Universidad de Cienfuegos "Carlos Rafael Rodríguez"
Facultad de Informática
Carrera de Ingeniería Informática

Trabajo de diploma para optar por el título de Ingeniero en Informática

Título:

"Estrategia para la introducción del Software Libre en la disciplina Ingeniería y Gestión de Software del Plan de Estudio D para la carrera de Ingeniería Informática"

Autora:
Yohanna Fernández Galbán

Tutor:
Ing. Mario Merino Escoto

Consultante:
Ing. Willian Feal Delgado

Cienfuegos, Cuba
Curso 2008 – 2009

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

Richard Stallman

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo la presente a los **25** días del mes de **junio** del **2009**.

Yohanna Fernández Galbán	Mario Merino Escoto
Nombre completo del autor	Nombre completo del tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

_____

Tutor : Ing. Mario Merino Escoto

_____	_____
Firma ICT	Firma Vicedecano

A mi pequeña familia
porque siempre están apoyandome y por su inmenso amor

Agradecimientos

A mi abuela, Hilda por su constante preocupación por mi bienestar y por su infinito amor.

A Yuviny, mi novio por estar siempre a mi lado apoyándome y soportando mis malas caras y mi inseguraridad al empezar cada capítulo de la tesis. Por su amor, su cariño y su comprensión.

A Marisol y Jesús, mis padres por su incondicional ayuda y amor. Los quiero mucho.

A mi tío Eddy por estar siempre ahí para cuando lo necesito y su cariño.

A mi hermanita Yisel por alegrarme en los momentos más estresantes.

A mi prima Janny por su ayuda a pesar de que está lejos, gracias por enviarme todos los correos con la documentación que me hacía falta para la tesis.

A mi tía Nena por siempre estar preocupada por mí, por siempre buscar la forma de mantenerme a la moda con las carteras, por ser la tía que más me quiere. Gracias.

A mi primo Javier por su apoyo que aunque extraño es sincero porque sé que él me quiere a su forma.

A tía Margarita por su apoyo y preocupación a pesar de no ser más que su sobrina política.

A Estrella por quererme como su propia hija y por apoyarme en en los momentos de inseguridad y agotamiento, por sus sabios consejos.

A Delvis, Dagmaris y Rosabel por ser tan buenos amigos, por considerame parte de su familia.

Agradecimientos

A Rosa mi amiga incondicional por estar siempre al pendiente de mi abuela cuando yo no estoy allá. También a su mamá Aleja por quereme como una hija.
Gracias a ambas.

A mis compañeros que también aportaron su granito de arena a pesar de no valorar los beneficios que aporta el Software Libre.

A mi tutor Mario por dedicarme pedacitos de su tiempo y apoyarme.

A mis profesores de la carrera por los conocimientos que me brindaron durante la carrera y por formarme como un profesional de la especialidad.

En fin les agradezco a todos a aquellos que me brindaron su ayuda y apoyo incondicional para que la investigación fuera un éxito,
a todos muchas gracias.

Resumen

El mundo abre las puertas al Software Libre como alternativa fiable para su desarrollo pues brinda amplias libertades mediante las cuales se rompe el yugo que nos ata a las grandes trasnacionales del software. Muchas son las instituciones tanto nacionales como internacionales en las diferentes ramas que están enfrascadas en la migración. Este proceso ha sido objeto de estudio de ilustres personalidades en los últimos años resultando riesgoso pues conduce por dos caminos, el éxito o el fracaso.

Los software utilizados en los distintos centros educacionales de nuestro país son generalmente de carácter propietario por lo que se deben buscar alternativas libres en aras de ahorrar y de emplear las ventajas que nos ofrece el Software Libre.

En el presente trabajo se propuso una estrategia de migración a Software Libre que cuenta con cuatro etapas, cinco flujos de trabajo y las tareas a realizar en cada uno de estos flujos, restringida a las asignaturas de la disciplina Ingeniería y Gestión de Software del Plan de Estudio D para la carrera de Ingeniería Informática de la Universidad de Cienfuegos. Además se realizó un análisis de las características de las herramientas libres que son equivalentes a las herramientas privativas empleadas actualmente en las asignaturas de la disciplina.

Índice de Contenido

Índice de ilustraciones

I*ntroducción*

El Software Libre es un fenómeno tecnológico de la era digital el cual cuenta con gran impacto y significación social por su carácter ético. Además se convierte en la alternativa socializadora y antimonopolista al software propietario.

Este modelo de software según se manifiesta en “Implicaciones éticas del Software Libre” (Dr. Mario González Arencibia s.d.) es un sistema de trabajo totalmente voluntario y cooperativo, donde los líderes de cada proyecto son elegidos democráticamente, de acuerdo a sus méritos, por los propios integrantes. Este modelo de trabajo elimina la clásica alienación del trabajador en el sistema capitalista, porque todos están directamente interesados en el producto final, el producto final les pertenece y además lo donan para toda la humanidad.

La actitud de estos desarrolladores libres, o sea, su generosidad y humanidad entre otros valores anteponen el reconocimiento moral e intelectual de su trabajo a los beneficios económicos que le pueda brindar la comercialización, todo esto es digno de admiración y se debe convertir en un ejemplo a seguir por los demás programadores del mundo pues ellos no reciben pago por concepto de licencia pero si reciben, como es justo, retribuciones por su trabajo. En realidad, funciona como un gran almacén a nivel mundial, en la cual todos depositan sus conocimientos en forma de programas convirtiéndose en un gran repositorio del que se retira lo que necesitan para ellos mismos o para dar servicios a sus clientes.

El Software Libre facilita la soberanía tecnológica de todos los países, principalmente de países subdesarrollados, favoreciendo el proceso de identidad nacional a diferencia del software propietario que profundiza y fortalece la transculturación globalizante y conlleva a la pérdida de autonomía. Además puede llegar a convertirse en una forma de lucha de

clases contra la propiedad privada en el entorno digital.

Por las libertades que brinda como modificar y redistribuir copias, modificarlo y su mínimo costo , este movimiento está llamado a disminuir la brecha digital entre informáticos con grandes recursos e informáticos sin recursos.

La formación masiva de recursos humanos en “Implicaciones éticas del Software Libre” (Dr. Mario González Arencibia s.d.) se describe como una ventaja para la organización del desarrollo tecnológico de los países del tercer mundo, verificada a partir de la multiplicación de desarrolladores de software, a diferencia del software propietario que multiplica los piratas y reduce los desarrolladores. La piratería del software propietario incide negativamente en el desarrollo del Software Libre por el desperdicio de talentos tecnológicos que se enfrascan en esa actividad, pudiendo ser utilizados en el desarrollo de programas informáticos sobre plataformas libres.

Si el software propietario favorece los intereses de las grandes multinacionales de la información, el Software Libre propicia el desarrollo de la pequeña y mediana empresa productora de servicios informáticos así como de los centros educacionales, y es en ese sentido una actividad estratégica para el progreso de los países subdesarrollados.

Los grandes monopolios internacionales de la informática y las comuniciones se mantienen informados del avance del Software Libre por los peligros que les representa. Este último está llamado a convertirse en un factor potencial de integración latinoamericana en los marcos del nuevo escenario social y político de la región, donde la interacción ciencia, tecnología y el desarrollo social constituye un elemento significativo del proceso global que se desarrolla en las diferentes alternativas integracionistas, primordialmente el ALBA.

Muchos han sido los países que ven en el Software Libre la alternativas a sus problemas y se han enfrascado en el proceso de la migración para ello han confeccionado guías y estrategias.

Cuba un país bloqueado, no está exento de este proceso de migración pues se encuentra a merced de la empresa norteamericana Microsoft, que cuenta con la capacidad legal de pedirle a nuestro país que no continúe utilizando un sistema operativo de su propiedad basados en las leyes de propiedad industrial por las cuales nosotros también nos regimos y además el Software Libre brinda grandes facilidades.

La necesidad de llevar a cabo la migración a todas las instituciones del país conlleva a que el Ministerio de Educación Superior (MES) se trace una estrategia la cual sirve como base para todas las universidades pertenecientes al mismo y también se involucren en este gran proceso.

Un ejemplo de esto es la Universidad de Cienfuegos “Carlos Rafael Rodríguez” en la cual se realizó el Consejo de Carrera del curso 2007-2008 de Ingeniería Informática donde se llevó a cabo un análisis de los planteamientos del MES para el nuevo plan de estudio D dentro de ellos el uso del Software Libre en las disciplinas y además el proceso de migración a este en que se encuentra enfrascado el país . Para dar cumplimiento a lo planteado se propuso en este consejo de carrera el diseño de estrategias para llevar a cabo la migración a Software Libre de las disciplinas de este nuevo plan de estudio así como la búsqueda de herramientas libres para el desarrollo de las diferentes asignaturas de cada una de las disciplinas.

La disciplina Ingeniería y Gestión de Software partiendo de que es una de las integradoras de la carrera, en la cual se abarcan aspectos esenciales para la formación de profesionales informáticos, además cuenta con gran cantidad de horas clases y de asignaturas se puede considerar como una guía para las demás disciplinas del Plan de

Estudio en todo el proceso de la migración.

Partiendo de lo expuesto anteriormente se identifica como ***Problema a resolver*** la necesidad de una estrategia que permita la migración a Software Libre, de las asignaturas de la disciplina de Ingeniería y Gestión de Software, en la carrera de Ingeniería Informática de la Universidad de Cienfuegos.

Se plantea como ***Objetivo General*** de este trabajo diseñar una estrategia para la introducción del Software Libre en el Plan de Estudio D para la carrera de Ingeniería Informática que permita una transición correcta de la filosofía de Software Libre en las asignaturas de la disciplina de Ingeniería y Gestión de Software manteniendo la calidad del Proceso Docente Educativo.

Para dar cumplimiento al Objetivo General enunciado se plantean como ***Objetivos Específicos:***

- Identificar los objetivos de las asignaturas de la disciplina “Ingeniería y Gestión de Software” en la carrera de Ingeniería Informática.
- Analizar otras experiencias de migración a Software Libre y las tecnologías aplicadas.
- Diseñar las etapas y flujos de trabajo de la estrategia para la migración a Software Libre de la disciplina Ingeniería y Gestión de Software del Plan de Estudio D para la carrera de Ingeniería Informática.

Para lograr los objetivos se proponen las siguientes ***Tareas:***

- Revisión del plan de estudio D de la carrera de Ingeniería Informática.
- Entrevista con el jefe de la disciplina “Ingeniería y Gestión de Software”.
- Estudio de la aplicación del Software Libre tanto en la escala mundial como nacional.

- Estudio de otras estrategias y guías de migración a Software Libre a nivel internacional y nacional.
- Revisión de tecnologías y herramientas de Software Libre.
- Elaboración de una estrategia para la introducción del Software Libre en la disciplina de Ingeniería y Gestión de Software del Plan de Estudio D para la carrera de ingeniería Informática.

Se considera como ***Objeto de estudio*** el proceso Docente Educativo en la carrera de Ingeniería Informática y como ***Campo de acción*** la estrategia de migración a Software Libre en el proceso Docente Educativo de la disciplina Ingeniería y Gestión de Software de la carrera de Ingeniería Informática.

Se propone como ***Idea a defender*** la aplicación de la estrategia para la introducción del Software Libre en la disciplina Ingeniería y Gestión de Software del Plan de Estudio D permitirá incorporar a la formación del estudiante de Ingeniería Informática de la Universidad de Cienfuegos el manejo de las herramientas libres.

Este trabajo se encuentra estructurado en 3 capítulos:

Capítulo I:" Fundamentación Teórica" en este capítulo se aborda todo lo referente a la historia,licencias del Software Libre así como la necesidad de migrar y su vínculo con la educación.

Capítulo II:"Análisis y propuesta de herramientas " en este capítulo se caracteriza tanto el Software Libre como el Propietario. También se da a conocer las características de las herramientas propietarias actualmente utilizadas y de algunas alternativas libres y se proponen las más factibles.

Capítulo III:"Propuesta de la Estrategia " en este capítulo se realiza un análisis de guías internacionales como: las Directrices IDA europeas, Plan Nacional de migración de venezolano y la guía peruana,además de la guía cubana dando lugar a la estrategia particular de migración para la disciplina Ingeniería y Gestión de Software.

Capítulo I: "Fundamentación Teórica"

1.1. Introducción

En este capítulo, se hace un acercamiento a la historia del Software Libre así como se caracterizan sus licencias, además se explica la necesidad de migrar. Incluye una panorámica sobre el estado del Software Libre y la educación tanto en la escala internacional como en la nacional. Describe lo referente a las características y objetivos expuestos en el nuevo Plan de Estudio D para la carrera de Ingeniería Informática y especificando los de la disciplina Ingeniería y Gestión de Software.

1.2. Historia del Software Libre.

El software se considera como el gran intermediario entre la información y la inteligencia humana. Se pueden clasificar de varias formas tales como su precio, apertura de su código fuente, protección, legalidad y concepción social.

No se puede olvidar como muestra de software cuando a finales de la década de los 60 principios de los 70 tuvo su aparición el sistema operativo UNIX creado en los laboratorios Bell de la AT&T por Ken Thompson a partir de un experimento empresarial para ayudar a controlar la nueva generación de redes telefónicas, que estaban convirtiéndose en computadoras especializadas. Distribuyendo más tarde licencias para su uso a las universidades a un bajo costo contribuyendo a las innovaciones.

Como se plantea en "Implicaciones Sociales del Software Libre, Universidad de las Ciencias Informáticas, La Habana." (Orlando Cárdenas Fernández s.d.) a partir de la aparición de un software portable en los primeros años de la década de los 80 diseñado por los hackers del staff para el laboratorio de inteligencia artificial del MIT (Instituto Tecnológico de Massachussets) comienzan los fabricantes a detener las distribuciones de los códigos fuentes restringiendo las copias y redistribuciones de estos, partiendo del Derecho de Autor (Copyright).

Se considera que la producción de software tuvo en sus inicios un carácter totalmente solidario y de transparencia en la información proporcionando a todos los programadores su código fuente, o sea, tienen acceso abierto pero la causa de su conversión en software privado de código fuente cerrado y secreto estuvo dada por la comercialización con fines de lucro. A raíz de esto surge una nueva alternativa para una mejor unión del conocimiento informático la que conocemos como Software Libre.

El Software Libre es aquél que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios, sin embargo, no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

En "Implicaciones Sociales del Software Libre, Universidad de las Ciencias Informáticas, La Habana." (Orlando Cárdenas Fernández s.d.) se expone que el 27 de septiembre de 1983 Stallman, principal figura de este movimiento anuncia la creación de un sistema operativo totalmente libre que toma como punto de partida a UNIX y que a su vez era compatible con él, nombrándolo GNU (GNU no es Unix). Además en 1985 crea la Fundación para el Software Libre (FSF), una organización sin ánimos de beneficio que coordinar los esfuerzos de la comunidad y proveería de una infraestructura financiera, logística y legal al proyecto. Esto se unió a la publicación del "Manifiesto del GNU", documento en el que se declaraban los objetivos, intenciones y motivaciones de crear un sistema operativo libre. Ya en 1989 aparece el Copyleft, un mecanismo para proteger los derechos de modificación y redistribución del Software Libre. Este concepto está plasmado en la Licencia Pública General (GNU- GPL), creada por la Fundación.

Además en 1991 el estudiante finés Linus Torvald con el objetivo de desarrollar una versión mejorada de MINIX (sistema operativo de la IBM) para el procesador 386 sobre

todo la conmutación de tareas en modo protegido liberó el kernel Linux *(Anexo III)* bajo los términos de una de las licencias del GNU, o sea, la licencia GPL v. 2.0, lográndose un sistema operativo completo y libre llamado GNU/Linux.

Pero también para que un software sea libre debe garantizar las siguientes **cuatro libertades** planteadas en "La promoción del uso del Software Libre por parte de las universidades" (Ana María Delgado García & Rafael Oliver Cuello s.d.) :

- Libertad 0 de ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.).
- Libertad 1 de estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente).
- Libertad 2 de copiar el programa.
- Libertad 3 de mejorar dicho programa y hacer públicas las mejoras, de forma que se beneficie toda la comunidad**.**

Las libertades que brinda el Software Libre han despertado gran interés en la población, las empresas y otras instituciones o países, los cuales planifican y buscan las mejores vías para migrar a Software Libre.

A partir de la definición anterior de las libertades, la contraposición de Software Libre es software propietario y no software comercial como se expresa en "Acceso abierto y Software Libre: premisas para la independencia tecnológica " (Raúl G Torricella Morales & Francisco Lee Tenorio). El software propietario no es ni libre ni semilibre; sus productores prohíben expresamente su redistribución y modificación y generalmente lo distribuyen en forma comercial, aunque no siempre. El software comercial se desarrolla por una entidad que tiene como objetivo obtener ganancias a partir de su venta. La mayoría del software propietario es comercial; sin embargo, el Software Libre puede ser comercial, lo que no contradice el espíritu del movimiento.

Para una mejor comprensión de lo que realmente es la definición de Software Libre

resulta de gran utilidad el mapa conceptual elaborado por *René Mérou (Anexo I)*, que muestra de forma exhaustiva las definiciones, ventajas y algunos ejemplos de Software Libre .

1.2.1. El Software Libre en el ámbito político, económico y tecnológico

El empleo del Software Libre es sustentable en Cuba dado por los beneficios que brinda con respecto a los Software Propietario. Su aplicación como plataforma informática de trabajo adquiere una relevante significación que puede verse en ámbitos diferentes, como se expresa en la "Guía Cubana para la Migración a Software Libre" (Ramón Paumier Samón & Yoandy Pérez Villazón s.d.) :

Político

Desde un primer punto de vista, representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación. Es válido recordar que actualmente Cuba se encuentra a merced de la empresa norteamericana Microsoft, que tiene la capacidad legal de reclamar a Cuba que no siga utilizando un sistema operativo de su propiedad, basado en leyes de propiedad industrial por las cuales también Cuba se rige; esto provocaría una interrupción inmediata del programa de informatización de la sociedad que como parte de la batalla de ideas que está desarrollando el país, además pudiera implementarse una campaña de descrédito a la isla, abogando el uso de la piratería informática por parte de las instituciones estatales cubanas.

Desde un segundo punto de vista, el Software Libre representa la alternativa para los países pobres, y es por concepción, propiedad social, si se tiene en cuenta que una vez que comienza a circular rápidamente se encuentra disponible para todos los interesados sin costo alguno o en su defecto a muy bajo costo.

En tercer lugar, es desarrollado de forma colectiva y cooperativa, tanto en su creación

como en su desarrollo, cuantitativa y cualitativamente, mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad.

La posibilidad de usar, copiar, estudiar, modificar y redistribuir libremente el software como un bien social, que brinda esta plataforma, cumple los preceptos enunciados por la sociedad socialista cubana y está acorde con el tipo de economía socialista, donde el valor social está por encima de la ganancia.

Económico

Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

La adquisición de cualquiera de sus distribuciones puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por cuanto no hay que pagar absolutamente nada por su utilización (no requiere de licencia de uso, las cuales son generalmente muy caras), distribución y/o modificación.

El uso del Software Libre desarrollado con Estándares Abiertos, fortalecerá la industria del software nacional, aumentando y fortaleciendo sus capacidades. Facilitará la reducción de la brecha social y tecnológica en el menor tiempo y costo posibles. Su uso en la Institución Pública y en los servicios públicos, facilitará la interoperabilidad de los sistemas de información del Estado, contribuyendo a dar respuestas rápidas y oportunas a los ciudadanos, mejorando la gobernabilidad.

Tecnológico

Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además de la corrección de sus errores de programación y la obtención de las actualizaciones y las nuevas versiones.

Todas las mejoras que se realicen no tienen restricciones. De este modo, cualquier otra administración, empresa, institución o organismo se puede beneficiar de las mejoras introducidas.

Se fomenta la innovación tecnológica del país. Al disponer del código fuente de la aplicación, podamos realizar el desarrollo de mejoras, en vez de encargarlas a empresas de otros países que trabajan con sistemas de licencia propietaria. De este modo, contribuimos a la formación de profesionales en nuevas tecnologías y al desarrollo local bajo nuestros propios planes estratégicos.

Proceso de corrección de errores muy dinámico. Los usuarios del programa de todo del mundo, gracias a que disponen de su código fuente, pueden detectar los posibles errores, corregirlos, y contribuir con sus mejoras.

Más dificultad para introducir código malicioso, espía o de control remoto. Debido a que el código es revisado por muchos usuarios que pueden detectar posibles puertas traseras.
Por las razones detalladas anteriormente, el uso del Software Libre es, sin lugar a dudas, sustentable para Cuba y en ese sentido desde octubre del 2002, se puso en marcha una estrategia para alcanzar la independencia en el terreno del software, garantizando la seguridad informática y por sobre todas las cosas afianzando el uso de los principios del Software Libre, pues la negación de dichos preceptos constituiría el rechazo de los principios del socialismo y el comunismo.

1.2.2. Licencias en el Software Libre

Todo autor de software tiene su cesión de derechos que se puntualiza por un contrato denominado licencia. Pero estas se pueden ser empleadas para programas privativos los cuales generalmente se distribuyen por medio de licencias de uso no exclusivo, o sea, se aceptan automáticamente al abrir o instalar el producto. Estas licencias estipulan lo que

los usuarios pueden hacer con el programa en cuanto a uso, redistribución, modificación, etc., y en qué condiciones.

Pero por otra parte se encuentran las que se emplean para programas libres que no restringen precisamente el uso, la redistribución y la modificación, aunque pueden especificar condiciones a satisfacer esencialmente en caso de que se quiera redistribuir el programa. Por ejemplo, pueden exigir que se respeten las indicaciones de autoría o que se incluya el código fuente si se quiere redistribuir el programa preparado para ejecutar.

Existen un gran número de licencias para programas libres, pero generalmente se emplean pocas de ellas.

Estas licencias libres se pueden dividir en dos grandes grupos o familias, las permisivas y las robustas.

No se puede obviar expuesto en "Software Libre: licencias y propiedad intelectual" (Jesús M. González Barahona, & Joaquín Seoane Pascual 2004) donde expresan que las licencias robustas son aquellas llamadas habitualmente en inglés *copyleft,* estas tratan de garantizar las libertades que otorga el autor no sólo a quien recibe el programa directamente de él, sino también a quienes los reciben, más adelante en la cadena de distribución, de otras terceras partes. La primera de este tipo de licencias fue la GNU GPL.

La licencia **GNU GPL** fue diseñada por la FSF (Free Software Foundation) con el objetivo de que todos sus software tuvieran dicha licencias pero se extendió mucho más hasta adentrarse en los proyectos cumbres del mundo del Software Libre, tal como el núcleo de Linux. Garantiza los derechos de los usuarios permite la redistribución binaria y la del código fuente así como realizar modificaciones sin restricciones; pero sin embargo, sólo se puede redistribuir código licenciado bajo GPL de forma integrada con otro código si éste tiene una licencia compatible, conocido esto como efecto viral de la GPL, ya que código publicado una vez con esas condiciones nunca puede cambiar de condiciones.

Otros ejemplos de licencias robustas señalados en "Software Libre: licencias y propiedad

intelectual" (Jesús M. González Barahona, & Joaquín Seoane Pascual 2004) son:

- La **Licencia Pública General Menor del proyecto GNU** (comúnmente conocida por sus iniciales en inglés **LGPL**) es la otra licencia de la Free Software Foundation. Pensada en sus inicios para su uso en bibliotecas (la L en sus comienzos venía de *library*, biblioteca) fue modificada recientemente para ser considerada la hermana menor de la GPL. La LGPL permite el uso de programas libres con software propietario. El programa en sí se redistribuye como si estuviera bajo la licencia GPL, pero se permite la integración con cualquier otro software sin prácticamente limitaciones.
- **Licencia de Sleepycat** es la que la empresa Sleepycat distribuye sus programas, esta obliga a ciertas condiciones siempre que se redistribuye el programa o trabajos derivados del programa. En particular, exige ofrecer el código fuente (incluidas las modificaciones, si se trata de un trabajo derivado), así como a que la redistribución imponga al receptor las mismas condiciones. Aunque es mucho más corta que la GNU GPL, es muy similar a ella en sus efectos principales.
- **eCos License 2.0** es la licencia bajo la que se distribuye eCos, un sistema operativo de tiempo real. Es una modificación de la GNU GPL que no considera que el código que se enlace con programas protegidos por ella queden sujetos a las cláusulas de la GNU GPL si se redistribuyen. Desde este punto de vista, sus efectos son similares a los de la GNU LGPL.
- **Affero General Public License** importante transformación de la GNU GPL que considera el caso de los programas que ofrecen servicios vía Web o vía redes de ordenadores. Como el uso del programa no implica haberlo recibido mediante una redistribución, aunque el mismo esté licenciado bajo la GNU GPL, alguien puede modificarlo y ofrecer un servicio en la red usándolo, sin redistribuirlo de ninguna forma, y por tanto sin estar obligado a distribuir el código fuente. La Affero GPL tiene una cláusula que obliga que, si el programa tiene un medio para proporcionar su código fuente vía Web a quien lo use, no se pueda desactivar esa característica.
- **IBM Public License Version 1.0** es una licencia que permite la redistribución

binaria de trabajos derivados sólo si se tiene algún mecanismo para que quien reciba el programa pueda recibir su código fuente. La redistribución de este código se ha de hacer bajo la misma licencia. Además, esta licencia obliga al que redistribuye el programa con modificaciones a licenciar automática y gratuitamente las patentes que puedan afectar a esas modificaciones, y a su vez sean propiedad del redistribuidor, a quien reciba el programa.

- **Mozilla Public License 1.1** Ejemplo de licencia libre con origen en una empresa. Es una evolución de la primera licencia libre que tuvo el Netscape Navigator, y en su momento fue muy importante por ser la primera vez que una empresa muy conocida decidió distribuir un programa bajo su propia licencia libre.

Por otra parte se encuentra las licencias permisivas o liberales que no le imponen al usuario que reciba el software prácticamente ninguna condición y estos les es permitido usarlos, modificarlos y redistribuirlos. Pero estas en ocasiones pueden considerarse como un alto nivel de despreocupación o la mayor garantía de libertad a los usuarios, también permite las redistribuciones bajo licencias privativas aquel software publicado con licencias permisivas.

Como se expresa en "Software Libre: licencias y propiedad intelectual" (Jesús M. González Barahona, & Joaquín Seoane Pascual 2004) entre estas licencias encontramos las llamadas **BSD** (Berkeley Software Distribution) cuya única obligación que exige dar crédito a los autores, mientras que permite tanto la redistribución binaria y la de los fuentes, aunque no obliga a ninguna de las dos en ningún caso. Asimismo se da permiso para realizar modificaciones y ser integrada con otros programas casi sin restricciones.

Además existen otras con elementos similares a las BSD, algunas de la gran familia son:

- **Licencia de XWindow versión 11 (X11).** es la licencia usada para la distribución del sistema XWindow, el sistema de ventanas más ampliamente usado en el mundo Unix, y también en entornos GNU/Linux.

- **Zope Public License 2.0** esta licencia (habitualmente llamada *"ZPL"*) es usada para la distribución de Zope (un servidor de aplicaciones) y otros productos relacionados. Esta licencia tiene el interesante detalle de prohibir expresamente el uso de marcas registradas por Zope Corporation.
- **Licencia de Apache**. Licencia bajo al que se distribuyen la mayor parte de los programas producidos por el proyecto Apache.

Los programas libres también pueden carecer de licencias específicas, es decir, los autores los pueden declarar de dominio común o público perdiendo estos todos sus derechos sobre su producto.

1.3. Necesidad de Migrar a Software Libre.

Siempre las grandes compañías han empleado para el control de sus sistemas software privados teniendo que pagar por su uso, eso sin contar que no tienen acceso a su código fuente. Con esto están expuestos a cualquier manipulación de sus informaciones por parte de los programadores de los mismos.

Existen varios ejemplos de problemas presentados por el empleo de estos tipos de software de los cuales no se conoce su funcionamiento interno, este inconveniente contribuyó a un paro prácticamente general con una duración de siete días a una de las empresas más importantes de Venezuela, PDVSA (compañía petrolera estatal) donde su administración se mostró en desacuerdo con las políticas del Presidente Chávez, que incluían redirigir los beneficios de las élites de la petrolera estatal a programas sociales (como planes de salud y de alfabetización).

Según se comenta en "División de Formación General de la Facultad de Ciencias, La Universidad del Zulia, Venezuela" (Mg. Jorge Hinestroza 2003) la participación de la empresa INTESA en el colapso provocado a PDVSA fue determinante, considerando que esta empresa desde 1997 controlaba toda la infraestructura, facilidades, equipos, datos financieros, geológico, técnica, presupuestaria y de negocios de la empresa y, además, el personal de soporte tecnológico de información de PDVSA, en el cual reposa el más decisivo control de la producción, refinación y distribución del crudo y sus derivados.

INTESA ejerció su poder de control computarizado para paralizar la carga, descarga y almacenamiento de crudo en los diferentes terminales de embarque, así como para detener el funcionamiento de la mayoría de las estaciones de flujo, plantas compresoras, plantas de procesamiento, llenadores de combustible, tuberías automatizadas, etc., asegurándose que la manipulación de las redes informáticas fuera posible solamente por parte de los poseedores de las claves secretas de acceso al sistema, todos empleados de INTESA sumados al "paro", quienes secuestraron el sistema mediante un acceso clandestino preparado cuidadosamente con anticipación.

Además en "División de Formación General de la Facultad de Ciencias, La Universidad del Zulia, Venezuela" (Mg. Jorge Hinestroza 2003) se resalta que este contundente sabotaje informático implicó la utilización de módems ocultos en el interior de paredes y escritorios para operar con acceso remoto, vía telefónica. Cabe anotar el uso de la red INTERNET para lograr acceso a la intranet de la corporación, en donde tanto las comunicaciones satelitales como de microondas y radio son parte del sistema. También se concentraron en destruir las bases de datos contentivas de las operaciones rutinarias y evitar la identificación de operadores en distintas instalaciones y maquinarias; fortuitamente, suprimir los sistemas operativos de los servidores, desconfigurar los routers, y en último lugar, desmantelar todo el sistema de información y control a distancia de las operaciones automatizadas.

Este problema sirvió de modelo para que las grandes compañías no confíen en los programas privativos y de esta forma evitar estar a merced de las empresas productoras de estos software; es por ello que se ha hecho más atractiva la idea de emplear Software Libre en el control y funcionamiento de las importantes empresas pues con estos la utilización de maniobras perversas ya se hace más difícil.

Otro problema que surge a raíz del uso de software privativo es el empleo de software especializados en contar automáticamente los votos en las urnas electorales , uno de estos es el comentado en "El Lado Oscuro del Voto Electrónico" (Beatriz Busaniche 2008)

donde expone que el 5 de febrero de 2008, en las elecciones primarias del Estado de Nueva Jersey, los trabajadores electorales detectaron que el total de votos registrados en algunas urnas de la empresa Sequoia Voting Systems no coincidía con la suma de los votos para cada candidato.
Este estado llamó auditores independientes para evaluar las urnas. El profesor Edward Felten, de la Universidad de Princeton, a cargo de la auditoría, recibió una comunicación urgente de Sequoia Voting Systems en la que se lo intimaba a no realizar el estudio. Si evaluaba las urnas sería llevado a juicio por violación de la propiedad intelectual de la Empresa Sequoia ya que la firma sólo autoriza auditorías bajo su supervisión.
Demostrando esto que por términos de licencia de software, Nueva Jersey tiene prohibido hacer auditoría independiente sobre su sistema electoral.

Igualmente en "El Lado Oscuro del Voto Electrónico" (Beatriz Busaniche 2008) se hace referencia al documental titulado Hacking Democracy realizado por la cadena HBO y que revela en forma contundente las grietas de seguridad de las urnas usadas y muestra una larga serie de irregularidades. Al escándalo desatado por las preferencias políticas del CEO de Diebold se suman problemas en las urnas y dificultades para emitir el voto en distritos con alta concurrencia de votantes afroamericanos (distritos de mayoría demócrata) o las vulnerabilidades del sistema GEMS, la tabuladora central que se ocupa de sumar los votos emitidos.
En la Universidad de Princeton, el Center for Information Technology Policy dirigido por Edward Felten, profesor de Ciencias de la Computación y Asuntos Públicos, realizó una serie de pruebas de seguridad sobre las urnas Diebold AccuVote Systems.
Planteando que los expertos pudieron manipularlas en minutos sin dejar huella del fraude. Inyectaron un virus en una de ellas, lograron que se reproduzca, modifique los resultados y se autoelimine para que no queden pruebas. Ningún trabajador electoral encargado de auditar el proceso tendría capacidad alguna para detectar esta manipulación.

Estos son algunos de los ejemplos de manipulación que puede existir por parte de los

expertos y personal autorizados a los software de los cuales no se tiene acceso su código fuente, o sea, son los riesgos que se corren por usar este tipo de sistema.

1.4. El Software Libre en la educación.

Existen muchos motivos por los cuales es importante usar Software Libre en la Educación. De acuerdo con la visión de Richard Stallman expresada en "Guía Práctica sobre Software Libre, su selección y aplicación local en América Latina y el Caribre"(Fernando Da Rosa & Federico Heinz 2007) , se debe tener en cuenta al estudiante como un ser integral y enseñarle la importancia de la libertad, guiarlo en el sentido de saber hacer uso de esa libertad. Hacerle entender al estudiante que debe tener la posibilidad de estudiar hasta donde su curiosidad lo lleve, que debe poder profundizar en el conocimiento, y que además existe una responsabilidad inherente a ello, un Software Libre es tan bueno como la responsabilidad con que encararan sus realizadores su trabajo alrededor de él.

Además la educación como en otras ramas también ha tomado la decisión de la migración a Software Libre debido a sus grandes perspectivas desde el ámbito económico como el social, es decir, el costo en la compra de licencias disminuiría y el índice de piratería desaparecería considerablemente.

1.4.1. El Software Libre y la educación en el contexto internacional

El *Gobierno de Brasil* como se señala en "Más de 50 millones de estudiantes brasileños utilizarán tecnologías abiertas a partir de 2009" (Software Libre Maxorata 2008) ha dado un paso más en favor de las tecnologías de código abierto y ha decidido implementar el sistema operativo libre GNU/Linux en las escuelas públicas. La medida supone que 52 millones de estudiantes utilizarán Software Libre a partir del año 2009.

La decisión del ejecutivo brasileño se enmarca en el proyecto "Proinfo", el programa nacional de alfabetización informática, en el que colabora el Ministerio de Educación. Este departamento proporcionará la infraestructura necesaria de hardware, terminales y

conectividad con el Software Libre que se utilizará en todas las escuelas.

Uno de los casos más recientes que ha decidido apostar de manera fuerte por el uso y la promoción de sistemas libres como se plantea en "Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI)" (Ramón Paumier Samón 2007) es la *institución regional de Extremadura*. La acción más emblemática hasta el momento ha sido la de elaborar una distribución de GNU/LINUX llamada LinEx, poniendo especial énfasis en la facilidad de instalación, y que cuenta con todas las herramientas que cualquier usuario medio utiliza habitualmente. De esta forma los gastos en licencias de ordenadores para la enseñanza secundaria, etc. se reducirían considerablemente.

Argentina cuenta con un proyecto educativo cooperativo (Gleducar,) como se expresa en "Argentina: Software Libre y educación en el Año de la Enseñanza de las Ciencias - Definición - MasterMagazine"(Marcos Guglielmetti 2008) que tiene por fin adecuar las aulas del país a las Nuevas Tecnologías de la Comunicación y la Información, reunido en forma de Asociación Civil sin fines de lucro.

Según este proyecto la incursión del Software Libre en realidad debería profundizar los ideales originarios de educación universal, obligatoria, gratuita y laica pues con software privativo como por ejemplo Microsoft Windows un estudiante no podría obtener una copia sin pagar por la licencia a esta empresa estadounidense, por lo cual la educación dejaría de ser gratuita (cabe acotar que en varios establecimientos del país, ya sea primarios, secundarios, terciarios y universitarios, se utiliza software de Microsoft, por el cual se pagan grandes sumas de dinero).

La Universidad de Cádiz tras evaluar los beneficios que brindaría el uso de Software Libre en la misma, su Consejo de Gobierno aprobó en el año 2004 una declaración institucional como apoyo al Software Libre, así como la instauración de la Oficina de Software Libre de

la Universidad de Cádiz (OSLUCA).

Esta declaración marcó un importante paso, como se expresara en "El Software Libre en la educación – Experiencias"(Nsmlinux 2006) pues ha situando a la universidad como abanderada en la acogida de estrategias de la información basadas en Software Libre en su comunidad, a la par con la Junta de Andalucía y con muchas más universidades, asociaciones e instituciones públicas españolas. La OSLUCA coordinará los esfuerzos de la comunidad universitaria para promover el conocimiento y el Software Libre.

Esta oficina (OSLUCA) es fruto de un ambicioso plan de intenciones tanto a corto como a largo plazo, en el que se apuesta indudablemente por el uso de Software Libre, y en general por la libre difusión del conocimiento para potenciar el papel de la universidad como fuente de educación, innovación y tecnología.

Igualmente la Universidad de las Palmas de Gran Canarias (ULPGC) apuesta por las nuevas tecnologías y ha decidido crear la Oficina del Software Libre. Dicha oficina esta destinada a fomentar y divulgar el uso de Software Libre. También se espera, en un futuro, poder ofrecer soporte a todo aquel que lo solicite, ya sea mediante cursos, documentación u otros medios. Esta oficina surge a raíz de la idea de convertir esta universidad en un ejemplo de reutilización de los propios recursos, de un mejor aprovechamiento de los fondos públicos y de una apuesta por la innovación tecnológica y la pluralidad.

En "El Software Libre en la educación – Experiencias"(Nsmlinux 2006) se hace referencia a el sistema de migración de las aulas en la ULPGC , esta migración se planificó para el curso académico 2003/2004 con la instalación de Software Libre que funcione en el sistema operativo MS Windows, de las aulas existentes. Se prevee que durante el primer cuatrimestre y parte del segundo convivan en muchas de estas aulas, aplicaciones libres y no libres, para que el cambio no sea traumático para los usuarios. En una segunda fase se eliminarán todas aquellas aplicaciones informáticas propietarias cuyas funcionalidades

queden cubiertas por Software Libre. Y el objetivo de la última fase es mantener las mismas aplicaciones libres que ya estén usando los usuarios, pero se cambiará el sistema operativo MS Windows, por uno libre como es GNU/Linux.

Según lo planteado en "La promoción del uso del Software Libre por parte de las universidades" (Ana María Delgado García & Rafael Oliver Cuello s.d.) el Departamento de Educación y Universidades de la Generalidad de Catalunya cuenta con un portal denominado *zonaClic, el cual* acoge el proyecto de *Software Libre JClic*, formado por un conjunto de herramientas informáticas que permiten a los educadores crear aplicaciones interactivas para los estudiantes, así como un proyecto de cooperación entre escuelas de diversos países y comunidades que ha permitido la creación de una extensa base de intercambio de materiales educativos.
A la par se debe destacar el proyecto CAMPUS, que fue promovido por la Secretaría de Telecomunicaciones y Sociedad de la Información (STSI) de la Generalidad de Catalunya, que nace de la voluntad de las universidades catalanas de poder disponer de un campus virtual basado en *Software Libre* y bajo licencia libre GPL (*General Public License*), que permite impartir enseñanza superior exclusivamente en línea, así como de forma semipresencial.

La Universidad Oberta de Catalunya se responsabiliza de la coordinación y liderazgo tecnológico del proyecto CAMPUS, mientras que las otras universidades catalanas se hacen cargo de otras partes del desarrollo, aportando sus conocimientos y experiencia.

1.4.2. El Software Libre y la educación en el contexto nacional

Una de las entidades que contribuye de alguna manera con la estrategia de migración del país es el organismo rector del Sistema de Educación Superior en Cuba, o sea, el *Ministerio de Educación Superior (MES)*, el cual se trazó como se expresa en "Estrategia Maestra de Informatización" (Ministerio de Educación Superior: Dirección de Informatización 2008) un programa de migración al Software Libre que tiene como

objetivo general: Desarrollar y cumplir con la Estrategia para la Migración al Software Libre aprobadas por los Consejos de Dirección del MES y de los Centros el pasado curso 2007/08.
Este plan cuenta con dos etapas una de migración de los servicios y otra de migración de las aplicaciones, la primera se estructura por cuatro fases: diagnóstico de recursos, formación y capacitación del personal, preparación para la migración y la migración.

La *Universidad de Ciencias Informáticas* no se quedó al margen de este proceso para ello ha hecho un amplio estudio y según lo planteado en "Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI)" (Ramón Paumier Samón 2007) para una buena estrategia, se utilizará una nomenclatura que divide el proceso de migración en 3 grandes grupos fundamentales: Preparación, Migración y Consolidación, considerando cada etapa como sigue:

- **Preparación**: Etapa previa al desarrollo del proceso cuyo contenido dependerá de la guía en estudio.
- **Migración**: Etapa en la que se acomete como tal el proceso de migración y cuyo contenido dependerá de la guía en estudio.
- **Consolidación**: Etapa post-migración cuyo contenido dependerá de la guía en estudio.

Fue aprobado como expresa "Estrategia Maestra de Informatización" (Ministerio de Educación Superior: Dirección de Informatización 2008) por el consejo de dirección de la *Universidad de La Habana* un plan de migración a Software Libre. Este plan ubica a la más antigua y prestigiosa casa de altos estudios cubana en la vanguardia de este proceso.

Dentro de los aspectos propuestos en ese plan encontramos: Realizar el análisis en las distintas áreas del centro de las necesidades de software e identificación de programas

sustitutos, y donde no deben o no pueden producirse sustituciones. También comenzar la migración de los sistemas operativos en aquellas computadoras donde se haya determinado que todos los software necesarios en ella tienen un equivalente en GNU\Linux.

Otra institución que no se queda rezagada con respecto a esta migración es el Ministerio de Educación (MINED) que para llevar a cabo este proceso parte de las etapas antes expuestas en "Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI)"(Ramón Paumier Samón 2007) y de esa forma alcanzar su **objetivo** de definir una metodología de migración a Software Libre para los centros educacionales del MINED, proporcionando una mayor integración en el sector educativo con los países del ALBA, América Latina y el mundo en general como fue planteado en "Metodología para la Migración a Software Libre para los Centros Educacionales del Ministerio de Educación (MINED)" (Ing. Eduardo Manuel Macías Sotolongo s.d.).

1.5. Plan de Estudio D

El Ministerio de Educación Superior (MES) es el encargado de la confección de planes de estudio para las universidades del país con las especificaciones para cada tipo de especialidad. Un ejemplo de ello es el nuevo plan de estudio D que se basa en las transformaciones ocurridas en el país.

Como se señala en "Plan de Estudio D INGENIERÍA INFORMÁTICA Presencial" (Ministerio de Educación Superior 2007) sus diferencias fundamentales están en la intensidad, semestres y modalidad en que se cursan las asignaturas; así como en la forma en que se desarrolla el componente laboral e investigativo. Además se añaden aspectos propios como el cambio en los niveles de abstracción al manejar recursos y herramientas informáticas, la diversidad de enfoques locales en cuanto a tecnologías, lenguajes de programación y áreas de aplicación, el surgimiento de la carrera en todas las provincias, y la experiencia de algunas universidades en la realización de ajustes a los planes de estudio a grupos de estudiantes relacionados con colectivos de desarrollo o

investigación en cuanto a los enfoques de esencialidad, presencialidad y flexibilidad exigidos.

Al mismo tiempo la organización del trabajo docente-educativo debe fundamentarse en el trabajo de los colectivos de año y en el trabajo diferenciado y personalizado con los estudiantes, determinando sus principales necesidades de formación y tomando las acciones necesarias para satisfacerlas, de modo que se logre una elevada retención y permanencia de los estudiantes en los diferentes años académicos. En los primeros años los colectivos de año, profesores guías y tutores definirán los lineamientos para la atención personalizada de los estudiantes y una adecuada caracterización individual del grupo docente. En tercero, cuarto y quinto años el trabajo docente educativo se centrará en la consolidación de los valores y objetivos educativos a través de la formación académica y científica, la cual será orientada y controlada por los colectivos de año y los departamentos docentes que directamente intervienen en el desarrollo profesional de los estudiantes.

1.5.1. Disciplina Ingeniaría y Gestión de Software.

En el nuevo plan de estudio D en la carrera de Ingeniería Informática se incluye según lo planteado en "Plan de Estudio D Ingeniería Informática Presencial" (Ministerio de Educación Superior 2007) la disciplina de Ingeniería y Gestión de Software que surge de la unificación de las disciplinas consideras como integradoras en el plan de estudio C´: Técnicas de Programación de Computadoras (TPC) e Ingeniería y Gestión de Software (IGS).

Esta disciplina se propone un conjunto de transformaciones fundamentales que se basan en los siguientes aspectos:

- Incorporar las nociones de captura de requisitos funcionales desde las primeras asignaturas de programación.
- Destacar, desde momentos tempranos de la enseñanza de la programación, la importancia del modelado del sistema y la formalización de los patrones.
- Crear hábitos en la definición y uso de estándares, tanto de diseño de interfaces, como

de codificación.

- Crear hábitos en el registro de tiempo y defectos, así como en el manejo de riesgos, que aseguren las bases para una adecuada gestión de proyecto y de su calidad.
- Crear habilidades en la definición de la arquitectura de un producto software.
- Incorporar habilidades para la definición y ejecución de casos de prueba de un producto de software, así como el desarrollo de revisiones utilizando listas de chequeo en cada flujo de trabajo.
- Garantizar la formación de diferentes roles dentro de un equipo de proyecto a lo largo de las distintas asignaturas de la disciplina y controlar su desempeño en la realización de los proyectos.

Además en "Plan de Estudio D Ingeniería Informática Presencial" (Ministerio de Educación Superior 2007) se resalta sus objetivos instructivos:

- Aplicar, con un alto nivel de profesionalidad, los principios de la programación prescriptiva en la automatización de cualquier aplicación.
- Diseñar las estructuras de datos adecuadas para la solución automatizada de cualquier aplicación.
- Utilizar mecanismos automatizados de ayuda a la generación de código.
- Manejar literatura en idioma extranjero con respecto a las técnicas y los productos de programación que utilicen.
- Analizar un sistema en una organización de base productiva o de servicio o en cualquier otro medio para determinar la factibilidad económica, las tareas a automatizar, los medios técnicos necesarios y la planificación del proyecto.
- Aplicar metodologías modernas para organizar y dirigir el desarrollo, implantación y puesta en marcha de los sistemas informáticos.
- Desarrollar los procesos de gestión de software, asegurar la calidad de los productos de software.
- Dirigir procesos de desarrollo software.

- Desarrollar en equipo procesos de software.
- Utilizar herramientas CASE para auxiliarse en todas las etapas de trabajo de desarrollo de sistemas informáticos.
- Aplicar el enfoque sistemático al desarrollo de los sistemas informáticos.
- Preparar y entrenar al personal necesario para la puesta en marcha de los sistemas informáticos. Confeccionar la documentación técnica del sistema informático y la orientada a los usuarios finales.

Pero a su vez también cuenta con objetivos educativos tales como:

- Desarrollar en los estudiantes las formas de pensamiento lógico y capacidad de razonamiento, mediante la modelación conceptual y el análisis algorítmico de los problemas, como recursos metodológicos que les posibilite enfrentarse, con un adecuado nivel de profesionalidad, a las exigencias del desarrollo científico - técnico y a los problemas concretos y prácticos que a diario se presentan en nuestro país.
- Lograr que los estudiantes avancen en la consolidación de una concepción científica del mundo, mediante el estudio de las técnicas de modelación de datos y de programación, analizando el papel del hombre y las máquinas en el desarrollo social, político y económico, en consonancia con los principios del materialismo dialéctico, reconociendo el papel de la práctica social en el proceso del conocimiento
- Formar en los egresados los hábitos de organización personal y responsabilidad que se requiere en el desarrollo de sistemas automatizados.
- Contribuir a formar profesionales con una alta calificación en cuanto a las técnicas de desarrollo de sistemas informáticos y que posean como cualidad distintiva la consagración, la sencillez, y la modestia que les permita trabajar en equipo con especialistas de diferentes perfiles y con todo tipo de usuario.
- Contribuir a que los estudiantes utilicen y desarrollen una forma dialéctica de pensamiento y que lo apliquen consecuentemente en su enfoque sistémico de

análisis.

- Contribuir a crear en el egresado hábitos de disciplina, independencia, creatividad y responsabilidad en las diferentes etapas de desarrollo de un sistema informático.
- Contribuir a educar a los estudiantes en la consideración de las limitantes que pueden existir en la utilización de los medios técnicos y en la búsqueda de soluciones económicas.
- Propiciar el desarrollo de un estilo profesional de trabajo en el cual sea objeto permanente de atención la calidad de los resultados de éste, lo que estará dado fundamentalmente por la sencillez y elegancia de las soluciones adoptadas y de la documentación técnica confeccionada.
- Desarrollar en los estudiantes la necesidad de búsqueda de información adicional acerca de los avances en las técnicas e instrumentos para el desarrollo de un software y en la capacidad de asimilar y desarrollar nuevas tecnologías, contribuyendo de esta forma a la creación de un fuerte espíritu de autopreparación.
- Lograr que el profesional se plantee y ejecute su trabajo tomando en cuenta prioritariamente las necesidades e intereses sociales.

El mundo actualmente se ha propuesto capacitar bien a sus futuros profesionales donde la vía más factible es el uso de Software Libre por las grandes ventajas que este proporciona.

Nuestro país no está exento de estos planes pero pertenecemos al gran grupo de países subdesarrollados y nos es muy difícil de adquirir los Software Privativos necesarios para nuestro desarrollo. Además en nuestro caso somos un país bloqueado en busca de las mejores alternativas para incrementar la tecnología, es por ello que con el uso de Software Libre los estudiantes adquirirán los conocimientos necesarios los cuales empleará para satisfacer las necesidades sociales e incrementar su espíritu para enfrentar el desarrollo científico - técnico y a los problemas concretos y prácticos que a diario se presentan.

1.6. Conclusiones

En este capítulo se abordaron ejemplos donde se manifiesta la necesidad de migrar a Software Libre así como se incursionó en la historia y licencias del mismo. También se caracterizó el nuevo Plan de Estudio D para la carrera de Ingeniería Informática, profundizando en la disciplina Ingeniería Gestión de Software de la cual se especificaron sus objetivos tanto instructivos como educativos. Además se expuso el uso y beneficio del Software Libre en la educación en las diferentes escalas.

Capítulo II: " Análisis y propuesta de herramientas"

2.1. Introducción

En este capítulo, se abordan las características de herramientas propietarias que se utilizan actualmente para el desarrollo de las asignaturas y de sus equivalentes libres, además cuál o cuáles de estas últimas son las que complementan los objetivos instructivos de cada asignatura de la disciplina Ingeniaría y Gestión de Software. Igualmente se realiza una caracterización del Software Libre y del Software Privativo.

2.2. Caracterización del Software Libre y del Software Propietario.

¿Qué es *Software Libre*? Es aquel programa o conjunto de ellos de los que el usuario puede disponer del código fuente, sin restricciones, y el cual puede modificar y redistribuir también sin limitaciones. Estas libertades garantizadas al usuario del software (o a aquel que lo recibe) no son diferentes a los derechos legítimos del autor del programa.

Entonces el *Software propietario* es lo contrario, o sea, el distribuido bajo una licencias prohibitivas y que no garantiza estas cuatro libertades antes citadas. Su normativa reserva la mayoría de los derechos de modificación, duplicación y redistribución para el titular de los derechos de propiedad intelectual.

2.2.1. Características del Software Libre

El Software Libre es rico en cuanto a sus características por los privilegios brindados a sus usuarios, entre las expuestas en "Software Libre vs software propietario Ventajas y desventajas" (Montserrat Culebro Juárez 2006) encontramos que:

- Tiene un alto nivel de *innovación tecnológica* pues su objetivo principal es compartir la información, trabajando de manera cooperativa y cuya ideología de sus defensores es que el conocimiento le pertenece a la humanidad, sin hacer distinciones. Por lo que sus usuarios tienen un destacado papel al influir en los errores que quieren que sean corregidos, proponiendo nueva funcionalidad al

programa, o contribuyendo ellos mismos en el desarrollo del software.

- Los *requisitos de hardware* son menores aunque es imposible hacerlo de forma general existen ejemplos documentados que lo demuestran. Como ejemplo encontramos los sistemas de GNU Linux que se emplean como servidores pueden ser utilizados sin la interfaz gráfica con la consecuente reducción de requisitos de hardware necesarios.
- La *durabilidad de las soluciones* está dada por la toma de decisiones por toda una comunidad no cuando el autor del software decide en un momento dado no continuar el proyecto para una cierta plataforma, para un hardware que considera antiguo, o descontinuar el soporte para una versión de su software.
- El *proceso de revisión o inspección* es público lo que motiva a contribuir de alguna forma en la corrección de errores, es decir, los usuarios de de todo del mundo, gracias a que disponen del código fuente de dicho programa, pueden detectar sus posibles errores, corregirlos y apoyar a su desarrollo con sus mejoras. Son comunes los casos en que un error de seguridad en GNU Linux se hace publicó y con él la solución al mismo. Con el software propietario la solución de los errores no llega hasta que el fabricante del programa puede asignar los recursos necesarios para solventar el problema y publicar la solución.
- Proporciona una total *independencia del proveedor* pues al sus usuarios disponer de su código fuente puede hacerle mejoras sin necesidad de esperar a que surja una nueva versión. Además puede hacer uso del Software Libre en cuantas veces le sea necesario, en cuantas máquinas desee y para cualquier fin. Se libera de toda dependencia de un proveedor único, y puede administrar su crecimiento y operación con total autonomía, sin temor de costos ocultos ni extorsiones.
- El *desarrollo de software* está dado por las facilidades que le proporciona a los profesionales el conocimiento necesario para dar solución a los problemas que surjan así como el entendimiento de los productos. Las herramientas le permiten crecer y operar con seguridad y total libertad.
- La *seguridad y privacidad de los datos* es mucho más factible pues los sistemas de

almacenamiento y recuperación de la información del software son públicos y cualquier programador puede ver y entender cómo se almacenan los datos en un determinado formato o sistema, lo que garantiza la durabilidad de la información y su posterior migración. Además por su carácter abierto se dificulta la introducción de código malicioso, espía o de control remoto, en razón de que el código es revisado por infinidad de usuarios y desarrolladores que pueden detectar posibles puertas traseras.

- La *adaptación de los software* resulta más fácil debido a se dispone del código fuente y por tanto los programas pueden ser modificados, o sea, adaptados tanto como sea necesario hasta que cubran exactamente la necesidad.
- La *curva de aprendizaje es mayor* dado a que resulta más complejo para aquellos usuarios que utilizaban software propietario pero para los principiantes el nivel es igual.
- El Software Libre no tiene garantía proveniente del autor.
- Se necesita dedicar *recursos a la reparación de errores* pero a diferencia de los propietarios de los cuales solo puedes esperar que surja una nueva versión porque no puedes solucionar los errores por la carencia del código fuente.
- No existen compañías únicas que respalden toda la tecnología.
- Las *interfaces gráficas de usuario (GUI) y la multimedia apenas se están estabilizando* aunque aumentan cada vez más los usuarios que aseguran que las interfaces gráficas más populares del Software Libre (KDE, GNOME y el manejador de ventanas WindowMaker) son ya lo suficientemente estables para el uso cotidiano y lo suficientemente amigables para los principiantes de la informática.
- La mayoría de la *configuración de hardware no es intuitiva* pues se requieren de conocimientos previos acerca del funcionamiento del sistema operativo y fundamentos del equipo a conectar para lograr un funcionamiento adecuado. Sin embargo la documentación referente a la configuración del hardware es tan explícita y detallada que permite al usuario neófito profundizar en el conocimiento de su hardware en muy pocas horas y una vez teniendo ese conocimiento la

configuración se vuelve ligera.

- El *usuario debe tener nociones de programación* pues la administración del sistema recae en la automatización de tareas y esto se logra utilizando, en muchas ocasiones, lenguajes de guiones (perl, python, shell, etc). Sin embargo, existen en la actualidad muchas herramientas visuales que permiten al usuario no técnico llevar a cabo tareas de configuración del sistema de una manera gráfica muy sencilla sin la necesidad de conocimientos de programación.
- La *diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas con un mismo fin, etc., pueden crear confusión en cierto número de personas* pues algunos ven esto como una fortaleza porque se pueden encontrar desde distribuciones especializadas en sistemas embebidos con muchas limitantes de almacenamiento y dispositivos periféricos de uso especializado hasta distribuciones optimizadas para su uso en servidores de alto rendimiento con varios procesadores y gran capacidad de almacenamiento; pasando por las distribuciones diseñadas para su uso en computadoras de escritorio y entre las cuales se encuentran las diseñadas para el usuario neófito que son muy fáciles de instalar y utilizar y las diseñadas para el usuario avanzado con todas las herramientas necesarias para explotar el Software Libre en todo su potencial. Cabe notar que la posibilidad de crear distribuciones completamente a la medida para atacar situaciones muy específicas es una ventaja que muy pocas marcas de software propietario pueden ofrecer y que Microsoft ha sido completamente incapaz de hacer.

2.2.2. Características del Software Propietario.

También en "Software Libre vs software propietario Ventajas y desventajas" (Montserrat Culebro Juárez 2006) se hace mención a determinadas características del software propietario, entre las cuales se encuentran:

- El control de calidad está dado por las compañías productoras de software propietario, las cuales por lo general tienen departamentos de control de calidad

que llevan a cabo muchas pruebas sobre el software que producen.

- Se *destinan recursos a la investigación* sobre los usos del producto.
- *Personal altamente capacitado*, o sea, se contratan programadores muy capaces y con mucha experiencia para la producción de este software.
- *Uso común por los usuarios*. El software propietario de marca conocida ha sido usado por muchas personas y es relativamente fácil encontrar a alguien que lo sepa usar.
- *Software para aplicaciones muy específicas* que no existe en ningún otro lado más que con la compañía que lo produce.
- *Amplio campo de expansión de uso en universidades* donde los planes de estudios de la mayoría de estas tienen tradicionalmente un marcado enfoque al uso de herramientas propietarias, de ahí que muchos egresados pueden comenzar su vida productiva utilizando estos productos de inmediato. No obstante, en algunos los centros de estudio se observa cambios en esta tendencia.
- *Difusión de publicaciones acerca del uso y aplicación del software* pues existe gran cantidad de publicaciones, ampliamente difundidas, que documentan y facilitan el uso de las tecnologías proveídas por compañías de software propietario, aunque el número de publicaciones orientadas al Software Libre va en aumento.
- Posee una *Curva de aprendizaje pequeña* al utilizar productos en los sistemas operativos de Microsoft pues ya las personas que cotidianamente utiliza este le resulta más difícil el uso de un sistema operativo nuevo y de herramientas nuevas, aunque ya existen reportes de experiencias con usuarios reales en quienes la curva de aprendizaje de GNOME o KDE fue mínima.
- *Soporte de las herramientas por diversas compañías* dado que las herramientas de Microsoft son soportadas por una gran cantidad de compañías de todos tamaños a nivel nacional e internacional. Además existe una red de certificaciones que proveen de "credibilidad" a las soluciones creadas por cada compañía.
- *Existen numerosas aplicaciones desarrolladas para la plataforma Win32 que no han sido portadas a otras plataformas* aunque existen numerosas herramientas,

libres o no, que facilitan la tarea de migración o reemplazos disponibles en las demás plataformas.

- *Mejoras para desempeño en ambientes de red* debido a que Microsoft ha estado mejorando mucho sus productos, para que tengan un mejor desempeño en ambientes de red. Sin embargo aún mantienen un rezago importante, ya que estas mejoras no han sido probadas lo suficiente por el mercado y la falta de interés por la seguridad es evidente.
- Soporte técnico ineficiente.
- *Ilegal o costosa la adaptación de un módulo del software a necesidades particulares* debido a que solo la compañía fabricante tiene el poder para realizar esta adaptación cobrando altos precios y solo cuando es inminente el cambio.
- *Derecho exclusivo de innovación* para la compañía fabricante. Si alguien tiene una idea innovadora con respecto a una aplicación propietaria, tiene que elegir entre venderle la idea a la compañía dueña de la aplicación o escribir desde cero su propia versión de una aplicación equivalente, para una vez logrado esto poder aplicar su idea innovadora.
- Ilegalidad de copias sin licencias.
- *Quedar sin soporte técnico* si la compañía fabricante del software propietario se arruina el soporte técnico desaparece, la posibilidad de en un futuro tener versiones mejoradas de dicho software desaparece y con el la posibilidad de corregir los errores de dicho software. Los clientes que contrataron licencias para el uso de ese software quedan completamente abandonados a su propia suerte.
- *Descontinuación de una línea de software* debido a que una compañía fabricante de software es comprada por otra más poderosa, es probable que esa línea de software quede descontinuada y nunca más vuelva a tener una modificación.
- Dependencia a proveedores.
- *Nulificación de desarrollo tecnológico de la industria nacional* respecto de la extranjera (las aplicaciones de consumo masivo se desarrollan en otros países).

Tras el análisis expuesto anteriormente respecto a las características de cada uno de ellos, es decir, del Software Libre y el Software Propietario es válido resaltar que el Software Libre representa el camino para alcanzar la independencia tecnológica de todos los países del mundo principalmente la de los subdesarrollados, también es una forma de contribuir a la erradicación de la piratería.

Además es la opción más factible para el desarrollo en las diferentes esferas, entre ellas la educación donde el empleo de este tipo de software le proporciona a los profesores la libertad de buscar e indagar a fondo cada herramienta y así escoger la más propicia para impartir sus clases e igualmente le brinda a los estudiantes la posibilidad de profundizar en otras diferentes a las utilizadas por las asignaturas y desarrollar sin restricciones. El Software Libre es una fuente de conocimientos para la población en general.

2.3. Caracteísticas de las herramientas para las asignaturas de la disciplina.

Existen grandes grupos de herramientas libres que les proporcionan a los usuarios disímiles beneficios y generalmente su ambiente es en cierta forma similar al de las propietarias usadas por la mayoría de las empresas y de administración pública.

Para una mejor información respecto a algunos equivalentes de los software privativos empleados se debe consultar la tabla de equivalencias *(Anexo II)*.

2.3.1. Asignatura: Introducción a la programación

La asignatura Introducción a la programación es la base para la incursión de los estudiantes en la programación, en esta se utiliza la herramienta privativa ***Borland C++ Builder*** pero dada las dificultades que esto provoca se puede valorar alternativas libres como:***Kdevelop, Code:: Block, Anjuta + Glade,*** entre otras.

Borland C++ Builder cuenta con características como:

- Ambiente de desarrollo rápido de aplicaciones (RAD) muy flexible.
- Ofrece un amigable entorno visual de desarrollo.
- Posee una gran cantidad de clases y objetos reusables.
- Permite la importación de código C++ existente
- Las aplicaciones creadas solo funcionan sobre la plataforma de trabajo Windows.
- Cuenta con una buena ayuda, pero en Internet no se pueden encontrar muchos foros de discusión sobre el mismo.
- Requiere del pago de una licencia para su uso

A continuación se caracterizan los equivalentes a esta herramienta propietaria.

Code:: Block es un servicio gratuito de C + + IDE construido para satisfacer las más exigentes necesidades de sus usuarios. Está diseñado para ser muy extensible y absolutamente configurable, es un IDE con todas las características que se necesitan, su funcionamiento es a través de plataformas.
Cualquier tipo de funcionalidad se puede completar mediante la instalación o codificación de un plugin.

Anjuta es un ambiente integrado de desarrollo para programar en C y C++ en GNU/Linux, creado para trabajar con GTK/GNOME. Posee licencia GPL y cuenta con un administrador de proyectos, plantillas, depurador interactivo y un poderoso editor.

Este se tiene varias características que fueron enumeradas en "Manual Glade y Anjuta." (Carlos Joa & Ninoska Cardona 2006):

- Editor de textos que integra las herramientas GNU (gcc, automake, autoconf, make y gdb) en un entorno que hace más fácil el seguimiento de los cambios.
- Permite desarrollar proyectos en varios lenguajes de programación.
- Sirve de soporte para resaltar la sintaxis en función del lenguaje.

- Ofrece una visión de la estructura del directorio del proyecto.
- Permite la integración del depurador gdb con el texto.
- Ofrece la generación automática de los archivos necesarios para la configuración, previa a la instalación del proyecto.

Glade es una herramienta para el desarrollo visual de aplicaciones, entre sus lengüetillas de componentes resaltan las de GTK y GNOME. También es capaz de generar código para implementar la interfaz visual, que se detalla en XML, en C, C++, Ada 95, Perl y Eiffel. El código generado por este se compila e instala con: configure, make, make install, entre otros.
Además como se especifica en "Manual Glade y Anjuta " (Carlos Joa & Ninoska Cardona 2006) puede crear la interfaz de usuario de las aplicaciones, de dos formas diferentes, bien sea generando código fuente, o, cargando dinámicamente un fichero XML de descripción de la misma en tiempo de ejecución. Cualquiera de las dos alternativas está disponible para una gran cantidad de lenguajes de programación.
Generalmente se encuentra incorporado en el programa Anjuta.

KDevelop es un IDE idóneo para cualquier tarea de sistema de programación para desarrollo RAD (ambiente de desarrollo rápido de aplicaciones) de GUI. Incluye todas las herramientas de desarrollo estándar de UNIX, proporciona soporte para equipo de desarrollo con varias herramientas de sistema de control de versiones y está disponible en diferentes plataformas como Mac OS X y Windows/Cygwin además de GNU/Linux, posiblita el desarrollo multiplataforma.

Dentro de las características expresadas en "KDevelop" (Alexander Dymo 2004) se encuentran:

- Soporta alrededor de 15 lenguajes de programación y numerosos sistemas de control de versiones, depuradores, formatos de documentación y herramientas de construcción. Cualquier editor está soportado por los interfaces de

KTextEditor que pueden ser usados como editor nativo de KDevelop, incluyendo kate, QEditor y kvim. Herramientas de construcción como automake, qmake están soportadas de forma nativa, esta herramienta no mantiene su propio repositorio de objeto. Los cambios en los ficheros del proyecto nativo se reflejarán en Kdevelop y viceversa.

- El soporte C++ ofrece un visor de clase mostrando todos los símbolos de forma plano o en jerarquía de namespaces y permite una fácil navegación por el código. Un parser en segundo plano actualiza los símbolos al vuelo y también muestra los errores sintácticos en el código fuente. La completación de código incluye las signals y slots de Qt.
- Con su nueva arquitectura abierta, KDevelop puede ser personalizado en todos los aspecto. La interfaz de usuario puede cambiarse al vuelo del modo clásico MDI al moderno modo IDEAl, Representa una aproximación centrado en el código con vistas de herramienta bajo demanda.
- Soporta amplias funcionalidades para la navegación por el código fuente. Se pueden acceder a símbolos y nombres de ficheros simplemente escribiendo parte de su nombre, se pueden tener acceso a los ficheros modificados recientemente con una simple pulsación de tecla.
- Ofrece generadores de documentación, depurador de código, comprobador de memoria, "code refactoring", gestión de marcadores y docenas de herramientas para un desarrollo más fácil y cómodo. Está traducido a más de 16 idiomas, que van del Inglés al Tamil.

Esta herramienta, es decir, ***KDevelop*** es la más completa y factible para el desarrollo de esta asignatura pues contribuye a facilitar la comprensión por parte de los estudiantes de la carrera debido a las grandes ventajas dentro de ellas encontramos que soporta alrededor de 15 lenguajes de programación y numerosos sistemas de control de versiones, depuradores, formatos de documentación y herramientas de construcción y amplias funcionalidades para la navegación por el código fuente. Se pueden acceder a

símbolos y nombres de ficheros simplemente escribiendo parte de su nombre, se pueden tener acceso a los ficheros modificados recientemente con una simple pulsación de tecla.

2.3.2. Asignatura: Introducción a la Informática.

En esta asignatura los objetivos instructivos se enfocan en brindar a los estudiantes el conocimiento de los principales campos de estudio y de aplicación del Ingeniero Informático y la utilización de software de uso general lo que puede ser una bueno para fomentar en los alumnos la importancia del Software Libre e incentivar a el uso del mismo.

Además es una primera aproximación a las demás asignaturas de la disciplina por lo que en ella se deben usar las mismas herramientas que proponemos en cada una de estas así como el uso de navegadores, el paquete de ofimática y herramitas para el tratamiento de imágenes, actualmente se emplean ***Internet Explorer, Adobe Photoshop y Microsoft Office*** todas propietarias por lo que se deben buscar equivalentes libres tales como: ***Mozilla Firefox, Gimp y Open Office.***

Internet Explorer

Internet Explorer 7 según se expresa en "Características" (Microsoft Corporation 2009) ofrece mejoras en la navegación gracias al buscador en pestañas, la posibilidad de búsqueda directamente desde la barra de herramientas, opciones avanzadas de impresión, facilidades de encontrar los envíos RSS, leerlos y suscribirse a ellos además de otras muchas cosas.

Dentro de sus características encontramos:

- Visita varios sitios en una sola ventana del explorador. Puedes pasar fácilmente de un sitio a otro a través de las pestañas que se encuentran en la parte superior del marco del explorador.
- Navega de forma sencilla entre las pestañas abiertas: basta con mostrar sus miniaturas en una sola ventana.

- Las pestañas se pueden agrupar y guardar en categorías lógicas, lo que posibilita abrir varias con un solo clic. Se puede establecer fácilmente un grupo de pestañas como grupo de la página principal, de modo que se abra todo el conjunto cada vez que se inicia Internet Explorer.
- Se ha vuelto a diseñar la interfaz y se ha mejorado para maximizar el área de la pantalla donde se muestra la página web, de modo que se ve mejor la parte necesaria y menos la que no lo es.
- Internet Explorer 7 reduce de forma automática las páginas web para imprimirlas, de forma que toda la página quepa en la hoja. Entre las opciones de impresión se encuentran los márgenes ajustables, los diseños de páginas personalizables, los encabezamientos y pies de página prescindibles y una opción para imprimir sólo el texto seleccionado.
- Ahora puedes realizar las búsquedas en Internet a través de tu proveedor preferido desde el cuadro de búsqueda de la barra de herramientas, con lo que se evita la acumulación de barras. Es muy fácil elegir un proveedor: sólo hay que seleccionarlo en la lista desplegable, a la que además se pueden agregar otros proveedores.
- Internet Explorer 7 detecta de forma automática las fuentes RSS de los sitios e ilumina un icono en la barra de herramientas. Un solo clic en el icono permite al usuario obtener una vista previa y, opcionalmente, suscribirse a las fuentes RSS del sitio para recibir notificaciones de forma automática cuando se actualiza el contenido. Es posible leer las fuentes RSS directamente desde el explorador, buscar historias importantes y filtrar la vista que se presenta con ayuda de términos de búsqueda o por categorías específicas.
- Amplía una página web en concreto, incluidos el texto y los gráficos, para centrarte en un contenido específico o para facilitar la accesibilidad a usuarios con problemas de visión.

Adobe Photoshop como se plantea en "Grandes características de Adobe Photoshop – artículo" (DiscoveryArticles 2008) cuenta con características como:

- Cuenta con una interfaz amigable muy fácil de navegar.
- Las herramientas se ordenan en grupos de gamas de colores, se puede modificar según la preferencia del trabajo.
- Emplea efectos de capas.
- Permite aplicar fácilmente efectos a una imagen o un texto como sombras de las gotas, los cartabones internos y externos y otros efectos.
- Tiene también la capacidad para utilizar modelos de color tales como: RGB, CMYK, BITMAP binaria, grayscale, y duotone. También utiliza formatos como: .PNG, .EPS, .GIF, .JPEG, etc.

Microsoft Office

Microsoft Office 2007 es la versión más reciente de la suite ofimática de Microsoft, como se expresa en "Cuáles son las características de Microsoft Office 2007" (Yahoo 2009) cuenta con:

- La nueva interfaz gráfica llamada Office Fluent, también conocido como cinta de opciones, que reemplaza al menú y a la barra de herramientas que fueron características desde su inicio.
- Incluye Microsoft Office Server 2007, un sistema de revisión en red de aplicaciones de Office, tales como Excel o Word.
- Elimina Microsoft Office FrontPage e incluye Microsoft Office SharePoint Designer y Microsoft Expression Web.
- Funciona bajo plataformas operativas Microsoft Windows y Apple Mac OS, aunque también lo hace en GNU Linux si se utiliza un emulador como Wine o CrossOver Office.

Mozilla Firefox

Esta herramienta cuenta con disímiles característica manifiestadas en ¨Características de Firefox, suficientes cosas buenas como para cambiar la forma en la que usas la web¨ (Mozilla Europe y Mozilla Foundation s.d.):

- Añade a marcadores, busca y organiza páginas web rápida y fácilmente.
- Evita los fraudes en línea, transacciones no seguras y falsificaciones gracias a una identificación sencilla de sitios.
- Lee las páginas web más rápidamente, usando menos memoria de tu ordenador.
- Ve cualquier parte de una página web, tan cerca y legible como quieras, en cuestión de segundos.
- Recuerda las contraseñas de los sitios sin ver ni una ventana emergente.
- Encuentra tus sitios favoritos en segundo, para un resultado instantáneo introduce un término relacionado.
- Si Firefox se cierra de golpe, no tienes que perder tiempo recuperando datos o siguiendo tus pasos a través de la web. Si estabas escribiendo un mensaje de correo electrónico, volverás justo donde lo habías dejado, incluyendo hasta la última palabra que habías escrito. La restauración de sesiones te devuelve instantáneamente tus ventanas y pestañas, recuperando el texto que hayas introducido y cualquier descarga en curso. Reinicia el navegador sin perder el trabajo.
- Cada nuevo sitio aparece en una nueva pestaña (no en una nueva ventana) y se puede acceder a ella con un solo clic.
- Ordena tus sitios. Simplemente coloca tus pestañas en orden arrastrándolas con un sencillo movimiento de ratón.

Open Office es una suite ofimática que incluye procesador de documentos de textos, planilla de cálculos, editor de presentaciones, editor de gráficos vectoriales, editor de páginas Web y diseñador de bases de datos. Dentro de sus características encontramos:

- Los formatos propios de este paquete a diferencia de los de Microsoft Office

cumplen con estándares abiertos .

- Es multiplataforma, los documentos realizados en el mismo tienen asegurada la portabilidad.
- Es distribuido gratuitamente bajo la Licencia Pública General (GPL).
- Se integra además con bases de datos como MySQL y PostgresSQL, con una funcionalidad similar o superior a Microsoft Access.
- Desde la versión 3 de Openoffice es sencillo añadir extensiones (plugins) para añadir otras funcionalidades adicionales, a la manera de Firefox.

Gimp es un programa de edición de imágenes muy conocido dentro del mundo de GNU/Linux, también nos proporciona ventajas tales como:

- Es *libre*. Una de sus ventajas es su licencia, siendo uno de los más importantes programas dentro del mundo del Software Libre.
- Es *gratuito*.
- Es *multiplataforma*. Los puedes usar en muchos sistemas operativos, entre ellos Windows, Linux y MacOs.

Esta herramienta, es decir, ***Gimp*** es la mejor alternativa al Adobe Photoshop por sus características y similitud con esta última. Además para una mejor navegación por los sitios Web se debe emplear ***Mozilla Firefox*** pues cuenta con facilidades como pestañas para acceder a diferentes sitios sin cambiar de ventanas, en casa de cualquier problema ya sea reinicio o cierre de la aplicación se restaura las sesiones, es decir, no se pierde en lo que estabas trabajando y el consumo de memoria es menor. Para la realización de documentos de textos y presentaciones es factible el uso de ***OpenOffice*** es una gran alternativa al paquete privativo Microsoft Office pues posee una compatibilidad prácticamente absoluta con los formatos de este. Además brinda facilidades como la exportación de datos al formato .pdf y soporta el estándar ISO OpenDocument con lo que es fácil el intercambio de documentos con muchos otros programas.

2.3.3. Asignaturas: Diseño y Programación Orientada a Objetos & Estructutras de datos

Estas asignaturas persiguen objetivos como el desarrollo de algoritmos ya sean de alta o mediana complejidad partiendo del uso del paradigma de la programación orientada a objetos y el diseño de clases para los algoritmos e igualmente prever los errores y hacer tratamiento adecuado de estos, para estos se está utilizando la herramienta privativa ***Borland C++ Builder*** caracterizada en el subepígrafe 3.3.1, para lograr la independencia de estas herramientas existen opciones libres como ***Eclipse*** y ***NetBeans.***

NetBeans como se resalta en "Conozca el nuevo NetBeans" (Sun Microsystems 1994) posee amplias posibilidades de desarrollo multiplataforma que atraen a muchos desarrolladores que han trabajado con otras herramientas, por ejemplo el importador del proyecto NetBeans permite a los usuarios guardar el trabajo desarrollado con JBuilder y continuar los proyectos en esta herramienta. Su flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio. Este está más inclinado al ámbito comercial puesto que garantiza que las aplicaciones creadas con el IDE se adhieran a los estándares de la industria e igualmente permite que los desarrolladores se concentren en la lógica comercial durante todo el desarrollo de las aplicaciones, es también una reflexión de cómo el software debe satisfacer los requerimientos del mercado actual.

Algunas de sus características son:

- Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).
- Cuenta con un sistema de proyectos basado en Ant, control de versiones, etc.
- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)

- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

Eclipse es una comunidad de fuente abierta, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles y herramientas para crear, desplegar y gestionar software en todo el ciclo de vida.
IBM lanzó la Plataforma de Eclipse en Código Abierto, convirtiéndose este en un órgano independiente que impulsa la evolución de la plataforma en beneficio de los proveedores de ofertas de desarrollo de software y usuarios finales. Toda la tecnología y el código fuente siempre y desarrollado por esta comunidad en rápido crecimiento está disponible gratuitamente a través de la Eclipse Public License.

Algunas de las características más importantes son:

- Autocompletamiento (no necesita incluir el objeto pues escanea el proyecto).
- Muestra la estructura en métodos y las propiedades de las clases.
- Muestra la herencia entre las clases.
- Autocompleta llaves, paréntesis, autotabula el código de acuerdo a reglas predefinidas o propias.
- Permite múltiples proyectos a la vez.

Esta herramienta (***Eclipse***) es la más factible para el desarrollo de esta asignatura pues cuenta con múltiples plugins dentro de ellos para programación Web, programación en lenguajes como C++, java, etc; así como mejor interpretación del código exportado. Además puede proporcionar un óptimo aprendizaje a los estudiantes y les permite desarrollar una buena práctica de los conocimientos adquiridos contando con todas sus facilidades.

2.3.4. Asignaturas: Bases de datos & Bases de datos avanzadas

Las asignaturas de Bases de Datos y Bases de Datos Avanzadas son de gran importancia en la formación de los estudiantes de la carrera, ellas cuentan con objetivos como:

representar fenómenos de la realidad objetiva a través del modelo global de los datos así como el desarrollo de algoritmos complejos para el procesamiento de los datos almacenados en las bases de datos partiendo del uso de gestores de bases de datos, además el desarrollo de trabajos de diseño de bases de datos también avanzadas. Se emplea ***SQL Server*** y ***Microsoft Acces*** ambas privativas pero para ello se tienen alternativas libres de gestores como: ***PostgreSQL***, ***MySQL*** .Además para el diseño se pueden utilizar ***PgDesigner, DBDesigner 4*** y para la administración ***PgAdmin III*** y ***phpMyAdmin***..

SQL Server cuenta con características como las expuestas en "Manual de SQL" (2004) ***:***

- Opera en una arquitectura cliente/servidor con gran rendimiento.
- Orientado para hacer posible manejar grandes volúmenes de información, y un elevado número de transacciones.
- Depende de la potencia del hardware del equipo en el que esté instalado.
- Permite la creación de procedimientos almacenados,
- Solamente corre sobre plataforma Windows.
- Requiere del pago de una licencia para su uso.

Microsoft Acces

- Cuenta con una buena ayuda para usuarios de poca experiencia.
- Tiene una interfaz interactiva que permite la creacion de consulta, vistas y reportes a personas inespertas.
- Soporta un número limitado de conexiones desde aplicaciones externas.
- Solo es para plataforma Windows .
- Requiere del pago de una licencia para su uso.

La siguiente tabla expuesta en "Diseño de un Datamart y de los procesos ETL, para la toma de decisiones en las áreas de portadores energéticos y Transporte del MES" (Abdiel Alejandro Caballero Legrá s.d.) muestra una comparación entre los dos gestores de bases de datos más importantes entre las herramientas libres: **PostgreSQL** y **MySQL.**

Parámetros	MySQL 5.x	PostgreSQL 8.0.x
Sistema Operativo	Linux, Windows, FreeBSD, MacOS X, Solaris, HP UX, AIX, entre otros.	Windows, más de 2 docenas de la familia Unix (Linux, todos los BSDs, HP-UX, AIX, OS X, Unixware, Netware...)
Soporta Subconsultas	Sí	Sí
Soporta Transacciones	Sí	Sí
Replicación	Sí	Sí
Soporta Llaves Foráneas	Sí	Sí
Vistas	Sí	Sí
Procedimientos Almacenados	Sí	Sí
Disparadores (Triggers)	Sí	Sí
Full joins	No	Sí
Restricciones (Constraints)	No	Sí
Cursores (Cursors)	Parcial (Solo lectura)	Sí
Dominios definidos por el usuarios	No	Sí
Lenguajes procedurales	ANSI SQL 2003	PL/pgSQL, PL/Tcl, PL/Perl, PL/Python PL/PHP, PL/Java o definido por el usuario
ODBC	Sí	Sí
JDBC	Sí	Sí
Velocidad en las Consultas (grandes volúmenes de datos)	Lento	Rápido

Fig. 1 "Comparación entre MySQL y PostgresSQL"

Ambos gestores son firmes y sólidos, poseen conjuntamente amplio respaldo de la comunidad de desarrolladores, por lo que su sostenimiento y desarrollo están garantizados. Existen diferencias que permiten considerar a ***PostgreSQL*** el más factible para el desarrollo de las asignaturas pues:

- Es más eficiente a la hora satisfacer peticiones que involucren grandes volúmenes de datos, lo cual constituye un requisito indispensables para un sistema de toma de decisiones.
- Ofrece más funcionalidades y tiene mayor experiencia en módulos que fueron incluidos en ***MySQL*** recientemente.
- Manipula mayor cantidad de usuarios concurrentes.
- Agregación en memoria usando tablas de hashing para hacer que las consultas de data-warehousing sean más rápidas.
- Implementa módulos adicionales para el trabajo con cubos en el entorno multidimensional.
- Mayor soporte de los estándares SQL 92 y SQL 99 que MySQL.
- Tiene mejor performance en Querys complejos que ***MySQL.***
- Brinda servicios adicionales como: autentificación de Apache Web Server, OpenLdap, variedad de Software Libre de aplicaciones específicas: foros, cms, erp, crm, galerías fotográficas, etc y también comunicaciones usando el protocolo IPv6.
- Cuenta con PgCluster, un parche adicional que nos permite manejar cluster de bases de datos, balanceo de carga y replicación de servidores todo en uno (solo disponible para la versión 7.3.6).

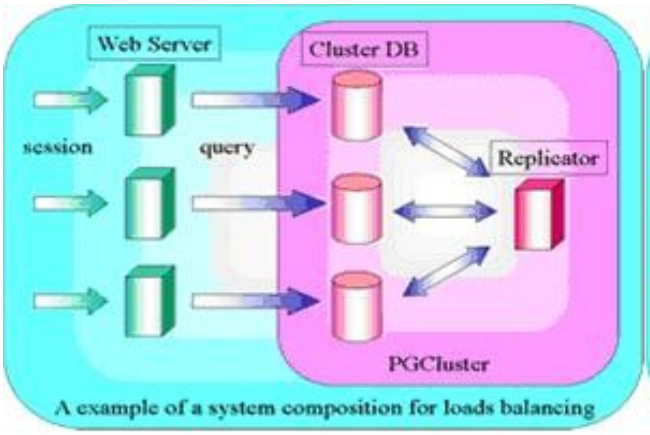


Fig. 2 "Composición del sistema por balance de carga "

Fig. 3 " Composición del sistema de alta fiabilidad "

Además presenta vistas, funciones store procedures (que pueden desarrollarse en varios lenguajes, cada lenguaje debe ser inscrito en la base de datos que lo usará), triggers y reglas (integridad referencial), transacciones planas (no soporta transacciones anidadas), tipos de datos definidos por el usuario, herencia, funciones especiales para administrar concurrencia y bloqueos de usuarios, índices comunes e índices funcionales, vistas materializadas, soporte de Esquemas en XML.

Para las funciones de crear un modelo visual de cualquier base de datos existe herramientas libres tales como:

PgDesigner que como se resalta en "PgDesigner" (Desconocido s.d.) su código está escrito en la lengua de Gambas, y actualmente sólo funciona en sistema operativo GNU Linux. Además está en un estado de continuo desarrollo y actualización, teniendo en cuenta la continua evolución de Gambas y PostgreSQL, tanto aplicaciones tecnológicas requeridas por la evolución continua de software en general. A pesar de que está en constante evolución, pgDesigner puede usarse de manera segura para construir sobre la base del motor de base de datos PostgreSQL.

Este cuenta con determinadas características que reflejan su funcionamiento, estas son:

- Multiproject gestión.
- Puntos de vista de la gestión del proyecto.
- Copia de objetos entre los diferentes proyectos.
- Creación de objetos a base de PostgreSQL, tales como tablas, campos e índices, puntos de vista, las relaciones, de tablas, procedimientos, disparadores, los tipos, y las secuencias de los dominios.
- Creación de zonas de delimitación rectangular, con la personalización de llenar de color y texto.
- Creación de texto como gráficos en el diagrama del proyecto.
- Cómo guardar los archivos en los proyectos, tanto en formato INI y XML.
- La gestión de liberación de PostgreSQL, a partir de la versión 7.0 a 8.2.
- Ingeniería inversa llevada a cabo por servidores PostgreSQL, local o remota.
- Personalización visual de medio ambiente en general, la configuración básica de cada proyecto o cada objeto, la gestión de los detalles visuales.
- Gestión del proyecto en un diagrama gráfico.
- Actualización automática de las relaciones entre los objetos del proyecto.
- Cada tipo de objeto distinto del icono especial.
- Colocación de objetos arrastrando con el ratón o mediante la inserción manual de las coordenadas.
- Actualización automática de las relaciones entre tablas.
- Asistente para la construcción de puntos de vista.
- Panel de proyectos, visualización y gestión de múltiples proyectos simultáneamente.
- Lista de temas en cada proyecto.
- Lista de puntos de vista para cada proyecto.
- Panel de instrumentos con objetos o entidades que se pueden insertar en el

proyecto.

- Panel de estado para ver el estado actual del proyecto actual.
- Ver y escribir archivos en los comandos SQL necesarios para crear la base de datos.
- Creación de base de datos directamente en un servidor PostgreSQL.
- Gestión previa para el diseño de la impresión gráfica, con la posibilidad de enviar archivos de imagen o de la impresora.
- Informes de gestión de proyectos con una vista previa, archivos de texto o enviar a la impresora, en formatos: texto, html y pdf.
- Personalización mundial color, o de cada uno de los objetos.
- Creación de proyectos sobre la carga de archivos que contienen comandos SQL.
- Función de búsqueda de objetos en el proyecto.

DBDesigner 4 es un sistema visual de diseño de bases de datos que integra el diseño, modelado, creación y mantenimiento en un único entorno sin fisuras. Combina características profesionales y una clara y sencilla interfaz de usuario para ofrecer la forma más eficaz de manejar sus bases de datos.
El mismo es un proyecto de código abierto disponible para Microsoft Windows y Linux 2k/XP de KDE / Gnome.

Partiendo de lo expuesto en "MySQL Workbench 5.0 - El DBDesigner 4 Sucesor" (fabFORCE.net 2003) este sistema es muy rico en cuanto a sus características, por ejemplo:

- Interfaz de usuario basada en la distribución de software estándar de la industria:
    1. Lienzo de navegación similar al de Adobe Illustrator ® y Photoshop ®
    2. Paletas (acoplado / flotantes)
    3. Objetos disponibles incluyen tablas, relaciones, las etiquetas, las

regiones, las imágenes

4. Redacción Avanzada
5. Copiar/Cortar/Pegar en portapapeles funciones (XML, DDL)
6. Alinear las funciones

- Modo de diseño / Modo consulta (El modo de consulta se utiliza para trabajar con los datos de las tablas y crear la consulta SQL compleja estados para el uso en PHP, Kylix u otro lenguaje de programación y el modo de diseño se utiliza para crear y mantener el modelo de bases de datos visuales.)
- La ingeniería inversa de MySQL, Oracle, MSSQL y cualquier base de datos ODBC.
- Soporta dos interfaz de usuario conmutable.
- Modelo para Base de datos de sincronización.
- Índice de apoyo.
- Colocación automática de claves foráneas.
- Entidad débil.
- Inserta el almacenamiento y el estándar de sincronización.
- Capacidades de toda la documentación.
- Modelo avanzado de impresión.
- Todos de datos MySQL con todas las opciones.
- De datos definidos por el usuario.
- Almacenamiento de base de datos, la capacidad de guardar en la base de datos modelo.
- Red multiusuarios para el acceso a través de la base de datos de almacenamiento.
- De control de versiones (El control de versiones sólo están disponibles si se utiliza la base de datos de almacenamiento).
- Comando de SQL de almacenamiento en el modelo.
- Plugin de interfaz.

- Con un parche genera SQL para PostgreSQL.

Para la realización de los diseños de las bases datos en las asignaturas es más recomendable el uso de ***DBDesigner 4*** por sus innumerables características y cuenta con un parche mediante el cual genera SQL también para ***PostgreSQL*** manteniendo sus mismas facilidades***,*** a pesar, de que fue desarrollado con el fin de generar SQL solo para ***MySQL*** .

Para la administración de las bases de datos encontramos a: ***PgAdmin III*** y ***phpMyAdmin,*** ambas herramientas son factibles para el desarrollo práctico de estas asignaturas por sus características.

PhpMyAdmin es una herramienta de Software Libre escrito en PHP destinado a manejar la administración de MySQL a través de la WWW. Además es compatible con una amplia gama de operaciones con MySQL. Su uso más frecuente son las operaciones de apoyo de la interfaz de usuario (manejo de bases de datos, tablas, campos, relaciones, índices, usuarios, permisos, etc).
Sus características más específicas destacadas en "php MyAdmin" (equipo de desarrollo de phpMyAdmin 2003) son:

- Intuitiva interfaz Web.
- Buscar y borrar bases de datos, tablas, vistas, campos e índices.
- Crear, copiar, eliminar y cambiar el nombre de alterar las bases de datos, tablas, campos e índices.
- Mantenimiento del servidor, bases de datos y tablas, con propuestas sobre la configuración del servidor
- Ejecutar, editar y favorito de cualquier base de datos de la declaración, incluso las consultas por lotes.
- Gestión de usuarios y privilegios de MySQL.

- Importar datos de CSV y SQL
- Los datos de exportación a diferentes formatos: CSV, SQL, XML, PDF, ISO / IEC 26300 - OpenDocument Text, Word, Excel y Latex.
- Administración de múltiples servidores.
- Creación de gráficos en PDF de su base de datos de diseño.
- Creación de consultas complejas utilizando Consulta por ejemplo (QBE).
- Búsqueda en todo el mundo en una base de datos o un subconjunto de ella.
- La transformación de los datos almacenados en cualquier formato a través de un conjunto de funciones predefinidas, como mostrar datos BLOB como imagen o enlace de descarga.

PgAdmin III es el más popular y ricos dentro de la característica de código abierto en la parte de administración y plataforma de desarrollo, es para PostgreSQL, la más avanzada base de datos de código abierto en el mundo.

Este no se queda atrás en con respecto a sus características, algunas de las comentadas en "PgAdmin PostgreSQL Tools" (Desconocido s.d.) son:

- Está diseñado para responder a las necesidades de todos los usuarios, de la escritura simples consultas SQL a la elaboración de bases de datos complejos.
- La interfaz gráfica de PostgreSQL soporta todas las características y hace fácil la administración.
- Conexión con el servidor puede hacerse utilizando TCP / IP o Unix Domain Sockets y puede ser encriptado SSL para la seguridad.
- No se requieren drivers adicionales para comunicarse con el servidor de base de datos.

2.3.5. Asignatura: Programación Web

La asignatura Programación Web abarca objetivos tales como el diseño de aplicaciones Web basadas en los principios de ingeniería de software para un proceso adecuado de desarrollo al igual que desarrollar trabajos de programación en entornos Web, se está trabajando con ***Macromedia Dreamweaver***, software comercial, es decir, privativo por ello se debe buscar alternativas libres, algunas de ellas pueden ser: ***Bluefish, Nvu*** y ***Kompozer.***

Macromedia Dreamweaver cuenta con características como:

- Corre sobre plataformas Windows y MAC, está disponible en varios idiomas.
- Completamiento de código y coloreado en las sintaxis del mismo.
- Compatible con lenguajes como: HTML, Coldfusion, PHP, ASP VBScript, ASP.NET C#, Javascript, XML, XSLT, CSS, ActionScript, JSP, entre otros.
- Incluye Framework para AJAX y administrador CSS.
- Requiere de conexión a Internet para registrar el producto y pagar por su uso.

Bluefish es un potente editor para los programadores y diseñadores Web, cuenta con muchas opciones para escribir páginas Web, scripts y código de programación. Además soporta muchos lenguajes de programación y de marcas, y se centra en la edición de sitios dinámicos e interactivos. Está publicado bajo la licencia GNU GPL.
Entre sus características como se expresa en "Características de Bluefish" (Olivier Sessink 2008) resaltan:

- Su compatibilidad con los sistemas operativos POSIX, incluyendo GNU Linux, FreeBSD, MacOS-X, OpenBSD y Solaris.
- Su referencia de funciones del navegador, incluyendo archivos de referencia para PHP, CSS, Python y HTML.
- Posee etiqueta de cierre automático de documentos HTML y XML.
- Creación automática de miniaturas y la vinculación de la miniatura con la imagen

original.

- Insertar imágenes, tablas, marcos, etc.

Nvu, es multiplataforma con ejecución independiente. El mismo está disponible para GNU Linux, MacOSX y Microsoft Windows, aunque puede compilarse para cualquier plataforma con el Netscape Portable Runtime.
Dentro de sus características se aprecia su:

- Soporte integrado de CSS y mejor gestión del soporte FTP para actualización de los ficheros.
- Gestión de trabajo mediante proyectos.
- Soporte para todos los elementos típicos: marcos, formularios, tablas, plantillas de diseño, etc.

Kompozer es un editor de páginas web derivado de Nvu, otra aplicación de edición de páginas web basado en Mozilla Composer, *c*uyo desarrollo llegó a su fin hace un algunos años. Es multiplataforma y portátil, funcionan en Windows, en Mac y en GNU Linux (tiene versiones para Ubuntu y Fedora, las dos distribuciones Linux más populares).
Además es un programa gratuito y OpenSourse, e incluye todas las herramientas necesarias para diseñar las páginas más asombrosas, sin necesidad de ser un experto. Incluye todas o casi todas las funcionalidades de DreamWeaver o FrontPage y algunas mejoradas.

Entre sus características más interesantes planteadas en "Kompozer: el arte del diseño web " (Carlos Rojas s.d.) se destacan:

- Administrar sitios FTP de forma intuitiva y completa.
- Seleccionar colores y crear tonalidades personalizadas.
- Utilizar pestañas diferentes para cada trabajo, cada una con sus opciones de Deshacer y Rehacer.
- Personalizar las barras de tareas.

- Verificar la ortografía automáticamente.
- Permite usar estilos CSS, y es compatible con XML y JavaScript. EL código HTML creado es confiable y funcionará con la mayoría de los navegadores más populares.
- Al ser un programa completamente independiente, Kompozer consume pocos recursos del sistema: es liviano y veloz.
- Su facilidad de uso y la posibilidad de ver el diseño de la página en tiempo real al editar, hacen que sea un programa sumamente útil para los apasionados del diseño.

Esta herramienta, o sea, ***Kompozer*** es la más completa para la preparación de los estudiantes en esta asignatura pues dispone de pestañas que le permiten trabajar en varias páginas al mismo tiempo, entre las cuales se puede cambiar rápidamente; un excelente cuadro de diálogo para generar tablas, más otro muy completo para modificar estas. También ofrece varias vistas: donde se diseñan las páginas sin necesidad de escribir código; Etiquetas HTML, una vista esquemática de cómo están distribuidas las etiquetas; código fuente, que muestra el código HTML tal cual, y vista preliminar, que exhibe las páginas como se verán en el navegador semejantes a las de la Macromedia Dreamweaver. Todo esto facilita la preparación de los profesores para sus clases.

2.3.6. Asignaturas: Ingeniería de Software I, II, III

Estas asignaturas coinciden entre sus objetivos instructivos con el empleo de herramientas CASE para el análisis y diseño de proyectos. Actualmente se emplea el ***Rational Rose*** el cual no es libre y esto dificulta su uso debido a sus restricciones de licencias pero existen homólogos que si son libres tales como: ***Umbrello, ArgoUML, BoUML***, etc.

Rational Rose se caracteriza por:

- Herramienta CASE (Ingeniería Asistida por Computadora) que da soporte al

modelado visual con UML ofreciendo distintas perspectivas del sistema.

- Da soporte al Proceso Unificado de Rational (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la etapa de pruebas.

- Rational Rose ofrece un diseño dirigido por modelos que redundan en una mayor productividad de los desarrolladores, admitiendo UML, COM, OMT y Booch

- El diseño está centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.

- Posee capacidades de ingeniería inversa.

- Se encuentra disponible en múltiples plataformas.

Umbrello es una herramienta que contribuye a la automatización de los procesos de desarrollo de software, es decir, es un modelador *Unified Modeling Language* (UML) que facilitará la creación de un producto de alta calidad, especialmente durante fases de análisis y diseño del proyecto. Además está diseñado principalmente para KDE, aunque funciona en otros entornos de escritorio.

Además maneja gran parte de los diagramas estándar UML pudiendo crearlos, además de manualmente, importándolos a partir de código en C++, Java, Python, IDL, Pascal/Delphi, Ada, o también Perl. También permite crear un diagrama y generar el código automáticamente en los lenguajes antes citados.

BoUML es un servicio gratuito para UML 2 y a su vez una caja de herramientas (en desarrollo) que le permite especificar y generar código en C + +, Java, IDL, PHP y Python.

Se ejecuta en Unix / Linux / Solaris, MacOS X (Power PC e Intel) y Windows. Además es muy rápido y no requiere de mucha memoria para manejar miles de clases. Proporciona versiones compiladas para los principales sistemas operativos gracias al framework Qt. Además de su ligereza y facilidad para su instalación permite visualizar la documentación del proyecto en html e importar el diseño a xmi.

ArgoUML es una herramienta muy ligera que no requiere de instalación. Está desarrollada en java por lo que es portable para cualquier Sistema Operativo (Windows, GNU/Linux).

Posee características destacadas en *"*Estudio de herramientas CASE de análisis y diseño orientado a objetos*"* (Ivan Obeso Agüera s.d.) tales como:

- Permite hacer la mayor parte de diagramas de diseño (9 en total), agrupándolos en el mismo proyecto. Su manejo es muy intuitivo.
- Es Open Source (bajo licencia BSD) y se basa en UML 1.4 Las opciones se muestran en paneles con imágenes, desde mi punto de vista es una herramienta muy usable y fácil de manejar.
- En el paquete de idiomas que usa, se encuentra el español. Además existe mucha documentación y manuales sobre el uso de la herramienta en la página Web del proyecto.
- Vuelca el contenido en ficheros .XML, concretamente usa .XMI que es un formato XML usado entre herramientas UML.
- Los diagramas se exportan en imágenes con formato GIF, PNG, PS, EPS, PGML y SVG.
- Permite incluir el código fuente de los archivos representados en los diagramas, es decir, las clases. También se encarga de generar código automáticamente siempre que los diagramas sean correctos y suficientemente aclaratorios. Soporta los lenguajes Java, C++, C# y PHP.
- El proyecto se puede organizar en función de los paquetes usados, las clases, los

diagramas, las herencias,etc.

- La forma de construir diagramas es muy sencilla, mediante drag & drop se van desplegando elementos sobre el área de trabajo.
- También gestiona las tareas a realizar, organizándolas por prioridad, decisión, objetivo… Es una buena forma de organizar el trabajo.

Para un mejor desempeño de los estudiantes de la carrera en las asignaturas Ingeniaría de Software I, II, III y se cumplan sus objetivos planteados la herramienta más factible es ***ArgoUML*** pues además de ser multiplataforma permite hacer la totalidad de diagramas UML y los agrupa dentro del proyecto, contribuyendo esto a una buena organización de la información. Su interfaz es de fácil manejo para los usuarios. Aunque el modelo de usuario no está completamente implementado, el mismo consiste en modificar la herramienta para hacerla más usable en función de la información que se maneje del usuario .

2.3.7. Asignatura: Introducción a la Gestión de Software.

Esta asignatura cuenta con objetivos instructivos como el empleo de listas de chequeo del código para la corrección y almacenamiento de errores así como la utilización de medios formales para la documentación de todas las actividades realizadas durante la verificación, validación y el registro de los resultados obtenidos; recientemente se emplea la herramienta denominada ***Process Dashboard*** no obstante existen otras alternativas como ***DotProject*** que también es libre.

DotProject fue elaborada por aplicaciones de código abierto y es mantenida por un pequeño grupo de voluntarios. Además es Software Libre, multiusuario y soporta varios lenguajes. La plataforma recomendada para su uso es nombrada LAMP (Linux + Apache + MySQL + PHP). Aunque, existen binarios para instalarlo bajo otros sistemas operativos tales como Microsoft Windows (NT, 2000, XP) y Mac.

DotProject cuenta con un determinado conjunto de entidades ordenadas jerárquicamente, entre las principales encontramos:

- Compañías: Son las entidades que agrupan proyectos, actividades y usuarios.
- Departamentos: Son áreas dentro de las compañías, que permiten agrupar usuarios en dicho nivel.
- Usuarios/Contactos: DotProject tiene usuarios los cuales son capaces de loguearse en él y trabajar dentro del esquema de permisos que posea el rol de dicho usuario. Los contactos son usuarios especiales que asignados a un determinado proyecto pueden recibir por ejemplo: correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema DotProject. Los usuarios y contactos pertenecen a una compañía.
- Proyectos: Es la entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto.
- Actividades: son las tareas asignadas dentro de un proyecto. Son los componentes sobre los cuales se controla: la duración, dependencias, recursos asignados y progreso. Las actividades deben de pertenecer a un único proyecto.
- Diagramas de Gantt: Permite ver en forma gráfica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.
- Tickets: para administrar todos los problemas relacionados a un proyecto.
- Archivos: Permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.
- Foros: Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.
- Administración del Sistema: Contiene la actividades relacionadas a la administración de usuarios, roles y configuración del sistema.
- Recursos: Permite asignar recursos no humanos (oficinas, equipamiento, etc) a un proyecto.

Process Dashboard el cual en "Process Dashboard" (Desconocido s.d.) lo describen como un proceso actual de PSP (SM), herramienta de apoyo. Está disponible gratuitamente para su descarga bajo las condiciones de la Licencia Pública GNU.

Además este proceso apoya:

- *Recopilación de datos*: Tiempo, defectos, tamaño del plan en comparación con los datos reales.
- *Planificación:* Integrado de scripts, plantillas, formularios, y los resúmenes, PROBE, valor ganado.
- *Sigue:* Potente valor obtenido apoyo.
- *Análisis de Datos:* Gráficos e informes de la ayuda en el análisis de los datos históricos las tendencias.
- *Exportación de datos:* Los datos de exportación a Excel, o exportar datos a formato de texto para su uso con herramientas externas.

Los principales puntos fuertes del proceso de Dashboard son:

Facilidad de uso:

- Optimiza la facilidad de la recogida de los más comúnmente realizados métrica (tiempo y defectos).
- Otro proceso de ayudas (scripts, formas, plantillas, y los resúmenes).
- Su pequeña huella en pantalla de la herramienta permite a coexistir con entornos de desarrollo integrado.
- Las tareas se organizan jerárquicamente, lo que refleja el trabajo del proyecto de desglose de la estructura.

Flexibilidad / extensibilidad:

- Nuevos procesos y nuevos tipos de datos se pueden añadir sin programación.

- Proceso de secuencias de comandos, plantillas, formularios y resúmenes son HTML, y por lo que pueden ser creados con cualquier editor HTML.

Independencia de plataforma:

- 100% de ejecución de Java se ejecuta en Windows, Unix, GNU Linux, Macintosh, etc.

Esta herramienta, es decir, ***Process Dashboard*** es la más factible para dar cumplimiento a sus objetivos pues cuenta con una interfaz amigable y de fácil manejo para los usuarios, su pequeño tamaño permite coexistir con entornos de desarrollo integrado, no requiere de entidades para su funcionamiento lo que lo hace más sencillo para los profesores y los estudiantes.

2.4. Conclusiones

En este capítulo se caracterizan teniendo en cuenta varios aspectos tanto negativos como positivos el Software Libre y el Software Propietario así como las alternativas libres factibles para el desarrollo de las asignaturas de la disciplina Ingeniería y Gestión de Software. También se mencionan las características de las herramientas privativas empleadas actualmente.

Capítulo III : "Propuesta de la Estrategia"

3.1. Introducción

En este capítulo, se abordan los aspectos a tener en cuenta para llevar a cabo una buena migración, también se realizará un amplio análisis de las diferentes guías de migración internacionales tales como: las Directrices Europeas, la Guía Peruana y el Plan nacional de Venezuela. Además se manifestará lo planteado por la Guía cubana la cual servirá de base para la elaboración de la estrategia de migración para la disciplina de Ingeniería y Gestión de Software de la carrera.

3.2. Proceso de Migración a Software Libre

Para que el proceso de migración a Software Libre sea efectivo se deben tener en cuenta determinados aspectos expuestos en "Guía cubana de migración a Software Libre" (Ramón Paumier Samón & Yoandy Pérez Villazón s.d.), tales como:

1. **Crear un equipo con la capacitación y el respaldo de gestión adecuados**. Es importante que se disponga de apoyo de gestión, pues de lo contrario habrá resistencia a un cambio de la norma de sistemas propietarios. Este apoyo tendrá que ser suficiente para permitir por lo menos la construcción de pilotos representativos, permitiendo elaborar un caso de negocio básico, y quizá uno más detallado después, cuando se disponga de más datos.
2. **Entender el entorno final, tanto el Software Libre como la arquitectura básica, junto con las diferentes opciones y posibilidades disponibles**. Esto significa que hay que formar al personal, contratar personal o recurrir a consultores. Lo que implicará algunos costos iniciales y por ello es necesario disponer de respaldo de los responsables de la gestión. A veces existe la expectativa de que el Software Libre se puede entender y usar sin costo alguno.

3. **La migración es una oportunidad de revisar la arquitectura de base así como el software de aplicaciones**. La arquitectura que se recomiende se debe basar en el control centralizado y debe tener ciertas ventajas. Puede haber ciertos costos al hacer el cambio, y hay que tenerlos en cuenta.

4. **Es muy importante entender bien en qué consiste el SWL**. Hay algunos aspectos que hay que tener en cuenta antes de tomar alguna decisión:

- Hay que tener claro cuáles son las implicaciones de las licencias para SWL especialmente si se considera que la institución va a distribuir los cambios de software.
- Cuando hay varias opciones para una función (por ejemplo, hay por lo menos tres buenas hojas de cálculo de SWL) los ejecutores del proceso han de entender los pros y los contras de cada producto.
- Se deben tener en cuenta las diferencias entre las distintas distribuciones. Algunas distribuciones están respaldadas por empresas comerciales que prestan su apoyo y correcciones. Algunas tienen características diferentes: Gentoo, por ejemplo, da una distribución basada en un código fuente que facilita una adaptación del software para que satisfaga necesidades concretas. Todas estas diferencias han de ser valoradas antes de hacer cualquier elección.
- Los ejecutores deben determinar qué nivel de apoyo es necesario. Se puede conseguir apoyo comercial de los creadores de la aplicación o la distribución si la suministran. Si no es así, hay terceros que pueden prestar ese apoyo ya que se dispone del código fuente y hay muchas empresas internacionales que dan ese apoyo.

Lo antes descrito marca una diferencia con respecto al Software Propietario pues estos dependen de una compañía que es la responsable del mantenimiento por ser la única con acceso al código fuente mientras que el Software Libre cuenta con una gran comunidad de usuarios y una lista de correos que correos activa donde una pregunta o petición de

ayuda recibirá la respuesta de alguien relacionado con la aplicación.

5. **Estudiar los sistemas existentes**. Estos datos no solo serán necesarios para hacer la migración en sí, sino que muchos de ellos serán también muy necesarios para construir un modelo de costo total de propiedad para un caso concreto de negocio.

6. **Elaborar un caso detallado de migración**, que se basará en los datos recogidos y que consistirá en los siguientes puntos:

- El costo del entorno existente en un período de tiempo razonable.
- El costo de entornos alternativos y el costo de la migración a cada uno de ellos en el mismo período.
- Los puntos fuertes y débiles del entorno actual y las distintas alternativas.

7. **Consultar a los usuarios**. Explicar las razones que hay detrás de la migración y cómo les afectará.

- Estudiar sus preocupaciones con seriedad y permitirles que practiquen con la tecnología, sin pérdida de tiempo. Cuanto antes se impliquen los usuarios mejor será.
- Habilitar un espacio de atención al cliente que pueda dar respuesta a las preocupaciones de los usuarios. Más adelante, cuando la migración esté configurada, podrá resolver los problemas y convertirse en un centro de excelencia y buenas prácticas.
- Crear un sitio de Intranet con una sección dedicada a "consejos y cómo se hace" que los propios usuarios puedan actualizar. Es importante que los usuarios sientan que forman parte y éste sitio a su vez puede proporcionarle a la ventanilla de atención una idea del tipo de problemas a los que se enfrentan los usuarios.

8. **Comenzar con proyectos pilotos a pequeña escala, de preferencia en un entorno auto-contenido con pocos usuarios**. Esto facilitará, entre otras cosas:

- Datos más ajustados de modelos de costo total de propiedad.
- La reacción de los usuarios, que se puede emplear para facilitar la introducción a otros sistemas.
- La validación o modificación de la arquitectura final y el caso de ejemplo.

9. **Decidir sobre la velocidad del proceso de migración una vez iniciado**. Estas son las principales alternativas:

- **Big bang:** Todos los usuarios cambian del viejo sistema al nuevo el mismo día. En la práctica, esto significa programar el cambio en un fin de semana o fiesta nacional. La ventaja es que no se necesitan disposiciones de doble acceso y el personal no se va a encontrar pasando constantemente de un sistema a otro. Entre las desventajas está el alto riesgo y la gran exigencia de recursos durante el cambio. Además cuenta con determinadas variantes a controlar, las que casi siempre fallan pero este fallo no necesariamente es del Software Libre, sino de la gestión, por esta razón se recomienda este tipo de migración en pequeñas instituciones.

- **Transición en grupos:** Se pasa a los usuarios del antiguo sistema al nuevo en grupos. Puede que los grupos funcionales completos se trasladen juntos para minimizar tener que compartir datos y los problemas de trabajo en el grupo. Se pueden contener los riesgos y gestionar los recursos eligiendo grupos del tamaño adecuado. También es posible hacer un cambio del hardware de las PC al mismo tiempo, reemplazando las máquinas en un grupo y luego instalando las sustituidas en lugar de las viejas máquinas del siguiente grupo.

- **Transición de usuario a usuario:** Básicamente la misma opción de la transición en grupos, pero con un grupo compuesto por una sola persona. Ese

método de "goteo" tiene escasos requisitos en cuanto a los recursos, pero no resulta eficaz ni apropiado para grandes instituciones. Pero sí puede ser una buena manera de ejecutar los proyectos pilotos.

Existe gran probabilidad que los viejos sistemas y los nuevos funcionen paralelamente durante algún tiempo. Es de sutil importancia contar con una estrategia de transición que permita que ambos sistemas funcionen juntos, de manera que las actividades de producción se puedan continuar correctamente durante el período de transición.

10. **Extender la migración a toda la Institución**. Esto implicará más formación de los usuarios y del personal técnico.

11. **Supervisar la respuesta de los usuarios y tomar nota de los problemas que surjan**. Algunas necesidades de los usuarios pueden ser tan poco claras que no se pueden detectar, ni descubrir, durante los proyectos piloto. Hay que asegurarse de que se dispone de recursos suficientes para hacer frente a esas necesidades tras la transición.

3.3. Guías de Migración a Software Libre

¿Qué es una guía de migración a Software Libre?
Esta no es más que un documento en el cual se exponen los diferentes puntos que se deben tener en cuenta para realizar la sustitución del sistema operativo propietario Windows que esté en cada uno de los servidores y las estaciones de trabajo. Además en la misma se establecen las etapas de desarrollo de dicho proceso y las tareas específicas para cada una de estas así como los responsables de cada una, igualmente los recursos que deberán ser asignados y el tiempo a emplear en su desempeño, garantizando un flujo de trabajo constante y eficiente.

3.3.1. Guías internacionales

Se realiza un estudio de determinadas Guías de Migración existentes en todo el mundo. En las guías de la Unión Europea, Venezuela y Perú se profundiza el análisis debido a sus características, la correcta definición del contenido y sus facilidades de entendimiento.

Otros ejemplos también importantes son las guías de migración de las universidades de los Andes en Venezuela y la universidad de Misiones en Argentina.

La universidad de los Andes, realiza sublimes aportes en cuanto a la divulgación, utilización y fortalecimiento del Software Libre, igualmente constituye una referencia necesaria para la búsqueda de soluciones ante determinados imprevistos.

Para un mejor entendimiento y organización se muestra la opinión de cada una de las guías antes mencionadas en las siguientes tres ***etapas***:

- *Preparación*
- *Migración*
- *Consolidación*

Etapa: Preparación

En la guía europea "Directrices IDA de migración a software de fuentes abiertas" (European Communities 2003) se define lo que sería la preparación de un proceso de este tipo, en 3 grandes fases:

1. Una fase de definición del proyecto y de recopilación de datos en la que se contemplen:
    - La arquitectura o arquitecturas de los sistemas,
    - Aplicaciones y sus datos asociados,
    - Protocolos y normas empleados,
    - Hardware,
    - El entorno físico,

- El ancho de banda de la red,
- La ubicación,
- Los requisitos sociales como el idioma o idiomas
- La capacitación del personal.

2. Una justificación de la migración, incluido el costo asociado a la misma.
3. Una o más fases piloto preparadas para probar si el plan y la justificación funcionan. Los datos de estas fases piloto pueden luego alimentar el modelo de costos usado en el plan.

Los peruanos en la "Guía para la migración de Software Libre en las entidades públicas de Perú" (Instituto Nacional de Estadísticas e Informática 2002) definen igualmente la preparación de este tipo de proceso en 3 grandes fases:

1. ***Planeamiento de la migración***. Etapa que corresponde a la planificación global, donde es necesario que tanto la alta dirección de las instituciones, las áreas de gestión informática y los usuarios responsables de los sistemas estratégicos de la institución, tomen conciencia de la importancia del plan de migración a Software Libre y además se genere el compromiso de apoyo de la alta dirección en la implementación.

   El plan de migración esta conformado por una serie de acciones agrupadas estratégicamente en etapas o fases, para lograr migrar los sistemas operativos, aplicaciones y herramientas informáticas que poseen las entidades públicas a Software Libre, con la participación activa del área informática y diferentes dependencias usuarias encargadas de generar, procesar, mantener y aplicar la información.

   Algo que resulta imprescindible y que propone esta guía para dar inicio al proceso de migración a Software Libre, consiste en efectuar dos tareas importantes: la sensibilización institucional respecto a la migración a Software Libre y la organización institucional para la implementación del Software Libre.

2. ***Capacitación y Diagnóstico de los sistemas de información de las entidades del estado.*** En esta etapa se registran todos los sistemas de información que posee la institución, a través de la cual se evaluará el grado de factibilidad para la migración a Software Libre mediante la verificación de:

- Las herramientas o aplicaciones equivalentes en GNU/LINUX,
- El grado de seguridad,
- Confiabilidad,
- Información técnica disponible
- Soporte que ofrecen las distribuciones más conocidas y recomendadas.

Lo anterior será logrado a través de un levantamiento o inventario informático que incluya:

- Inventario de los equipos informáticos de la institución.
- Inventario del software informático disponible en la institución.
- Inventario de los equipos empotrados.
- Inventario de los sistemas de información o aplicativos utilizados por la institución.

Y no debe faltar en esta etapa un inventario del conocimiento especializado del personal de informática existente en la institución, que serán los encargados de la implementación del plan de migración de los sistemas de información a Software Libre.

3. ***Alternativas de migración y capacitación***. En la determinación de alternativas de migración de los sistemas de información a Software Libre, será necesario analizar una serie de elementos, entre los que se encuentran:

- La disponibilidad presupuestal con la que cuenta la institución para la ejecución del proyecto de migración.
- La factibilidad total o parcial de migración a Software Libre de los sistemas de información existentes en la institución.
- El hardware que contiene a los sistemas de información.
- La cantidad y calidad de los recursos humanos disponibles

Se plantea que las alternativas de migración de los sistemas de información a Software Libre, dependen de:

- El sistema de información a migrar.
- La estructura del hardware que lo contiene.
- La dimensión de los sistemas de información que posee la institución.
- Los recursos humanos de la institución o terceros.

En esta etapa se elaboraran el Plan de acción y los Cronogramas de ejecución, los que comprenderán una estimación final de los costos implicados, y se insertarán en el Plan de Acción Institucional.

Se incluyen además la asignación de recursos humanos, soporte tecnológico y partida presupuestaria lo que será controlado por la alta dirección de la institución, que debe velar por ello para que se cumplan las actividades y cronogramas establecidos.

Por otra parte Venezuela en su "Plan Nacional de Migración a Software Libre en la Administración Pública Nacional" (Oficina de Tecnologías de Información 2005) define para esta etapa de preparación una determinada cantidad de tareas como:

1. Establecimiento de convenios para:
   - Garantizar formación y capacitación de los funcionarios antes, durante y después del proceso de migración.
   - Incluir el Software Libre y su filosofía en los planes de estudio de todas las enseñanzas.
   - Lograr incentivos financieros y fiscales para el apoyo de la industria del software nacional.
   - Garantizar la creación del Laboratorio Nacional de Software Libre, los semilleros de desarrolladores en Software Libre y los centros regionales de certificación.
2. Diseño de modelos replicables de enseñanza y aprendizaje en Software Libre.
3. Inicio de la campaña nacional para la difusión de la filosofía del Software Libre.
4. Diseño y aplicación del levantamiento de recursos informáticos en las entidades del

estado.

5. Publicación del plan nacional de migración, una vez aprobado por el presidente de la República.
6. Diseño y puesta en marcha del Portal de Software Libre.
7. Conformación de grupos de expertos y el diseño de trabajo de estos grupos.

Después de haber analizado la propuesta planteada en cada una de las guías para esta etapa de preparación se puede apreciar que existe coincidencia en varios aspectos aunque también tienen puntos de vistas diferentes.

En las Directrices Europeas se intenta definir el proceso desde el comienzo con la mayor cantidad de detalles posibles lo que facilita más información y mejor comprensión de lo que se desea realizar. Además eligen la justificación temprana del por qué llevar a cabo la migración. No obstante, ven la formación de los usuarios como algo ligero que debe efectuarse una vez que se haya migrado, es decir, que se enmarcan en la definición de estrategias para darle frente a las necesidades de los usuarios cuando choquen con el nuevo entorno.

Igualmente en Perú se elaboran conjuntamente con la administración de la institución, el plan de migración y solicitan a esta, que elaboren un plan de acción institucional para apoyar la migración dejando claro quienes son los responsables de cada tarea y el presupuesto para efectuarla y exigen que se revise su cumplimiento. Esta guía alerta sobre la necesidad de un levantamiento informático, de hardware, software e intelectual, para evaluar el grado de factibilidad de la migración y el conocimiento especializado del personal de informática existente en la institución a diferencia de las Directrices Europeas.

Diferente a los documentos anteriores analizados, que intentan empequeñecer el proceso de migración y hacerlo en el óptimo y menor tiempo posible, el Plan Nacional de Migración de Venezuela sugiere un proceso a largo plazo y en el que definitivamente

habrá que hacer ajustes, aunque este constituye un claro ejemplo de los sólidos pasos del proceso revolucionario venezolano. Esta propuesta ajusta su etapa de preparación en la introducción de la filosofía del Software Libre tanto en el sistema educacional como en la administración pública, en el lanzamiento de campañas de divulgación del mismo y la firma de convenios para garantizar todo lo anterior. También concuerdan con los peruanos en que debe hacerse un levantamiento informático y proponen la creación y puesta en funcionamiento de un portal para dar información y soporte a los usuarios.

Etapa: Migración

Esta etapa es nombrada en las guías analizadas de manera diferentes, por ejemplo las Directrices de la unión europea la llaman, despliegue del plan; en Perú, instalación, configuración y pruebas; y en Venezuela, simplemente migración.

Las Directrices Europeas son de la opinión de que los primeros cambios no deben incluir a la comunidad de usuarios sino que se realicen en los servidores los cuales serán los encargados de proporcionar la plataforma para la posterior introducción de estas transformaciones en el lado del cliente.

En la guía "Directrices IDA de migración a software de fuentes abiertas" (European Communities 2003) se proponen dos rutas a tener en cuenta para la migración.

RUTA 1

Añadir máquinas GNU/LINUX a los dominios Windows existentes e ir trasladando gradualmente los datos y los usuarios, y luego eliminar a los antiguos servidores propietarios. Es posible transferir a clientes y servidores independientemente. Añadir servidores al dominio Windows es uno de los modos más rápidos de sacar provecho del sistema libre.

Ejecutar clientes GNU/LINUX en un dominio Windows es una forma de coexistencia de

escaso riesgo, ya que no es necesario cambio alguno en relación a los servidores. Se puede usar donde un pequeño número de personas utilizarán escritorios con GNU/LINUX en un entorno sólo de Windows.

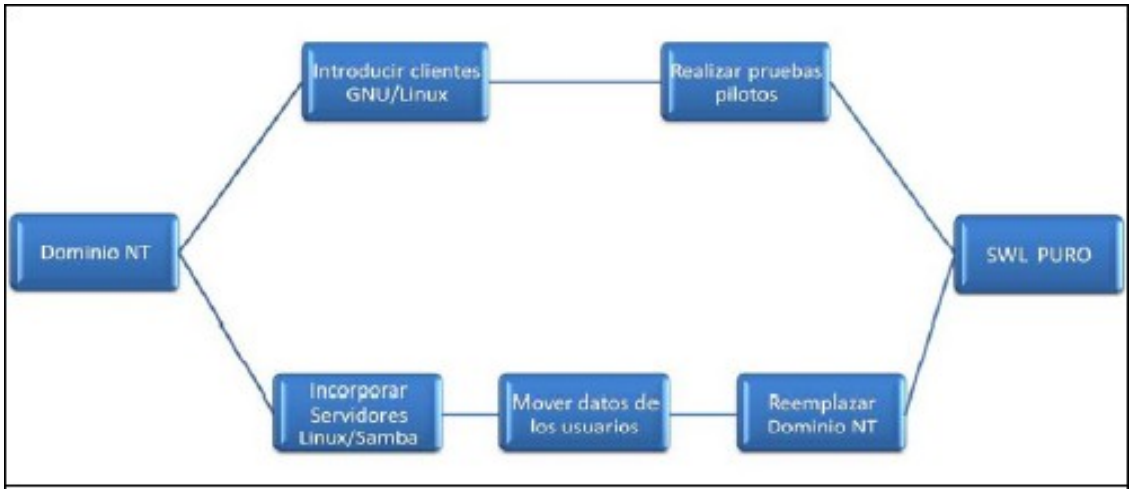


Fig. 4 "Ruta1 propuesta para del desarrollo de la etapa de migración"

RUTA 2

Construir una infraestructura paralela de tipo GNU/LINUX y transferir a los usuarios y sus datos en grupos, con mínima interacción entre el sistema antiguo y el nuevo.

Su criterio resalta que es mucho más sencillo que ejecutar un sistema mixto GNU/LINUX-Windows, pero crea una cooperación entre la gente que usa Windows y la que usa sistemas GNU/LINUX más difícil.

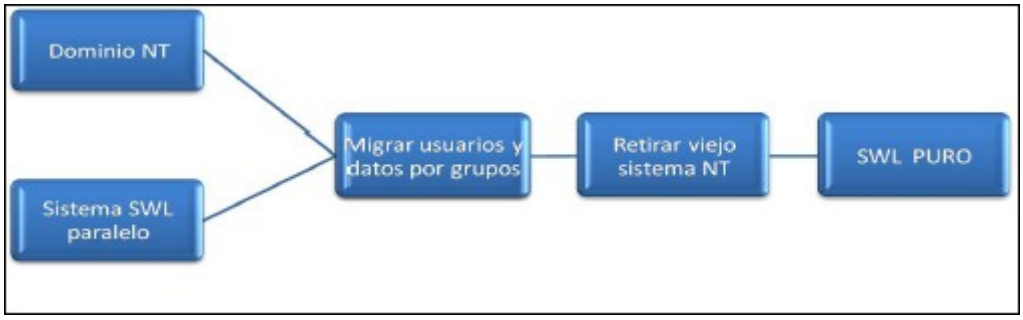


Fig. 5 "Ruta2 propuesta para del desarrollo de la etapa de migración"

A la par en "Guía para la migración de Software Libre en las entidades públicas de Perú" (Instituto Nacional de Estadísticas e Informática 2002), propone que los primeros cambios deben ser a nivel de servidores, indica que esta fase tendrá como objetivo la realización de la migración de acuerdo con lo planificado en la etapa anterior. La institución deberá proceder a la entrega de los recursos económicos que se pondrán en función de la migración, quedando por parte de los responsables del proceso, la definición del conjunto de procedimientos que deberán llevarse a cabo para evaluar e implementar cada una de las aplicaciones y herramientas que posee GNU/LINUX, así como, la incorporación del personal técnico, profesional o administrativo con responsabilidad sobre alguna de las tareas que se proponen a continuación para esta etapa:

1. **La instalación y configuración de servidores de comunicaciones y de bases de datos**: cuyo objetivo será realizar la implementación del sistema operativo GNU/LINUX y las aplicaciones específicas para cada servidor, dependiendo de la dimensión de la red de datos de la institución.

   Este sistema permite que los usuarios con más conocimientos del mismo, puedan configurarlo para trabajar de forma avanzada, y por consiguiente, ser utilizado no solamente a nivel de usuario, sino que puede ser configurado para trabajar de forma personal como servidor de comunicaciones, sin carecer de ninguna de las funcionalidades que poseen otros sistemas operativos propietarios. El sistema operativo GNU/LINUX esta configurado para trabajar como: Servidor Web (Apache),

Servidor FTP, Servidor de Correo Electrónico (Send Mail), Servidor de Red (Samba), Servidor DHCP, Servidor NFS, Servidor NIS, Servidor DNS (Bind), Servidor Proxy (Proxy Squid), Servidor Seguridad (Firewall Linux), Servidor de Impresión (Samba), Servidor de Administración y Monitoreo (SNMP Linux).

2. **Instalación del GNU/Linux tipo servidor**: que es más apropiada si se desea que el sistema funcione como un servidor de tipo GNU/LINUX y no se desea personalizarlo mucho.

3. **Instalación del sistema operativo para clientes**: que puede tomar determinadas variantes de acuerdo con la premisa, de que en el Software Libre el cambio es paulatino, en el sentido de que el usuario puede comparar las ventajas del Software Libre con el que normalmente utiliza en el ordenador. Además de que el Software Libre, posee interfaces gráficas, el uso de ventanas y la filosofía de operación, muy similares a la plataforma Windows; con la diferencia de que es más robusto, diseñado para grandes y pequeñas redes y para usuarios sin conexión a red.

La proposición planteada por Perú contempla la realización de pruebas de operatividad en los sistemas migrados. También resalta que la calidad en los sistemas puede ser medida por características externas, tales como que sea fácil de usar, fácil de implementar o de rápida ejecución; o por características internas, como el diseño modular.

Por otra parte Venezuela en su "Plan Nacional de Migración a Software Libre en la Administración Pública Nacional" (Oficina de Tecnologías de Información 2005) tiene en cuenta, en la etapa de migración, una serie de acciones que tributan a:

- La creación de un Laboratorio Nacional de Software Libre y Centros Regionales de Certificación.
- El inicio del apoyo a los grupos expertos creados en su etapa de preparación.
- La formalización de los documentos de normalización venezolanos: Estándares de

calidad, Gestión y evaluación de riesgos tecnológicos, que servirán de base al proceso de migración.

El diseño de la campaña de divulgación del Laboratorio Nacional y la continuación de la campaña nacional para la difusión de la filosofía libre, se incluyen en los planes de la guía Bolivariana, que pretende además, el monitoreo de (estadísticas de uso) y mantenimiento del portal de Software Libre.

La propuesta realizada por las Directrices Europeas no es la más apropiada pues ellas instan a utilizar la 2da ruta de migración, fundamentando que es mucho más fácil que ejecutar un sistema mixto GNU/LINUX /Windows. Pero no se puede obviar el factor de apoyo que debe brindar la institución al proceso, tanto en presupuesto como en gestión. Si no es completo o simplemente no se cuenta con este, entonces esa vía no es conveniente y obliga a decidirse por la 1ra ruta, cuya práctica ha demostrado, no ser del todo difícil.

Mientras que la propuesta peruana aborda de forma abstracta los mismos conceptos e ideas sobre el trabajo, primero con los servidores y luego la introducción paulatina del cambio en las terminales clientes. Pobremente, proponen la configuración de los servicios mediante discos de instalación, alejándose por completo de las posibles variantes ante factores que impliquen cambios sustanciales en el proceso.

Venezuela por su lado, a pesar, de arribar a la 2da fase de su migración, centra sus esfuerzos en la incesante divulgación de la filosofía del Software Libre, tema que no es nada errado pero que no ha de formar el eje central del proceso en una etapa tan avanzada como la migración de los sistemas.

Etapa: Consolidación

Para pasar a esta etapa ya se deben haber pasado a Software Libre todos los servidores y la mayoría de las PC de las estaciones de trabajo y la cantidad de Software Propietarios

sea mínima. Todo eso tiene un carácter normal y de alguna forma necesario para enfrentar cualquier eventualidad que se presente.

En "Directrices IDA de migración a software de fuentes abiertas" (European Communities 2003) se viene insistiendo desde tempranas etapas la introducción de aplicaciones libres en el entorno donde se utiliza Software Propietario para hacer mínimo el rechazo al cambio, además del establecimiento de estándares para el desarrollo de aplicaciones.

Estas también planean que una vez llegado a esta etapa, se establezcan con carácter definitivo las disposiciones que hasta la migración, se habían hecho solo a modo de sugerencias, lo cual implicará, insistir en que los desarrollos Web hechos que se produzcan utilizando un contenido que se pueda visualizar en todos los navegadores actuales de la Web, particularmente los navegadores de GNU/LINUX, pudiendo utilizarse herramientas como Weblint para comprobar la compatibilidad de las páginas Web.

Además sugieren:

- No fomentar el uso indiscriminado de macros y scripts en documentos y hojas de cálculo ya que de forma habitual los virus se valen de las macros y los scripts para infectar los sistemas.
- Insistir en el uso de formatos de archivos abiertos y estándar, como Proscript y PDF. Utilizar protocolos abiertos estándar.
- Desarrollar sistemas basados en por lo menos un modelo de tres niveles donde el código de aplicación es independiente de la interfaz humana y de los métodos de acceso a los datos.
- Insistir en que las nuevas aplicaciones se escriban de manera que sean portables, lo que incluye, usar lenguajes estandarizados portables como ANSI C, Java, Python y Perl, y usar sólo librerías multiplataforma y juegos de herramientas GUI, evitando lenguajes y APIs de arquitecturas específicas, además de la construcción de aplicaciones que requieran la presencia de otras aplicaciones propietarias.

La "Guía para la migración de Software Libre en las entidades públicas de Perú" (Instituto Nacional de Estadísticas e Informática 2002) no se queda detrás y para esta etapa de consolidación contempla la continuación de la capacitación del personal, priorizando el proceso de formación a:

- Funcionarios y directivos de la entidad: Lo que les permitirá difundir el impacto del Software Libre en las entidades del estado, así como las características, bondades y beneficios de este para las instituciones públicas. Esto implicará una mejor comprensión del Plan de migración, lo que tributará a un mejor apoyo y comprensión del plan.
- Personal informático: Con el objetivo de planificar, organizar, ejecutar y supervisar el proceso e implementar el Plan de migración así como la elaboración de la documentación del proceso.
- Usuarios del nuevo sistema: Que son los encargados de contribuir en la implementación del Software Libre en los ordenadores y operarán los nuevos sistemas de información GNU/LINUX.

Según esta guía, el equipo de capacitación empleará para llevar a cabo lo anterior expuesto las técnicas y medios de mayor facilidad, tales como:

- Reuniones de trabajo
- Cursos talleres,
- Exposiciones,
- Seminarios,
- Informes,
- Presentaciones,
- Multimedia,
- Videoconferencias
- Videos.

Esta etapa debe culminar según la guía peruana con la documentación de todos los procedimientos que se hayan llevado a cabo en el proceso de migración al Software Libre y la confección de los manuales correspondientes a cada uno de los niveles.

Por otra parte "Plan Nacional de Migración a Software Libre en la Administración Pública Nacional" (Oficina de Tecnologías de Información 2005) venezolano encierra para la etapa de consolidación

- La ejecución y puesta en marcha del Laboratorio Nacional de Software Libre, los grupos de desarrolladores de Software Libre y los centros regionales de certificación.
- El seguimiento, control y evaluación de procesos según los Planes de Implantación Progresiva del Software Libre desarrollado con Estándares Abiertos.
- La consolidación de una bitácora nacional de migración, con especial énfasis en los procesos exitosos, para publicarlos como casos referenciales. Continuar la Campaña Nacional para la difusión de la filosofía del Software Libre, el apoyo técnico de los grupos expertos y el monitoreo (estadísticas de uso) y mantenimiento del portal.
- La publicación de los parámetros o indicadores de gestión obtenidos antes, durante y al finalizar el proceso de migración de los órganos y entes de la Administración Pública Nacional.

Para la etapa de consolidación la propuesta de las Directrices Europeas es una alternativa a tener en cuenta pues implica la implementación con carácter obligatorio de los lineamientos que se habían estado realizando a modo de sugerencias, para evitar posibles barreras durante la migración. Es efectivo recordar que estos lineamientos se vienen aplicando desde la etapa de preparación, comprobándose durante el proceso su efectividad. También resulta enteramente acertada la idea de convertir las sugerencias en normas o principios y a su vez comenzar a exigir por el cumplimiento de estos. Todo para

evitar molestias recargadas a los usuarios y el rechazo a la plataforma libre.

Por otra parte las ideas peruanas para esta etapa no se deben ser descartadas totalmente, no obstante existen algunos planteamientos cuya efectividad radica en el momento más propicio para ser efectuados, estos deben ser analizados profundamente. También capacitar, es una tarea de gran importancia por lo que no se debe olvidar su introducción en todos los momentos posibles, pues es considerado la fuente principal para el surgimiento de los usuarios de la nueva plataforma. La capacitación no debe ser dejada para el final debido al cúmulo de acciones que encierra. Lo mismo ocurre con la acción de la redacción de la documentación.

Muchos piensan que porque en algún lugar algo está funcionando sin problemas y por eso deben hacer una copia fiel del mismo, no saben que están errados pues hay que tener en cuenta determinadas condiciones tecnológicas y sociales. Conllevando a la búsqueda de posibles soluciones, alternativas y variantes, las cuales deberán ser bien documentadas en el momento justo para no olvidar la información que con el paso del tiempo tiende a fraccionarse y se incurre en la redundancia ante determinadas trabas que puedan haber sido solucionadas en algún momento.

Para la etapa de consolidación la propuesta realizada en Venezuela ponen en marcha los centros desarrolladores de Software Libre que han sido preparados durante las 2 etapas anteriores, los que instituirán un notable e imprescindible impulso al desarrollo del proceso de migración. Resulta interesante la idea de publicar artículos donde se refleje el desarrollo alcanzado por los procesos en las diferentes entidades, así como colocar los problemas que se hayan presentado y la solución dada para que sirva de referencia para el resto, todo esto a partir de las facilidades brindadas por el Software Libre relacionadas con la distribución gratuita y la publicación de resultados que puedan ser mejorados y utilizados por todos.

Esta guía puede incrementar la cantidad de etapas o incrementar las tareas en cada una

de las propuestas en el mínimo tiempo posible, por lo que se recomienda un exhaustivo análisis de la misma.

3.3.2. Guía Nacional

La "Guía Cubana para la Migración a Software Libre" (Ramón Paumier Samón & Yoandy Pérez Villazón s.d.) contempla la migración mediante 4 Etapas o Fases y 6 Flujos de Trabajo, considerando como flujo de trabajo la secuencia de acciones, actividades o tareas utilizadas para la ejecución de un proceso, incluyendo el seguimiento del estado de cada una de sus etapas y el aporte de las herramientas necesarias para gestionarlo.

Las *ETAPAS* propuestas son:

- Preparación: Etapa en la que se realizarán las tareas de recopilación de datos y se lanzará una primera versión de la guía de migración.
- Migración Parcial: Etapa en la que se realizarán las pruebas y se validará la propuesta a pequeña escala, además de que tendrá gran actividad de trabajo.
- Migración Total: Cada vez que se ejecute una iteración de esta fase la cantidad de FLOSS (Free Libre Open Source Software) irá en aumento, será la etapa que marcará el fin del software privativo.
- Consolidación: Etapa que constituirá el soporte al proceso de migración, será el apoyo e indicará los niveles de éxito o fracaso de la Migración a Software Libre.

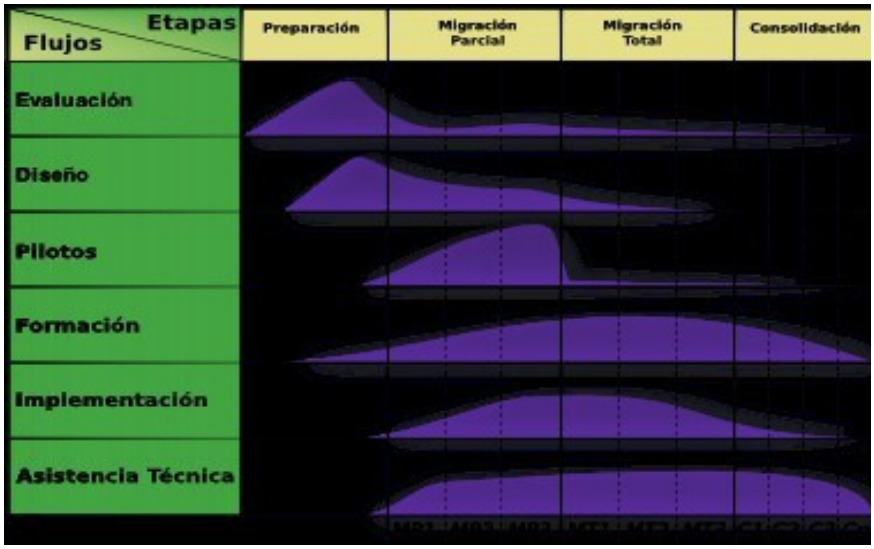


Fig. 6"Comportamiento de los flujos por etapas"

Los FLUJOS DE TRABAJO son:

- Evaluación: Hacer una evaluación de todos los procesos, tecnología y personal y adaptarlas al entorno actual.
- Diseño: Diseñar un plan de migración conforme a las necesidades, tomando como partida el resultado anterior.
- Pilotos: Poner en marcha el plan en un ambiente real de pruebas.
- Formación: Formación del personal y certificación del mismo por niveles de usuarios.
- Implementación: Instalación y migración definitiva de servicios y estaciones de trabajo a Software Libre.
- Asistencia y soporte técnico: Brindará atención y soporte a las infraestructuras, servicios instalados y al personal.

Esta guía cubana propone una determinada cantidad de tareas para llevar a cabo durante cada flujo, la misma puede servir de base para las instituciones del país. Además cada centro puede adaptarla a sus necesidades y de esta forma contribuir con las acciones que está emprendiendo nuestra Cuba.

También tras su análisis en ella se tienen en cuentan aspectos propuestos en guías

internacionales como las Directrices Europeas.

3.4. Estrategia para la migración a Software Libre para la disciplina "Ingeniería y Gestión de Software"

Después del análisis realizado de la Guía Cubana y apreciando la organización y distribución que plantean es válido tenerla en cuenta, o sea, tomarla como base para nuestra propuesta así como realizarle ajustes para adaptarla a nuestras necesidades.

La estrategia propuesta esta distribuida por etapas, flujos de trabajos y tareas específicas para cada uno de estos flujos.

3.4.1. Etapas

Las **Etapas** propuestas son: Preparación, Migración (Parcial y Total) y Consolidación.

1. *Preparación:* Etapa de estudio y recopilación de la información necesaria para el proceso de migración
2. *Migración Parcial*: Se pone en práctica la propuesta en una pequeña parte como prueba.
3. *Migración Total*: Es donde se eliminará totalmente el Software Propietario de todas las máquinas de los usuarios y los servidores y se le abrirán las puertas al Software Libre.
4. *Consolidación:* Etapa donde se brindará el soporte y apoyo al proceso así como se indicará el éxito y el fracaso del mismo.

El jefe de la disciplina conjuntamente con el jefe de carrera y los profesores deben definir el tiempo de ejecución de cada una de estas etapas sin olvidar la prioridad que requieren.

3.4.2. Flujos de Trabajo

Se plantearon los siguientes **Flujos de Trabajo:** Evaluación, Diseño, Formación, Implementación y Apoyo y soporte técnico.

1. *Evaluación:* Realizar una evolución de los objetivos instructivos de la disciplina Ingeniería y Gestión de Software así como las tecnologías empleadas en la misma para lograr una adaptación al nuevo entorno.
2. *Diseño:* Diseñar un plan de migración para cada asignatura de la disciplina partiendo de sus necesidades y de los resultados obtenidos durante la evaluación.
3. *Formación:* Preparación tanto de los profesores como los alumnos para los nuevos cambios.
4. *Implementación:* Instalación y migración total de cada herramienta y servicio en los laboratorios de los estudiantes y de los profesores.
5. *Apoyo y soporte técnico:* Brindará a estudiantes y profesores el apoyo y atención necesarias así como soporte y mantenimiento a las herramientas y servicios.

3.4.3. Tareas para cada flujo de trabajo

Flujo de Trabajo: Evaluación

1. Delimitar las prioridades de los profesores y los alumnos involucrados en el proceso durante la preparación.
2. Analizar el estado actual de los software utilizados en las asignaturas y del hardware de los laboratorios.
3. Evaluar las posibles soluciones disponibles para migrar los software existentes para cada asignatura de la disciplina y dictaminar cuales son los más factibles para el cumplimiento de los objetivos de cada una de ellas.
4. Evaluar los distintos mecanismos para facilitar el soporte y asistencia técnica.
5. Cuantificar las herramientas privadas a migrar.

Flujo de Trabajo: Diseño

1. Establecer un plan de acciones que abarquen todos los elementos a migrar.
2. Basados en las ventajas y desventajas del Software Libre elaborar la justificación del proceso de migración.
3. Definir la cantidad de los estudiantes y profesores que estarán involucradas en el

proceso así como el número de sistemas informáticos a migrar.

4. Se deben establecer períodos para la implementación y el soporte de las aplicaciones.
5. Gestionar el personal capacitado necesario para llevar a cabo el proceso así como los discos de las distribuciones a instalar y los repositorios de las mismas.
6. Recoger por medio de las diferentes vías que existen ya sean encuestas, conversaciones, revisión de los sistemas que se encuentren funcionando, etc. los elementos fundamentales para llevar a cabo las mejoras más convenientes a los puntos de la estrategia propuesta.
7. Hacer un continuo chequeo del plan de acciones propuestas para cada etapa.

Flujo de Trabajo: Formación

1. Elaborar e impartir guías de formación para los profesores y los alumnos:
   - Formación de especialistas de Software Libre capaces de explicar a profesores y estudiantes como se utilizan las herramientas propuestas.
   - Formación del Soporte Técnico para ser capaces de apoyar y ayudar técnicamente ya sea a profesores o estudiantes a través de cursos teóricos o prácticos así como el empleo de mecanismos adecuados para el uso del Software Libre y lograr un buen soporte.
   - Formación de los profesores y los estudiantes para que tras la instalación de las nuevas aplicaciones los mismos sean capaces de adquirir y recepcionar el máximo conocimiento sobre el Software Libre.
   - Formación a desarrolladores (profesores y estudiantes) cuyo fin es prepararlos mediante cursos prácticos o teóricos de programación en determinados lenguajes y de esta forma le den mantenimiento y desarrollo de aplicaciones futuras.
2. Otorgar certificados como muestra de su tránsito por los cursos impartidos así como establecer niveles entre los usuarios.

Flujo de trabajo: Implementación

1. Realizar la migración concretamente partiendo de los cambios respectivos de las tecnologías.
   - Sobre cualquier sistema operativo propietario (Windows u otro) instalar herramientas libres para que los profesores y los alumnos se vayan familiarizando con ellas.
   - Sustitución de los sistemas operativos propietarios (Windows u otro) por distribuciones del sistema operativo libre GNU/Linux el cual será establecido en los laboratorios tanto de los profesores como de los estudiantes.
2. Hacer con más firmeza los planes de formación y certificación de los estudiantes y profesores.
3. Fortalecer los mecanismos de soporte y mantenimientos.
4. Cambiar los datos y la información existentes a los nuevos sistemas y formatos.

Flujo de trabajo: Apoyo y soporte técnico

1. Brindar los repositorios de las aplicaciones y servicios actualizados periódicamente.
2. Implantar políticas de estándares a la hora de la creación de las aplicaciones, documentos, librerías gráficas y lenguajes de programación.
3. Las necesidades de los profesores y de los estudiantes así como de las aplicaciones más usadas se deberán someter a estudios para obtener mejoras en los servicios y optimización de las mismas a fin de incrementar su rendimiento.

3.4.4. Descripción por etapas

Etapa: Preparación

Esta etapa abarcara la planificación general donde lo más esencial es que todos los involucrados tengan noción lo que implica una migración, es decir, desde la más alta dirección de la facultad, los jefes de departamentos y de la carrera, los profesores reconozcan la importancia de la migración.

Se debe partir de la elaboración de un Plan de de Migración basados en las acciones de

la universidad y contando con el compromiso de la alta dirección para la implementación de los procesos.

El Plan de migración está conformado por una serie de acciones estratégicas para lograr migrar los sistemas operativos, aplicaciones y herramientas informáticas que poseen las entidades públicas a Software Libre, con la participación activa de todos los involucrados.

Resulta imprescindible según se plantea en la "Guía Cubana para la Migración a Software Libre" (Ramón Paumier Samón & Yoandy Pérez Villazón s.d.) que para llevar a cabo el proceso de migración a Software Libre, se deben realizar dos tareas importantes: la sensibilización institucional respecto a la migración a Software Libre y la organización institucional para la implementación del Software Libre.

Serán chequeadas:

- las herramientas o aplicaciones y sus equivalente en GNU/Linux,
- el grado de seguridad,
- la confiabilidad de la información,
- la información técnica disponible,
- el soporte que ofrecen las distribuciones más conocidas y recomendadas,
- los recursos humanos de que dispone la institución para el proceso.

Para lograr lo anterior planteado se realizara a través de un levantamiento o inventario informático que incluya:

- Inventario de los equipos informáticos de la institución.
- Inventario del software informático disponible en la institución.
- Inventario de los equipos empotrados.
- Inventario de los sistemas de información o aplicaciones utilizados por la institución.

En esta etapa además del plan de acciones se debe distribuir cuando se realizarán cada una mediante un cronograma que comprenderá los costos y asignación de soporte tecnológico.

Etapa: Migración Parcial

Esta etapa debe dirigirse a ir llevando a cabo la migración de forma paulatina,o sea, instalar sobre el sistema operativo Windows herramientas libres para una que tanto los profesores como los estudiantes se familiaricen con estas herramientas en un entorno conocido teniendo la posibilidad de capacitarlos en las mismas y la conversión de archivos a formatos estándares y evitar conflictos de compatibilidades entre las aplicaciones. Después de lograr la preparación en las herramientas libres que emplearan sobre GNU/Linux y las ventajas de su uso, entonces se puede establecer el nuevo sistema operativo libre en una parte lo que requiere la mayor atención de los implicados haciendo esto la parte más vulnerable de la migración.

Resulta importante tener en cuenta que puede que no existan equivalentes libres para determinadas aplicaciones propietarias pero para estos casos se pueden utilizar aplicaciones como Wine, que permiten emular software de Windows sobre GNU/Linux; unificar varias herramientas libres para lograr los mismos resultados que la propietaria o la re-programación de la misma.

Etapa: Migración Total

En esta etapa se lleva a cabo el mismo proceso que en la etapa anterior excepto que ya la migración se extiende a todos los laboratorios involucrados. Se convierte en una etapa compleja debido a la concentración de flujos de trabajo que están implicadas en la misma. Incluye un mayor número de personas, el cumplimiento exacto del cronograma de trabajo y la elaboración de la documentación de todo el proceso.

Se debe estar atento a cualquier eventualidad o irregularidad que se pueda presentar para poder recurrir a las soluciones más apropiadas. Es de vital importancia atender las opiniones de los usuarios y recopilar cualquier tipo de quejas o inconvenientes, para evitar

inconformidades y retraso del proceso.

Etapa: Consolidación

Durante el proceso de migración se ha podido detectar la necesidad de la formación de los estudiantes y profesores desde las primeras etapas y reiterarla aunque se concluya la misma para obtener una documentación detallada de todo lo que se ha llevado a cabo y el aprendizaje fuera lo más óptimo posible.

3.5. Conclusiones

En este capítulo se describen los aspectos de los cuales se debe partir para una buena migración. Además se realiza un análisis exhaustivo de las guías propuestas por las Directrices Europeas, el Plan Nacional de Venezuela y la Guía peruana, las cuales coinciden en determinados aspecto y a su vez cuentan con particularidades propias de cada una. También se propone la estrategia de migración para la disciplina Ingeniería y Gestión de Software de la carrera tomando como base la Guía cubana, de la cual se manifiestan los principales puntos tratados en la misma.

Conclusiones

Se realiza un estudio del empleo del Software Libre en las diferentes ramas de la educación, los beneficios y logros obtenidos tras el uso del mismo. Además las diferentes vías utilizadas para fomentar el interés en los estudiantes.

También se lleva a cabo un análisis de los objetivos propuestos para la disciplina Ingeniería y Gestión de Software del Plan de Estudios D para la carrera de Ingeniería Informática con el fin de proponer como alternativas herramientas libres factibles para cada una de sus asignaturas. Para esta propuesta se parte de la revisión de las características particulares de cada herramienta libre, las cuales son equivalentes a las propietarias que se encuentran en uso.

Se realiza un estudio de algunas de las estrategias de migración en los distintos ámbitos, este análisis incluye documentos como las Directrices IDA de la Unión Europea, el Plan Nacional de Migración de Venezuela y de las instituciones públicas de Perú así como la de Estremadura, etc. A fin de detectar los mejores adaptadas a la necesidad existente.

Se confecciona una estrategias para la migración de la disciplina Ingeniería y Gestión de Software como resultado del análisis de la documentación mencionada. Esta tomó como base la Guía Cubana. Para esta estrategia se diseñan cuatro etapas, cinco flujos de trabajos y varias tareas a realizar en cada uno de los flujos. Además se hace una descripción de cada etapa de la misma.

Resulta vital el vínculo del personal como la migración sin olvidar los pasos para este proceso. Es imprescindible la unión de todas las fuerzas es un solo objetivo pues con la organización de cada facultad y concientización de todos sus miembros se obtendrán resultados satisfactorios.

Recomendaciones

1. El estudio de nuevas guías, metodologías y estrategias a fin de introducir o suprimir fases en caso de ser necesario y mantener actualizada la que se propuso.

2. La implementación de la estrategia planteada como parte de la estrategia de migración de la Universidad de Cienfuegos.

3. Crear un grupo de soporte que permita el asesoramiento a los profesores y estudiantes en el uso de las herramientas propuestas.

4. Seguir la evaluación de nuevos software y actualizaciones de los propuestos para no quedar obsoleto.

5. Extender la estrategia propuesta a la disciplina Práctica Profesional.

Referencias Bibliógraficas

2004. Manual de SQL. Available at: http://walter.freeservers.com [Accedido Mayo 20, 2009].

Abdiel Alejandro Caballero Legrá, *Diseño de un Datamart y de los procesos ETL, para la toma de decisiones en las áreas de portadores energéticos y Transporte del MES*.

Alexander Dymo, 2004. KDevelop.

Ana María Delgado García & Rafael Oliver Cuello, La promoción del uso del software libre por parte de las universidades. Available at: http://www.um.es/ead/red/17.

Beatriz Busaniche , 2008. El Lado Oscuro del Voto Electrónico. Available at: http://www.criticadigital.com/impresa/index.php?secc=nota&nid=3007 [Accedido Enero 28, 2009].

Carlos Joa & Ninoska Cardona, 2006. Manual Glade y Anjuta.

Carlos Rojas, Kompozer: el arte del diseño web .

Desconocido, PgAdmin PostgreSQL Tools. Available at: www.pgadmin.org/ [Accedido Marzo 1, 2009].

Desconocido, PgDesigner. Available at: http://pgdesigner.sourceforge.net/en/index.html [Accedido Marzo 1, 2009].

Desconocido, Process Dashboard . Available at: http://processdash.sourceforge.net/pspdash.html [Accedido Marzo 1, 2009].

DiscoveryArticles, 2008. Grandes características de Adobe Photoshop - artículo. Available at: http://www.discoveryarticles.com/es/articles/175391/1/Great-Features-of-Adobe-Photoshop/Page1.html [Accedido Mayo 20, 2009].

Dr. Mario González Arencibia, Implicaciones éticas del software libre. Available at: www.eumed.net [Accedido Noviembre 28, 2008].

equipo de desarrollo de phpMyAdmin , 2003. php MyAdmin . Available at: www.phpmyadmin.net/ [Accedido Marzo 1, 2009].

European Communities, 2003. Directrices IDA de migración a software de fuentes abiertas. Available at: www.netproject.com.

fabFORCE.net, 2003. MySQL Workbench 5.0 - El DBDesigner 4 Sucesor. Available at: fabforce.net/dbdesigner4/ [Accedido Marzo 1, 2009].

Fernando Da Rosa & Federico Heinz, 2007. Guía Práctica sobre Software Libre, su selección y aplicación local en América Latina y el Caribre. Available at: http://intranet.mes.edu.cu/index.php?
option=com_remository&Itemid=27&func=startdown&id=1.

Ing. Eduardo Manuel Macías Sotolongo, Metodología de Migración a Software Libre para los
Centros Educacionales del Ministerio de Educación (MINED).

Instituto Nacional de Estadísticas e Informática, 2002. Guía para la migración de Software Libre en las entidades publicas de Perú.

Ivan Obeso Agüera, Estudio de herramientas CASE de análisis y diseño orientado a objetos.

Jesús M. González Barahona, & Joaquín Seoane Pascual, 2004. Software libre: licencias y propiedad intelectual.

Marcos Guglielmetti, 2008. Argentina: Software Libre y educación en el Año de la Enseñanza de las Ciencias - Definición - MasterMagazine. Available at: [Accedido Enero 28, 2009].

Mg. Jorge Hinestroza, 2003. División de Formación General de la Facultad de Ciencias, La Universidad del Zulia, Venezuela.

Microsoft Corporation, 2009. Características. Available at: http://www.microsoft.com/spain/windows/products/winfamily/ie/features.mspx [Accedido Mayo 20, 2009].

Ministerio de Educación Superior , 2007. Plan de Estudio D Ingeniería Informática Presencial.

Ministerio de Educación Superior: Dirección de Informatización , 2008. Estrategia Maestra de Informatización.

Montserrat Culebro Juárez, 2006. Software libre vs software propietario Ventajas y desventajas.

Mozilla Europe y Mozilla Foundation , Características de Firefox, suficientes cosas buenas como para cambiar la forma en la que usas la web. Available at: http://www.mozilla-

europe.org/es/firefox/features/ [Accedido Mayo 20, 2009].

Nsmlinux, 2006. “El software libre en la educación – Experiencias”. Available at: http://es.wikibooks.org/wiki/El_software_libre_en_la_educaci%C3%B3n .

Oficina de Tecnologías de Información, 2005. Plan Nacional de Migración a Software Libre en la Administración Pública Nacional.

Olivier Sessink , 2008. Características de Bluefish. Available at: http://bluefish.openoffice.nl/features.html&prev=/search%3Fq%3Dbluefish%26hl%3Des%26sa%3DG&usg=ALkJrhhETvwvetrc7a-hZWZjBL0k4-NqXw [Accedido Marzo 1, 2009].

Orlando Cárdenas Fernández, Implicaciones Sociales del Software Libre , Universidad de las Ciencias Informáticas , La Habana.

Ramón Paumier Samón, 2007. *Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI).*

Ramón Paumier Samón & Yoandy Pérez Villazón, Guía cubana de migración a Software Libre.

Raúl G Torricella Morales & Francisco Lee Tenorio, Acceso abierto y software libre: premisas para la independencia tecnológica. , Vol. 17 Issue 2, p33-44. http://search.ebscohost.com/login.aspx?direct=true&db=zbh&AN=33007639&lang=es&site=ehost-live

Software Libre Maxorata, 2008. Más de 50 millones de estudiantes brasileños utilizarán tecnologías abiertas a partir de 2009. Available at: http://www.gnumax.net.

Sun Microsystems, 1994. Conozca a NetBeans. Available at: http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html [Accedido Mayo 20, 2009].

Yahoo, 2009. Cuáles son las características de Microsoft Office 2007. Available at: http://espanol.answers.yahoo.com/question/index?qid=20080316155106AAlGoyM [Accedido Mayo 20, 2009].

Bibliografía

.

2004. Manual de SQL. Available at: http://walter.freeservers.com [Accedido Mayo 20, 2009].

Abdiel Alejandro Caballero Legrá, *Diseño de un Datamart y de los procesos ETL, para la toma de decisiones en las áreas de portadores energéticos y Transporte del MES*.

Administrdor, 2008. Proteger la Soberanía nacional con software libre . Available at: http://www.csol.org/?q=node/54 [Accedido Enero 28, 2009].

Alexander Dymo, 2004. KDevelop.

Ana María Delgado García & Rafael Oliver Cuello, La promoción del uso del software libre por parte de las universidades. Available at: http://www.um.es/ead/red/17.

Andreas Viklund. , 2006. Lista de herramientas libres para Ingeniería « Asociación Tumbesina de Software Libre. Available at: http://atusol.wordpress.com/2006/09/28/lista-de-herramientas-libres-para-ingenieria/ [Accedido Diciembre 11, 2008].

Beatriz Busaniche , 2008. El Lado Oscuro del Voto Electrónico. Available at: http://www.criticadigital.com/impresa/index.php?secc=nota&nid=3007 [Accedido Enero 28, 2009].

Carlos Cañedo Iglesias, 2008. Estrategia Maestra de Informatización de la Universidad de Cienfuegos “Carlos Rafael Rodríguez”.

Carlos Joa & Ninoska Cardona, 2006. Manual Glade y Anjuta.

Carlos Rojas, Kompozer: el arte del diseño web .

Daniel Pagés Chacón , 2006. *Sistema de Gestion de la Nueva Universidad* . Instituto Superior Técnico "José Antonio Ecchevarría".

Daniel F Moisset & Federico Heinz , 2004. Uso de software libre en el gobierno. Available at: http://proposicion.org.ar/doc/freegov-faq.html [Accedido Marzo 1, 2009].

Desconocido, Cuba otro país listo para el Software Libre Principal.

Desconocido, El Software Libre en el mundo de la seguridad. Available at: http://mmc.geofisica.unam.mx/LuCAS/Presentaciones/200103hispalinux/ferrer/html/sw-libre.html [Accedido Diciembre 11, 2008].

Desconocido, 2006. Guía de migración y despliegue - Software libre para los países en desarrollo. Available at: [Accedido Noviembre 28, 2008].

Desconocido, PgAdmin PostgreSQL Tools. Available at: www.pgadmin.org/ [Accedido Marzo 1, 2009].

Desconocido, PgDesigner. Available at: http://pgdesigner.sourceforge.net/en/index.html [Accedido Marzo 1, 2009].

Desconocido, Process Dashboard . Available at: http://processdash.sourceforge.net/pspdash.html [Accedido Marzo 1, 2009].

DiscoveryArticles, 2008. Grandes características de Adobe Photoshop - artículo. Available at: http://www.discoveryarticles.com/es/articles/175391/1/Great-Features-of-Adobe-Photoshop/Page1.html [Accedido Mayo 20, 2009].

Dr. Edgar David Villanueva Nuñez, 2002. Señor:Juan Alberto González, Gerente General de Microsoft del Perú. Available at: http://www.gnu.org.pe/rescon.html [Accedido Marzo 1, 2009].

Dr. Mario González Arencibia, Implicaciones éticas del software libre. Available at: www.eumed.net [Accedido Noviembre 28, 2008].

Dr. Ricardo J Castello, 2005. Software Libre Modelo de factibilidad económica - financiera .

eiraworks , 2006. Migración a Software Libre - eiraworks. Available at: http://www.eiraworks.com/ [Accedido Noviembre 28, 2008].

equipo de desarrollo de phpMyAdmin , 2003. php MyAdmin . Available at: www.phpmyadmin.net/ [Accedido Marzo 1, 2009].

European Communities, 2003. Directrices IDA de migración a software de fuentes abiertas. Available at: www.netproject.com.

fabFORCE.net, 2003. MySQL Workbench 5.0 - El DBDesigner 4 Sucesor. Available at: fabforce.net/dbdesigner4/ [Accedido Marzo 1, 2009].

Fernando Da Rosa & Federico Heinz, 2007. Guía Práctica sobre Software Libre, su selección y aplicación local en América Latina y el Caribre. Available at: http://intranet.mes.edu.cu/index.php?option=com_remository&Itemid=27&func=startdown&id=1.

Grupo de Desarrollo Global de PostgreSQL , 1996. Sobre PostgresSQL. Available at: http://www.postgresql.org/about/ [Accedido Marzo 1, 2009].

Hypatia, 2003. Usos de Software Libre en los Estados de la Región Centro y Sud Americana. Available at: http://www.hipatia.info.

Ing. Eduardo Manuel Macías Sotolongo, Metodología de Migración a Software Libre para los Centros Educacionales del Ministerio de Educación (MINED).

Instituto Nacional de Estadísticas e Informática, 2002. Guía para la migración de Software Libre en las entidades publicas de Perú.

Ivan Obeso Agüera, Estudio de herramientas CASE de análisis y diseño orientado a objetos.

Javier García Diz, 2006. La migración a software libre. The soft revolution. Available at: www.ladinamo.org [Accedido Noviembre 28, 2008].

Jesús M. González Barahona, & Joaquín Seoane Pascual, 2004. Software libre: licencias y propiedad intelectual.

Jose Aguilar, El Software Libre: un nuevo fenómeno informático.

Julio Cabero Almenara & M.C. LLorente Cejudo, Software Libre y sus posibilidades en la educación.

La provincia cubana de Holguín inicia migración al Software Libre . Available at: http://www.ceslcam.com/index.php?option=com_content&view=article&id=173:la provincia-cubana-de-holguin-inicia-migracion-al-software-libre&catid=82&Itemid=265 [Accedido Marzo 1, 2009].

Manual de Umbrello,Capítulo1.Introducción. Available at: http://docs.kde.org/stable/es/kdesdk/umbrello/introduction.html [Accedido Marzo 1, 2009].

Marcos Guglielmetti, 2008. Argentina: Software Libre y educación en el Año de la Enseñanza de las Ciencias - Definición - MasterMagazine. Available at: [Accedido Enero 28, 2009].

Marianela Diaz Rosales & Liané Díaz Boza, 2007. *Módulo de Recuperación Web de Información Docente del Sistema de Gestión de la Nueva Universidad SIGENU*. Instituto Superior Politécnico “José Antonio Echeverría”.

Mauricio Nunes, 2006. Experiencias I.U.T.CARIPITO Primeras Experiencias de Nuestra Migración a Software Libre. Available at: http://www.iutcaripito.tec.ve/migracion/experiencias.html [Accedido Enero 8, 2009].

Mg. Jorge Hinestroza, 2003. División de Formación General de la Facultad de Ciencias, La Universidad del Zulia, Venezuela.

Microsoft Corporation, 2009. Características. Available at: http://www.microsoft.com/spain/windows/products/winfamily/ie/features.mspx [Accedido Mayo 20, 2009].

Ministerio de Educación Superior: Dirección de Informatización , 2008. Estrategia Maestrade Informatización.

Ministerio de Educación Superior , 2007. Plan de Estudio D Ingeniería Informática Presencial.

Montserrat Culebro Juárez, 2006. Software libre vs software propietario Ventajas y desventajas.

Mozilla Europe y Mozilla Foundation , Características de Firefox, suficientes cosas buenas como para cambiar la forma en la que usas la web. Available at: http://www.mozilla-europe.org/es/firefox/features/ [Accedido Mayo 20, 2009].

Nsmlinux, 2006. “El software libre en la educación – Experiencias”. Available at: http://es.wikibooks.org/wiki/El_software_libre_en_la_educaci%C3%B3n .

Oficina de Tecnologías de Información, 2005. Plan Nacional de Migración a Software Libre en la Administración Pública Nacional.

Olivier Sessink , 2008. Características de Bluefish. Available at: http://bluefish.openoffice.nl/features.html&prev=/search%3Fq%3Dbluefish%26hl%3Des%26sa%3DG&usg=ALkJrhhETvwvetrc7a-hZWZjBL0k4-NqXw [Accedido Marzo 1, 2009].

Orlando Cárdenas Fernández, Implicaciones Sociales del Software Libre ,Universidad de las Ciencias Informáticas, La Habana.

PgAdmin. Available at: http://www.pgadmin.org/ [Accedido Marzo 1, 2009].

Rafael Oliver Cuello & Ana M. Delgado García, 2006. “Algunas experiencias en la utilización de software libre en la educación superior”. Available at: http://www.cibersociedad.net/congres2006 [Accedido Marzo 14, 2009].

Ramón Paumier Samón, 2007. *Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI).*

Ramón Paumier Samón & Yoandy Pérez Villazón, Guía cubana de migración a Software Libre.

Raúl G Torricella Morales & Francisco Lee Tenorio, Acceso abierto y software libre: premisas para la independencia tecnológica, Vol. 17 Issue 2, p33-44.http://search.ebscohost.com/login.aspx?direct=true&db=zbh&AN=33007639&lang=es&site=ehost-live

Rene Merou, Mapa conceptual del Software Libre. Available at: http://danubuntu.wordpress.com/2009/03/10/mapa-conceptual-del-software-libre/ [Accedido Mayo 19, 2009].

Software Libre Maxorata, 2008. Más de 50 millones de estudiantes brasileños utilizarán tecnologías abiertas a partir de 2009. Available at: http://www.gnumax.net.

Sun Microsystems, 1994. Conozca a NetBeans. Available at: http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html [Accedido Mayo 20, 2009].

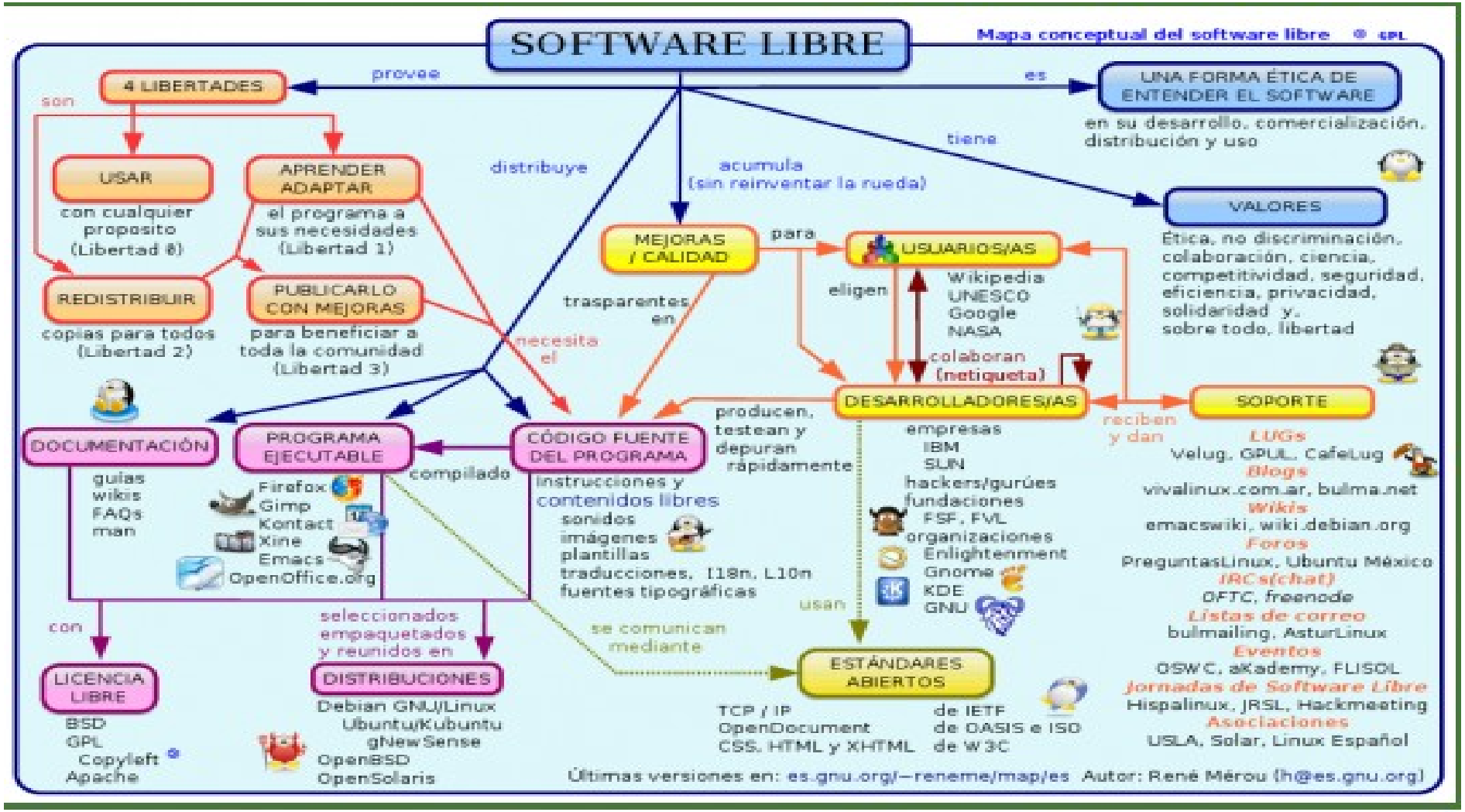
Sun Microsystems , Sun Microsystems Conozca el nuevo NetBeans. Available at: http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html [Accedido Mayo 3, 2009].

Yahoo, 2009. Cuáles son las características de Microsoft Office 2007. Available at: http://espanol.answers.yahoo.com/question/index?qid=20080316155106AAlGoyM [Accedido Mayo 20, 2009].

Yudivián Almeida Cruz, 2007. Plan de migración a Software Libre de la Universidad de La Habana . Available at: http://profesores.matcom.uh.cu/%7Eyudy/blog/feeds/index.rss2 [Accedido Noviembre 28, 2008].

Yudivián Almeida Cruz , 2008. Ecuador decreta migrar a Software Libre . Available at: [Accedido Noviembre 28, 2008].

Anexo I


SOFTWARE LIBRE
Mapa conceptual del software libre
provee
es
4 LIBERTADES
son
UNA FORMA ÉTICA DE ENTENDER EL SOFTWARE
en su desarrollo, comercialización, distribución y uso
USAR
con cualquier propósito (Libertad 0)
APRENDER ADAPTAR
el programa a sus necesidades (Libertad 1)
distribuye
acumula (sin reinventar la rueda)
tiene
VALORES
Ética, no discriminación, colaboración, ciencia, competitividad, seguridad, eficiencia, privacidad, solidaridad y, sobre todo, libertad
MEJORAS / CALIDAD
para
USUARIOS/AS
Wikipedia
UNESCO
Google
NASA
eligen
REDISTRIBUIR
copias para todos (Libertad 2)
PUBLICARLO CON MEJORAS
para beneficiar a toda la comunidad (Libertad 3)
trasparentes en
necesita el
colaboran (netiqueta)
DESARROLLADORES/AS
SOPORTE
producen, testean y depuran rápidamente
reciben y dan
DOCUMENTACIÓN
guías
wikis
FAQs
man
PROGRAMA EJECUTABLE
Firefox
Gimp
Kontact
Xine
Emacs
OpenOffice.org
compilado
CÓDIGO FUENTE DEL PROGRAMA
instrucciones y contenidos libres
sonidos
imágenes
plantillas
traducciones, I18n, L10n
fuentes tipográficas
empresas
IBM
SUN
hackers/gurúes
fundaciones
FSF, FVL
organizaciones
Enlightenment
Gnome
KDE
GNU
LUGs
Velug, GPUL, CafeLug
Blogs
vivalinux.com.ar, bulma.net
Wikis
emacswiki, wiki.debian.org
Foros
PreguntasLinux, Ubuntu México
IRCs(chat)
OFTC, freenode
Listas de correo
bulmailing, AsturLinux
Eventos
OSWC, aKademy, FLISOL
Jornadas de Software Libre
Hispalinux, JRSL, Hackmeeting
Asociaciones
USLA, Solar, Linux Español
usan
se comunican mediante
con
seleccionados empaquetados y reunidos en
LICENCIA LIBRE
BSD
GPL
Copyleft
Apache
DISTRIBUCIONES
Debian GNU/Linux
Ubuntu/Kubuntu
gNewSense
OpenBSD
OpenSolaris
ESTÁNDARES ABIERTOS
TCP / IP
OpenDocument
CSS, HTML y XHTML
de IETF
de OASIS e ISO
de W3C
Últimas versiones en: es.gnu.org/~reneme/map/es Autor: René Mérou (h@es.gnu.org)

Anexo II

La siguiente lista de programas una muestra de equivalentes libres para los software privativos.

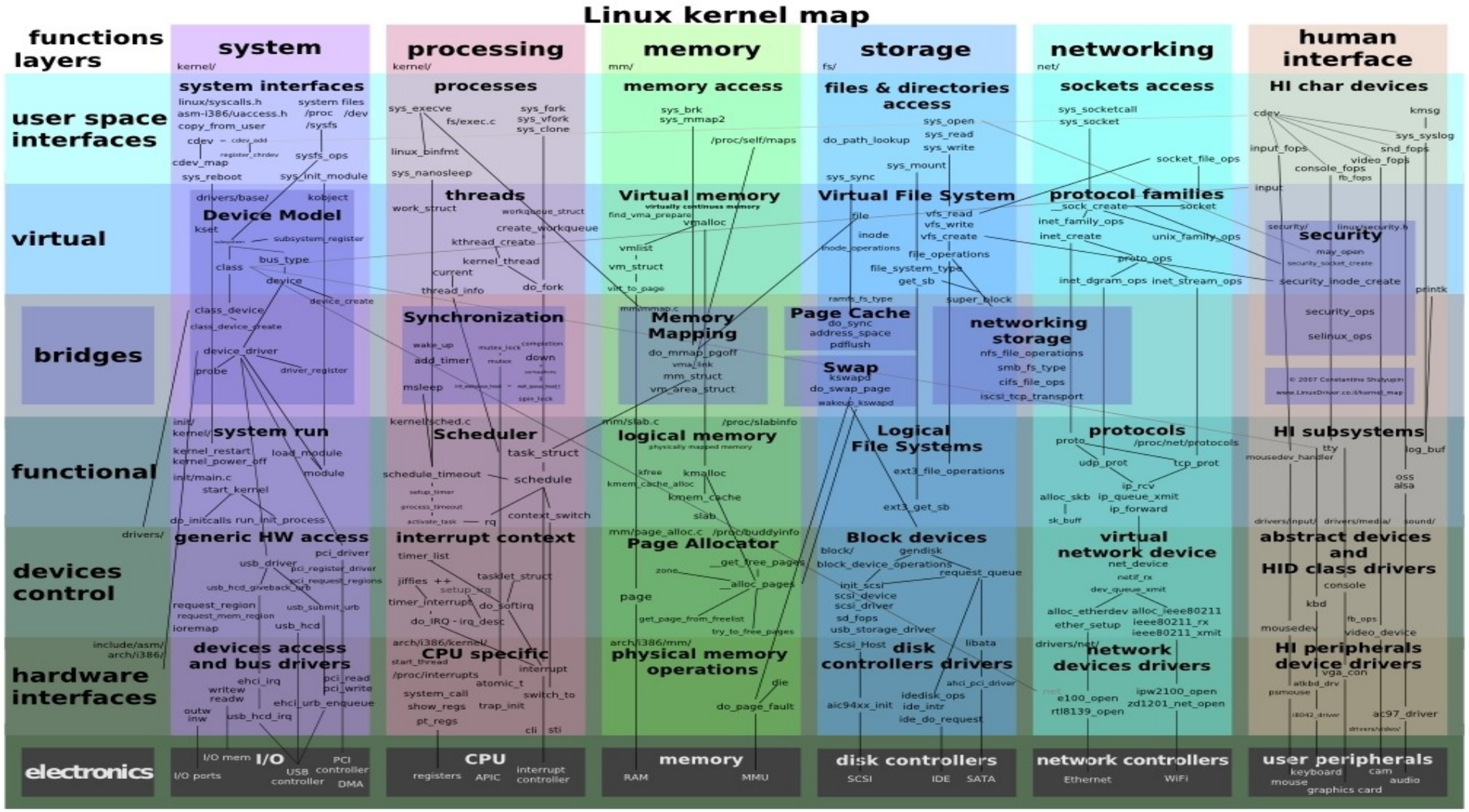
No es una lista exhaustiva, ni final, es simplemente una introducción a la amplia gama de opciones que ofrece el Software Libre para centros educacionales y desarrolladores aunque todos los programas abajo mencionados no han sido licenciados bajo la licencia GPL, todos respetan las cuatro libertades que definen al Software Libre.

Descripción del programa, tareas ejecutadas	***Windows***	***Linux***
Redes y Conectividad.		
Navegadores Web	Internet Explorer, Netscape / Mozilla for Windows, Opera, , ...Phoenix for Windows	1) Netscape / Mozilla. 2) Galeon. 3) Konqueror. 4) Opera. [Prop] 5) Phoenix. 6) Nautilus. 7) Epiphany. 8) Links. (with "-g" key). 9) Dillo. (Parches lenguage Ruso - aquí).
Sistemas de Software para Escritorio.		
Editor de Texto	Notepad, WordPad, TextPad, ...	1) Kedit (KDE). 2) Gedit (Gnome). 3) Gnotepad. 4) Kate (KDE). 5) KWrite (KDE). 6) Nedit. 7) Vim. 8) Xemacs. 9) Xcoral. 10) Nvi.
Visualizador de PDF	Adobe Acrobat Reader	1) Acrobat Reader para Linux. 2) Xpdf. 3) GV.
Offimática		
Paquetes de oficina	MS Office, StarOffice / OpenOffice, 602Software	1) Openoffice. 2) Staroffice. [Prop] 3) Koffice. 4) HancomOffice. [Prop]

		5) Gnome Office. 6) Applixware Office. 7) Siag Office. 8) TeX, LaTeX, ...
Procesador de Word	Word, StarOffice / OpenOffice Writer, 602Text	1) Abiword. 2) WordPerfect. 3) Ted. 4) StarOffice / OpenOffice Writer. 5) Kword. 6) LyX. 7) Kile (KDE Integrado a LaTeX).
Hojas de Cálculo	Excel, StarOffice / OpenOffice Calc, 602Tab	1) Gnumeric. 2) Abacus. 3) StarOffice / OpenOffice Calc. 4) Kspread.
Dibujo de Gráficos	Excel, ...	1) Kivio. 2) Dia. 3) KChart. 4) xfig. 5) Gnuplot. 6) GtkGraph.
Creación de Presentaciones	MS PowerPoint, StarOffice Presentation, OpenOffice Impress	1) StarOffice Presentation. 2) OpenOffice Impress. 3) Kpresenter. 4) MagicPoint. 5) Kuickshow & gimp :).
Bases de Datos Locales	Access	1) KNoda. 2) Gnome DB Manager. 3) OpenOffice + MySQL. 4) InterBase. 5) BDB. 6) Rekall. [Prop] 7) StarOffice Adabase.
Programación y Desarrollo.		
Visual C++ IDE	Borland C++ Builder, MS Visual C	1) Anjuta + Glade + Devhelp. 2) KDE Studio Gold. [Prop] 3) Dev-C++. 4) Kylix. [Prop] (Kylix la edición personal es libre). 5) vtkBuilder. 6) foxBuilder. 7) wxDesigner. 8) Arriba. [Prop] 9) Code Crusader.

		[Prop] 10) CodeWarrior. [Prop] 11) Gbuilder. 12) Source Navigator. 13) TimeStorm. [Prop] 14) Understand for C++. [Prop] 15) SlickEdit. [Prop] 16) Vide. 17) Kdevelop 18) Eclipse.
Bases de Datos (Plataforma)	SQL Server, PostgreSQL, MySQL, Oracle	1) PostgreSQL 2) MySQL 3) Oracle [Prop]
Bases de Datos (Diseño)	Erwin	1) DBDesigner 2) PgDesigner 3) Gerwin 4) Dia + Code
Bases de Datos (Administración)	SQL Server	1) Pgaccess(Postgre) 2)pgadmin3(Postgre) 3) phpmyadmin (MySQL) 4) webmin 5) EMS postgres Manager[Prop]
Editores HTML	Dreamweaver, FrontPage	1)Quanta Plus. 2) Bluefish. 3) WebMaker. 4) Screem. 5) Toppage. 6) WebDesigner. 7) ScriptEditor. 8) August. 9) Kompozer
IDE de Java	JBuilder	Link: Java Tools for Linux. 1) Jbuilder äëÿ Linux. 2) NetBeans. 3) Eclipse. 4) Sun ONE Studio. [formerly Forte] 5) Vide.
Ingeniería de Software	Rational Rose.	1) Umbrello UML Modeller. 2) Dia+Dia2Code. 3) PoceidonCE (community edition). 4) ArgoUML. 5) Together ControlCenter [Prop] 6)Rational Rose [Prop] 7) Visual Paradim [Prop]

Anexo III


Linux kernel map
functions layers
system
processing
memory
storage
networking
human interface
user space interfaces
virtual
bridges
functional
devices control
hardware interfaces
electronics
system interfaces
processes
memory access
files & directories access
sockets access
HI char devices
Device Model
threads
Virtual memory
Virtual File System
protocol families
security
Synchronization
Memory Mapping
Page Cache
networking storage
Swap
system run
Scheduler
logical memory
Logical File Systems
protocols
HI subsystems
generic HW access
interrupt context
Page Allocator
Block devices
virtual network device
abstract devices and HID class drivers
devices access and bus drivers
CPU specific
physical memory operations
disk controllers drivers
network devices drivers
HI peripherals device drivers
I/O
CPU
memory
disk controllers
network controllers
user peripherals

Anexo IV

La Federación Estudiantil Universitaria

Otorga el presente **RECONOCIMIENTO**

A: "Estrategia para la introducción del Software libre en la disciplina Ingeniería y gestión de Software del Plan de estudio D para la carrera de Ingeniería Informática"

Por haber obtenido la categoría de Relevante **en el**

Evento Joven Ciencia 2009

Por ayudarnos a enriquecer la actividad científica de nuestra Organización.

Dado a los 7 días del mes de mayo del 2009

Año del 50 Aniversario del Triunfo de la Revolución

Dr. Juan B. Cogollos Martínez
Rector UCf

Oslién Ramírez González
Presidente FEU UCf

Leonardo Cortina González
Docencia- Historia e Investigación