



FACULTAD 8

Título: “Estrategia para la migración a software libre de la producción de software educativo”

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

Autor: Dania Marrero Quesada

Tutor: Ing. Abel E. Lorente Rodríguez

Ciudad de la Habana, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

"Es intentando lo imposible como se realiza lo posible."

Henri Barbusse

"Después de escalar una montaña muy alta, descubrimos que hay muchas otras montañas por escalar."

Nelson Mandela

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de ella para lo que necesite, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2009.

Autores: Dania Marrero Quesada

Tutor: Ing. Abel E. Lorente Rodríguez

Autor(es):

Nombre: Dania Marrero Quesada

Correo Electrónico: dquesada@estudiantes.uci.cu

Tutor(es):

Nombre: Ing. Abel E. Lorente Rodríguez

Correo Electrónico: abel@uci.cu

Le dedico todos mis esfuerzos en el desarrollo de esta tesis:

A mis padres por su amor, su comprensión y su apoyo en todo momento, gracias por estar siempre conmigo y por ser mi guía a seguir en la vida, los quiero y los admiro mucho.

A mi hermano por su cariño, por su amistad y por confiar siempre en mí, gracias por entenderme y por ser el mejor hermano del mundo.

A mis abuelos por su apoyo y cariño, gracias por creer en mí.

A la Revolución cubana por haberme dado la oportunidad de ser ingeniera.

AGRADECIMIENTOS

Le agradezco la ayuda y la confianza depositada en mí:

A mi tutor por brindarme apoyo y permitirme conocer el mundo del software libre.

A mis profesores por haberme guiado en el camino del conocimiento.

A mi oponente sus sugerencias y criterios los cuales han enriquecido mi formación como ingeniera

A Deymis y Daynis por haber puesto a mi disposición sus vastos conocimientos en metodología de la investigación

A Roberto y Yonnys por su preocupación y ayuda

A mis padres por seguir alentándome cuando me invadía el espíritu derrotista.

La voluntad del país de migrar hacia software libre (SWL) es una vía al desarrollo independiente de la informática. Es necesario ir eliminando todas las dependencias y ataduras, de las tecnologías privativas en la producción de software educativo (SWE). La UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS (UCI) está implicada en este proceso de migración por ser una de las empresas productoras de software que actualmente aporta más ingresos al país. Siguiendo esta voluntad, los Ministerios de Educación y Educación Superior también ha tomado medidas para alinearse y adoptar el software libre en los centros educativos.

Cuba es una de las mayores potencias en la educación. En concordancia con esto ha sido la producción de software educativo una de las líneas de desarrollo estratégicas para la UCI y solo una estrategia de migración permitirá fiabilidad en el cambio de las herramientas que propone este proceso.

Para el desarrollo de esta estrategia se hizo necesario estudiar las herramientas con las que se producía este software y un grupo de guías de migración que tenían puntos de coincidencia entre ellas; pero mostraron algunas deficiencias o pasos que no estaban acorde a esta área productiva. Se llegó a la conclusión de que había que crear una estrategia apoyándose en el estudio de las mismas, pero con características muy propias. Se determinó como metodología a seguir, realizar un proceso evolutivo e iterativo que transitara por 4 fases y 6 flujos de trabajo. Se designa un grupo de migración con roles para realizar las actividades de cada flujo de trabajo.

PALABRAS CLAVES: Multimedia, software educativo (SWE), software libre (SW)

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 EL SOFTWARE LIBRE	5
1.1.1 ¿Qué es el software libre?.....	5
1.1.2 Unix	6
1.1.3 El Proyecto GNU.....	7
1.1.4 Linux.....	8
1.1.5 Open Source.....	9
1.2 CONSIDERACIONES PARA ADOPTAR SOFTWARE LIBRE	10
1.2.1 Nivel de seguridad proporcionado	10
1.2.2 Recopilación del Kernel.....	11
1.2.3 Configuración de archivos	11
1.2.4 Independencia tecnológica.....	12
1.2.5 Costes	12
1.2.6 Requerimientos de hardware	13
1.3 EL SOFTWARE LIBRE EN CUBA	13
1.3.1 Las comunidades en Cuba.....	14
1.4 CONCLUSIONES DEL CAPÍTULO.....	15
CAPÍTULO 2: ESTUDIO DE LAS HERRAMIENTAS Y GUÍAS DE MIGRACIÓN.....	16
2.1 HERRAMIENTAS USADAS EN LOS PROYECTOS DE SOFTWARE EDUCATIVO	16
2.1.1 Ofimática	16
2.1.2 Correo electrónico, acceso a Web y servicios.....	17
2.1.3 Los calendarios y herramientas colaborativas	17
2.1.4 Bases de Datos.....	17
2.1.5 Ingeniería de Software	18
2.1.6 Herramientas para el desarrollo	18
2.2 HERRAMIENTAS ANÁLOGAS.....	19
2.2.1 Análisis de las herramientas del primer grupo (Ofimática)	19
2.2.2 Análisis de las herramientas del segundo grupo (Correo Electrónico, Acceso a Web y Servicios)..	20
2.2.3 Análisis de las herramientas del tercer grupo (Los Calendarios, Herramientas Colaborativas y	
Gestión de Documentos)	20
2.2.4 Análisis de las herramientas del quinto grupo (Paquete de herramientas para el desarrollo).....	20
2.3 ESTUDIO DE LA DOCUMENTACIÓN DE ESTRATEGIAS DE MIGRACIÓN.....	22
2.3.1 Directrices IDA de migración a software de fuentes abiertas.....	23
2.3.2 Plan Institucional de Liberación de Software y de Migración a Plataformas de Software Libre	24
2.3.3 Metodología para la migración a Software Libre de la UCI.....	28
2.3.4 Metodología para la migración a software libre de las universidades del ministerio de educación	
superior (MES).....	32
2.4 CONCLUSIONES DEL CAPÍTULO.....	34
CAPITULO 3: ESTRATEGIA DE MIGRACIÓN A SOFTWARE LIBRE PARA LOS PROYECTOS DE SOFTWARE	
EDUCATIVO.....	35
3.1 GENERALIDADES DE LA ESTRATEGIA.....	35
3.1.1 Aspectos organizativos	37
3.1.2 Grupo de Migración.....	37
3.2 DESCRIPCIÓN DE LOS FLUJOS DE TRABAJO	38
3.2.1 Flujo de trabajo: Levantamiento informático	39
3.2.2 Flujo de trabajo: Diseño	41
3.2.3 Flujo de trabajo: Pilotaje.....	42
3.2.4 Flujo de trabajo: Formación.....	43
3.2.5 Flujo de trabajo: Implantación	47
3.2.6 Flujo de trabajo: Supervisión y soporte informático.....	49
3.3 APLICACIÓN DEL MÉTODO DE EXPERTOS	52

TABLA DE CONTENIDOS

CONCLUSIONES	54
RECOMENDACIONES	55
BIBLIOGRAFÍA.....	56
TRABAJOS CITADOS	57
SITIOS DE INFORMACIÓN.....	58
GLOSARIO.....	59
ANEXOS	64

INTRODUCCIÓN

La UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS constituye un eslabón fundamental en la informatización del país, vinculando la educación y la producción, con lo cual se visiona tener amplios resultados incorporando a la sociedad cubana los software que produce, aportando estos elementos en la construcción del prototipo de una sociedad de la información.

En la actualidad la producción de software educativo, en la universidad, se ha desarrollado con herramientas privativas, tales como: Adobe Flash, Adobe Director, Adobe Dreamweaver, Toolbook, etc., las cuales, pese a ser de empresas radicadas en los Estados Unidos, se han utilizado sin tener presentes los problemas de derechos de autor y licencias que esto puede traer.

El bloqueo del gobierno de los Estados Unidos a nuestro país no permite la comercialización de software desarrollados con tecnologías de empresas norteamericanas. El país tampoco puede adquirir libremente estas herramientas, las cuales se obtienen a través de terceros (en ocasiones cuartos y quintos) los precios que hay que pagar por estos softwares y sus licencias son insostenibles.

El surgimiento, desarrollo y la potenciación del software libre ha permitido al centro una nueva posibilidad de desligarse del uso de estas herramientas privativas. Es esta sin dudas una vía hacia el desarrollo independiente de nuestro país en la rama de la informática.

En la actualidad se han hecho convenios de trabajo con empresas y gobiernos extranjeros los cuales tienen interés en el desarrollo de software para la educación, los cuales deben ser desarrollados con herramientas libres y ejecutarse sobre las distribuciones de GNU/Linux. La situación real en la producción de SWE no ha permitido cumplir fielmente los compromisos adquiridos.

Los proyectos más grandes y de mayores compromisos que se ejecutan hoy en el Polo de Software Educativo son: Proyecto “Alfaomega”, Proyecto “Español para no Hispanohablantes”, Proyecto “Migración de las colecciones Multisaber¹ y El Navegante²” y Proyecto “Educandus”. Haciendo un análisis global de los proyectos de SWE, que hoy se están desarrollando en la UCI, se puede concluir lo siguiente:

Alfaomega es un grupo editorial en expansión con sede en México, que ha mostrado su interés en establecer relaciones comerciales con la UCI, en el desarrollo y edición de publicaciones de

¹ Colección Multisaber: Colección de software educativo desarrollada por el Ministerio de Educación de Cuba para la educación primaria

² Colección El Navegante: Colección de software educativo desarrollada por el Ministerio de Educación de Cuba para la educación

profesores cubanos y en el desarrollo de software educativo para escuelas mexicanas. Inicialmente el proyecto consiste en el desarrollo de los softwares educativos de la colección Futuro³ del Ministerio de Educación de Cuba (MINED). Uno de los puntos abordados en el contrato con la editorial Alfa omega fue que estos productos deben ser desarrollados con tecnologías libres. Es esta sin dudas una de las oportunidades de negocios e inicio de nuevos mercados que hoy tiene la Universidad y el país.

El proyecto *Migración de las colecciones Multisaber y El Navegante* fue el primero de los desarrollos, que impuso un gran desafío, el cual consistió en migrar a SWL las colecciones educativas Multisaber y El Navegante para la escuela venezolana en un primer momento y luego para la escuela cubana, lo cual facilitaría el proceso de migración del MINED y las escuelas cubanas a SWL.

Español para no Hispanohablantes es un producto, que en un primer momento, tiene una audiencia bien definida (estudiantes chinos que estudian en Cuba idioma español), se decidió que el producto se generalizara para el estudio de lengua española destinado a otros países no hispanohablantes. Aunque en un principio se ha desarrollado con herramientas privativas, una vez que se quiera comercializar, esto lo impediría.

Por último *Educandus*, es una empresa brasileña y tiene la intención de comenzar a desarrollar productos educacionales con la UCI, por lo tanto se crea un proyecto de igual nombre a la empresa, con el objetivo producir estos SWE. Los cursos que hay que implementar deben correr sobre una plataforma Web desarrollada con tecnologías libres y aunque estos cursos se implementan con Adobe Flash, el hecho de que se puedan desarrollar con herramientas libres, daría una mejor visión internacional, permitiendo que la UCI se acreditase como autor.

Los principales problemas que se han detectado y de alguna forma han afectado el desarrollo de estos y otros proyectos son los siguientes:

- Las herramientas de desarrollo que se conocen no permiten cumplir con los requerimientos de los productos a desarrollar.
- No se tiene conocimiento o se tiene escasos conocimientos de cómo trabajar con algunas de estas herramientas.
- Los miembros de los equipos de proyectos (estudiantes y profesores) hacen rechazo a Linux y las herramientas de desarrollo libres, cuando se enfrentan a ellas las primeras veces.

³ Colección Futuro: Colección de software educativo desarrollada por el Ministerio de Educación de Cuba para la educación preuniversitaria.

En estos momentos constituye un reto desarrollar un software educativo o una multimedia para plataformas libres y aun más desarrollarlo con herramientas y tecnologías libres. A partir de lo anteriormente mencionado se puede concluir que: el empleo de las herramientas privativas para el desarrollo de software educativo en la UCI está afectando la migración a SWL, debido al rechazo que hace el personal de los proyectos, por adolecer de una estrategia efectiva en el cambio de tecnología.

Para lo cual se estudiará la producción de software educativo en la UCI, dentro del proceso de desarrollo de estos softwares, que requieran tecnologías libres. El objetivo general de la investigación es desarrollar una estrategia para la migración a SWL en la producción de SWE, tratando de dar respuesta a las siguientes incógnitas:

- ¿Cómo lograr la independencia tecnológica en los proyectos de desarrollo de software educativo?
- ¿Cómo realizar el proceso de migración a software libre en los mismos?

Se trazaron un conjunto de tareas investigativas a desarrollar:

- Analizar y documentar las condiciones que permiten la adopción del SWL para la producción de SWE.
- Realizar un estudio investigativo de las tecnologías privativas y open source, que se están utilizando en el desarrollo de SWE.
- Hacer un estudio de las herramientas posibles para el desarrollo de productos con tecnologías libres.
- Estudiar las diferentes guías de migración escritas por comités técnicos en diferentes países e instituciones y las realizadas en la UCI
- Elaborar una estrategia para la migración a SWL de la línea de productos de SWE.
- Realizar la evaluación de la estrategia propuesta.
- Incorporar a la estrategia de migración los resultados obtenidos en la validación de la misma.

Se utilizará para ello el método Histórico-Lógico en sus dos tipologías: el analítico-sintético, para la recolección de información sobre el SWL, dígase ventajas y desventajas, manipulación, herramientas que serán de utilidad para la producción específica a desarrollar y diferentes guías de migración escritas por comités técnicos en disímiles países e instituciones. Con estos datos se podrá

entonces hacer uso del método Inductivo-Deductivo, con el cual se llegará a conclusiones y a un documento que describa la estrategia de migración para el polo de SWE.

Las técnicas investigativas o de recopilación de información estarán estrechamente vinculadas, tanto entrevistas, análisis documental y observación; mediante las cuales se determinarán las necesidades del polo de SWE a los que atañe el problema que se trata de resolver. La investigación impulsa a la búsqueda de información sobre el tema en Internet, o de experiencias que hayan acumulado los ingenieros del centro (con un nivel curricular que permita su fiabilidad) y catalogarla según su utilización.

Con el desarrollo de esta investigación se pretende definir los flujos de trabajo y las fases de la migración, para guiar la adopción de tecnologías libres, que permitan desligarse de las herramientas privativas y disminuir los altos costos en la producción de SWE. Una vez que se genere el proceso propuesto, se espera lograr minimizar la brecha informacional, con respecto a la cultura del SWL y el rechazo que hace el personal de los proyectos cuando se cambian los paradigmas de desarrollo a sistemas libres.

El presente trabajo consta de tres (3) capítulos estructurados de la forma siguiente:

Capítulo 1 Fundamentación teórica: En este capítulo, se da a conocer que es GNU/LINUX y cómo surgió. Se explican un grupo de condiciones que permiten la adopción de las tecnologías libres y el desarrollo de las comunidades virtuales cubanas que promueven la cooperación en el uso de distribuciones GNU/LINUX.

Capítulo 2 Herramientas y guías de migración: En este capítulo, se hace un exhaustivo análisis de los principales intentos de migración a Software Libre en países desarrollados o en desarrollo, así como en determinadas compañías, a través de guías, documentos y publicaciones con estos fines, presentes en internet. Se enfatiza en las tareas de cada una de las etapas de la migración a estos sistemas de código libre. Se determinaron las herramientas que actualmente utilizan los proyectos de SWE y las análogas a estas en SWL.

Capítulo 3 Estrategia de migración a software libre de los proyectos de software educativo: En este capítulo se realiza la propuesta de estrategia de migración a SWL, se definen y explican las fases de la misma y los flujos de trabajo asociados que darán cumplimiento a la estrategia, haciendo uso artefactos como resultado de estos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En la transición hacia una sociedad del conocimiento altamente interconectada, la información es el motor principal de la misma. Es capaz de modificar las relaciones en nuestro entorno, empresas, personas y hasta los gobiernos han cambiado sus medios de comunicación, la forma de brindar sus servicios e incluso la forma de gestionar el conocimiento. La gestión del conocimiento corresponde al conjunto de procesos y sistemas que permiten que el capital intelectual de una organización aumente significativamente, mediante la gestión de sus capacidades de resolución de problemas de forma eficiente, y la manera de generar ventajas competitivas a través del tiempo. (1)

La importancia de controlar la información, así como su transparencia, veracidad y confidencialidad es disputada mundialmente; puesto que inflige poder y autoridad. Son los softwares, los intermediarios encargados de soportar, autorizar y manipular la información. Surge entonces una incógnita - ¿Qué es el software? - que como respuesta tiene una definición según Microsoft: "El "Software" es un programa o grupo de programas que muestra al equipo informático cómo funcionar de la manera deseada y contiene un conjunto de líneas de texto, que son las instrucciones que debe seguir la computadora para ejecutar dicho programa: el código fuente.

En el código fuente se define por completo el funcionamiento del SO, escrito por un programador en algún lenguaje de programación; pero no es directamente ejecutable por la computadora, sino que debe ser traducido a otro lenguaje (el lenguaje máquina o código objeto) que sí pueda ser ejecutado por el hardware de la computadora. A partir del concepto anterior se puede inferir que sin acceso al código, no se puede ver cómo está hecho o introducir mejoras. En este sentido, el tener acceso al código fuente juega un papel fundamental en cómo se explotan las libertades del software. Dada la importancia del mismo en la sociedad, es imprescindible garantizar su transparencia y fiabilidad.

1.1 El Software Libre

A continuación se definirán los conceptos más importantes relacionados con el SWL. Además de algunos de los principales proyectos que dieron lugar a su surgimiento, los cuales han permitido ganar fuerzas y prestigio al movimiento que se hace llamar con este nombre (Software Libre).

1.1.1 ¿Qué es el software libre?

El software libre es aquel que se encuentra disponible gratuita o comercialmente. Se basa en la cooperación, la transparencia y garantiza una serie de libertades a los usuarios (2). Estas libertades son (3):

- Libertad 0.

Libertad para utilizar el programa para cualquier propósito.

- Libertad 1.

Libertad para poder estudiar cómo funciona el programa. Implica acceso al código fuente del mismo.

- Libertad 2.

Libertad para redistribuir el programa.

- Libertad 3.

Libertad para hacer modificaciones y distribuir las mejoras. Implica también acceso al código fuente del mismo.

1.1.2 Unix

Un sistema operativo (SO) es un conjunto de programas destinados a permitir la comunicación entre el usuario y el ordenador, gestionando sus recursos de manera eficiente (4). Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos. Unix ha sido el sistema operativo que más influencia ha tenido en la historia de la microinformática⁴ y constituye los fundamentos técnicos sobre los cuales se desarrolló el software libre.

Unix era un SO que los propios usuarios debían mantener según la política de AT&T⁵, porque esta no brindaba ningún soporte. Tratando de dar solución a estos problemas, se comenzó a colaborar e intercambiar información entre los programadores. Para esto AT&T, por una pequeña suma de dinero, facilitaba una copia del código fuente de Unix permitiendo adaptabilidad en universidades y centros de computación.

La universidad de Berkeley⁶ fue una de las primeras en la adopción y desarrollo de Unix liberando en 1978 la primera versión de Unix BSD (Berkeley Software Distribution). Esta distribución incluía varias herramientas diseñadas en la propia universidad y mejoraba considerablemente la versión original. Uno de los programadores involucrados fue Bill Joy quien creó más tarde Sun Microsystems, en donde adoptarían Unix como sistema operativo para todos sus hardware.

⁴ Microinformática: Es la informática basada en ordenadores que utilizan como unidad central de proceso (CPU) un microprocesador.

⁵AT&T: Compañía Estadounidense de telecomunicaciones. Una división de esta compañía, la Bells Lab, creó el primer Unix.

⁶ <http://www.berkeley.edu/>

1.1.3 El Proyecto GNU

Con el descenso de los precios de los sistemas y la popularidad alcanzada, aparecieron las primeras empresas de software. Se iniciaron contratando programadores (hackers), que radicaban en los centros de cálculos de las universidades. Muchas de estas empresas comenzaron a denegar el acceso al código fuente de las aplicaciones que desarrollaban o mejoraban, para conseguir algunas ventajas competitivas. Este fue uno de los factores de vital importancia que rompió con la ética hacker y la tradición de compartir el código.

Este modelo en el cual, el software se vendía sin el código fuente se fue expandiendo cada vez más y las libertades de los usuarios se fueron acortando. Esta fue la época en que aparecieron técnicas como por ejemplo: las bombas de tiempo, que limitaban el período durante el cual un usuario podía utilizar un producto. Los programas shareware popularizarían más tarde estas como sistema para inducirlos a adquirir una licencia.

El proyecto GNU fue iniciado con el objetivo de crear un sistema operativo completamente libre. Una de las personas que había vivido de cerca toda aquella evolución era Richard Stallman⁷, quien fue pionero en defender las libertades que se habían perdido y acuñó el término 'software libre'. El 27 de septiembre de 1983 Richard Stallman muy preocupado por esta pérdida de libertades anunciaba en el foro Usenet net-unix.wizards, que empezaba a trabajar sobre una implementación libre de un sistema inspirado en Unix, denominándolo GNU y que estaría libre de código de AT&T, es decir, una implementación desde cero sin posibles problemas legales con AT&T. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar el proyecto, entre las que destaca "retornar al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras". GNU es un acrónimo recursivo que significa "GNU No es Unix". (5)

En 1984, Stallman fundó la Free Software Foundation, con el objetivo de crear el sistema libre GNU y la potenciación del software libre. El sistema fue diseñado para ser totalmente compatible con UNIX. Este hecho implica, que GNU está compuesto de pequeñas piezas individuales de software, muchos de los cuales estaban disponibles en aquel momento, como el sistema de edición de textos

⁷ <http://www.stallman.org>

Tex y el sistema gráfico X Windows, que pudieron ser adaptados y reutilizados; otros en cambio tuvieron que ser reescritos.

Para asegurar que el software permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos, al tiempo que evitase restricciones posteriores de los mismos. La idea se conoce en inglés como copyleft (en clara oposición a copyright), y está contenida en la Licencia General Pública de GNU (GPL) (3).

En el manifiesto GNU, Stallman mencionó que "un núcleo inicial existe, pero se necesitan muchos otros programas para emular Unix". TRIX es un núcleo de llamadas remotas a procedimientos, desarrollado por el MIT⁸ y cuyos autores decidieron que fuera libremente distribuido, era totalmente compatible con UNIX versión 7. En Diciembre de 1986 ya se había trabajado para modificar este núcleo. Sin embargo, los programadores decidieron que no era inicialmente utilizable, debido a que solamente funcionaba en "algunos equipos sumamente complicados y caros", razón por la cual debería ser portado a otras arquitecturas antes de que se pudiera utilizar. (6)

1.1.4 Linux

Stallman continuó trabajando en las herramientas necesarias para crear aquel sistema operativo libre y sus protecciones, por lo que trabajo en la versión 1.0 de la licencia GPL (General Public Licence). Consistía en un contrato entre el autor y el usuario que garantizaba la cesión de los derechos que definían al software libre. La licencia GPL era una herramienta legal muy importante, dado que muchos programadores cogían códigos que eran libres, hacían modificaciones y no las aportaban a la comunidad.

Ya en 1991 Linus Torvald, un estudiante finlandés de 21 años, publica en la Red su propio núcleo de sistema operativo inspirado en Unix, que más tarde se denominaría Linux. Rápidamente, muchos desarrolladores se sumaron a un esfuerzo colectivo, para ayudarlo a mejorar el núcleo del sistema operativo, creciendo en importancia. Éste era precisamente el corazón que le faltaba y que Stallman estaba diseñando. (5)

Fue así, que los hackers de Linux fueron adaptando las herramientas de GNU al núcleo de Linux y crearon las primeras distribuciones GNU/Linux. Estas distribuciones incluían Linux y todas las herramientas necesarias para que un usuario lo pudiera utilizar. Richard Stallman insistió mucho en el hecho de que el sistema GNU/Linux existe gracias a las aportaciones de la Free Software Foundation y

⁸ <http://www.mit.edu>

propone usar la denominación GNU/Linux para hacer referencia a sistemas completos que incorporan un núcleo Linux. La mayoría de los usuarios y distribuciones comerciales siguen hablando de Linux para identificar al sistema y al núcleo. (5)

GNU/Linux se ha convertido en el buque insignia del movimiento del software libre, ya que es un sistema operativo completamente libre, que incluye todo tipo de herramientas para servidores y usuarios finales. (7)

1.1.5 Open Source

Durante el año 1998, Eric S. Raymond, Bruce Perens y otros involucrados en el desarrollo de software libre lanzaron la Open Software Initiative⁹ y propusieron el uso del término open source (código abierto¹⁰) en contraposición al de free software (software libre) determinándolo más atractivo al entorno empresarial. En el mundo anglófono se creaba una situación incómoda debido a la doble acepción que en inglés tiene el término free (que puede significar gratuito¹¹ o libre). La gran mayoría de empresas en Estados Unidos, usan principalmente el término código abierto, para evitar dar la percepción de que el software libre es un recurso totalmente gratuito y para poner énfasis en el valor diferencial que representa el hecho de que el código fuente está disponible.

Bruce Perens, de la Open Source Initiative creó una lista de condiciones, que debe cumplir un programa para poder ser considerado Open Source y son las siguientes: (8)

1. Libre distribución: No se puede impedir la venta o distribución del programa o parte de él. Así mismo, tampoco se puede exigir el pago de un canon o tasa a cambio de su distribución por parte de terceros.
2. Código fuente: El programa debe incluir su código fuente y no se puede restringir su redistribución.
3. Trabajos derivados: No debe impedirse realizar modificaciones o trabajos derivados del programa y debe permitirse que éstos sean distribuidos bajo los mismos términos del software original.
4. Integridad del código de fuente original: Puede exigirse que una versión modificada del programa tenga un nombre y número de versión diferente que el programa original, para poder proteger al autor original de la responsabilidad de estas versiones.

⁹Open Source Initiative: Es una corporación sin fines de lucro creada para educar y abogar por sobre los beneficios del código abierto y de construir puentes entre los diferentes grupos en la comunidad open-source <http://www.opensource.org>

¹⁰ <http://www.hispalinux.es/informes/wheeler/index.html> (Español)

¹¹ http://es.wikipedia.org/wiki/C%C3%B3digo_abierto

5. No discriminación contra personas o grupos: Las condiciones de uso del programa no pueden discriminar contra una persona o un grupo de personas.
6. No discriminación contra usos: No se puede negar a ninguna persona hacer uso del programa para ningún fin como, por ejemplo, comercial o militar.
7. Distribución de la licencia: Los derechos del programa deben aplicarse a todos los que se redistribuyen el programa sin ninguna condición adicional.
8. La licencia no debe ser específica de un producto: Los derechos garantizados al usuario del programa no deben depender de que el programa forme parte de una distribución o paquete particular de software.
9. La licencia no debe restringir otro software: La licencia no debe poner restricciones en otros programas que se distribuyen junto con el software licenciado.
10. La licencia debe ser tecnológicamente neutra: No puede existir ninguna disposición de la licencia que obligue al uso de una tecnología concreta.

1.2 Consideraciones para adoptar software libre

Se analizarán algunas tentativas que hacen elegir software libre para el trabajo diario. Comúnmente para decidir qué sistema operativo utilizar, sobresalen aquellos que mayores beneficios proporcionen, o se basa en la búsqueda de una característica distintiva y de gran peso para el que lo elija. Sería importante destacar por lo tanto, con respecto al SWL, cómo se comporta, el nivel de seguridad que proporciona, el acceso al código fuente, la configuración de archivos y el costo de adquisición (que es el que más pesa). Estas no son las únicas cualidades a destacar en el software libre, existen muchas más; pero de la misma manera hay algunas desventajas en su utilización. Realmente es imprescindible eliminar la dependencia tecnológica y actualmente el software libre es la vía para el desarrollo informático con costos mínimos.

1.2.1 Nivel de seguridad proporcionado

Existen muchas razones para realizar una instalación y configuración enfocadas hacia la máxima seguridad en servidores con el sistema operativo Linux. Una de ellas es que una configuración segura implica casi por definición, instalar sólo y exclusivamente los componentes indispensables para su operación, mantenimiento y administración del servidor. (9)

Esto significa que serán muy pocos los componentes y/o programas cargados, por lo tanto el procesador trabajará menos para realizar las mismas tareas. Otra razón más es la de disminuir las

posibilidades de que un ataque (interno o externo) pueda derivar en denegación de servicio o robo de información y una adicional (hay más todavía) es que será muy eficiente su operación. (9)

Todavía hay personas, que no notan la diferencia entre la instalación de servicios en una PC de escritorio de pruebas, y un servidor de producción. Hay básicamente dos corrientes de pensamiento en términos de servicios de información: (9)

La primera se refiere a poner en operación un solo servidor muy potente, grandes cantidades de memoria, muchos discos y varios procesadores y como consecuencia de toda esa inversión, le se añade la mayor cantidad de servicios posibles como: sistemas de bases de datos, aplicaciones, correo electrónico, páginas Internet, páginas Intranet, DNS, etc. para que “desquite” lo que costó. Es decir; se ponen todos los huevos en una canasta. Problema: cuando falla el servidor o se compromete la seguridad del sistema fallan todos los servicios ahí instalados.

La segunda corriente del pensamiento sería distribuir cada servicio de información, en uno o varios servidores con mucha menor capacidad en procesador, velocidad, memoria y espacio en discos. Ventajas: mayor eficiencia en la prestación de servicios. Es decir, si se asigna una máquina por cada servicio como correo, páginas Internet, páginas Intranet, y sistemas de bases de datos, se tendrán cuatro procesadores atendiendo a los usuarios de forma simultánea. En caso de falla o de compromiso de seguridad, sólo se ve afectado el servicio que corría en esa máquina. Una ventaja adicional es que en caso de ser requerido un mantenimiento preventivo o correctivo, será posible realizarlo sin alterar más servicios. ¡Y la más importante! Será mucho más sencilla la administración de la seguridad de ese servidor.

1.2.2 Recopilación del Kernel

Otro aspecto importante es el hecho de que en cada distribución se puede obtener el código fuente y realizar las modificaciones al kernel que se crean adecuadas a las funciones del servidor, que se va a instalar. El nuevo será más pequeño y eficiente, ya que sólo incorporará y cargará los módulos indispensables para la operación y procesamiento de información.

1.2.3 Configuración de archivos

Si se poseen y/o se desarrollan los conocimientos suficientes, podrán realizar los cambios a la configuración original de los archivos que controlan las variables de ambiente. De esta forma cancelarán, limitarán o ampliarán los permisos de acceso al sistema y las aplicaciones que corren en él.

Linux incorpora en una instalación típica, herramientas de desarrollo y depuración del kernel, de los programas que lo componen, lenguajes compiladores e intérpretes, editores de texto, bases de

datos, filtros de paquetes, etc. lo que permitirá emplearlo como servidor de correo, DNS, base de datos relacionales, ruteador, firewall, DNS, servidor de páginas web, servidor de archivos, de aplicaciones de oficina, servidor de monitoreo de redes, etc. sin necesitar nada más, ya que por lo general cualquier distribución integra estos programas. Con estos conocimientos se podrá pensar en la implementación de seguridad en este sistema operativo que no tiene nada de sencillo ni humilde y que requiere ciertas habilidades que no siempre son fáciles de encontrar.

1.2.4 Independencia tecnológica.

Una licencia de software (10) es un contrato, entre el titular del derecho de autor (propietario) y el usuario del programa informático (usuario final), para utilizar éste en una forma determinada y de conformidad con unas condiciones de uso convenidas.

Las licencias de software pueden establecer entre otras cosas: la cesión de determinados derechos del propietario al usuario final, sobre una o varias copias del programa informático, los límites en la responsabilidad por fallos, el plazo de cesión de los derechos, el ámbito geográfico de validez del contrato e incluso pueden establecer determinados compromisos del usuario final hacia el propietario, tales como la negación del programa a terceros o la no reinstalación del programa en equipos distintos al que se instaló originalmente.

Este tipo de forma de hacer software lleva al principio de la solidaridad y cooperación. Demuestra entonces, que compartir no es un delito, sino que lleva a la evolución y permite desarrollar la comunidad. Esta forma de pensar no es tan sólo técnica, sino también filosófica, conceptual y promueve la reflexión sobre qué es lo que pasa con el conocimiento en ésta era en la que se vive.

1.2.5 Costes

El costo es uno de los factores más importantes de realizar una compra de software, para las empresas. Para la mayoría de usuarios individuales y las instituciones públicas el software libre es una opción atractiva por las libertades que garantiza, sin necesidad de verse lastrados por el precio.

Cuando se analiza el precio de una solución tecnológica, se suele hablar del TCO (Total Cost of Ownership), es decir, del coste total de la propiedad que tiene una determinada solución de software. Este concepto fue creado por el Gartner Group 23 en 1987, como herramienta de análisis exhaustiva de los costes de una solución de mercado y desde entonces se ha convertido en un estándar. En este análisis se reflejan el coste del programa, la ayuda, y el mantenimiento tecnológico de la solución. Si parte de la base que el SWL no tiene prácticamente costo de licencia, esa parte del dinero se puede emplear en la formación del personal, para el uso de esa tecnología y en la adaptabilidad de los programas.

1.2.6 Requerimientos de hardware

Los sistemas operativos y aplicaciones necesitan recursos de hardware para su instalación. Los requerimientos de hardware se orientan en definir características como: capacidad de la RAM, tipo de procesador, espacio en disco, entre otros. En la Tabla 7 (Ver Anexo 14) se listan requisitos que debe cumplir el hardware donde se implante alguna distribución de GNU/LINUX. Existen distribuciones las cuales necesitan como mínimo desde 16 hasta 128 MB solamente, estos son por ejemplo Debian 3.0, Xubuntu 7.10 o Fedora Core 5. Aunque quizás no sean los más idóneos para el trabajo en un centro o empresa no se necesita de muchos recursos.

1.2.6.1 Evitar hardware cerrado o propietario

Algunos fabricantes de hardware simplemente no dicen cómo escribir drivers para su hardware, o no permiten acceder a la documentación sin un acuerdo de no divulgación que prevenga de revelar su código fuente para Linux. Un ejemplo es el sistema de sonido DSP¹² para portátiles IBM y algunos de esos sistemas también acoplan el sistema de sonido al módem. Otro ejemplo es la disquetera en los antiguos modelos Macintosh.

Puesto que no se ha tenido acceso a la documentación de esos dispositivos, simplemente no funcionarán bajo Linux. Puede ayudar a cambiar esta situación reclamando a los fabricantes de este hardware liberar la documentación. Si esto lo hace una cantidad suficiente de personas, entonces comprenderán que la comunidad de software libre sería un importante mercado.

1.3 El software libre en Cuba

La mayor parte de la industria del software cubano, se basa en el consumo de publicaciones propietarias desarrolladas en otros países y el desarrollo de productos se basa en tecnologías privativas extranjeras. Las empresas creadoras se reservan los derechos de uso y propiedad intelectual, por lo que es imposible realizar mejoras en esa tecnología, que permita usarla para dar solución a los problemas.

El uso del software libre permite introducir modificaciones a las aplicaciones. De esta manera, se contribuye a la formación de profesionales en nuevas tecnologías y al desarrollo de la industria nacional; pero además las mejoras que se realicen tendrán restricciones y se podrán compartir con cualquier otra empresa, institución u organismo que las necesite, al igual que las propias aplicaciones.

¹²"DSP", cuyo significado era *Digital Sound Processor* se creó para designar la parte del audio digital de la Sound Blaster, que es una familia de tarjeta de sonido para PC.

1.3.1 Las comunidades en Cuba

En el mundo y todo el tiempo surgen comunidades de personas que se agrupan de acuerdo a intereses comunes, ya sea de manera organizada o espontánea. El arribo de Internet propició la creación de comunidades virtuales, aun con limitantes (problemas de conectividad conocidos y de enlace satelital), han sido numerosas, con distintos fines e intereses (Bloggers Cuba es un ejemplo fehaciente)

Una de las más antiguas nacida en un entorno virtual es la Comunidad Cubana de Usuarios de Software Libre (CCSWL). La CCSWL (11) nace en torno a la lista de discusión Linux –L entre 1998 y 1999 para la cual el Ministerio de la Informática y las Comunicaciones (MIC) aportó un nuevo servidor. En sus comienzos la integraban administradores de redes, que intentaban encontrar respuestas a sus problemas con los sistemas libres. Luego se fueron haciendo asiduos muchos más usuarios (no sólo cubanos), a medida que ganaban conciencia de cuál era el verdadero significado del software Libre (SWL) y la importancia de su uso, comenzaron debates no sólo de cuestiones técnicas y filosóficas.

En ese entonces se inicia la organización de usuarios en algunas provincias del país, teniendo como grupo insignia a los que utilizaban Linux de Ciudad de La Habana (LiHab), que comienzan a alternar el espacio virtual con el terrenal. Estos inician colaboraciones sin remuneración con los Joven Club y otras instituciones impartiendo cursos y ofreciendo consultoría sobre el uso del SWL (11). En el año 2002 surge un acuerdo del Consejo de Estado de la República de Cuba, que propone la utilización de SWL y en el 2004 se anuncia la decisión y la voluntad política del país de comenzar una migración hacia SWL.

La migración cubana ha sido un proceso irregular con altas y bajas, momentos donde se hacen cosas y otros donde parece que no existe, pero si algo se ha mantenido constante ha sido el deseo de colaborar y contribuir de la CCSWL. Ya Linux-L tiene 10 años y se considera como el núcleo funcional de la CCSWL. (11)

En estos tiempos parece tomar un nuevo impulso, le corresponde a la UCI jugar su papel continuador e impulsor de la migración. En este centro se cuenta también con una comunidad, así como una facultad que soporta sólo esa tecnología libre. El objetivo fundamental es lograr una independencia total de las aplicaciones propietarias, por eso es necesaria una estrategia de migración para los proyectos productivos, que han estado trabajando con software propietario. En ese caso se incluye la esfera de producción de software educativo.

1.4 Conclusiones del capítulo

En este capítulo se hizo un estudio en los principios básicos de la tecnología libre, se dió respuesta a muchas de las inquietudes que se formulan, cuando se utiliza el concepto de software libre y otros que muy pocas veces se explican cuando se habla en este sentido. Se justificaron las características que posee el SWL, que hacen elegirlo para suplir las necesidades tecnológicas actuales. Se manifiesta como Cuba se ha ido adentrando en el mundo de las comunidades que abogan por la independencia tecnológica.

Al finalizar la fundamentación teórica se concluyó el estudio de los principales conceptos del software libre, además de algunos de los proyectos de más importancia, que dieron origen a su surgimiento, lo que permitió crear la base teórica que fundamentarán la estrategia a desarrollar. Se deriva de este estudio que se puede utilizar cualquier sistema operativo libre que se desee o se defina en la planificación de migración para los proyecto del polo de SWE en la UCI.

CAPÍTULO 2: ESTUDIO DE LAS HERRAMIENTAS Y GUÍAS DE MIGRACIÓN.

El éxito de un proceso de migración a software libre depende del conjunto de pasos a seguir para cumplimentar ese objetivo. Para ello se hace indispensable regirse por un documento, que establezca las etapas y fases de desarrollo, las cuales presentan objetivos, acciones y metas definidas: Guías de migración. Es necesario dejar claro en el documento el tiempo a emplear en la implantación de dicha estrategia, garantizando un trabajo continuo, que se ha de redefinir en un lenguaje claro, mediante las acciones a llevar a cabo en cada migración.

Además se deben analizar cuáles son aquellas herramientas informáticas que se están utilizando en la producción de software educativo. Luego se debe realizar un estudio de las análogas a las antes mencionadas, así como otras, que sean de importancia para los proyectos. De ahí que se hace necesario después sugerir el uso aceptado de estas, ya sea por el criterio de especialistas o porque han sido utilizadas y obtuvieron aceptación, en la producción de los proyectos donde se implantó.

2.1 Herramientas usadas en los proyectos de software educativo

En toda migración es importante hacer un estudio de las herramientas que se están utilizando, lo cual indica que es necesario listarlas y definir el uso que se le están dando, además del apoyo que brindaron para esta investigación las guías de migración realizadas en el centro y la tabla de homólogos que se presenta en el Anexo 2. Objetivamente se analizó con los Jefes de Proyectos del Polo de SWE de la facultad 8 (UCI) las herramientas que han estado usando, de cualquier sistema operativo que soportara la labor del proyecto. Se pudo definir, que ya se está haciendo uso de algunas herramientas libres, aunque están sobre Windows el personal posee dominio en ellas. Resumiendo la información recogida se elaboró la Tabla 1 (Ver Anexo 1), que esboza el listado de las herramientas con las que se trabaja actualmente. Para mejor comprensión de la investigación se dividieron las herramientas en cinco (5) grupos, atendiendo a la utilidad que se les puede dar.

2.1.1 Ofimática

Ofimática es uno de los grupos más importantes y consiste en la creación, modificación e impresión de archivos, que contienen datos de negocio, como informes, cartas y presentaciones. Los formatos de los archivos Microsoft (doc., xls, ppt) se deben poder leer y escribir con gran precisión. Entonces es claro pensar en el paquete de office de Windows, el cual contiene varias herramientas que son imprescindibles para el trabajo con archivos de datos.

2.1.2 Correo electrónico, acceso a Web y servicios

El correo electrónico permite mantener informado a todo el personal de la universidad, de todo el quehacer diario. Por lo tanto fue declarado como otro grupo, consiste en la creación, recepción y visualización de correos electrónicos, incluido el apoyo para lograr un correo seguro. Actualmente en algunas PC se puede encontrar Outlook, Internet Explorer, Mozilla Firefox y Opera como los navegadores más populares y utilizados. También para el acceso a la Web y servicios utiliza esas herramientas de visualización y manipulación del correo del lado del cliente, ya que consiste en la capacidad de acceder a los protocolos de servicio de Internet y visualizar los resultados. Esto se hace normalmente con un navegador. Además de la capacidad de crear contenido y hacerlo accesible tanto interna como externamente.

2.1.3 Los calendarios y herramientas colaborativas

Los calendarios y herramientas colaborativas de trabajo en grupo, no son más que la creación y gestión de calendarios de grupos y libretas de direcciones. Los calendarios deberían permitir concertar reuniones y reservar alojamiento. Las libretas de direcciones se integrarían con los otros grupos funcionales. Los calendarios, junto con la gestión de contactos y el correo, suelen agruparse con el encabezamiento general de Gestión de la Información Personal (PIM). Algunos paquetes integrados, como el *Outlook* de Microsoft, proporcionan las tres funciones en una sola interfaz. La gestión de documentos permiten su almacenaje central con mecanismos de recuperación eficaces para el trabajo colaborativo y actualmente todos los proyectos de software educativo están utilizando para ese objetivo el Tortoise SVN. También hay algunos casos, donde no se utiliza ninguna herramienta colaborativa, pero es una propuesta que deberían tomar todos los proyectos, porque brinda facilidades en la producción de software, en función de la organización del desarrollo de los productos.

2.1.4 Bases de Datos

Las bases de datos (BD) y sus tecnologías tienen un impacto decisivo con el creciente uso de las computadoras. Desempeñan un papel crucial en casi todas las áreas de aplicaciones, como los negocios, la ingeniería, la medicina, el derecho, la educación y la bibliotecología. Son un conjunto de datos relacionados entre sí y por dato entiende: los hechos conocidos, que pueden registrarse y que tienen significado implícito.

Los proyectos que necesitan la manipulación de datos estructurados (en base de datos, tanto centrales como personales) lo están haciendo con las siguientes plataformas EMS MySQL Manager, Oracle y Microsoft Office Access. El trabajo con base de datos en el desarrollo de software permite

almacenar información que luego será requerida por las aplicaciones, el ejemplo más ilustrativo: el desarrollo de un SWE para el aprendizaje de las matemáticas, donde se necesitan almacenar todos los resultados de operaciones aritméticas en un rango numérico del cero al nueve y sin una base de datos sería práctica o realmente imposible.

2.1.5 Ingeniería de Software

La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos) Los proyectos se rigen por el establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico, que sean fiables y funcionen eficientemente sobre máquinas reales. La suite de Rational permite el modelado de los artefactos de la metodología Proceso Unificado de Desarrollo (RUP) y ofrece varios productos, destacándose los siguientes:

- Rational Requisite Pro: Mantiene a todo el equipo de desarrollo actualizado a través del proceso de desarrollo de aplicaciones haciendo que los requerimientos se puedan escribir, comunicar y cambiar fácilmente.
- Rational ClearQuest: Un producto Windows y basado en Web de administración de solicitudes de cambio que permite a los equipos de proyecto rastrear y administrar todas las actividades de cambio que ocurren durante el desarrollo del ciclo de vida.
- Rational Rose: La herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes.
- Rational SoDA: Automatiza la producción de documentación para todo el proceso de desarrollo de software, reduciendo dramáticamente el tiempo y el costo de documentar el software.
- Rational ClearCase: Herramienta de administración de configuración de software, líder en el mercado, que da a los administradores de proyecto la posibilidad de rastrear la evolución de cada proyecto de desarrollo de software

2.1.6 Herramientas para el desarrollo

En el desarrollo de software educativo, el paquete de multimedia contiene muchas de las herramientas para el desarrollo de ese tipo de software, en dependencia de las condiciones

estructurales mínimas: Dreamweaver 8.0, Flash 8.0, Director y Firework. Pero se están usando otras, como es el caso de CS3, Toolbook y Aptana Studio. Para el tratamiento de imágenes de forma complementaria a las herramientas antes mencionadas adiciona el uso de: Adobe Photoshop y CorelDraw.

2.2 Herramientas análogas

Uno de los principales problemas que afrontan hoy las personas, que desean migrar a SWL, es que existe una gran diversidad de aplicaciones para elegir a la hora de seleccionar qué software se va a usar, cuando se migra de Windows a GNU/Linux. Generalmente las herramientas se escogen en función de lo que se desee hacer y las facilidades que el software debe brindar. Aunque una migración a SWL sea implementada en el área de producción de SWE, no indica que las herramientas propuestas sean las únicas que se puedan usar. Según las experiencias adquiridas mediante las técnicas de investigación desde todas las fuentes, dígase análisis documental de la información encontrada en Internet y entrevistas con especialistas del tema, se reunió un gran número de herramientas, que tienen utilidad para realizar un software educativo, considerándose imprescindibles al definir las herramientas necesarias para el polo de software educativo. En la Tabla 2 (Ver Anexo 2) se muestran las que se usan o se pudieran utilizar para el desarrollo de software educativo y otras necesarias en las estaciones de trabajo.

2.2.1 Análisis de las herramientas del primer grupo (Ofimática)

Si hasta el momento se estaba utilizando la suite de Microsoft para oficina “Microsoft Office” que incluye las siguientes aplicaciones: Microsoft Word (procesador de texto y utiliza la extensión de archivo *doc.*), Microsoft Excel (planilla de cálculo/hoja de cálculo que utiliza la extensión de archivos *xls*), Microsoft PowerPoint (programa de presentaciones, con extensión de archivos *ppt*), Microsoft Access (programa de bases de datos con extensión de archivos *dbo*), Microsoft Outlook (agenda y cliente de correo electrónico), Microsoft Photo Manager (editor fotográfico), Microsoft Publisher (editor para crear varios tipos de publicaciones como tarjetas, pancartas, etc.), Microsoft InfoPath (Editor y creador de documentos XML), Microsoft Project (gestor de proyectos) entre otros, que se recomienda sustituirlo por OpenOffice.org, puesto que sus componentes son muy similares al que se estaban usando anteriormente y han sido recomendada en la guía de Migración de la UCI.

Las herramientas incluidas en la suite ofimática de OpenOffice.org son por ejemplo OpenOffice.org Writer (Procesador de textos), OpenOffice.org Calc (Hoja de cálculo), OpenOffice.org

Impress (Programa de presentación), OpenOffice.org Base (Programa de bases de datos), entre otros y aunque todos realizan las mismas funciones que sus análogos, no se limitan a ello, realizan otras funciones que facilitan el trabajo de oficina (Ver anexo 4). Almacena todos sus datos en un formato estándar abierto internacional y también puede leer y escribir archivos de otros paquetes de software de oficina común. Puede ser descargado y utilizado completamente gratuito para cualquier propósito.

2.2.2 Análisis de las herramientas del segundo grupo (Correo Electrónico, Acceso a Web y Servicios)

Los proyectos de SWE estaban utilizando como clientes de correo, acceso a la Web y servicios Outlook, Internet Explorer, Mozilla Firefox y en casos aislados Opera, se recomienda utilizar Mozilla Firefox, Opera o Konqueror; puesto que tienen la facilidad de ser multiplataforma y estar ambos disponibles para Windows, Mac OS X, GNU/Linux, OS/2 y Solaris, además son aplicaciones en las que ya se tiene un conocimiento.

2.2.3 Análisis de las herramientas del tercer grupo (Los Calendarios, Herramientas Colaborativas y Gestión de Documentos)

Outlook además de ser un cliente de correo permite trabajar con calendarios, libreta de notas, etc; es recomendable sustituirlo por e-Groupware, es una herramienta colaborativa, utilizándose mucho en el proyecto GALM¹³ con muy buenos resultados. Su instalación es necesaria, aún cuando los proyectos nunca hubieran utilizado este tipo de herramienta, porque permite lograr organización en todas las tareas una vez que se comienza a utilizar; pero en la gestión de documentos se utiliza Endnote y Tortoise SVN, para tener un repositorio desde el cual todos puedan trabajar y acceder al trabajo de otros, para revisar documentos, entre otras muchas facilidades, entonces es recomendable sustituirlo por el KDESVN igualmente utilizado en el proyecto GALM o por cualquier otro que considere el personal del grupo de migración .

2.2.4 Análisis de las herramientas del quinto grupo (Paquete de herramientas para el desarrollo)

Para el desarrollo software educativo se hace uso de herramientas tales como Photoshop, 3DMas, Dreamweaver o cualquier otro editor de HTML y CSS, Toolbook, Flash, etc., se debería

¹³ Proyecto perteneciente a la facultad 8, el cual tiene como objetivo elaborar herramientas libres con las que se puedan crear softwares educativos, específicamente multimedia.

realizar un estudio a fondo de las posibilidades que brindan otras herramientas (Ver Anexo 2) y sustituirlas. Photoshop o CorelDraw, es sustituible por GIMP o Inkscape.

Inkscape es una aplicación multiplataforma, disponible principalmente para Windows, Linux y Mac OS X (mediante X11), en el que realizan objetos vectoriales escalables y cuenta con una interface similar a la presentada por CorelDraw. El GIMP es una herramienta de manipulación fotográfica multiplataforma y casi todas las distribuciones de GNU/Linux incluyen al GIMP como una aplicación estándar. Es una aplicación de Software Libre, cubierta por la Licencia Pública General (GPL license) que brinda a los usuarios la libertad de acceder y modificar el código fuente del que se construyen los programas.

Flash permite diseñar animaciones 2D y utiliza programación en ActionScript. En una sustitución de software privativo por software libre se podría combinar el trabajo de Eclipse más ASDT (esto es MTASC, SWFMILL, HAXE) para ActionScript y para animaciones 2D, KToon o Synfig. Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma, para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Synfig, es un editor vectorial 2D, aplicación capaz de realizar animaciones elaboradas, programa gratuito y de código abierto. KToon es una herramienta para desarrollo de animación en 2D, diseñada por animadores (Toonka Films)¹⁴ para animadores, enfocada a la industria de la animación profesional. Este proyecto está cubierto por la Licencia GPL¹⁵ usando G++¹⁶, OpenGL¹⁷ y QT¹⁸ como recursos de programación desde KDevelop¹⁹, como plataforma de desarrollo.

Para el caso de Toolbook se propone HAEDUC, herramienta desarrollada por el MINED. Esta se puede utilizar para la realización de libros electrónicos en general. (12)

Si el software educativo a desarrollar es sobre Web utilizando editores de HTML, CSS, XML, tal vez como el Dreamweaver es aconsejable la migración a Bluefish²⁰ o NVU / Kompozer. Bluefish es un

¹⁴ <http://www.toonka.com/>

¹⁵ <http://www.gnu.org/licenses/gpl.html>

¹⁶ <http://gcc.gnu.org/>

¹⁷ <http://www.opengl.org/>

¹⁸ <http://www.trolltech.com/>

¹⁹ <http://www.kdevelop.org/>

²⁰ <http://es.wikipedia.org/wiki/Bluefish>

software editor HTML multiplataforma POSIX²¹ y con licencia GPL, lo que le convierte en software libre. Está dirigido a diseñadores web experimentados y programadores, se enfoca en la edición de páginas dinámicas e interactivas. Es capaz de reconocer diversos lenguajes de programación y de marcas. NVU es un editor de páginas Web WYSIWYG²² multiplataforma basado en Mozilla Composer²³, pero de ejecución independiente. Añade características nuevas, como soporte integrado de CSS y mejor gestión del soporte FTP para actualización de los ficheros.

Para las animaciones 3D es recomendable Blender²⁴, herramienta utilizada en la facultad 10 (UCI) promotora del trabajo con herramientas libres. Es un programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita; pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre.

Los criterios analizados en este epígrafe son el resultado de las propuestas obtenidas en las entrevistas realizadas, según modelo de encuesta (Ver anexo 14), a los ingenieros Roberto Ferrer Obregón y Yonnys Pablo Martín Olivera del proyecto Galm en el cual cumplen el rol de scrum máster.

2.3 Estudio de la documentación de estrategias de migración

Para analizar las migraciones, la información se dividió en dos grupos. En el primer grupo se escogió el “**Plan Institucional de Liberación de Software y de Migración a Plataformas de Software Libre, Universidad de Los Andes**” por ser un centro universitario destacado de la República venezolana; el de Europa” **Directrices IDA de migración a software de fuentes abiertas**” por ser un documento muy completo, con una explicación detallada sobre qué herramientas implantar en cada caso. Según el Ing. Abel Meneses Abad los europeos han logrado mucho, si de migración se habla y se mantienen actualizando las migraciones según los cambios que realizan. En el segundo grupo se analizaran las guías desarrolladas en los años 2007 y 2008 en la UCI como trabajo de PRE-grado de estudiantes de la facultad 10.

²¹ <http://es.wikipedia.org/wiki/POSIX>

²² WYSIWYG: Se aplica a los procesadores de texto y otros editores de texto con formato

²³ http://es.wikipedia.org/wiki/Mozilla_Composer

²⁴ <http://es.wikipedia.org/wiki/Blender>

2.3.1 Directrices IDA25 de migración a software de fuentes abiertas²⁶

Los europeos (13) plantean que cualquier ejercicio de migración debe reflejar lo siguiente:

- I. **Una fase de definición del proyecto y de recopilación de datos, en la que se contemplen (13):**
 - a) La descripción del conjunto de condiciones iniciales relevantes consistentes, por ejemplo, en:
 - Arquitectura o arquitecturas de los sistemas.
 - Aplicaciones y sus datos asociados.
 - Protocolos y Norma empleados.
 - Hardware.
 - El entorno físico, como el ancho de banda de la red, la ubicación.
 - Los requisitos sociales como el idioma o idiomas y la capacitación del personal.
 - b) Un conjunto de condiciones finales con el mismo detalle.
 - c) Una descripción de cómo llegar de las condiciones iniciales a las condiciones finales.
- II. **Una justificación de la migración, incluido el coste asociado a la misma.**
- III. **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 costes usado en el plan.**
- IV. **Despliegue del plan.**
- V. **Seguimiento de la experiencia real en relación con el plan.**

Durante el proceso descrito anteriormente los europeos definen al punto 1 como el escenario en el que se realizará el proceso y explican las directrices para poder cumplirlo, además simplifican el estado inicial del proceso a uno solo (Entorno inicial usando Windows) y de los posibles estados finales eligen el uso de GNU/Linux completamente; no cuentan con la posibilidad de que el objetivo final de la migración (obtener un entorno FLOSS)²⁷ no llegue a cumplimentarse de manera

²⁵ Intercambio de Datos entre Administraciones

²⁶ Europa, 15 de octubre de 2003

²⁷ FLOSS: Free/ Libre Open Source Software

total y quede al final de este alguna estación con Windows por motivos específicos, es decir una migración parcial.

Por otro lado dividen las directrices para la migración en dos grupos: las directrices de gestión y las técnicas. Las primeras encargadas de regular todo el proceso del control, planificación de los recursos tanto humanos como materiales y la confección de una guía de migración. Mientras que la segunda se responsabiliza por los aspectos técnicos del proceso, tales como el estudio, selección de aplicaciones y los pasos necesarios para la implantación de las mismas.

Para lograr el despliegue del plan definen una sección en el capítulo 7: “Una vida más fácil” en la que mencionan la inclusión de herramientas libres en el entorno actual, para lograr una familiarización de los usuarios con las herramientas que posteriormente usarán en dicho entorno. Utilizar este enfoque y posibilitar la existencia de herramientas libres y privativas ayudará a la conversión de los formatos de archivos de una forma más sencilla. Muchas herramientas libres están disponibles en entornos privativos, como pueden ser el navegador Web Mozilla Firefox, la suite ofimática OpenOffice.org y el servidor Web Apache entre otros. A la vez afirman que los primeros cambios deberán ser en el servidor, siendo estos transparentes para los usuarios, y luego en los clientes para reducir al máximo las dificultades. La administración debe tomar una serie de medidas que contemplen el uso de estándares para el desarrollo de sitios Web, la utilización de lenguajes de programación portable para la creación de aplicaciones y la introducción y uso de formatos de fuente abierta y portable (PDF).

Aunque la guía es bastante extensa, se puede deducir que una vez realizado el despliegue, no refleja un procedimiento sólido de seguimiento del plan, por lo que se le podrían incorporar nuevos elementos al proceso propuesto. El incluirle supervisión por un líder que guiará la migración y controlara el desarrollo de cada fase lograría fiabilidad en el desarrollo del proceso. Este rol lo desempeñaría una persona con experiencias en otras migraciones y que conozca las necesidades de la empresa o institución inmersa en la migración.

2.3.2 Plan Institucional de Liberación de Software y de Migración a Plataformas de Software Libre²⁸

El Consejo de Computación Académica (CCA) de la Universidad de los Andes (ULA) para el plan define 6 etapas:

²⁸

Universidad de Los Andes, 2006-2010

- Etapa de diagnóstico
- Etapa de evaluación
- Etapa de capacitación
- Etapa de pilotaje
- Etapa de implementación
- Etapa de mantenimiento

La ULA, realizó un estudio de los principales procesos de la universidad, haciendo un levantamiento de información que abarca toda la estructura de la misma, definiendo los procesos críticos a enfrentar durante la migración, identificando estos como: correo electrónico, conectividad (routing y DNS), portales y repositorios, servicios bibliotecarios entre otros.

Para la etapa de implementación se definen dos aspectos importantes, el primero relacionado con los aspectos organizativos del plan de migración tecnológica, en la que se tratan un conjunto de elementos a considerar y las etapas del plan. El segundo, relacionado con temas de carácter técnico y orientado a programadores.

La guía de la ULA define algunos aspectos organizativos a tener en cuenta en la migración, como son (17):

- a) Evaluar el tipo de usuarios que participará en el proceso y el impacto corporativo que tienen en el plan de migración a Software Libre.
- b) Analizar el estado actual de la organización en lo referentes a sistema informáticos y uso de herramientas privativas.
- c) Evaluar las potenciales soluciones para la migración de los sistemas informáticos.
- d) Evaluar los distintos escenarios o ambientes de trabajo que existen en la organización en lo referente al uso de herramientas de productividad, para determinar las aplicaciones de SL que se requieren.
- e) Evaluar según el tipo de ambiente y usuario cuáles serían los planes de formación más adecuados, lo cual definirá los tipos de cursos y actividades de concientización que se puedan aplicar.
- f) Evaluar la disponibilidad de recursos humanos dentro de la organización, que pueda participar en el proceso de migración, tanto en los planes de formación (instructores, soportes técnicos, etc.) como en el soporte, instalación y mantenimiento de las aplicaciones de Software Libre, pues podría existir

muchos usuarios finales que no realicen estas actividades por lo menos en las etapas iniciales del proceso.

g) Evaluar los mecanismos más adecuados que faciliten el soporte, mantenimiento y atención de los usuarios en lo referente al uso de aplicaciones de Software Libre, tales como: creación de lista, foros y canales IRC, que faciliten a los usuarios el intercambio de información con expertos u otros usuarios, atención de los usuarios vía telefónica, email, Web, y atención en sitio, ante dudas o fallas de software que se puedan presentar; en este punto se podría considerar la creación, por aplicación de Software Libre, una cuenta de e-mail que sea moderada por un experto, creación de repositorios, entre otros, que faciliten la descarga y actualización de las aplicaciones de Software Libre, creación de portales instituciones que faciliten el adiestramiento de los usuarios en línea (cursos virtuales, etc.), acceso a manuales, ubicación de los recursos de Software Libre, acceso a noticias y eventos, distribución de aplicación y herramientas de productividad (Live CD. ULAnix), entre otros.

h) Establecer un plan de acciones concretas, durante el período comprendido en los años 2006 al 2008.

En este plan de acciones se podría requerir algunos de los siguientes datos e información (19):

- 1) Definir el número de usuarios que participarán en los planes de formación según su impacto corporativo en cada fase del plan de migración.
- 2) Establecer el número y tipos de sistemas informáticos que migrarán hacia aplicaciones de Software Libre anualmente.
- 3) Períodos en que se implementará el mecanismo de soporte de acuerdo a los recursos financieros disponibles, entre otros.

El CCA recomienda la implantación de proyectos pilotos (17) en determinadas facultades y unidades administrativas, con los objetivos de implementar el plan y ajusta algunos elementos del mismo de acuerdo a los resultados obtenidos. Propusieron además, un conjunto de etapas a tener en cuenta en la migración tecnológica de la Universidad de los Andes:

Etapa 1: La etapa se encarga de inventariar las unidades administrativas de la Universidad de los Andes, los sistemas informáticos y herramientas de productividad que se utilizan, así como también evaluar los sistemas informáticos, que están desarrollados bajo plataformas y aplicaciones de Software Libre, y cuáles no.

Etapa 2: Esta etapa consiste en definir estratégicamente los usuarios, que según su impacto corporativo, serán atendidos progresivamente, y los sistemas informáticos que pueden migrar a SL, durante el período comprendido entre los años 2006 al 2008. Adicionalmente se deben definir los distintos ambientes de trabajo, tanto a nivel de usuarios finales, como de desarrolladores de sistemas

informáticos, para evaluar potenciales soluciones que se podrían implementar con aplicaciones de Software Libre.

Etapa 3: Etapa que permite definir planes de adiestramiento sobre aplicaciones de Software Libre y las políticas comunicacionales y de promoción (charlas, foros informativos, etc.) sobre los beneficios de la integración en la Universidad de los Andes. Para ello se necesita: determinar y establecer prioridades de los usuarios que participarán en los planes de formación discriminados por los distintos períodos que dure el plan, definir los cursos que se dictarán, los recursos humanos, laboratorios, materiales de apoyo, etc. que se requieren, y los costos que están involucrados en los mismos.

Etapa 4: Se definen las estrategias necesarias para proceder a la distribución, instalación y soporte de las aplicaciones en los distintos ambientes de trabajo de las unidades administrativas de la Universidad de los Andes, así como también, definir las estrategias para consolidar los mecanismos de atención y soporte de los usuarios; ya que estos mecanismos implican costos para la organización.

Etapa 5: Permite la migración de los sistemas informáticos administrativos de la Universidad de los Andes, tanto del área académica como administrativa. El proceso se podría iniciar por aquellas aplicaciones menos críticas, y en la medida que se adquieren las pericias y habilidades respectivas sobre el manejo de aplicaciones de desarrollo de Software Libre, pasar a los sistemas informáticos más críticos. En las fases iniciales del plan de migración se deben asignar recursos humanos para determinar, evaluar, probar y depurar las aplicaciones de Software Libre que podrían ser utilizadas en el desarrollo de los sistemas informáticos.

Etapa 6: Para esta etapa en cada fase del proceso de migración se garantiza el mantenimiento y actualización de las aplicaciones y herramientas de productividad de Software Libre, ya que los desarrolladores de Software Libre frecuentemente realizan mejoras en sus aplicaciones, lo cual requiere de un soporte bien organizado y canales de promoción sobre las mismas, para que los usuarios tomen las medidas respectivas.

La estrategia desarrollada por la Universidad de los Andes se inicia con un levantamiento de todos los recursos disponibles hasta el fin del proceso de migración, asegurando la continuidad del soporte al proceso. La capacitación a los usuarios se brinda por los niveles de impacto que tengan estos, estableciendo así planes de formación en tres posibles niveles educacionales (para instructores en software libre, la formación de soporte técnico y para los usuarios finales), lo cual demuestra la importancia de la capacitación en el proceso de migración.

Otro elemento importante para la transición de un sistema es precisamente los mecanismos de soporte, en ese sentido propone la consolidación de comunidades virtuales para el soporte a las

herramientas, la creación y recolección de documentación disponibles en medios digitales de fácil acceso, consolidación de portales que permitan el acceso a esos recursos, el fortalecimiento del soporte técnico para asegurar la instalación y soporte de aplicaciones, centros de atención a los usuarios, para evitar fallas en ese sentido.

El plan no explica claramente como se produce el cambio de tecnología libre en sustitución a las privativas. La idoneidad se alcanzaría si definieran pasos para el cambio de SO. El utilizar un transito gradual, introduciendo algunas herramientas libres multiplataforma en las estaciones de trabajo, luego migrar solo algunas PC y por último realizar la instalación del SO escogido, abriría una nueva puerta a la sensibilización y familiarización con las herramientas que ofrece la tecnología libre.

2.3.3 Metodología para la migración a Software Libre de la UCI²⁹

Nuestro país puso en marcha en el año 2002 una estrategia para alcanzar la soberanía nacional de la isla, en la cual la UCI formaba parte de la misma. La facultad 10 estaría encargada de las soluciones relacionadas con los aspectos técnicos del proceso de migración y como parte este se realizó la "Metodología para la Migración a Software Libre de la UCI " en el trabajo de diploma del Ing. Ramón Paumer Samón. La estrategia para la migración descrita en la guía, propone el proceso de cambio en tres etapas (14):

- 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.

Cada etapa presenta un grupo tareas para dar cumplimiento a los objetivos trazados. La primera tarea de la etapa preparación es la fase de justificación del proceso, donde se presentarán los argumentos del por qué migrar, expresando las ventajas del cambio, ejemplos comparativos de los costos de utilización de las herramientas, la gran variedad de herramientas libres existentes y la calidad de las mismas con respecto a las privativas.

²⁹La Habana, Cuba, 2006-2007

Así mismo define la fase de planificación del proceso, que contiene tareas significativas como la realización de un levantamiento informático a gran escala, que involucre el hardware y software de la institución a migrar; la elaboración de un plan de migración; la sensibilización de la institución y preparación de la misma para acometer el proceso de cambio; la formación y capacitación del personal para asegurar el soporte técnico mientras dure. Después del proceso de migración, la creación y puesta en funcionamiento de un portal Web para el soporte, tanto de actualizaciones de software como para la capacitación.

Para culminar la etapa de preparación, la metodología propuesta sugiere la fase de pruebas piloto, implementando el plan propuesto en el punto anterior en un ambiente más reducido para permitir hacer una valoración del mismo e introducir posibles cambios. La propuesta para la UCI resulta interesante pues está orientada al peor de los casos, o sea brinda una solución de migración sobre cómo lograr el proceso sin un apoyo de gestión adecuado por parte de la institución, lo que sin dudas resulta difícil.

En la etapa de migración se exponen dos tareas esenciales y unidas a este, otro gran número de sub.-tareas (14):

- 1) La creación de una unidad de formación y desarrollo, compuesta en su gran mayoría por el personal formado durante la fase de preparación y que será la encargada de:
 - La creación de los puestos de trabajos necesarios para acometer cualquier tipo de acción, convirtiéndose estos en los primeros escritorios migrados, lo que posibilitará el comienzo de las pruebas a las que constantemente deberán estar sometidas las estaciones de trabajo migradas.
 - La generación de entusiasmo hacia el proyecto mediante la exposición de los beneficios del uso de este tipo de sistemas, la necesidad de su implementación a todos los niveles y la introducción de este tipo de filosofía en aquellos usuarios, que no conocen del tema mediante la promoción. El desarrollo de aplicaciones y proyectos que incluirá el desarrollo y personalización de distribuciones GNU/Linux cada vez más estables y adaptadas a las necesidades de los usuarios, la realización de las primeras aplicaciones y programas en la nueva plataforma y la migración de aplicaciones específicas que sean utilizadas como parte de la gestión de determinados recursos.
 - La recogida de comentarios y sugerencias se convertirá en la retroalimentación que permita el mejor acabado de las distribuciones, que indique los cambios que deban hacerse a ciertas configuraciones incómodas para los usuarios y que permitirá saber cuál es el momento adecuado para la migración de nuevos servicios.

- La realización de la migración estará marcada por la introducción de un servicio migrado, con su correspondiente configuración y el movimiento de los datos que posibilitará su funcionamiento. Llegado este punto se tendrá en cuenta la realización de pruebas piloto y la constante retroalimentación, que será el factor determinante en la introducción de nuevos servicios.
- 2) La capacitación, que se convierte en baluarte fundamental para la continuidad del proceso en esta etapa, debido a que una vez iniciada la migración, será cada vez mayor el número de especialistas que se necesite para la realización de cualquier tarea relacionada con esta. Es por eso que se incluye en esta etapa:
- La definición del programa de formación, documento que estará sujeto a constantes modificaciones introducidas durante el desarrollo del proceso, lo que garantizará que la formación sea sobre la base de los cambios introducidos y no sobre modelos obsoletos o poco usados.
 - El registro de grupos de usuarios del nuevo sistema, que posibilitará la detección de los próximos usuarios y sistemas más propicios para la migración. La formación de usuarios del nuevo sistema, que constituirá el pilar fundamental para la creación de las nuevas Unidades de Formación y Desarrollo, estos tendrán a su cargo otras tareas no menos importantes dentro del proceso de migración y la creación de nuevos puestos de trabajos, entendiéndose la migración de nuevas estaciones clientes, con su afinada y validada configuración, sometidas a las pruebas recurrentes necesarias para la detección y corrección de errores. Esta etapa es sin duda alguna la fundamental en el proceso de migración, será la que enmarcará el fin del software privativo en la institución. Como fase final para la culminación de la migración la guía propone la especialización de las unidades de formación y desarrollo, lo cual implicará dejar de hacer efectivo el proceso de migración y la creación de nuevos puestos de trabajo con un conjunto de tareas entre las que se pueden identificar:
 - Continuar elaborando la documentación, tutoriales, manuales de instalación y de uso, y se añade todo aquello que facilite los procesos de auto-aprendizaje, educación a distancia y teleformación. Incrementar el proceso de recogida de opiniones a fin de trabajar con más exactitud, la personalización de las distribuciones. Sumar el mayor personal posible al desarrollo de aplicaciones, lo cual contemplará el incremento de la producción sobre la plataforma libre y la migración de aquellas aplicaciones específicas

que formen parte aún de la gestión de determinados recursos en algunos de los departamentos.

- Asumir las labores de soporte a través de la mayor cantidad de vías posibles, ya sea online, mediante foros de discusión, portales de Software Libre, servidores de documentación y repositorios, personalmente, mediante la habilitación de un lugar con los recursos indispensables (personal con los conocimientos) para que los usuarios acudan y reciban toda la información que necesiten o cualquier otra iniciativa. El perfeccionamiento del programa de formación, adaptando su contenido a los nuevos aportes realizados durante la migración o posterior a esta. Además de valorar las tecnologías que se utilizan y la posibilidad de actualizarlas o sustituirlas por otras más acabadas. Continuar registrando y formando los usuarios, ya motivados por la necesidad de utilizar el nuevo entorno como única alternativa para el correcto desempeño en su entorno laboral.

La propuesta culmina haciendo un análisis crítico del proceso de migración a Software Libre en la facultad 10 de la UCI destacando los logros y tropiezos de la misma con respecto a la propuesta antes comentada.

La confección de esta guía de migración resulta de suma importancia para el estudio y futura propuesta, pues se encuentra adaptada a las condiciones de una universidad cubana y es la primera guía de migración a Software Libre realizada en el centro, de la que se tenga conocimiento. Se determina en la fase de preparación un levantamiento informático dependiendo de las características generales del centro y no se incluyen las necesidades particulares para cada facultad. La definición de cómo llevar a cabo el cambio de tecnología privativa a libre no se pauta o explica claramente. La propuesta metodológica crecería en importancia si se definieran como mínimo tres acciones en este sentido:

- Realizar la instalación de herramientas multiplataforma libres sobre Windows e ir estudiando el comportamiento que manifiesta el personal del centro hacia estas. Paralelamente implantar, en torno a esta acción, una estrategia de concientización sobre el uso que puedan dárseles.
- Migrar sólo un por ciento de las máquinas y luego recoger información referente a la aceptación que ha tenido esta acción.
- Migrar completamente todas las estaciones y servicios.

2.3.4 Metodología para la migración a software libre de las universidades del ministerio de educación superior (MES)³⁰

Este documento, al igual que el anterior, fue elaborado en la UCI como trabajo de diploma. Surge a raíz de satisfacer la siguiente incógnita: ¿Cómo migrar las universidades cubanas a Software Libre? y la respuesta, aunque corta y sencilla; queda plasmada en la metodología de migración propuesta para el MES. Para lograr una correcta Migración a Software Libre como se mencionó anteriormente, se definieron seis flujos de trabajo, cuyos nombres y características generales son:(15)

- 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.

De manera análoga se proponen varias etapas por las que pasará el proceso dejando cada una de ellas un resultado, que servirá de entrada a la próxima (15):

- Preparación: Etapa en la que se realizarán las tareas de recopilación de datos y se lanzará una primera versión del plan 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.

³⁰Ciudad de la Habana, Cuba, 2007-2008

- Migración Total: Cada vez que se ejecute una iteración de esta fase la cantidad de FLOSS 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.

Los flujos de trabajo y las etapas las proponen de manera iterativa e interrelacionadas, donde en determinadas etapas los flujos tendrán mayor esfuerzo, siempre aclarando que en todas estarán presentes. Luego define las tareas de cada flujo de trabajo, explicando cada una de las iteraciones.

La evaluación es el primer flujo de trabajo durante la migración para el MES, es donde se realiza una valoración de todos los procesos y tecnología presentes, pues define elementos que serán la base para esta migración, este flujo constituye un hito fundamental en el proceso y se realiza una sola vez, su mayor impacto lo dirigen a la etapa de preparación. Luego de la realización de una evaluación de los procesos de la institución. El segundo flujo es diseño, donde se define el plan de migración de acuerdo a las necesidades requeridas como resultados de la evaluación, dicho flujo tiene su mayor impacto en la etapa de preparación aunque también estará presente en la migración parcial.

Se explica, con el objetivo de validar el contenido y crear las configuraciones correctas para las aplicaciones, la definición y creación de un ambiente real de pruebas en un marco reducido, el cual permitirá la retroalimentación y ajuste de algunas variables del plan propuesto. Haciendo extensiva la migración con un número de fallos mínimo y el encargado de ello es el flujo de pilotos, cuyo mayor impacto está en la etapa de la migración parcial. Específicamente el flujo de formación propone 4 niveles básicos de usuarios, a los que estaría dirigida la capacitación.

La implementación es el flujo de trabajo donde se marca el fin de los sistemas privativos, declarando la introducción de los libres. Los mayores esfuerzos dentro de la implementación los concentran en la fase de migración total e irá disminuyendo a medida que se avance en esta, quedando al final esta tarea en la etapa de consolidación. El último flujo propuesto es al igual que la formación, uno de los más extensos del proceso, la asistencia y soporte técnico, estarán presentes a lo largo de todo el tiempo que dure el cambio, comenzando con la instalación de la primera computadora e irá disminuyendo a medida que avanza el proceso. Es importante destacar que el soporte nunca deberá terminar ya que siempre habrá problemas a los cuales darle solución, sólo que estos tienden al decremento. Su objetivo principal es brindar el soporte para el mantenimiento de todas las infraestructuras y al personal.

Este documento propone una metodología de migración en la cual el error está en darle el esfuerzo a los flujos de trabajo y no a las iteraciones definidas dentro de estos. En las migraciones de carácter iterativo, aun cuando a un flujo se le dé el esfuerzo que le corresponde, si el peso que se le

atribuye y el tiempo, de una sola iteración (en cualquier flujo), no fuesen cumplidos ocasionaría errores en toda la migración. El esfuerzo destinado para cada flujo de trabajo debería estar distribuido por las iteraciones que contengan cada uno, definiéndose en la planificación de la migración y destinar tareas en función al seguimiento y control del esfuerzo destinado para las iteraciones. La entrega de resultados palpables de los flujos de trabajo cumplimentaría el tránsito de una iteración a otra.

2.4 Conclusiones del capítulo

En la propuesta se recomienda el uso de un grupo de herramientas libres como sustitución a las privativas, herramientas estas que han sido probadas en la universidad, aunque esto no implique el uso exclusivo de ellas pues hay otras que pueden utilizarse. Se hizo necesario consultar algunas guías migratorias, y aunque están en función de un centro o institución específico, definen pasos para realizar la migración, que son comunes en todos los casos. Los métodos utilizados resultan interesantes y novedosos, pues determinan etapas y fases del proceso de migración a los cuales tributa la estrategia de migración del próximo capítulo.

Es imprescindible destacar determinados criterios que contribuyeron a que no se utilizara alguna de estas guías, para los proyectos de software educativo. Algunas obvian pasos y explicaciones importantes que se deberían contemplar, por ejemplo: las directrices IDA de Europa³¹ no definen un conjunto de pasos para el proceso de post-migración. Por otra parte los venezolanos³² definen una guía muy completa, aunque no describen claramente el cambio de un sistema a otro, ni las aplicaciones propias del software privativo. La guía para la UCI no establece en la fase de preparación un levantamiento del personal que satisfaga las necesidades de las facultades, lo que es necesario para la creación de los planes de formación más adecuados a las características de cada sección del centro, ni la implantación de herramientas libres sobre Windows, de la cual depende la aceptación de la migración en los usuarios que están implicados en el proceso migratorio.

³¹Directrices IDA de migración a software de fuentes abiertas. Europa, 15 de octubre de 2003.

³²Plan Institucional de Liberación de Software y de Migración a Plataformas de Software Libre, Universidad de Los Andes, 2006-2010.

CAPITULO 3: ESTRATEGIA DE MIGRACIÓN A SOFTWARE LIBRE PARA LOS PROYECTOS DE SOFTWARE EDUCATIVO

Luego del análisis de los conceptos fundamentales del software libre, del proceso de desarrollo de software educativo, especialmente de las herramientas para gestión y desarrollo de estos proyectos y del estudio de procedimientos y guías de migración tratados en el capítulo anterior, se plantea una estrategia para la migración a software libre organizada por fases definiendo en cada una de estas una serie de flujos de trabajo que permitan una vez concluidos, pasar a una próxima fase, hasta culminar y sostenerse en la continua actualización de las herramientas de trabajo.

3.1 Generalidades de la estrategia.

Toda estrategia de migración debe basarse en una buena planificación, por muy sencilla que sea esta, se debe hacer pruebas a pequeña escala antes de implementar cualquier cambio, más aún en el ámbito de una universidad y empresa desarrolladora de software, en la cual el proceso se llevará a cabo con proyectos reales los cuales tienen compromisos de entregas contra tiempo y dinero. Es necesario disponer de un equipo de respaldo durante el desarrollo de la migración. Concientizar a los usuarios en el uso de las nuevas herramientas y sistema operativo, así como supervisar toda la marcha del proceso.

La alternativa es implementar el proceso de migración mediante una transición por etapas o fases, ejecutándose en cada una de ellas una serie de actividades. Se propone migrar, del antiguo sistema al nuevo, en grupos funcionales, los cuales se podrían definir de acuerdo a los diferentes tipos de proyectos que se desarrollan en el Polo de SWE, siempre teniendo en cuenta las particularidades de cada uno de estos y sin poner en riesgo la producción.

Se define una estrategia evolutiva e iterativa que tendrá definidas cuatro (4) fases y seis (6) flujos de trabajo, haciendo una analogía, se puede comparar con la organización en fases y flujos que define RUP en su proceso de desarrollo, permitiendo lograr la eficacia deseada y obteniendo resultados parciales, de forma tal que el proceso se vaya evaluando y refinando de una iteración a otra.

La **Preparación** es la fase inicial, en la que se realizarán todas las tareas de recopilación de datos que permitirán caracterizar el trabajo de los proyectos, las herramientas que se emplean en el desarrollo de estos y sus análogas en SWL, además de la capacitación y concientización del personal. La segunda fase es la **Migración Parcial**, donde se irán introduciendo parcialmente las herramientas libres en los proyectos. La tercera fase es la **Migración Total**, la actividad que requerirá mayor

esfuerzo, en esta etapa será la instalación la que deberá haber comenzado al finalizar la primera fase y se extenderá hasta el principio de la última. El objetivo final será la eliminación de las herramientas privativas y la implantación total de SWL. El principal flujo de trabajo de la última etapa, **Consolidación**, será el proceso de soporte a la migración el cual lo garantizará un equipo destinado para esto por un tiempo determinado hasta que el personal este lo suficiente capacitado y pueda crear uno propio.

Los flujos de trabajo contendrán una serie de actividades que tributarán a los objetivos de la migración. El **levantamiento informático** permitirá la recogida de datos de interés de todos los procesos, tecnología y personal para adaptarlas a la situación actual. El **Diseño** se encargará de conformar un plan de migración conforme a las necesidades, tomando como partida el resultado anterior. Poner en marcha el plan en un ambiente real de pruebas, es una tarea bien difícil y se realizará en el flujo de **Pilotos**. La **Formación** incluirá la capacitación del personal y certificación del mismo. La instalación y migración definitiva de servicios y estaciones de trabajo a Software Libre correrá a cargo de la **Implantación**. Como último flujo de trabajo está la **Supervisión y Soporte** que brindará atención y soporte a las infraestructuras, servicios instalados y al personal.

Siguiendo esta idea, cada fase en principio debe tener más de una (1) iteración. En la planificación se proveerá cada iteración, de forma tal que se defina el inicio, el final y los objetivos que se persiguen. Las iteraciones deben ser de períodos cortos en los cuales se obtengan resultados de forma rápida y de tal manera que se corrijan los errores que puedan aparecer en el proceso. Un ejemplo clásico en una Fase es la formación o capacitación donde la primera (1ra) iteración se deberá hacer al propio grupo que servirá de soporte y luego se planificarán otras para el resto del personal. (Ver Anexo 11)

Siempre se debe dejar constancia de los resultados obtenidos, los cuales en algunos casos son imprescindibles para el inicio a la próxima tarea. Estos resultados son entonces los artefactos que se deberán generar en cada una de las actividades propuestas, tal como se ilustra en la tabla que se muestra en el Anexo 3. De esta forma en la primera fase el grupo de migración será capaz de realizar determinadas tareas que concluirán con un resultado y este se presentará en un grupo de planillas, las que luego serán la entrada a la próxima actividad y así sucesivamente, hasta que termine el proceso. (Ver además el Anexo 13)

3.1.1 Aspectos organizativos

Es necesario definir algunos aspectos organizativos, los cuales van a determinar la importancia de muchas de las acciones que se desarrollarán en el proceso migratorio, estos son:

- Evaluar el tipo de usuarios que participará en el proceso y el impacto que tienen en el plan de migración a SWL.
- Analizar el estado actual de los proyectos en lo referente a sistemas informáticos y uso de herramientas privativas, para determinar las aplicaciones de SWL que se requieren.
- Analizar cuáles serían los planes de formación más adecuados atendiendo al ambiente y tipo de usuario, lo cual además, definirá los tipos de cursos y actividades de concientización que se puedan aplicar.
- Analizar la disponibilidad de recursos humanos dentro de la organización, que puedan participar en el proceso de migración, tanto en los planes de formación (instructores, soportes técnicos, etc.) como en el soporte, instalación y mantenimiento de las aplicaciones de SWL.
- Evaluar los mecanismos más adecuados que faciliten el soporte, mantenimiento y atención de los usuarios en lo referente al uso de aplicaciones de SWL.
- Utilizar las listas y foros que ya existen en la Universidad, que faciliten a los usuarios el intercambio de información con expertos u otros usuarios de SWL, ante dudas o fallas de software que se puedan presentar.
- Utilizar los repositorios, que facilitan la descarga y actualización de las aplicaciones de SWL para el desarrollo de aplicaciones de SWE, acceso a manuales y documentación para el uso de las aplicaciones.
- Establecer un plan de acciones concretas, durante el período que se realice la migración.

3.1.2 Grupo de Migración

Antes de realizar cualquier actividad que tribute a una migración se deben seleccionar los responsables en llevarla a cabo, aquellas personas que cumplirán un rol determinado dentro de este

proceso. La dimensión del grupo estará dada por la cantidad de proyectos con que cuente el Polo de SWE. Este grupo debe tener los siguientes roles:

- **Jefes de proyecto:** Los jefes de proyectos deben ser los encargados de proporcionar la información necesaria de su proyecto que se necesite para el proceso de migración, además debe seguir de cerca el proceso y estar pendiente de problemas o fallas en la producción una vez finalizada la fase de migración total. Como es un jefe por proyecto el número de estos en el grupo de migración dependerá de la cantidad de proyectos de SWE en existencia.
- **Técnicos:** Su mayor esfuerzo será en las fases de migración total, pero desempeñará un papel importante en la fase de migración parcial y consolidación. Debe tener conocimientos referentes a la instalación de sistemas, asistencia técnica y soporte informático. Se recomienda que sean técnicos de laboratorios o estudiantes que la actividad docente les permita disponer de tiempo suficiente para las tareas que se les asignarán, se sugiere buscar estudiantes de cuarto año que tengan algún conocimiento de software libre.
- **Expertos en Linux:** Ellos serán los principales encargados de la formación y capacitación, así como de la investigación y definición de las herramientas libres y sistema operativo a instalar. El mayor esfuerzo del personal que este ejerciendo este rol será en la actividad de formación, lo cual no indica que no realice algunas tareas en otras fases. Serán profesores con experiencia en software libre y uno de estos expertos será el jefe de grupo de migración.
- **Planificador:** Su mayor esfuerzo será en la fase de preparación, debe tener experiencia en la planificación de tareas. Será el encargado de realizar el diseño y planificación del proceso de migración, laborará también en el pilotaje junto al equipo. Puede ser un profesor, no necesariamente con perfil técnico.
- **Jefe del grupo de migración:** Estará a cargo de la supervisión del proceso de migración y debe ser un experto en términos de SWL, con experiencia en procesos de migración.

3.2 Descripción de los flujos de trabajo

Para comenzar con la migración se debe seguir el cumplimiento lógico de un conjunto de actividades contenidas en los flujos de trabajo, los cuales tendrán un esfuerzo mayor o menor según la fase donde se encuentren. Las tareas permitirán la elaboración de una serie de artefactos como

resultado de cada flujo, que serán llenados por los responsables del grupo de migración (Ver Anexo19).

3.2.1 Flujo de trabajo: Levantamiento informático

El mayor esfuerzo del flujo de levantamiento informático se desarrollará en la fase de preparación y disminuirá gradualmente, teniendo una leve presencia en las restantes. Si se analiza la posibilidad de la adopción de algún nuevo proyecto, aun cuando se esté inmerso en otra fase, lo anteriormente expuesto eliminará la idea de detener el proceso para incluirlo.

Este flujo permitirá la recogida de datos de interés para la definición del estado actual de las condiciones de los proyectos. Es importante destacar que no se hace necesario definir características de hardware puesto que actualmente todos los proyectos de software educativo tienen las condiciones y requerimientos idóneos para la migración a software libre, lo cual se traduce a que perfectamente se puede instalar SWL con los recursos existentes.

Aquí no existen artefactos de entrada, puesto que marca el inicio de la migración, por el contrario serán los resultados que se deriven de los esfuerzos realizados, la entrada a los próximos flujos de trabajo. Se puede por lo tanto afirmar que si se está laborando en este flujo, ya ha comenzado el proceso de migración.

Actividades y Tareas: Para este flujo se determinan un conjunto de tareas que tributan a la elaboración de las actividades correspondientes a la recogida de datos de interés. Estas acciones serán ejecutadas por quienes ocupen los siguientes roles del grupo de migración: jefes de los proyectos inmersos en el proceso, el jefe del grupo de migración y los técnicos.

Recopilar los datos informáticos

- Describir del trabajo con herramientas privativas
- Describir del trabajo con herramientas libres.
- Estudiar las herramientas libres análogas.
- Determinar las herramientas con mayor importancia para el proyecto

Recopilar los datos del personal

- Definir la categoría del personal
- Determinar el nivel cognoscitivo con respecto al SWL.
- Identificar el rol de cada trabajador del proyecto.

Proponer herramientas libres para el trabajo de los proyectos

Listar las herramientas libres que sustituirán las privativas
Justificar la selección

Artefactos de salida: El llenar correctamente las plantillas que a continuación se explicaran, permitirá la recogida de datos de interés para la definición del estado actual de las condiciones de los proyectos. Se deben elaborar las siguientes plantillas para recogida de datos que se debe hacer en los proyectos:

La **plantilla**³³ de esta tarea se nombra **Tecnologías utilizadas** y recogerá los siguientes datos:

H.P: Se definirán las herramientas privativas que se utilizan en el proyecto.

H.L: Se definirán las herramientas libres que se utilizan en el proyecto.

Productividad: Determinará el grado de importancia y uso de una herramienta dada en el trabajo productivo. Se medirá el rango de 1 a 5.

En esta plantilla los responsables (Ver Anexo 3) del flujo serán capaces de explicar cómo se utilizan las herramientas en la producción de SWE y recoger un listado de todas, identificándolas por privativas o libres. Esto da la posibilidad de que en la tarea de Diseño donde se van a proponer las herramientas libres, estén acorde y permitan realizar correctamente la producción de SWE, de la misma manera que lo hacían las antiguas.

Está además la **plantilla** nombrada **Recursos Humanos** y permitirá recoger los siguientes datos:

Categoría: Para conocer si es estudiante o profesor.

Nombre: Se debe escribir claramente el nombre y los dos apellidos.

Conocimientos: Recoger los curso que se tengan de SWL.

Desempeño: Determinar el rol que cumple en el proyecto.

Los responsables(Ver Anexo 3) de esta tarea al recoger la evaluación del personal en el proyecto sobre conocimientos que tengan en SWL, brindaran la posibilidad de que en la tarea de Diseño de esta fase, no se le tenga que dar capacitación a un individuo que ya está calificado en una determinada herramienta. Se logra entonces economizar tiempo para la tarea de formación de la próxima fase, la cual es una de las que más tiempo de duración tendrá. Pero además se logra incrementar el número de expertos, porque este individuo puede ser incluido en el grupo de migración, o dar apoyo al trabajo de capacitación del personal.

Como se hace un estudio de las herramientas análogas es por lo tanto dejar claro cómo se irá introduciendo y para ello está la **plantilla Plan de Instalación de las herramientas libres**. (Ver Anexo

³³ Puede encontrar todas las plantillas que se nombran en el adjunto a este documento

3) En esta se deben determinar los pasos a realizar en las tres fases de instalación que expone el flujo de trabajo Implantación y la fundamentación.

3.2.2 Flujo de trabajo: Diseño

Luego de la realización de una evaluación de todos los procesos del área, el segundo flujo será el diseño, donde se definirá el plan de migración de acuerdo a las necesidades requeridas, dicho flujo tiene su mayor impacto en la etapa de preparación, pero aun así, será redefinido en la de migración parcial. Se evitará con esto, posibles problemas o cambios en el diseño de la migración.

Actividades y tareas: A continuación se muestran las actividades de este flujo, cada una con sus respectivas tareas. Estas acciones la realizarán los siguientes roles del grupo de migración: Expertos en SWL, el planificador y los técnicos.

Elaborar del plan de migración y la ruta crítica

Identificar y fundamentar la duración del proceso.

Identificar las iteraciones de cada flujo

Realizar un cronograma de las Actividades y Tareas de migración.

Diseñar la configuración

Determinar los pasos para la configuración

Modificar el plan de migración y la ruta crítica

Listar los problemas de planificación de la migración durante las pruebas a menor escala.

Re planificar las tareas del cronograma

Definir la ruta crítica más aceptada según las pruebas y justificar

Modificar la configuración

Listar los problemas de configuración durante las pruebas a menor escala.

Re planificar los pasos de configuración e instalación

Fundamentar los cambios

Artefactos de entrada: Para este flujo se necesitara utilizar las plantillas de Tecnologías utilizadas, la de Recursos Humanos y del Plan de Instalación de las herramientas libres. Puesto que contienen información vital para la planificación del proceso.

Artefactos de salida: Uno de los resultados será, la **plantilla Plan de migración y ruta crítica** (Ver Anexo 3), el cual será realizado por un grupo de expertos SWL y el planificador. Este contendrá un cronograma que se creará definiendo dentro del rango de tiempo determinado para cada fase, las fechas de inicio y fin de cada una de las tareas, propiciando entonces dar cumplimiento a todas las tareas dentro de ese rango y la supervisión de las acciones en tiempo. Además determinará la ruta crítica para el desarrollo de la migración y la planificación de las iteraciones de cada flujo y tarea. Esta plantilla será de gran importancia en el transcurso del proceso, puesto que se define el tiempo de duración de cada tarea y las iteraciones necesarias. El líder del grupo de migración será el encargado de velar por su cumplimiento; pero en el caso de contingencias en la planificación se propone una nueva reestructuración y se continúa con el proceso.

El otro, la **plantilla** que se nombra **Configuración propuesta** (Ver Anexo 3) la cual permitirá explicar la configuración que se le debe dar a cada estación de trabajo, para lograr un estándar en el trabajo, que permita agilidad en el proceso.

3.2.3 Flujo de trabajo: Pilotaje

Una vez que se haya concluido con la fase de preparación, con el objetivo de validar su contenido y crear las configuraciones correctas para las aplicaciones es importante crear un ambiente real de pruebas en un marco reducido, mediante este se hará una retroalimentación y se ajustarán los resultados del diseño para poder hacer extensiva la migración con la seguridad de que el número de fallos va a ser mínimo. Se realizarán pruebas del trabajo con las herramientas propuestas para el proyecto, las que permitan determinar si se ajustan a las características de desarrollo del proyecto, cuyo mayor impacto está en la etapa de la migración parcial.

Actividades y Tareas: Podrá encontrar las actividades de este flujo, así como las tareas que permitirán el cumplimiento de las mismas. Estas acciones la realizarán los siguientes roles del grupo de migración: jefes de los proyectos inmersos en el proceso, el jefe del grupo de migración, los técnicos y los expertos en SWL.

Implantar el plan de migración a menor escala

Estudiar el Plan de migración propuesto

Realizar cada iteración propuesta en el Plan

Recoger los elementos importantes para introducir mejoras en los distintos puntos de la guía de migración, para ello pueden utilizarse diversos métodos como la encuesta, conversaciones con los usuarios y revisión de los sistemas en funcionamiento.

Validar la configuración

- Definir los criterios y pruebas que definirá la validez
- Listar los problemas de cada paso de la configuración
- Proponer la solución de cada problema.

Validar el plan de migración y la ruta crítica

- Definir los criterios y pruebas que definirá la validez
- Listar los problemas de cada iteración
- Proponer y aplicar soluciones a los problemas

Artefactos de entrada: Para poder realizar las actividades aquí encomendadas se hace necesario trabajar con las plantillas de Plan de migración y Configuración propuesta. Al implantar el proceso a menor escala es necesario registrarse por estos documentos, permitiendo definir los puntos de fallos que pueda tener. De esta manera se logrará validar estos dos, que serán la guía fundamental para el tiempo que reste de migración.

Artefactos de salida: De surgir algún tipo de falla será registrada en la **plantilla** titulada **Reportes de fallas** (Ver Anexo 3), puesto que ésta se encargará de recoger todo tipo de errores que puedan surgir y la solución que se le dé a los mismos, desde ésta tarea hasta que concluya el proceso de migración. Permitiendo que una vez solucionados los problemas vayan quedando documentados, por si volviese a ocurrir.

Lo más importante de ésta tarea es que permite desarrollar la base del conocimiento necesaria para obtener una planificación de la migración replicable a mayor escala, eficaz y a corto plazo. Además se debe documentar utilizando la **planilla** que se nombra **Validación de la configuración propuesta y el plan de migración** (Ver Anexo 3) en la que se determinan las pruebas que se deben realizar en la propuesta de configuración y plan de migración, así como los criterios de validez de la misma. Este documento será el que dará la certeza de que se puede dar inicio a la siguiente fase o si es necesario cambiar la configuración y/o el plan de migración y detenerse en este punto hasta que la propuesta sea válida.

3.2.4 Flujo de trabajo: Formación

Inicialmente implantar un sistema de capacitación según la planificación en la fase de Preparación con la documentación necesaria para estos cursos, porque de este modo los cursos que

se impartirán estarán en función de las necesidades reales de cada proyecto. En la medida que se van dando los cursos se realiza la instalación de éstas en las estaciones de trabajo, lo que brinda la posibilidad de que los usuarios vayan relacionándose con estas herramientas mediante un proceso paulatino. El objetivo es que una vez concluido cada uno de los cursos, el usuario tenga dominio de la herramienta, por lo que la evaluación sistémica de los conocimientos adquiridos dentro de los cursos brindaría seguridad para el posterior trabajo productivo que debe desarrollar ese personal una vez realizada la migración.

Este flujo estará presente en toda la migración. Al pasar los usuarios por las distintas iteraciones y fases, deben adquirir el conocimiento básico al menos para el trabajo con el sistema, por ejemplo:

Al pasar por la iteración perteneciente a la culminación del pilotaje (en la fase de Migración Parcial) y uso de herramientas libres sobre Windows los usuarios involucrados deben saber ya trabajar con algunas herramientas de las propuestas en el Anexo 2.

Al llegar al final de la fase de migración total ya todos los usuarios involucrados en el proceso deberán tener al menos los conocimientos básicos para el trabajo con las nuevas herramientas.

Actividades y tareas: A continuación se definen las actividades de este flujo de trabajo y las atareas a realizar en cada una de ellas. Estas acciones la realizarán los siguientes roles del grupo de migración: el jefe del grupo de migración, los expertos en SWL y el planificador.

Justificar del proceso

Identificar las ventajas y desventajas que proporciona el SWL frente al privativo.

Definir los métodos de sensibilización adecuados de acuerdo a la diversidad del personal que participará en el proceso de migración en función a su categoría, nivel de conocimiento y rol.

Ejecutar el proceso de sensibilización.

Elaborar planes de formación y certificación

Elaborar programas analíticos de acuerdo a los planes de formación

Preparar los cursos

Planificar la documentación

Impartir los cursos por los niveles del personal

Agrupar al personal por niveles

Orientar la documentación: que puede ser complementaria o necesaria para los cursos y el objetivo es que todos los involucrados en éste flujo puedan continuar el estudio de las herramientas, así como llegar a tener un dominio más profundo de las mismas. Los documentos tienen que estar publicados en algún sitio al cual el personal debe tener acceso, deben ser legibles y en formatos adecuados para todas las estaciones.

Comenzar la capacitación en la fecha designada por el cronograma

Los niveles del personal por los que se puede agrupar al personal son los siguientes:

Formación de instructores en Software Libre: El objetivo de estos planes de formación es preparar un grupo de instructores que faciliten el proceso de formación de herramientas libres, tanto del personal técnico como de los usuarios finales. Los instructores recibirán cursos prácticos y teóricos de las herramientas que serán utilizadas en los cursos de soporte técnico y usuarios finales.

Formación de soporte técnico: Los planes de formación orientados a este contingente tienen la finalidad de preparar un conjunto de personas para dar soporte a los usuarios cuando éstos los necesiten a través de diversos métodos, estos usuarios recibirán un plan de formación que contendrá cursos prácticos y teóricos, así como que se les indicarán los mecanismos más adecuados para acceder a los recursos de software libre que les garanticen brindar soporte efectivamente.

Formación al personal del proyecto: Este tipo de formación puede ser orientada a desarrolladores o para los usuarios finales.

Para la capacitación de los desarrolladores de cada proyecto, los planes deben contener un grado mayor de conocimientos tanto prácticos como teóricos orientados a este tipo de personas, por lo tanto incluirá cursos de programación en los lenguajes elegidos y sobre el que se desarrollarán las futuras aplicaciones de la institución, se es del criterio de usar como se enuncia en el flujo de asistencia técnica lenguajes y librerías multi-plataformas.

La finalidad de estos planes de formación tienen como objetivo formar en software libre a los usuarios finales que usarán las aplicaciones instaladas en el nuevo sistema, para ello se prepararán de igual manera cursos prácticos y teóricos a la medida de las herramientas utilizadas en éste grupo pueden incorporarse personal administrativo, algún número de estudiantes sin vínculo a proyecto con expectativas de estar en alguno.

Artefactos de entrada: Para poder darle inicio a las actividades de este flujo se hace necesario analizar la plantilla Tecnologías utilizadas, la de Plan de instalación de las herramientas libres y la de Recursos Humanos, puesto que estas engloban los datos de interés para realizar las tareas necesarias en este flujo, ya que dependiendo de las herramientas que anteriormente se estaban utilizando se instalarán las necesarias para los cursos y hay que agrupar al personal por los niveles descritos.

Artefactos de salida: Para lograr la sensibilización es necesario explicar el por qué del cambio y otros argumentos de interés, la **plantilla** que se nombra ***Justificación del proceso de migración*** recoge precisamente esos datos (Ver Anexo 3):

¿Por qué migrar?

Ventajas del cambio.

Herramientas libres existentes

Calidad de las herramientas libres sobre las privativas

Esta planilla permitirá definir los argumentos que propiciaron la migración, poder regir la explicación de igual forma para todo el personal y lograr la sensibilización del mismo con el proceso que se ha de desarrollar. A éste documento se le pueden adicionar nuevos criterios (siempre y cuando todo el grupo de migración esté de acuerdo); pero como mínimo se hace indispensable que los antes enumerados aparezcan.

Para poder impartir un curso se hace necesario que los responsables de ésta tarea trabajen con la **plantilla Planes de Formación** (Ver Anexo 3). Se necesita determinar desde la información recogida anteriormente, como planificar los cursos y los profesores o alumnos ayudantes capacitados para impartir la capacitación y acreditar los cursos. Esta plantilla será otro resultado imprescindible de esta fase. Incluye:

Nombre del proyecto

Herramientas libres a implantar

Cursos y Profesores

Se agrupa la distribución para cada proyecto, de las herramientas a utilizar, el nombre el(los) curso (s) que explicarán el trabajo en esa herramienta propuesta y el profesor que lo impartirá. El Programa analítico de los cursos también, porque permite determinar el alcance de estos, así como otros datos de importancia.

Se debe realizar además, la **plantilla** que se nombra **Documentación** (Ver Anexo 3), con toda la bibliografía que necesita el personal para su auto superación, catalogada por cursos y la dirección en la que se puede encontrar. Además este documento debe ser entregado al personal de los proyectos antes de dar inicio a los cursos. Por lo cual se puede decir que las tareas de formación y documentación se deben realizar paralelamente, se determina entonces como un artefacto de ésta tarea.

De surgir algún tipo de falla será registrada en la **plantilla** titulada **Reportes de fallas** (Ver Anexo 3), puesto que ésta se encargará de recoger todo tipo de errores que puedan surgir y la solución que se le dé a los mismos, desde este flujo hasta que concluya el proceso de migración. Permitiendo que una vez solucionados los problemas vayan quedando documentados, por si volviese a ocurrir.

3.2.5 Flujo de trabajo: Implantación

La implementación es el flujo de trabajo donde se hará efectiva la migración, ésta marcará el fin del software privativo, de forma paulatina se irán migrando las herramientas e instaurando las aplicaciones y sistemas libres, es una etapa donde existirán grandes cambios en los sistemas por lo que la formación y soporte deberá reforzarse.

Será el momento en el que la mayoría de los usuarios podrán tocar de cerca la migración y hacerse partícipes de ella, por lo que todas las medidas que se tomen para el correcto desenvolvimiento de la misma serán pocas, deberán reforzarse las charlas de sensibilización y generar entusiasmo hacia los usuarios con el objetivo de lograr una buena aceptación del proceso, los mayores esfuerzos dentro de la implementación se concentrarán en la fase de migración total e irá disminuyendo a medida que se avance en la migración quedando al final ésta tarea en la fase de consolidación.

Actividades y tareas: A continuación se definen las actividades para éste flujo, que las mismas son además las tareas a realizar, sólo con la excepción de la tercera que se define con exactitud. Estas acciones la realizarán los siguientes roles del grupo de migración: jefes de los proyectos inmersos en el proceso, los expertos en SWL y los técnicos.

Instalar algunas herramientas sobre Windows

Instalar las herramientas para los planes de formación y certificación

Remplazar los clientes de Windows por GNU/Linux y las herramientas de desarrollo

Escoger de cada proyecto un grupo de estaciones del proyecto para la instalación y orientarles a los usuarios de esas máquinas que hagan una salva de todo el trabajo que tengan de importancia.

Escoger el sistema operativo libre a instalar, todas las herramientas necesarias para el trabajo del proyecto en cuestión y proceder a la instalación de imágenes en un grupo escogido de estaciones del proyecto.

El grupo de migración debe validar y afinar la configuración del puesto de trabajo.

El grupo de migración debe prestar soporte activo al primer grupo mientras se acostumbran a trabajar con el sistema instalado.

El grupo de migración debe generar entusiasmo por el proyecto, recoger comentarios y sugerencias sobre la interfaz de usuario.

El grupo de migración debe hacer extensivo el segundo y tercer paso para todos los puestos de trabajo.

Prestar soporte activo a todo el proyecto mientras se acostumbra a trabajar con el sistema instalado en lo cual pueden ser de gran ayuda las experiencias de los problemas que se pudieron presentar en la migración del primer grupo de cada proyecto.

Artefactos de entrada: Los artefactos de salida y las actividades a realizar, deben estar en correspondencia con plantilla de Configuración propuesta, puesto que una vez realizado el pilotaje se tengan experiencias sobre la implantación del sistema libre y ésta configuración haya quedado refinada. Permitirá entonces, realizar las tareas de instalación y configuración con el mismo modelo obtenido de las pruebas pilotos pero replicable a mayor escala. También es importante tener en cuenta la de Reporte de fallas, puesto que recogidos ya los problemas y soluciones a éstos, ocurridos en el flujo de Pilotos, permitirá no tener que volver a caer en ellos o darles una solución más rápida.

Artefactos de salida: Al igual que en el flujo anterior, es necesario hacer uso de la **plantilla Reporte de fallas** (Ver Anexo 3) pero definiendo la fase en la que se encuentra la migración. De esta manera se podrá diferenciar los errores de una a la otra, permitiendo entonces sacar conclusiones sobre la trayectoria recorrida hasta el momento y tener documentados los posibles errores de cada fase.

3.2.6 Flujo de trabajo: Supervisión y soporte informático

El último flujo propuesto en la migración es al igual que la formación uno de los más extensos del proceso, la asistencia y soporte técnico estará presente a lo largo de todo el tiempo que dure el cambio, comenzando con la instalación de la primera computadora e irá disminuyendo a medida que avanza el proceso es importante destacar que el soporte nunca deberá terminar ya que siempre habrán problemas para darles soluciones sólo que estos tienden a decrementos , su objetivo principal es brindar el soporte para el mantenimiento de todas las infraestructuras y al personal.

La asistencia y soporte técnico debe estar presente el tiempo que sea necesario de acuerdo a las necesidades de los proyectos, por lo que el número de iteraciones en la fase de consolidación, donde mayor impacto posee dicho flujo, no está definido a ciencia cierta.

3.2.6.1 Características de la asistencia técnica y el soporte informático

Es muy importante que al brindar asistencia técnica y soporte informático en los proyectos donde se haya realizado la migración se cumplan con determinadas pautas, que propiciarán un mejor desarrollo de la producción, luego de haber transitado por las fases anteriores.

- Atención completa a incidencias, dudas, consultas, investigaciones y recomendaciones que haga el personal al cual se le está prestando el servicio.
- Trato personal al usuario con su interlocutor directo, buscando una solución según las necesidades de individuales.
- Seguimiento a la incidencia, análisis e investigación con el objetivo de intentar solucionar la incidencia con la mayor brevedad posible.
- Atención inmediata, se traduce en que la recepción de la petición se realiza de forma inmediata dentro del horario establecido en el cual debería estar el personal del proyecto.
- Estado de soporte, el usuario debe tener acceso a toda la información útil de soporte de su proyecto.
- Comunidades, los usuarios deben estar informados sobre los sitios que tiene la universidad donde puede encontrar algunas respuestas a sus problemas y vincularlos a la comunidad del software libre de la universidad, para que tengan un espacio de comunicación (chat o correo) donde poder hacer preguntas, evacuar dudas y compartir conocimientos.

3.2.6.2 Modalidades de atención al personal

Para realizar ésta tarea se dispone de varias modalidades, las cuales pueden satisfacer las necesidades que se presenten.

- Asistencia presencial: Asistencia "in situ" para todo tipo de sistemas y entornos, modalidad donde se debe garantizar una rápida respuesta.
- Asistencia telefónica: Para resolver cualquier duda que pueda surgir o para guiarle en la instalación de un programa, configuración de drivers, mantenimiento de sus equipos, etc. El proyecto donde surgió el problema será el encargado de reportar la falla y el técnico acudirá, se debe garantizar que siempre sea el mismo técnico asignado, garantizando la familiarización con su sistema.
- Asistencia remota: Su problema no podría ser resuelto de una manera más eficiente. Los técnicos se conectarán al ordenador con fallas mediante mecanismos de alta seguridad (comunicaciones con cifrados de 1024 bits y clave asimétrica) para tratar de solucionar cualquier incidencia en cuestión de minutos.

Es necesario que ante cualquier falla se documenten los reportes y las soluciones que se le dan, así se podrán disminuir los esfuerzos en ésta tarea. Esta documentación de fallas está a cargo del técnico, pero siempre tiene que tener conocimiento un encargado por el proyecto. Este servicio además debe ser de lunes a sábado de las 8 am a las 2am, para que no se atrase el trabajo del proyecto porque haya que reportarlo un día en específico, será una vinculación entre el técnico y el proyecto.

Actividades y tareas: Se definen las actividades para este flujo y las tareas a realizar en cada una de ellas. Estas acciones la realizarán los siguientes roles del grupo de migración: jefes de los proyectos inmersos en el proceso, los expertos en SWL y los técnicos.

Evaluar los mecanismos de soporte

Verificar la existencia de elementos de soporte

Listar los mecanismos que existan (en caso de que existan)

Divulgar los métodos de atención al usuario que existen en la UCI

Divulgar listas y foros que existen en la universidad para facilitar el debate de temas relacionados.

Crear canales para la atención a los usuarios vía telefónica ante dudas.

Mantenimiento de sitios y cursos virtuales para el auto-aprendizaje.

Divulgación sobre el trabajo de la comunidad virtual de la UCI y como formar parte de ella.

Planificar mecanismos de soporte y asistencia

Realización de estudios de las aplicaciones más usadas

Estudiar las necesidades de los usuarios

Lograr mejoras en el servicio de aplicaciones y optimización de las mismas, para aumentar su rendimiento, permitiendo saber en cada momento cuáles son las necesidades de los clientes.

Creación de manuales de las herramientas implantadas.

Lograr la automatización de la mayor cantidad de tareas de soporte

Implantar mecanismos para la actualización automática de los repositorios,

Instalar herramientas que permitan la instalación automática de herramientas en las máquinas de los clientes.

Fortalecer los mecanismos de soporte y mantenimiento

Habilitar algún espacio físico a donde los usuarios puedan dirigirse para recibir asesoría.

Establecer políticas que promuevan el uso de estándares para la creación de aplicaciones (ejemplo librerías gráficas o lenguajes de programación)

Creación de repositorios de aplicaciones y datos. Y servicios asociados: personalización de repositorios, secciones propias, etc.

Brindar asistencia técnica y soporte informático a los proyectos

Utilizar el grupo de soporte propio del proyecto

Implementar las modalidades de atención al usuario

Brindar los servicios tratando de cumplir las características definidas para la asistencia técnica y el soporte informático

Artefactos de entrada: Se necesitará tener en cuenta los reportes obtenidos de otras fases, recogidos en la plantilla de Reportes de fallas para estudiar los problemas y las soluciones que se le fueron dando para mitigarlos. De esta manera los encargados del soporte estarán mejor preparados y poder ganar tiempo, al eliminar fallas que ya se presentaron con anterioridad.

Artefactos de salida: La **plantilla Reportes de fallas** (Ver Anexo 3), por las características de la tarea de ella. Se definirá la fase de la migración, en el documento y ello propiciará que para los problemas recogidos en su tiempo de duración los responsables sean capaces de apoyarse en la solución de las fallas de otra fase en caso, que sea la misma o con características similares. Siempre a consideración de los trabajadores que vayan a darle solución al problema.

3.3 Aplicación del método de expertos

El método de expertos permite tomar decisiones para aceptar o rechazar determinada propuesta de acuerdo a los criterios definidos y a la evaluación dada por los expertos seleccionados (18).

Paso 1: Los criterios que se han tenido en cuenta para la valoración de los expertos han sido desglosados en cuatro grupos (Ver Anexo 5)

Paso 2: A partir de la importancia conferida a cada grupo de criterios en función de los aspectos a evaluar, al asignar el peso a cada grupo la suma total de pesos debe ser 100.(Ver anexo 5)

Paso 3: El comité de expertos está compuesto por un total de 7 personas.

Paso 4: Entregar a cada experto la propuesta para que estudien el tema a evaluar y 2 modelos: en el primero deben valorar el peso relativo de cada criterio (Ver Anexos 5 y 6) y otro para realizar una evaluación cuantitativa de los criterios en una escala del 1 al 5, luego deben registrar su apreciación cualitativa de la propuesta con una clasificación final que puede ser: excelente, bueno, aceptable, cuestionable y malo; además pueden registrar sus consideraciones respecto a la propuesta.

Paso 5: Determinar por cada criterio el peso promedio, a partir de los pesos dados por los expertos (Ver Anexo 8).

Paso 6: Determinación de la consistencia en el trabajo de los expertos:

Dado C el número total de criterios a evaluarse, y E el número de expertos involucrados en la evaluación, se realiza el siguiente procedimiento para determinar la consistencia trabajo de los expertos:

Calcular para cada criterio: $\sum E$, que representa la sumatoria del peso dado por los expertos

Determinar el valor de E_p : puntuación promedio de cada criterio

Se calcula el peso medio de cada criterio $M\sum E$

Hallar el valor de ΔC , que representa la diferencia entre $\sum E$ y $M\sum E$

Determinar la desviación de la media, que posteriormente se eleva al cuadrado obtener la dispersión S, dada por la expresión: $S = \sum (\sum E - \sum \sum E / C)^2$

Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall dado por la expresión: $W = S / E^2 (C^3 - C) / 11$

Calcular el Chi cuadrado real a partir del valor del coeficiente de Kendall teniendo en cuenta la siguiente expresión: $X^2 = E (C-1) W$ (Ver Anexo 9)

Posteriormente, se compara el X^2 real, con el valor del dato estadístico, siendo $\alpha=0.01$, y $C = 12$ y debe cumplirse que $X^2_{real} < X^2(\alpha, c-1)$ para que el trabajo realizado por los expertos sea valorado de consistente.

$$X^2(\alpha, c-1) = X^2(0,025;11) = 28,7291$$

Comparando el real y el obtenido de la tabla sería $27,7 < 28,7291$; quedando demostrada la consistencia en el trabajo realizado por los expertos.

Paso 7: Cálculo de $P \times c$:

P: peso promedio de cada criterio.

C: calificación promedio de cada criterio concebida por los expertos. (Ver Anexo 10)

Paso 8: Índice de aceptación de la propuesta:

Se calcula a partir de la siguiente fórmula teniendo que $(P * c) = 4,54$, $IA = (P * c) / 5$

Al despejar los valores, obtener el índice de aceptación: $IA = 0,908$

Paso 9: Probabilidad de éxito de la propuesta:

Se determina a partir de los rangos predefinidos del índice de aceptación:

$IA > 0,7$ Existe alta probabilidad de éxito

$0,7 > IA > 0,5$ Existe probabilidad media de éxito

$0,5 > IA > 0,3$ Probabilidad de éxito baja

$0,3 > IA$ Fracaso seguro

Dado el resultado de IA igual a 0,908 entonces se puede concluir que la probabilidad de éxito es: ALTA.

CONCLUSIONES

Mediante el presente trabajo se logró dar cumplimiento a su principal objetivo, el cual consistía en crear una estrategia para la migración a software libre de los proyectos de software educativo. Se realizó un estudio del origen de los sistemas libres, de las diferencias entre tecnologías privativas y de código abierto “open source” utilizadas para el desarrollo de SWE, concluyendo que el uso del SWL es una vía al desarrollo y soberanía tecnológica. Se estudiaron las diferentes guías de migración escritas por comités técnicos en diferentes países e instituciones así como las de nuestro país. Lo cual permitió describir la estrategia para la migración y fundamentar la utilidad de las herramientas propuestas. Dándosele de esta forma respuesta a las preguntas científicas que se plantearon al inicio de la investigación.

La estrategia propuesta permite guiar la adopción de tecnologías libres en los proyectos de SWE, que permitan desligarse de las herramientas privativas y disminuir los altos costos en la producción de SWE. Una vez que se haya implantado la estrategia, se logrará minimizar la brecha informacional, con respecto a la cultura del SWL y el rechazo que hace el personal de los proyectos cuando se cambian los paradigmas de desarrollo a sistemas libres. La migración incluye la capacitación en las nuevas herramientas de desarrollo y por lo tanto cumplir con los requerimientos de los productos. La estrategia propuesta fue validada mediante el método de experto, buscando la máxima fiabilidad en las fases y flujos de trabajo que define y se obtuvo un índice de aceptación alto; lo que se traduce en resultados eficientes en términos de validación.

RECOMENDACIONES

De acuerdo a los resultados obtenidos en dicha investigación se recomienda:

- Describir detalladamente las responsabilidades de los roles definidos en el grupo de migración.
- Llevar a cabo el piloto con un proyecto del Polo de SWE que tenga pocos compromisos de entregas, tiempo y dinero.
- Definir una segunda versión de la estrategia, de manera que se integren el resto de los procesos que intervienen en el desarrollo de software educativo, Diseño, Gestión y Producción de medias y Pedagogía.

BIBLIOGRAFÍA

1. *La Gestión del conocimiento: Contexto, concepto y aplicaciones*. **SECURED**. 3, España : Sociedad Española de documentación e información científica, 2000-2007, Vol. I.
2. **Stallman, Richard M.** *Software libre para una sociedad libre*. s.l. : Free Software Foundation, 1995-1999.
3. *Aspectos legales y de explotación del software libre*. **Malcolm, Bain**. 2, 2000, Vol. I.
4. *Sistemas Operativos: Conceptos Fundamentales*. **Botaro, Estero**. Cadiz : Universidad de Cadiz. Servicio de publicaciones, 2002, Vol. I.
5. **Vidal, Miguel**. Richard Stallman y El proyecto GNU. *forge.morfeo-project.org*. [Online] marzo 30, 2008. [Cited: abril 1, 2009.] http://forge.morfeo-project.org/docman/view.php/83/312/3.3_proyecto-gnu.pdf.
6. *Conceptos de Sistema Operativo*. **Cortes, Jose Benitez**. Miraflores : Colegio Miraflores, 1999.
7. **Raymond, Eric S.** *La Catedral y el Bazar*. 1997.
8. *Revista SL*. **Gonzalez, Gonzalo Javier**. s.l. : SL Tras Pantomimas, 2009, Vol. II.
9. *Software libre vs Software propietario*. **Montserrat, Culebro Juarez**. Mexico : s.n., 2006, Vol. I.
10. *Seguridad informática y software libre*. **Fernandez, Jorge Ferrer Sanguino y Javier**. version 1.0, s.l. : Estructura de Hispalinux, 2006-2009, Vol. I.
11. *Software libre: Conceptos y licencias*. **Labrador, Josuka Diaz**. Deusto : Universidad de Deusto: Facultad de Ingeniería (EDIDE)+ DELI, 2004, Vol. I.
12. **Almeida, Yudivian Cruz**. De comunidades: Comunidad Cuba de software libre. [Online] febrero 1, 2008. [Cited: marzo 20, 2009.] <http://www.comunidades.cu>.
13. **Osmely Fernández Llitas, Ronny Rodríguez Lloró, José Seidel L. García, Aley Valdés González, Eliezer**. *Manual de referencias Sora script*. Cuba : Centro de Estudios de Software Educativo CEnSoE, 2008. registro:2821.
14. **Netproject, Ltd. Grupo**. *Directrices IDA de migración a software de fuentes abiertas*. Europa : s.n., 2003.
15. **Samón, Ramon Paumier**. *Metodología para la migración a software libre de las universidades de las Ciencias Informáticas (UCI)*. Ciudad de la Habana : s.n., 2007.
16. **Villazon, Yoandy Perez**. *Metodología para la migración a software libre de las universidades del Ministerio de Educación Superior (MES)*. Ciudad de la Habana : s.n., 2008.
17. **Unicornios, Grupo**. Herramientas Floss para aplicaciones de Windows. *tabla de equivalencia por asignatura Facultad 10*. Ciudad de la Habana : s.n., 2008. Vol. I.
18. **Campos, Rafael Paré**. Base de Datos en entornos libres. *Bases de datos*. 2006-2008. Vol. I.
19. **Andes U. d.** *Plan Institucional de liberación de software y de Migración a Plataformas de Software libre*. Venezuela: s.n, 2006.
20. **León, R.A.H.** *El paradigma cuantitativo de la investigación científica*. 2002.

TRABAJOS CITADOS

1. *La Gestión del conocimiento: Contexto, concepto y aplicaciones*. **SECURED**. 3, España : Sociedad Española de documentación e información científica, 2000-2007, Vol. I.
2. **Stallman, Richard M.** *Software libre para una sociedad libre*. s.l. : Free Software Foundation, 1995-1999.
3. *Aspectos legales y de explotación del software libre*. **Malcolm, Bain**. 2, 2000, Vol. I.
4. *Sistemas Operativos: Conceptos Fundamentales*. **Botaro, Estero**. Cadiz : Universidad de Cadiz. Servicio de publicaciones, 2002, Vol. I.
5. **Vidal, Miguel**. Richard Stallman y El proyecto GNU. *forge.morfeo-project.org*. [Online] marzo 30, 2008. [Cited: abril 1, 2009.] http://forge.morfeo-project.org/docman/view.php/83/312/3.3_proyecto-gnu.pdf.
6. *Conceptos de Sistema Operativo*. **Cortes, Jose Benitez**. Miraflores : Colegio Miraflores, 1999.
7. **Raymond, Eric S.** *La Catedral y el Bazar*. 1997.
8. *Revista SL*. **Gonzalez, Gonzalo Javier**. s.l. : SL Tras Pantomimas, 2009, Vol. II.
9. *Software libre vs Software propietario*. **Montserrat, Culebro Juarez**. Mexico : s.n., 2006, Vol. I.
10. *Seguridad informática y software libre*. **Fernandez, Jorge Ferrer Sanguino y Javier**. version 1.0, s.l. : Estructura de Hispalinux, 2006-2009, Vol. I.
11. *Software libre: Conceptos y licencias*. **Labrador, Josuka Diaz**. Deusto : Universidad de Deusto: Facultad de Ingeniería (EDIDE)+ DELI, 2004, Vol. I.
12. **Almeida, Yudivian Cruz**. De comunidades: Comunidad Cuba de software libre. [Online] febrero 1, 2008. [Cited: marzo 20, 2009.] <http://www.comunidades.cu>.
13. **Osmely Fernández Lliteras, Ronny Rodríguez Lloró, José Seidel L. García, Aleyn Valdés González, Eliezer**. *Manual de referencias Sora script*. Cuba : Centro de Estudios de Software Educativo CEnSoE, 2008. registro:2821.
14. **Netproject, Ltd. Grupo**. *Directrices IDA de migración a software de fuentes abiertas*. Europa : s.n., 2003.
15. **Samón, Ramon Paumier**. *Metodología para la migración a software libre de las universidades de las Ciencias Informáticas (UCI)*. Ciudad de la Habana: s.n., 2007.
16. **Villazon, Yoandy Perez**. *Metodología para la migración a software libre de las universidades del Ministerio de Educación Superior (MES)*. Ciudad de la Habana: s.n., 2008.
17. **Andes U. d.** *Plan Institucional de liberación de software y de Migración a Plataformas de Software libre*. Venezuela: s.n, 2006.
18. **León, R.A.H.** *El paradigma cuantitativo de la investigación científica*. 2002.

SITIOS DE INFORMACIÓN

- <http://blender.org/cms/Home.2.0.html>
- <http://es.openoffice.org/>
- <http://audacity.sourceforge.net/>
- <http://heroinewarrior.com/cinelerra.php3>
- <http://www.gimp.org>
- <http://www.inkscape.org>
- [http:// www.scribus.org.uk](http://www.scribus.org.uk)
- <http://linux.bankhacker.com/software/KIllustrator+Kontour/>
- <http://ktoon.toonka.com/>
- <http://www.synfig.com/download/>

GLOSARIO

Código abierto (Open Source): Código abierto (del inglés open source) es el término por el que se conoce al software distribuido y desarrollado en forma libre. Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software). Fecha de consulta: 7/1/2009

Código cerrado: Es el código fuente que no se encuentra disponible para cualquier usuario, es decir no se hace público. Se le llama así en contraposición al código abierto. El software no libre utiliza un código cerrado. Por su calidad de secreto industrial, su divulgación (que puede ocurrir ocasionalmente) constituye un delito. Fecha de consulta: 26/3/2009

Código fuente: Es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento. El código fuente de un programa está escrito por un programador en algún lenguaje de programación, pero en este primer estado no es directamente ejecutable por la computadora, sino que debe ser traducido a otro lenguaje (el lenguaje máquina o código objeto) que sí pueda ser ejecutado por el hardware de la computadora. Fecha de consulta: 26/3/2009

Estrategia: Es "un conjunto de objetivos y políticas para lograr objetivos amplios", brindar a las organizaciones una guía para lograr un máximo de efectividad en la administración de todos los recursos en el cumplimiento de la misión. Fecha de consulta: 10/2/2009

Gestión del Conocimiento: La Gestión del Conocimiento corresponde al conjunto de procesos y sistemas que permiten que el capital intelectual (es el conocimiento intelectual de esa organización, la información intangible (que no es visible, y por tanto, no está recogida en ninguna parte) que posee y que puede producir valor.) de una organización aumente de forma significativa, mediante a la gestión de sus capacidades de resolución de problemas de forma eficiente, de manera de generar ventajas competitivas a través del tiempo. Fecha de consulta: 21/1/2009

Herramientas libres: Las herramientas libres son aquellas que pueden ser usadas, copiadas, estudiadas, modificadas y redistribuidas libremente. No está sujeta

a limitaciones de mejora puesto que su licencia lo permite explícitamente; está disponible en forma de código fuente abierto y, por lo tanto, todo el mundo puede accederlas y emplearlas como quiera. Fecha de consulta: 25/1/2009

IBM: International Business Machines (NYSE: IBM) (conocida coloquialmente como **el Gigante Azul**) es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911, pero lleva operando desde 1888. Fecha de consulta: 2/3/2009

Kernel: Es la parte fundamental de un sistema operativo, el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso para el programador. Fecha de consulta: 2/3/2009

Licencia General Pública de GNU: Las licencias que cubren la mayor parte del Software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas. Fecha de consulta: 28/3/2009

Linux: Es un sistema operativo, compatible Unix. Dos características muy peculiares lo diferencian del resto de los sistemas que se puede encontrar en el mercado, la primera, es que es libre, esto significa que no se tiene que pagar ningún

tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente. El sistema lo forman el núcleo del sistema (kernel) más un gran número de programas / librerías que hacen posible su utilización. Fecha de consulta: 28/12/2008

Migración a software libre: Proceso que aboga por la sustitución del concepto de la comercialización de software como negocio para convertirlo en la de servicio o valor añadido, que es lo que realmente atrae al usuario. El software debe ser de código abierto, de modo que cualquier usuario pueda modificarlo y adaptarlo a sus necesidades. El desarrollo de software, pues, se convierte en un gran esfuerzo colectivo donde cada miembro de la comunidad obtiene beneficio del trabajo de los otros miembros; y donde el negocio surge de explotar este software a través de servicios de mantenimiento, instalación, soporte técnico, etc. El paradigma de este movimiento es el sistema operativo Linux, creado a partir de una idea original de Linus Torvald, que posteriormente ha ido perfeccionando un gran grupo de desarrolladores repartido por todo el mundo gracias a las facilidades de colaboración y trabajo en grupo que ofrece la red Internet. Fecha de consulta: 25/1/2009

Multimedia educativa: Documentos informáticos que combinan textos, imágenes y sonidos, es decir, a recursos que nos ofrecen la posibilidad de obtener información de diversos medios de comunicación interconectados y que pueden ser controlados por el alumno mediante el ordenador. El entorno multimedia permite al usuario interactuar activamente con la información y reestructurarla de forma personal. Fecha de consulta: 7/1/2009

Open Source Initiative: La Open Source Initiative es una organización dedicada a la promoción del software abierto. Fue fundada en febrero de 1998 por Bruce Perens y Eric S. Raymond. Fecha de consulta: 25/1/2009

Protocolo TCP/IP: Conjunto básico de protocolos de comunicación de Internet, que permiten la transmisión de información en redes de computadoras. El nombre TCP / IP proviene de dos protocolos importantes de la familia, el *Transmission Control Protocol (TCP)* y el *Internet Protocol (IP)*. Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto, algunos de ellos son HTTP, que es el que está usando usted ahora, FTP para transferencia de archivos, SMTP y POP para correo, entre otros. El TCP / IP es la base de Internet que sirve para enlazar

computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa.

Fecha de consulta: 2/3/2009

Software educativo: El Software educacional o educativo es software cuya finalidad primordial es, dependiendo el punto de vista: la enseñanza, o el aprendizaje. Abarca desde programas para el nivel preescolar, que tienen un gran componente lúdico, a programas de mecanografía y aplicaciones para aprender lenguas extranjeras. Fecha de consulta: 7/1/2009

Software libre: La filosofía de código abierto, propia del movimiento anárquico y auto-sostenido de la red Internet, aboga por la sustitución del concepto de la comercialización de software como negocio para convertirlo en la de servicio o valor añadido, que es lo que realmente atrae al usuario. El software debe ser de código abierto, de modo que cualquier usuario pueda modificarlo y adaptarlo a sus necesidades. Fecha de consulta: 7/1/2009

Socket: Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada. Un *socket* queda definido por una dirección IP, un protocolo y un número de puerto. Fecha de consulta: 21/3/2009

Software propietario: La expresión software propietario proviene del término en inglés "*proprietary software*". En la lengua anglosajona, "*proprietary*" significa «poseído o controlado privadamente» («*privately owned and controlled*»), que destaca la mantención de la reserva de derechos sobre el uso, modificación o redistribución del software. Fecha de consulta: 7/1/2009

TRIX: Es un núcleo de llamadas remotas a procedimientos, desarrollado por el MIT y cuyos autores decidieron que fuera libremente distribuido, era totalmente compatible con UNIX versión 7. Fecha de consulta: 5/2/2009

Usenet: Es el acrónimo de **Users Network** (Red de usuarios), consistente en un sistema global de discusión en Internet, que evoluciona de las redes UUCP. Fue creado por Tom Truscott y Jim Ellis, estudiantes de la Universidad de Duke, en 1979. Los usuarios pueden leer o enviar mensajes (denominados artículos) a distintos

grupos de noticias ordenados de forma jerárquica. El medio se sostiene gracias a un gran número de servidores distribuidos y actualizados mundialmente, que guardan y transmiten los mensajes. Fecha de consulta: 26/3/2009

Wizards (asistente): Es un elemento de interfaz de usuario donde el usuario se presenta con una secuencia de cuadros de diálogo. A través de estos cuadros de diálogo, el usuario es llevado a través de una serie de medidas, la realización de tareas en una secuencia específica. A veces puede ser más fácil llevar a cabo las tareas mediante un asistente, especialmente para los complejos o con poca frecuencia en tareas que el usuario no está familiarizado con los pasos necesarios. Fecha de consulta: 26/3/2009

ANEXOS

Anexo 1-Tabla1: Herramientas para la producción en la facultad 8

Categorías	Herramientas Privativas	Herramientas libres
Ofimática:	Paquete de office 2007	
	Microsoft Word	
	Microsoft Excel	
	Microsoft Power Point	
Tratamiento de imagenes	Adobe Photoshop	
	Paint	
	Corel Draw	
Edición de sonido, audio y video	Adobe Premiere	
Correo electrónico	Outlook	Mozilla Firefox
	Internet Explorer	
	Opera	Opera
Calendarios y herramientas colaborativas	Outlook	eGroupware
Gestión de Documentos	End Note	SVN
	TortoiseSVN	
Bases de datos	Personal:Microsoft Access	Postgre SQL
	Plataformas: SQL server, Oracle	MySQL manager
	Diseño:Erwin	
	Administración: SQL server	
Ingeniería de software	Rational Rose	Visual Paradigm
Gestión de Software	MS Proyect	
Paquete de multimedia	Dreamweaver 8.0	Aptana Studio
	CS3	
	Flash 8.0	
	Toolbook	
	Director	
	Firework	
Programación (HTML, PHP, Java, Action script 2.0,C #)	Editores HTML	
	Zend Studio	
	Deamweaver 8.0	
	FrontPage	
	Visual Studio	
	Jbuilder	
	Flash 8.0	
Servidor Web	Internet Information Service	Apache

Anexo 2-Tabla 2: Herramientas análogas (libres)

Descripción del programa, tareas ejecutadas	Windows	Linux	
<i>1) Redes y conectividad</i>			
Navegador Web	Internet Explorer	Mozilla Firefox	
	Mozilla Firefox	Galeon	
	Opera	Konqueror	
	Phoenix for Windows	Opera Phoenix	
Clientes de Email	Outlook Express	Evolution.	
	Mozilla Thunderbird	Mozilla ThunderBird	Sylpheed, Sylpheed-claws.
		Kmail.	Gnus.
		Balsa.	Bynari Insight GroupWare Suite
		Arrow.	Gnumail.
		Althea.	Liamail.
		Aethera.	
Clientes Jabber IM	JAJC	Tkabber.	
	TKbber	Gabber.	
	Psi	Psi.	
	Exodus	Pidgin	
	WinJab	Gajim	
	myJabber	Gossip	
	RhymBox	Kopete	
	Rival	SIM	
	Skabber		
	Tipic IM		
	Vista		
	Yabber		
Administración Remota	VNC	VNC.	
	Administrador Remoto	ssh / openssh/ nx	
	Asistencia Remota	Remote management is built-in in XFree86.	
	Symantec pcAnywhere	Rdesktop Client.	
	Windows Terminal Server	freeS/WAN	
	Rdesktop		
	Radmin		
<i>2) Sistemas de Software para Escritorio.</i>			

Editor de Texto	Notepad	Kedit (KDE).
	WordPad	Gedit (Gnome).
	TextPad	Gnotepad.
Visualizador de PDF	Adobe Acrobat Reader	Acrobat Reader para Linux. Xpdf, Okular,
Trabajando con compresores de archivos	WinZip	FileRoller.
	WinRar	Gnozip, Ark
Herramientas Ofimatica	Microsoft Office cualquier versión	Open Office.org 3.0
Antivirus	AVG AntiVirus, NAV, Dr. Web, TrendMicro, F-Prot, Kaspersky	Dr. Web for Linux.
		Trend ServerProtect for Linux.
		RAV Antivirus.
		OpenAntivirus + AMaViS / VirusHammer.
		F-Prot.
		Sophie / Trophie.
		Kaspersky for Linux.
3) Multimedia (audio / CD).		
Reproductores Música / mp3 / ogg s	Winamp	XMMS (X multimedia system).
		Noatun.
		Zinf. (former Freeamp)
		Amarok
		Exaile
		GQmpeg.
		SnackAmp
Programas para quemar CD con GUI	Nero, Roxio Easy CD Creator,	K3b. (KDE)
		XCDRoast.
		KOnCd.
		Eclipt Roaster.
		Brasero
		CD Bake Oven.
		KreateCD.
		SimpleCDR-X.
GCombust.		
Reproductores de CD	CD player	KsCD.
		Orpheus. (console)
		Sadp.
		WorkMan.
		Xmcd.
		Grip.
		XPlayCD.
		ccd / cccd. (consola)
CD ripping / grabación	Windows Media Player, AudioGrabber, Nero,	Grip.
	VirtualDrive, VirtualCD, ...	Audacity.

		RipperX.
		tkcOggRipper.
		cdda2wav.
Trabajando con protocolos en Tiempo Real	RealPlayer	RealPlayer for Linux. Mplayer + librerías.
Editores de Audio	SoundForge, Cooledit	Sweep.
		WaveForge.
		Sox.
		Audacity.
		GNUSound.
		Ecasound
<i>4) Multimedia (Gráficos).</i>		
Visualizador de Archivos Gráficos	ACDSee, IrfanView	Xnview.
		GQview.
		Qiv.
		CompuPic.
		Kuickshow.
		GTKSee.
		xv.
		pornview.
		imgv.
		Gwenview.
		Gliv.
		Showimg.
		Fbi.
Gthumb		
Editores Simples de Gráficos	Paint	Kpaint.
		Tuxpaint.
		Xpaint.
		Gpaint.
		Killustrator.
Editores potentes de Gráficos tipo PhotoShop	Adobe Photoshop	Gimp, FilmGimp.
	Gimp para Windows	ImageMagick
	Paint Shop Pro	
Programas para Trabajos gráfico con Vectores	Adobe Illustrator, Corel Draw 14, Freehand, AutoSketch	Inkscape
		Sodipodi.
		xfig.
		Sketch.
		Karbon and Kontour.
		OpenOffice Draw.
		Dia.
Tgif.		
Visualizadores de Flash	Flash Player	GNASH
		Flash Player para Linux

		swfdec
Creación de Flash	Macromedia Flash	DrawSWF.
		Ming. (Creación de flash swf desde lenguajes de programación)
Creación de gráficos Web	Macromedia Fireworks	Gimp.
Gráficos 3D	3D Studio MAX	Blender.
	Maya	Maya.
	Povray	KPovModeler.
		K3Studio.
		Moonlight.
		GIG3DGO.
		Povray.
		K3D.
		Wings 3D.
Softimage XSI.		
Editores de Iconos	Microangelo	Gnome-iconedit.
		Kiconedit.
Gráficos 2D		Ktoon
		Synfig
<i>5) Programación y Desarrollo.</i>		
Plataforma .NET	Microsoft VisualStudio .net	MonoDevelop
Visual C++ IDE	Borland C++ Builder	Anjuta + Glade + Devhelp.
	MS Visual C	Codeblocks
		Kdevelop
		QtCreator
		Arriba. [Prop]
		Gbuilder.
		Source Navigator.
	Vide.	
Objetos en Pascal IDE	Delphi	Lazarus + FPC.
Pascal	Pascal, BP	Freepascal.
		GNU Pascal.
		RShell (al estilo de Borland Pascal 7.0)
Basic	Basic	Hbasic.
		X-basic.
		Yabasic.
		SmallBASIC
Prolog	VisualProlog,	GNU Prolog.
Editor Html WYSIWYG	Deamweaver	NVU / Kompozer

		Quanta Plus
		Bluefish
		Screem
Editores HTML	HomeSite	Quanta Plus.
		2) Bluefish.
		3) WebMaker.
		4) Screem.
		5) Toppage.
		6) WebDesigner.
		7) ScriptEditor.
		8) August.
IDE de Java	Jbuilder, NetBeans, Eclipse	NetBeans.
		Eclipse.
		Sun ONE Studio. [formerly Forte]
		Vide
IDE para Desarrollo de Bases de Datos Oracle	T.O.A.D., SQL Navigator, PL/SQL Developer	Tora.
CASE-facilesity para UML	ArgoUML, Together ControlCenter [Prop]	Umbrello UML Modeller.
		Dia+Dia2Code.
		PocidonCE (community edition).
		ArgoUML.
		Together ControlCenter
Elevados sistemas de CASE	Rational Rose.	Rational Rose for Linux.
Editores HEX	Hiew	Biew.
		KHexEdit.
Compiladores y preprocesadores Clipper	CA-Clipper	Clip.
	The Harbour Project	The Harbour Project.
		xHarbour.
Plataformas al estilo dot-net	M\$.Net	Mono.
		DotGNU/Portable.NET
Trabajando con CVS	WinCVS	cvs (consola).
	TortoiseCVS	Cervisia (KDE).
	cvs for Windows	Lincvs. (Front-end para CVS)
		SubVersion.
Visual Basic	Visual Basic	Phoenix.
Librerías Gráficas	-	gtk.
		qt.
		wxwidgets

<i>6) Software para Servidores.</i>		
Servidores Web	Internet Information Server, Apache para Windows, roxen / roxen2, wn, cern-httpd, dhttpd, caudium, aolserver, boa, ...	Apache.
		Xitami.
		Thhttp.
		Tux.
		PublicFile.
Servidores FTP	Internet Information Server	pure-ftp.
	ServU	vsftpd.
	War FTP	wu-ftp.
	BulletProof FTP server	proftpd.
		gl-ftp.
		ftp.
		PublicFile.
Languages para Desarrollo Web	PHP para Windows	PHP.

Anexo 3- Tabla 3: Responsables por artefactos

Artefactos o Plantillas	Responsables
Tecnologías utilizadas	Jefe de proyecto y Jefe del grupo de migración
Recursos Humanos	Jefe de proyecto y Jefe del grupo de migración
Plan de instalación de las herramientas libres	Expertos en SWL y Técnico
Plan de migración y ruta crítica	Expertos en SWL y Planificador
Configuración propuesta	Expertos en SWL y Técnico
Reporte de fallas	Expertos en SWL , Técnico y Jefe de proyecto
Validación de la configuración y el plan de migración	Experto en SWL y Jefe del grupo de migración
Justificación del proceso	Expertos en SWL y Planificador
Planes de Formación	Expertos en SWL y Planificador
Programa analítico	Expertos en SWL y Planificador o el jefe del grupo de migración
Documentación propuesta	Expertos en SWL y Planificador

Anexo 4- Características de las herramientas incluidas en la suite ofimática de OpenOffice.org



Procesador de textos similar en funcionalidad a Microsoft Word y WordPerfect. Se puede exportar desde el formato de Documento Portátil (PDF) sin software adicional, y puede funcionar como un editor WYSIWYG para crear y editar páginas web.



Hoja de cálculo similar Microsoft Excel o Lotus 1-2-3. Calc ofrece una serie de características no presentes en Excel, incluyendo un sistema que automáticamente define serie de gráficas, sobre la base de la disposición de la información del usuario. Calc también puede exportar hojas de cálculo para el formato PDF.



Impress es un programa de presentación similar a Microsoft PowerPoint o Keynote de Apple. Puede exportar presentaciones al formato SWF, permitiendo que se reproduzca en cualquier computadora con un reproductor de Flash instalado. También incluye la capacidad de crear archivos PDF, y la capacidad de leer archivos de Microsoft PowerPoint (ppt).



Programa de base de datos similar a Microsoft Access. OpenOffice.org Base permite la creación y manejo de bases de datos, elaboración de formularios e informes que proporcionan a los usuarios finales un acceso fácil a los datos. Al igual que Microsoft Access, es capaz de trabajar como un *front-end* para diversos sistemas de bases de datos tales como el de Access (JET), fuente de datos ODBC y MySQL/PostgreSQL. Base está disponible desde la versión 2.0.x.



Editor de gráficos vectoriales y herramienta de diagramación, similar a Microsoft Visio y comparable en características a las primeras versiones de CorelDRAW. Ofrece "conectores" entre las formas, que están disponibles en una gama de estilos de línea y facilitan la construcción de los dibujos, como diagramas de flujo. Tiene características similares a la edición de software de escritorio como Scribus y Microsoft Publisher. Draw también puede exportar sus creaciones al formato PDF y al estándar SVG.



Math

Programa para la creación y edición de fórmulas matemáticas, similar a Microsoft Equation Editor. Las fórmulas pueden ser incorporadas dentro de otros documentos de OpenOffice.org, tales como los creados por Writer. Soporta múltiples fuentes y puede exportar a PDF.

Anexo 5- Modelo para definir el peso de los criterios

Nombre del evaluador: _____

Fecha de entrega: _____

Fecha de recogida: _____

Usted debe otorgarle a cada criterio un peso en dependencia del peso total dado al grupo al que pertenece:

Grupo 1: Criterios de interés científico30	
Criterio	Peso
Calidad de la investigación	
Valor científico de la propuesta	
Novedad científica de las fases definidas	
Grupo 2: Criterios de implantación20	
Criterio	Peso
Necesidad de emplear la estrategia al polo de software educativo	
Posibilidad de aplicación de las fases	
Integración con la metodología de migración	
Grupo 3: Criterios de impacto30	
Criterio	Peso
Agilidad en la producción de SWE	
Facilidad para el cambio de SO	
Nivel de utilidad	
Grupo No.3: Criterios de flexibilidad	
Criterio	Peso
Adaptabilidad a proyectos productivos.	
Flexibilidad en la retroalimentación de las fases y tareas.	
Flexibilidad de entendimiento de las fases	

Anexo 6- Modelo para calificar los criterios

Nombre del evaluador: _____

Fecha de entrega: _____

Fecha de recogida: _____

1- Evaluación de los criterios en una escala del 1 al 5:

Grupo 1: Criterios de interés científico	
Criterio	Evaluación
Calidad de la investigación	
Valor científico de la propuesta	
Novedad científica de las fases definidas	
Grupo 2: Criterios de Implantación	
Criterio	Evaluación
Necesidad de emplear la estrategia al polo de software educativo	
Posibilidad de aplicación de las fases	
Integración con la metodología de migración	
Grupo 3: Criterios de impacto	
Criterio	Evaluación
Agilidad en la producción de SWE	
Facilidad para el cambio de SO	
Nivel de utilidad	
Grupo No.3: Criterios de flexibilidad	
Criterio	Evaluación
Adaptabilidad a proyectos productivos.	

Flexibilidad en la retroalimentación de las fases y tareas.	
Flexibilidad de entendimiento de las fases	

Categoría final de la propuesta:

___ Excelente: alta novedad científica, grandes posibilidades de aplicabilidad y relevantes resultados esperados.

___ Bueno: novedoso científicamente y con buenos resultados esperados.

-----No es lo suficientemente bueno, pero puede aplicarse.

-----Cuestionable: sin relevancia científica y con resultados esperados no satisfactorios.

-----Malo: no aplicable

___ Aceptable: no es lo suficientemente bueno, pero puede aplicarse.

___ Cuestionable: sin relevancia científica y con resultados esperados no satisfactorios.

___ Malo: no aplicable

Evaluación final:

- Sugerencias del evaluador para mejorar los procesos propuestos.
- Elementos que deben mejorarse.

Anexo 7-Modelo de entrevista

Nombre y apellidos del entrevistado: _____

Cargo que ocupa: _____

- ¿Qué tipo de herramientas se deberían utilizar para el desarrollo de software educativo?
- ¿Conoce algunas herramientas libres que pueda proponer, como de utilidad en una futura migración hacia las tecnologías libres?
- ¿En la universidad, en el área productiva de software educativo cuales de las mencionadas se han usado o se tiene algún tipo de conocimiento al respecto?

Anexo 8-Tabla 4 Plantilla – Peso Promedio / Criterio

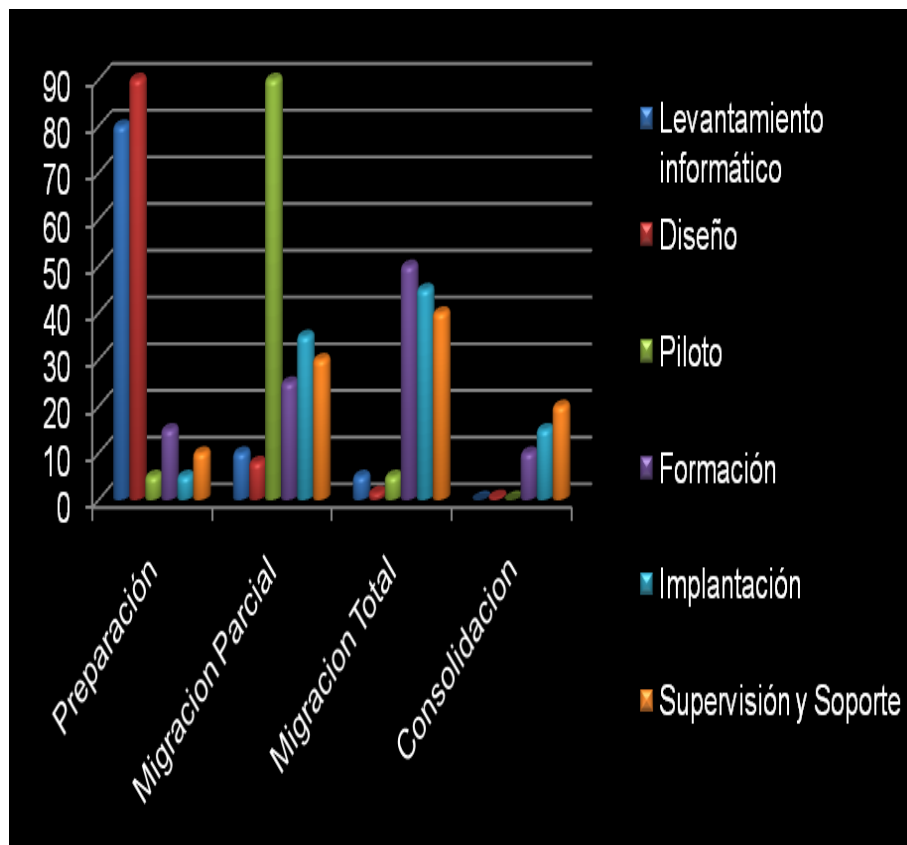
G	C/E	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	∑E	E _p
---	-----	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----	----------------

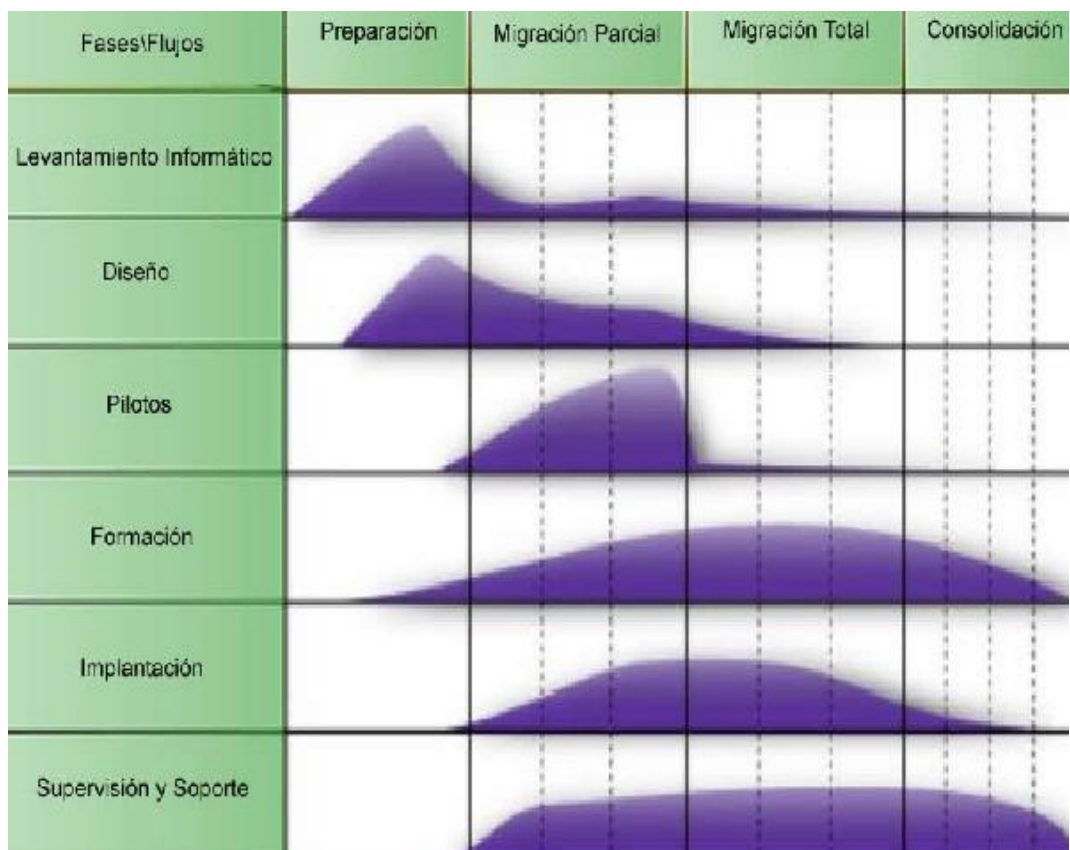
30	C1	8	12	14	8	8	15	8	73	10,4
	C2	13	8	9	13	13	10	11	77	11
	C3	9	10	7	9	9	5	11	60	8,57
20	C4	10	5	7	12	12	10	8	64	9,14
	C5	3	10	7	3	3	5	7	38	5,43
	C6	7	5	6	5	5	5	5	38	5,43
30	C7	5	15	6	5	5	15	5	56	8
	C8	15	5	8	15	15	5	9	72	10,3
	C9	10	10	16	10	10	10	16	82	11,7
20	C10	7	10	6	5	8	10	7	53	7,57
	C11	7	5	7	7	6	5	7	44	6,29
	C12	6	5	7	8	6	5	6	43	6,14
T		100	100	100	100	100	100	100	700	100

Anexo 9-Tabla 5 Plantilla – Consistencia en trabajo de Expertos

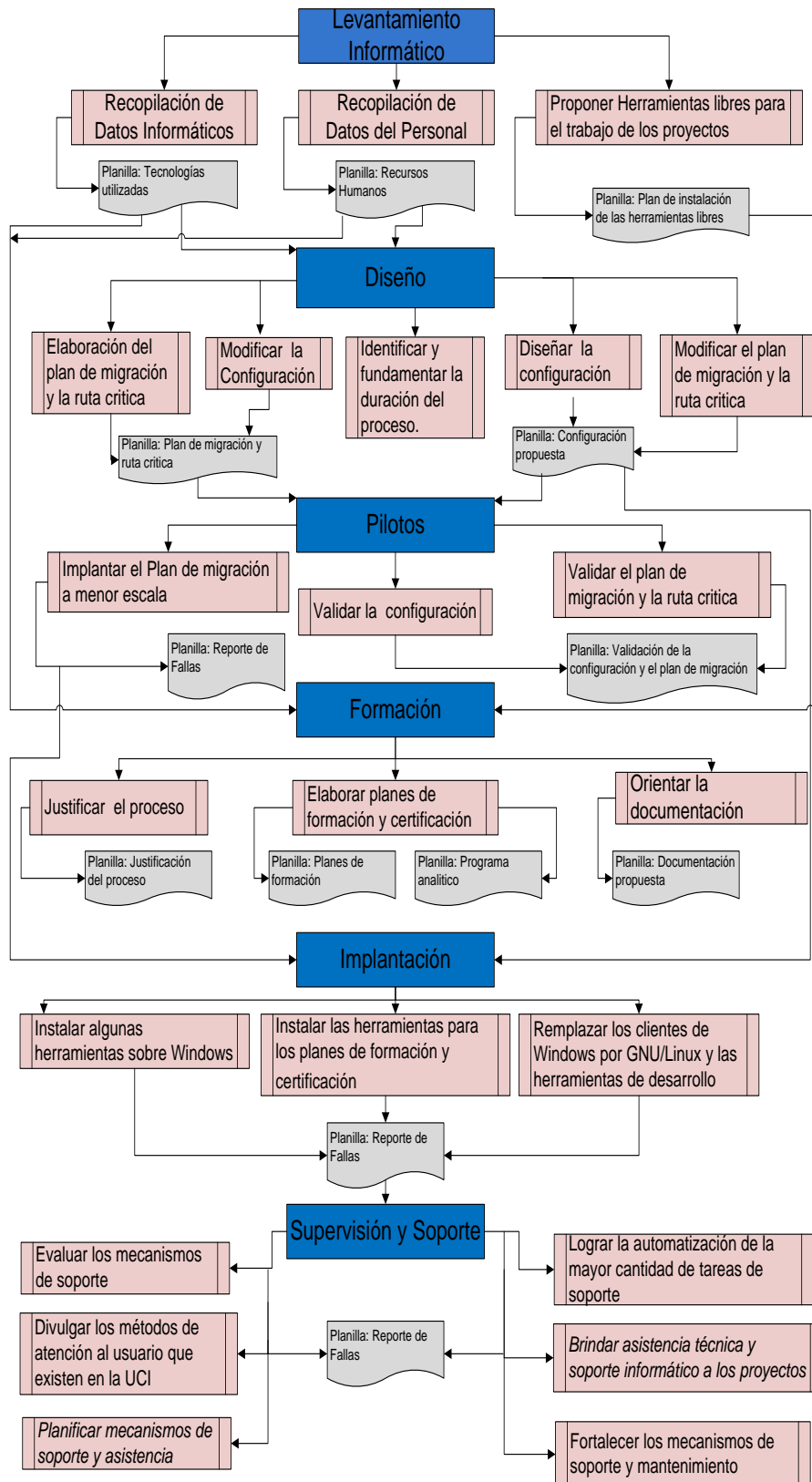
C/E	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	Ep	ΣE/C	ΣE- ΣΣE/C	(ΣE- ΣΣE/C) ²	ΔC	ΔC*ΔC
C1	8	12	14	8	8	15	8	73	10	6,08	17,0833	291,840278	64,7	4715
C2	13	8	9	13	13	10	11	77	11	6,42	21,0833	444,506944	68,67	4225
C3	9	10	7	9	9	5	11	60	8,6	5	4,08333	16,6736111	51,67	2669
C4	10	5	7	12	12	10	8	64	9,1	5,33	8,08333	65,3402778	55,67	3099
C5	3	10	7	3	3	5	7	38	5,4	3,17	17,9167	321,006944	29,67	880,1
C6	7	5	6	5	5	5	5	38	5,4	3,17	17,9167	321,006944	29,67	880,1
C7	5	15	6	5	5	15	5	56	8	4,67	0,08333	0,00694444	47,67	2272
C8	15	5	8	15	15	5	9	72	10	6	16,0833	258,673611	63,67	4053
C9	10	10	16	10	10	10	16	82	12	4,42	26,0833	680,340278	73,67	5427
C10	7	10	6	5	8	10	7	53	7,6	4,42	2,91667	8,50694444	44,67	1995
C11	7	5	7	7	6	5	7	44	6,3	3,67	11,9167	142,006944	35,67	1272
C12	6	5	7	8	6	5	6	43	6,1	3,58	12,9167	166,840278	34,67	1202
T	100	100	100	100	100	100	100	700	100	55,9	156,167	2716,75	600	32690
n*k								84						
MΣE								8,3						
S												2716,75		
W = S / E2 (C3 – C) / c-1 ----Kendall								0,4						
X2=E(C-1)*W-----Ji cuadrado								28						

Anexo 11-Gráfica de porcentaje de esfuerzo de los flujos por fase



Anexo 12-Gráfica de Flujos de trabajo por Fases

Anexo 13-Diagrama de Relación entre los flujos de trabajo y los artefactos



Leyenda:



Flujo de trabajo



Actividad del Flujo de trabajo



Plantilla que generan las actividades



Relación entre Flujo de trabajo y actividad, así

como relación entre la actividad y la plantilla. Además relaciona las plantillas que son necesarias para dar inicio a un flujo de trabajo determinado.

Anexo 14-Tabla 7 Requerimientos de hardware para distribuciones de Linux

Distro gnu/Linux	CPU (procesador)	RAM (Mínimo)	RAM (Recom.)	Espacio en disco (mínimo)	Espacio en disco (Recom.)
openSUSE 10.3	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Athlon 64, Sempron u Opteron	256 MB	512 MB	500 MB	3 GB
openSUSE 10.1	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	256 MB	512 MB	500 MB	3 GB
CentOS 5	Intel Pentium I/II/III/IV/Celeron/Xeon, AMD K6/II/III, AMD Duron, Athlon/XP/MP, (Pendiente Itanium, Athlon 64, PowerPC)	128 MB	512 MB	1.2 GB	2 GB
CentOS 3/4	Intel Pentium I/II/III/IV/Celeron/Xeon, AMD K6/II/III, AMD Duron, Athlon/XP/MP, Itanium, Athlon 64, PowerPC	128 MB	256 MB	800 MB	2 GB
Mandriva 2008	Cualquier procesador Intel o AMD, 1Ghz o superior. Soporte para procesadores dual-core.	256 MB	512 MB	3 GB	4 GB
Mandriva 2007	Pentium 1-4, Celeron, Athlon, Duron, Sempron.	256 MB	512 MB	500 MB	4 GB
Mandriva 2006	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Sempron, Opteron, K6, Via C3	128 MB	256 MB	500 MB	4 GB
Fedora 8	Intel Pentium 2-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	128 MB*/194 MB**	256 MB	500 MB	3 GB
Fedora 7	Intel Pentium 2-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	128 MB*/194 MB**	256 MB	500 MB	3 GB
Fedora Core 5	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	64 MB*/194 MB**	256 MB	500 MB	3 GB

Debian 3.1	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	32 MB*/194 MB**	256 MB	500 MB	3 GB
Debian 3.0	Intel Pentium 1-4, Celeron, AMD Duron, Athlon, Sempron u Opteron	16 MB*/64 MB**	128 MB	450 MB	4 GB
Xubuntu 7.10	Intel o AMD con velocidad de 500mhz	64 MB/192 MB***	128 MB	1.5 GB	4 GB
Kubuntu 7.10	Intel o AMD con velocidad de 500mhz	384 MB	384 MB	4 GB	4 GB
Ubuntu 7.10	Intel o AMD con velocidad de 500mhz	256 MB	384 MB	4 GB	4 GB
Ubuntu 6.06	Intel o AMD con velocidad de 500mhz	256 MB	256 MB	2 GB	3 GB
Ubuntu 5.10	Intel o AMD con velocidad de 500mhz	192 MB	256 MB	2 GB	3 GB
Slackware	486 o superior	16 MB	32 MB	100 MB	3.5 GB
Damn Small 3.0	486DX o superior	16 MB	128 MB	–	–
Damn Small 1.0	486DX o superior	16 MB	64 MB	–	–
Slax 5.1	486, Pentium o AMD	36 MB	fluxbox 96 MB KDE 144 MB	–	–
Puppy 2	Pentium 166MMX o superior (Intel o AMD)	128 MB	128 MB	–	–
simplyM EPIS 6	Intel Pentium o AMD Athlon	128 MB	512 MB	2 GB	3 GB
Knoppix 5	Intel Pentium o AMD Athlon	32 MB*/96 MB**	128 MB	–	–