



**METODOLOGIA PARA LA MIGRACION A SOFTWARE LIBRE DE LAS
UNIVERSIDADES DEL MINISTERIO DE EDUCACION SUPERIOR
(MES)**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TITULO DE INGENIERO EN CIENCIAS
INFORMATICAS.

AUTOR: YOANDY PEREZ VILLAZON

TUTOR: ING. ABEL MENESES ABAD

**CIUDAD DE LA HABANA 30 DE MAYO DEL 2008
"AÑO 50 DE LA REVOLUCION"**

DECLARACION DE AUTORIA:

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

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

YOANDY PEREZ VILLAZON

Firma del autor.

Ing. ABEL MENESES ABAD

Firma del tutor(a).

AGRADECIMIENTOS

Muchas son las personas que de una forma u otra han colaborado con la realización de este trabajo, sin las cuales hubiese sido imposible la elaboración del mismo.

Agradezco entonces a:

- ✓ **Abel Meneses Abad**, incansable en los trabajos de la Migración a Software Libre y profundo conocedor de los procesos asociados a la misma, quién con sus críticas y consejos siempre estuvo al tanto de la realización de este trabajo.
- ✓ **Ramón Paumier Samón**, elaborador de la primera metodología de migración a Software Libre cubana sobre la cual se elaboró este trabajo, quien además ha tocado en la realidad un proceso de migración de gran envergadura y estuvo al tanto de la realización de esta metodología.
- ✓ **David Padrón Alvarez**, quien con sus enormes conocimientos de diseño gráfico en Software Libre facilitó el sueño de hacer esta tesis de grado solo con herramientas libres ayudándome en la elaboración de los diagramas.

A todos los que colaboraron,

Gracias.

DEDICATORIA

A nuestro eterno Comandante en Jefe Fidel Castro Ruz, creador de la Universidad de las Ciencias Informáticas, alta casa de estudios que hoy me permite graduarme como profesional en Ciencias Informáticas.

A mis padres que me apoyaron en todo momento y derrocharon esfuerzo y dedicación con el fin de educarme como un hombre de bien graduando el 1er ingeniero de la familia.

A mi hermana y hermanos jimaguas para los cuales soy la guía a seguir, a ellos les dedico este triunfo que es por ellos.

A mis abuelos que me han apoyado y dado fuerzas.

A mi novia que ha sido mi apoyo en estos últimos años de carrera.

A mis amigos de 5 años en la universidad que me han ayudado.

RESUMEN

El uso del Software Libre es cada vez más necesario en nuestro país por lo que este representa para los cubanos en los aspectos económicos, políticos y sociales, pero para poder realizar una migración en el país se hace necesario poseer personal capacitado para acometer el proceso.

Las universidades son la fuente de investigación y conocimientos más grande del país, contamos hoy con 3150 sedes universitarias y 65 escuelas de altos estudios por todo el país. La incrustación de Software Libre en estos centros sin duda alguna sería el paso más grande que daría el país en materia de migración. Para que el proceso de migración y esparcimiento de Software Libre tenga éxito es importante la realización de un modelo de migración que guíe el proceso. Para la realización de dicha tarea se elaboró un estado del arte del Software Libre y el impacto de este en la educación.

Se analizaron las estrategias de migración de dos universidades: la primera el Plan Institucional de Liberación y Migración a Plataformas de Software Libre de la Universidad de los Andes (ULA), y la segunda la Metodología de Migración de la Universidad de las Ciencias Informáticas (UCI); además se estudió la guía de migración a Software Libre de la Unión Europea, un documento muy completo que explica cómo hacer procesos de este tipo.

Como último punto se realizó bajo la base del análisis de los documentos anteriores y el estado de la universidad cubana hoy una propuesta de metodología de migración para los centros anteriormente mencionados, siendo este el objetivo general de esta investigación.

Índice de contenido

RESUMEN.....	V
INTRODUCCION.	1
1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Historia y definiciones del Software Libre.....	4
1.2. ¿Por qué migrar a Software Libre?.....	8
1.3. Software Libre a nivel internacional. ¿Estado actual?.....	13
1.4. Significación del uso del Software Libre en Cuba.....	18
1.4.1. Político.....	19
1.4.2. Social.....	19
1.4.3. Tecnológico.....	20
1.4.4. Económico.....	20
1.5. La universidad cubana.....	20
1.6. Conclusiones.....	24
2. CAPÍTULO II: ANALISIS DE LAS PROPUESTAS.....	25
2.1. Introducción.....	25
2.2. Elementos de un ejercicio de migración.....	25
2.3. Análisis de las directrices IDA de la Unión Europea.....	31
2.4. Análisis del Plan de Migración de la Universidad de los Andes.....	35
2.5. Análisis de la Metodología de Migración de la Universidad de las Ciencias Informáticas.	39
2.6. Conclusiones.....	43
3. CAPÍTULO III: METODOLOGIA DE MIGRACION A SOFTWARE LIBRE PARA LAS UNIVERSIDADES DEL MES.....	45
3.1. Introducción.....	45
3.2. Metodología propuesta. Generalidades.....	45
3.3. Flujo de trabajo : Evaluación.....	48
3.4. Flujo de trabajo : Diseño.....	49
3.5. Flujo de trabajo : Pilotos.....	51
3.6. Flujo de trabajo : Formación.....	53
3.7. Flujo de trabajo : Implementación.....	55
3.8. Flujo de trabajo : Asistencia y soporte técnico.....	57
4. CONCLUSIONES.....	60
5. RECOMENDACIONES.....	61
6. REFERENCIAS BIBLIOGRAFICAS.....	62
7. BIBLIOGRAFIA.....	65
8. ANEXOS.....	66
8.1. ANEXO 1: HERRAMIENTAS FLOSS, PARA APLICACIONES WINDOWS.....	66
8.2. ANEXO 2: HERRAMIENTAS LIBRES PARA MICROSOFT WINDOWS.....	70
8.3. ANEXO 3: NIVELES DE USUARIOS PROPUESTOS PARA LA FORMACION.....	71
8.4. ANEXO 4: SOFTWARE UTILIZADO EN LOS PRINCIPALES SERVICIOS SOBRE GNU/LINUX.....	71
8.5. ANEXO 5: FLUJO DE PROCESOS EN LA METODOLOGIA.....	72
9. GLOSARIO DE TERMINOS.....	73

INTRODUCCION

La industria cubana del software hoy en día se ve seriamente afectada por el bloqueo norteamericano impuesto a la isla por parte del gobierno de los Estados Unidos de América (EUA), el cual prohíbe usar en territorio cubano cualquier producto que provenga de empresas norteamericanas, por lo que la exportación de software hacia el extranjero de las aplicaciones construidas se ve seriamente afectada, muchas veces tenemos que recurrir a terceros países para vender los productos y servicios informáticos que se producen en la isla lo cual disminuye en gran medida el costo de venta de estos productos produciendo pérdidas en las ventas que más tarde repercuten en la economía nacional.

El software propietario usado hoy en Cuba proviene en más del 80% de empresas norteamericanas, por lo que además de representar problemas para la economía nuestro país no posee licencias de uso del mismo y en gran medida es software pirateado el que usamos en casi toda la infraestructura nacional – representando esto una gran amenaza – aunque el bloqueo yanqui nos “permite” piratear las aplicaciones ya que las leyes de la propiedad intelectual vigentes en la constitución de los EUA no pueden ser aplicadas a Cuba. Si es posible que Microsoft u otra empresa de software pueda lanzar una campaña de desacreditación moral contra la isla por usar software sin el permiso de las empresas que lo producen, lo cual sería muy lamentable para nuestro país que defiende los principios más altruistas que existen. Por otra parte el uso del software privativo tiene asociado principios poco éticos como pueden ser el no acceso a la información, no permitida la colaboración, por solo citar dos elementos que están muy asociados al sistema socialista que defendemos hoy en día en nuestro país y que es el sistema que queremos siga vigente para nuestros hijos.

La alternativa para estos males que afectan la nación tanto en el aspecto social, económico y político, existe, se expande por el mundo a una velocidad incalculable arrastrando a organizaciones, empresas y gobiernos de todas las latitudes y se denomina Software Libre que más que una simple forma de distribuir el software es un movimiento tecnológico considerado por muchos hoy el principal enemigo del capitalismo; este movimiento tiene arraigado los principios más nobles a sí mismo, promoviendo el trabajo en equipo, la libertad de acceso a la información, la no privatización del código fuente que desarrollamos; en fin son los principios para llegar a la Sociedad de la Información justa

que defiende la humanidad hoy.

Para poder llegar a esa sociedad a la que aspira la humanidad hoy es necesario comenzar con una fuerte tarea de formación de la misma que se inicie a todos los niveles comenzando por la escuela siendo este el medio donde se formarán los principales cuadros que representan la sociedad en las distintas áreas de la misma. Cuba posee un consolidado sistema de educación por lo que la implantación de un modelo educativo basado en Software Libre no sería una tarea de gran complejidad.

Las tareas asociadas a la migración hacia software de fuente abierta además del cambio en materia de tecnología tienen asociadas otras tareas de suma importancia como es la capacitación del personal, en Cuba un tercio de la población esta influenciada directamente por el proceso de formación académica, la universidad cubana constituye la principal fuente de capacitación de pregrado y postgrado para la formación en Software Libre y el alcance de la soberanía tecnológica nacional, surge entonces la necesidad de migrar la educación superior cubana siendo el **problema científico** de esta investigación el ¿cómo migrar las universidades cubanas a Software Libre?

Para poder ejecutar este proceso de migración de manera satisfactoria se plantea como **objeto de estudio** el análisis de los procesos de migración hacia Software Libre de distintas universidades del mundo y cubanas que hayan sido exitosos o no, así como los distintos grupos de medidas tomadas por algunas organizaciones; es el **objetivo general** de esta investigación definir la propuesta de una metodología de migración a Software Libre para los centros asociados al Ministerio de Educación Superior (MES) en Cuba.

Como parte de la estrategia a seguir para poder lograr el objetivo general de esta investigación se han enumerado los siguientes **objetivos específicos** que tienen como **campo de acción** la migración a Software Libre de la Educación Superior Cubana:

1. Estudiar los intentos de migración a Software Libre de universidades cubanas y extranjeras.
2. Elaborar una propuesta de migración propia para los centros asociados al MES.

El presente trabajo consta de 3 capítulos, estructurados de la siguiente forma:

Capítulo I Fundamentación Teórica : En este se definen una serie de conceptos acerca del

Software Libre, GNU/Linux incluyendo elementos históricos acerca del surgimiento de los mismos, el capítulo aborda también lo relacionado con las licencias de software, se muestran las ventajas y desventajas del uso de GNU/Linux como sistema operativo libre y se comenta acerca del por qué no ha triunfado el Software Libre a pesar de las ventajas del mismo, se muestra un estudio acerca del uso del Software Libre a nivel internacional y la aceptación del mismo por diversas organizaciones, gobiernos e instituciones a nivel internacional, se resalta la importancia que representa para Cuba, se concluye realizando un resumen del estado de la universidad cubana hasta finales del año 2007 haciendo énfasis en cambios recientes sufridos como la universalización y el uso de herramientas libres en la Educación Superior.

Capítulo II Análisis de las propuestas : A lo largo de este capítulo se realiza un profundo estudio de los documentos, guías, artículos y materiales de manera general encontrados en internet que tratan los elementos relacionados con temas de migración hacia Software Libre, se definen un conjunto de elementos que debe contener las guías de migración y se someten a un estudio exhaustivo-crítico tres documentos, el primero de ellos las Directrices IDA de la Unión Europea, en segunda instancia el Plan Institucional de Liberación y Migración a Plataformas de Software Libre de la Universidad de los Andes (ULA) de Venezuela y por último se analizó la Metodología para la Migración a Software Libre de la UCI elaborada en el año 2007, cuyo documento constituye una base para la elaboración de este.

Capítulo III Metodología de Migración a Software Libre para las Universidades del MES: En este capítulo se realiza la propuesta de metodología de migración para el MES, se definen y explican las etapas de la misma y los flujos de trabajo asociados que darán cumplimiento a la metodología.

1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1. Historia y definiciones del Software Libre.

En los albores de la computación el software no era considerado un producto independiente , sino un añadido que se vendía junto al hardware sin precio alguno, los principales productores de aquellas enormes computadoras entregaban este a los consumidores y la cultura existente favorecía el intercambio entre programadores y desarrolladores sin interés alguno, en la década de 1970 se comenzó el desarrollo, en los laboratorios de AT&T Bell Labs, del sistema operativo UNIX.

A finales de los 70 las compañías comenzaron a imponer licencias de software a los compradores, estas concedían autorización o permiso por el titular de forma contractual al usuario de un programa para utilizar este producto de acuerdo a un conjunto de condiciones previamente estipuladas.

A inicios de los años '80 la transnacional International Bussines Machine (IBM) decidió comenzar a comercializar un ordenador personal en respuesta al éxito que la empresa Apple comenzaba a tener en el mercado, y se comienza a distribuir el IBM PC.

Con el surgimiento de estas licencias de software se comenzaba a crear una cultura bastante diferente a la que existía anteriormente, ya los desarrolladores no podían acceder al código fuente de las aplicaciones que compraban o adquirían, cuando se detectaba un problema en el funcionamiento de un producto la única opción posible era escribir al desarrollador del mismo y reportar el problema y esperar a que estos le dieran solución.

Tal caso como el descrito anteriormente tuvo lugar en el Massachusetts Institute of Technology (MIT), el laboratorio donde trabajaba Richard Mattews Stallman poseía una impresora la cual utilizaban en la red para que un grupo de personas pudiesen imprimir en ella, resulta que esta poseía problemas y se atoraba seguidamente y cuando las personas enviaban trabajos para imprimir que los iban a buscar se encontraban con la impresoras en tal estado y con una gran cantidad de tareas sin procesar, Stallman que era programador destacado decide crear un pequeño sistema de avisos para notificar cuando dicha impresora estuviese atascada, pero el driver de la misma era privativo por lo que no se podía acceder al código de la misma, entonces decide escribir a la empresa que fabricó dicho producto para que le dieran al acceso para poder modificar este pequeño elemento,

la empresa se negó ante la petición.

Richard que era desarrollador de programas informáticos se vió en la disyuntiva de aceptar la nueva corriente que se avecinaba: crear software privativo y no revelar su contenido, o cambiar el futuro.

Con estas condiciones existentes en enero de 1984 el neoyorkino anunció como respuesta a las tendencias de licenciamiento de software del momento la creación de GNU, un acrónimo recursivo que aún hoy significa GNU is Not Unix, cuyo objetivo era crear un sistema operativo basado en UNIX pero libre. Las características de la arquitectura de UNIX lo hacían un sistema estable y seguro, por lo que lograr un sistema con características similares a este pero libre habría sido ideal, el siguiente fragmento de correo (Richard M. Stallman 1983) tomado del anuncio oficial explicaba:

From: RMSMIT-OZ@mit-eddie

Newsgroups: net.unix-wizards,net.usoft

Subject: new Unix implementation

Date: Tue, 27-Sep-83 12:35:59 EST

Organization: MIT AI Lab, Cambridge, MA

.....

Porqué debo escribir GNU

Considero que la regla de oro exige que si yo quiero un programa debo compartirlo con otras personas que también lo quieren. No puedo, conscientemente, firmar un acuerdo de confidencialidad o un acuerdo de licencia de software.

Para que yo pueda continuar utilizando las computadoras sin violar mis principios, he decidido reunir suficiente software libre de manera que

podré continuar sin necesidad de utilizar algún software que no sea libre.

.....

Para más información, contácteme

Correo Arpanet:

RMS@MIT-MC.ARPA

El término de la libertad no se refiere en ningún momento a gratis referente al precio, sino a libre en términos de la libertad de expresión, el término de la libertad hace referencia a :

- ✓ La libertad de usar el programa, con cualquier propósito (libertad 0).
- ✓ La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1).
- ✓ La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- ✓ La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3).

Nota : La libertad 1 y 3 exigen el acceso al código fuente de la aplicación.

Un programa es software libre si lo usuarios tienen todas estas libertades sobre el mismo, se debe tener la libertad de distribuir copias ya sea con modificaciones o sin las mismas, sea gratis o cobrando cierta cantidad por las mismas, es, de forma general hacer uso del mismo sin pedir o pagar permisos (Proyecto GNU 2001).

Para lograr que el software cumpliera estas disposiciones una vez estuviese en manos de los usuarios y evitar restricciones posteriores del mismo en 1985 se crea la Fundación para el Software Libre (FSF), con el objetivo de proveer soportes logísticos, legales y financieros al proyecto GNU, esta organización creada sin ánimo de lucro para el desarrollo contrató varios programadores y desarrolladores para crear aplicaciones para el sistema operativo GNU, aunque vale decir que la mayor parte de los productos han sido escritos por la comunidad internacional de forma voluntaria; ya para finales de los 80 se crea la Licencia

Pública General (GPL) para servir de sostén legal a todo el software que se iba creando, creando el término de copyleft o copia permitida en contrapartida al copyright.

En la década de los 90 el sistema GNU contaba con aplicaciones importantes tales como el editor de textos multifuncional Emacs, el compilador GCC y muchas librerías que formaban parte de los sistemas UNIX, pero faltaba un componente esencial, el núcleo encargado de hacer interactuar el software del sistema con el hardware y los dispositivos del mismo gestionando los recursos a través de llamadas al sistema .

En la otra parte del mundo, Linus Torvalds, un joven universitario terminaba de graduarse y continuaba como diversión la creación de un núcleo basado en Minix (Linus Benedict Torvalds 1991), dicho sistema fue puesto público en internet bajo licencia de código compartido en el año 1991 bajo el nombre de Linux.

El kernel creado por Torvalds era el complemento para el conjunto de aplicaciones que se habían creado en el proyecto GNU, posteriormente dichos proyectos se unieron para crear GNU/Linux.

De forma general la evolución de este movimiento según Lena Zuñiga en su investigación social sobre el Software Libre en América Latina (Lena Zuñiga Blanco 2006). puede resumirse en el siguiente esquema.

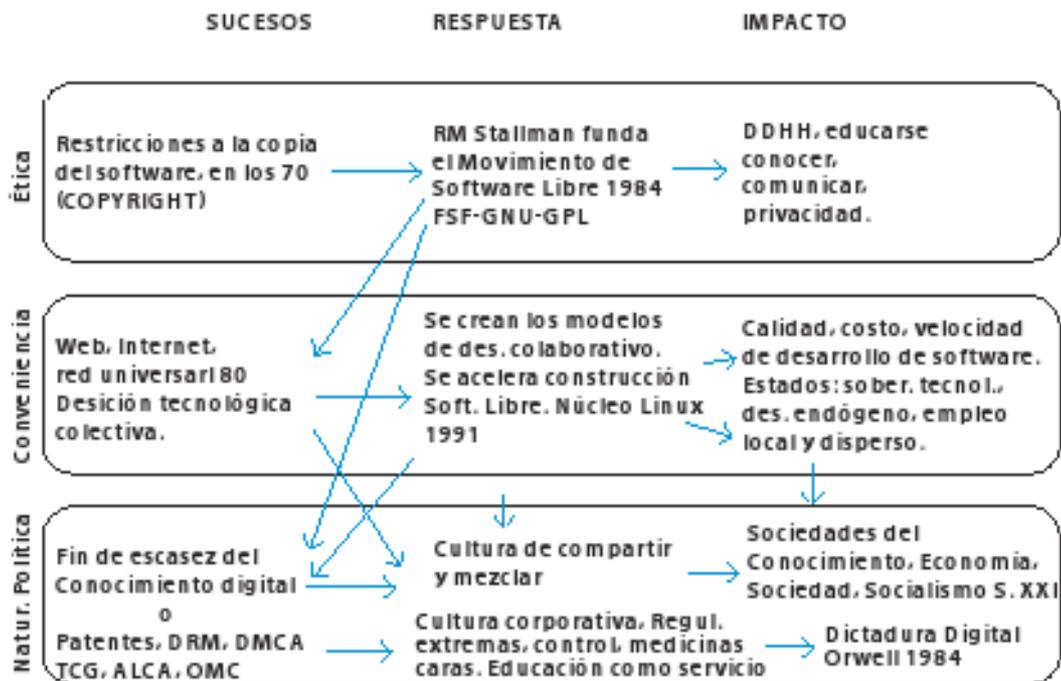


Figura 1.1. Evolución del Software Libre.

1.2. ¿Por qué migrar a Software Libre?

GNU/Linux es la familia de sistemas operativos libres denominados distribuciones. En sus distintas variantes incorporan un conjunto de paquetes de software específicos acordes con la funcionalidad a la que están destinados. De forma general todas llevan el kernel desarrollado por Linus y el conjunto de paquetes de software desarrollados por la GNU y otras fundaciones. El Software Libre cada día gana más espacio en las aplicaciones de escritorio; y en los servidores se ha consolidado como un fuerte oponente al software privativo por sus características de robustez, seguridad, estabilidad y alta configurabilidad.

En la tabla se visualizan algunos aspectos, y la forma en que se manifiestan en cada uno de estas clasificaciones de software, y elementos que muestran características superiores en el Software Libre.

Tabla 1.1 : Comparación entre Software Privativo y Software Libre.

Aspectos a considerar	Software Privativo	Software Libre
Acceso al código fuente	Prohibido por licencia	Si, garantizado
Corrección de errores en el programa por el cliente usuario	No	Si
Duplicación del software	Prohibido	Posible y recomendada
Libertad de competencia para el mantenimiento	No, depende del fabricante	Si, imposible limitarla
Posibilidad de examinar el código del producto	Prohibido salvo permiso del fabricante	Si
Venta de 2da mano	Prohibido	N/A
Respeto a estándares globales	En función del fabricante	En la mayoría de los casos
Adaptaciones al cliente	En función del fabricante	Disponible
Virus, gusanos	Frecuentes	Muy infrecuentes

Origen: Libro blanco de Software Libre en España (2004).

Para los proveedores del software las ventajas del Software Libre sobre el software privativo van mucho más allá de la parte técnica. Tanto los clientes consumidores de software como sus proveedores mejoran su actividad mediante el uso de Software Libre. Los proveedores pueden ofrecer tiempos de respuesta más bajos, mejor calidad, menos tiempo en I+D, planificaciones más exactas, ahorro en marketing (las aplicaciones libres

suelen tener su propia Web, documentación, explicación de ventajas y son de libre descarga y uso), menos pruebas piloto y demostraciones, entre otras. Todo este ahorro se suele invertir en adaptación del software y soporte técnico al cliente (Ramón Paumier Samón 2007).

Por otra parte es importante decir que las aplicaciones de Software Libre en muchos casos son desarrolladas o contribuyen a su desarrollo empresas transnacionales y poderosas económicamente que sin duda alguna lo impulsan de manera increíble, un ejemplo es IBM que colabora donando sumas de dinero representativas para reforzar aspectos del Software Libre como es el caso de la documentación, mediante el sitio web DevelopersWorks, también contribuye al kernel de Linux con su sistema de ficheros transaccional JFS (Steve Best 2000) y desarrolla la herramienta eclipse (IBM 2007) para la programación en Java que es análoga a Jbuilder pero libre, los escritorios de la empresa utilizan sistemas GNU/Linux, posee un gran número de anuncios publicitarios incitando al uso de plataformas libres que han tenido gran impacto sobre la comunidad internacional.

A niveles empresariales y gubernamentales, de manera general, implica no tener que pagar grandes cantidades de dinero por conceptos de licencias, y no verse obligado a cumplir estrictamente las disposiciones legales que implican los contratos establecidos por los productores de software. El ahorro de dinero que se obtiene de no pagar estos impuestos puede ser utilizado para otras tareas de mayores prioridades. El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. El secretismo tecnológico es uno de los grandes frenos y desequilibrios existentes para el desarrollo en el modelo de propiedad intelectual. Otra ventaja significativa es la relacionada con la independencia tecnológica, al proveer el acceso al código de la aplicación cualquier entidad o persona puede ofrecer los servicios asociados al software, eliminándose la dependencia de empresas y si estas últimas salen del mercado por cualquier motivo o se discontinúa el soporte, pueden ser mantenidas por terceros.

A diferencia de la familia de sistemas operativos privativos, cuando el proveedor abandona el mercado los clientes están obligados a migrar hacia versiones superiores de los productos pagando grandes costos asociados. En ocasiones debido al cambio de software o por la necesidad inducida de comprar hardware mejorado (generalmente el hardware caduca con las aplicaciones, esta es una técnica de mercado que favorece a las

transnacionales, y casi nunca a los clientes).

Otra temática importante es la seguridad y privacidad que acompañan al Software Libre, a lo largo de la historia el software a sido revisado por cientos de programadores en todo el mundo por lo que este se encuentra libre de errores y/o puertas traseras lo que constituye un elemento de basta seguridad para los usuarios y empresas evitando la posibilidad de que los creadores puedan “espiarte” y saber que haces en cada momento, un estudio realizado por las empresas Covery especializada en auditoría de código y evaluación de riesgos y Symantec en conjunto con la Universidad de Stanford (The White House Washington 2003) analizaron 32 proyectos de Software Libre más utilizados por todo el mundo y que pueden considerarse como críticos para la seguridad nacional de los Estados Unidos en la infraestructura de internet, se comparó cada aplicación libre con su equivalente propietaria arrojando el resultado siguiente:

Tabla 1.2. Errores encontrados por cada mil líneas de código

Tipo de Software	Errores por 1000 líneas de código
Software Libre	0.434
Software Privativo	20-30

Las ventajas de usar GNU/Linux para los usuarios finales también es evidente, muchas distribuciones están orientadas al escritorio y el usuario puede descargar el software de internet, instalarlo y probarlo libremente eligiendo dentro del gran número de herramientas existentes cual usar para una determinada tarea; para usuarios avanzados puede incluso adaptarla a sus necesidades. Existen muchas aplicaciones libres de igual o mejores prestaciones que las privadas que van desde aplicaciones tan sencillas como un reproductor para escuchar música hasta potentes navegadores como es el caso de Mozilla Firefox considerado por la revista PC World como el mejor producto informático del año 2005 (PC World 2005).

Existen muchas personas y organizaciones que plantean no usar Software Libre porque no

existen empresas que le den soporte a estos productos, y que se responsabilice de estas aplicaciones, algo que no es totalmente cierto. Si bien existen diversas distribuciones de GNU/Linux sin asistencia técnica personalizada a los usuarios existen otras como Red Hat, Suse o Mandriva que sí ofrecen tales bondades (RedHat Inc 2008),(Novell 2008),(Mandriva 2008). Esta industria sin embargo vive de los servicios y muchos grupos se encargan de dicho tarea y así obtienen los recursos necesarios para sustentarse.

Es válido preguntarse entonces ¿por qué si el Software Libre posee tantas bondades las empresas, gobiernos y demás miembros de la sociedad no usan estas tecnologías? La respuesta a esta interrogante implica hacer referencia a varios factores, en primera instancia nos referiremos a la resistencia al cambio, muchas personas creen aún que el Software Libre tiene un pobre desarrollo. Otros criterios señalan con desdén que GNU/Linux no es usado por ser solo una consola de comandos; otros que sí conocen el sistema simplemente no lo usan por el echo de tener que aprender algo nuevo. Esto implica destinar tiempo a esta tarea. Otro elemento interesante es la política seguida por empresas desarrolladoras de software privativo, al crear convenios con empresas creadoras de hardware para poner a Microsoft Windows instalado de forma predeterminada en sus máquinas. Esto ha comenzado a cambiar en los últimos tiempos (Marcos Guglielmetti 2008) y cuando el usuario compre el computador ya tendrá este sistema operativo previamente instalado. Lo que sin duda es una estrategia de mercado para adquirir clientes, forzando a estos a usar las herramientas disponibles en la plataforma y encadenándolo a las políticas escritas en la licencia.

Por su parte la utilización de programas ilegales es una práctica que está de moda en el mundo actual, si todas las personas que usan software privativo tuviesen que pagar los altos costos de las licencias las tecnologías libres fueran más ampliamente usadas. Otro punto que sustenta el amplio uso del software privativo es la creación por parte de empresas productores de hardware como memorias flash y otros dispositivos de manejadores (drivers) para que puedan ser solo utilizados por el sistema operativo Microsoft Windows y estas no revelan el código o funcionamiento de las mismas de modo que muchas veces los desarrolladores de GNU/Linux deben hacer ingeniería inversa para lograr un resultado similar.

La última razón no menos importante está relacionada con el conocimiento. Como bien dijo hace ya más de un siglo el filósofo francés Augusto Comte (1798 – 1857) “saber es poder”.

Precisamente muchos de los usuarios administradores de redes no poseen el conocimiento suficiente respecto a las nuevas tecnologías libres, y creen perder el poder, haciendo resistencia al cambio tecnológico.

Independientemente a todas estas ataduras y manipulaciones por parte de las grandes empresas una migración a Software Libre es totalmente factible y actualmente muchas organizaciones la implementan.

1.3. Software Libre a nivel internacional. ¿Estado actual?

El Software Libre más que una forma de producción y distribución de software es un movimiento tecnológico que se ha extendido por el mundo entero los últimos años donde a través de comunidades virtuales la gente intercambia y comparte bajo una filosofía cooperativista; el hecho de que el software sea abierto ha permitido que se haya extendido a tantos lugares del planeta gracias a los proyectos de internacionalización y traducción de herramientas y documentación. Tomemos como ejemplo la distribución de software disponible en ruso, sueco, finés, danés, japonés y muchos otros.

El Software Libre se encuentra hoy sumamente extendido en el mundo logrando mayor ocupación en países del tercer mundo y suele ser como un impulso en los países en desarrollo que lo están acogiendo en gran medida como es el caso del grupo de los BRIC compuesto por Brasil, Rusia, India y China, en un estudio realizado por Jesús Javier Estepa Nieto en Julio del 2007, se detalla el impacto que ha tenido el Software Libre en los países del tercer mundo y como se están organizando estos para los procesos asociados a la migración.

Muchos individuos y organizaciones a nivel internacional poseen diversos criterios sobre el Software Libre, por ejemplo:

- ✓ Algunos, consideran el Software Libre como un competidor del capitalismo, una forma de anarquismo práctico.
- ✓ Otros, como otra forma de competición en el mercado libre, y que el copyright es una restricción gubernamental sobre el mercado.
- ✓ No faltan los que consideran el Software Libre como una forma de cooperación en un modelo de mercado en una línea próxima al mutualismo.

- ✓ Hay quienes comparan el Software Libre a una economía del regalo, donde el valor de una persona está basado en lo que esta da a los demás, así como un parentesco a la economía participativa, algo por supuesto que ofrece ventajas a los tercermundistas.

Independientemente a estas posturas el nuevo modelo que trae arraigado el Software Libre es más justo para las pequeñas y medianas empresas, el nuevo paradigma de una empresa basada en servicios hace más justa la competencia y se basa más en la capacidad de las personas que en el poder financiero que pueda poseer una empresa.

Actualmente son innumerables los gobiernos empresas y entidades que usan o están organizando procesos migratorios en todos los continentes del planeta, a continuación exponemos algunos ejemplos:

- ✓ **Europa**

- Ayuntamiento de Munich, Alemania, ciudad de 1,3 millones de habitantes que migraron sus infraestructuras de información para aislarse del dominio de Microsoft.
- Extremadura, España, ciudad pionera del Software Libre en España y ejemplo de referencia para la comunidad internacional.(Jesus Javier Estepa Nieto 2007)
- Andalucía, España , ciudad migrada a partir de lo elaborado por extremadura crearon su propia junta.
- Ayuntamiento de Amsterdam, Holanda, la ciudad de Amsterdam destinó para el 2007 una suma 300 000 euros para probar Software Libre y hacer pruebas, si todo era satisfactorio se migraría la ciudad.
- Otra países involucrados en procesos de migración en el área Europea son : Inglaterra y Francia que poseen varias ciudades involucradas en estos procesos.

- ✓ **África**

- Ministerio de la Agricultura Desarrollo Rural y Pesca, durante julio del 2006 anunció su intención de usar Software Libre, es el primer ministerio marroquí que

le saca provecho a las tecnologías soberanas.

- Sudafrica, este fue el primer país africano que mostró su interés en migrar a Software Libre mostrando su interés desde el año 2001.

✓ **América**

- Argentina, es otro de los países a seguir en el uso del Software Libre, posee su propio distribución de GNU/Linux llamada Ututo considerada por el propio Richard Stallman como una distribución de GNU/Linux libre completamente.
- Brasil esta ejecutando cambios en la Administración Pública y posee un desarrollo de Software Libre a un ritmo impresionante que abarca todas la regiones del país.
- Venezuela, es otro de los países que va delante en la región, desarrolla actualmente un Plan Nacional de Nuevas Tecnologías para liberarse de las aplicaciones privativas.

Brasil es uno de los países por excelencia en el uso del Software Libre, comenzando por iniciativas propias como la de Río Grande do Sul que aprobó la primera ley obligatoria sobre el uso de aplicaciones libres hasta un decreto federal en el 2003 enunciada por el presidente Lula Da Silva, durante este año y como resultado de migrar a OpenOffice.org el Congreso Nacional de Brasil ahorró 1.41 millones de euros lo que supone el 15% del presupuesto de tecnologías para ese período de tiempo.(Jesus Javier Estepa Nieto 2007)

Otro ejemplo claramente evidente de los procesos de migración lo constituye Venezuela cuyo presidente tras al paro petrolero iniciado en diciembre del 2002 que paralizó la producción de petrolero en el país y causó grandes daños en la economía que le costó grandes perdidas al país de recursos económicos por usar software privativo de Microsoft, decidió migrar a Software Libre con este fin se creó el decreto 3390 (Hugo Rafael Chávez 2004)que busca empujar a las instituciones gubernamentales del país a buscar alternativas libres migrando los sistemas, proyectos y servicios informáticos a esta nueva plataforma de desarrollo, instando a la Institución Pública Nacional a tomar cartas en el asunto (Venezolana de televisión 2007).

Además de los gobiernos y empresas el Software Libre tiene un gran auge en los entornos

universitarios donde se reúne la mayor fuerza intelectual e investigadora de la rama, no por gusto el núcleo de Linux nació en la universidad, la lista de universidades que usan aplicaciones libres es muy amplia encontrándose la mayoría en España, país que como se mostró posee gran desarrollo en esta rama entre ellas vemos la Universidad de Cádiz, Universidad del Rey Juan Carlos, Universidad de Murcia; en América también encontramos la Universidad de Misiones en Argentina y la de los Andes por solo citar algunos ejemplos.

En la actualidad existe una gran gama de aplicaciones libres que han hecho posible estos procesos de migración y que además garantizan que los mismos sean factibles, las mismas están disponibles tanto para clientes como para servidores, apache por ejemplo es el servidor web libre por excelencia y el más usado en internet (Ver figura 1.2).

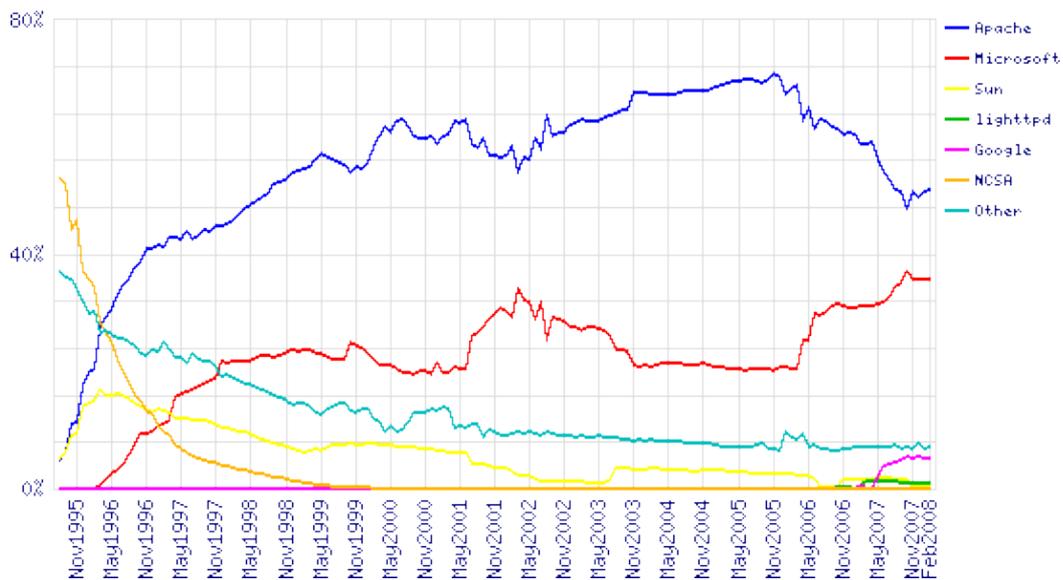


Figura 1.2 Uso de servidores web (Agosto 1995- Febrero 2008)

Otros productos libres de gran éxito en el campo de los supercomputadores que prestan servicios a internet son:

- ✓ Sendmail, sistema líder en institución de cuentas de correo al que a finales del 2001 se le atribuía la mayor cuota de mercado de todos los servidores de correo, seguido por Microsoft Exchange.

- ✓ MySQL, base de datos que ostentaba con el 20 % del mercado hasta finales del 2003.
- ✓ PHP, uno de los lenguajes de programación más utilizados para realizar páginas dinámicas en la construcción de aplicaciones web.
- ✓ Distribuciones GNU/LINUX, las grandes estrellas del movimiento de Software Libre. Se trata del sistema operativo que más rápidamente ha evolucionado y mejorado de la historia.
- ✓ Navegador Mozilla/ Firefox: Ha generado una gran expectación en el mercado, 30% de uso en Europa en solo 3 años.
- ✓ Open Office: Paquete ofimático de los sistemas operativos de Software Libre. Incluye múltiples mejoras y nuevas funcionalidades. Tiene la mejor integración de formatos de archivos de Microsoft.

No solo en el campo de los servidores es exitoso el Software Libre, aunque es válido decir que la mayor inserción del mismo está en esta área, para el uso de los clientes de escritorio existen innumerables aplicaciones (Anexo 1).

Tabla 1.3 Lista de herramientas equivalentes más comunes.

Aspectos a considerar	Software privativo	Software Libre
Navegadores	Internet Explorer, Netscape, Mozilla, Opera.	Mozilla, Galeón, Konqueror, Nautilus, Firefox.
Suite ofimática	MS Office, StarOffice, OpenOffice	StarOffice, OpenOffice, Koffice.
Hoja de cálculo	Excel, StarOffice, OpenOffice Cal	Gnumeric, OpenOffice Calc, StarOffice, Kspread
Gráficos y dibujos	Excel	Kivio, Día, KChart, Gnuplot

Aspectos a considerar	Software privativo	Software Libre
Creación de presentaciones	MS PowerPoint, StarOffice Presentation, OpenOffice Impress	StarOffice Presentation, OpenOffice Impress, Kpresenter, MagicPoint.
Base de Datos local	Access	KNoda, Gnome DB Manager, OpenOffice + MySQL.
Gestor de finanzas personales	MS Money, Quicken	GNUcash, GnoFin, Kmymoney, Grisbi.
Gestión de proyectos	MS project	Mr project, Planner, Dot Project, Gant Project, Track.
Cliente correo electrónico como MS Outlook	Outlook	Evolution, Kmail.
Base de datos	MS SQL, MySQL para Windows	PostgreSQL, MySQL, mSQL, SAP DB
Servidor Web	Internet Information Server, Apache para Windows, roxen	Apache

Existen actualmente otras intenciones y procesos de migración forjándose en las distintas regiones siendo este el único camino hacia un mundo mejor.

1.4. Significación del uso del Software Libre en Cuba.

Nuestro país a diferencia de otros países del mundo no posee un documento legal de como

debe hacerse el proceso de migración y que guíe el proceso como lo hace Venezuela con el decreto 3390 ó la municipalidad del rosario en Argentina (Pablo Javkin 2004) que establece una ordenanza que contiene las disposiciones de como debe hacerse el proceso, sin embargo desde el año 2005 y posteriormente después de la constitución del Grupo Técnico Nacional se emitió un documento que contiene los lineamientos (Grupo técnico nacional 2005) fundamentales de cómo debe hacerse el proceso de migración a Software Libre, en Cuba el proceso migratorio responde a una estrategia que se evidencia a través de cuatro puntos fundamentales para el país :

1.4.1. Político

El proceso de revolución informática en Cuba hacia las tecnologías soberanas posee una gran significación política, la mayoría de los computadores de la infraestructura cubana están instalados con el sistema operativo de Redmond y en la gran mayoría son copias ilegales al igual que el software que utilizamos lo que nos pone a disposición de la transnacional Microsoft productora de la mayoría de dichos productos y le da el derecho a esta empresa con el respaldo de la ley propiedad industrial por la cual Cuba se rige también de decir que deje de usar dicho sistema o pudiera desatar una campaña de descrédito contra la isla en la comunidad internacional dañando la imagen de Cuba acusándonos de piratería informática, frenando en el primer caso el proceso de informatización de la sociedad cubana que desarrolla la isla en el marco actual de la batalla de ideas.

En segunda instancia esta la filosofía de trabajo cooperativo con la que se desarrolla el Software Libre donde el producto final es propiedad social y no propiedad privada como suele ser el privativo, estando muy acorde con el justo sistema socialista que defendemos como dijera el apóstol “Con todos y para el bien de todos”.

1.4.2. Social

En el ámbito social es evidente que formar jóvenes bajo la filosofía socialista de compartir el conocimiento forjara mejores personas, el carácter cooperativista y colectivista que se deslumbra en comunidades es parte de la solución de la sociedad más justa que cada día construimos.

La mejora de la relaciones humanas es un factor presente en la comunidad, brindando la

posibilidad de crear un tejido tecnológico por todo el país y que todos puedan aportar y dar sus puntos de vista y contribuciones a los proyectos que se desarrollen.

1.4.3. Tecnológico

Cuba es un país que alcanzó su independencia en el año 1959 sin embargo tecnológicamente la mayor parte de la infraestructura esta atada a empresas internacionales productoras de la mayoría de las aplicaciones que aquí usamos un proceso de adquisición y reemplazo de Software Libre significaría eliminar dicha dependencia y convertirnos en un país soberano en el ámbito de las tecnologías y con las posibilidades ofrecidas en la libertades del software podremos con la fuerza de trabajo de la isla personalizar dichas aplicaciones a la medida brindando así mejores soluciones y servicios informáticos.

Un segundo punto esta vinculado con las bondades que ofrece el Software Libre referente a los pocos recursos que se necesitan para correr las aplicaciones lo que implicaría que se pueden usar los mismos recursos de hardware existentes e incluso darle mejor rendimiento a máquinas con menor recursos; ha esto va unido el gran soporte que ofrecen muchas distribuciones de GNU/Linux para las distintas arquitecturas existentes.

1.4.4. Económico.

En el plano económico no tendremos que comprar hardware nuevo ya que GNU/Linux es totalmente operable con el hardware que disponemos en la actualidad, los productos de software libre son en su mayoría gratuitos por lo que tendríamos que pagar muy poco por su adquisición, la mayoría de los mismos están disponibles en internet para descargar y usar libremente.

La concepción de lograr una Industria Cubana de Software rentable basada en servicios es una idea totalmente asequible, existen empresas en el mundo que viven de los servicios y ofrecen resultados halagadores como es el caso de IBM (Roberto Di Cosmo 2007) que más de la mitad de sus ganancias son por esta vía.

1.5. La universidad cubana.

Los orígenes de la Educación Superior Cubana se remontan al año 1728 cuando fue creada La Real y Pontificada Universidad de La Habana, 219 años después es creada la segunda

universidad cubana, la Universidad de Oriente y así sucesivamente se fueron creando los distintos centros, en el año 1959 con el triunfo de la Revolución Cubana la educación sufre profundos cambios en sus concepción, se crean becas en todo el país y se establece la educación gratuita para todos, dándole acceso a los estudios a las clases más humildes de la sociedad, los principales cambios estaban dirigidos a lograr una educación científica basado en un enfoque más experimental y romper con aquella educación arcaica que existía con conceptos académicos que la alejaban de la realidad objetiva de las necesidades de nuestro país.

La red de centros de Educación Superior continúa ampliándose por el país y en 1976 se crea el Ministerio de Educación Superior (MES), siendo su objetivo dirigir, organizar y ejecutar la política educacional a ese nivel del gobierno y del estado socialista.

El MES como organismo rector se interrelaciona con los distintos organismos que responden a las particularidades de desarrollo del país, por lo que los centros educativos están altamente comprometidos con la sociedad, Cuba ha sido capaz de desarrollar una impetuosa revolución educacional que comenzó en el año 1961 con la campaña de alfabetización hasta nuestro días con la inmensa batalla de idea que llevamos a cabo reflejada en la universalización de la enseñanza a través de las Sedes Universitarias Municipales (SUM).

La universidad actual es el fruto de profundas transformaciones cuyo objetivo es brindar equidad de oportunidades a todos los miembros de la sociedad y se hace notar no solo con las creación de las SUM sino que va más allá llegando a los bateyes, municipios, hospitales, prisiones, centrales azucareros brindando igual de acceso a todos.

Estas nuevas transformaciones en la concepción del modelo educacional ha dado lugar a la "Nueva Universidad", las SUM parecen prefigurarse como un actor que, dentro de un conjunto de actividades incluida la investigación, privilegie la transferencia de tecnologías y saberes, los evalúe, adapte y utilice eficientemente en función del desarrollo social. No está demás decir que esa actividad que se orienta a conectar conocimiento y necesidades sociales es una labor altamente creativa que solo pueden desarrollar personas con la debida formación como para actuar como agentes del desarrollo científico y tecnológico local. El desarrollo necesita mucho de ese tipo de actores.

Todo el desarrollo científico y tecnológico experimentado por la sociedad en el pasado y en el actual siglo, ha implicado igualmente nuevas demandas de formación de profesionales por las universidades. Por lo que le corresponde el papel privilegiado de ser la institución social que más integralmente puede hacerlo, no existe otra institución capaz de acometer tal empeño, la universidad en su nueva concepción tiene la tarea de preservar, desarrollar y promover a través de sus procesos y su estrecho vínculo con la sociedad la cultura de la humanidad.

Por tanto pueden identificarse tres procesos inherentes cuya integración permite dar respuesta a lo planteado anteriormente : Formación, Investigación y Extensión Universitaria, estos tres procesos están estrechamente ligados entre si y entre ellos aseguran el cumplimiento de los objetivos de la universidad actual.

La universidad actual es científica, tecnológica, y humanista, otras características son :

- ✓ Formación sobre la base de un amplio perfil con dos ideas rectoras principales.
 - Unidad entre la educación y la instrucción.
 - Vinculación del estudio con el trabajo.
- ✓ Amplia cobertura de las necesidades de educación de postgrado.
- ✓ Investigación e innovación tecnológica como elementos intrínsecos de todo el quehacer universitario.
- ✓ Plena integración con toda la sociedad.
- ✓ Presencia en todo el territorio nacional.

Las carreras universitarias en el país están divididas en nueve ramas (ONE 2007) agrupadas de la siguiente forma: ciencias técnicas, ciencias naturales y matemática, ciencias agropecuarias, ciencias económicas, ciencias sociales y humanísticas, ciencias médicas, pedagogía, cultura física y arte distribuidas en 65 instituciones de educación superior y 3150 sedes universitarias a lo largo de la isla, el ingreso inicial de este curso fue de 747554 estudiantes y cuenta con un total de 40865 profesores universitarios, 9042 a tiempo completo y 31 443 a tiempo parcial. (Juan Vela Valdés 2007)

La docencia esta distribuida en cuatro modalidades de cursos diferentes: El curso regular diurno con 48 carreras, el curso para trabajadores con 29, el de continuidad de estudios con 12, y la educación a distancia con tres, de forma general Cuba dedicó en el 2007 el 6% del presupuesto a la Educación Superior.(Juan Vela Valdés 2007)

El uso de software privativo para la enseñanza de algunas disciplinas en las carreras universitarias constituye un costoso elemento para la universidad de hoy en todo el mundo un ejemplo de ello lo constituye Adobe Photoshop CS3, una potente herramienta para artistas gráficos, cuyo costo asciende a los 835 dólares (Adobe 2007), y es ampliamente usada en las carreras de diseño gráfico e industrial.

El uso en Cuba de estas herramientas no constituye de forma económica una barrera para su uso, debido al bloqueo económico impuesto a la isla por parte de los EE.UU. estas aplicaciones en su mayoría no son pagadas, existe otras herramientas que son libres y no tienen para el país implicaciones legales que se están usando a lo largo del país y constituyen un estandarte en la formación del hombre nuevo forjado bajo concepciones socialistas con herramientas libres, la más importante y que constituye una piedra angular en la formación de los universitarios en el nuevo modelo de enseñanza a distancia lo constituye moodle, una herramienta para la gestión de cursos de libre distribución que ayuda a educadores a crear comunidades de aprendizaje en línea, es, de las herramientas de su tipo la más usada a nivel mundial compitiendo en el mercado con herramientas privativas.

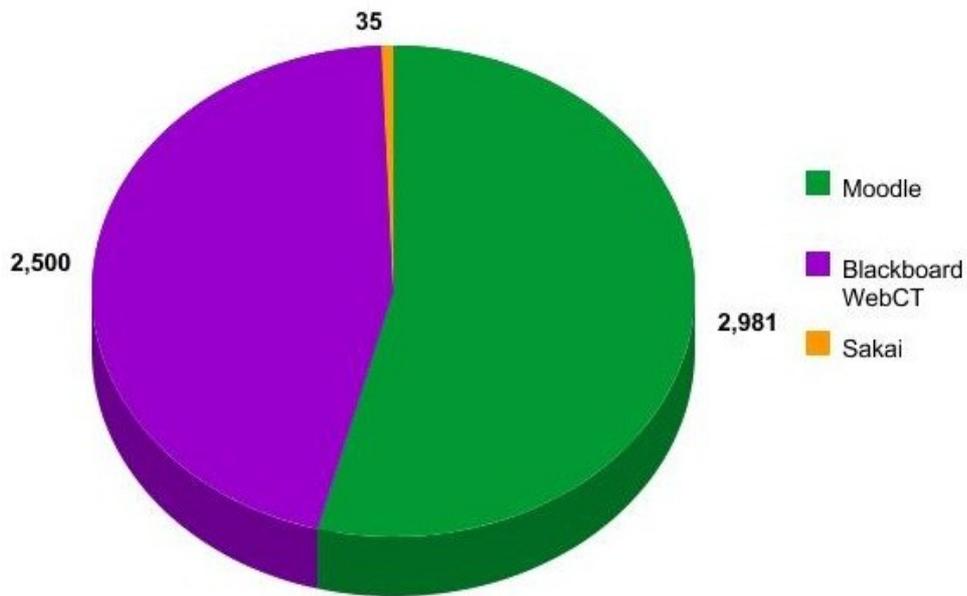


Figura 1.3. Penetración en el mercado de herramientas LMS.

En Cuba la teleformación y la enseñanza asistida por computadora está presente a distintos niveles, en la universalización está vigente con cursos de formación a distancia, ejemplo de ello es el Portal de Moodle brindado por la red cubana de salud Infomed (Infomed 2008), la plataforma de enseñanza a distancia de la Universidad de Holguín (UCO 2008), y el portal de teleformación de la UCI (UCI 2008).

1.6. Conclusiones

En la universalización está la mayor interacción con la sociedad y es donde se concentra la mayor fuerza investigadora del país, el uso de tecnologías que promuevan el trabajo en equipo y la colaboración aumentará en poco tiempo el nivel de conocimientos del país, la cultura general integral de los ciudadanos por la que apostamos todos estará más cerca de la realidad en un menor tiempo y será más sólida.

2. CAPÍTULO II: ANALISIS DE LAS PROPUESTAS

2.1. Introducción

El éxito de un proceso de migración a Software Libre depende en gran medida del conjunto de pasos que usen para acometerla, siempre lo más recomendable es la redacción de un documento que contenga dichos elementos el cual definiremos como: Guía de Migración a Software Libre, dicho documento establece las etapas de desarrollo y las tareas de cada una de ellas, los recursos que deberán ser asignados y el tiempo a emplear en su desempeño, garantizando un flujo de trabajo constante y eficiente, debe describir en lenguaje técnico y claro cómo se debería llevar a cabo dicha migración.

A continuación se muestran algunos resultados de un estudio realizado a algunas guías de migración existentes y elegidas por su similitud con el objetivo de este trabajo, resulta interesante antes de comenzar mencionar la similitud existente entre algunos de estos estudios, por ejemplo la “Guía Livre” brasileña (Equipo técnico de migración 2003) es muy similar a la realizada por los europeos, esto demuestra que este tipo de procesos no es nuevo en el mundo por lo que es importante la realización de un análisis preliminar. Con el objetivo de realizar este tipo de estudio se tomaron como objetos de análisis la Guía de Migración de la Unión Europea, la propuesta de la Universidad de los Andes y la metodología realizada en el año 2007 para la migración a Software Libre de la Universidad de las Ciencias Informáticas.

2.2. Elementos de un ejercicio de migración.

En el año 2003 la Comisión de la Unión Europea publicó el documento “The IDA Open Source Migration Guidelines” (netProject Ltd and European Communities 2003), el cual define un conjunto de elementos que deben estar incluidos en cualquier ejercicio de migración que se lleve a cabo, estas directrices se han validado en el proceso de migración a OSS realizado por el Tribunal de Cuentas de Schwerin en el lander alemán de Mecklenberg Vorpommern. dichos elementos se han constituido por lo tanto como una especie de estándar a la hora de evaluar y crear guías de migración ideales asociadas a estos procesos.

1. Crear un equipo con la capacitación y el respaldo de gestión adecuados. Es importante que se disponga de apoyo de gestión pues de lo contrario habrá resistencia a un cambio de la norma de sistemas privativos. Este apoyo habrá de ser suficiente para permitir por lo menos la construcción de pilotos representativos, así habrá que elaborar un caso de negocio básico, y quizás uno más detallado después cuando se disponga de más datos.
2. Entender el entorno final, tanto el software FLOSS (Free/Libre Open Source Software) como la arquitectura básica, junto con las diferentes opciones y posibilidades disponibles. Esto significa que hay que formar al personal, contratar personal o recurrir a consultores. Esto implica algunos costes iniciales y por ello es necesario disponer de respaldo de los responsables de la gestión.
3. La migración es una oportunidad de revisar la arquitectura de base así como el software de aplicaciones. Puede haber ciertos costes al hacer el cambio, y hay que tenerlos en cuenta.
4. Es muy importante entender bien en qué consiste el FLOSS. Hay algunos aspectos que hay que tener en cuenta antes de tomar ninguna decisión.
 - Hay que tener claro cuáles son las implicaciones de las licencias para FLOSS especialmente si se considera que la administración va a distribuir los cambios de software.
 - Cuando hay varias opciones para una función (por ejemplo, hay por lo menos tres buenas hojas de cálculo de FLOSS) los administradores han de entender las ventajas y las desventajas de cada producto.
 - Se deben tener en cuenta las diferencias entre las distintas distribuciones. Algunas distribuciones están respaldadas por empresas comerciales que prestan su apoyo y correcciones. Algunas tienen características diferentes: Gentoo, por ejemplo, da una distribución basada en un código fuente que facilita a la administración una adaptación del software para que satisfaga sus necesidades concretas. Todas estas diferencias han de ser valoradas antes de hacer cualquier elección.

- Los administradores deben determinar qué nivel de apoyo es necesario. Se puede conseguir apoyo comercial de los creadores de la aplicación o la distribución si la suministran. Si no es así, hay terceros que pueden prestar ese apoyo ya que se dispone del código fuente y hay muchas empresas que dan ese apoyo.
- Esta es una diferencia clara respecto al mercado de software privativo donde un apoyo detallado sólo lo facilitan las empresas que tienen el privilegio de acceder al código fuente. Y esto es importante, si el vendedor propietario abandona el negocio sin revelar el código fuente.

Y si todo falla, la mayoría de las aplicaciones tienen listas de correo activas donde una pregunta o petición de ayuda recibirá la respuesta de alguien relacionado con la aplicación. La presencia de una lista de correo activa y una comunidad de usuarios suele ser uno de los criterios a tener en cuenta en primer lugar en la elección de los componentes de software.

5. Estudiar los sistemas existentes. Estos datos no solo serán necesarios para hacer la migración en sí, sino que muchos de ellos serán también muy necesarios para construir un modelo de coste total de propiedad para un caso concreto de negocio. Hay que compilar los inventarios de lo siguiente:

- ✓ Para cada aplicación usada:
 - El nombre de la aplicación, el número de la versión y el punto de contacto para obtener respuesta a cualquier consulta.
 - Cuántos usuarios necesitan acceder a la misma.
 - Qué sistema operativo se está usando. Con qué sistemas operativos se puede ejecutar la aplicación.
 - Qué otras aplicaciones necesitan el cliente y el servidor para que la aplicación funcione.
 - Qué hardware se necesita. En particular si se necesita algún hardware de perfil especial o no estándar.

- Qué protocolo utiliza para comunicarse con otras aplicaciones.
 - Qué formatos de archivos necesita.
 - Qué internacionalización y localización se necesita. Se pueden necesitar múltiples idiomas y monedas.
- ✓ Requisitos en cuanto a los datos. Esto habría de interpretarse en el sentido amplio que contempla, por ejemplo, el proceso de textos y hojas de cálculo, voz y audio e imagen así como bases de datos normales. En general, todo lo que un ordenador ha de procesar.
- ¿Cuáles son las condiciones de interfaz con otros sistemas o usuarios que no controle la administración?
 - ¿Qué requisitos hay para conservar los datos y procesarlos en el futuro? ¿Hay un depósito de datos heredados existentes que haya que respaldar? Si es así, ¿son necesarias aplicaciones especiales para ese respaldo?
 - i. Dividir los datos en las siguientes categorías:
 - ii. Datos que no es necesario conservar y se pueden eliminar. Eliminarlos.
 - iii. Datos que deben conservarse y están actualmente en formato abierto como PDF o Postscript, o que pueden traducirse fácilmente a uno de ellos. Se debe estudiar con cuidado el coste de esa traducción.
 - iv. Datos que deben conservarse pero que están en un formato cerrado propietario que no puede traducirse fácilmente a otro abierto. Estos datos pueden requerir la conservación de copias de la aplicación propietaria concreta. Se debe estudiar el coste de estas aplicaciones. El número necesario de copias de la aplicación se puede determinar según el grado de acceso a los mismos que se requiere. Por ejemplo, si se accede poco a los datos entonces bastará con una copia en una máquina central. También puede ser necesario conservar un hardware específico para ejecutar esas aplicaciones.

- ✓ Requisitos de seguridad:
 - ¿Cuál es el sistema actual para asignar los nombres de usuarios y las contraseñas? ¿Tienen los nombres de usuario una estructura y si es así cuál? ¿Cuál es la política para actualizar las contraseñas?
 - ¿Hay sistemas que requieran otra autenticación más allá que un simple nombre de usuario y una contraseña?
 - ¿Qué políticas tiene la Administración en relación con el uso de los ordenadores? Por ejemplo ¿hay restricciones sobre el uso de Internet y el correo electrónico?
 - ¿Hay disposiciones de seguridad que exijan el uso de un hardware o software concretos?
- 6. Elaborar un caso detallado de migración, que se basará en los datos recogidos según lo dicho anteriormente y que consistirá en los siguientes puntos:
 - ✓ El coste del entorno existente en un período de tiempo razonable digamos de cinco años con supuestos adecuados a la administración.
 - ✓ El coste de entornos alternativos y el coste de la migración a cada uno de ellos en el mismo período.
 - ✓ Los puntos fuertes y débiles del entorno actual y las distintas alternativas.

La correspondiente hoja de cálculo será de ayuda para comparar costes.

7. Consultar a los usuarios. Explicar las razones que hay detrás de la migración y cómo les afectará. Estudie sus preocupaciones con seriedad y permítales que practiquen con la tecnología sin pérdida de tiempo. Cuanto antes se impliquen los usuarios mejor será. Esto puede ser un requisito legal en algunos países pero habría que hacerlo en todo caso para facilitar la introducción de lo que puede ser un cambio significativo en las prácticas de trabajo.

Crear una ventanilla de atención al cliente que pueda dar respuesta a las preocupaciones de los usuarios. Más adelante, cuando la migración esté configurada, podrá resolver los

problemas y convertirse en un centro de excelencia y buenas prácticas. Crear un sitio de intranet con una sección dedicada a “consejos y cómo se hace” que los propios usuarios puedan actualizar. Es importante que los usuarios sientan que forman parte y este sitio a su vez puede proporcionarle a la ventanilla de atención una idea del tipo de problemas a los que se enfrentan los usuarios.

8. Suponiendo que el caso de ejemplo se ha hecho, comenzar con proyectos piloto a pequeña escala, de preferencia en un entorno auto-contenido con pocos usuarios. Esto facilitará, entre otras cosas:
 - ✓ Datos más ajustados de modelos de coste total de propiedad.
 - ✓ La reacción de los usuarios, que se puede emplear para facilitar la introducción a otros sistemas.
 - ✓ La validación o modificación de la arquitectura final y el caso de ejemplo.
9. Decidir sobre la velocidad del proceso de migración una vez iniciado. Estas son las principales alternativas:
 - ✓ Big bang: Todos los usuarios cambian del viejo sistema al nuevo el mismo día. En la práctica, esto significa programar el cambio en un fin de semana o fiesta nacional. La ventaja es que no se necesitan disposiciones de doble acceso y el personal no se va a encontrar yendo y viniendo de un sistema a otro. Entre las desventajas está el alto riesgo y la gran exigencia de recursos durante el cambio. Este esquema de migración sólo cabe en el caso de las pequeñas administraciones.

En todo caso, si es posible hay que evitar las migraciones de tipo Big Bang. Las migraciones “big bang” tendrán tantas variantes que controlar que casi siempre fallan. Y si lo hacen no parece probable que sea por un fallo del OSS sino de gestión.

- ✓ Transición por fases en grupos: Se pasa a los usuarios del antiguo sistema al nuevo en grupos. Puede que los grupos funcionales completos se trasladen juntos para minimizar tener que compartir datos y los problemas de trabajo en el grupo. Se pueden contener los riesgos y gestionar los recursos eligiendo grupos del tamaño adecuado. También es posible hacer un cambio del hardware de los

PC al mismo tiempo, reemplazando las máquinas en un grupo y luego instalando las sustituidas en lugar de las viejas máquinas del siguiente grupo.

- ✓ Transición de usuario a usuario: Básicamente la misma opción de la transición en grupos, pero con un grupo compuesto por una sola persona. Ese método de “goteo” tiene escasos requisitos en cuanto a los recursos, pero no resulta eficaz ni apropiado para grandes administraciones. Pero sí puede ser una buena manera de ejecutar los proyectos piloto.

Es probable que tanto los viejos como los nuevos sistemas tengan que funcionar “codo con codo” durante cierto tiempo. Es importante contar con una estrategia de transición que permita que ambos sistemas funcionen juntos, de manera que las actividades de producción se puedan continuar correctamente durante el período de transición. La sustitución de la máquina vieja puede llevar bastante tiempo (o no tener lugar), por lo que la coexistencia puede ser muy importante.

10. Extender la migración a toda la administración. Esto implicará más formación de los usuarios y del personal técnico.
11. Supervisar la respuesta de los usuarios y tomar nota de los problemas que surjan. Algunas necesidades de los usuarios pueden ser tan poco claras que no se pueden prever ni descubrir durante los proyectos piloto. Hay que asegurarse de que se dispone de recursos suficientes para hacer frente a esas necesidades tras la transición.

2.3. Análisis de las directrices IDA de la Unión Europea.

Los europeos (netProject Ltd and European Communities 2003) estiman que cualquier ejercicio de migración debe reflejar lo siguiente:

1. Una fase de definición del proyecto y de recopilación de datos, en la que se contemplen:
 - A. La descripción del conjunto de condiciones iniciales relevantes consistentes, por ejemplo, en:
 - a) Arquitectura o arquitecturas de los sistemas.

- b) Aplicaciones y sus datos asociados.
 - c) Protocolos y normas empleados.
 - d) Hardware.
 - e) El entorno físico, como el ancho de banda de la red, la ubicación.
 - f) 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.
2. Una justificación de la migración, incluido el coste asociado a la misma.
 3. Una o más fases piloto preparadas para probar si el plan y la justificación funcionan. Los datos de estas fases piloto pueden luego alimentar el modelo de costes usado en el plan.
 4. Despliegue del plan.
 5. 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 el cumplimiento del mismo, abogan además acerca de lo gran general que es la guía y 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 dejan escapar la posibilidad de que el objetivo final de la migración que es obtener un entorno FLOSS completamente no se cumpla de manera total y queden al final del mismo algunas máquinas con Windows por motivos específicos, lo que ellos denominan una migración parcial (netProject Ltd and European Communities 2003).

Por otra parte los europeos dividen las directrices para la migración en dos grandes grupos, las directrices de gestión y las directrices técnicas, las primeras encargadas de

forma general de regular todo el proceso del control y planificación de los recursos tanto humanos como materiales y la confección de una guía de migración, mientras que la segunda se encarga de los aspectos técnicos del proceso tales como el estudio y selección de aplicaciones y los pasos necesarios para la implantación de las mismas, la arquitectura de referencia a seguir, etc.

Para lograr el despliegue del plan luego de cumplir los aspectos iniciales (1,2,3) expresados al inicio de esta sección los europeos definen una sección en el capítulo 7: “Una vida más fácil” denominada “Introducir nuevas aplicaciones en un entorno familiar” 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 el entorno libre, utilizar este enfoque y posibilitar la existencia de herramientas libres y privativas ayudará a la conversión de los formatos de archivos 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 por solo citar algunos ejemplos.

A la vez afirman que los primeros cambios deberán ser en el servidor siendo estos transparentes para los usuarios y posteriormente en los clientes, para poder hacer efectivo el proceso y 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 portables para la creación de aplicaciones como es el caso de ANSI, JAVA o Python al igual que el empleo de aplicaciones de construcción de GUI como wxWindows y Fox-toolkit por solo citar algunos ejemplos, la introducción y masivización del uso de formatos de fuente abierta y/o portable como PDF es una buena práctica y reduce bastante también los riesgos y la resistencia de los usuarios en el proceso de migración hacia FLOSS.

Para el proceso de migración de los servidores pueden ser diversas las condiciones iniciales las cuales abarcan el modelo de grupo de trabajo, dominio NT de Windows y Dominio de Active Directory cada una de ellos posee características específicas a la hora de la migración.

De forma general dicha guía propone dos rutas (netProject Ltd and European Communities 2003) de migración :

Ruta 1 de migración :

Añadir máquinas FLOSS a los dominios existentes e ir trasladando gradualmente los datos y los usuarios, y luego eliminar a los antiguos servidores privados. Es posible transferir a clientes y servidores independientemente.

Añadir servidores al dominio Windows es uno de los modos más rápidos de sacar provecho del FLOSS. Por ejemplo, la combinación del sistema operativo GNU/Linux con Samba da un servidor de impresión/archivos potente y de bajo coste que se puede usar en lugar de un sistema Windows sin gastos para el entorno de cliente existente.

Ejecutar clientes FLOSS en un dominio Windows es una forma de coexistencia de escaso riesgo, ya que no es necesario cambio alguno en relación a los servidores. Se puede usar donde un pequeño número de personas van a usar escritorios FLOSS en un entorno sólo de Windows.

Ruta 2 de migración :

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

Es mucho más sencillo que ejecutar un sistema mixto FLOSS/Windows, pero crea una cooperación entre la gente que usa Windows y la que usa sistemas FLOSS más difícil.

Ambas rutas las podemos resumir en los siguientes diagramas:

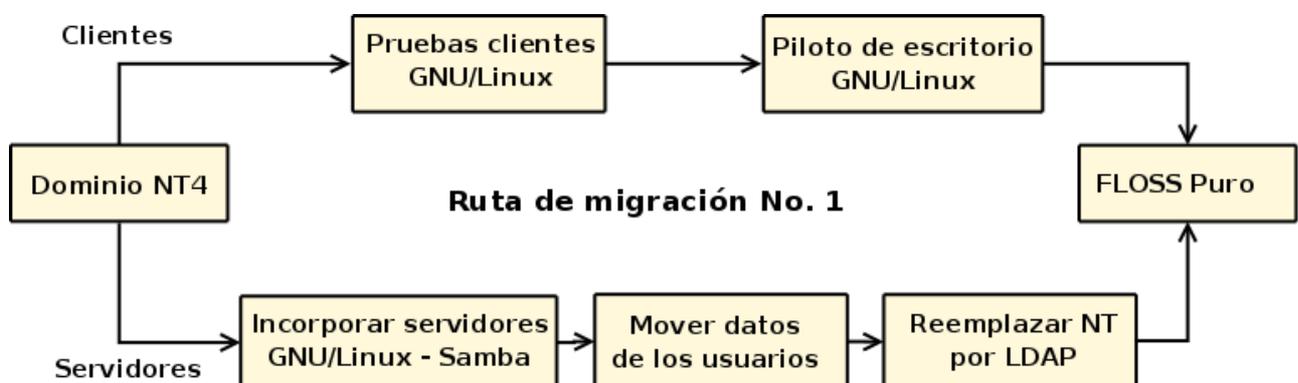


Figura 2.1 Ruta de migración No.1

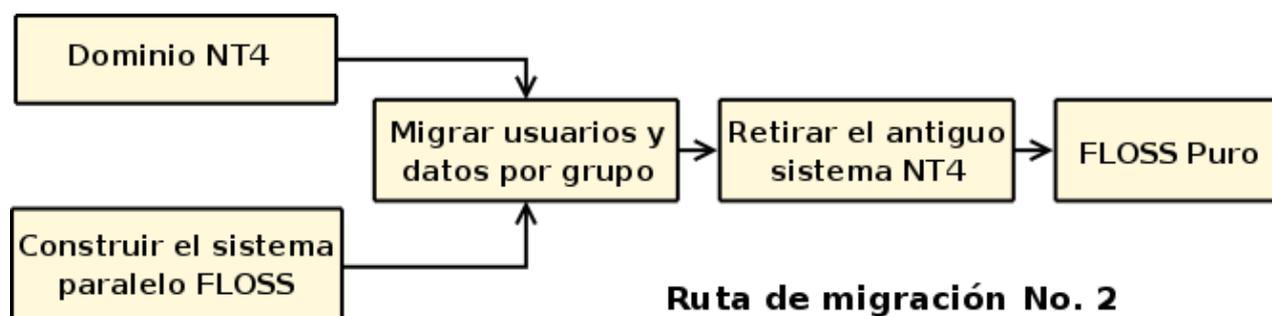


Figura 2.2 Ruta de migración No.2

Para el proceso de migración de las estaciones de trabajo a FLOSS a realizarse posterior a los servidores los europeos les han dedicado un amplio espacio en su guía estudiando cada herramienta privativa que existe y buscando su equivalente libre, recomendando estas últimas de acuerdo a sus funcionalidades, para lo cual han dividido las mismas en dos grandes grupos que han denominado “Grupos funcionales”, el primero llamado grupos principales que representa lo descrito en los procesos del negocio y el segundo denominado grupos secundarios encargados de servir de apoyo a los anteriores; ofimática, correo, calendario y herramientas colaborativas, son algunos ejemplos de principales a la vez que sistemas operativos, servidores de archivos, detección de virus y spam son ejemplos de las segundas.

Los europeos además de proponer la metodología proponen paso por paso los elementos a tener en cuenta para llegar de las condiciones iniciales a la finales en el proceso sin dejar escapar detalle alguna en una amplia guía de migración que contiene 156 hojas, no obstante, una vez analizado dicho proceso se puede deducir que una vez realizado el despliegue como bien lo han denotado los europeos la guía no refleja un procedimiento sólido para la fase de seguimiento del plan (punto 5), por lo se le podrían incorporar nuevos elementos al proceso propuesto.

2.4. Análisis del Plan de Migración de la Universidad de los Andes.

En diciembre del 2004 el gobierno bolivariano de Venezuela, hizo público el decreto 3390 (Hugo Rafael Chávez 2004) en el que el presidente Hugo Rafael Chávez Frías enuncia la necesidad de usar Software Libre a todo el país haciendo un llamado a migrar los organismos de la Administración Pública Nacional (APN); la Universidad de los Andes (ULA), lanzó en mayo del 2006 la versión 1.0 de su estrategia para migrar bajo el título "Plan Institucional de Liberación y Migración a Plataformas de Software Libre" (ULA 2006) en el cual definen un conjunto de elementos sobre la realización de dicho proceso.

EL Consejo de Computación Académica (CCA) de la ULA realizador de dicha estrategia realizó un estudio de los principales procesos de la universidad, haciendo un levantamiento de información que abarca la información general del organismo y la estructura de la misma, definiendo los procesos críticos a enfrentar durante la migración, identificando como críticos aquellos de los que depende la vida universitaria del centro como pueden ser correo electrónico, conectividad (routing y dns), portales y repositorios, servicios bibliotecarios entre otros.

Para la etapa de implementación se definen de forma general 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 y el segundo relacionado con temas de carácter técnico y orientado a programadores.

Según la guía de la ULA (ULA 2006) algunos aspectos organizativos a tener en cuenta en la migración son:

- 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 de productividad.
- 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 cuales 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 institucionales 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:

- 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 implementarán los mecanismo de soporte de acuerdo a los recursos financieros disponibles, entre otros.

Como un último aspecto el CCA recomienda la implantación de proyectos pilotos (ULA 2006) en determinadas facultades y unidades administrativas con el objetivo de implementar el plan y ajustar determinados elementos del mismo de acuerdo a los resultados obtenidos.

Además de los aspectos anteriormente enunciados los venezolanos propusieron un conjunto de etapas a tener en cuenta en la migración tecnológica de la ULA:

Etapas 1: La etapa de diagnóstico consiste en identificar y cuantificar (inventario) en las distintas unidades administrativas de la ULA, tanto del área académica como administrativa, 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 cuales no.

Etapas 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 en los años 2006 al 2008.

Adicionalmente se debe 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.

Etapas 3: Esta etapa consiste en 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 ULA. Para ello se necesita: determinar y establecer prioridades de los usuarios que participaran 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. Así como también lograr la consolidación, por entes institucionales, sobre las políticas comunicacionales de Software Libre en la ULA.

Etapas 4: Esta etapa consiste en definir 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 ULA. Así como también, definir las estrategias para consolidar los mecanismos de atención y soporte de los usuarios, los cuales fueron mencionados en la sección anterior; ya que estos mecanismos implican costos para la

organización. En el proceso de distribución, instalación y soporte de las aplicaciones de Software Libre, es necesario determinar los recursos (RRHH, CD de distribución, etc.) que se requieren en cada fase del plan de migración, pues en la medida que se incrementa el soporte de los usuarios estos recursos podrían ser mayores.

Etapa 5: Esta etapa consiste en definir las estrategias que permitan la migración de los sistemas informáticos administrativos de la ULA, tanto del área académica como administrativa. Para ello, los responsables y desarrolladores de dichos sistemas deben presentar un plan de migración, durante los años 2006 y 2008. 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 aquellas aplicaciones de Software Libre que podrían ser utilizadas en el desarrollo de los sistemas informáticos o actualizaciones parciales de los mismos (si ya existen sistemas desarrollados bajo plataformas de Software Libre) dependiendo de las características de cada sistema.

Etapa 6: Esta etapa consiste en definir las estrategias para que en cada fase del proceso de migración se garantice 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.

Como bien se muestra la estrategia descrita por la ULA abarca desde el inicio con un levantamiento de todos los recursos disponibles hasta el fin del proceso de migración asegurando la continuidad del soporte al proceso, resulta sorprendente sin embargo la importancia que se le da al proceso de capacitación en el plan, como bien especifican en el anexo A (ULA 2006) la capacitación a los usuarios se brinda por los niveles de impacto que tengan estos, estableciendo planes de formación para los distintos niveles de usuarios involucrando a todos los miembros de la ULA en el proceso e identifican tres posibles niveles de planes de formación : para instructores en software libre, la formación de soporte técnico y para los usuarios finales , lo cual da una medida de la importancia de la

capacitación en el proceso de migración.

Los mecanismos de soporte son otro elemento de gran importancia para el proceso de transición de un sistema a otro , la ULA en tal sentido propone la consolidación de comunidades virtuales para el soporte a las herramientas libres que se introducen en el proceso, creación de manuales, materiales de apoyo y guías prácticas disponibles en medios digitales de fácil acceso, consolidación de portales que permitan el acceso a los recursos, fortalecimiento de las unidades de soporte técnico para garantizar las instalación y soporte de aplicaciones, centros de atención a los usuarios entre otros mecanismos para garantizar un soporte adecuado y evitar fallas del proceso en ese sentido

2.5. Análisis de la Metodología de Migración de la Universidad de las Ciencias Informáticas.

Nuestro país como parte de la política de informatización de la sociedad y por las ventajas que brinda el software libre expresadas en el epigrafe 1.4 puso en marcha en el año 2002 una estrategia para alcanzar la soberanía nacional de la isla, en la que la Universidad de las Ciencias Informáticas (UCI) formaba parte del núcleo de la misma consolidándose en la facultad 10 el GTN, encargado de las soluciones relacionadas con los aspectos técnicos del proceso, como parte de la migración de la propia universidad en el año 2006-2007 se realizó la Metodología para la Migración a Software Libre del UCI como 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 (Ramón Paumier Samón 2007):

- **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 descrita anteriormente tiene un grupo tareas para dar cumplimiento a los objetivos trazados, según la guía en estudio se propone como primera tarea de la etapa de

preparación una justificación del proceso donde se presentaran los argumentos del porque migrar, expresando las ventajas del cambio, ejemplos comparativos de los costos de utilización de las herramientas, la gran variedad de herramientas existentes libres y la calidad de las mismas con respecto a las privativas.

Como un segundo elemento y el más importante está la 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, sensibilización de la institución y preparación de la misma para acometer el proceso de cambio, formación y capacitación del personal en la mayor cantidad posible para asegurar el soporte técnico mientras dure y después del proceso de migración y 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 realización 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 como 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 estas otro gran número de subtareas :

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 sometidos 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 cual 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 pilotos y la constante retroalimentación, que será el factor determinante en la introducción de nuevos servicios.

2) Y 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 que tendrán a su cargo otras tareas no menos importantes dentro del proceso de migración y la creación de nuevos puestos de trabajos, entiéndase la migración de nuevas

estaciones clientes, con su afinada y validada configuración y 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 las 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 on-line, 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 es la única guía de migración a Software Libre realizada en nuestro país hasta la fecha y se encuentra adaptada a las condiciones de una universidad cubana.

2.6. Conclusiones

Luego del estudio anterior sobre las guías mencionadas es válido destacar que todas de una forma u otra describen los pasos para realizar una migración a Software Libre, algunas de una forma más implícita y otras lo dejan claramente enunciando las etapas del proceso, los métodos utilizados resultan interesantes y novedosos en algunos casos lo que sin duda alguna tributa a la realización en el próximo capítulo de una metodología de migración más acabada para las universidades adjuntas al MES.

Es importante destacar algunos elementos que se es del criterio pueden incorporarse en algunos de estos documentos para mejorarlos, las directrices IDA por ejemplo no definen un conjunto de pasos sólidos para el proceso de post-migración, aunque si es válido decir que establecen algunas pautas para evitar rechazo en esta etapa, por otra parte los venezolanos que definen una guía muy completa para la migración de la ULA no definen a ciencia cierta como va a realizarse la transferencia de un sistema a otro, o sea desde el punto de vista del cambio del sistema operativo y de las aplicaciones propias del sistema no esclarecen como va a realizarse, finalmente la guía propuesta para la UCI se le puede señalar que no contempla en su fase de preparación un levantamiento del personal lo cual resulta importante para la creación de los planes de formación mas adecuados y tampoco tiene en cuenta el uso en fases iniciales de la migración de Windows con herramientas libres de cuya implantación dependerá el nivel de aceptación de la futura transición a los sistemas completamente libres.

3. CAPÍTULO III: METODOLOGIA DE MIGRACION A SOFTWARE LIBRE PARA LAS UNIVERSIDADES DEL MES

3.1. Introducción

El éxito de todo proceso en la vida cotidiana está regido por el orden de como se desarrolle el mismo, una migración a Software Libre no es solo cambiar un sistema operativo por otro o eliminar y colocar un conjunto de nuevas aplicaciones, va mas allá de eso, los cambios mas grandes están en la forma de pensar de las personas, la migración es un proceso ordenado donde las personas están convencidas de las ventajas de implantación de las nuevas tecnologías y apoyan dicho proceso para el desarrollo exitoso del mismo.

Luego del análisis de varios procesos o intentos de migración estudiados en el capítulo anterior y como garantía de la realización de una migración exitosa para las universidades del MES, a continuación se muestra una metodología para la Migración a Software Libre de dichos centros, la misma está organizada por etapas definiendo un conjunto de flujos de trabajo constantes que tienen distintos grados de impacto en cada etapa del proceso.

3.2. Metodología propuesta. Generalidades.

Para lograr una correcta Migración a Software Libre como habíamos mencionado anteriormente se definieron seis flujos de trabajo cuyos nombres y características generales son :

Evaluación : Hacer una evaluación de todos los procesos, tecnología y personal y adaptarlas al entorno actual.

Diseño : Diseñar un plan de migración conforme a las necesidades, tomando como partida el resultado anterior.

Pilotos : Poner en marcha el plan en un ambiente real de pruebas.

Formación : Formación del personal y certificación del mismo por niveles de usuarios.

Implementación : Instalación y migración definitiva de servicios y estaciones de trabajo a Software Libre.

Asistencia y soporte técnico : Brindará atención y soporte a las infraestructuras ,

servicios instalados y al personal.

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 :

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.

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.

De forma general, la estrategia propuesta se resume en el siguiente gráfico :

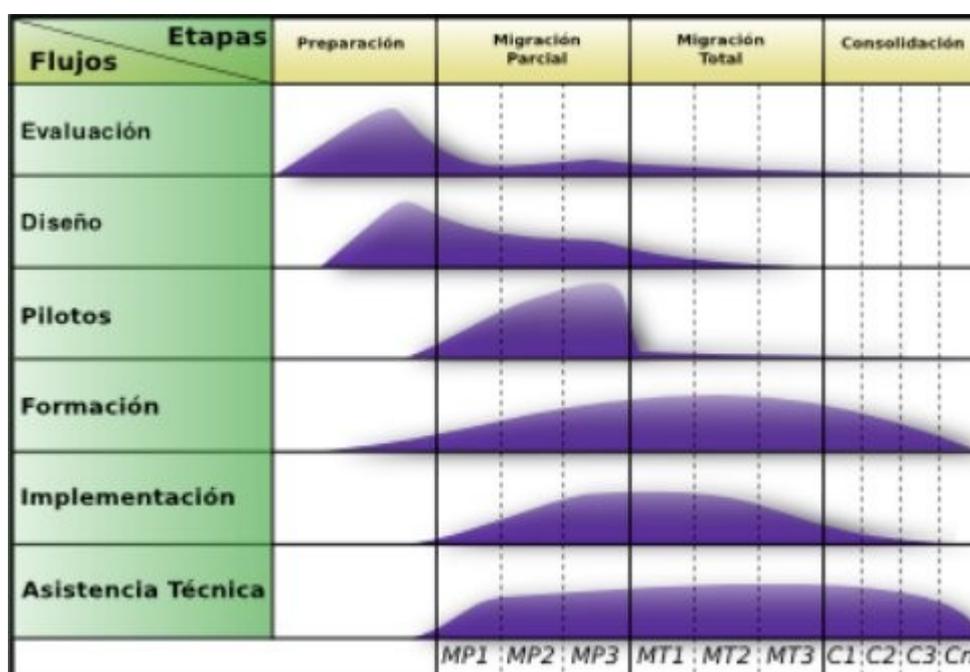


Figura 3.1 Metodología de migración.

La metodología propone para las etapas de Migración Parcial y Migración Total tres

iteraciones en cada una de ellas, la primera de estas orientadas a la implantación de Software Libre en los servidores de la institución, es importante esto como primer paso pues de esta forma contribuimos a disminuir una de las grandes barreras de un proceso de migración “Miedo a perder el poder” enunciadas en las Directrices IDA de la UE (netProject Ltd and European Communities 2003) analizada en el capítulo anterior, lo más importante en el proceso es lograr que las personas se sientan bien en el nuevo ambiente y apoyen en el mismo por lo que lograr que los administradores de red sean los primeros en tener los conocimientos necesarios al respecto será un punto a favor del futuro.

Luego de tener la infraestructura de servidores completamente en FLOSS como próxima iteración se propone el uso de herramientas libres en el sistema actual (Windows), esto proporcionará que los usuarios puedan probar las herramientas con las que trabajarán en GNU/Linux en un entorno ya conocido teniendo la posibilidad de capacitarlos en estas herramientas sobre este entorno y de ir facilitando la conversión de archivos a formatos estándares, para evitar problemas de compatibilidad entre las aplicaciones. Existe una lista bastante amplia de aplicaciones libres para Windows que pueden sustituir a todas las aplicaciones que comúnmente usamos (Anexo 2).

Una vez que estén los usuarios ya preparados en las herramientas libres que usarán sobre GNU/Linux y convencidos de las ventajas de utilizar FLOSS se propone el cambio de la base y la instalación de GNU/Linux como nuevo sistema operativo a usar lo que supone gran atención por todos los implicados en la tarea, pues es aquí donde más “frágil” resulta la migración.

Complementariamente a las dos primeras iteraciones se puede utilizar virtualización para algunas herramientas con el objetivo de hacer pruebas, es importante no dejar escapar que la migración tal vez no se realice de forma total debido a la existencia de aplicaciones que no posean equivalente libre, en tal caso se pueden utilizar aplicaciones como Wine que permiten emular software de Windows sobre GNU/Linux, siempre es importante probar la aplicaciones para verificar el rendimiento de las mismas una vez emuladas; también para tal caso es posible utilizar varias herramientas libres que en su conjunto realicen la tarea de una sola herramienta privativa o realizar la reprogramación de la misma.

3.3. Flujo de trabajo : Evaluación

La evaluación es el primer flujo de trabajo durante la migración, es donde se realiza una valoración de todos los procesos y tecnología presentes, es de suma importancia pues en el se definen elementos que serán la base en la futura migración, este flujo constituye un hito fundamental en el proceso y se realiza una sola vez, su mayor impacto tiene lugar en la etapa de preparación (Ver figura 3.2) donde se destina un mayor esfuerzo a la misma - punto [1] - al pasar este momento el esfuerzo destinado a esta tarea irá disminuyendo gradualmente y estará presente de forma muy leve en las restantes etapas pronunciándose un poco más en la iteración migración parcial 3 (MP3) debido a la gran actividad que puede haber presente en esta por las pruebas de las herramientas de escritorio y la gran variedad de las mismas lo que ocasionará la evaluación de nuevas herramientas.

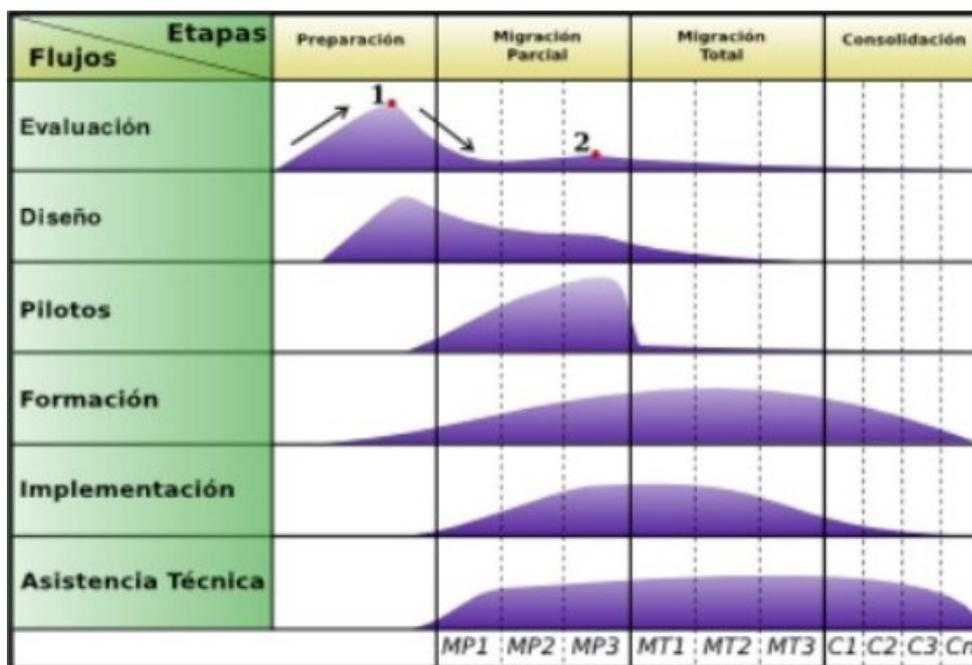


Figura 3.2 Flujo de trabajo de evaluación.

Para este flujo se definen un conjunto importante de tareas :

- ✓ Evaluar el tipo de usuarios que participan en el proceso y clasificar los mismos según el grado de impacto para definir prioridades durante los procesos de formación y concientización, y ver cuales pueden ser partícipes del proceso de migración.
- ✓ Analizar el estado actual de los sistemas informáticos de la institución en cuanto a software y hardware. Para esta tarea puede ser de utilidad, debido a la magnitud de la misma, apoyarse en aplicaciones para automatizar el proceso como pueden ser OCS Inventory (OCS Inventory team) .
- ✓ Analizar el estado actual de los servicios que brinda la institución y determinar el grado de criticidad de los mismos para definir prioridades durante el proceso de cambio tecnológico.
- ✓ Evaluar los distintos escenarios posibles que se pueden seguir para acometer la migración.
- ✓ Evaluar las potenciales soluciones de migración disponibles para los sistemas informáticos presentes y definir los más adecuados en cada caso así como definir cuales serán necesarios desarrollar y valorar posibilidades reales de desarrollo de las mismas.
- ✓ Hacer una evaluación de los mecanismos más adecuados que faciliten el soporte y la asistencia técnica dentro de estructura de la institución.
- ✓ Cuantificar las herramientas privativas a migrar.
- ✓ Hacer una evaluación del costo de la migración.

3.4. Flujo de trabajo : Diseño

Luego de la realización de una evaluación de todos procesos de la institución, el segundo flujo será el diseño donde se definirá 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 como se muestra en el punto [1] (Figura 3.3) aunque es importante destacar que el diseño se redefinirá durante la etapa de migración parcial como

retroalimentación de dicha fase donde el impacto será un poco mayor que en las etapas posteriores.

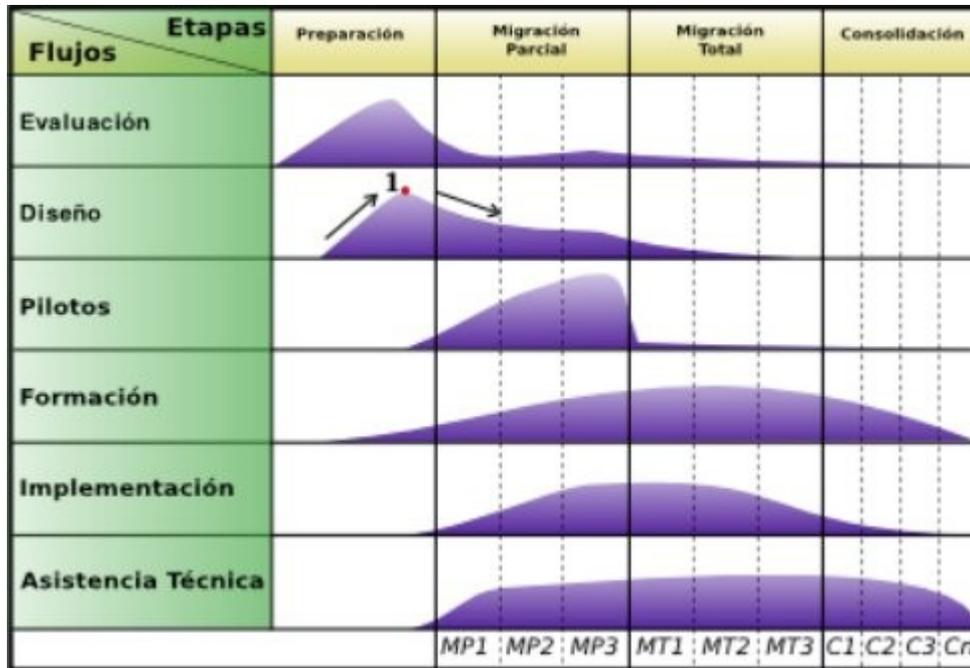


Figura 3.3 Flujo de trabajo de diseño.

Constituyen tareas importantes a realizar durante este flujo:

- ✓ Establecer un plan de acciones concreto que abarque todos los elementos a migrar.
 - Definir una estrategia de sensibilización de acuerdo a los distintos grupos de usuarios, haciendo énfasis en los usuarios que mayor impacto poseen en la toma de decisiones para lograr un mayor apoyo de gestión.
 - Diseñar una justificación para el proceso basado en las ventajas y desventajas que posee la utilización de Software Libre.
 - Definir el número de usuarios que participan en cada fase del plan.
 - Establecer los números y sistemas informáticos que migrarán hacia Software Libre.

- Establecer los períodos de implementación y soporte de las aplicaciones.
- Definir el orden de atención a los usuarios durante la formación según el impacto de los mismos.
- Definir la estrategia para el proceso de instalación soporte y distribución de aplicaciones, esto incluye gestionar los recursos humanos necesarios, discos de distribuciones a instalar, repositorios, etc.
- Diseñar la ruta de migración.
- Diseñar junto al plan de acción la asignación del presupuesto necesario, es importante garantizar el mismo para el desarrollo exitoso del proceso.

3.5. Flujo de trabajo : Pilotos

Una vez que se haya diseñado el plan de migración y 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 nos retroalimentaremos y ajustaremos algunas variables del plan propuesto para poder hacer extensiva la migración con la seguridad de que el número de fallos va a ser mínimo, a dicho flujo lo hemos denominado pilotos, cuyo mayor impacto esta en la etapa de la migración parcial durante la iteración tres, en el momento en que hagamos el pilotaje de migración de las computadoras de escritorio.

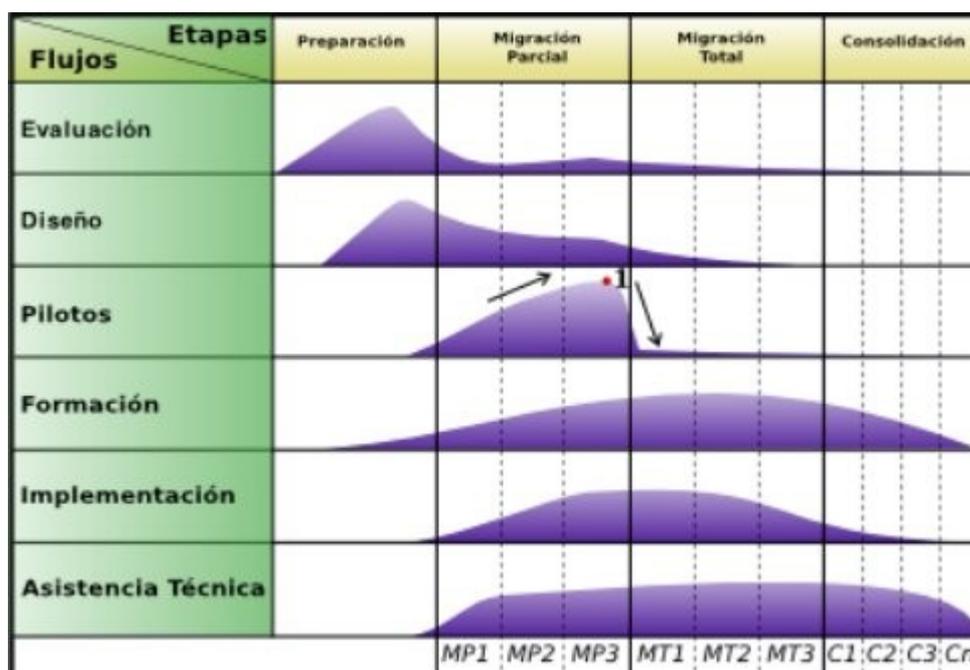


Figura 3.4 Flujo de trabajo de los pilotos.

Las tareas a acometer durante el flujo son :

- ✓ Implementar cada iteración a pequeña escala y validar la configuración propuesta de las aplicaciones.
- ✓ 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, revisión de los sistemas en funcionamiento, etc.
- ✓ Chequear el plan de acciones propuesto.
- ✓ Introducir cambios en el plan de migración propuesto.

Lo más importante de este flujo es, además de lo mencionado anteriormente, que permite desarrollar la experticia y la base de conocimiento necesaria para obtener un modelo replicable de migración eficaz y a corto plazo.

3.6. Flujo de trabajo : Formación

Para lograr que los usuarios acepten la transición lo más importante para ellos es conocer el nuevo sistema, por lo que la capacitación se convierte en el baluarte fundamental de la migración; el flujo de trabajo de formación estará presente durante toda la migración comenzando por la justificación del proceso y abarcando hasta las etapas posteriores al soporte siendo el objetivo fundamental de la misma capacitar a todo el personal en los nuevos sistemas informáticos que se implantan en la institución. Al pasar los usuarios por los distintos puntos del gráfico cada uno ubicado al final de cada etapa e iteración deben adquirir los conocimientos básicos al menos para trabajar con el sistema, por ejemplo:

- ✓ Al pasar por el punto [2] perteneciente a la culminación del pilotaje de instalación y uso de herramientas libres sobre Windows los usuarios involucrados deben saber ya trabajar con herramientas como las propuestas en el anexo 2.
- ✓ Al llegar al punto [6] ya todos los usuarios involucrados en el proceso deberán tener al menos los conocimientos básicos para el trabajo con las nuevas herramientas.

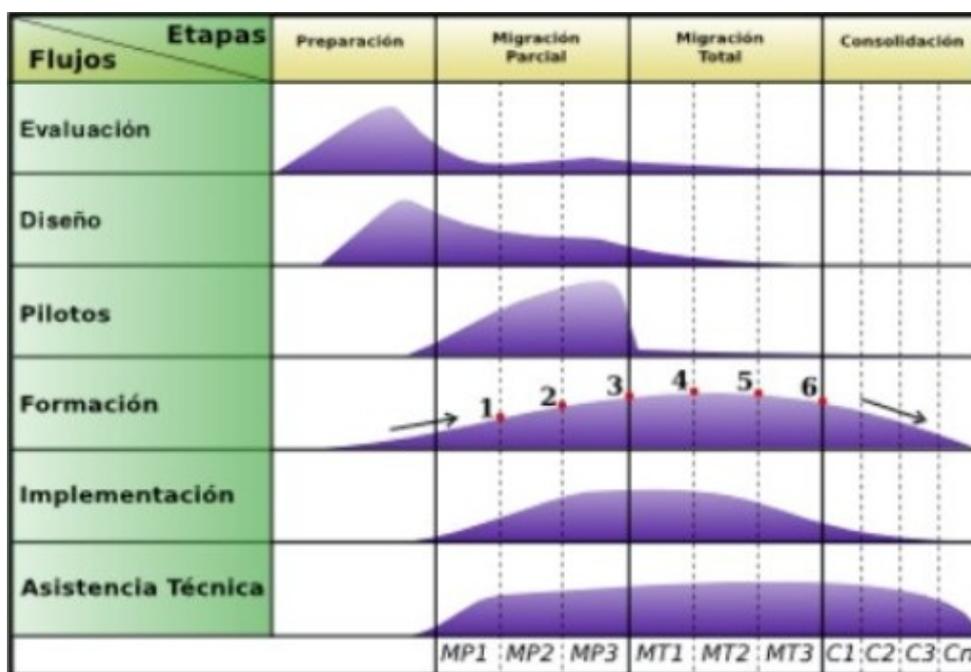


Figura 3.5 Flujo de trabajo de formación.

Como tareas fundamentales se proponen :

- ✓ Confeccionar e impartir planes de formación a los usuarios según el nivel de los mismos, se proponen cuatro niveles básicos (Anexo 3) de usuarios que se pueden identificar, esta propuesta puede desglosarse en más niveles en dependencia del lugar:
 - 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 cómo 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 estos 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 de excelencia.
 - Formación de desarrolladores: La formación orientada a los desarrolladores debe contener planes con 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.
 - Formación de los usuarios finales: 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 este grupo pueden incorporarse secretarías, personal administrativos, algún número de estudiantes, entre otros. Se les debe dotar además las principales direcciones de los recursos donde estos puedan apoyarse en dependencia de los

medios establecidos para dar soporte y capacitación.

Es importante la creación de un mecanismo de certificación para acreditar cada curso vencido e ir estableciendo determinados niveles en los usuarios para lograr una formación más adecuada.

3.7. Flujo de trabajo : Implementación

La implementación es el flujo de trabajo donde se hará efectiva la migración, esta marcará el fin del software privativo según la iteración, 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 esta tarea en la etapa de consolidación.

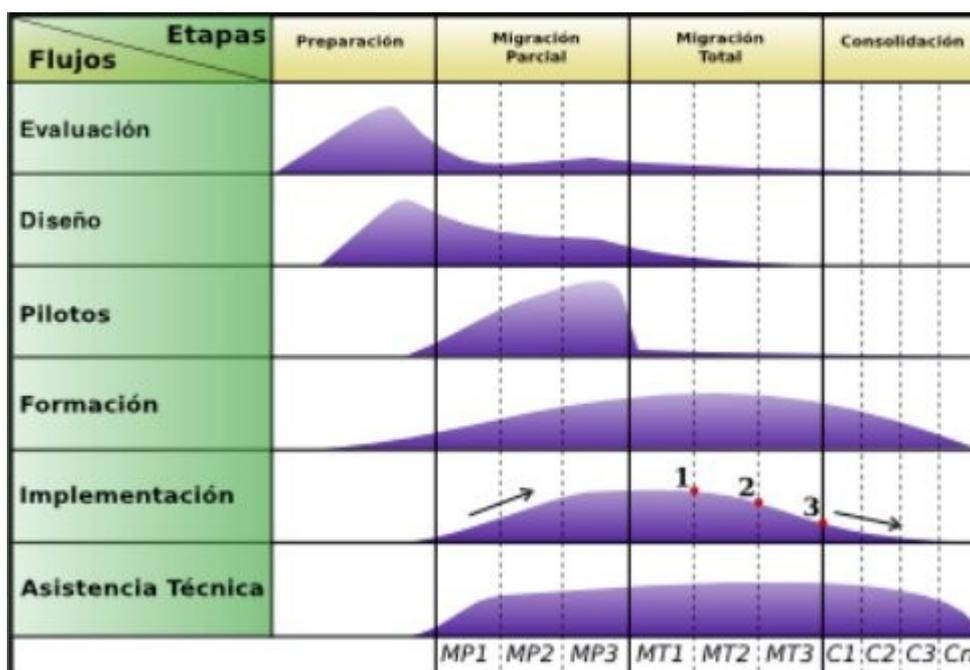


Figura 3.6 Flujo de trabajo implementación.

La implementación de la migración contiene un conjunto de tareas asociadas como otros flujos, algunas de las mismas variarán de un centro a otro en dependencia de las condiciones existentes, las principales tareas concernientes a esta etapa son :

- ✓ Hacer efectiva la migración en dependencia de la iteración, introduciendo los cambios respectivos en la tecnología, cada equipo migrado y configurado significará un elemento significativo en el avance de la migración.
 - Sustitución de los servidores y dominios de Windows a aplicaciones libres. El anexo 4 se definen las herramientas libres más utilizadas para el montaje de servidores en Software Libre.
 - Instalación de herramientas libres sobre Windows (Anexo 2).
 - Cambio de Windows por la distribución(es) de GNU/Linux que se usará(n) como sistema operativo una vez migrada la institución.
- ✓ Fortalecer los planes de formación y certificación de usuarios.
- ✓ Fortalecer los mecanismos de soporte y mantenimiento.

- ✓ Creación de unidades de desarrollo que puedan servir para el desarrollo de aplicaciones locales, así como en la personalización de herramientas e imágenes de GNU/Linux.
- ✓ Movimiento masivo de datos hacia los nuevos formatos y sistemas.

3.8. Flujo de trabajo : Asistencia y soportetécnico.

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 solo que estos tienden a decrementarse , 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 necesidad del lugar por lo que el número de iteraciones en la fase de consolidación donde mayor impacto posee dicho flujo no esta definido a ciencia cierta.

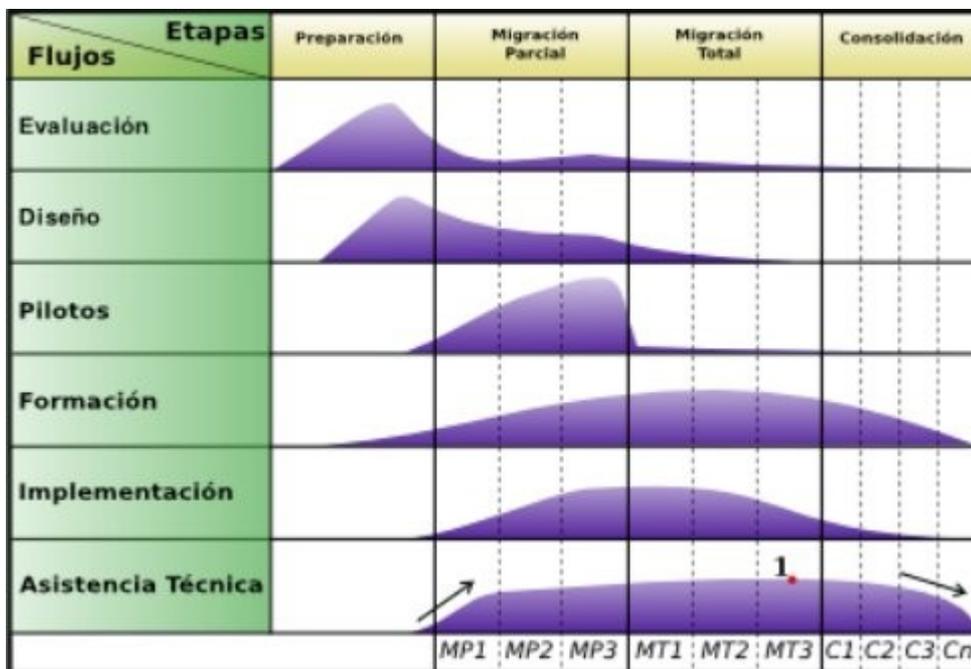


Figura 3.7 Flujo de trabajo de asistencia técnica.

Las tareas y acciones más importantes a llevar a cabo en este momento del proceso son :

- ✓ Creación de listas, foros y canales IRC de ser posible para facilitar el debate de temas relacionados.
- ✓ Crear canales para la atención a los usuarios vía telefónica ante dudas.
- ✓ Creación y mantenimiento de sitios y cursos virtuales para el auto-aprendizaje y la auto-certificación .
- ✓ Creación de repositorios de aplicaciones y datos. Y servicios asociados: personalización de repositorios, secciones propias, etc.
- ✓ Creación de manuales, FAQs, HowTo.
- ✓ Creación en la medida de las posibilidades del centro, comunidades virtuales en la red.
- ✓ Habilitar algún local, stand o espacio físico a donde puedan dirigirse los usuarios a recibir asesoría.
- ✓ Establecer – para los centros en que se realicen desarrollos – políticas que promuevan el uso de estándares para la creación de aplicaciones como pueden ser librerías gráficas, ejemplo: wxwidgets o foxtoolkit y lenguajes de programación como: ANSI C, Java, PHP y otros.
- ✓ Realización de estudios de las aplicaciones más usadas, estudios de las necesidades de los usuarios a fin de lograr mejoras en el servicio de aplicaciones optimización de las mismas para aumentar su rendimiento y saber en cada momento cuales son las necesidades de los clientes.
- ✓ Creación de un portal informativo como centro del proceso que integre los métodos mencionados anteriormente y contenga un Service – Desk o escritorio de servicios para aumentar los tiempos de respuesta ante las preguntas más frecuentes.
- ✓ Lograr la automatización de la mayor cantidad de tareas, por ejemplo implantar

mecanismos para la actualización automática de los repositorios, instalar herramientas que permitan la instalación automática de herramientas en las máquinas de los clientes.

Es importante señalar que el uso de esta metodología no es una camisa de fuerza para las instituciones, la metodología propuesta solo define las pautas de como hacer el proceso ordenado y los pasos que se deben seguir para lograr un mayor grado de éxito, las tareas no tienen que ser necesariamente en ese orden, el mismo varía en dependencia de las condiciones.

4. CONCLUSIONES

El proceso de migración hacia Software Libre de las universidades cubanas es un proceso netamente viable, y sin duda alguna constituirá una verdadera revolución en la cultura de nuestro país, la UCI como centro rector de la informatización del país ya presenta adelantos en tal materia con soluciones prácticas a problemas que enfrenta hoy la migración, por lo que se podría utilizar la fuerza que poseemos para dar un espaldarazo y adelantar dicho proceso en el país.

El Software Libre es un movimiento tecnológico que trae consigo una filosofía de trabajo muy ajustable a nuestros principios socialistas por lo que contar con ella en gran parte de la sociedad sería un gran apoyo a la batalla de ideas que libera nuestra isla.

Se estudiaron a lo largo de este trabajo los principales documentos de migración encontrados que se ajustan al objetivo del mismo lo que permitió crear una metodología más completa que respondiera al objetivo de esta investigación, se analizaron las Directrices IDA de la Unión Europea, el Plan Institucional de Liberación y Migración a Plataformas de Software Libre de la ULA en Venezuela y la Metodología para la Migración a Software Libre de la UCI, siendo este último la base para la elaboración de este documento por constituir muy completo y ajustado a los objetivos propuestos.

Por último se elaboró la Metodología para la Migración a Software Libre de las universidades del MES, definiendo en ella los pasos para acometer la migración en estos centros, definiendo un conjunto de etapas y flujos de trabajo que guían el proceso hasta su culminación, recomendando de igual manera un conjunto de aplicaciones para usar.

Paralelamente a este trabajo se desarrolla una propuesta para la migración a del Ministerio de Educación (MINED), la guía de migración de la empresa comercial DIVEP y se están sentando las bases para la migración de la Empresa de Telecomunicaciones de Cuba (ETECSA) y el Ministerio de Auditoría y Control (MAC), lo que sin dudas significa un gran emprendimiento hacia la soberanía tecnológica.

Está entonces en manos de los cuadros de dirección del MES la efectivización de esta metodología en la universidades, del apoyo de gestión que le presten al mismo dependerá el éxito de la migración hacia Software Libre del los centros de altos estudios.

5. RECOMENDACIONES

Sin temor a equivocarnos la migración a Software Libre avanza en estos momentos más que nunca en la isla, ETECSA, MAC, MINED y MES son ejemplos de ministerios que evidencian este proceso de cambio tecnológico.

Por la importancia que revierte la migración hacia Software Libre para nuestro país se recomienda :

- ✓ Aplicar esta metodología en los centros universitarios asociados al MES.
- ✓ Realizar un análisis de los posibles resultados a fin de introducir o suprimir elementos de la propuesta y validar la gráfica propuesta.
- ✓ La elaboración de una guía de migración nacional.
- ✓ De manera general y para los nuevos tesis que realicen guías de migración con empresas, organismos e instituciones externas que realicen antes iniciarse a trabajar una especie de compromiso entre las partes que garantice la participación de ambos en la tarea.

6. REFERENCIAS BIBLIOGRAFICAS

- Adobe. Adobe - Adobe Store - International - Adobe Photoshop CS3. 2007. [cited 22 May 2008]. Available from world wide web:
<https://store3.adobe.com/cfusion/store/index.cfm?store=OLS-AP&view=ols_prod&category=/Applications/Photoshop&distributionMethod=FULL&nr=0>.
- Equipo técnico de migración. *Guía Libre. Referencia de migración para Software Libre del Gobierno Federal* . 2003 Available from world wide web:
<<http://www.governoeletronico.gov.br/anexos/versao-em-espanhol-do-guia-livre>>.
- Grupo técnico nacional. *Lineamientos para la migración a Software Libre en Cuba*. 2005 Available from world wide web: <<ftp://softwarelibre.cu>>.
- Hugo Rafael Chávez. *Decreto 3390*. December 2004 Available from world wide web:
<<http://www.cenit.gob.ve/cenitcms/servlet/com.mvdcomm.cms.andocasociado?5,64>>.
- IBM. IBM Research | Eclipse. 2007. [cited 22 May 2008]. Available from world wide web:
<<http://www.research.ibm.com/eclipse/>>.
- Infomed. AULA VIRTUAL DE SALUD. May 2008. [cited 22 May 2008]. Available from world wide web: <<http://aulauvs.sld.cu/>>.
- Jesus Javier Estepa Nieto. Software Libre para el tercer mundo. August 2007, 75,87,82,95,146,162,171,206.
- Juan Vela Valdés. La universidad Cubana crea una nueva sociedad. December 2007. Available from world wide web: <<http://www.juventudrebelde.cu/cuba/2007-12-13/la-universidad-cubana-crea-una-sociedad-nueva/>>.
- Lena Zuñiga Blanco. *Voces libres de los campos digitales*. [Costa Rica], 2006.
- Linus Benedict Torvalds. What would you like to see most in minix ? August 1991. Available from world wide web:
<<http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b?dmode=source>>.
- Mandriva. Support & Services - Mandriva Store. 2008. [cited 22 May 2008]. Available from world wide web:
<<http://store.mandriva.com/index.php?cPath=27&osCsid=0f74eff7a1fa3e9850948c70cb5de5bc>>.
- Marcos Guglielmetti. Dell vende Notebooks y PC de escritorio con GNU/Linux en Latinoamérica. February 2008. Available from world wide web:
<<http://www.mastermagazine.info/articulo/12667.php>>.

- netProject Ltd, and European Communities. *Directrices IDA de migración a fuentes abiertas*. [Union Europea], October 2003 Available from world wide web: <<http://www.netproject.com/docs/migoss/v1.0/>>.
- Novell. Novell Provides the Best Linux Support in the World. 2008. [cited 22 May 2008]. Available from world wide web: <http://www.novell.com/linux/support-survey.html?sourceid=supbanner_bestlinux>.
- OCS Inventory team. *Open Computer Software Inventory*. Available from world wide web: <<http://www.ocsinventory-ng.org/>>.
- ONE. Oficina Nacional de Estadísticas. Cuba. 2007. [cited 22 May 2008]. Available from world wide web: <<http://one.cu/>>.
- Pablo Javkin. Ordenanza para la migración a Software Libre municipalidad rosario. March 2004. [cited 22 May 2008]. Available from world wide web: <<http://lugro.org.ar/biblioteca/textos-gral/ordenanza.html>>.
- PC World. PC World - The 100 Best Products of 2005. July 2005. [cited 22 May 2008]. Available from world wide web: <<http://www.pcworld.com/article/id,120763-page,1/article.html>>.
- Proyecto GNU. La Definición de Software Libre - Proyecto GNU - Fundación para el Software Libre (FSF). 2001. [cited 21 May 2008]. Available from world wide web: <<http://www.gnu.org/philosophy/free-sw.es.html>>.
- Ramón Paumier Samón. Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas . May 2007, 18,36.
- RedHat Inc. redhat.com | Red Hat Support. 2008. [cited 22 May 2008]. Available from world wide web: <<https://www.redhat.com/apps/support/>>.
- Richard M. Stallman. new Unix implementation. September 1983. Available from world wide web: <<http://www.gnu.org/gnu/initial-announcement.html>>.
- Roberto Di Cosmo. Software Libre y competitividad para las empresas. August 2007. Available from world wide web: <<http://www.pps.jussieu.fr/~dicosmo/TALKS/Cordoba-Empresas-08-2007.pdf>>.
- Steve Best. JFS overview. January 2000. [cited 22 May 2008]. Available from world wide web: <<http://www.ibm.com/developerworks/library/l-jfs.html>>.
- The White House Washington . The National Strategy for Secure Cyberspace . February 2003. Available from world wide web: <<http://www.whitehouse.gov/pcipb/>>.
- UCI. Bienvenidos al Entorno Virtual de Aprendizaje. May 2008. [cited 22 May 2008]. Available from world wide web: <<http://teleformacion.uci.cu/>>.
- UCO. Universidad Católica de Oriente. May 2008. [cited 22 May 2008]. Available from world wide web: <<http://www.uco.edu.co:8081/moodle/>>.

ULA. *Plan Institucional de Liberación y Migración a Plataformas de Software Libre*. May 2006
Available from world wide web:
<http://nux.ula.ve/documentos/Plan_de_Migracion_Mayo_2006_CCA.pdf>.

Venezolana de televisión. *Hace cinco años: El nefasto paro petrolero del 2002*. Venezolana de
televisión, February 2007 Available from world wide web:
<<http://www.aporrea.org/oposicion/n105719.html>>.

7. BIBLIOGRAFIA

- Equipo técnico de migración. *Guía Libre. Referencia de migración para Software Libre del Gobierno Federal* . 2003 Available from world wide web: <<http://www.governoeletronico.gov.br/anexos/versao-em-espanhol-do-guia-livre>>
- Grupo técnico nacional. *Lineamientos para la migración a Software Libre en Cuba*. 2005 Available from world wide web: <<ftp://softwarelibre.cu>>.
- Hugo Rafael Chávez. *Decreto 3390*. December 2004 Available from world wide web: <<http://www.cenit.gob.ve/cenitcms/servlet/com.mvdcomm.cms.andocasociado?5,64>>.
- Jesus Javier Estepa Nieto. *Software Libre para el tercer mundo*. August 2007.
- Lena Zuñiga Blanco. *Voces libres de los campos digitales*. [Costa Rica], 2006.
- netProject Ltd, and European Communities. *Directrices IDA de migración a fuentes abiertas*. [Union Europea], October 2003 Available from world wide web: <<http://www.netproject.com/docs/migoss/v1.0/>>.
- Pablo Javkin. Ordenanza para la migración a Software Libre municipalidad del rosario. March 2004. [cited 22 May 2008]. Available from world wide web: <<http://lugro.org.ar/biblioteca/textos-gral/ordenanza.html>>.
- Ramón Paumier Samón. Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas . May 2007.
- Roberto Di Cosmo. Software Libre y competitividad para las empresas. August 2007. Available from world wide web: <<http://www.pps.jussieu.fr/~dicosmo/TALKS/Cordoba-Empresas-08-2007.pdf>>.
- The White House Washington . The National Strategy for Secure Cyberspace . February 2003. Available from world wide web: <<http://www.whitehouse.gov/pcipb/>>
- ULA. *Plan Institucional de Liberación y Migración a Plataformas de Software Libre*. May 2006 Available from world wide web: <http://nux.ula.ve/documentos/Plan_de_Migracion_Mayo_2006_CCA.pdf>.
- Instituto Nacional de Estadísticas e Informática. *Guía para la migración de Software Libre en las entidades publicas*. 2002.
- Universidad de Misiones. *Proyecto de migración de software*. 2006.

8. ANEXOS

8.1. ANEXO 1: HERRAMIENTAS FLOSS, PARA APLICACIONES WINDOWS

Grupo UNICORNIOS
(Servicios Especializados para la Migración a Software Libre)

Tabla de equivalencias por asignaturas. (12 de abril de 2008)
Versión 0.8.1

Asignatura	Windows	Linux
Programación (C & C++)	C++ Builder x.x	KDE Studio Gold (1) Dev-C++ (1) Kylix3 [Prop] (6) vtkBuilder (1) foxBuilder (1) wxDesigner (1) Code Crusader[Prop] (1) CodeWarrior[Prop] (1) Gbuilder (6) Source Navigator (1) TimeStorm[Prop] (1) Understand for C++[Prop] (1) SlickEdit.[Prop] (1) Vide (1) Ultimate (1) Anjuta + Glade (2) Code::Blocks (2) Eclipse + Plugin C++(2) Kdevelop + QtDesigner (3)
Programación	VisualStudio.net	CodeForge (1) Kdevelop + Qt3 Designer (2) Eclipse (2) Emacs (6) Xemacs (6) Monodevelop (3)
Programación	Pascal	FreePascal (1)

Asignatura	Windows	Linux
		GNU Pascal (1) Rshell (Borland Pascal 7.0) (1) Lazarus (3)
Programación (HTML)	Editores HTML Dreamweaver FrontPage	Quanta Plus (3) Bluefish (3) WebMaker (1) Screem (6) Toppage (1) WebDesigner (1) ScripEditor (1) Webdevelopment (plugging firefox) (2) Amaya (1) August (1) Nvu (2) Dreamweaver Emulado (6) ZendStudio (3)
Programación(Java)	JBuilder	Jbuilder for Linux (1) NetBeans (2) Sun One Studio (1) Vide (1) Eclipse (3)
Programación(PHP)	Dreamweaver	ZendStudio (3) gphpedit (5) Dreamweaver Emulado (6)
Programación (VisualBasic)	VisualBasic	Gambas (2)
Programación(Python)		Boa Constructor. (2) WingIDE (3) Spe (1)
Máquinas Computadoras	TASM MASM	NASM + ddd (3) GAS (2) MSDOS emulado con qemu (2)
Ingeniería del software	R.Rose	Umbrello (2) ArgoUML (6) Dia + Dia2Code (6)

Asignatura	Windows	Linux
		PocceidonCE (1) TogetherControCenter[Prop] (1) Rational Rose[Prop] (6) Visual Paradigm [prop] (3)
Bases de Datos [Plataforma]	SQL Server PostgreSQL MySQL Oracle	PostgreSQL (3) MySQL (3) Oracle (6)
Bases de datos [Diseño]	Erwin	Gerwin (2) DBDesigner (3) Dia + code
Bases de Datos [Administración]	SQL Server	Pgacces(Postgre) (6) pgadmin3(Postgre) (3) phpmyadmin (MySQL) (3) webmin (6) EMS postgres Manager[prop] (6)
Matemática	MatLab	MatLab[Prop] (2) Octave (3) Scilab + Entorno Gráfico (1) R. (1) Yorick (1) rlab (1) Yacas (1) Euler (1)
Matemática	Maple	Maple para Linux[Prop].(5) Maxima (3) MuPad (1)
Matemática	Derive	Maxima (1) Xmaxina (2) Wxmaxima (3)
Matemática [Editor de ecuaciones]	Mathtype MSEquation Editor	OpenOffice Math (3) MathMLed (6) Kformula(KOffice) (6) LyX (6)
Gestión de Software	MSProyect	Planner (6) GantProject (6) Dotproject (3) Track (2)
Diseño Gráfico	Photoshop	Gimp (3)

Asignatura	Windows	Linux
(mapa de bits)		
Diseño Gráfico (vectorial)	CorelDRAW	Inskape (3) Skencil (2)
Edición de video	Adobe Premiere	Cinelerra (3) Jahshaka (2) avs video converter (1)
Convertidores de Formato		Media Coder (2) Mencoder (3) Avidemux (1)
Modelado 3D	3D Max Studio	Blender (3)
Edición de Audio		Audacity (3)

Leyenda	Herramientas
1	En estudio
2	Propuestas Estudiadas
3	Propuesta óptima
4	En estudio con URGENCIA
5	Estudio planificado en breve, IMPORTANTE
6	No recomendada

Documento oficial:

Herramientas FLOSS, para aplicaciones de Windows.

Grupo UNICORNIOS

(Servicios Especializados para la Migración a Software Libre)

Autor: Ing. Abel Meneses Abad

Mantenedores: Ing. Abel Meneses Abad

Est. Yoandy Perez Villazón

Est. David Padrón Álvarez

Est. Jorge Mijail Vázquez Paredes

Fundamentado por Grupo de Investigación de Migración, UNICORNIOS,

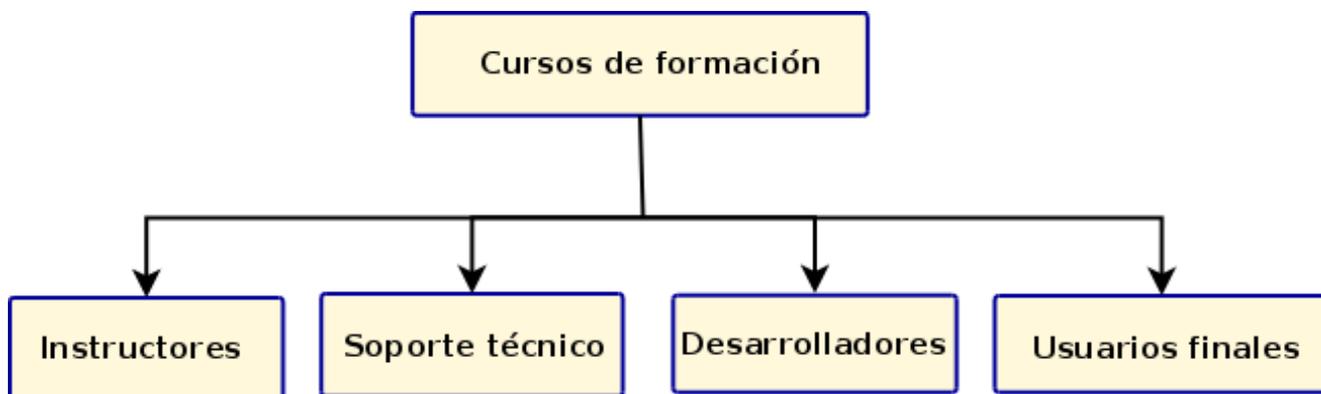
Polo de SWL UCI

v0.8.1, 12 de Abril de 2008

8.2. ANEXO 2: HERRAMIENTAS LIBRES PARA MICROSOFT WINDOWS.

Aspecto a considerar	Software Privativo	Software Libre
Editor de diagramas	MS Visio	Dia
Ofimática	MS Office	OpenOffice.org
Gestión de finanzas	MS Money	GNUCash, Grisbi
Gestión de proyectos	MS Project	DotProject
Maquetación	FreeHand, Adobe Illustrator	Scribus
Agenda y calendario	MS Outlook	Mozilla Sunbird
Cliente de correo	MS Outlook	Mozilla Thunderbird
Navegadores	Internet Explorer	Mozilla Firefox
IRC	Xchat	XChat
Mensajería instantánea	Pandion, MSN	Pidgin, aMSN
Bases de datos	MS SQL	PostgreSQL, MySQL
Desarrollo .NET	Visual Studio .NET	Mono
Desarrollo C/C++	C++ Builder	Code::Blocks
Desarrollo python	WingIDE	SPE
Editores	Notepad	Scite
UML	Rational Rose	ArgoUML
Gráficos 3D	3DMax	Blender
Gráficos 2D	Adobe Photoshop	Gimp
Herramienta CAD	AutoCAD	FreeCAD
Fuentes	Arial	Bitstream Vera, DejaVu
Video y multimedia	Windows Media Player	Mplayer, VLC
Gráficos infantiles	Paint	TuxPaint
Antivirus	Nod 32, Kasperky	ClamWin

8.3. ANEXO 3: NIVELES DE USUARIOS PROPUESTOS PARA LA FORMACION.



8.4. ANEXO 4: SOFTWARE UTILIZADO EN LOS PRINCIPALES SERVICIOS SOBRE GNU/LINUX.

Servicio	Propuesta Libre
Servidor Web	Apache
Servidor de bases de datos	MySQL, PostgreSQL
Servidor de nombres (DNS)	BIND
Servidor DHCP	DHCPd
Servicio de impresión	SAMBA, CUPS
Servidor de correo	Postfix + Cyrus IMAP + SASL + TLS + LMTP + Sieve + SpamAssassin + ClamAV + Mailman + SquirrelMail.
Transferencia de archivos (FTP)	Proftpd, Pureftpd, Vsftpd
Compartición de archivos de Windows	SAMBA
Proxy	SQUID
Mensajería instantánea	Jabber

8.5. ANEXO 5: FLUJO DE PROCESOS EN LA METODOLOGIA.

Entrada	Proceso	Salida
<p>Sistemas informáticos (Hardware & Software)</p> <p>Listado de usuarios involucrados</p> <p>Escenario del proceso</p>	<p>Evaluación</p>	<p>Inventario completo (Hardware & Software)</p> <p>Listado de usuarios por nivel y usuarios potenciales</p> <p>Objetivos y necesidades</p>
<p>Inventario completo (Hardware & Software)</p> <p>Listado de usuarios por nivel y usuarios potenciales</p> <p>Objetivos y necesidades.</p>	<p>Diseño</p>	<p>Guía de migración, acciones concretas, justificación, ruta de migración, prioridades.</p>
<p>Guía de migración, acciones concretas, ruta de migración, prioridades.</p>	<p>Pilotos</p>	<p>Ajustes para el documento.</p> <p>Experticia del equipo.</p> <p>Validación de la configuración.</p>
<p>Niveles de usuarios según su impacto.</p>	<p>Formación</p>	<p>Usuarios preparados.</p> <p>Nuevos preparadores y formadores.</p> <p>Mayor visión y conocimiento</p>
<p>Personal preparado</p> <p>Modelo replicable de migración.</p>	<p>Implementación</p>	<p>Servidores con FLOSS</p> <p>Computadoras Windows con FLOSS.</p> <p>Computadoras de escritorio con FLOSS totalmente.</p>
<p>Dudas ante el nuevo sistema.</p> <p>Problemas de SW & HW</p>	<p>Asistencia técnica</p>	<p>Sistemas de soporte adecuados.</p> <p>Satisfacción de los usuarios.</p> <p>Soluciones de alto nivel.</p>

9. GLOSARIO DE TERMINOS

AT&T Bell Labs: Son varios centros de investigación científica y tecnológica ubicados en más de diez países y que pertenecen a la empresa estadounidense Lucent Technologies (Alcatel-Lucent).

Unix: Es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

IBM: 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.

IBM PC: (Personal Computer -computadora personal) es una marca comercial de IBM.

Apple: Es una empresa estadounidense de tecnología informática creada en 1976 con sede en Cupertino, California (Estados Unidos), sus principales productos son Apple Macintosh, Mac OS X, iPod, QuickTime, iLife, iWork, Final Cut Studio, Aperture, Logic Pro, Cinema Display, AirPort, iPhone.

MIT: El Instituto Tecnológico de Massachusetts, es una de las principales instituciones dedicadas a la docencia y a la investigación en Estados Unidos, especialmente en ciencia, ingeniería y economía, es considerada como una de las mejores universidades de ciencia e ingeniería del mundo.

GNU: Proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

GPL: Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

GCC: GNU Compiler Collection es un conjunto de compiladores creados por el proyecto GNU. GCC es software libre y lo distribuye la FSF bajo la licencia GPL.

Emacs: Es un editor de texto con una gran cantidad de funciones, muy popular entre programadores y usuarios técnicos, desarrollado por GNU su nombre deviene de editor de macros.

Minix : Es un clon del sistema operativo Unix distribuido junto con su código fuente y desarrollado por el profesor Andrew S. Tanenbaum en 1987.

Ubuntu GNU/Linux: Es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a computadoras de escritorio aunque también proporciona soporte para servidores, su desarrollo esta basado en Debian.

Debian GNU/Linux: Debian GNU/Linux es la principal distribución Linux del proyecto Debian, que basa su principio y fin en el software libre. Creada en el año 1993.

Red Hat : Es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux, y de otra más, Fedora.

Suse Linux: Es una de las más conocidas distribuciones Linux existentes a nivel mundial, se basó en sus orígenes en Slackware, es desarrollada por Novell y originaria de los Estados Unidos.

Mandriva : Es la empresa francesa de software dedicada a ofrecer y hacerse cargo de su distribución Linux, Mandriva Linux tiene su sede en Paris, anteriormente denominada Mandrake.

Ututo: Es una distribución de GNU/Linux, La fundación del software libre, FSF reconoce al proyecto Ututo como una distribución GNU/Linux totalmente libre, su país de origen es argentina.

Gentoo: Es una distribución GNU/Linux orientada a usuarios con cierta experiencia en este sistema operativo, fue fundada por Daniel Robbins, basada en la inactiva distribución llamada Enoch Linux. Ya para el año 2002, ésta última pasa a denominarse Gentoo Linux.

I+D: La expresión Investigación y Desarrollo (ó I+D, en abreviatura) tiene una relevancia comercial especial aparte de su significado convencional de investigación y desarrollo tecnológico. En el contexto de comercio, "investigación y desarrollo" se suele referir a

actividades de largo recorrido orientadas al futuro en tecnología o ciencia copiando la investigación científica que no se vincula tanto a la búsqueda de beneficios.

Symantec: Fundada en 1982, es una corporación internacional que vende software para computadoras, particularmente en el dominio de la seguridad informática. Con la sede central en Cupertino, California, Symantec opera en más de cuarenta países.

BRIC: En economía internacional, se emplea la sigla BRIC para referirse conjuntamente a Brasil, Rusia, India y China, que tienen en común una enorme población, y lo que es más importante, que en los últimos años han presentado cifras de crecimiento de su PIB y de participación en el comercio mundial muy elevados, lo que les hace atractivos como destino de inversiones.

OpenOffice.org: Es una suite ofimática de software libre y código abierto de distribución gratuita que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para muchas y diversas plataformas, como Microsoft Windows, sistemas de tipo Unix con el Sistema X Window como GNU/Linux, BSD, Solaris y Mac OS X. También está disponible para Mac OS X un programa similar derivado denominado NeoOffice.

Apache: Servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, es modular, extensible, y muy popular.

Sendmail: Es un popular MTA (Mail Transport Agent) en internet, cuya tarea consiste en "encaminar" los mensajes correos de forma que estos lleguen a su destino.

Microsoft Exchange Server : Es un software de colaboración entre usuarios, desarrollado por Microsoft, es parte de la familia Microsoft Server ya que es una de las aplicaciones destinadas para el uso de servidores. Exchange es utilizado para brindar soluciones para grandes empresas corporativas.

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. En enero de 2008, MySQL AB fué adquirida por Sun Microsystems, y por tanto MySQL también.

PHP: Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK.

Decreto 3390: Decreto aprobado por el presidente venezolanos Hugo Rafael Chavez Frías para impulsar el uso del Software Libre en la República Bolivariana de Venezuela el 23 de diciembre del 2004.

FLOSS: Es la sigla que identifica, también en inglés, al concepto del "Free/Libre Open Source Software", pero haciendo mención a la idea filosofía del Software Libre (Free Software), donde "Free" menciona el concepto de "Libre", y no la idea del software gratis, de ahí lo de "F/L", ya que en inglés se tiende a confundir y hasta a trasladarse la confusión al español.

Iteración: Iteración se refiere a la acción de repetir una serie de pasos un cierto número de veces.

Protocolo: Protocolo de red o también Protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red.

PDF: Sigla en inglés Portable Document Format, Formato de Documento Portátil. Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Está especialmente ideado para documentos susceptibles de ser impresos, ya que especifica toda la información necesaria para la presentación final del documento, determinando todos los detalles de cómo va a quedar, no requiriéndose procesos anteriores de ajuste ni de maquetación.

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

Ancho de banda: Es la anchura, medida en hercios, del rango de frecuencias en el que se concentra la mayor parte de la potencia de la señal. Puede ser calculado a partir de una

señal temporal mediante el análisis de Fourier. También son llamadas frecuencias efectivas las pertenecientes a este rango.

GUI: Interfaz gráfica de usuario, es el artefacto tecnológico de un sistema inactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

ANSI C: La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C.

Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.

Python: Lenguaje de programación creado por Guido van Rossum en el año 1990. Es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

WxWindows: Librería y conjunto de componentes que permite a los desarrolladores crear aplicaciones para plataformas como Mac OS X, GNU/Linux, Win32, Motif, y otros , utilizando lenguajes como C++, Python, Perl, y C#/.NET.

Foxt toolkit: Conjunto de herramientas utilizado para el desarrollo de interfaces gráficas de usuario de forma fácil, contiene controles, soporte y componentes para el trabajo con OpenGL, facilidades de arrastrar y soltar, la principal ventaja es la independencia de la plataforma por lo que podremos desarrollar aplicaciones para Mac OS X, GNU/Linux, Win32 y otros.

LiveCD: Un LiveCD o LiveDVD es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí su nombre), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro del ordenador, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de ficheros.

Herramientas colaborativas: Herramienta o conjunto de herramientas que permite la colaboración y el intercambio durante el trabajo , algunos ejemplos de ellas constituyen

Subversion (control de versiones), dotproject (gestión de proyecto, tareas, calendario), egroupware (correo, jabber, proyecto, wiki, gestión documental y más), entre muchas otras.

Ofimática: Se llama ofimática al equipamiento usado para idear y crear, coleccionar, almacenar, manipular y transmitir digitalmente la información necesaria en una oficina para realizar tareas y lograr objetivos básicos.

Spam: Es la denominación dada para los mensajes no solicitados, habitualmente de tipo publicitario, enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor, también es llamado correo basura aunque no es el e-mail la única forma en que estos arriban a los receptores.

DNS: El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

Repositorio: Sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Está preparado para distribuirse habitualmente sirviéndose de una red informática como Internet o en un medio físico como un disco compacto.

IRC: Internet Relay Chat es un protocolo de comunicación en tiempo real basado en texto, que permite debates en grupo o entre dos personas.

Wine: Es una reimplementación de la API de Win16 y Win32 para sistemas operativos basados en Unix bajo plataformas Intel. Permite la ejecución de programas para MS-DOS, Windows 3.11, Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP y Windows Vista.

OCS Inventory: Poderoso paquete de herramientas para hacer inventario de software y hardware de computadoras de una red, desarrollado para sistemas Windows y Unix, basa su funcionamiento en el modelo cliente-servidor.

FAQ: Es el acrónimo de Frequently Asked Questions en inglés, que en español significa

preguntas frecuentes. El término se refiere a una lista de preguntas y respuestas, dentro de un determinado contexto y para un tema en particular.

How to: Constituye un documento informal, generalmente corto, que describe cómo cumplir con una cierta tarea, en español es denominado *como*.