

CARACTERÍSTICAS Y PERSPECTIVAS DEL SOFTWARE LIBRE PARA LA EDUCACIÓN MATEMÁTICA.

Master Eduardo Chaves B

Educología, CIDE, UNA
Universidad Nacional
echav@una.ac.cr

Master Yuri Morales L

Escuela de Matemática
Universidad Nacional
ymorales@una.ac.cr

Objetivo de la ponencia: Promover el uso del software libre como fuente de recursos didácticos para la educación Matemática.

Equipo: Computadora, Video Beam.

Resumen: en este trabajo se desea mostrar una breve cronología histórica, una caracterización y justificación del diseño y uso del software libre; además, se pretende señalar las ventajas de su uso mediante ejemplificaciones de sistemas operativos, software dinámico de geometría y CAS [Computer Algebra Systems].

Palabras clave: Software libre, historia del software, Matemática.

1. INTRODUCCIÓN

Diversos estudios han señalado el potencial de los recursos tecnológicos en los procesos educativos de la Matemática; en consecuencia, las instituciones realizan esfuerzos para financiar la compra de equipo, la instalación de software, mantenimiento de laboratorios, capacitación, entre otros.

Bajo esta perspectiva, se deben analizar temas como: el software que está a disposición de los docentes, el mecanismo que ellos tienen para seleccionarlo, los procesos de capacitación y auto capacitación en los que participan, el acceso, el intercambio de experiencias y la filosofía que sustenta la creación y uso de estos recursos.

En Costa Rica poco a poco nacen las asociaciones que reúnen a interesados en la utilización y divulgación de software libre en todas las áreas. Para nuestro interés, este trabajo tiene como objetivo divulgar a la comunidad educativa matemática algunas herramientas útiles en el desarrollo de la mediación pedagógica.

2. MARCO REFERENCIAL

En los siguientes apartados se explora el desarrollo histórico de la creación y uso de software libre, así como su conceptualización y se exponen una serie de argumentos con

el fin de justificar la importancia del uso de estos recursos como una ventaja real en nuestro país. Para esto, nos referimos a su impacto en Costa Rica y posibles aplicaciones, principalmente, en el sector educativo matemático.

2.1 **Qué es software libre**

Un excelente punto de partida para la comprensión del concepto de software libre es volver la mirada a quien lo concibió por primera vez de la manera en que hoy es aceptado; Richard Stallman propuso que un software se puede considerar software libre (SL) si el usuario tiene **libertad** de para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Uno de los primeros problemas que encontró este concepto se formó a partir de su traducción. El uso “*free software*” contó con el inconveniente que *free* en Ingles puede comprenderse tanto libre como gratis. Por eso la primera condición para entender qué es software libre es comprender que lo que está en juego **no es** necesariamente el dinero (si es gratis o no) sino la libertad.

En el sitio oficial Free Software Foundation. **FSF** (*GNU Operating System*) se encuentran las 4 libertades que propuso Stallman:

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

Veamos con una analogía otra percepción sobre el software libre: *supongamos por un momento que su gobierno decide que los cuadernos deben ser gratis. Cada persona tiene derecho a una calidad ilimitada de los mismos, pero deben ser retirados en la casa presidencial. Para algunos de zonas alejadas será sumamente difícil traerlos por lo que una persona decide establecer un negocio en el cual obtiene ganancias únicamente por el traslado de los mismos; ¿está esa persona cometiendo un delito?. La respuesta, intuitivamente, debe ser No.*

Lo anterior explica el por qué en algunos sitios o empresas cobran al **descargar** un software libre. Es aquí donde se debe aclarar que lo opuesto a software libre **no es** software comercial; por lo que un software puede ser comercializado y seguir siendo libre. *Además, no importa como consiguió usted los cuadernos, sino que usted tiene libertad de hacer con ellos lo que desee.*

La definición más aceptada para el software que no cumple del todo con las 4 libertades es software privativo (propietario) según la **FSF** y es válido usarlo como opuesto. Hace un par de años también está permitido usar software semi – libre para los software que cumplen algunas de las 4 libertades. En general, estos son los conceptos más significativos en este trabajo, pero existe una clasificación mucho más exhaustiva descrita en la **FSF** (Software libre, Código abierto «Open source», Software de dominio público, Software protegido con copyleft, Software libre no protegido con copyleft, Software cubierto por la GPL, Software de GNU, Software no libre, Software semi – libre, Software privativo, Shareware, Freeware, Software privado -a medida-, Software comercial).

2.2 **Cronología**

La evolución histórica del software libre podemos resumirla en tres grandes etapas: niñez, adolescencia y madurez.

- **Etapa de Niñez** [Desde el nacimiento del software hasta 1970]

Los sistemas más comunes como los de IBM y DECUS incorporaban el software a la compra del hardware por lo que no se concebía, en esos momentos, que se pudiera lucrar con ellos. Según Gonzáles y otros (2003), *“Podría decirse que durante estos primeros años de la informática el software era libre, al menos en el sentido de que los que tenían acceso a él podían disponer habitualmente del código fuente, estaban acostumbrados a compartirlo, a modificarlo y a compartir las modificaciones”* (p 32)

- **Etapa de adolescencia** [1970 – 1983]

A principios de la década de 1970, IBM comercializa parte del software como producto aparte del hardware de manera que los dos debían pagarse por separado. Esto significó que el software se convirtió en un bien comercial y prosperaba el desarrollo de software como una forma de ganar dinero. Respecto al software libre nacieron Spice, TEX y Unix.

- **Etapa de madurez [1984 – a la fecha]**

En esta etapa inicia la contra propuesta al software privativo; un evento define el nacimiento de la misma: en 1984 Richard Stallman renuncia a los laboratorios del Instituto de Tecnología de Massachusetts (MIT) en Boston para desarrollar un sistema de software completamente **libre**. Este proyecto lo llamó GNU (se lee en español como Ñu) que es un acrónimo recursivo GNU's Not Unix. Además, fundó la **FSF** y definió los principios políticos, comerciales y éticos del software libre. La madurez empieza en el momento que se desea que el software vuelva a su estado original (Se recomienda el libro Introducción al Software Libre de Gonzáles y otros, para ahondar en la historia).

2.3 Qué se obtiene con su uso

Como se señaló en los apartados anteriores, el software libre guarda una estrecha relación con la libertad de los individuos; a esto se debe agregar que para algunos, la sola idea de poder compartir sin fin de lucro es más estimulante que “hackear” el software propietario. Para otros, el paradigma del software libre estimula ideas tan loables como la producción en beneficio de la sociedad.

Aún así, existen una serie de preguntas esenciales, a nuestro criterio válidas, que rondan la implementación e impacto del SL; unas de tipo técnico, ¿es el software libre tan seguro y confiable como el software propietario? ¿Al existir una entidad detrás de la venta de un software se asegura la atención rápida y segura a los problemas que presente el mismo? Otras preguntas guardan una estrecha relación con el comercio, beneficio y ventajas del SL.

Un ejemplo sobre los beneficios y los argumentos para su empleo es su utilidad en las universidades, según Delgado y Oliver (2006)

“Bajo coste; la promoción de la industria local; el fomento de la innovación tecnológica, empleando la metodología típicamente científica; el escrutinio público o proceso de revisión pública al que está sometido el desarrollo del software y que imprime un gran dinamismo a la corrección de errores; la independencia del proveedor; la mayor protección de los datos personales, al impedir que algunos fabricantes puedan introducir códigos de control remoto en el software elaborado;

el mejor tratamiento de las lenguas minoritarias; y los estándares abiertos, que garantizan la libertad de los usuarios para intercambiar información con todo el mundo, independientemente de la aplicación que utilicen” (s.p)

Otros, como Valverde (2005), señalan que a nivel general, los beneficios como las desventajas son idénticos en ambas líneas (SL y software propietario) pero establece que ciertas situaciones pueden ocurrir con el uso del software libre:

- No tiene garantía proveniente del autor; esto es, el software libre se compra o se adquiere sin garantías explícitas del fabricante, quienes no se responsabilizan de perjuicios económicos en contra del usuario.
- Cualesquiera modificaciones a problemas encontrados requiere de la dedicación de recursos institucionales, así como la adaptación a las necesidades del usuario.
- Los usuarios deben tener conocimientos de programación para modificarlo y adaptarlo al contexto educativo
- Los usuarios deben estar al tanto de modificaciones que se le haga para evitar confusiones.

Ciertamente hemos enfocado esta sección, principalmente al sector educativo, pero se debe resaltar que el impacto de este paradigma conlleva repercusiones en toda nuestra sociedad. ¿Es justo que el software para la medicina y su investigación sea libre?, ¿y en la producción de alimentos?, ¿es justo el beneficio económico desmedido de unos pocos sobre las necesidades mundiales?

2.4 Comunidades de SL en CR

Un elemento notable en el mundo del Software Libre, es la conformación de comunidades dedicadas al uso, defensa, difusión y desarrollo de sus programas. Más aún, la existencia y el crecimiento de este fenómeno tecnológico se deben al continuo ejercicio de las cuatro libertades en amplias y profundas redes de interacción humana.

Por ejemplo, en Costa Rica, existe una organización denominada Red Costarricense de Software Libre (RCSL). Esta organización mostró su capacidad gestiona y logística en el Festival Latinoamericano de Instalación de Software Libre, realizado en la Universidad

de Costa Rica el 26 de abril del 2008. En esta actividad treinta voluntarios de la RCSL instalaron sistemas operativos GNU-Linux en computadoras personales de más de setenta personas, ante una concurrencia de más de trescientos participantes (RCSL, 2008).

Además, tienen dos listas de correos de la Red, una busca mantener a la comunidad de usuarios de la RCSL enlazada, su dirección es usuarios@softwarelibre.cr. La otra busca mantener a esta comunidad informada de las actividades de la organización, su dirección es anuncios@softwarelibre.cr. Esta última es unidireccional, es decir, sólo los administradores pueden enviar mensajes.

A la mencionada red están asociadas las comunidades de la Universidad de Costa Rica, del Instituto Tecnológico y más recientemente de la Universidad Nacional, denominada UNA Alternativa, la cual fue recientemente integrada el pasado viernes 25 de julio (RCSL, 2008b; Comunidad de Software Libre UCR, 2008). Estos organismos funcionan gracias al voluntarioso y desinteresado esfuerzo de profesores, administrativos y estudiantes universitarios que se sienten comprometidos con los principios filosóficos, técnicos y sociales del Software Libre.

Otra comunidad asociada a la RCSL, es Sibü, que busca la difusión de la filosofía GNU con fines didácticos en la zona Atlántica de Costa Rica, especialmente en el cantón de Pococí. Su nombre lo tomaron de Sibö, el dios bribri creador de la tierra y del hombre. Uno de sus grandes logros es el desarrollo de un sistema operativo, Sibü Linux, basado en Ubuntu Hardy Heron, que se caracteriza por ser liviano y de fácil utilización (Comunidad Sibü, 2008).

También, se pueden mencionar la comunidad denominada Grupo de Usuarios de Linux de Costa Rica, que inicialmente nace como un foro de opiniones y que ha evolucionado como un portal de comunicación (Ramírez, 2008). Similarmente, el Loco Team Ubuntu Costa Rica, comunidad que cuenta con un sitio que ofrece chat, blogs, foros y lista de correos (Loco Team, 2008).

2.5 Ubuntu Linux como alternativa de Sistema Operativo

El sistema operativo Ubuntu ha sido desarrollado a partir de una selección de paquetes de Debian, quizás la distribución más prestigiosa de núcleo Linux. Sus desarrolladores son treinta y ocho empleados contratados por la empresa sudafricana Canonical, complementados con una multitud de voluntarios de la comunidad mundial Ubuntu, la cual se siente identificada con el significado de la palabra "ubuntu"; humanidad hacia los demás (en las lenguas de las etnias Zulú y Xhosa). De modo que Ubuntu tiene una gran importancia, tanto a nivel tecnológico como a nivel ideológico.

Aunque está financiado por una empresa privada, esta distribución de núcleo Linux es completamente software libre, y ha sido concebida desde sus inicios para facilitar el desarrollo de distribuciones derivadas, como es el caso de Sibu en Pococí (Costa Rica) y de Guadalinux en Andalucía, España (Muñoz, 2008). El objetivo es que este sistema operativo sea un vínculo de convivencia que conecte a la personas.

Ubuntu goza de características que la hacen conveniente para un educador, entre ellas están:

1. Tiene una avanzada, robusta y actualizada detección de hardware, particularmente es muy eficiente con la frecuencia de los monitores; asimismo, con las impresoras, "llaves mayas", ratones, audífonos, parlantes, WEBCAM, micrófonos, wireless, módems, dispositivos Bluetooth, entre otros.
2. Cuenta con un ciclo de liberación definido: una versión cada seis meses con actualizaciones durante dieciocho meses para cada una. Adicionalmente, a partir de Hardy Heron 8.04, la instalación de nuevas versiones puede hacerse preservando la partición /home con todo su contenido. Éste es el directorio donde los usuarios guardan sus archivos y aplicaciones personales. Además, posee eficientes herramientas para eliminar recursos inservibles u obsoletos.
3. Posee un respetable rango de posibilidades para la personalización, por ejemplos, la apariencia del escritorio, la pantalla de selección del usuario, efectos visuales mediante el Compiz Fusion, entre otros.
4. Gracias a su infraestructura de traducción, honra su compromiso de darle al

usuario el software en su propio lenguaje.

5. Gran compatibilidad con paquetes de otros entornos Linux, como el KDE y el Xfce, gracias a los proyectos Kubuntu y Xubuntu.
6. Compatibilidad con otros sistemas operativos, por ejemplo, algunos paquetes ofimáticos, que trae por defecto o que pueden instalarse, permiten abrir y guardar documentos en formato .doc. También, exportan directamente en formato PDF. Además, el software WINE permite instalar y ejecutar paquetes .exe.
7. Tanto el navegador de archivos, Nautilus, como el navegador WEB, Mozilla Firefox, son herramientas ágiles, amigables, eficientes y poderosas; fácilmente utilizables por usuarios que han migrado de otros sistemas operativos.
8. Excelente soporte para instalación de plugins, el sistema Ubuntu es muy "inteligente" en este sentido y guía al usuario a ir instalando los recursos conforme se van necesitando.
9. En lugar de instalarse con una multitud de paquetes que pueden ser innecesarios para el usuario, ofrece métodos diversos, limpios y eficientes de instalación y desinstalación de software, entre ellos el programa de "añadir y quitar" aplicaciones, el synaptic y la Terminal. De esta forma el usuario puede probar entre varias alternativas de software, y de esta forma elegir el que más le agrada y conviene.
10. Entre las opciones de software libre que pueden instalarse hay varios que ofrecen prestaciones convenientes para la educación, particularmente la educación matemática. Tal es el caso de KPercentage, Kayali, Xabacus, KBruch, Graph Thing, VYM (View Your Mind), Kig, KmPlot, GeoGebra y wxMaxima entre otros. Otros son especialmente útiles para la edición de texto matemático como Kile, TexMaker y Lyx, o el Open Office Fórmulas.
11. El mundo de Ubuntu ofrece diversas y múltiples opciones para realizar una misma tarea. Por ejemplo, considerando únicamente la lista de aplicaciones del repositorio de Ubuntu, para graficar una función podría utilizarse Ygraf, Libnyz, wxMaxima, Euler, Kalgebra, Dia, Octave, entre otros. El usuario tiene la posibilidad de probar y elegir la herramienta que más se ajuste a sus necesidades.

Por otra parte, gran parte de los proyectos de Software Libre son compatibles con Ubuntu y un amplio sector del software propietario. Además, conforme crece el número de sus usuarios de este sistema, crece la demanda a nivel mundial de servicios para esta distribución y con ella crece la presión por drivers y compatibilidad en general, ante los

productores de software y hardware.

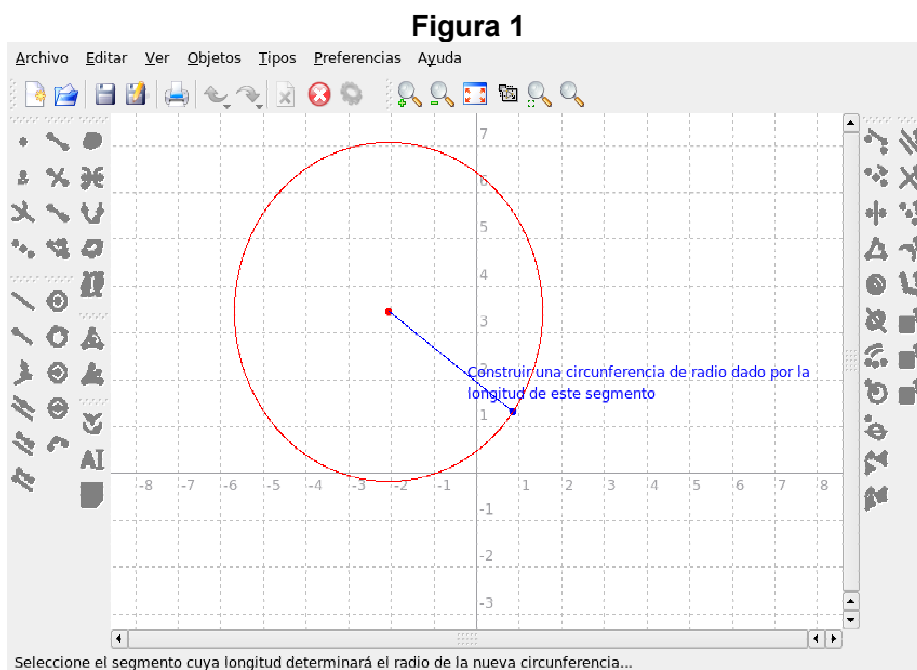
2.6 Software libre en la educación matemática

A continuación se presentan varias aplicaciones basadas en la filosofía de software libre.

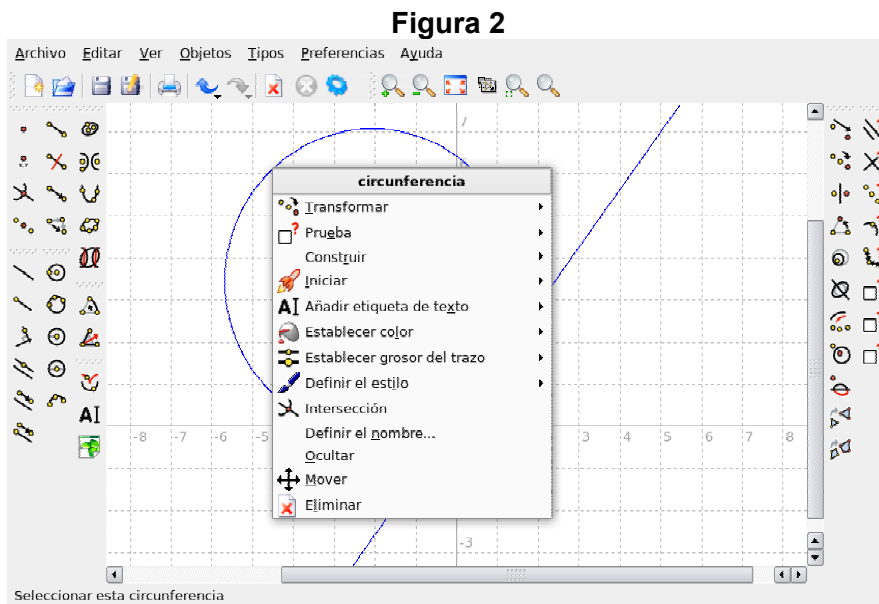
2.6.1 La aplicación Kig

Kig (KDE Interactive Geometry) es un software de licencia GNU- General Public License que puede utilizarse para el estudio interactivo de figuras geométricas y sus relaciones, guardando los trabajos en formato .kig, el cual puede ser incorporado en otros programas de la familia KDE. También, puede utilizarse como un editor de figuras WYSIWYG que permite exportar figuras en formatos .xfig y Latex, para incorporalas en otros documentos (Devriese, 2005). Asimismo, con Kig, pueden abrirse archivos en formatos Kgeo y Kseg, y en alguna medida los formatos de los programas Cabri y Dr. Geo. Básicamente, con Kig pueden incorporarse objetos dando clic sobre los iconos y otros objetos, y esto puede dar lugar al planteamiento de problemas geométricos.

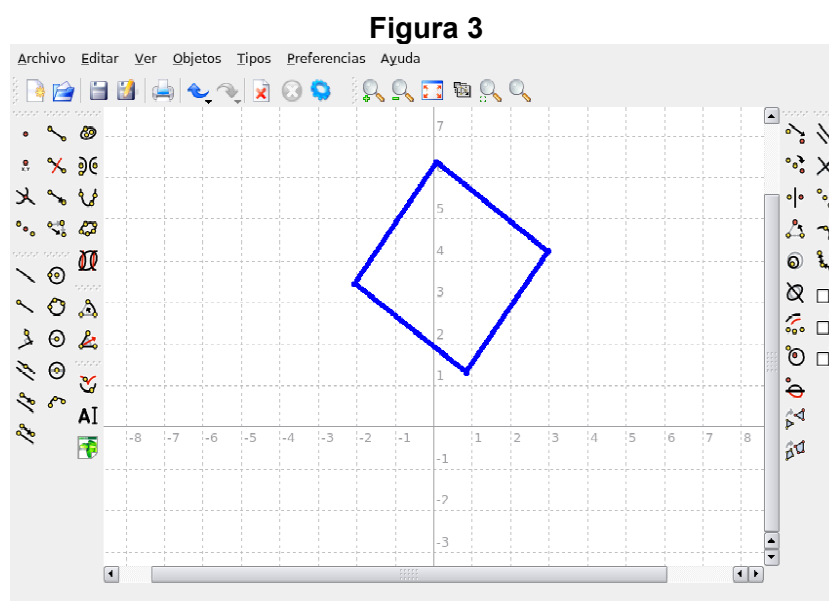
Un ejemplo, puede mostrarse cómo trazar un segmento, mediante el icono respectivo y luego una circunferencia con radio igual a la longitud del segmento (*Figura 1*).



Luego, se puede indicar cómo trazar una recta perpendicular al segmento dado, y luego, a ocultar objetos, mediante la opción derecha del ratón (*figura 2*)



A partir de estos conocimientos puede solicitarse a los alumnos que exploren la forma de trazar un cuadrado, de tal manera que exista la posibilidad de tomarlo de un vértice, con el cursor, y que se pueda cambiar de tamaño y de posición conservando siempre su calidad de cuadrado, como se muestra en la *figura 3*.



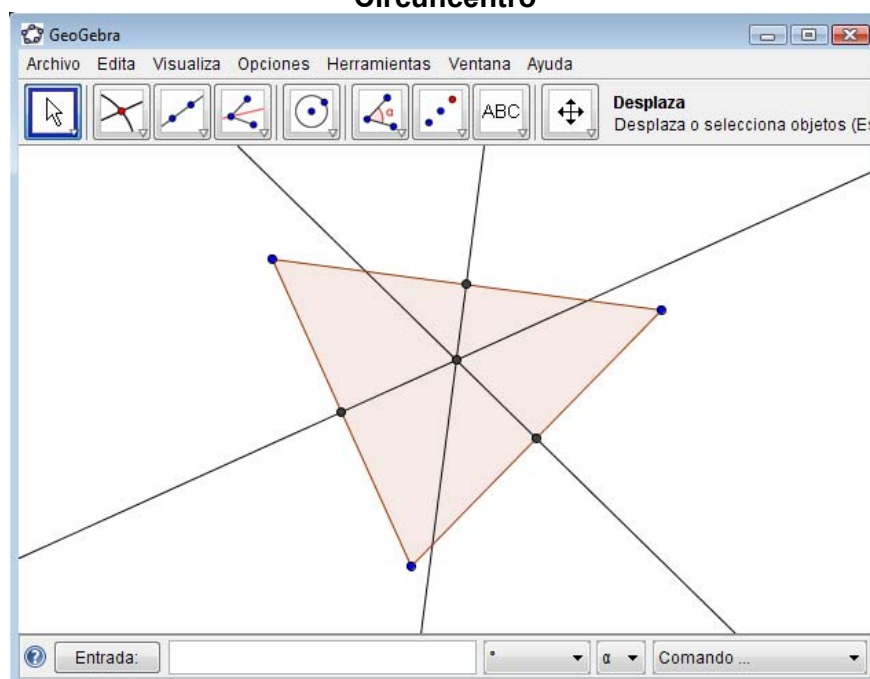
Acciones como seleccionar, mover, eliminar objetos, las opciones de pantalla completa, guardar, imprimir, acercar y alejar y cancelar, son bastante evidentes, igualmente la introducción de etiquetas de texto para las construcciones. También, para los más avanzados, Kig permite la definición de macros y de scripts (objetos a partir de objetos padre, utilizando lenguaje Python).

2.6.2 GeoGebra

GeoGebra es un software de matemática que reúne geometría, álgebra y cálculo. Lo ha desarrollado Markus Hohenwarter en la Universidad Atlantic de Florida (Florida Atlantic University) para la enseñanza de matemática escolar con licencia GPL. GeoGebra relaciona la geometría dinámica y a la vez permite realizar construcciones tanto de objetos y figuras geométricas.

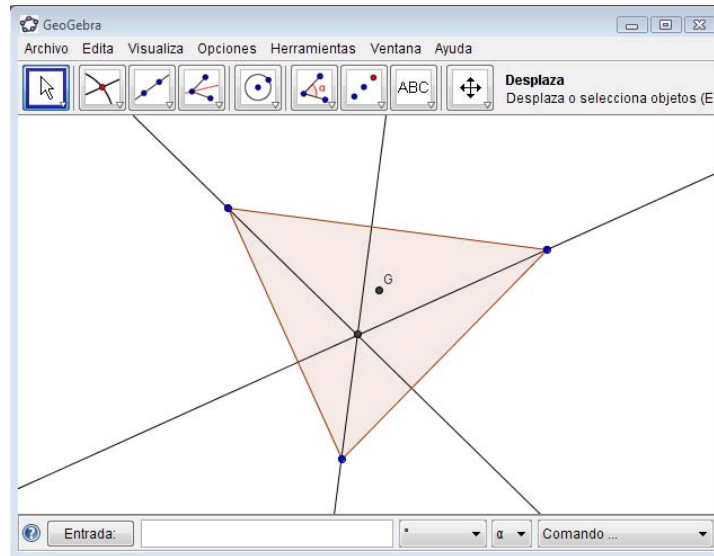
Como ejemplo se puede construir un triángulo para mostrar la relación conocida como la recta de Euler. Primero se encuentra determina el punto medio de los lados y la intersección de las mediatrices (circuncentro), como se muestra en la *figura 4*. Las *figuras 5 y 6* muestran la intersección de las alturas (ortocentro) y la intersección de las medianas (baricentro).

Figura 4
Circuncentro



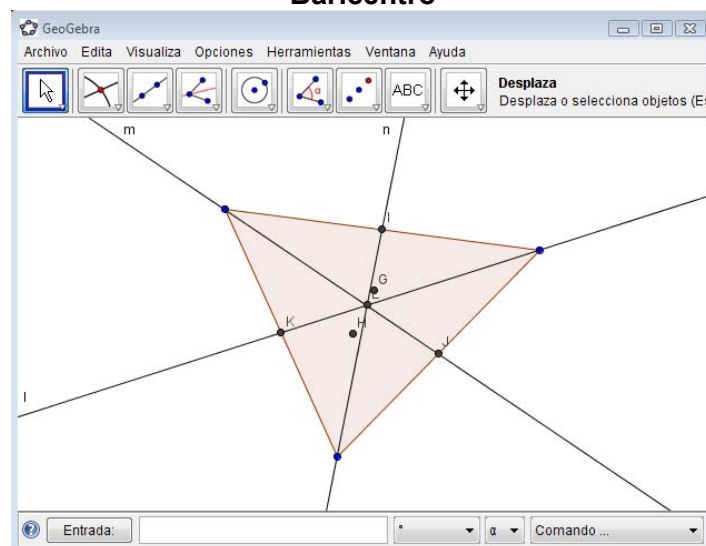
Luego de construir las mediatrices de los lados del triángulo se pueden ocultar las construcciones para dar paso a otras. Un elemento importante es que estas variantes pueden ser aplicadas fijando cierta cantidad de objetos.

**Figura 5
Ortocentro**



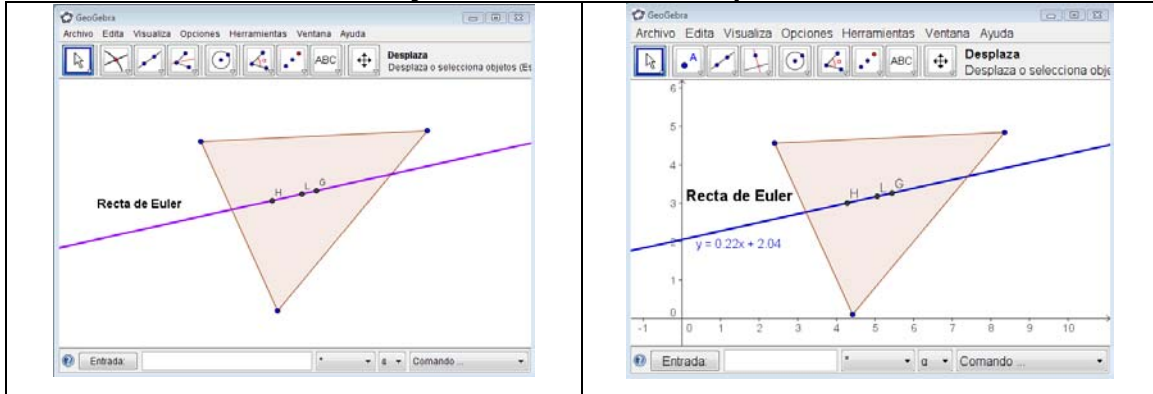
Seguidamente pueden ocultarse las construcciones anteriores y construir el baricentro. Si en algún momento se desea recordar los pasos que se realizaron se puede acceder al módulo de protocolo de construcción en la barra menú

**Figura 6
Baricentro**



La *figura 7* muestra la representación de la recta de Euler y la construcción llevada a un plano coordenado donde se calcula la ecuación de la recta de manera analítica.

Figura7
Recta de Euler y su ecuación sobre el plano cartesiano



2.6.3 XWMaxima

El software libre xwMaxima es un programa tipo CAS (Computer Algebra System); es decir, está diseñado para efectuar operaciones numéricas, realizar cálculos matemáticos simbólicos, y trazar diversos tipos de gráficas.

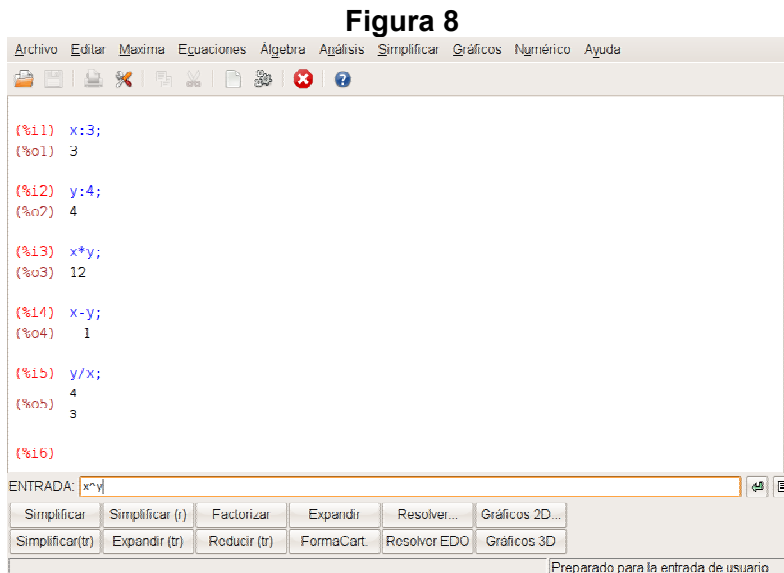
Según Rodríguez (2007), sus orígenes se remontan al año 1967 en el Laboratorio de IA del MIT, con fondos del Departamento de Energía de EE.UU., con el nombre original fue Machine Aided Cognition Symbolic Manipulator.

La versión que el Departamento de Energía le concedió a la empresa Symbolic y que luego esta le entrega a Macsyma Inc. es la que se conoce con el nombre de DOE-Macsyma, y con estas administraciones el proyecto perdió terreno ante dos programas que le precedieron: Maple y Mathematica.

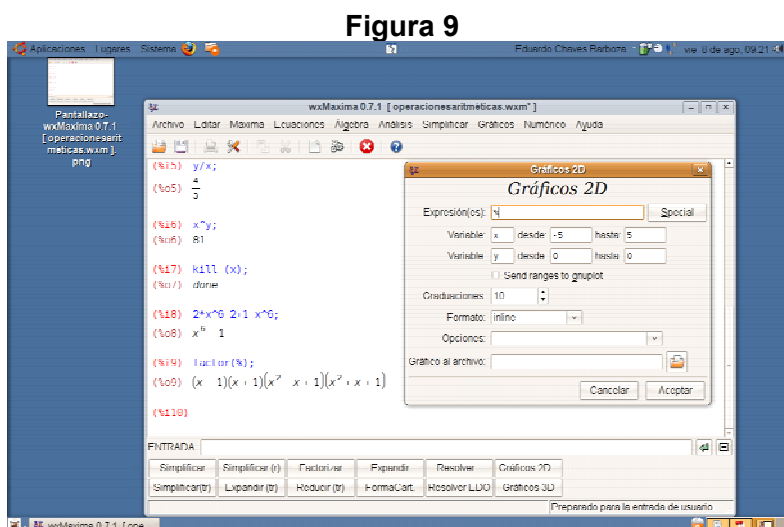
Otra versión fue conservada desde 1982, por William Shelter en la Universidad de Texas, donde se le conoció con el nombre de Maxima. El Departamento de Energía en 1998 permitió distribuir esta versión bajo licencia GNU, y a partir de ese momento el proyecto se ha desarrollado mediante el esfuerzo de una comunidad internacional, actualmente

enlazada mediante la lista de correos. La versión conocida como wxMaxima fue desarrollada inicialmente por Andrej Vodopivec, y se caracteriza por tener una interfase muy amigable.

Por ejemplo, tiene un espacio de entrada en la que puede realizar operaciones numéricas o simbólicas. Como muestra la siguiente figura (figura 8)

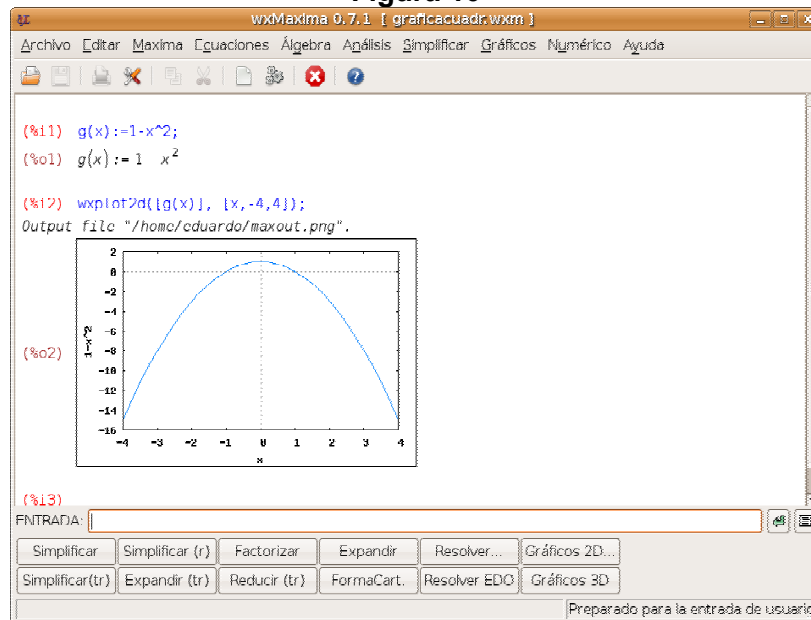


Las pestañas que tiene integradas hacen que algunas operaciones matemáticas sean muy sencillas, tal es caso de la factorización y de la graficación (figura 9).



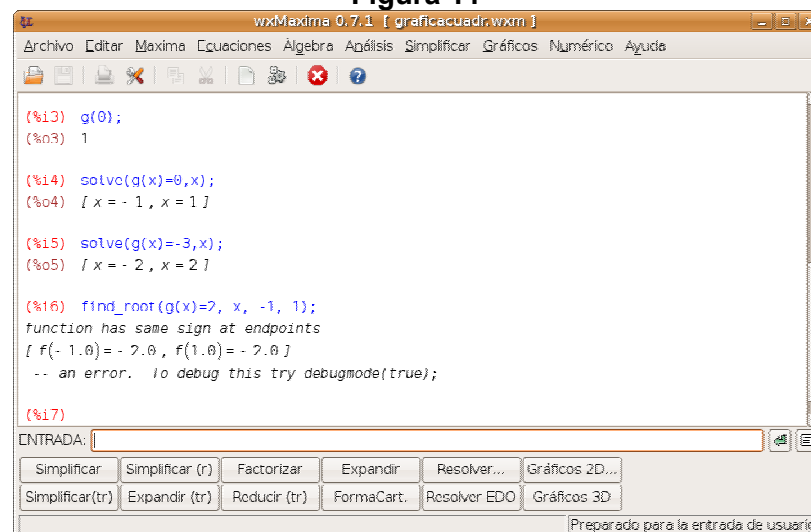
En efecto, con las prestaciones de wxMáxima se pueden desarrollar objetivos de la teoría de funciones de una forma muy analítica. Por ejemplo, analizarse una gráfica mediante una discusión de los resultados obtenidos con el programa. Primeramente trazamos la gráfica de la función g (figura 10).

Figura 10



Luego puede realizarse algunas operaciones relativas a algunas de sus imágenes y de sus pre – imágenes, e iniciar con los estudiantes una discusión generada por los resultados obtenidos con el programa (figura 11).

Figura 11



Mediante preguntas como ¿es posible visualizar en la gráfica que la ecuación $g(x) = 0$ tiene un conjunto solución de dos elementos?, o pueden solicitarse a los alumnos resolver problemas analíticos como “Utilice la gráfica para explicar brevemente por qué la entrada %i6 nos señala un error”.

3. CONCLUSIONES

El Software Libre cada vez se toma más en serio, es notable cómo Neelie Kroes, la Comisionada en Competencia de la Unión Europea, señaló el pasado 11 de junio en una conferencia en Berlín, "Reconozco un negocio inteligente cuando lo veo", al referirse al uso de software de código abierto en instancias gubernamentales y empresariales (Kanter, 2008). Del mismo modo ha reflexionado el diputado Alberto Salom, al defender su proyecto del Software Libre en intensas columnas que se publican en los diarios nacionales.

Particularmente, el educador matemático, debe considerar seriamente el universo de posibilidades que se abre ante ellos. Dentro del Software Libre tiene acceso a herramientas que puede utilizar para elaborar sus documentos. También, existen adecuados paquetes informáticos que le permiten diseñar sus guías didácticas y pruebas matemáticas con la simbología requerida. Asimismo, herramientas poderosas y amigables le pueden ayudar a planificar y desarrollar sus lecciones.

Pero no solamente se trata de una discusión sobre herramientas tecnológicas. Como formador de personas, la alternativa del Software Libre, tiene un enorme potencial ideológico, pues la filosofía del software GNU está saturada de solidaridad, cooperación, humanismo, colaboración y esfuerzo creativo en comunidad.

En este sentido, es deseable que estos valores sean ejemplificados, reflexionados y, más aún, vivenciados por los estudiantes. Por ejemplo, el profesor de matemáticas que trabaje con este tipo de software puede vivenciar la colaboración, al compartir libremente los paquetes con sus alumnos. También, puede motivar el valor del agradecimiento y del humanismo, mientras pone en contacto a sus alumnos con los arquitectos de las

herramientas utilizadas. Asimismo, el profesor y los alumnos, pueden compartir sus experiencias con las comunidades de diseñadores y desarrolladores, como una forma de contribuir, de forma creativa y solidaria, a sus esfuerzos.

Por otra parte, el desarrollo del pensamiento crítico es un compromiso del docente de Matemáticas y éste sólo puede desarrollarse donde hay libertad de criterio; en efecto, desde el escudo moral que brinda la libertad tecnológica, el profesor puede alentar a sus alumnos a analizar críticamente las acusaciones de prácticas monopolísticas realizadas contra Microsoft (Parra, 2006), al tiempo que reflexionan sobre las posibles razones que explican la existencia de la piratería en Latinoamérica, todo ello apoyado con estadísticas tomadas de la realidad mundial (International Data Corporation, 2006).

En conclusión, en el Software Libre, el docente de matemática costarricense tiene una alternativa que debe considerar desde las dimensiones tecnológica, ética y social.

4. BIBLIOGRAFÍA

Comunidad de Software Libre UCR (2008) **Comunidad de Software Libre UCR**. Descargado de [<http://www.softwarelibre.ucr.ac.cr>] en la fecha [1/08/2008]

Comunidad Sibú (2008) Acerca de Sibú. Descargado de [<http://sibu.homelinux.org/dex.p/sibu>] en la fecha [02/06/08]

Delgado, A. M., y Oliver R. (2006). La promoción del uso del software libre por parte de las universidades. RED. *Revista de Educación a Distancia*, número 17 Consultado (23/06/2008) en <http://www.um.es/ead/red/17>

Devriese, D. (2005) **El Manual Kig**. Descargado de [<http://docs.kde.org/stable/es/kdeedu/kig/index.html>] el [10/7/2008]

Free Software Foundation (2008) GNU Operating System. Descargado de [<http://www.gnu.org/home.es.html>] en la fecha [12/07/2008]

González J, Seoane J, Robles G. (2003) *Introducción al software libre*. Fundació per a la Universitat Oberta de Catalunya, España.

Hohenwarter M, Preiner J. (2007) *Documento de Ayuda para GeoGebra, Manual Oficial de la Versión 3.0* Descargado de [www.GeoGebra.org] en la fecha [8/6/2008]

International Data Corporation (2006) Piracy Study. Descargado de [<http://global.bsa.org/>]

cglobalstudy2007] en la fecha [10/06/08]

Kanter, J. (New York Times, 11 jun 2008) **Hard words for Microsoft Technology**. Disponible en [http://www.nytimes.com/2008/06/11/technology/11soft.html?_r=1&emc=tnt&tntemail0=y&oref=slogin]

Loco Team Costa Rica (2008) **Ubuntu-cr**. Descargado de [<http://www.crubuntu.org>] en la fecha [1/08/2008]

Muñoz, J.D. (2008) Programa Educativo Guadalinx. Descargado de [<http://www.josedomingo.org/web/mod/resource/view.php?id=1452>] en la fecha [15/07/08]

Parra, A. (Marzo 2006) *Homo Insipiens*. Begins 1:6

Ramírez, A. (2008) **Linux Costa Rica**. Descargado de [<http://linuxcostarica.org/LCR>] en la fecha [2/08/2008]

RCSL (2008) **CSL-TEC**. Descargado de [<http://wiki.softwarelibrecr.org/index.php/CSL-TEC>] en la fecha [01/08/2008]

RCSL (2008) **Inscríbete e Inscribe a tu Comunidad**. Descargado de [http://www.softwarelibrecr.org/index.php?option=com_frontpage&Itemid=1&limit=4&limitstart=28] en la fecha [30/07/2008]

Rodríguez, R. (2007) Software Libre en el aula de Matemáticas. Oficina de Software Libre de la Universidad de Cádiz. Cádiz: España

Valverde, J (2005). Software Libre, alternativa tecnológica para la educación. Revista *Actualidades Investigativas en Educación*, Vol. 5, número 002, p1-9, UCR