

Vitalinux y software libre para el profesorado de Matemáticas

por

PABLO BELTRÁN-PELLICER y CARLOS RODRÍGUEZ-JASO

(Universidad de Zaragoza; IESO Bárdenas Reales, Cortes de Navarra)

Las Tecnologías de la Información y la Comunicación (TIC) son ya una constante en los programas de formación del profesorado, conformando igualmente un recurso didáctico sobre el que se elaboran propuestas de enseñanza y aprendizaje. Además de la función didáctica y pedagógica, las TIC también están presentes en el día a día de los docentes, facilitando las tareas de gestión, como el cuaderno del profesor o la programación de aula.

En esta comunicación, hemos querido centrarnos en el software libre, presentando una serie de aplicaciones que resultan relevantes para los profesores de matemáticas, aunque varias de ellas son de carácter general. Las que se han seleccionado constituyen solamente una muestra de las posibilidades existentes.

¿Qué es software libre?

La creencia de que el software libre es aquel que es gratuito está muy extendida. Sin embargo, esto no es así en realidad. La confusión tiene su origen, en parte, en que en idioma inglés se utiliza la misma palabra para designar libre y gratis (*free*). Y resulta que libre no es lo mismo que gratis. No vamos a abrir aquí un debate en profundidad. Simplemente, nos limitaremos a señalar que hay diferentes acepciones para el término software libre y algunos de los diferentes tipos de licencias libres que hay.

Un programa o aplicación software será libre, según la Free Software Foundation, si los usuarios tienen las siguientes cuatro libertades <<https://www.gnu.org/philosophy/free-sw.es.html>>:

- La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Como bien hace notar León-Rojas (2005), estas cuatro libertades se han traspasado a lo que se conoce como conocimiento libre, sin más que sustituir el término «programa» por «conocimiento», de forma que se expande la filosofía del movimiento libre a todos los campos del saber.

Sobre la definición de software libre se han elaborado multitud de licencias, de las que mostramos las más extendidas en la tabla 1.

Nos gustaría comentar brevemente el caso de GNU/Linux. Cualquier distribución Linux se asocia al software libre, casi como sinónimo. Siendo puristas, en el momento en el que una distribución utiliza paquetes privativos, como los necesarios para algunas tarjetas gráficas o de audio, ya no podríamos calificarla como *libre*. Es por ello que, por ejemplo, a partir de Ubuntu se desarrolla otra distribución, Trisquel, una distribución libre que restringe la utilización de drivers o paquetes privativos. Esto es algo posible, desde luego, pero ya hay que tener en cuenta el equipo en cuestión y todo el rango de aplicaciones que se pretenden usar. No pretendemos aquí ser los adalides de la ortodoxia. Simplemente, mostrar una alternativa cercana y usable. Pasamos a mostrar el caso de Vitalinux.

	Condiciones más relevantes	Observaciones
 GPL	Libertad de usar, estudiar, compartir (copiar) y modificar el software.	Mediante el copyleft se protege esa creación de intentos de apropiación que restringen las libertades originales a los nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada.
 LGPL	Libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.	La diferencia con la GPL es que la LGPL puede enlazarse o ser utilizada por un programa no-GPL (libre o privativo).
 MPL	Libertad de usar, estudiar, compartir (copiar) y modificar el software.	Cumple con las definiciones de la Open Source Initiative (OSI) y con las cuatro libertades del software libre de la Free Software Foundation (FSF). Como la LGPL, deja abierto el camino a una posible reutilización no libre del software, sin restringir la reutilización ni el relicenciamiento
 Varias	Permite configurar una variedad de licencias mediante conjuntos de condiciones. Las verdaderamente libres son las que incluyen únicamente atribución (BY) y/o compartir igual (SA).	Orientadas a proporcionar cobertura legal no solamente al software, sino a cualquier tipo de creación.

Tabla 1. Algunas de las licencias libres más populares

Linux en educación. El caso de Vitalinux

En el fondo, para la enseñanza y aprendizaje de contenido matemático, nos daría igual utilizar Windows, Linux, iOS o el sistema operativo que nos diera la gana, ya fuera gratuito, privativo, costoso o libre. Si nos limitamos a las matemáticas, lo que nos interesa a fin de cuentas como profesores de esta materia, es tener a nuestra disposición herramientas que podamos usar como recurso para facilitar y promover el aprendizaje. Esa es la prioridad. Ahora bien, la cuestión, para nosotros, surge en el momento que existen alternativas plenamente válidas, desde la perspectiva de la enseñanza de las matemáticas y que además fomentan una escala de valores éticos más elevada.

Solamente por el hecho de utilizar software con licencia gratuita ya estamos facilitando el acceso a la tecnología, puesto que no estamos obligando a nadie a pagar por una licencia que, en realidad, no necesita. Pero es que, si la licencia es libre, estamos evitando el riesgo de que, si la empresa que desarrolla dicho software desaparece, el software también desaparezca.

Desde hace tiempo, diversas administraciones han ido incorporando el software libre en sus instituciones. En el ámbito público resulta una práctica más deseable incluso que en el privado, pues permite una gestión más transparente de los proveedores.

En Aragón, se ha desarrollado Vitalinux EDU (DGA), a partir de los proyectos Vitalinux y Migasfree. ¿Por qué este esfuerzo en preparar una nueva distribución Linux y no utilizar cualquiera de las ya existentes? La respuesta es clara, lo que se busca es facilidad de uso, de instalación y, sobre todo, de gestión. Todo ello, en el entorno propio de los centros de nuestra comunidad autónoma.

De esta forma, puede decirse que Vitalinux EDU (DGA) es un Lubuntu con una serie de aplicaciones predefinidas, entre ellas, el paquete Migasfree. Lubuntu, a su vez, es una distribución de Ubuntu con el escritorio LXDE. La elección de este escritorio hace que Lubuntu sea un sistema operativo ligero, apto para los equipos, normalmente de bajas prestaciones, de los colegios e institutos. Por otro lado, Migasfree es una pequeña aplicación que permite la gestión remota de todo el software de forma desatendida, lo que facilita enormemente la labor del responsable de medios informáticos de cada centro, actualmente conocido en Aragón como MIA. Migasfree, además, permite tener inventariado todo el hardware y el software, todo mediante la aplicación de unas sencillas etiquetas.

Toda la información relativa a la instalación y funcionamiento de Vitalinux EDU (DGA) está disponible en su wiki <<http://wiki.vitalinux.educa.aragon.es/>>. De hecho, los contenidos del curso de formación que se oferta en Aularagon <<http://aularagon.catedu.es/>>, están basados en dicha wiki, que incluye videotutoriales elaborados por los responsables del proyecto. Además, en caso de dudas e incidencias, existe una página de soporte <<http://soporte.vitalinux.educa.aragon.es/>>.

Algunas aplicaciones libres de interés

En esta sección se presentan algunas de las aplicaciones libres más extendidas que resultan de utilidad e interés para el profesor de matemáticas. Obviamente, es imposible cubrir todas las existentes en un documento como este, ya que además de las que nombramos hay muchas más. Incluso, habrá docentes que vean interesantes ciertas aplicaciones que no muestran una relación directa con las matemáticas.

El omnipresente Geogebra

Una de las herramientas más utilizadas en niveles preuniversitarios es Geogebra, cuya licencia para aplicaciones no comerciales es de tipo GNU GPL. Por lo tanto, cualquier utilización o modificación del mismo requiere, simplemente, atribuir la autoría especificando el tipo de licencia. Por motivos de espacio, no profundizaremos en Geogebra, ya que además hay otras comunicaciones dedicadas al mismo tema. Simplemente, apuntaremos que Geogebra se puede clasificar como software de geometría dinámica (Hohenwarter, Hohenwarter y Lavicza, 2009; Hohenwarter y Preiner, 2007), en el que confluyen un procesador geométrico y otro algebraico. Existe muchísima literatura al respecto, así como investigaciones y experiencias didácticas.

Una suite ofimática libre: LibreOffice

Muchas distribuciones de Linux incluyen LibreOffice como la suite ofimática por defecto. No obstante, los usuarios de Windows también pueden instalarlo en sus equipos. Resulta curioso que las trabas de compatibilidad, cuando existen, son casi siempre en el sentido inverso. Es decir, aplicaciones nativas de Windows que no se compilan para otros sistemas operativos. LibreOffice está desarrollado por The Document Foundation y tiene su origen en 2010, a partir de OpenOffice, por lo que su núcleo hereda la licencia de este, de tipo LGPLv3. En realidad, LibreOffice posee una licencia dual, ya que los añadidos vienen con licencia MPL. LibreOffice está compuesta por los siguientes elementos:

- Procesador de textos: Writer.
- Hoja de cálculo: Calc.
- Elaboración de presentaciones: Impress.
- Gestor de bases de datos: Base.
- Editor de gráficos vectoriales: Draw.
- Editor de fórmulas matemáticas: Math.

Todos estos elementos ofrecen las funcionalidades de una suite ofimática estándar, como MS Office o las propias de Google Drive. Es otras palabras, con ellos vamos a poder redactar documentos con el formato que queramos, elaborar complicadas hojas de cálculo con gráficos estadísticos, dibujar y realizar presentaciones, etc. Lo que nos puede preocupar es la compatibilidad entre LibreOffice y MS Office. Aunque es bastante buena, sí que es verdad que al importar determinados documentos, nos podemos encontrar con algún pequeño problema. Como profesores de matemáticas, nos puede interesar utilizar Writer para elaborar fichas o exámenes. Si en MS Word tenemos el editor de ecuaciones, en LibreOffice nos encontramos con uno muy similar, Math, y con el plugin Texmaths, que facilita este trabajo al poder introducir las fórmulas en lenguaje LaTeX.

La hoja de cálculo es, en sí misma, un recurso ampliamente utilizado en las aulas y sobre el que se ha investigado en profundidad. Algunas de estas experiencias didácticas incluyen la resolución (algebraica) de problemas con el apoyo de la hoja de cálculo, como entorno de trabajo (Arnau y Puig, 2013).

Por otro lado, existen en la red multitud de cuadernos del profesor realizados en hoja de cálculo (tanto en Excel, Drive como Libreoffice) y que sustituyen al tradicional cuaderno de papel. Es interesante el realizado en Excel por Antonio J. Calvillo, compatible con LibreOffice, en cuya página podemos descargarlo <<http://www.musikawa.es/cuaderno-del-profesor-en-excel-muy-facil-manual-musikawa>>.

El sistema de composición de textos LaTeX

Tanto el procesador de textos Writer como MS Word son de tipo WYSIWYG. Este acrónimo tan curioso viene del inglés *What You See Is What You Get* (lo que ves es lo que obtienes). Es decir, que lo que ves por pantalla en cada momento es el documento final, lo que saldría por la bandeja de la impresora al darle al botón de imprimir. Pues bien, LaTeX no es WYSIWYG, y lo que se ve en pantalla es más parecido a un programa que a un documento. Sin embargo, los trabajos elaborados en LaTeX presentan una calidad tipográfica muy superior a las suites ofimáticas habituales, así como otras ventajas, como la escritura de fórmulas matemáticas sobre la marcha, maquetación automática o gestión de bibliografía. Eso sí, su curva de aprendizaje es más lenta que para un procesador estándar.

LaTeX requiere de la instalación de un sistema de programas, fuentes tipográficas y paquetes, siendo uno de los más populares TEXLive. Si bien TEXLive puede ejecutarse tanto en Linux como en Windows, en estos últimos sistemas es más popular el entorno MikTeX. A este entorno hay que añadir un editor y, aunque para esta función valdría cualquier editor de ficheros de texto, como Notepad++ <<https://notepad-plus-plus.org>>, hay opciones específicamente diseñadas, como TeXstudio <<http://www.texstudio.org>> y Texmaker <<http://www.xmlmath.net/texmaker>>, que facilitan mucho la tarea.

A medio camino entre un sistema WYSIWYG y LaTeX, nos encontramos con LyX <<https://www.lyx.org>>, que encapsula la complejidad del lenguaje de marcado de LaTeX, mostrando por pantalla un aspecto muy similar al de un procesador de textos tradicional.

Si sentimos curiosidad, siempre podemos realizar un acercamiento al potencial de LaTeX a través de aplicaciones online, como Code Cogs <<http://www.codecogs.com/latex/eqneditor.php>>, con la que podemos escribir ecuaciones para luego copiarlas como imagen e insertarlas en nuestros documentos. LaTeX, junto con MathML es el estándar para escribir fórmulas matemáticas. De hecho, si queremos incluir fórmulas en nuestros blogs o en cualquier página web, podemos hacerlo mediante las librerías libres de MathJax <<https://www.mathjax.org>>.

Nuevos horizontes: pensamiento computacional y programación (Scratch)

La elaboración de programas informáticos con fines educativos ha constituido un foco de interés desde la aparición de las primeras computadoras, incluso mucho antes de la popularización de los ordenadores personales en la década de los 80. En 1967, salió a la luz el lenguaje de programación Logo, diseñado por Danny Bobrow, Wally Feurzeig y Seymour Papert, cuyo conjunto de instrucciones se empleaba, por ejemplo, para dibujar en la pantalla moviendo una tortuga, o para construir programas algo más sofisticados. A los estudios originales de Papert, Watt, diSessa y Weir (1979) les sucedieron diversas propuestas didácticas e investigaciones acerca del impacto que este tipo de tareas tenían en los procesos de resolución de problemas, en los patrones de interacción de los alumnos, etc. (Nastasi, Clements y Battista, 1990; Godino y Batanero, 1986; Pea, 1983). Una de las características de Logo es que su conjunto de instrucciones podía ser traducido a diferentes idiomas, de manera que cualquier niño podía leer el programa, sin necesidad de traducirlo por su cuenta. Hoy en día, siguen existiendo diversos intérpretes libres, como MSWLogo <<http://www.softronix.com/logo.html>> y la biblioteca de instrucciones en español LogoES <<http://neoparaiso.com/logoes>>. Sigue existiendo una comunidad activa de usuarios de Logo.

Poco a poco se fue formalizando la noción de pensamiento computacional. Aunque han ido surgiendo varias definiciones, la más extendida es la que diera Wing (2006):

El pensamiento computacional engloba los procesos de pensamiento implicados en la formulación de problemas y representación de sus soluciones, de forma que dichas soluciones puedan ser procesadas por un agente, bien sea humano o tecnológico. Es lo que precede a cualquier forma de programación, el pensamiento que lleva a cabo una persona conocedora de los mecanismos y de la potencia de cálculo de la automatización.

Scratch <<https://scratch.mit.edu/>> retoma de alguna manera el testigo que dejara Logo. Es un lenguaje de programación visual libre, diseñado en el MIT Media Lab con fines didácticos, fuertemente ligado al desarrollo del pensamiento computacional. Si la característica principal de Logo era que resultaba fácil de leer, en Scratch apenas hay texto. Los bloques lógicos se identifican visualmente, como si de piezas de un puzzle se tratase. De vez

en cuando, aparece algún número para ajustar las propiedades de algún bloque, pero poco más. Ahora bien, permite realizar las más variadas aplicaciones, desde escenas de dibujos animados, hasta videojuegos o enlazar con sensores reales y robots basados en Arduino o en otros sistemas.

SageMath, un peso pesado

SageMath <<http://www.sagemath.org>>, conocido antes sencillamente como Sage, es un sistema de álgebra computacional (CAS) que integra varios paquetes de software libre escritos en diferentes lenguajes con una interfaz común. Algunos de estos paquetes son NumPy y Sympy, de largo recorrido en Python; PARI/GP, un CAS avanzado escrito en C y con su propio lenguaje de script; Maxima, otro CAS especializado en la manipulación simbólica de expresiones; o R, muy conocido para realizar tratamiento estadístico de datos. El proyecto, iniciado por William A. Stein (Stein y Joyner, 2005), se ejecuta de forma nativa sobre plataformas Linux, siendo necesario el empleo de máquinas virtuales para usarlo en otros sistemas, como Windows. No obstante, existe desde hace unos años una implementación de SageMath en la nube, SageMathCloud <<https://cloud.sagemath.com>>.

Decimos que es un peso pesado porque está muy orientado hacia la docencia e investigación universitaria, tanto de matemáticas como de otras disciplinas, ya sean estadística, tratamiento de datos, física o ingeniería. Aunque hay otras alternativas más apropiadas para primaria y secundaria, como hemos visto, SageMath podría resultar interesante, por ejemplo, en bachillerato para la manipulación de matrices o cálculo simbólico. De hecho, SageMathCloud incluye, además de hojas de trabajo interactivas, editor de LaTeX y un terminal Linux, un sistema de gestión de cursos virtuales (Croucher, Stein, Hawke, Jeng y Furnass, 2016). Mediante este gestor de cursos se pueden asignar tareas a los estudiantes y evaluarlas de diferentes maneras, de forma similar a otras plataformas actuales.

En la página de Sagemath está disponible un tutorial que comienza presentando cómo hacer matemáticas elementales con Sage (Tábara, 2009). En cualquier caso, su utilización en el aula requeriría de un aprendizaje previo del sistema y tener nociones de programación.

Modelado e impresión en 3D

Desde hace unos años, y en consonancia con esa búsqueda de fomentar el pensamiento computacional y establecer relaciones con la tecnología, han ido surgiendo interesantes aplicaciones didácticas en torno al modelado y la impresión en 3D (Eisenberg, 2013, 2008). El ecosistema de la impresión en 3D está repleto de aplicaciones libres, lo cual no es de extrañar, ya que el proyecto RepRap (primera máquina auto-replicante de uso general de la humanidad) nació con la finalidad de ser compartido (Jones, Haufe, Sells, Irvani, Olliver, Palmer y Bowyer, 2011).

Son necesarias aplicaciones de diseño y modelado 3D, como Freecad <<http://www.freecadweb.org>> y Openscad <<http://www.openscad.org>>. Lo interesante de esta última es que permite parametrizar los diseños, dando más juego para crear objetos teniendo presente el pensamiento computacional. Por otro lado, hay una implementación de Openscad, llamada Blockscad <<https://www.blockscad3d.com>>, que permite la utilización de bloques gráficos de programación, al estilo de Scratch, lo que facilita acercar esta tecnología a nuestros estudiantes.

Las tareas que se pueden diseñar con este recurso pueden ser de lo más variado. Más allá de la realización de proyectos interdisciplinares con el área de Tecnología o Informática, pensemos en una actividad concreta. Por ejemplo, el modelado y posterior fabricación de un dado. En el proceso de diseño, los alumnos tendrán que trabajar competencias propias del bloque de geometría, pero al mismo tiempo, tendrán que reflexionar sobre el propio diseño para que sea un buen dado. Posteriormente, una vez impreso, el dado se prueba con series de tiradas largas, enlazando los significados frecuencial y clásico de la probabilidad (Beltrán-Pellicer, en prensa).

Otros programas: Inkscape y Audacity

Como hemos adelantado, es imposible tratar todas las aplicaciones software existentes, libres o no, cuya utilización sea pertinente por el profesor de matemáticas. Por ejemplo, consideremos Inkscape <<https://inkscape.org/es/>>, un programa de dibujo vectorial, en el que los dibujos se guardan en función de sus atributos matemáticos, al

contrario que los mapas de bits, basados en píxeles. La principal diferencia de unos y otros es que los vectoriales pueden ampliarse a voluntad sin que se produzca un deterioro de la calidad, empleando mucho menos espacio en disco. Por otro lado, Inkscape resulta tan sencillo de utilizar como el clásico Paint, pero ofrece muchas otras funcionalidades, pudiendo hacer diagramas y figuras con el grado de sofisticación que deseemos.

De la misma forma, Audacity <<http://www.audacityteam.org>> es un editor de audio libre multiplataforma. Al igual que Inkscape, se trata de una aplicación que conocen nuestros alumnos, normalmente, en las asignaturas de Informática o Tecnología. Con Audacity se podrían diseñar tareas muy sugerentes en el bloque de funciones. Así, se podrían grabar sonidos naturales, silbatos, una pelota rebotando en el suelo, etc., para estudiar la forma de onda y sus propiedades, como el período y la frecuencia.

Conclusiones

Si algo caracteriza al software libre, es que ha sido concebido para compartirse. Su expansión solamente ha sido posible con herramientas que permiten compartir el código de forma eficiente y fluida. No estamos hablando únicamente de código software, sino cualquier tipo de documentación. Un ejemplo de ello es Git, el sistema de control de versiones creado por Linus Torvalds y que han popularizado plataformas como GitHub <<https://github.com>>.

Más allá de los recursos que se han presentado a lo largo de esta comunicación, lo realmente interesante, desde el punto de vista del docente, es cómo pueden emplearse de forma efectiva en nuestras aulas y en las tareas del día a día. Por el mero hecho de utilizar las TIC no significa que se esté innovando en la práctica educativa. Tampoco estamos diciendo que debamos emplear recursos TIC en todas las sesiones. Se trata de un recurso que se añade a los de otro tipo, como los juegos, fragmentos películas y series, lecturas, fotografía matemática, historia de las matemáticas, etc. Todos ellos, además de tratar contenidos propiamente matemáticos, inciden en el plano afectivo, sobre el sistema de creencias del alumnado, fomentando una visión de las matemáticas más cercana a la vida cotidiana.

Las aplicaciones que hemos considerado muestran cómo es posible llevar a cabo una utilización de las TIC mediante el empleo de software libre.

Referencias

- Arnau, D. y Puig, L. (2013). Actuaciones de alumnos instruidos en la resolución algebraica de problemas en el entorno de la hoja de cálculo y su relación con la competencia en el método cartesiano. *Enseñanza de Las Ciencias*, 3, 49–66.
- Beltrán-Pellicer, P. (en prensa). Modelado e impresión 3D como recurso didáctico en el aprendizaje de la probabilidad. *Épsilon*.
- Croucher, M., Stein, W., Hawke, I., Jeng, J. y Furnass, W. (2016). *SageMathCloud Tutorial*. DOI: <http://doi.org/10.5281/zenodo.58304>.
- Eisenberg, M. (2008). Pervasive Fabrication: Making Construction Ubiquitous in Education. *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, 193–198.
- Eisenberg, M. (2013). 3D printing for children: What to build next? *International Journal of Child-Computer Interaction*, 1(1), 7–13.
- Godino, J. D. y Batanero, C. (1986). *Microordenadores en la escuela: una introducción didáctica a los lenguajes Basic y Logo*.
- Hohenwarter, M. y Preiner, J. (2007). Dynamic mathematics with GeoGebra. *The Journal of Online Mathematics and Its Applications*, 7, 1448.
- Hohenwarter, J., Hohenwarter, M. y Lavicza, Z. (2009). Introducing dynamic mathematics software to secondary school teachers: The case of GeoGebra. *Journal of Computers in Mathematics and Science Teaching*, 28(2), 135-146.
- Jones, R., Haufe, P., Sells, E., Iravani, P., Olliver, V., Palmer, C. y Bowyer, A. (2011). RepRap – the replicating rapid prototyper. *Robotica*, 29, 177-191.
- León-Rojas, J. M. (2005). Liberalidad del conocimiento desde la cesión de derechos de propiedad intelectual. *Encuentro Internacional sobre Conocimiento Libre*, pp. 190-200, Junta de Extremadura, Mérida. España.
- Nastasi, B. K., Clements, D. H. y Battista, M. T. (1990). Social-cognitive interactions, motivation, and cognitive growth in logo programming and CAI problem-solving environments. *Journal of Educational Psychology*, 82(1), 150-158.
- Özüsaglam, E. y Tekin, P. P. (2016). Comparison of open source softwares in mathematics education. *Konuralp Journal of Mathematics*, 4(1), 225–238.
- Papert, S., Watt, D., diSessa, A., y Weir, S. (1979). *Final Report of the Brookline LOGO Project. Part III: Profiles of Individual Student's Work*. Cambridge, MA: MIT Artificial Intelligence Laboratory.
- Pea, R. D. (1983). Logo programming and problem solving. *Symposium of the American Educational Research Association*, 150, 1-10.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Stein, W. y Joyner, D. (2005). Sage: System for algebra and geometry experimentation. *ACM SIGSAM Bulletin*, 39(2), 61-64.
- Tábara, J. L. (2009). Matemáticas elementales con Sage. Recuperado de <http://www.sagemath.org>.