

Herramientas de Software Libre para Aplicaciones en Ciencias e Ingeniería

Herrera R. *

* Empresa Pública de Hidrocarburos del Ecuador, Quito, Ecuador
(e-mail: roberto.herrera.lara@gmail.com)

Resumen: *El presente artículo desarrolla una revisión de las principales características y funcionalidades de grupo de herramientas informáticas licenciadas bajo el paradigma de software libre muy populares dentro del mundo universitario y de investigación aplicada. El objetivo central del presente trabajo es brindar una visión actual de las funcionalidades que presentan dichas herramientas, tomando como base artículos publicados anteriormente, en donde se recogen las experiencias de usar estas aplicaciones de software en la enseñanza de cátedras como: cálculo simbólico, métodos numéricos, sistemas de control, procesamiento de señales e imágenes, etc. así como en investigación aplicada y de las nuevas funcionalidades que estas herramientas han adquirido hasta la actualidad. La contribución de este trabajo radica en conjuntar un gran conjunto de herramientas en una sola fuente de literatura, donde se enfatizan sus posibles usos y aplicaciones, ampliando así su alcance y aplicabilidad en comparación a trabajos anteriores. Primeramente, se realiza un análisis de las ventajas y desventajas del software libre frente al software comercial, luego se realiza una descripción resumida de estas herramientas, en base a sus principales características, funcionalidades y a través de ejemplos prácticos de uso. Finalmente en la última parte se presentan las principales conclusiones de la elaboración de este trabajo.*

Palabras clave: *software , scilab, scicos, octave, python, maxima, R.*

Abstract: *This article provides an overview of the main features and functionality of software tools group licensed under the open source paradigm, which are popular in the academic world and applied research. The objective of this paper is to provide a current overview of the main characteristics and functionalities that have these tools, based on previously published articles, where are collected the experiences of using these software applications in teaching subjects as: symbolic computation, numerical methods , control systems, signal and image processing, etc, applied research and the new features that these tools have acquired to date. The contribution of this work is to combine a large set of tools in a single source of literature, which emphasize its possible uses and applications, expanding its scope and applicability compared to previous works. First, an analysis of the advantages and disadvantages of open source software versus commercial software is developed, then a brief description of these tools, based on their main characteristics, functionality and through practical examples of use. Finally the last section presents the main conclusions of the development of this work.*

Keywords:*software , scilab, scicos, octave, python, maxima, R.*

1. INTRODUCCIÓN

El manejo de Software y herramientas informáticas especializadas representa hoy en día un pilar fundamental en la enseñanza universitaria de pregrado, postgrado e investigación científica. Sin embargo, a pesar de la inversión económica realizada en Ecuador, en campos como la educación, ciencia e investigación para la adquisición de paquetes de software por parte de los centros educativos, ins-

titutos y universidades, existen herramientas cuyo costo, las vuelve prácticamente inaccesibles a nivel institucional y peor aún por parte de los estudiantes.

Para eliminar los problemas que representan adquirir la licencia para un sistema operativo o herramienta de software, existen alternativas, el manejo de Software Libre [1, 2, 3, 4, 5, 6, 7]. El Software Libre, a veces mal entendido entre los hispanohablantes como Software Gratis, representa el esfuerzo de varios especialistas por elaborar

y producir herramientas especializadas y de acceso gratuito. Muchas de estas herramientas están orientadas a solucionar problemas de ciencia e ingeniería o a su vez orientadas a aplicaciones de investigación. En este trabajo se presentan de manera resumida las más populares herramientas de software de este tipo, haciendo énfasis en sus características, ventajas frente a sus similares comerciales y sus correspondientes desventajas o limitaciones. El objetivo principal de este trabajo, es servir de guía a profesores y estudiantes para la integración de este tipo de herramientas en sus actividades curriculares [8, 9, 7, 1, 10, 11].

En las siguientes secciones se abordará primeramente las principales ventajas del Software Libre vs el Software Comercial, luego se analizará una a una, una serie de herramientas informáticas populares en el campo universitario de ciencias e ingeniería, así como también en investigación, citando en medida de lo posible ejemplos prácticos de uso de este tipo de herramientas. Finalmente se recogen las principales conclusiones de las experiencias recogidas en la elaboración de este artículo.

2. SOFTWARE LIBRE VS SOFTWARE COMERCIAL

Durante los últimos 20 años, los sistemas operativos, herramientas informáticas especializadas y software en general se han desarrollado enormemente. Este desarrollo ha tenido un gran impacto en la sociedad, dentro de muchos campos tales como el entretenimiento, educación, medicina, biología, genética, matemáticas aplicadas, simulación, ingeniería, etc. [12, 13, 14, 15, 16].

En la actualidad el Software como tal se clasifica en dos grandes grupos: el Software Propietario y el Software Libre. Dentro de esta clasificación, cada uno de estos grupos como poseen sus ventajas y desventajas, dentro de las que, una de las principales desventajas del Software Propietario frente al Software Libre es el costo de adquisición.

El Software Propietario como tal presenta la filosofía, “use de forma fácil”, sistemas operativos tales como Microsoft Windows vienen preinstalados en la mayoría de ordenadores comerciales, herramientas de Omática como Microsoft Office, de igual manera viene preinstalada o es fácil de instalar dando unos cuantos clics. Además de forma general este tipo de Software presenta una serie de características agradables al usuario común tales como interfases gráficas simples e intuitivas de usar, gran compatibilidad de hardware, amplio acceso al mercado de entretenimiento, como son las películas y los juegos de video y soporte técnico por parte de la empresa productora del software. Claro, que así como este tipo de software tiene una serie de ventajas, también posee una serie de desventajas, entre las que se tienen: primeramente, la imposibilidad de realizar copias o instalaciones múltiples, debido a que cada copia esta licenciada y protegida mediante el uso de códigos

de instalación, son muy populares, por lo tanto menos seguras, como es el caso de la mayoría de software de Microsoft, victimas a diario de hacking y cracking, existe un costo de adquisición que puede ser representativo dependiendo de la herramienta a adquirir y por último, es incompatible con versiones anteriores o en el peor de los casos salen del mercado volviéndose caducas como ha sido el caso de los diferentes sistemas operativos anteriores a Windows 7.

Del otro lado, el Software Libre, posee un espíritu joven y en constante evolución. Este tipo de software está diseñado para aprender y evoluciona bajo ese paradigma, el aprender y evolucionar continuamente. Las principales ventajas de este tipo de software son: bajo o nulo costo de adquisición, libertad de copiar, modificar, mejorar y redistribuir los resultados en copias nuevas, compatibilidad con versiones anteriores, soporte en foros o grupos de colaboración disponibles en Internet, mayor seguridad y fiabilidad y finalmente, promueve la innovación. A su vez, sus principales desventajas son: la falta o inexistencia de soporte técnico por parte de los autores, interfases gráficas poco amigables, o a su vez están diseñadas para usar en modo texto a través de comandos. Su orientación no comercial ha limitado su acceso al mercado del entretenimiento. En general este tipo de herramientas están orientadas a usuarios con un conocimiento mínimo sobre el manejo de entornos informáticos, lo que limita su uso a grupos determinados como: personal técnico, estudiantes universitarios, matemáticos, físicos, científicos en general, ingenieros, etc.

Pocos son los casos los lineamientos descritos anteriormente no se cumplan. Cleve Barry Moler de la Universidad de Nuevo México, rompe en parte este esquema con Matlab de Mathworks, ya que este paquete de software es una de las contadas excepciones en donde el fabricante del producto permite y motiva compartir código fuente de programación en su foro www.mathworks.com/matlabcentral/fileexchange. De este breve análisis de las ventajas y desventajas del Software Propietario vs el Software Libre se pueden rescatar los siguientes puntos:

- El Software Propietario está orientado al usuario común donde en la mayoría de aplicaciones de software prima la funcionalidad sobre el costo, mas no el conocer cómo éste funciona y de sus posibles correcciones y modificaciones para volver a determinada herramienta más funcional.
- El Software Libre está orientado a determinados grupos de usuarios, sobre los que se exige un conocimiento mínimo del entorno informático donde se usará determinada herramienta de software, su orientación es innovativa, es decir prima el entender cómo funciona y sus posibles usos, sobre la practicidad o facilidad de uso que determinada herramienta tenga.

3. HERRAMIENTAS DE SOFTWARE LIBRE PARA APLICACIONES EN CIENCIAS E INGENIERÍA

Dentro de los Curriculums y Pensums sobre los que se desarrollan Ciencias como Física y Matemáticas Aplicadas, Ingenierías en Mecánica, Electricidad, Electrónica, Informática, Software, entre otras, existen un sin número de cátedras de carácter práctico, que si bien no son exactamente iguales, son similares en concepto y aplicaciones, tales como: calculo simbólico, calculo numérico, simulación de sistemas, teoría de control y automática, procesamiento de datos, etc.

Dichas cátedras exigen la utilización de herramientas que permitan comprobar los principios y leyes que son descritos en teoría. O a su vez, herramientas sobre las cuales, implementar y simular de manera rápida posibles soluciones a problemas, usando las herramientas teóricas descritas en clases. Un ejemplo práctico de esto es, el resolver un problema de maximización, donde la herramienta básica es la diferenciación, ahora bien, una vez que en clases se entiendan y manejen los principios de diferenciación, volver a realizar una derivada se convierte en un problema mecánico, problema en el que posible optimizar el tiempo de cálculo a través del uso de software. Lo importante aquí viene a ser el entender, cómo denir la ecuación o ecuaciones que denan el sistema, saber que diferenciar, lo siguiente puede ser realizado de forma manual a la vieja usanza, o forma computacional, reduciendo el tiempo de solución y mejorando el aprendizaje, debido a la posibilidad de realizar pruebas sobre la solución encontrada.

Herramientas computaciones tales como Scilab, Scicos, Octave, R Commander, Python corriendo en plataformas computacionales como Ubuntu-Linux o en su variante Kubuntu-Linux, debido a su nivel de acesibilidad y costo, representan claras opciones a considerar al momento de implementar herramientas computacionales a cátedras universitarias donde sea necesario el cálculo simbólico o numérico y las simulaciones a gran escala.

A continuación en las siguientes secciones se describen estas y otras herramientas, desde un punto de vista práctico haciendo énfasis en sus posibles aplicaciones y usos.

3.1 Ubuntu y sus versiones

Ubuntu – Linux es una de las distribuciones de la familia de los Sistemas Operativos Linux que han logrado tener gran acogida dentro de los distintos tipos de usuarios. Sin bien en su inicio Linux exigía un cierto nivel de conocimiento para su instalación, en la actualidad Ubuntu dispone de muchas variantes de distribución que se adaptan a las necesidades del usuario. A continuación en la tabla 1 se presentan las principales derivaciones de Ubuntu – Linux y sus principales aplicaciones.

Tabla 1: Ubuntu – Linux y sus derivaciones.

NOMBRES	APLICACIONES	OBSERVACIONES
Ubuntu	Ordenadores de escritorio, portátiles y servidores en distribuciones de 32 y 64 bits.	Gran compatibilidad de hardware y un amplio número de herramientas de software.
Kubuntu	Similar a Ubuntu, en versiones de 32 y 64 bits.	Entorno grafico similar las versiones de Microsoft Windows.
Lubuntu	Disponible en versiones de 32 y 64 bits para ordenadores de escritorio y portátiles.	Está diseñado para trabajar con limitados recursos de RAM y CPU y optimizar el consumo de energía.
Xubuntu	Disponible en versiones de 32 y 64 bits para ordenadores de escritorio y portátiles.	Similar a Lubuntu, usa el entorno grafico Unity optimizando recursos de procesamiento.

En base a la Tabla 1 se puede visualizar plenamente la flexibilidad de los sistemas operativos de la distribución Ubuntu. Haciendo una comparación breve con las versiones comerciales de Sistemas Operativos, se puede concluir que no es necesario poseer un ordenador de última tecnología para trabajar adecuadamente, como es el caso de Windows 7/8. Ubuntu – Linux por lo tanto representa una opción a tener en cuenta en cuanto a sistemas operativos para ordenadores de todo tipo y sin altas exigencias de Hardware.

3.2 Scilab, Xcos, Scicos, Octave y Python

Scilab, Xcos, Scicos, Octave y Python son un grupo de herramientas orientadas al calculo numerico a gran escala, simulaciones, implementación y prueba de sistemas, entre otras aplicaciones.

Scilab disponible en www.scilab.org, es un paquete de software que utiliza lenguaje de programación interpretado de alto nivel, para la realización de cálculo y simulaciones numéricas con aplicaciones en ciencias e ingeniería. Está disponible para plataformas, Linux, Mac OS X y Windows en arquitecturas de 32 y 64 bits. Posee herramientas para la solución de problemas de algebra lineal, matrices dispersas, polinomios, funciones, solución de ecuaciones diferenciales, optimización, estadística y cálculo simbólico básico. Scilab incluye además un entorno de trabajo similar a Simulink de Matlab/Mathworks, llamado Xcos, para la simulación de sistemas dinámicos, circuitos eléctricos, sistemas hidráulicos y análisis y solución de ecuaciones diferenciales. Una variante más completa de este simulador de sistemas ScicosLab, liberado en 2011 y disponible en www.scicoslab.org, mismo que cuenta con un set de herramientas y paquetes extras de simulación, que lo hacen tan funcional como su versión comercial, Simulink. Scilab viene empaquetado en una version básica, la misma que puede ser ampliada usando el Scilab – ATOMS (Auto-matic mOdules Management for Scilab). En la Figura 1,

se muestra un ejemplo de la aplicación del uso de Scilab para el Procesamiento de Imágenes, usando los operadores Sobel, Prewitt y Canny para detectar los bordes de la imagen ejemplo [17, 18].

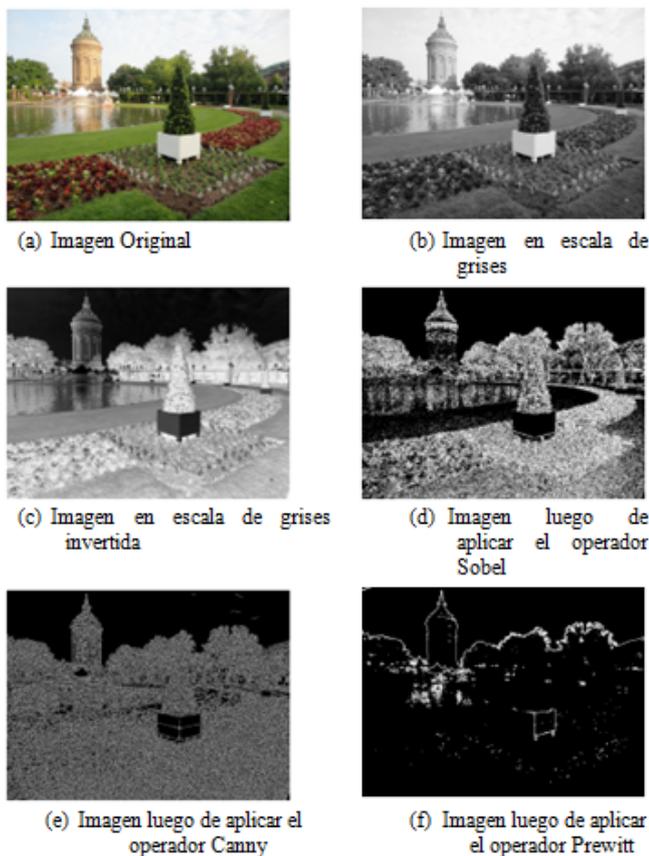


Figura 1: Ejemplo de procesamiento de imágenes usando Scilab y su Toolbox. Imagen: Torre de Agua, Mannheim – Alemania, 2011 (C) Roberto Herrera.

Para conocer más, sobre Scilab y ScicosLab, se puede ingresar a sus sitios oficiales de internet, en www.scilab.org, para Scilab y en www.scicoslab.org, para ScicosLab.

Octave es una herramienta de software libre, para realizar cálculo y análisis numérico, disponible en la dirección de internet www.gnu.org/software/octave/. Su funcionamiento y manejo es similar a Matlab, es decir que se pueden cargar funciones y procedimientos en archivos escritos en Matlab y ejecutarlos en consola, siempre y cuando estos no dependan de toolboxes especializados de Matlab. Una vez instalado el módulo principal, se pueden ir añadiendo módulos adicionales en función de las aplicaciones requeridas. [19].

Python y PythonXY- Scientific-oriented Python son dos herramientas que se han hecho un espacio en el mercado tecnológico y de usuarios de forma muy acelerada y hoy tienen un gran número de adeptos. Usado por organizaciones y empresas tales como: Google, Organización Europea para la Investigación Nuclear (CERN), la Agencia para la Administración Nacional de Aeronáutica y del

Espacio (NASA), Industrial Light Magic (ILM), ITA Software, Yahoo!, Python representa en la actualidad una herramienta revolucionaria dentro del mundo académico y de la investigación aplicada. Como tal Python es un lenguaje de programación interpretado, multiparadigma y que soporta programación orientada a objetos. Fácil de leer e ideal para quienes empiezan a trabajar con el desarrollo de Software y Computación Científica. Funciona en plataformas Windows, Linux y Mac OS X. El consumo de memoria frente a otros lenguajes populares como Java es relativamente mejor y equiparable a C o C++ [30]. En combinación con interfaces gráficas y complementos, Python puede llegar a ser desde una gran calculadora científica a un sustituto de muy buen nivel de Matlab. A continuación en la Tabla 2 se enlistan una serie de complementos y librerías más populares de Python. Una lista más detallada de los complementos y librerías de Python se encuentra disponible en <http://wiki.python.org/moin/NumericAndScientific>.

Tabla 2: Complementos y librerías para aplicaciones científicas en Python

PAQUETE	APLICACIONES
NumPy	Calculo con arrays multidimensionales
SciPy	Complementa a NumPy, en aplicaciones de algebra lineal, optimización, integración, manejo de funciones numéricas, procesamiento de señales e imágenes, estadística, algoritmos genéticos, ecuaciones diferenciales, entre otras.
SymPy	Calculo simbólico (CAS).
APM Python	Programación no lineal y optimización.
ALGLIB	Calculo numérico basado en C++ y C# con interfaces para Python.
Python Data Analysis Library	Análisis numérico.
SpaceFuncs	Modelado geométrico de Funciones N-Dimensionales, calculo paramétrico, optimización numérica y resolución de sistemas de ecuaciones geométricas.
PIL	Procesamiento de Imágenes.
PyDSTool	Modelado de ecuaciones diferenciales ordinarias y ecuaciones diferenciales algebraicas.
mlpy	Aprendizaje Automático.
Matplotlib	Visualización y representación de datos.

A breves rasgos, de lo mostrado en la Tabla 2, se puede visualizar que existen una pseudo - equivalencia entre las aplicaciones de Matlab y sus toolboxes con Python y sus complementos. Claro esta, que esta equivalencia no es total, ni tampoco directa, ya que ambos manejan conceptos diferentes en el desarrollo de sus herramientas. Matlab desarrolla sus toolboxes en función de aplicaciones específicas, mientras que para Python son desarrollados complementos de carácter genérico, que en ciertos casos

comparten funcionalidades, como es el caso de SciPy y ALGLIB. Además, el número de aplicaciones de Matlab es superior a Python, sin embargo Python es una muy buena opción a tener en cuenta para aplicaciones de cálculo simbólico, numérico y simulación de sistemas.

PythonXY es un paquete instalador de las principales librerías y aplicaciones que Python posee, el Kernel de Python, además de complementos para cálculo diferencial e integral, métodos numéricos, estadística descriptiva, algebra lineal, computación paralela, biofísica, visión artificial, entre otras aplicaciones, están disponibles en www.python.org y PythonXY en <https://code.google.com/p/pythonxy/> [28].

3.3 R, PSPP, Weka

PSPP es una aplicación de software libre, para aplicaciones estadísticas y de análisis de datos, descargable desde la dirección web www.gnu.org/software/pspp. Sus principales funcionalidades son el manejo de tablas de frecuencias, regresiones, pruebas no paramétricas, histogramas, distribuciones, etc. Presenta facilidades para la realización de documentación en formatos de como ASCII, PDF, PostScript o HTML [20].

R al igual que PSPP es un paquete de software para aplicaciones estadísticas, que incluye una amplia variedad de herramientas que incluyen el modelamiento lineal y no li, pruebas clásicas de estadística, análisis de series temporales, clasicación, clustering, entre otras [13, 21]. R posee una versatilidad muy grande en cuanto a aplicaciones, ya que se presenta como un lenguaje de programación definido, lo que diversifica su uso. A continuación en el Código de Programa 1, se muestra en pequeño ejemplo de uso de R en el interfaz gráfico R Commander, donde se planea analizar las 10 mediciones realizadas usando un sensor de temperatura.

```
# Vector de Tiempos de medicion
tiempo <- c(1,2,3,4,5,6,7,8,9,10)
# Valores de medición del sensor
datos <- c(81.324, 82.213,83.123, 84.58767,
85.543, 86.345, 89.565, 83.34897,88.34598, 89.213)
# Visualizacion de los datos
print(datos)
# Calculo de los principales parametros
# de los datos
summary(datos)
# Visualizacion de los datos en una grafica
plot(tiempo, datos,type="o", col="blue")
# Agregar leyenda a los valores de las mediciones
legend("topleft", inset=0.05, "datos", pch=1,
col="blue", bty="n")
```

Código de programa 1: Ejemplo de uso de R sobre un problema de estadística descriptiva.

Una vez ingresado el código de programa 1 en el R commander se obtendrán los resultados expuestos en la Figura 2.

```
> # Vector de Tiempos de medicion
> tiempo <- c(1,2,3,4,5,6,7,8,9,10)
> # Valores del sensor
> datos <- c(81.324, 82.213,83.123, 84.58767, 85.543, 86.345, 89.565,
83.34897,88.34598, 89.213)
> # Visualizacion de los datos
> print(datos)
[1] 81.32400 82.21300 83.12300 84.58767 85.54300 86.34500 89.56500
83.34897
[9] 88.34598 89.21300
> # Calculo de los principales parametros
> # de los datos
> summary(datos)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
81.32  83.18   85.07  85.36  87.85  89.56
> # Visualizacion de los datos en una grafica
> plot(tiempo, datos,type="o", col="blue")
> # Agregar leyenda a los valores de las mediciones
> legend("topleft", inset=0.05, "datos", pch=1, col="blue", bty="n")
>
```

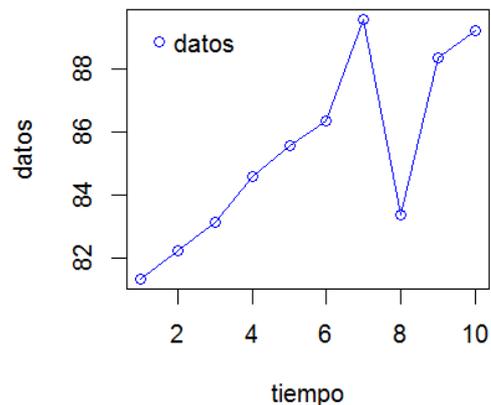


Figura 2: Ejemplo de uso de R en R commander para solucionar un problema de estadística descriptiva.

Weka es una plataforma de software basada en Java, por lo que puede trabajar indistintamente sobre plataformas de Windows, Linux o MacOS. Desarrollada en la Universidad de Waikato para aplicaciones de Minería de Datos y Aprendizaje Automático. Posee una amplia variedad de herramientas para el análisis y visualización de conjuntos de datos, modelado predictivo, clasificación, clustering, regresiones, etc[14].

3.4 Maxima y SAGE

Maxima y Sage son herramientas informáticas para aplicaciones de algebra computacional y cálculo simbólico. Maxima posee un amplio conjunto de funciones para el manejo simbólico de polinomios, matrices, funciones racionales, integración, derivación, manejo de gráficos, cálculo numérico en formato otante, expansión de expresiones, series de fourier, series de taylor, por citar las más importantes, descargable desde maxima.sf.net [23]. A continuación se muestra un ejemplo de aplicación de ésta herramienta computacional. El ejemplo define dos funciones, $f(x) = x^4 + 3x^3 + 2x^2 + x + 1$ y $g(x) = \text{sen}(x^4) + \text{cos}(3x^3)$, se plantea primeramente, dibujar estas dos funciones en un rango definido, de -2 a 2 y luego se pide derivarlas y volver a imprimirlas el mismo rango de valores. Este ejercicio se soluciona mediante el siguiente código.

```

/* Ejemplo de uso de Maxima */
f(x):=x^4+3*x^3+2*x^2+x+1;
g(x):=sin(x^4)+cos(3*x^3);
wxplot2d([f(x),g(x)], [x,-2,2],
[plot_format, gnuplot],
[gnuplot_preamble, "set size ratio 1;
set zeroaxis;"])$
define(df(x),diff(f(x),x));
define(dg(x),diff(g(x),x));
wxplot2d([df(x),dg(x)], [x,-2,2],
[plot_format, gnuplot],
[gnuplot_preamble, "set size ratio 1;
set zeroaxis;"])$

```

Código de programa 2: Solucion en Maxima al problema de ejemplo planteado.

Luego de proporcionarle a Maxima el código de programa 2, éste devolverá los resultados expuestos en la Figura 3. Los resultados expuestos en la Figura 3 pueden ser impresos o a su vez exportados a formatos de edición digital Tex o HTML.

Sage al igual que Maxima, es un sistema algebraico computacional, programado en Python, que presenta funcionalidades similares a las de Maxima, a las que se incluyen, estadística multivariable, procesamiento de imágenes, análisis y visualización de gráficas, manejo de números complejos, aritmética binaria, combinatoria, geometría analítica, geometría algebraica y la posibilidad de usar diferentes interfases gráficas basadas en paquetes comerciales tales como Mathematica o Maple y, documentación directa para exportar en formato Tex . Esta herramienta está disponible en <http://www.sagemath.org/> [30].

3.5 LibreOffice, Latex, MikTex, Texlive

Finalmente se tiene el grupo de LibreOce, Latex, MikTex, Texlive. Todos estos paquetes de software están orientados a la documentacion y levantamiento de textos.

LibreOffice es una suite de paquetes de omática, que ha tenido mejor acogida que el tradicional Open Office. Dispone de paquetes de edición de textos (Writer), otro para hojas de cálculo (Calc), editor de diapositivas (Impress), gestor de base de datos (Base), editor de gráficos vectoriales (Draw) y un editor de fórmulas (Math) matemáticas y está disponible para sistemas operativos como Windows, Mac OS X y Linux, se lo puede descargar desde <http://es.libreoffice.org/>. LibreOffice tiene como característica adicional, la capacidad de leer y modificar formatos comerciales ofimáticos de Microsoft Office, aunque su compatibilidad no es total, brindan esta interoperabilidad en su uso, pudiendo editar documentos ya sea en LibreOffice o Microsoft Office.

Latex, MikTex y Texlive conforman un kit de herramientas para la edición de documentación en lenguaje Tex. Tex es un lenguaje orientado a la composición de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.

```

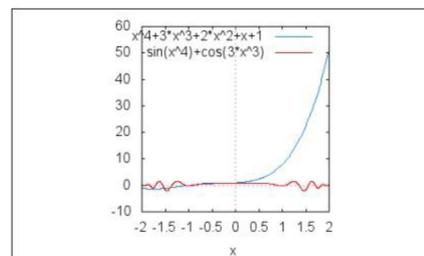
(%i1) /* Ejemplo de uso de Maxima */
f(x):=x^4+3*x^3+2*x^2+x+1;
g(x):=sin(x^4)+cos(3*x^3);
wxplot2d([f(x),g(x)], [x,-2,2],
[plot_format, gnuplot],
[gnuplot_preamble, "set size ratio 1; set zeroaxis;"])$
define(df(x),diff(f(x),x));
define(dg(x),diff(g(x),x));
wxplot2d([df(x),dg(x)], [x,-2,2],
[plot_format, gnuplot],
[gnuplot_preamble, "set size ratio 1; set zeroaxis;"])$

```

(a) Ingreso de comandos a Maxima

(%o1) $f(x) := x^4 + 3x^3 + 2x^2 + x + 1$

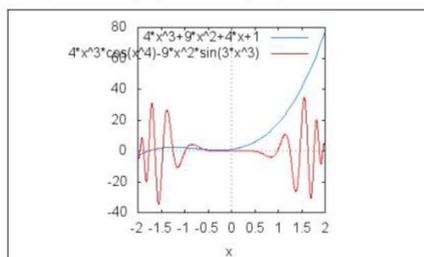
(%o2) $g(x) := \sin(x^4) + (\cos(3x^3))$



(%t3)

(%o4) $df(x) := 4x^3 + 9x^2 + 4x + 1$ (%i5)

(%o5) $dg(x) := 4x^3 \cos(x^4) - 9x^2 \sin(3x^3)$



(%t6)

(b) Procesamiento y salida de resultados en Máxima

Figura 3: Ejemplo de uso de Máxima para un problema básico de cálculo diferencial.

Latex está conformado por un conjunto de paquetes escritos en lenguaje Tex. Latex puede ser escrito en Windows o Linux y los resultados serán los mismos independientemente del sistema operativo. Miktex es una distribución de paquetes Latex para Microsoft Windows. TexLive es el equivalente de Miktex, pero para distribuciones Linux. La edición se hace en texto plano, para mediante un proceso de compilación obtener archivos de salida en formatos PDF, SGML, HTML, RTF, etc. Las versiones descargables de MikTex y de TexLive están disponibles en <http://www.miktex.org/> y <http://www.tug.org/texlive/> respectivamente.

4. CONCLUSIONES

Del presente artículo se recogen las siguientes conclusiones:

El software libre representa una opción a tener en cuenta por parte de estudiantes universitarios, profesores e investigadores, debido a su alto rendimiento y funcionalidades, que en algunos casos esta a la altura de sus equivalente comerciales

Ubuntu-Linux presenta una gran ventaja frente a Microsoft Windows, la adaptabilidad del software sobre la plataforma de Hardware, es decir, no es exigente en cuanto a recursos mínimos de procesamiento como lo es Microsoft Windows. Para esto se puede usar derivaciones de Ubuntu como Lubuntu o Xubuntu.

Scilab, Octave, Python en conjunto con sus complementos presentan funcionalidades equiparables a las del tradicional Matlab para aplicaciones de cálculo numérico, análisis y simulación de sistemas, procesamiento de señales e imágenes, estadística, entre otras.

R, PSPP aplicaciones para el análisis de datos, están altamente desarrolladas, lo que las vuelve equiparables y en casos superiores en cuanto a rendimiento y funcionalidades frente a sus equivalentes comerciales como Excel o SPSS.

Weka es una herramienta orientada al análisis de datos y aprendizaje automático, que sigue fielmente el lineamiento del software libre. Su código está liberado, es escalable y funciona independientemente del sistema operativo, ya sea este Windows o Linux.

Máxima presenta una interfaz muy fácil de usar y de alta potencialidad para realizar cálculos simbólicos.

SAGE es una herramienta potente, que combina el cálculo algebraico computacional con herramientas como Python, Maxima, SciPy, NumPy, entre otras.

LibreOffice, la evolución histórica de OpenOffice permite la edición en formatos comerciales y libres. Presenta funcionalidades similares a Microsoft Office.

Tex es sistema de tipografías, similar a un lenguaje de programación para la edición de macros y paquetes que conforman Latex. Latex está orientado a la edición de textos científicos, posee una gran variedad de herramientas para la escritura de expresiones matemáticas y manejo de gráficos vectorizados.

De manera general el presente trabajo muestra una serie de herramientas computacionales que invitan al usuario a experimentar e innovar, sin limitaciones, por lo cual estas herramientas se han ganado su reputación y acogida.

REFERENCIAS

- [1] P. S. M. Pires and D. A. Rogers, "Free/open source software: An alternative for engineering students," *Frontiers in Education*, 2002. FIE 2002. 32nd Annual, vol. 1, pp. T3G-7 – T3G-11 vol.1, 2002.
- [2] P. Tona, "Teaching process control with scilab and scicos," *American Control Conference*, 2006.
- [3] K. Zaková, "Maxima - an open alternative for engineering education" *Global Engineering Education Conference (EDUCON)*, 2011 IEEE, pp. 1022 – 1025, 2011.
- [4] F. T. Liao Wenjiang, Dong Nanping, "The application of scilab/scicos in the lecture of automatic control theory," *Open-source Software for Scientific Computation (OSSC)*, 2009 IEEE International Workshop on, pp. 85 – 87, 2009.
- [5] Q. W. Tao Zhao, "Application of matlab/simulink and pspice simulation in teaching power electronics and electric drive system," *Electrical Machines and Systems*, 2005. ICEMS 2005. Proceedings of the Eighth International Conference on, vol. 3, pp. 2037 – 2041 Vol. 3, 2005.
- [6] Z. Y. Jin Lihong, Bi Chongrong, "Application of matlab software for linear algebra," *Circuits, Communications and Systems (PAACCS)*, 2011 Third Pacific - Asia Conference on, pp. 1 – 3, 2011.
- [7] G. Lindstrom, "Programming with python," *IT Professional*, vol. 7 , Issue: 5, pp. 10 – 16, 2005.
- [8] S. Y. K. K. Hiroaki Koshikawa, Masataka Kaneko and S. Takato, "Handier use of scilab to draw ne latex figures: Usage of ketpic version for scilab," *Computational Science and Its Applications (ICCSA)*, 2010 International Conference on, pp. 39 – 48, 2010.
- [9] K. Z. Zoltán Magyar, "Using scilab for building of virtual lab," *Information Technology Based Higher Education and Training (ITHET)*, 2010 9th International Conference on, pp. 280 – 283, 2010.
- [10] T. F. Liao Wenjiang, Nanping Dong, "Application of scilab in teaching of engineering numerical computations," *Open-source Software for Scientific Computation (OSSC)*, 2009 IEEE International Workshop on, pp. 88 – 90, 2009.
- [11] M. F. R. O. Jimmy W. Ramírez Cano, "Scilab as a tool to increase learning in courses of communication systems," *Electronics, Robotics and Automotive Mechanics Conference*, 2008. CERMA '08, pp. 201 – 205, 2008.
- [12] B. L. F. Daku and K. Jerrey, "An interactive computer-based tutorial for matlab," in *Frontiers in Education Conference*, 2000. FIE 2000. 30th Annual, vol. 2, 2000, pp. F2D/2–F2D/7 vol.2.
- [13] J. Meys and A. de Vries, *R For Dummies*, ser. –For dummies. Wiley, 2012.
- [14] Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [15] "Organizations Using Python". Python Software Foundation, 2009.
- [16] "Python : the holy grail of programming". CERN Bulletin (CERN Publications) (31/2006).
- [17] Shafer, Daniel G. (17 January 2003). "Python Streamlines Space Shuttle Mission Design". Python Software Foundation.

-
- [18] Fortenberry, Tim (17 January 2003). "Industrial Light Magic Runs on Python". Python Software Foundation. Retrieved 11 February 2012.
- [19] Taft, Darryl K. (5 March 2007). "Python Slithers into Systems". eWeek.com. Ziff Davis Holdings. Retrieved 24 September 2011.
- [20] Z. Dobesova, "Programming language python for data processing," in *Electrical and Control Engineering (ICECE)*, 2011 International Conference on, 2011, pp. 4866–4869.
- [21] E. Anderson, G. Preston, and C. Silva, "Using python for signal processing and visualization," *Computing in science Engineering*, vol. 12, no. 4, pp. 90–95, 2010.
- [22] A. Backer, "Computational physics education with python," *Computing in Science Engineering*, vol. 9, no. 3, pp. 30–33, 2007..
- [23] S. Campbell, J. Chancelier, and R. Nikoukhah, *Modeling And Simulation in Scilab/Scicos*. Springer-Verlag New York Incorporated, 2006.
- [24] C. Gómez, "Engineering and Scientific Computing with Scilab, ser. Engineering and scientific computing with Scilab". Birkhauser Verlag GmbH, 1999.
- [25] J. W. Eaton, D. Bateman, and S. Hauberg, *GNU Octave Manual Version 3*. Network Theory Ltd., 2008.
- [26] C. Halter, *The PSPP Guide: An Introduction to Statistical Analysis*. CreativeMinds Press Group, 2012.
- [27] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.
- [28] L. Prechelt, "An empirical comparison of C, C++, Java, Perl, Python, REXX, and Tcl for a search/string-processing program", *IEEE Computer*, 2000.
- [29] K. J. Millman and M. Aivazis, "Python for scientists and engineers," *Computing in Science Engineering*, vol. 13, no. 2, pp. 9–12, 2011.
- [30] Maxima. (2011) Maxima, a computer algebra system. version 5.25.1. <http://maxima.sourceforge.net/>. [Online]. Available: <http://maxima.sourceforge.net/>
- [31] W. Stein and D. Joyner, "SAGE: system for algebra and geometry experimentation," *SIGSAM Bull.*, vol. 39, no. 2, pp. 61–64, 2005.