

# Lista de herramientas libres para Ingeniería

Diego Essaya

26 de marzo de 2005

## Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Lista de software</b>	<b>1</b>
2.1. Creación de documentos e informes	1
2.2. Edición de imágenes / bitmaps	2
2.3. Gráfica de datos / funciones matemáticas	2
2.4. Sistemas de álgebra y cálculo simbólico	2
2.5. Herramientas/Lenguajes de cálculo numérico	3
2.6. Compiladores/Intérpretes	3
2.7. Entornos de programación (IDEs - Integrated Development Environments)	4
2.8. Debugging	4
2.9. Herramientas auxiliares para programación	4
2.10. Redes	4
2.11. Electrónica	5

## 1. Introducción

En este documento se ha recopilado una lista del software disponible para ingeniería. A menos que se indique lo contrario, todas las herramientas listadas son [software libre](#).

En lugar de usar software pirateado (como muchas veces nos obligan en las mismas cátedras), deberíamos tratar de comenzar a usar software libre, sobre todo en una institución educativa como la Universidad de Buenos Aires. Algunos de estos programas son tanto o más poderosos que los comúnmente utilizados. Casi todos están disponibles para GNU/Linux, y muchos de ellos también para Windows.

La lista de ningún modo está completa. Si encontrás algún programa que no está listado aquí, creés que es de utilidad para ingeniería y además es software libre, por favor avisanos.

## 2. Lista de software

### 2.1. Creación de documentos e informes

[OpenOffice.org](#) Es una suite de oficina inspirada en la de Microsoft. Si estás totalmente acostumbrado a las herramientas de M\$, ésta es tu opción. Además trae el OpenOffice Draw, que es un programa de dibujo vectorial similar al Corel Draw.

Hay muchos otros paquetes similares ([KOffice](#), [SIAG Office](#), [GNOME Office](#)...) y también programas como [Abiword](#) y [Gnumeric](#), todos con sus pros y sus contras.

No digas que no hay para elegir... :)

**LaTeX** ¿Realmente pensás hacer el informe de tu tesis en M\$ Word...?

LaTeX es un lenguaje para crear documentos. El método es bastante similar a la creación de un programa. Tenés que escribir el “código fuente” de tu documento (en cualquier editor de texto plano), y después compilarlo. Tiene una fuerte orientación a las matemáticas, permitiendo escribir fórmulas y símbolos de todo tipo. Además, es totalmente estable.

La curva de aprendizaje es inversa a la de programas como M\$ Word: al principio cuesta arrancar, pero una vez que te sentís cómodo, podés hacer un informe técnico de calidad profesional en poco tiempo.

El formato de los documentos creados en LaTeX es generalmente Postscript (PS) o PDF, pero pueden ser fácilmente convertidos a otros formatos como HTML o incluso texto plano. Este documento fue creado en LaTeX.

**Kile** Sirve para editar archivos fuente de LaTeX, con syntax highlighting y macros para los comandos LaTeX más utilizados. Ayuda muchísimo en el ciclo de editar/compilar/ver resultado.

**LyX** Es un front-end de LaTeX, casi WYSIWYG (What You See Is What You Get), permitiendo crear documentos con la calidad de LaTeX, pero con la facilidad de los procesadores de textos usuales.

**DocBook** Es otro lenguaje de creación de documentos similar a LaTeX, pero basado en XML / SGML.

## 2.2. Edición de imágenes / bitmaps

**The GIMP** Es el programa de edición de bitmaps. Le pisa los talones al Photoshop.

**Dia** Permite dibujar todo tipo de diagramas, grafos, cuadros sinópticos, etc, que pueden ser exportados a Postscript (para incluir, por ejemplo, en un documento LaTeX).

**XFig** Es un programa para crear gráficos vectoriales (es decir, para dibujo general). Las imágenes creadas pueden ser convertidas a muchos formatos usando el programa transfig. Permite incluir fórmulas de LaTeX en los dibujos, por lo que es ideal para acompañar los documentos LaTeX con figuras.

**Inkscape** Un programa de dibujo vectorial, similar al Corel Draw.

## 2.3. Gráfica de datos / funciones matemáticas

**Gnuplot** Puede graficar datos numéricos o funciones matemáticas, en 2D y 3D. Es muy flexible y totalmente configurable, pero para usarlo bien hay que aprender el lenguaje, ya que es un programa manejado mediante línea de comandos.

**Kile** Ya fue mencionado, pero acá está de nuevo, porque trae un front-end gráfico para gnuplot. Significa que podés usar gnuplot sin aprender los comandos.

## 2.4. Sistemas de álgebra y cálculo simbólico

**Maxima** Es un sistema de álgebra bastante completo.

**YACAS** Otro sistema de álgebra, de propósito general, liviano y fácil de usar.

**Pari-GP** Es un paquete para teoría numérica asistida por computadora.

**Giac/XCas** Otro sistema de álgebra, disponible como un programa independiente, o bien como una librería C++.

**Mathomatic** Un sistema de álgebra de propósito general, altamente portable.

**Qalculate!** A primera vista parece una calculadora de escritorio sencilla, pero bajo su interfaz simple e intuitiva se esconde un paquete matemático muy completo y fácil de usar, que permite hacer cálculo numérico y simbólico, conversión de unidades, gráficos (usando gnuplot) y muchas cosas más.

## 2.5. Herramientas/Lenguajes de cálculo numérico

**Octave** Un sistema de cálculo numérico muy completo con un lenguaje casi compatible con Matlab. Utiliza a Gnuplot como back-end para graficar.

**Scilab** Otro paquete de cálculo numérico. Tal vez ofrece más funcionalidades que Octave, pero su lenguaje es menos compatible con Matlab.

**Yorick** Es un lenguaje de programación interpretado, para cálculos y simulaciones numéricas.

**PDL** Perl Data Language añade al lenguaje Perl la habilidad de almacenar y manipular grandes cantidades de datos en arreglos  $n$ -dimensionales, que son el pan y la manteca de la computación científica. La idea es transformar a Perl en un lenguaje numérico similar a Matlab. Se incluye un entorno interactivo de línea de comandos, además del módulo Perl (PDL) para usar en scripts.

**Euler** Es un laboratorio numérico y un lenguaje de programación. El sistema puede manejar números, vectores y matrices reales, complejos y en forma de intervalos.

También existe una versión con interfaz gráfica, [Euler for GTK+](#).

## 2.6. Compiladores/Intérpretes

**gcc** GNU Compiler Collection, (y no GNU C Compiler). Es el compilador de GNU, que contiene front-ends para C, C++, Objective-C, Fortran, Java, y Ada.

Está incluido en cualquier distribución de GNU/Linux (de hecho, casi todo el sistema operativo se compila en gcc).

**Free Pascal** Es un compilador Pascal semánticamente compatible con TP 7.0, así como la mayoría de las versiones de Delphi.

**GNU Pascal** El compilador Pascal de GNU, que utiliza a gcc como back-end. Es compatible con Borland Pascal, y soporta muchas de sus unidades (units).

**Perl** Perl es un lenguaje interpretado de alto nivel, optimizado para procesar archivos de texto arbitrarios, extraer datos de ellos, e imprimir informes basados en esa información.

**Python** Un lenguaje de programación interpretado, interactivo y orientado a objetos, que combina una gran potencia con una alta claridad en la sintaxis. Esta característica hace que Python sea ideal, tanto para el que está dando sus primeros pasos en programación, como para el programador avanzado.

**PHP** Lenguaje de programación interpretado, fuertemente orientado a desarrollo web.

**gprolog** Un intérprete/compilador Prolog, que cumple casi completamente con el estándar ISO Prolog, añadiendo algunas extensiones bastante útiles. Incluye resolución de condiciones sobre dominios finitos (FD), y una potente interfaz bidireccional entre Prolog y C.

**SWI-Prolog** Otro intérprete/compilador Prolog con licencia libre. Incluye recolector de basura (garbage collection) e interfaz C/C++, y ofrece una buena performance general.

## 2.7. Entornos de programación (IDEs - Integrated Development Environments)

**Anjuta** Es un IDE para crear aplicaciones GNOME/Gtk+ con Glade, o simplemente para crear aplicaciones genéricas, en C/C++, o cualquier otro lenguaje.

**KDevelop** Puede generar aplicaciones genéricas, o GUIs usando Qt, KDE o GNOME, en C/C++.

**Kylix** Kylix := Delphi in [Linux]. Es software propietario, pero el Kylix Personal Edition es gratuito.

**ViM / Emacs** Son dos editores de texto plano tan flexibles que pueden utilizarse como entornos de programación para prácticamente cualquier lenguaje.

**Quanta Plus** Es un entorno de desarrollo para HTML con soporte PHP. Está diseñado para el desarrollo rápido de sitios web.

**Eclipse** Un entorno de desarrollo integrado, especialmente orientado a proyectos en Java.

**Mono** Entorno de desarrollo .NET.

## 2.8. Debugging

**gdb** El debugger de GNU. Puede depurar cualquier programa compilado en gcc con la opción -g. Se maneja mediante línea de comandos.

**DDD** Data Display Debugger es un front-end gráfico muy completo para varios debuggers, incluyendo el gdb, y para varios lenguajes (C/C++, Java, Pascal...).

**insight** Es otro front-end gráfico para gdb.

## 2.9. Herramientas auxiliares para programación

**Doxygen** Sistema para generar automáticamente la documentación de programas escritos en C/C++.

**Umbrello** Modelador UML.

**RCS / CVS / Subversion / Darcs** Sistemas de control de versiones.

## 2.10. Redes

**Ethereal** Sniffer y analizador de tráfico de red, gráfico.

**tcpdump** Sniffer y analizador de tráfico de red, de consola.

**Apache** Servidor web.

## 2.11. Electrónica

**Spice** Es la versión original del programa de simulación creado en la Universidad de California, Berkeley. Se maneja mediante línea de comandos, así que no esperes ver un programa de dibujo de esquemáticos con lucecitas y colores.

El circuito a simular debe ser ingresado en un archivo de texto, y el Spice se limita a simularlo y entregarte los resultados. El paquete viene con nutmeg, que es un programa que sirve para interpretar y graficar los datos de la simulación, pero no es demasiado flexible.

Debido a problemas de licenciamiento, el Spice no puede ser distribuido en binarios, así que es necesario bajar el código fuente y compilarlo. Si usás Debian, en:

<http://www.eda.ei.tum.de/mcp/spicedeb/>

se explica cómo crear un paquete .deb a partir del código fuente del spice3f5.

**ngspice** Es un proyecto que tiene como objetivo crear una versión mejorada del spice (con algunos bugs arreglados) y con licencia GPL. El paquete viene con ngnutmeg, que es similar al nutmeg del spice. Sirve para graficar los datos de salida de las simulaciones.

**Gnucap** GNU Circuit Analysis Package. Este simulador no está basado en Spice, y su funcionamiento es algo diferente. La principal ventaja de esto es que se trata de un software enteramente libre. Se incluyen modelos para BJT, MOSFET y diodos.

**Oregano** Es un front-end gráfico para spice, ngspice o Gnucap. Permite dibujar los circuitos con una interfaz amigable y simularlos fácilmente.

Richard Hult, el autor de Oregano, abandonó el proyecto hace algún tiempo, y algunos miembros del LUGFI decidieron continuarlo. Aun está en desarrollo.

**Kicad** Un grupo de programas para la creación de esquemáticos y circuitos impresos. La interfaz es similar a la de Orcad. La documentación está en francés, pero el programa en sí está en inglés.

**Electric** Es un sistema de dibujo asistido, simulación y producción de circuitos eléctricos, muy completo y sofisticado, pero también bastante difícil de usar.

**TkGate** Es un simulador de circuitos digitales, con una interfaz amigable e intuitiva, y varias características interesantes, como un compilador genérico de microcódigo/macrocódigo para crear archivos de inicialización de memorias.

**Xcircuit** Es un programa que permite dibujar circuitos eléctricos (aunque puede ser utilizado también para dibujar otras cosas). Es liviano, fácil de usar y produce resultados con calidad profesional y en formato EPS, por lo que resulta ideal para incluir circuitos en documentos LaTeX. También permite generar archivos Spice para simular el circuito.

**gEDA** GPL Electronic Design Automation. La buena noticia es que GNU/Linux ya cuenta con un sistema integrado de diseño electrónico, que contiene herramientas para creación de esquemáticos, simulación y producción de circuitos, distribuido íntegramente como software libre. La mala noticia es que aun está en fase de desarrollo.

**Programación de microcontroladores** Una lista (no completa) de los distintos ensambladores, compiladores, simuladores, depuradores y programadores de micros:

[sdcc](#), [ASEM-51](#), [uisp](#), [avrp](#), ...