

INSTALACION DE SOFTWARE LIBRE EN WINDOWS: PSICRO, SIMUSOL y SCEPTRE. INTERFAZ GRAFICA PARA SIMUSOL Y SCEPTRE BASADA EN XUL.

Diego Saravia

Facultad de Ciencias Exactas – Universidad Nacional de Salta (U.N.Sa.)

INENCO (U.N.Sa – CONICET)

Fax: 54-387-4255489, Av. Bolivia 5150. 4400 Salta, Argentina

mail: dsa@unsa.edu.ar, Diego.Saravia@gmail.com

RESUMEN: Se presentan los avances obtenidos en la compilación, empaquetamiento, instalación y uso de Software Libre (Free Software Foundation 2010a) en Windows. En particular los casos de Psicro (Saravia, 2010a; Saravia, D. y Lesino, 1992a), y Simusol – Sceptre (Alia et al., 2003 y 2009), que antes sólo estaban disponibles para GNU/Linux. También se presenta una interfaz gráfica local, basada en XUL (Mozilla Foundation, 2010) para Simusol - Sceptre. Se plantean las relaciones del presente trabajo con Summapack (Saravia, D. 2010b).

Palabras clave: Psicro, Sceptre, Simusol, GNU/Linux, Windows, Sumapack, Software Libre, XUL.

INTRODUCCION

En este trabajo se describen los avances en la estandarización del proceso de compilación, empaquetamiento e instalación de Software Libre creado en el INENCO, en particular Psicro y Simusol – Sceptre para su uso en diversos sistemas operativos, distribuciones y arquitecturas. El proyecto **Summapack**, resume los esfuerzos en esta área, pero en éste artículo nos centraremos específicamente en las interfaces gráficas locales y en el uso de Psicro, Sceptre y Simusol en Windows. Summapack se escribe indistintamente con una o dos emes.

ANTECEDENTES

En el INENCO se desarrolla con Software Libre y Gnu/Linux desde los 90, en particular con dgjpp (Delorie, 2003), TeX y LaTeX (Knuth, 1978). Se han venido construyendo metodologías y prácticas a tal efecto desde entonces. Así, por ejemplo, se desarrollaron las primeras versiones de Psicro en “C”, utilizando las Autotools (Wikipedia, 2010b) y liberándolas vía la GPL (Free Software Foundation, 2007) en el 2000 (Saravia D., 2000). Se comenzaron a utilizar repositorios “Subversion” (Apache Software Foundation, 2010), documentación con “Muse” (Olson, 2009) e interfaces gráficas vía web en el 2006: Puyuspa (Saravia, D. 2010a). Se viene desarrollando Simusol y Sceptre desde el 2000, que fuera en particular distribuido mediante Ututo (Saravia, D. y Saravia, L., 2000).

Psicro es un software para realizar cálculos Psicrométricos basados en la Termodinámica del Aire Húmedo. **Simusol** es un software de simulación de sistemas de energía solar. Facilita la descripción de diferentes circuitos físicos y su simulación numérica a partir de diagramas día que traduce a archivos aptos para el Sceptre. **Sceptre** (Novender, 1999) es un software para analizar circuitos, redes y sistemas dinámicos, con capacidad para determinar condiciones iniciales, transitorias y de estado estacionario. Fue originariamente desarrollado por IBM para las Fuerzas Armadas Norteamericanas; liberado y portado a sistemas tipo GNU/Linux por el Prof. Novender. En el INENCO se lo ha mejorado y empaquetado bajo el sistema de las Autotools. Psicro, Simusol y Sceptre son Software Libre.

DEBATES FILOSOFICOS, POLITICOS Y ETICOS

El debate sobre ampliar las libertades.

El Software Libre se define como aquel en el que sus usuarios y desarrolladores tienen las “4 libertades”: conocer, usar, modificar y distribuir el software sin impedimentos legales o contractuales de tipo alguno (Free Software Foundation 2010). Existen varios sistemas legales que en el mundo permiten la existencia de Software que no es Libre, denominado propietario o privativo; software que las personas no pueden modificar o compartir libremente. En particular el sistema de “Copyright” o “Derecho de Autor”, mediante el cual algunos autores, o empresas editoriales deciden distribuir su software con licencias restrictivas.

El sistema legal vigente plantea que es un derecho absoluto del autor o del derecho-habiente (empleador, por ejemplo), el definir “la licencia” de distribución legal del software. El movimiento del Software Libre plantea que los autores tienen derechos sobre su obra, pero no tan extremos, y que los usuarios deben tener también sus derechos.

En las comunidades que impulsan el Software Libre, existen diferentes posiciones (Saravia, 2005, Saravia, 2007) sobre cómo tratar el problema de la convivencia entre el Software Libre y el que no lo es. Algunos vemos sólo como algo transitorio el problema de la convivencia pues aspiramos a que todo el software sea libre. Esta es la posición oficial del movimiento fijada por su fundador Richard Stallman.

“El movimiento de Software Libre busca un cambio social: hacer que todo el software sea libre” y para ello desarrolla software que reemplace al propietario para cada necesidad (Stallman, R. 2010).

Así se pueden plantear dos posiciones filosóficas de fondo: aceptar o no la posibilidad de que los estados instrumenten sistemas que eviten que los usuarios compartan el software si así lo desean sus autores. Y tres posibilidades prácticas. Para quienes apoyan el derecho absoluto del autor la convivencia es algo positivo. Para los que no, hay en principio dos alternativas: la convivencia como mecanismo y camino de apoyo al cambio, y la no convivencia planteando que sólo es ético apoyar el uso de software 100% libre. La cuestión que plantea el objeto de éste artículo es si distribuir o no Software Libre para sistemas no libres. Todo el software cuya creación o modificación se presenta en éste artículo es libre. Simplemente que ahora funciona también en sistemas no libres.

Mi posición es que el sistema actual de copyright da excesivos derechos a los autores, en realidad a las empresas que los contratan, sobre todo en el marco de Internet; que los usuarios y desarrolladores deben tener las 4 libertades para todo el software y que las personas deben poder conocer toda la tecnología que usan. Así pienso que todo el Software debe ser libre. En tal sentido considero que cualquier medida que incremente el porcentaje de instrucciones libres que ejecuta el total de procesadores del planeta es bienvenida. Por lo tanto es positivo desarrollar distribuciones 100% libres, pero también es positivo distribuir software 100% libre que funcione en sistemas no libres, si esto aumenta el uso de Software Libre. En tales condiciones es positivo también instalar mezclas, aunque no es el caso que ahora presentamos. No me parece malo, ni contradictorio, con el objeto de liberar todo el software del planeta, ayudar a que el Software Libre funcione en Windows. Producir sistemas 100% libres es el objetivo central y último, pero es cierto que esto puede acelerarse en determinados casos ayudando a generar usos combinados. Es claro que el usuario es libre de hacer lo que quiera en sus sistemas. No es una cuestión de libertad de elección del usuario. Es una cuestión de qué cosas se permiten legalmente restringir en el marco de las leyes “de autor”, y qué hacemos los que desarrollamos software con ello. Dado que ahora la ley permite a un autor o empresa restringir a sus usuarios o no, decidimos no hacerlo por razones políticas, éticas y de conveniencia. Y decidimos apoyar el uso de Software Libre en Windows porque esto incrementa el uso de Software Libre y nos acerca al objetivo final. De hecho la FSF adopta una postura similar (Free Software Foundation, 2009b).

Es interesante que actividades que plantean alternativas al uso convencional por parte de la humanidad de determinadas tecnologías como en el caso de las energía solar y las que funcionan con Software Libre, puedan confluír y apoyarse mutuamente. Hoy por ejemplo, no es posible pensar, ni remotamente, en el uso exclusivo de fuentes dispersas y poco contaminantes de energía. Por lo tanto plantear una ética que instale como malo y niegue el uso de energías como carbón, petróleo o nuclear no parece sustentable. Pero muchos de los que trabajamos en Energías Alternativas pensamos que sería bueno que poco a poco vayamos llegando a sociedades con mayor uso de estas energías. En el campo del software hoy ya es posible vislumbrar el uso exclusivo de Software Libre. Es sólo una cuestión de decisión política de los estados, e individual de los actores involucrados, y su efectivización está más vinculada a procesos de transferencia y apropiación masiva que de otro tipo.

METODOLOGIAS Y HERRAMIENTAS

Presentamos herramientas para poder compilar, usar e instalar Software Libre en Windows, como también para realizar interfaces gráficas locales en sistemas tipo GNU/Linux y Windows.

Compilado y entorno POSIX

Las herramientas utilizadas anteriormente servían para compilar nuestro software en sistemas del tipo POSIX (Wikipedia, 2010c): Unix, GNU/Linux, BSD, etc.. En particular se usaba Gcc proporcionado por la FSF y el proyecto GNU (Free Software Foundation 2010b). A los efectos de su uso en Windows se utilizan las herramientas de Cygwin (Red Hat, 2008), también basadas en Gcc.

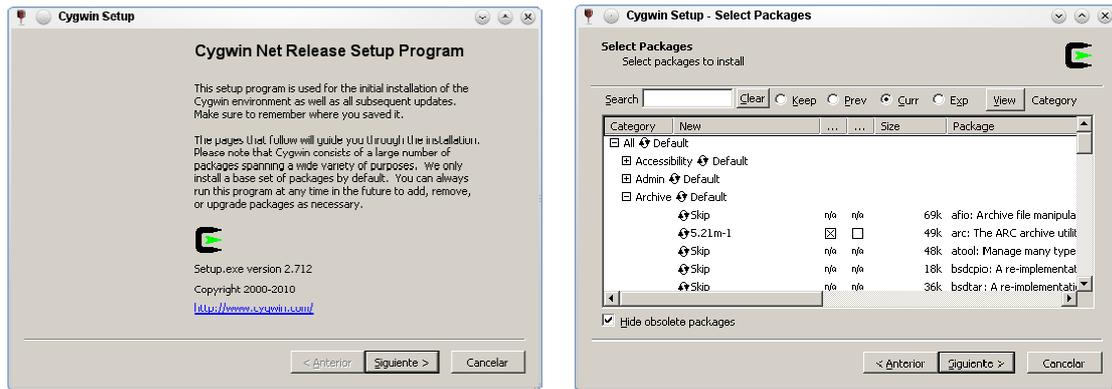
Cygwin aporta las funciones características de los sistemas POSIX a Windows en el marco de un entorno de trabajo consistente en un árbol de directorios similar al de un sistema UNIX y el acceso a una shell Bash. En este trabajo se lo usa para compilar software pensado para GNU/Linux, compilando sus fuentes en el entorno de trabajo de Cygwin que funciona en Windows y que produce ejecutables para ese sistema.

Cygwin provee una biblioteca Cygwin1.dll que instrumenta el API (Application Programming Interface) POSIX en Windows, y otras librerías diversas. Junto con las herramientas Gnu como Awk, Sed, Bash, Cp, y otro software, en un entorno de directorios similar al encontrado en un GNU/Linux. Además podría usarse una librería MinGW que instrumenta el API nativo de Windows (MSVCRT).

CygWin comprende un conjunto de paquetes que ejecutan en Windows una buena parte del software que se encuentra en un GNU/linux. En tal sentido Cygwin puede equipararse a una distribución que funciona sobre un “núcleo” Windows.

Cygwin se instala en Windows a partir de un programa “setup.exe”, que se descarga de Internet (Red Hat, 2008), especificando que paquetes se instalarán. Ver figura 1 y 2. Para compilar Psicro o Sceptre hay que instalar toda la serie de las Autotools, el Gcc, librerías, Make, etc..

A partir de su instalación se trabaja mediante una consola de texto que ejecuta Bash, mediante la cual se pueden instalar los paquetes fuentes a compilar, por ejemplo vía Internet con Wget o con Subversion. Luego se desempaquetan con tar y gzip. Se realizan los cambios que sean necesarios, se autoreconfigura con “autoreconf”, se añaden los archivos faltantes con “automake –add-missing”, se vuelve a ejecutar “autoreconf”, se ejecuta “./configure”, “make” y “make DESTDIR=xxxx install” y se tienen los ejecutables que habrá que incorporar al Nsis (Nullsoft,2009), generador de paquetes para Windows.



Figuras 1 y 2, Instalador de Cygwin (ejecutado desde Wine en este caso).

En tal sentido es más simple para el usuario final producir paquetes con Nsis, que paquetes para el Cygwin,. Posiblemente se crearán paquetes para ser instalados desde dentro del Cygwin cuando Sumapack esté portado a Windows. Otra posibilidad sería crear paquetes tradicionales del Cygwin que se instalan con el aplicativo setup.exe del mismo Cygwin.

Empaquetado para Windows (NSIS y HMNE).

Uno de los campos donde Windows ha quedado muy relegado en relación a GNU/Linux es en el área de los instaladores. No hay metodologías estandarizadas de empaquetado que tengan la riqueza de sistemas como zypper, yast o apt, ni repositorios centralizados para descargar aplicaciones. Cada paquete debe ser empaquetado individualmente con el software apropiado para instalarlo, y debe además instalar sus propios des-instaladores, careciendo de una base de datos central de archivos y paquetes, que pueda estructurar todas las aplicaciones de un computador que use Windows. Mas allá de los avances de sistemas libres creados por Microsoft ,como Wix, Windows esta muy lejos de los sistemas GNU/Linux en este sentido.

Para construir instaladores de los aplicativos ya compilados se usa Nsis, que es totalmente libre y contiene una funcionalidad adecuada para la tarea. Solamente añade 34 Kb a cada paquete. Permite elegir el sistema de compresión utilizado y tiene un editor de configuración independiente denominado “Hm nis edit”. No requiere librerías especiales durante la instalación. Es compatible con diversos Windows. Es basado en scripts, contiene diálogos e interfaces estándares predefinidos. Su configuración se hace con archivos de texto. El generador de instaladores puede ser corrido en sistemas como GNU/Linux y BSD. Existen otras alternativas libres como Inno, pero Nsis es en principio más flexible al utilizar scripts. Se puede también pensar en la posibilidad de utilizar nvin que produce archivos msi de acuerdo a las especificaciones del toolset libre de Microsoft: Wix, pero esto requiere el uso de mucho software específico como el “Windows installer engine”.

Debemos avanzar en la integración de estos procesos con Summapack, a los efectos de contar con un sistema de empaquetamiento moderno para Windows.

Otras programas que es necesario instalar en Windows

En todos los casos se utiliza Software Libre específicamente compilado y empaquetado para Windows: Strawberry Perl (Vanilla Perl Project, 2010), Tail (Perkins, 2002), Dia (Gnome, 2010), NSIS, “Hm nis edit” (Rodriguez Segura, 2005), Apache (Apache Software Foundation, 2009), Emacs (Free Software Foundation, 2008), Tortoise SVN (TortoiseSVN team, 2010), que se descargan desde sus sitios, listados en las referencias y se instalan mediante sus propios instaladores.

Interfaces Gráficas XUL para GNU/Linux y Windows (entre otros), XUL::Gui

XUL es un lenguaje basado en XML (Extensible Markup Language) utilizado para describir y crear interfaces de usuario, que permite desarrollar aplicaciones multi-plataforma sofisticadas o complejas sin necesidad de herramientas especiales.

Con XUL se desarrolla en forma dinámica, sin precompilar la interfaz, a diferencia de los esquemas de programación en los entornos tipo KDE, Gnome o Windows.

Estas aplicaciones son fácilmente personalizadas con textos, gráficos y demás. Pueden ser rápidamente localizadas y adaptadas mediante plantillas CSS.

XUL es una evolución de HTML y puede ser utilizado en su lugar, cuando se requiera desarrollar una interfaz de usuario portable y compleja. A diferencia de HTML, XUL provee un gran conjunto herramientas para crear menús, paneles, barras de herramientas, wizards, entre otras. Gracias a esto, no es necesario utilizar un lenguaje de programación especial o incluir un gran código JavaScript para manejar el comportamiento de la interfaz de usuario.

XUL sólo funciona en los navegadores del tipo Mozilla y Firefox y estos habilitan todas sus tecnologías para el mismo y pueden ser ejecutadas en cualquier sistema donde funcionen (muchos).

XUL::Gui es un módulo Perl que facilita el uso de XUL con este lenguaje. El módulo inicia un servidor propio y el cliente

Firefox. Este módulo expone toda la funcionalidad del motor de “rendering” del Firefox, instrumentando todos los tags XUL y HTML y permitiendo la interacción del programador con estos objetos directamente desde Perl.

Tanto las interfaces al estilo Puyuspa (Saravia, 2010a) como las creadas con XUL utilizan a fondo las ideas de separar el contenido de la forma, ideas originadas en LaTeX y continuadas con el HTML (css) y el XML en general.

El uso de XUL es un avance que permite extender estas ideas a otros ámbitos. Puyuspa genera un archivo HTML que es cargado vía Internet por un navegador. Cuando no tiene sentido plantearse la realización de interfaces remotas pues se producen gran cantidad de archivos locales como resultado, y deben ejecutarse procedimientos en la máquina del usuario (no en el servidor) el modelo utilizado para el Puyuspa basado en un computador que ejecuta una aplicación en un servidor web tipo Apache, y muestra sus resultados en un cliente de red, no es simple de instrumentar. Se debe utilizar un modelo que facilite la ejecución de aplicaciones en la máquina del cliente. Para esto se usa XUL que además provee tags mas ricos que HTML.

RESULTADOS

Compilación de Psicro en Windows

Se instalan las fuentes de Psicro en Cygwin mediante el comando svn del paquete Subversion, o con Wget.

Psicro se configura y compila sin demasiados problemas en Cygwin, hay que instalar entre otros la librería Gsl (Free Software Foundation, 2009a). La única particularidad es que la librería glibc de Cygwin no provee argp (Kögler, 2005) por lo que hay que instalar y compilar el paquete argp-standalone (./configure; make; make install) que contiene software que habitualmente vienen en glibc pero que fue removida de allí en Cygwin.

Debe modificarse el Makefile.am indicando que se debe linkear Psicro con argp.

Desarrollo de Instaladores.

Se han desarrollado instaladores para Psicro y Puyuspa en Windows, los que están disponibles en el servidor web del proyecto. Se trabaja en los instaladores para Sceptre y Simusol. Continúan disponibles los paquetes fuentes y los paquetes para GNU/Linux en sus diferentes versiones, rpm, deb, tbz2, etc..

Para instalar Psicro hay que copiar el ejecutable producido, junto con varios dll de Cygwin, en particular cygwin1.dll, cyggcc_s-1.dll, cygreadline7.dll, cyggsl-0.dll, cygblas-0.dll, cygncurses-9.dll.

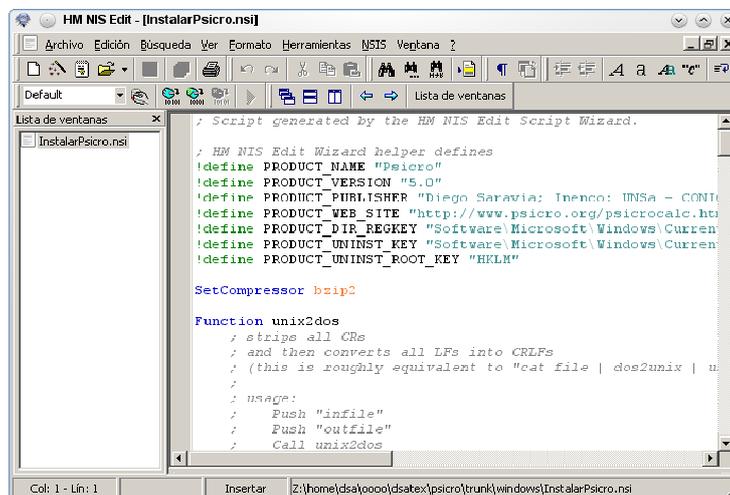


Figura 3: HM NIS edit, creando el instalador de Psicro en este caso desde Wine.

Con el programa “HM NIS edit” se edita el archivo InstalarPsicro.nsi, provisto en el paquete fuente, con el cual se construye el Instalador InstalarPsicro.exe, utilizando Nsis. Ver figura 3.

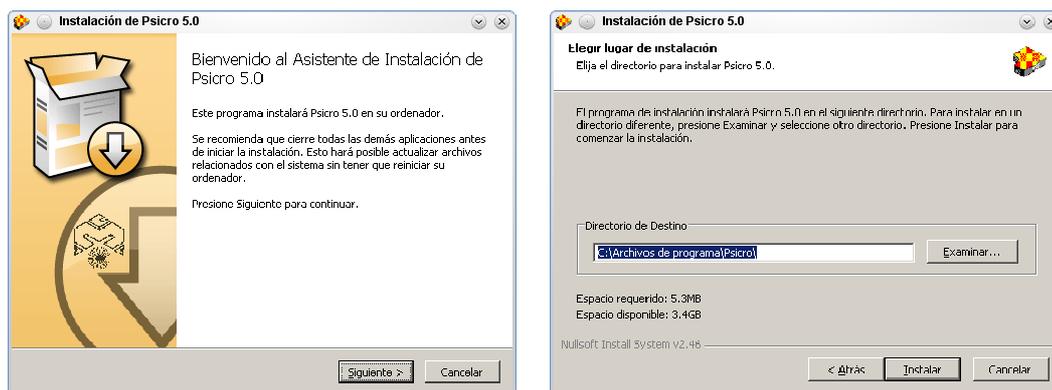
Instalación

Se debe ejecutar el programa InstalarPsicro.exe, creado con Nsis. Ver figuras 4 y 5.

Uso

Para llamar a Psicro en forma directa se puede usar la propia consola del Psicro.exe, que es muy limitada. Un poco mejor es

invocarlo en la consola de Windows o "Símbolo del Sistema". Se usa por ej.: "Psicro -p:Ba:t=30,wr=1,alt=0 ". indicando tres datos independientes para calcular un punto, entre los siguientes tipos: t, x, p, v, h, pv, xs, wr, alt, s, td, tbh, gs, xbh, th w, ws, ha, sa, va (Saravia, 2000; Saravia, 2010a). Mejor todavía es utilizar la interfaz gráfica del Psicro que se describe más abajo.



Figuras 4 y 5: Pantallas del instalador de Psicro para Windows.

No es habitual usar Psicro en forma directa, sino llamado desde otros programas. Para ello puede ejecutar:

```
"Programa 1 | Psicro -pimp > "archivo de resultados",
```

o se captura la salida por parte del programa que lo llama. Así invoca Puyuspa a Psicro, que es la interfaz provista, tanto vía web, como para ser instalada localmente.

Interfaz gráfica del Psicro: Puyuspa

El instalador de Puyuspa para Windows se construye e instala en forma similar a la del Psicro, con la particularidad que no es necesario compilar nada pues funciona en Perl.

Puyuspa requiere el siguiente Software Libre en su Windows: Strawberry Perl, Apache y Firefox. Las instrucciones para instalarlos se encuentran en el mismo instalador de Puyuspa.

Uso

Se apunta el navegador Firefox a <http://localhost/Psicrocalc.html>

También puede usar Puyuspa (Saravia, 2008b), vía web, apuntando su navegador al sitio de la calculadora psicrométrica (Saravia, 2008a; Saravia, 2010a).

Simusol y Sceptre en Windows

Simusol puede funcionar sin problema con Strawberry Perl en Windows y Sceptre puede ser compilado en Cygwin en forma directa. Gnuplot del Windows también muestra los resultados de la misma forma que lo hace con GNU/Linux. Se ha incorporado también un programa para mostrar los mensajes del Simusol y del Sceptre, integrado a la interfaz gráfica basada en XUL.

A los efectos de una mayor usabilidad se ha creado una versión del Sceptre denominada "sceptre-inenco", modificada desde un repositorio Subversion, para poder trabajar comodamente con Sumapack, transformarlo con las Autotools y facilitar la generación de tgz, tbz2, rpms y debs. Se han modificado varios archivos Fortran para mejorar su estabilidad, todo ello en un marco de tests a los efectos de poder comprobar su buen funcionamiento. En algún momento quizás los cambios efectuados puedan ser incorporados al proyecto Sceptre original.

En el momento en que se presenta este artículo se construyen los instaladores basados en Nsis, para Sceptre y Simusol en Windows. No esperamos tener ningún problema en esta tarea habiendo probado que copiando los archivos necesarios es posible ejecutar Sceptre y el compilador de Cygwin para Fortran en Windows.

XUL para Simusol

Se ha desarrollado una interfaz gráfica para Simusol, escrita en Perl y que utiliza el módulo XUL::Gui (Strom, 2009).

Dicha interfaz contiene botones y otros dispositivos para ejecutar las tareas que habitualmente se realizan con Simusol. En particular para: seleccionar un directorio, o proyecto, donde se coloca el archivo del Día del mismo nombre; llamar al Día; procesar con Simusol el archivo .dia; simular con Sceptre interpretando sus resultados; y mostrar los resultados.

Todo ello se controla desde un Makefile. Se tuvo que rediseñar los scripts del Simusol a tal efecto.

La interfaz se ejecuta vía un programa Perl, que mediante el módulo XUL::Gui, ejecuta un servidor web en un puerto aleatorio y simultáneamente llama a un Firefox que automáticamente se conecta con dicho servidor y muestra la interfaz.

El botón “Muestra” invoca un programa que muestra los “logs” de cada uno de los comandos efectuados. Dichos logs se cierran al Reiniciar o al Ocultarlos con sus botones respectivos. También queda un registro somero de las actividades en la pantalla de interfaz, a los efectos de mostrar la interacción con las acciones. Ver figura 6.

Al seleccionar un directorio crea un archivo .dia vacío a los efectos de que el programa Día pueda abrirlo cuando se lo pida, e instala el Makefile,

Se incorporó un sistema para hacer y administrar backups con los resultados de diferentes corridas, con la posibilidad de compararlas.

Se está trabajando en incorporar visores de los principales archivos generados desde el Simusol o el Sceptre.

Todo ello funciona tanto en GNU/Linux como en Windows. Pero para Windows es imprescindible su existencia, salvo que se use directamente la consola Cygwin, lo que esta muy alejado de la experiencia habitual de un usuario Windows. Sería razonable pensar también en desarrollar una serie de botones para el software Día que actúen de la misma forma que esta interfaz. Puede pensarse el presente trabajo como una primera etapa en tal dirección.

La única particularidad que diferencia la interfaz en Windows es que en vez de usar la función “fork” para ejecutar aplicaciones externas en forma no bloqueante, como requiere Día, por ejemplo, se usa la función “Win32::Process::Create”, del módulo Win32 del Perl, y el uso de funciones del tipo “Win32::GetFolderPath”, para encontrar directorios específicos de Windows.

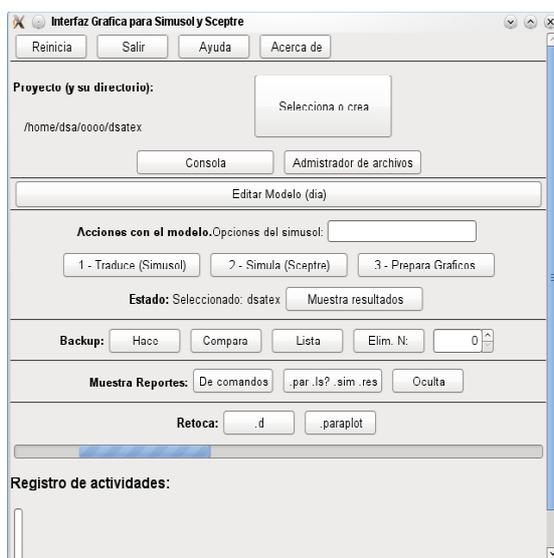


Figura 6: Interfaz gráfica de Simusol

CONCLUSIONES

Se ha logrado portar programas como Psicro, Simusol y Sceptre a plataformas varias, en particular a Windows. Se ha desarrollado software a tal efecto en el marco del proyecto Summapack. Se ha desarrollado una interfaz gráfica para el manejo de Simusol. Todo ello como Software Libre. Se continúan desarrollando los instaladores automatizados para estos programas. Es posible y relativamente sencillo utilizar Software Libre desarrollado con las técnicas y las herramientas libres habituales en entornos GNU/Linux en Windows. Esto permite utilizar estas poderosas técnicas con la confianza de que podrán ser usadas en todo sistema y en tal sentido impulsar el desarrollo de sistemas específicos de las energías alternativas y distribuirlos en forma libre. Todo el software creado es libre bajo la GPL o la AGPL (Free Software Foundation, 2007)

REFERENCIAS

Alía de Saravia, D., Saravia, L. y Saravia, D. (2003). Simusol: Simulating thermal systems using Sceptre and Dia
http://www.fcu.utn.edu.ar/deptos/depto_3/32JAIIO/jsl/JSL_01.pdf

Alía de Saravia, D., Saravia, L. y Saravia, D. (2009). Simusol
<http://www.simusol.org.ar>

- Apache Software Foundation, The (2009). Using Apache with Microsoft Windows
<http://httpd.apache.org/docs/2.0/es/platform/windows.html>
- Apache Software Foundation, The (2010). Apache Subversion
<http://subversion.apache.org/>
- Delorie, DJ (2003). djgpp
<http://www.delorie.com/djgpp>
- Free Software Foundation (2007). Licencia AGPL. Ver también la GPL.
<http://www.gnu.org/licenses/agpl.html>
- Free Software Foundation (2008). GNU Emacs FAQ For MS Windows
<http://www.gnu.org/software/emacs/windows/nemacs.html>
- Free Software Foundation (2009a). GSL - GNU Scientific Library
<http://www.gnu.org/software/gsl>
- Free Software Foundation (2009b). Free Software for Windows
<http://www.gnu.org/software/for-windows.html>
- Free Software Foundation (2010a). Software Libre
<http://www.gnu.org/philosophy/free-sw.es.html>
- Free Software Foundation (2010b). GCC, the GNU Compiler Collection
<http://gcc.gnu.org/>
- Gnome (2010). Dia
<http://live.gnome.org/Dia>
http://dia-installer.de/index_en.html
- Knuth, D. (1978). TeX
<http://www.tug.org>
<http://es.wikipedia.org/wiki/TeX>
LaTeX: <http://es.wikipedia.org/wiki/LaTeX>
- Kögler, M. (2005). Argp-standalone Package.
<http://www.auto.tuwien.ac.at/~mkoegler/eib/argp-standalone-1.3.tar.gz>
- Mozilla Foundation (2010). XUL. Mozilla Development Center
<https://developer.mozilla.org/En/XUL>
- Novender, W. (1999). Sceptre: Simulation of Nonlinear Electrical Circuits Issue 63. July.
<http://www.linuxjournal.com/article/3008>
- Nullsoft (2009). Nullsoft Scriptable Install System (NSIS)
http://nsis.sourceforge.net/Main_Page
- Olson, M. (2009). Emacs Muse.
<http://mwolson.org/projects/EmacsMuse.html>
- Perkins, P. (2002). Tail for Win32
<http://tailforwin32.sourceforge.net>
- Red Hat (2008). Cygwin.
<http://www.cygwin.com>
- Rodriguez Segura, H. M (2005). HM NIS EDIT: A Free NSIS Editor/IDE
<http://hmne.sourceforge.net>
- Saravia, D. y Lesino, G. (1992a). Desarrollo de una calculadora Psicrométrica. Actas de ``15 Reunión Nacional de Energía Solar y fuentes alternativas, ASADES 92 Encuentro de las Nuevas Energías'', Catamarca, 3 al 6 de noviembre. Tomo II, pp. 439 - 446
<http://Psicro.org/calpsi/index.html>
- Saravia, D. y Saravia, L. (2000). Ututo GNU/Linux. Inenco UNSa, ISBN 987-9381-06-8. Disco Compacto.
- Saravia, D. (2000). Calculadora Psicrométrica. Su recodificación y preparación para Internet. AVERMA, Vol 4, N2. pp 08.05.

<http://salta.org.ar/wiki/bin/view/Psicro/PresentacionPsicro?slideshow=on:skin=print#GoSlide1>

- Saravia, D. (2005). Ontología de la libertad del conocimiento y del software libre: luchas y debates
<http://docs.hipatia.net/ontologia/>
- Saravia, D. (2007). Software Apropriado
<http://g.unsa.edu.ar/softlibre/SoftwareApropriado.html>
- Saravia, D. (2008a). Calcula
<http://www.calcula.org>
- Saravia, D. (2008b) Avances en Psicro, La calculadora Psicrométrica. Presentación de Puyuspa y Calcula. AVERMA, Vol. 12, Sección 8, pp79, 20-08
<http://www.cricyt.edu.ar/lahv/asades/averma/2008/20-08.pdf>
- Saravia, D. (2010a). Psicro
<http://www.psicro.org>
Repositorio SVN: <https://www.inenco.net/repos/dsateX/psicro/trunk/>
Programas necesarios en Windows: <http://www.Psicro.org/windows-bin>
Instalador Psicro para Windows: <http://www.Psicro.org/download/InstalarPsicro.exe>
Instalador Puyuspa para Windows: <http://www.Psicro.org/download/InstalarPuyuspa.exe>
Puyuspa: <http://Psicro.org/Psicrocalc.html>
- Saravia, D. (2010b). SumaPack
<http://www.sumapack.org>
- Stallman, R. (2010). Evitando compromisos ruinosos
<http://www.gnu.org/philosophy/compromise.es.html>
- Strom, E. (2009). XUL::Gui - render cross platform gui applications with firefox from Perl
<http://search.cpan.org/~asg/XUL-Gui-0.36/lib/XUL/Gui.pm>
- TortoiseSVN team, The (2010). The coolest Interface to (Sub)Version Control
<http://tortoisesvn.net/>
- Ututo, El proyecto, (2000). Ututo, Un GNU/Linux Simple,
<http://softwarelibre.unsa.edu.ar/ututo/ututo.html>
Actual: <http://www.ututo.org>
- Vanilla Perl Project (2010), Strawberry Perl
<http://strawberryperl.com>
http://win32.perl.org/wiki/index.php?title=Strawberry_Perl
- Wikipedia (2010a). XUL
<http://en.wikipedia.org/wiki/XUL>
- Wikipedia (2010b). GNU build system
http://es.wikipedia.org/wiki/GNU_build_system
- Wikipedia (2010c). POSIX
<http://es.wikipedia.org/wiki/POSIX>

ABSTRACT

This paper shows the progress achieved on the compilation, packaging and installation of Free Software in Windows. In particular, Psicro, Simusol and Sceptre cases are treated. These were only available for GNU/Linux before. It also presents a local XUL-based GUI for Simusol and Sceptre and this work's relations with Summapack project.

Keywords: Psicro, Sceptre, Simusol, GNU/Linux, Windows, Sumapack, Free Software, XUL.