



Interfaz gráfica de usuario PHR GUI v0.1

Desarrollo de la interfaz gráfica

Documento versión 0.9
Abril de 2013

1. Primeras pruebas

Se reconocía la necesidad de facilitar el uso del software `xc3sprog` debido a que éste funciona en un terminal y puede necesitar de varios argumentos). Sobre todo porque la PHR está diseñada para estudiantes que se inician con los lenguajes de descripción de hardware.

Surgió la idea de realizar una interfaz gráfica de usuario (GUI) que sea capaz de correr en sistemas Linux y Windows, y se eligió un lenguaje de programación: Python.

El proceso comenzó haciendo pequeñas pruebas para analizar las opciones disponibles.

1.1. Interfaz gráfica

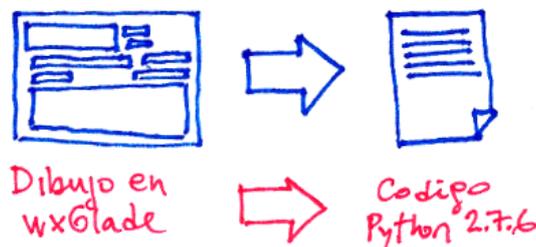
Se buscaba hacer algo aprovechando recursos de licencias libres. En cuanto a la interfaz gráfica se analizaron las siguientes librerías:

1. GTK
2. Qt
3. Tcl/Tk
4. **WxWidgets – Elegida!**

Y se eligió por ésta última por su portabilidad, e integración con librerías nativas de los distintos sistemas operativos.

En el caso de MS Windows hizo falta instalar al interprete python el módulo *wxPython*. Primero se usó la versión 3.0.0 (la última versión) pero luego de un tiempo se detectó un funcionamiento errático con Windows XP, por lo cual se pasó a la versión 2.9.5 y no se tuvieron mas problemas.

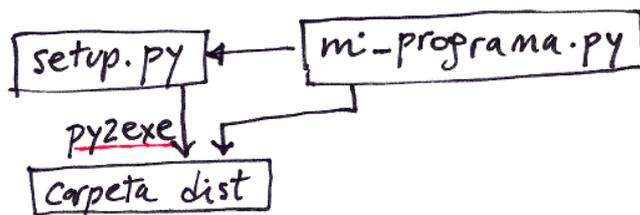
Para el desarrollo con wxPython se encontró el software *wxGlade* que permite un desarrollo rápido de la aplicación (RAD) ya que el usuario dibuja la interfaz y luego la exporta en forma de código de Python. Luego quedan por escribir los handlers para los eventos.



1.2. Independencia del intérprete

Para el caso de la ejecución de un script de Python en GNU/Linux no existen complicaciones puesto que las distribuciones cuentan con el intérprete ya instalado. No ocurre lo mismo con el sistema Windows.

Para no tener que instalar el interprete en cada computadora donde se desee usar PHR GUI se recurrió al módulo de Python denominado *py2exe* que genera archivos de distribución. Este modulo lee el script de interés y lo procesa para generar un directorio (denominado *dist*) en donde ubica un archivo ejecutable (.exe) y librerías asociadas al mismo.



Si bien hay otras alternativas a *py2exe* para compilar código de Python, se eligió por sobre las otras por recomendaciones en la documentación de wxPython¹.

No obstante la tarea efectuada, *py2exe* no resolvió el problema de la distribución por completo. Por usar la librería gráfica wxWidgets, que a su vez usa librerías de Microsoft (y de licencia no libre), el programa no puede correrse en algunas versiones de Windows. La solución es preinstalar un paquete para desarrolladores de Microsoft: *vc9dist_x86.exe*. Según Microsoft²:

The Microsoft Visual C++ 2010 Redistributable Package installs runtime components of Visual C++ Libraries required to run applications developed with Visual C++ on a computer that does not have Visual C++ 2010 installed.

En el próximo paso se busca una solución amigable a la distribución de PHR GUI.

1.3. Instalador

Para generar el instalador se recurrió a NSIS³.

NSIS (Nullsoft Scriptable Install System) is a professional open source system to create Windows installers. It is designed to be as small and flexible as possible and is therefore very suitable for internet distribution.

¹<http://wiki.wxpython.org/DistributingYourApplication>

²<http://www.microsoft.com/en-us/download/details.aspx?id=5555>

³http://nsis.sourceforge.net/Main_Page

Mediante un lenguaje permite describir instaladores que luego son creados por dicha aplicación.

Para generar el script de NSIS se recurrió a:

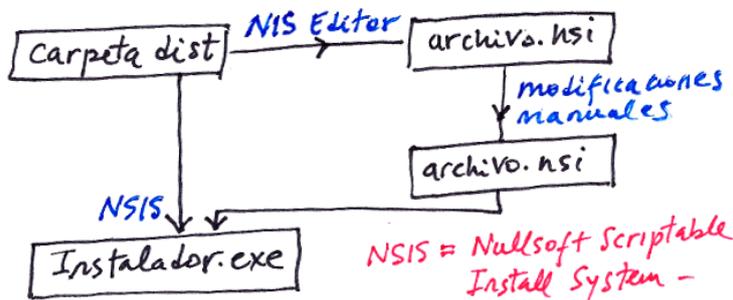
Software *HM NIS Editor 2.0.3* Este software tiene un asistente al que se le puede indicar el directorio de distribución (entre otros datos), para crear el archivo NSIS.

Edición manual del archivo anterior Se realiza la edición manual para añadirle una línea en la sección Post del documento.

```
ExecWait '$INSTDIR\vc8dist_x86.exe /q'
```

Esto instala de forma silenciosa las librerías que el módulo py2exe no incluye al momento de compilar por incompatibilidades de la licencia.

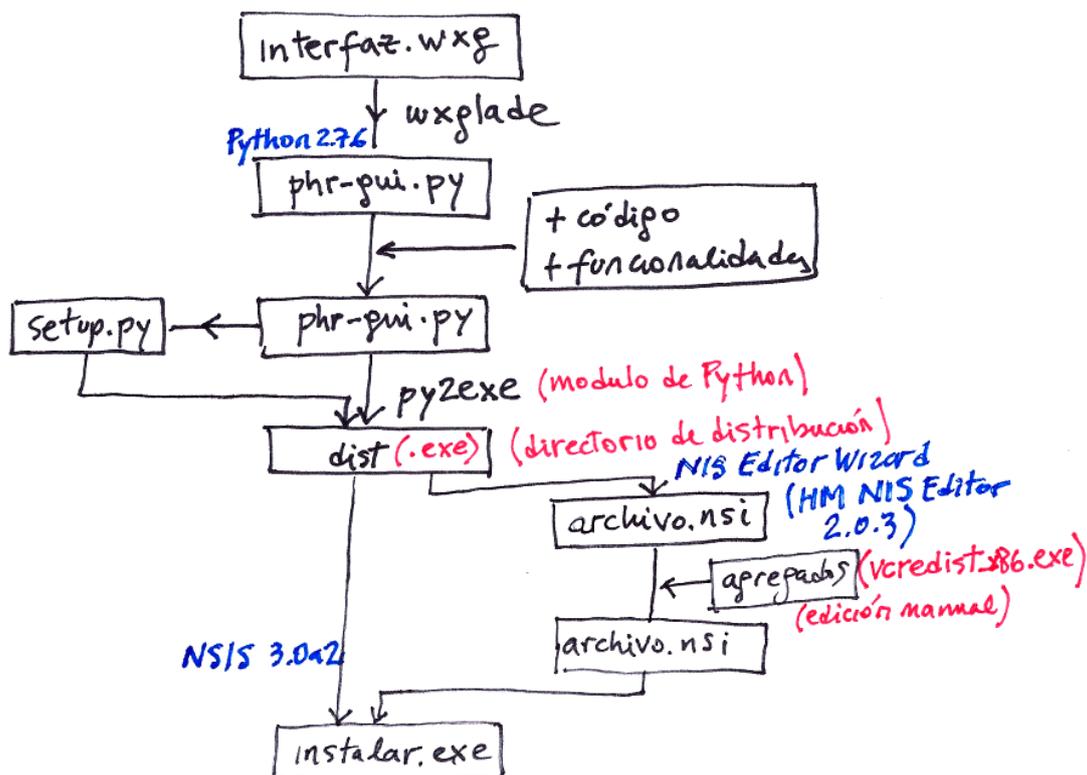
Así se obtuvo el archivo NSIS con el que finalmente se hizo el instalador.



2. Desarrollo de la GUI en Windows

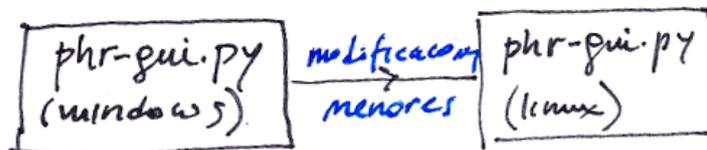
Debido a que Windows presentaba una mayor cantidad requisitos en cuanto a desarrollo y portabilidad se inició el desarrollo de PHR GUI en este sistema operativo.

A continuación un diagrama bosquejando el proceso hasta la obtención del instalador.



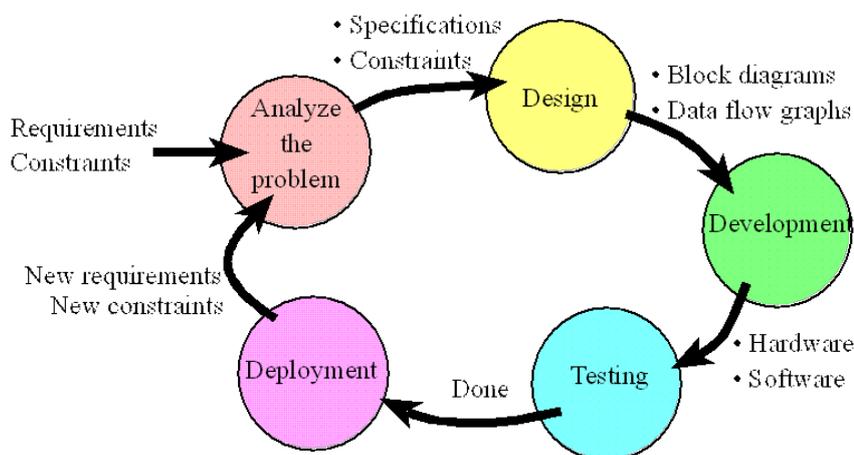
3. PHR GUI v0.1 para GNU/Linux

Para hacer funcionar la GUI en GNU/Linux se hicieron algunas modificaciones al programa desarrollado para Windows. En su mayor parte referidas a la forma en que se invoca el xc3sprog, debido a la diferencia entre el shell de linux y la consola de windows.



4. Futuro

En la vida de cualquier producto, se pasan por distintas etapas y de acuerdo al proceso de diseño, cíclicamente es necesario adaptar el trabajo para cumplir con los objetivos del proyecto y acomodarse a nuevas especificaciones.



De acuerdo a lo expuesto, se prevé una nueva versión de PHR GUI y en el siguiente texto se listan algunas ideas a tener en cuenta en el momento de continuar con el desarrollo.

4.1. TODO

1. Respecto de la interacción con xc3sprog:
 - a) Añadir soporte para programar con archivos MCS además de BIT.
 - b) Solucionar la indicación de los archivos en Windows cuando la ruta incluye dos puntos (':').
 - c) Crear una ventana de configuración en la cual se pueda elegir la ruta al xc3sprog (podría estar en el PATH), y que además permita elegir el comando que se ejecuta para cada evento, haciéndolo compatible para distintas versiones de xc3sprog.
 - d) Dar la opción para elegir librería libre o propietaria.
 - e) Detectar y corroborar un archivo de configuración al inicio (o en cualquier momento).
 - f) Procesar la salida de xc3sprog para brindar mensajes mas claros.
2. Respecto de la usabilidad:
 - a) Mejorar el dialogo de ayuda. Incluir foto explicando el seteo de los jumpers.
 - b) Añadir aceleradores a los botones y hacerlos cambiar entre ellos con la tecla tab.

3. Respecto de la portabilidad:

a) Generar los mismos archivos capaces de correr en Windows y Linux.