

# Introducción al Desarrollo de Aplicaciones Gráficas con PyGTK

Alejandro Valdés Jimenez  
avaldes@gnome.org

November 9, 2010

# Agenda

- 1 Algunos conceptos
- 2 Herramientas
- 3 Ejemplos
- 4 Recursos

# GUI

- Graphical User Interface.
- Permite interacción con una computadora a través de ventanas, botones, etc.
- Acciones dependen de la manipulación de estos objetos, a través del teclado, ratón.

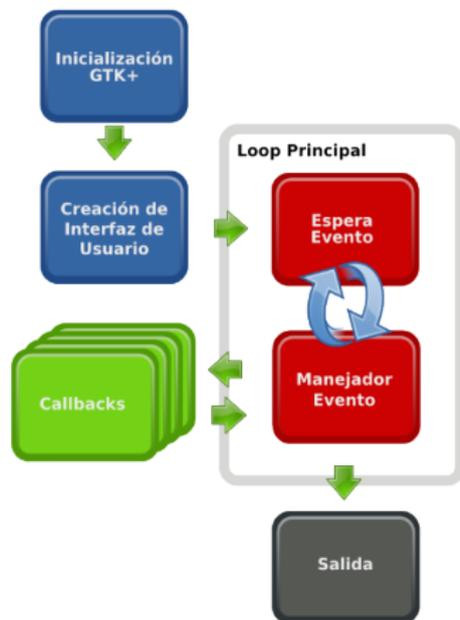
# GUI

- Widget.
  - Nombre genérico para botones, ventanas, barras de menú, etc.
  - Usuario interactúa con ellos.
  - Pueden contener otros widgets.
- Signals.
  - Un evento produce una señal (clicked, edited, etc).
- Callbacks.
  - Función llamada por una señal.

# Programación por eventos

- Ejecución del programa depende de los eventos que ocurran (presionar un botón, cerrar una ventana, etc).
- GTK funciona de esta manera.
  - Loop principal (`gtk.main()`) a la espera de algún evento y pasar control a alguna función (callback).

# Programación por eventos



# python

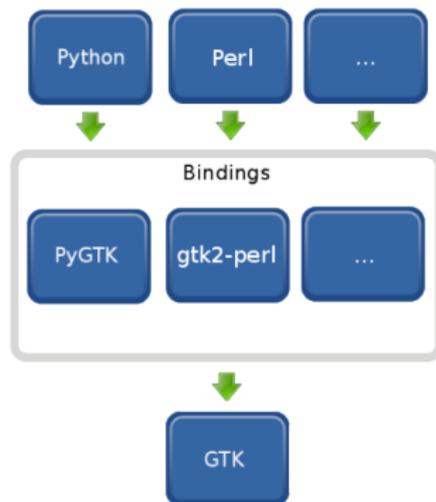
- VHLL (very high-level programming language).
- Interpretado y OO.
- Muy fácil de leer y mantener.
- Bastante documentación.

# GTK

- Toolkit multiplataforma para crear interfaces gráficas de usuario, escrito en C.
- Bindings en C++, Java, perl, python, etc.

## pygtk

- Binding de GTK+
- Parte del proyecto GNOME.



# glade

- Asistente para diseño de GUIs.
- Almacena información en archivos XML.
- <http://deb.otalca.cl/ogv/glade.ogv>

# libglade

- Cargar y procesar archivos XML.
- Separa parte lógica de la aplicación.
- Modificar interfaz (layout) sin necesidad de modificar código fuente.

# gtkbuilder

- GtkBuilder similar a Libglade, pero?
- Libglade no forma parte de GTK.
- Libglade sigue desarrollo independiente (demora en soporte de nuevos widget y propiedades).
- Desarrolladores de GTK mantienen esta biblioteca integrada en GTK.
- gtk-builder-convert: Convierte archivos glade para ser cargados por GtkBuilder.

# devhelp

- Programa de ayuda para desarrolladores (python-gtk2-doc).
- <http://deb.otalca.cl/ogv/devhelp.ogv>

# HIG - Human Interface Guidelines

- Proyecto de usabilidad para GNOME.
- "Recomienda" como debemos crear las interfaces, para que:
  - Los usuarios debieran aprender a usar de forma mas rápida la aplicación.
  - La aplicación sea atractiva.
  - La aplicación sea accesible para todos
  - La aplicación se vea bien con los cambios de temas de escritorio
  - etc.

# Editores

- Donde escribir nuestros programas?
- A gusto del consumidor...
  - Vi (Vim)
  - Gedit
  - Geany (<http://deb.otalca.cl/ogv/geany.ogv>)
  - Anjuta
  - ...

# Manos a la obra!

- Sin señales
- Con señales
- Con gtkbuilder
- Con Clases
- UIManager
- SQLite + Treeview
- Internacionalización (gettext)
- Preferencias (gconf)

- Videos ejemplo.
  - <http://deb.ugalca.cl/ogv/>
- Enlaces
  - Glade: <http://glade.gnome.org/>
  - Python: <http://www.python.org/>
  - PyGTK: <http://www.pygtk.org/>
  - GNOME: <http://www.gnome.org/>
  - GNOME Chile: <http://www.gnome.cl/>
  - FSF: <http://www.fsf.org/>
  - gettext: <http://www.gnu.org/software/gettext/>
  - gconf: <http://projects.gnome.org/gconf/>
  - HIG: <http://library.gnome.org/devel/hig-book/stable/>
  - Graphviz: <http://www.graphviz.org/>
- Lista de correo
  - <http://mail.gnome.org/mailman/listinfo/gnome-cl-list>
  - [gnome-cl-list@gnome.org](mailto:gnome-cl-list@gnome.org)