



Tema 5 Diseño con Glade

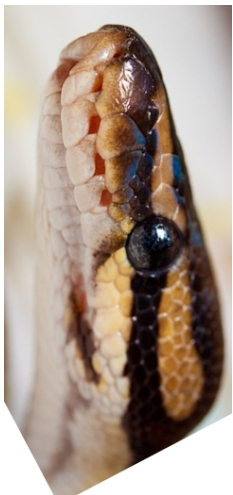
Curso de Python Avanzado

Juan Pedro Bolívar Puente

Instituto Andaluz de Astrofísica

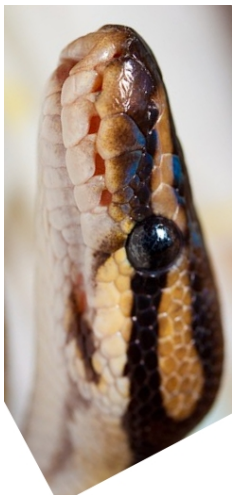
Mayo de 2011

Índice



- 1 Introducción
- 2 Un paseo por Glade
- 3 Proyecto

Índice



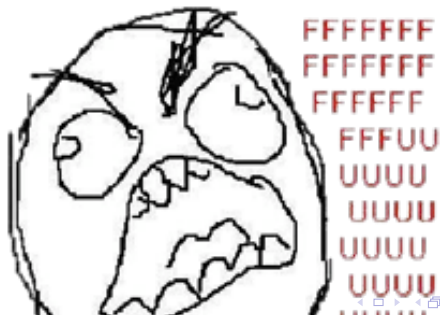
- 1 Introducción
- 2 Un paseo por Glade
- 3 Proyecto

¡Yo también quiero botones!

O sea que...

Le damos botones a los usuarios...
¡Y nosotros no paramos de picar código!

¡Esto no es justo!



... y usuarios

Solución:

¡Editores de interfaces gráficas!

- **Glade**: editor de interfaces gráficas de GTK
- Editamos la interfaz visualmente ...
- La guardamos en un ficherito XML ...
- Luego cargamos la interfaz ...
- Conectamos las señales automáticamente ...

El ficherito de Glade...

Un ficherito editor por Glade

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE glade-interface SYSTEM "glade-2.0.dtd">
<!--Generated with glade3 3.4.0 on Tue Nov 20 14:05:37 2011-->
<glade-interface>
  <widget class="GtkWindow" id="window1">
    <property name="events">GDK_POINTER_MOTION_MASK |
GDK_POINTER_MOTION_HINT_MASK | GDK_BUTTON_PRESS_MASK |
GDK_BUTTON_RELEASE_MASK</property>
    <child>
      <placeholder/>
    </child>
  </widget>
</glade-interface>
```

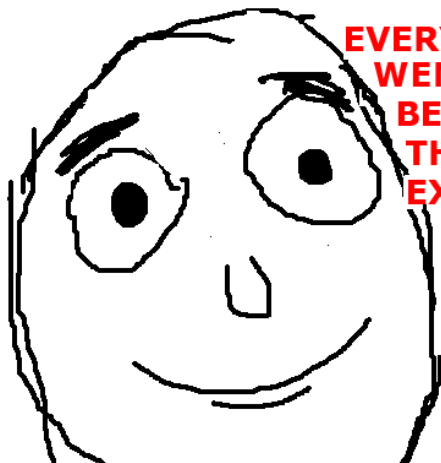
Ventajas

¡Todo son ventajas!

- Podemos **cambiar el diseño** sin cambiar el código
- Podemos **centrarnos en el diseño**
⇒ **¡Nuestros usuarios nos lo agradecerán!**
- Glade ayuda *por defecto* a seguir las **Gnome Human Interface Guidelines**

Ventajas

¡Todo son ventajas!



**EVERYTHING
WENT
BETTER
THAN
EXPECTED**

Gnome Human Interface Guidelines

¡El diseño es importante!

- Design for **People**
- **Don't Limit** Your User Base
- Match Between **Application** → **Real World**
- Make Your Application **Consistent**
- Keep the User **Informed**
- Keep It **Simple and Pretty**
- Put the User in **Control**
- **Forgive** the User
- Provide **Direct Manipulation**

Gnome Human Interface Guidelines

¡El diseño es importante!

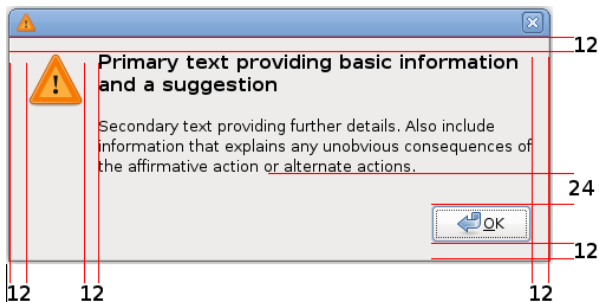
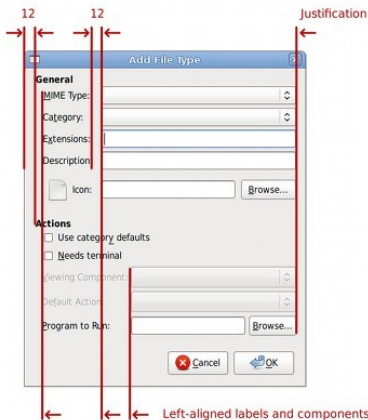


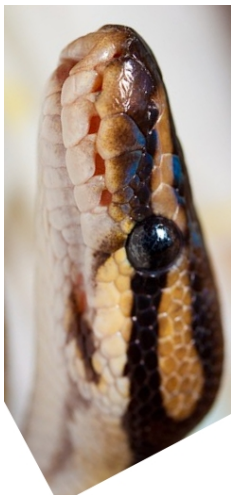
Figura: Da recomendaciones estilísticas concretas

Gnome Human Interface Guidelines

¡El diseño es importante!

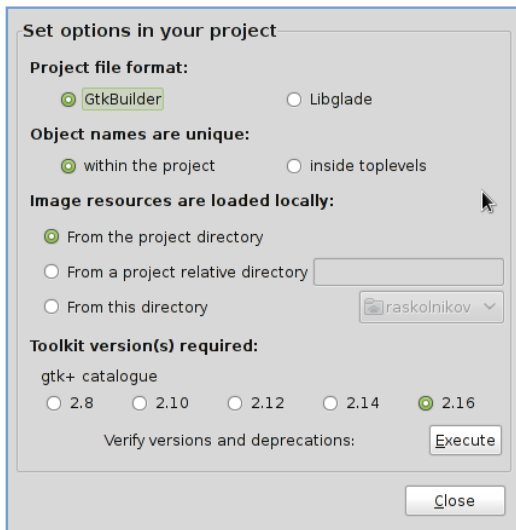


Índice

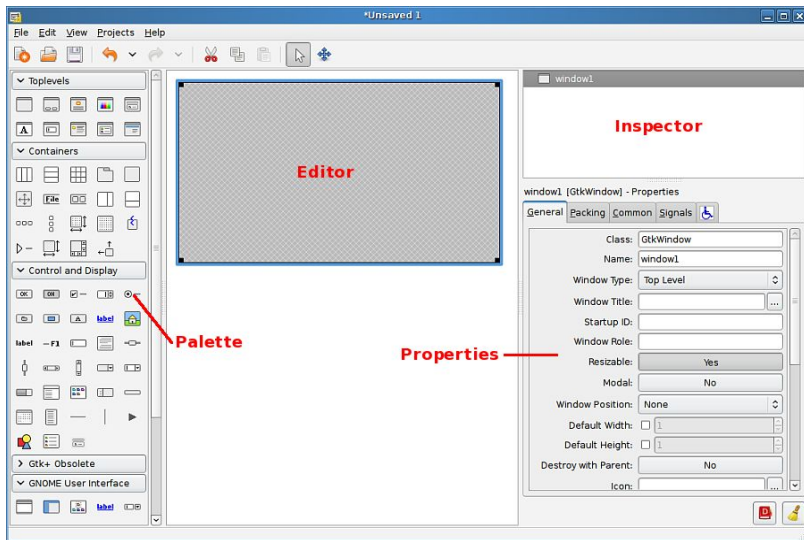


- 1 Introducción
- 2 Un paseo por Glade
- 3 Proyecto

Nada más abrir Glade ...



Estructura del editor ...



Cargando la interfaz en el código

¿Cómo cargamos la interfaz en el código?

Antaño ...

- Usábamos `libglade`
<http://developer.gnome.org/libglade/stable/>
- Sigue siendo mantenida por GNOME

Ahora ...

- ¡GTK nos da todo lo necesario!
- Nos ahorramos `dependencias`

Cargando la interfaz en el código ...

La clase `gtk.Builder`

- `add_from_file (fname)` → Carga el fichero `fname`
- `connect_signals (dic_or_obj)` → Conecta los manejadores con:
 - Métodos con ese nombre de un objeto
 - Funciones especificadas en un diccionario
- `get_object (objname)` → Devuelve el objeto con nombre `objname` del fichero ...

Conectando señales con un botón ...

The screenshot shows the Glade GUI designer interface. The central canvas displays a button widget with the text "¡Pinchame!". The right-hand panel is set to the "Signals" tab for the selected widget, "button1".

Widget Hierarchy:

- Widgets
 - window
 - button1

Signal Handler Table:

Signal	Handler
activate	<Type here>
▶ clicked	handle_clicked
enter	<Type here>
leave	<Type here>
pressed	<Type here>
released	<Type here>
▶ GtkContainer	
▶ GtkWidget	
▶ GObject	

Nuestra primera aplicación ...

```
import gtk
import util

class TutorialApp (object):
    def __init__ (self, *a, **k):
        super (TutorialApp, self).__init__ (..

        b = gtk.Builder()
        b.add_from_file ("tutorial.glade")
        b.connect_signals (self)

        self.window = b.get_object ("window")
        self.window.show ()
```

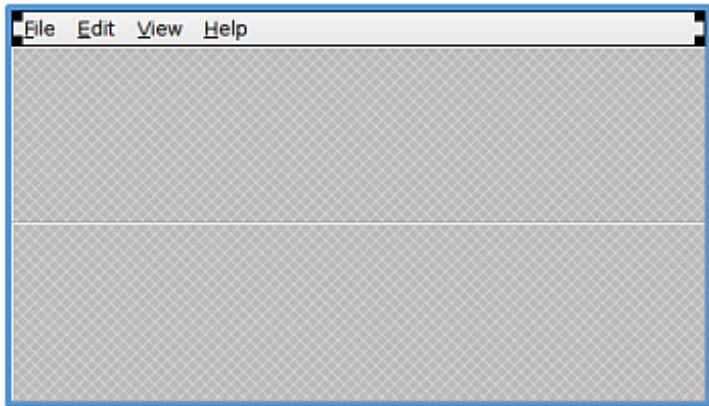
Nuestra primera aplicación ...

```
def handle_clicked (self, obj):
    util.show_message_dialog (
        "Mmmmmmm...",
        "Que gustito!")

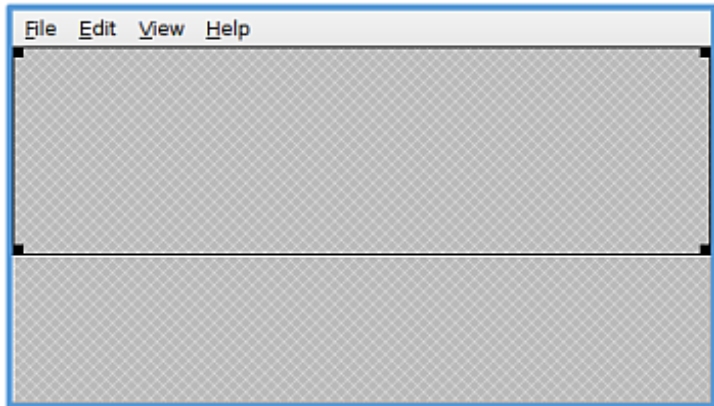
def handle_window_destroy (self, obj):
    gtk.main_quit ()

if __name__ == "__main__":
    app = TutorialApp ()
    gtk.main ()
```

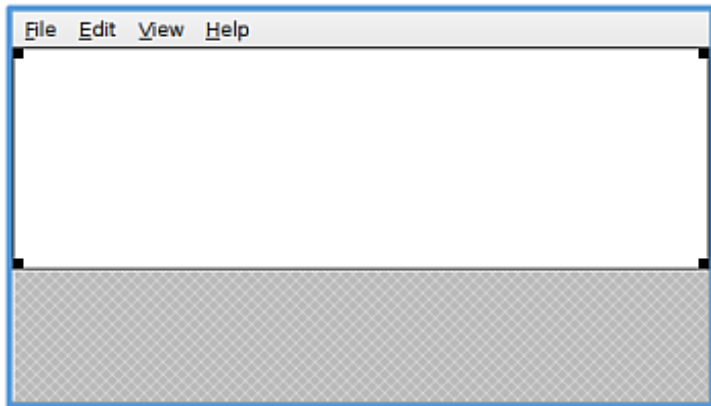
Vamos creando cajas y rellenando huecos



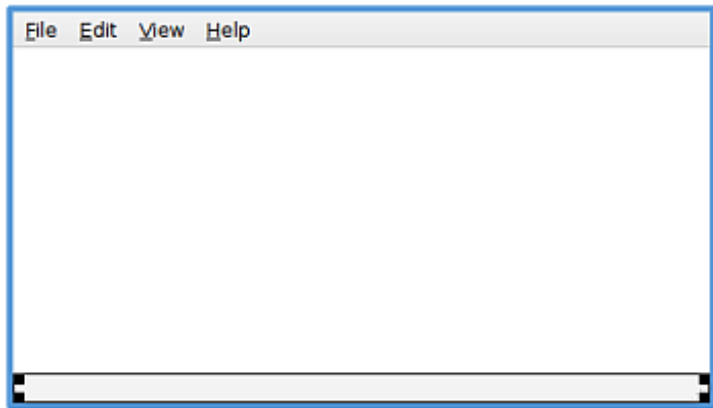
Vamos creando cajas y rellenando huecos



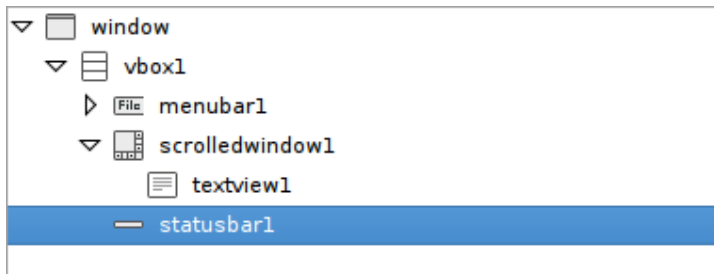
Vamos creando cajas y rellenando huecos



Vamos creando cajas y rellenando huecos



Vamos creando cajas y rellenando huecos



Las propiedades importantes del *layout*

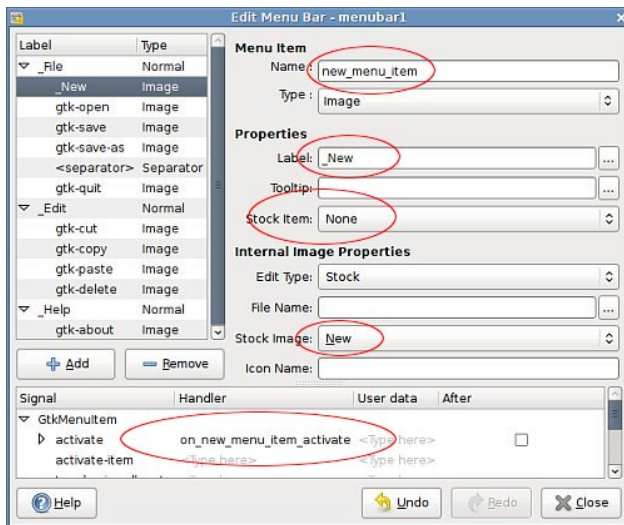
En las cajas ...

- `homogeneous` → Si es TRUE los huecos tendrán el mismo espacio dedicado fijo

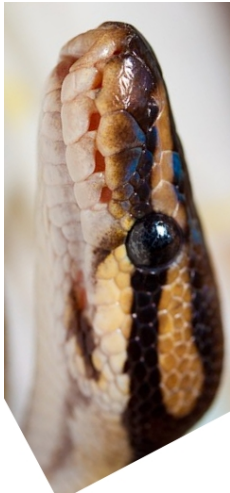
En los *widgets* de dentro ...

- `fill` → Rellenar el hueco extra disponible
- `expand` → Solicitar todo el espacio posible

También podemos editar menús ...



Índice



- 1 Introducción
- 2 Un paseo por Glade
- 3 Proyecto**

Vamos a hacer algo más chulo...

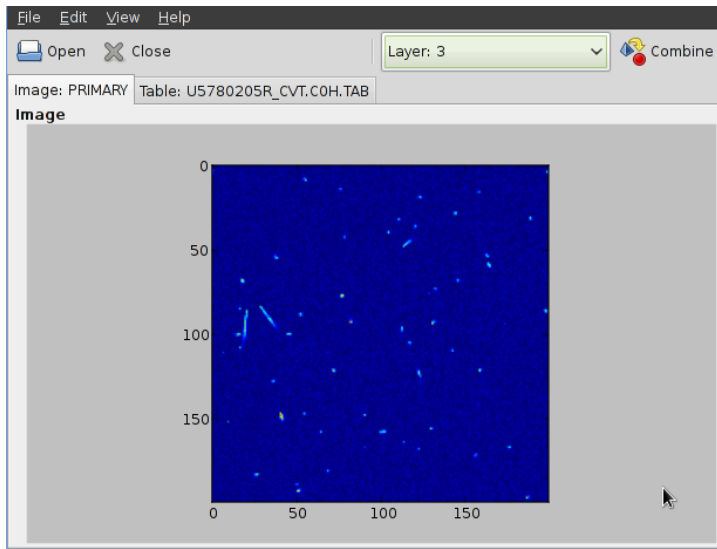
Vamos a programar `gfits`

Aplicación para visualizar ficheros FITS

- Que permita ver las imagines
- Mezclar cubos como imágenes RGB
- Ver las tablas que contiene

Dedicaremos el resto del tema a `estudiar su código e intentar mejorarlo ...`

¡Ta dá!



Recursos adicionales



Gnome Human Interface Guidelines

The GNOME Team

<http://developer.gnome.org/hig-book/>



Glade Tutorial

Micah Carrick

<http://www.micahcarrick.com/gtk-glade-tutorial-part-1.html>



Python code to convert FITS files to images

Min-Su Shin

http://www.astro.lsa.umich.edu/~msshin/science/code/Python_fits_image/

¿Preguntas?

Muchas gracias por su atención.

