Aplicaciones de escritorio con python3 y Gtk+

danigm
http://danigm.net
danigm@wadobo.com

wadobo
it isn't magic, it's wadobo!

¿Qué es Gtk+?

wadobo
it isn't magic, it's wadobo!

Aplicación básica de ejemplo con python3

wadobo
it isn't magic, it's wadobo!

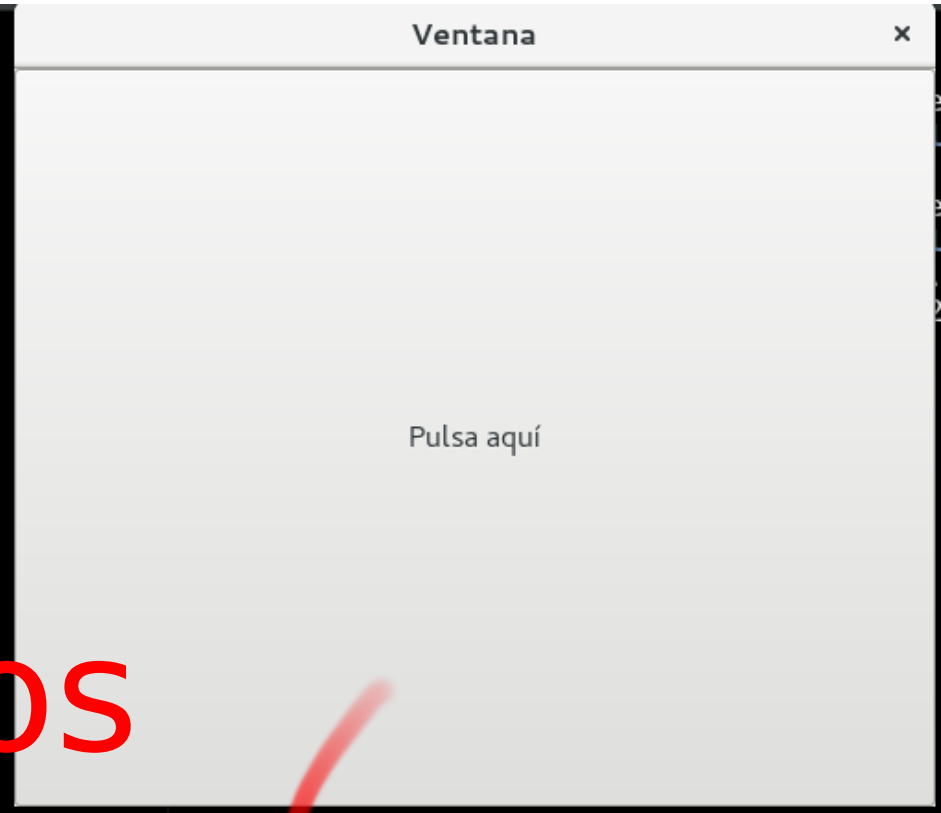
```python
#!/usr/bin/env python3

import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk


class MyWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Ventana")
        self.set_default_size(500, 400)
        self.connect("delete-event", Gtk.main_quit)

        self.label = Gtk.Label('¡Hola Mundo!')
        self.add(self.label)

        self.show_all()


win = MyWindow()
Gtk.main()
```

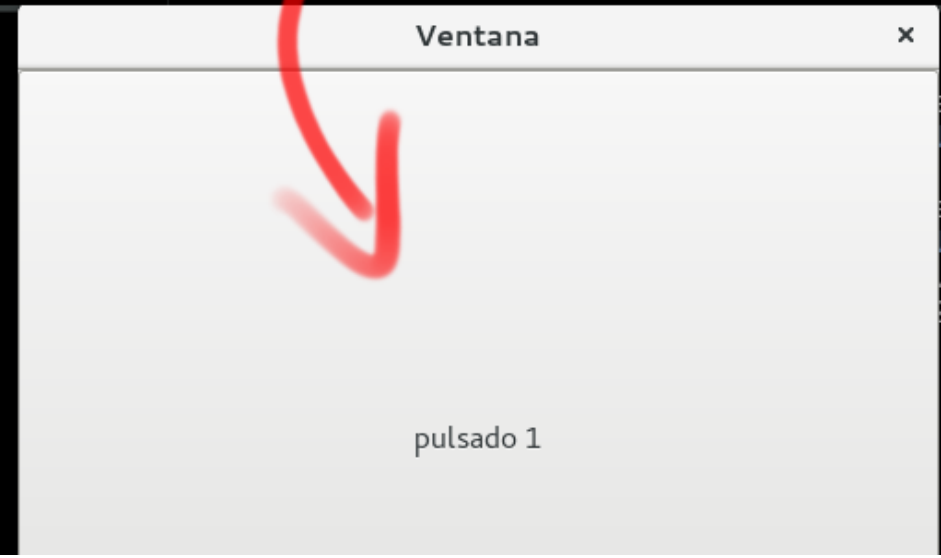
Main loop, eventos y señales

wadobo
it isn't magic, it's wadobo!

```python
#!/usr/bin/env python3

import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk


class MyWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Ventana")
        self.set_default_size(500, 400)
        self.connect("delete-event", Gtk.main_quit)

        self.button = Gtk.Button('Pulsa aquí')
        self.button.connect('clicked', self.button_clicked)

        self.add(self.button)
        self.n = 0

        #self.button.emit('clicked')

        self.show_all()

    def button_clicked(self, button):
        self.n += 1
        self.button.set_label("pulsado {}".format(self.n))


win = MyWindow()
Gtk.main()
```

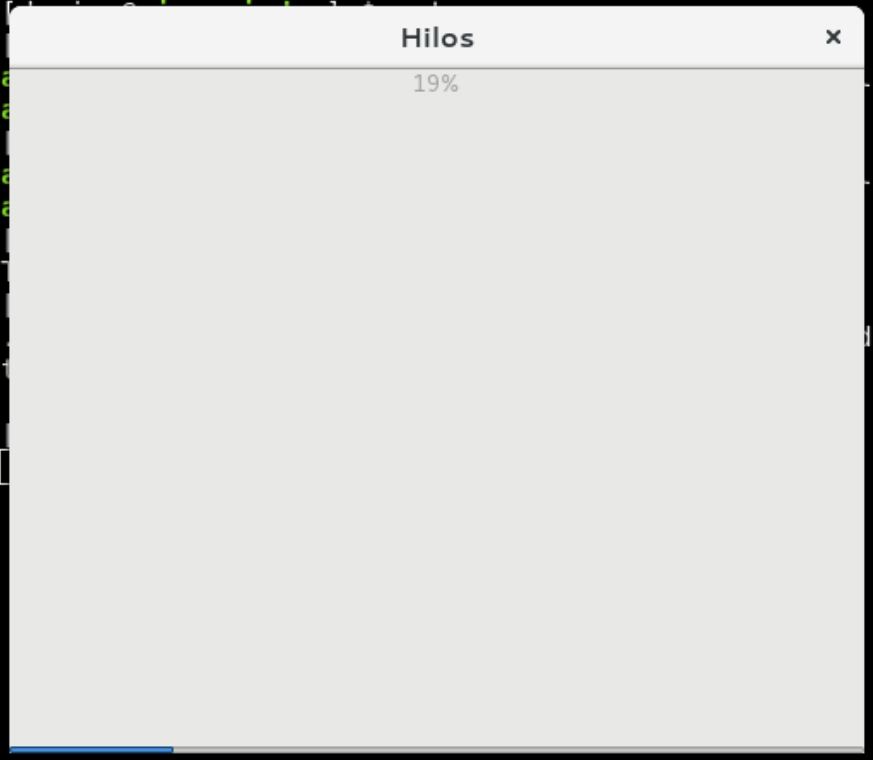
Eventos





```python
class MyWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Hilos")
        self.set_default_size(500, 400)
        self.stopped = False
        self.connect("delete-event", self.quit)

        self.progressbar = Gtk.ProgressBar()
        self.progressbar.set_show_text("0%")

        self.add(self.progressbar)
        self.n = 0

        self.show_all()

    def quit(self, *args):
        self.stopped = True
        Gtk.main_quit()

    def update_progressbar(self):
        self.progressbar.set_fraction(self.n / 100)

    def time_pass(self):
        self.n += 1

        #self.update_progressbar()
        GLib.idle_add(self.update_progressbar)

        if not self.stopped and self.n < 100:
            t = Timer(0.5, self.time_pass)
            t.start()


win = MyWindow()

t = Timer(0.5, win.time_pass)
t.start()

Gtk.main()
```

Hilos



Contenedores

wadobo
it isn't magic, it's wadobo!

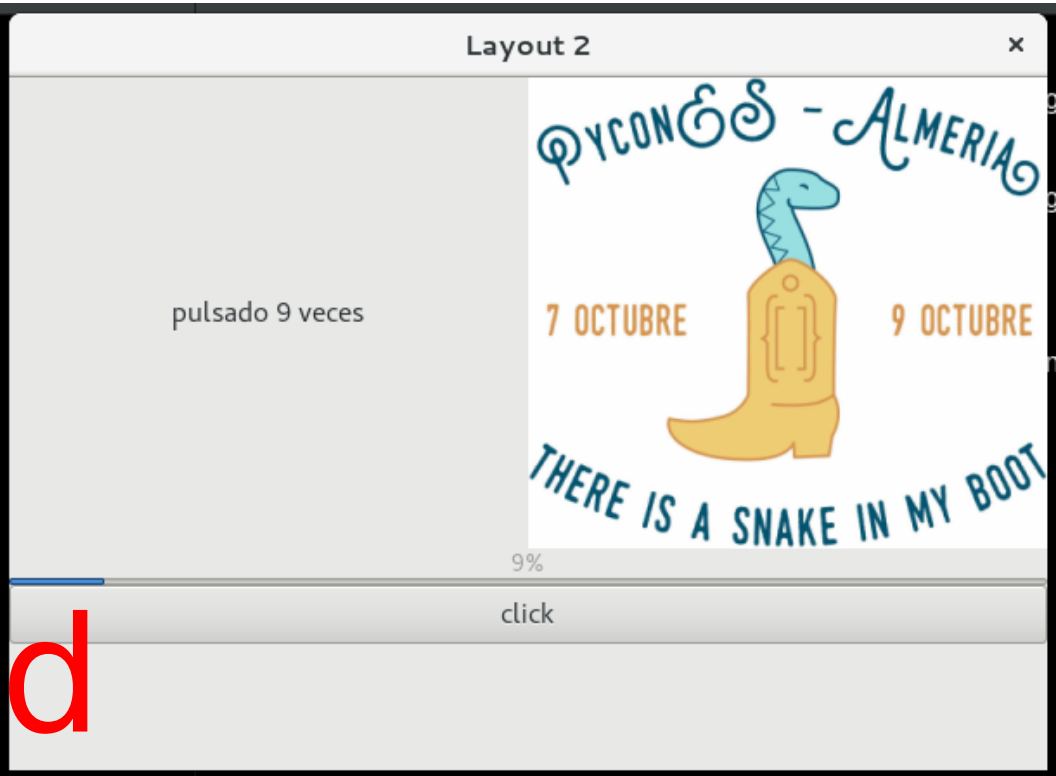
```python
class MyWindow(Gtk.Window):
    def __init__(self):
        Gtk.Window.__init__(self, title="Layout 1")
        self.set_default_size(500, 400)
        self.stopped = False
        self.connect("delete-event", self.quit)

        self.box = Gtk.Box(orientation=Gtk.Orientation.VERTICAL)
        self.box2 = Gtk.Box(orientation=Gtk.Orientation.HORIZONTAL)
        self.img = Gtk.Image.new_from_file("pycon-small.png")
        self.label = Gtk.Label("texto")
        self.button = Gtk.Button("click")
        self.progressbar = Gtk.ProgressBar()
        self.progressbar.set_show_text("0%")

        self.button.connect('clicked', self.callback)

        self.box2.pack_start(self.label, True, True, 0)
        self.box2.add(self.img)

        self.box.pack_start(self.box2, True, True, 0)
        self.box.add(self.progressbar)
        self.box.add(self.button)

        self.add(self.box)
        self.n = 0

        self.show_all()

    def quit(self, *args):
        self.stopped = True
        Gtk.main_quit()

    def callback(self, button):
        self.n += 1
        self.progressbar.set_fraction(self.n / 100)
        self.label.set_text("pulsado {} veces".format(self.n))


win = MyWindow()
Gtk.main()
```

app-04.py [+] 12,1 88%

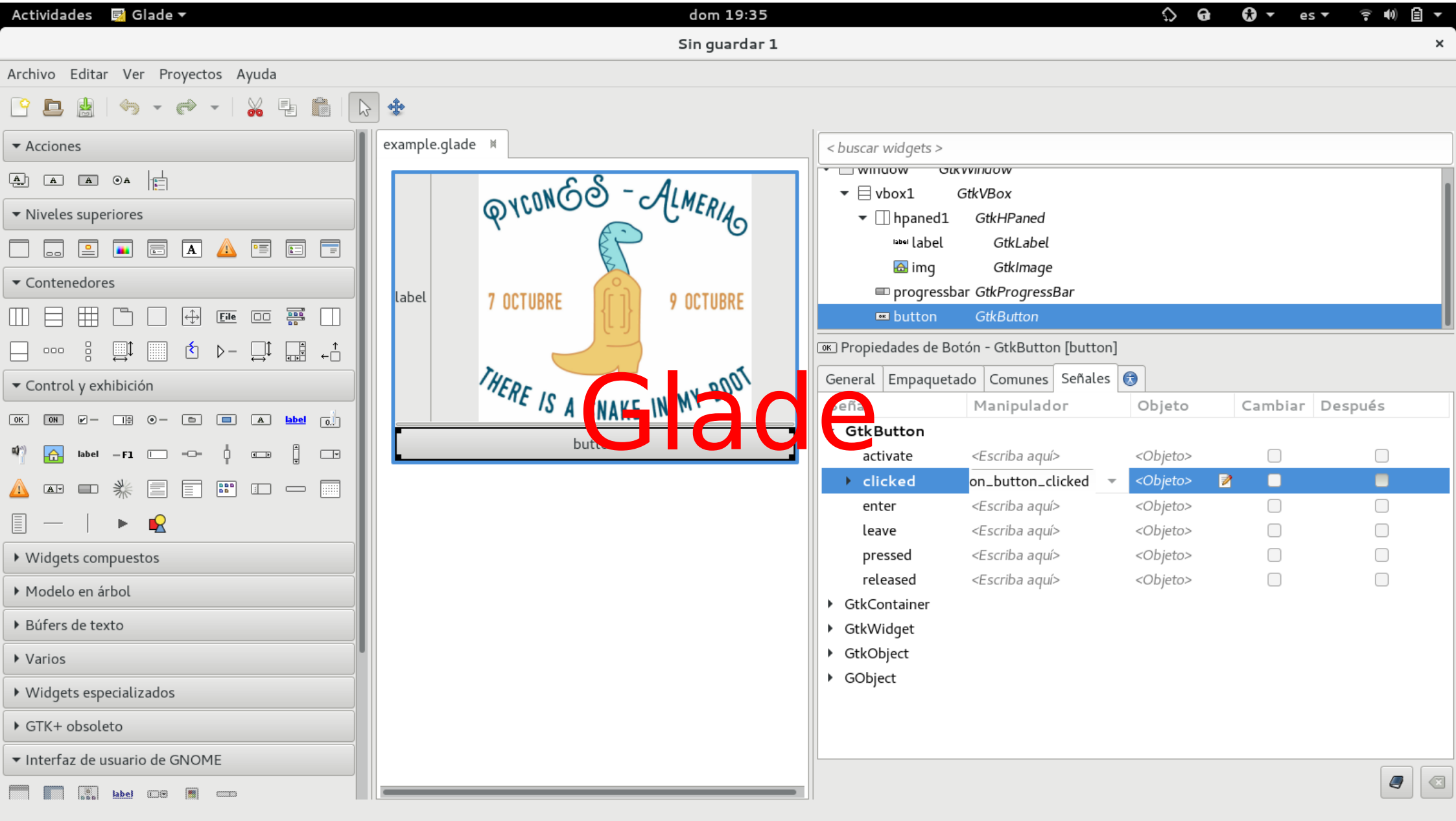
Box

```python
class MyWindow(Gtk.Window):
    def __init__(self):
        Gtk.Window.__init__(self, title="Layout 2")
        self.set_default_size(500, 400)
        self.stopped = False
        self.connect("delete-event", self.quit)

        self.grid = Gtk.Grid()

        self.img = Gtk.Image.new_from_file("pycon-small.png")
        self.label = Gtk.Label("texto")
        self.button = Gtk.Button("click")
        self.progressbar = Gtk.ProgressBar()
        self.progressbar.set_show_text("0%")

        self.button.connect('clicked', self.callback)

        self.grid.set_column_homogeneous(True)
        # left, top, with, height
        self.grid.attach(self.label, 0, 0, 1, 1)
        self.grid.attach(self.img, 1, 0, 1, 1)
        self.grid.attach(self.progressbar, 0, 1, 2, 1)
        self.grid.attach(self.button, 0, 2, 2, 1)

        self.add(self.grid)
        self.n = 0

        self.show_all()

    def quit(self, *args):
        self.stopped = True
        Gtk.main_quit()

    def callback(self, button):
        self.n += 1
        self.progressbar.set_fraction(self.n / 100)
        self.label.set_text("pulsado {} veces".format(self.n))

win = MyWindow()
Gtk.main()
```

Grid



wadobo
it isn't magic, it's wadobo!


Actividades
Glade
dom 19:35
es
Sin guardar 1
Archivo Editar Ver Proyectos Ayuda
Acciones
Niveles superiores
Contenedores
Control y exhibición
Widgets compuestos
Modelo en árbol
Búfers de texto
Varios
Widgets especializados
GTK+ obsoleto
Interfaz de usuario de GNOME
example.glade
label
7 OCTUBRE
9 OCTUBRE
Glade
< buscar widgets >
vbox1 GtkVBox
hpaned1 GtkHPaned
label GtkLabel
img GtkImage
progressbar GtkProgressBar
button GtkButton
Propiedades de Botón - GtkButton [button]
General
Empaquetado
Comunes
Señales
Manipulador
Objeto
Cambiar
Después
GtkButton
activate
clicked
on_button_clicked
enter
leave
pressed
released
GtkContainer
GtkWidget
GtkObject
GObject

```python
#!/usr/bin/env python3

import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk
from gi.repository import GLib



class Handler:
    def __init__(self, builder):
        self.builder = builder
        self.n = 0

    def on_window_delete_event(self, *args):
        Gtk.main_quit(*args)

    def on_button_clicked(self, button):
        self.n += 1

        button = self.builder.get_object("button")
        label = self.builder.get_object("label")
        progressbar = self.builder.get_object("progressbar")

        progressbar.set_fraction(self.n / 100)
        label.set_text("pulsado {} veces".format(self.n))


builder = Gtk.Builder()
builder.add_from_file("example.glade")
builder.connect_signals(Handler(builder))

window = builder.get_object("window")
window.show_all()

Gtk.main()
```

GtkBuilder

```
[danigm@rincewind ~] $ rst
```

Gobject-Introspection

wadobo
it isn't magic, it's wadobo!

```
2375
2376 /**
2377  * gtk_window_get_title:
2378  * @window: a #GtkWindow
2379  *
2380  * Retrieves the title of the window. See gtk_window_set_title().
2381  *
2382  * Returns: (nullable): the title of the window, or %NULL if none has
2383  * been set explicitly. The returned string is owned by the widget
2384  * and must not be modified or freed.
2385  **/
2386 const gchar *
2387 gtk_window_get_title (GtkWindow *window)
2388 {
2389   g_return_val_if_fail (GTK_IS_WINDOW (window), NULL);
2390
2391   return window->priv->title;
2392 }
2393
```

GIR

Más info:

* https://developer.gnome.org/
* https://python-gtk-3-tutorial.readthedocs.io/en/latest/index.html
* https://wiki.gnome.org/Projects/PyGObject/Threading
* https://developer.gnome.org/gtk3/unstable/index.html

wadobo
it isn't magic, it's wadobo!