

Aplicaciones de escritorio con python3 y Gtk+

danigm

<http://wadobo.com>

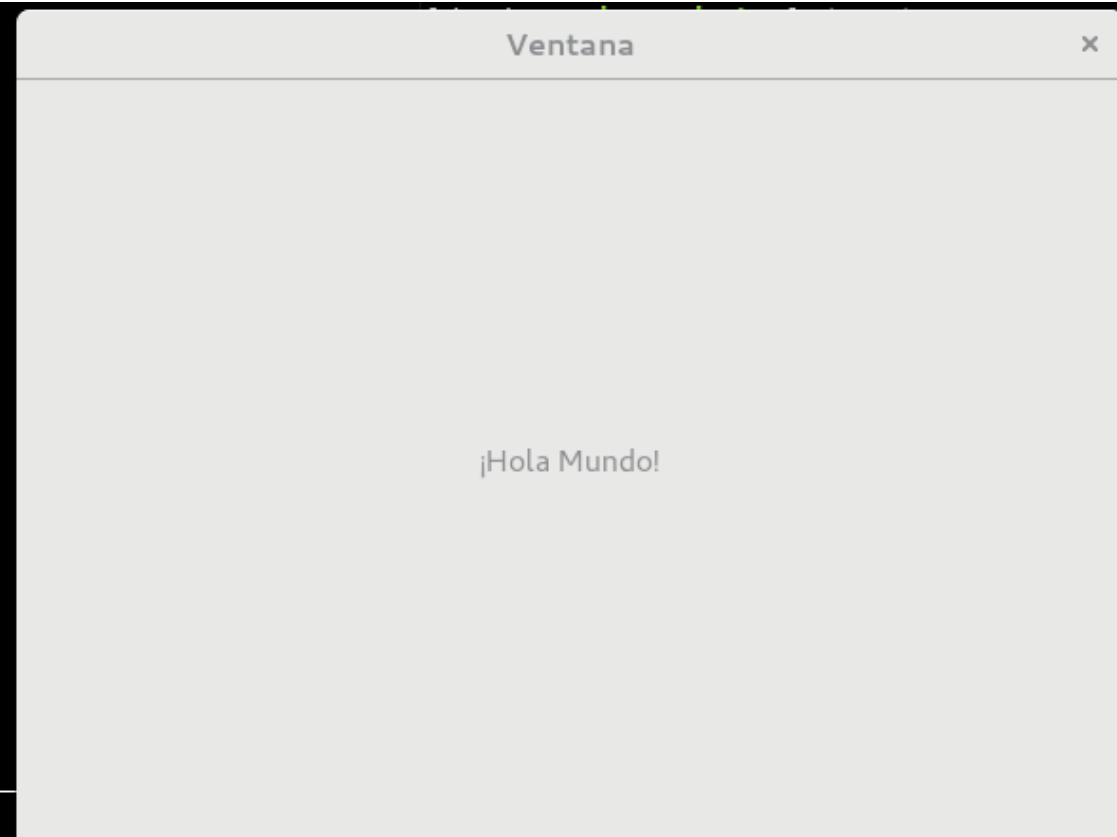
danigm@wadobo.com

<https://github.com/danigm/python3-gtk>

¿Qué es Gtk+?

Aplicación básica de ejemplo con python3

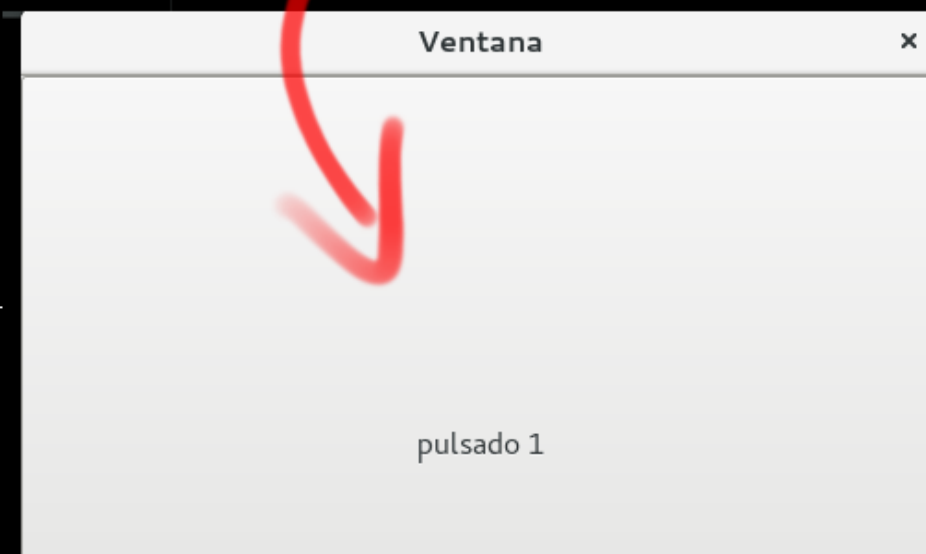
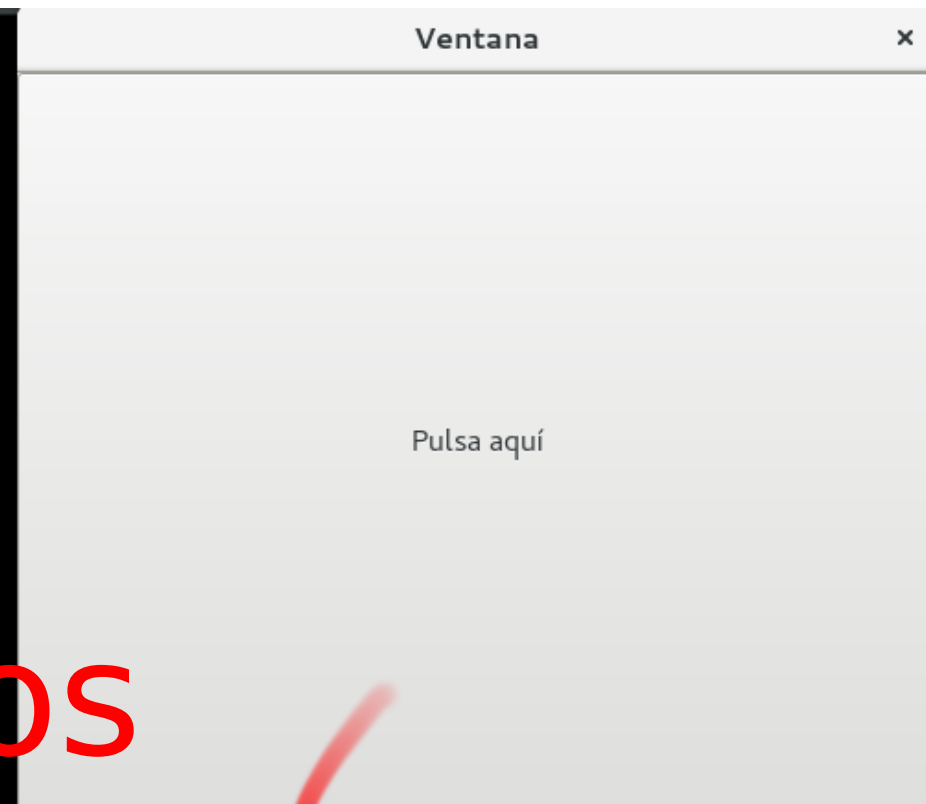
```
1 #!/usr/bin/env python3
2
3 import gi
4 gi.require_version('Gtk', '3.0')
5 from gi.repository import Gtk
6
7
8 class MyWindow(Gtk.Window):
9
10     def __init__(self):
11         Gtk.Window.__init__(self, title="Ventana")
12         self.set_default_size(500, 400)
13         self.connect("delete-event", Gtk.main_quit)
14
15         self.label = Gtk.Label('¡Hola Mundo!')
16         self.add(self.label)
17
18         self.show_all()
19
20
21 win = MyWindow()
22 Gtk.main()
```



Main loop, eventos y señales

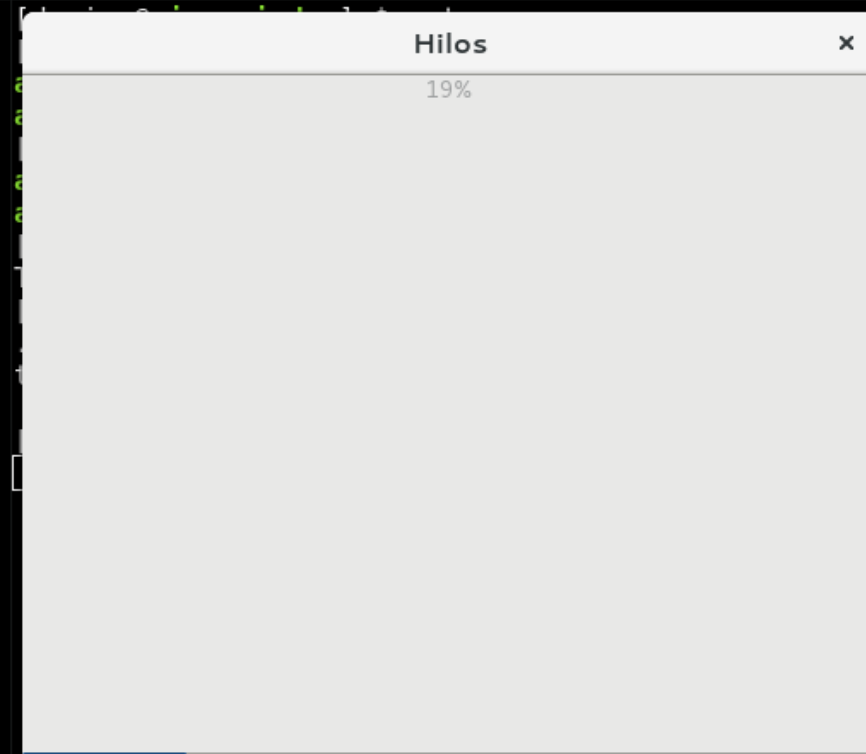
Eventos

```
1 #!/usr/bin/env python3
2
3 import gi
4 gi.require_version('Gtk', '3.0')
5 from gi.repository import Gtk
6
7
8 class MyWindow(Gtk.Window):
9
10     def __init__(self):
11         Gtk.Window.__init__(self, title="Ventana")
12         self.set_default_size(500, 400)
13         self.connect("delete-event", Gtk.main_quit)
14
15         self.button = Gtk.Button('Pulsa aquí')
16         self.button.connect('clicked', self.button_clicked)
17
18         self.add(self.button)
19         self.n = 0
20
21         #self.button.emit('clicked')
22
23         self.show_all()
24
25     def button_clicked(self, button):
26         self.n += 1
27         self.button.set_label("pulsado {}".format(self.n))
28
29
30 win = MyWindow()
31 Gtk.main()
32
```



Hilos

```
11 class MyWindow(Gtk.Window):
12
13     def __init__(self):
14         Gtk.Window.__init__(self, title="Hilos")
15         self.set_default_size(500, 400)
16         self.stopped = False
17         self.connect("delete-event", self.quit)
18
19         self.progressbar = Gtk.ProgressBar()
20         self.progressbar.set_show_text("0%")
21
22         self.add(self.progressbar)
23         self.n = 0
24
25         self.show_all()
26
27     def quit(self, *args):
28         self.stopped = True
29         Gtk.main_quit()
30
31     def update_progressbar(self):
32         self.progressbar.set_fraction(self.n / 100)
33
34     def time_pass(self):
35         self.n += 1
36
37     #self.update_progressbar()
38     Glib.idle_add(self.update_progressbar)
39
40     if not self.stopped and self.n < 100:
41         t = Timer(0.5, self.time_pass)
42         t.start()
43
44
45 win = MyWindow()
46
47 t = Timer(0.5, win.time_pass)
48 t.start()
49
50 Gtk.main()
51
```



Contenedores


```

9 class MyWindow(Gtk.Window):
10     def __init__(self):
11         Gtk.Window.__init__(self, title="Layout 1")
12         self.set_default_size(500, 400)
13         self.stopped = False
14         self.connect("delete-event", self.quit)
15
16         self.box = Gtk.Box(orientation=Gtk.Orientation.VERTICAL)
17         self.box2 = Gtk.Box(orientation=Gtk.Orientation.HORIZONTAL)
18         self.img = Gtk.Image.new_from_file("pycon-small.png")
19         self.label = Gtk.Label("texto")
20         self.button = Gtk.Button("click")
21         self.progressbar = Gtk.ProgressBar()
22         self.progressbar.set_show_text("0%")
23
24         self.button.connect('clicked', self.callback)
25
26         self.box2.pack_start(self.label, True, True, 0)
27         self.box2.add(self.img)
28
29         self.box.pack_start(self.box2, True, True, 0)
30         self.box.add(self.progressbar)
31         self.box.add(self.button)
32
33         self.add(self.box)
34         self.n = 0
35
36         self.show_all()
37
38     def quit(self, *args):
39         self.stopped = True
40         Gtk.main_quit()
41
42     def callback(self, button):
43         self.n += 1
44         self.progressbar.set_fraction(self.n / 100)
45         self.label.set_text("pulsado {} veces".format(self.n))
46
47
48 win = MyWindow()
49 Gtk.main()

```

Box



Grid

```
9 class MyWindow(Gtk.Window):
10     def __init__(self):
11         Gtk.Window.__init__(self, title="Layout 2")
12         self.set_default_size(500, 400)
13         self.stopped = False
14         self.connect("delete-event", self.quit)
15
16         self.grid = Gtk.Grid()
17
18         self.img = Gtk.Image.new_from_file("pycon-small.png")
19         self.label = Gtk.Label("texto")
20         self.button = Gtk.Button("click")
21         self.progressbar = Gtk.ProgressBar()
22         self.progressbar.set_show_text("0%")
23
24         self.button.connect('clicked', self.callback)
25
26         self.grid.set_column_homogeneous(True)
27         # left, top, with, height
28         self.grid.attach(self.label, 0, 0, 1, 1)
29         self.grid.attach(self.img, 1, 0, 1, 1)
30         self.grid.attach(self.progressbar, 0, 1, 2, 1)
31         self.grid.attach(self.button, 0, 2, 2, 1)
32
33         self.add(self.grid)
34         self.n = 0
35
36         self.show_all()
37
38     def quit(self, *args):
39         self.stopped = True
40         Gtk.main_quit()
41
42     def callback(self, button):
43         self.n += 1
44         self.progressbar.set_fraction(self.n / 100)
45         self.label.set_text("pulsado {} veces".format(self.n))
46
47 win = MyWindow()
48 Gtk.main()
```

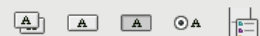


Sin guardar 1

Archivo Editar Ver Proyectos Ayuda



Acciones



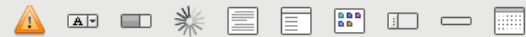
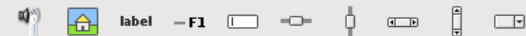
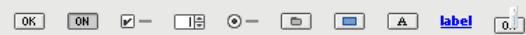
Niveles superiores



Contenedores



Control y exhibición



Widgets compuestos

Modelo en árbol

Búfers de texto

Varios

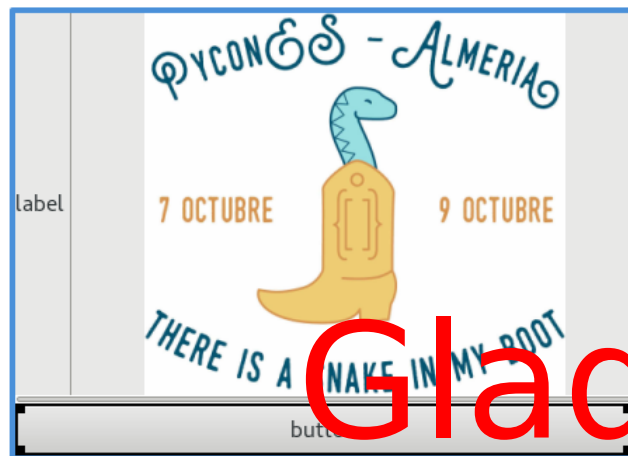
Widgets especializados

GTK+ obsoleto

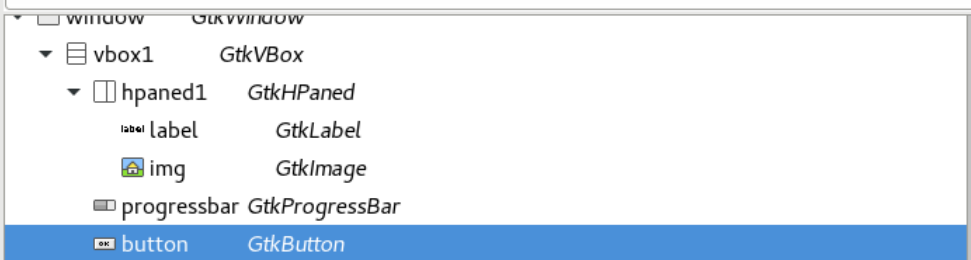
Interfaz de usuario de GNOME



example.glade



< buscar widgets >



Propiedades de Botón - GtkButton [button]

General Empaquetado Comunes Señales

Señal	Manipulador	Objeto	Cambiar	Después
-------	-------------	--------	---------	---------

GtkButton

activate	<Escriba aquí>	<Objeto>	<input type="checkbox"/>	<input type="checkbox"/>
----------	----------------	----------	--------------------------	--------------------------

clicked	on_button_clicked	<Objeto>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
---------	-------------------	----------	-------------------------------------	-------------------------------------

enter	<Escriba aquí>	<Objeto>	<input type="checkbox"/>	<input type="checkbox"/>
-------	----------------	----------	--------------------------	--------------------------

leave	<Escriba aquí>	<Objeto>	<input type="checkbox"/>	<input type="checkbox"/>
-------	----------------	----------	--------------------------	--------------------------

pressed	<Escriba aquí>	<Objeto>	<input type="checkbox"/>	<input type="checkbox"/>
---------	----------------	----------	--------------------------	--------------------------

released	<Escriba aquí>	<Objeto>	<input type="checkbox"/>	<input type="checkbox"/>
----------	----------------	----------	--------------------------	--------------------------

GtkContainer

GtkWidget

GtkObject

GObject

GtkBuilder

```
1 #!/usr/bin/env python3
2
3 import gi
4 gi.require_version('Gtk', '3.0')
5 from gi.repository import Gtk
6 from gi.repository import GLib
7
8
9
10 class Handler:
11     def __init__(self, builder):
12         self.builder = builder
13         self.n = 0
14
15     def on_window_delete_event(self, *args):
16         Gtk.main_quit(*args)
17
18     def on_button_clicked(self, button):
19         self.n += 1
20
21         button = self.builder.get_object("button")
22         label = self.builder.get_object("label")
23         progressbar = self.builder.get_object("progressbar")
24
25         progressbar.set_fraction(self.n / 100)
26         label.set_text("pulsado {} veces".format(self.n))
27
28
29 builder = Gtk.Builder()
30 builder.add_from_file("example.glade")
31 builder.connect_signals(Handler(builder))
32
33 window = builder.get_object("window")
34 window.show_all()
35
36 Gtk.main()
```

[danigm@rincewind ~] \$ rst



Gobject-Introspection

```
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>