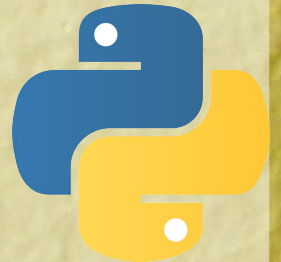


python

Interfaces gráficas de usuario con wxPython

Por Marcos Pérez



Índice

- Objetivos de wx
- Evolución de wx
- Ejemplos
- Profundizando

Objetivos de wxWidgets

- Crear aplicaciones gráficas multiplataforma
- En C++, mismo API en cualquier plataforma
- En cada plataforma aspecto e integración como si fuera nativa
- Bastante libertad de uso:
 - Licencia como LGPL + enlace dinámico / estático sin liberar código

Evolución de wxWidgets

- Comenzó en 1992 con Windows y Unix → wxWindows
- En 1997 comenzó la versión 2.
 - Rehacer código
 - Añadir nuevas plataformas / interfaces
- En 2013 versión 3 → Unicode unificado

Plataformas soportadas

- GTK
- MS Windows
- OSX (Carbon, Cocoa)
- X11
- Motif
- ~~Windows CE, GPE, ...~~



WX

Aplicaciones reales

Pylot

 Pylot - Web Performance | Version .01_alpha

File Tools

Run Stop Agents (count) 15 Interval (ms) 2000
Rampup (s) 120 Duration (s) 120 ☐ Log Responses

Summary

Run Time	Agents	Requests	Errors	Avg Resp Time	Avg Throughput	Cur Throughput
00:01:25	11/15	234	0	1.688	2.753	4.667

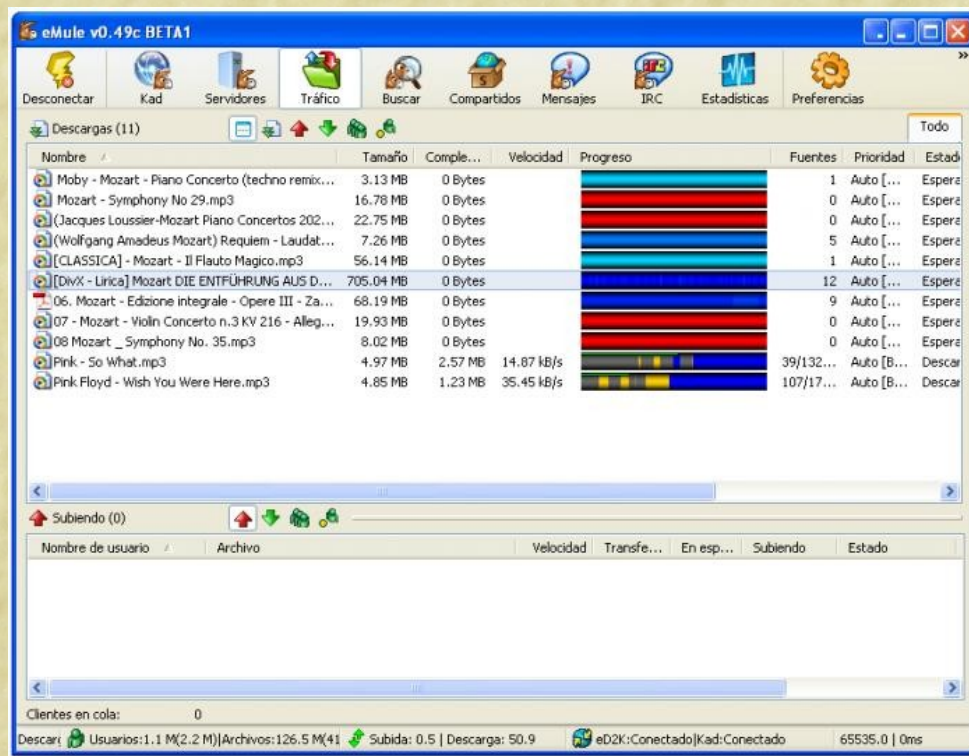
Agent Monitor

Agent Num	Status	Requests	Last Resp Time	Avg Resp Time	Bytes Received
1	running	41	2.875	1.424	1625199
2	running	37	2.344	1.490	1466643
3	running	34	1.984	1.558	1347726
4	running	29	2.734	1.733	1149531
5	running	26	2.016	1.733	1030614
6	running	21	2.640	1.840	832419
7	running	17	2.375	1.953	673863
8	running	13	2.344	2.085	515307
9	running	9	2.547	2.094	356751
10	running	5	2.641	2.041	198195
11	running	2	2.516	2.539	79278
12	waiting	0	-	-	-
13	waiting	0	-	-	-
14	waiting	0	-	-	-
15	waiting	0	-	-	-

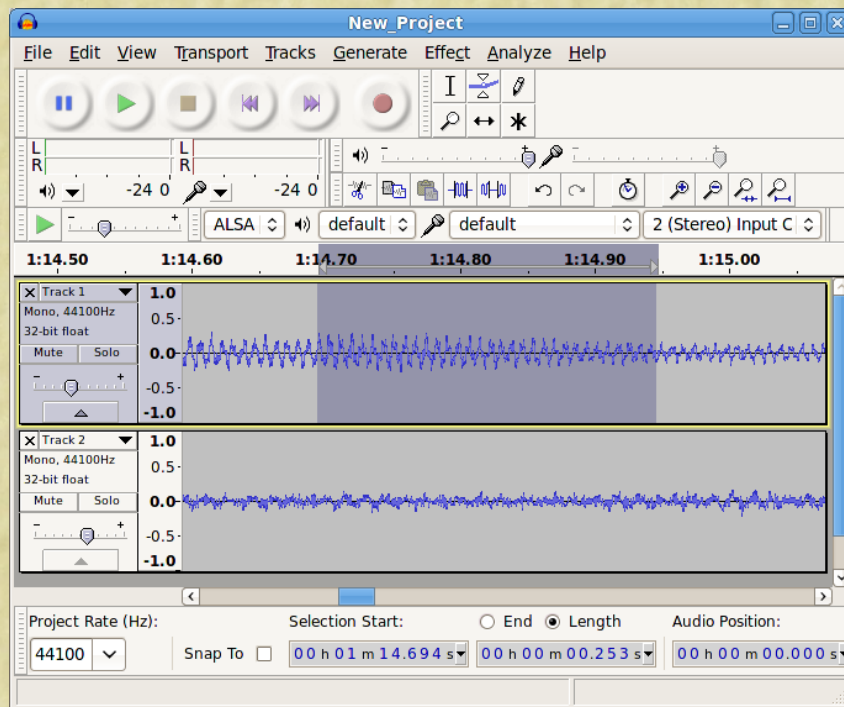
Errors

Pause Monitoring Resume Monitoring

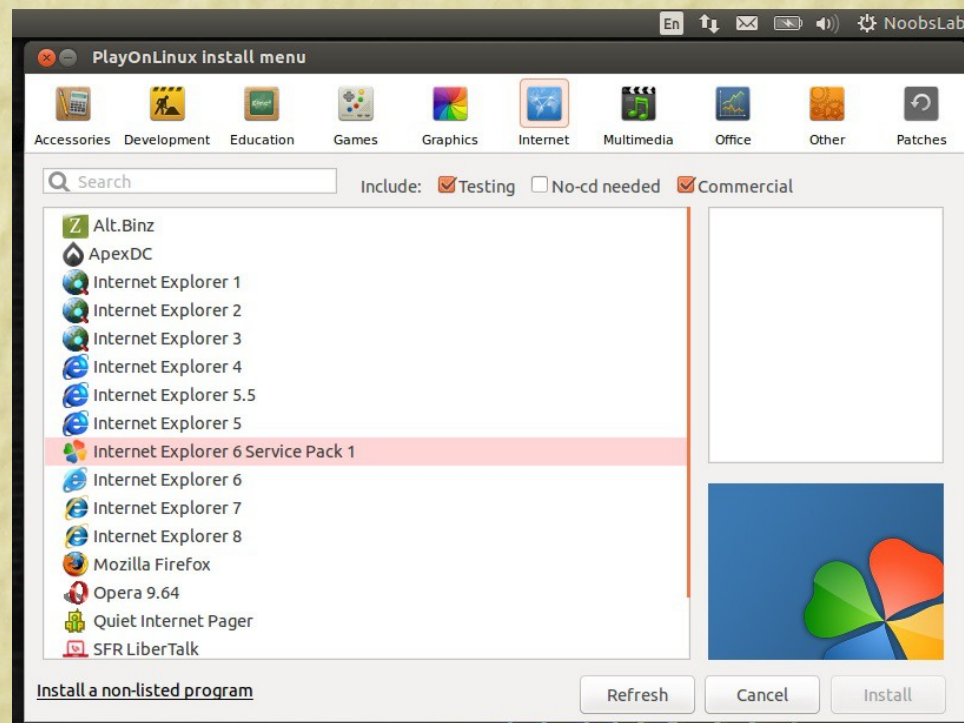
Mule (variantes)



Audacity

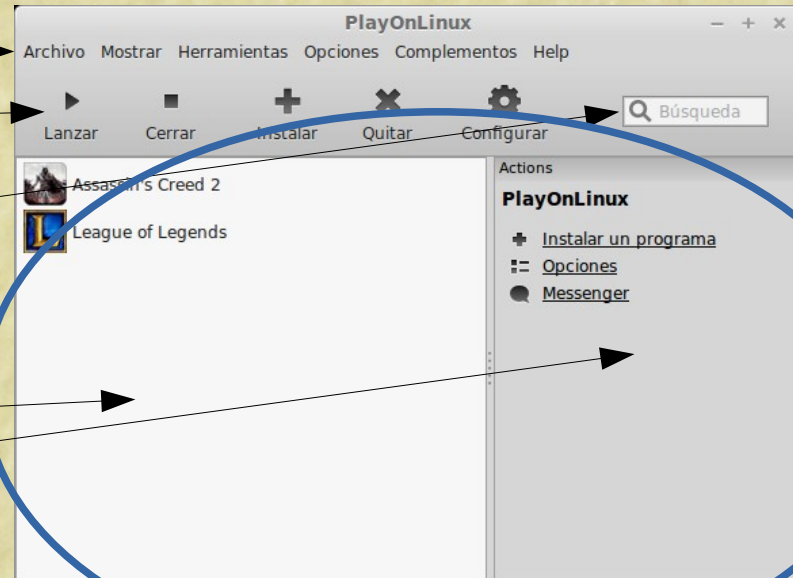


PlayOnLinux

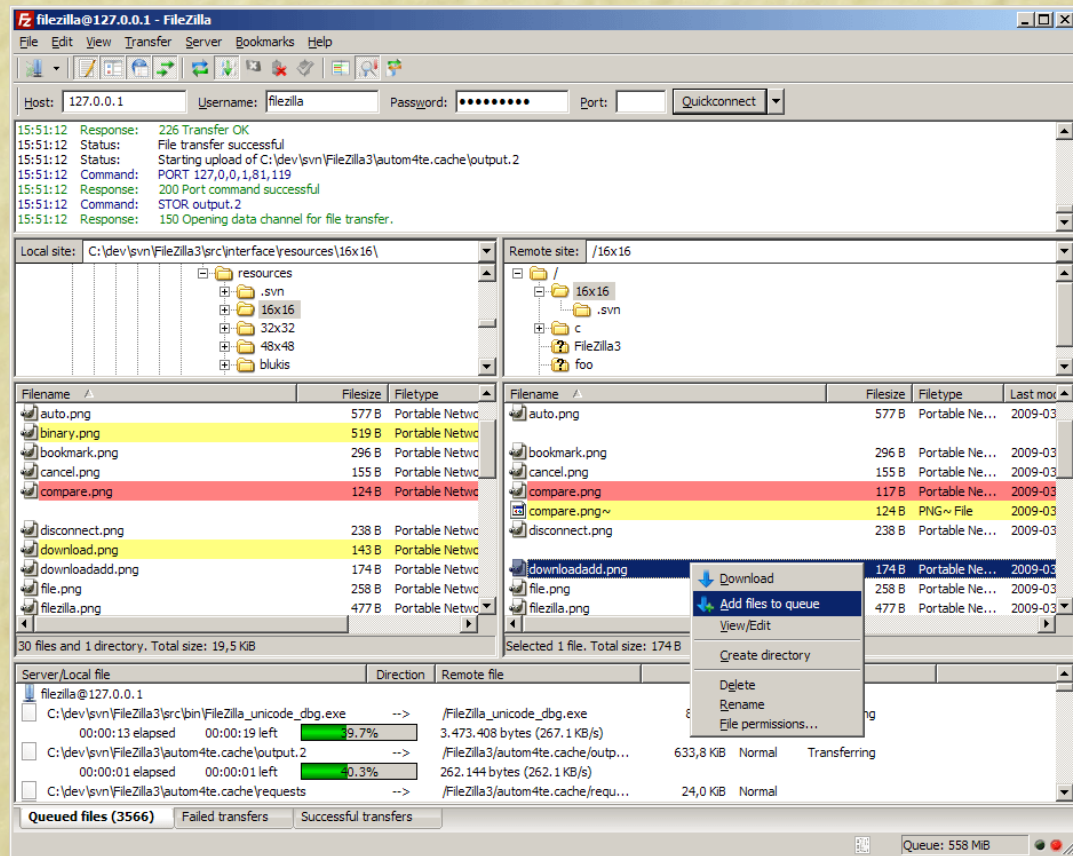


PlayOnLinux pantalla

- Ventana principal → Frame
- Menu
- ToolBar
 - SearchCtrl
- AuiManager con un Panel
 - TreeCtrl
 - auiPanelInfo
- StatusBar



FileZilla



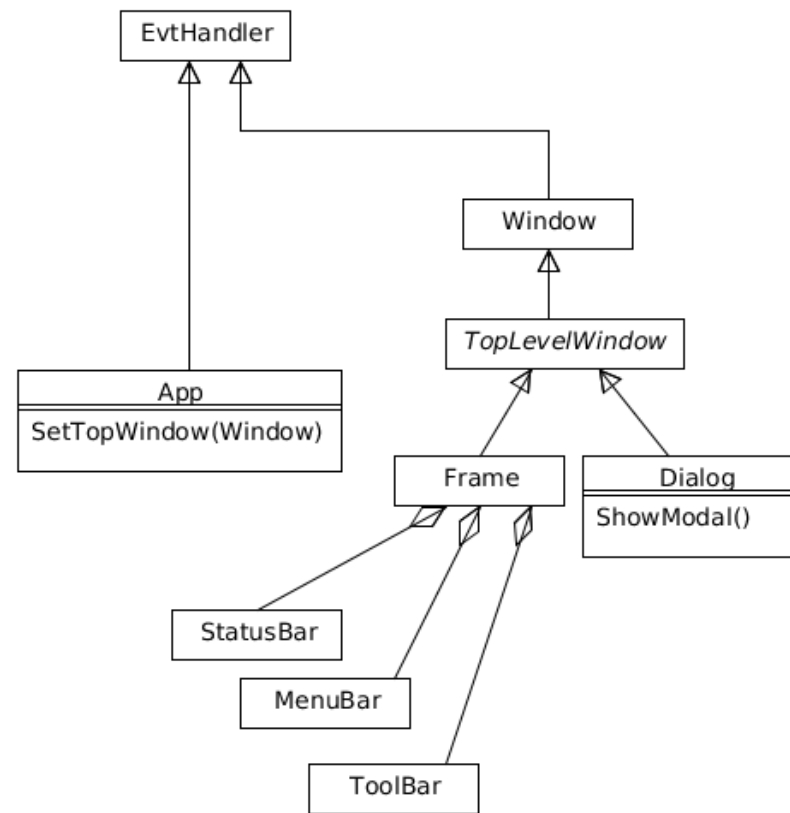
wxPython

Bajando a tierra

Conceptos básicos

- Orientado a eventos
- Raíz de todo → Aplicación
- Una o varias ventanas raíz
- Resto de ventanas forman una jerarquía
- Componentes se colocan mediante *Sizers*
- Creación de componentes mediante código o como recursos
-

Aplicación básica



Primer caso

```
import wx
```

```
app = wx.App(False) # Creación del objeto Aplicación.  
frame = wx.Frame(None, wx.ID_ANY, "¡Hola!") # Creación de  
Frame como ventana principal.  
frame.Show(True)    # Mostrar la ventana  
app.MainLoop()      #Bucle de eventos de aplicación  
#Si llega aquí, es que ha terminado aplicación
```

Creación de interfaces

- XRCed → Caso mínimo
- Dev-C++ (wxDev-C++)
- WxGlade
- wxFormBuilder

Futuro

- WxWidgets 3
- WxPython – Phoenix
 - Rehacer forma de bindings
 - Python 3
 - Widgets nuevos (PDF viewer)
- AMBOS muy activos. WxWidgets en GSOC 2014

Comparado con Qt

- Antes, ridículo 2 versiones: ANSI y UTF-8
- Ambos orientados a eventos, a su manera
- Qt “modifica C++”
- Qt mejor documentación, acercando diferencias
- Wx tiene una licencia más permisiva con (no-)comercial
- Wx más “cercano” al aspecto del S.O. de ejecución
- Python → wx lo tiene como “primerísimo” lenguaje
- Qt, 2 bindings

Comparando con GTK

- GTK es aspecto GTK en cualquier entorno
- GTK en Windows no esta “maduro”
- En X, mejor GTK



Sin límites

Gracias