



- ¿Qué es Qt?
- ¿Quién usa Qt?
- ¿Qué es PyQt?
- ¿Quién usa PyQt?



- ¿Qué es Qt?
 - Toolkit para interfaz gráfica
 - Maduro (más de 15 años de vida)
 - Origen comercial (actualmente propiedad de Nokia)
 - Open Source
 - Multiplataforma
 - C++

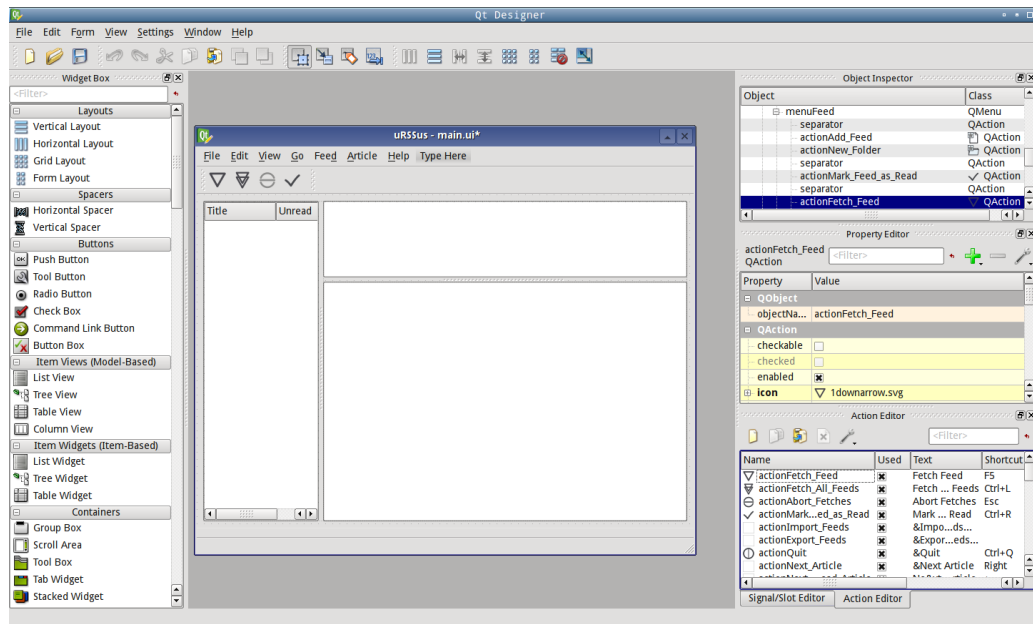
- ¿Quién usa Qt?
 - Google Earth
 - Adobe Photoshop Elements / Photoshop Album
 - KDE
 - VLC
 - Skype
 - Feko
 - Dispositivos con Qt Embedded:
<http://is.gd/2TTgk>

- ¿Qué es PyQt?
 - Qt desde Python
 - Open Source
 - Maduro (10 años de vida)
 - Soporte comercial (Riverbank Computing)

- ¿Quién usa PyQt?
 - Eric
 - Camelot
 - Treeline
 - Calibre
 - Yo
 - Muchos más!



Empezás con Designer:



- Genera XML
- XML => Python

Y si eso no te gusta...

- XML cargado desde Python
- Separa código de diseño
- Soporta
Signals/Slots/Actions/Resources/i18n

Si no lo usás...



Si no lo usás...



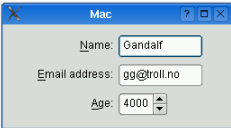
No estamos conectando!

```
1  # Este es el módulo generado por designer
2  from windowUi import Ui_MainWindow
3
4  class Main(QtGui.QMainWindow):
5      def __init__(self):
6          QtGui.QMainWindow.__init__(self)
7
8          # This is always the same
9          self.ui=Ui_MainWindow()
10         self.ui.setupUi(self)
11
12     def main():
13         # Esto es lo mismo en cada aplicación que escribas.
14         app = QtGui.QApplication(sys.argv)
15         window=Main()
16         window.show()
17         sys.exit(app.exec_())
18
19     if __name__ == "__main__":
20         main()
```

Es como Python: viene con las pilas puestas!

- Webkit (objeto web browser)
- OpenGL (para hacer 3D)
- Phonon (Sonido / Video)
- Red (cliente y servidor)
- Browser de ayuda (para usar desde tu aplicación)
- SVG
- Y cosas que capaz no necesitás
 - Threading
 - SQL
 - Javascript

- Windows
- Linux
- Mac
- S60
- Embedded
- Se pone ***mucho*** cuidado en hacer que se comporte como una aplicación nativa en cada plataforma soportada.

QCommonStyle derived styles (except QPlastiqueStyle)	QMacStyle	QPlastiqueStyle	Qt Extended styles
			
Traditional style used for Windows, GNOME, and earlier versions of KDE. Labels are left aligned, and expanding fields grow to fill the available space. (This normally corresponds to what we would get using a two-column QGridLayout .)	Style based on the Mac OS X Aqua guidelines. Labels are right-aligned, the fields don't grow beyond their size hint, and the form is horizontally centered.	Recommended style for KDE applications . Similar to MacStyle, except that the form is left-aligned and all fields grow to fill the available space.	Default style for Qt Extended styles. Labels are right-aligned, expanding fields grow to fill the available space, and row wrapping is enabled for long lines.

Es exactamente el mismo código. Porqué son distintos?

Mientras uses solo las APIs de PyQt, y la biblioteca standard de python, tu aplicación funciona en todas estas plataformas sin modificaciones.

- Qt es multi-licencia



- Qt es multi-licencia
- Pero la que importa es la LGPL



- Qt es multi-licencia
- Pero la que importa es la LGPL
- PyQt es GPL



- Qt es multi-licencia
- Pero la que importa es la LGPL
- PyQt es GPL
- Sí, eso es un problema para algunos



- Qt es multi-licencia
- Pero la que importa es la LGPL
- PyQt es GPL
- Sí, eso es un problema para algunos
- Usen PySide entonces!



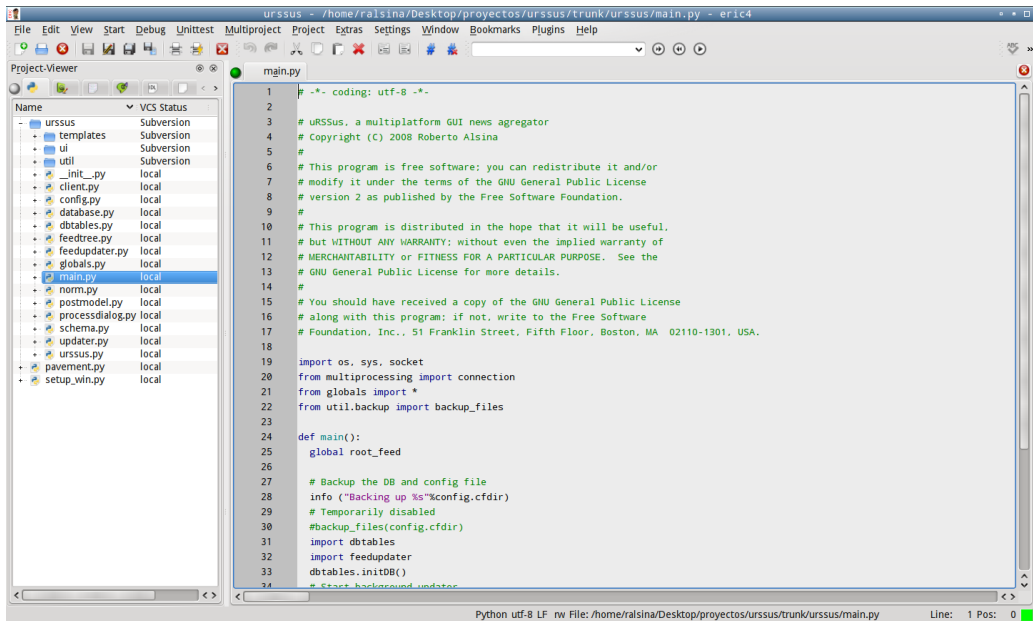
- Como todos
 - pyinstaller
 - py2exe
 - upx
 - etc, etc.
- Mide alrededor de 12MB máximo (incluye runtime python, y TODO PyQt)
- Se puede achicar a unos 5MB



ES TÉCNICAMENTE CIERTO!

xkcd, licencia Creative Commons
Attribution-Noncommercial 2.5

¿Alguna pregunta?



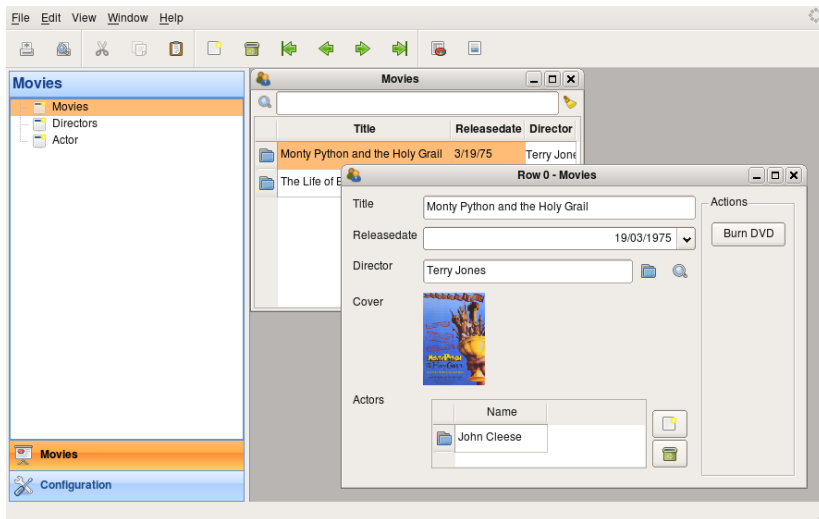
The screenshot shows the Eric IDE interface. On the left is the 'Project-Viewer' pane showing a tree structure of the project 'urssus'. The tree includes folders like 'templates', 'ui', 'util', and files like 'main.py', 'norm.py', 'postmodel.py', 'processdialog.py', 'schema.py', 'updater.py', 'urssus.py', 'pavement.py', and 'setup_win.py'. The 'urssus' folder is expanded, and 'main.py' is selected. On the right is the code editor, displaying the content of 'main.py'. The code is a Python script for a multiplatform GUI news aggregator, featuring a main function that initializes a database and starts a background updater.

```
1  # -*- coding: utf-8 -*-
2
3  # urssus, a multiplatform GUI news aggregator
4  # Copyright (C) 2008 Roberto Alsina
5  #
6  # This program is free software; you can redistribute it and/or
7  # modify it under the terms of the GNU General Public License
8  # version 2 as published by the Free Software Foundation.
9  #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program; if not, write to the Free Software
17 # Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
18
19 import os, sys, socket
20 from multiprocessing import connection
21 from globals import *
22 from util.backup import backup_files
23
24 def main():
25     global root_feed
26
27     # Backup the DB and config file
28     info ("Backing up %s"%config.cfdir)
29     # Temporarily disabled
30     #backup_files(config.cfdir)
31     import dbtables
32     import feedupdater
33     dbtables.initDB()
34     # Start background updater
```

Volver a slide 6

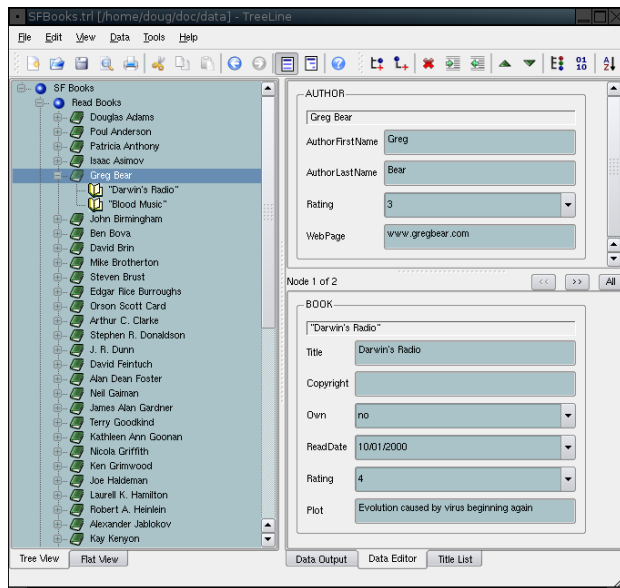
Camelot: Django, pero Desktop!

PyQt



Volver a slide 6

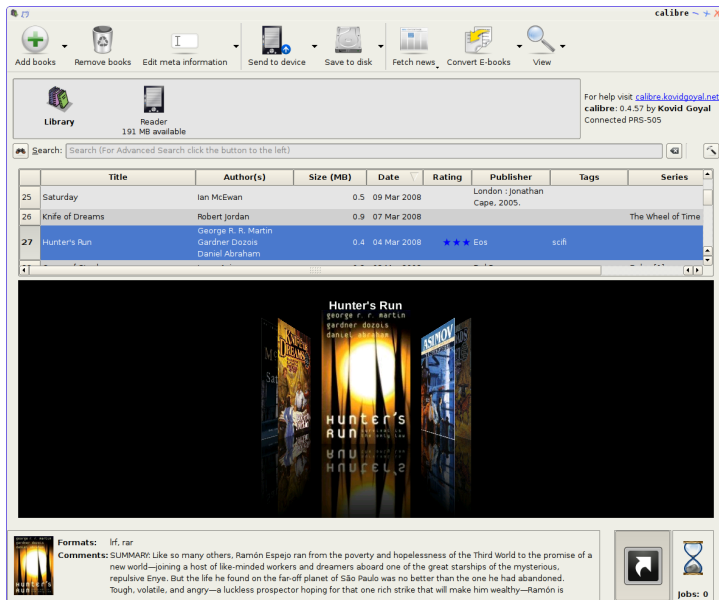
TreeLine: no es fácil de describir



Volver a slide 6

Calibre: increíble eBook manager

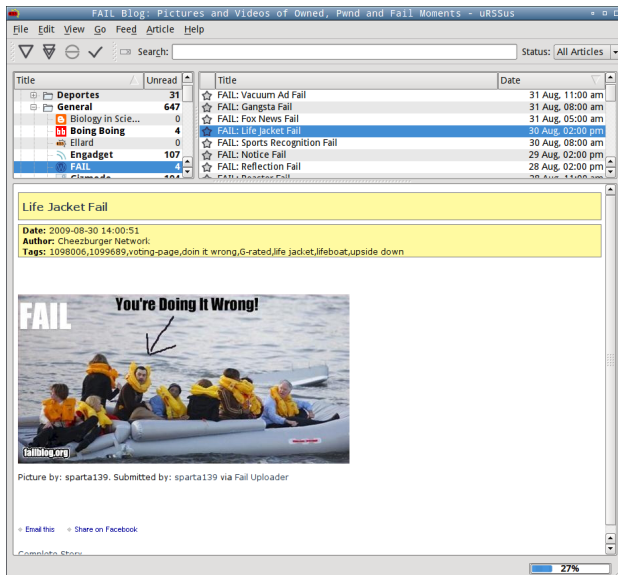
PyQt



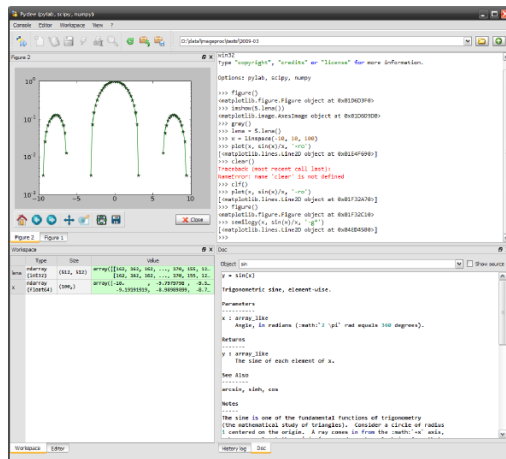
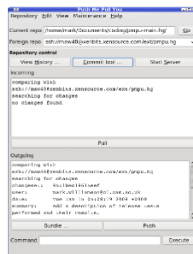
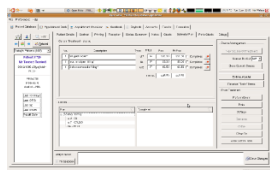
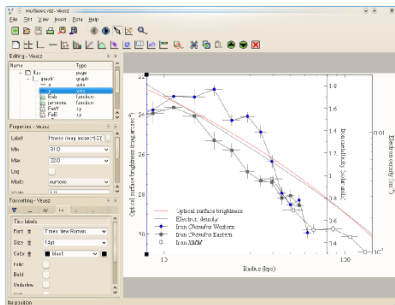
Volver a slide 6

Yo! en un montón de medias aplicaciones! PyQt

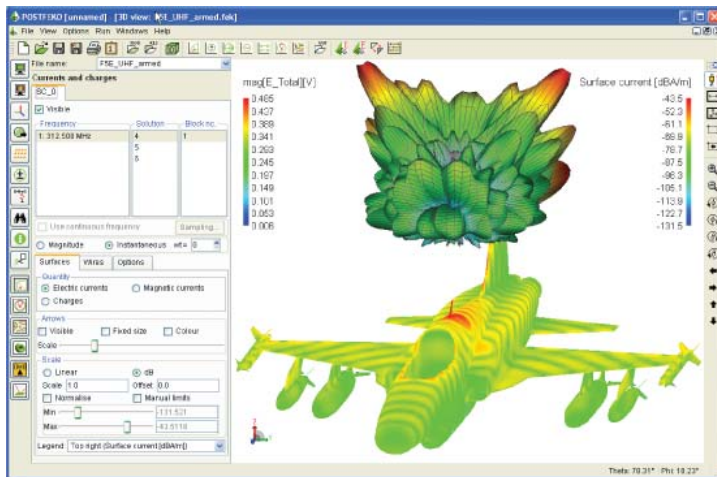
Esta es uRSSus: <http://urssus.googlecode.com>



Volver a slide 6



Volver a slide 6



¡Lo puse porque la imagen es linda!

Volver a slide 4