



**Linux Week
PUCP 2007**

PROGRAMACIÓN MULTIPLATAFORMA CON Qt

Geffrey Velásquez Torres

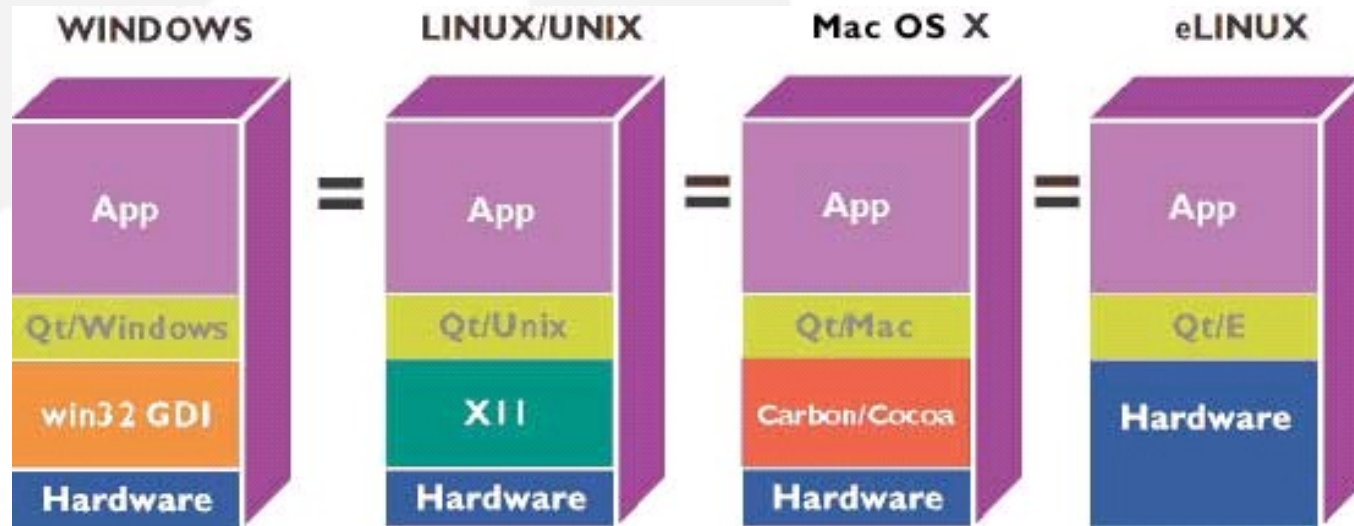


Qt: Framework C++ para Desarrollo Multiplataforma

- Qt es un framework en C++ para desarrollo de aplicaciones, el cual incluye:
 - Una biblioteca (library) de aproximadamente 400 clases C++, las cuales brindan las funcionalidades: GUI, Base de Datos, Core, XML, Networking, Open GL, Multithreading.
- Herramientas de desarrollo multiplataforma para: Desarrollo de Interfaces Gráficas (Designer), Internacionalización (Linguist), Documentación (Assistant), Generación de Makefiles (qmake).



Qt: Principales Plataformas Soportadas



- Linux and embedded Linux
- AIX, BSD/OS, FreeBSD, HP-UX, IRIX, NetBSD, OpenBSD, Solaris, Tru64 UNIX
- Desde Windows® 95 hasta Windows Server 2003
- Mac OS® X



Proyectos Desarrollados con Qt: Google Earth

TROLLTECH™

TROLLTECH
Hall of Fame

A **3D** Interface
to the **Planet!**

Google™

Want to know more about a specific location? Dive right in -- Google Earth combines satellite imagery, maps and the power of Google Search to put the world's geographic information at your fingertips. Google Earth puts a planet's worth of imagery and other geographic information right on your desktop. View exotic locales like Maui and Paris as well as points of interest such as local restaurants, hospitals, schools, and more.

Qt has enabled the Google Earth team to bypass tedious application infrastructure development and focus on delivering a world-class innovative user interface, providing a fresh new look at our physical environment.

Thanks to the cross-platform nature of Qt, it was also possible to provide a Mac version of Google with only a small fraction of the resources required for



Google™

Adobe

skype

myVR

PERFORCE

SYNOPSIS™

Perfect TablePlan

Crumhorn
Labs™



Proyectos Desarrollados con Qt: Skype

TROLLTECH™

TROLLTECH
Hall of Fame

Free **Calling**
Worldwide



Skype offers consumers free, superior-quality calling worldwide. Skype uses P2P (peer-to-peer) technology to connect users with other Skype users.

"Qt has greatly helped us to bring Skype for Linux to users across various Linux desktops and distributions. It neatly integrates with the user's desktop theme and helps us provide a consistent, easy-to-use, yet powerful user interface that is central to Skype across all the platforms it runs on."

www.skype.com



Google™

Adobe

skype™

myVR

PERFORCE

SYNOPSIS™

Perfect TablePlan

Crumhorn
Labs™



Proyectos Desarrollados con Qt: Adobe Photoshop Elements

TROLLTECH

TROLLTECH
Hall of Fame

Adobe® **Photoshop®**
Elements



Adobe Photoshop Elements helps photo enthusiasts find, fix, share and protect their digital photos. Adobe Photoshop Elements combines the functionality of Adobe Photoshop Album with some of the Adobe Photoshop functionality. The software received the "Best of CES" award in Digital Imaging at the 2006 Consumer Electronics Show in Las Vegas.

"Qt simplified our task by providing high-level tools we could customize to meet our needs," said Mike DePaoli, Photoshop Album Engineering Manager. "The product is excellent, the support was outstanding and we are extremely pleased with our decision to go with Qt".

<http://www.adobe.com>



Google



skype



PERFORCE

SYNOPSYS

Perfect TablePlan

Crumhorn
Labs



Proyectos Desarrollados con Qt: K Desktop Environment (KDE)

TROLLTECH

TROLLTECH
Hall of Fame

The **K** Desktop Environment



The K Desktop Environment (KDE) is a network transparent contemporary desktop environment for UNIX workstations. KDE seeks to fill the need for an easy to use desktop for Unix workstations, similar to the desktop environments found under the Mac® OS or Microsoft® Windows®.

In addition to the core KDE applications like KMail, Konqueror, KOffice or KDevelop, a great number of Qt-based applications have been built for the K Desktop Environment, see <http://kde-apps.org>.

<http://www.kde.com>



PERFORCE

SYNOPSYS

PerfectTablePlan

Crumhorn
Labs





Primeros Pasos: Compilación e Instalación en LINUX / UNIX

- Requiere el tener instalado las herramientas de desarrollo GNU GCC:

```
tar -xvzf qt-x11-opensource-src-4.2.3.tar.gz
cd qt-x11-opensource-src-4.2.3
./configure
gmake
gmake install
```

- La instalación se realiza por defecto en /usr/local/Trolltech/Qt-4.2.3, siendo necesario agregar al \$PATH:

```
export PATH=$PATH:/usr/local/Trolltech/Qt-4.2.3/bin
```



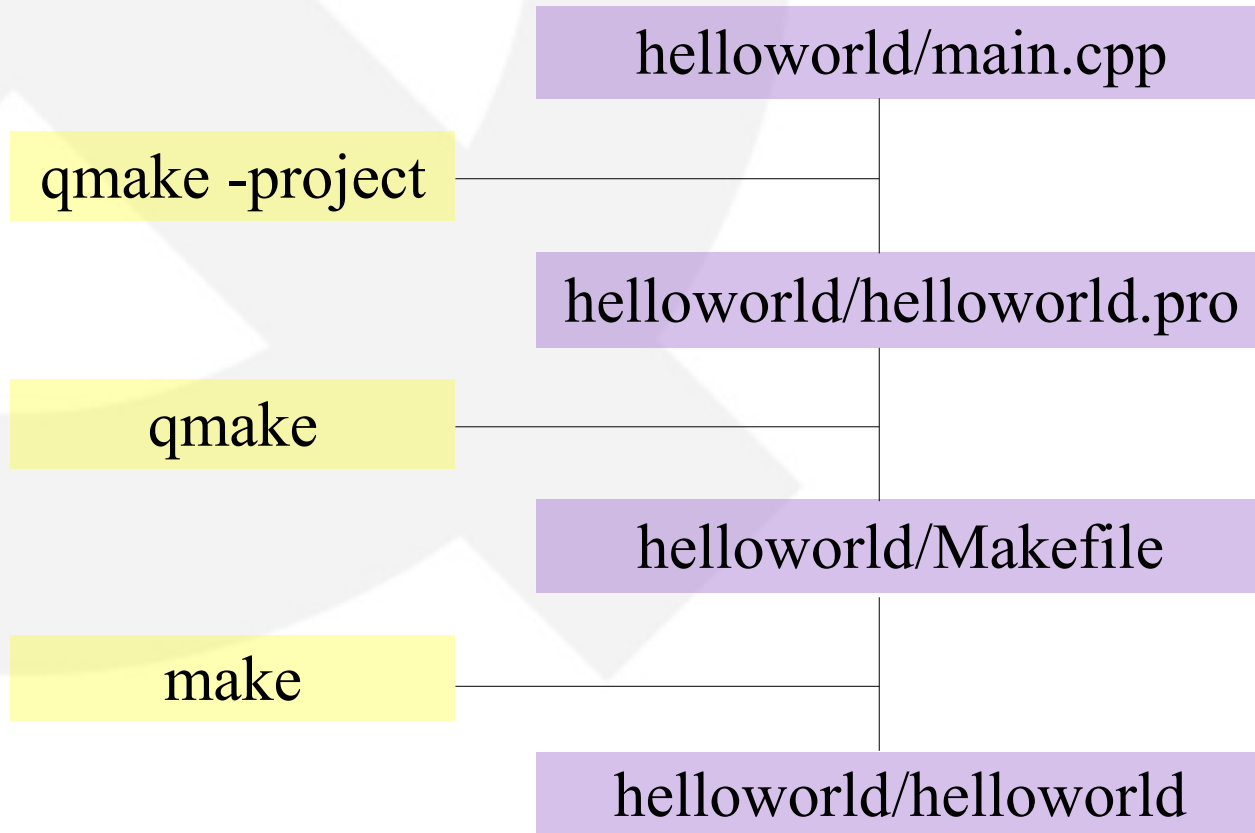

Primeros Pasos: Instalación en Windows (Next, Next, Next ...)

- La versión Open Source requiere Mingw, el cual es descargado durante el proceso de instalación.
- La versión comercial se integra con el compilador de Visual C++ 6, 2003, 2005.
- Técnicamente es posible utilizar la versión Open Source con Visual C++, aplicando un parche y recompilando la fuentes:

http://qtnode.net/wiki/Qt4_with_Visual_Studio



Primeros Pasos: ¡Hola Mundo!





Primeros Pasos: ¡Hola Mundo!

Archivo: helloworld.cpp

```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QPushButton hello("Hello world!");
    hello.resize(100, 30);
    hello.show();
    return app.exec();
}
```

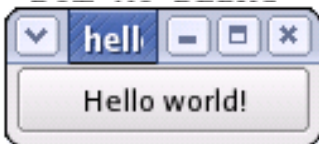


Primeros Pasos: ¡Hola Mundo!

Compilando y ejecutando:

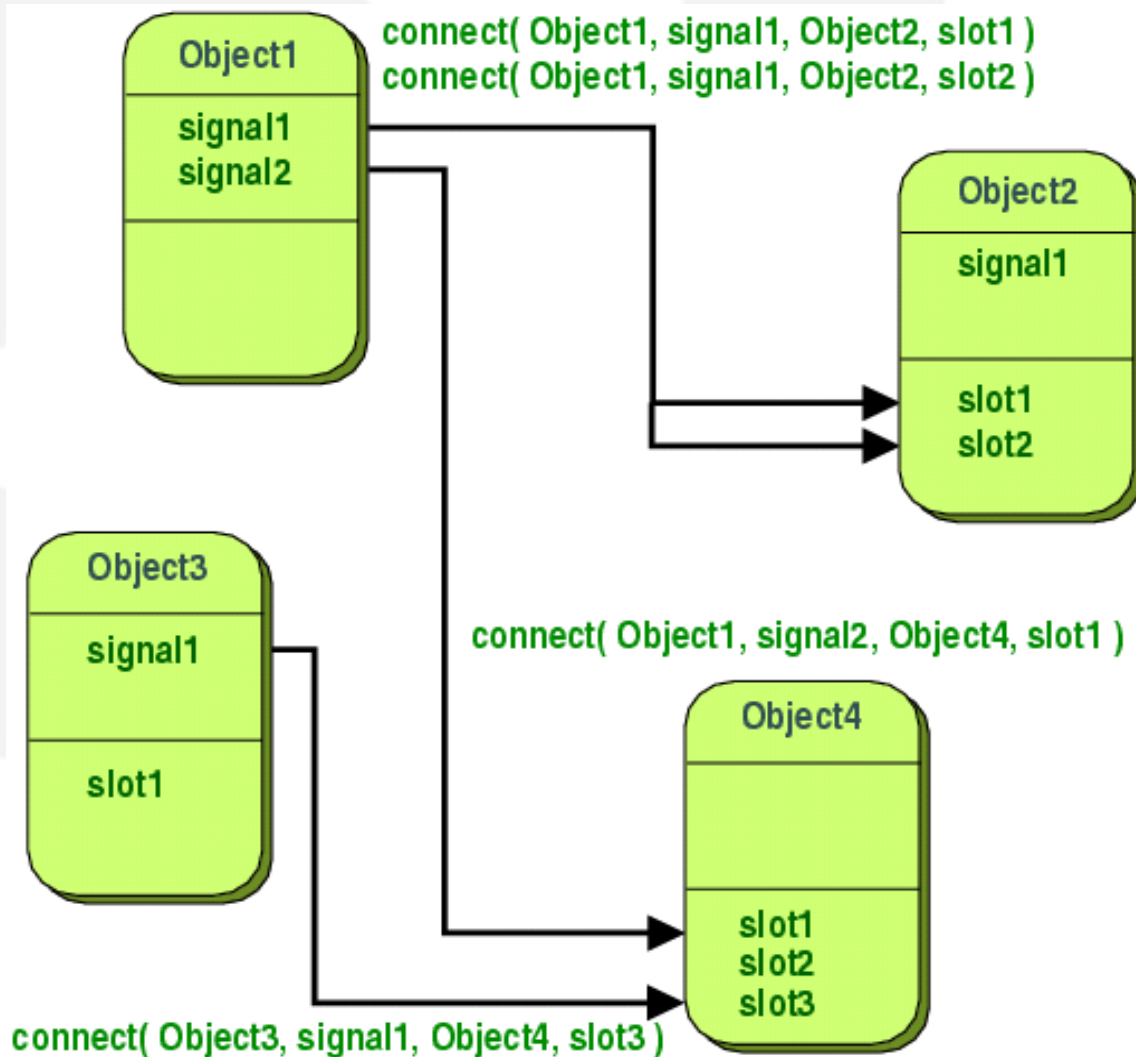
```
genfrey@dev:/usr/local/Trolltech/Qt-4.2.3/examples/tutorial/helloworld
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

[root@dev helloworld]# ls
main.cpp
[root@dev helloworld]# qmake -project
[root@dev helloworld]# qmake
[root@dev helloworld]# make
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_GUI_LIB -DQT_CORE_LIB -DQT_SHARED -
../../../../mkspecs/linux-g++ -I. -I../../../../include/QtCore -I../../../../in
de/QtGui -I../../../../include/QtGui -I../../../../lib -I. -I. -o main.o main.cpp
g++ -Wl,-rpath,/usr/local/Trolltech/Qt-4.2.3/lib -o helloworld main.o -L/usr/local/Trolltec
t-4.2.3/lib -lQtGui -L/home/geffrey/Desktop/qt-x11-opensource-src-4.2.3/lib -L/usr/X11R6/lib -
g -lSM -lICE -pthread -pthread -lXi -lXrender -lXrandr -lXfixes -lXcursor -lXinerama -lfreetyp
lfontconfig -lXext -lX11 -lQtCore -lz -lm -pthread -lgthread-2.0 -lglib-2.0 -ldl -lpthread
[root@dev helloworld]# ./helloworld
```





Primeros Pasos: Signals and Slots



- Signals and Slots son utilizados para la comunicación entre objetos, este mecanismo es una característica principal de Qt.



Primeros Pasos: Ejemplo de Signals and Slots

```
class Contador
{
    public:
        Contador() { m_valor = 0; }
        int valor() const { return m_valor; }
        void setValor(int valor);
    private:
        int m_valor;
};
```



Primeros Pasos: Ejemplo de Signals and Slots

```
#include <QObject>

class Contador : public QObject
{
    Q_OBJECT

public:
    Contador() { m_valor = 0; }
    int valor() const { return m_valor; }

public slots:
    void setValor(int valor);

signals:
    void valorChanged(int nuevoValor);

private:
    int m_valor;
};
```



Primeros Pasos: Ejemplo de Signals and Slots

Implementación de Contador::setValor(int valor):

```
void Contador::setValor(int valor)
{
    if (valor != m_valor) {
        m_valor = valor;
        emit valorChanged(valor);
    }
}
```




Primeros Pasos: Ejemplo de Signals and Slots

Contador a, b;

```
QObject::connect(&a, SIGNAL(valorChanged(int)),  
                &b, SLOT(setValor(int)));
```

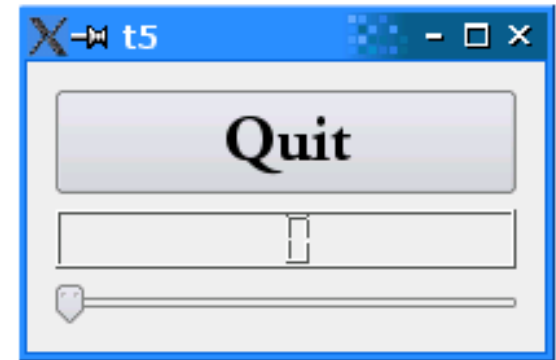
```
a.setValor(12); // a.valor()==12, b.valor()==12
```

```
b.setValor(48); // a.valor()==12, b.valor()==48
```



Primeros Pasos: Ejemplo Completo

- El movimiento del Slider deberá hacer cambiar el valor mostrado por el display LCD.
- Se realizará las conexiones utilizando Signals and Slots de los Objetos: QSlider y QLCDNumber.
- De igual forma al hacer click en el botón Quit deberá salirse de la aplicación.





Primeros Pasos: Ejemplo Completo

```
#include <QApplication>
#include <QFont>
#include <QLCDNumber>
#include <QPushButton>
#include <QSlider>
#include <QVBoxLayout>
#include <QWidget>

class MyWidget : public QWidget
{
public:
    MyWidget(QWidget *parent = 0);
};
```



Primeros Pasos: Ejemplo Completo

```
MyWidget::MyWidget(QWidget *parent)
    : QWidget(parent)
{
    QPushButton *quit = new
        QPushButton(tr("Quit"));
    quit->setFont(QFont("Times", 18,
        QFont::Bold));

    QLCDNumber *lcd = new QLCDNumber(2);
    lcd->setSegmentStyle(QLCDNumber::Filled);

    QSlider *slider = new
        QSlider(Qt::Horizontal);
    slider->setRange(0, 99);
    slider->setValue(0);
```



Primeros Pasos: Ejemplo Completo

```
connect(quit, SIGNAL(clicked()), qApp,  
        SLOT(quit()));  
connect(slider, SIGNAL(valueChanged(int)),  
        lcd, SLOT(display(int)));
```

```
QVBoxLayout *layout = new QVBoxLayout;  
layout->addWidget(quit);  
layout->addWidget(lcd);  
layout->addWidget(slider);  
setLayout(layout);  
}
```



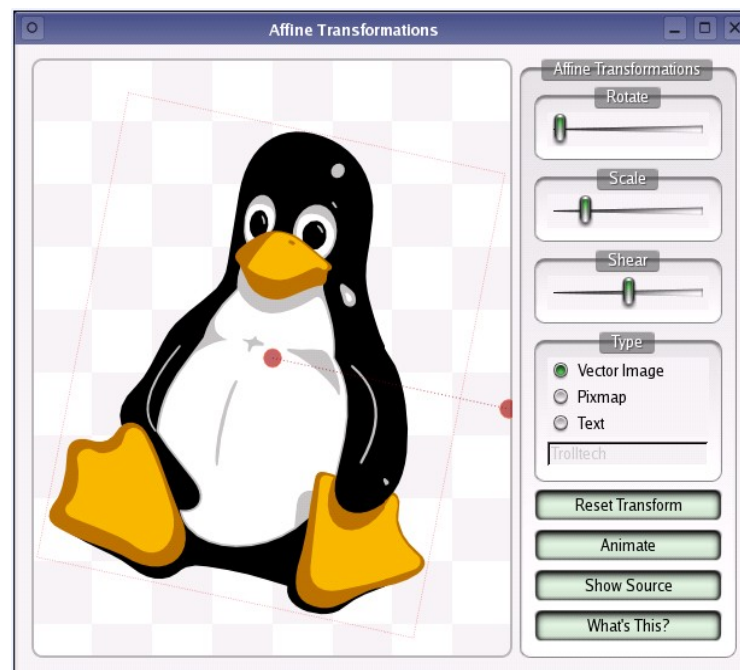
Primeros Pasos: Ejemplo Completo

```
int main(int argc, char *argv[])  
{  
    QApplication app(argc, argv);  
    MyWidget widget;  
    widget.show();  
    return app.exec();  
}
```



Características Destacadas: Subsistema de Dibujo

- El sistema de dibujo de Qt, permite el pintado en pantalla y dispositivos de impresión, utilizando la misma API.
- Provee funciones para el dibujo de puntos, líneas, rectángulos, polígonos, elipses, curvas.
- Las figuras pueden ser llenadas, utilizando la clase QBrush.
- Soporte de varios formatos de archivos raster y SVG.





Características Destacadas: Contenedores Genéricos

- La biblioteca Qt provee un conjunto de contenedores de propósito general, basados en C++ Templates, los cuales pueden almacenar elementos de todo tipo, por ejemplo: `Qlist<T>`, `QlinkedList<T>`, `Qvector<T>`, `Qstack<T>`, `Qqueue<T>`, `Qset<T>`, `QMap<Key, T>`, `QMultiMap<Key, T>`, `QHash<Key, T>`, `QMultiHash<Key, T>`.
- Están diseñadas para ser veloces, seguras y más fáciles de usar que los contenedores STL, pudiéndose utilizar iteradores basados en STL o Java. Se provee la palabra clave “foreach”.
- Los contenedores pueden ser utilizados en conjunto con los algoritmos genéricos provistos por Qt: `qBinaryFind`, `qSort`, `qSwap`, etc.

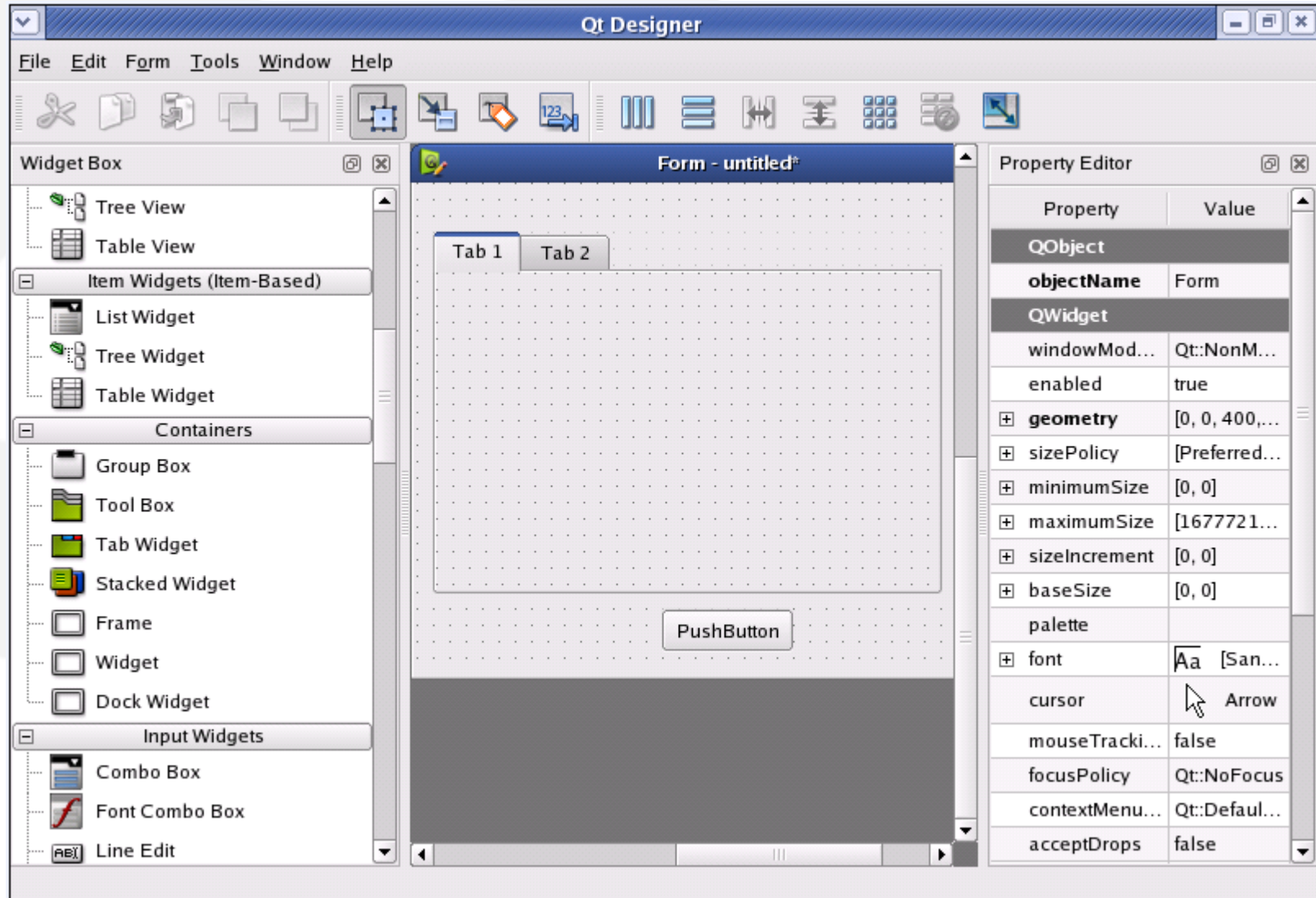


Características Destacadas: Otros componentes

- Programación Modelo / Vista (MVC) y Graphics View.
- Programación Multihilos.
- Programación de Redes: QAbstractSocket, QFtp, QHttp, QHttpHeader, QHttpRequestHeader, QHttpResponseHeader, QTcpServer, QTcpSocket, QUdpSocket, etc.
- Programación OpenGL.
- Soporte de BDs SQL.
- Soporte de XML (SAX, DOM).



Herramientas: Qt Designer





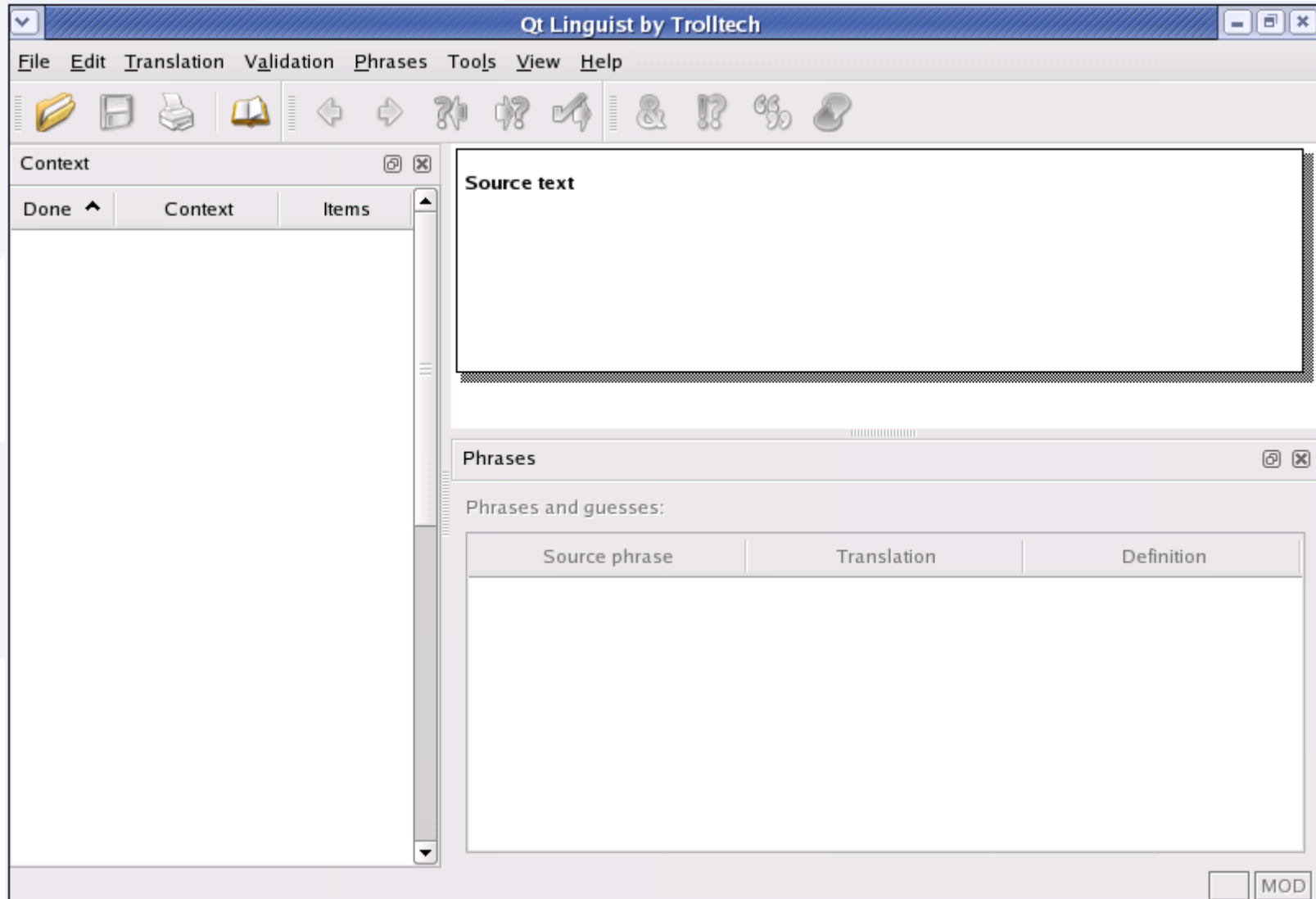
Herramientas: Qt Assistant

The screenshot shows the Qt Assistant application window titled "Qt Assistant by Trolltech - Qt Reference Documentation (Open Source Edition)". The window has a menu bar (File, Edit, View, Go, Bookmarks, Help) and a toolbar with various navigation icons. A sidebar on the left contains a "Contents" pane with a tree view of manuals: Qt Assistant Manual, Qt Designer Manual, Qt Linguist Manual, Qt Reference Documenten, and qmake Manual. The main content area displays the "Qt Reference Documentation (Open Source Edition)" page, which includes the Trolltech logo, navigation links (Home, All Classes, Main Classes, Grouped Classes, Modules, Functions), and a table of contents with the following sections:

Getting Started	General	Developer Resources
<ul style="list-style-type: none">• What's New in Qt 4.2• How to Learn Qt• Installation• Tutorial and Examples• Porting from Qt3 to Qt 4	<ul style="list-style-type: none">• About Qt• About Trolltech• Commercial Edition• Open Source Edition• Frequently Asked Questions	<ul style="list-style-type: none">• Mailing Lists• Qt Community Web Sites• Qt Quarterly• How to Report a Bug• Other Online Resources
API Reference	Core Features	Key Technologies
<ul style="list-style-type: none">• All Classes• Main Classes• Grouped Classes• Annotated Classes	<ul style="list-style-type: none">• Signals and Slots• Object Model• Layout Management• Paint System	<ul style="list-style-type: none">• Multithreaded Programming• Main Window Architecture• Rich Text Processing



Herramientas: Qt Linguist





Resumen: La Ventaja Competitiva de Qt

- Qt incrementa la productividad de los desarrolladores, haciendo la programación en C++ más veloz, fácil e intuitivo.
- Las herramientas de desarrollo eliminan los cuellos de botella comunes en el proceso de desarrollo:
 - Diseño de interfaces Gráficas – Qt Designer
 - Soporte de múltiples idiomas – Qt Assistant
 - Sistema de construcción multiplataforma – qmake
- Un solo código fuente, permite menor tiempo invertido en mantenimientos, multiplicando los resultados de los esfuerzos de desarrollo.
- Acceso completo a su código fuente, permite adaptar y extender Qt a necesidades particulares.



¡Muchas Gracias!

Geffrey Velasquez Torres

**email: geffrey.velasquez@pucp.edu.pe
geffrey@gmail.com**