



Programação Orientada a Objetos

Prof. Hugo Marcondes
hugo.marcondes@ifsc.edu.br

Aula 08

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina



- Introdução ao QT
 - Instalação do QT
 - Hello QT !
 - Conectando componentes
 - Layout dos componentes
- Desafio: Notepad !



"In the summer of 1990 Haavard [Nord] and Eirik [Chambe-Eng] were working together on a C++ database application for ultrasound images. The system needed to be able to run with a GUI on Unix, Macintosh, and Windows. One day that summer, Haavard and Eirik went outside to enjoy the sunshine, and as they sat on a park bench, Haavard said, "We need an object-oriented display system." The resulting discussion laid the intellectual foundation for the object-oriented cross-platform GUI framework they would soon go on to build."

Blanchette, J & Summerfield, M. (2006). C++ GUI Programming with Qt 4. Prentice Hall.

Quem usa QT ?



Fonte: [https://en.wikipedia.org/wiki/Qt_\(software\)#Organizations_using_Qt](https://en.wikipedia.org/wiki/Qt_(software)#Organizations_using_Qt)

QT History - Principais Marcos



- 1995 - Lançamento da primeira versão (Troll Tech)
- 1996 - KDE projeto estabelece o QT como biblioteca base
- 2008 - Nokia compra Trolltech, "QT Software at Nokia"
- 2011 - Digia compra o negócio de licenciamento comercial do QT
- 2012 - Digia compra todos os direitos sobre o QT
- 2014 - Digia cria uma subsidiária para cuidar do QT ("The Qt Company")
- 2016 - Separação da Digia e QT

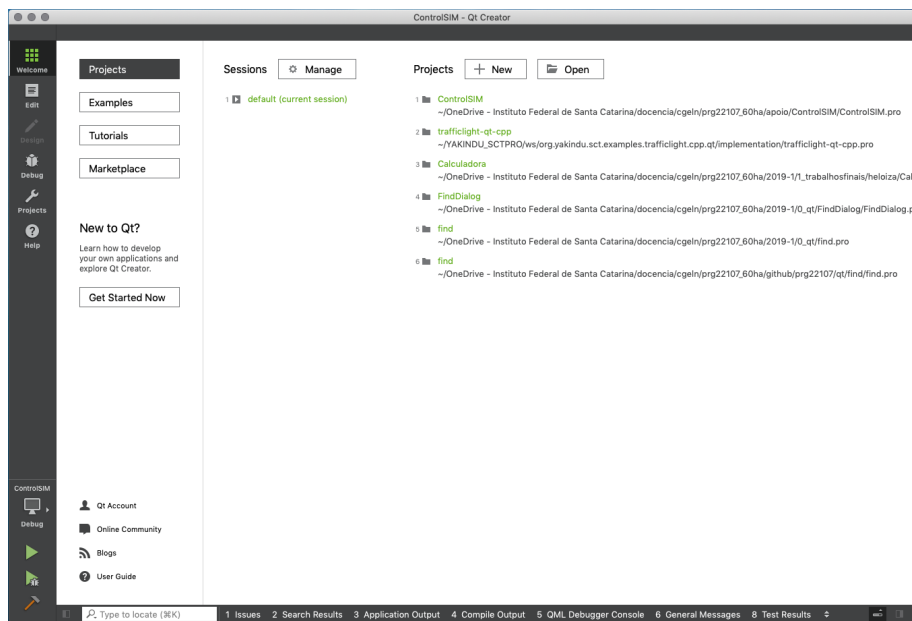


- Framework para o desenvolvimento de aplicações baseado em C++
 - Mais que uma simples biblioteca gráfica!
 - Pode ser utilizado também para programas sem interface gráfica
 - Prover interoperabilidade de código !
- Funcionalidade provida por diversos módulos

- **Qt Core**: Principais componentes não gráficos do QT.
- **Qt GUI**: Componentes gráficos do QT
- Qt Multimedia: Componentes Multimídia (audio, vídeo, rádio, etc)
- **Qt Network**: Componentes para programação da rede (interoperabilidade)
- Qt QML: Componentes do QML e JavaScript
- Qt Quick: Framework para UI gráfica declarativa
- **Qt SQL**: Componentes para acesso a banco de dados
- Qt Test: Componentes para testes unitários
- Qt Webkit: Componentes para renderização WWW (Webkit2)
- Qt Webkit Widgets: Webkit 1 (compatibilidade versões antigas)
- Qt Widgets: Extensão do Qt GUI para C++ Widgets

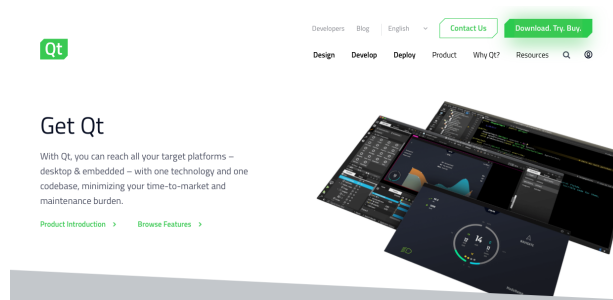


- IDE para o desenvolvimento de aplicação QT
 - Ferramentas de auxílio para o uso do QT
- Pode ser utilizado apenas com GCC + QT Lib
 - qmake para produzir a configuração do projeto, utilizando Makefiles (GNU make)
 - Conjunto de regras que instruem como compilar um programa
 - Automatização do processo





- QT 5.12.3
 - Pacotes de instalação para diversas plataformas
 - Bibliotecas QT 5.13.3
 - QT Creator 4.12.4
- <http://www.qt.io/download-open-source/>





- Nosso primeiro exemplo clássico:
 - Hello World, QT-based :)

```
1 #include <QApplication>
2 #include <QLabel>

3 int main(int argc, char *argv[])
4 {
5     QApplication app(argc, argv);
6     QLabel *label = new QLabel("Hello Qt!");
7     label->show();
8     return app.exec();
9 }
```

- O QLabel pode receber em seu construtor uma “string” formatada com HTML :)

```
QLabel *label = new QLabel("<h2><i>Hello</i>"  
                           "<font color=red>Qt!</font></h2>");
```



- Este exemplo demonstra como o QT responde as ações que são realizadas pelo usuário
- Uma aplicação simples com um botão para clicar e sair do programa

```
1 #include <QApplication>
2 #include <QPushButton>

3 int main(int argc, char *argv[])
4 {
5     QApplication app(argc, argv);
6     QPushButton *button = new QPushButton("Quit");
7     QObject::connect(button, SIGNAL(clicked()),
8                      &app, SLOT(quit()));
9     button->show();
10    return app.exec();
11 }
```

- Os componentes podem ser organizados em uma janela do QT através de componentes específicos para Layout
- Componentes são “linkados” através de uma hierarquia (pai - filho)
 - QWidget - Cria uma janela
 - QHBoxLayout - Organiza seus componentes filhos de forma horizontal
 - QVBoxLayout - Organiza seus componentes filhos de forma vertical
 - QGridLayout - Organiza os componentes em uma grade



```
1 #include <QApplication>
2 #include <QHBoxLayout>
3 #include <QSlider>
4 #include <QSpinBox>

5 int main(int argc, char *argv[])
6 {
7     QApplication app(argc, argv);

8     QWidget *window = new QWidget;
9     window->setWindowTitle("Enter Your Age");

10    QSpinBox *spinBox = new QSpinBox;
11    QSlider *slider = new QSlider(Qt::Horizontal);
12    spinBox->setRange(0, 130);
13    slider->setRange(0, 130);

14    QObject::connect(spinBox, SIGNAL(valueChanged(int)),
15                     slider, SLOT(setValue(int)));
16    QObject::connect(slider, SIGNAL(valueChanged(int)),
17                     spinBox, SLOT(setValue(int)));
18    spinBox->setValue(35);

19    QHBoxLayout *layout = new QHBoxLayout;
20    layout->addWidget(spinBox);
21    layout->addWidget(slider);
22    window->setLayout(layout);

23    window->show();

24    return app.exec();
25 }
```

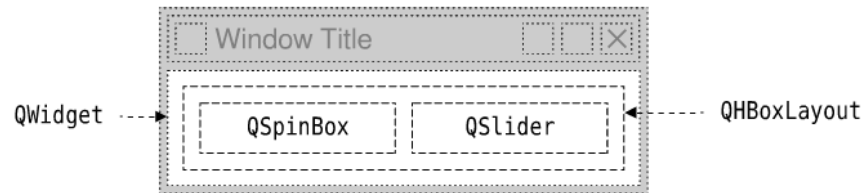


Figure 1.7. The Age application's widgets and layout

- QT Creator
 - Apresenta diversas opções de ajuda em contexto (F1)
- Documentação
 - Disponível em HTML no diretório de instalação do QT
 - Acessível por qualquer browser
- QT Assistant
 - Aplicativo para navegar na documentação com ferramentas de busca e indexação da informação

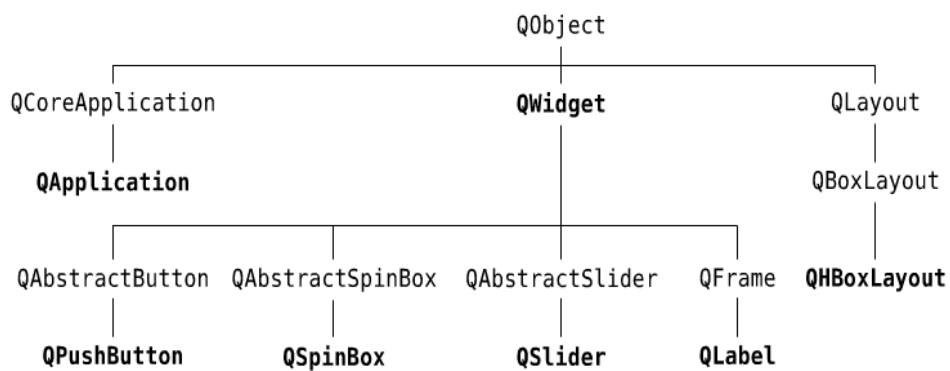


Figure 1.9. Inheritance tree for the Qt classes seen so far

Desafio !



INSTITUTO FEDERAL
SANTA CATARINA

- Vamos explorar mais QT
- Acesso a seguinte página e siga as instruções para programar o seu próprio Notepad, utilizando o QT
 - Contato com componentes distintos do QT
 - Desenhando interfaces com o QT Designer (UI)

<http://doc.qt.io/qt-5/qtwidgets-tutorials-notepad-example.html>