



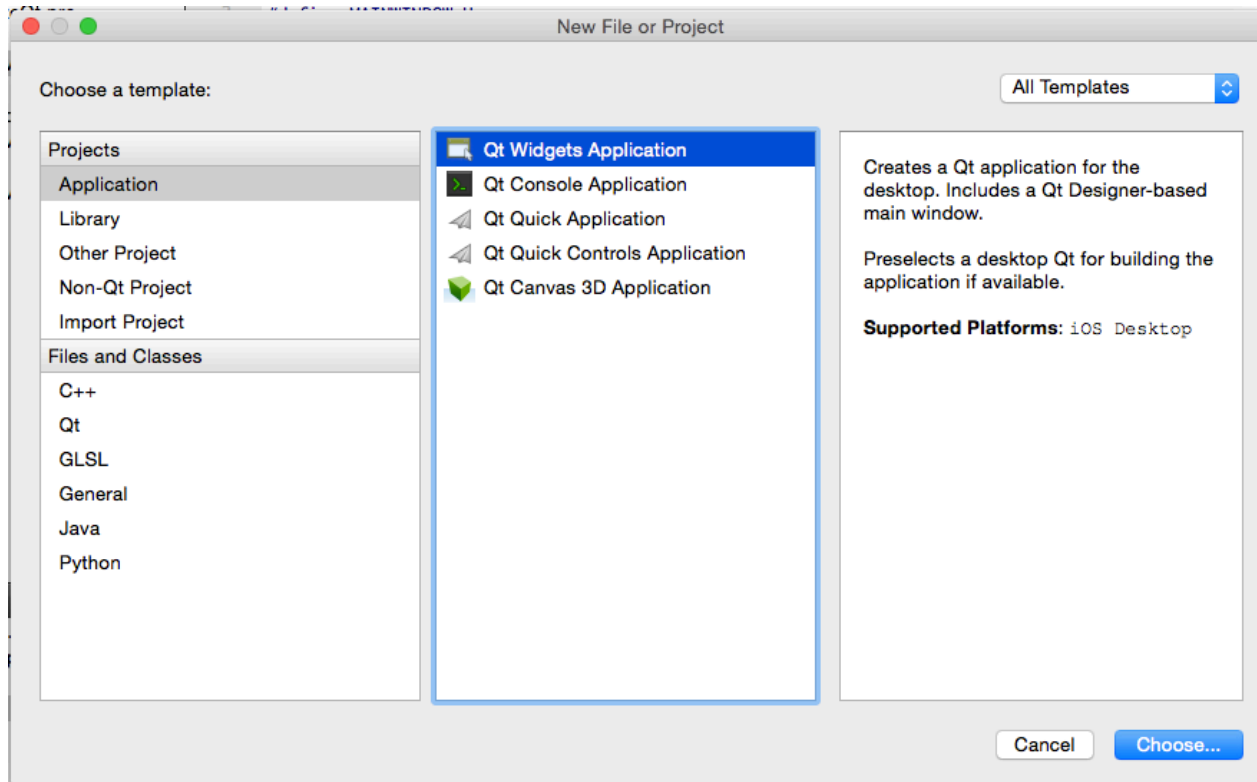
Usando GUI - QT

O QT

- Biblioteca aberta para criação de GUI
- Normalmente torna simples a utilização de interface gráfica
- Porta o build para várias plataformas
- Onde obter:
 - <http://www.qt.io/download/>
 - Free para usos sem fins comerciais

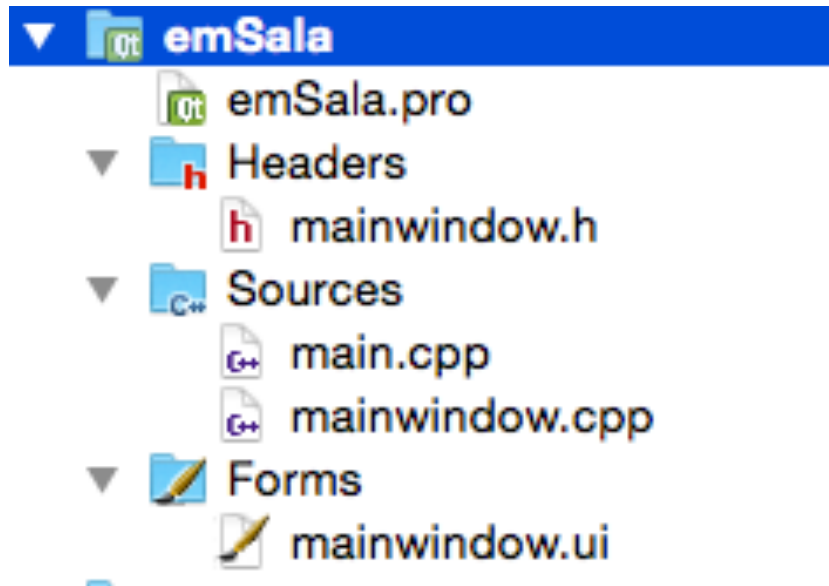
Começando

- Instale o QT
- Abra o QtCreator
- Clique para criar um projeto, selecione: QtWidgetsApplication



Configurando

- A seguinte árvore deve ter sido criada



.pro = arquivo do projeto
.ui = arquivo de interface

O arquivo de interface é sempre linkado com uma Classe em C++ onde estará as operações

Integrando com OpenCV

- Abra o arquivo .pro
- E inclua as configurações de INCLUDEPATH e LIBS

Mac e Linux:

```
INCLUDEPATH += /usr/local/include/
```

```
LIBS += -L/usr/local/lib -lopencv_core -lopencv_imgcodecs -  
lopencv_imgproc
```

Windows (eu acho):

```
INCLUDEPATH += C:\OpenCV2.2\include\
```

```
LIBS += -LC:\OpenCV2.2\lib -lopencv_core220 -lopencv_highgui220 -  
lopencv_imgproc220 -lopencv_features2d220 -lopencv_calib3d220
```

Como ficou

```
#-----  
#  
# Project created by QtCreator 2015-11-29T17:09:39  
#  
#-----  
  
QT      += core gui  
  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
TARGET = novoTesteQt  
TEMPLATE = app  
  
SOURCES += main.cpp\  
          mainwindow.cpp  
  
HEADERS  += mainwindow.h  
  
FORMS    += mainwindow.ui  
  
INCLUDEPATH += /usr/local/include/  
LIBS      += -L/usr/local/lib -lopencv_core -lopencv_imgcodecs -lopencv_imgproc
```

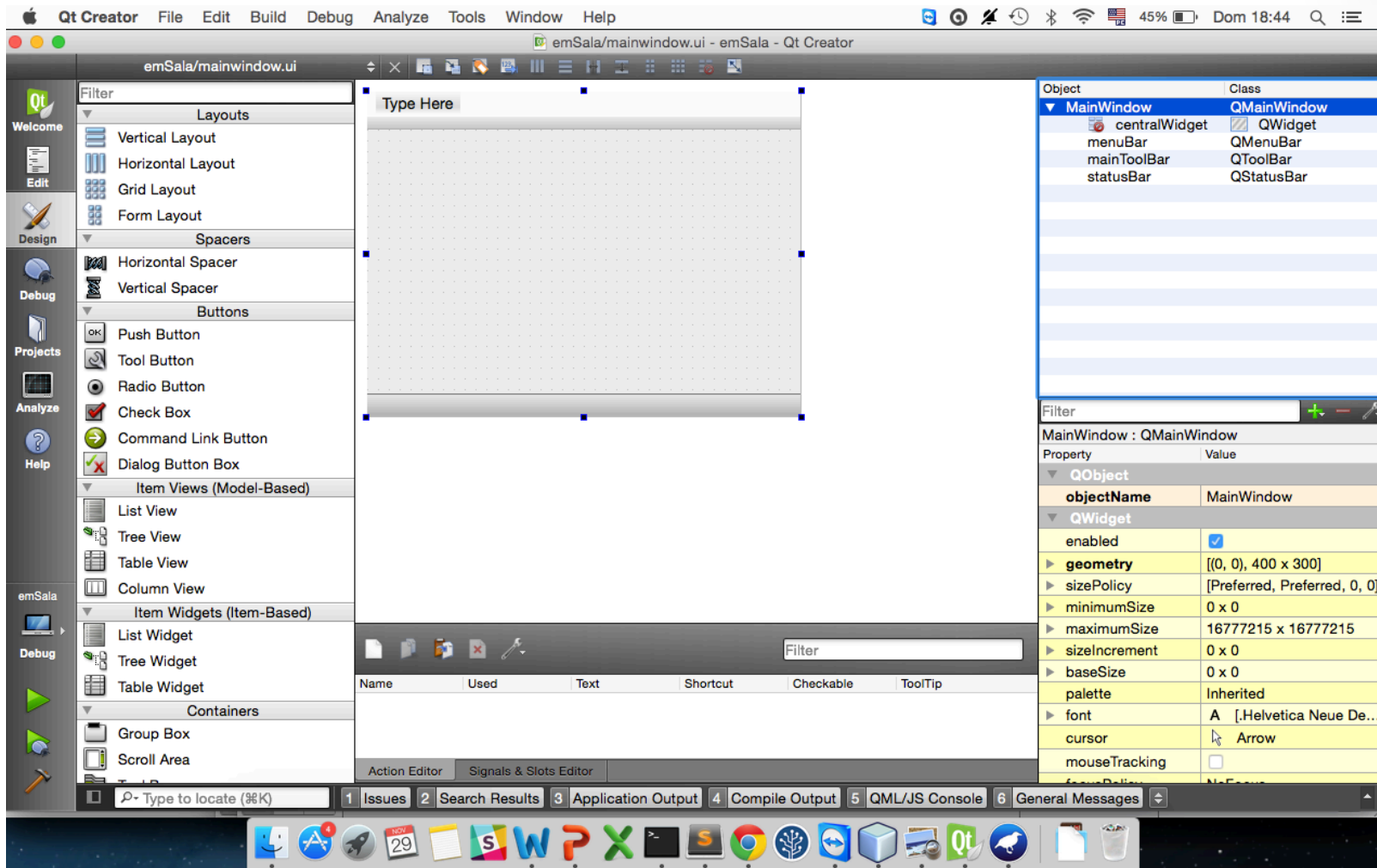
Abrindo o arquivo main.cpp

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

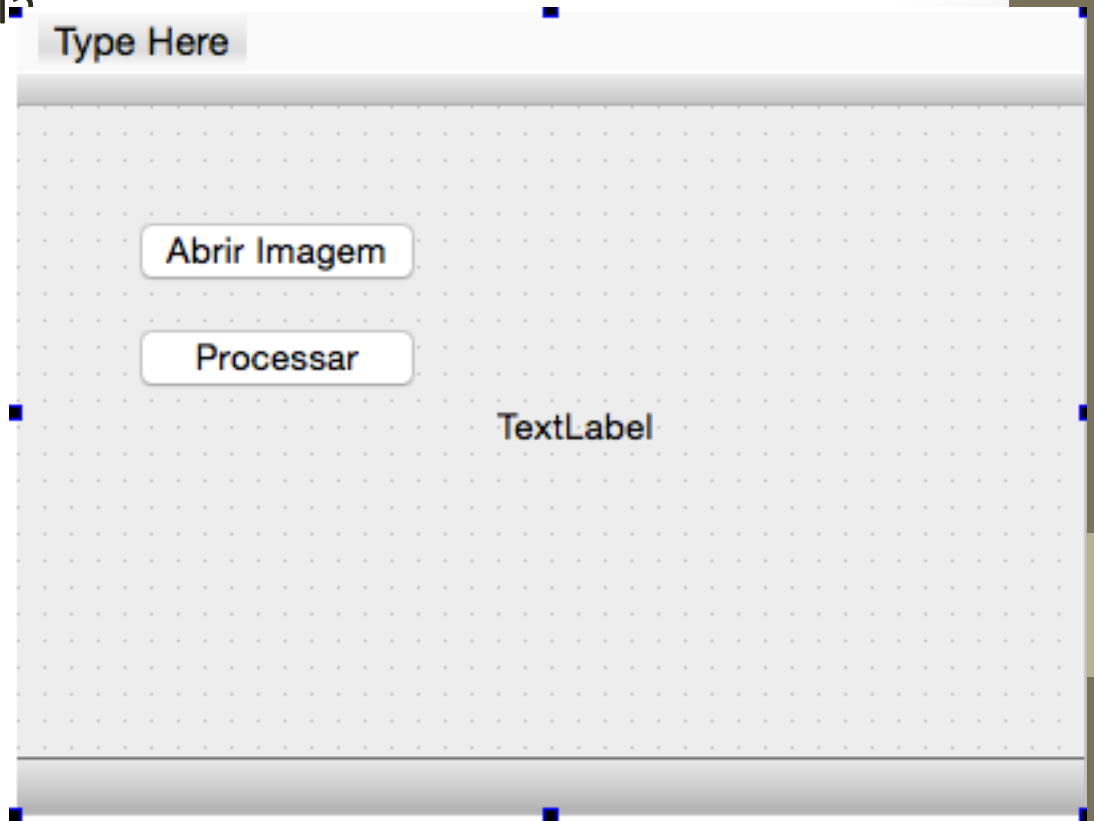
    return a.exec();
}
```

Abrindo o mainwindow.ui



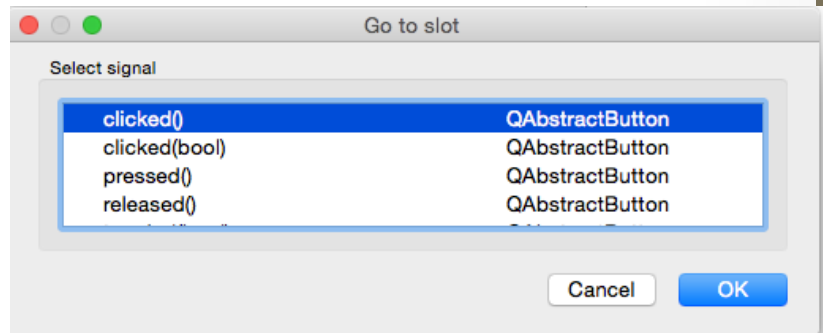
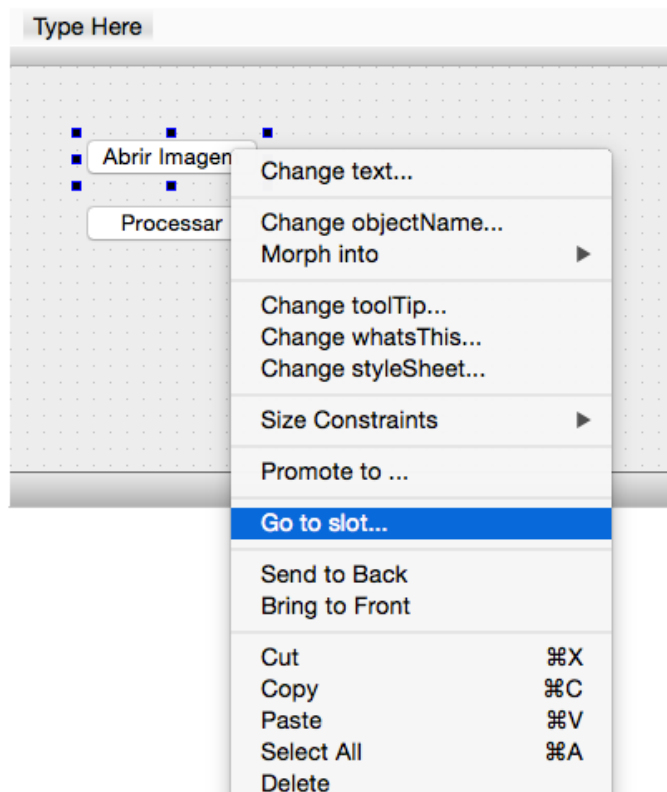
Hands on

- Inclua dois botões (Push Button):
 - Abrir imagem
 - Processar imagem
- Um label dentro da janela



Adicione eventos aos botões

- Botão direito sobre o botão, escolha go to slot
- Depois escolha o evento clicked()



Adicione o seguinte código para Abrir Imagem

```
void MainWindow::on_openBtn_clicked()
{
    QString fileName = QFileDialog::getOpenFileName(this,
        tr("Open Image"), ".",
        tr("Image Files (*.png *.jpg *.jpeg *.bmp)"));
    image= cv::imread(fileName.toStdString());

    cv::cvtColor(image,image,CV_BGR2RGB);

    QImage img= QImage((const unsigned char*)(image.data),
        image.cols,image.rows,QImage::Format_RGB888);

    ui->label->setPixmap(QPixmap::fromImage(img));
    ui->label->resize(ui->label->pixmap()->size());
}
```

Faça o seguinte para o Processar

```
void MainWindow::on_procBtn_clicked()
{
    cv::flip(image,image,0);

    QImage img= QImage((const unsigned char*)(image.data),
        image.cols,image.rows,QImage::Format_RGB888);

    ui->label->setPixmap(QPixmap::fromImage(img));
    ui->label->resize(ui->label->pixmap()->size());
}
```

Modifique a classe: mainwindow.h

- Inclua os cabeçalhos

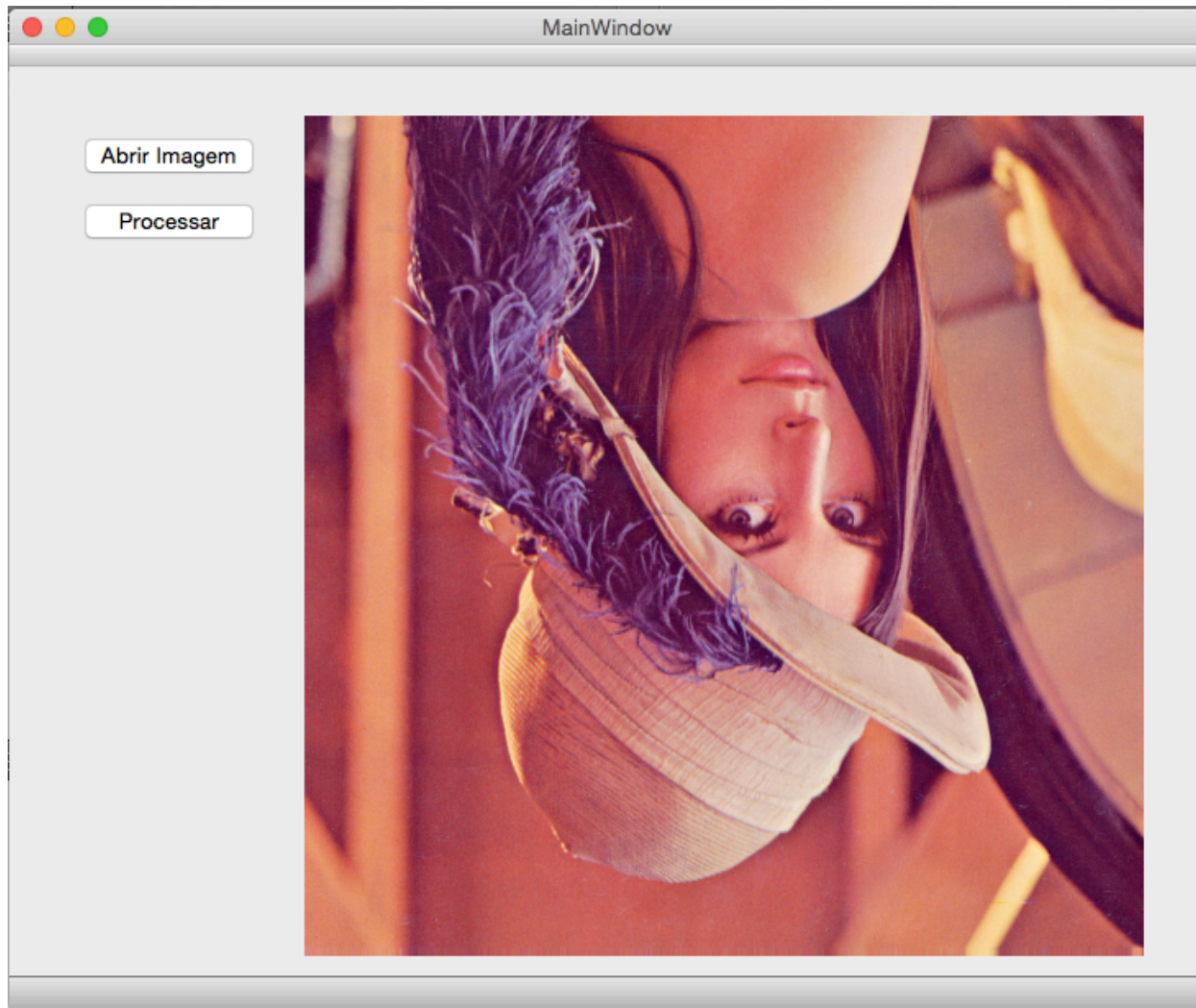
```
#include <QFileDialog>

#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
|
```

- Adicione em private, a declaração do atributo image:
- cv::Mat image;

```
private:
    Ui::MainWindow *ui;
    cv::Mat image;
```

Build and RUN!



Agora é sua vez

- Tente se acostumar com os componentes do QT
- E use sua documentação para criar aplicativos legais