

Universidade Tecnológica Federal do Paraná
UTFPR – Campus Curitiba

Orientação a Objetos Programação em C++

Grupo de Slides 17 – Parte D:

Programação Visual - Orientada a Eventos e Objetos Gráficos (Formulários, Botões, Caixas de Texto etc)
– ambiente RAD (*Rapid Application Development*)

Exemplificado no software **Qt/Qt Creator**

Prof. Jean Marcelo **SIMÃO**

Aluno monitor em 2011: Fabiano Cezar Domingos

Qt

- Qt é um *framework* para desenvolvimento de *softwares* em C++ para várias plataformas, incluindo dispositivos móveis.
- Prima pela facilidade da criação de GUIs (*Graphical User Interface*) e possibilita a utilização de diversos objetos gráficos em ambiente RAD (*Rapid Application Development*).
- É largamente empregado na indústria, sendo utilizado por empresas como Google, HP, NASA, Volvo e Disney.
- Qt Creator é o ambiente de desenvolvimento ou IDE (*Integrated Development Environment*) necessário para programar utilizando o Qt.



Qt

- O Qt é um *software* pago, mas também possui uma versão *open source* e gratuita para fins acadêmicos disponível em: <http://qt-project.org/downloads>
- Faça o download do *framework* Qt, ele já contém o ambiente Qt Creator incluso.
- Escolha o arquivo compatível com o seu sistema operacional.
- Prefira as versões com instalador online (*Online Installer*). Nelas é possível escolher qual será o compilador instalado com o programa, como por exemplo o MingW para Windows. Assim, o compilador será configurado automaticamente e não será preciso ajustar manualmente o compilador no Qt Creator.

Fazendo o Download

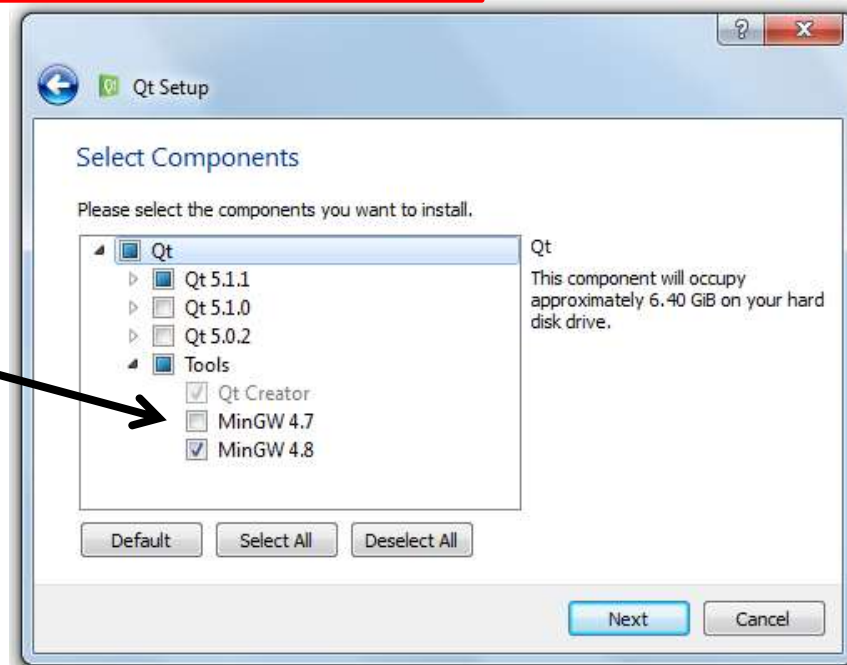


Qt 5.1

Select the file according to your operating system from the list below to get the latest Qt 5.1 for your computer. The binary packages include Qt 5.1.1 libraries and Qt Creator 2.8.1

- Qt Online Installer for Linux 32-bit (22 MB) (Info)
- Qt Online Installer for Linux 64-bit (22 MB) (Info)
- Qt Online Installer for Mac (9 MB) (Info)
- Qt Online Installer for Windows (13 MB) (Info)

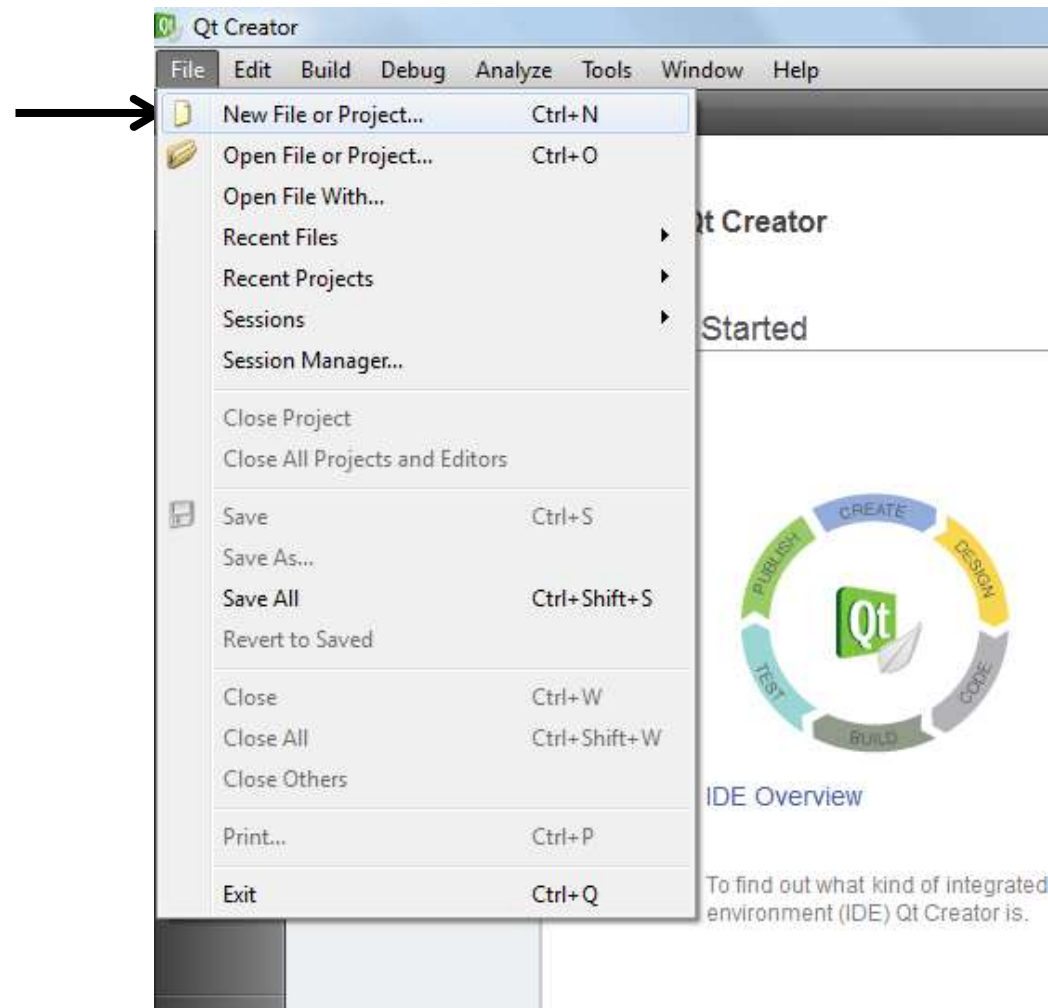
Online Installer:
Escolher
compilador



Exemplo 01: Introdução

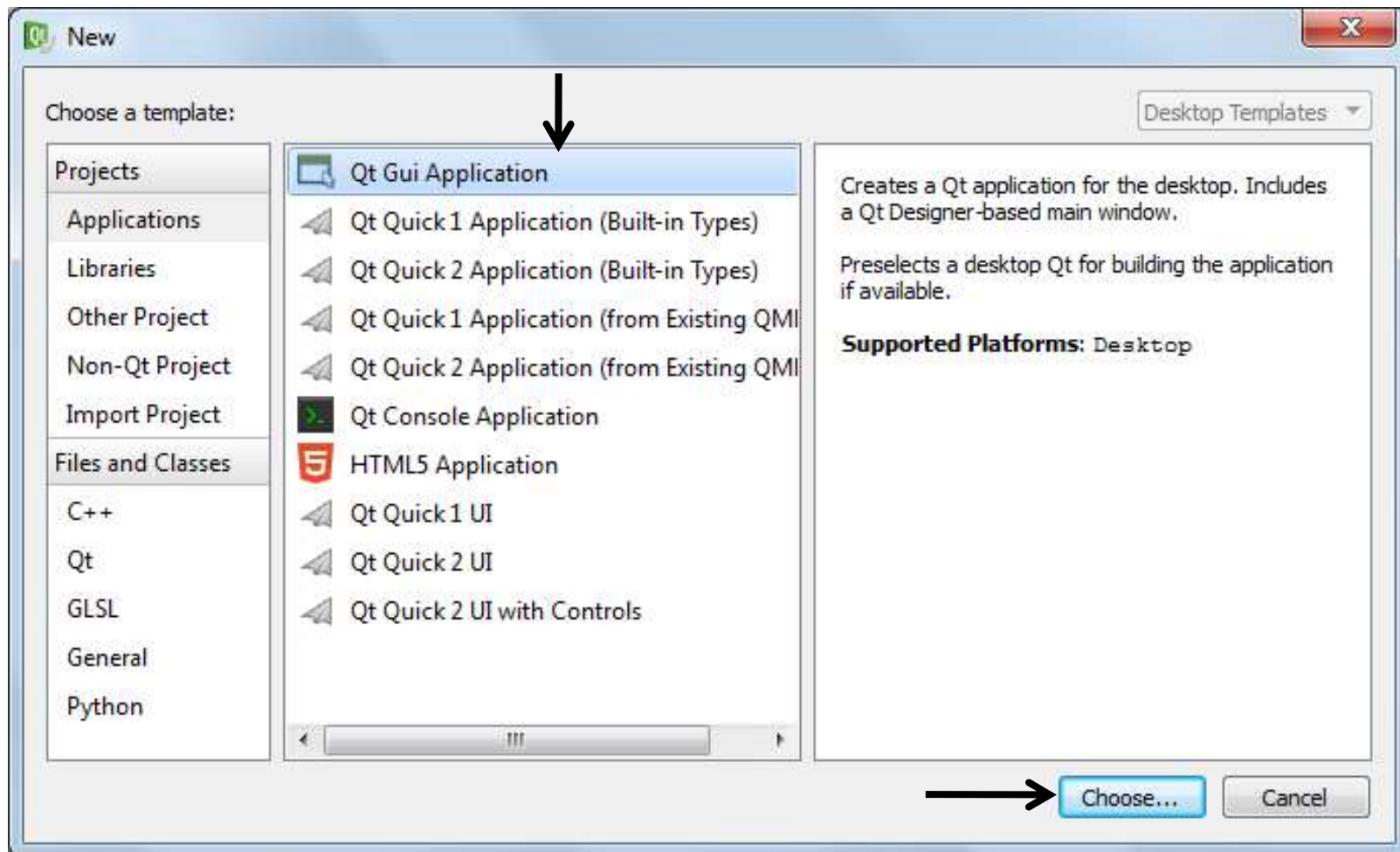
Exemplo 01

- Crie um novo projeto.



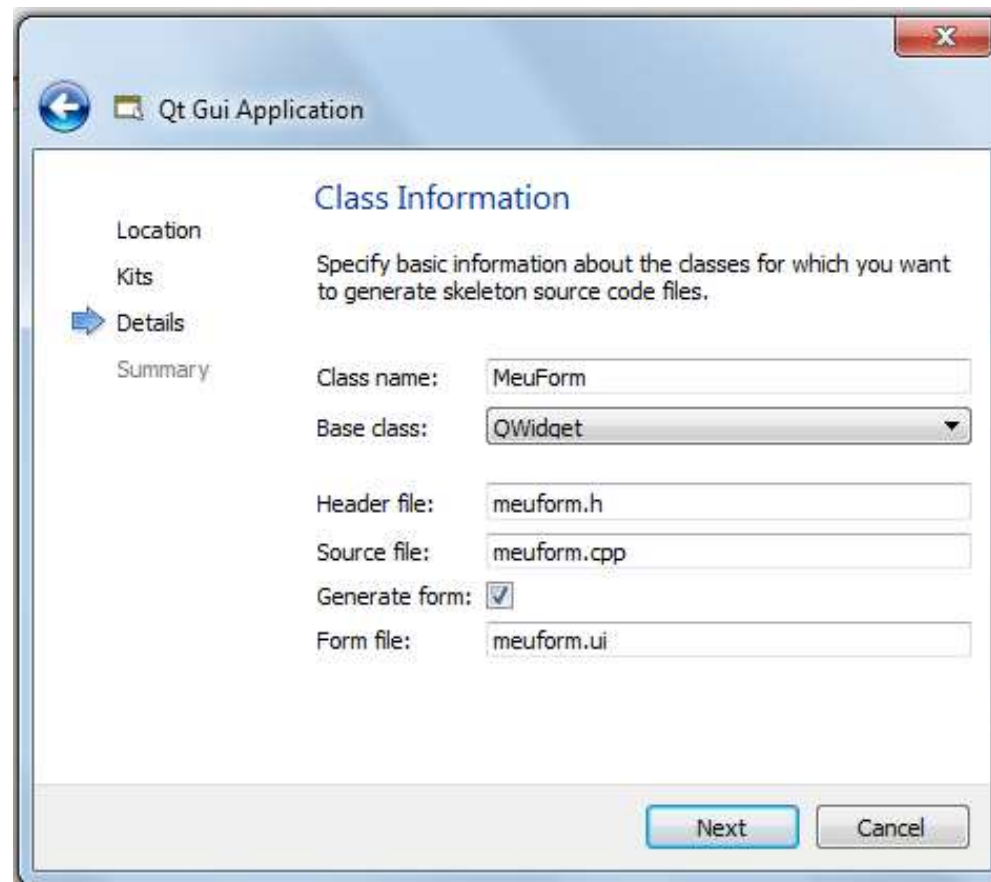
Exemplo 01

- Escolha o *template* “Qt Gui Application” e clique em “Choose”.



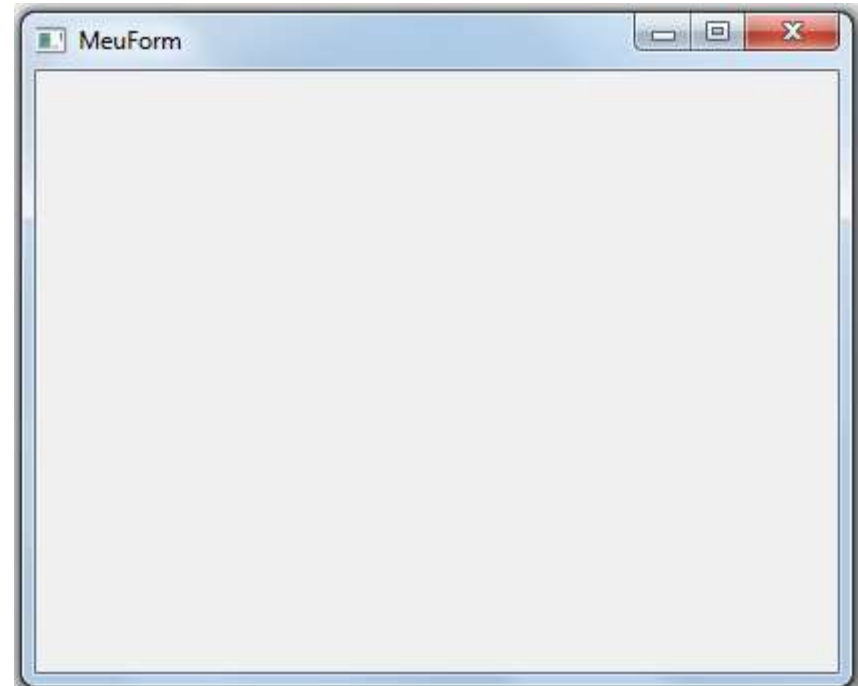
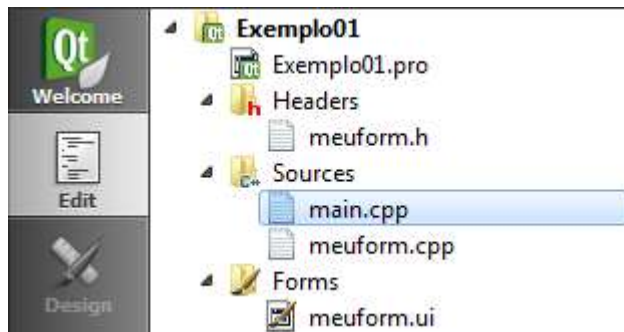
Criando o Projeto

- Uma nova janela será aberta e escolha o nome e diretório do projeto. Na aba “*Details*”, nomeie a classe como “MeuForm” e escolha a classe base “QWidget”.



Criando o Projeto

- O projeto será criado. O Qt Creator cria 3 arquivos de código automaticamente: meuform.h, meuform.cpp e main.cpp, mais o arquivo de design do formulário, meuform.ui.
- Basta clicar em “*Build*” > “*Run*” ou Ctrl+R para executar o código. O formulário “MeuForm” será mostrado.



Código

```
// meuform.h

#ifndef MEUFORM_H
#define MEUFORM_H
#include <QWidget>
namespace Ui { class MeuForm; }

class MeuForm : public QWidget
{
    Q_OBJECT
public:
    explicit MeuForm(QWidget *parent = 0);
    ~MeuForm();
private:
    Ui::MeuForm *ui;
};
#endif // MEUFORM_H
```

```
// meuform.cpp

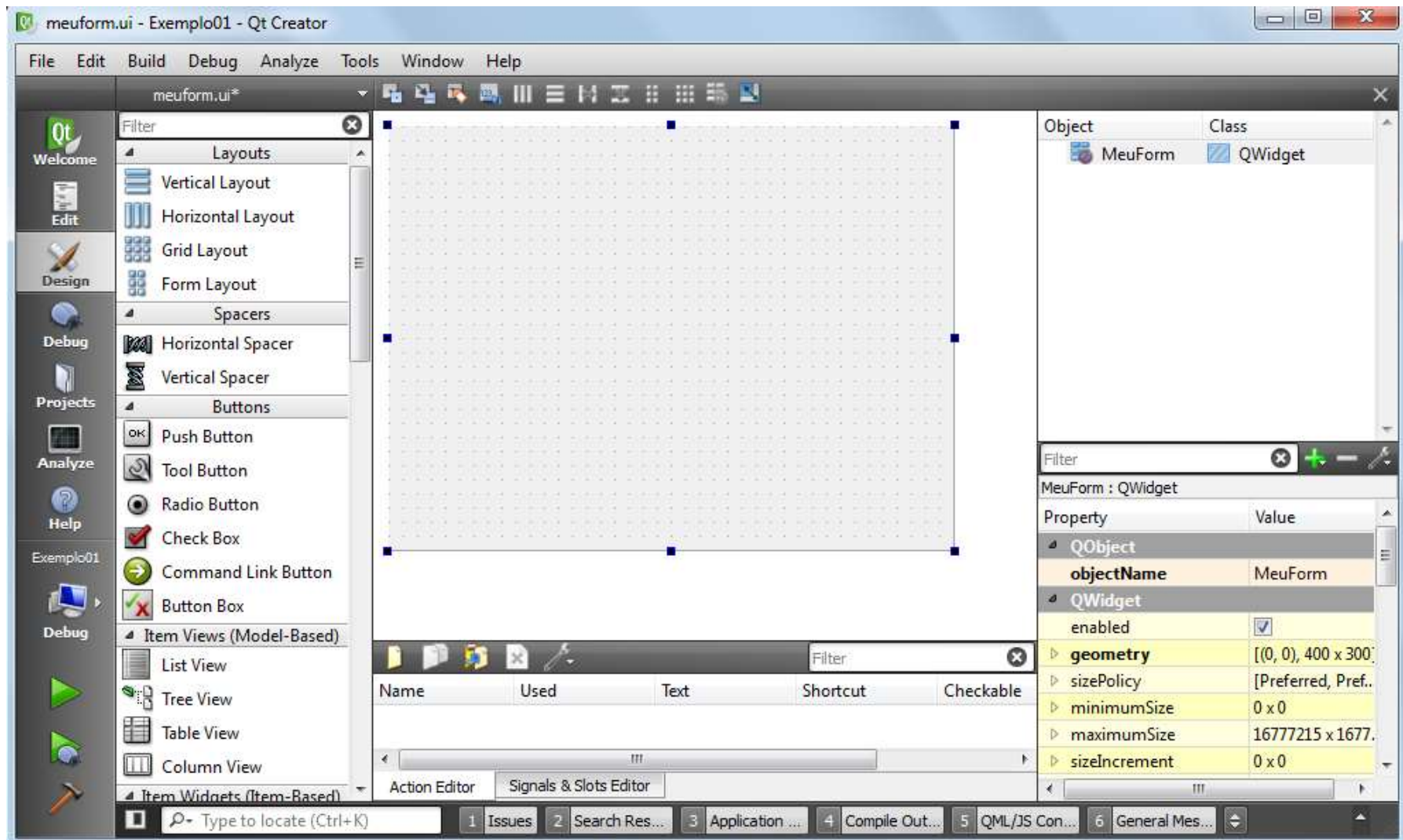
#include "meuform.h"
#include "ui_meuform.h"

MeuForm::MeuForm(QWidget *parent)
    : QWidget(parent),
      ui(new Ui::MeuForm)
{
    ui->setupUi(this);
}

MeuForm::~MeuForm()
{
    delete ui;
}
```

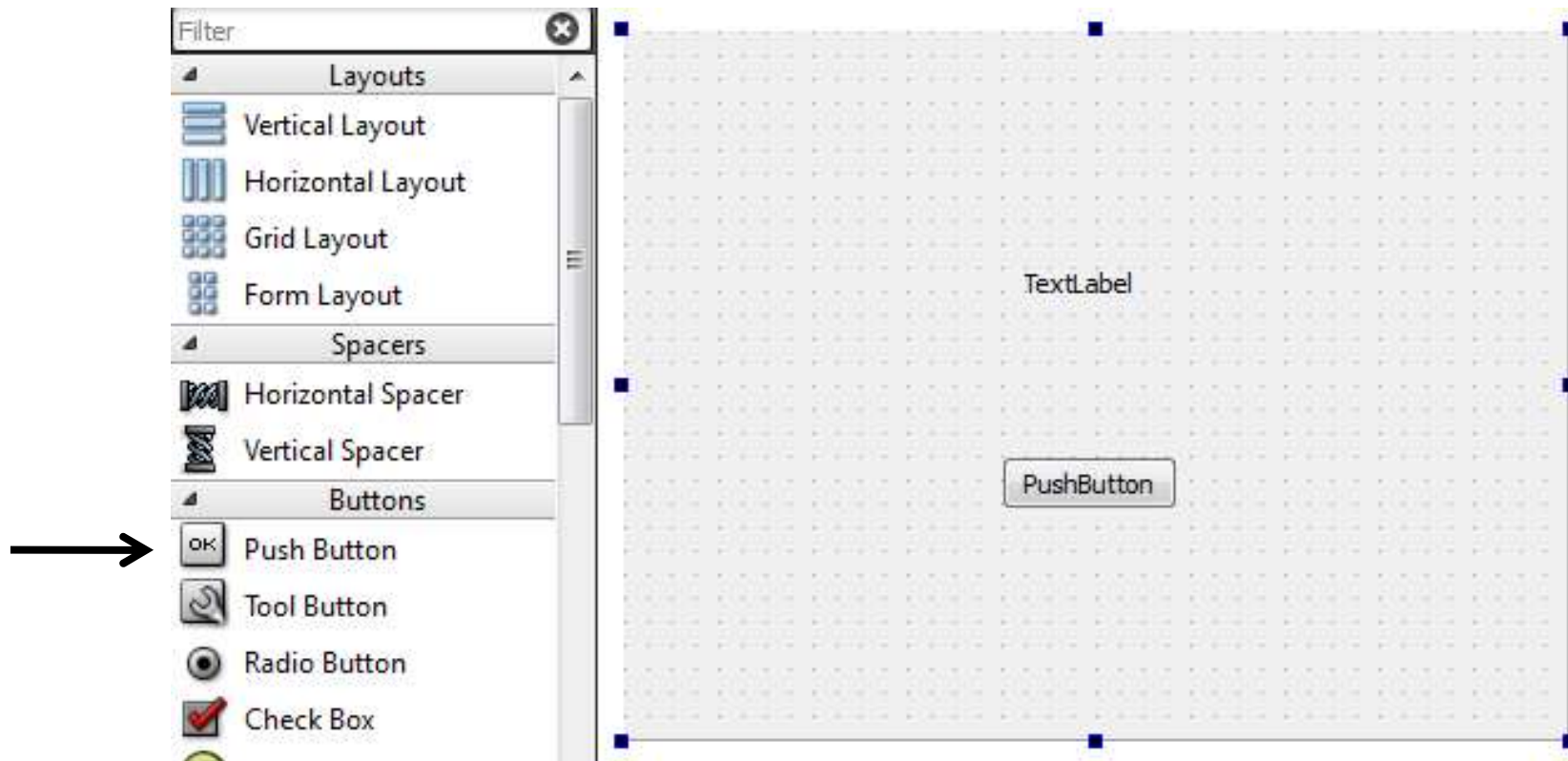
Design do Formulário

- Para entrar no modo Design de Formulário, dê dois cliques no arquivo meuform.ui ou clique na aba lateral “*Design*”.



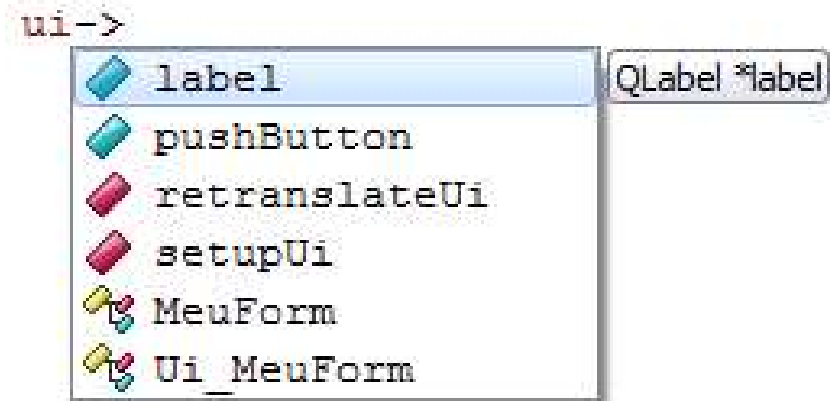
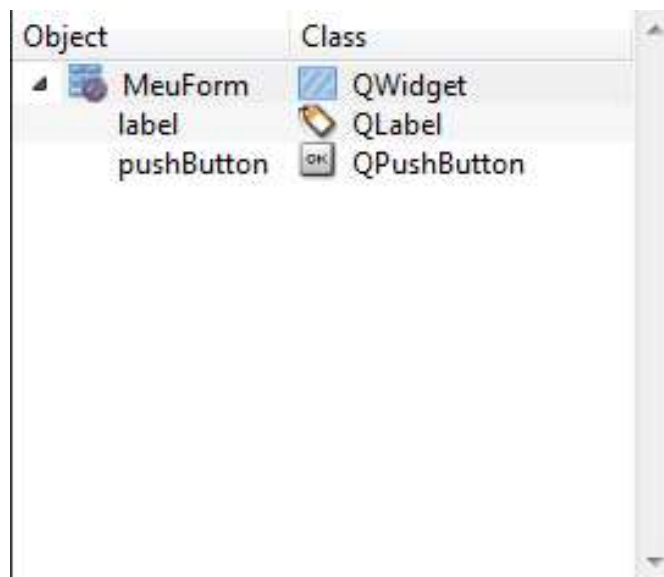
Design do Formulário

- Na esquerda, encontra-se a caixa de ferramentas com todos os componentes disponíveis para utilização no Formulário. Para adicionar um botão, clique em PushButton e arraste-o para o Formulário. Adicione também um objeto Label.



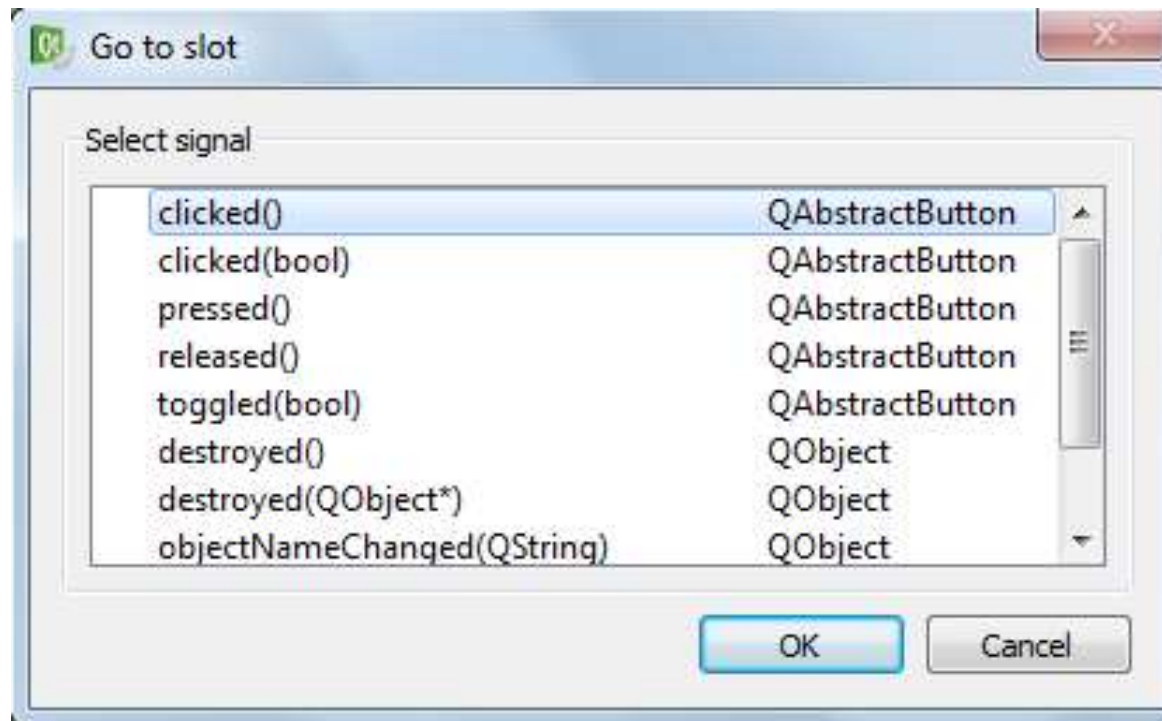
Design do Formulário

- Na parte superior direita, existe uma lista de todos os componentes presentes no formulário e o nome do respectivo objeto. Ela é útil pois o Qt Creator não mostra explicitamente os objetos gráficos como atributos dentro do código da classe MeuForm.
- Para acessar esses objetos via código e alterar alguma das suas propriedades, utiliza-se o ponteiro `Ui::MeuForm *ui` existente na classe `MeuForm`.



Eventos

- É uma ação externa ao programa, por exemplo: um clique de mouse ou uma tecla pressionada. Quando um evento ocorre, uma determinada função pode ser executada.
- Para adicionar um evento ao clique do QPushButton, clique com o botão esquerdo nele e escolha “Go to slot...” Selecione o evento clicked().



Eventos

- Automaticamente será criada uma função que será chamada quando o evento acontecer.
- O Qt guarda apontamento para todos os objetos presentes no formulário via o ponteiro `Ui::MeuForm *ui`. Através dele, por exemplo, é possível acessar o objeto `label` e mudar o seu texto.

```
void MeuForm::on_pushButton_clicked()
{
    string Text = "Clicou!";

    // Convertendo de string do C++ para QString (string do Qt)

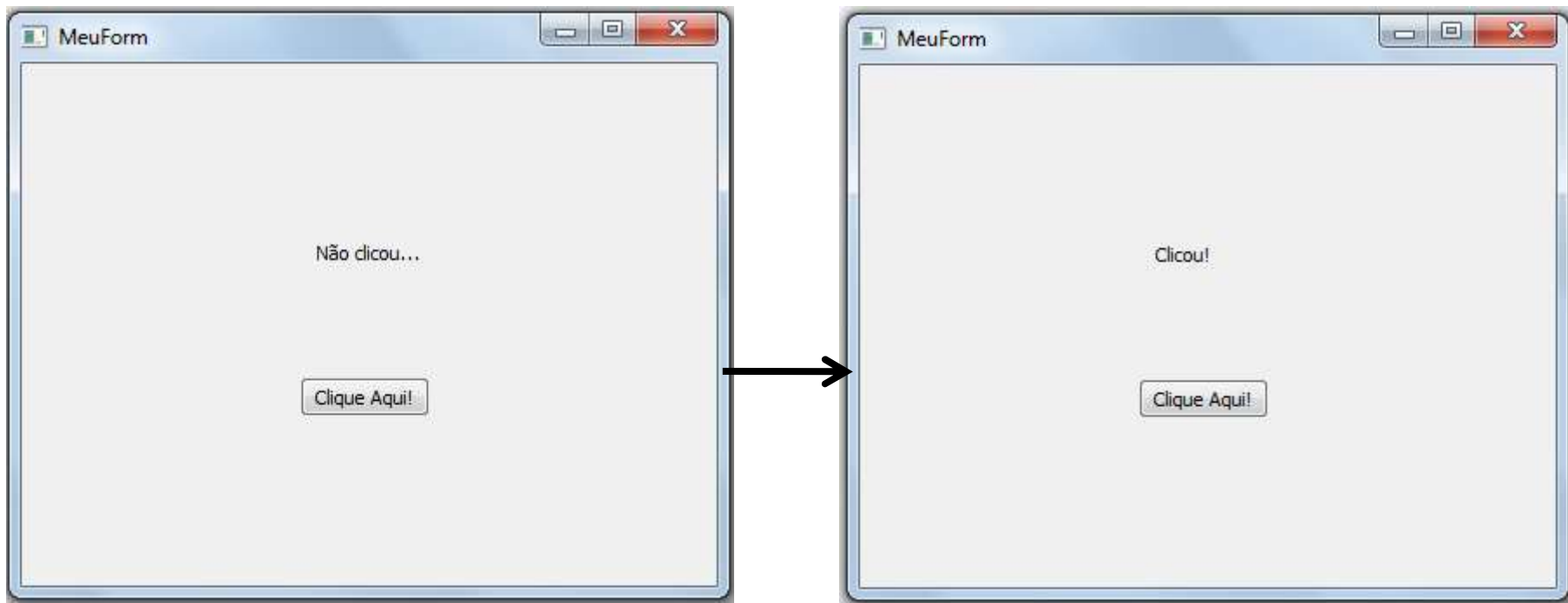
    QString aux = Text.data();

    ui->label->setText(aux);

}
```

Eventos

- Quando o usuário clicar no PushButton, o evento ocorrerá e o texto da Label é mudado automaticamente.



Exemplo 02: Calculadora

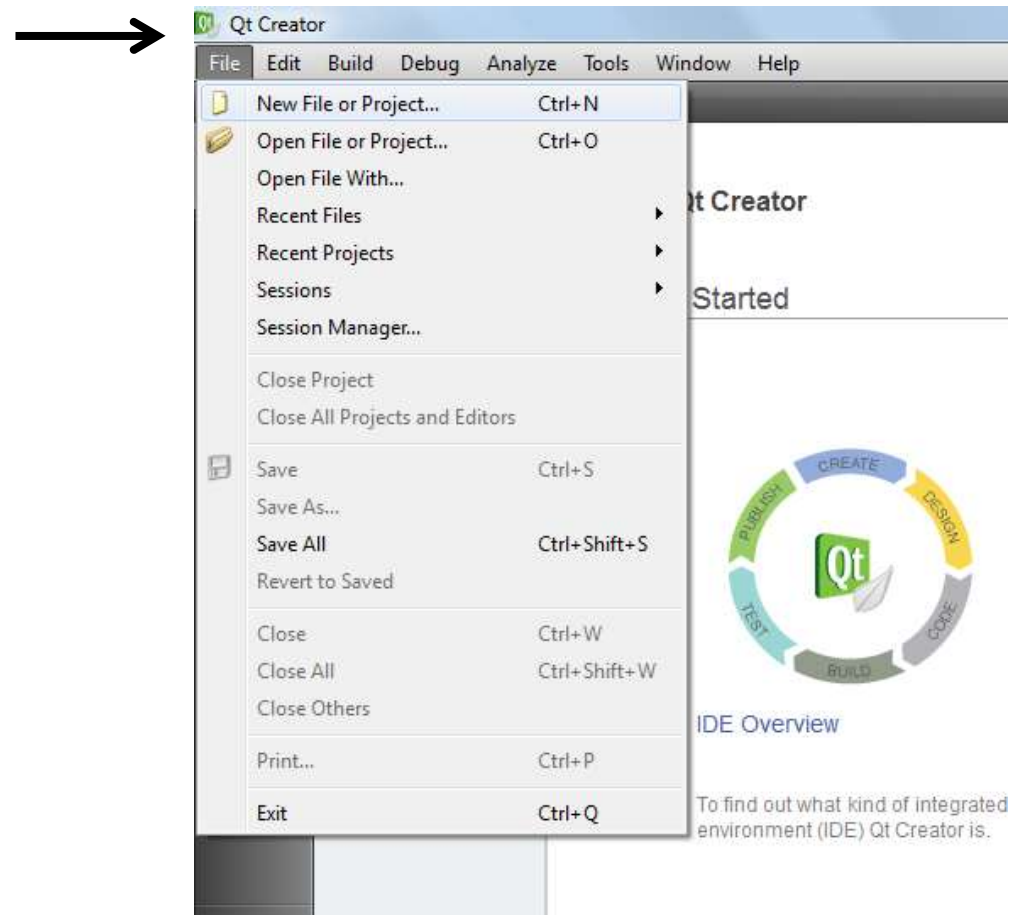
Exemplo 02

- O segundo exemplo busca demonstrar outros recursos do Qt, como o uso de mais de um formulário e outros componentes gráficos.
- Ele consiste em uma calculadora capaz de realizar as quatro operações básicas: adição, subtração, multiplicação e divisão.
- O formulário MeuFormPrincipal é criado da mesma maneira feita no Exemplo 1.
- Ele é composto por 2 LineEdit, 1 Push Button, 1 GroupBox e 4 RadioButton, todos estes encontrados na caixa de ferramentas de Componentes.



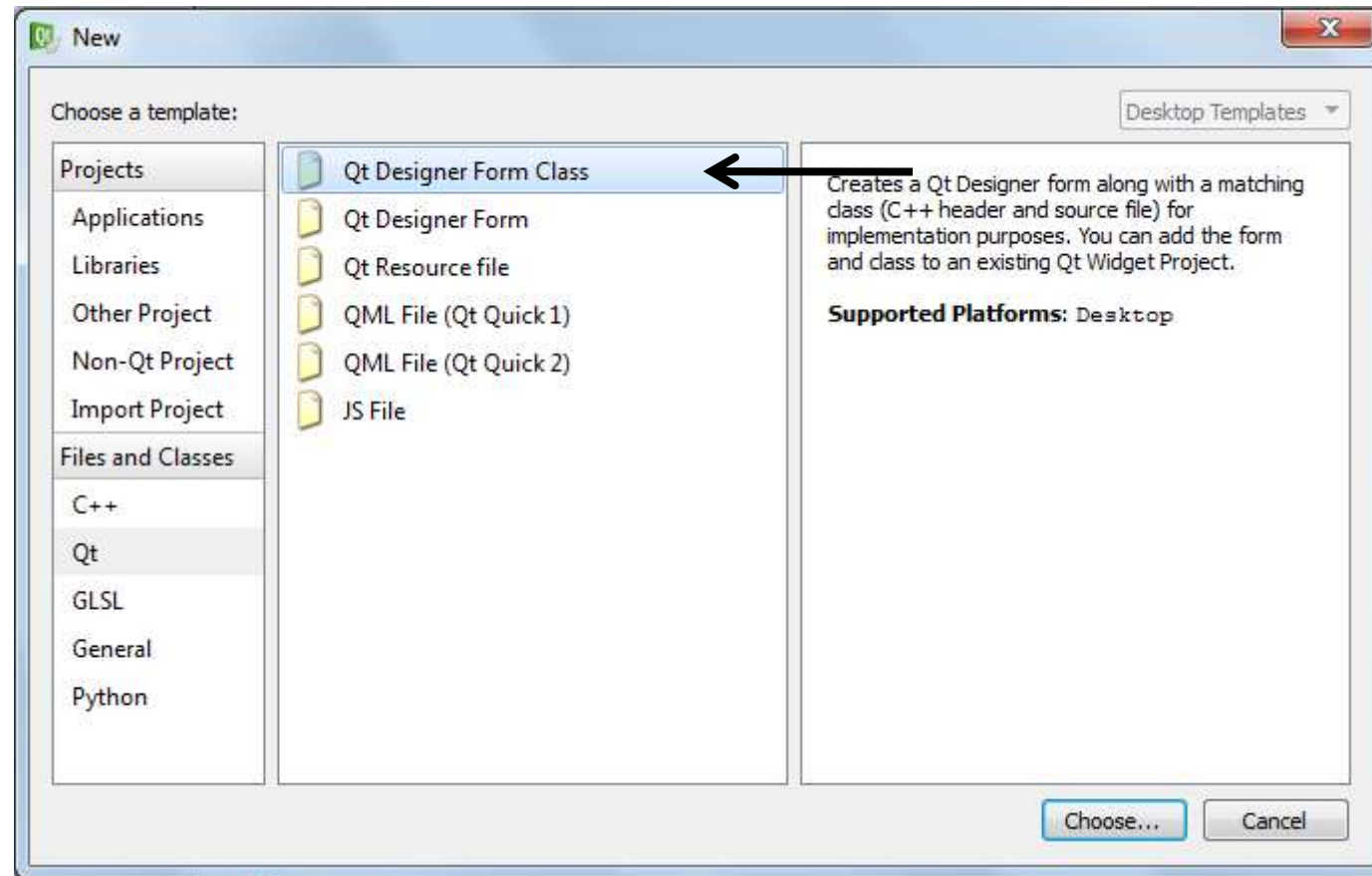
Exemplo 02

- Após ter criado o MeuFormPrincipal, crie o formulário MeuFormSecundário através do menu “*File*”.



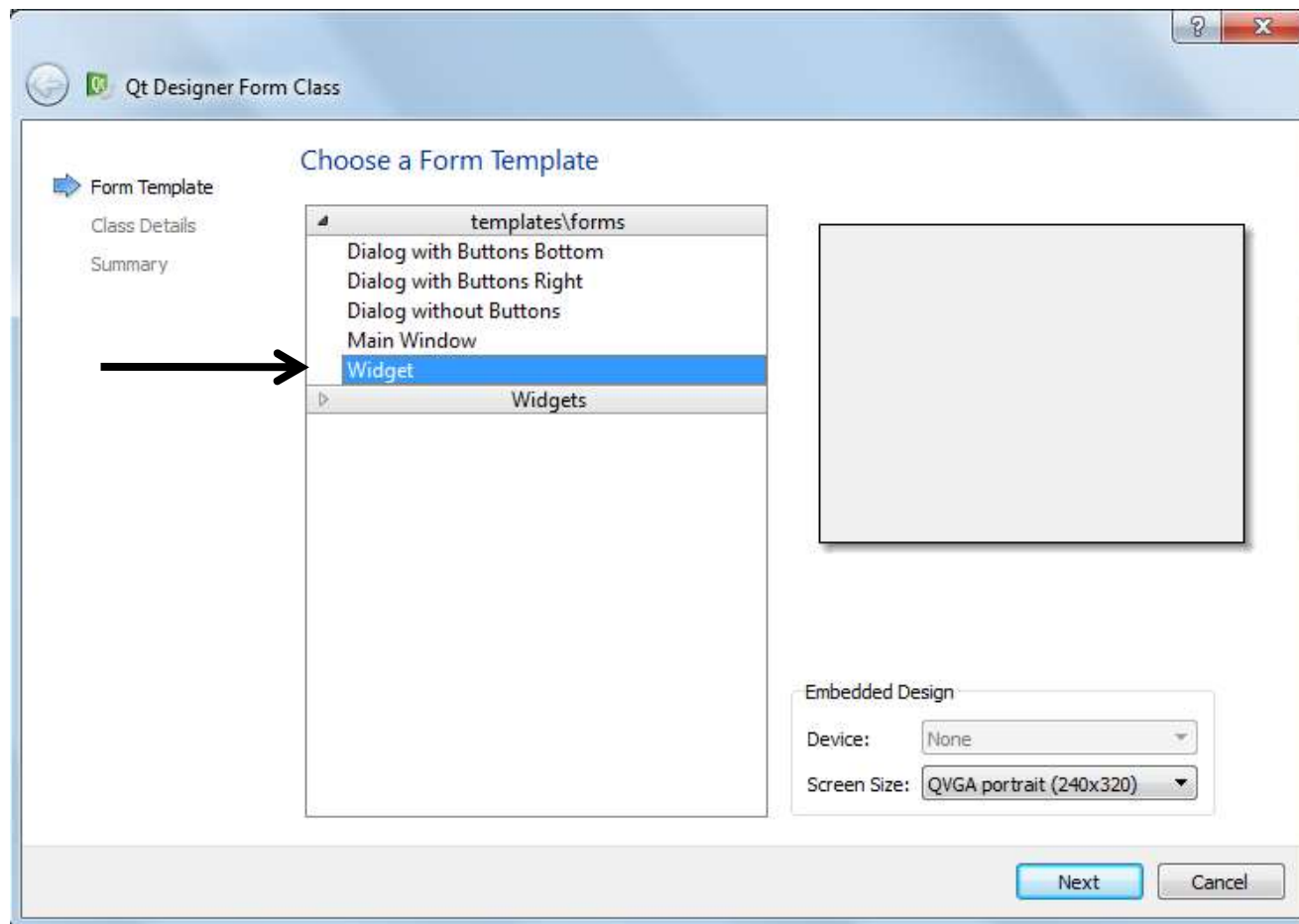
Exemplo 02

- Selecione o tipo de arquivo “Qt Designer Form Class”



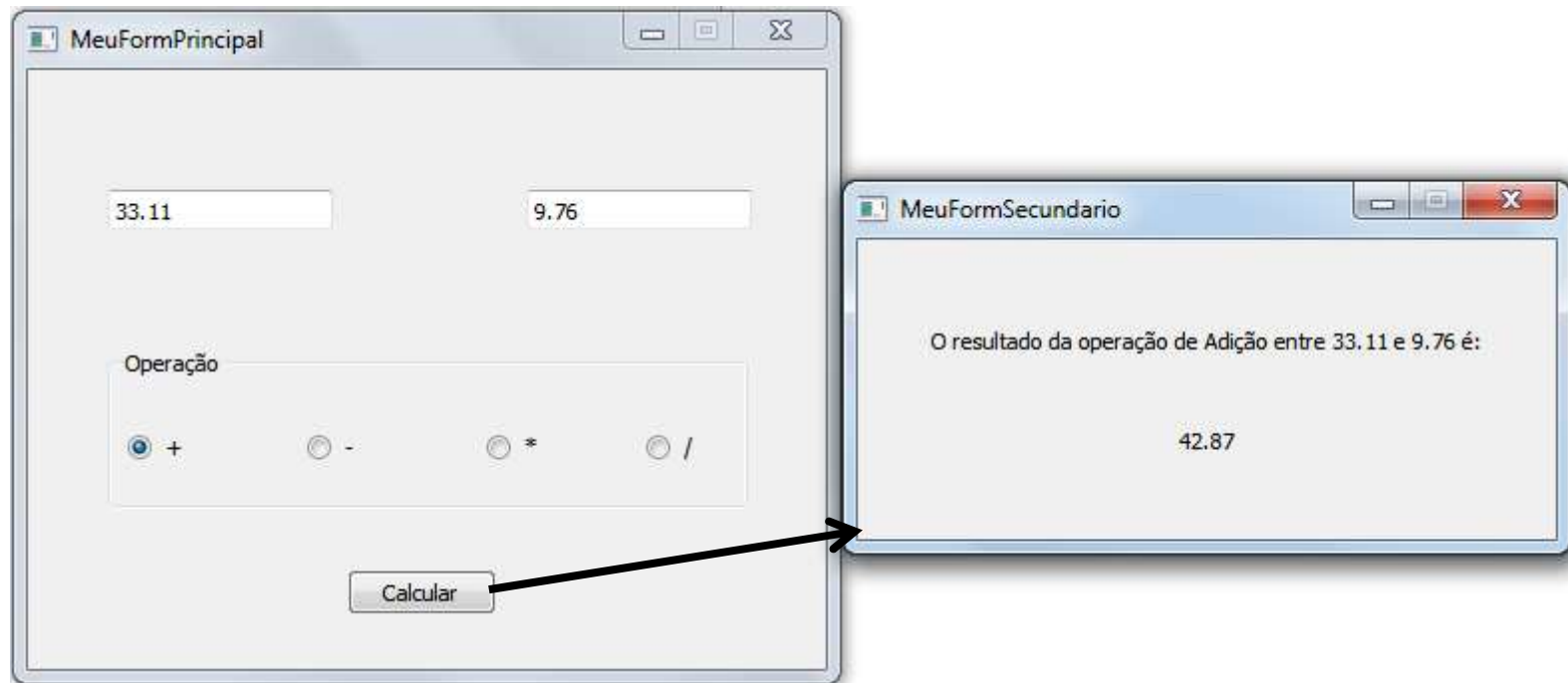
Exemplo 02

- Escolha o template “*Widget*” e nomeia a classe como MeuFormSecundario.



Exemplo 02

- O MeuFormSecundario é composto por 2 Labels. O usuário irá digitar os dois operandos, selecionar a operação e clicar em Calcular. Em seguida, o MeuFormSecundario será mostrado, contendo o resultado da operação.



```

// meuformsecundario.h
#ifndef MEUFORMSECUNDARIO_H
#define MEUFORMSECUNDARIO_H
#include <QWidget>
#include <string>

using namespace std;

namespace Ui
{ class MeuFormSecundario; }

class MeuFormSecundario : public QWidget
{ Q_OBJECT

public:
    explicit MeuFormSecundario(QWidget *parente = 0);
    ~MeuFormSecundario();

// Método set criado para alterar textos do Form
    void setLabelText ( float n1, float n2,
                       string operation, float result);
private:
    Ui::MeuFormSecundario *ui;
};

#endif // MEUFORMSECUNDARIO_H

```

```

// meuformsecundario.cpp
#include "meuformsecundario.h"
#include "ui_meuformsecundario.h"
#include <string>
#include <sstream>
MeuFormSecundario::
MeuFormSecundario (QWidget *parent) :
QWidget(parent), ui(new Ui::MeuFormSecundario) {
    ui->setupUi(this);
}
MeuFormSecundario::~MeuFormSecundario() {
delete ui;
}

void MeuFormSecundario::setLabelText(
float n1, float n2, string operation, float result) {
    // Concatenando textos
    stringstream aux; stringstream aux2;
    // Mudando o texto da label
    aux << "O resultado da operação de "
    << operation.data() << " entre "
    << n1 << " e " << n2 << " é:";
    ui->label->setText((aux.str()).data());
    //Mudando o texto da label_2
    aux2 << result;
    ui->label_2->setText ((aux2.str()).data());
}

```

```

// meuformprincipal.h
#ifndef MEUFORMPRINCIPAL_H
#define MEUFORMPRINCIPAL_H
#include <QWidget>
#include "meuformsecundario.h"
namespace Ui
{
    class MeuFormPrincipal;
}

class MeuFormPrincipal : public QWidget
{ Q_OBJECT
public:
    explicit MeuFormPrincipal(QWidget *parent = 0);
    ~MeuFormPrincipal();

private slots:
    void on_pushButton_clicked();

private:
    Ui::MeuFormPrincipal *ui;
    // MeuFormPrincipal agrega via alocação
    // dinâmica um objeto de MeuFormSecundario
    MeuFormSecundario * pForm;
};
#endif // MEUFORMPRINCIPAL_H

```

```

// meuformprincipal.cpp
#include "meuformsecundario.h"
#include "ui_meuformsecundario.h"
#include <string>
#include <sstream>

MeuFormPrincipal::
MeuFormPrincipal (QWidget *parent) :
    QWidget(parent),
    ui(new Ui::MeuFormPrincipal)
{
    ui->setupUi(this);

    // Adição é a operação default
    ui->additionRadioButton->setChecked(true);

    // Criando um novo objeto de FormSecundario
    pForm = new MeuFormSecundario();
}

MeuFormPrincipal::~~MeuFormPrincipal()
{
    // Fecha o FormSecundario e desaloca memória
    pForm->close();
    delete pForm;
    delete ui;
}

```



```

// Continuação meuformprincipal.cpp
void MeuFormPrincipal::on_pushButton_clicked()
{
//Evento de clicar no botão calcular
    pForm->clearFocus();
    //Variáveis float auxiliares
    float num1, num2 = 0.0;
    //Converter o texto digitado em cada lineEdit para float
    num1 = ui->lineEdit->text().toFloat();
    num2 = ui->lineEdit_2->text().toFloat();

    //Realiza soma
    if ( true == ui->additionRadioButton->isChecked ( ) )
    { pForm->setLabelText(num1, num2, "Adição", num1 + num2); }
    //Realiza subtração
    else if ( true == ui->subtractionRadioButton->isChecked ( ) )
    { pForm->setLabelText(num1, num2, "Subtração", num1 - num2); }
    //Realiza multiplicação
    else if ( true == ui->multiplicationRadioButton->isChecked ( ) )
    { pForm->setLabelText(num1, num2, "Multiplicação", num1 * num2); }
    //Realiza divisão
    else if ( true == ui->divisionRadioButton->isChecked ( ) )
    { pForm->setLabelText(num1, num2, "Divisão", num1 / num2); }

    //Mostra o FormSecundario
    pForm->show(); pForm->setFocus();
}

```

Após o usuário clicar no botão “Calcular”, os dados digitados em cada QLineEdit são convertidos em float.

A partir de qual RadioButton está selecionado, executa-se a operação desejada e os dados são enviados para o objeto de MeuFormSecundário, que é mostrado na tela.

Referências

- O Qt é uma biblioteca muito rica e possui vários recursos avançados. Os links a seguir são sugeridos como referência complementar e para o aprofundamento no *software* em questão:
 - <http://qt-project.org/doc/>
 - <http://antoniomenezes.net/down/minicursoQt4/oficinaQt4.pdf>
 - <http://www.hardware.com.br/comunidade/calculadora-tutorial/1302622/>
 - Acesso em 30 de março de 2014.