



**UNIVERSIDADE FEDERAL
DE OURO PRETO**

LISTA 5 I - DESENVOLVIMENTO DE GUIs COM O FRAMEWORK QT

ENYA LUÍSA GOMES DOS SANTOS

19.2.4201

Relatório apresentado por exigência da
disciplina BCC322 - ENGENHARIA DE
SOFTWARE I, da Universidade Federal de
Ouro Preto.

Professor: TIAGO GARCIA DE SENNA
CARNEIRO

OURO PRETO - MG

2021

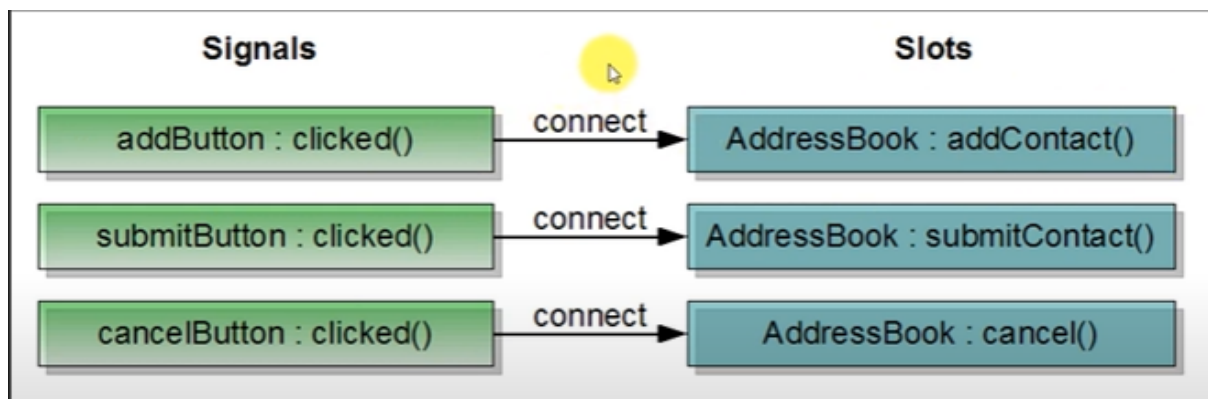
Questões teóricas

1.

- a. E uma chamada de retorno é uma função passada para outras funções como um argumento a ser chamado mais tarde na execução do programa. Em suma, uma função callback é um código passado como parâmetro para outro código. Assim, a camada de software de nível mais baixo pode invocar funções de nível mais alto.

```
float someFunc(int a, int b, float (*func)(int,int));
```

2.



Um *signal* é uma mensagem que está presente em uma classe como uma declaração de uma função membro void. Signals não são invocados, mas emitidos (via emit) por um objeto da classe.

signals: são métodos de classes que sinalizam a mudança do estado interno de um objeto.

slots: são métodos de classes que são disparados quando um *signal* é emitido.

O conceito é que widgets de interfaces gráficas podem enviar sinais contendo informações de eventos que podem ser recebidos por outras widgets / controles usando funções especiais conhecidas como slots. Isso é semelhante às callbacks em C/C++ (ponteiros de função), mas o sistema de sinais/slots garante que os argumentos da callback tenham os tipos corretos.

Signals e Slots implementam um mecanismo de comunicação entre objetos ou widgets.

3.

- a. Todas as classes que herdam de `QObject` ou uma de suas subclasses (por exemplo, `QWidget`) podem conter sinais e slots.
- b. Um *signal* pode estar associado a *n slots*. Um mesmo slot pode receber signals de objetos diferentes; e um mesmo *signal* pode enviar informações para vários slots,
- c. Nesse caso, o código após a `emit` palavra-chave continuará imediatamente e os slots serão executados mais tarde.

Se vários slots estiverem conectados a um sinal, os slots serão executados um após o outro, na ordem em que foram conectados, quando o sinal for emitido.

5.

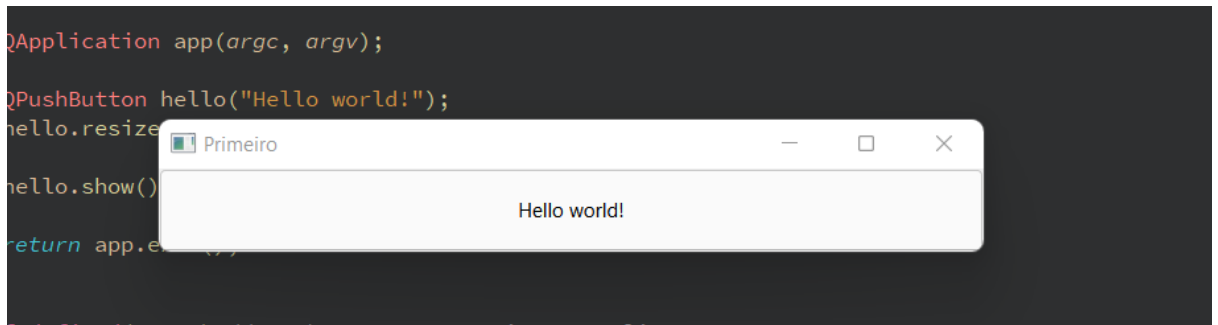
```
class MyClass : public QObject {
    Q_OBJECT
public:
    MyClass(QObject *parent = 0);
    ~MyClass();
signals:
    void mySignal();
public slots:
    void mySlot();
};
```

Programa principal:

```
MyClass x, y;
connect(&x, SIGNAL(mySignal()), &y, SLOT(mySlot()));
x.mySlot();
```

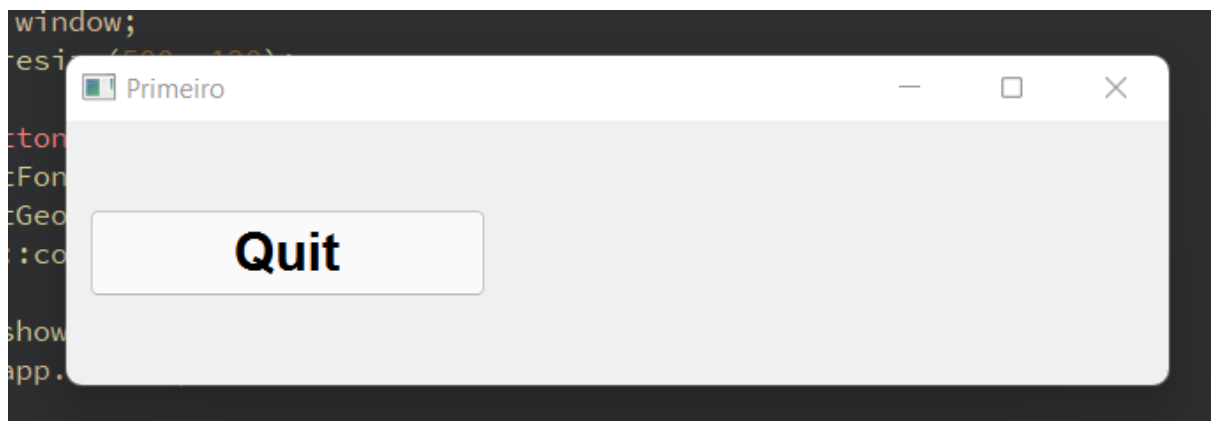
Questões práticas

1. O botão acompanhou o tamanho da janela.



2.

3. Nesse caso o botao nao seguiu o tamanho da tela, ele seguiu seu tamanho fixo.



4. Não foi possível alterar o tamanho da janela.

5. Duas janelas iguais foram abertas durante a execução .

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    MyWidget widget;
    MyWidget widget2;
    widget.show();
    widget2.show();
}
```

```

        return app.exec();
    }

```

6. A interface não funciona mais quando o número de dígitos do Visor LCD é menor que a quantidade de dígitos do número da faixa da Barra de Rolagem. Por exemplo:

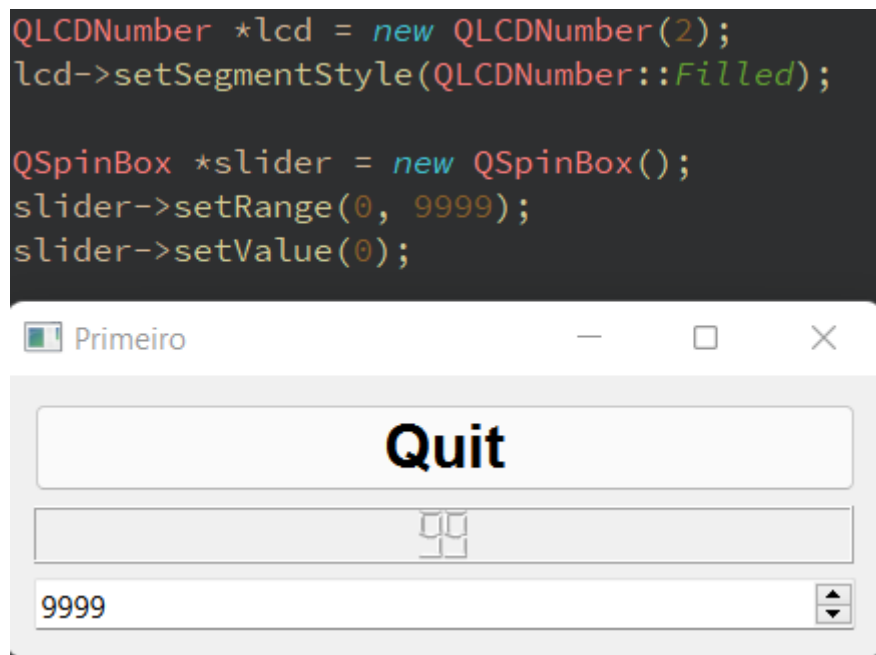
```

QLCDNumber *lcd = new QLCDNumber(3);
lcd->setSegmentStyle(QLCDNumber::Filled);

QSlider *slider = new QSlider(Qt::Horizontal);
slider->setRange(0, 9999);
slider->setValue(0);

```

7. Com o QSpinBox, por mais que o QLCDNumber tenha o valor de dígitos menor que a quantidade de dígito do QSpinBox, a interface ainda funciona, mesmo que o Visor LCD esteja limitado a mostrar o numero de digitos declarado.



9.

Linhas de código adicionadas:

```

srand(time(0));
QRandomGenerator valueRandom = QRandomGenerator::global()->bounded(0,99);

```

Linha de código alterada:

```

slider->setValue(valueRandom.generate());

```

