

# Programação Orientada a Objetos

## Aula 08 – QT Signals

Hugo Marcondes

Departamento Acadêmico de Eletrônica  
DAELN

[hugo.marcondes@ifsc.edu.br](mailto:hugo.marcondes@ifsc.edu.br)



### Signal e Slots

- Signals e Slots são mecanismos fundamentais da programação utilizando o QT
  - Permitem a interação de objetos de forma ALTAMENTE desacoplada
- Slots são similares a funções membro do C++, a diferença é que estas podem ser conectadas a um ou mais signal, sendo executada automaticamente toda vez que o signal é emitido.
  - Esta associação é realizada pelo método `connect()` da classe `QObject`

```
connect(sender, SIGNAL(signal), receiver, SLOT(slot));
```



Notes

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

## Signal e Slots

### ■ A conexão entre sinais e slots é muito versátil

#### ■ Um sinal pode ser conectado em diversos slots

```
connect(slider, SIGNAL(valueChanged(int)),
        spinBox, SLOT(setValue(int)));
connect(slider, SIGNAL(valueChanged(int)),
        this, SLOT(updateStatusBarIndicator(int)));
```

#### ■ Muitos sinais podem ser conectados a um mesmo slot

```
connect(lcd, SIGNAL(overflow()),
        this, SLOT(handleMathError()));
connect(calculator, SIGNAL(divisionByZero()),
        this, SLOT(handleMathError()));
```

#### ■ Um sinal pode ser conectado a outro sinal

```
connect(lineEdit, SIGNAL(textChanged(const QString &)),
        this, SIGNAL(updateRecord(const QString &)));
```



## Notes

---

---

---

---

---

---

## Signal e Slots

### ■ Para conectar sinais e slots é fundamental

- Ambos devem possuir a mesma assinatura de parâmetros (tipo e ordem)
- Excepcionalmente, se um sinal possuir um número maior de parâmetros, os parâmetros adicionais são ignorados.

```
connect(ftp, SIGNAL(rawCommandReply(int, const QString &)),
        this, SLOT(checkErrorCode(int)));
```



## Notes

---

---

---

---

---

---

## Signal e Slots

- Qualquer classe pode implementar sinais e slots para uso pelo QT
  - A classe deve herdar a classe QObject
  - Declarar a macro Q\_OBJECT
  - Utilizar o escopo `public slot:` para declarar os slots
    - São implementados como métodos comuns, podem inclusive ser chamados diretamente
  - Utilizar o escopo `signals:` para declarar os sinais
    - São declarados com uma assinatura na classe, mas não devem ser implementados
    - A implementação dos sinais são lidadas diretamente pela biblioteca QT
  - O objeto que emite o sinal deve utilizar a macro `emit` fornecida pelo QT



## Notes

---

---

---

---

---

---

---

## Exemplo

```
1 class Employee: public QObject {
2     Q_OBJECT
3
4     public:
5         Employee() { mySalary = 0; }
6         int salary() const { return mySalary; }
7     public slots:
8         void setSalary(int newSalary);
9     signals:
10        void salaryChanged(int newSalary);
11
12    private:
13        int mySalary;
14 };
15
16 void Employee::setSalary(int newSalary) {
17     if (newSalary != mySalary) {
18         mySalary = newSalary;
19         emit salaryChanged(mySalary);
20     }
21 }
```



## Notes

---

---

---

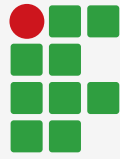
---

---

---

---

That's all folks!



**INSTITUTO  
FEDERAL**  
Santa Catarina  

---

Câmpus  
Florianópolis



Notes

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---