

Unidade 2 – Graphical User Interfaces (GUI's)

## Exemplo 5: Lista

Vamos criar um aplicativo simples que permite fazer operações básicas com uma Lista.

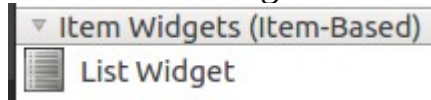
1. Criar Criar novo projeto (idêntico aos exercícios anteriores):

**File > New File or Project > Application > Qt Widgets Application > Choose**

Nome do projeto: **MyList**

Nome da Classe: **MainWindow** (Base class: **QMainWindow**)

2. Adicionar widget **List Widget** na forma principal



3. Primeiro vamos adicionar itens na lista via código. Para isso temos que adicionar:

### mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

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

    ui->listWidget->addItem("AAA");
    ui->listWidget->addItem("BBB");
    ui->listWidget->addItem("CCC");

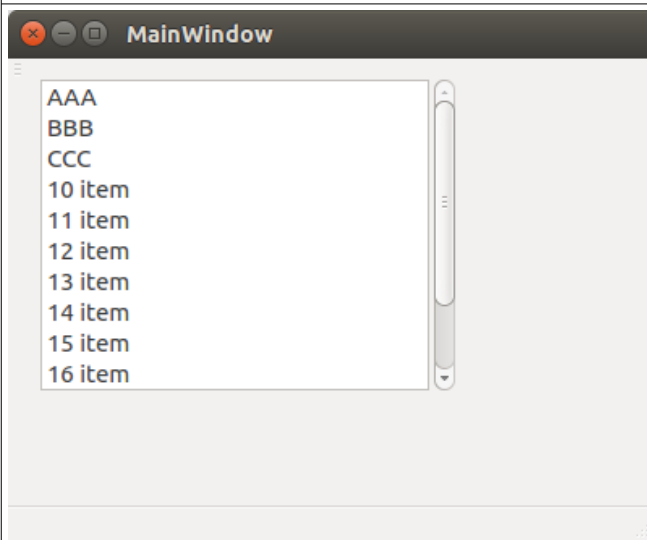
    for(int i = 10; i < 20; i++)
    {
        ui->listWidget->addItem(QString::number(i) + " item" );
    }
}

MainWindow::~MainWindow()
{
    delete ui;
```

```
}
```

Depois de compilar e executar o programa temos algo assim:

### Aplicativo



4. Vamos criar um botão para seleccionar um elemento da lista. Para isso vamos adicionar um **widget** do tipo **push Button** e modificar as características dele da seguinte forma:

tipo	push Button
nome	selectButton
texto	Select

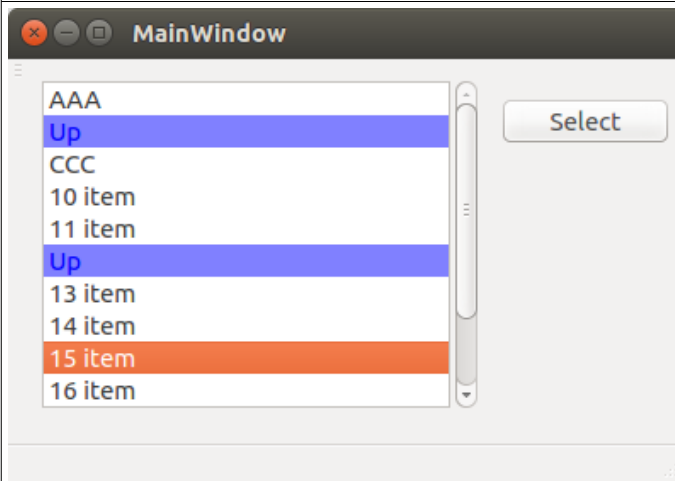
Adicionar um **slot** → **clicked()** com o seguinte código:

### mainwindow.cpp

```
void MainWindow::on_selectButton_clicked()
{
    // ui->listWidget->currentItem()->setText("Up");
    QListWidgetItem *itm = ui->listWidget->currentItem();
    itm->setText("Up");
    itm->setTextColor(Qt::blue);
    itm->setBackground(QColor(0, 0, 255, 127));
}
```

Esse código vai fazer com que o item selecionado seja substituído pelo texto “Up” e vai alterar a cor da fonte e do fundo.

## Aplicativo



5. Vamos criar um botão para remover um elemento da lista. Para isso vamos adicionar um **widget** do tipo **push Button** e modificar as características dele da seguinte forma:

tipo	push Button
nome	removeButton
texto	Remove

Adicionar um **slot** → **clicked()** com o seguinte código:

### mainwindow.cpp

```
void MainWindow::on_removeButton_clicked()
{
    delete ui->listWidget->currentItem();
}
```

6. Vamos criar um botão para adicionar um elemento. Para isso vamos adicionar um **widget** do tipo **line Edit** e um **widget** do tipo **push Button** e modificar as características dele da seguinte forma:

tipo	push Button
nome	addButton
texto	Add

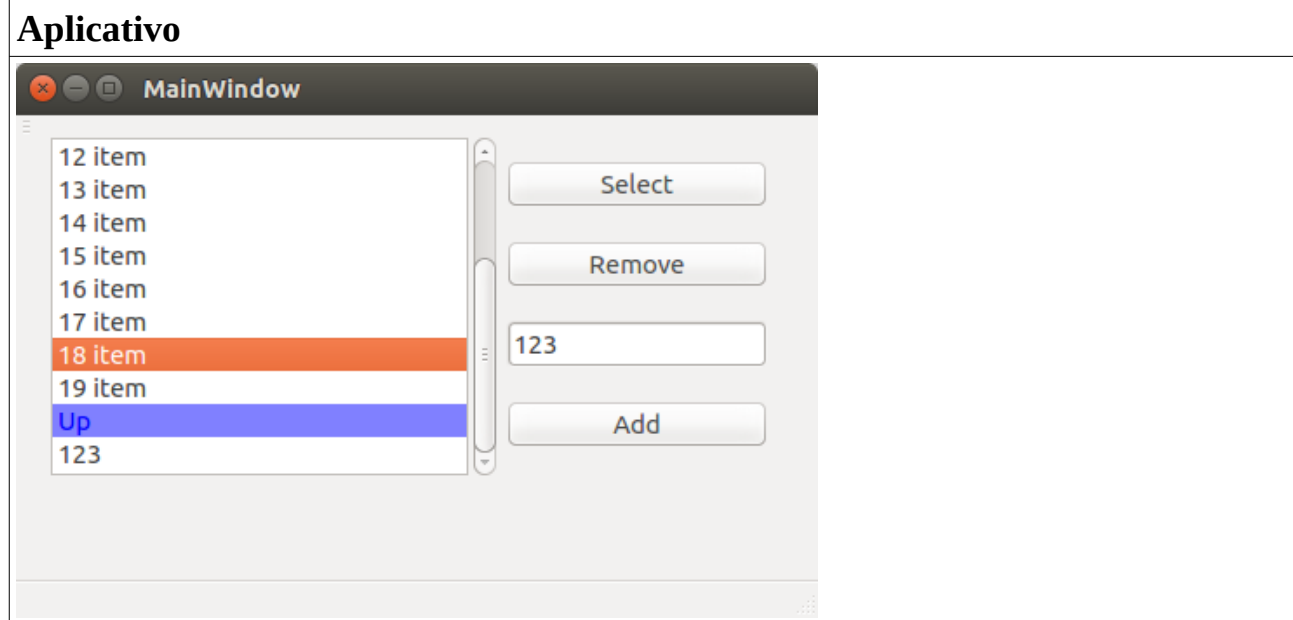
Adicionar um **slot** → **clicked()** com o seguinte código:

### mainwindow.cpp

```
void MainWindow::on_addButton_clicked()
{
    ui->listWidget->addItem(ui->lineEdit->text());
}
```

Os itens serão adicionados no final da lista.

Aplicativo agora deve ser parecido com:

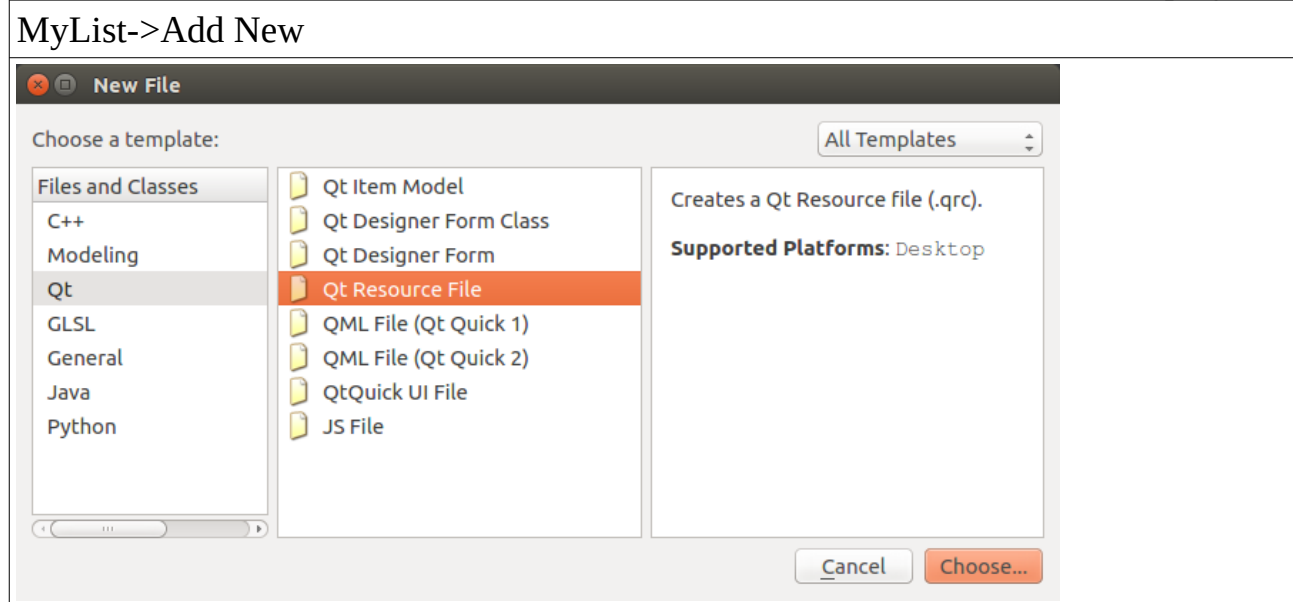


7. Finalmente vamos adicionar os ícones para alguns itens da lista.

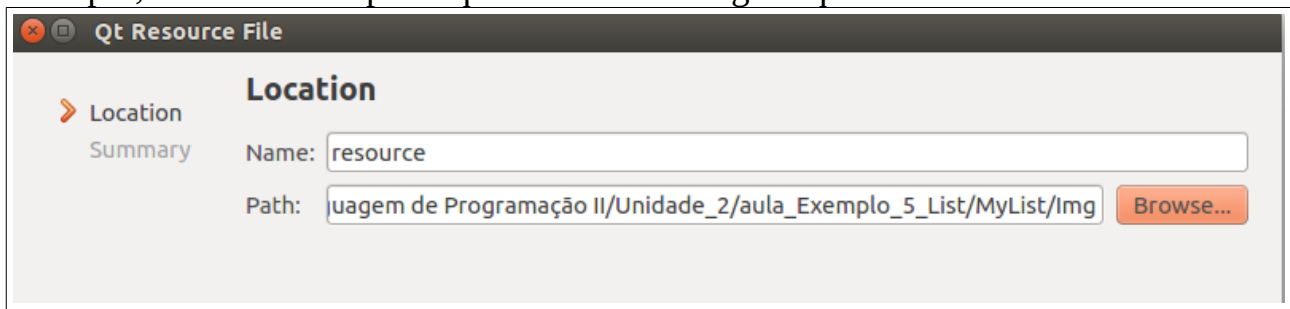
Para isso precisamos fazer mais alguns passos:

7.1. Adicionar a pasta com imagens para ícones na pasta do Projeto.

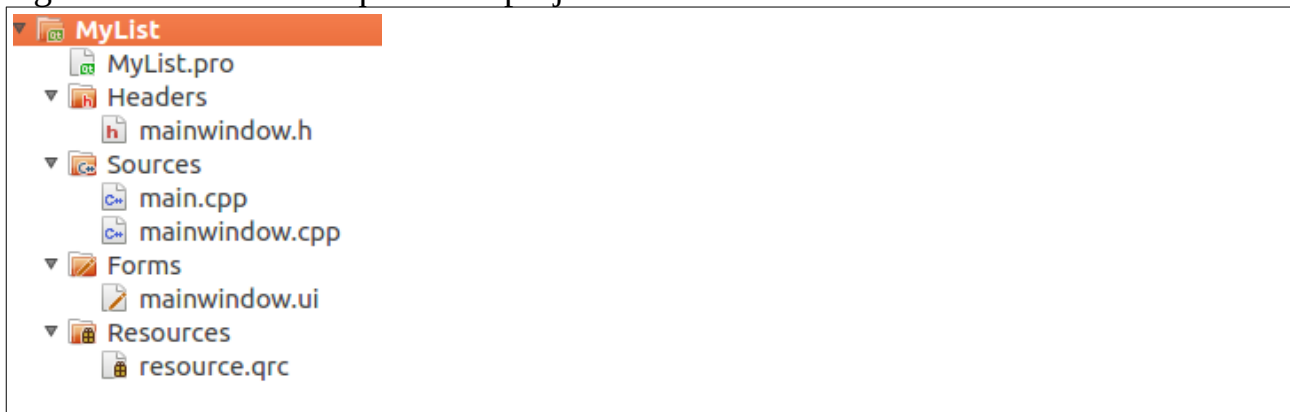
7.2. Adicionar **Qt Resource File** selecionando item Add New... em cima do projeto:



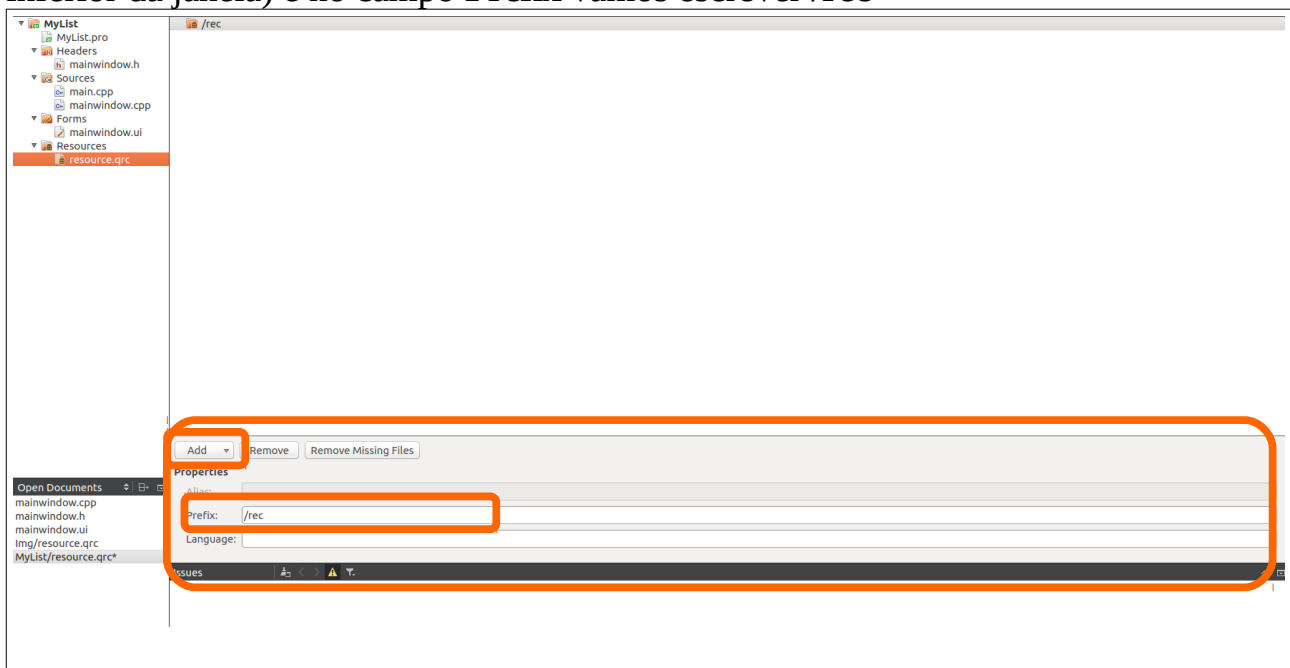
Na próxima janela escolher um nome para esse tipo de arquivo, **resource** por exemplo, e selecionar a pasta que contém as imagens que serão usados como ícones.



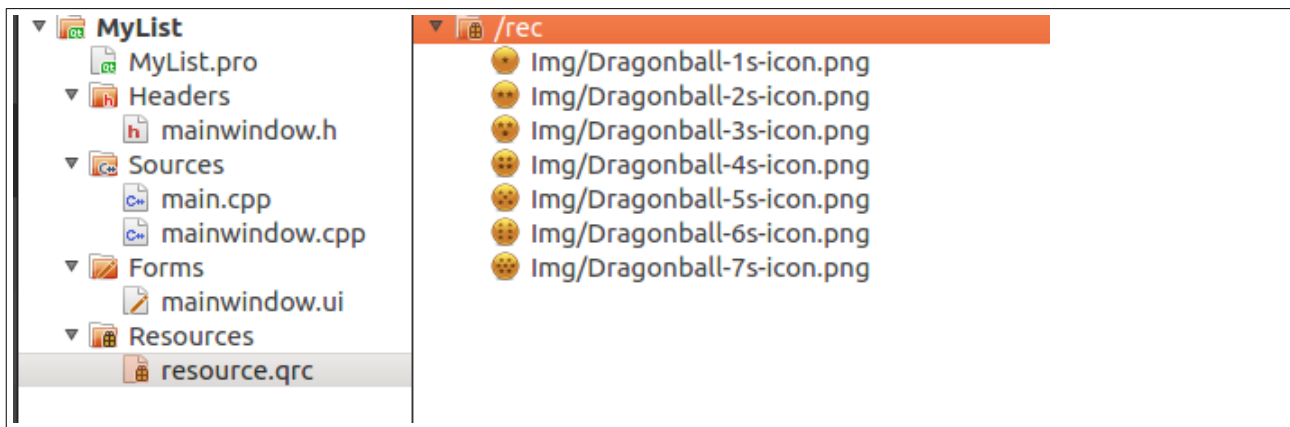
Agora na árvore dos arquivos do projeto vamos ter:



**7.3** Agora vamos selecionar esse arquivo e a opção **Add** → **Add Prefix** (na parte inferior da janela) e no campo **Prefix** vamos escrever **/rec**



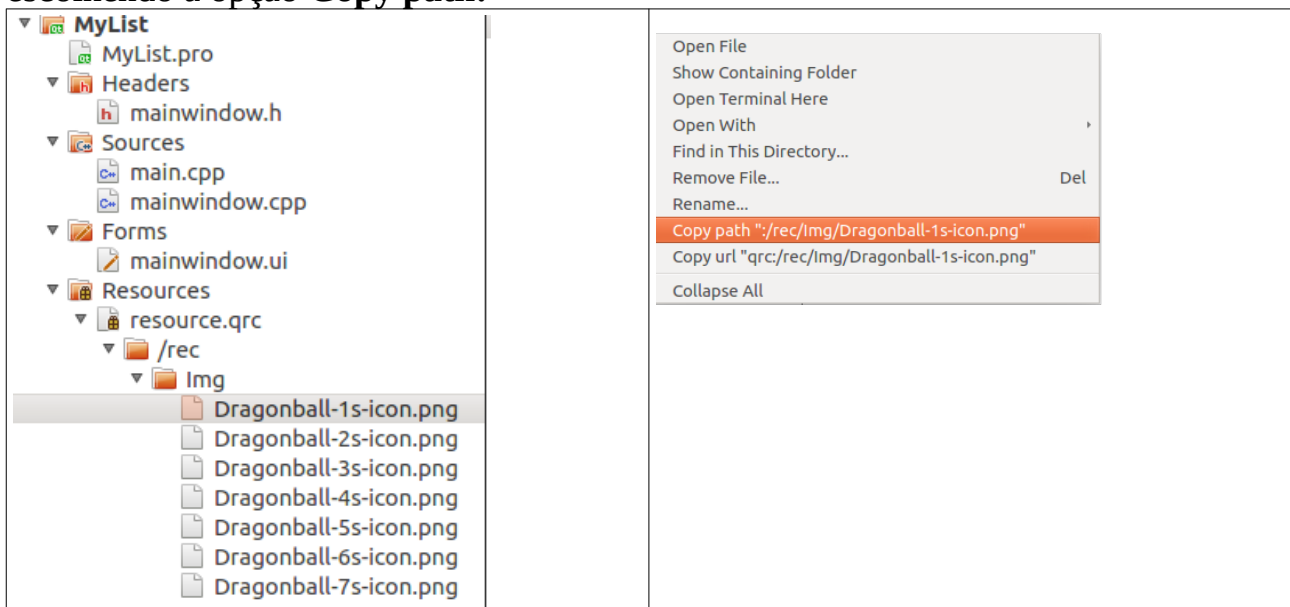
Depois no mesmo menu vamos escolher **Add** → **Add Files** e vamos selecionar os arquivos com as imagens. Depois de adicionar todos arquivos eles devem aparecer quando o arquivo **resource.qrc** é escolhido:



Podem colocar outros ícones de referência.

## 7.4 Inserção dos arquivos das imagens

**Observação:** para pegar a localização das imagens para os ícones podemos escrever o caminhão manualmente ou então podemos chamar menu em cima do **resource.qrc** e escolher opção **Open With -> Resource Editor** agora vamos ver a lista dos nossos arquivos com imagens e podemos copiar o caminho chamando menu em cima deles e escolhendo a opção **Copy path:**



Agora podemos fazer as seguintes modificações no código:

### mainwindow.cpp

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

    QListWidgetItem *item1 = new QListWidgetItem (QIcon(":/rec/Img/Dragonball-1s-icon.png"), " item 1");
```

```

    ui->listWidget->addItem(item1);

    QListWidgetItem *item2 = new QListWidgetItem (QIcon(":/rec/Img/Dragonball-
2s-icon.png"), " item 2");
    ui->listWidget->addItem(item2);

    QListWidgetItem *item3 = new QListWidgetItem (QIcon(":/rec/Img/Dragonball-
3s-icon.png"), " item 3");
    ui->listWidget->addItem(item3);

    for(int i = 10; i < 20; i++)
    {
        ui->listWidget->addItem(QString::number(i) + " item" );
    }
}

```

Depois de compilar e executar o programa vamos ter algo assim:

