

### Exemplo 3: TextFinder

(<http://doc.qt.io/qtcreator/creator-writing-program.html>)

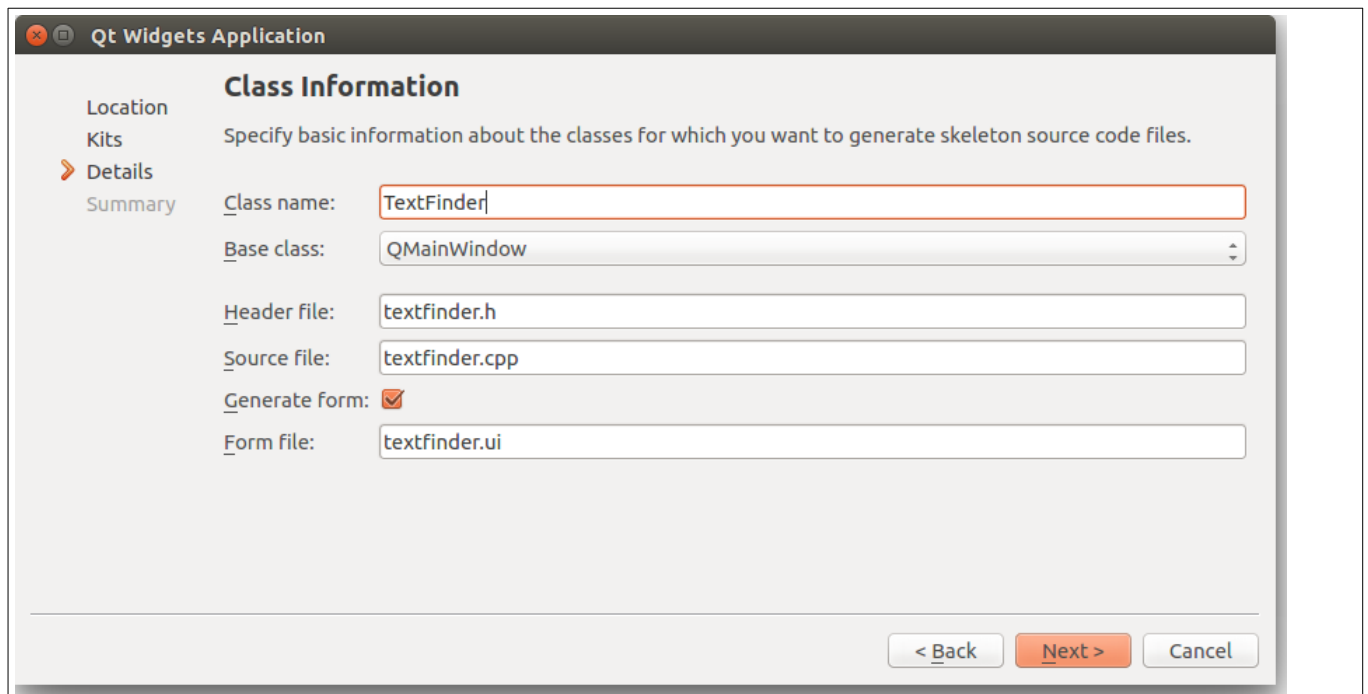
Vamos criar um aplicativo simples que realiza a busca por palavras-chave no arquivo texto que é automaticamente carregado (consultado) no início da execução do programa.

Criar novo projeto (idêntico aos exercícios anteriores):

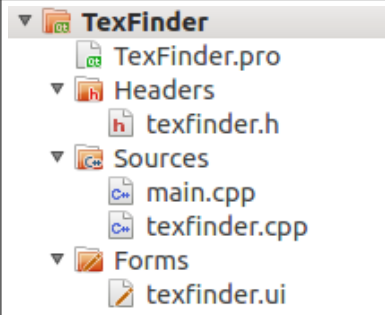
**File > New File or Project > Application > Qt Widgets Application > Choose**

Nome do projeto: **TextFinder**

Nome da Classe: **TextFinder**







O projeto **TextFinder** vai conter os arquivos:

<ul style="list-style-type: none"> <li>• textfinder.h</li> <li>• textfinder.cpp</li> <li>• main.cpp</li> <li>• textfinder.ui</li> <li>• textfinder.pro</li> </ul>	
---	--

### Interface gráfica

Em modo de **QT Designer** vamos adicionar os seguintes componentes na nossa janela principal:

**1)** Em **textfinder.ui** acrescentar os seguintes elementos:

Elemento	Quantidade	Imagem
<b>Label (<a href="#">QLabel</a>)</b>	<b>1</b>	 Label
<b>Line Edit (<a href="#">QLineEdit</a>)</b>	<b>1</b>	 Line Edit
<b>Push Button (<a href="#">QPushButton</a>)</b>	<b>1</b>	 Push Button
<b>Text Edit (<a href="#">QTextEdit</a>)</b>	<b>1</b>	 Text Edit

**2)** Aplicar o **Layout Horizontal** nos elementos:

- **Label ([QLabel](#))**
- **Line Edit ([QLineEdit](#))**
- **Push Button ([QPushButton](#))**

**3)** Selecionar todos os elementos e aplicar o **Layout Vertical**.

**4)** Trocar o texto do **Label** para “**Keyword:**”

**5)** Trocar o texto do **Push Button** para “**Find**”

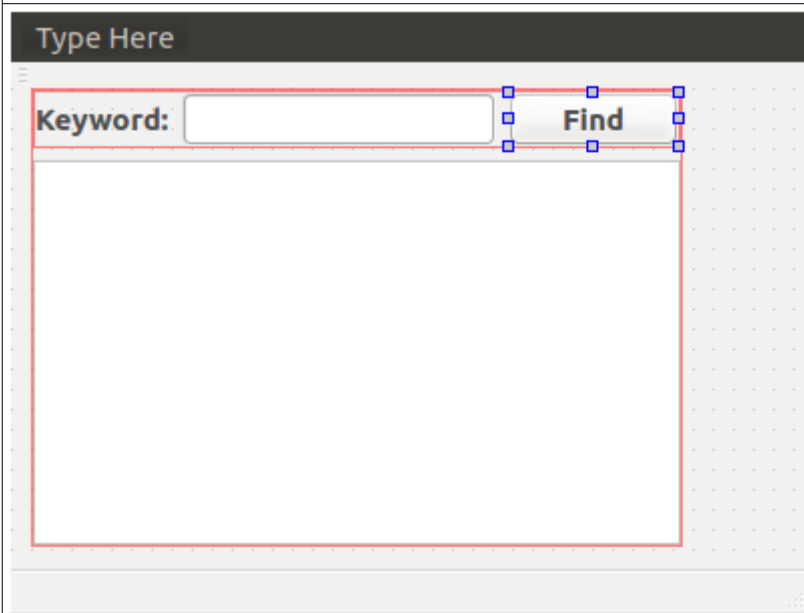
**6)** Trocar o nome do objeto de **Push Button** para “**findButton**” na janela **Properties**

**textfinder.ui**

Object	Class
▼ TextFinder	QMainWindow
▼ centralWidget	QWidget
▼ verticalLayout	QVBoxLayout
▼ horiz...ayout	QHBoxLayout
findButton	QPushButton
label	QLabel
lineEdit	QLineEdit
textEdit	QTextEdit
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

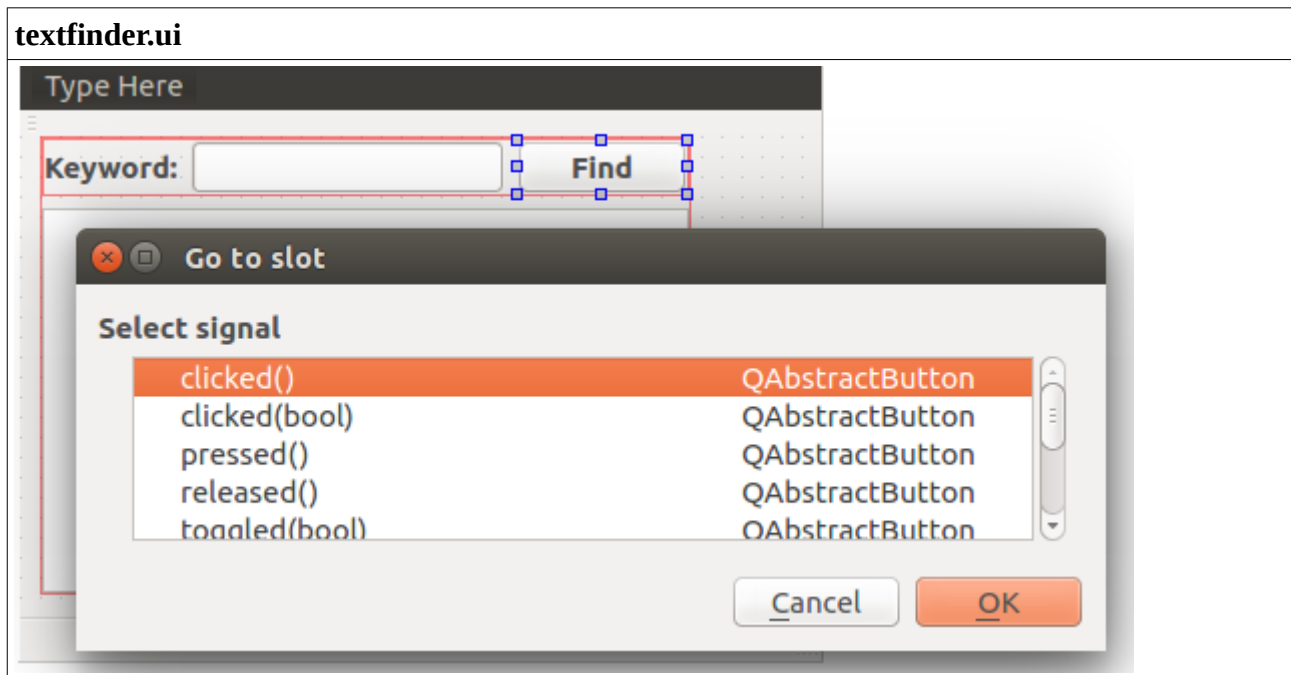
A aparência da janela principal será parecida com isso:

#### textfinder.ui



## Funcionalidade

7) Em **mainwindow.ui** chamar o menu em cima do boton **findButton** e escolher **Go to Slot** → **clicked()** → **ok**



Qt Designer vai automaticamente adicionar o seguinte **private slot** no arquivo **header (.h)**:

```
textfinder.h
private slots:
    void on_findButton_clicked();
```

A função correspondente será adicionada no arquivo **source (.cpp)**:

```
textfinder.cpp
void TextFinder::on_findButton_clicked()
{
}
}
```

8) Vamos criar uma função **private** chamada **loadTextFile()** para carregar o conteúdo do arquivo com dados de entrada para componente [QTextEdit](#).

Para isso vamos adicionar o seguinte código em arquivo **.h**:

```
textfinder.h
#ifndef TEXTFINDER_H
#define TEXTFINDER_H

#include <QMainWindow>

namespace Ui {
class TextFinder;
```

```

}

class TextFinder : public QMainWindow
{
    Q_OBJECT

public:
    explicit TextFinder(QWidget *parent = 0);
    ~TextFinder();

private slots:
    void on_findButton_clicked();

private:
    Ui::TextFinder *ui;
    void loadTextFile();
};

#endif // TEXTFINDER_H

```

9) Em arquivo **.cpp** vamos adicionar a implementação da função **loadTextFile()** e o código que realizará a busca de texto quando o botão **findButton** é acionado.

A chamada da função **loadTextFile()** será adicionada no construtor.

#### textfinder.cpp

```

#include "textfinder.h"
#include "ui_textfinder.h"

#include <QFile>
#include <QTextStream>

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

TextFinder::~~TextFinder()
{
    delete ui;
}

void TextFinder::on_findButton_clicked()
{
    QString searchString = ui->lineEdit->text();
    ui->textEdit->find(searchString, QTextDocument::FindWholeWords);
}

void TextFinder::loadTextFile()
{
    QFile inputFile(":/input.txt");
    inputFile.open(QIODevice::ReadOnly);

```

```

QTextStream in(&inputFile);
QString line = in.readAll();
inputFile.close();

ui->textEdit->setPlainText(line);
QTextCursor cursor = ui->textEdit->textCursor();
cursor.movePosition(QTextCursor::Start, QTextCursor::MoveAnchor, 1);
}

```

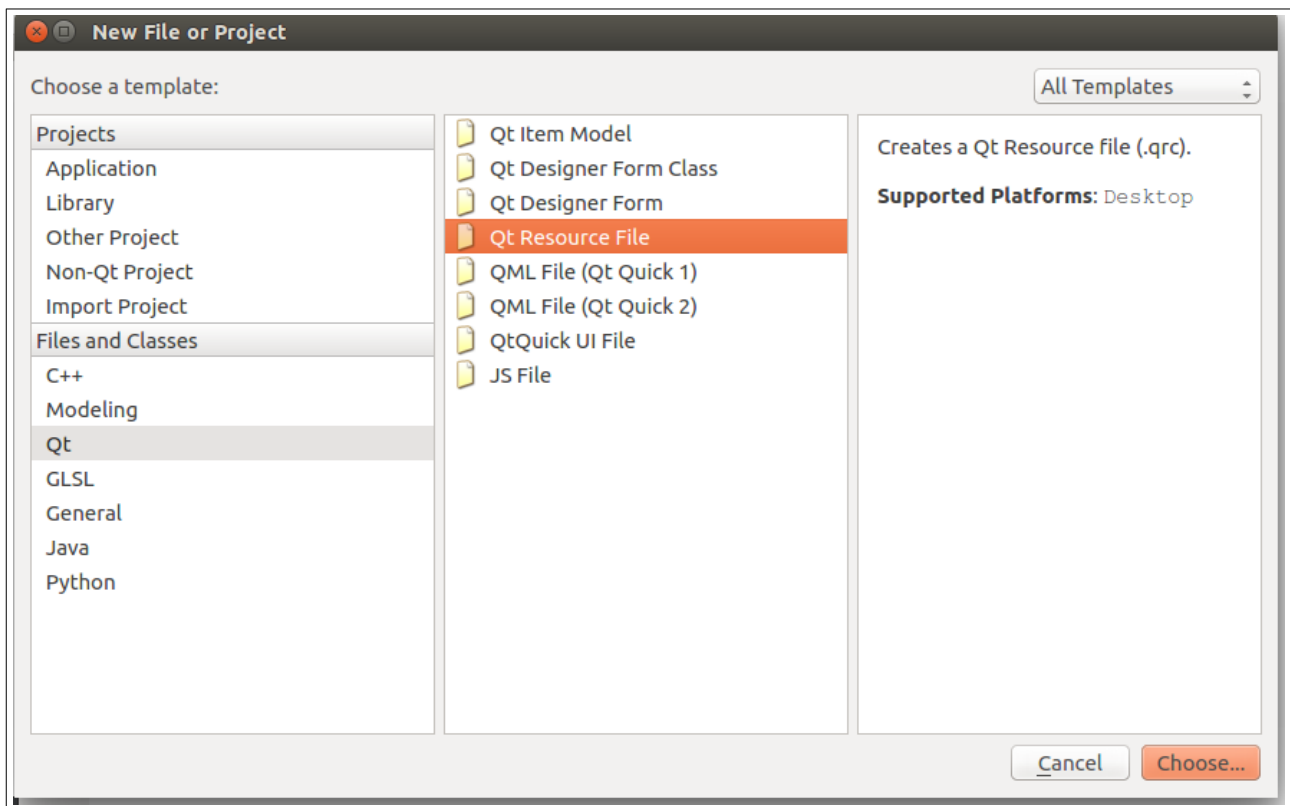
### Criação de um arquivo para consulta

Nesse exemplo vamos criar um arquivo do tipo **resource file (.qrc)** que será responsável pela leitura de arquivo com dados de entrada **input.txt**

O arquivo **input.txt** é um simples arquivo de texto que pode ser criado pelo qualquer editor de texto. Ele pode conter qualquer tipo de texto e deve ser armazenado na mesma pasta do projeto **TextFinder**.

**10) (A)** Para adicionar o arquivo do tipo **resource file** temos que escolher:

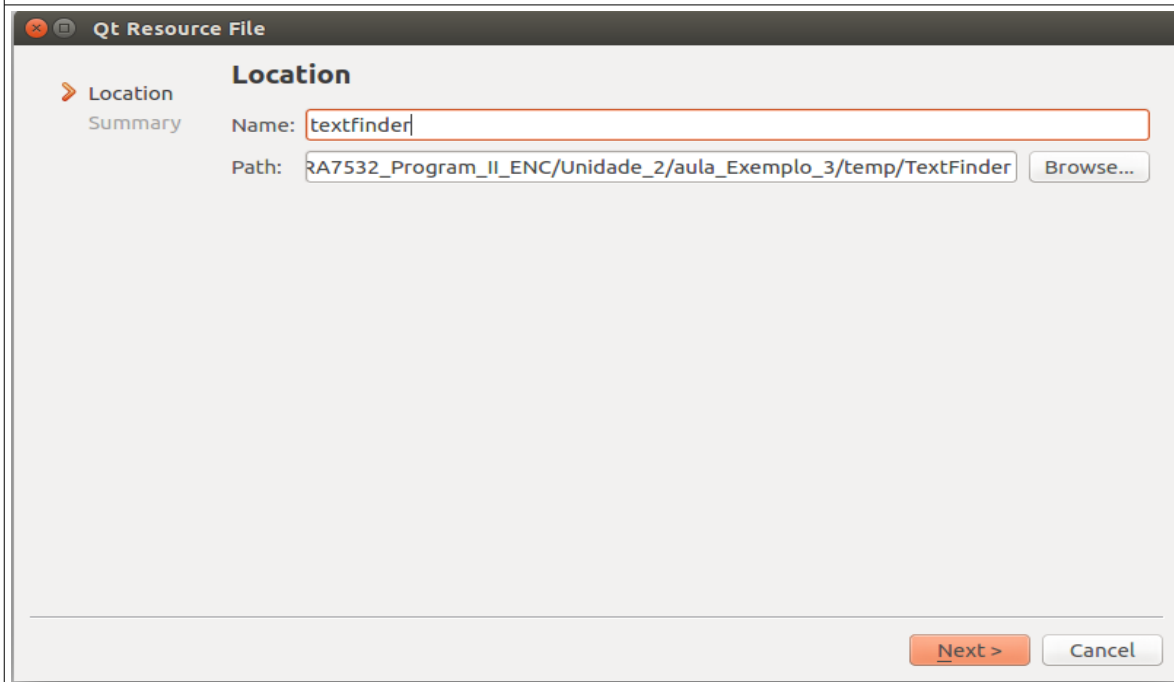
**File > New File or Project > Qt > Qt Resource File > Choose**



**10) (B)** Na próxima janela que vai abrir em seguida **Location** temos que colocar:

Campo	Valor
Name:	textfinder
Path:	caminho para pasta do projeto

### Location



The image shows the 'Qt Resource File' dialog box with the 'Location' tab selected. The 'Name' field contains 'textfinder'. The 'Path' field contains 'RA7532\_Program\_II\_ENC/Unidade\_2/aula\_Exemplo\_3/temp/TextFinder'. There is a 'Browse...' button next to the path field. At the bottom right, there are 'Next >' and 'Cancel' buttons.

Qt Resource File

Location

Location Summary

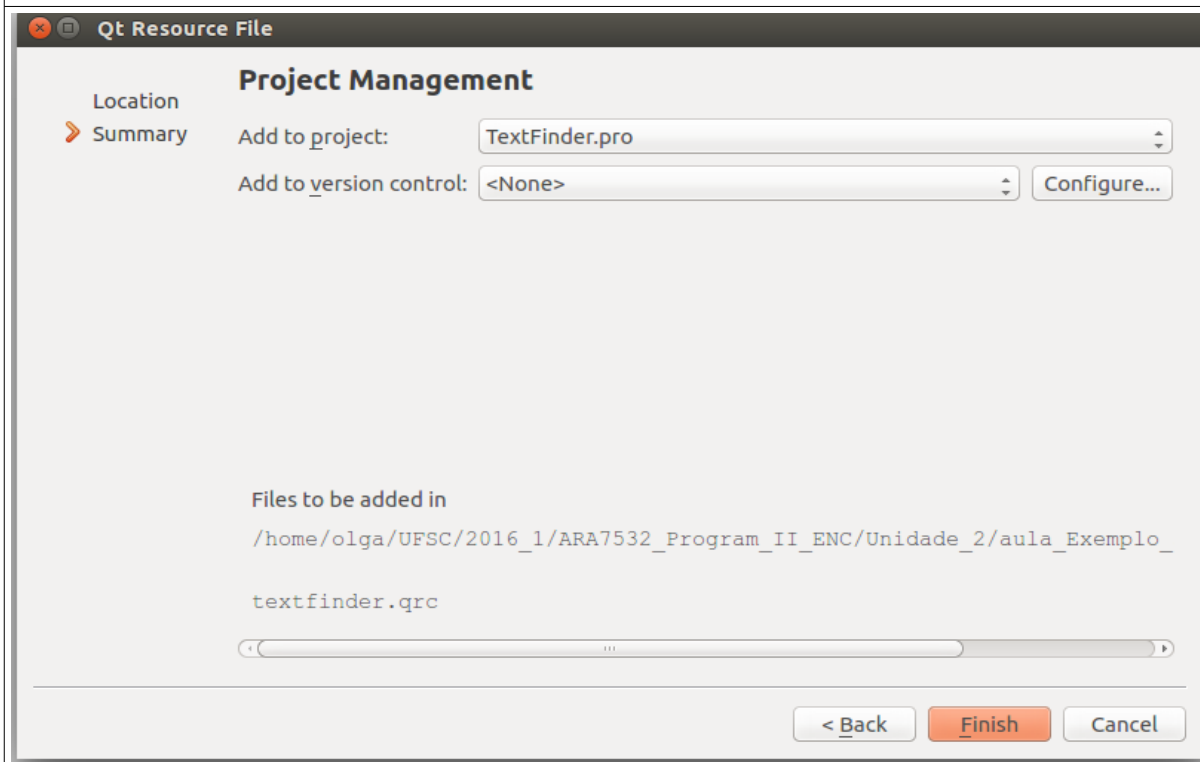
Name: textfinder

Path: RA7532\_Program\_II\_ENC/Unidade\_2/aula\_Exemplo\_3/temp/TextFinder Browse...

Next > Cancel

**10) (C)** Na janela **Project Management** deixar as opções **default**

### Project Management



The image shows the 'Qt Resource File' dialog box with the 'Project Management' tab selected. The 'Add to project' dropdown is set to 'TextFinder.pro'. The 'Add to version control' dropdown is set to '<None>'. There is a 'Configure...' button next to the version control dropdown. At the bottom, there are '< Back', 'Finish', and 'Cancel' buttons. The 'Files to be added in' section shows the path '/home/olga/UFSC/2016\_1/ARA7532\_Program\_II\_ENC/Unidade\_2/aula\_Exemplo\_textfinder.qrc'.

Qt Resource File

Project Management

Location Summary

Add to project: TextFinder.pro

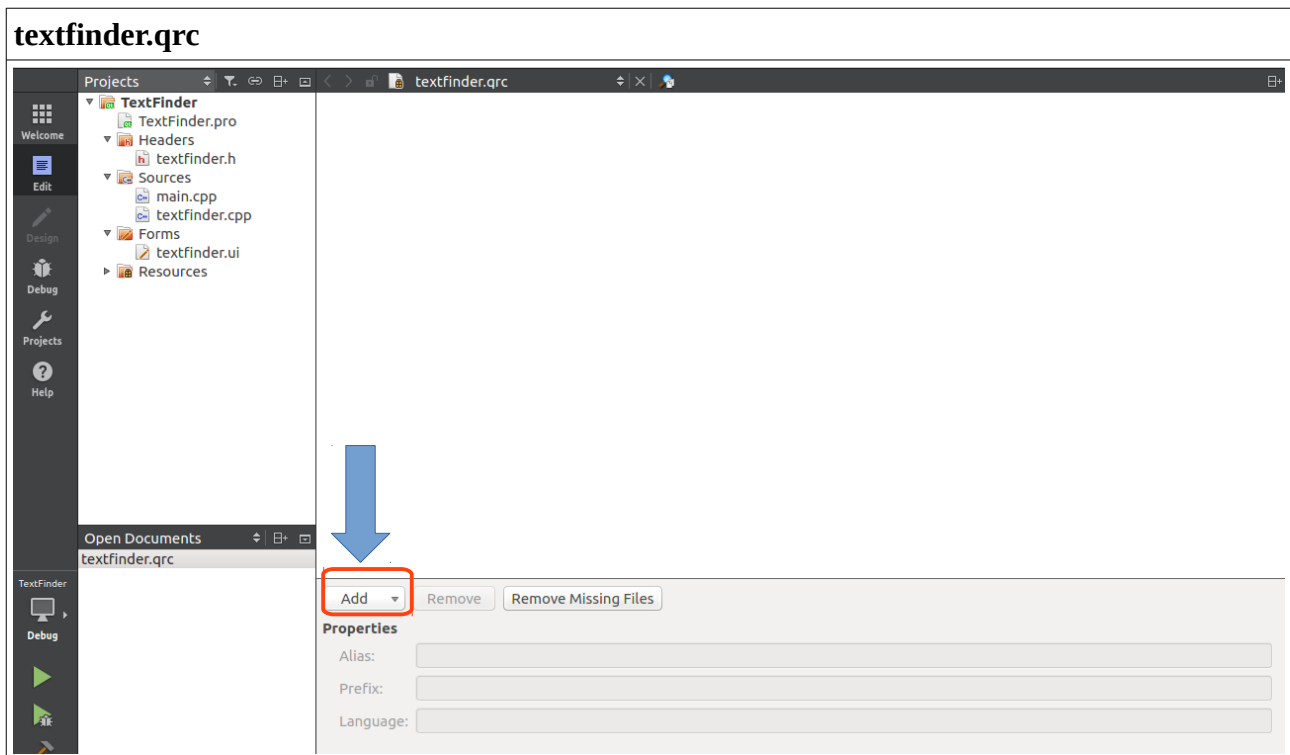
Add to version control: <None> Configure...

Files to be added in

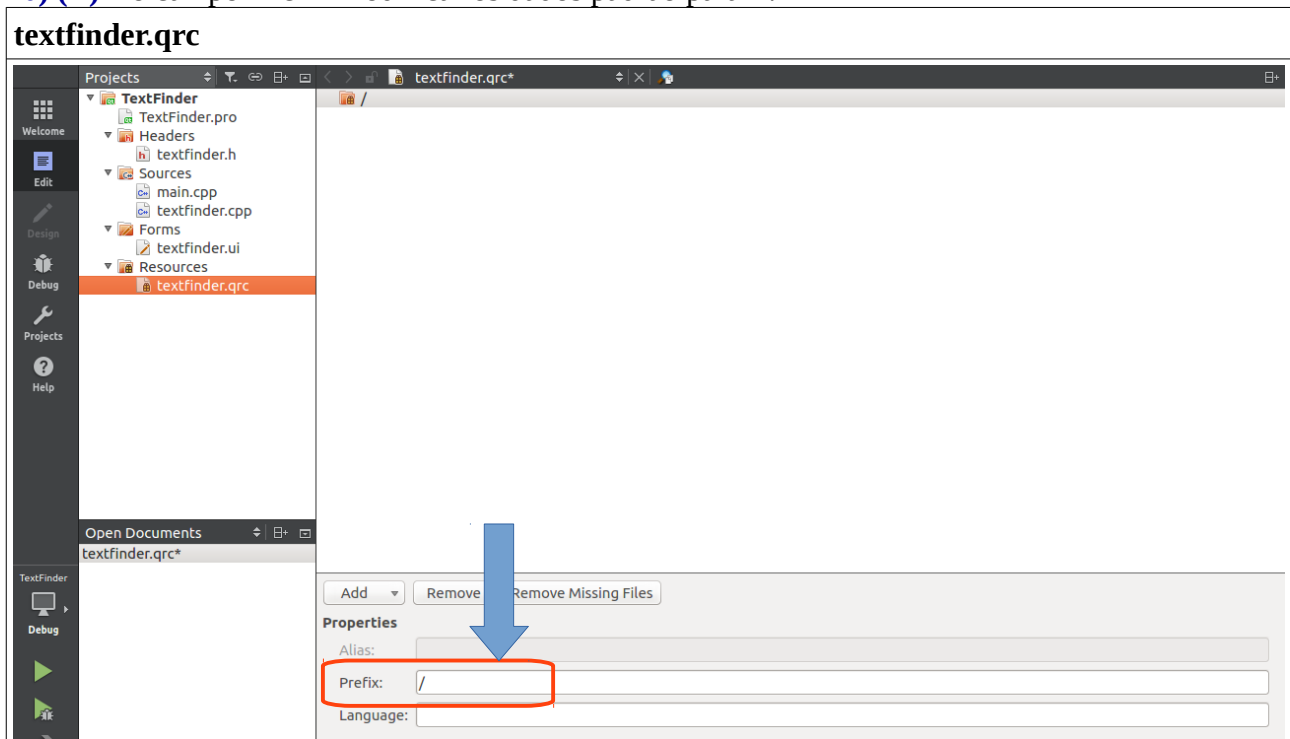
/home/olga/UFSC/2016\_1/ARA7532\_Program\_II\_ENC/Unidade\_2/aula\_Exemplo\_textfinder.qrc

< Back Finish Cancel

10) (D) Selecionar **Add** -> **Add Prefix**

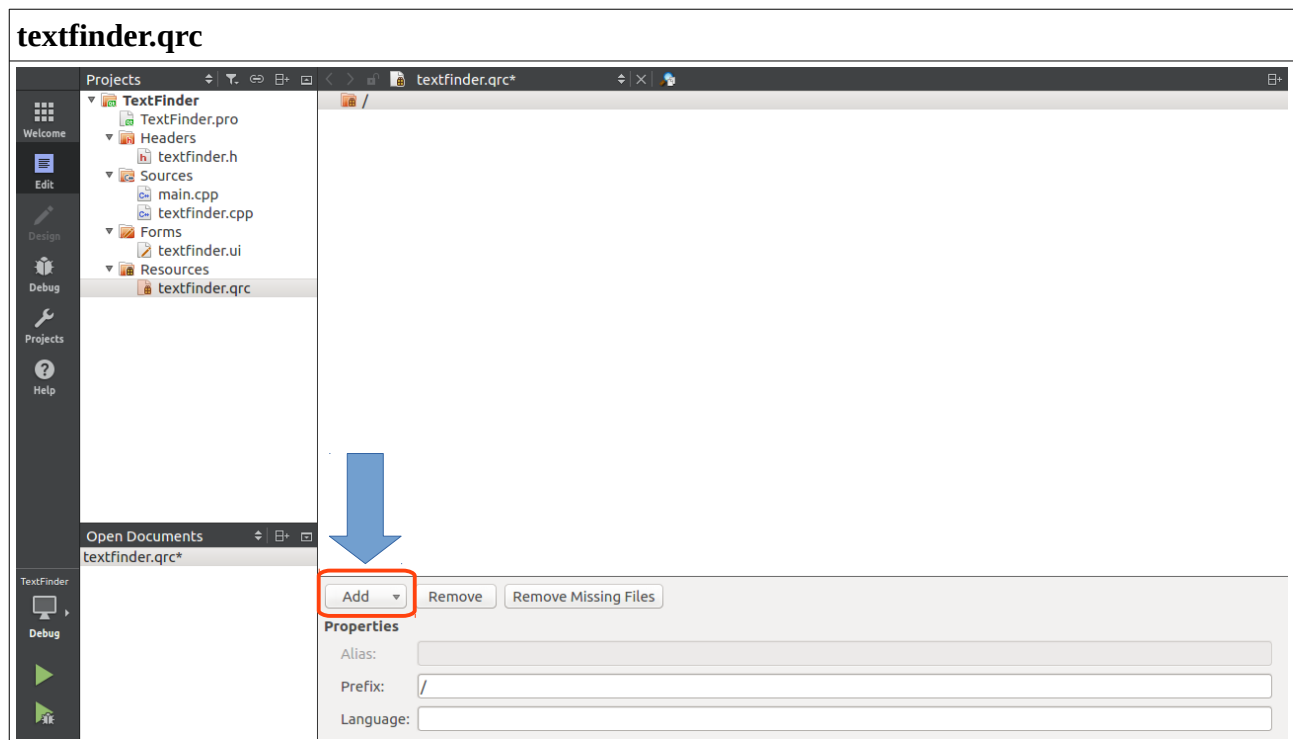


10) (E) No campo **Prefix** modificar os dados padrão para “/”

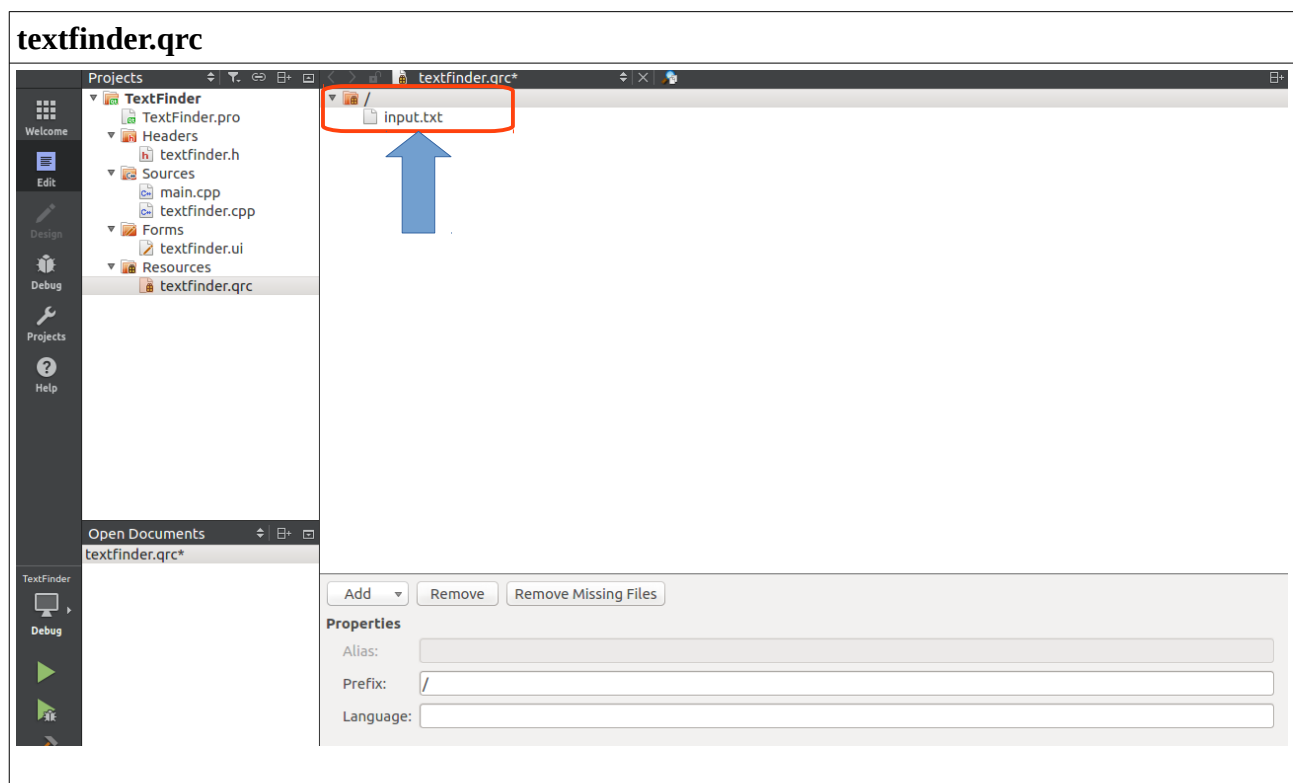




10) (F) Selecionar **Add -> Add Files** e localizar e adicionar o arquivo **input.txt**



10) (G) O arquivo **input.txt** vai aparecer adicionado no projeto



11) Compilar e executar o projeto.

A janela principal terá mais ou menos essa aparência (a aparência depende do sistema operacional).

O programa deve buscar e marcar no texto as palavras especificadas no campo Keyword.

Execução do programa

