

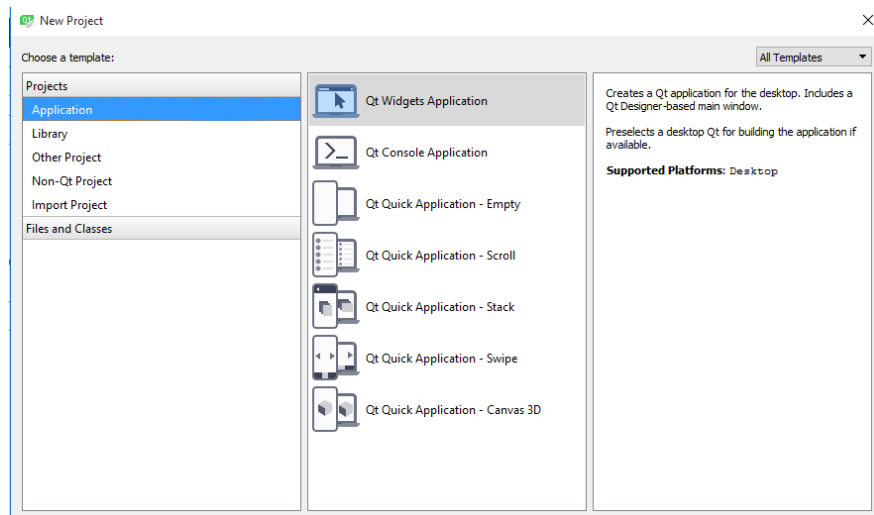
Introducción al desarrollo de aplicaciones con interfaz gráfica de usuario (GUI) con Qt Creator

Para desarrollar los ejemplos aquí presentados y cualquier futura asignación en la que se vaya a implementar interfaces gráficas de usuario será estrictamente necesario el uso de Qt Creator, en estos casos es improcedente el uso de compiladores en línea.

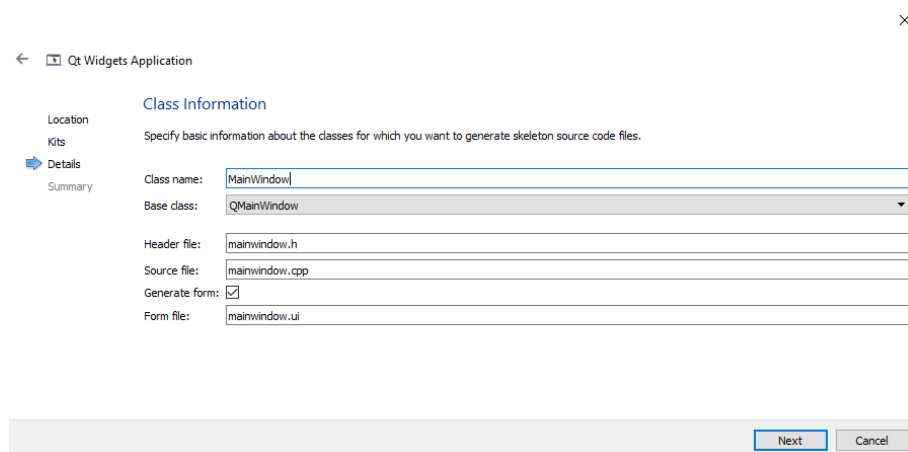
Ejemplo 1: Contador simple

Objetivos: Introducir el uso de los elementos (widgets) “QLineEdit”, “QPushButton”, “QLabel” y los conceptos generales relacionados con la interfaz gráfica de Qt: “ui”, “widget”, “slot”, etc.

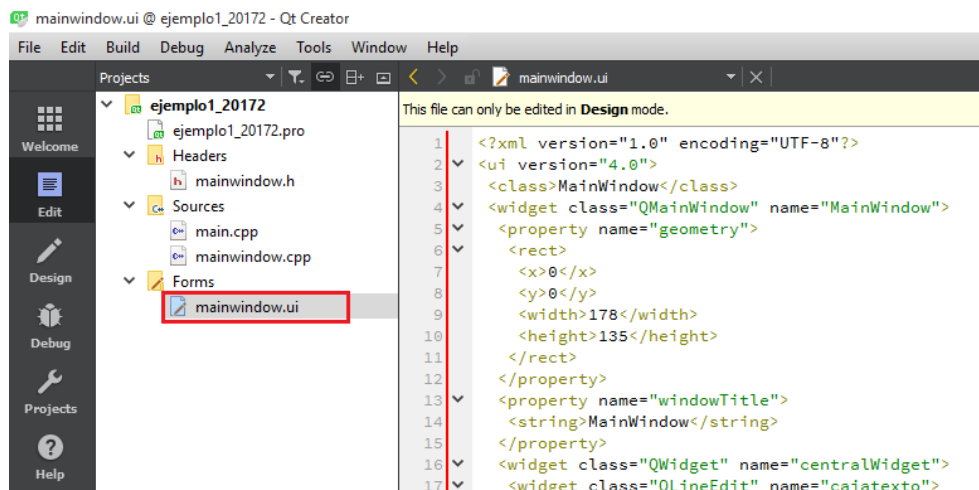
1. Crear un nuevo proyecto gráfico con Qt (Qt Widgets Application). En Projects seleccione Application y en la parte derecha seleccione Qt Widgets Application.



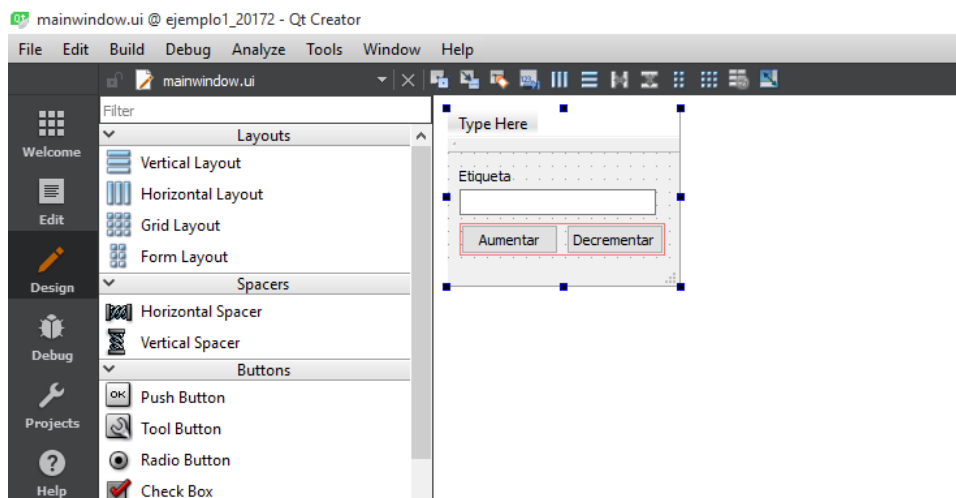
2. En la información de clase (Class Information), puede dejar la información que sale por defecto. Asegúrese que en el campo Base class la opción seleccionada sea QMainWindow.



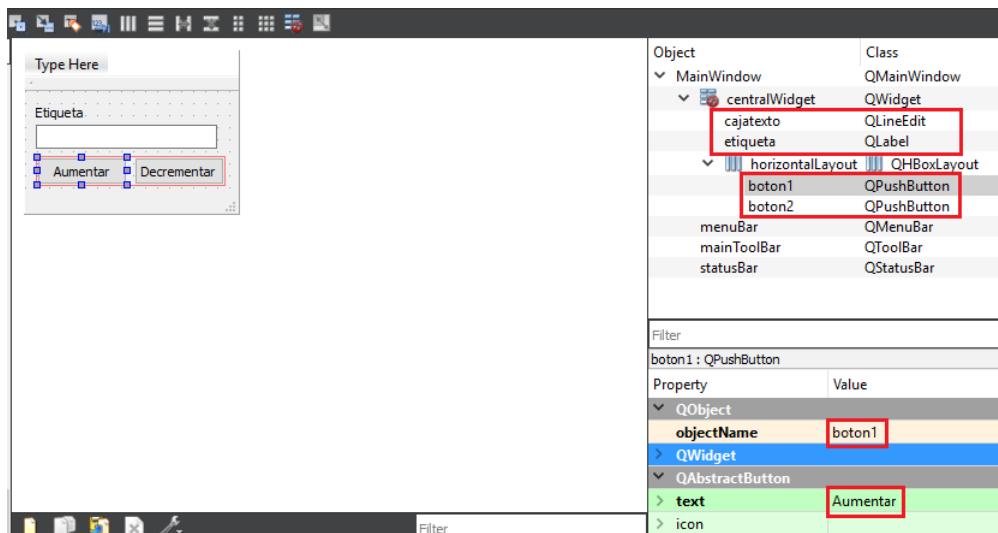
3. De doble clic en el archivo mainwindow.ui de la carpeta Forms.



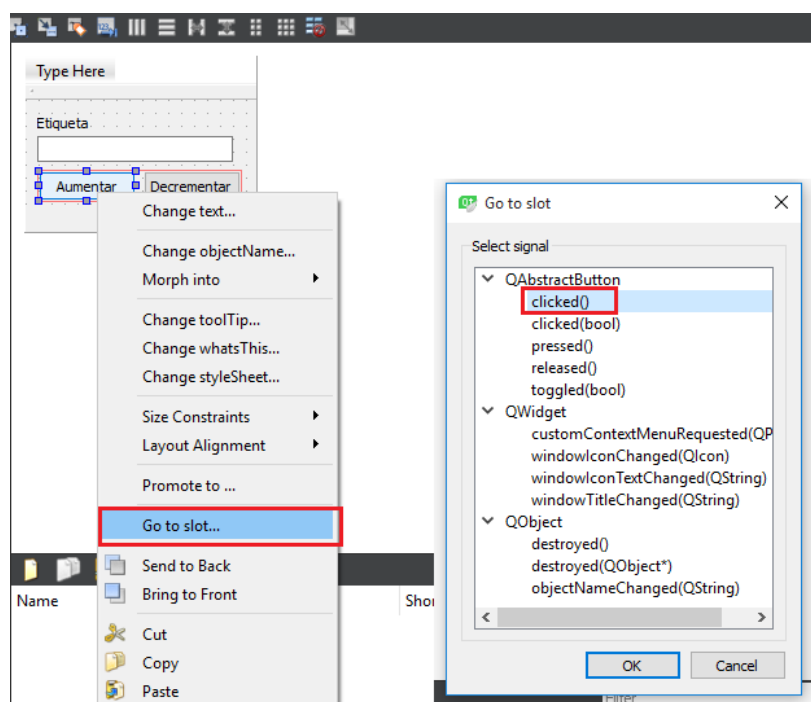
4. Diseñe la interfaz gráfica de su aplicación. En el lienzo de la interfaz, conocido como “centralWidget” agregue los siguientes elementos: dos QPushButton (Buttons -> QPushButton), un QLineEdit (Input Widgets -> QLineEdit) y un QLabel (Display Widgets -> QLabel).



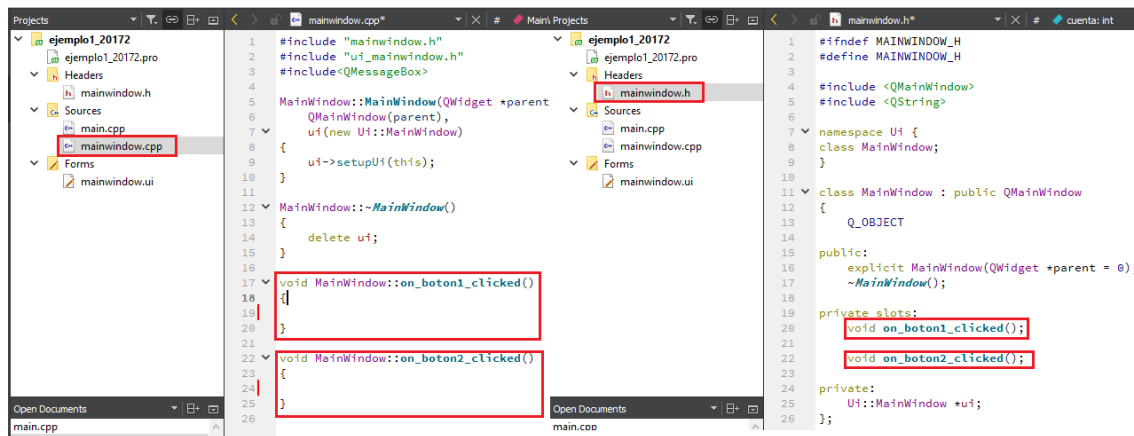
5. Utilizando el menú ubicado en la parte derecha del editor de la interfaz gráfica cambie el texto por defecto que aparece en los botones por “Aumentar” y “Disminuir” respectivamente y cambie los nombres de los elementos (Widgets) por denominaciones que sean para usted más familiares y de fácil recordación.



6. Definir la funcionalidad de los botones. De clic derecho en los botones y luego clic en la opción “Go to Slot” y luego seleccione el método “clicked()”. De esta forma se está seleccionando un método que permite realizar acciones luego de dar clic en los botones. Luego clic en OK.



7. Verifique que los métodos (slots) “clicked()” de los botones se agregan a la clase “MainWindow”.



8. Agregue la lógica necesaria para que al dar clic en el botón “Aumentar” aumente una cuenta y dicho aumento se muestre en el widget QLabel y en el widget QLineEdit debe mostrar un texto que diga que aumento la cuenta. De igual manera para el botón “Disminuir”.

Agregue a la clase “MainWindow” 3 atributos. Un atributo tipo int para llevar la cuenta, y dos atributos de tipo QString para las frases que va a poner en el widget QLineEdit.

```

1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3
4  #include <QMainWindow>
5  #include <QString>
6
7  namespace Ui {
8      class MainWindow;
9  }
10
11 class MainWindow : public QMainWindow
12 {
13     Q_OBJECT
14
15 public:
16     explicit MainWindow(QWidget *parent = 0);
17     ~MainWindow();
18
19 private slots:
20     void on_boton1_clicked();
21
22     void on_boton2_clicked();
23
24 private:
25     Ui::MainWindow *ui;
26     QString aumentar = "Aumento";
27     QString disminuir = "Disminuyo";
28     int cuenta = 0;
29 };
30
31 #endif // MAINWINDOW_H

```

En los métodos clicked() de los botones codifique el aumento y disminución del atributo cuenta, también lleve el valor de la cuenta al widget QLabel y lleve las frases de los

atributos QString al widget QLineEdit. Nota: Para poder llevar la cuenta que es de tipo int a un QLabel que recibe datos de tipo QString, debe hacer un casting con la función `QString::number(dato_int_a_convertir)`.

Para poder invocar los widgets desde los métodos clicked() debe en primer lugar invocar la interfaz de usuario "ui", luego con el operador flecha puede invocar los elementos que tenga esa interfaz de usuario, en este caso QLabel y QLineEdit (por los nombres que se le asignaron). Ejemplo: `ui->myLineEdit`. Luego con el operador flecha puede llamar los métodos de cada uno de los widgets, en este caso necesita métodos para poner líneas de texto en ellos. Ejemplo: `ui->myLineEdit->setText("mi_linea_de_texto")`.

```
1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3  #include <QMessageBox>
4
5  MainWindow::MainWindow(QWidget *parent) :
6      QMainWindow(parent),
7      ui(new Ui::MainWindow)
8  {
9      ui->setupUi(this);
10 }
11
12 MainWindow::~MainWindow()
13 {
14     delete ui;
15 }
16
17 void MainWindow::on_boton1_clicked()
18 {
19     ui->cajatexto->setText(aumentar);
20     cuenta++;
21     ui->etiqueta->setText(QString::number(cuenta));
22 }
23
24 void MainWindow::on_boton2_clicked()
25 {
26     ui->cajatexto->setText(disminuir);
27     cuenta--;
28     ui->etiqueta->setText(QString::number(cuenta));
29 }
```

Actividad evaluativa 1 Tarea 4: Diseñe una calculadora “suma-huevos” con interfaz gráfica. La calculadora debe tener teclas para: los números del 0 al 9; para las 4 operaciones básicas (suma, resta, multiplicación y división); para el punto decimal; para el resultado (tecla signo igual =) y para resetear la pantalla a cero en caso de equivocarse en la operación (tecla CE/C). Al ingresar los números los debe mostrar en un widget `QLCDNumber` al igual que el resultado de la operación. Adicionalmente, al presionar la tecla igual (=) despliegue un mensaje en una ventana que diga la operación que se ha realizado y el resultado de la misma, para esto puede hacer uso del widget `QMessageBox` que viene incluido en la librería de C++ del mismo nombre.