
	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 1

INFORME DE LABORATORIO

INFORMACION BASICA					
ASIGNATURA:	Tecnología de Objetos				
TITULO DE LA PRACTICA:	Introducción a Qt				
NUMERO DE LA PRACTICA:	04	AÑO LECTIVO:	2025 - B	N° SEMESTRE:	VI
FECHA DE PRESENTACION:	08 / 10 / 2025	HORA DE PRESENTACION:	--:-- PM		
INTEGRANTE (s): ■ Huayhua Hillpa, Yourdyy Yossimar				NOTA:	
DOCENTE (s): ■ Mg. Escobedo Quispe, Richart Smith					

1. Tarea

1.1. Objetivo



- Introducción al lenguaje Qt.

1.2. Problema propuesto:

- [1] Crear una interface gráfica (que implemente señales y slots) que muestre una lista de nombres de colores, al dar clic sobre alguno que se muestre un Label o un Text con el nombre del color.
- [2] Averiguar la implementación por signals y Slots, para comunicar los objetos del formulario del Task.ui (por ejemplo una caja de texto, o un Label) desde el MainWindows.h y CPP.

2. Equipos, materiales y temas utilizados

- Subsistema de Windows para Linux (WSL) con Ubuntu.
- Sistema operativo: Microsoft Windows [Versión 10.0.26100.6584]
- TeX Live 2025
- Helix 25.01.1 (e7ac2fcd)
- Visual Studio Code 1.104.0 x64
- Git version 2.41.0.windows.1
- Cuenta activa en GitHub para la gestión de repositorios remotos.
- Qt Creator
- Leguaje de programación C++
- Librería Qt

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 2</p>

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/yhuayhuahi/Teo.git>
- URL para el laboratorio (04) en el Repositorio GitHub.
- <https://github.com/yhuayhuahi/Teo/tree/main/laboratorios/lab04>

4. Desarrollo de las actividades

4.1. Actividad 01: Implementación en C++ con Qt

4.1.1. Función Main en C++

A continuación se muestra la función main implementada en C++ utilizando la librería Qt para crear una aplicación gráfica que muestra una lista de colores y actualiza un label con el nombre del color seleccionado.

Listing 1: Función Main en cpp - Primera implementación

```

1  #include "mainwindow.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      QApplication app(argc, argv);
7
8      MainWindow w;
9      w.show();
10
11     return app.exec();
12 }
```



4.1.2. Clase MainWindow en C++

A continuación se muestra el archivo de encabezado (header) de la clase MainWindow en C++ que define la interfaz gráfica y los elementos necesarios para la aplicación.

Listing 2: Header de la clase MainWindow en C++

```

1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3
4  #include <QMainWindow>
5  #include <QListWidgetItem>
6
7  QT_BEGIN_NAMESPACE
8  namespace Ui { class MainWindow; }
9  QT_END_NAMESPACE
10
11  class MainWindow : public QMainWindow {
12      Q_OBJECT
13
14  public:
15      explicit MainWindow(QWidget *parent = nullptr);
16      ~MainWindow();
17 }
```

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 3

```

18 private slots:
19     void onColorItemClicked(QListWidgetItem *item);
20
21 private:
22     Ui::MainWindow *ui;
23 };
24
25 #endif // MAINWINDOW_H

```

A continuación se muestra la implementación de la clase MainWindow en C++ que maneja la interfaz gráfica y la lógica para actualizar el label con el nombre del color seleccionado de la lista.

Listing 3: Clase MainWindow en C++



```

1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3
4  MainWindow::MainWindow(QWidget *parent)
5      : QMainWindow(parent)
6      , ui(new Ui::MainWindow)
7  {
8      ui->setupUi(this);
9
10     // Agregamos algunos colores a la lista
11     ui->colorListWidget->addItem("Rojo");
12     ui->colorListWidget->addItem("Verde");
13     ui->colorListWidget->addItem("Azul");
14     ui->colorListWidget->addItem("Amarillo");
15     ui->colorListWidget->addItem("Negro");
16
17     // Conectamos la señal al slot
18     connect(ui->colorListWidget, &QListWidget::itemClicked,
19           this, &MainWindow::onColorItemClicked);
20 }
21
22 MainWindow::~MainWindow() {
23     delete ui;
24 }
25
26 void MainWindow::onColorItemClicked(QListWidgetItem *item) {
27     ui->colorLabel->setText("Color seleccionado: " + item->text());
28 }

```

4.1.3. Pruebas de ejecución:

Se realizaron pruebas de ejecución para verificar que la aplicación funcione correctamente. Al seleccionar un color de la lista, el label se actualiza con el nombre del color seleccionado.

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 4</p>

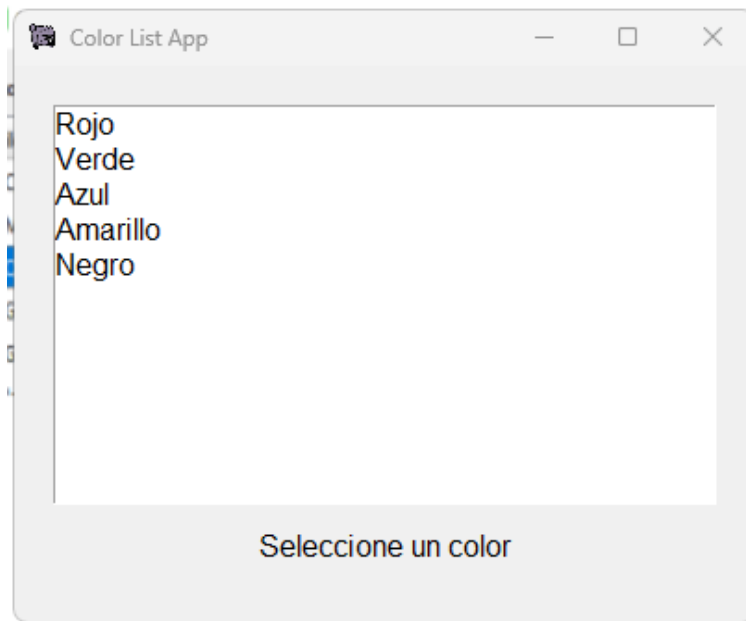


Figura 1: Interfaz inicial de la aplicación Qt

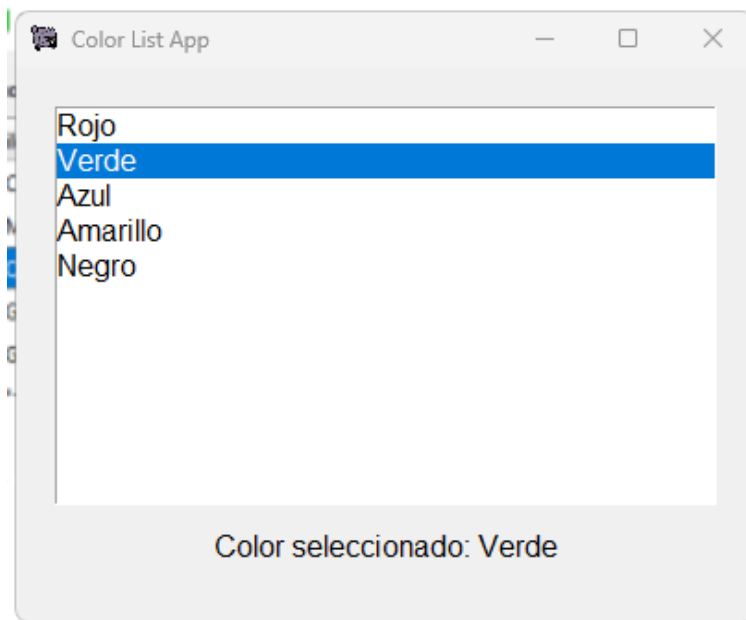




Figura 2: Interfaz de la aplicación Qt después de seleccionar un color

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 5</p>

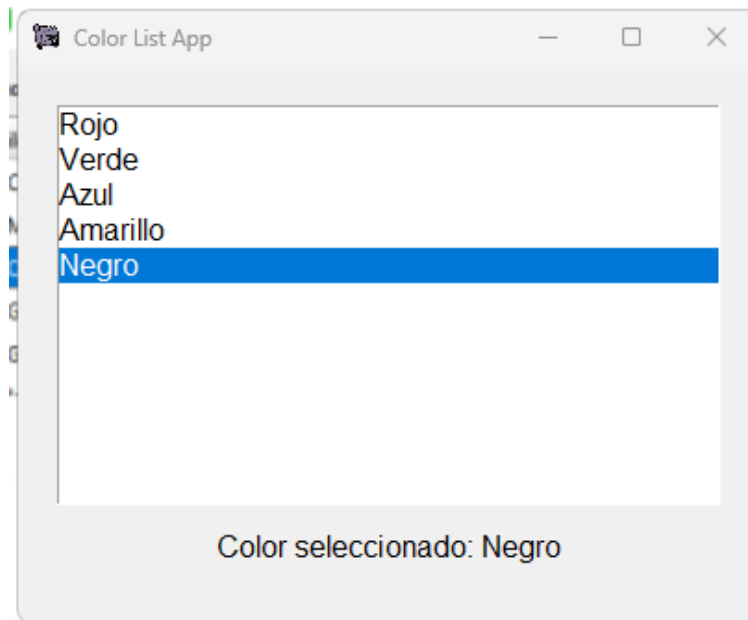


Figura 3: Interfaz de la aplicación Qt después de seleccionar otro color

4.2. Actividad 02: Averiguar la implementación por signals y Slots, para comunicar los objetos del formulario del Task.ui (por ejemplo una caja de texto, o un Label) desde el MainWindows.h y CPP.

En Qt, la forma correcta de comunicar widgets definidos en un archivo .ui con otra clase (por ejemplo MainWindow) es mediante el mecanismo de señales y ranuras (signals & slots). Este sistema permite que los objetos se comuniquen sin depender directamente entre sí.

- El Task.ui se convierte (por uic) en una clase Ui::Task con los punteros a los widgets (ej. lineEdit, label).
- Se crea una clase TaskForm que herede de QWidget y use esa UI.
- Se define un signal en TaskForm para avisar a MainWindow de cambios (por ejemplo, cuando el usuario escribe).
- Se define un slot para permitir que MainWindow actualice los widgets.
- En MainWindow, se conectan ambas partes con connect().



A continuación se muestra un ejemplo de cómo implementar esta comunicación entre TaskForm y MainWindow utilizando señales y ranuras en Qt.

Listing 4: TaskForm.h

```

1  #include <QWidget>
2  namespace Ui { class Task; }
3
4  class TaskForm : public QWidget {
5      Q_OBJECT
6  public:
7      explicit TaskForm(QWidget *parent = nullptr);
8      ~TaskForm();

```

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 6

```

9
10 signals:
11     void textChanged(const QString &text); // avisará al MainWindow
12
13 public slots:
14     void setLabel(const QString &msg); // llamado por MainWindow
15
16 private:
17     Ui::Task *ui;
18 };

```

Listing 5: TaskForm.cpp

```

1  #include "TaskForm.h"
2  #include "ui_Task.h"
3
4  TaskForm::TaskForm(QWidget *parent) : QWidget(parent), ui(new Ui::Task) {
5      ui->setupUi(this);
6
7      // cuando el usuario escribe, emitimos la señal
8      connect(ui->lineEdit, &QLineEdit::textChanged,
9              this, &TaskForm::textChanged);
10 }
11
12 TaskForm::~TaskForm() { delete ui; }
13
14 void TaskForm::setLabel(const QString &msg) {
15     ui->label->setText(msg);
16 }

```

Listing 6: MainWindow.cpp

```

1  #include "MainWindow.h"
2  #include "TaskForm.h"
3
4  MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent) {
5      auto *task = new TaskForm(this);
6      setCentralWidget(task);
7
8      // TaskForm -> MainWindow
9      connect(task, &TaskForm::textChanged, this, [this, task](const QString &txt){
10         task->setLabel("Texto: " + txt);
11     });
12 }



```

1. Usa signals para notificar eventos (de TaskForm a MainWindow).
2. Usa slots para reaccionar o modificar el formulario (de MainWindow a TaskForm).
3. Usa la sintaxis moderna de connect() (con punteros a miembros o lambdas).

4.3. Commits realizados

4.3.1. Primer Commit

- Este commit se hizo después de terminar la primera implementación de código para C++ con Qt,

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 8

```
(venv) ~/.../Teo/laboratorios/lab04
23:36:35 tree /f
Listado de rutas de carpetas para el volumen Windows
El número de serie del volumen es A025-E3A0
C:.\
├── ColorListApp
│   ├── .gitignore
│   ├── CMakeLists.txt
│   ├── main.cpp
│   ├── mainwindow.cpp
│   ├── mainwindow.h
│   └── mainwindow.ui
├── informe-latex
│   ├── desarrollo.tex
│   ├── main.tex
│   └── rubricas.tex
└── build
```



Figura 6: Estructura de archivos y carpetas del laboratorio

6. Rúbricas

6.1. Entregable Informe

Cuadro 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 9

6.2. Rúbrica para el contenido del Informe y demostración



- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Cuadro 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Cuadro 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1.5	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		16	

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 10</p>

7. Referencias

- [1] <https://doc.qt.io/qt-6/signalsandslots.html>
- [2] <https://doc.qt.io/qt-6/qtwidgets-widgets-mainwindow-example.html>
- [3] <https://doc.qt.io/qt-6/qtwidgets-widgets-qmainwindow.html>