



UTN Facultad Regional La Rioja

Trabajo Practico N° 02

2024



Índice

Introducción.....	3
Entrega.....	3
Criterio de evaluación.....	3
Rubrica de Evaluación.....	4
Enunciados.....	5
Ejercicio 1: Signal & Slot.....	5
Ejercicio 2: Deque	5
Ejercicio 3: Calculadora.....	6
Ejercicio 4: Formulario de Personas	7
Ejercicio 5: Torneo de Artes Marciales.	8



Introducción

El presente trabajo práctico tiene como objetivo poner en práctica los conocimientos adquiridos en la materia hasta ahora.

Los ejercicios propuestos abarcan el uso de los conocimientos adquiridos en programación I (variables, el manejo de estructuras condicionales, estructuras cíclicas, vectores, matrices, funciones, punteros, estructura y ficheros) además de los conocimientos adquiridos en programación II (funciones cin, cout, clases, QT Disegned, Signals & Slot, QMessageBox, QStyleSheets, Biblioteca de Plantillas Estándar (STL), Algoritmos).

Es fundamental comprender y aplicar adecuadamente estos conceptos para resolver eficientemente los problemas planteados. Además, se espera que los estudiantes adquieran habilidades en la organización del código, la lectura de entradas desde el teclado, y la presentación de resultados en la terminal.

Entrega

Se deberá entregar un archivo comprimido con WinRAR en formato .zip o .rar, este archivo deberá contener:

1. Un archivo PDF con los códigos realizados por cada ejercicio junto con la imagen de la salida de la Terminal (Aplicación Programada).
2. Un archivo de código fuente por cada ejercicio realizado, en este archivo deber estar el código funcional y comentado correctamente.

Nombres de los Archivos:

Los archivos deberán tener el siguiente formato de nombre:

- **Archivo Comprimido:** "TP 01 - Comisión X - Apellido Nombre"
 - *Ejemplo:*
 - TP 02 – Comisión B – Oviedo Carlos.rar
- **Archivo PDF:** "TP 02 - Comisión X - Apellido Nombre "
 - *Ejemplo:*
 - TP 02 – Comisión B – Oviedo Carlos.pdf
- **Código Fuente:** " TP 02 - Comisión X - Apellido Nombre – Ejercicio xx"
 - *Ejemplo:*
 - TP 02 – Comisión B – Oviedo Carlos – Ejercicio 01.cpp

Fecha de Entrega: *Lunes 04 de Octubre del 2024*

Criterio de evaluación

Los trabajos prácticos de programación se calificarán según una rúbrica de 10 puntos por ejercicio, luego para la nota final del práctico se dividirá el puntaje total en 10. Criterios a evaluar:

- **Funcionamiento del programa:** Cada programa debería funcionar correctamente en todas las entradas. Además, si hay alguna especificación sobre cómo funciona el programa debería escribirse, o cómo debe aparecer el resultado, se deben seguir esas especificaciones.
- **Legibilidad:** Las variables y funciones deben tener nombres significativos. El código debe organizarse en funciones cuando corresponda. Debe haber una cantidad adecuada de espacio en blanco para que el código sea legible y la sangría debe ser coherente.
- **Documentación:** El código y funciones deben estar comentados apropiadamente. Sin embargo, no todas las líneas deben ser comentadas porque eso hace que el código esté demasiado ocupado. Pensar detenidamente dónde se necesitan comentarios.
- **Elegancia del código:** Hay muchas formas de escribir la misma funcionalidad en el código y algunas de ellas son innecesariamente lentas o complicadas. Por ejemplo, si está repitiendo el mismo código, debería estar dentro de una nueva función o bucle for.
- **Especificaciones y entrega del práctico:** Cada programa debe ser guardado con un determinado nombre de archivo u otras especificaciones similares y debe ser entregado en tiempo y forma. Ver especificaciones de entregas indicadas anteriormente.



Tecnicatura en Programación Universitaria - Programación II
TP Nº 02

Rubrica de Evaluación

Funcionamiento	4	3	2	0
	El programa siempre funciona correctamente y cumple con las especificaciones.	El programa no cumple con detalles menores de las especificaciones, o funciona incorrectamente en algunas entradas.	El programa no cumple con detalles significativos de las especificaciones, o exhibe un comportamiento incorrecto.	El programa no funciona, o no compila.
Legibilidad	2	1	0,5	0
	El código es limpio, entendible y bien organizado.	Problemas menores como sangría inconsistente, denominación de variables, organización general	Al menos un problema importante que dificulta la lectura.	Varios problemas importantes que dificultan su lectura.
Documentación	2	1	0,5	0
	El código está bien documentado.	Unos o dos lugares pueden beneficiarse de comentarios, o el código está comentado de más	La falta de comentarios dificulta la comprensión del código.	El código no presenta comentarios.
Elegancia		1	0,5	0
		El código utiliza adecuadamente bucles for y métodos para código repetido, y la codificación es mínima.	El código utiliza un enfoque mal elegido al menos en un lugar, por ejemplo, codificar algo que podría implementarse a través de un bucle for.	En varias ocasiones en que el código podría haber utilizado un enfoque más fácil/rápido/mejor.
Especificaciones		1	0,5	0
		El ejercicio cumple con las especificaciones.	El ejercicio no cumple con especificaciones menores.	El ejercicio no cumple con especificaciones significativas.
Entrega			0	-3
			Se entregó dentro del plazo establecido.	Se entregó fuera del plazo establecido.



Enunciados

Ejercicio 1: Signal & Slot

Generar un programa en QT que usando Signal & Slot (connect) poner tres botones en el front usando el layout horizontal (uno debajo del otro) y cambiar a 3 colores diferentes el fondo de la ventana Widget.

Los colores son los siguientes:

1. #E9967A
2. #B2FF9E
3. #DBFCFF

Ejercicio 2: Deque

El programa deberá tener una ventana principal que permita al usuario agregar números al frente o al final de un deque, eliminar un elemento, mostrarlos en la pantalla y ordenar de mayor a menor o de menor a mayor la visualización de los datos cargados. Además de tener una ventana secundaria que permita buscar un elemento y si este se encuentra muestra la ubicación y cuantas veces se repite en el Deque usando los algoritmos correspondientes.

El programa deberá tener estos QObjects básicos (puede agregarse los que sean necesarios para mejorar el programa para la usabilidad):

- Ventana Principal:
 - 1 QLineEdit para ingresar un número.
 - 4 QPushButton ("Agregar al Frente", "Agregar al Final", "Eliminar Elemento" y "Mostrar Deque", "Buscar Elemento").
 - 1 QLabel para mostrar el contenido.
- Ventana Secundaria:
 - 1 QLineEdit para ingresar un número.
 - 1 QPushButton ("Buscar Elemento").
 - 2 QLabel para mostrar la ubicación y la cantidad de veces que se repite.

Validaciones: El usuario solo podrá ingresar números positivos, y podrá eliminar los elementos.... Usar **QMessageBox** para advertir al usuario del error en caso de no cumplir la validación.



Ejercicio 3: Calculadora

Un cliente nos pide desarrollar una calculadora tal como está en la imagen. Las operaciones que debe realizar son:



- Borrar (C)
- Suma
- Restar
- Dividir
- Multiplicar
- Cambiar signo (+/-)
- Calcular porcentaje (%)
- Agregar paréntesis ().
- Igual

La calculadora deberá tener un archivo .css con el estilo diferente al estilo básico de la imagen.

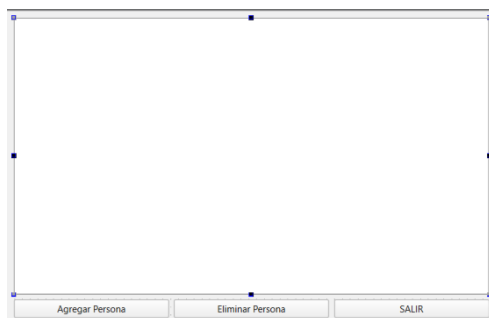
- **Validaciones:** El programa deberá utilizar **QMessageBox** para las validaciones de todas las operaciones que no se puedan realizar por ejemplo: Dividir por cero.



Ejercicio 4: Formulario de Personas

Un cliente no solicita un programa que permita cargar Nombre, Apellido, DNI (Solo números), Edad, Peso (Kgr), Altura (cm) de personas y que estos datos cargados se visualicen en una tabla, para ello vamos a usar dos ventanas diferentes una para la tabla y otra para el formulario, además de utilizar QMessageBox para indicar si el usuario se equivoca al ingresar los datos.

Se deberá usar una lista e incluir un archivo css que modifique el estilo de las ventanas propuesta a continuación.



- QWiget:
 - QTable
 - Botón Agregar Persona (agrega a la lista)
 - Botón Eliminar Persona (elimina de la lista)
 - Botón SALIR (cerrar programa)

- QDialog:
 - 07 QLabel
 - 03 QLineEdit
 - 01 QSpinBox
 - 02 QDouble SpinBox
 - 01 QPlainTextEdit
 - 02 botones (OK/CANCEL).

- **Validaciones:** DNI Solo números, Números negativos. Usar **QMessageBox** para advertir al usuario del error en caso de no cumplir la validación



Ejercicio 5: Torneo de Artes Marciales.

Simula un Torneo de Artes Marciales, donde participarán diferentes luchadores (usar clases), y el sistema decidirá quién es el ganador.

El programa deberá tener un Formulario (Luchadores) con

- Nombre.
- Tres atributos:
 - Velocidad. (con valores entre 0 a 100).
 - Ataque. (con valores entre 0 a 100).
 - Defensa. (con valores entre 0 a 100).
- Comienza cada batalla con 100 de salud.

La Ventana Principal deberá simular la batalla, para ello deberá tener dos QComboBox donde estarán los jugadores (cargados desde una Lista) y seleccionados para la simulación, además debe incluir el botón para comenzar la simulación. **Batalla:**

- En cada batalla se enfrentan 2 luchadores.
 - El luchador con más velocidad comienza atacando.
 - El daño se calcula restando el daño de ataque del atacante menos la defensa del oponente.
 - El oponente siempre tiene un 20% de posibilidad de esquivar el ataque.
 - Si la defensa es mayor que el ataque, recibe un 10% del daño de ataque.
 - Después de cada turno y ataque, el oponente pierde salud.
 - La batalla finaliza cuando un luchador pierde toda su salud.
-
- **Validaciones:** Se deberá validar los datos cargados por el usuario Nombre (Solo letras) y Atributos (rango correcto). Usar **QMessageBox** para advertir al usuario del error en caso de no cumplir la validación
-