



UTN Facultad Regional La Rioja

Trabajo Practico Nº 03

2024



Índice

Introducción.....	3
Entrega.....	3
Criterio de evaluación.....	3
Rubrica de Evaluación.....	4
Enunciados.....	5
Ejercicio 01:.....	5
Ejercicio 02:.....	5
Ejercicio 03:.....	5
Ejercicio 04:.....	6
Ejercicio 05:.....	6



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 variables, funciones básicas como printf y scanf, así como el manejo de estructuras condicionales (if, if/else, switch) y estructuras cíclicas (while, do while, for), vectores, matrices, funciones, punteros, estructura y ficheros.

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 wirar 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 03 - Comisión X - Apellido Nombre"
 - *Ejemplo:*
 - TP 03 – Comisión B – Oviedo Carlos.rar
- **Archivo PDF:** "TP 03 - Comisión X - Apellido Nombre "
 - *Ejemplo:*
 - TP 03 – Comisión B – Oviedo Carlos.pdf
- **Código Fuente:** " TP 03 - Comisión X - Apellido Nombre – Ejercicio xx"
 - *Ejemplo:*
 - TP 03 – Comisión B – Oviedo Carlos – Ejercicio 01.c

Fecha de Entrega: _____

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 I
TP Nº 03

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 01: Calculadora Básica

Escribe un programa en C que implemente una calculadora básica. El programa debe:

1. Pedir al usuario que ingrese dos números.
2. Mostrar un menú con las opciones: 1. Sumar, 2. Restar, 3. Multiplicar, 4. Dividir.
3. Leer la opción seleccionada por el usuario.
4. Usar una estructura switch para realizar la operación correspondiente.
5. Mostrar el resultado en pantalla.
6. Preguntar al usuario si desea realizar otra operación (utilizando un bucle do-while).

Objetivo:

Desarrollar una calculadora básica que permita al usuario realizar operaciones aritméticas (suma, resta, multiplicación, división) utilizando funciones, estructuras condicionales y bucles.

NOTA: Validar todos los ingresos de datos del usuario.

Ejercicio 02: Promedio de Calificaciones

Escribe un programa en C que calcule el promedio de las calificaciones de un grupo de estudiantes. El programa debe:

1. Pedir al usuario que ingrese la cantidad de estudiantes.
2. Declarar un vector para almacenar las calificaciones.
3. Utilizar un bucle for para leer las calificaciones desde el teclado.
4. Implementar una función que reciba el vector y la cantidad de estudiantes, y devuelva el promedio de las calificaciones.
5. Mostrar el promedio en pantalla.

Objetivo:

Trabajar con vectores y funciones para calcular el promedio de un conjunto de calificaciones ingresadas por el usuario.

NOTA: Validar todos los ingresos de datos del usuario.

Ejercicio 03: Transpuesta de una Matriz

Escribe un programa en C que calcule la transpuesta de una matriz. El programa debe:

1. Pedir al usuario que ingrese el tamaño de la matriz (número de filas y columnas).
2. Declarar una matriz bidimensional y leer sus elementos desde el teclado.
3. Implementar una función que reciba la matriz y sus dimensiones, y calcule la transpuesta.
4. Mostrar la matriz transpuesta en pantalla.

Objetivo:

Manejo de matrices y funciones para calcular la transpuesta de una matriz.

NOTA: Validar todos los ingresos de datos del usuario.



Ejercicio 04: Gestión de Inventario con Estructuras

Escribe un programa en C que permita gestionar un inventario de productos. El programa debe:

1. Definir una estructura para representar un producto (nombre, código, precio, cantidad).
2. Declarar un vector de productos.
3. Implementar funciones para:
 - a. Agregar un nuevo producto.
 - b. Mostrar todos los productos.
 - c. Buscar un producto por código.
4. Guardar y cargar el inventario desde un archivo.
5. Utilizar un menú para que el usuario pueda seleccionar las distintas opciones.

Objetivo:

Uso de estructuras y archivos para gestionar un inventario simple.

NOTA: Validar todos los ingresos de datos del usuario.

Ejercicio 05: Operaciones con Punteros

Escribe un programa en C que realice las siguientes operaciones utilizando punteros:

1. Declarar dos variables enteras e inicializarlas con valores ingresados por el usuario.
2. Implementar una función que reciba dos punteros enteros y los intercambie (swap).
3. Implementar una función que reciba dos punteros enteros y calcule su suma y producto, almacenando los resultados en variables pasadas por referencia.
4. Mostrar los resultados de las operaciones en pantalla.

Objetivo:

Ejercitación al uso de punteros y funciones en C.

NOTA: Validar todos los ingresos de datos del usuario.