

# FINAL PROGRAMACIÓN II

Cátedra Programación II

– Lic. Natalia Colussi –

03/07/2024

Nombre Apellido:	Legajo:
email:	Nota:

## ¿Cómo resolver los ejercicios del final?

Las funciones que definan para cada programas tienen que seguir los pasos *La Receta*. Recuerde realizar el testing con por lo menos tres casos de test verificable con la librería `pytest` y el header `<assert.h>`. En el caso de que el testing no sea posible, enunciar los ejemplos correspondientes. Al finalizar la resolución del final, todos los programas generados se subirán al sitio de comunidades.

## Ejercicio 1: Midiendo el Paraná (Lenguaje C)

El/La director/a del Historial de Alturas Hidrométricas de la Cuenca del Paraná, en la Universidad del Litoral, nos facilitó las alturas del Río Paraná, medidas en los últimos 12 días del mes de Junio, las cuáles se indican a continuación:

$a_{\text{Alturas}} = [1.95, 1.89, 1.88, 1.86, 1.86, 2.0, 2.0, 2.0, 2.0, 2.07, 2.09, 2.12]$

Diseñe un programa el cual utilizando los valores anteriores, permita al director/a de esta institución calcular rápidamente mediante distintas funciones los siguientes ítems:

1. Promedio aritmético de la altura del Paraná,
2. ¿Cuál es la altura máxima registrada?
3. Cálculo de la Moda: sobre el conjunto de datos dados, establecer aquel valor que más se repite. El conjunto de datos puede tener una sola Moda, o tener varias Modas (multimodal) o no tener Moda cuando los datos se repitan de manera igualitaria, se dice amodal
  - Ejemplo 1: 1 2 3 3 4 5 5 5 5 5 6 7 8 3, la moda es 5 porque es el número que más se repite.
  - Ejemplo 2: 1 - 1 - 2 - 2 - 2 - 4 - 7 - 8 - 8 - 8 - 25, *Muestra multimodal*, porque el 2 y el 8 se repiten 3 veces. Cabe destacar que el 1 aunque se repite 2 veces, este se repite menos veces que el 2 y el 8.

- Ejemplo 3: 1 - 3 - 4 - 5 - 6 - 8 - 17- 25 *Muestra amodal*, no tiene moda.
4. Desviación del promedio: Dado el arreglo con la medidas del río Paraná y el valor de promedio de alturas, obtener otro arreglo con el valor absoluto de la diferencia entre cada medición con el promedio.

Guarde su solución en el archivo: `ejercicio1-APELLIDO.c`.

## Ejercicio 2: Enviando Palabras (Lenguaje C)

Dado un mensaje en inglés (no hay acentos, ni letras con diéresis, o ñs) se debe calcular mediante una función el costo de enviar el mensaje por un telégrafo. La agencia postal aplica el siguiente criterio para calcular el costo.

- Las letras cuestan, cada una, 10 pesos.
- Los dígitos tienen un valor de 20 pesos cada uno.
- Los caracteres especiales, que no sean letras ni números, cuestan 30 y .
- Los espacios en blanco no tienen valor alguno.

El main pedirá los datos por la entrada standard y mostrará por pantalla el costo del envío. En el pedido de datos se solicitará al usuario la longitud del mensaje y el mensaje en si mismo. Esto permitirá alocar la memoria para la longitud solicitada.

Guarde su solución en el archivo: `ejercicio2-APELLIDO.c`

## Ejercicio 3: Cumpleaños! (Lenguaje C)

Implementar en C un programa que contenga una arreglo de estructuras para almacenar las fechas de cumpleaños de 5 amigxs. Los datos a almacenar sobre cada amigx son nombre, día, mes y año. El programa debe indicarnos cuántos de nuestros amigxs cumplen los años un día y mes determinados introducidos por teclado.

- Diseñar la estructura `Cumple` en C y generar 5 instancias, inicializando sus valores de día, mes, año, y nombre. Crear el arreglo que contenga estos 5 datos.
- Diseñar la función `busqueda_cumple: Cumple[5] Int Int Int → Void`

La cual nos mostrará en pantalla los resultados de la busqueda. Deberán contemplarse los casos que no haya cumpleaños, y/o que mas de un amigx cumpla los años en el día especificado.

Guarde su solución en el archivo: `ejercicio3-APELLIDO.c`

## El Comisionista (todo se resuelve en Python)

Necesitamos organizar los envíos del comisionista de Ciudad Gótica. El comisionista realiza tres tipos de envíos a entre diferentes ciudades de la región, transportando: *documentos, cajas, y/o personas*. Para ayudarlo en su labor crearemos un programa que lo asista tal cual lo indicamos a continuación.

Guarde su solución en el archivo: `ejercicio4-APELLIDO.py`.

### Ejercicio 4.1: Tupla Envío

1. Diseñar una tupla llamada Envío la cuál almacenará la siguiente información: *origen, destino, transporte, precio, estado*.
2. Una función llamada `crearEnvío`, la cual dados un origen, un destino, un transporte, y un precio, creará y retornará un dato de tipo Envío. Por defecto se seteará el estado en no entregado.
3. Una función llamada `imprimirEnvío`, la cual toma un Envío e imprime los datos de éste en pantalla.
4. Una función llamada `EnvíoEntregado`, la cual toma un Envío y actualiza el campo estado para manifestar que el mismo ha sido entregado retornando el Envío actualizado.
5. Una función llamada `EnviosIguales`, la cual toma dos Envios y determina si son iguales.

### Ejercicio 4.2: Carga y Almacenamiento de Envíos

En relación con la tupla Envío se desea ahora que el programa incorpore la posibilidad de procesar más de un Envío. Resuelva:

1. El encargado de ventas, procesará todos los pedidos hasta que el gerente de compras indique que no hay más envíos que realizar. Este momento es desconocido y variable en cada jornada. ¿Qué tipo de ciclo usarás entonces?
2. Crear una lista de Envios llamada `LstEnvios` donde almacenará todos los envíos cargados del ítem anterior.
3. Cargar los siguientes datos:
  - `envio-1: ( Ciudad Gótica, Rosario, documentos, $130, 0 )`
  - `envio-2: ( Rafaela, Rosario, persona, $370, 1 )`

4. Diseñar una función que recupere de un archivo de texto la siguiente información y la transforme a una tupla Envío y la almacene en la lista LstEnvíos.

Ciudad Gótica, Rafaela, caja, 450, 0

San Francisco, Ciudad Gótica, documentos, 45, 0

San Francisco, Rosario, persona, 500, 1

Ciudad Gótica, Rafaela, caja, 450, 1

Ciudad Gótica, Rosario, caja, 350, 0

5. Diseñar una función que almacene en un archivo de texto todos los datos de Envíos de la lista LstEnvíos.

## Ejercicio 4.3: Respondiendo Preguntas sobre los Envíos

Diseñar funciones que le permitan responder las siguientes preguntas al Comisionista:

1. ¿Cuántas personas debo entregar en una ciudad determinada?
2. ¿Cuánto dinero tengo acumulado con los pedidos que ya entregué?
3. Por un problema en la ruta todos los pedidos con destino a Rafaela serán cancelados de la lista.
4. ¿Cuáles son los envíos de ida y vuelta que tengo mi lista?
5. ¿Cuál es el envío mas costoso de cada categoría de traslado: documento, caja, o persona?

## Ejercicio 4.4: Un Diccionario

Crear un diccionario para guardar la información de los Envíos de una forma mas organizada. Utilizar como clave la ciudad destino.

Una vez diseñado el diccionario:

1. Crear una función que tome la lista de envíos y construya un diccionario tal cual se especificó en el enunciado.
2. Diseñar una función que tome el diccionario anterior y promedie por ciudad los montos por traslados realizados.

## Ejercicio 5: Cálculos

Calcule mediante:

1. un ciclo for.

2. un ciclo while.
3. por recursión.

la siguiente sumatoria:

$$\sum_{k=1}^n (2 \times k - 1) = n^2.$$

Desarrolle una función para cada caso y utilice la relación que está dada en la ecuación anterior, para elaborar los casos de *test* de las funciones.

Guarde su solución en el archivo: **ejercicio5-APELLIDO.py**.

## Ejercicio 6: Carreras (Lenguaje C)

En una competencia de maratón se necesita registrar tres datos de 20 competidores (número de corredor, peso y ubicación). Crear una estructura para almacenar esta información y llamarla Competidor. Luego se pide lo siguiente:

1. Halle y muestre en pantalla los corredores que están en las primeras 5 posiciones. Cree una función para realizar esta tarea.
2. Halle y guarde en un arreglo los corredores que pesan entre 76 y 80 kg. Cree una función para realizar esta tarea.