

## PARADIGMAS DE PROGRAMACION

Primer Semestre 2025

### Guía N° 1

#### OBJETIVOS

La presente guía tiene por propósito explorar problemas de programación, de diversas áreas, que pueden ser resueltos con la materia vista en la primera unidad del curso.

Esta es sólo una pequeña muestra de problemas que pueden encontrarse en pruebas antiguas, libros y otros materiales como, por ejemplo, Internet.

#### COMO PROCEDER

Para cada problema se espera que se realice un estudio (análisis) de lo solicitado, de los medios disponibles para la solución y, luego, se decida por un curso de acción. Casi todos los problemas suponen la escritura —y prueba— de código y, también, de documentación.

El código que se escriba se espera que sea, en su mayoría ANSI C. La razón reside en que se encuentra su influencia, en la sintaxis y semántica, en la mayor parte de los lenguajes imperativos modernos.

#### ENUNCIADOS

1. En un cierto problema se ha llegado a la siguiente función recursiva:

$$f(n) = \begin{cases} n + f(n - 1) & \text{si } n > 0 \\ 0 & \text{si } n = 0 \end{cases}$$

Para esta función se le solicita diseñar un algoritmo recursivo para calcular su valor para un  $n$  arbitrario. A continuación, identifique la función y determine si existe una forma más efectiva para el cálculo de esta función.

2. Escriba un programa para calcular  $n!$ . Empiece con una versión recursiva y luego diseñe otra versión, esta vez no iterativa. Determine cual el mayor factorial obtenible con su solución.

3. Escriba una función recursiva para el cálculo de los números de Fibonacci. Obtenga una versión iterativa y compárelas. ¿Cual es el rango de validez de las soluciones?
4. Investigue algoritmos para realizar los siguientes cálculos estadísticos:
  - media
  - varianza
  - desviación estándar
  - coeficiente de dispersión

A continuación, escriba un programa que lea un conjunto (grande) de números de punto flotante y despliegue cada uno de estos indicadores (calculadora estadística).

5. Dado un entero que representa un año, escriba un algoritmo que determine si un año es o no es bisiesto. El valor del año se ingresará por la consola.
6. Escriba un programa que calcule

$$S = \sum_{n=1}^{n=10.000} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{10.000}$$

primero de izquierda a derecha (modo usual) y luego de derecha a izquierda (modo reverso). Determine si hay diferencias en el valor calculado por ambos modos. Pruebe con números de tipo `float` y `double`.

7. Escriba una función equivalente a la función de biblioteca `sqrt(x)` para calcular el valor de la raíz cuadrada de una variable `x`, utilizando el *algoritmo babilónico*. Deberá probar la función desarrollada construyendo una función `main` que lea un número, calcule la raíz cuadrada usando la función escrita por Ud. y compare el valor obtenido con `sqrt`. Comente los resultados.
8. La función exponencial se expresa como la serie de términos  $e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ . Programe una función que calcule  $e^x$ .
9. Escriba un algoritmo para encontrar el elemento mayor contenido en un arreglo. A partir de este algoritmo derive uno para encontrar el elemento menor.

10. Diseñe un algoritmo que invierta el orden de los caracteres de una cadena de caracteres, sin usar una cadena auxiliar.
11. Dado un texto que contiene exclusivamente los caracteres '(' y ')', escriba un programa que determine si el texto está bien formado. Un texto está bien formado si la cantidad de '(' y ')' es la misma y el texto empieza termina por '(' y termina con ')'. Por ejemplo, '(()())()' es una cadena bien formada, en tanto que '(())' no lo es (lenguaje de paréntesis).
12. Escriba una función que determine si un array de caracteres es un palíndromo, o sea, si se lee igual de izquierda a derecha que de derecha a izquierda. Las palabras *radar* o *anilina*, por ejemplo, son palíndromos.
13. En estadística, es frecuente que se requiera el cálculo de la fórmula 
$$S^2 = \frac{\sum (x_i - M)^2}{n - 1}$$
 donde  $n$  es el tamaño de una muestra,  $x_i$  es cada uno de los elementos de la muestra y  $M = \frac{1}{n} \sum_{i=1}^n x_i$  es el promedio de los valores de la muestra. Construya un programa en C para calcular  $S^2$ . Los datos se leerán desde el teclado, serán números enteros y su cantidad no está determinada. Prepare datos de prueba para comprobar el funcionamiento del programa.
14. Dadas 2 matrices A y B de cada una, escriba un programa que retorne una matriz C conteniendo el producto de A y B.
15. Escriba un programa que cuente el número de caracteres que componen un texto. Luego:
  - Modifique el programa para que, además, cuente el número de líneas del texto. Una línea está terminada por el carácter especial '\n'
  - Modifique el programa anterior para que también cuente el número de palabras del texto. Una palabra se inicia con un carácter distinto de espacio y se termina cuando se encuentra un carácter igual a espacio.
 Su programa debe utilizar la entrada estándar y la salida estándar para obtener y entregar datos. Ref.: Kernighan y Ritchie. **El Lenguaje de Programación C**.
16. Diseñe un algoritmo que conjugue el tiempo presente del modo indicativo de un verbo regular, en castellano.

17. Dado un array con caracteres y terminado por '\0', escriba una función que obtenga el valor numérico equivalente. Por ejemplo, si el array contiene "123\0", entonces se debe entregar un `int` con valor 123. *Ref.: función `atoi` de la biblioteca estándar y el libro de Kernighan y Ritchie.*
18. Estudie la función de manejo de strings llamada `strlen` y escriba su propia versión de ella. Su versión debe ser compatible con `strlen` en cuanto a parámetros y resultados. Para verificar esta parte, escriba un programa que lea strings y compare los resultados de la función propia con los resultados de la función de biblioteca.
19. Repita el ejercicio anterior, ahora con la función `strcat`.
20. Repita el ejercicio anterior, ahora con la función `strcmp`.
21. Construya un algoritmo para calcular e imprimir la distancia entre dos puntos del plano. Cada punto se caracteriza por dos números que representan sus coordenadas  $x$  e  $y$  (`struct`).
22. Construya funciones para implementar aritmética de números complejos.
23. Dado un archivo con datos con la siguiente estructura:

```
struct PERSONA {  
    struct RUT rut;  
    char nombre[256];  
    int telefono;  
};
```

donde la estructura RUT se define como

```
struct RUT {  
    int rutnum;  
    char rutdv;  
};
```

construya un programa que imprima el contenido del archivo en páginas de 50 líneas cada una. Cada página debe llevar un encabezado y un pie de página. El encabezado será el `string` "Universidad San Sebastián" seguido por 2 líneas en blanco y el pie de página será el número de página, centrado horizontalmente y precedido por 2 líneas en blanco. *NOTA:* para crear un archivo de prueba, es

necesario construir un programa que lea datos desde el teclado y los almacene en el formato solicitado.

24. Escriba algoritmos para efectuar una búsqueda secuencial de un elemento dentro de un arreglo de enteros.
25. Diseñe un algoritmo para efectuar una búsqueda binaria de un elemento dentro de un arreglo de enteros. El arreglo está ordenado en forma creciente.
26. Recorra a algún libro de texto y obtenga una versión de un programa para ordenar por el método de la burbuja (`bubblesort`). Programe esta versión y ejecútela con archivos de un tamaño importante (10.000, 20.000 números).

-o-O-o-