

---

# ELEMENTOS DE CÁLCULO NUMÉRICO

Primer Cuatrimestre 2026

---

## Laboratorio N° 1: Introducción a la Programación

**Ejercicio 1 (Inestabilidad numérica de una recurrencia.)** Defina  $I_n = \int_0^1 x^n e^x dx$  para  $n \geq 0$ .

1. Calcule  $I_0$  exactamente.
2. Integrando por partes, demuestre que  $I_n = e - n I_{n-1}$  para  $n \geq 1$ .
3. Programe la recurrencia y calcule  $I_n$  para  $n = 1, 2, \dots, 25$ . ¿Qué observa? ¿Los valores obtenidos son razonables? (Observe que  $0 < I_n < e$  para todo  $n$ , y que  $I_n \rightarrow 0$ .)

**Ejercicio 2 (Año bisiesto.)** En el calendario gregoriano, un año es bisiesto si es divisible por 4, *excepto* los años divisibles por 100, que no son bisiestos, *salvo* que también sean divisibles por 400, en cuyo caso sí lo son.

1. Escriba una función `es_bisiesto(n)` que dado un entero  $n$  devuelva `True` si el año  $n$  es bisiesto y `False` en caso contrario.
2. Verifique que su función da los resultados correctos para los años 1900 (no bisiesto), 2000 (bisiesto), 2024 (bisiesto) y 2025 (no bisiesto).
3. Use su función para contar cuántos años bisiestos hay entre 1 y 2026.

**Ejercicio 3 (Conjetura de Collatz.)** Dado un entero positivo  $n$ , defina la sucesión:

$$a_{k+1} = \begin{cases} a_k/2 & \text{si } a_k \text{ es par,} \\ 3a_k + 1 & \text{si } a_k \text{ es impar,} \end{cases}$$

con  $a_0 = n$ . La conjetura de Collatz afirma que esta sucesión siempre llega al valor 1, independientemente del valor inicial  $n$ .

1. Escriba una función `collatz(n)` que devuelva la lista de valores de la sucesión desde  $a_0 = n$  hasta el primer  $a_k = 1$ .
2. Grafique la sucesión para  $n = 27$ . ¿Cuántos pasos tarda en llegar a 1? ¿Cuál es el valor máximo que alcanza?
3. Verifique la conjetura para todo  $n$  entre 1 y 10000. Para cada  $n$ , registre la cantidad de pasos hasta llegar a 1. Grafique esta cantidad en función de  $n$ .

**Ejercicio 4 (Búsqueda binaria.)** Dada una lista ordenada  $a_0 \leq a_1 \leq \dots \leq a_{n-1}$  y un valor  $v$ , la búsqueda binaria encuentra un índice  $i$  tal que  $a_i = v$  (o determina que  $v$  no está en la lista) comparando  $v$  con el elemento central y descartando la mitad de la lista en cada paso.

1. Escriba una función `busqueda_binaria(lista, v)` que devuelva el índice  $i$  tal que `lista[i] = v`, o  $-1$  si  $v$  no está en la lista.
2. Pruebe su función buscando varios valores en la lista  $[2, 5, 8, 12, 16, 23, 38, 42, 77, 91]$ .