

## ✓ Utilizando R como calculadora

Dado que aún no se configura en local el entorno R con jupyter vscode, se exportará en un ipynb a PDF las soluciones con el código probado en R

### ✓ Pregunta 1

Si hubiéramos empezado a contar segundos a partir de las 12 campanadas que marcan el inicio de 2018, ¿a qué hora de qué día de qué año llegaríamos a los 250 millones de segundos? ¡Cuidado con los años bisiestos!

```
1 # Función para verificar si un año es bisiesto
2 es_bisiesto <- function(ano) {
3   return((ano %% 4 == 0 & ano %% 100 != 0) | (ano %% 400 == 0))
4 }
5
6 # Fecha de inicio
7 inicio <- as.POSIXct("2018-01-01 00:00:00", tz = "UTC")
8
9 # Número total de segundos
10 segundos_totales <- 250000000
11
12 # Calcular la fecha y hora final
13 fecha_hora_final <- inicio + segundos_totales
14
15 # Iterar para encontrar la fecha y hora final teniendo en cuenta los años bisiestos
16 while (segundos_totales > 0) {
17   # Aumentar la fecha y hora en un segundo
18   fecha_hora_final <- fecha_hora_final + 1
19
20   # Verificar si el año actual es bisiesto
21   if (es_bisiesto(as.numeric(format(fecha_hora_final, "%Y")))) {
22     segundos_totales <- segundos_totales - 366
23   } else {
24     segundos_totales <- segundos_totales - 365
25   }
26 }
27
28 # Imprimir la fecha y hora final
29 print(format(fecha_hora_final, "%Y-%m-%d %H:%M:%S"))
```

### ✓ Pregunta 2

Crea una función en R que resuelva una ecuación de primer grado (de la forma  $Ax + B = 0$ ). Es decir, los parámetros deben ser los coeficientes (en orden) y la función tiene que devolver la solución. Por ejemplo, si la ecuación es  $2x + 4 = 0$ , la función tendría que devolver -2. Una vez creada la función, utilízala para resolver las siguientes ecuaciones de primer grado:

- $5x + 3 = 0$
- $7x + 4 = 18$
- $x + 1 = 1$

```

1 # Función para resolver ecuaciones de primer grado Ax + B = 0
2 resolver_ecuacion_primer_grado <- function(A, B) {
3   if (A == 0) {
4     if (B == 0) {
5       return("Infinitas soluciones (identidad)")
6     } else {
7       return("No tiene solución (contradicción)")

```

### ▼ Pregunta 3

Da una expresión para calcular  $3e^{-\pi}$  con R y a continuación, da el resultado obtenido redondeado a 3 cifras decimales.

```

##
1 # Calcular 3e^(-pi)
2 resultado <- 3 * exp(1) * -pi
3
4 # Redondear el resultado a 3 cifras decimales
5 resultado_redondeado <- round(resultado, 3)
6
7 # Imprimir el resultado
8 print(resultado_redondeado)
22 cat("Para la ecuación 7x + 4 = 18, la solución es:", solucion2, "\n")

```

### ▼ Pregunta 4

Da una expresión para calcular el módulo del número complejo  $((2 + 3i)^2) / (5 + 8i)$  y, a continuación, da el resultado obtenido redondeado a 3 cifras decimales.

```

1 # Definir el número complejo
2 numero_complejo <- ((2 + 3i)^2) / (5 + 8i)
3
4 # Calcular el módulo del número complejo
5 modulo_resultado <- Mod(numero_complejo)
6
7 # Redondear el resultado a 3 cifras decimales
8 modulo_resultado_redondeado <- round(modulo_resultado, 3)
9
10 # Imprimir el resultado
11 print(modulo_resultado_redondeado)
12

```