

# RESOLUCIÓN DE EJERCICIOS BÁSICOS DE R. Ejercicio 1

*Enunciado, protocolo a seguir para la elaboración del script y ejemplos de scripts.*

## **Introducción**

Este recurso tiene como objetivo ser una herramienta completa, práctica y comprensible para estudiantes que cursan la asignatura de Fundamentos de Programación. El contenido está diseñado para que los estudiantes, con conocimientos iniciales en algoritmia y programación, puedan programar desde cero un script que resuelva el problema propuesto en clase o en el examen, siempre que sea de un nivel parecido a estos. Incluye descripciones detalladas de procedimientos, ejemplos de scripts y consejos para evitar errores comunes durante el desarrollo de los ejercicios.

Nota: Te recomendamos encarecidamente que leas y entiendas los recursos sobre sumatorios y productorios, vectores y matrices, y otras cosas básicas como bucles *for*, *while* y condiciones *if*. Todos estos recursos los encontrarás en [esta web](#).

## **Procedimiento a seguir para la elaboración del script:**

1. **Leer cuidadosamente el enunciado al completo.** Puede parecer una obviedad, pero muchas veces, con las prisas, no hacemos bien este paso y se nos pasan detalles, como los signos negativos. Esto puede llevarnos a interpretar mal el enunciado, provocando errores tontos o bloqueándonos en algún punto. Si leemos detenidamente hasta el final, generalmente podremos continuar sin problemas. En este paso identificamos todas las variables, que valores toman, cuales de estas variables son vectores o matrices (en estos ejemplos no tenemos matrices), la fórmula que debemos programar, cómo lo haremos y qué resultado exactamente nos pide el ejercicio.
2. **Comenzamos a programar.** Inicia el script con las líneas de código o comandos que limpian la consola y el entorno de trabajo. Esto se hace para evitar confusiones (por nuestra parte) y errores al ejecutar el programa. Este paso debes saltártelo cuando estés programando un subprograma (en estos ejemplos no programaremos subprogramas).
3. **Definir las variables dadas en el enunciado del problema.** Habitualmente tendremos vectores entre estos datos/variables de entrada. A algunas personas les gusta iniciar primero variables de un solo dato y después los vectores, pero tú hazlo como mejor te apañes.
4. **Programar bucles.** Generalmente, después de definir las variables, es el momento de programar los bucles que identificamos en el paso 1. No profundizaremos en este paso dado que ya tenemos recursos que tratan este tema como por ejemplo [este](#).
5. **Programar la fórmula,** si no se incluye en uno de los bucles, para combinar los resultados de los bucles y sacar el resultado pedido.

6. **Mostrar el resultado.** Te recomendamos utilizar la función que aparece en el script para mostrar el resultado. Esta función muestra el resultado en una sola línea realizando un salto de línea. También puedes usar otras como: `print(paste("variable"))`.
7. **Comentar el script.** Esto lo puedes hacer mientras programas en vez de en este punto. Esto sirve para que otro día recuerdes y entiendas lo que habías hecho, y para facilitar el entendimiento del script a otros. Es más útil con programas muy largos y complejos, pero es bueno que te vayas acostumbrando a hacerlo.
8. **Prueba el script ejecutándolo.** Esto también lo puedes ir haciendo mientras programas, para ir comprobando que lo vas haciendo bien y detectar más fácilmente donde está el error en caso de que ocurra. Si tienes un script del profesor de ejemplo para ese ejercicio, comprueba que te sale el mismo resultado.
9. **Guarda el script** como te lo pida el ejercicio.

### **Enunciado:**

En biología, el crecimiento celular en un entorno dinámico puede modelarse considerando factores de crecimiento y aportes externos de nutrientes en cada unidad de tiempo. Un modelo matemático sencillo para describir este fenómeno es:

$$P(t) = P_0 \prod_{i=1}^t (1 + r_i) + \sum_{i=1}^t F_i$$

Descripción del modelo:

- $P(t)$ : Población celular en el tiempo  $t$ .
- $P_0$ : Población inicial de células.
- $r_i$ : Tasa de crecimiento relativa en el intervalo  $i$ , donde  $i = 1, 2, \dots, t$ .
- $F_i$ : Cambio neto en la población celular debido a factores externos como aportes de nutrientes o pérdidas.

Supongamos que una colonia celular inicia con una población de  $P_0 = 500$  células. Durante los siguientes cinco períodos, la tasa de crecimiento y los aportes externos varían de la siguiente manera:

- **Tasa de crecimiento relativa en cada período ( $r$ ):**  $r_1 = 0.01, r_2 = 0.02, r_3 = 0.03, r_4 = 0.01, r_5 = 0.04$
- **Aportes externos en cada período ( $F$ ):**  $F_1 = 10, F_2 = 15, F_3 = -5, F_4 = 20, F_5 = 10$

Calcula la población celular al final del período  $t = 5$ , es decir,  $P(5)$ .

### **Ejemplos de scripts:**

- Propuesto por un miembro del equipo:

```

1  cat("\014") #limpiar la consola
2  rm(list=ls()) #resetear el entorno de trabajo
3
4  #dar valor a las variables de entrada
5  P0<-500 #población inicial de células
6  t<-5 #número de periodos
7
8  #inicializar vectores
9  r<-c(0.01,0.02,0.03,0.01,0.04) #tasa de crecimiento relativa
10 F<-c(10,15,-5,20,10) #cambio neto en la población celular
11
12 prod<-1 #inicializar variable para el productorio
13 for (i in 1:t) { #productorio con for
14   prod=prod*(1+r[i])
15 }
16
17 sum<-0 #inicializar variable para el sumatorio
18 for (i in 1:t) { #sumatorio con for
19   sum=sum+F[i]
20 }
21
22 #cálculo de la población celular al final del periodo
23 Pt=P0*prod+sum
24
25 cat("población celular al final del periodo:", Pt, "\n")
26 #cat(): concatena y muestra texto y valores en la consola en una sola línea
27 #"\n": salta de línea después de imprimir el mensaje

```

- Ejemplos dados por el profesor a los que hemos añadido algunos comentarios:

```

1  # Sin usar prod y sum
2  P0 <- 500 # Población inicial de células.
3  r <- c(0.01, 0.02, 0.03, 0.01, 0.04) # Tasa de crecimiento relativa en cada período.
4  F <- c(10, 15, -5, 20, 10) # Cambios externos en la población celular.
5
6  productorio <- 1 # Inicializa el producto acumulativo.
7  for (i in 1:length(r)) {
8    productorio <- productorio * (1 + r[i]) # Multiplica acumulativamente el término (1 + r[i]).
9  }
10
11 sumatorio <- 0 # Inicializa la suma acumulativa.
12 for (i in 1:length(F)) {
13   sumatorio <- sumatorio + F[i] # Suma los términos de aportes externos.
14 }
15
16 P_t <- P0 * productorio + sumatorio # Calcula la población total al final del período.
17
18 cat("Población total después de 5 días:", P_t, "\n") # Imprime la población total calculada.

```

```

1  # Usando prod y sum
2  P0 <- 500 # Población inicial de células.
3  r <- c(0.01, 0.02, 0.03, 0.01, 0.04) # Tasa de crecimiento relativa en cada período.
4  F <- c(10, 15, -5, 20, 10) # Cambios externos en la población celular.
5
6  productorio <- prod(1 + r) # Calcula el producto acumulativo usando la función `prod`.
7
8  sumatorio <- sum(F) # Calcula la suma acumulativa usando la función `sum`.
9
10 P_t <- P0 * productorio + sumatorio # Calcula la población total al final del período.
11
12 cat("Población total después de 5 días:". P t. "\n") # Imprime la población total calculada.

```

### Resultado:

En todos los casos el resultado es: **607.2929**