



## Programación Declarativa



Ingeniería Informática  
Cuarto curso. Primer cuatrimestre.

Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

Curso académico: 2022 - 2023

---

**Práctica número 4.- Tipos compuestos de datos y funciones con argumentos obligatorios y opcionales**

### VECTORES Y MATRICES

1. Codifica una función que calcule el **módulo** de un vector:
  - Si  $\vec{v} = (x_1, x_2, \dots, x_n)$  entonces su módulo se define como
    - $|\vec{v}| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$
  - **Importante**
    - La función se podrá aplicar a vectores de cualquier longitud.
  - **Ejemplos**
    - $(\text{modulo } \#(1\ 0\ 2)) \rightarrow 5$
    - $(\text{modulo } \#(1\ 0\ 2\ 0\ 3)) \rightarrow 14$
2. Codifica una función iterativa que permita calcular la **media aritmética** de un vector  $\vec{v} = (x_1, x_2, \dots, x_n)$ 
  - **Media aritmética**
$$\frac{x_1 + x_2 + \dots + x_n}{n}$$
  - **Importante**
    - La función se podrá aplicar a vectores de cualquier longitud.
  - **Ejemplo**
    - $(\text{mediaAritmetica } \#(1.\ 2.\ 3.)) \rightarrow 2.0$
    - $(\text{mediaAritmetica } \#(1.\ 2.\ 3.\ 4.\ 5.)) \rightarrow 3.0$
3. Codifica una función, denominada “**extraer-columna**”, que permita extraer la columna de una matriz de dos dimensiones
  - **Ejemplos**
    - 1. 2. 3. 2.
    - 4. 5. 6. 4.
    - 7. 8. 9. 7.
  - $(\text{extraer-columna } \#(\#(1.\ 2.\ 3.\ 2.) \#(4.\ 5.\ 6.\ 4.) \#(7.\ 8.\ 9.\ 7.))\ 0) \rightarrow \#(1.\ 4.\ 7.)$
  - $(\text{extraer-columna } \#(\#(1.\ 2.\ 3.\ 2.) \#(4.\ 5.\ 6.\ 4.) \#(7.\ 8.\ 9.\ 7.))\ 1) \rightarrow \#(2.\ 5.\ 8.)$
  - $(\text{extraer-columna } \#(\#(1.\ 2.\ 3.\ 2.) \#(4.\ 5.\ 6.\ 4.) \#(7.\ 8.\ 9.\ 7.))\ 2) \rightarrow \#(3.\ 6.\ 9.)$
  - $(\text{extraer-columna } \#(\#(1.\ 2.\ 3.\ 2.) \#(4.\ 5.\ 6.\ 4.) \#(7.\ 8.\ 9.\ 7.))\ 3) \rightarrow \#(3.\ 4.\ 7.)$

4. Codifica una función, denominada “*maxiMin*”, que reciba una matriz (no necesariamente cuadrada) y devuelva el máximo de los valores mínimos de las columnas de la matriz.

- **Ejemplo**

- 1. 2. 3. 2.
- 4. 5. 6. 4.
- 7. 8. 9. 7.
- $(\text{maxiMin} \#(\#(1. 2. 3. 2.) \#(4. 5. 6. 4.) \#(7. 8. 9. 7.))) \rightarrow 3.$

5. Determinante de una matriz de dimensión  $3 \times 3$  y área del triángulo.

- a) Codifica una función que permita calcular el **determinante** de una matriz de dimensión  $3 \times 3$ :

- $$\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = x_1y_2z_3 + x_2y_3z_1 + x_3y_1z_2 - x_1y_3z_2 - x_2y_1z_3 - x_3y_2z_1$$

- b) Utiliza la función anterior para calcular el **área de un triángulo** definido por sus vértices:  $P_1=(x_1,y_1)$ ,  $P_2=(x_2,y_2)$  y  $P_3=(x_3,y_3)$

- $\text{área}(P_1, P_2, P_3) = \text{valor\_absoluto} \left( \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \right)$

## LISTAS Y PARÁMETROS OBLIGATORIOS U OPCIONALES

6. Codifica una función, denominada *primosMenores*, que reciba un número natural y genere una lista con los números **primos menores o iguales** que dicho número.

- **Ejemplos**

- $(\text{primosMenores } 9) \rightarrow (2 3 5 7)$
- $(\text{primosMenores } 11) \rightarrow (2 3 5 7 11)$

- **Nota:**

- Utiliza el predicado “*primo?*” codificado en la práctica nº 3.

7. Codifica una función, denominada *descomposiciónEnPrimos*, que reciba un número natural y genere una lista con su **descomposición en números primos**.

- **Ejemplos**

- $(\text{descomposiciónEnPrimos } 2) \rightarrow (2)$
- $(\text{descomposiciónEnPrimos } 12) \rightarrow (2 2 3)$
- $(\text{descomposiciónEnPrimos } 60) \rightarrow (2 2 3 5)$

8. Codifica una función, denominada *filtrarListaPrimos*, que reciba una lista de números naturales y devuelva otra lista compuesta por los números primos.

- **Ejemplos**

- $(\text{filtrarListaPrimos } '() ) \rightarrow ()$
- $(\text{filtrarListaPrimos } '(2 3 4 5 6)) \rightarrow (2 3 5)$
- $(\text{filtrarListaPrimos } '(2 3 4 5 6 7 8 9 10)) \rightarrow (2 3 5 7)$

9. Codifica una función denominada *filtrarPrimos* que reciba un número variable de números naturales y devuelva una lista compuesta por los números primos.

- **Ejemplos**

- *(filtrarPrimos )*  $\rightarrow ()$
- *(filtrarPrimos 2 3 4 5 6)*  $\rightarrow (2 \ 3 \ 5)$
- *(filtrarPrimos 2 3 4 5 6 7 8 9 10 11 12)*  $\rightarrow (2 \ 3 \ 5 \ 7 \ 11)$

10. Codifica una función denominada *filtrarPrimosDelimitados* que reciba dos números “*inicial*” y “*final*” (parámetros obligatorios) y un número variable de números naturales y devuelva una lista compuesta por los números primos delimitados por “*inicial*” y “*final*”.

- **Ejemplos**

- *(filtrarPrimosDelimitados 2 10)*  $\rightarrow ()$
- *(filtrarPrimosDelimitados 2 10 3 4 5 6)*  $\rightarrow (3 \ 5)$
- *(filtrarPrimosDelimitados 2 10 3 4 5 6 7 8 9 10 11 12)*  $\rightarrow (3 \ 5 \ 7)$
- *(filtrarPrimosDelimitados 10 2 3 4 5 6 7 8 9 10 11 12)*  $\rightarrow ()$

11. Codifica una función **recursiva**, denominada *separar*, que reciba como parámetro una lista de números y los reparta en dos listas, dependiendo de que ocupen un “lugar o posición” par o impar.

- **Ejemplos**

- *(separar '())*  $\rightarrow (() \ ())$
- *(separar '(2))*  $\rightarrow ((2) \ ())$
- *(separar '(3 2))*  $\rightarrow ((3) \ (2))$
- *(separar '(1 3 2))*  $\rightarrow ((1 \ 2) \ (3))$
- *(separar '(4 1 3 2))*  $\rightarrow ((4 \ 3) \ (1 \ 2))$
- *(separar '(5 4 1 3 2))*  $\rightarrow ((5 \ 1 \ 2) \ (4 \ 3))$

12. Codifica una función **recursiva**, denominada *unir*, que reciba como parámetros dos listas ordenadas de números y devuelva otra lista con los números ordenados

- **Ejemplos**

- *(unir '() '())*  $\rightarrow ()$
- *(unir '(1) '())*  $\rightarrow (1)$
- *(unir '(1) '(2))*  $\rightarrow (1 \ 2)$
- *(unir '(1 3) '(2))*  $\rightarrow (1 \ 2 \ 3)$
- *(unir '(1 3) '(2 4 5))*  $\rightarrow (1 \ 2 \ 3 \ 4 \ 5)$
- *(unir '(1 3 5) '(2 4))*  $\rightarrow (1 \ 2 \ 3 \ 4 \ 5)$
- *(unir '() '(1 2 3 4 5))*  $\rightarrow (1 \ 2 \ 3 \ 4 \ 5)$

13. Método de ordenación *mergeSort*

- **Descripción**

- Datos de entrada: 5 4 1 3 2
  - División

✓ Primera: 5 1 2 | 4 3

- ✓ Segunda: 5 2 | 1 || 4 | 3
  - ✓ Tercera: 5 | 2 || 1 ||| 4 | 3
  - Fusión:
    - ✓ Primera: 2 5 | 1 || 3 4
    - ✓ Segunda: 1 2 5 | 3 4
    - ✓ Tercera: 1 2 3 4 5
  - Codifica una función que permita ordenar una lista de números utilizando el método *mergeSort*.
    - Ejemplo
      - ✓ *(mergeSort ' (5 4 1 3 2))* → (1 2 3 4 5)
  - Observación
    - Utilizas las funciones “*separar*” y “*unir*” de los ejercicios anteriores.
14. Codifica una función denominada *mergeSortDatos* que permita ordenar una cantidad variable de números utilizando el método *mergeSort*.
- Ejemplo
    - *(mergeSortDatos )* → ()
    - *(mergeSortDatos 2)* → (2)
    - *(mergeSortDatos 1 3 2)* → 1 2 3)
    - *(mergeSortDatos 5 4 1 3 2)* → (1 2 3 4 5)