



# Programación Declarativa

Ingeniería Informática  
Cuarto curso. Primer cuatrimestre



Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

Curso académico: 2024 - 2025

## Práctica número 2. Predicados y sentencias condicionales

- Observación:

- Sólo se han de presentar los ejercicios marcados con un asterisco (\*), que deberán estar contenidos en un mismo fichero.

- IMPORTANTE:

- Todas las funciones deberán tener un comentario de cabecera:
  - Nombre de la función
  - Objetivo
  - Descripción de la solución (salvo que se deduzca de forma inmediata).
  - Significado de los parámetros de entrada.
  - Significado del resultado que devuelve.
  - Funciones auxiliares a las que llama.
- Ejemplos de ejecución de las funciones
  - Después de cada función, se deben poner comentarios con ejemplos de ejecución de dicha función.

1. Indica los valores que resultan al aplicar los predicados primitivos.

```
(boolean? #t)      (boolean? (> 2 3))  (boolean? (+ 2 4))
(number? 3)
(define a 2)
(number? a)
(negative? a)      (positive? a)      (zero? a)
(even? a)          (even? (+ a 1))    (odd? a)      (odd? (+ a 1))
(define (par? x)
  (= 0 (remainder x 2))
)
(procedure? par?)  (procedure? 'par?)
(complex? 3+4i)    (complex? 3)
(real? 3.5)        (real? 3.2+0.0i)    (real? 3+4i)
(rational? 6/10)   (rational? 3)      (rational? 3+4i)
(integer? 2)       (integer? 3.2)     (integer? 3/5)

(define letra1 "w") (define letra2 'w) (define letra3 #\w)
(char? letra1)      (string? letra1)  (char? letra2)
(string? letra2)    (char? letra3)    (string? letra3)
(char? "w")         (string? "w")     (char? #\w)
(string? #\w)
```

2. Comprueba los resultados de los siguientes predicados de equivalencia:

|  |  |  |
|--|--|--|
| <code>(eq? 9/2 9/2)</code>                       | <code>(eqv? 9/2 9/2)</code>                      | <code>(equal? 9/2 9/2)</code>            |
| <code>(define a 9/2)</code>                      | <code>(define b 9/2)</code>                      |  |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b) (= a b)</code>        |
| <code>(define a 3)</code>                        | <code>(define b 3)</code>                        |  |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b) (= a b)</code>        |
| <br><code>(define a 3)</code>                    | <br><code>(define b 3.)</code>                   | <br><code>(define c 3.0)</code>          |
| <code>(eq? a b)</code>                           | <code>(eq? a c)</code>                           | <code>(eq? b c)</code>                   |
| <code>(eqv? a b)</code>                          | <code>(eqv? a c)</code>                          | <code>(eqv? b c)</code>                  |
| <code>(equal? a b)</code>                        | <code>(equal? a c)</code>                        | <code>(equal? b c)</code>                |
| <code>(= a b)</code>                             | <code>(= a c)</code>                             | <code>(= b c)</code>                     |
| <br><code>(define a (+ 3. 2))</code>             | <br><code>(define b (+ 3 2.))</code>             |  |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b) (= a b)</code>        |
| <br><code>(define a "dato")</code>               | <br><code>(define b "dato")</code>               |  |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b)</code>                |
| <br><code>(define a (cons 'a 'b))</code>         |  | <br><code>(define b (cons 'a 'b))</code> |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b)</code>                |
| <br><code>(define a (lambda (x) (* x x)))</code> | <br><code>(define b (lambda (x) (* x x)))</code> |  |
| <code>(eq? a b)</code>                           | <code>(eqv? a b)</code>                          | <code>(equal? a b)</code>                |

3. (\*) Utiliza la forma especial **case** para definir una función que permita indicar el código de las provincias de Andalucía.

- La función recibirá como parámetro el nombre de la provincia y deberá devolver su código:
  - Almería: 04
  - Cádiz: 11
  - Córdoba: 14
  - Granada: 18
  - Huelva: 21
  - Jaén: 23
  - Málaga: 29
  - Sevilla: 41

4. (\*) Utiliza la forma especial **case** para definir una función que permita calcular la letra del DNI o NIF.

- La función recibirá como parámetro el número y deberá devolver la letra que le corresponde.
- Algoritmo
  - Se calcula el resto de la división del número del DNI entre 23 y se devuelve la letra que le corresponde según la siguiente tabla:

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Resto | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| Letra | T | R | W | A | G | M | Y | F | P | D | X  | B  | N  | J  | Z  | S  | Q  | V  | H  | L  | C  | K  | E  |

5. (\*) Codifica una función que reciba las coordenadas de cuatro puntos  $P_1$ ,  $P_2$ ,  $P_3$  y  $P_4$  y calcule el **ángulo** que forman los vectores definidos por dichos puntos:

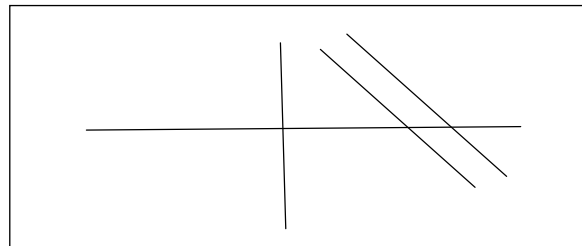
- $\vec{u} = \overrightarrow{P_1P_2} = (x_2 - x_1, y_2 - y_1)$
- $\vec{v} = \overrightarrow{P_3P_4} = (x_4 - x_3, y_4 - y_3)$
- Si  $\vec{u} = (u_1, u_2)$  y  $\vec{v} = (v_1, v_2)$  entonces el ángulo  $\alpha$  se puede calcular como:

$$\alpha = \arccos\left(\frac{u_1 \times v_1 + u_2 \times v_2}{\sqrt{u_1^2 + u_2^2} \times \sqrt{v_1^2 + v_2^2}}\right)$$

- **Observaciones**
  - El resultado se devuelve en **radianes**.
  - Si los vectores son nulos entonces el ángulo es 0.0.
  - Téngase en cuenta la **precisión** de los números reales al calcular los ángulos. Utilícese una cota de error: 1e-6.

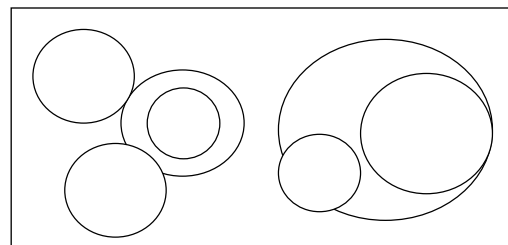
6. (\*) Codifica una función, denominada **posiciónRelativaRectas**, que determine la posición relativa de dos rectas:

- Iguales.
- Paralelas.
- Secantes.
- Perpendiculares.



- La función recibirá como parámetros los coeficientes de las dos rectas
  - $r_1: a_1X + b_1Y + c_1 = 0$
  - $r_2: a_2X + b_2Y + c_2 = 0$

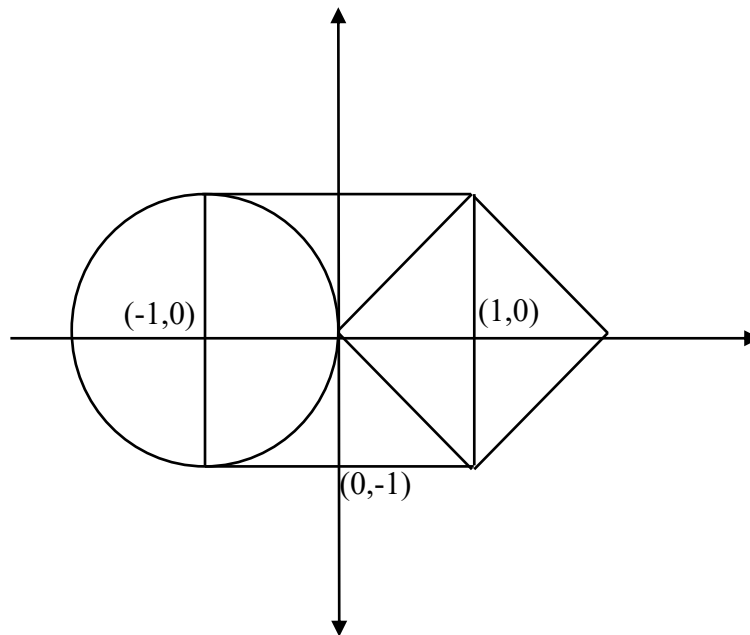
7. (\*) Codifica una función que determine la **posición relativa de dos circunferencias**.



- La función recibirá como parámetros las coordenadas de los centros y los radios de las circunferencias y devolverá los siguientes valores:

- Iguales: 1
- Concéntricas: 2
- Tangentes por dentro: 3
- Tangentes por fuera: 4
- Interiores: 5
- Exteriores: 6
- Secantes por dentro: 7
- Secantes por fuera: 8

8. (\*) Dada las siguientes figuras geométricas

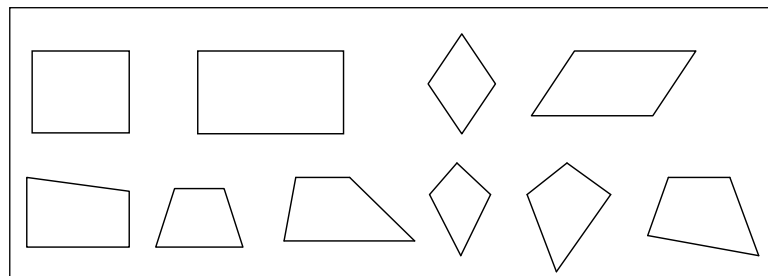


- Codifica una función que asigne a un punto  $P = (x, y)$  el valor que le corresponde según su posición:
  1. El punto pertenece a la circunferencia o a uno de los lados del cuadrado o del rombo.
  2. El punto está dentro del círculo y fuera del cuadrado.
  3. El punto está dentro del círculo y del cuadrado.
  4. El punto está dentro del cuadrado y fuera del círculo y del rombo.
  5. El punto está dentro del cuadrado y del rombo.
  6. El punto está dentro del rombo y fuera del cuadrado.
  7. En otro caso.
- **Observación**
  - Se deben utilizar las funciones auxiliares que calculan las distancias euclidiana, de Manhattan y de ajedrez (véase la práctica 1).

9. (\*) Clasificación de triángulos según sus ángulos

- Codifica una función, denominada `tipoTrianguloPorAngulos`, que reciba las coordenadas de los vértices de un triángulo y devuelva el valor que le corresponde según su tipo:

- Recto: tiene un ángulo de  $90^\circ$ , es decir,  $\pi/2$  radianes.
  - Acutángulo: todos sus ángulos son menores que  $90^\circ$ .
  - Obtusángulo: tiene un ángulo mayor de  $90^\circ$  y menor de  $180^\circ$ .
  - Nulo: los tres vértices están alineados.
  - **Observación**
    - Téngase en cuenta la **precisión** de los números reales al calcular los ángulos.
    - Utilícese una cota de error:  $1e-6$ .
10. (\*) Codifica los siguientes predicados que reciben las coordenadas de cuatro puntos del plano  $P_1, P_2, P_3$  y  $P_4$ :
- **ladosParalelos?**
    - Determina si la recta definida por los dos primeros puntos es **paralela** a la recta definida por los otros dos puntos.
  - **perpendiculares?**
    - Determina si la recta definida por los dos primeros es **perpendicular** a la recta definida por los otros dos puntos.
11. (\*) Codifica una función que calcule el **área de un trapecio**:
- La función ha de recibir como parámetros las coordenadas de los vértices, pero sin especificar su orden.
  - La función deberá determinar **previamente** los lados que forman las bases utilizando el predicado “**ladosParalelos?**”
12. (\*) Codifica una función que calcule el **área de un rombo**:
- La función ha de recibir como parámetros las coordenadas de los vértices del rombo, pero sin especificar su orden.
  - La función deberá usar el predicado “**perpendiculares?**” para determinar previamente los vértices que forman las diagonales del rombo.
13. (\*) **Clasificación de cuadriláteros convexos**
- Codifica una función que reciba las coordenadas de cuatro puntos del plano y determine qué tipo de **cuadrilátero convexo** forman:



- **Paralelogramos:** lados paralelos dos a dos
  1. **Cuadrado:** lados iguales y ángulos rectos.
  2. **Rectángulo:** lados desiguales y ángulos rectos.
  3. **Rombo:** los lados son todos iguales y los ángulos opuestos son iguales.

- 4. **Romboide:** “lados opuestos” y “ángulos opuestos” iguales.
  - **Trapecios:** solamente tiene un par de lados paralelos
  - 5. **Trapecio rectangular:** dos lados paralelos y un ángulo recto
  - 6. **Trapecio isósceles:** dos lados paralelos y otros dos lados iguales
  - 7. **Trapecio escaleno:** dos lados paralelos y otros dos lados desiguales
  - **Cometas y trapezoides:** no tiene lados paralelos
  - 8. **Cometa:** lados contiguos iguales dos a dos.
  - 9. **Cometa oblicuo:** tiene un solo par de lados contiguos iguales
  - 10. **Trapezoide:** todos los lados son distintos.
- **Observación**
  - Téngase en cuenta la **precisión** de los números reales al calcular los ángulos y los lados. Utilícese una cota de error:  $1e-6$ .