

Tema 08 - Programacion Orientada a Objetos (Soluciones)

Nota: Estos ejercicios son optativos para hacer al final de la unidad y están pensados para apoyar tu aprendizaje.

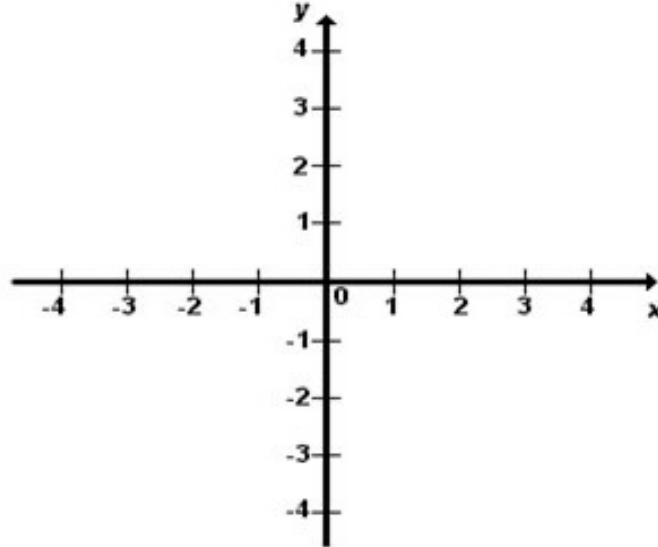
En este ejercicio vas a trabajar el concepto de puntos, coordenadas y vectores sobre el plano cartesiano y cómo la programación Orientada a Objetos puede ser una excelente aliada para trabajar con ellos. No está pensado para que hagas ningún tipo de cálculo sino para que practiques la automatización de tareas.

Nota: Creo que es un ejemplo muy interesante, punto de partida en la programación de gráficos, pero si consideras que esto no lo tuyo puedes simplemente pasar de largo. Ahora bien, debes ser consciente de que te vas a perder uno de los ejercicios más interesantes del curso.

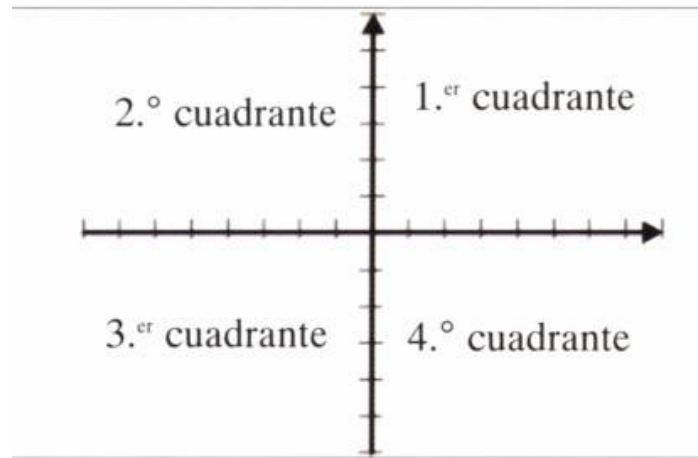
Antes de continuar voy a explicar brevemente los conceptos básicos por si alguien necesita un repaso.

El plano cartesiano

Representa un espacio bidimensional (en 2 dimensiones), formado por dos rectas perpendiculares, una horizontal y otra vertical que se cortan en un punto. La recta horizontal se denomina eje de las abscisas o **eje X**, mientras que la vertical recibe el nombre de eje de las ordenadas o simplemente **eje Y**. En cuanto al punto donde se cortan, se conoce como el **punto de origen O**.



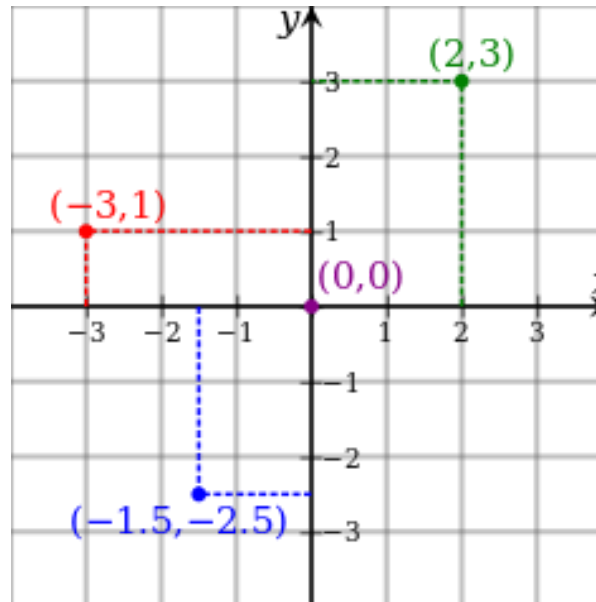
Es importante remarcar que el plano se divide en 4 cuadrantes:



Puntos y coordenadas

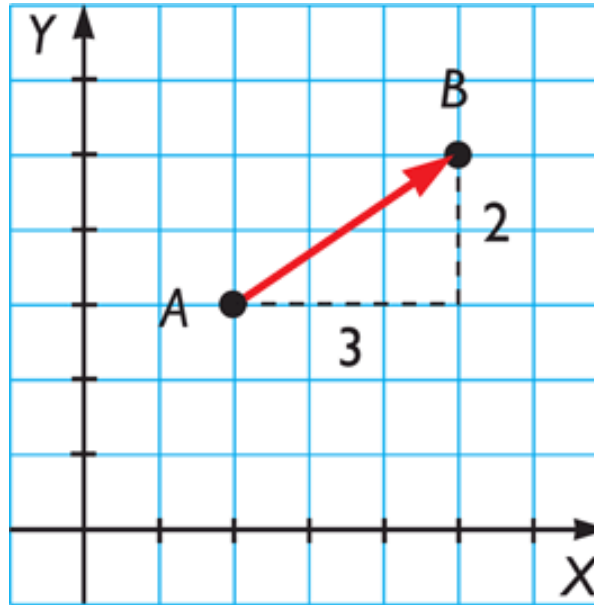
El objetivo de todo esto es describir la posición de **puntos** sobre el plano en forma de **coordenadas**, que se forman asociando el valor del eje de las X (horizontal) con el valor del eje Y (vertical).

La representación de un punto es sencilla: **P(X,Y)** dónde X y la Y son la distancia horizontal (izquierda o derecha) y vertical (arriba o abajo) respectivamente, utilizando como referencia el punto de origen (0,0), justo en el centro del plano.



Finalmente, un vector en el plano hace referencia a un segmento orientado, generado a partir de dos puntos distintos.

A efectos prácticos no deja de ser una línea formada desde un punto inicial en dirección a otro punto final, por lo que se entiende que un vector tiene longitud y dirección/sentido.



En esta figura, podemos observar dos puntos A y B que podríamos definir de la siguiente forma:

- $A(x_1, y_1) \Rightarrow A(2, 3)$
- $B(x_2, y_2) \Rightarrow B(5, 5)$

Y el vector se representaría como la diferencia entre las coordenadas del segundo punto respecto al primero (el segundo menos el primero):

- $AB = (x_2 - x_1, y_2 - y_1) \Rightarrow (5 - 2, 5 - 3) \Rightarrow (3, 2)$

El ejercicio

que se definió no deja de ser: 3 a la derecha y 2 arriba.

Y con esto finalizamos este mini repaso.

Preparación

- Crea una clase llamada **Punto** con sus dos coordenadas X e Y.
- Añade un método **constructor** para crear puntos fácilmente. Si no se reciben una coordenada, su valor será cero.
- Sobreescribe el método **string**, para que al imprimir por pantalla un punto aparezca en formato (X,Y)
- Añade un método llamado **cuadrante** que indique a qué cuadrante pertenece el punto, o si es el origen.
- Añade un método llamado **vector**, que tome otro punto y calcule el vector resultante entre los dos puntos.
- (Opcativo) Añade un método llamado **distancia**, que tome otro punto y calcule la distancia entre los dos puntos y la

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Nota: La función raíz cuadrada en Python `sqrt()` se debe importar del módulo `math` y utilizarla de la siguiente forma:

```
import math
math.sqrt(9)
> 3.0
```

- Crea una clase llamada **Rectángulo** con dos puntos (inicial y final) que formarán la diagonal del rectángulo.
- Añade un método **constructor** para crear ambos puntos fácilmente, si no se envían se crearán dos puntos en el origen por defecto.
- Añade al rectángulo un método llamado **base** que muestre la base.
- Añade al rectángulo un método llamado **altura** que muestre la altura.
- Añade al rectángulo un método llamado **area** que muestre el area.

*Puedes identificar fácilmente estos valores si intentas dibujar el cuadrado a partir de su diagonal. Si andas perdido, prueba de dibujarlo en un papel, ¡seguro que lo verás mucho más claro! Además recuerda que puedes utilizar la función **abs()** para saber el valor absolute de un número.*

Experimentación

- Crea los puntos A(2, 3), B(5,5), C(-3, -1) y D(0,0) e imprimelos por pantalla.
- Consulta a que cuadrante pertenecen el punto A, C y D.
- Consulta los vectores AB y BA.
- (Optativo) Consulta la distancia entre los puntos 'A y B' y 'B y A'.
- (Optativo) Determina cual de los 3 puntos A, B o C, se encuentra más lejos del origen, punto (0,0).
- Crea un rectángulo utilizando los puntos A y B.
- Consulta la base, altura y área del rectángulo.

```
In [1]: import math

class Punto:

    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def __str__(self):
        return "({}, {})".format(self.x, self.y)
```

```

        if self.x > 0 and self.y > 0:
            print("{} pertenece al primer cuadrante".format(self))
        elif self.x < 0 and self.y > 0:
            print("{} pertenece al segundo cuadrante".format(self))
        elif self.x < 0 and self.y < 0:
            print("{} pertenece al tercer cuadrante".format(self))
        elif self.x > 0 and self.y < 0:
            print("{} pertenece al cuarto cuadrante".format(self))
        else:
            print("{} se encuentra sobre el origen".format(self))

    def vector(self, p):
        print("El vector entre {} y {} es ( {}, {} )".format(self, p, p.x - self.x, p.y - self.y))

    def distancia(self, p):
        d = math.sqrt( (p.x - self.x)**2 + (p.y - self.y)**2 )
        print("La distancia entre los puntos {} y {} es {}".format(self, p, d))
)

```

class Rectangulo:

```

    def __init__(self, pInicial=Punto(), pFinal=Punto()):
        self.pInicial = pInicial
        self.pFinal = pFinal

    def base(self):
        self.v_base = abs(self.pFinal.x - self.pInicial.x)
        print("La base del rectángulo es {}".format( self.v_base ) )

    def altura(self):
        self.v_altura = abs(self.pFinal.y - self.pInicial.y)
        print("La altura del rectángulo es {}".format( self.v_altura ) )

    def area(self):
        self.v_base = abs(self.pFinal.x - self.pInicial.x)
        self.v_altura = abs(self.pFinal.y - self.pInicial.y)
        self.v_area = self.v_base * self.v_altura
        print("El área del rectángulo es {}".format( self.v_area ) )

```

```

A = Punto(2,3)
B = Punto(5,5)
C = Punto(-3, -1)

```

A.cuadrante()
C.cuadrante()
D.cuadrante()

A.vector(B)
B.vector(A)

A.distancia(B)
B.distancia(A)

A.distancia(D)
B.distancia(D)
C.distancia(D)

R = Rectangulo(A, B)
R.base()
R.altura()

~~R2,3()~~ pertenece al primer cuadrante

(-3, -1) pertenece al tercer cuadrante

(0, 0) se encuentra sobre el origen

El vector entre (2, 3) y (5, 5) es (3, 2)

El vector entre (5, 5) y (2, 3) es (-3, -2)

La distancia entre los puntos (2, 3) y (5, 5) es 3.605551275463989

La distancia entre los puntos (5, 5) y (2, 3) es 3.605551275463989

La distancia entre los puntos (2, 3) y (0, 0) es 3.605551275463989

La distancia entre los puntos (5, 5) y (0, 0) es 7.0710678118654755

La distancia entre los puntos (-3, -1) y (0, 0) es 3.1622776601683795

La base del rectángulo es 3

La altura del rectángulo es 2

El área del rectángulo es 6