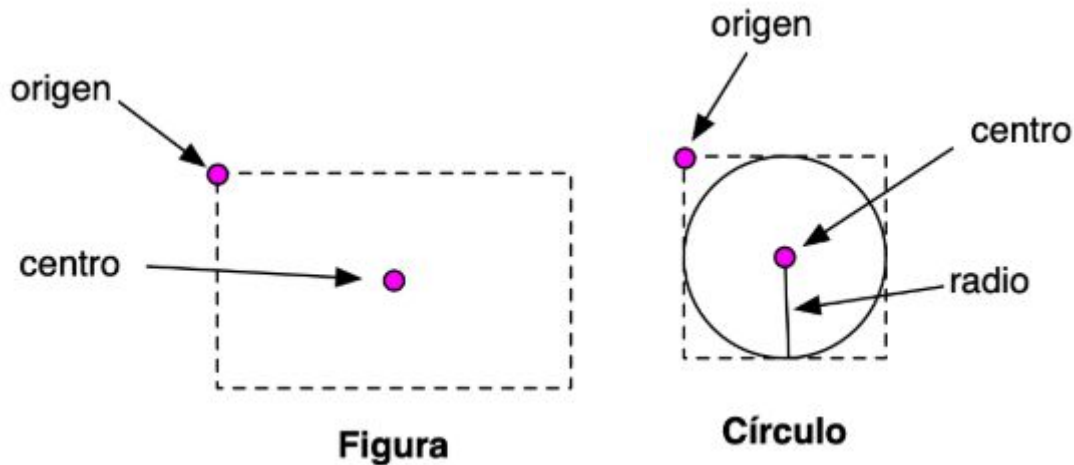


POO con Swift

Ejercicio: Define las siguientes clases y estructuras:



- **Estructuras Punto y Tamaño**, con dos variables: x e y, y alto y ancho respectivamente.
- **Clase padre figura**, con:
 - Constructor:
Figura(origen: Punto, tamaño: Tamaño)
 - Propiedades de instancia almacenadas:
 - origen** (Punto) que define las coordenadas de la esquina superior izquierda del rectángulo que define la figura
 - tamaño** (Tamaño) que define el alto y el ancho del rectángulo que define la figura.
 - Propiedades de instancia calculadas:
 - area (Double, solo lectura)** que devuelve el área del rectángulo que engloba la figura.
 - centro (Punto, propiedad de lectura y escritura)**. Es el centro del rectángulo que define la figura. Si modificamos el centro se modifica la posición del origen de la figura.
- **Clase derivada círculo**, con:
 - Constructor:
Círculo(centro: Punto, radio: Double). Al crear el círculo deberemos actualizar las propiedades origen y tamaño de la figura.
 - Propiedades de instancia almacenadas:
 - radio** (Double) que contiene la longitud del radio.
 - Propiedades de instancia calculadas:
 - centro** (Punto, de lectura y escritura), heredada de la clase padre.
 - area** (Double, de lectura y escritura) que devuelve el área del círculo. El setter modifica el tamaño del círculo (su radio), manteniendo el centro en la misma posición.

Solución:

```
struct Punto {  
    var x = 0.0, y = 0.0  
}  
  
struct Tamaño {  
    var ancho = 0.0, alto = 0.0  
}  
  
class Figura {  
    var origen: Punto  
    var tamaño: Tamaño  
  
    var area: Double {  
        return tamaño.ancho * tamaño.alto  
    }  
  
    var centro: Punto {  
        get {  
            let centroX = origen.x + (tamaño.ancho / 2)  
            let centroY = origen.y + (tamaño.alto / 2)  
            return Punto(x: centroX, y: centroY)  
        }  
        set {  
            origen.x = newValue.x - (tamaño.ancho / 2)  
            origen.y = newValue.y - (tamaño.alto / 2)  
        }  
    }  
  
    init(origen: Punto, tamaño: Tamaño) {  
        self.origen = origen  
        self.tamaño = tamaño  
    }  
}
```

```
class Circulo: Figura {
    var radio: Double {
        didSet {
            let alto = radio * 2
            let ancho = radio * 2
            tamaño = Tamaño(ancho: ancho, alto: alto)
        }
    }
    override var area: Double {
        get {
            Double.pi * radio * radio
        }
        set {
            let radioCuadrado = newValue / Double.pi
            radio = radioCuadrado.squareRoot()
        }
    }
}

init(centro: Punto, radio: Double) {
    self.radio = radio
    let minX = centro.x - radio
    let minY = centro.y - radio
    let alto = radio * 2
    let ancho = radio * 2
    super.init(origen: Punto(x: minX, y: minY), tamaño: Tamaño(ancho: ancho, alto: alto))
}
```

Bibliografía

Los enunciados de los ejercicios resueltos, y los resúmenes, se han elaborado a partir del material publicado en <https://domingogallardo.github.io/> , material del que es propietario el Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante, Domingo Gallardo, Cristina Pomares, Antonio Botía y Francisco Martínez.