Nombre y Apellidos:

S	O	L	U	C	I	Ó	N														

Considere la siguiente definición, que consiste en una declaración que permite crear figuras geométricas a las que interesa obtener su área.

```
Class Figura:
    INIT_CLASS(){}
    DOUBLE area(){}
```

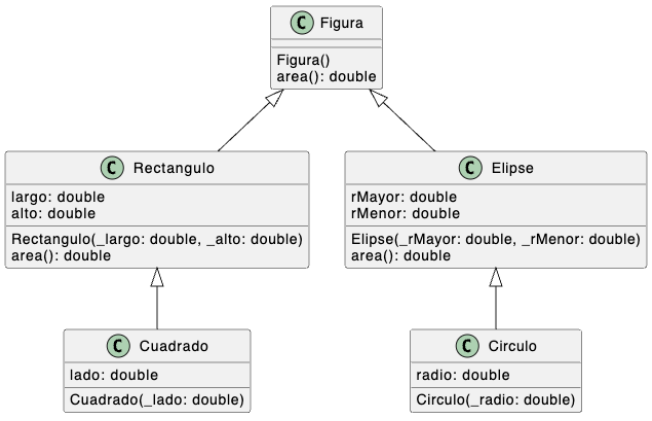
Esta clase se utiliza como base para crear figuras geométricas clásicas, como *rectángulos* y *elipses*. Además, se quieren crear casos específicos de los dos figuras anteriores, como *cuadrados* y *círculos*. La medida de interés de cada figura que es creada es su área.

1) A partir de la clase `Figura`, diseñe una jerarquía eficiente de herencia para las clases `Rectangulo`, `Elipse`, `Cuadrado` y `Circulo`, tal que permita tener un alto nivel de cohesividad y bajo nivel de acoplamiento. Cada clase nueva debe tener declarada sus atributos y métodos. Asuma que no se utilizan constructures vacíos, sólo constructures con parámetros. La interfaz de cada clase no implementa métodos para modificar sus atributos respectivo. (4pts)

a) Una jerarquía eficiente significa que debe crear una herencia que permite crear clases con un alto grado de cohesividad y bajo nivel de acoplamiento.

b) Si bien rectángulo, elipse, cuadrado y círculo son figuras geométricas y aparentemente todas con subclases de Figura, esto no es del todo correcto. Según la definición que se da en el enunciado, rectángulos y elipses son las figuras derivadas de figura. Cuando se menciona que cuadrado y círculo son casos específicos, esto se traduce a que son clases derivadas.

c) En base a lo anterior, la jerarquía que establecida como:



Si bien no es incorrecto agregar los atributos lado y radio en las clases `Cuadrado` y `Circulo` respectivamente, se pueden omitir. Esto es debido a que, por las

características de *la interfaz de las clase, no se deben implementar métodos para modificar sus atributos respectivo*.

2) Implemente en pseudo-código la clase `Rectangulo` (3pts)

```
Class Rectangulo(Figura):
    double largo, alto

    INIT_CLASS(double _largo, double _alto):
        largo = _largo
        alto  = _alto

    double area():
        return largo*alto
```