

Cumple con los 4 pilares de la POO porque respeta el encapsulamiento, la abstracción, hay herencia (simple) y polimorfismo.

- Entendiendo la abstracción como el proceso de definición de los métodos y atributos de cada clase, al indicar las clases: cuadrado y rectángulo, se indican también los datos necesarios para definirlo (medidas de los lados) y a su vez se indican dos métodos que pueden aplicarse a ambas clases (cálculo del perímetro y el área).
- Encapsulamiento se presenta porque en la parte de prueba el sujeto solo puede acceder a las clases, sin alterar el contenido de cada una de ellas. Al usar las distintas clases, será necesario que pasen los parámetros solicitados y eso no se puede modificar. Si no estuviesen encapsulados se dejan visibles tanto los atributos como los métodos por lo que no se está protegiendo la información de manipulaciones no autorizadas.
- Herencia, en este ejemplo las clases hijo son cuadrado y rectángulo que heredan atributos y métodos de la clase padre que es Figura. Como no hay definido ninguno de ellos para la clase Figura, por lo tanto no reciben estos parámetros.
- Polimorfismo: implica dar la misma orden a varios objetos para que respondan de diferente manera, así es que la orden es calcular el perímetro pero dependiendo si el objeto es un rectángulo o es un cuadrado, responderá de distinta manera. De igual forma ocurre con el cálculo del área.

3) Investigar:

- Atributos públicos vs atributos privados
- ¿Aplica este mismo criterio para los métodos?
- Utilizando la función `dir` aplicada a un objeto: ¿Qué devuelve? ¿Puede llamar a un atributo o método privado?
- ¿Cómo adaptaría el código fuente implementado para atender a este nuevo criterio? Impacte el cambio en su repositorio.

En Python, por defecto, todos los atributos y métodos son públicos. ¿Qué implica esto? Que se puede acceder y modificar desde otra parte del código, lo cual generaría fallas de seguridad y perdería posiblemente la esencia de la clase. Al generar atributos y métodos privados lo que se logra es encapsular a éstos a la clase en la cual fueron creados y una clase que extiende otra clase con un método privado no heredará esos métodos y provocará un error si intenta acceder a ellos.

Para declarar tanto atributos como métodos privados, en Python se colocan como prefijos dos guiones bajos.

La función `dir(objeto)` devuelve la lista de los atributos del objeto. En el caso que algún atributo tenga `__` es decir, este planteado como privado, al imprimir la lista, se mostrará escrito antepuesto por la clase.