

1. En la clase pájaro debemos definir el método ruido(), para que este oculte el de la clase padre.
2. La clase Pez heredar  el constructor de la clase Animal, el cual espera los par metros nombre, edad y raza. Adem s, heredar  los m todos set y get correspondientes a los atributos privados nombre y raza. Tambi n recibir  otros m todos como caminar, correr, ruido y getTotalCreados. A su vez, heredar  los m todos set y get asociados al atributo edad de la clase SerVivo.
3. Para evitar la duplicidad de los atributos nombre y edad, se debe eliminar la privacidad de estos atributos nombre y edad tanto en la clase SerVivo como en la clase Animal. De este modo, ambos atributos har an referencia al mismo espacio de memoria en las dos clases.
4. Para evitar que los m todos sobrescriban el comportamiento de la clase padre, se puede utilizar la funci n super() dentro de los m todos getNombre y setNombre. Esto permite llamar a los m todos correspondientes de la clase base SerVivo.
5. Si se respeta la privacidad del atributo, este no se hereda. Sin embargo, si se omite esa privacidad y el atributo totalCreados se define nuevamente en una subclase, lo que ocurre es una ocultaci n del atributo original. Esto sucede porque el nuevo atributo queda asociado a la subclase y no a la clase padre.
6. S , se sobrescribe, porque en Python, si un m todo de la clase hija tiene el mismo nombre que uno en la clase padre, se considera una sobrescritura.
7. Una clase que hereda de Persona obtiene los m todos adue arAnimal, getTotalCreados y los m todos set y get correspondientes a los atributos privados nombre y edad.
8. El m todo adue arAnimal acepta cualquier objeto que sea de tipo Animal o sus subclases. Por lo tanto, no se le puede pasar directamente un objeto de tipo SerVivo si no pertenece a esa jerarqu a.
9. Para obtener esta cantidad, se debe agregar la siguiente l nea en el constructor de la clase Animal: super().__init__(nombre, edad), De esta forma, el inicializador de la clase SerVivo contar  todos los objetos creados. Tambi n se puede consultar directamente el atributo totalCreados.
10. El m todo getRaza en la clase Perro sobrescribe al m todo con el mismo nombre en la clase Animal. Esto sucede porque ambos m todos tienen la misma firma (nombre y par metros), por lo que el m todo de la clase hija reemplaza al de la clase padre.