

# Taller Python 5

By: Andrés Felipe Muñoz Ortiz

Documento: 1011395924

Solución:

1. Como el metodo ruido es un atributo de instancia que esta siendo heredado por la clase animal a pajar, entonces simplemente basta con sobrescribirla dentro de la clase pájaro, por ejemplo: `def ruido(self): {print"Cantar y silvar"}` dentro de la clase pájaro
2. Si creamos la clase pez como una subclase de Animal recibirá todos sus métodos y atributos no privados, atributo seria TotalCreados y métodos tales como `setRaza`, `getRaza`, `setNombre`, `getNombre`, `caminar`, `correr`, `ruido` y `getTotalCreados`, en cuanto al constructor, este no es heredado, y como Animal no tiene uno por defecto entonces pez tampoco, por tanto debe crear un nuevo constructor para que la clase pueda funcionar bien
3. La clase animal los esta sobrescribiendo, no llamando al método `__init__` de `SerVivo` en su propio `__init__` y por esto no son el mismo, lo que debe hacer es llamar al `__init__` de `SerVivo` en el propio `__init__` de Animal para así inicializar los mismos atributos, quedaría: 

```
def __init__(self, nombre, edad, raza):  
    super().__init__(nombre, edad)  
    self._raza = raza  
    Animal._totalCreados += 1
```
4. Pues, sí se eliminan estos métodos de la clase Animal entonces los métodos que heredan no serían sobrescritos y podrían ser usados sin ocultar algún valor de la clase padre, o usando `super()` sería simplemente llamar a los métodos respectivos de la clase padre dentro de los métodos de la clase hija, como:

```
def setNombre(self, nombre):  
    super().setNombre(nombre)  
def getNombre(self):  
    super().getNombre()
```
5. Si se hereda a las clases hijas, pero como se vuelve a definir en estas entonces se sobrescribe
6. Sí, pues python si permite sobrescribir métodos estáticos, entonces este sería nuevamente establecido reemplazando al método del papa
7. Sí la clase hija no sobrescribe el `init` este se heredaría, además `aduenarAnimal` y `getTotalCreados` también son métodos que se heredarían, pues Python hereda incluso métodos de clase y estáticos
8. Sí, cualquier tipo de atributo se puede pasar al método, incluso de tipo `serVivo` ya que en Python no existen problemas de dar métodos de tipos superiores

9. Pues, como el atributo totalCreados y el método gettotalCreados se redefine en cada clase entonces no es compartido y no contaría el total de objetos creados, lo único sería sumar todos o eliminar el atributo y el método de cada subclase para que así la compartan todas las clases y cuente cuantos objetos se han creado en total
10. El método se sobrescribe, pues Python no tiene sobrecarga de métodos y por ser el mismo nombre esta se reemplaza