

Respuestas

1. Se debe redefinir el método desde la clase pájaro, escribiendo un método con el mismo nombre.
2. En caso de que la clase `Pez` herede de `Animal`, tendrá los atributos y métodos de `Animal` y de `Ser Vivo`, y su inicializador será el de la clase padre(`Animal`) que recibe como argumentos, nombre, edad, raza. Los métodos de `Pez` serán: `getRaza`, `setRaza`, `getNombre`, `setNombre`, `caminar`, `correr`, `ruido`, `getTotalCreados`, `setEdad`, `getEdad`. Sus atributos serán: nombre, edad, raza
3. Como en Python no se realiza una separación de espacios de memoria como en JAVA, la redefinición de los atributos en `Animal` ocupan el mismo espacio que los de la clase `SerVivo`
4. No es necesario redefinir esos métodos en la clase `Animal`, ya que directamente los hereda de `SerVivo`, por lo tanto se pueden eliminar tranquilamente el `get` y `set` nombre de la clase `Animal`, sin tener que usar `super()`
5. Sí el atributo se hereda, y al redefinirlo lo oculta,
6. Sí, se están sobrescribiendo en cada clase hija.
7. Hereda los métodos; `getNombre`, `setNombre`, `getEdad`, `setEdad`, `getTotalCreados`, `aduenarAnimal` y el inicializador(que cuenta como método) de `Persona`.
8. Se le puede pasar cualquier tipo, porque Python así lo permite, lo ideal es pasarle uno de clase `Animal` o sus subclases.
9. Agregando en `Animal` el `super().__init__(nombre, edad)` en su inicializador.
10. Se sobrescribe, la sobrecarga como tal no existe en Python, y si se quisiera sobrecargar, necesitamos importar una librería especial y escribir el método en la clase que ya lo tenga definido, no en su clase hija.