

1. Se debe sobrecargar el método ruido en la clase pájaro. Para ello se agrega:

```
def ruido(self):
```

```
    return "cantar y silbar"
```

2. Si se crea la clase pez (suponiendo que herede de la clase Animal), contaría con el constructor de la clase Animal, ya que es su padre, y por lo tanto recibiría nombre, edad y raza. Al mismo tiempo recibe los métodos definidos en la clase Animal: caminar, correr, ruido.

3. Los atributos nombre y edad se esta manejando desde dos lugares distintos y de manera independiente. Para hacer que los atributos sean los mismos se debe de agregar un super() en el constructor de la clase Animal. Quedando de la siguiente manera:

```
def __init__(self, nombre, edad, raza):
```

```
    super().__init__(nombre, edad)
```

```
    self._raza = raza
```

```
    Animal._totalCreados += 1
```

4. Usar la función super() para llamar los métodos correspondientes permite el comportamiento deseado. Los métodos quedarían de la siguiente manera:

```
def setNombre(self, nombre):
```

```
    Super().setNombre(nombre)
```

Y

```
def getNombre(self):
```

```
    return super().getNombre()
```

5. Si, esta siendo heredado, pero al redefinir el atributo de una clase hija, de cierto modo se esta ocultando el atributo heredado. Sin embargo, el atributo de la clase padre sigue estando disponible, pero al ser redefinido el atributo de la clase hija tiene prioridad.

6. Si, se está sobrescribiendo el método y estos usan su propio totalCreados por asi decirlo, ya que esta llamando su propio método, y no se hace uso de la función super().

7. Una clase hija de la clase persona, heredaría los métodos de la clase persona y los métodos que haya heredado la clase persona de la clase serVivo.

8. el método recibe objetos que tengan definido el metodo llamado ruido(), en nuestro contexto esto son objetos de tipo Animal, de cierta manera el método está recibiendo objetos de tipo serVivo, ya que los objetos de tipo Animal también son de tipo serVivo.

9. Llamando el método de getTotalCreados desde serVivos.

10. El método se esta sobrescribiendo y este está siendo “remplazando” desde las instancias de sus clases hijas.