**¿Al realizar el ejercicio, cuantos objetos se han creado, 1, 2 o 3 o más?**

En el ejercicio, solo se ha creado un objeto de tipo SerVivo, pero este objeto ha sido referenciado por la variable Juana para comportarse como tres tipos diferentes (Humano, Perro, y Puerco) en distintos momentos. Aunque la referencia cambia a lo largo del programa, solo se está utilizando una única variable de referencia y no se crean nuevos objetos para cada clase.

En resumen, solo se crea un objeto a la vez, pero es reutilizado.

**¿Qué es más conveniente usar, "polimorfismo real" o "polimorfismo con Overriding"?**

Polimorfismo real (a través de UpCasting y DownCasting): Este enfoque permite que un objeto de una clase hija sea tratado como una instancia de su clase padre. Es más flexible cuando trabajas con colecciones de objetos heterogéneos (como listas de distintos tipos de objetos) y te permite mantener una mayor reutilización del código, ya que puedes tratar diferentes tipos de objetos de manera uniforme.

Polimorfismo con Overriding: Aunque sigue siendo polimorfismo, en este caso, depende de la sobrescritura de métodos en las clases derivadas para cambiar el comportamiento. Es útil cuando necesitas implementar diferentes comportamientos en cada subclase y es más sencillo cuando la cantidad de subclases es pequeña.

**¿Qué conviene más?**

Depende del contexto del programa, polimorfismo real es más útil en sistemas más grandes o dinámicos, donde puedes tener varios tipos de objetos y deseas manipularlos a través de una misma interfaz. Es más robusto para manejar múltiples tipos de objetos de forma genérica.Overriding es más adecuado para casos donde el comportamiento específico de cada subclase es crucial, como cuando trabajas con un número limitado de clases con comportamientos claramente diferenciados.