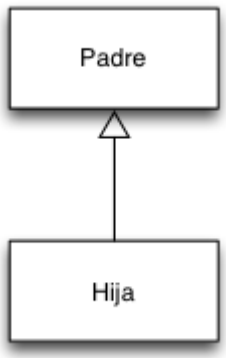
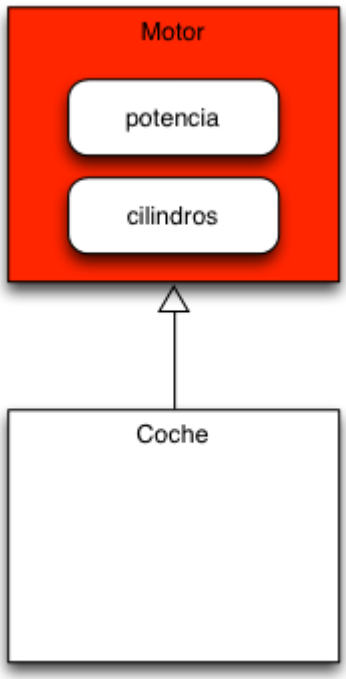
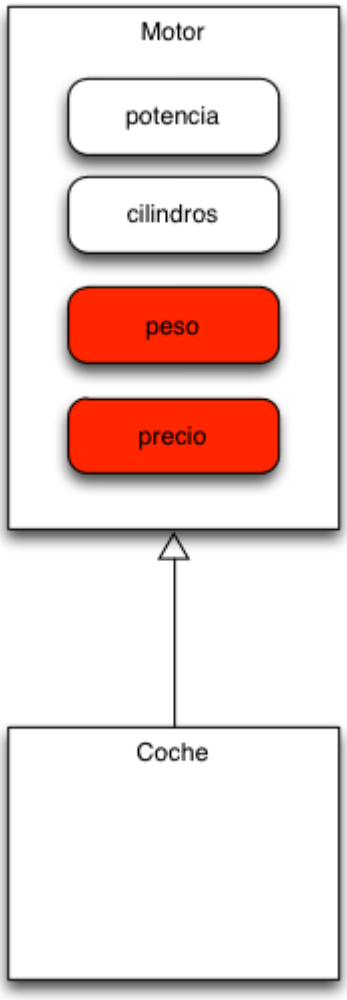
Una de las características mas destacadas de los lenguajes de programación orientada a objetos es su capacidad de reutilizar el código a traves del concepto de herencia.Usando la herencia una clase (hija) puede heredar todas las características de una clase padre.



Esta característica tan comentada por todos es muchas veces mal interpretada a la hora de ponernos a desarrollar .Vamos a ver un ejemplo sencillo.

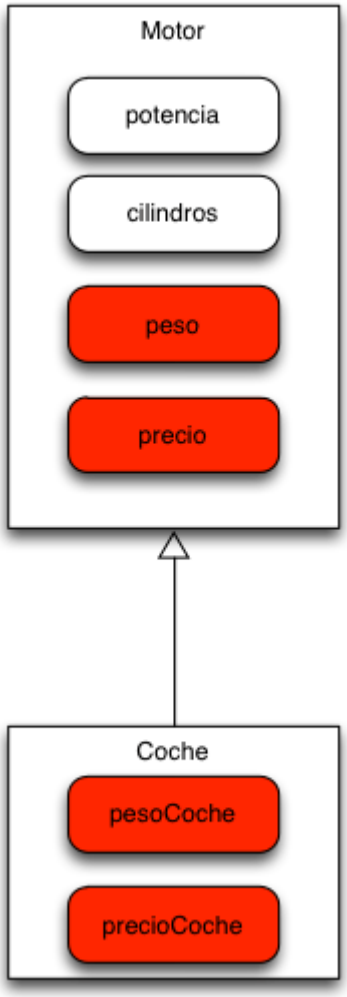


Aqui hemos construido una jerarquia de clases elemental . Tenemos a nuestra disposición la clase Motor y la clase Coche que es su clase hija. Por lo tanto la clase Coche hereda las propiedades de potencia y cilindros que el motor tiene. Todo parece correcto en un primer momento ya que el coche hereda la potencia y los cilindros.Ahora bien imaginemonos que añadimos un par de de propiedades más(peso y precio).

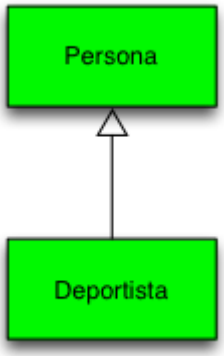


En un principio podemos pensar que son propiedades similares a las anteriores .Sin embargo el motor tiene un peso y el coche tiene otro peso completamente distinto y lo mismo sucede con el precio una cosa es el precio del coche y otra el precio de un motor. Por

lo tanto es mas que probable que un diseño tuviera que tener en cuenta esto reflejandolo en el diagrama de la siguiente forma.



Es un pequeño problema aunque puede ser asumible .Sin embargo en cuanto comencemos a implementar métodos la situación empeorará una cosa es arrancar el motor y otra cosa es arrancar el coche con toda su electronica. Ni que decir tiene que no se parecerá nada el método de parar motor al de parar el coche . Es evidente que nuestro diseño tiene un problema importante.Ahora bien si echamos echamos un vistazo al siguiente diagrama de herencia.



Nos daremos cuenta que si añadimos la propiedad nombre a la clase Persona la clase Deportista lo heredará de una forma natural .Ocurrirá lo mismo con un método como andar o con propiedades como el peso . Esta Jerarquia de clases es mucho mas sólida . No siempre la herencia es la mejor solución para todos nuestros problemas a la hora de reutilizar el código . En el proximo post veremos como usar la herencia y como distinguir situaciones optimas de situaciones problematicas y como solventar estas.