**Capítulo 19. Principios de POO**

Programación orientada a objetos Es probable que te toque hablar con amiguetes que programan en la lengua de Mordor (visualbasic) o gente que programa en c ofuscado, o lo que es peor, desconocidos que te dicen que "programan" en HTML; estos intercambios de experiencias, esas afirmaciones sobre rendimientos de ejecución pueden hacer tambalearse los cimientos de tu fe en la POO. Gracias a estas anotaciones rescatamos del olvido las excelencias de la POO y nos armamos de argumentos ante los herejes que nos salgan al paso con listados de código en ristre.

• Encapsulación: los detalles de implementación están ocultos. Esto reduce que se reproduzcan errores cuando se hacen cambios. Se facilita enormemente la interacción entre otros objetos encapsulados ya que no tienen que conocer los detalles de uso.

• Herencia: este mecanismo es una de las claves de la OOP. Todos los atributos,

métodos, programaciones contenidas en una clase pueden heredarse y extenderse a otras clases facilitando la reutilización de código. Cualquier cambio se propaga en toda la jerarquía de clases y nos vuelve a ahorrar trabajo haciendo un sistema mas fácil de mantener.

• Polimorfismo: gracias a el facilitamos la ampliación del sistema ya que en lugar de crear código por cada tipo de dato lo podemos agrupar todo en uno utilizando la sobrecarga de funciones. El copa-paste puede parecer lo mismo, pero a la hora de cambiar/mantener un sistema se acaban metiendo muchas horas, cosa que con la POO evitas.