

Manifiesto de los objetos

Tres etapas ha recorrido el desarrollo de *software*: la *artesanal* (desde mediados del siglo hasta 1975), la *estructurada* (de entonces hasta 1990) y la *orientada a objetos* (a partir de la presente década).

Al igual que los métodos estructurados en su tiempo, el paradigma de los objetos aporta nuevas técnicas de análisis, diseño y construcción de *software*. Ambos elaboran modelos del mundo real que luego implantan en computadoras; sólo que, mientras que los modelos estructurados conciben la realidad como un *flujo de procesos* que manejan información, los modelos orientados a objetos la entienden de modo similar a como la hace la investigación científica: como un *sistema de objetos* portadores de información y procesos que comparten estructuras de información y comportamiento con todos los de su clase y que se comunican a través de mensajes a los cuales reaccionan polimórficamente.

En efecto, los objetos del mundo real poseen *información* y realizan *procesos* en su interior; y así son los objetos en el nuevo enfoque. Los objetos reales *heredan* de su clase propiedades y comportamiento; y así sucede con los objetos informáticos. Los objetos reales surgen y eventualmente desaparecen; los objetos informáticos se construyen y destruyen a partir de su clase. Los objetos reales se comunican de diversas maneras; los objetos implementados en computadoras reciben y emiten *mensajes*. Los objetos reales reaccionan de muchas formas ante el mismo estímulo, son *polimórficos*; y también lo son los objetos elaborados con el nuevo paradigma.

Los métodos estructurados fueron instrumentos poderosos para reducir la *complejidad*; los métodos orientados a objetos constituyen el siguiente paso en esta dirección. Los métodos estructurados permitieron multiplicar varias veces la capacidad humana para desarrollar *software* frente a los métodos artesanales; los métodos orientados a objetos permiten multiplicar otras tantas veces esta capacidad frente a los métodos estructurados.

La programación estructurada nos enseñó a modularizar procedimientos, pero mantuvo globales los datos; la programación orientada a objetos nos enseña a *encapsular* en una sola unidad —el objeto— tanto procedimientos como datos. Por ello, la orientación a objetos nos permite obtener *software* más modificable y mantenible y hace su administración más efectiva.

La programación estructurada demandó más recursos de *hardware* que la programación artesanal; la programación orientada a objetos permite obtener *software* más poderoso y compacto que la programación estructurada. Los métodos estructurados redujeron el *costo* de fabricación del *software* y aumentaron considerablemente la *productividad* del personal de desarrollo; los métodos orientados a objetos hacen otro tanto con relación a la programación estructurada.

Manuel Antonio Cota Aguilar

Aguascalientes, Ags., la noche del 14 de abril de 1993.

Unos meses después de la redacción de este Manifiesto, el autor y su hijo mayor fundaron la organización de *software* Óptica con el objeto principal de hacer desarrollo e investigación en ingeniería de software orientado a objetos en México, y también para contribuir a su divulgación.

Este Manifiesto se escribió originalmente para apoyar al INEGI a adoptar la tecnología de objetos en sus desarrollos, y lo utilizamos por varios años en conferencias, cursos y diplomados. A pesar del tiempo transcurrido, en la mayoría de las universidades del país todavía no se enseña a programar con objetos correctamente, y en no pocas, los profesores apenas y tocan el tema.

En parte como consecuencia de esto último, muchas empresas de *software* —no sólo en México— siguen construyendo sus productos en forma *artesanal* en perjuicio de sus clientes y usuarios, atentando además contra su propia viabilidad financiera y su permanencia en el mercado. Se puede demostrar con facilidad que hacer *software* de mala calidad es una pésima inversión tanto para quien lo hace como para quien lo adquiere.

Hoy sábado 4 de enero de 2014, lo compartimos para celebrar nuestros primeros 20 años de esta magnífica y apasionante aventura que nos ha obsequiado el hacer buen *software*.