

Taller de Herramientas Computacionales

Valeria Ortiz Cervantes

2 de marzo del 2019

**Universidad Nacional Autónoma de
México.
Facultad de Ciencias.**



Paradigma Orientado a Objetos.

En los años 60, en el Centro de Cálculo Noruego, un equipo de investigación, liderado por Krinsten Nygaard, tuvo problemas en el desarrollo de simulaciones de sistemas físicos, pues los programas eran muy complejos y las modificaciones, al ser una simulación, eran muy frecuentes. La solución que idearon fue diseñar el programa paralelamente al objeto físico, es decir, si el objeto físico tenía treinta componentes, el programa también tendría treinta módulos. Seccionando el programa de esta manera, había una total correspondencia entre el sistema físico y el sistema informático.

Además de resolverse las dificultades anteriores, se obtuvo otro beneficio: la reusabilidad, poder guardar cosas para futuros programas. Para implementar esto crearon el lenguaje Simula 67.

Después, en los años 70 en Estados Unidos, Alan Kay estaba creando un ordenador que pudiera ser utilizado por niños, Dynabook, pero también le resultó muy complicado porque era una programación nueva y experimental, por lo cual creó un lenguaje llamado Smalltalk.

Smalltalk tuvo una gran difusión y cuando en los años 80 en ATT-Bell quisieron crear un sucesor al lenguaje C, incorporaron las principales ideas de Smalltalk y de Simula, creando el lenguaje C++. Puede afirmarse que se debe a este último la gran extensión de los conceptos de la orientación a objetos.

Python

La historia de Python como lenguaje de programación inicia a finales de los 80s y principios de los 90s con Guido Van Rossum. En una navidad de 1989, Guido Van Rossum, quien trabajaba en el CWI, un centro de investigación en Ámsterdam, decidió empezar un proyecto como pasatiempo dándole continuidad a ABC, un lenguaje de programación que se desarrolló ahí mismo a principios de los 80s como alternativa a BASIC, fue pensado para principiantes por su facilidad de aprendizaje y uso. Su código era compacto pero legible.

El proyecto no trascendió ya que el hardware disponible en la época hacía difícil su uso. Así que Van Rossum le dio una segunda vida creando Python. En 1991, Van Rossum publicó el código, en el que ya se tenían disponibles clases con herencias, manejo de excepciones, funciones y tipos modulares.

Además en este lanzamiento inicial aparecía un sistema de módulos adoptado de Modula-3; van Rossum describe el módulo como "uno de las mayores unidades de programación de Python". En el año 1994 se formó el foro de discusión principal de Python, marcando un hito en el crecimiento del grupo de usuarios de este lenguaje.

Paradigmas de programación.

Paradigma declarativo.

En la programación declarativa un programa se describe en términos de proposiciones y afirmaciones que son declaradas para describir el problema, sin especificar los pasos para resolverlo; en este tipo de programas, el estado no puede ser modificado ya que todos los tipos de datos son inmutables. En los lenguajes de programación declarativos, los mecanismos de control están dados por funciones o expresiones puramente matemáticas que carecen de efectos secundarios.

Paradigma imperativo.

En la programación imperativa un programa se describe en términos de instrucciones, condiciones y pasos que modifican el estado de un programa al permitir la mutación de variables, todo esto con el objetivo de llegar a un resultado. Los lenguajes de programación imperativos generalmente hacen uso de procedimientos, rutinas o funciones impuras para establecer mecanismos de control, que potencialmente generan efectos secundarios y mutan el estado del programa durante su ejecución.