

## TOPICOS INTRODUCTORIOS

### QUÉ ES UN PARADIGMA?

Según el historiador Thomas Kuhn un paradigma es: "un conjunto de teorías, estándares y métodos que en conjunto representan una forma de organizar el conocimiento, es decir, una forma de ver la realidad".

### PARADIGMAS DE PROGRAMACION

Es un conjunto de elementos, estructuras y reglas para construir programas. Cada paradigma define un programa con diferentes términos.

Básicamente, un programa es la "simulación computacional de una porción de la realidad". No exactamente la realidad sino lo que creemos haber comprendido de ella. Por ser este un modelo de la realidad existe una distancia con la realidad que simula: GAP (separación) Semántico. Dicha distancia depende del esfuerzo que se debe realizar para reflejar la realidad.

El mejor paradigma es aquel que tiene un menor GAP semántico.

De acuerdo con el Paradigma de Programación, podemos clasificar los lenguajes en las siguientes categorías:

- **Imperativos:** Son aquellos lenguajes, que basan su funcionamiento en un conjunto de instrucciones secuenciales, las cuales, al ejecutarse, van alterando las regiones de memoria donde residen todos los valores de las variables involucradas en el problema que se plantea resolver. Es decir, se cambia progresivamente el estado del sistema, hasta alcanzar la solución del problema [CONTRERAS 01].

Como un ejemplo ilustrativo vamos a escribir un programa en un lenguaje de este tipo para calcular el factorial de un número positivo x.

```
READ(x);  
fac := 1 ;  
for i = 1 to x  
{  
  fac := fac * i ;  
}  
WRITEln(fac);
```

- **Declarativos:** En este paradigma, más que el ¿cómo? desarrollar paso a paso un proceso, nos interesa el ¿qué? deseamos obtener a través del programa. Quizás el lenguaje declarativo que nos sea más familiar, es SQL, el cual es utilizado para interactuar con la información de bases de datos, concentrándose (como se podrá observar en el siguiente ejemplo), sólo en los resultados que van a ser obtenidos, dejándole al traductor (O motor de base de datos) la tarea de cómo llegar a ellos y presentárnoslos. [SANDERS-PRICE 02]

```
SELECT * FROM alumnos WHERE sexo = "M" ORDER BY edad
```

Dentro de este paradigma, se encuentran dos estilos distintos de programación, cada uno de los cuales posee su propia lógica [SANFÉLIX 00].

o **Funcionales**: Son lenguajes basados en funciones, las cuales se representan mediante expresiones, que nos permiten obtener ciertos resultados a partir de una serie de argumentos [BIBBY 00]. De hecho las expresiones están formadas por un conjunto de términos, que a su vez pueden encapsular otras expresiones, para con la evaluación de todas ellas, llegar a la solución deseada.[GAULD 01]. Para describir la idea, retomaremos el ejemplo del factorial escrito en el lenguaje funcional Haskell.

```
fac :: Integer -> Integer
fac 0 = 1
fac x = x * fac (x-1)
```

o **Lógicos**: Este tipo de lenguajes se basan en el cálculo de predicados, la cual es una teoría matemática que permite entre otras cosas, lograr que un ordenador basándose en un conjunto de hechos y de reglas lógicas, pueda derivar en soluciones inteligentes. [DIMARE 90]. El mismo ejemplo del factorial, se vería de la siguiente manera, escrito en PROLOG.

```
factorial (0, 1) factorial (X, Fac) :- Y is X-1, fac(Y, F2), Fac is F2 * X.
```

. **Orientados a Objetos**: Este último paradigma, como se puede observar en la figura 1, algunas veces se mezcla con alguno de los otros 2 modelos, sin embargo mantiene características propias, que lo diferencian claramente. Los programas de este tipo, se concentran en los objetos que van a manipular, y no en la lógica requerida para manipularlos [MARBUS 00]. Ejemplos de objetos pueden ser: estudiantes, coches, casas etc, cada uno de los cuales tendrá ciertas funciones (métodos) y ciertos valores que los identifican, teniendo además, la facultad de comunicarse entre ellos a través del paso de mensajes.

Tratando de resumir un poco, presentaremos los siguientes cuadros evolutivos, donde aparecen los lenguajes que por su uso y comercialización, han resultado ser los más populares a lo largo de este medio siglo. [LABRA 98] [RUS 01].

Como ya lo citamos anteriormente y como se puede observar en los cuadros, la existencia de tantos lenguajes obedece a que cada uno de ellos está encaminado a resolver ciertas tareas, dentro de la amplia problemática de la explotación de la información, o bien, a que su arquitectura, o su forma de llevar a cabo la programación, tiene un enfoque particular.

Ahora bien, si tomamos como referencia las herramientas usadas en el proceso de traducción y ejecución de los programas esbozada en la figura 2, vamos a tener la siguiente clasificación de lenguajes [AHO 77]:

Programa Objeto Es el que Entiende la Máquina

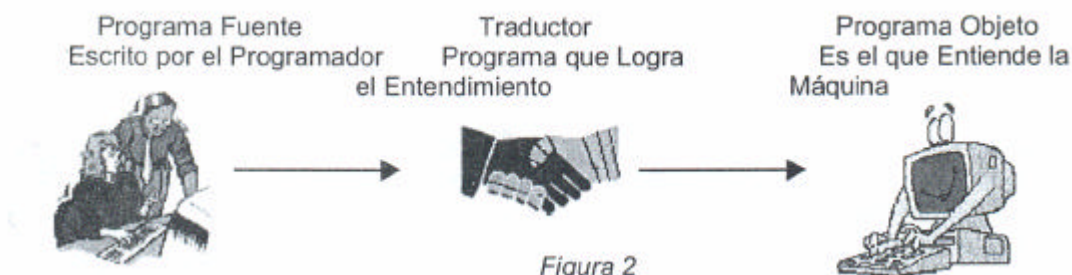


Figura 2

- **Lenguajes Ensamblados:** Se refieren al lenguaje ensamblador, que viene a ser una representación simbólica de las instrucciones correspondientes al lenguaje ensamblador de alguna arquitectura específica, con lo que, casi siempre, la correspondencia entre las instrucciones de este lenguaje, y las del lenguaje máquina son de 1 a 1, si bien existen algunas excepciones, que dan lugar a lo que se conoce como lenguajes macro-ensambladores [CUEVA 88]

- **Lenguajes Compilados:** Son aquellos, que son traducidos de un lenguaje de alto nivel (como FORTRAN, PASCAL, C, C++) a lenguaje máquina o bien a lenguaje ensamblador, produciendo un programa objeto permanente.

- **Lenguajes Interpretados:** Estos lenguajes, tienen la particularidad, de que no producen código objeto, sino que cada instrucción es analizada y ejecutada a la vez, lo que ofrece mucha interacción con los usuarios, pero a la vez resultan ineficientes, cuando se desea ejecutar repetitivamente un programa.

- **Lenguajes Preprocesados:** Son lenguajes que son traducidos primeramente a un lenguaje intermedio de más bajo nivel, para posteriormente volverlos a traducir y producir el programa objeto. Este tipo de lenguajes fueron creados, con la idea de proporcionar un lenguaje más potente que el lenguaje intermedio, mediante la implementación de algunas macroinstrucciones. [SANCHIS-GALAN 86].

- **Híbrido,** (a medio camino entre compilado e interpretado). El lenguaje que mas usamos, Java, es un lenguaje orientado a objetos creado por James Gosling e introducido por Sun Microsystems en junio de 1995. Fue diseñado como un lenguaje de sintaxis muy similar a la del C++, pero con ciertas características diferentes que lo hacen más simple y **portable** a través de diferentes plataformas y sistemas operativos (tanto a nivel de código fuente como nivel binario). A grandes rasgos, se compila el código fuente \*java a un formato binario \*.class (bytecode), que luego es interpretado por una máquina virtual java, la cual debe estar implementada para la plataforma particular usada. A diferencia de los lenguajes ejecutables (Aplicaciones .exe) que corren directamente sobre el sistema operativo, (y necesitan ser compilados para ese sistema operativo) el formato binario .class corre sobre la máquina virtual Java, y es **ella quien se conecta con el sistema operativo**. En definitiva, máquinas virtuales Java adecuadas al sistema operativo, aplicaciones .class genéricas, portables de un SO a otro.



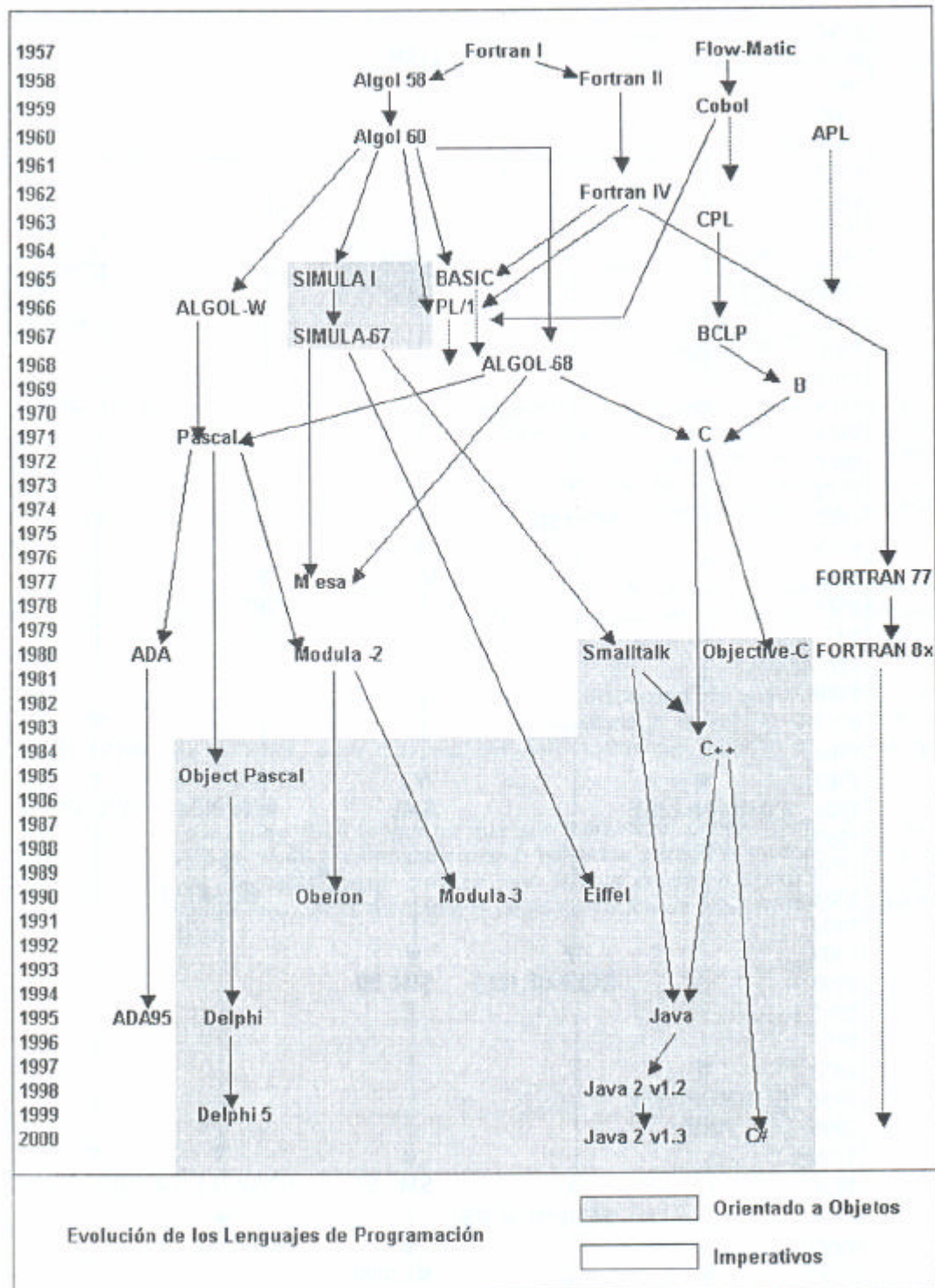
**Figura a. Evolución de los Lenguajes Imperativos y Orientados a Objetos**

Figura b. Evolución de los lenguajes declarativos

