Evolución y Clasificación de los Lenguajes de Programación

Entornos de Desarrollo
1º DAW Semipresencial

Alberto Requena Sáez
20 de Octubre de 2023

1. Evolución y Clasificación de los Lenguajes de Programación

Un **Lenguaje de Programación** es el conjunto de símbolos y palabras, instrucciones y sentencias que el usuario tiene a su disposición para elaborar un programa. Es básicamente como el usuario se puede comunicar con el ordenador.

Se puede clasificar a los lenguajes de programación según a la generación a la que pertenezcan. Son 5 generaciones, y cuanto menor es el número más antigua es su aparición, y más próximo es al lenguaje de la máquina, es decir, menos parecido al lenguaje humanizado es.

1.1 1ª Generación

Comienza a mitad del siglo XX, antes de los años 50.

A esta primera generación pertenecen los lenguajes máquina, que son aquellos que entienden los procesadores de los ordenadores. Para ello utilizamos el código binario.

Su ejecución es extremadamente rápida pues el procesador entiende perfectamente el lenguaje y no necesita de ninguna interpretación intermedia.

Como inconvenientes, estos lenguajes son muy complicados de utilizar para el ser humano. Además, no existe un lenguaje máquina universal, hay multitud de ellos según el tipo de procesador, aunque todos utilicen el código binario.

En estos lenguajes se utilizaban conmutadores eléctricos y tarjetas perforadas.

1.2 2ª Generación

Comienza a mitad del siglo XX, durante los años 50.

A esta generación pertenecen los lenguajes ensambladores, los cuales ya no utilizan el código binario. La traducción al lenguaje máquina es muy sencilla y extremadamente rápida, pero siguen siendo muy complicados para los humanos.

```
title helloWorld

include Irvine32.Inc

.data
msg byte "Hola!", 0

.code
main proc
    mov edx, offset msg
    call writestring
    exit
main endp
end main
```

Como en el caso de los lenguajes máquina, no hay un lenguaje ensamblador universal, hay multitud de ellos, lo cual es un gran inconveniente.

Hoy en día aún se utilizan estos lenguajes para la programación de drivers.

1.3 3ª Generación

Aparecen a finales de los años 50 del siglo XX.

Son ya lenguajes considerados de alto nivel, mucho más comprensibles para el ser humano, lo que facilita enormemente el trabajo para los programadores. También su mantenimiento es mucho más sencillo.

Por contra, se necesita de compiladores e intérpretes para utilizarlos y se pierde en velocidad de ejecución pues se necesitan muchos más recursos para utilizarlos.

A esta generación pertenecen lenguajes como Fortran, Cobol, Basic, Pascal, C, C++ y Java.

```
#include <stdio.h>

int main() {
   printf("Hola!");
   return 0;
}
```

A principios de los años 70, con estos lenguajes se empiezan a desarrollar la programación estructurada y la programación modular.

1.4 4ª Generación

Se desarrollan a partir de los años 70 y 80.

A esta generación pertenecen los lenguajes de propósito específico como Sql o Matlab. Estos lenguajes se centran en proporcionar un mayor nivel de abstracción que los de 3ª generación, pareciéndose más al lenguaje humano, y abordando tipos concretos de problemas, que suelen operar con conjuntos de datos muy grandes.

```
SELECT nombre FROM Alumnos;
```

Se utilizan habitualmente con entornos visuales, aprovechando las herramientas que estos entornos ofrecen.

Puesto que hay un alto grado de reutilización del código, la productividad suele ser muy grande. Por otro lado, al ser lenguajes desarrollados para tareas específicas, no son tan versátiles como los de 3ª generación.

1.5 5ª Generación

Aparecen en los años 80 y 90.

Los lenguajes declarativos, el lógico y el funcional, son lenguajes de 5ª generación.

Estos lenguajes se caracterizan en que se centran en **qué** hacer para resolver el problema. En el caso del lenguaje de programación lógico la resolución de problemas está basada en restricciones, en lugar de que haya un algoritmo creado por el programador que sea el encargado de resolverlo. En el lenguaje de programación funcional se hace la utilización continua de funciones.

```
write('Hello World').
```

Son lenguajes que suelen consumir más recursos que los de generaciones menores.