
Introducción al desarrollo del software

Entornos de desarrollo

ÍNDICE DE CONTENIDOS

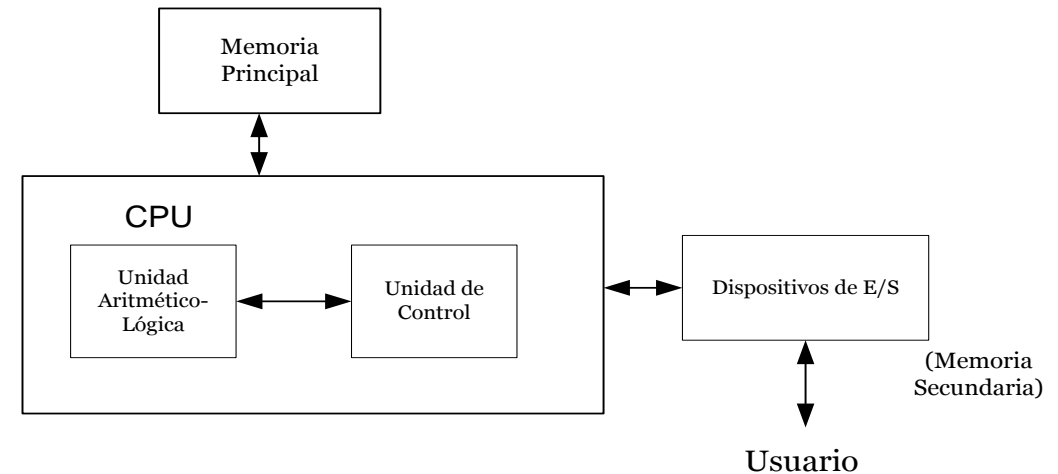
- Hardware.
- Software.
- Lenguajes de programación.

Hardware

El hardware es la parte **tangible** de un sistema informático, equipamiento físico.

Generalmente, un **ordenador digital** se basa en la arquitectura de Von Neumann 1945, la cuál divide un ordenador en las siguientes partes:

- Unidad de Control.
- Unidad Aritmético-Lógica.
- Memoria principal
- Dispositivos de Entrada/Salida



Hardware

Unidad de control - es el componente básico de un ordenador ya que controla la ejecución de las operaciones y dirige el funcionamiento de todos los demás componentes de tal forma que el trabajo conjunto de todos conduzca a la consecución de las tareas específicas programadas en el sistema.

Unidad Aritmético-Lógica – es la parte encargada de realizar las operaciones aritméticas (suma, resta, ...) y las operaciones lógicas (comparación, ...). También realiza otras funciones más complejas (raíces, funciones trigonométricas, ..).

Al conjunto Unidad de Control y Unidad Aritmético-Lógica se le conoce como **CPU (Central Process Unit – Unidad Central de proceso)**. Así tenemos los procesadores Intel, Motorola, AMD.

Memoria principal – es la memoria de almacenamiento interno. Opera a gran velocidad. Aquí se ubican los programas: las instrucciones junto con los datos sobre los que actúan.

La memoria principal está formada por una serie de *celdas* o posiciones identificadas por una *dirección* de memoria.

Por otro lado está la *memoria secundaria* o *memoria externa* que permite resolver los problemas de volatilidad y capacidad de la memoria principal (discos duros, CD, ...).

Dispositivos de Entrada / Salida – son los que facilitan la interacción del usuario (el mundo exterior) con la máquina (teclado, monitor, impresora, ...).

Hardware

El ciclo de una instrucción es:

- 1) La Unidad de Control indica a la Memoria Principal la siguiente instrucción que quiere ejecutar.
- 2) La U.C. recibe la instrucción, procediendo a su análisis para determinar los operandos sobre los que actúa y su localización.
- 3) Bajo las directrices de la U.C. la U.A.L realiza la operación y se guarda el resultado en su destino.
- 4) Una vez ejecutada la instrucción se incrementa el contador de programa y se pasa a realizar la ejecución de la siguiente instrucción.



Hardware

Generaciones de un ordenador digital:

- 1) Primera generación
- 2) Segunda generación
- 3) Tercera generación
- 4) Cuarta generación
- 5) Quinta generación

Hardware

Generaciones de un ordenador digital:

1) Primera generación

- ❖ 1945-1955
- ❖ Tubos de vacío
- ❖ Se programaba en binario
- ❖ Se requería la intervención humana



2) Segunda generación

3) Tercera generación

4) Cuarta generación

5) Quinta generación

Hardware

Generaciones de un ordenador digital:

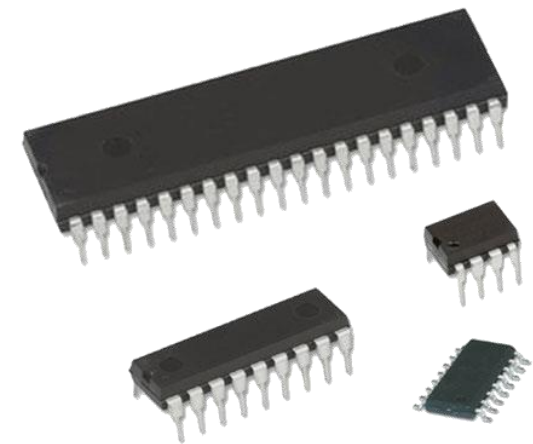
- 1) Primera generación
- 2) Segunda generación
 - ❖ 1955-1965
 - ❖ Transistores
 - ❖ Se utilizaba lenguaje ensamblador
 - ❖ Programación en lotes
- 3) Tercera generación
- 4) Cuarta generación
- 5) Quinta generación



Hardware

Generaciones de un ordenador digital:

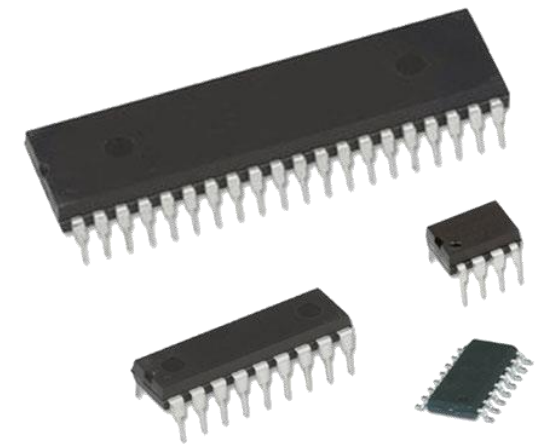
- 1) Primera generación
- 2) Segunda generación
- 3) Tercera generación
 - ❖ 1965-1971
 - ❖ Circuitos integrados
 - ❖ Se utilizaban lenguajes de alto nivel
 - ❖ Se desarrolló la multiprogramación y los sistemas compartidos.
- 4) Cuarta generación
- 5) Quinta generación



Hardware

Generaciones de un ordenador digital:

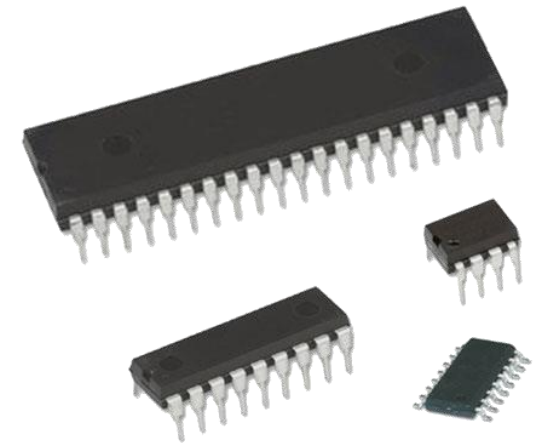
- 1) Primera generación
- 2) Segunda generación
- 3) Tercera generación
- 4) Cuarta generación
 - ❖ 1971-1985
 - ❖ Circuitos integrados de alta densidad LCI
 - ❖ Se utilizaban lenguajes de alto nivel
 - ❖ Aparecen los ordenadores personales.
- 5) Quinta generación



Hardware

Generaciones de un ordenador digital:

- 1) Primera generación
- 2) Segunda generación
- 3) Tercera generación
- 4) Cuarta generación
- 5) Quinta generación
 - ❖ 1985-actualidad
 - ❖ Circuitos integrados de ultra alta densidad ULCI
 - ❖ Se utilizaban lenguajes de alto nivel
 - ❖ Computación en red, distribuida y auge del desarrollo del software

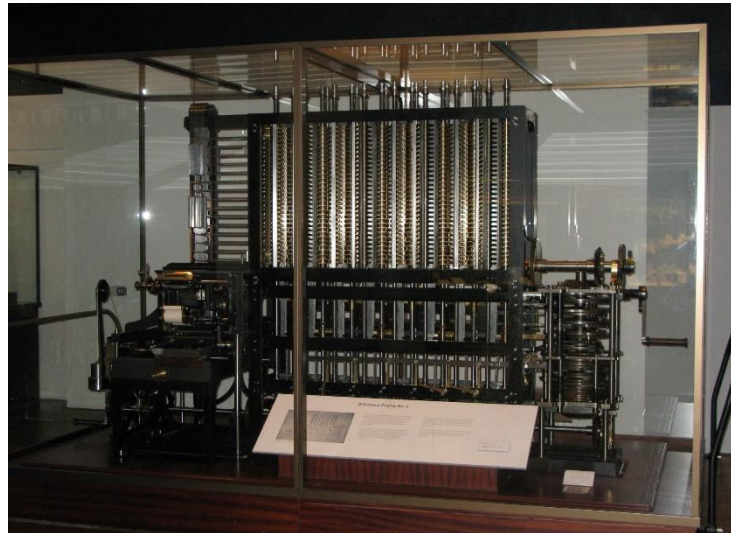


Software

El software es la parte **intangible** de un sistema informático, equipamiento lógico.

El software es el encargado de comunicarse con el hardware, es decir, se encarga de traducir las órdenes dadas por el usuario en órdenes comprensibles por el hardware.

El concepto de software fue utilizado por primera vez por Charles Babbage (1822), máquina diferencial.



Software

Alan Turing (1912-1954) profundizó en el concepto de software y destacó como criptógrafo gracias a la creación de la máquina que lleva su mismo nombre.

Actualmente es considerado como el padre de la computación, diseñó el sistema de programación del primer ordenador comercial y sentó las bases de la [inteligencia artificial](#).

Software: Características.

- 1) El software es **lógico**, intangible.
- 2) El software **se desarrolla**, no se fabrica.
- 3) El software **no se estropea**, una copia suya da lugar a un clon con las mismas características.
- 4) Se puede construir **a medida**, puede construirse a medida o enlatado.

Software: Tipos.

Según sus funciones, podemos destacar tres tipos de software:

- 1) El **software de sistema** es el software base que interactúa con el resto de programas y, además, controla el hardware. Podemos dividirlos en: Sistema operativo, controladores de dispositivos y programas utilitarios.
- 2) El **software de aplicación** son aquellos programas que tienen una finalidad más o menos concreta. Estos han sido diseñados para facilitar al usuario la realización de ciertas tareas, por ejemplo: aplicaciones ofimáticas, software educativo, editores de audio,...
- 3) El **software de programación** es el conjunto de herramientas que nos permiten desarrollar programas informáticos. Este tipo de software comprende: compiladores, intérpretes, ensambladores, enlazadores,...

Software: Tipos.

Según su tipo de distribución, podemos destacar tres tipos de software:

- **Shareware:** Es una modalidad de distribución de software, para que el usuario pueda evaluar el producto de forma gratuita ***durante un tiempo especificado***. Por ejemplo WinZip, WinRar, Virus Scan ...
- **Freeware:** Software que se distribuye sin cargo. A veces se incluye el código fuente, pero no es lo habitual.
- **Adware:** Programas shareware que automáticamente descargan publicidad. Al comprar la licencia del programa se elimina la publicidad.

Lenguajes de Programación. Concepto

Según la RAE:

Un lenguaje de programación es el lenguaje que facilita la comunicación con un computador mediante signos convencionales cercanos a los de un lenguaje natural. Es decir, un lenguaje de programación es una herramienta que nos permite crear programas ejecutables por el ordenador.

Lenguajes de Programación. Características

- a) **Facilidad de lectura y comprensión:** Los programas deben ser corregidos y modificados, por lo tanto, el lenguaje debe facilitar la comprensión de los programas una vez escritos.
- b) **Facilidad de codificación:** Debe ser simple utilizar el lenguaje para desarrollar programas que resuelvan el tipo de problemas al que está orientado.
- c) **Fiabilidad:** Es fiable si funciona bien en cualquier condición. Depende de la comprobación de tipos de datos, del control de excepciones y del manejo de la memoria.
- d) **Posibilidad de usar un entorno de programación:** Se refiere a la existencia de herramientas para el desarrollo.
- e) **Portabilidad:** Un mismo programa debe funcionar en ordenadores diferentes.
- f) **Coste:** Relacionado con la curva de aprendizaje, codificación, ejecución, fiabilidad y mantenimiento.
- g) **Detallabilidad:** Define el número de pasos que es necesario definir en un programa para dar solución a un determinado problema.
- h) **Generalidad:** Indica las opciones de uso que tiene un lenguaje para resolver problemas de distintito tipo. Cuantos más problemas diferentes pueda solucionar más general es el lenguaje.

Lenguajes de Programación. Elementos

Todo lenguaje de programación está constituido por tres niveles, donde la mayoría de ellos, tienen los siguientes elementos comunes dentro de cada nivel:

a) Nivel léxico: Son los símbolos que se pueden usar en el lenguaje:

- **Identificadores:** Nombres simbólicos que se da a ciertos elementos de programación, por ejemplo, nombre de variables, tipos, etc.
- **Constantes:** Una constante es un dato cuyo valor permanece inalterable durante toda la ejecución del programa.
- **Operadores:** Símbolos que representas operaciones entre variables y constantes. Los operadores más comunes son: aritméticos, lógicos, incrementales, relacionales y de asignación.
- **Palabras claves:** Son aquellas que palabras predefinidas por el lenguaje que tienen un significado especial para este. La lista de palabras claves de Java [aquí](#).
- **Comentarios:** Símbolo que se utiliza para documentar los programas explicando el código fuente.
- **Separadores:** Los separadores están formados por espacios en blanco, tabuladores y saltos de línea, y su función es ayudar al compilador a descomponer el programa en token y facilitar la legibilidad del programa.

Lenguajes de Programación. Elementos

Todo lenguaje de programación está constituido por tres niveles, donde la mayoría de ellos, tienen los siguientes elementos comunes dentro de cada nivel:

- a) **Nivel léxico:** Son los símbolos que se pueden usar en el lenguaje.
- b) **Nivel sintáctico:** Es el conjunto de normas que determinan cómo se pueden utilizar los símbolos del lenguaje para crear sentencias. Las reglas sintácticas pueden tener elementos terminales (símbolos léxicos) y elementos no terminales (construcciones gramaticales intermedias).

Una vez definida la sintaxis del lenguaje, el compilador determinará si las sentencias de un código fuente son válidas, de acuerdo a esta sintaxis. Para esto se utilizan árboles de análisis sintáctico y algoritmos de parsing.

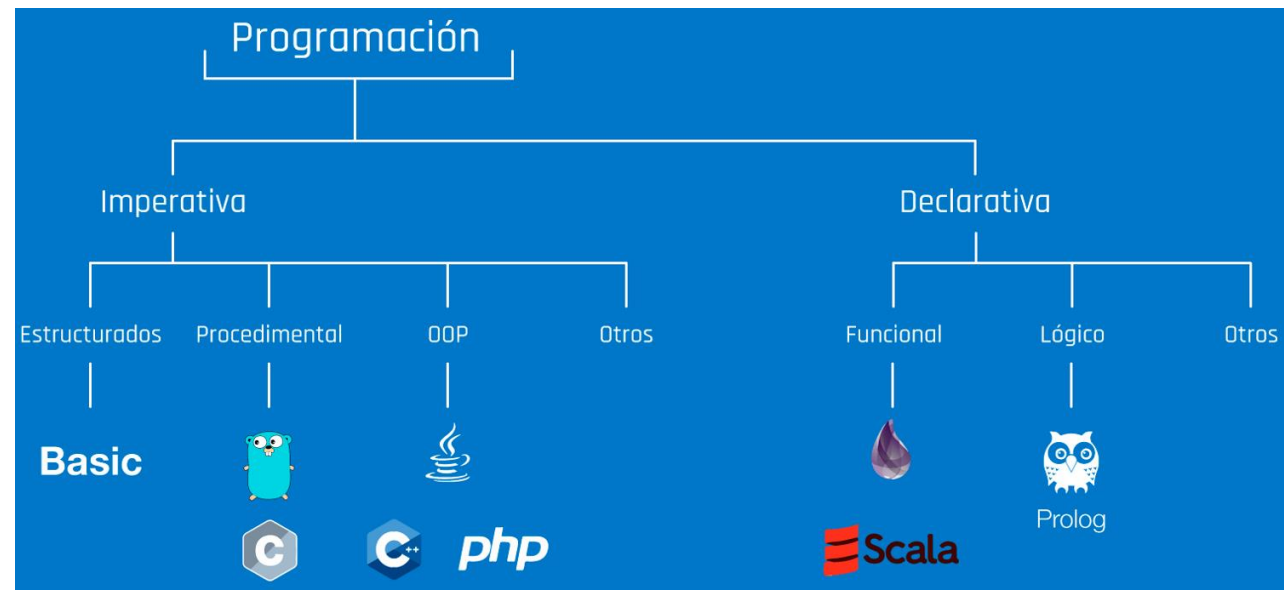
- a) **Nivel semántico:** La semántica de un lenguaje de programación se refiere al significado que adoptan las distintas sentencias, expresiones y enunciados de un programa. La semántica engloba aspectos sensibles al contexto. Los principales elementos de la semántica son:

Lenguajes de Programación. Elementos

- c) **Nivel semántico:** La semántica de un lenguaje de programación se refiere al significado que adoptan las distintas sentencias, expresiones y enunciados de un programa. La semántica engloba aspectos sensibles al contexto. Los principales elementos de la semántica son:
- **Variables:** Una variable representa un espacio de memoria para almacenar un valor de un determinado tipo. El valor de una variable, puede cambiar durante la ejecución de un programa.
 - **Valores y referencias:** Los valores son el estado de determinada celda o grupo de celdas de la memoria, mientras que las referencias indican la posición de esa celda en memoria. Estos dos conceptos están muy involucrados con los punteros. La mayoría de los lenguajes los soportan, pero son una gran fuente de errores de programación.
 - **Expresiones:** Son construcciones sintácticas que permiten combinar valores con operadores y producir nuevos valores. Las expresiones pueden ser aritméticas, relacionales, lógicas y condicionales. Cada una de estas tiene una semántica específica que la define.
 - **Gramática de atributos:** Las gramáticas de atributos permiten formalizar aspectos sensibles al contexto. Por ejemplo, el chequeo de tipos depende del contexto porque debemos saber el tipo esperado y el actual y determinar si son compatibles. El tipo esperado lo obtenemos del contexto analizando la definición de la variable.

Lenguajes de Programación. Paradigmas

Un paradigma de programación es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.



Lenguajes de Programación. Tipos

A continuación, clasificaremos los lenguajes de programación según:

- **Nivel de abstracción:** Esta clasificación se realiza atendiendo a la forma en que los lenguajes son validados por el ordenador, es decir, lo próximo que está el lenguaje programación al lenguaje máquina.
- **Clasificación cronológica:** Esta clasificación se basa en la fecha de creación de los lenguajes
- **Clasificación por paradigmas:** Los paradigmas de programación consisten en el método que llevan a cabo los cálculos y la forma que deben estructurarse y organizarse las tareas que debe realizar el programa.

La mayoría de los paradigmas deberían de dos tipos de programación:

- **Imperativa:** Es aquella en la que se describe paso a paso un conjunto de instrucciones que deben realizarse para ejecutarse para variar el estado del programa y hallar la solución.
- **Declarativa:** Las sentencias describen el problema que se quiere solucionar; se programa diciendo lo que se quiere resolver a nivel de usuario, pero no las instrucciones necesarias para realizarlo.

Lenguajes de Programación. Tipos

Según su nivel de abstracción, nos encontramos con los siguientes tipos:

- **Lenguaje máquina:** Estas instrucciones están formada por secuencias interminables de 1 y 0 que conforman operaciones que la máquina puede entender sin ningún tipo de interpretación.
 - Este lenguaje depende de la arquitectura de la máquina, CPU, lo cual implica baja portabilidad.
 - No existen comentarios.
 - Los datos se referencian directamente con la dirección de memoria en la que se encuentra.
 - Las instrucciones realizan operaciones muy simples.
 - Código muy rígido, lo cuál conlleva poca versatilidad a la hora de escribir el programa.

Lenguajes de Programación. Tipos

Según su nivel de abstracción, nos encontramos con los siguientes tipos:

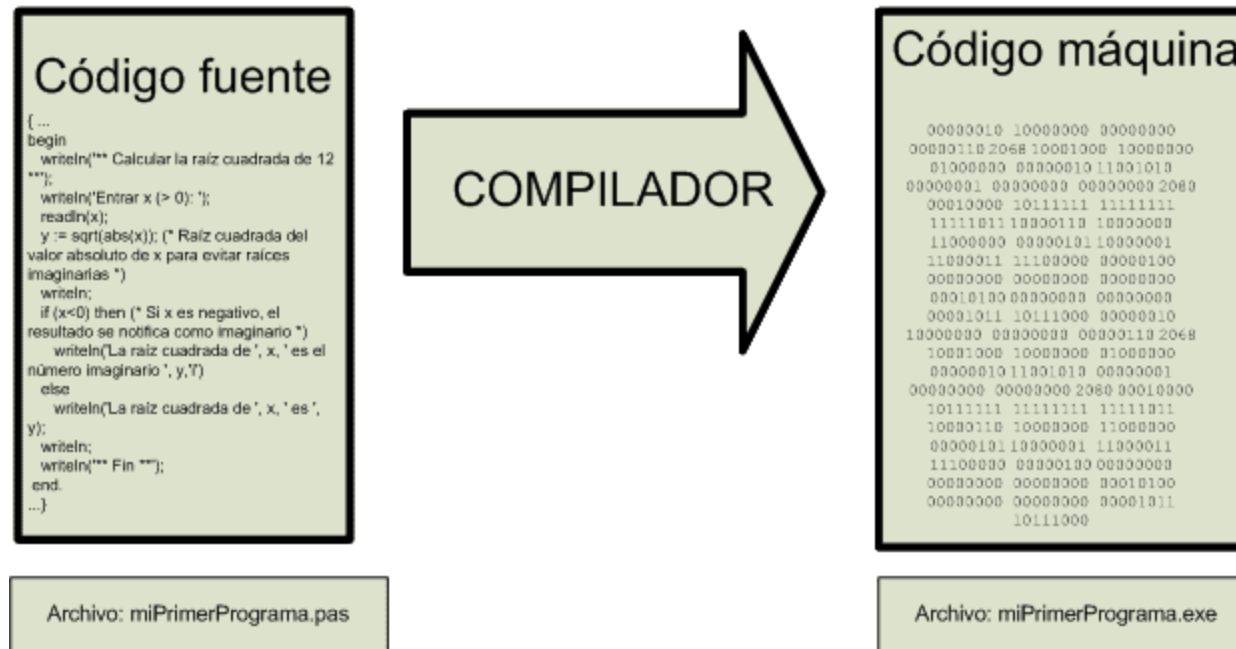
- **Lenguaje ensamblador o de bajo nivel:** Las secuencias interminables de 1 y 0 son reemplazadas por símbolos reconocibles llamados **mnemotécnicos** (normalmente abreviaturas del inglés) para representar las instrucciones.
 - Permite la utilización de **direccionamiento simbólico**, es decir, en vez de utilizar direcciones binarias podemos identificar los datos con nombres (variables).
 - Permite el uso de **comentarios**.
 - Sigue siendo dependiente de la arquitectura de la CPU.
 - Realiza una gestión óptima de los recursos hardware, permitiendo un código muy eficiente.
 - Es necesario traducir este código a lenguaje máquina, para ello se utiliza un **compilador**.

Lenguajes de Programación. Tipos

Según su nivel de abstracción, nos encontramos con los siguientes tipos:

- **Lenguaje de alto nivel:** Estos lenguajes permiten programar sin necesidad de conocer el funcionamiento interno de la máquina. Normalmente una instrucción de estos lenguajes equivale a varias instrucciones máquina. Se diseñaron con la finalidad de que las personas escriban y entiendan los programas de un modo más fácil que los lenguajes máquina y ensambladores.
 - Son independientes de la arquitectura física del ordenador, lo cuál, los hace portables de un ordenador a otro.
 - Las operaciones se expresan de una forma muy similar al lenguaje matemático o lenguaje natural (generalmente inglés).
 - Permiten el uso de los comentarios y de las variables.
 - Requieren de un compilador para su traducción a lenguaje máquina.

Lenguajes de Programación. Tipos



Lenguajes de Programación. Tipos

Según aparición cronológica, nos encontramos con:

- **Primera generación:** Coincide con el lenguaje máquina visto en la clasificación anterior.
- **Segunda generación:** Aparición de los lenguajes simbólicos, coincide con el lenguaje ensamblador o bajo nivel.
- **Tercera generación:** Aparición de los lenguajes de alto nivel con técnicas de programación (estructurada, concurrente, orientada a objetos) que logran un gran rendimiento computacional con respecto a los lenguajes anteriores.
- **Cuarta generación:** Esta generación está marcada por la aparición de herramientas orientadas al desarrollo de aplicaciones de gestión que permiten al programador automatizar operaciones como, definir BBDD, informes, consultas...
 - Permiten el acceso a base de datos.
 - Generación automática de código.
 - Mejoran el rendimiento y la productividad del programador.

Lenguajes de Programación. Tipos

Según aparición cronológica, nos encontramos con:

- **Primera generación:** Coincide con el lenguaje máquina visto en la clasificación anterior.
- **Segunda generación:** Aparición de los lenguajes simbólicos, coincide con el lenguaje ensamblador o bajo nivel.
- **Tercera generación:** Aparición de los lenguajes de alto nivel con técnicas de programación (estructurada, concurrente, orientada a objetos) que logran un gran rendimiento computacional con respecto a los lenguajes anteriores.
- **Cuarta generación:** Esta generación está marcada por la aparición de herramientas orientadas al desarrollo de aplicaciones de gestión que permiten al programador automatizar operaciones como, definir BBDD, informes, consultas...
- **Quinta generación:** Son aquellos lenguajes asociados a sistemas expertos y a la inteligencia artificial. Se caracterizan por utilizar programación declarativa y lógica.

Lenguajes de Programación. Tipos

Según al paradigma de programación al que pertenezcan:

- **Programación estructurada:** Orienta a mejorar la claridad, calidad y tiempo de desarrollo de un programa recurriendo a subrutinas y tres estructuras básicas: secuencia, selección (if y switch) e iteración (for y while).
- **Programación modular:** Consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.
- **Programación orientada a objetos:** Está basada en la programación imperativa, pero el núcleo central de este paradigma es la unión de datos y procesamiento en una entidad llamada “objeto”, la cuál a su vez es relacionable con otras entidades “objeto”.
- **Programación dinámica:** Se basa en dividir un problema en más pequeños para analizarlos y resolverlos de forma más cercana al óptimo. Este paradigma se basa en el modo de realizar los algoritmos, por lo que se puede usar cualquier lenguaje imperativo.

Lenguajes de Programación. Tipos

Según al paradigma de programación al que pertenezcan:

- **Programación funcional:** Se basa en la definición de predicados y es de corte más matemático.
- **Programación lógica:** Se basa en la definición de relaciones lógicas, son lenguajes en los que se especifican un conjunto de hechos y una serie de reglas que permiten la deducción de otros hechos. Esto permite al sistema encontrar la solución.
- **Programación con restricciones:** Similar a la anterior, pero usando ecuaciones.
- **Lenguaje específico del domino o DSL:** Se denomina así a los lenguajes desarrollados para resolver un problema específico, pudiendo entrar dentro de cualquier grupo anterior.
- **Programación multiparadigma:** Se basa en el uso de dos o más paradigmas dentro de un mismo lenguaje de programación.

Ejercicio

El ranking de [TIOBE](#), muestra los lenguajes de programación más populares a la hora de desarrollar software.

Clasifica los diez primeros lenguajes de programación dentro su nivel de abstracción y paradigma de programación.

Además, clasifica los siguientes lenguajes:

➤ Go

➤ Dart

➤ Kotlin

➤ Lisp

➤ Ada

➤ Fortran