



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Materia: Fundamentos de la Programación

Tarea #2 Historia de la Programación

Alumno: Enrique Ichazo Bautistas

Fecha: 02/10/2020



## Historia del cómputo:

Todo comenzó con máquinas destinadas a manejar números, es así como nos remitimos a él Ábaco, inventado por los babilonios allá por el año 1000 A.C. Utilizado sobre todo por los chinos para la realización de operaciones sencillas, está formado por una tablilla con una serie de cuentas que sirven para efectuar sumas y restas.

En 1617, el matemático escocés John Napier desarrolló un juego de palitos para calcular a las que llamó «Napier Bones.» Así llamados porque se tallaron las ramitas de hueso o marfil, los «bones» incorporaron el sistema logarítmico.

En 1623 la primera calculadora mecánica fue diseñada por Wilhelm Schickard en Alemania. Llamado «El Reloj Calculador», la máquina incorporó los logaritmos de Napier, hacia rodar cilindros en un albergue grande.

Uno de los antepasados más directos de la computadora actual, fue creada por el científico francés **Blaise Pascal** en el siglo XVII (1642). A sus 18 años, Pascal inventó su primera **maquina** calculadora, capaz de sumar y restar; y todo ello a base de engarzar múltiples ruedas dentadas.

Años más tarde, en 1673, Gottfried Von Leibnitz perfeccionó los estudios de Pascal, y llegó a construir una máquina que no solo sumaba y restaba, sino que también multiplicaba, dividía e incluso calculaba raíces cuadradas.

Se inventó la primera máquina lógica en 1777 por Charles Mahon, el Conde de Stanhope. El «demostrador lógico» era un aparato tamaño bolsillo que resolvía silogismos tradicionales y preguntas elementales de probabilidad. Mahon es el precursor de los componentes lógicos en computadoras modernas.

Fue en 1830, cuando se establecieron los principios de funcionamiento de las modernas computadoras. Su paternidad se debe al matemático inglés Charles Babbage, quien tras lanzar en 1822 la denominada maquina diferencial– con nada menos que 96 ruedas dentadas y 24 ejes, se lanzó en pos de su proyecto más relevante: la máquina analítica (1833). Mientras tanto Charles Jacquard (francés), fabricante de tejidos, había creado un telar que podía reproducir automáticamente patrones de tejidos leyendo la información codificada en patrones de agujeros perforados en tarjetas de papel rígido. Al enterarse de este método Babbage abandonó la máquina de diferencias y se dedicó al proyecto de la máquina analítica que se pudiera programar con tarjetas perforadas para efectuar cualquier cálculo con una precisión de 20 dígitos.

En 1944 se construyó en la Universidad de Harvard, la Mark I, diseñada por un equipo encabezado por Howard H. Aiken. Esta máquina no está considerada como computadora electrónica debido a que no era de propósito general y su

funcionamiento estaba basado en dispositivos electromecánicos llamados relevadores.

## **Historia de la programación:**

El conjunto de Ordenes e instrucciones que se dan al ordenador para que resuelva un problema o ejecute una determinada misión, recibe el nombre de programa. En los primeros tiempos de la Informática, la programación se efectuaba en el único lenguaje que entiende el microprocesador: su propio código binario, también denominado lenguaje máquina o código máquina. Pero la programación en lenguaje máquina resulta muy lenta y tediosa, pues los datos e instrucciones se deben introducir en sistema binario y, además, obliga a conocer las posiciones de memoria donde se almacenan los datos.

Por este motivo, a principios de los 50 se creó una notación simbólica, denominada código de ensamblaje (ASSEMBLY), que utiliza una serie de abreviaturas mnemotécnicas para representar las operaciones:

ADD (sumar), STORE (copiar), etc. Al principio, la traducción del código de ensamblaje al código máquina se realizaba manualmente, pero enseguida se vio que el ordenador también podría encargarse de esa traducción; se desarrolló así un programa traductor, llamado ensamblador<sup>1</sup> (ASSEMBLER).

Conforme los ordenadores fueron introduciéndose en el mundo empresarial y académico, aquellos primitivos lenguajes fueron sustituidos por otros más sencillos de aprender y más cómodos de emplear. Estos lenguajes, llamados de alto nivel, tienen una estructura que se adapta más al pensamiento humano que a la forma de trabajar del ordenador. Por ejemplo, seguro que le suenan lenguajes como BASIC, PASCAL, C, etc.

Las órdenes dadas en cualquier lenguaje siempre deben traducirse al código binario del ordenador, que es el único que realmente comprende su unidad central. Esta labor de traducción se lleva a cabo mediante un intérprete o un compilador.

El intérprete informático, traduce instrucción a instrucción y, de esta forma, favorece la interactividad, la depuración y puesta a punto del programa, la ejecución inmediata de una orden, etc. Por ejemplo, entre los lenguajes que suelen ser interpretados, se encuentran BASIC, LOGO, etc.

el compilador traduce todo el programa de golpe, dejándolo listo para ser ejecutado<sup>3</sup>. De esta forma, se logra mayor rapidez en la ejecución y, además, se liberan recursos de la memoria, pues el programa, una vez compilado, no exige que el traductor esté residente en memoria, como sucede con los intérpretes. Por ejemplo, entre los lenguajes que siempre son compilados se pueden destacar PASCAL, FORTRAN, COBOL, etc.

## Lista de lenguajes de programación:

### 1. Java

Es el más actualizado actualmente y esto se debe a su simplicidad y legibilidad. No en vano, más de 9 millones de usuarios lo usan y está presente en miles de millones de dispositivos, lo que significa un 15% del total en el mundo frente a otros lenguajes.

### 2. Lenguaje de programación C

Surgió en los años 70 y tenía un nombre que no cambió mucho: "B".

Con un alto porcentaje de uso al igual que java, 12%, es el segundo lenguaje más usado a nivel mundial. Su uso también es muy variado y se puede ejecutar en la mayoría de los sistemas operativos.

### 3. Python

Un lenguaje de programación multiplataforma y multiparadigma, que también tiene un propósito general. Esto significa que soporta la orientación a objetos, la programación imperativa y funcional.

### 4. C++

Es una evolución del antes mencionado C. Este, sin embargo, tiene un enfoque en desarrollo de aplicaciones y softwares más complejos, que exigen, por ejemplo, una intervención visual.

### 5. C#

Conocido también como "**C Sharp**" entre los programadores, también es una evolución de C y C++. C# es un lenguaje de programación orientado a objetos. Creado en el año 2000, se destaca por su simplicidad, trabaja con aplicaciones bajo el framework .net como el visual studio de Windows.

### 6. Visual Basic. NET

Ha tenido una rápida evolución en número de usuarios en los últimos años. Conocido por ser una herramienta mucho más amigable, que no exige tanto conocimiento como, por ejemplo, el C#.

Otros lenguajes no tan utilizados:

### 7. SQL

### 8. PHP

### 9. Ruby

### 10. Lenguaje de programación R

11. Rust
12. TypeScript
13. Swift
14. Perl
15. Lenguaje de programación Go
16. Kotlin
17. Scheme
18. Erlang
19. Elixir
20. Pascal
21. Postscript
22. Haskell
23. Objective-C
24. Scala
25. Lava
26. Fortran
27. Cobol
28. Logo
29. Prolog
30. Ada

## **Clasificación de los lenguajes de programación:**

### Lenguaje máquina

Es el sistema de códigos interpretable directamente por un circuito microprogramable, como el microprocesador de una computadora. Este lenguaje se compone de un conjunto de instrucciones que determinan acciones que serán realizadas por la máquina. Y un programa de computadora consiste en una cadena de estas instrucciones de lenguaje de máquina (más los datos). Normalmente estas instrucciones son ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas.

## Lenguajes de bajo nivel

Un lenguaje de programación de bajo nivel es el que proporciona poca o ninguna abstracción del microprocesador de una computadora. Consecuentemente, su traslado al lenguaje máquina es fácil. El término *ensamblador* (del inglés *assembler*) se refiere a un tipo de programa informático encargado de traducir un archivo fuente, escrito en un lenguaje ensamblador, a un archivo objeto que contiene código máquina ejecutable directamente por la máquina para la que se ha generado.

## Lenguaje de alto nivel

Los lenguajes de programación de alto nivel se caracterizan porque su estructura semántica es muy similar a la forma como escriben los humanos, lo que permite codificar los algoritmos de manera más natural, en lugar de codificarlos en el lenguaje binario de las máquinas, o a nivel de lenguaje ensamblador

## **Paradigmas de la programación:**

### Lenguajes imperativos

Emplean instrucciones como unidad de trabajo de los programas (Cobol, Pascal, C, Ada).

### Lenguajes declarativos

Los programas se construyen mediante descripciones de funciones o expresiones lógicas (*Lisp*, *Prolog*).

### Lenguajes orientados a objetos

El diseño de los programas se basa más en los datos y su estructura. La unidad de proceso es el objeto y en él se incluyen los datos (variables) y operaciones que actúan sobre ellos (*Smalltalk*, C++).

### Lenguajes orientados al problema

Están diseñados para problemas específicos, principalmente de gestión; suelen ser generadores de aplicaciones.

### Lenguajes naturales

Están desarrollándose nuevos lenguajes con el objetivo de aproximar el diseño y construcción de programas al lenguaje de las personas.

## **Bibliografía:**

[https://www.acta.es/medios/articulos/informatica\\_y\\_computacion/034083.pdf](https://www.acta.es/medios/articulos/informatica_y_computacion/034083.pdf)

<http://wiki.elhacker.net/programacion/introduccion/historia-de-la-programaci%C3%B3n>

<https://www.gestiopolis.com/historia-de-la-computacion-y-la-informatica/>

<https://rockcontent.com/es/blog/tipos-de-lenguaje-de-programacion/>

[https://www.ecured.cu/Paradigmas\\_de\\_programaci%C3%B3n](https://www.ecured.cu/Paradigmas_de_programaci%C3%B3n)

[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html)