



**Universidad Nacional Autónoma
de México**
Facultad de Ingeniería



Materia: Fundamentos de Programación

Profesor(a): Marco Antonio Martínez Quintana

Semestre 2021-1

Alumno: Castro Rodríguez Ulises

Email de contacto: ulicastro5@gmail.com

Tarea 1: Historia de la Programación

Fecha: 14/10/2020

Historia del cómputo

La primera computadora fue la *máquina analítica* creada por *Charles Babbage*, profesor matemático de la Universidad de Cambridge e Ingeniero Ingles en el siglo XIX. La idea que tuvo Charles Babbage sobre un computador nació debido a que la elaboración de las tablas matemáticas era un proceso tedioso y propenso a errores. Las características de esta máquina incluye una memoria que puede almacenar hasta 1000 números de hasta 50 dígitos cada uno. Las operaciones a ejecutar por la unidad aritmética son almacenados en una *tarjeta perforadora*. Se estima que la máquina tardaría un segundo en realizar una suma y un minuto en una multiplicación.



En 1944 se construyó en la *Universidad de Harvard*, la *Mark I*, diseñada por un equipo encabezado por *Howard H. Aiken*. Este computador tomaba seis segundos para efectuar una multiplicación y doce para una división. Computadora basada en rieles (tenía aprox. 3000), con 800 kilómetros de cable, con dimensiones de 17 metros de largo, 3 metros de alto y 1 de profundidad. Al Mark I se le hicieron mejoras sucesivas, obteniendo así el *Mark II*, *Mark III* y *Mark IV*.



En 1947 se construyó en la Universidad de Pennsylvania la *ENIAC* (Electronica Numerical Integrator And Calculator) que fue la *primera computadora electrónica* que funcionaba con tubos al vacío, el equipo de diseño lo encabezaron los ingenieros *John Mauchly* y *John Eckert*. Este computador supera ampliamente al *Mark I*, ya que llegó 1500 veces más potente. En el diseño de este computador fueron incluidas nuevas técnicas de electrónica que permitían

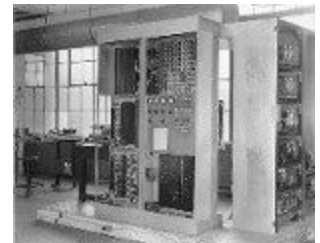


minimizar el uso de partes mecánicas. Esto trajo como consecuencia un incremento significativo en la velocidad de procesamiento. Así , podía efectuar 5000 sumas o 500 multiplicaciones en un segundo y permitía el uso de aplicaciones científicas en astronomía , meteorología, etc.

Durante el desarrollo del proyecto ENIAC, el matemático Von Neumann propuso unas mejoras que ayudaron a llegar a los modelos actuales de computadoras:

- 1.- Utilizar un sistema de numeración de base dos (Binario) en vez del sistema decimal tradicional.
- 2.- Hacer que las instrucciones de operación estén en la memoria , al igual que los datos. De esta forma , memoria y programa residirán en un mismo sitio.

La *EDVAC* (Electronic Discrete Variable Automatic Computer), construida en la Universidad de Manchester, en Connecticut (EE.UU), en 1949 fue el primer equipo con capacidad de almacenamiento de memoria e hizo desear a los otros equipos que tenían que ser intercambios o reconfigurados cada vez que se usaban. Tenía aproximadamente cuatro mil bulbos y usaba un tipo de memoria basado en tubos llenos de mercurio por donde circulaban señales eléctricas sujetas a retardos. EDCAV pesaba aproximadamente 7850 kg y tenía una superficie de 150 m².



En realidad EDVAC fue la primera verdadera computadora electrónica digital de la historia, tal como se le concibe en estos tiempos y a partir de ella se empezaron a fabricar arquitecturas más completas.



El *UNIVAC* fue la primera computadora diseñada y construida para un propósito no militar. Desarrollada para la oficina de *CENSO* en 1951, por los ingenieros *John Mauchly* y *John Presper Eckert*, que empezaron a diseñar y construirla en 1946.

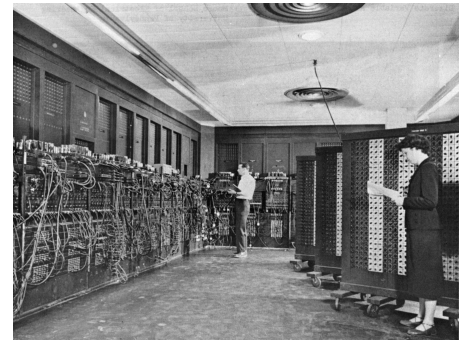
La computadora pesaba 7257 kg aproximadamente, estaba compuesta por 5000 tubos de vacío, y podía ejecutar unos 1000 cálculos por segundo.

Era una computadora que procesaba los dígitos en serie. Podía hacer sumas de dos números de diez dígitos cada uno, unas 100000 por segundo.

Así *Von Neumann*, junto con *Babbage* se consideran hoy como los *padres de la Computación*.

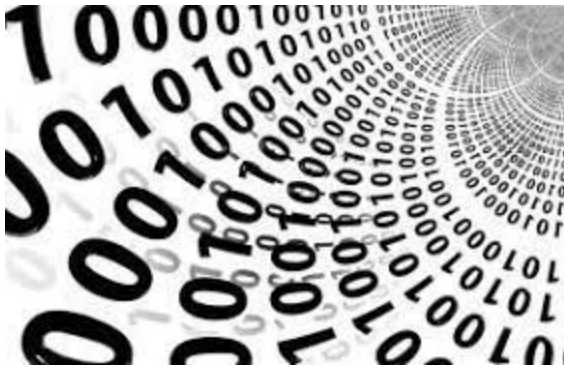
Historia de la Programación

La historia de la Programación está relacionada directamente con la aparición de los computadores, que ya desde el siglo XV tuvo sus inicios con la construcción de una máquina que realizaba operaciones básicas y raíces cuadradas (Gottfried Wilheml von Leibniz); aunque en realidad la primera gran influencia hacia la creación de los computadores fue la máquina diferencial para el cálculo de polinomios, proyecto no concluido de Charles Babbage con el apoyo de Lady Ada Countess of Lovelace, primera persona que incursionó en la programación y de quien proviene el nombre del lenguaje de programación ADA creado por el DoD (Departamento de defensa de Estados Unidos) en la década de 1970.



Los avances en las ciencias informáticas han sido muy acelerados, se reemplaza los tubos al vacío por transistores en 1958 y en el mismo año, se sustituyeron por circuitos integrados, y en 1961 se miniaturizan en chips de silicio. En 1971 apareció el primer microprocesador de Intel; y en 1973 el primer sistema operativo CP/M. El primer computador personal fue comercializado por IBM en el año 1980.

De acuerdo a este breve viaje por la historia, la programación está vinculada a la aparición de los computadores, y los lenguajes tuvieron también su evolución. Inicialmente se programaba en código binario, es decir en cadenas de 0s y 1s, que es el lenguaje que entiende directamente el computador, tarea extremadamente difícil; luego se creó el lenguaje ensamblador, que aunque era lo mismo que programar en binario, al estar en letras era más fácil de recordar. Posteriormente aparecieron lenguajes de alto nivel, que en general, utilizan palabras en inglés, para dar las órdenes



a seguir, para lo cual utilizan un proceso intermedio entre el lenguaje máquina y el nuevo código llamado código fuente, este proceso puede ser un compilador o un intérprete.

Un compilador lee todas las instrucciones y genera un resultado; un intérprete ejecuta y genera resultados línea a línea. En cualquier

caso han aparecido nuevos lenguajes de programación, unos denominados estructurados y en la actualidad en cambio los lenguajes orientados a objetos y los lenguajes orientados a eventos.

Lenguajes de Programación

- **Java**

Desarrollado en 1995 por Sun Microsystems, su principal punto fuerte es que **es muy versátil**. Puede usarse prácticamente para todo, desde aplicaciones webs a programas de ordenadores, apps de Android, juegos, etc.

Otra de sus ventajas es que **es independiente del hardware**. Solo tienes que escribir la aplicación una vez, y funcionará en cualquier ordenador o móvil, mediante un intérprete de Java. Así que de lo único que debes preocuparte es de programar. Java es un lenguaje estable y seguro, y existe mucha documentación, así que es fácil de aprender. Es **un lenguaje orientado a objetos**, una forma de programar más sencilla e intuitiva en donde se trabaja con entidades que hacen una función y manejan unos datos. Se trata de relacionar esas entidades, según los datos, para crear los programas.



- **Javascript**

Javascript es el lenguaje más utilizado para escribir aplicaciones dentro de las **páginas webs**. Funciona directamente en el navegador, así que lo único que necesita

para usar Javascript es escribir el programa como un documento de texto, y abrirlo en el navegador como si fuese una página web. El navegador se encargará de ponerlo en marcha. Es un lenguaje jerarquizado y muy bien estructurado, en donde resulta sencillo seguir el flujo del programa.



Se trata de un lenguaje orientado a objetos, lo que significa que hay una serie de bloques de programación que aceptan ciertos datos, y ofrecen ciertos resultados. Estos objetos son creados por los programadores y compartidos para que todo el mundo pueda usarlos. Así que muchas funciones y herramientas ya están programadas, y simplemente tienes que añadirlas a tus aplicaciones.

- **Python**

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).



Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

- **Lenguaje C/C++**

Fue creado nada menos que en 1969 por Dennis Ritchie, cuando no existía la informática doméstica y solo algunas empresas privilegiadas tenían ordenadores.

Es un lenguaje que se creó para diseñar sistemas operativos, como Unix y otros muchos. Pero al tratarse de un lenguaje de bajo nivel (muy cercano al hardware de la máquina, pero más difícil de aprender), y muy rápido y eficiente (si está bien escrito), también comenzó a usarse para escribir aplicaciones y programas. Buena parte de los programas comerciales que usan las grandes empresas, incluyendo bases de datos, programas de contabilidad, etc., están escritos en C. Y como cuesta mucho sustituirlos por completo, el lenguaje C es muy solicitado para mantener o mejorar aplicaciones ya existentes. Por eso es el más usado.

Otra de las ventajas del lenguaje C es que **muchos otros lenguajes se basan en él, incluido Java y Javascript.**

La variante más popular es **el lenguaje C++**, también muy utilizado. Es un lenguaje orientado a objetos, así que resulta algo más sencillo de usar, y reutilizar funciones y código. También es más fácil de depurar. Es más fácil de aprender que C, y está disponible para todas las plataformas. Se usa mucho para crear bases de datos, sistemas operativos, videojuegos, y mucho más.



- **HTML**

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.

El lenguaje consta de etiquetas que tienen esta forma o <P>. Cada etiqueta significa una cosa, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo, <A> es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta. Por ejemplo se utiliza para indicar que se deje de escribir en

negrita. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir el contenido del documento y algún estilo básico. `Esto está en negrita`.



Clasificación de los lenguajes de Programación

- **Clasificación histórica**

A medida que surgían nuevos lenguajes que permitían nuevos estilos de programación más expresiva, se distinguieron dichos estilos en una serie de **generaciones**, cada una representando lenguajes de programación surgidos en una época similar y con características genéricas comunes.

- **Lenguajes de alto y de bajo nivel**

Los lenguajes de programación se suelen clasificar dentro de dos amplias categorías que se refieren a su "nivel de abstracción", es decir, en cuanto a lo específico o general que es respecto a la arquitectura de computación inherente al sistema que se está utilizando.

- **Clasificación por paradigmas**

Los paradigmas de programación distinguen distintos modelos de cómputo y de estilos de estructurar y organizar las tareas que debe realizar un programa. Un lenguaje de programación puede ofrecer soporte a uno o varios paradigmas de programación, total o parcialmente.

- **Clasificación por propósito**

Se distinguen los lenguajes de programación de **propósito general** de aquellos de **propósito específico**.

Tipos de paradigmas de Programación

Un paradigma de programación es una manera o estilo de programación de software. Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores. Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales. Los lenguajes de programación adoptan uno o varios paradigmas en función del tipo de órdenes que permiten implementar.

- **Paradigma Imperativo**

Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas. El desarrollador describe en el código paso por paso todo lo que hará su programa.

Algunos lenguajes: **Pascal, COBOL, FORTRAN, C, C++**, etc.

Otros enfoques subordinados al paradigma de programación imperativa son:

- ❑ **Programación estructurada:** La programación estructurada es un tipo de programación imperativa donde el flujo de control se define mediante bucles anidados, condicionales y subrutinas, en lugar de a través de GOTO.
- ❑ **Programación procedimental:** Este paradigma de programación consiste en basarse en un número muy bajo de expresiones repetidas, englobadas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse.
- ❑ **Programación modular:** consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más manejable y legible. Se trata de una evolución de la programación estructurada para resolver problemas de programación más complejos.

- **Paradigma Declarativo**

Este paradigma no necesita definir algoritmos puesto que describe el problema en lugar de encontrar una solución al mismo. Este paradigma utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas.

Este paradigma a su vez se divide en dos:

- ❑ **Programación Lógica**

- ❑ **Programación funcional**

- **Programación orientada a objetos**

En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaremos las cosas en la vida real.

Podemos definir un objeto como una estructura abstracta que, de manera más fiable, describe un posible objeto del mundo real y su relación con el resto del mundo que lo rodea a través de interfaces.

- **Programación Reactiva**

Este Paradigma se basa en escuchar lo que emite un evento o cambios en el flujo de datos, en donde los objetos reaccionan a los valores que reciben de dicho cambio. Las librerías más conocidas son Project Reactor, y RxJava. React/Angular usan RxJs para hacer uso de la programación reactiva.

Bibliografía

- Alvarez, M. A. (2001, 1 enero). *Qué es HTML*. Desarrollo Web. <https://desarrolloweb.com/articulos/que-es-html.html>
- Cervantes, N. C., & Pineda, C. P. (s. f.). *Fundamentos de programación-Un poco de historia*. GPL. Recuperado 4 de octubre de 2020, de http://www.utn.edu.ec/reduca/programacion/fundamentos/un_poco_de_historia.html
- colaboradores de Wikipedia. (2020a, septiembre 14). *Python*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Python#Caracter%C3%ADsticas_y_paradigmas
- colaboradores de Wikipedia. (2020b, septiembre 28). *Lenguaje de programación*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Clasificaci%C3%B3n_de_los_lenguajes_de_programaci%C3%B3n
- *Historia de la Computación*. (s. f.). Historia de la Computación. Recuperado 4 de octubre de 2020, de http://www.cad.com.mx/historia_de_la_computacion.htm#:~:text=La%20primera%20computadora%20fue%20la,Ingles%20en%20el%20siglo%20XIX.&text=La%20idea%20que%20tuvo%20Charles,tedioso%20y%20propenso%20a%20errores.
- Martínez, M. M. C. (s. f.). *¿Qué son los paradigmas de programación?* Profile. Recuperado 4 de octubre de 2020, de <https://profile.es/blog/que-son-los-paradigmas-de-programacion/>

- Pascual, J. A. (2020, 9 mayo). *Los lenguajes de programación más populares del mundo*. ComputerHoy.
<https://computerhoy.com/listas/industria/lenguajes-programacion-mas-populares-633547>