



***UNIVERSIDAD NACIONAL  
AUTONOMA DE MÉXICO***



*Facultad de Ingeniería*

***Fundamentos de  
programación***

***Tarea asincrónica #1***

Cuevas Antúñez Samantha

No. lista: 12

04/10/2020

# Historia del computo

Primero que nada, es primordial el conocer el concepto de *computadora*, según la RAE<sup>1</sup> se define como; 'Máquina electrónica capaz de realizar un tratamiento automático de la información y de resolver con gran rapidez problemas matemáticos y lógicos mediante programas informáticos', por otro lado un *ordenador* es una máquina formada por multitud de elementos físicos, la mayoría de ellos de carácter electrónico (placas, circuitos integrados, chips), capaz de aceptar información, manipularla mediante operaciones aritméticas y lógicas a gran velocidad y devolver una serie de resultados. Todas estas operaciones se realizan sin la intervención directa de las personas y controladas por un programa o conjunto de instrucciones que previamente ha sido almacenado en el ordenador. Una vez definidos dichas palabras claves es prudente el hablar de sus antecedentes.

## Precedentes:

- El ábaco.  
Tiene **5.000 años de antigüedad**, por lo cual es posiblemente el primer dispositivo mecánico de contabilidad de la historia, inclusive aún se sigue usando. Consta de un marco rectangular con cuentas, al desplazar cada cuenta al otro extremo se contabiliza dicha cuenta (la cuenta representa lo que se está contabilizando).
- Calculadoras mecánicas.  
En **1642 Blaise Pascal** <sup>2</sup> construyó la primera sumadora mecánica que llamó *Pascalina*, funcionaba con un complicado mecanismo de engranes y ruedas: la rotación completa de una de las ruedas dentadas hacía girar un paso a la rueda siguiente. La Pascalina sólo realizaba sumas y restas, pero está fue un fracaso ya que resultaba más costosa que hacer las operaciones manualmente.  
**Gottfried Wilhelm Von Leibniz** <sup>3</sup> diseñó en **1671** otra sumadora mecánica, que concluyó definitivamente en 1694, conocida como la Calculadora

---

<sup>1</sup> La Real Academia Española (RAE) es una institución cultural con sede en Madrid (España).

<sup>2</sup> Filósofo y matemático francés (1623-1662)

<sup>3</sup> Alemán (1646-1716)

Universal o Rueda Escalada de Leibniz, capaz de realizar sumas, restas, divisiones y raíces cuadradas.

A partir de estas creaciones comenzaron a darse nuevos modelos de calculadoras mecánicas, con mejoras y variaciones.

- **Primera tarjeta perforada.**

**Joseph-Marie Jacquard**<sup>4</sup> ideó en **1801** un telar, todavía utilizado en la actualidad, que podía reproducir automáticamente patrones de tejidos leyendo la información codificada en patrones de agujeros perforados en tarjetas de papel rígido. Las tarjetas se perforaban estratégicamente y se acomodaban en cierta secuencia para indicar un diseño de tejido en particular. Para cambiar de diseño, basta con preparar una nueva colección de tarjetas.

El telar consta de una serie de varillas, sobre las que pasan las tarjetas, y de las que están prendidos hilos de distintos colores. Las perforaciones de las tarjetas determinan de manera mecánica qué varillas intervienen en la formación del tejido y en la disposición de los dibujos.

El telar de Jacquard supone una triple aportación teórica para el futuro desarrollo computacional:

- ✓ Proporciona un modelo de automatización de los procesos de producción diversificada.
- ✓ Por primera vez se realiza una codificación de la información. Las tarjetas son la información suministrada (*input*)<sup>5</sup> y el tejido es el resultado (*output*).<sup>6</sup>
- ✓ **Por primera vez se realiza la programación** de las instrucciones. La cadena de tarjetas perforadas prefigura la organización de los procesos mediante técnicas de programación.

- Maquina analítica de Babbage

**Charles Babbage** (1791-1871), visionario científico y matemático inglés, fue el más claro precursor del hardware computacional, hasta el punto de que se le considera el **padre histórico de la computación**:

- ✓ Ideó la Máquina de Diferencias (*Difference Engine*), cuyo modelo definitivo es de **1823**, capaz de calcular e imprimir tablas matemáticas de hasta veinte cifras con ocho decimales y polinomios de sexto grado.

---

<sup>4</sup> Fabricante de tejidos francés (1752-1834)

<sup>5</sup> La palabra input traducido al español expresa "entrada"

<sup>6</sup> El término output traducido al español significa "salida"

- ✓ **1834** concibió su revolucionaria **Máquina Analítica** que puede considerarse un auténtico prototipo decimonónico de ordenador; era una calculadora polivalente con capacidad para operar de forma distinta según el problema que se le planteara. En dicha máquina aparecen ya los elementos básicos de los modernos ordenadores: dispositivos de entrada y de salida, unidad de control, unidad lógico-aritmética y memoria. La programación se debía realizar mediante fichas perforadas.
- ✓ Sus proyectos no dieron más allá, debido a la época, problemas económicos e industriales y falta de tecnología.
- ✓ En **1843**, **Lady Ada Augusta Lovelace**, estrecha colaboradora de Babbage, sugirió la idea de que las tarjetas perforadas pudieran adaptarse de manera que propiciaran que el motor de Babbage repitiera ciertas operaciones. Debido a esta sugerencia algunas personas consideran a Lady Lovelace la **primera programadora**.

- Primeras computadoras

#### 1943. Howard Aiken

"Automatic Sequence Controlled Calculator". Este artefacto era de 51 pies de largo, 8 pies de altura y 2 pies de espesor; contaba con 750,000 partes y 500 millas de cable; y su peso era de 5 toneladas. Era muy ruidosa, pero capaz de realizar tres calculaciones por segundo. Este computador, aceptaba tarjetas perforadas, las cuales eran luego procesadas y almacenadas esta información. Los resultados eran impresos en una maquinilla eléctrica. Esta primera computadora electromecánica fue la responsable de hacer a IBM un gigante en la tecnología de las computadoras.



#### 1939 John Atanasoff

En la Universidad de Iowa State, John Atanasoff diseñó y construyó la primera computadora digital mientras trabajaba con Clifford Berr, un estudiante graduado. Esta computadora, completada en el 1942, usaba circuitos lógicos binarios y tenía memoria regenerativa.



#### 1946 Dr. John Mauchly y J. Presper Eckert

Surgió una computadora electrónica digital operacional, llamada ENIAC (Electronic Numerical Integrator And Computer). Esta máquina fue desarrollada a gran escala, siendo derivada de las ideas no patentadas de Atanasoff. Este aparato trabajaba con el sistema decimal y tenía todas las características de las computadoras de hoy día. Las dimensiones de la ENIAC eran inmensas, ocupando un espacio de 30 X 50 pies, un peso de 30 toneladas, y un consumo de 160 kilovatios de potencia. Conducía electricidad a través de 18,000 tubos de vacío, generando un calor inmenso; contaba con un aire acondicionado especial para mantenerla fría.

### 1945 John Von Newmann

Como resultado de la colaboración del equipo de Moore, surgió la EDVAC. Hasta este momento, la computadora almacenaba sus programas externamente, ya fuera en tarjetas conectadas, cintas perforadas y tarjetas. Mauchly y Eckert descubrieron que una línea de demora de mercurio podría reemplazar docenas de estos tubos al vacío. Ellos figuraron que las líneas de demoras significarían ahorros gigantescos en los costos de los tubos y espacio de memoria. Este avance contribuyó a la creación de la computadora EDVAC. El EDVAC almacenaba información en memoria en la misma manera que los datos. La máquina, entonces, manipulaba la información almacenada.

Antes del 1951, las computadoras no fueron manufacturadas a grande escala. En el 1951, con la llegada del UNIVAC, comienza la era de las computadoras comerciales. Con tan solo dentro de tres años, IBM comenzó a distribuir su IBM 701 y otras compañías manufacturaron computadoras, tal como el Burroughs E. 101 y el Honeywell Datamatic 1000. Las computadoras que fueron desarrolladas durante los años 1950 y 1960 se conocieron como las computadoras de la primera generación porque tenía una característica en común, el tubo en vacío.



*Diagrama de flujo 1. Muestra las primeras computadoras que se desarrollaron.*

## Historia de la programación

**Programación:** es el acto de programar, es decir, organizar una secuencia de pasos ordenados a seguir para hacer cierta cosa.

El lenguaje de programación es un idioma artificial prediseñado formado por signos, palabras y símbolos que permite la comunicación entre el programador y el ordenador. Las instrucciones que sigue el ordenador para la ejecución de aplicaciones y programas están escritas en lenguaje de programación y luego son traducidas a un lenguaje de máquina que puede ser interpretado y ejecutado por el hardware del equipo.

El código fuente está formado por líneas de texto que expresan en lenguaje de programación las instrucciones que debe llevar a cabo el ordenador. Este código es creado, diseñado, codificado, mantenido y depurado a través de la

programación. Existen diferentes lenguajes de programación (Java, Pearl, Python) que se valen de diversos programas en los que se vuelcan las instrucciones. Estos lenguajes varían con el tiempo, se expanden y evolucionan.

### OBJETIVO

Definir instrucciones para que un ordenador pueda ejecutar sistemas, programas y aplicaciones que sean eficaces, accesibles y amigables para el usuario.

Los programas informáticos suelen seguir algoritmos, que son el conjunto de instrucciones organizadas y relacionadas entre sí que permiten trabajar al software de los equipos de computación.

7

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 y como ya vimos; 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<sup>8</sup> en la década de 1970.

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

---

<sup>7</sup> Algoritmo. Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.

<sup>8</sup> Departamento de defensa de Estados Unidos

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

## Elementos de la programación

Existen ciertos elementos que son clave a la hora de conocer o ejecutar un lenguaje de programación, entre los más representativos están:

- ❖ Palabras reservadas. Palabras que dentro del lenguaje significan la ejecución de una instrucción determinada, por lo que no pueden ser utilizadas con otro fin.
- ❖ Operadores. Símbolos que indican la aplicación de operaciones lógicas o matemáticas.
- ❖ Variables. Datos que pueden variar durante la ejecución del programa.
- ❖ Constantes. Datos que no varían durante la ejecución del programa.
- ❖ Identificadores. Nombre que se le da a las diferentes variables para identificarlas.

## Lista de los lenguajes de programación

Los tipos **más importantes** de lenguajes de programación son:

### ○ Lenguajes Imperativos

Su origen es la propia arquitectura de Von Neumann, que consta de una secuencia de celdas (memoria) en las cuales se pueden guardar datos e instrucciones, y de un procesador capaz de ejecutar de manera secuencial una serie de operaciones principalmente aritméticas y booleanas. En general, un lenguaje imperativo ofrece al programador conceptos que se traducen de forma natural al modelo de la máquina.

Ejemplos: FORTRAN, Algol, Pascal, C, Modula-2, Ada

- **Lenguajes Funcionales**

Los matemáticos resuelven problemas usando el concepto de función, que convierte datos en resultados. Sabiendo cómo evaluar una función, usando la computadora, podríamos resolver automáticamente muchos problemas. Además, se aprovechó la posibilidad que tienen las funciones para manipular datos simbólicos, y no solamente numéricos, y la propiedad de las funciones que les permite componer, creando de esta manera, la oportunidad para resolver problemas complejos a partir de las soluciones de Inteligencia en Redes de Comunicaciones.

También se incluyó la posibilidad de definir funciones recursivamente. Un lenguaje funcional ofrece conceptos que son muy entendibles y relativamente fáciles de manejar.

El lenguaje funcional más antiguo y popular es LISP, diseñado por McCarthy en la segunda mitad de los años 50. Se usa principalmente en Inteligencia Artificial. En los 80 se añadió a los lenguajes funcionales la tipificación y algunos conceptos modernos de modularización y polimorfismo, un ejemplo es el lenguaje ML. Programar en un lenguaje funcional significa construir funciones a partir de las ya existentes.

- **Lenguajes Lógicos**

El conocimiento básico de las matemáticas se puede representar en la lógica en forma de axiomas<sup>9</sup>, a los cuales se añaden reglas formales para deducir cosas verdaderas. Gracias al trabajo de algunos matemáticos, de finales de siglo pasado y principios de éste, se encontró la manera de automatizar computacionalmente el razonamiento lógico que permitió que la lógica matemática diera origen a otro tipo de lenguajes de programación, conocidos como lenguajes lógicos. También se conoce a estos lenguajes, y a los funcionales, como *lenguajes declarativos*, porque para solucionar un problema el programador solo tiene que describirlo con axiomas y reglas de deducción en el caso de la programación lógica y con funciones en el caso de la programación funcional.

En los lenguajes lógicos se utiliza el formalismo de la lógica para representar el conocimiento sobre un problema y para hacer preguntas que se vuelven teoremas si se demuestra que se pueden deducir a partir del conocimiento dado en forma de axiomas y de las reglas de deducción estipuladas. Así se encuentran soluciones a problemas formulados como preguntas. Con base

---

<sup>9</sup> Axioma. Proposición o enunciado tan evidente que se considera que no requiere demostración.



en la información expresada dentro de la lógica de primer orden, se formulan las preguntas sobre el dominio del problema y el intérprete del lenguaje lógico trata de encontrar la respuesta automáticamente. El conocimiento sobre el problema se expresa en forma de predicados (axiomas) que establecen relaciones sobre los símbolos que representan los datos del dominio del problema.

El PROLOG surgió a principio de los 70 y es el primer lenguaje lógico. Las aplicaciones en la Inteligencia Artificial lo mantienen vivo y útil.

En el caso de la programación lógica, el trabajo del programador es la buena descripción del problema en forma de hechos y reglas. A partir de ésta se pueden encontrar muchas soluciones dependiendo de cómo se formulen las preguntas, que tienen sentido para el problema. Si el programa está bien definido, el sistema encuentra automáticamente las respuestas a las preguntas formuladas. En programación lógica, al igual que en programación funcional, el programa, en este caso los hechos y las reglas, están muy alejados del modelo von Neumann que posee la máquina en la que tienen que ser interpretados; por lo que la eficiencia de la ejecución es inferior a la de un programa equivalente en lenguaje imperativo. Sin embargo, para cierto tipo de problemas, la formulación del programa mismo puede ser mucho más sencilla y natural.

- **Lenguajes Orientados a Objetos**

A mediados de los años 60 se empezó a usar las computadoras para la simulación de problemas del mundo real. Pero el mundo real está lleno de objetos, en la mayoría de los casos complejos, los cuales difícilmente se traducen a los tipos de datos primitivos de los lenguajes imperativos. Así surgió el concepto de objeto y sus colecciones, permitieron introducir abstracciones de datos a los lenguajes de programación. La posibilidad de reutilización del código y sus indispensables modificaciones, se reflejaron en la idea de las jerarquías de herencia de clases. También surgió el concepto de polimorfismo introducido vía procedimientos virtuales. Todos estos conceptos fueron presentados en el lenguaje Simula 67, desde el año 1967, aunque este lenguaje estaba enfocado a aplicaciones de simulación discreta.

Fue en los años 80 cuando surgieron lenguajes de programación con conceptos de objetos encabezada por Smalltalk, C++, Eiffel, Modula-3, Ada 95 y terminando con Java.

El modelo de objetos, y los lenguajes que lo usan, parecen *facilitar* la construcción de sistemas o programas en forma modular. Los objetos ayudan

a expresar programas en términos de abstracciones del mundo real, lo que aumenta su comprensión. La clase ofrece cierto tipo de modularización que facilita las modificaciones al sistema. La reutilización de clases previamente probadas en distintos sistemas también es otro punto a favor. Los programas en lenguajes orientados a objetos son *ineficientes*, en tiempo y memoria, contra los programas equivalentes en lenguajes imperativos, aunque les ganan en la comprensión de código.

- **Lenguajes Concurrentes, Paralelos y Distribuidos**

El origen de los conceptos para el manejo de concurrencia, paralelismo y distribución está en el deseo de aprovechar al máximo la arquitectura von Neumann y sus modalidades reflejadas en conexiones paralelas y distribuidas. Esto fue un tema importante sobre todo cuando las computadoras eran caras y escasas; el sistema operativo tenía que ofrecer la ejecución concurrente y segura de programas de varios usuarios, que desde distintos terminales utilizaban un solo procesador, y así surgió la necesidad de introducir algunos conceptos de programación concurrente para programar los sistemas operativos.

Cuando los procesadores cambiaron de tamaño y de precio, se abrió la posibilidad de contar con varios procesadores en una máquina y ofrecer el procesamiento en paralelo (procesar varios programas al mismo tiempo). Esto dio el impulso a la creación de “Historia de la programación” Inteligencia en Redes de Comunicaciones -4- Jessica Rivero Espinosa lenguajes que permitían expresar el paralelismo. Finalmente, llegaron las redes de computadoras, que también ofrecen la posibilidad de ejecución en paralelo, pero con procesadores distantes, lo cual conocemos como la programación distribuida. Históricamente encontramos soluciones conceptuales y mecanismos (semáforos, regiones críticas, monitores, envío de mensajes, llamadas a procedimientos remotos (RPC)) que se incluyeron después en lenguajes de programación como Concurrent Pascal o Modula (Basados en monitores), Ada ó SR (Basada en Rendez Vous (Tipo de RPC)), ALGOL 68(Semáforos), OCCAM (Envío de mensajes), Java, etc.

Otros tipos de lenguajes de programación son:

- Procedural Language
- Declarative Language
- Applicative Language

- Definitional Language
- Single Assignment Language
- Dataflow Language
- Constraint Language
- Lenguaje de cuarta generación(4GL)
- Query Language
- Specification Language
- Assembly Language
- Intermediate Language
- Metalenguajes

## *Clasificación de los programas de programación*

Los circuitos microprogramables son sistemas digitales, lo que significa que trabajan con dos únicos niveles de tensión simbolizados con el cero (0) y el uno (1). Por eso, el lenguaje de máquina utiliza sólo dichos signos.

- Un lenguaje de **bajo nivel** es trasladado fácilmente a lenguaje de máquina (la palabra bajo se refiere a la abstracción reducida entre el lenguaje y el hardware).
- Los lenguajes de programación de **alto nivel** se caracterizan por expresar los programas de una manera sencilla.

### **Lenguaje máquina.**

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.

### C++

Fue diseñado a mediados de la década de 1980 por Bjarne Stroustrup. Abarca dos paradigmas de la programación: la estructurada y la orientada a objetos.

### Fortran

Fue desarrollado en la década de 1950 y es empleado activamente desde entonces. Acrónimo de *formula translator*, Fortran se utiliza principalmente en aplicaciones científicas y análisis numérico.

### Python

Fue hecho por Guido van Rossum en 1990. En la actualidad se desarrolla como un proyecto de código abierto administrado por la *Python Software Foundation*. La última versión estable del lenguaje es la 2.5 (septiembre 2006).

### Java

Lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principios de 1990. Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible.

## 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.







### Perl

El lenguaje práctico para la extracción e informe es este lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del C, del lenguaje interpretado *shell* *sh*, *AWK*, *sed*, *Lisp* y, en grado inferior, de muchos otros lenguajes de programación.

# Tipos de paradigmas de programación

**Paradigmas de programación:** propuesta tecnológica que es adoptada por una Comunidad de Programadores cuyo núcleo central es incuestionable en cuanto a que unívocamente trata de resolver uno o varios problemas claramente delimitados. La resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de Software. Tiene una estrecha relación con la formalización de determinados lenguajes en su momento de definición. Un paradigma de programación está delimitado en el tiempo en cuanto a aceptación y uso ya que nuevos paradigmas aportan nuevas o mejores soluciones que la sustituyen parcial o totalmente.

## Tipos

-  Programación Imperativa
-  Programación funcional
-  Programación lógica
-  Declarativo
-  POO Orientado a Objetos
-  Por procedimientos

Habitualmente se mezclan todos los tipos de paradigmas a la hora de hacer la programación. De esa manera se origina la programación multiparadigma, pero el que actualmente es más usado de todos esos paradigmas es el de la programación orientada a objetos.

# Bibliografía

- ✓ Espinosa, J. R. (2012). Historia de la Programación. *Obtenido de Historia de la Programación:*  
[http://www.it.uc3m.es/jvillena/irc/practicas/estudios/Lenguajes\\_de\\_Programacion.pdf](http://www.it.uc3m.es/jvillena/irc/practicas/estudios/Lenguajes_de_Programacion.pdf).
- ✓ <https://www.google.com/search?q=axioma+definicion&oq=axioma+de&aqs=chrome..69i57j0l5.3023j1j7&sourceid=chrome&ie=UTF-8>
- ✓ <https://concepto.de/programacion/>
- ✓ <https://www.google.com/search?q=definicion+de+algoritmo&oq=definicion+de+algoritmo&aqs=chrome..69i57j0l7.9982j1j7&sourceid=chrome&ie=UTF-8>
- ✓ [https://www.google.com/search?xsrf=ALeKk00dstLuWUrRdZMfnWPvE33AN4Zupg%3A1601865451204&ei=64Z6X5SHDNq1tAaw5oylAQ&q=input+y+output+definicion&oq=input+y+output+def&gs\\_lcp=CgZwc3ktYWlQARgAMgcIABAUElcCMgIIADICCAyAggAMgYIABAWEb4yBggAEByQHjIGCAAQFhAeMgYIABAWEb4yBggAEByQHjIGCAAQFhAeOgQIABBDUNYQWJQcYLskaABwAngAgAHUCIgbVhCSAQM3LTKYAQCgAQGqAQdnd3Mtd2l6yAEIwAEB&sclien t=psy-ab](https://www.google.com/search?xsrf=ALeKk00dstLuWUrRdZMfnWPvE33AN4Zupg%3A1601865451204&ei=64Z6X5SHDNq1tAaw5oylAQ&q=input+y+output+definicion&oq=input+y+output+def&gs_lcp=CgZwc3ktYWlQARgAMgcIABAUElcCMgIIADICCAyAggAMgYIABAWEb4yBggAEByQHjIGCAAQFhAeMgYIABAWEb4yBggAEByQHjIGCAAQFhAeOgQIABBDUNYQWJQcYLskaABwAngAgAHUCIgbVhCSAQM3LTKYAQCgAQGqAQdnd3Mtd2l6yAEIwAEB&sclien t=psy-ab)
- ✓ <http://biblio3.url.edu.gt/Libros/provinciales/computadoras.pdf>
- ✓ <https://www.rae.es/dpd/computador>
- ✓ [http://www.utn.edu.ec/reduca/programacion/fundamentos/un\\_poco\\_de\\_historia.html](http://www.utn.edu.ec/reduca/programacion/fundamentos/un_poco_de_historia.html)
- ✓ <http://faa.unse.edu.ar/apuntes/ccaunidad1.pdf>
- ✓ [http://paginaspersonales.deusto.es/airibar/Ed\\_digital/INF/Intro/Historia.html](http://paginaspersonales.deusto.es/airibar/Ed_digital/INF/Intro/Historia.html)
- ✓ <https://www.uv.mx/personal/gerhernandez/files/2011/04/historia-compuesta.pdf>
- ✓ [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)
- ✓ [https://www.ecured.cu/Paradigmas\\_de\\_programaci%C3%B3n](https://www.ecured.cu/Paradigmas_de_programaci%C3%B3n)