



El futuro digital
es de todos

MinTIC



Lenguajes

Comunicación, lenguajes
de programación y
generaciones de los
computadores

Research Group on Artificial Life
Grupo de investigación en vida artificial (Alife)
Computer and System Department
Engineering School
Universidad Nacional de Colombia

Jonatan Gomez Perdomo, Ph. D.
jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.
aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D. (c)
eccubidesg@unal.edu.co

Carlos Andrés Sierra, M.Sc.
casierrav@unal.edu.co



UNIVERSIDAD
NACIONAL
DE COLOMBIA



Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

Clasificación de los lenguajes de programación

Paradigmas de programación

4

Generaciones de los computadores



Comunicación

Definición

La comunicación es un proceso mediante el cual se transmite la información. Para que haya comunicación, debe haber una coincidencia en el tipo de lenguaje, es decir que, para codificar la información se requiere seguir unos lineamientos o reglas preestablecidas por un sistema con el cual está familiarizado tanto emisor como receptor. El emisor codifica, el receptor decodifica.

Los elementos de la comunicación I

Los elementos de la comunicación son:

- Fuente de información: Genera la información que será transmitida.
- Mensaje: Dato o conjunto de datos a transmitir. Surge de la selección de posibilidades en un conjunto de combinaciones simbólicas posibles.
- Emisor: Codifica el mensaje.

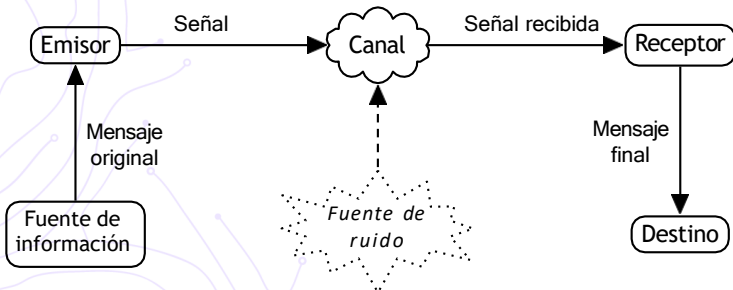


Figure: Modelo matemático de la comunicación de Claude Elwood Shannon (1948)



Los elementos de la comunicación II

Los elementos de la comunicación son:

- Señal: Signos o símbolos del sistema convencional.
- Canal: Medio por el cual se transmite el mensaje codificado.
- Fuente de ruido: interferencia que distorsiona la señal y puede cambiar el mensaje.

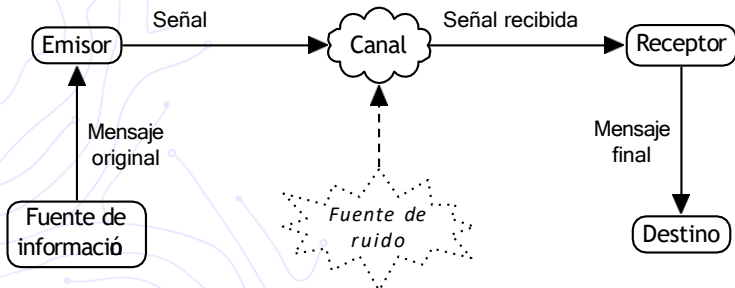


Figure: Modelo matemático de la comunicación de Claude Elwood Shannon (1948)

Los elementos de la comunicación III

Los elementos de la comunicación son:

- Receptor: Decodifica para poder ser recibido por el destino
- Destino: Ente al que se dirige el mensaje.

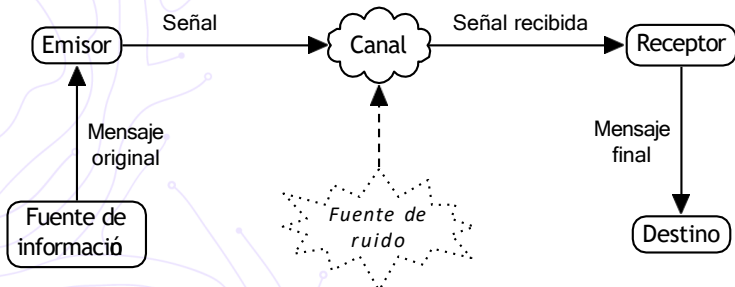


Figure: Modelo matemático de la comunicación de Claude Elwood Shannon (1948)



Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

Clasificación de los lenguajes de programación

Paradigmas de programación

4

Generaciones de los computadores



Lenguaje

Definición (Lenguaje)

Un lenguaje está formado por tres elementos (el léxico, la sintaxis y la semántica), que permiten expresar y comunicar información entre entes, ya sean personas, animales, computadores, etc.



Componentes del lenguaje I

Léxico

El léxico de un lenguaje lo conforman las unidades mínimas con significado completo. A cada uno de estas unidades mínimas con significado se le conoce como *lexema*.

Ejemplo

En castellano, las palabras y los símbolos de puntuación (que son usados para formar frases, oraciones y párrafos) conforman el léxico. A tales lexemas se les asocia un significado preciso en términos de las frases construidas con ellos.



Componentes del lenguaje II

Sintaxis I

La sintaxis de un lenguaje explica la forma en que se pueden construir frases en el lenguaje a partir del léxico. Usualmente la sintaxis se presenta como una colección de reglas de reescritura que se definen con una gramática. Estas son reglas que indican como unos símbolos de la gramática pueden ser reescritos por otros símbolos de la gramática o por lexemas. La idea es que al final del proceso de reescritura sólo se tengan lexemas.

Componentes del lenguaje III

Sintaxis II

Ejemplo

Por ejemplo en castellano una *frase* se puede reescribir como un *sujeto* y un *predicado*, a su vez un *sujeto* se puede reescribir como un *artículo* y un *sustantivo*; un *predicado* se puede reescribir como un *verbo* y un *complemento* y así sucesivamente.

La derivación de la frase

"El profesor hará un examen muy difícil"

se puede modelar mediante un árbol de derivación como el de la siguiente figura.

Componentes del lenguaje IV

Sintaxis III

Ejemplo (continuación)

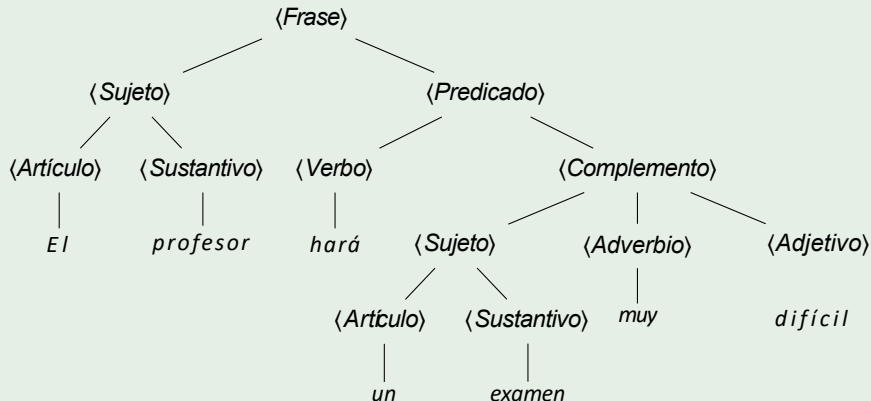


Figure: Árbol de derivación



Componentes del lenguaje V

Semántica

La semántica de un lenguaje define la forma en que se le asocia significado (sentido) a las frases construidas mediante la gramática. En castellano la semántica no es fácil de definir ya que intervienen elementos muy elaborados que han sido contruidos de manera natural a través del tiempo (cada objeto/idea conocido(a) por el ser humano esta asociado(a) con una palabra).

Componentes del lenguaje VI

Semántica

El sentido de una frase o una oración en castellano depende mucho del contexto en el que se escribe o dice la frase y del posible conjunto de significados el cual es muy grande. Este hecho es lo que hace difícil, para los computadores actuales, trabajar directamente en lenguaje natural.

Ejemplo

¿Cuál es el significado de la palabra “*mañana*” en la siguiente frase?:

Hasta mañana en la mañana el profesor Mañana viajará y regresará pasado mañana.

Ejemplo

¡Me estás tomando del pelo!.

Componentes del lenguaje VII

Semántica

La ambigüedad es una característica de los lenguajes naturales en general, por ejemplo en inglés

Ejemplo

- *It's raining cats and dogs*
- *Are you pulling my leg?*

Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

- Clasificación de los lenguajes de programación
- Paradigmas de programación

4

Generaciones de los computadores



Lenguajes de programación I

“Odio esta máquina, desearía que la vendieran, no hace lo que quiero que haga, sólo hace lo que le digo.”

Lamento de un programador (Anónimo)



Lenguajes de programación II

Para ordenarle a una máquina de cómputo (computador) que ejecute cierto procedimiento o realice un cálculo predeterminado, se dispone de los lenguajes de programación, los cuales son definidos a partir del lenguaje matemático, por eso los computadores hacen exactamente lo que se les dice, no lo que se quiere que hagan. De esta manera, en programación se tiene un lenguaje bien definido donde los significados de las frases son únicos (no ambiguos). Esto exige que el programador exprese de forma precisa lo que desea hacer.

Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

- Clasificación de los lenguajes de programación
- Paradigmas de programación

4

Generaciones de los computadores



Clasificación de los lenguajes de programación I

De acuerdo a la complejidad de la sintaxis y la abstracción necesaria los lenguajes de programación se pueden clasificar en las siguientes categorías:

- Lenguajes de máquina o de bajo nivel.
- Lenguaje ensamblador o de intermedio nivel.
- Lenguajes de alto nivel.
- Lenguajes de muy alto nivel.
- Lenguajes naturales.



Clasificación de los lenguajes de programación II

Lenguajes de máquina I

Lenguajes de máquina o de bajo nivel. Es el único lenguaje que entiende el *hardware* (máquina) y usa exclusivamente el sistema binario (ceros y unos). Este lenguaje es específico para cada *hardware* (procesador, dispositivos, periféricos, etc.). El programa (tanto códigos de instrucción como datos) es almacenado en memoria.

Ejemplo

La estructura de una instrucción en lenguaje máquina es la siguiente:

CÓDIGO	ARGUMENTO(S)
0010	00011010
1010	10111000
0110	11010001





Clasificación de los lenguajes de programación III

Lenguaje ensamblador I

Lenguaje ensamblador o de nivel intermedio. El *lenguaje ensamblador* surgió de la necesidad de desarrollar un lenguaje de nivel mayor, que fuese más comprensible que el de la máquina pero que permitiera acceder a los detalles de éstas. Por esta razón, se desarrolló una forma de construir un lenguaje intermedio que empleara mnemónicos (palabras cortas escritas con caracteres alfanuméricos), para codificar las operaciones. Los datos y/o direcciones son codificados generalmente como números en un sistema hexadecimal. Generalmente es específico (aunque no único) para cada lenguaje de máquina.



Clasificación de los lenguajes de programación IV

Lenguaje ensamblador II

Entre los mnemónicos típicos se tienen:

ADD: Utilizado para sumar dos direcciones de memoria.

SUB: Utilizado para restar dos direcciones de memoria.

MUL: Utilizado para multiplicar dos direcciones de memoria.

MOV: Utilizado para mover un dato de un registro de la memoria en otro.

CALL: Utilizado para ejecutar una subrutina.

INT: Utilizado para invocar una interrupción.



Clasificación de los lenguajes de programación V

Lenguaje ensamblador III

Ejemplo

La estructura de una instrucción en este lenguaje es la

```
MNEMONICO ARGUMENTO(S)  
ADD R1, F4  
MOV F4, C2  
SUB AX, AX  
MOV AX, 18D  
SUB AX, 18D  
INT 20h
```




Clasificación de los lenguajes de programación VI

Lenguaje ensamblador IV

Un Ensamblador es un software, generalmente escrito en lenguaje de máquina, que es capaz de traducir de lenguaje ensamblador a lenguaje de máquina. Con este lenguaje se dio un salto fundamental, donde se logró separar el programa de la máquina empleando los conceptos de máquina de Turing y la arquitectura de Von Neumann, almacenando el programa en memoria y empleando el *hardware* como elemento de control.

Lo anterior dio origen a los sistemas operativos, logrando que la máquina completa pudiera controlar otro programa.



Clasificación de los lenguajes de programación VII

Lenguajes de alto nivel I

Aunque útil, el lenguaje ensamblador es aún muy difícil de entender, por eso se planteó la idea de generar un lenguaje más parecido al lenguaje natural que tiene facilidades de aprendizaje, lectura, escritura, corrección, transformación y conversión.

Los lenguajes de alto nivel están basados en una estructura gramatical para codificar estructuras de control y/o instrucciones. Cuentan con un conjunto de palabras reservadas (escritas en lenguaje natural).

Adicionalmente, éstos permiten el uso de símbolos aritméticos y relacionales para describir cálculos matemáticos, y generalmente representan las cantidades numéricas mediante sistema decimal.



Clasificación de los lenguajes de programación VIII

Lenguajes de alto nivel II

Ejemplo

La estructura de un fragmento de un programa escrito en un lenguaje de alto nivel tal como Python es la siguiente:

```
def biseccion(a, b, eps):  
    while(abs(f(a) - f(b)) >= eps):  
        x = (a + b) / 2  
        if(f(a) * f(x) < 0):  
            b = x  
        else:  
            a = x  
    return (a + b) / 2
```



Clasificación de los lenguajes de programación IX

Lenguajes de alto nivel III

Gracias a su estructura gramatical, estos lenguajes permiten al programador olvidar el direccionamiento de memoria (donde cargar datos y/o instrucciones en la memoria), ya que esto se realiza automáticamente por parte de un programa compilador o intérprete.

Clasificación de los lenguajes de programación X

Lenguajes de alto nivel IV

Los *programas fuente* escritos en lenguajes de alto nivel se compilan y a partir de estos se genera un *programa objeto* de código de máquina. El lenguaje de programación C entra en la clase de lenguajes compilados.

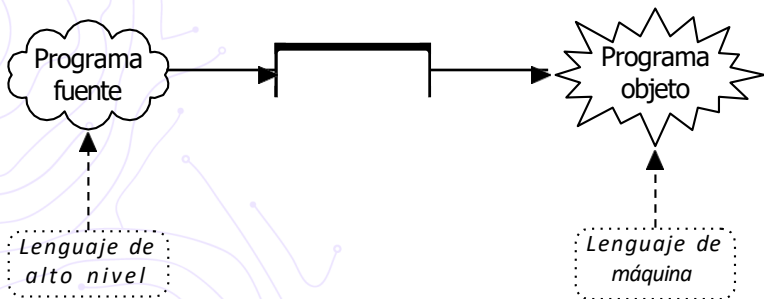


Figure: Estructura general de un compilador.

Clasificación de los lenguajes de programación XI

Lenguajes de alto nivel V

Otra forma de tratar los programas fuente es utilizando interpretes, en estos se toma instrucción por instrucción, estas se traducen a un código intermedio (Bytecode) similar al Ensamblador y de hay se traducen a código de máquina, usualmente utilizando un programa que depende de la plataforma y que se suele denominar *La maquina virtual*.

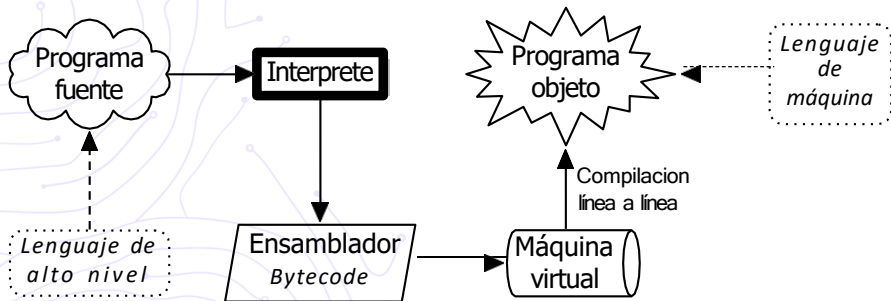


Figure: Estructura general de un interprete.



Clasificación de los lenguajes de programación XII

Lenguajes de muy alto nivel I

También conocidos como *lenguajes declarativos*. Su definición es más complicada que la de los anteriores. Se trata esencialmente de lenguajes taquigráficos que usan macroinstrucciones (se escribe más con menos). Cuando una operación que requiere de cientos de líneas en un lenguaje de alto nivel, en un lenguaje de muy alto nivel se requiere típicamente de unas cinco a diez líneas. Entre las características de estos lenguajes está el no uso de procedimientos. En los lenguajes de procedimientos se dice con detalle a la computadora las tareas a realizarse. En estos lenguajes se define solamente lo que se necesita computar, es decir, se enfatizan el qué hacer, en lugar del cómo hacerlo. Para esto, el compilador se encarga de los detalles relativos a cómo obtener el conjunto completo o parcial, y correcto de las soluciones.



Clasificación de los lenguajes de programación XIII

Lenguajes de muy alto nivel II

Ejemplo

Entre tareas típicas de estos lenguajes se pueden:

- Generar reportes sobre un criterio específico sobre un conjunto de datos.
 - Listar todas la personas que nacieron después de 1990.
- Encontrar las soluciones a una consulta realizada a un sistema con respecto a una base de conocimientos.
 - Pedro es padre de Jesus.
 - María es madre de Jesus.
 - ¿Quién es progenitor de Jesus?.



Clasificación de los lenguajes de programación XIV

Lenguajes de muy alto nivel III

Los lenguajes de muy alto nivel son fáciles de leer, comprender y programar, no requieren altos conocimientos de arquitecturas computacionales, esto los hace altamente transportables.

El principal inconveniente de estos lenguajes es que no hacen uso eficiente de los recursos computacionales.



Clasificación de los lenguajes de programación XV

Lenguajes naturales I

Se denominan así por su acercamiento a la escritura de los lenguajes humanos (inglés, francés, castellano, etc). El uso de un lenguaje natural con una base de conocimientos produce un sistema basado en el conocimiento. Una clase de estos sistemas son los *Sistemas Expertos*, que son una parte de la Inteligencia Artificial. Estos están todavía en su infancia, y actualmente puede considerarse que están más cerca de los lenguajes de muy alto nivel que de los lenguajes humanos.

Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

Clasificación de los lenguajes de programación

- Paradigmas de programación

4

Generaciones de los computadores



Paradigma

El filósofo e historiador de la ciencia, Thomas S. Kuhn fue el primero en dar el significado contemporáneo al término **paradigma** cuando lo adoptó para referirse a *“realizaciones científicas universalmente reconocidas que, durante cierto tiempo, proporcionan modelos de problemas y soluciones a una comunidad científica”*.



Paradigmas de programación I

Existen muchos lenguajes de programación de alto nivel con sus diferentes versiones. Por esta razón, es difícil su tipificación, pero una clasificación muy extendida desde el punto de vista de su estilo, forma de codificación y la filosofía de su creación es la siguiente:

- Lenguajes de programación imperativos.
- Lenguajes de programación estructurados o procedurales.
- Lenguajes de programación orientados a objetos.
- Lenguajes de programación declarativos.
- Lenguajes de programación multiparadigma



Paradigmas de programación II

Lenguajes de programación imperativos

Describe la programación en términos del estado de la memoria del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que se ejecutan secuencialmente y que le indican al computador cómo realizar una tarea. La implementación de hardware de la mayoría de computadores es imperativa ya que el hardware está diseñado para ejecutar código de máquina el cual es imperativo. Ejemplos: Ensamblador, Cobol, Pascal, Fortran, BASIC, C, Perl, Ada, MathLab, SciLab, Python, Java Script.



Paradigmas de programación III

Lenguajes de programación estructurados

En un lenguaje estructurado un programa se descompone en procedimientos individuales (funciones) que realizan una tarea específica, para resolver un problema en conjunto. Ejemplos: Pascal, C++, Visual Basic, Python, Java, MathLab, SciLab.



Paradigmas de programación IV

Lenguajes de programación orientados a objetos

En la Programación Orientada a Objetos se organiza el código en unidades denominadas clases, de las cuales los objetos son instancias de éstas, que al relacionarse e interaccionar entre sí, permiten resolver los retos planteados.

La Programación Orientada a Objetos trabaja de esta manera: todo el programa está construido con base a diferentes componentes (objetos), cada uno tiene un rol específico en el programa y todos los componentes pueden comunicarse entre ellos de forma predefinida.

Para la construcción de las clases se utilizan técnicas tales como: abstracción, herencia, modularidad, polimorfismo, encapsulamiento y ocultamiento. Ejemplos: Smalltalk, C++, Java, Python, R.



Paradigmas de programación V

Lenguajes de programación declarativos

Basado en la utilización de predicados lógicos (lógicos) o funciones matemáticas (funcionales), su objetivo es conseguir lenguajes expresivos en los que no sea necesario especificar cómo resolver el problema (programación convencional imperativa), sino qué problema se desea resolver. Los interpretes de los lenguajes declarativos tienen incorporado un motor de inferencia genérico que resuelve los problemas a partir de su especificación. Ejemplos: Lisp, ML, Haskell, Maude, Prolog, SQL.

Paradigmas de programación VI

Lenguajes de programación multiparadigma

La programación multiparadigma es la integración de muchos paradigmas de programación en un simple modelo. La misma facilita a los programadores utilizar la mejor herramienta para cada trabajo.

- En el caso de Oz, éste incluye programación imperativa, lógica, funcional, orientada a objetos, con restricciones, distribuida, concurrente.
- En el caso de Leda, éste incluye programación imperativa, orientada a objetos, lógica y funcional.

Agenda

1

Comunicación

2

Lenguajes

3

Lenguajes de Programación

Clasificación de los lenguajes de programación

Paradigmas de programación

4

Generaciones de los computadores



Generaciones de los computadores I

Primera generación de los computadores (1946 -1958) I

- Programación en lenguaje de máquina (el programa se escribe en código binario).
- La tecnología electrónica era a base de bulbos o tubos de vacío y válvulas.
- Desprendían bastante calor y tenían una vida relativamente corta. Máquinas grandes y pesadas.
- Alto consumo de energía. El voltaje de los tubos era de 300V y la posibilidad de fundirse era grande.



Generaciones de los computadores II

Primera generación de los computadores (1946 – 1958) II

- Almacenamiento de la información en cilindros magnéticos, donde se almacenan los datos y las instrucciones internas.
- La reprogramación se hacía intercambiando el cableado. Continuás fallas o interrupciones en la ejecución de los procesos. Requerían sistemas auxiliares de aire acondicionado especial.
- Alto costo.
- Uso de tarjetas perforadas para suministrar datos de programas.



Generaciones de los computadores III

Primera generación de los computadores (1946 – 1958) II

Las computadoras de esa generación fueron:

- 1946 ENIAC. Primera computadora digital electrónica de la historia.
- 1949 EDVAC. Primera computadora programable.
- 1951 UNIVAC I. Primera computadora comercial.
- 1953 IBM 701. Se usaban tarjetas perforadas para introducir los datos.
- 1954 IBM desarrolló otros modelos que usaban tambor magnético. Z1 (Alemana).
- Mark II.



Generaciones de los computadores IV

Segunda generación de los computadores

- Comunicación mediante el uso de lenguajes de alto nivel.
- Uso de los transistores contruidos en base al uso de un trozo de semiconductor que reemplazaron a los bulbos en los circuitos de los computadores. Fueron inventados por John Bardeen, Walter Houser Brattain y William Bradford Shockley.
- Tamaño más reducido que sus antecesoras de la primera generación.
- Consumían menos electricidad y producían menos calor que sus antecesoras.
- Aumento en la velocidad de las operaciones que ya no se mide en segundos sino en microsegundos.
- Costo más bajo que el de sus antecesoras.



Generaciones de los computadores V

Segunda generación de los computadores (1959 -1964) II

- Almacenamiento en cintas y discos magnéticos.
- Aparece gran cantidad de empresas dedicadas a la fabricación de los computadores.
- Programación con cintas perforadas y otras por medio de un cableado en un tablero.
- La transferencia de información de una computadora a otra requería un mínimo esfuerzo.
- Uso de impresoras para visualizar los resultados obtenidos a partir de los cálculos hechos.



Generaciones de los computadores VI

Segunda generación de los computadores (1959 – 1964) III

Las computadoras de esa generación fueron:

- Philco 212.
- UNIVAC M460.
- Control Data Corporations serie 1604.
- Control Data Corporations serie 3000.
- IBM 7090.
- NCR 315.
- Burroughs serie 5000. ATLAS.



Generaciones de los computadores VII

Tercera generación de los computadores (1965 – 1971) I

- Circuitos integrados desarrollado en 1958 por Jack Kilbry.
- Miniaturización y reunión de centenares de elementos en una placa de silicio o (chip).
- Menor consumo de energía.
- Reducción de espacio utilizado.
- Aumento de fiabilidad y flexibilidad.
- Aumenta la capacidad de almacenamiento y se reduce el tiempo de ejecución.
- Disponibilidad de gran cantidad de lenguajes de programación de alto nivel.



Generaciones de los computadores VIII

Tercera generación de los computadores (1965 – 1971) II

- Compatibilidad para compartir software entre diversos equipos.
- Construcción de computadoras en serie.
- Teleproceso.
- Multiprogramación.
- Tiempo compartido.
- Aparición de periféricos.
- Aparición de aplicaciones.
- Aparición del sistema operativo llamado OS.
- Aparición de la mini computadora.



Generaciones de los computadores IX

Tercera generación de los computadores (1965 – 1971) III

Las computadoras de esa generación fueron:

- IBM 360.
- Control Data Corporations serie 6000.
- Control Data Corporations serie 6600.
- Considerada la más rápida de su época.
- IBM 370.



Generaciones de los computadores X

Cuarta generación de los computadores (1972 – 1981) I

- Microprocesador: un único circuito integrado en el que se reúnen los elementos básicos de la máquina desarrollado por Intel Corporation (1971).
- Se minimiza el tamaño de los circuitos.
- Aumenta la capacidad de almacenamiento.
- Reemplazo de las memorias con núcleos magnéticos, por las de chips de silicio.
- Colocación de muchos componentes electrónicos en un sólo chip.



Generaciones de los computadores XI

Cuarta generación de los computadores (1972 – 1981) II

- Se aumenta la velocidad de computo.
- Reducción significativa de los costos de los computadores.
- Popularización del uso de los computadores.
- Steve Woziniak y Steve Jobs inventan la primera microcomputadora de uso masivo, fundadores de APPLE (1976) .
- Sistemas de tratamiento de base de datos.
- Generalización de las aplicaciones.
- Multiproceso.



Generaciones de los computadores XII

Quinta generación de los computadores (1982 – 1989) I

- Japón lanzó en 1983 el llamado "programa de la quinta generación de computadoras".
- Traductores de lenguajes.
- Creación de la primera supercomputadora con capacidad de proceso paralelo, diseñada por Seymour Cray (1982).
- Fuerte aplicación de la inteligencia artificial: sistemas expertos, redes neuronales, teoría del caos, programación heurística, algoritmos genéticos.



Generaciones de los computadores XIII

Quinta generación de los computadores (1982 – 1989) II

- Fibras ópticas.
- Telecomunicaciones.
- DVD.
- Uso del ratón (mouse).
- Robots con capacidad de movimiento.
- Juegos.
- Reconocimientos de formas tridimensionales, voz e imágenes.



Generaciones de los computadores XIV

Sexta generación de los computadores (1990–actualidad) I

- Arquitecturas combinadas Paralelo / Vectorial.
- Masificación del uso de redes de área mundial (Wide Area Network, WAN).
- Comunicación a través de fibras ópticas y satélites.
- 1997- El Pentium II
- 1999- El Pentium III
- 2001- El Pentium 4
- AMD Phenom II
- Intel Core i3, i5, i7, i9, i10
- AMD Ryzen 5, 7, 9
- Tablets y Smartphones.



Videos de apoyo

¿Qué es una máquina de Turing?

url: https://www.youtube.com/watch?v=iaXLDz_UeYY

Arquitectura de Von Neumann

url: https://www.youtube.com/watch?v=KpAHKQrco_s

Generaciones de las computadoras

url: <https://www.youtube.com/watch?v=0fB-B82Epx0>