

# Informática I

## Unidad 1: Introducción a la Computación

Cristian A. Jimenez C.

Departamento de Ingeniería Electrónica y Telecomunicaciones  
Facultad de Ingenierías  
Universidad de Antioquia

[cjimenez.castano@udea.edu.co](mailto:cjimenez.castano@udea.edu.co)  
Oficina 20-405

1

## Guía de Estudio

2

### Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

3

### Tema 1.2. Hardware y Software

- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

4

### Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

5

### Bibliografía

## 1 Guía de Estudio

### 2 Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

### 3 Tema 1.2. Hardware y Software

- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

### 4 Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

### 5 Bibliografía

# Aspectos Básicos I

## Objetivo

La primera unidad del curso busca hacer una **introducción general a los principales aspectos alrededor de la tecnología de los computadores**. Se definirán conceptos que se usarán a lo largo del curso, y se realizará un viaje a través de las capas de abstracción de software y hardware, que permitirán entender cómo un problema abstracto formularlo por un humano puede, finalmente, ser resultado por una máquina electrónica; el computador.

## Laboratorio

- Se plantea para esta unidad una primera actividad práctica, que puede ser desarrollada a la vez que avanza en los temas teóricos ya que no hay dependencia en estos. El objetivo es conocer e instalar el sistema operativo Linux, así como estudiar el manejo básico de la consola de comandos.
- El estudiante debe iniciar viendo las [videoclase](#) sobre **Linux** y luego leer [la parte 1 de la guía de la practica 1](#).
- Luego de la instalación, el estudiante debe ver la [videoclase](#) de **manejo de consola de comandos** y leer [la parte 2 de la guía de la practica 1](#), así como desarrollar los ejercicios indicados.

# Aspectos Básicos II

## Evaluación

- Un examen escrito sobre los **temas 1.1, 1.2 y 1.3** pero que también incluirá la **Unidad 2**. Este examen tendrá un valor del **20 %**.
- Un **informe corto** sobre la *instalación de Linux*. Este tendrá un valor del **2 %**.
- Un **quiz** sobre **comandos** para el manejo de la consola de Linux. Este tendrá un valor del **2 %**.

## Bibliografía



Y. Patt, S. Patel, *Introducción a los sistemas de computación*, 2<sup>a</sup> ed., McGraw-Hill, 2005.

# Contenido

1 Guía de Estudio

2 Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

3 Tema 1.2. Hardware y Software

- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

4 Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

5 Bibliografía

## Definición

Es el área de la ciencia y la tecnología que estudia el **procesamiento de la información** usando computadores.

Dentro de la informática se realizan dos principales preguntas:

- ① ¿Cómo **procesar datos** existentes para extraer nueva información útil para algún propósito?.
- ② ¿Y cómo hacerlo de manera **eficiente**?

## Definición

Es el área de la ciencia y la tecnología que estudia el **procesamiento de la información** usando computadores.

Dentro de la informática se realizan dos principales preguntas:

- ① ¿Cómo **procesar datos** existentes para extraer nueva información útil para algún propósito?.

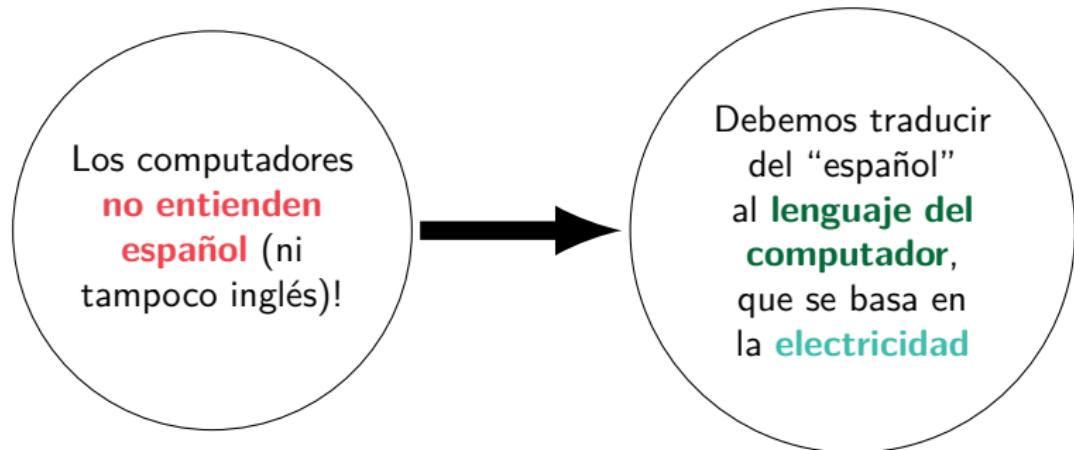
**Esto se realiza por medio de la programación.**

- ② ¿Y cómo hacerlo de manera **eficiente**?

**Se realiza por medio del entendimiento sobre la tecnología y siendo metódicos.**

## Definición

La programación es el proceso mediante el cual transmitimos al computador las operaciones que queremos que haga.



# El Computador I

## Definición

El computer es una **máquina electrónica programable** capaz de recibir datos, memorizarlos, hacer cálculos con ellos y entregar nuevos datos (resultados).



Ver Video



Partes del Computador  
Obtenido de [7]

## ¿Qué hace bien el computador?

- Cálculos Repetitivos.
- Cálculos aritméticos complejos.
- Grandes volúmenes de información.
- Precisión numérica.
- Velocidad de procesamiento y respuesta ( $1 \times 10^9$  operaciones mientras tu celular car al suelo.).

## ¿Y para qué **no** es bueno?



# Historia de la Computación I

-500

0

**Antikythera, Grecia año 1 BC** Usada para calcular eventos astronómicos, encontrada en el fondo del mar [6].



500

1642

**Calculadora de Pascal, Francia, 1642** Servía para sumar y restar, luego mejorada por Leibniz multiplicación y división. [8].



# Historia de la Computación II

1810



1837

*Máquina analítica de Babbage, Reino Unido* Primer computador de propósito general, sólo pudo ser puesto a funcionar 110 años más tarde [5].



1943

- ) Se crearon COLOSUS y Bombe para descifrar las comunicaciones nazis (Enigma), con la ayuda de Alan Turing.
- ) Jhon von Neumann, Mauchley y Eckert: ENIAC, EDVAC, EDSAC, etc..
- ) Máquinas de enormes tamaños y consumo de energía: **tubos de vacío**.



# Historia de la Computación III

1950

1960

1970

1980

1990

2005

## Los 60's

- ) Schockley, Bardeen y Brattain, inventaron el **transistor** en 1948, trabajando para Bell Labs.
- ) Kilby y Noyce inventaron el **circuito integrado** en 1958. (imágenes obtenidas de [1, 2])



## Los 80's (El computador personal)

- ) Incremento en **miniaturización**: millones de transistores en un sólo chip.
- ) IBM y Apple comercializaron el computador personal o de escritorio.



## Siglo XXI

- ) Varios procesadores (**cores**) en un sólo chip.
- ) Sistemas embebidos: "Computadores" como parte de otro tipo de dispositivos.



## ¿Quién fue el Primer Programador(a)?



### Ava Lovelace [4]

- 1815 - 1852 (36 años)
- Asistente de Charles Babbage.
- Talentosa Matemática inglesa y escritora
- Aprendio a “programar” la máquina analítica de Babbage.

## Reglas

- Deben de ingresar al [enlace](#), o haciendo uso del QR, a un formulario de Google.
- Contestan las preguntas, ya sea con la información dada en clase o por su propia investigación para las mismas.
- Esta Tarea no es obligatoria.



# Contenido

## 1 Guía de Estudio

## 2 Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

## 3 Tema 1.2. Hardware y Software

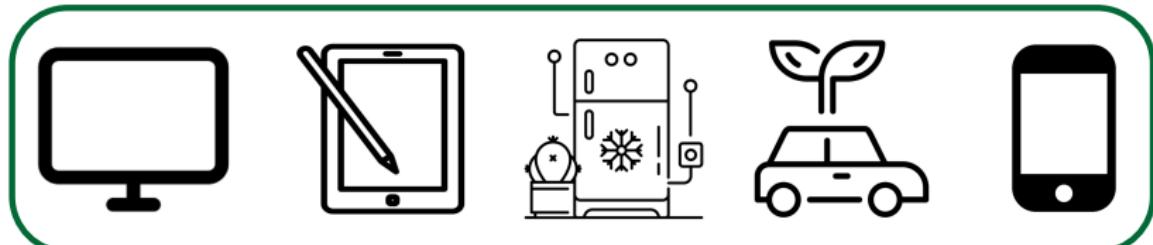
- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

## 4 Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

## 5 Bibliografía

# Componentes del Computador



## *SOFTWARE: "Los programas"*



## *HARDWARE: "Los circuitos"*

**Figura:** Primera parte, ejemplos de computadoras. Segunda y tercera parte, software y hardware respectivamente.

# Arquitectura Von Neumann

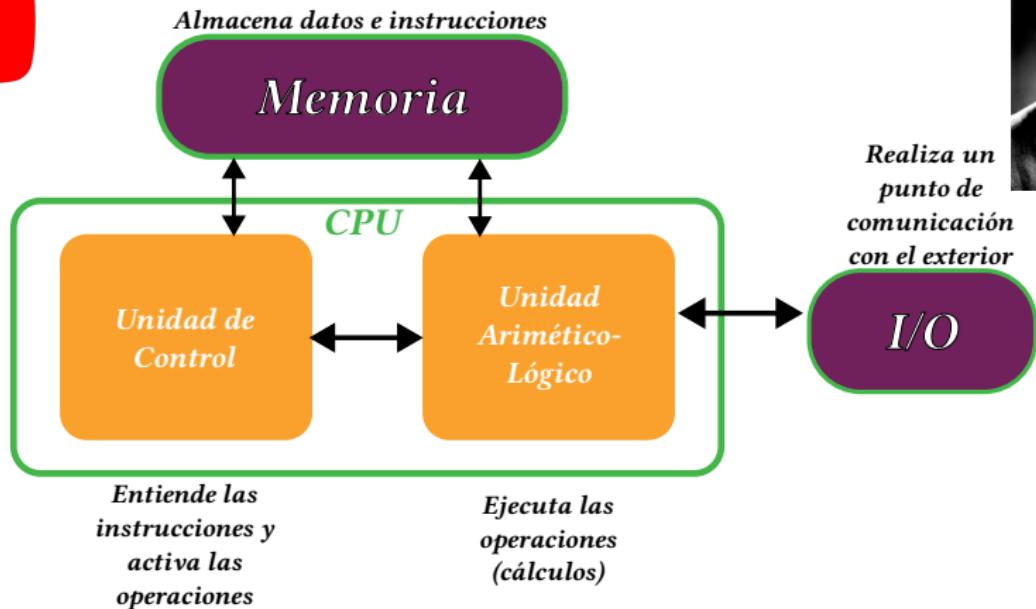


Figura: Arquitectura Von Neuman [3]. Fotografía obtenida de [11].

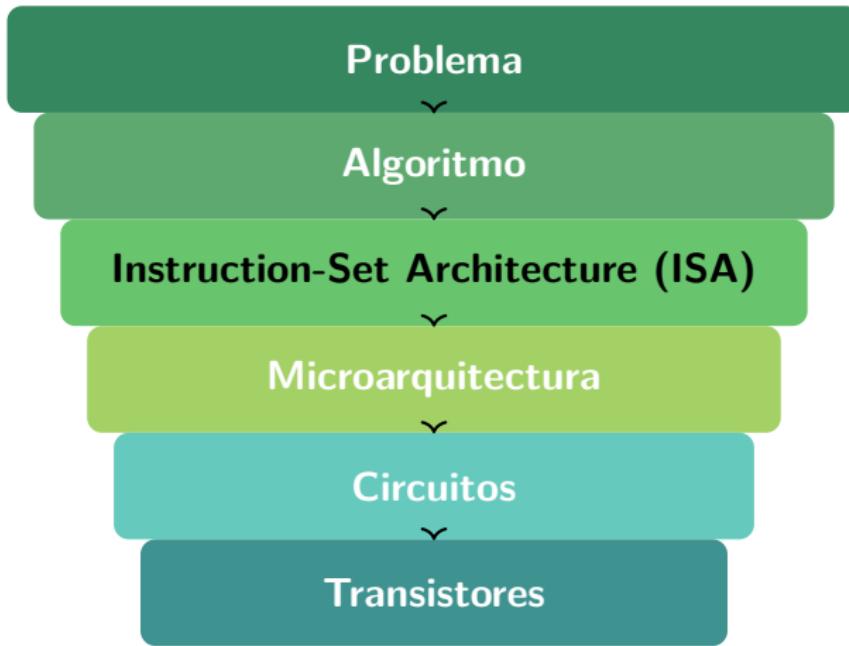
# ¡No es magia!

- Todos los computadores son esencialmente capaces de hacer lo mismo (mismas operaciones).
- Para hacerle **entender** a un computador lo que debe hacer, y cómo lo debe hacer, existe una serie de **transformaciones** que cruzan múltiples **capas de abstracción**.

Lo anterior nos pueden llevar a preguntarnos:

- ¿Por qué es importante la abstracción?
- ¿Cuándo es importante la **concreción**?

# Capas de abstracción



# Problema

- Estas son ideas, dificultades, oportunidades.

# Problema

- Estas son ideas, dificultades, oportunidades.
- Estas se plantean de forma **ambigua**.

# Problema

- Estas son ideas, dificultades, oportunidades.
- Estas se plantean de forma **ambigua**.
- Se expresan por medio de “Lenguaje Natural”.

# Problema

- Estas son ideas, dificultades, oportunidades.
- Estas se plantean de forma **ambigua**.
- Se expresan por medio de “Lenguaje Natural”.

“Necesito algo para  
crear documentos  
matemáticos, más  
fácil”

“Qué tal si llevo las  
matrices contables  
a mi computado-  
ra, para realizar los  
calculos más sim-  
ples.”

# Problema

- Estas son ideas, dificultades, oportunidades.
- Estas se plantean de forma **ambigua**.
- Se expresan por medio de “Lenguaje Natural”.

“Necesito algo para  
crear documentos  
matemáticos, más  
fácil” (Se creo La-  
Tex)

“Qué tal si llevo las  
matrices contables  
a mi computado-  
ra, para realizar los  
calculos más sim-  
ples.” (Aparece el  
software contable)

# Algoritmo I

## Definición

“Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.”<sup>a</sup>

<sup>a</sup><http://dle.rae.es/algoritmo>

## Importante

- **Finito:** El procedimiento debe terminar.
- **Preciso:** Pasos definidos de manera clara y precisa.
- **Computable:** Pasos que pueden ser calculados por el computador.

# Algoritmo I

## Definición

“Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.”<sup>a</sup>

<sup>a</sup><http://dle.rae.es/algoritmo>

## Importante

- **Finito:** El procedimiento debe terminar.  
“Calcular todos los dígitos decimales del numero  $\pi$ ” $\Rightarrow$  ¡Proceso Infinito!.
- **Preciso:** Pasos definidos de manera clara y precisa.  
“Esperar un rato” $\Rightarrow$  ¡Instrucción Imprecisa!.
- **Computable:** Pasos que pueden ser calculados por el computador.  
“Encontrar el número primo más grande” $\Rightarrow$  ¡No existe de forma general!.

# Algoritmo II



Figura: Ejemplo Interacción Problema-Algoritmo.

# Programa I

## Definición

Es la representación de un algoritmo en un “lenguaje” que **pueda ser procesado** por el computador.

## ¿Cómo los creamos?

Implementamos (hacemos) **programas** usando **lenguajes de programación**.

# Programa II

Utiliza reglas y hechos para representar el conocimiento, y emplea inferencia lógica para resolver problemas. Es común en aplicaciones de inteligencia artificial y sistemas expertos.

- Sistemas expertos e inteligencia artificial (**Prolog**).
- Planificación y resolución de problemas.
- Procesamiento de lenguaje natural y razonamiento.

Se basa en la ejecución de una serie de instrucciones de manera secuencial, dividiendo el programa en procedimientos o subrutinas. Cada paso describe una acción a realizar.

- Aplicaciones Embebidas.
- Sistemas Operativos (**C**).
- Cálculos numéricos (**Fortran**).

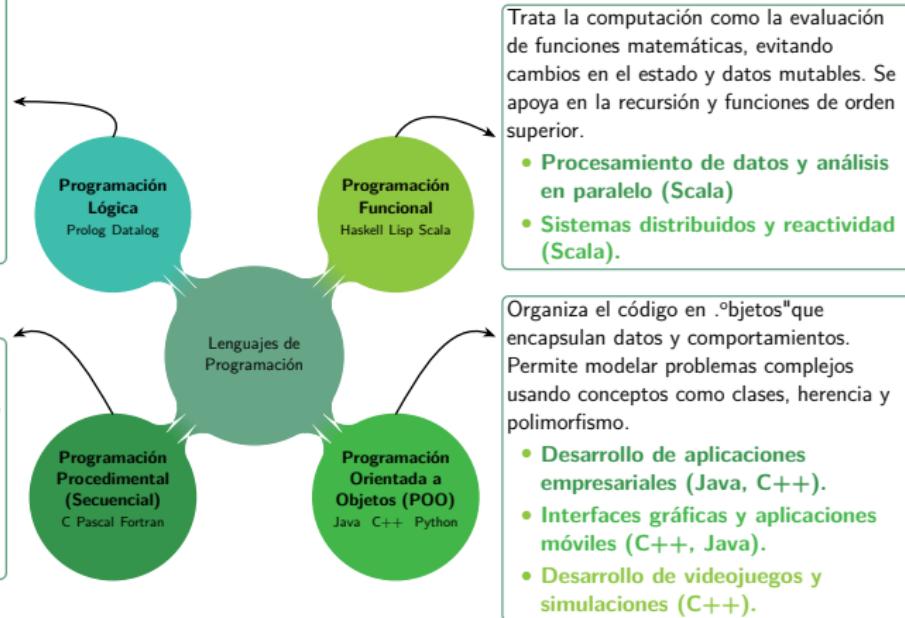


Figura: Clasificación de Lenguajes de Programación

# Programa III

```

1 from tensorflow.examples.tutorials import mnist
2 from tensorflow.examples.tutorials.mnist import input_data
3
4 # Import and preprocess data
5
6 # Train, validate, test
7 x_train, y_train, x_val, y_val, x_test, y_test = cifar10.load_data()
8
9 x_train = x_train / 255.0
10 y_train = to_categorical(y_train, 10)
11
12 x_val = x_val / 255.0
13 y_val = to_categorical(y_val, 10)
14
15 x_test = x_test / 255.0
16 y_test = to_categorical(y_test, 10)
17
18 # Familiar with Keras? You may prefer to use "from tensorflow.keras.layers import Dense, Activation, ..."
19 # MNIST = input_data.read_data_sets('MNIST_data', one_hot=True, validation_size=0, reshape=True)
20
```

## Lenguajes de Alto Nivel

Independiente de la CPU

## Alejado de la CPU

## Ejemplos: Python, Java, C#



A graphic consisting of a grid of green binary digits (0s and 1s) on a black background. The digits are arranged in a pattern that forms a stylized letter 'E' shape.

## Lenguajes de Bajo Nivel

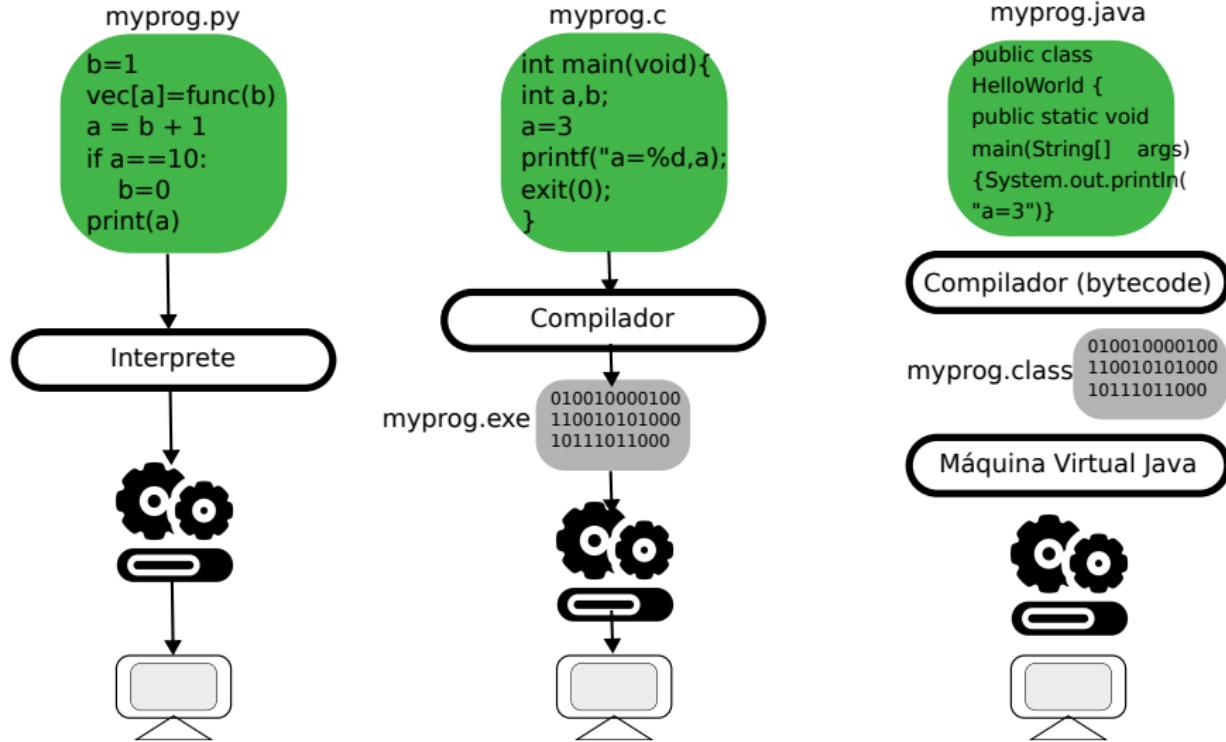
## Definidos según la CPU

## Cercano a la CPU

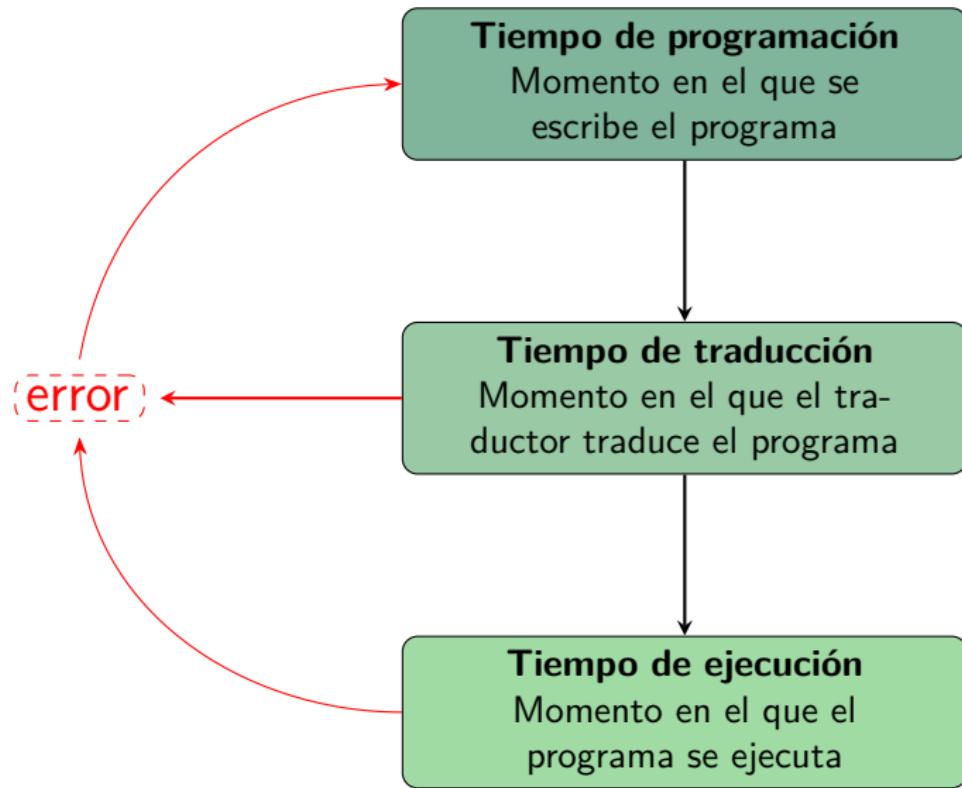
**Ejemplos:** Assembly, Código Máquina.

- Los lenguajes de **Alto Nivel** tienen una aproximación al lenguaje natural, humano. Se hace uso de **interpretes**, o **compiladores**, para **“traducirle”** al computador las instrucciones.
  - Por otra parte, los lenguajes de **Bajo Nivel** hacen traducen su programa en código máquina (binario, {0, 1}) por medio de **ensamblador** o directamente a ella.

# Interprete Vs. Compilador



# Ciclo de la Programación

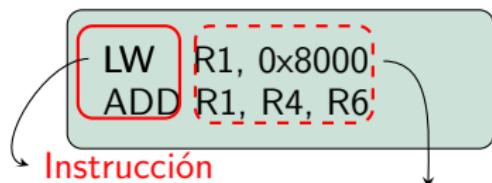


# ISA: Instruction-Set Architecture I

## Definición

- Conjunto de Instrucciones para controlar el procesador.
- Interface entre el software y el hardware.

### Assembly



Instrucción

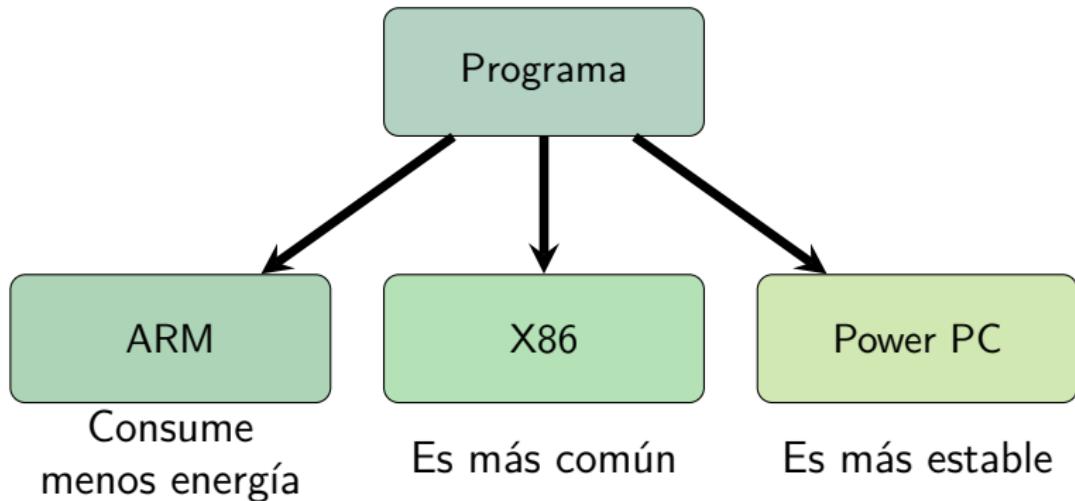
Operandos ⇒ Tipos de Datos

Vs.

### Machine code

The diagram shows a light green rectangular box containing two lines of binary machine code. The first line is '0100110111011011' and the second line is '0110111000101010'. A curved arrow points from the left side of the box to the left of the first line of binary code.

0100110111011011  
0110111000101010



## Nota

Lo van a estudiar en **Electrónica Digital II, III** (Electrónica) y **Fundamentos de Electrónica Digital** (Telecomunicaciones).

# Micro-arquitectura

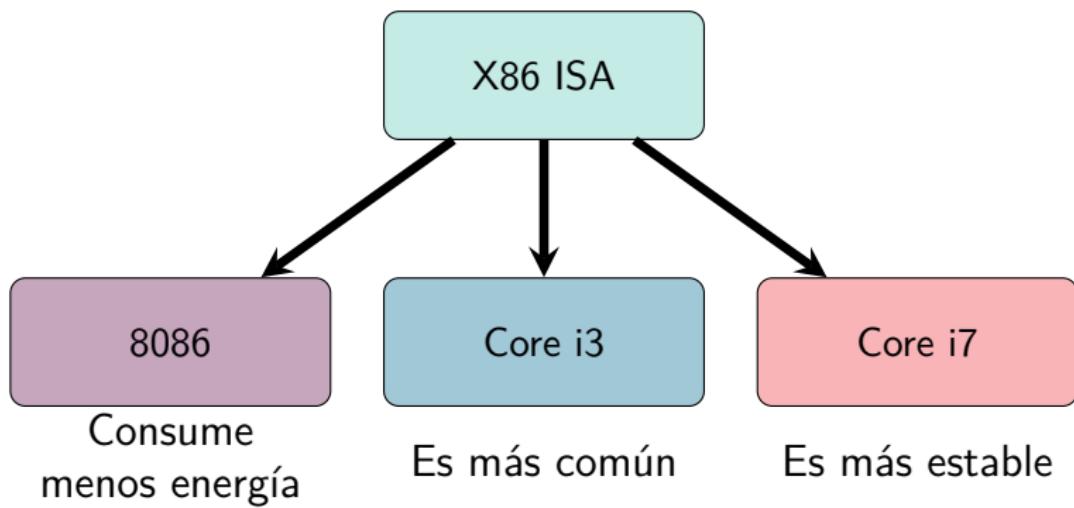
## Definición

Conjunto de mecanismos y estructuras de hardware que no están expuestas al programador o programa.



## Nota Electrónica

Lo van a estudiar en **Electrónica Digital II** y la electiva **Arquitecturas Avanzadas de Computadores**.



# Circuitos

Compuertas lógicas que permiten hacer operaciones elementales con bit:

INPUT				
		AND	OR	NOT
0	0	0	0	1 1
0	1	0	1	1 0
1	0	0	1	0 1
1	1	1	1	0 0

Lo van a estudiar en Matemáticas Discretas, **Electrónica Digital I** (Electrónica) y **Fundamentos de Electrónica Digital** (Telecomunicaciones).

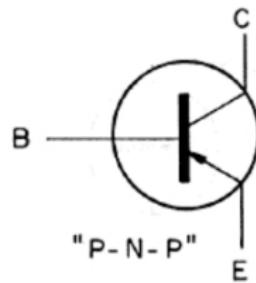


# Transistor

- Elemento base de la electrónica que funciona como **switch**.
- Fabricados con silicio (CMOS).
- Litografía de  $10\text{ }\mu\text{m}$ .

## Nota

Lo van a estudiar en **Electrónica Analógica I y II** (Electrónica), y en **Fundamentos en Electrónica Analógica** (Telecomunicaciones).



# ¿Qué es un Sistema Operativo?



Figura: Imagen obtenida de [9] (Acceso 20025/03/11)

- Conjunto de programas que administra y coordina los recursos de hardware y software.
- Proporciona una interfaz (gráfica o de línea de comandos) para la interacción con el usuario.
- Gestiona tareas esenciales: memoria, procesos, archivos, dispositivos de entrada/salida y seguridad.

(Fuente: [10])

# Tipos de Sistemas Operativos Populares

- **Windows:** Sistema operativo de propósito general, muy extendido en PCs.
- **GNU/Linux:** Sistema operativo de código abierto, reconocido por su seguridad y alta personalización; muy usado en servidores y entornos de desarrollo.
- **macOS:** Basado en **Unix**, destaca por su diseño elegante, interfaz intuitiva y alta estabilidad. Está optimizado para el hardware de **Apple**, lo que garantiza una integración fluida y una experiencia de usuario consistente, especialmente apreciada en áreas creativas y profesionales.

# Ventajas y Desventajas: Windows

## Ventajas:

- ✓ Gran compatibilidad con hardware y una amplia oferta de software comercial.
- ✓ Interfaz gráfica intuitiva y amigable para el usuario.
- ✓ Amplio soporte técnico y comunidad activa.

## Desventajas:

- ✗ Mayor vulnerabilidad a virus y malware.
- ✗ Requiere licencia, lo que incrementa el costo.
- ✗ Actualizaciones frecuentes que pueden generar problemas de compatibilidad.

# Ventajas y Desventajas: GNU/Linux

## Ventajas:

- ✓ Gratuito y de código abierto, con una comunidad activa.
- ✓ Alta seguridad, estabilidad y buen rendimiento, ideal para servidores.
- ✓ Gran capacidad de personalización en entornos y aplicaciones.

## Desventajas:

- ✗ Curva de aprendizaje más pronunciada para usuarios nuevos.
- ✗ Menor disponibilidad de software comercial y algunos juegos.
- ✗ Soporte técnico formal limitado; mayor dependencia de foros y documentación.

# Ventajas y Desventajas: macOS

## Ventajas:

- ✓ Alta estabilidad y seguridad.
- ✓ Interfaz gráfica intuitiva y cuidada.
- ✓ Excelente integración con el ecosistema Apple (iPhone, iPad, etc.).
- ✓ Buena optimización del rendimiento en hardware específico.

## Desventajas:

- ✗ Hardware exclusivo, lo que limita las opciones de compra.
- ✗ Precio elevado comparado con otras alternativas.
- ✗ Menor grado de personalización y flexibilidad en comparación con sistemas abiertos como GNU/Linux.

# Comparativa y Conclusiones

## Comparativa Rápida

	Windows	GNU/Linux	macOS
<b>Facilidad de uso</b>	Alta	Media	Alta
<b>Seguridad</b>	Media	Alta	Alta
<b>Costo</b>	Media	Gratis	Alta
<b>Personalización</b>	Media	Alta	Baja

- **Windows** es ideal si necesitas gran compatibilidad y acceso a software comercial.
- **GNU/Linux** es perfecto para usuarios avanzados, servidores y quienes valoran la seguridad y personalización.
- **macOS** ofrece una experiencia segura y consistente en sus dispositivos.

**Software Libre** se basa en la idea de que los programas deben ser accesibles para todos, permitiendo:

- **Uso:** Uso sin restricciones.
- **Estudio:** Acceso al código fuente para aprender y comprender su funcionamiento.
- **Modificación:** Adaptar y mejorar el software según las necesidades.
- **Redistribución:** Compartir y distribuir copias o versiones modificadas.

Este movimiento se impulsó en 1983 con el Proyecto *GNU*—(GNU's Not Unix), liderado por [Richard Stallman](#).



# Linux y su Historia

- En 1991, [Linus Torvalds](#) lanzó el kernel **Linux**, sentando las bases del sistema GNU/Linux.
- La combinación de Linux con las herramientas GNU dio lugar a un sistema operativo completamente libre y de código abierto.
- Su evolución ha sido impulsada por una comunidad global de desarrolladores.



Figura: Linus Torvalds [12]

# Importancia de Linux en el Desarrollo Actual

- Es la base de la mayoría de servidores, supercomputadoras y sistemas embebidos.
- Su naturaleza abierta fomenta la innovación y permite reducir costos de desarrollo.
- Es el pilar en sistemas móviles (Android se basa en Linux).
- Facilita la colaboración y personalización para proyectos específicos.

# Contenido

## 1 Guía de Estudio

## 2 Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

## 3 Tema 1.2. Hardware y Software

- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

## 4 Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

## 5 Bibliografía

# Unidades de Información: Bit, Byte y Palabra (Word)

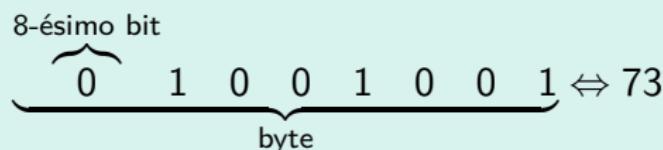
- **Bit:** Unidad mínima de información en una computadora, que puede representar dos valores posibles: 0 ó 1.
- **Byte:** Conjunto de 8 bits, utilizado para representar un carácter en la computadora.
- **Palabra (Word):** Unidad de máxima de información que una micro-arquitectura (ejemplo CPU) puede manejar, depende de la arquitectura del sistema, comúnmente de 16, 32 ó 64 bits.

# Unidades de Información: Bit, Byte y Palabra (Word)

- **Bit:** Unidad mínima de información en una computadora, que puede representar dos valores posibles: 0 ó 1.
- **Byte:** Conjunto de 8 bits, utilizado para representar un carácter en la computadora.
- **Palabra (Word):** Unidad de máxima de información que una micro-arquitectura (ejemplo CPU) puede manejar, depende de la arquitectura del sistema, comúnmente de 16, 32 ó 64 bits.

## A considerar

☞ 1 byte puede representar  $2^8$ , 256, valores distintos. Ejemplo:



☞ Arquitecturas de 64 bits, o 8 byte, manejan como máximo **palabras** de este tamaño.

# Unidades de Almacenamiento y su Relación

Unidad	Símbolo	Tamaño
Bit	bit	1 bit
Byte	B	8 bits
Kilobyte	KB	1,024 byte
Megabyte	MB	1,024 KB
Gigabyte	GB	1,024 MB
Terabyte	TB	1,024 GB
Petabyte	PB	1,024 TB
Exabyte	EB	1,024 PB

Cuadro: Unidades de almacenamiento y sus equivalencias

# Conversión de Unidades: Ejemplos I

## Ejemplo 1: Convertir 16,384 bits a Kilobyte (KB)

- **Paso 1:** Convertir bits a byte:

$$16,384 \text{ bits} \times \frac{1 \text{ byte}}{8 \text{ bits}} = 2,048 \text{ byte}$$

- **Paso 2:** Convertir byte a Kilobyte:

$$2,048 \text{ byte} \times \frac{1 \text{ KB}}{1024 \text{ byte}} = 2 \text{ KB}$$

- **Resultado:** 16,384 bits equivalen a 2 KB.

# Conversión de Unidades: Ejemplos II

## Ejemplo 2: Convertir 5 Megabyte (MB) a bits

- **Paso 1:** Convertir Megabyte a byte:

$$5 \text{ MB} \times \frac{1024 \text{ KB}}{1 \text{ MB}} \times \frac{1024 \text{ byte}}{1 \text{ KB}} = 5'242,880 \text{ byte}$$

- **Paso 2:** Convertir byte a bits:

$$5'242,880 \text{ byte} \times \frac{8 \text{ bits}}{1 \text{ byte}} = 41'943,040 \text{ bits}$$

- **Resultado:** 5 MB equivalen a 41'943,040 bits.

# Dirección de Memoria I

## Concepto

En una arquitectura de 16 bits, las direcciones de memoria tienen 16 bits de longitud, permitiendo direccionar hasta  $2^{16}$  posiciones de memoria, es decir, 65,536 bytes (64 KiB).

Dirección (Hex)	Dirección (Binario)	Contenido (Binario)
0x0000	0000 0000 0000 0000	0100 1101
0x0001	0000 0000 0000 0001	1010 0110
0x0002	0000 0000 0000 0010	1111 0000
0x0003	0000 0000 0000 0011	0001 1010
:	:	:
0xFFFFE	1111 1111 1111 1110	1100 0011
0xFFFF	1111 1111 1111 1111	0010 1001

**Cuadro:** Ejemplo de direcciones de memoria y sus contenidos en una arquitectura de 16 bits

## Importante

- Cada dirección apunta a una celda que almacena 1 byte (8 bits) de información,  $2^8 = 256$ .
- La organización de la memoria es crucial para el funcionamiento eficiente del sistema.

## Tamaño Memoria Vs. Contenido

- En una arquitectura de 32 bits, el procesador puede direccionar hasta 4 GB de memoria, ya que  $2^{32}$  bytes equivalen a 4,294,967,296 bytes.
- En una arquitectura de 64 bits, la capacidad teórica de direccionamiento es de 16 EX (exabytes), aunque las implementaciones actuales suelen soportar hasta 256 TB (terabytes) debido a limitaciones prácticas.

# Sistema Binario

## Definición

Sistema numérico que consta de secuencias de 0s (ceros) ó 1s (unos).

$$\begin{array}{l} \text{Sistema Decimal} \rightarrow \begin{array}{ccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 1 & 0 & 1 & 0 & 1 \end{array} \\ \text{Sistema Binario} \rightarrow \end{array}$$

¿Cómo se representa un entero de base decimal en base binaria?

$$\begin{aligned} (10101)_2 &= (2^4 * 1 + 2^3 * 0 + 2^2 * 1 + 2^1 * 0 + 2^0 * 1)_{10} \\ &= (16 * 1 + 8 * 0 + 4 * 1 + 2 * 0 + 1 * 1)_{10} \\ &= (16 + 0 + 4 + 0 + 1)_{10} \\ &= (21)_{10} \end{aligned}$$

# Sistema Binario

## Capacidad

Dependiendo del número de bits que conforman el sistema binario, se determina la capacidad de elementos a representar. En el caso de tener un sistema de,  $d$  bits, entonces tiene capacidad de representar  $2^d$  valores; ejemplo,  $d = 4$  entonces este sistema representar  $2^4 = 16$  valores.

- $(0000)_2 = (0)_{10}$
- $(0001)_2 = (1)_{10}$
- $(0010)_2 = (2)_{10}$
- $(0011)_2 = (3)_{10}$
- $(0100)_2 = (4)_{10}$
- $(0101)_2 = (5)_{10}$
- $(0110)_2 = (6)_{10}$
- $(0111)_2 = (7)_{10}$
- $(1000)_2 = (8)_{10}$
- $(1001)_2 = (7)_{10}$
- $(1010)_2 = (10)_{10}$
- $(1011)_2 = (11)_{10}$
- $(1100)_2 = (12)_{10}$
- $(1101)_2 = (13)_{10}$
- $(1110)_2 = (14)_{10}$
- $(1111)_2 = (15)_{10}$

# Ejercicios Sistema Binario

## Ejercicios

Pasa los siguientes valores desde sistema binario,  $(\cdot)_2$ , a su respectivo valor decimal,  $(\cdot)_{10}$ .

-   $(010010)_2$
-   $(01010)_2$
-   $(10010)_2$
-   $(011100)_2$
-   $(11101)_2$
-   $(101100)_2$
-   $(001110)_2$

# Tipos de Datos, Número de Bits y Rangos (Ejemplo)

TIPO	Número de bits	Rango
char	8	-128 a 127
unsigned char	8	0 a 255
signed char	8	-128 a 127
short	16	-32768 a 32767
int	16	-32768 a 32767
long int	32	-2147483648 a 2147483647
unsigned long int	32	0 a 4294967295
float	32	3.4E-38 a 3.4E+38
double	64	1.7E-308 a 1.7E+308
long double	64 ó 80	1.7E-308 a 1.7E+308 ó 3.4E-4932 a 1.1E+4932

Cuadro: Ejemplo de tipos de datos, número de bits y rangos. El simbolo E es notación científica,  $10E+3 = 10 \times 10^3 = 10000$

# Tabla ASCII Completa (0 a 95)

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr
0	00	000	NUL	32	20	040	Space	64	40	100	Ø
1	01	001	SOH	33	21	041	!	65	41	101	A
2	02	002	STX	34	22	042	"	66	42	102	B
3	03	003	ETX	35	23	043	#	67	43	103	C
4	04	004	EOT	36	24	044	\$	68	44	104	D
5	05	005	ENQ	37	25	045	%	69	45	105	E
6	06	006	ACK	38	26	046	&	70	46	106	F
7	07	007	BEL	39	27	047	'	71	47	107	G
8	08	010	BS	40	28	050	(	72	48	110	H
9	09	011	HT	41	29	051	)	73	49	111	I
10	0A	012	LF	42	2A	052	*	74	4A	112	J
11	0B	013	VT	43	2B	053	+	75	4B	113	K
12	0C	014	FF	44	2C	054	,	76	4C	114	L
13	0D	015	CR	45	2D	055	-	77	4D	115	M
14	0E	016	SO	46	2E	056	.	78	4E	116	N
15	0F	017	SI	47	2F	057	/	79	4F	117	O
16	10	020	DLE	48	30	060	0	80	50	120	P
17	11	021	DC1	49	31	061	1	81	51	121	Q
18	12	022	DC2	50	32	062	2	82	52	122	R
19	13	023	DC3	51	33	063	3	83	53	123	S
20	14	024	DC4	52	34	064	4	84	54	124	T
21	15	025	NAK	53	35	065	5	85	55	125	U
22	16	026	SYN	54	36	066	6	86	56	126	V
23	17	027	ETB	55	37	067	7	87	57	127	W
24	18	030	CAN	56	38	070	8	88	58	130	X
25	19	031	EM	57	39	071	9	89	59	131	Y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z
27	1B	033	ESC	59	3B	073	:	91	5B	133	[
28	1C	034	FS	60	3C	074	<	92	5C	134	\
29	1D	035	GS	61	3D	075	=	93	5D	135	]
30	1E	036	RS	62	3E	076	>	94	5E	136	
31	1F	037	US	63	3F	077	?	95	5F	137	_

# Tabla ASCII Completa (96 a 177)

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr
96	60	140	'	112	70	160	p
97	61	141	a	113	71	161	q
98	62	142	b	114	72	162	r
99	63	143	c	115	73	163	s
100	64	144	d	116	74	164	t
101	65	145	e	117	75	165	u
102	66	146	f	118	76	166	v
103	67	147	g	119	77	167	w
104	68	150	h	120	78	170	x
105	69	151	i	121	79	171	y
106	6A	152	j	122	7A	172	z
107	6B	153	k	123	7B	173	{
108	6C	154	l	124	7C	174	
109	6D	155	m	125	7D	175	}
110	6E	156	n	126	7E	176	~
111	6F	157	o	127	7F	177	DEL

# Tabla ASCII Extendida (Parcial)

Dec	Hex	Chr									
128	80	Ç	129	81	ü	192	C0	À	193	C1	Á
130	82	é	131	83	â	194	C2	Â	195	C3	Ã
132	84	ã	133	85	à	196	C4	Ã	197	C5	Ä
134	86	å	135	87	ç	198	C6	ä	199	C7	Å
136	88	ê	137	89	ë	200	C8	È	201	C9	É
138	8A	è	139	8B	ï	202	CA	Ê	203	CB	Ê
140	8C	î	141	8D	ì	204	CC	Í	205	CD	Í
142	8E	Ä	143	8F	Å	206	CE	Î	207	CF	Ï
144	90	É	145	91	æ	208	D0	ð	209	D1	Ð
146	92	Æ	147	93	ô	210	D2	Ê	211	D3	Ë
148	94	ö	149	95	ò	212	D4	È	213	D5	í
150	96	ú	151	97	ù	214	D6	Í	215	D7	î
152	98	ÿ	153	99	Ö	216	D8	Ø	217	D9	Ù
154	9A	Ü	155	9B	€	218	DA	Ú	219	DB	Û
156	9C	£	157	9D	¥	220	DC	Û	221	DD	—
158	9E	đ	159	9F	ƒ	222	DE	Ï	223	DF	SS
160	A0	á	161	A1	í	224	E0	Ó	225	E1	ß
162	A2	ó	163	A3	ú	226	E2	Ô	227	E3	Ö
164	A4	ñ	165	A5	Ñ	228	E4	ö	229	E5	Õ
166	A6	ª	167	A7	º	230	E6	µ	231	E7	þ
168	A8	¿	169	A9	ŀ	232	E8	Þ	233	E9	Ú
170	AA	߱	171	AB	߲	234	EA	Ӯ	235	EB	ӹ
172	AC	ߴ	173	AD	ߵ	236	EC	܍	237	ED	܍
174	AE	߸	175	AF	߹	238	EE	ܻ	239	EF	ܻ
176	B0	ߺ	177	B1	߻	240	F0	ܺ	241	F1	ܺ
178	B2	߻	179	B3	߻	242	F2	ܻ	243	F3	ܻ
180	B4	߻	181	B5	߻	244	F4	ܻ	245	F5	ܻ
182	B6	߻	183	B7	߻	246	F6	ܻ	247	F7	ܻ
184	B8	߸	185	B9	߻	248	F8	ܻ	249	F9	ܻ
186	BA	߻	187	BB	߻	250	FA	ܻ	251	FB	ܻ
188	BC	߻	189	BD	߻	252	FC	ܻ	253	FD	ܻ
190	BE	߻	191	BF	߻	254	FE	ܻ	255	FF	ܻ

# Codificación de Información Binaria

## Fragmento de Memoria

Dirección	Contenido	Código ASCII (Dec)	ASCII
3981	01001110	78	
3982	01000101	69	
3983	01010101	85	
3984	01001101	77	
3985	01000001	65	
3986	01001110	78	
3987	01001110	78	

# Codificación de Información Binaria

## Fragmento de Memoria

Dirección	Contenido	Código ASCII (Dec)	ASCII
3981	01001110	78	N
3982	01000101	69	E
3983	01010101	85	U
3984	01001101	77	M
3985	01000001	65	A
3986	01001110	78	N
3987	01001110	78	N

# Quiz



# Contenido

## 1 Guía de Estudio

## 2 Tema 1.1. Introducción a la computación y su historia

- Definiciones
- Historia

## 3 Tema 1.2. Hardware y Software

- Capa de Abstracción
- Sistema Operativo
- Linux y Software Libre

## 4 Tema 1.3. Codificación de la Información

- Almacenamiento de Información en memoria

## 5 Bibliografía

# Bibliografía I



Circuit board, resistor and computer.

<https://images.pexels.com/photos/163100/circuit-circuit-board-resistor-computer-163100.jpeg>, n.d.  
Imagen obtenida de Pexels. Accedido: 18 de febrero de 2025.



Imagen de diagrama.

<https://time.graphics/uploadedFiles/500/fb/2e/fb2ec024c7f867170a6487ddfbaca482.png>, n.d.  
Imagen obtenida de time.graphics. Accedido: 18 de febrero de 2025.



Martin Campbell-Kelly.

The development of computer programming in britain (1945 to 1955).  
*Annals of the History of Computing*, 4(2):121–139, 1982.



Wikipedia contributors.

Ada lovelace — wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Ada\\_Lovelace](https://en.wikipedia.org/wiki/Ada_Lovelace), n.d.  
Accessed: 18 February 2025.



Flickr.

Imagen en flickr.  
[https://live.staticflickr.com/1192/5119474042\\_8ee30b5c50\\_b.jpg](https://live.staticflickr.com/1192/5119474042_8ee30b5c50_b.jpg), n.d.  
Imagen obtenida de Flickr. Accedido: 18 de febrero de 2025.



<https://cdn.sci.news>.

Antikythera.  
[https://cdn.sci.news/images/enlarge/image\\_2201\\_2e-Antikythera.jpg](https://cdn.sci.news/images/enlarge/image_2201_2e-Antikythera.jpg), 2025.  
Imagen obtenida de Sci.News. Accedido: 18 de febrero de 2025.

# Bibliografía II



Pinterest.

Pinterest pin.

<https://in.pinterest.com/pin/740490363730500075/>, 2025.

Accedido: 18 de febrero de 2025.



[reyeskiller.wordpress.com](http://reyeskiller.wordpress.com).

Máquina calculadora de pascal.

<https://reyeskiller.wordpress.com/wp-content/uploads/2013/10/maquina-calculadora-de-pascal.jpg>, 2013.

Imagen obtenida de ReyesKiller Wordpress. Accedido: 18 de febrero de 2025.



[vecteezy.com](http://vecteezy.com).

Operating systems icons: Linux, windows, android, mac, ios.

<https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fstatic.vecteezy.com%2Fsystem%2Fresources%2Fpreviews%2F002%2F774%2F872%2Foriginal%2Foperating-systems-icons-linux-windows-android-mac-ios-icons-vector.jpg&f=1&nofb=1&ipt=09b90ff463906665a18417417201dd3ff8a909a1432d2993e72dca981afe13e1&ipo=images>, 2025.

Imagen descargada de vecteezy.com.



[Wikipedia](http://wikipedia.org).

Sistema operativo — wikipedia, la enciclopedia libre.

[https://es.wikipedia.org/wiki/Sistema\\_operativo](https://es.wikipedia.org/wiki/Sistema_operativo), 2025.

Consulta: 11 de marzo de 2025.



[Wikipedia contributors](http://wikipedia.org).

John von Neumann, 2025.

[Último acceso: 19 de febrero de 2025].

# Bibliografía III



Wikipedia contributors.

Linus torvalds — wikipedia, the free encyclopedia, 2025.  
[Online; accessed 11-March-2025].