



# Lógica binaria y computacional

## Introducción

Antes de ahondar en cómo se ejecutan los distintos programas y operaciones dentro del mundo informático, debemos conocer la lógica que regula los mismos.

De esta forma es que llegamos a la lógica binaria. Una lógica totalmente matematizada, ya que tiene su base en el sistema binario.

## Sistemas numéricos

Un sistema de numeración es un conjunto de símbolos y reglas, que permiten construir todos los números válidos. En nuestra cotidianidad, tenemos contacto mayoritariamente, con el sistema decimal. En él, a partir de 10 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9), podemos construir todos los números posibles.

Como para establecer una contraposición y detallar algunos conceptos más, pensemos ahora en el sistema de numeración romano. En él, los símbolos son otros, e incluso las reglas de construcción son otras.

El sistema decimal se considera posicional, dado que un mismo dígito, no tiene el mismo valor, si está en una posición u otra. En el tres romano (III), cada uno de los símbolos unitarios (I), vale exactamente lo mismo. Entre muchas otras reglas que difieren.

## Sistema Binario

Al igual que el sistema decimal, el sistema binario también es posicional. En este caso, construiremos todos los números, sólo a partir de los símbolos 1 y 0. El resto de la lógica es idéntico a nuestro sistema, una vez alcanzado el máximo valor posible, se van agregando dígitos.

Otro sistema muy difundido dentro de la lógica computacional, es el sistema hexadecimal. En él se agregan símbolos que permiten representar, hasta el número 15 como máximo valor, agregando las letras (A, B, C, D, E y F) al sistema decimal, ya mencionado. Las respectivas equivalencias, se detallan en la siguiente tabla.

Hexadecimal	Decimal	Binario
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Su difundido uso en el campo informático, se debe a que nos permiten representar, números mucho más grandes, con menor cantidad de dígitos, a su vez, como se verá a continuación, posee una gran sencillez para ser traducido a números binarios y viceversa.

## ¿Por qué las computadoras utilizan lógica binaria?





Para establecerlo de manera sencilla, debemos tratar de imaginar una representación de los números binarios, previamente mencionados. Cada número, se puede representar a través de una señal eléctrica. Pudiendo pensar en qué dispositivos se encuentran, encendidos o apagados.

Por esto es que las primeras computadoras, del tamaño de una habitación usaron esta lógica para construir sus sistemas, y aunque usaron hardware mucho más antiguo y voluminoso, hemos mantenido los mismos principios fundamentales. Las computadoras modernas usan lo que se conoce como un transistor para realizar cálculos con números binarios. A partir de los avances tecnológicos, es que se logró desarrollar transistores de tamaños nanométricos, que permiten sistemas compactos

## Compuertas lógicas

Un paso previo a la arquitectura computacional, descrita anteriormente, fueron las compuertas lógicas. Las mismas, también pueden ser construidas a partir de transistores, y previamente también llevadas a cabo a partir de válvulas eléctricas. Estas compuertas, se confeccionan a partir de componentes, estratégicamente conectados, para lograr distintas operaciones, entre ellas encontramos: NOT (Compuerta negada), AND (producto), OR (suma), XOR (suma exclusiva).

Previo a detallar su funcionamiento, deberemos definir lo que se conoce como tabla de verdad. Se conoce bajo este término a las tablas de doble entrada, encargadas de especificar el funcionamiento de situaciones proposicionales. Demos un ejemplo:




Si creo que va a llover, salgo con paraguas de mi casa. Caso contrario, no.




¿Vá a llover?	¿Llevo paraguas?
Si	Si
No	No

De ahora en más, los únicos valores que manejaremos en nuestros análisis, son 1 y 0. Correspondientes a Sí o No, Verdadero o Falso y analizaremos distintas salidas, para diferentes combinaciones de entradas. En este caso, la entrada es la condición de si va a llover o no (la causa) mientras que la salida, es si llevo paraguas o no (la consecuencia)

Detallamos ahora las compuertas lógicas más difundidas:



<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	A 	Y									
A	Y																
0	1																
1	0																
<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	A 	Y
A	B	Y															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	A 	Y
A	B	Y															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	A 	Y
A	B	Y															
0	0	0															
0	1	1															
1	0	1															
1	1	0															
<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	A 	Y
A	B	Y															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	A 	Y
A	B	Y															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

## Bloques condicionales (si), (sinó) e (sinó si)

Más adelante, estudiaremos en detalle qué son los bloques de programación. Por lo pronto, diremos qué es una forma de dividir nuestros programas y dirigir el flujo de los mismos. Una de las más difundidas formas de hacerlo, es a través de los bloques condicionales, permitiéndonos trasladar, instancias de decisión, ante ciertas condiciones, a código de programación.

Se pueden representar de la siguiente forma:

```

si (condiciones) {
    //Instrucciones a ejecutar si la condición se cumple
} sino {
    //Instrucciones a ejecutar, si la segunda condición se cumple
}

```

Son altamente compatibles con situaciones proposicionales como las anteriormente descritas (llueve o no llueve), veamos el ejemplo:

Si creo que va a llover, salgo con paraguas de mi casa. Caso contrario, no.

¿Vá a llover?	¿Llevo paraguas?
Si	Si
No	No

```

si (va a llover) {
    llevo paraguas
} sino {
    no llevo paraguas
}

```



# Descripción Funcional de un Computador

Una computadora, es una máquina electrónica que recibe y procesa datos con la misión de llevar adelante procesos específicos. Su composición, a nivel hardware, está compuesta por una serie de circuitos integrados y otros tantos elementos relacionados que son los que permiten la ejecución de una variedad de secuencias y/o rutinas de instrucciones que indicará el usuario del mismo.

En la unidad anterior, dimos explicación a la lógica computacional, implementada a partir de transistores, los cuales sirven para fabricar compuertas lógicas y también procesadores.

El microprocesador es, probablemente, el dispositivo más importante dentro de una computadora. Es un circuito integrado y constituye la CPU o unidad central de procesamiento en una computadora. Actualmente estos componentes se fabrican en pastillas de silicio, donde se integran cientos de miles de transistores.

## La unidad central de procesamiento (CPU) y buses de comunicación

La CPU de un ordenador ejecuta las instrucciones que proporciona la memoria y procesa los datos recibidos desde los módulos de entrada. Los resultados que se originan se presentan al exterior a través de los módulos o puertos de salida.

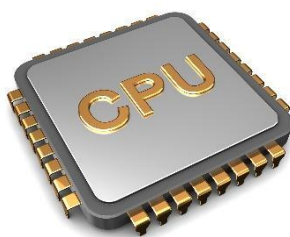
Para que estas partes que componen la estructura básica de una computadora se comuniquen entre sí, se utiliza el bus de comunicación, por ejemplo: El mismo se compone de finos conductores, que conectan todos los elementos de la máquina y que conduce también hacia los conectores externos que enlazan con los demás elementos del sistema informático.

La CPU es la parte del ordenador que gobierna su funcionamiento, pudiendo considerarse como el cerebro de la misma. La CPU se compone de las siguientes partes:

Unidad de Control. Interpreta y ejecuta las instrucciones.

ALU, Unidad Aritmético-Lógica. Realiza cálculos y comparaciones y toma decisiones lógicas.

Registro de trabajo. En ellos se almacena información de una forma temporal, realizan un seguimiento de las instrucciones y conservan la ubicación y los resultados de las operaciones efectuadas.



## Ciclos de instrucción

El ciclo de instrucción podemos dividirlo en tres partes:

1. Tomar la siguiente instrucción (fetch).
2. Decodificar la instrucción.
3. Ejecutar la instrucción.

### 1. Tomar la siguiente instrucción (fetch)

El procesador siempre tiene su registro PC (Contador de Programa) apuntando a la dirección de memoria de la siguiente instrucción a ejecutar. Durante esta fase el procesador extrae la instrucción de la memoria apuntada por el PC volcando el contenido de este registro al bus de direcciones y pidiendo una operación de lectura a la memoria.

### 2. Decodificar la instrucción

Una vez tiene el código de operación alojado en el IR (registro de instrucciones), el procesador decodifica éste para saber de qué instrucción se trata y obtiene los parámetros de la memoria (si los tuviera), incrementando el PC en una unidad por cada parámetro extraído.

### 3. Ejecutar la instrucción

Llegado a este paso, el procesador ya sabe de qué instrucción se trata y los parámetros que necesita, luego simplemente la ejecuta de la forma apropiada.

Al finalizar este paso finaliza una iteración del ciclo de instrucción, volveremos al fetch de la siguiente instrucción y así sucesivamente.

## Memoria

La memoria es un componente imprescindible del ordenador que mantiene disponibles las instrucciones para que el microprocesador o CPU pueda ejecutarlas. También la memoria se encarga de almacenar temporalmente el resultado de los procesos ejecutados.

Para almacenar información, la memoria está formada por un conjunto de casillas o células llamadas posiciones de memoria, en las que se colocan instrucciones y datos. Para que el ordenador pueda acceder a los que necesite en cada momento, cada una de las posiciones de memoria está identificada por un número, denominado dirección de memoria.





## Memoria auxiliar

La Memoria Auxiliar o Secundaria suele estar conformada por un grupo de dispositivos de almacenamiento preparados para administrar gran cantidad de información, poseyendo además de esta diferencia cuantitativa con la memoria principal o RAM, diferencias cualitativas, como por ejemplo su menor costo y su mayor lentitud.

Ejemplos de memoria Auxiliar lo son los discos rígidos, las unidades ópticas como los CDs o DVDs, memorias FLASH o diskettes. La memoria RAM recibe datos desde estos dispositivos de almacenamiento masivo.

## Memoria central

La memoria central está formada por bloques de circuitos integrados o chips capaces de almacenar, retener o "memorizar" información digital, es decir, valores binarios; a dichos bloques tiene acceso el microprocesador de la computadora. En las computadoras son utilizados dos tipos de memorias centrales:

**ROM** o memoria de sólo lectura (Read Only Memory) - viene grabada de fábrica con una serie de programas. El software de la ROM se divide en dos partes: rutina de arranque o POST (realiza el chequeo de los componentes de la computadora) y rutina BIOS (permite la activación de los periféricos de entrada/salida).

**RAM** o memoria de acceso aleatorio - es la memoria del usuario que contiene de forma temporal el programa, los datos y los resultados que están siendo usados por el usuario del computador. En general es volátil, pierde su contenido cuando se apaga el computador, es decir que mantiene los datos y resultados en tanto el bloque reciba alimentación eléctrica.

## Memorias de video

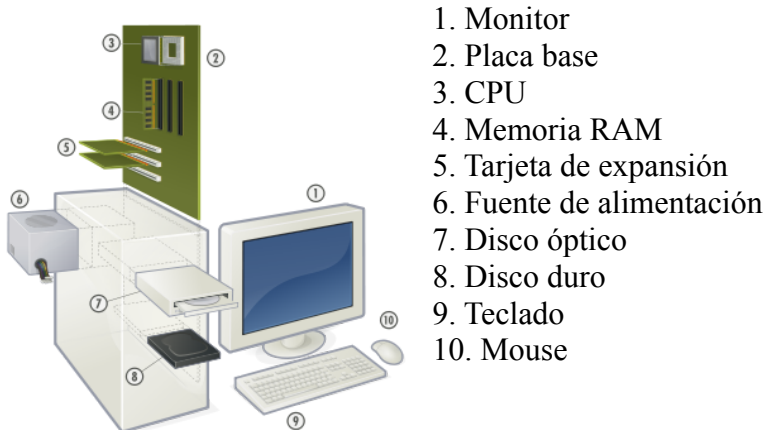
Una tarjeta de vídeo, también llamada tarjeta gráfica (entre otros nombres) tiene a su cargo el procesamiento de los datos que provienen del procesador principal (CPU o UCP) y convertirlos en información que se pueda representar en dispositivos tales como los monitores y los televisores.

Desde su concepción, las tarjetas gráficas han incluido diversas prestaciones y funciones, tales como la posibilidad de sintonizar la televisión o de capturar secuencias de vídeo de un aparato externo. Es importante notar que no se trata de un componente hallado exclusivamente en los ordenadores actuales, sino que han existido desde hace ya más de cuatro décadas y hoy en día también son parte indispensable de las consolas de videojuegos, tanto de las portátiles como de las caseras.

## Hardware

Corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Estos son los elementos de hardware típicos de una computadora.



## Software

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

### Clasificación del software

Si bien esta distinción es, en cierto modo, arbitraria, y a veces confusa, a los fines prácticos se puede clasificar al software en tres grandes tipos:

- **Software de sistema:** Su objetivo es desvincular adecuadamente al usuario y al programador de los detalles del sistema informático en particular que se use, aislandolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc.  
Dentro de este grupo, también encontramos:

- Sistemas operativos
- Controladores de dispositivos
- Herramientas de diagnóstico
- Herramientas de Corrección y Optimización
- Servidores
- Utilidades



- **Software de aplicación:** Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre muchos otros:
  - Aplicaciones para Control de sistemas y automatización industrial
  - Aplicaciones ofimáticas
  - Software educativo
  - Software empresarial
  - Bases de datos
  - Software de programación

**Software de programación:** Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica. Incluyen básicamente:

- Editores de texto
- Compiladores
- Intérpretes
- Enlazadores
- Depuradores



Utilizaremos varios de estos softwares durante el curso para crear nuestros programas.

## Estructura funcional de los computadores

Habíamos mencionado que el funcionamiento de una computadora se basa en la captura de datos que se van a procesar por medio de alguna unidad de entrada; en su almacenamiento en la unidad central de procesamiento; en la ejecución de un programa que transforma los datos de entrada en resultados, y en la comunicación de esos resultados (información) al exterior, por medio de una unidad de salida. Tanto la captura de los datos como la salida de la información se ejecutan a partir de una unidad de almacenamiento. Las unidades encargadas de estas operaciones se denominan periféricos.

Las unidades periféricas se clasifican en:

- Unidades de entrada (permiten el ingreso de datos y programas en la CPU para su tratamiento):
  - Teclado, escáner, joystick, lápiz óptico, lector de código de barras, etc.
- Unidades de salida (distribuyen los datos provenientes de la CPU al exterior por medio de una representación visual o auditiva).
  - Plotter, monitor, impresora
- Unidades de entrada/salida. MODEM, plaqueta digitalizadora de audio.





- Unidades de almacenamiento (según la instrucción que ejecuten en un momento determinado, pueden realizar una operación de entrada (recuperar un archivo) o una de salida (grabar un archivo).
- Unidades de disco o drive (discos flexibles, rígidos, CD, DVD, cintas magnéticas, etc.)

## El microprocesador

También habíamos mencionado qué se le llama microprocesador a un componente electrónico compuesto por cientos de miles de transistores integrados en una placa de silicio, el cual compone la parte principal de un CPU.

Sabemos qué se trata del elemento clave en la conformación de un ordenador. A pesar de que comúnmente se los confunde, el microprocesador no es lo mismo que el CPU. El microprocesador puede soportar a una o varias CPU, y varios microprocesadores pueden soportar a un CPU, pero en el caso de la Unidad Central de Procesamiento se trata de un concepto lógico que agrupa a todos los componentes que hacen al funcionamiento electrónico de la máquina.

La pastilla de un microprocesador tiene entre 40 y 132 patas, mediante las cuales se realizan los contactos con el exterior. Existen tres tipos de patas: dirección, control y datos. Todas ellas están conectadas a patas similares en las pastillas de memoria y de entrada/salida por medio de un conjunto de finos alambre denominados bus.

## Autómatas

La palabra autómatas es un término que se emplea en nuestro idioma para referir varias cuestiones, buena parte de ellas asociadas a la noción de automático, que como sabemos implica a aquello que funciona, en casi todo o en todo, por sus propios medios, y generalmente, como consecuencia que se le ha provisto de un mecanismo especial que le permite realizar tareas y acciones de manera autosuficiente.

Al dispositivo que presenta un mecanismo que le facilita la realización de determinados movimientos se lo denomina autómatas.

### El autómatas programable

Entendemos por Autómatas Programable, o PLC (Controlador Lógico Programable), toda máquina electrónica, diseñada para controlar en tiempo real y en medio industrial procesos secuenciales. Su manejo y programación puede ser realizada por personal eléctrico o electrónico sin conocimientos informáticos ya que se encarga solo de realizar funciones lógicas.

La función básica de los autómatas programables es la de reducir el trabajo del usuario a realizar el programa, es decir, la relación entre las señales de entrada que se tienen que cumplir para





activar cada salida, puesto que los elementos tradicionales (como relés auxiliares, de enclavamiento, temporizadores, contadores...) son internos.

## Concepto de sistema de automatizado

Un sistema automatizado consta de dos partes principales, parte de mando, parte operativa:

La Parte Operativa: es la parte que actúa directamente sobre la máquina. Son los elementos que hacen que la máquina se mueva y realice la operación deseada. Los elementos que forman la parte operativa son los accionadores de las máquinas como motores, cilindros, compresores. Y los captadores como fotodiodos, finales de carrera.

La Parte de Mando: En un sistema de fabricación automatizado el autómatas programable está en el centro del sistema. Este debe ser capaz de comunicarse con todos los constituyentes de sistema automatizado.

## Partes de un sistema automático

El grafcet (Gráfico Funcional de Etapas y Transiciones):

Ha sido especialmente diseñado para resolver problemas de automatismos secuenciales. Las acciones están asociadas a las etapas y las condiciones a cumplir a las transiciones. Este lenguaje resulta enormemente sencillo de interpretar por operarios sin conocimientos de automatismos eléctricos. Muchos de los autómatas que existen en el mercado permiten la programación en grafcet, tanto en modo gráfico o como por lista de instrucciones. También podemos utilizarlo para resolver problemas de automatización de forma teórica y posteriormente convertirlo a plano de contactos.

La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

## Fuente de alimentación

Es la encargada de convertir la tensión de la red, 220v corriente alterna, a baja tensión de corriente continua, normalmente a 24v. Siendo esta la tensión de trabajo en los circuitos electrónicos que forma el Autómata.

## Periféricos

Los periféricos no intervienen directamente en el funcionamiento del autómatas, pero sin embargo facilitan la labor del operario.





Puede coordinarse con otros autómatas o con un nivel superior de supervisión. Un sistema automatizado está formado por diversos componentes:

Accionadores: motores, cilindros, etcétera.

Preaccionadores: contactores, variadores de velocidad, válvulas, etcétera.

Captadores: detectores, interruptores, etcétera.

Paneles de mando.

Terminales de intervención.

## Unidad Central de Procesos o CPU

Se encarga de recibir las órdenes del operario por medio de la consola de programación y el módulo de entradas. Posteriormente las procesa para enviar respuestas al módulo de salidas. En su memoria se encuentra residente el programa destinado a controlar el proceso.

## Módulo de entrada

Es al que se unen los captadores (interruptores, finales de carrera, pulsadores,...). Cada cierto tiempo el estado de las entradas se transfiere a la memoria imagen de entrada. La información recibida en ella, es enviada a la CPU para ser procesada de acuerdo a la programación. Se pueden diferenciar dos tipos de captadores conectables al módulo de entradas: los pasivos y los activos.

## Módulo de salidas

Es el encargado de activar y desactivar los actuadores (bobinas de contactores, lámparas, motores pequeños,...) La información enviada por las entradas a la CPU, una vez procesada, se envía a la memoria imagen de salidas, de donde se envía a la interfaz de salidas para que estas sean activadas y a la vez los actuadores que en ellas están conectados.

## Terminal de programación

El terminal o consola de programación es el que permite comunicar al operario con el sistema.

## Los objetivos de automatización

- Mejorar la productividad de una empresa, reduciendo los costos de producción y la calidad
- Mejorar las condiciones de trabajo del personal, suprimiendo los trabajos penosos y reiterados.
- Realizar las operaciones imposibles de controlar intelectual o manualmente.
- Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.





- Simplificar el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.

## Problema - Algoritmo - Programa

### Introducción

### Problema

Es una cuestión que requiere una solución a veces no alcanzable. Un problema se puede definir como una situación en la cual se trata de alcanzar una meta y para lograrlo se deben hallar y utilizar unos objetivos y unas estrategias.

La mayoría de problemas tienen algunos elementos en común: un estado inicial; una meta, lo que se pretende lograr; un conjunto de recursos, lo que está permitido hacer y/o utilizar; y un dominio, el estado actual de conocimientos, habilidades y energía de quien va a resolverlo (Moursund, 1999).

A través del tiempo, la humanidad ha utilizado diversas estrategias generales para resolver problemas. Schunk (1997), Woolfolk (1999) y otros, destacan los siguientes métodos o estrategias de tipo general:

**Ensayo y error:** Consiste en actuar hasta que algo funcione. Puede tomar mucho tiempo y no es seguro que se llegue a una solución. Es una estrategia apropiada cuando las soluciones posibles son pocas y se pueden probar todas, empezando por la que ofrece mayor probabilidad de resolver el problema. Por ejemplo, una bombilla que no prende: revisar la bombilla, verificar la corriente eléctrica, verificar el interruptor.

**Iluminación:** Implica la súbita conciencia de una solución que sea viable. Es muy utilizado el modelo de cuatro pasos formulado por Wallas (1921): preparación, incubación, iluminación y verificación. Estos cuatro momentos también se conocen como proceso creativo. Algunas investigaciones han determinado que cuando en el periodo de incubación se incluye una interrupción en el trabajo sobre un problema se logran mejores resultados desde el punto de vista de la creatividad. La incubación ayuda a "olvidar" falsas pistas, mientras que no hacer interrupciones o descansos puede hacer que la persona que trata de encontrar una solución creativa se estanque en estrategias inapropiadas.

**Heurística:** Se basa en la utilización de reglas empíricas para llegar a una solución. El método heurístico conocido como "IDEAL", formulado por Bransford y Stein (1984), incluye cinco pasos: Identificar el problema; definir y presentar el problema; explorar las estrategias viables; avanzar en las estrategias; y lograr la solución y volver para evaluar los efectos de las actividades (Bransford &







Stein, 1984). El matemático Polya (1957) también formuló un método heurístico para resolver problemas que se aproxima mucho al ciclo utilizado para programar computadores. A lo largo de esta Guía se utilizará este método propuesto por Polya.

**Modelo de procesamiento de información:** El modelo propuesto por Newell y Simon (1972) se basa en plantear varios momentos para un problema (estado inicial, estado final y vías de solución). Las posibles soluciones avanzan por subtemas y requieren que se realicen operaciones en cada uno de ellos.

**Análisis de medios y fines:** Se funda en la comparación del estado inicial con la meta que se pretende alcanzar para identificar las diferencias. Luego se establecen submetas y se aplican las operaciones necesarias para alcanzar cada submeta hasta que se alcance la meta global. Con este método se puede proceder en retrospectiva (desde la meta hacia el estado inicial) o en prospectiva (desde el estado inicial hacia la meta).

**Razonamiento analógico:** Se apoya en el establecimiento de una analogía entre una situación que resulte familiar y la situación problema. Requiere conocimientos suficientes de ambas situaciones.

**Lluvia de ideas o Brainstorming:** Consiste en formular soluciones viables a un problema. El modelo propuesto por Mayer (1992) plantea: definir el problema; generar muchas soluciones (sin evaluarlas); decidir los criterios para estimar las soluciones generadas; y emplear esos criterios para seleccionar la mejor solución.

**Sistemas de producción:** Se basa en la aplicación de una red de secuencias de condición y acción (Anderson, 1990).

**Pensamiento lateral:** Se apoya en el pensamiento creativo, formulado por Edwar de Bono (1970), el cual difiere completamente del pensamiento lineal (lógico). El pensamiento lateral requiere que se exploren y consideren la mayor cantidad posible de alternativas para solucionar un problema. Su importancia para la educación radica en permitir que el estudiante: explore (escuche y acepte puntos de vista diferentes, busque alternativas); avive (promueva el uso de la fantasía y del humor); libere (use la discontinuidad y escape de ideas preestablecidas); y contrarreste la rigidez (vea las cosas desde diferentes ángulos y evite dogmatismos). Este es un método adecuado cuando el problema que se desea resolver no requiere información adicional, sino un reordenamiento de la información disponible; cuando hay ausencia del problema y es necesario apercebirse de que hay un problema; o cuando se debe reconocer la posibilidad de perfeccionamiento y redefinir esa posibilidad como un problema (De Bono, 1970). Según Polya (1957), cuando se resuelven problemas, intervienen cuatro operaciones mentales:

1. Entender el problema
2. Trazar un plan
3. Ejecutar el plan (resolver)





## 4. Revisar

**Algoritmos:** Consiste en aplicar adecuadamente una serie de pasos detallados que aseguran una solución correcta. Por lo general, cada algoritmo es específico de un dominio del conocimiento. La programación de computadores se apoya en este método.

### Pasos para resolver un problema

Numerosos autores de libros sobre programación, plantean cuatro fases para elaborar un procedimiento que realice una tarea específica utilizando como estrategia algoritmos. Estas fases concuerdan con las operaciones mentales descritas por Polya para resolver problemas:



### Algoritmo

Como se mencionó anteriormente, podemos definir un algoritmo como una serie de pasos detallados que aseguran una solución correcta a un problema. Los pasos son acciones y condiciones llamadas instrucciones. En general existen varios algoritmos para resolver un problema.

### Características importantes de un algoritmo

**Secuencia:** Los pasos que la componen tienen que estar ordenados, el orden es importante.

**Finita:** La secuencia en algún momento tiene que tener un final.

**Repetible:** La secuencia tiene que ser repetible, si partimos de los mismos datos de entrada, el resultado tiene que ser siempre el mismo. Como una cuenta,  $2 + 3$  siempre tiene que dar 5, lo mismo ocurre con los algoritmos.

**Correcto:** debe resolver el problema para el cuál fue diseñado.

 Importante!

Es posible que diferentes autores indiquen otras palabras claves para representar las características, lo importante es que el algoritmo cumpla con las mismas.

## Programas

Durante el transcurso del curso, vamos a resolver problemas mediante el uso de la computadora logrando crear un programa con el objetivo de resolver un problema planteado.

Un buen hábito es utilizar una metodología que nos permita resolver los problemas, por más mínimo que éste sea. A continuación, enumeramos los pasos que debemos seguir para resolver un problema diseñando un algoritmo y creando un programa a partir del mismo.

### Pasos para resolver los problemas

#### Paso 1: Análisis del problema

En principio, debemos leer y comprender qué es lo que se quiere resolver, o sea, cuál es el objetivo que se quiere lograr.

Básicamente, en esta etapa nos encargamos de identificar todos los datos que necesitamos conocer para resolver el problema (Datos de Entrada) e identificamos que información se desea obtener (Datos de Salida).

No importa cuánto tiempo se dedica al análisis, lo importante es que quede bien claro que se quiere resolver. Al principio es probable que lleve mucho más tiempo, a medida que se practica, se va adquiriendo experiencia y por consecuencia se va mejorando en el razonamiento para realizar este análisis.

Para realizar este análisis utilizaremos el esquema EPS: Entrada-Proceso-Salida y aplicaremos el modelo TOP-Down.

 Importante!

Si no está claro el objetivo, no será posible crear un programa que resuelva el problema planteado.

#### Paso 2: Diseño del Algoritmo

Una vez que tenemos analizado el problema, identificadas las entradas y las salidas que se desean obtener. Comenzamos el diseño del algoritmo que consiste básicamente en detallar todos los pasos a seguir para obtener el resultado deseado.

#### Paso 3: Codificación – Crear el Programa o Código Fuente

La solución diseñada es luego traducida a un lenguaje de programación, respetando la sintaxis del mismo para lograr crear un programa ejecutable.





#### Paso 4: Verificación y Depuración

El programa creado se ejecuta, se comprueba rigurosamente y se eliminan todos los errores que se detectan (en general llamados “bugs” entre los programadores).

#### Paso 5: Documentación y mantenimiento

Finalmente generamos la documentación que nos permite tener claro el objetivo. Se suele documentar supuestos considerados para resolver el problema o notas que consideramos importantes recordar.



Lectura requerida:

Joyanes Aguilar, Luis: Fundamentos de Programación Editorial Mc. Graw Hill.  
Capítulo 2 – Tema 2.2.1 - Pág.(46)



Lectura ampliatoria:

Las 5 fases para resolver problemas en Programación

“¿Cuál es la forma adecuada para resolver problemas en programación?”

Resolución de problemas de programación

[https://es.wikipedia.org/wiki/Resoluci%C3%B3n\\_de\\_problemas\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/Resoluci%C3%B3n_de_problemas_de_programaci%C3%B3n)

