**Algoritmo Cotidiano**

**Lo primero que hago al despertar:**

* Levantarme de la cama
* Mirar la hora
* Tomar la toalla
* Dirigirme al baño
* Quitarme la ropa
* Abrir la ducha
* En jabonarme
* Enjuagar
* Tomar el champo y aplicarlo
* Volver a enjuagar
* Cerrar la ducha
* Tomar la toalla
* Secarme la cara, el pelo y ultimo el cuerpo

**Lenguaje de Programación**

Un**lenguaje de programación** es un lenguaje informático especialmente diseñado para describir el conjunto de acciones consecutivas o instrucciones que un equipo informático debe ejecutar.

De esta manera, un lenguaje de programación pasa a ser la manera práctica de lograr que el equipo ejecute **las acciones que el usuario desea**. Los lenguajes de programación obedecen a un conjunto de reglas que permiten expresar las instrucciones que serán interpretadas.

**Ejemplos de lenguajes de programación**

|  |  |
| --- | --- |
| ADA | PHP |
| BASIC | Perl |
| C | Prolog |
| C++ | ASP |
| Cobol | Action Script |
| Fortran | Python |
| Java | JAVA |
| MATLAB | Java Script, etc. |
| LISP | Assembler |
| Pascal | Logo |

## Tipos de lenguaje de programación

Existen diferentes clases de lenguajes de programación, como los **funcionales o procedimentales**, los imperativos, los lógicos, los híbridos, los orientados a objetos.

Los primeros crean programas **mediante funciones** y recibe como entrada el resultado de otras funciones. Los segundos crean programas mediante una secuencia de comandos agrupados en bloques; las órdenes son condicionales, pues le permiten al programa volver a un bloque de comandos si se cumple cierta condición.

Referencia: https://www.ejemplos.co/20-ejemplos-de-lenguaje-de-programacion/#:~:text=Un%20lenguaje%20de%20programaci%C3%B3n%20es,Java%2C%20C%2B%2B%2C%20Python.

**Algoritmo**

 Se puede entender un algoritmo como una secuencia de pasos finitos bien definidos que resuelven un problema. Por ejemplo, la ejecución de tareas cotidianas tan simples como cepillarse los dientes, lavarse las manos o seguir el manual de instrucciones de armado de un mueble, se pueden ver como un algoritmo.

Desde el punto de vista informático un algoritmo es cualquier procedimiento computacional bien definido que parte de un estado inicial y un valor o un conjunto de valores de entrada, a los cuales se les aplica una secuencia de pasos computacionales finitos, produciendo una salida o solución. Se puede considerar al algoritmo como una herramienta para resolver un cálculo computacional bien especificado

## Características y clasificaciones de los algoritmos

Independientemente de la clasificación y el tipo de algoritmo, básicamente todos están compuestos de tres partes principales:

* **Entrada**: Se trata del conjunto de datos que el algoritmo necesita como insumo para procesar.
* **Proceso**: Son los pasos necesarios aplicados por el algoritmo a la entrada recibida para poder llegar a una salida o resolución del problema.
* **Salida**: Es el resultado producido por el algoritmo a partir del procesamiento de la entrada una vez terminada la ejecución del proceso.

Referencia: https://ude.edu.uy/que-son-algoritmos/#:~:text=Se%20puede%20entender%20un%20algoritmo,pueden%20ver%20como%20un%20algoritmo.

**Lenguaje de Maquina**

Lenguaje de máquina es un [lenguaje de programación](https://muytecnologicos.com/diccionario-tecnologico/lenguaje-de-programacion) que se compone de instrucciones binarias y códigos numéricos, diseñado para ser entendido directamente por el [hardware](https://muytecnologicos.com/diccionario-tecnologico/hardware) de una computadora. Estas instrucciones se ejecutan directamente por la CPU (procesador) y permiten al usuario comunicarse con la máquina a un nivel básico para controlar su funcionamiento.

Entender el lenguaje de máquina requiere familiarizarse con los elementos básicos del lenguaje. Esto es necesario para programar una computadora para realizar tareas específicas. El lenguaje de máquina se compone de instrucciones binarias, que son codificadas en una secuencia de 0 y 1. Estas instrucciones le indican a la computadora qué hacer, y cómo hacerlo.

El 1 y el 0, son números que forman parte del conocido sistema binario un lenguaje que es utilizado por todas las computadoras y que se conoce con el nombre de lenguaje máquina. Son ejemplos de lenguaje de máquina los siguientes: 0010, 0000, 1001, 1001, 10001, 1110.

**Lenguaje de bajo nivel (ensamblador)**

En primer lugar, debes saber que un lenguaje de bajo nivel es un [lenguaje de programación](https://www.epitech-it.es/cuantos-lenguajes-existen/). Un lenguaje de programación de bajo nivel, **es aquel en el que sus instrucciones ejercen un control directo sobre el hardware y por lo tanto están condicionados por la estructura física** de las computadoras que lo soportan.

El ensamblador es el **lenguaje de bajo nivel más sencillo de utilizar, ya que surge ante la gran dificultad que supone programar en código binario o máquina**. En ensamblador se pueden **expresar las instrucciones directas a la máquina en un lenguaje más natural y sencillo** para el programador y que puede ser comprendido por la propia máquina.

Referencia: https://miformacion.eu/blog/lenguaje-bajo-nivel-en-programacion/#:~:text=Un%20lenguaje%20de%20programaci%C3%B3n%20de,o%20m%C3%A1quina%20que%20lo%20soporta.

**Lenguaje de alto nivel**

Cuando hablamos de un lenguaje de alto nivel nos referimos al tipo de lenguaje de programación que no expresa los algoritmos teniendo en cuenta la capacidad que tienen las máquinas para ejecutar órdenes, sino al que se utiliza **teniendo en cuenta las capacidades cognitivas de los seres humanos**. Existen desde la década de los 50 y nacieron con el objetivo de ir más allá respecto a las limitaciones de los [lenguajes de bajo nivel](https://www.universidadviu.com/es/actualidad/nuestros-expertos/lenguaje-de-bajo-nivel-caracteristicas-y-funciones), permitiendo a los usuarios resolver problemas de una forma sencilla y rápida. Desde entonces han aparecido distintos lenguajes de alto nivel y es conveniente conocer, al menos, los principales.

Se tratan de lenguajes **independientes de la arquitectura del ordenador** y de su hardware. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes migrar de una máquina a otra sin ningún tipo de problema. Depende del lenguaje pueden existir unas u otras limitaciones, pero siempre que exista un intérprete o compilador para ese lenguaje en el lugar donde lo quieras ejecutar, funcionará.

Referencia: https://www.universidadviu.com/es/actualidad/nuestros-expertos/lenguaje-de-alto-nivel-los-mas-utilizados#:~:text=Cuando%20hablamos%20de%20un%20lenguaje,cognitivas%20

de%20los%20seres%20humanos.

https://desarrolloweb.com/articulos/2358.php#:~:text=probar%20y%20mantener.-,Lenguajes%20de%20alto%20nivel,todos%20los%20lenguajes%20de%20programaci%C3%B3n.

**Algoritmos cualitativos**

Los algoritmos cualitativos son aquellos que se realizan por medio de las palabras, lo que quiere decir que las órdenes vienen dadas en forma verbal. Por ejemplo, una receta de cocina.

Los algoritmos cualitativos se emplean con frecuencia en la vida cotidiana para resolver problemas. Por ejemplo: las instrucciones de uso que traen los equipos electrónicos, las instrucciones para el montaje de un equipo, las técnicas de laboratorio para evaluar ácidos, etc.

Referencia: https://www.lifeder.com/algoritmos-cualitativos/

**Algoritmos cuantitativos**

Los **algoritmos cuantitativos** son aquellos algoritmos que utilizan operaciones algebraicas y cálculos numéricos específicos para definir un proceso, obteniendo valores concretos. Por ejemplo, el resultado de una resta o una multiplicación.

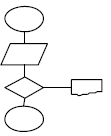
En las ciencias de la computación, en las matemáticas y otras disciplinas afines, un algoritmo es un conjunto finito y ordenado de instrucciones que permite efectuar una actividad por medio de pasos sucesivos que no generan dudas a quien deba ejecutar estas acciones, llevando a la solución de un determinado problema.

Resulta significativo destacar la importancia de los algoritmos, porque representan un elemento básico para la informática, la robótica y las matemáticas, ya que por medio de ellos se logra ordenar las ideas. Ellos llevan a la ejecución correcta de las actividades y a ideas con un orden, concernientes a cualquier aspecto.

Referencia: <https://www.lifeder.com/algoritmos-cuantitativos/>

**Lenguajes algorítmicos gráficos**

Es la representación gráfica de las operaciones que realiza un algoritmo también llamado diagrama de flujo. El diagrama de flujo se caracteriza por la forma detallada como representa de forma gráfica los pasos a seguir para encontrar la solución del problema propuesto.



Referencia: <https://sites.google.com/a/misena.edu.co/wiki-adsi/home/lenguajes-algoritmicos-graficos>

**Lenguajes no gráficos**

**No Gráficos**: También podemos representar los algoritmos con palabras, en lugar de utilizar gráficos. Esta descripción se realiza en un lenguaje intermedio llamado [**seudocódigo**](https://desarrolloweb.com/articulos/pseudocodigo.html), que permite especificar las operaciones de manera descriptiva que está mucho más cerca del propio lenguaje de programación.



Referencia: https://desarrolloweb.com/home/algoritmo

**Pseudocódigo**

El pseudocódigo es una forma de representar código, como algoritmos, funciones y otros procesos, utilizando una combinación de lenguaje natural y elementos similares al lenguaje de programación.

Se llama «pseudocódigo» porque no es realmente ejecutable. En cambio, es una forma de que los humanos comprendan y planifiquen la lógica de la programación — describir los pasos de un programa de forma que sea fácil de entender para los humanos, sin dejar de ser lo suficientemente detallado como para convertirse rápidamente en un lenguaje de programación específico.

REFERENCIA: https://kinsta.com/es/base-de-conocimiento/que-es-pseudocodigo/#:~:text=El%20pseudoc%C3%B3digo%20es%20una%20forma%20de%20representar%20c%C3%B3digo%2C%20como%20algoritmos,similares%20al%20lenguaje%20de%20programaci%C3%B3n.

**Definición del problema**

Un problema es un asunto o **cuestión que se debe solucionar o aclarar**. Puede ser una **contradicción**o un**conflicto**entre lo que es y lo que debe ser, una **dificultad**o un**inconveniente**para la consecución de un fin. >Un disgusto, una molestia o una preocupación.

**Pasos para redactar el planteamiento del problema**

1. Definir la variable principal.
2. Explicar el **problema** desde lo general a lo particular.
3. Explicar las causas de la situación descrita.
4. Explicar las consecuencias del **problema**.
5. **Redactar** el pronóstico de la situación.
6. **Redactar** el control del pronóstico.

Referencia:https://www.significados.com/problema/#:~:text=Un%20problema%20es%20un%20asunto,una%20molestia%20o%20una%20preocupaci%C3%B3n.

**Análisis del problema**

El análisis del problema define la **naturaleza, el alcance, las causas y las consecuencias** específicos del tema de la campaña para determinar el mejor modo de abordar esa cuestión y qué que puede cambiarse. Las pruebas fácticas, pertinentes y decisivas brindan una base sólida para cualquier estrategia de campaña, poniendo de manifiesto las formas en que puede lograrse el cambio previsto. Sin una comprensión clara y con base empírica de las causas y efectos de la cuestión, será difícil elaborar tácticas sólidas y una teoría del cambio.

**El método propone el siguiente orden en el tratamiento de problemas:**

1. **Problema**: Identificar el **problema**.
2. Observación: Analizar las características del **problema**.
3. **Análisis**: Determinar las causas principales.
4. Plan de acción: Diseñar un plan para eliminar las causas.
5. Acción: Actuar para eliminar las causas.

Referencia: https://www.endvawnow.org/es/articles/1176-herramientas-para-el-anlisis-del-problema.html

**Diseño del algoritmo**

El diseño de algoritmos o **algorítmica** es un área central de las ciencias de la computación, también muy importante para la [investigación de operaciones](https://es.wikipedia.org/wiki/Investigaci%C3%B3n_de_operaciones) (también conocida como investigación operativa), en [ingeniería del software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_del_software) y en otras disciplinas afines.

Referencia:https://es.wikipedia.org/wiki/Dise%C3%B1o\_de\_algoritmos#:~:text=El%20dise%C3%B1o%20de%20algoritmos%20o,y%20en%20otras%20disciplinas%20afines.

**Codificación**

Para entender qué es la codificación, por lo tanto, primero debe quedar en claro qué es un código. Se trata de una combinación de [**signos**](https://definicion.de/signos/) (números, letras, etc.) que tiene un cierto valor en el marco de un sistema o que posibilita la reformulación y la comprensión de un mensaje secreto. 

La codificación es un proceso mediante el cual se asignan claves numéricas a las respuestas de preguntas abiertas de un cuestionario. Este proceso se encuentra integrado a una etapa de procesamiento integral de la información, que incluye la captura, validación y explotación de la información.

Referencias: https://www.cepal.org/sites/default/files/presentations/sistema-codificacion-automatizada.pdf

**https://definicion.de/codificacion/**

**Compilación de ejecución**

Es una técnica para mejorar el rendimiento de sistemas de programación que compilan a bytecode, consistente en traducir el bytecode a código máquina nativo en tiempo de ejecución. La compilación en tiempo de ejecución se construye a partir de dos ideas anteriores relacionadas con los entornos de ejecución: la compilación a bytecode y la compilación dinámica.

En el caso del compilador, primero debe traducirse todo el código antes de poder resolver los errores o iniciar la aplicación. Sin embargo, una vez que se ejecuta el programa, los servicios del compilador ya no son necesarios, mientras que el intérprete continúa utilizando los recursos informáticos.

Referencias: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/compilador-e-interprete/#:~:text=En%20el%20caso%20del%20compilador,contin%C3%BAa%20utilizando%20los%20recursos%20inform%C3%A1ticos.

**Prueba y depuración**

Las prácticas de codificación y diseño le ayudan a crear programas de calidad y deben ir seguidas de pruebas exhaustivas de los programas. Tiene que prestar especial atención a la fase de prueba de desarrollo para que:

* Su programa sea completamente operativo tras el menos número posible de ejecuciones de prueba, minimizando el tiempo y el coste del desarrollo del programa.
* Su programa cumpla todos los objetivos de diseño antes de lanzarse al trabajo de producción.
* Su programa incluya comentarios suficientes permitir a quienes utilizan y se encargan del mantenimiento del programa realizar tareas sin ayuda adicional.

El proceso de prueba suele desvelar *bugs* (o errores), un término genérico que abarca todo lo que hace su programa que no se esperaba que hiciese. El proceso de suprimir estos errores del programa se conoce como *depuración*.

Referencias: <https://www.ibm.com/docs/es/developer-for-zos/9.5.1?topic=SSQ2R2_9.5.1/com.ibm.etools.pl1.win.doc/topics/debug.htm>

**Documentación**

La documentación es sobre todo información necesaria para conseguir un grado de eficacia en el desarrollo de cualquier ámbito profesional. La documentación como herramienta de trabajo dentro de los medios de comunicación, de los que la publicidad forma parte, es fundamental.

Referencia: <https://rua.ua.es/dspace/bitstream/10045/15240/6/UNIDAD-1-Documentaci%C3%B3n-Informativa.pdf>

**Tipos de datos**

En programación, un **tipo de dato** es la clasificación de un dato según sus características. Puede tratarse de una cadena, de un texto, de números, etc., y un **valor** es cualquier clase de dato que se halle dentro de un tipo de dato.

Entre los tipos de datos y valores se encuentran:

* Numéricos: incluyen números positivos, negativos; cifras decimales, naturales, etc.
* Texto: letras, caracteres, símbolos que representan otros idiomas.
* Valores *booleanos*: que son fundamentales para establecer condiciones de verdad o falsedad.
* Listas: para almacenar múltiples elementos de un mismo tipo.

Referencia: <https://edu.gcfglobal.org/es/conceptos-basicos-de-programacion/valor-y-tipos-de-datos/1/>

**Expresiones**

En programación, una expresión es una combinación de constantes, variables o funciones, que es interpretada de acuerdo a las normas particulares de precedencia y asociación para un lenguaje de programación en particular.

Referencia: https://es.wikipedia.org/wiki/Expresi%C3%B3n\_(inform%C3%A1tica)#:~:text=En%20programaci%C3%B3n%2C%20una%20expresi%C3%B3n%20es,lenguaje%20de%20programaci%C3%B3n%20en%20particular.

**Operadores y operandos**

## Qué son operadores

Antes de comenzar a ver cuáles son los operadores conviene explicar qué son, de un modo general, aunque creo que la mayoría de las personas lo tienen en mente del propio estudio de materias como las matemáticas.

Los operadores son elementos que relacionan de forma diferente, los valores con los que trabajamos en los lenguajes de programación. En los lenguajes de programación usamos los operadores para manipular los valores y transformarlos, con el objetivo de realizar los objetivos de los programas.

## Qué son los operandos

Los operandos son los valores que se utilizan para alimentar los operadores. Por ejemplo, en el operador suma necesitamos dos operandos para sumar ambos valores.

Los operandos pueden venir almacenados en variables o constantes, pero también pueden ser simplemente valores que están escritos de manera literal en el código del programa.

Referencia: https://desarrolloweb.com/articulos/operadores-operandos-programacion

**Identificadores (variables y constantes)**

Son elementos textuales también llamados símbolos que nombran entidades del lenguaje. Algunas de las entidades que un identificador puede denotar son las variables, las constantes, los tipos de datos, las etiquetas, las subrutinas y los paquetes.  
  
Una constante es un dato cuyo valor no puede cambiar durante la ejecución del programa.  
  
Una variable es nombre asociado a un elemento de datos que está situado en posiciones contiguas de la memoria principal y su valor puedes cambiar durante la ejecución del programa.

Referencia: <https://sites.google.com/a/misena.edu.co/wiki-adsi/home/identificadores-variables-y-constantes>

**Dispositivos de entrada**

En informática, se conoce como dispositivos de entrada o dispositivos de alimentación (input) a los aparatos que permiten ingresar información al sistema informático, ya sea proveniente del usuario o de otra computadora. Por ejemplo: el teclado, el mouse o el micrófono.

Referencia: <https://concepto.de/dispositivos-de-entrada/#:~:text=En%20inform%C3%A1tica%2C%20se%20conoce%20como,el%20mouse%20o%20el%20micr%C3%B3fono>.

**Dispositivos de salida**

En informática, los dispositivos de salida son aquellos aparatos que forman parte del hardware y que transmiten la información proveniente de una computadora hacia al usuario u otra computadora o red. Por ejemplo: monitor, parlantes, impresora.

Referencia: <https://www.ejemplos.co/ejemplos-de-dispositivos-de-salida/#:~:text=En%20inform%C3%A1tica%2C%20los%20dispositivos%20de,%3A%20monitor%2C%20parlantes%2C%20impresora>.

**Unidad central de procesamiento (c.p.u)**

La unidad central de procesamiento (conocida por las siglas CPU, del inglés Central Processing Unit) o procesador es un componente del hardware dentro de un ordenador, teléfonos inteligentes, y otros dispositivos programables.

Referencia: https://es.wikipedia.org/wiki/Unidad\_central\_de\_procesamiento#:~:text=La%20unidad%20central%20de%20procesamiento,inteligentes%2C%20y%20otros%20dispositivos%20programables.

**UNIDAD DE CONTROL**

La **unidad de control** (UC), en inglés: *control unit* (CU), es uno de los tres bloques funcionales principales en los que se divide una [unidad central de procesamiento](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento) ([CPU](https://es.wikipedia.org/wiki/Central_Processing_Unit)). Los otros dos bloques son la [unidad de proceso](https://es.wikipedia.org/wiki/Unidad_de_proceso) y el [bus de entrada/salida](https://es.wikipedia.org/wiki/Bus_(inform%C3%A1tica)).

Su función es buscar las instrucciones en la [memoria principal](https://es.wikipedia.org/wiki/Memoria_principal), decodificarlas (interpretación) y ejecutarlas, empleando para ello la unidad de proceso.

Existen dos tipos de unidades de control: las **cableadas**, usadas generalmente en máquinas sencillas y las **microprogramadas**, propias de máquinas más complejas. En el primer caso, los componentes principales son el circuito de [lógica secuencial](https://es.wikipedia.org/wiki/L%C3%B3gica_secuencial), el de control de estado, el de [lógica combinacional](https://es.wikipedia.org/wiki/L%C3%B3gica_combinacional) y el de emisión de reconocimiento de señales de control. En el segundo caso, la microprogramación de la UC se encuentra almacenada en una micromemoria, a la cual se accede de manera secuencial para posteriormente ir ejecutando cada una de las microinstrucciones.

Referencia: <https://es.wikipedia.org/wiki/Unidad_de_control>

**Unidad aritmético – lógica**

Una ALU es una unidad aritmética lógica. Es una unidad de creación fundamental de cualquier procesador de CPU en el mundo informático actual. Lógicamente, su rendimiento y aplicación son relativamente fáciles de entender.

Referencia: <https://www.arrow.com/es-mx/categories/standard-and-specialty-logic/specialty-logic/alu>

**Memoria Central (interna)**

MEMORIA CENTRAL: ¿Qué es la memoria central? Es el dispositivo encargado de almacenar de forma temporal las instrucciones y los datos que empleara la CPU del ordenador en la ejecución de algún proceso. Por ella pasan todas las operaciones de entrada/salida, así como los resultados parciales o finales del proceso.

Referencia:https://www.infor.uva.es/~cevp/FI\_II/fichs\_pdf\_teo/Trabajos\_Ampliacion/Memoria\_Central.pdf

**Memoria RAM**

¿Qué es la memoria de RAM?

Las siglas RAM significan memoria de acceso aleatorio, y es uno de los elementos más fundamentales de la informática. La memoria RAM es un banco de memoria temporal donde su ordenador almacena los datos que necesita recuperar rápidamente

Referencia: <https://www.avast.com/es-es/c-what-is-ram-memory#:~:text=Las%20siglas%20RAM%20significan%20memoria,datos%20que%20necesita%20recuperar%20r%C3%A1pidamente>.

**Memoria ROM**

Sus siglas tienen su origen del inglés Read Only Memory o memoria de solo lectura. Y es que la memoria ROM se utiliza para almacenar datos que no cambian o que lo hacen poco en el tiempo, como por ejemplo el sistema de arranque del dispositivo o BIOS.

Referencia: https://www.lg.com/es/posventa/microsites/movil/memoria-ram-flash-rom#:~:text=La%20memoria%20ROM,arranque%20del%20dispositivo%20o%20BIOS.

**Memoria auxiliar (externa)**

**La memoria secundaria o auxiliar es en un**[**ordenador**](https://www.definicionabc.com/tecnologia/ordenador.php)**el conjunto de dispositivos que permiten almacenar datos complementarios a aquellos que se archivan en la memoria**

**Si se habla de memoria auxiliar, puede hacerse referencia a los antiguos diskettes, un disco o**[**CD-ROM**](https://www.definicionabc.com/tecnologia/cd-rom.php)**, DVD, unidades de memoria flash, discos Zip y otras alternativas.** Cada una de ellas presenta diversas características que la adecuan a uno u otro propósito. Además, en la mayoría de los casos pueden adquirirse con distintos tamaños, de acuerdo con las necesidades del usuario

REFERENCIA: <https://www.definicionabc.com/tecnologia/memoria-auxiliar.php>