**WIKI GLOSARIO TECNICO**

**Lenguaje de programación:**

Un lenguaje de programación, es un conjunto de instrucciones a través del cual un usuario humano interactúa con las computadoras, a través de algoritmos e instrucciones escritas en una sintaxis que la computadora pueda entender e interpretar. Un lenguaje de programación cuenta de tres características importantes que son:

Sintaxis: el conjunto de símbolos y reglas para formar sentencias.

Semántica: las reglas para transformar sentencias en instrucciones lógicas.

Pragmática: utilizando las construcciones particulares del lenguaje.

Fuente: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/#qu%C3%A9-es-un-lenguaje-de-programaci%C3%B3n>

**Algoritmo:**

Se puede entender un algoritmo como una secuencia de pasos finitos que resuelven un problema, como por ejemplo tareas cotidianas como lavarse los dientes, las manos o seguir el manual de instrucciones de armado de un mueble.

Desde el punto de vista en la programación 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

Fuente: <https://ude.edu.uy/que-son-algoritmos/>

**Lenguaje de máquina:**

Es el lenguaje que pueden ejecutar los ordenadores y otros sistemas de computación, reúne las instrucciones que recibe la maquina a la hora de llevar los procesos para el cual haya sido programado. Hay que tener en cuenta que el lenguaje de maquina no da espacio para la interpretación humana.

En el momento en el que las instrucciones llegan al ordenador se pueden apreciar dos partes fundamentales: la operación y el comando. Esto es lo que deriva en que el equipo informático sepa cuál es la instrucción que se le ha proporcionado y cómo es la función que tendrá que poner en acción. Estos datos permitirán que el sistema pueda saber dónde consultar para poner en marcha el proceso o el lugar en el cual proceder con el almacenado de los datos.

Fuente: <https://www.universidadviu.com/es/actualidad/nuestros-expertos/conociendo-el-lenguaje-de-maquina>

**Lenguaje de bajo nivel (ensamblador):**

Lenguaje de programación que utiliza palabras clave e instrucciones que son un poco más complejas que los unos y ceros del lenguaje máquina. En sentido técnico, el lenguaje de bajo nivel incluye el lenguaje máquina, pero se refiere más comúnmente a un lenguaje ensamblador que emplea símbolos para crear instrucciones de máquina más fáciles de leer y entender por parte de los programadores. Cada instrucción de un lenguaje ensamblador representa un solo comando para el procesador, lo que permite al desarrollador solamente un bajo nivel de abstracción en cuanto a funciones rutinarias, tales como acceso a memoria y almacenamiento de registro, por lo que tales operaciones requieren suma atención por parte del desarrollador.

Fuente: <https://support.esri.com/es-es/gis-dictionary/low-level-language>

**Lenguaje de alto nivel:**

Lenguaje de programación que utiliza palabras clave e instrucciones similares a las expresiones del lenguaje humano o las matemáticas y que, por tanto, resulta más fácil de comprender y usar para las personas. Un lenguaje de alto nivel debe su nombre al alto nivel de abstracción que ofrece a los desarrolladores respecto a funciones de procesadores de bajo nivel, tales como acceso a memoria y almacenamiento de registro, por lo que tales operaciones no requieren la atención del desarrollador.

Fuente: <https://support.esri.com/es-es/gis-dictionary/high-level-language>

**Algoritmos cualitativos**

Los algoritmos cualitativos son todos aquellos algoritmos en los que los pasos que lo componen se describen de una forma narrada con un lenguaje natural. Un algoritmo es una secuencia de pasos ordenados y lógicos que se llevan a cabo con el objeto de resolver un problema determinado.

Se tiene una entrada y un estado inicial, al seguir los pasos en el orden indicado se llegará a un estado final. Obteniéndose una solución

Fuente: <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.

Los algoritmos cuantitativos son muy precisos en sus pasos e instrucciones, es una necesidad para evitar la ambigüedad y son definidos a la perfección, es decir debe obtenerse el mismo resultado siempre, sin importar cuanta veces se repita.

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

**Lenguajes algorítmicos gráficos**

Consiste en escribir los algoritmos mediante una representación gráfica de las operaciones que deben realizarse para su consecución. El lenguaje más habitual aquí sería el diagrama de flujo.

Fuente: <https://desarrolloweb.com/home/algoritmo>

**Lenguajes no gráficos**

También podemos representar los algoritmos con palabras, en lugar de utilizar gráficos. Esta descripción se realiza en un lenguaje intermedio llamado pseudocódigo, que permite especificar las operaciones de manera descriptiva que está mucho más cerca del propio lenguaje de programación.

Fuente: <https://desarrolloweb.com/home/algoritmo>

**Pseudocódigo**

El pseudocódigo es una forma de expresar los distintos pasos que va a realizar un programa, de la forma más parecida a un lenguaje de programación. Su principal función es la de representar por pasos la solución a un problema o algoritmo, de la forma más detallada posible, utilizando un lenguaje cercano al de programación. El pseudocódigo no puede ejecutarse en un ordenador ya que entonces dejaría de ser pseudocódigo, como su propio nombre indica, se trata de un código falso (pseudo = falso), es un código escrito para que lo entienda el ser humano y no la máquina.

Fuente: <https://openwebinars.net/blog/que-es-pseudocodigo/>

**Definición del problema**

Definir el problema permite pasar de un problema abstracto a un problema concreto permitiendo las instancias y soluciones estén codificadas en forma de lenguajes formales.

Los problemas abstractos suelen definirse en dos partes: en la primera se describe al conjunto de instancias y en la segunda se describe la solución esperada para cada instancia.• Análisis del problema

Fuente: <https://es.wikipedia.org/wiki/Problema_computacional>

**Análisis del problema**

Para analizar el problema que se quiera resolver y llegar a una solución satisfactoria se necesita una buena definición del problema, junto con una descripción detallada de las especificaciones en los datos de entrada y de salida, cumpliendo con estos requerimientos se puede pasar al diseño de algoritmo

Fuente: <https://plataforma.josedomingo.org/pledin/cursos/curso_cpp1/curso/u01/#:~:text=Definici%C3%B3n%20o%20an%C3%A1lisis%20del%20problema,de%20un%20problema%20dado%3A%20algoritmo>.

**Diseño del algoritmo**

Es la etapa donde se construye el algoritmo que resolverá el problema que ya definimos y analizamos anteriormente, sabiendo que son independientes tanto del lenguaje de programación como del ordenador que los ejecuta.

Cuenta con tres características importantes:

* Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
* Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
* Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

Fuente: <https://plataforma.josedomingo.org/pledin/cursos/curso_cpp1/curso/u01/#:~:text=Definici%C3%B3n%20o%20an%C3%A1lisis%20del%20problema,de%20un%20problema%20dado%3A%20algoritmo>.

**Codificación**

Se llama codificación a la transformación de la formulación de un mensaje a través de las reglas o normas de un código o lenguaje predeterminado. Conocemos a la codificación como cualquier operación que implique la asignación de un valor de símbolos o caracteres a un determinado mensaje verbal o no verbal con el propósito de transmitirlo a otros individuos o entidades que compartan el código.

Fuente: <https://www.definicionabc.com/tecnologia/codificacion.php>

**Compilación y ejecución**

La compilación es el proceso por el cual se traducen las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina. Además de un traductor, se pueden necesitar otros programas para crear un programa objeto ejecutable. Un programa fuente se puede dividir en módulos almacenados en archivos distintos. La tarea de reunir el programa fuente a menudo se confía a un programa distinto, llamado preprocesador. El preprocesador también puede expandir abreviaturas, llamadas a macros, a proposiciones del lenguaje fuente.

El modelo de ejecución especifica el comportamiento de los elementos de dicho lenguaje por lo que aplicándolo se puede derivar el comportamiento de un programa escrito en términos del lenguaje en cuestión, por ejemplo, cuando una programadora "lee" cierto código, ella puede recorrerlo línea a línea en su mente y analizar lo que hace cada una de estas líneas de código, emulando en su mente el comportamiento de dicho programa. De esta forma la programadora está aplicando el modelo de ejecución al código en su mente, lo que resulta en el comportamiento de dicho código. Cada lenguaje de programación tiene su propio modelo de ejecución, el cual determina la manera en cómo las unidades de trabajo (indicadas por la sintaxis de programación) serán organizadas para su ejecución.

Fuente: <https://es.wikipedia.org/wiki/Compilador>

<https://es.wikipedia.org/wiki/Modelo_de_ejecuci%C3%B3n>

**Prueba y depuración**

Probar: Es la ejecución del proceso de pruebas; consistente en la planificación, preparación y evaluación de un producto de software y sus artefactos de trabajo, para determinar si satisfacen los requerimientos, demostrar que son aptos para el propósito, y detectar defectos.

Depurar: Es la ejecución del proceso de depuración; consistente en el análisis de fallas y la reparación de defectos (bugs).

Fuente: <https://diariobug.com/diferencia-entre-probar-y-depurar/>

**Documentación**

La documentación es la ciencia que consiste en documentar, y se encuentra identificada por el procesamiento de información que otorgará datos específicos sobre un tema determinado; de acuerdo a esto puede identificarse como una técnica instrumental y auxiliar, para lograr informar a numerosas personas sobre un tema en específico. Un documento no es más que un texto redactado que tienen la finalidad de servir de soporte para el relato de una circunstancia, también se puede describir como una carta que contenga datos que deben ser identificados para comprobar una información.

Fuente: <https://es.wikipedia.org/wiki/Documentaci%C3%B3n>

**Tipos de datos**

En ciencias de la computación, un tipo de dato informático o simplemente tipo es un atributo de los datos que indica al ordenador (y/o al programador/programadora) sobre la clase de datos que se va a manejar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.

Los tipos de datos más comunes son: números enteros, números con signo (negativos), números de coma flotante (decimales), cadenas alfanuméricas (y unicodes), estados, etc.

Fuente: <https://es.wikipedia.org/wiki/Tipo_de_dato>

**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. Como en matemáticas, la expresión es su valor evaluado, es decir, la expresión es una representación de ese valor.

Fuente: <https://es.wikipedia.org/wiki/Expresi%C3%B3n_(inform%C3%A1tica)>

**Operadores y operandos**

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.

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.

Fuente: <https://desarrolloweb.com/articulos/operadores-operandos-programacion>

**Identificadores (variables y constantes)**

Un identificador es un conjunto de caracteres alfanuméricos de cualquier longitud que sirve para identificar las entidades del programa (clases, funciones, variables, tipos compuestos) Los identificadores pueden ser combinaciones de letras y números. Cada lenguaje tiene sus propias reglas que definen como pueden estar construidos. Cuando un identificador se asocia a una entidad concreta, entonces es el «nombre» de dicha entidad, y en adelante la representa en el programa. Nombrar las entidades hace posible referirse a las mismas, lo cual es esencial para cualquier tipo de procesamiento simbólico.

Fuente: <https://andresmtzg.wordpress.com/2012/09/27/identificadores-variables-y-constantes/>

**Dispositivos de entrada**

Los dispositivos periféricos de entrada son todos aquellos dispositivos que permiten introducir datos o información en una computadora para que ésta los procese u ordene. A pesar de que el término “periférico” implica a menudo el concepto de “adicional pero no esencial”, muchos periféricos son elementos fundamentales para un sistema informático. Sin embargo, al ser las fuentes primordiales de entrada, se pueden considerar como extensiones en un sistema.

Un dispositivo de entrada es cualquier periférico del equipamiento de la computadora, utilizado para proporcionar datos y señales de control a un sistema de procesamiento de la información. Los periféricos de entrada y salida componen la interfaz del hardware

Fuente: <https://es.wikipedia.org/wiki/Perif%C3%A9rico_de_entrada>

**Dispositivos de salida**

son dispositivos que muestran o proyectan información hacia el exterior del ordenador. La mayoría son para informar, alertar, comunicar, proyectar o dar al usuario cierta información, de la misma forma se encargan de convertir los impulsos eléctricos en información legible para el usuario. Sin embargo, no todos de este tipo de periféricos es información para el usuario. Un ejemplo: Impresora.

Fuente: <https://es.wikipedia.org/wiki/Perif%C3%A9rico_(inform%C3%A1tica)>

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

CPU es la abreviatura de Central Processing Unit (Unidad Central de Procesamiento en español) que se encuentra en un ordenador. Pero también se puede hablar simplemente de procesador. El procesador es el componente de hardware central y, por tanto, la base del ordenador. Sin ella, un ordenador no puede funcionar en absoluto. Esto se debe principalmente a que la CPU es responsable de todos los cálculos necesarios para que funcione.

Los cálculos del ordenador se llevan a cabo mediante los comandos de máquina, que se pueden considerar como instrucciones para el procesador. Cabe señalar que estos comandos pueden representarse con el código binario, es decir, como secuencias de ceros y unos. Esto es justamente lo que ocurre en el ordenador, ya que la CPU solo puede procesar instrucciones en código binario.

Fuente: <https://www.ionos.es/digitalguide/servidores/know-how/cpu/>

**Unidad de control**

La unidad de control es uno de los tres bloques funcionales principales en lo que se divide la unidad central de procesamiento(CPU), Su función es buscar las instrucciones en la 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 micro programadas, propias de máquinas más complejas. En el primer caso, los componentes principales son el circuito de lógica secuencial, el de control de estado, el de lógica 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 micro memoria, a la cual se accede de manera secuencial para posteriormente ir ejecutando cada una de las microinstrucciones.

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

**Unidad aritmética – lógica**

la unidad aritmética lógica o unidad aritmético-lógica, también conocida como ALU (siglas en inglés de arithmetic logic unit), es un circuito digital que realiza operaciones aritméticas (suma, resta) y operaciones lógicas (SI, Y, O, NO) entre los valores de los argumentos (uno o dos), Por mucho, los circuitos electrónicos más complejos son los que están construidos dentro de los chips de microprocesadores modernos. Por lo tanto, estos procesadores tienen dentro de ellos un ALU muy complejo y potente. De hecho, un microprocesador moderno (y los mainframes) puede tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución, cada una de ellas con múltiples ALU.

Fuente: <https://es.wikipedia.org/wiki/Unidad_aritm%C3%A9tica_l%C3%B3gica>

**Memoria central (interna):**

Es la memoria de la computadora donde se almacenan temporalmente tanto los datos como los programas que la unidad central de procesamiento (CPU) está procesando o va a procesar en un determinado momento. Por su función, la MP debe ser inseparable del microprocesador o CPU, con quien se comunica a través del bus de datos y el bus de direcciones. El ancho del bus determina la capacidad que posea el microprocesador para el direccionamiento de direcciones en memoria.

En algunas ocasiones suele llamarse “memoria interna” porque a diferencia de los dispositivos de memoria secundaria, la MP no puede extraerse tan fácilmente.

Fuente: <https://es.wikipedia.org/wiki/Memoria_principal>

**Memoria RAM**

La memoria de acceso aleatorio (Random Access Memory, RAM) es una memoria de almacenamiento a corto plazo. El sistema operativo de ordenadores u otros dispositivos utiliza la memoria RAM para almacenar de forma temporal todos los programas y sus procesos de ejecución. En la RAM se cargan todas las instrucciones que ejecuta la unidad central de procesamiento (CPU) y otras unidades del ordenador, además de contener los datos que manipulan los distintos programas.

Se denominan «de acceso aleatorio» porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder (acceso secuencial) a la información de la manera más rápida posible.

Fuente: <https://es.wikipedia.org/wiki/Memoria_de_acceso_aleatorio>

**Memoria ROM**

La memoria de solo lectura, conocida también como ROM (acrónimo en inglés de Read Only Memory), es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite solo la lectura de la información y no su escritura,​ independientemente de la presencia o no de una fuente de energía.

Los datos almacenados en la ROM no se pueden modificar, o al menos no de manera rápida o fácil. Se utiliza principalmente para contener el firmware​ (programa que está estrechamente ligado a hardware específico, y es poco probable que requiera actualizaciones frecuentes) u otro contenido vital para el funcionamiento del dispositivo, como los programas que ponen en marcha el ordenador y realizan los diagnósticos.

Fuente: <https://es.wikipedia.org/wiki/Memoria_de_solo_lectura>

**Memoria auxiliar (externa)**

La memoria secundaria, memoria auxiliar, memoria periférica o memoria externa, también conocida como almacenamiento secundario, es el conjunto de dispositivos y soportes de almacenamiento de datos que conforman el subsistema de memoria de la computadora, junto con la memoria primaria o principal.

Puede denominarse periférico de almacenamiento o “memoria práctica”, en contraposición a la ‘memoria central’, porque en ocasiones puede considerarse como periférico de Entrada/Salida.

La memoria secundaria es un tipo de almacenamiento masivo y permanente (no volátil) con mayor capacidad para almacenar datos e información que la memoria primaria que es volátil, aunque la memoria secundaria es de menor velocidad.

Fuente: <https://es.wikipedia.org/wiki/Memoria_secundaria>