**WIKI**

- **INDICADOR:** Es un dato o conjunto de datos que ayudan a medir objetivamente la evolución de un proceso o una actividad.

- **INSTRUCCION:** Conjunto de datos (números y letras) que se insertan en una secuencia y que indican a un procesador la operación que debe ejecutar.

**- LENGUAJE DE PROGRAMACIÓN:**Conjunto de instrucciones que se pueden utilizar para construir un programa, de manera tal que, permita la comunicación usuario/máquina.

**- ALGORITMO:** Método para resolver un problema mediante una serie de pasos precisos, definidos y finitos. Un algoritmo es independiente tanto del lenguaje de programación en que se exprese como del ordenador en el que se ejecute. Preciso porque indica el orden de realización en cada paso. Definido porque si se ejecuta muchas veces, se obtiene el mismo resultado cada vez. Finito porque tiene un número determinado de pasos haciendo que en un momento termine.

**- LENGUAJE DE MÁQUINA**: Lenguaje que está escrito directamente inteligible por la máquina, ya que sus instrucciones son cadenas binarias (0 y 1). Las instrucciones en este lenguaje dependen del hardware, por lo tanto, difieren de una computadora a otra.

**- LENGUAJE DE BAJO NIVEL:** El lenguaje de bajo nivel por excelencia es el ensamblador (assembly language). Sus instrucciones son nemotécnicas. Este lenguaje no puede ser ejecutado directamente por la computadora, sino que requiere una fase de traducción al lenguaje de máquina.

**- LENGUAJE DE ALTO NIVEL:** Son lenguajes desarrollados para reflejar los procedimientos utilizados en la solución de un problema. Son los más utilizados por los programadores.  Están diseñados para que las personas escriban y entiendan los programas de un modo más fácil. Estos programas son independientes de la máquina, es decir, las instrucciones del programa de la computadora no dependen del diseño del hardware.

**- ALGORITMO CUALITATIVO:**Aquel en el cual no intervienen cálculos numéricos, y siempre son expresados en forma de palabras. Son algoritmos usados para describir procesos de la vida cotidiana. Este tipo de algoritmo no es estrictamente computable porque prima en gran medida la parte subjetiva sobre la objetiva.

**- ALGORITMO CUANTITATIVO:** Aquel en el que sí intervienen cálculos numéricos. Son algoritmos muy objetivos, pues al tener siempre correspondencia con problemas matemáticos, demandan la inclusión de cálculos numéricos para la obtención de sus resultados.

**- LENGUAJES ALGORÍTMICOS GRÁFICOS:** Son las representaciones gráficas de las operaciones que realiza un algoritmo (diagrama de flujo).

**- LENGUAJE NO GRÁFICO:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

**- PSEUDOCÓDIGO:**Herramienta de programación en la que las instrucciones se escriben con palabras similares al inglés o al español. Es la representación de los pasos del algoritmo a través de palabras, utilizando una nomenclatura estandarizada para denotar el significado de cada paso.

**- DEFINICIÓN DEL PROBLEMA:** Se refiere a la comprensión del problema como tal. Para ello habrá que definir lo qué se requiere resolver, qué resultados se deben obtener, qué se tiene, lo que se desconoce, las condiciones que nos dan, las cantidades dadas.

**- ANÁLISIS DEL PROBLEMA:**El objetivo principal del análisis es profundizar minuciosamente en la comprensión de la mecánica del problema a solucionar, describiendo plenamente las especificaciones de entrada y salida, así como los cómputos a realizar. Es aquí donde se hace el planteamiento matemático y lógico de la solución del problema.

**- DISEÑO DEL ALGORITMO:**Descripción de los pasos lógico que dan solución al problema hata obtener los resultados requeridos.

Las características de un buen algoritmo son:

1. Tener un punto particular de inicio.
2. Ser definido, no debe permitir dobles interpretaciones.
3. Ser general, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
4. Ser finito en tamaño y tiempo de ejecución.

**- CODIFICACIÓN:** Consiste en pasar los gráficos o palabras con que se diseña un algoritmo a un lenguaje de programación reconocido por la computadora.

**- COMPILACIÓN:** El compilador analiza, una a una, las instrucciones del programa codificado para detectar si están bien escritas o no. Si existen errores se producirá un listado de éstos indicando su localización y causa del error, para corregirlos e iniciar de nuevo la fase de compilación. Cuando no existan errores el programa se traduce al lenguaje de máquina.

**- EJECUCIÓN:** Convierten las instrucciones escritas en lenguaje de programación de alto nivel en instrucciones escritas en lenguaje de máquina, para que esta lo pueda entender, y así se ejecute una a una las instrucciones dadas.

**- PRUEBA DE ESCRITORIO:** Aquella que permite detectar los posibles errores que comete el programador en el diseño del algoritmo.  Esta prueba consiste en la selección de diferentes datos de entrada al algoritmo y en el seguimiento de la secuencia de cada una de las etapas, hasta obtener los resultados.

**- PRUEBA O DEPURACIÓN:** Proceso de identificación y eliminación de errores.

**- TIPOS DE DATOS:** Hace referencia a las diferentes clases de datos que puede procesar una computadora.  Estos pueden ser simples o estructurados. Los simples ocupan una sola casilla de memoria, haciendo referencia a un único valor a la vez.  Dentro de este grupo tenemos: enteros, reales, caracteres, booleanos, double y reales. Los estructurados hacen referencia a un grupo de casillas en memoria.  Dentro de este grupo tenemos: arreglos, cadenas de caracteres, registros, conjuntos.

**- EXPRESIONES:**Es una combinación de constantes, variables, operadores de cálculo (aritméticos, de comparación, booleanos), paréntesis y nombres de funciones especiales; que arroja un resultado determinado por la naturaleza de todos los elementos contenidos en ella. Las expresiones se clasifican en aritméticas y booleanas.

Las expresiones aritméticas son análogas a las expresiones matemáticas, que incluyen: números, funciones, operadores matemáticos, de comparación, etc. Su resultado siempre es un número.

Las expresiones booleanas son sentencias o enunciados veritativos (valor de verdad de un enunciado u oración) que contiene operadores relacionales de comparación y/o booleanos.  Su resultado siempre es un valor verdadero o falso.

**- OPERADORES:** Elementos que relacionan de forma diferente los valores de una o más variables y/o constantes. Nombre que se le da a una variable, constante o cualquier elemento que necesite nombrarse. Debe tenerse en cuenta que para nombrar un identificador, el primer carácter debe ser una letra, los demás pueden ser letras, dígitos o símbolos.  Un identificador nunca podrá coincidir con ninguna de las palabras reservadas del propio lenguaje de programación.

Tipos de operadores:

* Lógicos
* Relacionales
* Aritméticos
* **Operandos**
* **Identificadores**

**- OPERANDO:**Elemento sobre el cual se aplica una operación. El concepto de operando y el de resultado están relacionados con el operador. El resultado es la respuesta que resulta cuando se completa la operación. Por ejemplo. la expresión 3+2, el + es el operador, el 3 y el 2 son los operandos, y el 12 es el resultado. Los operandos pueden ser constantes o variables. El tipo de dato de los operandos y el resultado dependen del operador.

**- IDENTIFICADOR:** Son nombres que se le dan a las variables, constantes, acumuladores y contadores para poder así diferenciarlos.

**- VARIABLE:** Estructura de datos que puede cambiar o no su valor o contenido durante la ejecución de un programa.

**- CONSTANTE:** Dato que no cambia durante la ejecución de un programa.

**- TÉCNICA DE DISEÑO TOP DOWN:** Proceso mediante el cual un problema se descompone en una serie de niveles o pasos sucesivos.

**- TÉCNICA DE DISEÑO BOTTOM UP:** Hace énfasis en la programación y pruebas tempranas, que pueden comenzar tan pronto se ha especificado el primer módulo. Este enfoque tiene el riesgo de programar cosas sin saber cómo se van a conectar al resto del sistema, y esta conexión puede no ser tan fácil como se creyó al comienzo. Uno de los mayores beneficios de este enfoque es la reutilización del código.

**- DISPOSITIVOS DE ENTRADA:** Aquel por el que el ordenador recibe datos desde el mundo exterior.  Estos dispositivos están diseñados para convertir señales de información de una fuente externa en un patrón de bits que el computador pueda procesar. Los dispositivos de entrada más comunes son el teclado y el ratón.

**- DISPOSITIVOS DE SALIDA:**Aquel por el que el procesador puede dar datos o señales al mundo exterior. Estos dispositivos aceptan cadenas de bits del computador y las transforman en un formato útil o significativo fuera de éste. Algunos dispositivos de salida son los monitores de video, impresoras, sintetizadores de voz, brazos robóticos.

**- UNIDAD CENTRAL DE PROCESAMIENTO:** Coordina y supervisa la operación de diversas funciones. Le indica a la unidad de entrada cuándo debe grabarse la información dentro de la memoria, a la unidad aritmética y lógica le indica cuándo debe utilizarse la información dela memoria para los cálculos, a la unidad de salida le indica cuándo debe enviar la información desde memoria hasta ciertos dispositivos de salida.

**- UNIDAD DE CONTROL:** Supervisa la transferencia de información entre los registros e instruye a la unidad aritmético lógica sobre cuál operación ejecutar.

**- UNIDAD ARITMÉTICO LÓGICA:** Es la responsable de realizar cálculos como suma, resta, multiplicación y división.  Contiene los mecanismos de decisión que permiten a la computadora hacer cosas como, por ejemplo, comparar dos elementos de la unidad de memoria para determinar si son iguales o no.