2. a DEFINICIÓN:

Un algoritmo es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

b. CARACTERÍSTICAS:

Las características que debe poseer una secuencia de tareas para considerarse algoritmo son:

PRECISIÓN: El algoritmo debe indicar el orden exacto de ejecución de cada tarea.

Definitud o determinismo: Si se sigue el algoritmo dos o más veces con los mismos datos de entrada, se deben obtener los mismos datos de salida.

Finitud: El algoritmo debe terminar en algún momento y debe usar una cantidad de recursos finita.

C. FORMAS DE REPRESENTAR UN ALGORITMO:

SEUDOCÓDIGO: es una forma de diagramar o representar un algoritmo para resolver un determinado problema, independiente de cualquier lenguaje de programación en especial.

DIAGRAMA DE FLUJO: es la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Es la representación de la solución utilizando figuras geométricas como circunferencias, flechas, rectángulos, rombos, etc.

DIAGRAMA N - S (Nassi-Shneiderman): también conocido como diagrama de Chapin es una técnica de especificación de algoritmos que combina la descripción textual, propia del seudocódigo, con la representación gráfica del diagrama de flujo.

El diagrama N-S cuenta con un conjunto limitado de símbolos para representar los pasos del algoritmo, por ello se apoya en expresiones del lenguaje natural; sin embargo, dado que el lenguaje natural es muy extenso y se presta para la ambigüedad, solo se utiliza un conjunto de palabras.

D.1 Simbolización y Nombre:

* **Óvalo** o **Elipse**: Inicio y término (Abre y cierra el diagrama).
* **Rectángulo**: Actividad (Representa la ejecución de una o más actividades o procedimientos).
* **Rombo**: Decisión (Formula una pregunta o cuestión).
* **Círculo**: Conector (Representa el enlace de actividades con otra dentro de un procedimiento).
* **Triángulo boca abajo**: Archivo definitivo (Guarda un documento en forma permanente).
* **Triángulo boca arriba**: Archivo temporal (Proporciona un tiempo para el almacenamiento del documento).

TIPOS DE ALGORITMOS:

**Cualitativos**: Son aquellos en los que se describen los pasos utilizando palabras.

* Son todos aquellos pasos o instrucciones descritos por medio de palabras que sirven para llegar a la obtención de una respuesta o solución de un problema cualquier.

Ejemplo**:**Construir un algoritmo cualitativo para realizar una llamada a un número nuevo vía celular

Proceso**:**

* 1. Marcar el número
* 2. Presionar el Botón llamar
* 3. Hablar
* 4. Terminar la llamada
* 5. Guardar número en la lista de contactos

**Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. Son aquellos pasos o instrucciones que involucran cálculos numéricos para llegar a un resultado satisfactorio.

Ejemplo 2:

Se quiere calcular el área de un rectángulo, se tiene en cuenta la forma que es (A x B) en la cual a y b son dos de los lados.

Se solicita la longitud del lado A. A=8

Se solicita la longitud del lado B. B=13

Luego se multiplica el lado A con el lado B y el resultado es el área del rectángulo.
C= AXB

C=104

F. QUÉ SON VARIABLES

Una variable es un símbolo constituyente de un predicado, fórmula, algoritmo o de una proposición. El término «variable» se utiliza aun fuera del ámbito matemático para designar una cantidad susceptible de tomar distintos valores numéricos dentro de un conjunto de números especificado.

F. 1. Cómo se le da un valor:

En programación, Para crear una variable escribiremos el nombre del tipo seguido del nombre que queremos que tenga la variable y tras un símbolo **=** el valor que queramos que tenga dicha variable. Las variables pueden cambiar de valor durante la ejecución del programa. Para modificar el valor de una variable usaremos la instrucción de asignación que consiste en indicar el nombre de la variable, un símbolo **=**, el valor que queremos que tome la variable y, como siempre, acabar la instrucción con un punto y coma.

G. QUE ES UNA CONSTANTE:

En general, una constante es un valor de tipo permanente, que no puede modificarse, al menos no dentro del contexto o situación para el cual está: geometría aritmética.

H. EJEMPLO:

**C**

En C las constantes se declaran con la directiva **#define**, esto significa que esa constante tendrá el mismo valor a lo largo de todo el programa.

El identificador de una constante así definida será una cadena de caracteres que deberá cumplir los mismos requisitos que el de una variable (sin espacios en blanco, no empezar por un dígito numérico, etc).

Ejemplo:

 #include <stdio.h>

 #define PI 3.1415926

 int main()

 {

 printf("Pi vale %f", PI);

 return 0;

 }

Lo cual mostrará por pantalla:

 Pi vale 3.1415926

Es decir, **PI** es una constante a la que le hemos asignado el valor **3.1415926** mediante la directiva **#define**.

I. PSEUDOCÓDIGO:

En ciencias de la computación, y análisis numérico, el pseudocódigo (o falso lenguaje) es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

I.1: Características:

Las principales características de este lenguaje son:

1. Se puede ejecutar en un ordenador (con un IDE como por ejemplo SLE, LPP, PilatoX, Maruga Script, Seudocódigo o PSeInt. Otros Ides de consideración son Inter-P y Algor)
2. Es una forma de representación sencilla de utilizar y de manipular.
3. Facilita el paso del programa al lenguaje de programación.
4. Es independiente del lenguaje de programación que se vaya a utilizar.
5. Es un método que facilita la programación y solución al algoritmo del programa.

Todo documento en pseudocódigo debe permitir la descripción de:

1. Instrucciones primitivas.
2. Instrucciones de proceso.
3. Instrucciones de control.
4. Instrucciones compuestas.
5. Instrucciones de descripción.

I.2: Qué operadores se pueden utilizar en un pseudocódigo

**Aritméticos:** son análogas a las fórmulas matemáticas, se usan para variables numéricas.

**Lógicas:** solo pueden dar dos valores **true**o **false.** Pueden ser:

* **Relacionales:** pueden comparar dos valores, mediante un símbolo
* **Lógicos**: combina condiciones, según el tipo de operador se deberá cumplir una u otra función.