Herramientas para la elaboración de algoritmos:

- El Diagrama de Flujo, es la representación gráfica de las operaciones que realiza un algoritmo.
- El Pseudocódigo, representa de forma descriptiva las operaciones que debe realizar un algoritmo en un lenguaje humano simplificado.

Para su elaboración se debe determinar los **datos de entrada**, los **datos de salida**, los **cálculos** o **procesos** que se deben realizar, las **condiciones** y **restricciones** del algoritmo.

Características:

Un algoritmo debe cumplir con las siguientes condiciones fundamentales:

- Ser preciso e indicar el orden de realización de cada paso.
- Debe ser definido, no debe permitir dobles interpretaciones, siguiendo un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Debe ser finito, contener un número finito de pasos en tamaño y tiempo de ejecución.
- Debe ser legible, el texto que lo describe debe ser claro, de manera que permita leerlo y entenderlo fácilmente.

En el algoritmo escrito en **pseudocódigo**, los **datos**, los **procesos** y las **operaciones** de **entrada** y **salida** se indican mediante **palabras claves**. Estas constituyen una parte requerida de la sintaxis de una instrucción. Alguna de ellas son: **Nombre del algoritmo**, **variable**, **inicio**, **fin**, **escribir**, **leer**.

Esta forma de escribir algoritmos guarda semejanza con las sentencias disponibles en cualquier lenguaje de programación, pero con reglas más flexibles, permitiendo así concentrarse en la estructura lógica del algoritmo, sin preocuparse por las limitaciones que impondría trabajar con un lenguaje de programación en particular.

Metodología para describir algoritmos

- **Entrada** ➡ Información dada al algoritmo.
- **Proceso** ➡ Operaciones o cálculos necesarios para encontrar la solución del problema.
- **Salida** ➡ Respuestas dadas por el algoritmo o resultados finales de los procesos realizados.

Especificaciones de entrada

- ¿Qué datos son de entrada?
- ¿Cuántos datos se introducirán?
- ¿Cuántos son datos de entrada válidos?

Especificaciones de salida

- ¿Cuáles son los datos de salida?
- ¿Cuántos datos de salida se producirán?
- ¿Qué formato y precisión tendrán los resultados?

Ejemplo:

El restaurante de comidas rápidas "***cofla***" desea que las hamburguesas base siempre salgan igual, según el orden de los ingredientes. Medio pan, 70Gr de Carne, 1 tajada de queso, 2 tajadas de tomate, una hoja de lechuga y por último medio pan.

Entrada ➡***Son 4 datos, ya que el pan se repite.***

Los datos son los ingredientes base para la hamburguesa, pan por mitad, queso tajado, tomate tajado, hojas de lechuga, carne en porciones de 70 Gr.

Salida ➡ es una Hamburguesa.

Proceso ➡ "armado de la hamburguesa"

1. **Inicio**
2. Se preparan todos los ingredientes (DECLARACIÓN DE VARIABLES y CONSTANTES)
3. Se coloca una Mitad de pan
4. Ahora sobre el pan se colocan 70 gr de Carne.
5. Se coloca 1 Tajada de queso sobre la Carne.
6. Se colocan 2 Tajadas de Tomate
7. Se coloca 1 Hoja de lechuga
8. Y se tapa con la otra mitad del pan.
9. Se entrega una hamburguesa armada
10. **FIN**

Estructuras de control

a) **Selección simple, doble y anidada**

El algoritmo al ser ejecutado toma la decisión de ejecutar o no ciertas instrucciones si se cumplen o no ciertas condiciones. Las condiciones devuelven un valor, verdadero o falso, determinando así la secuencia a seguir.

Simple es cuando: Si se cumple (condición) se ejecuta la instrucción.

Doble: Si se cumple la condición se ejecuta la sentencia 1, sino se ejecuta la sentencia 2. Su estructura es:

```
Si condición entonces
      sentencia 1
Sino
      sentencia 2
fin_si
```

Anidamiento de estructura SI

Existen casos en los que la condición es insuficiente porque el conjunto de valores probables sobrepasa solo dos, por ejemplo, si se quiere conocer si un número es positivo, negativo o cero, en este caso las posibles condiciones son que el número sea mayor que cero, sea menor que cero o sea igual a cero, en este caso, son tres las condiciones, entonces es necesario que como una de las acciones de la estructura, se utilice otra estructura si (operación que se conoce como anidamiento de estructuras si).

```
Algoritmo tipo_numero
Var a:entero
Inicio
      Leer (a)
      Si a > 0 entonces
            Escribir ("El número es positivo")
      Sino
```

```
            Si a < 0 entonces
                  Escribir ("El número es negativo")
            Sino
                  Escribir ("El número es cero")
            Fin_si
      Fin_si
Fin
```