

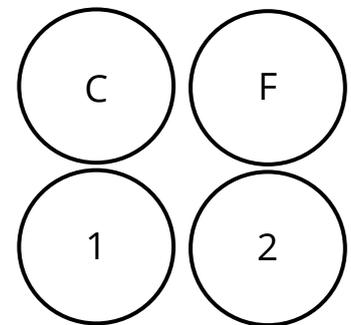
Organigramas

Para empezar a programar, lo primero que hay que hacer es entender los **algoritmos**, y la mejor manera de entender estas instrucciones, es representarlas en forma de **organigramas**.

Un organigrama es un **diagrama de flujo**, que representa la secuencia de pasos que hay que seguir para programar. Coloca las instrucciones dadas por los algoritmos de una forma ordenada y entendible.

Un diagrama de flujo se representa con diversos símbolos:

El **círculo** se utiliza para **iniciar** (C) o **terminar** (F) el organigrama. También puede servir de **conector**, ya que el organigrama solo puede ir en vertical, así que si se te acaba el papel, puedes seguir al lado.



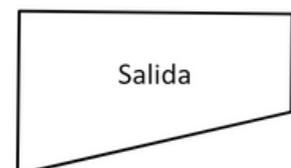
El **rectángulo** se utiliza para **procesos**, como operaciones o asignación de valores a las variables. También se utiliza para llamar funciones.



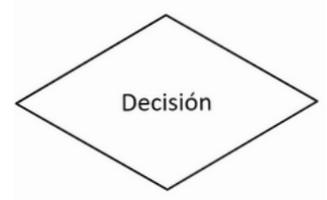
El **paralelogramo** con la punta hacia arriba se utiliza para la **entrada** de datos, por si el algoritmo necesita recibir información.



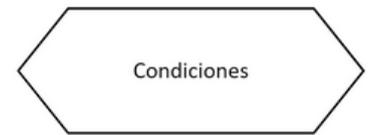
El **paralelogramo** con la punta hacia abajo se utiliza para la **salida** de datos, para mostrar datos, resultados o mensajes en pantalla.



El **rombo** se utiliza para **decisiones**. Se introduce una condición y dependiendo del resultado de dicha condición, el flujo seguirá un camino u otro. Se suele utilizar en los bucles.



El **hexágono alargado** se utiliza para poner **condiciones** a las operaciones, como por ejemplo, poner los intervalos de las variables.



Estos símbolos se unen mediante **flechas**, que señalan el flujo del diagrama, es decir, la **dirección** que siguen las instrucciones.

Aquí podemos ver un ejemplo de lo que sería hacer un algoritmo para el área de un trapecio, teniendo en cuenta que la base mayor debe ser mayor que la base menor, y que no sean iguales, para asegurar que sea un trapecio de verdad. Los pasos son:

1. Inicio.
2. Entrada de variables.
3. Inicio de bucle $B < b$, que pide la nueva entrada de variables si no se cumplen las condiciones.
4. Fin de bucle cuando se cumplen las condiciones.
5. Inicio de bucle $B == b$, que pide la nueva entrada de variables si no se cumplen las condiciones.
6. Fin de bucle cuando se cumplen las condiciones.
7. Cálculo del área del trapecio.
8. Salida de "área".

