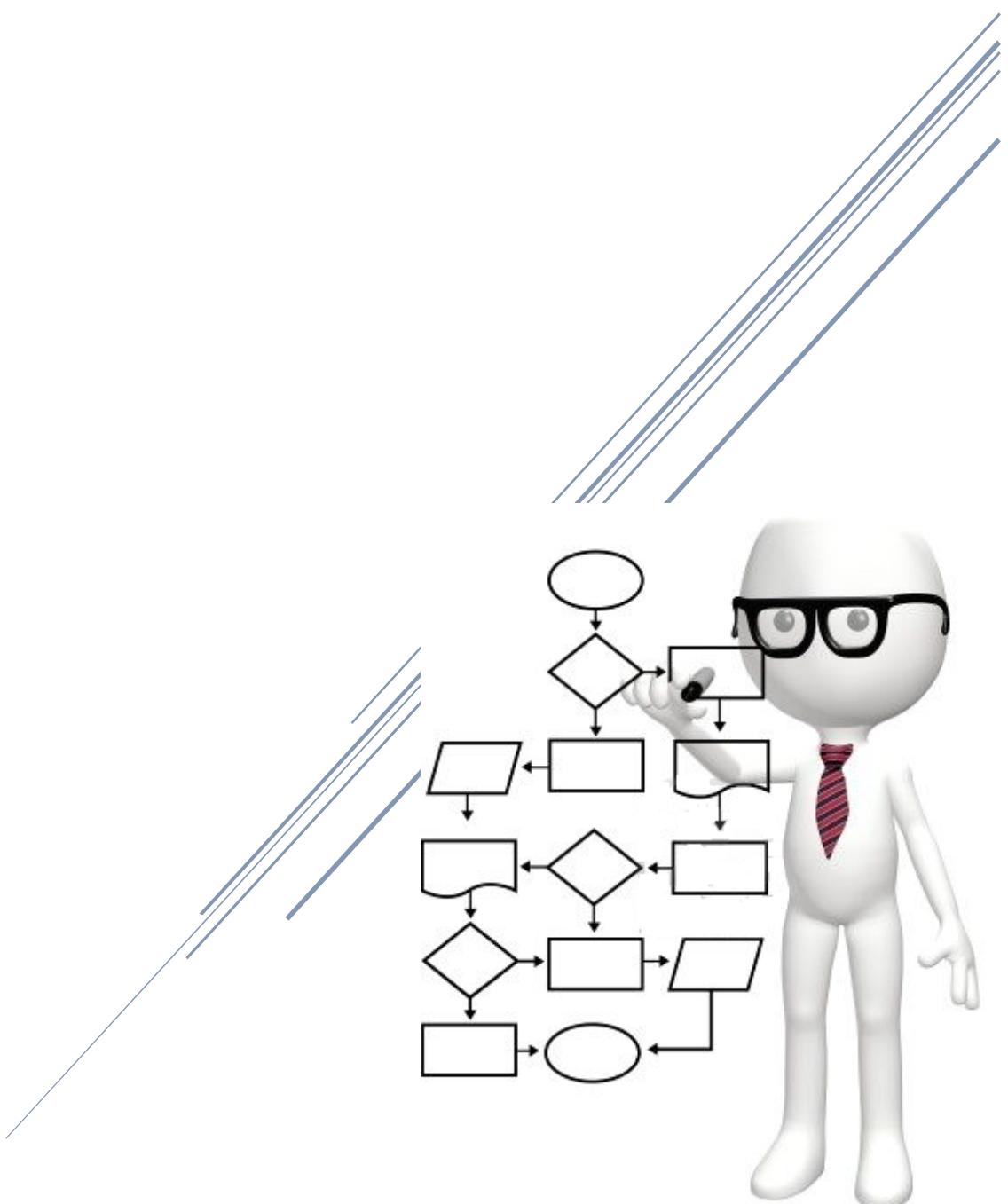


DIAGRAMAS DE FLUJO

PROGRAMACIÓN



DIAGRAMAS DE FLUJO

Podemos decir que un diagrama de flujo, a veces denominados **organigramas**, es una representación gráfica de un algoritmo, de forma que se describen las diferentes tareas y la secuencia en la que se ejecutan para alcanzar una solución.

En algoritmos complejos podemos usar más de un diagrama de flujo. Diseñaremos el diagrama de flujo principal y a partir de él iremos diseñando diagramas de cada tarea que se puede realizar a partir de la inicial.

Ya hemos visto representaciones de este tipo de elemento, a continuación, vamos a concretar qué tipo de figuras se usan y qué tipo de información proporcionan.

REGLAS PARA DISEÑAR DIAGRAMAS DE FLUJO

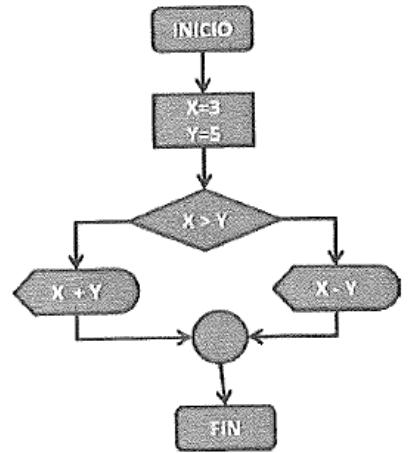
A la hora de crear nuestro propio diagrama de flujo debemos tener en cuenta que:

- Siempre debemos incluir una forma con el texto **Inicio** y otra con el texto **Fin** para delimitar los momentos en que empieza y acaba el algoritmo.
- Debemos usar una serie de símbolos estándar.
- Los elementos del diagrama de flujo deben estar concatenados mediante flechas que indican la dirección de ejecución derecha.
- Debemos evitar el cruce de líneas, para eso se define la forma conectora. El uso de conectores debe producirse cuando no exista otra opción.
- Todas las líneas de flujo deben estar conectadas a algún objeto.
- A la hora de escribir texto en las formas, este debe ser escueto y legible.
- Todos los símbolos de decisión deben tener más de una línea de salida, es decir, deben indicar qué camino seguir en función de la decisión tomada.

Existen otras formas que pueden usarse en un diagrama de flujo, pero no creemos que sea necesario profundizar en ellas.

Si el alumno lo desea puede localizar más información sobre diagramas de flujo u organigramas en la web o bibliografía aportada por el profesor.

	Indican inicio o fin de programa.
	Representan una instrucción, un paso a dar o proceso. Por ejemplo: nota = 4, suma=suma+1, etc.
	Operaciones de entrada/salida de datos. Por ejemplo: visualiza en pantalla suma.
	Usaremos este símbolo si nos encontramos en un punto en el que se realizará una u otra acción en función de la decisión que el usuario tome.
	Conector. Permite unir diferentes zonas del diagrama de forma que se redefine el flujo de ejecución hacia otra parte del diagrama.
	Representa una función o subprograma.
	Conector de página. Se usa cuando llegamos al final de una página y aún no se ha acabado el diagrama de flujo.
	Disco magnético.
	Unidad de cinta magnética.
	Salida. Visualización de los datos de salida por pantalla. Este suele ser siempre el periférico de salida por defecto.
	Salida de datos por impresora.
	Flechas para indicar la dirección de flujo.



ACTIVIDAD 1

Dado el siguiente organigrama, indica qué tarea realiza.

ACTIVIDAD 2

Realiza el diagrama de flujo tal que se dé un dato de entrada X. Si la X es igual a 1 mostraremos el texto “Has pulsado 1”. En caso de que sea 2 el texto a visualizar será “Has pulsado 2”.

PSEUDOCÓDSO

La palabra pseudocódigo puede ser traducida como falso lenguaje, y no hay nada más acertado que su definición. Un pseudocódigo es una descripción informal de un programa, siendo de gran utilidad a la hora de diseñar el algoritmo del mismo. En pseudocódigo usamos lenguaje natural o similar para representar estructuras propias del lenguaje de programación que usaremos en la codificación, es decir, simulamos un lenguaje de programación.

El pseudocódigo, aunque usa el lenguaje natural, no utiliza su amplia variedad de palabras, se seleccionan una serie de palabras como reservadas para representar todas las estructuras que posteriormente deberán ser codificadas mediante el lenguaje de programación.

Se han visto ejemplos de pseudocódigo en los apartados 1.3.2 y 1.3.6. Estudiaremos en más profundidad las palabras reservadas en pseudocódigo cuando empecemos a conocer las diferentes estructuras de control y su significado en el Capítulo 2.