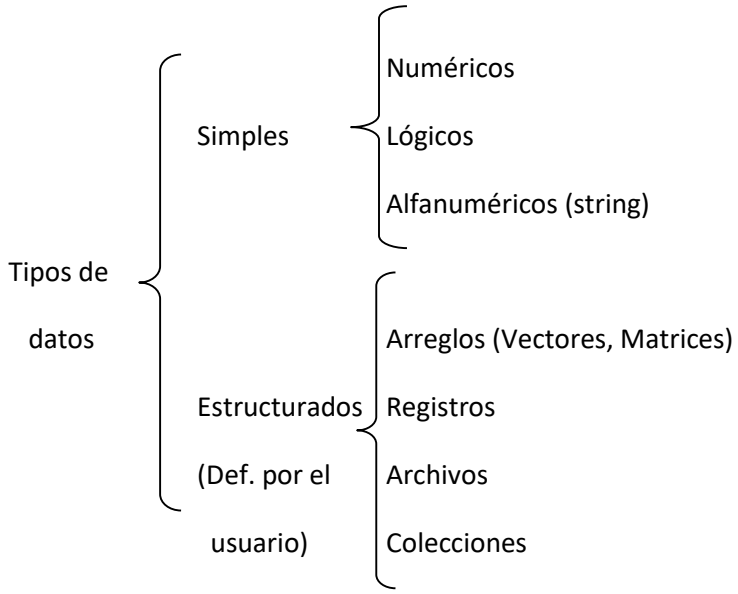
Uso de variables

Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como 'b', un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.



Expresiones y operadores

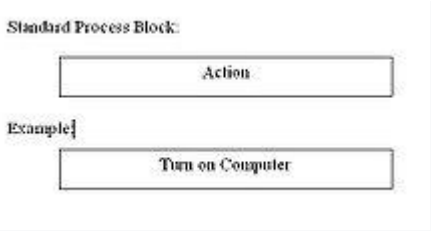
- Aritméticos: + - * / % y funciones definidas en el lenguaje.
- Relacionales: < > = !=
- Lógicos: and (&&) or (||) not (!)

Diagramación estrurada

Diagrama Nassi-Shneiderman (o **NSD** por sus siglas en inglés), también conocido como *diagrama de Chapin*. Fue desarrollado en 1972 por Isaac Nassi y Ben Shneiderman.

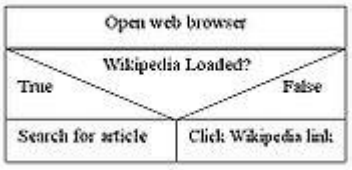
Es una representación gráfica que muestra el diseño de un programa estructurado.

Bloques de procesos El bloque de proceso representa el paso mas simple y no requiere ningún análisis específico.Cuando un bloque de proceso es encontrado, la acción dentro del bloque se realiza, y pasamos directamente al siguiente bloque



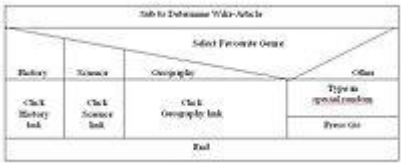
Process blocks

Bloques ramificados hay dos tipos de estos bloques. El primero y más sencillo de ellos es el bloque **verdadero-falso** el cual ofrece al programa dos caminos para tomar, dependiendo de si una determinada condición ha sido especificada. Estos bloques pueden ser usados como bucles que detienen el programahasta que una determinada condición se cumpla.



True/False branching blocks

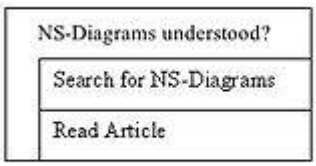
El segundo tipo es un bloque ramificado múltiple. Este tipo de bloque es utilizado cuando se necesita la selección de un caso en un programa . El bloque suele contener una pregunta. Además, el bloque le da al programa una cadena de oportunidades y es generalmente usado en las conjunciones con bloques de subprocesos para ahorrar espacio.



Multiple branching blocks

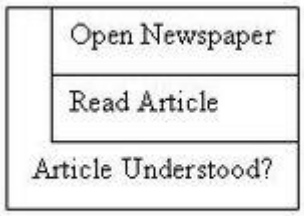
Bucles testeadores: este bloque permite al programa repetir un bloque o un conjunto de bloques hasta que una determinada condición se haya cumplido.

Hay dos tipos de estos bloques: de testeo inicial y testeo final. La única diferencia entre los dos es el orden en el cual se completan los pasos involucrados en el proceso. En los de la primera situcaión, cuando el programa encuentra el bloque, testea si la condición necesaria se cumple, si no, se repite el bucle. El test se repite hasta que se cumpla dicha condición. En el nivel que se cumpla la condición, el programa detiene la ejecución del blucle y pasa a analizar los bloques del siguiente nivel.



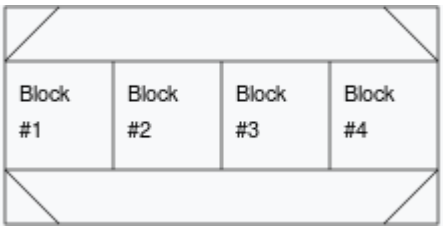
Test first loop block

Los de testeo final operan al revés.



Test last loop block

Una **Expresión concurrente** `puede ser dibujada así:[5]

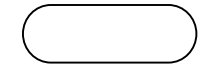
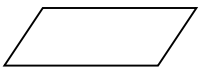


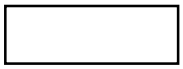
Diagramas de flujo

El **diagrama de flujo** o **flujograma** o **diagrama de actividades** es la representación gráfica de un algoritmo o proceso.

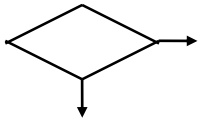
Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.

Los símbolos utilizados han sido normalizados por el instituto norteamericano de normalización (ANSI).

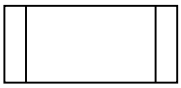
SÍMBOLO	***DESCRIPCIÓN***
	Indica el inicio y el final de nuestro diagrama de flujo.
	Indica la entrada de datos.



Símbolo de proceso y nos indica la asignación de un valor en la memoria y/o la ejecución de una operación aritmética.

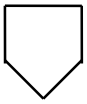


Símbolo de decisión indica la realización de una comparación de valores.



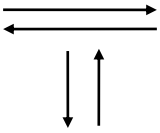
Se utiliza para representar los subprogramas.

Conector dentro de página. Representa la continuidad del diagrama dentro de la misma página.



Conector fuera de página. Representa la continuidad del diagrama en otra página.

Indica la salida de información por impresora.



Líneas de flujo o dirección. Indican la secuencia en que se realizan las operaciones.

Recomendaciones para el diseño de Diagramas de Flujo

- Se deben se usar solamente líneas de flujo horizontales y/o verticales.
- Se debe evitar el cruce de líneas utilizando los conectores.
- Se deben usar conectores sólo cuando sea necesario.
- No deben quedar líneas de flujo sin conectar.

- Se deben trazar los símbolos de manera que se puedan leer de arriba hacia abajo y de izquierda a derecha.
- Todo texto escrito dentro de un símbolo deberá ser escrito claramente, evitando el uso de muchas palabras.

Pseudocodigo

Mezcla de lenguaje de programación y español (o inglés o cualquier otro idioma) que se emplea, dentro de la programación estructurada, para realizar el diseño de un programa. En esencial, el pseudocodigo se puede definir como un lenguaje de especificaciones de algoritmos.

Es la representación narrativa de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. El pseudocodigo utiliza palabras que indican el proceso a realizar.

Ventajas de utilizar un Pseudocodigo a un Diagrama de Flujo

- Ocupa menos espacio en una hoja de papel.
- Permite representar en forma fácil operaciones repetitivas complejas.
- Es muy fácil pasar de pseudocodigo a un programa en algún lenguaje de programación.
- Si se siguen las reglas se puede observar claramente los niveles que tiene cada operación.