**LICEO MIXTO LA MILAGROSA**

**TECNOLOGÍA E INFORMATICA GRADO NOVENO**

**LIC. VIRGILIO MURILLO**

|  |
| --- |
| **2.1 ALGORITMOS** |

Un **ALGORITMO** es una secuencia finita ‘bien definida’ de tareas ‘bien definidas’, cada una de las cuales se puede realizar con una cantidad de recursos finitos.

Se dice que una tarea esta ‘bien definida’, si se sabe de manera precisa las acciones requeridas para su realización. Aunque los recursos que debe utilizar cada tarea deben ser finitos estos no están limitados, es decir, si una tarea bien definida requiere una cantidad inmensa (pero finita) de algún recurso para su realización, dicha tarea puede formar parte de un algoritmo. Además, se dice que una secuencia de tareas esta ‘bien definida’ si se sabe el orden exacto de ejecución de cada una de las mismas.

EJECUTAR un algoritmo es realizar las tareas del mismo, en el orden especificado y utilizando los recursos disponibles.

|  |
| --- |
| **2.1.1   CARACTERISTICAS DE UN ALGORITMO** |

Las características que debe poseer una secuencia de tareas para considerarse algoritmo son: precisión, definitud y finitud.

|  |  |  |
| --- | --- | --- |
| **Precisión** | **Definitud o determinismo** | **Finitud** |
| El algoritmo debe indicar el orden exacto de ejecución de cada tarea. | Si se sigue el algoritmo dos o más veces con los mismos datos de entrada, se deben obtener los mismos datos de salida. | El algoritmo debe terminar en algún momento y debe usar una cantidad de recursos finita. |

Dada una cantidad de datos de entrada de un algoritmo, se dice que la cantidad de un recurso usada por dicho algoritmo para su ejecución determina la**complejidad del algoritmo** respecto a tal recurso. Cuando se implementa un algoritmo en un computador digital, los recursos con los que se cuenta son **tiempo de proceso** y **memoria**. Por lo tanto, a un algoritmo implementado en un computador digital se le pueden calcular sus complejidades **temporal** y **espacial**.[[1]](http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/leccion101.htm" \l "_ftn1" \o ")

[[1]](http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/leccion101.htm" \l "_ftnref1" \o ") El estudio de la complejidad de un algoritmo respecto a sus recursos es conocido como algoritmia, la cual no es tratada en este curso.

|  |
| --- |
| **2.1.2 TECNICAS PARA LA FORMULACION DE ALGORITMOS** |

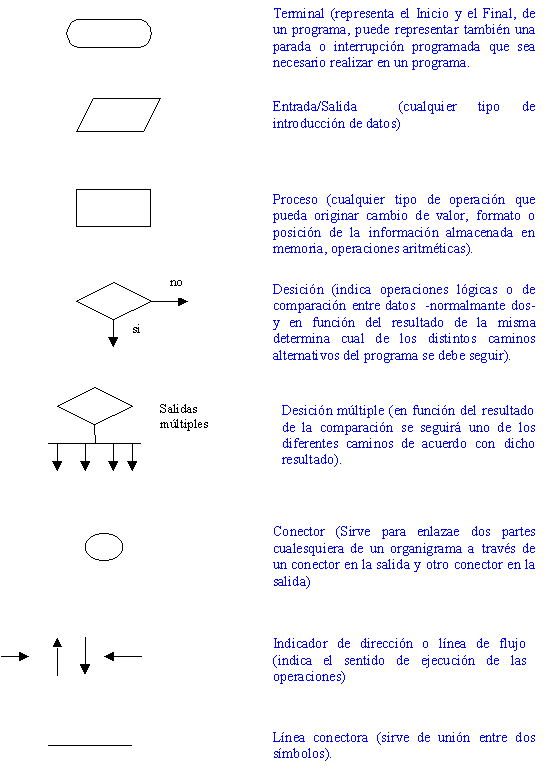
Las técnicas para la formulación de algoritmos más populares son:

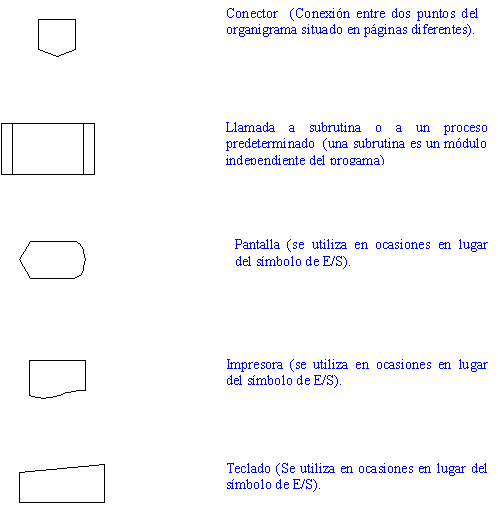
|  |
| --- |
| **2.1.2.1 DIAGRAMAS DE FLUJO** |

Se basan en la utilización de diversos símbolos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados Son.

**SIMBOLOGIA UTILIZADA EN LOS DIAGRAMAS DE FLUJO**

SIMBOLO                           FUNCION





|  |
| --- |
| **2.1.2.2 PSEUDOCODIGO** |

Es un lenguaje de especificación de algoritmos. El uso de tal lenguaje hace el paso de codificación final (esto es, la traducción a un lenguaje de programación) relativamente fácil.

El pseudocódigo nació como un lenguaje similar al lenguaje natural y era un medio para representar básicamente las estructuras de control de programación estructurada. Se considera un primer borrador, dado que el pseudocódigo tiene que traducirse posteriormente a un lenguaje de programación.

La ventaja del pseudocódigo es que en su uso en la planificación de un programa, el programador se puede concentrar en la lógica y en las estructuras de control y no preocuparse de las reglas de un lenguaje específico. Es también fácil modificar el pseudocódigo si se descubren errores o anomalías en la lógica del programa, además de todo esto es fácil su traducción a lenguajes como *Pascal*, *C* o *Basic*.

El pseudocódigo utiliza para representar las acciones sucesivas palabras reservadas (similares a sus homónimos en los lenguajes de programación), tales como inicio, fin, si-entonces-sino, mientras, repita-hasta….etc

Secuencia

|  |
| --- |
| **Inicio**  **acción1**  **acción2**  **.**  **.**  **.**  **acción n**  **Fin** |

Decisión

Simple

|  |
| --- |
| **si condición entonces**  **acción1**  **acción2**  **.**  **.**  **.**  **acción n** |

 Doble

|  |
| --- |
| **si condición entonces**  **acción1**  **acción2**  **.**  **.**  **.**  **en caso contrario**  **acción1**  **acción2** |

Iteracción

Fija

|  |
| --- |
| **para var. Entera inicial hasta final hacer**  **acción1**  **acción2**  **…..**  **acción n** |

Condicional al inicio

|  |
| --- |
| **mientras condición hacer**  **acción1**  **acción2**  **.**  **.**  **acción n** |

Condicional al final

|  |
| --- |
| **repita**  **acción1**  **acción2**  **…..**  **acción n**  **Hasta que condición** |

Selección

|  |
| --- |
| **casos selector de**  **valor1 : acción1**  **acción2**    **valor2 : acción1**  **acción2**  ...  **valor n : acción1**  **acción2** |

|  |
| --- |
| **2.1.2   EJEMPLOS DE ALGORITMOS** |

**1. PROBLEMA:** Un estudiante se encuentra en su casa (durmiendo) y debe ir a la universidad (a tomar la clase de programación!!), ¿qué debe haga el estudiante?

**ALGORITMO:**

|  |
| --- |
| **Inicio** Dormir  **haga**1**hasta** que suene el despertador (o lo llame la mamá).  Mirar la hora. ¿Hay tiempo suficiente? **Si** hay, **entonces**     Bañarse.     Vestirse.     Desayunar. **Sino,**        Vestirse. Cepillarse los dientes. Despedirse de la mamá y el papá.    ¿Hay tiempo suficiente? **Si**, Caminar al paradero. **Sino**, Correr al paradero. **Hasta** que pase un bus para la universidad **haga**:     Esperar el bus     Ver a las demás personas que esperan un  bus. Tomar el bus. **Mientras**no llegue a la universidad **haga**:      Seguir en el bus.     Pelear mentalmente con el conductor. Timbrar. Bajarse. Entrar a la universidad.  **Fin** |

***2. PROBLEMA:***Cambiar la rueda pinchada de un automóvil teniendo un gato mecánico en buen estado, una rueda de reemplazo y una llave inglesa.

**ALGORITMO:**

|  |
| --- |
| **Inicio** **PASO 1.**Aflojar los tornillos de la rueda pinchada con la llave inglesa. **PASO 2.**   Ubicar el gato mecánico en su sitio. **PASO 3.** Levantar el gato hasta que la rueda pinchada pueda girar libremente. **PASO 4.**   Quitar los tornillos y la rueda pinchada. **PASO 5.** Poner rueda de repuesto y los tornillos. **PASO 6**.    Bajar el gato hasta que se pueda liberar. **PASO 7.**    Sacar el gato de su sitio. **PASO 8.** Apretar los tornillos con la llave inglesa. **Fin** |

***3. PROBLEMA:*** Realizar la suma de los números 2448 y 5746.

**ALGORITMO:**

|  |
| --- |
| **Inicio** **PASO 1.** Colocar los números el primero encima del segundo, de tal manera que las unidades, decenas, centenas, etc., de los números queden alineadas. Trazar una línea debajo del segundo número. **PASO 2.**  Empezar por la columna más a la derecha. **PASO 3.**  Sumar los dígitos de dicha columna. **PASO 4.** Si la suma es mayor a 9 anotar un 1 encima de la siguiente columna a la izquierda y anotar debajo de la línea las unidades de la suma. Si no es mayor anotar la suma debajo de la línea. **PASO 5.**  Si hay más columnas a la izquierda, pasar a la siguiente columna a la izquierda y volver a 3. **PASO 6.** El número debajo de la línea es la solución. **Fin** |

**4. PROBLEMA:** Sean los puntos *P=(a,b)* y *Q=(c,d)* que definen una recta, encontrar un segmento de recta perpendicular a la anterior que pasa por el punto medio de los puntos dados.

**ALGORITMO:**

|  |
| --- |
| **Inicio**  http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/imagenes_02/image002.gif  **PASO 1.** Trazar un círculo con centro en el punto P que pase por el punto Q.  http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/imagenes_02/image004.gif  **PASO 2.** Trazar un círculo con centro en el punto Q que pase por el punto P.  http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/imagenes_02/image006.gif  **PASO 3.** Trazar un segmento de recta entre los puntos de intersección de las circunferencias trazadas.  http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/imagenes_02/image008.gif  **Fin.** El segmento de recta trazada es el buscado.  http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_02/imagenes_02/image010.gif |

|  |
| --- |
| **2.1.3   EJERCICIOS DE ALGORITMOS** |

Para los siguientes problemas dar un algoritmo y si es posible una ejecución del mismo.

1. Buscar en el directorio telefónico, el número de:

|  |
| --- |
| a. José González Pérez |
| b. Pedro Gómez Bernal. |
| c. Escribir un algoritmo que sirva para buscar a cualquier persona. |

2. Calcular el número de días entre las fechas:

|  |
| --- |
| a. Enero 17 de 1972 y Julio 20 de 1973 |
| b. Febrero 2 de 1948 y Agosto 11 de 1966 |
| c. Escribir un algoritmo que sirva para calcular la cantidad de días entre cualquier dos fechas. |

3. Solicitar en préstamo algún libro de una biblioteca.

4. haga una caja de cartón con tapa de:

|  |
| --- |
| a. 20 cm de largo, por 10 cm de ancho y 5 cm de alto. |
| b. 10 cm de largo, por 30 cm de ancho y 15 cm de alto. |
| c. Escribir un algoritmo que sirva para construir una caja de cartón con tapa de cualquier tamaño. |

5. Construir un avión de papel.

6. Calcular manualmente la división de cualquier par de números naturales. El resultado también debe ser un número natural. Escribir un algoritmo para calcular el residuo de la división.

7. Un juego muy famoso entre dos niños es el de *adivina mi número*, el cual consiste en que cada niño trata de adivinar el número pensado por el otro niño. Dicho número generalmente está entre 1 y 100. Las reglas del juego son las siguientes:

|  |
| --- |
| a. Cada niño posee un turno en el que trata de averiguar el número del otro. |
| b. En su turno el primer niño pregunta si un número que dice es el pensado por el segundo. |
| c. Si el número que ha dicho el primer niño es el que pensó el segundo, este último debe informarle al primero que ganó. |
| d. Si el número no es el segundo niño debe decir si su número pensado es menor o mayor al que el primer niño dijo. |
| e. Luego el segundo niño tiene su turno y de esta manera se van intercalando hasta que alguno de los dos gane.Desarrollar un algoritmo para jugar adivina mi número. |

8. Una balanza se encuentra en equilibrio cuando el producto de la carga aplicada sobre el brazo derecho por la longitud de este brazo, es igual al producto de la carga aplicada sobre el brazo izquierdo por la longitud de este otro brazo. Determinar si la balanza se encuentra en equilibrio si:

|  |
| --- |
| a. La longitud del brazo izquierdo es3 m, la del derecho es 2 m, la carga aplicada al brazo izquierdo es 5 Kg y la carga aplicada al derecho es 7 Kg. |
| b. La longitud del brazo izquierdo es4 m, la del derecho es 2 m, la carga aplicada al brazo izquierdo es 4 Kg y la carga aplicada al derecho es 4 Kg. |
| c. Desarrollar un algoritmo que sirva para cualquier conjunto de valores para las longitudes de los brazos y las cargas aplicadas. |