

Manual de Programación

ALGORITMOS



Facilitadores:

Telgo. César Vanegas
2014.

Índice

Presentación.....	1
Objetivos del Modulo	2
Objetivos Generales.....	2
Objetivos Específicos.....	2
Metodología de Trabajo y Evaluación.....	3
Introducción a los Algoritmos.....	4
1.1. Fundamentos.....	4
1.2. Metodología para la solución de problemas por medio de la computadora ..	5
1.3. Tipos de Algoritmos.....	6
Algoritmos Cualitativos:	7
Algoritmos Cuantitativos:	12
1.4. Técnicas para la formulación de algoritmos	14
Pseudocódigo	14
Diagrama de Flujo	14
Recomendaciones para el diseño de Diagramas de Flujo.....	14
1.5 Ejercicios Prácticos	17
Estructuras Algorítmicas	18
2.1. Variables, Constantes y Operadores	18
Cabecera.....	18
Sección de identificación de variables y constantes.....	19
2.1. Tipos de Datos.....	21
Datos Numéricos	22
Datos Lógicos	22
Datos de Carácter	22
Datos de Cadena (String).....	22
2.3. Operadores	23
Ejercicios Prácticos.....	29
Sentencias de Control.....	30
3.1. Sentencias de Control.....	30
3.2. Sentencias Secuenciales.....	31
Asignación.....	31
Ejercicios Resueltos.....	33
Ejercicios Propuestos.....	36
3.3. Sentencias de Decision o Condicionales.....	37
Estructura de Selección Simple SI.....	37
Estructura de Selección Doble SI - SINO.....	40
3.4. Sentencias de Selección Anidadas SI	42
Estructura de Selección Múltiple.....	43
Ejercicios Resueltos.....	45
Ejercicios Propuestos.....	50
3.5. Sentencias de Bucles o Lazos Repetitivos.....	54
Sentencia Para / Desde / Hasta / Haga.....	55
Ejercicios Resueltos.....	57

Ejercicios Propuestos.....	61
3.6. Sentencia Mientras Hacer	63
Contadores	65
Acumulador.....	65
Bucles Infinitos	66
Ejercicios Resueltos.....	66
Ejercicios Propuestos.....	68
3.7. Sentencia Repetir \Hasta	71
Ejercicios Resueltos.....	72
Ejercicios Propuestos.....	75
2.11. Elección Estructuras Repetitiva Adecuada.....	76
Autoevaluacion Final	77
Arreglos, Vectores y Matrices	79
4.1. Arreglos.....	79
4.2. Vectores	79
Ejercicios Propuestos.....	80
5.3. Matriz.....	81
Ejercicios Propuestos.....	82
Bibliografía.....	84
Anotaciones.....	85
Derechos Reservados.....	87

Presentación

El desarrollo vertiginoso de las computadoras ha hecho que cada tipo de persona comiencen tener acceso a la programación de sus propias máquinas, facilitando así el estudio y desarrollo de programas para la solución de problemas particulares en las empresas, oficinas y en mismo hogar.

La necesidad de implementar metodologías para el aprendizaje de la programación en diferentes lenguajes y paquetes de utilidad, obliga al cuestionamiento sobre el desarrollo de la lógica estructurada en las personas comunes obligándose, así, a pensar sobre la forma de ejecución de cualquier acción tanto humana como mecánica o de sistemas.

El desarrollo del módulo de algoritmos y flujogramas es un tema fundamental en el diseño de programas por lo cual el alumno debe tener buenas bases que le sirvan para poder desarrollar de manera fácil y rápida sus programas.

Este módulo de Algoritmos y Flujogramas le servirá de apoyo al maestro, en su labor cotidiana de aprendizaje y al estudiante le facilitará desarrollar su capacidad analítica y creadora, para de esta manera mejorar su destreza en la elaboración de algoritmos que sirven como base para la codificación de los diferentes programas que tendrá que desarrollar a lo largo de su carrera.

Se recomienda que el estudio de este material se lo realice en grupos, elaborando los ejercicios que se muestran en la Guía de Estudio. Éxitos en este módulo que solo trata de dar un enfoque actual y personal de las maravillosas capacidades que se puede lograr, dominando esta temática.

Los Autores.

Objetivos del Modulo

Objetivos Generales

- § Comprender la naturaleza de los algoritmos y flujogramas y su aplicación en la programación sobre computadoras.

Objetivos Específicos

- § Crear en el estudiante la necesidad y agilidad para pensar en forma lógica y ordenada con el fin de resolver problemas en forma eficiente.
- § Despertar en el estudiante el espíritu de programar
- § Dar las bases fundamentales para el desarrollo de cualquier tipo de programación simple, estructurada y orientada a Objetos como Pascal, C++ , Visual.Net

Metodología de Trabajo y Evaluación

La propuesta metodológica se basa en un modelo educativo centrado en el alumno y en los grupos colaborativos. En este sentido el participante es el responsable de su proceso de aprendizaje; debe interactuar con los contenidos del módulo mediante el sistema de estudio a distancia, desarrollar su juicio crítico y tener la iniciativa de aprender continuamente todo aquello que le sea esencial durante el proceso para cumplir con las intenciones del módulo de Algoritmos y flujogramas.

Así mismo, el participante debe realizar actividades con otros compañeros, de tal forma que entre ellos, construyen diferentes experiencias de aprendizaje que enriquezcan los contenidos y el desarrollo de distintas habilidades, por ejemplo, uso de las tecnologías, trabajo cooperativo, espacios de discusión de ideas, síntesis y análisis, juicio crítico, entre otras.

Durante el desarrollo de este módulo, tienen especial importancia actividades como:

- § La entrega de trabajos individuales y grupales por escrito, que se indicaran en los encuentros establecidos de clases presenciales.
- § Exposición de los trabajos grupales y de los trabajos de investigación utilizando herramientas tecnológicas o pedagógicas que usted conozca y domine.
- § La elaboración de las Autoevaluaciones y selección de temas para investigación y auto-estudio por parte de los alumnos, y aplicación de evaluaciones prácticas para reforzar el conocimiento adquirido por el alumno.
- § Ampliación de los temas por parte del facilitador.

CAPITULO

1

Introducción a los Algoritmos

- 1.1 Fundamentos
- 1.2 Metodología para la Solución de Algoritmos
- 1.3 Tipos de algoritmos
- 1.4 Técnicas para la formulación de Algoritmos
- 1.5 Ejercicios Prácticos

1.1. Fundamentos

Un elemento básico para el funcionamiento del computador es el Software, el cual está formado por una serie de instrucciones y datos que permiten el aprovechamiento de los diversos recursos de la computadora, para resolver gran cantidad de problemas. El software está constituido por programas los cuales son un conjunto detallado de instrucciones que instruyen al procesador para realizar determinados procesos particulares. Los datos pueden ser cualquier información que necesite el programa: caracteres, números, imágenes, etc.

Por otro lado un computador es incapaz de hacer cosas por su propia iniciativa, hace únicamente lo que se le ordene para lo cual es necesario especificarle detalladamente paso a paso de que manera debe ejecutar un conjunto de órdenes. Este proceso de enseñar al computador se conoce como programación: y su producto es el programa. Un programa se escribe en un lenguaje de programación el cual proporciona la interfaz hombre máquina. Así pues, los lenguajes utilizados para escribir programas de computadoras son los lenguajes de programación y los programadores son los escritores y diseñadores de programas. Un programador a su vez da solución a un problema mediante la resolución de algoritmos ó diagramas de flujo.



Un Algoritmo es un conjunto ordenado y finito de pasos o instrucciones que conducen a la solución de un problema específico. La naturaleza de los problemas varían con el ámbito o con el contexto donde están planteados; así, existen problemas matemáticos, químicos, filosóficos, etc. Según esto la naturaleza de los algoritmos también es variada y no todos ellos pueden ser ejecutados por la computadora..

La palabra algoritmo apareció por primera vez en el diccionario en 1957; antes se encontraba como la palabra algorismo, que significa, proceso de hacer cálculos aritméticos utilizando números arábigos.

En la edad media, los abacistas utilizaban el ábaco para hacer sus cálculos, mientras que los algoristas lo hacían con los algoritmos. Para esta época, no se sabía a ciencia cierta cual era el origen de la palabra algoritmo; los lingüistas intentaron hacer una aproximación, usando combinaciones de palabras como alirios (sin dolor) y arithmos (número); otros decían que la palabra provenía del "Rey Algor de Castile".

Finalmente, los historiadores de la matemática encontraron el verdadero origen de la palabra algoritmo: proviene del nombre del autor de un famoso libro persa, Abu Ja'far Mohammed ibn Musa al-Khowárizmí, literalmente "Padre de Ja'far, Mohammed, hijo de Moses, nativo de Khowárizm. Al-Khowárizm escribió un libro muy famoso que sentó las bases del álgebra.

1.2. Metodología para la solución de problemas por medio de la computadora

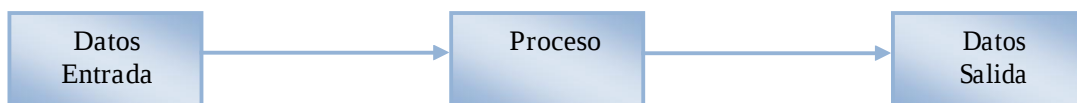
Definición del Problema: Esta fase está dada por el enunciado del problema y consiste en estudiar el problema planteado dividiendo en tantas partes como sea posible, para obtener una idea clara y concisa sobre lo que se desea obtener como resultado. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.

Análisis del Problema: Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

Los datos de entrada.

Cual es la información que se desea producir (salida)

Los métodos y fórmulas que se necesitan para procesar los datos.



Una recomendación muy práctica es el que nos pongamos en el lugar de la computadora y analicemos que es lo que necesitamos que nos ordenen y en que secuencia para producir los resultados esperados.

Diseño del Algoritmo: Consiste en describir las secuencias ordenadas de pasos que conducen a la solución de un problema dado. Las características de un buen algoritmo son:

Ser algoritmo: Tiene que consistir en una secuencia de instrucciones claras y finitas, es decir debe estar compuesto por un número determinado de pasos que tiene fin sin importar el número de instrucciones a utilizar.

Ser correcto: El algoritmo ha de resolver el problema planteado en todas sus facetas, teniendo en cuenta todas las posibles situaciones que puedan plantearse y no debe permitir dobles interpretaciones

Ser legible. Donde cada secuencia de un algoritmo debe ser definida en forma precisa, de tal manera que las acciones a ejecutar en cada uno, estén rigurosamente especificadas y no admitan ambigüedad.

Ser eficiente: todos los pasos, instrucciones u operaciones deben optimizarse de manera que se obtenga la solución más corta y eficiente de resolver el problema planteado

Codificación : La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora, la serie de instrucciones detalladas se le conoce como código fuente, el cual se escribe en un lenguaje de programación o lenguaje de alto nivel.

Prueba y Depuración: Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama depuración.

La depuración o prueba resulta una tarea tan creativa como el mismo desarrollo de la solución, por ello se debe considerar con el mismo interés y entusiasmo. Resulta conveniente observar los siguientes principios al realizar una depuración, ya que de este trabajo depende el éxito de nuestra solución.

1.3. Tipos de Algoritmos

El concepto de algoritmo se aplica a muchas situaciones cotidianas. Por ejemplo, cuando se quiere explicar a una persona, la ubicación de una casa o edificio, generalmente se le describe la ruta que debe seguir desde un determinado punto para llegar a la dirección deseada, en otras palabras, se le proporciona un algoritmo que le permite llegar a al destino sin ningún inconveniente. Una receta de cocina también puede considerarse como un algoritmo para elaborar una comida. Los planos de un modelo a escala, indican el orden y la forma en que deben ser unidas las piezas para construir el modelo. En este contexto existen dos tipos de algoritmos computacionales.

- § Algoritmos Cualitativos
- § Algoritmos Cuantitativos

Algoritmos Cualitativos:

Son aquellos en los que se describen cualquier acción o conjunto de acciones utilizando palabras pero siempre enmarcadas en tres estructuras fundamentales que son :

1. Secuencias de acciones
2. Decisión de acción
3. Ciclos de acciones

Secuencias de acciones: Para escribir una secuencia de ordenes o acciones todo lo que tiene que hacer es colocar una nueva orden o una nueva acción después de la última que haya colocado. De esta manera se entiende la secuencialidad y la ordinalidad en la ejecución de esas acciones.

Ejemplo:

Ejercicio para cambiar un foco quemando por uno nuevo.

- Situar una escalera bajo el foco quemado
- Elegir un nuevo foco
- Subir la escalera hasta alcanzar el foco
- Girar el foco en sentido antihorario hasta sacarlo
- Colocar el nuevo foco en el mismo lugar
- Girar en sentido horario hasta que quede apretado
- Bajar de la escalera

Puede usted notar que para utilizar la estructura de secuencia (que a veces parece ser todo lo que tenemos que hacer es ir colocando una acción tras otra y, por supuesto, racionales en el orden de dichas acciones, el orden de los factores sí altera el resultado en los diseños de algoritmos.

Decisión de acción: Son condiciones que nos permite que podamos decidir cuál es el camino lógico correcto a tomar para resolver un problema eficientemente.

Vamos a desarrollar el mismo algoritmo, para ello vamos a incorporar una líneas de decisión que nos permitan tener un algoritmo mas genérico y que nos permita lograr mejor el objetivo, así:

1. Situar una escalera bajo el foco quemado
2. Elegir un nuevo foco
 - Si la potencia no es igual a la del quemado Entonces (Pregunta)
 - Descartar el foco elegido (acción verdadera)
 - Caso contrario
 - Elegir un nuevo foco. (acción falsa)
3. Subir la escalera hasta alcanzar el foco
4. Girar el foco en sentido antihorario hasta sacarlo
5. Colocar el nuevo foco en el mismo lugar
6. Girar en sentido horario hasta que quede apretado
7. Bajar de la escalera

Ya puede usted notar que nuestro algoritmo ha cambiado un poco y por lo tanto ahora tiene unas condiciones que le permiten ser una secuencia de acciones mas racional y analiza mas detalladamente las situaciones posibles. Adicionalmente usted puede notar que una decisión completa involucra:

Una pregunta que evalúa una condición

Un conjunto de acciones a realizar en caso de que la condición sea verdadera

Un conjunto de acciones a realizar en caso de que la condición sea falsa

Ciclos de acciones: No es mas que la estructura que nos permite repetir una o varias acciones una cantidad definida de veces.

Para identificación de acciones repetitivas usted puede representar con la palabra Mientras/Hasta o Repetir/Hasta, bloques que establece en relación con una condición el inicio de un conjunto de acciones que se repiten precisamente Mientras esa condición lo permita. La estructura Mientras (por efectos de clarificación del algoritmo) debe tener un finalizador que indique hasta donde llega el bloque de acciones que debemos repetir.

1. Situar una escalera bajo el foco quemado
2. Elegir un nuevo foco
- Si la potencia no es igual a la del quemado Entonces
- Repetir
- Descartar el foco elegido
- Elegir un nuevo foco
- Hasta que la potencia sea igual
3. Mientras no se alcance el foco quemando Hacer
- Subir la escalera, peldaño tras peldaño
4. Repetir
- Girar el foco en sentido antihorario
- Hasta que el foco haya salido
5. Colocar el foco nuevo en el mismo lugar
6. Repetir
- Girar el foco en sentido horario
- Hasta que quede apretado
7. Bajar de la escalera

Este algoritmo informal están expresado tal como desprevenidamente cualquier persona los expresaría y puede entonces suponer usted que la variabilidad de algoritmos que cumplan los mismos objetivos sería inmensa si no existieran unas técnicas uniformes para facilitar la expresión de estas ideas, particularmente en algoritmos computacionales.

Ejemplo:

Algoritmo para usar un teléfono público. La operación puede expresarse en los siguientes pasos:

- Descolgar la bocina
- Introducir las monedas
- Discar el número
- Hablar
- Colgar la bocina

Los pasos anteriores parecen describir de forma correcta de usar un teléfono público. Ahora supóngase que le proporciona estas instrucciones a una persona que jamás en su vida ha utilizado un teléfono público. ¿Podrá esta persona completar exitosamente una llamada telefónica? Probablemente ¡no verdad!. Existen muchos aspectos que no son mencionados en el proceso anterior, como por ejemplo, que sucede si el número está ocupado, o si el teléfono no sirve, o si se agota el tiempo. Si bien el proceso tiene un inicio y un final, los pasos son poco específicos y no contempla todas las situaciones; por ende es posible que existan situaciones en que no pueda completarse una llamada telefónica, siguiendo los pasos. Por esta razón, el proceso antes descrito está un poco lejos de ser un algoritmo efectivo para usar un teléfono público.

Ahora procedemos a refinar el proceso anterior. Para esto, hay que especificar ciertos aspectos con respecto al uso de un teléfono público. Primero, ¿Qué sucede si al descolgar la bocina, no se oye ningún tono? Lo más seguro es que el teléfono esté dañado, por lo que no queda más remedio que colgar la bocina e intentar ubicar otro teléfono público. Así, el paso "Descolgar la bocina" del proceso anterior, puede ser extendido a lo siguiente:

- Descolgar la bocina
- Si no se oye un tono entonces
- Colgar la bocina
- Fin

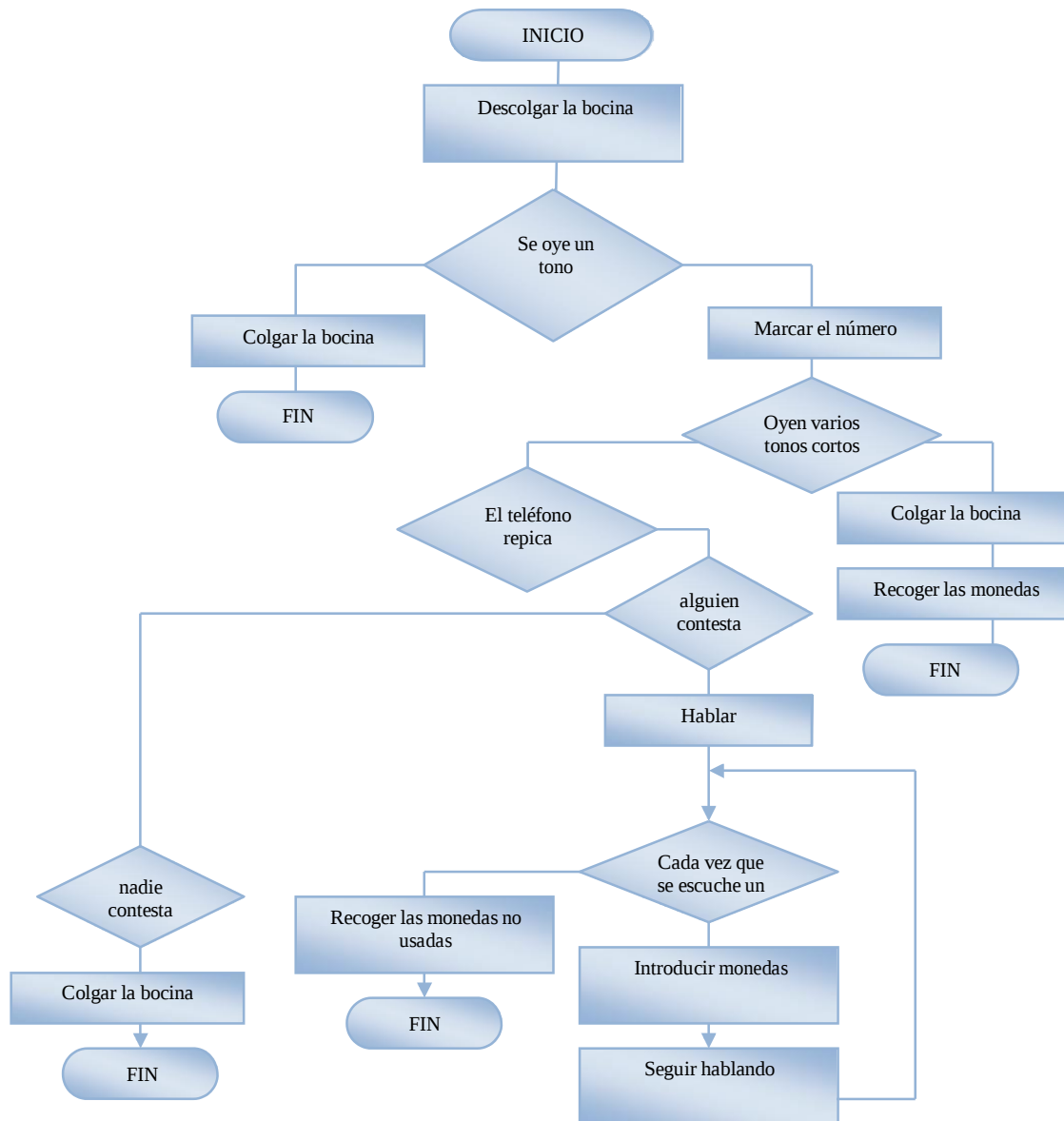
Cuando se termina de marcar el número, es posible que sucedan varias cosas: que alguien conteste, que nadie conteste o que el número esté ocupado. En el primer caso, la persona debe proceder a hablar; si nadie contesta o si está ocupado, la persona debe colgar la bocina y recoger las monedas. En caso de que alguien conteste, es posible que la persona deba introducir otras monedas, para mantener la llamada. Todas estas consideraciones se expresan en los siguientes pasos:

- Marcar el número
- Si se oyen varios tonos cortos seguidos (el número está ocupado)
- Colgar la bocina
- Recoger las monedas
- Fin
- Si se oyen varios tonos largos (el teléfono repica)
- Si alguien contesta
- Hablar
- Cada vez que se escuche un tono mientras se habla

Introducir monedas
Seguir hablando
Colgar la bocina
Fin
Si nadie contesta
Colgar la bocina
Fin

De esta forma se ha especificado de forma razonable los pasos a seguir para realizar una llamada usando un teléfono público. Todo algoritmo, de una forma u otra, indica el orden en que deben ejecutarse los pasos. En este caso, se agregan números a cada paso, para indicar la secuencia apropiada:

1. Descolgar la bocina
2. Si no se oye un tono entonces
 - 2.1. Colgar la bocina
 - 2.2. Fin
3. Marcar el número
4. Si se oyen varios tonos cortos seguidos
 - 4.1. Colgar la bocina
 - 4.2. Recoger las monedas
 - 4.3. Fin
5. Si se oyen varios tonos largos
6. Si alguien contesta
 - 6.1. Hablar
 - 6.2. Cada vez que se escuche un tono mientras se habla
 - 6.2.1. Introducir monedas
 - 6.2.2. Seguir hablando
 - 6.3. Colgar la bocina
 - 6.4. Recoger las monedas no usadas
 - 6.5. Fin
7. Si nadie contesta
 - 7.1. Colgar la bocina
 - 7.2. Recoger las monedas
 - 7.3. Fin



En los dos algoritmos anteriores aparecen ciertas palabras reservadas (en cursiva) que describen las estructuras de control fundamentales y procesos de toma de decisiones. Estas palabras incluyen los siguientes conceptos fundamentales:

Selección : expresada por *si-entonces-caso contrario*

Repetición: expresada por *repetir-hasta que* o a veces por *mientras-hacer*

La capacidad de decisión permite seleccionar alternativas de acciones a seguir o bien la repetición una y otra vez de operaciones básicas.

Un aspecto importante a considerar al momento de escribir un algoritmo es la indentación (sangrado) de las acciones interiores a las estructuras de selección y/o repetición.

Otro aspecto, también importante es el método por medio del cual se desarrolla un algoritmo. Se comienza con un enunciado muy general y se continúa hasta encontrar el algoritmo final incrementando sistemáticamente los detalles.

Algoritmos Cuantitativos:

Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. De igual forma estos tipos de algoritmos describen tres partes esenciales:



Ejemplo: Diseñe un algoritmo que permita hallar la suma y el promedio de tres números

Se definen: E-P/S

Entrada: Números enteros (Numero1, numero2, numero3)

Proceso: Sumar los 3 números y calcular su Promedio

Salidas: Suma, Promedio

Pseudocódigo

INICIO

LEER numero1, numero2, numero3

suma = numero1 + numero2 + numero3

promedio = suma / 3

IMPRIMIR suma, promedio

FIN

Notas: El término LEER significa obtener un dato de algún dispositivo de entrada, como el teclado, y almacenarlo en una variable. Una variable es una localización en la memoria que tiene un nombre y cuyo contenido puede cambiar a lo largo de la ejecución de un programa. Así numero1, numero2 y numero3 son variables. El término IMPRIMIR significa mostrar el valor de una variable en algún dispositivo de salida, como la pantalla

Ejemplo de un Algoritmo Cuantitativo:

Determinar la suma de los N primeros números enteros de acuerdo a la siguiente fórmula: $\text{Suma} = N*(N+1)/2$

Se definen: E/S

Entrada: Número entero (N)

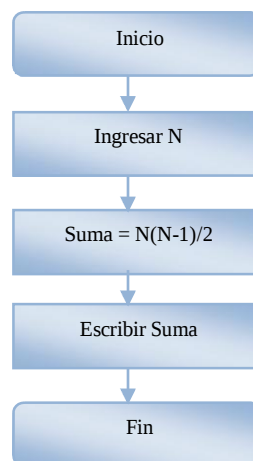
Proceso: Calcular suma por fórmula:

Salida: Suma

Pseudocódigo

```
INICIO
Ingresar un número entero
Leer (N)
 $\text{Suma} = N*(N+1)/2$ 
Escribir( suma)
FIN
```

Diagrama De Flujo



1.4. Técnicas para la formulación de algoritmos

Las dos herramientas utilizadas comúnmente para diseñar algoritmos son:

- § Pseudocódigo
- § Diagrama de Flujo

Pseudocódigo

Es un lenguaje algorítmico de programación informal, en donde se omiten aspectos técnicos, para considerar solamente operaciones que definen el comportamiento del algoritmo.). Es decir es una herramienta algorítmica que permite escribir pseudoprogramas (una imitación de un programa real) utilizando un lenguaje de pseudoprogramación que es una imitación de los lenguajes de programación de alto nivel.

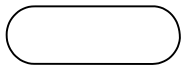
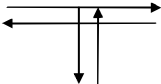
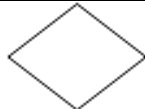
Diagrama de Flujo

Un diagrama de flujo es la representación gráfica de un algoritmo. También se puede decir que es la representación detallada en forma gráfica de como deben realizarse los pasos en la computadora para producir resultados

Un diagrama de flujo utiliza los símbolos estándar (caja) y que tiene los pasos del algoritmo escritos en esas cajas unidas por flechas, denominada línea de flujo, que indican la secuencia en que se deben ejecutar. Los símbolos utilizados han sido normalizados por el instituto norteamericano de normalización (ANSI).

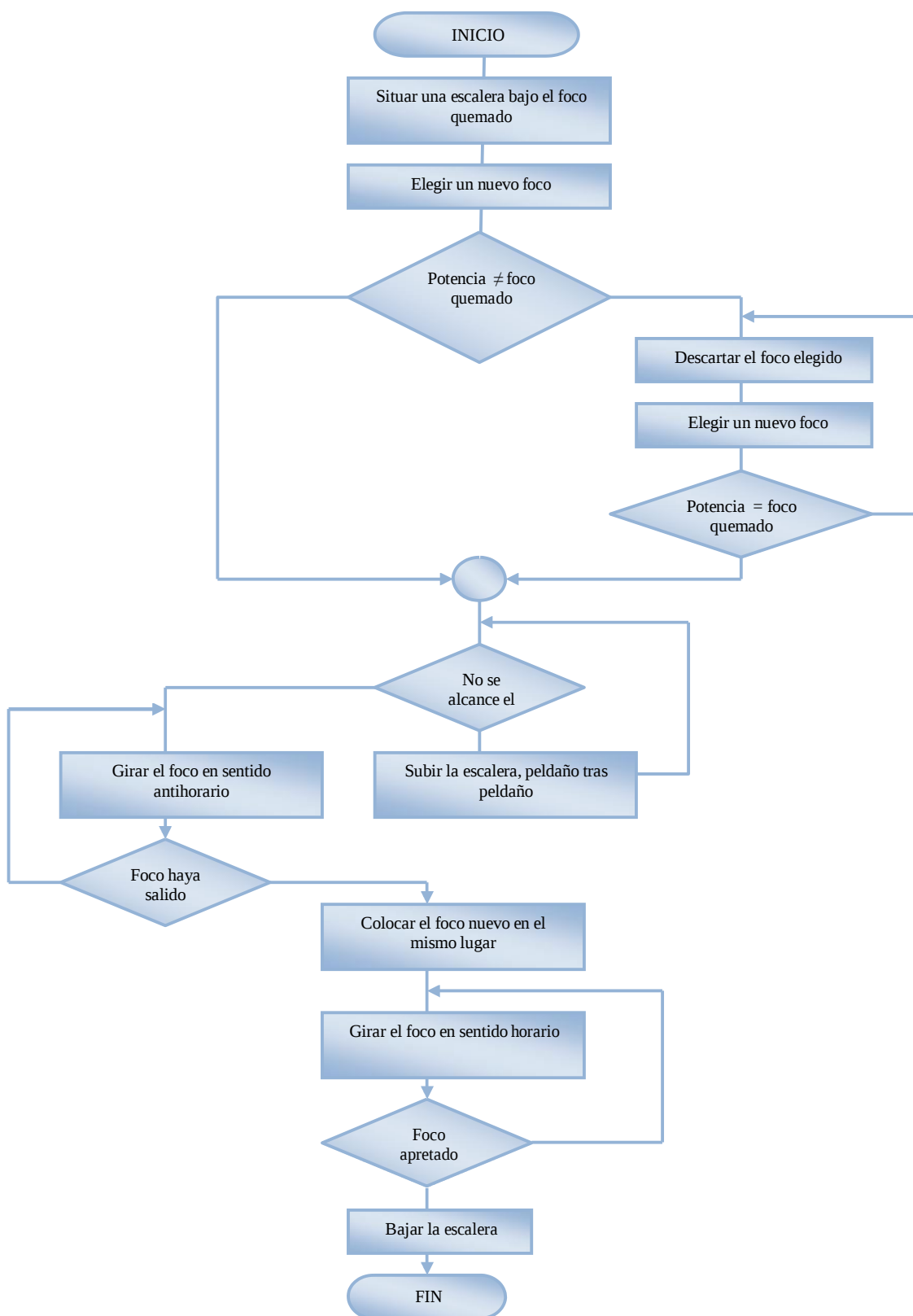
Recomendaciones para el diseño de Diagramas de Flujo

- § Se deben usar solamente líneas de flujo horizontales y/o verticales.
- § Se debe evitar el cruce de líneas utilizando los conectores.
- § Se deben usar conectores solo 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.

DIAGRAMA	DESCRIPCIÓN
	Establece el INICIO y el FIN.
	Indica un PROCESO interno de una operación aritmética.
	Introducción de datos por parte del usuario. (E/S)
	SALIDA de información POR PANTALLA.
	Indica la continuidad de un diagrama. CONECTOR.
	Volumen de DATOS.
	Indica el SENTIDO del flujo en que se realizan las operaciones.
	Especifica la realización de una comparación de valores. DECISIÓN.
	SALIDA de información POR IMPRESORA.
	Se utiliza en ocasiones en lugar del símbolo de E/S para representar el ingreso de datos solo por teclado
	Indica un proceso externo. SUBROUTINA.
	Se usa para enlazar dos partes cualesquiera de un ordinograma a través de un conector en la salida y otro conector en la entrada
	Conector fuera de pagina. Representa la continuidad del diagrama en otra pagina.

Ejemplo:

Diagrama de Flujo para cambiar un foco quemando por uno nuevo.



1.5 Ejercicios Prácticos

Diseñar los algoritmos que resuelvan los siguientes problemas:

- § Algoritmo y flujograma para colocarnos una camisa
- § Algoritmo Para Ver La Película Tiburon
- § Comprar una entrada para un show artístico.
- § Elabore un pseudocódigo y flujograma que sume, reste, multiplique y divida dos números: $x = 10$ y $y = 2$
- § Elabore un pseudocódigo y flujograma que calcule el área de un rectángulo:
Lado1 = 3 lado2 = 4, Area del rectángulo=lado1 * lado2
- § Elabore un pseudocódigo y flujograma que reciba un número entero y devuelva su inmediato siguiente. Por ejemplo si ingresa 28 la salida debe ser 29.
- § Una tienda de abastos ofrece un descuento del 8% sobre el total de la compra y un cliente desea saber cuanto deberá pagar finalmente por su compra. Elabore un pseudocódigo y flujograma que resuelva este problema.

CAPITULO

2

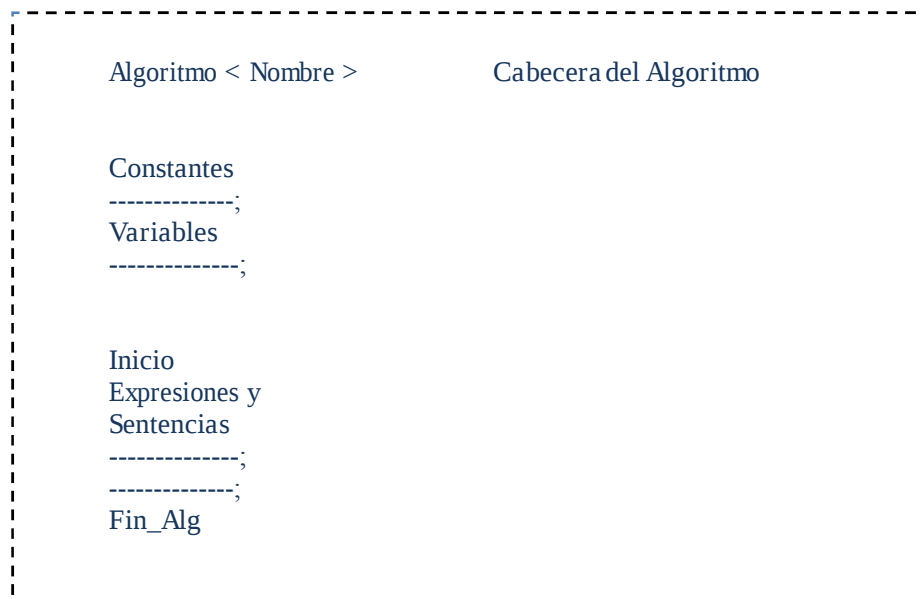
Estructuras Algorítmicas

- 2.1 Variables, Constantes y Operadores
- 2.2 Tipos de Datos
- 2.3 Operadores

2.1. Variables, Constantes y Operadores

El recurso fundamental utilizado por un programa es la memoria. Es en la memoria donde un programa almacena sus datos e instrucciones. Los datos se almacenan en localizaciones en la memoria denominadas variables..

Un algoritmo constara de tres componentes: Una cabecera, una sección de Identificadores, Variables y Constantes, y el cuerpo del Algoritmo.



Cabecera

Comienza con la palabra ALGORITMO, la cual está seguida por el nombre del programa completo, Ejemplo: Algoritmo suma;

Sección de identificación de variables y constantes

En ella se especifican los tipos de datos a utilizar (Cuales y de que tipo).

Identificadores

Un identificador es un nombre que puede darse a una variable, a una constante y en general a cualquier elemento de un programa que necesite nombrarse.

Así, por ejemplo, son válidos los siguientes identificadores:

sueldoBruto	Sueldo bruto de un empleado
Edad	Edad de una persona
numero_hijos	Número de hijos de un empleado
DIA	Nombre de un día de la semana

Reglas para formar un Identificador

- § Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no deben contener espacios en blanco.
- § Letras, dígitos y caracteres como la subraya (_) están permitidos después del primer carácter.
- § La longitud de identificadores puede ser de hasta 8 caracteres.

Constantes

Una constante es un dato que no cambia durante la ejecución del programa. Ejemplo:

pi = 3.1416

Variables

Una variable es una localización en la memoria principal que almacena un dato que puede cambiar a lo largo de la ejecución del programa.

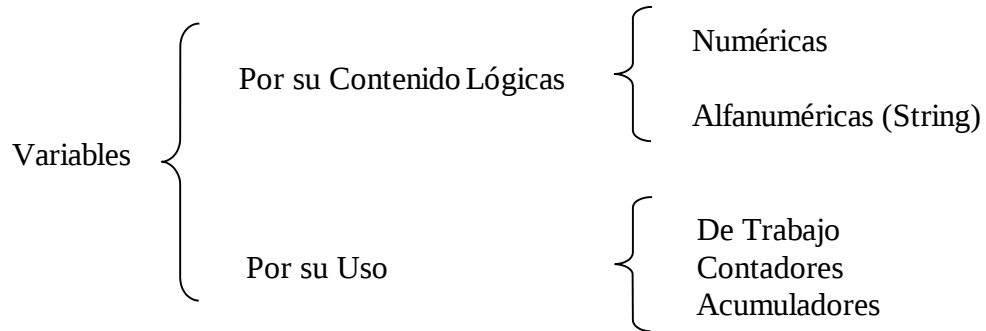
Una variable tiene asociada dos cosas fundamentales: un identificador y un tipo de dato.

- § El nombre identifica de manera única la localización de memoria donde se almacena el dato.
- § El tipo de dato especifica la naturaleza del dato que puede almacenar la variable.

En el gráfico siguiente se muestran dos variables con sus contenidos en memoria. La variable llamada Edad (de tipo entero) cuyo contenido es 27 y la variable llamada Descuento (de tipo real) cuyo contenido es 23.57.

Edad	301
Descuento	23.57

Según el tipo de dato de la variable podemos tener variables:



Variables por su Contenido

Variable Numéricas

Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal. Ejemplo:

```
iva=0.15
pi=3.1416
costo=2500
```

Variables Lógicas

Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.

Variables Alfanuméricas

Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales). Ejemplo:

```
letra='a'
apellido='lopez'
direccion='Av. Libertad #190'
```

Variables por su Uso

Variables de Trabajo:

Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa. Ejemplo:

```
suma=a+b/c
```

Contadores

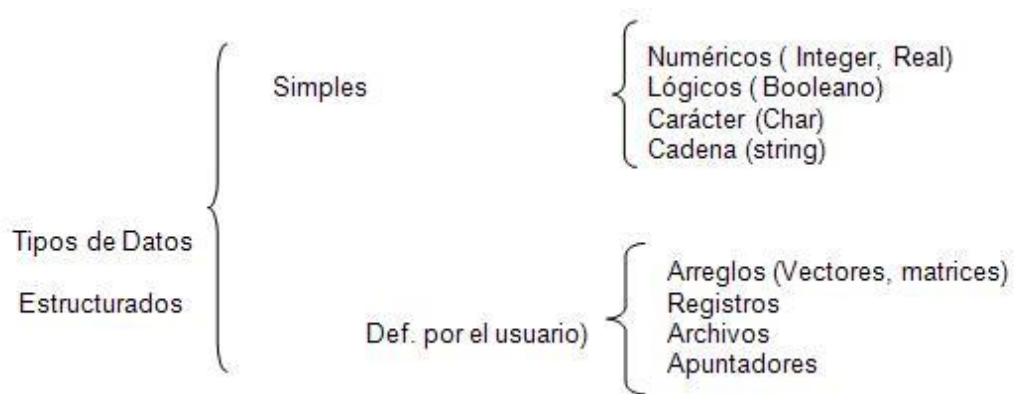
Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno.

Acumuladores

Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.

2.1. Tipos de Datos

El primer objetivo de la computadoras es el manejo de la información o datos . Todos los Algoritmos y programas operan sobre datos. 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.



Datos Numéricos

Permiten representar valores escalares de forma numérica, esto incluye a los números enteros y los reales. Este tipo de datos permiten realizar operaciones aritméticas comunes.

Enteros

Es un subconjunto finito de los números enteros que pueden ser negativos o positivos Ej.

5 1340 -5
20 -25 4

Reales

Tienen un punto decimal y pueden ser positivos o negativos.

0.08 3739.41 3.0
3.7452 -52.321 -8.12

Datos Lógicos

También denominado booleano y son aquellos que solo pueden tener dos valores (verdadero o falso; Si o No) ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).

Por ejemplo: Cuando se pide si un valor entero es par, la respuesta será verdadera o falsa, según sea par o impar.

Datos de Carácter

Es tipo carácter es el conjunto finito y ordenado de caracteres que la computadora entiende. Un dato tipo carácter contiene un solo carácter delimitado por apóstrofes. Dentro de los caracteres se reconocen:

Caracteres Alfabéticos ('A', 'B', 'C', ..., 'Z').
Caracteres numéricos ('1', '2', '3', ..., '9').
Caracteres especiales ('+', '-', '*', '/', '^', '.', ',', '<', '>', '..')

Datos de Cadena (String)

Es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombres de personas, direcciones, etc. Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir no es posible hacer operaciones con ellos. Este tipo de datos se representan encerrados entre comillas. Ejemplo:

“ Universidad Estatal de Bolívar ”

Operaciones de Asignación

Es el modo de darle valores ó tipo de dato a una variable. La asignación de un valor a una Variable se representa con el símbolo $\Leftarrow$

NOMBRE DE LA VARIABLE $\Leftarrow$ EXPRESIÓN

Ejemplos:

A,B,C,D $\Leftarrow$ Entero; Significa que a la variable A,B,C,D se le ha asignado el soporte de valores enteros

A $\Leftarrow$ 5 Significa que a la variable A se le ha asignado el valor 5

B $\Leftarrow$ 5+3+8 La variable B se le ha asignado el valor 16

C $\Leftarrow$ B*A La variable C se le ha asignado el valor de 80

D $\Leftarrow$ B/2 La variable D se le ha asignado el valor 8

Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales. Ej.

$a+(b+3)+c$

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.

Una expresión consta de operadores, operandos y funciones. Según sea el tipo de datos que manipulan, se clasifican las expresiones en:

- § Aritméticas
- § Relacionales
- § Lógicas

2.3. Operadores

Son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes. Es decir, los operadores nos permiten manipular valores.

Los Tipos de Operadores que existen son tres: los relacionales, los Aritméticos y por último los lógicos

Operadores Aritméticos

Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes).

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real.

+	Suma	^, **	Exponenciación
-	Resta	div	División Entera
*	Multiplicación	mod	Modulo (Residuo)
/	División		

Ejemplos:

Expresión	Resultado
5 + 3 + 10	18
10 - 7 - 1	2
2 * 5	10
11 / 2	5.5
3 ^ 3	27
10 div 3	3
10 mod 3	1

Operadores Relacionales:

Se utilizan para establecer una relación entre dos valores. Compara estos valores entre si y esta comparación produce un resultado de certeza o falsedad (verdadero o falso).

Los operadores relacionales comparan valores del mismo tipo (numéricos o cadenas). Tienen el mismo nivel de prioridad en su evaluación. Los operadores relacionales tiene menor prioridad que los aritméticos

Es decir es un operador que sólo puede tomar dos valores lógicos que son: Verdadero ó Falso. También se denominan expresiones booleanas en honor del matemático Británico George Boole, que desarrolló el Algebra lógica o de Boole.

Operadores de Relación	
Operador	Significado
<	Menor que
>	Mayor que
=	Igual que
<=	Menor o igual a
>=	Mayor o Igual a
<>	Distinto de

Ejemplo:

$3 < 6$	Verdadero
$0 > 1$	Falso
$4 = 2$	Falso
$(4 - 2) < (3 - 4)$	Falso

Operadores Lógicos

Los operadores lógicos o booleanos básicos son

OPERADOR	SIGNIFICADO
no(not)	negación de
Y (and)	intersección de
O (Or)	Unión de

Tabla de verdad:

P	Q	P y Q	P o Q	no P
v	v	v	v	f
v	f	f	v	f
f	v	f	v	v
f	f	f	f	v

En las expresiones lógicas se pueden mezclar operadores de relación y lógicos.

Ejercicio.

$(10 > 5)$ y $(4 \bmod 3) = 0$
v f Falso

$0 < 5$ ó $0 > 5$
v v Verdadero

$(a < b)$ and $(b < c)$
 $(10 < 20)$ and $(20 < 30)$
V and V
 V

Prioridad o Precedencia de los Operadores

Las Expresiones que tiene dos o más operandos requieren unas reglas matemáticas que permitan determinar el orden de las operaciones y estas son:

- § Las operaciones que están encerradas entre paréntesis se evalúan primero. Si existen diferentes paréntesis anidados (Interiores unos a otros) las expresiones más internas se evalúan primero
- § Las operaciones aritméticas dentro de una expresión suelen seguir el siguiente orden de prioridad:

1. ()
2. ^
3. *, /, Mod, Not
4. +, -, And
5. >, <, > =, < =, < >, =, Or

En caso de coincidir varios operadores de igual prioridad en una expresión o subexpresión encerrados entre paréntesis en orden de prioridad en este caso es de izquierda a derecha

Ejemplos

$$\begin{aligned} 3 + 6 * 14 \\ = 3 + 84 \\ = 87 \end{aligned}$$

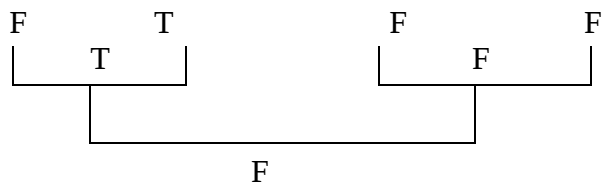
$$\begin{aligned} 8 + 7 * 3 + 4 * 6 \\ = 8 + 21 + 4 * 6 \\ = 8 + 21 + 24 \\ = 53 \end{aligned}$$

$$\begin{aligned} (9 + 11) / (8 + 4 / 2) &= 20 / (8 + 4 / 2) \\ &= 20 / (8 + 2) \\ &= 20 / 10 \\ &= 2 \end{aligned}$$

- § $4 + 2 * 5 = 14$
- § $23 * 2 / 5 = 9.2 \quad 46 / 5 = 9.2$
- § $3 + 5 * (10 - (2 + 4)) = 23 \quad 3 + 5 * (10 - 6) = 3 + 5 * 4 = 3 + 20 = 23$
- § $3.5 + 5.09 - 14.0 / 40 = 5.09 \quad 3.5 + 5.09 - 3.5 = 8.59 - 3.5 = 5.09$
- § $2.1 * (1.5 + 3.0 * 4.1) = 28.98 \quad 2.1 * (1.5 + 12.3) = 2.1 * 13.8 = 28.98$

Ejemplo:

$$\begin{aligned} a = 10 \quad b = 12 \quad c = 13 \quad d = 10 \\ ((a > b) \text{ or } (a < c)) \text{ and } ((a = c) \text{ or } (a > = b)) \end{aligned}$$



Funciones

Los programas exigen en numerosas ocasiones, además de las operaciones aritméticas básicas ya tratadas, un número determinado de operadores especiales que se denominan funciones.

FUNCION	DESCRIPCION	TIPO DE ARGUMENTO	RESULTADO
Abs (x)	Valor absoluto de x	Entero o Real	Igual que argumento
Arctan (X)	Arco tangente de x	Entero o real	Real
Cos (x)	Coseno de x (radianes)	Entero o real	Real
Exp (x)	Calcula e x	Entero o real	Real
Round(x)	Redondea el valor de x al entero más próximo	Real	Entero
sqr (X)	Calcula el cuadrado de x	Entero o real	Igual que x
Sqrt(X)	Calcula la raíz cuadrada de x	Entero o real	Real
Trunc(X)	Suprime la parte decimal de x	Real	Entero
Chr(x)	Devuelve el carácter de un entero x	Entero o real	Carácter
Ord(x)	Devuelve un entero de un carácter	Carácter	Entero
Pred(x)	Devuelve el predecesor de x	Entero carácter o booleano	Igual que x
Succ(x)	Devuelve el sucesor de x	Entero carácter o booleano	Igual que x

Ejemplo

Chr (65) = 'A'	Ord('A') = 65	round(2.3)=2	Trunc(2.3)=2
Chr (112) = 'p'	Ord('p') = 112	round(3.7)=4	trunc(3.7)=3
Chr (53) = '5'	Ord('5') = 53	Sin(30) = 0.5	sqrt(16) = 4

$\text{Pred}(3) = 2$
 $\text{Succ}('e') = 'f'$

$\text{Succ}(3) = 4$

$\text{Sqr}(3) = 9$

$\text{Pred}('e') = 'd'$

Ejercicios Prácticos

1. Obtener el valor de las siguientes expresiones:

- § $-4 * 7 + 2 ^ 3 / 4 - 5$
- § $6 / 2 * (7 + 5)$
- § $6.0 * (5 \bmod 3) + (15 \text{ div } 4) / (5 + (6 * (2 ^ 3 / 2) \text{ div } 3) - 1)$
- § $9 \text{ div } 2/5$
- § $2 \text{ div } 3 + 3/5$
- § $-4 * 7 + 2 ** 3 / 4 - 5$
- § $(5 < (4+10)) \text{ y } (6 = 6)$
- § $\text{no } ((5 < (4+10)) \text{ y } (6 = 6))$
- § $(10 \text{ div } 2) <> 0 \text{ ó } (\text{no } (4 < -10))$
- § $(2*3*5) <= (10/5+8)$
- § $(6=8) \text{ ó } (6 >= 8)$
- § $(2*8) <> (4 \bmod 3)$

2. Identifique a que tipo de dato corresponde los siguientes ejercicios

- | | |
|-----------|------------|
| § 0.5 | f. “xyz” |
| § 27,822 | g. 0515 |
| § 9.3e12 | h. ‘9’ |
| § 9.3e-12 | i. 1.66e+8 |
| § ‘a’ | j. “verde” |

3. Determinar cuáles de los siguientes son identificadores válidos. Si son válidos, explicar por qué.

- | | |
|----------------------|-----------------------|
| § a. Area | f. 1valor |
| § b. Altura | g. Costo neto |
| § c. Valor 1 | h. ÁREA |
| § d. Ventas95 | i. Nombre y dirección |
| § e. POSICION_ACTUAL | j. 123-45-6789 |

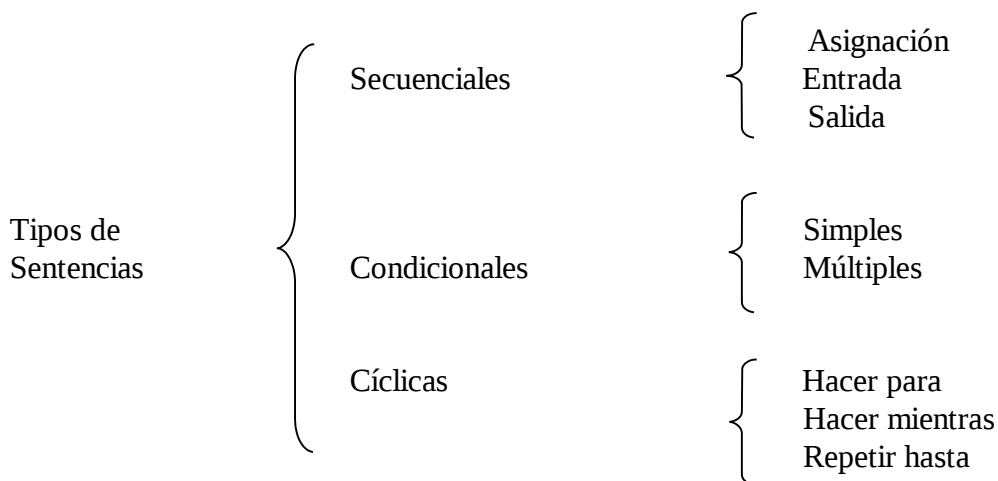
CAPITULO 3

Sentencias de Control

- 3.1 Sentencias de Control
- 3.2 Sentencias Secuenciales
- 3.3 Sentencias de Decisión o Secuenciales
- 3.4 Sentencias de Selección Anidada
- 3.5 Sentencias de Bucles o Lazos Repetitivos
- 3.6 Elección de Estructuras Repetitivas

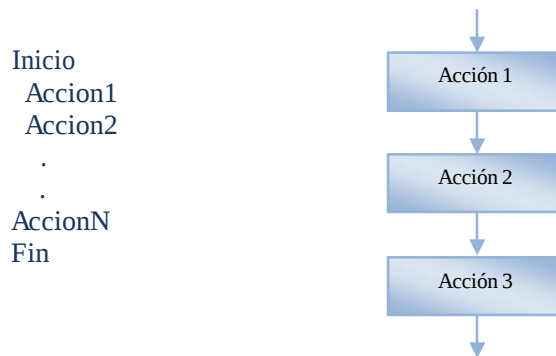
3.1. Sentencias de Control

Las Sentencias algorítmicas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas sentencias se clasifican de acuerdo con su complejidad en:



3.2. Sentencias Secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:



Asignación

La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor. La asignación se puede clasificar de la siguiente forma:

Simple

Consiste en pasar un valor constante a una variable ($a=15$)

Contador: Consiste en usarla como un verificador del número de veces que se realiza un proceso ($a=a+1$)

Acumulador: Consiste en usarla como un sumador en un proceso ($a=a+b$)

De trabajo: Donde puede recibir el resultado de una operación matemática que involucre muchas variables ($a=c+b*2/4$).

Entrada y Salida de Información

Todo cálculo que se realiza en la computadora requieren la entrada de datos necesarios para ejecutar las operaciones que posteriormente se convertirán en resultados es decir, salida.

Las operaciones de entrada permiten leer determinados valores y asignarlos a determinadas variables. Esta entrada se conoce como operación de lectura (read). La operación de salida se denomina escritura (write)

Lectura

La lectura consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor. Esta operación se representa en un pseudocódigo como sigue:

Las acciones de lectura de algoritmos se representan por el formato:

```
Leer ( lista de variables de entrada )  
Leer ( a, b )
```

Donde “a” y “b” son las variables que recibirán los valores

Escritura

Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje. Este proceso se representa en un pseudocódigo como sigue:

Las acciones de escritura de algoritmos se representan por el formato:

```
Imprimir ó Escribir (< ” Mensaje” >, lista de variables de salida )  
Imprimir ( “El resultado es:”, R)
```

Donde “El resultado es:” es un mensaje que se desea aparezca y R es una variable que contiene un valor.

Ejemplos

La siguiente instrucción de entrada, obtiene la edad de una persona desde un dispositivo de entrada, como el teclado, y lo almacena en la variable llamada edad:

```
LEER (edad);
```

La siguiente instrucción de entrada, obtiene tres números desde un dispositivo de entrada, como el teclado, y los almacena en las variables num1, num2 y num3:

```
LEER (num1, num2, num3);
```

La siguiente instrucción de salida, muestra el contenido de la variable sueldoNeto en un dispositivo de salida como la pantalla:

```
IMPRIMIR sueldoNeto
```

La siguiente instrucción de salida, muestra el contenido de las variables montoPag y montoDes en un dispositivo de salida como la pantalla:

```
IMPRIMIR montoPag, montoDes
```

Ejercicios Resueltos

Ejercicio 1

Algoritmo para sumar dos números

Análisis del Problema

Entradas : valor1 (Variable A)
valor2 (Variable B)

Salidas : La suma del valor1 más valor 2 (Variable Result)

Variables: A, B, Result de tipos de datos enteros

ALGORITMO SUMA;

Variables

A, B, Result $\mathbb{N}$ ENTEROS;

INICIO

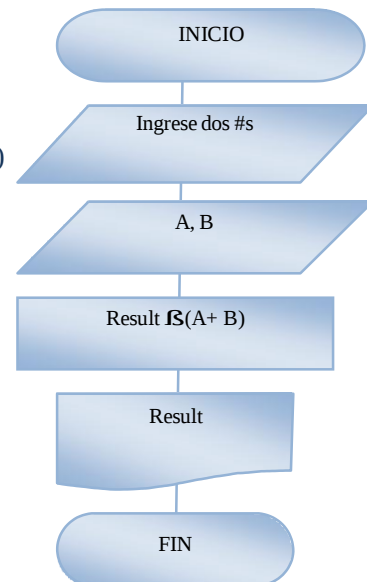
Escribir (“ Ingrese dos #s “);

Leer (A,B);

Result $\mathbb{N}$ (A+ B);

Escribir (Result) ;

FIN-ALG



Prueba de Escritorio

A	B	RESULT
5	6	11

SALIDA

Ejercicio 2

Realizar un algoritmo y Diagrama de Flujo para calcular el área de un rectángulo cuyos datos base y altura se leen desde teclado y el resultado se visualizara en la pantalla.

Algoritmo Rectángulo;

Variables

area , base, altura $\rightarrow$ enteros;

Inicio

Escribir (“ Ingrese la base y altura de una rectángulo”)

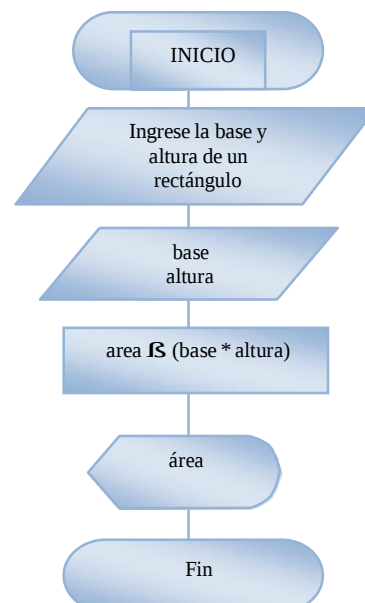
Leer (base);

Leer (altura);

area $\rightarrow$ (base * altura)

Imprimir(área);

Fin_Alg

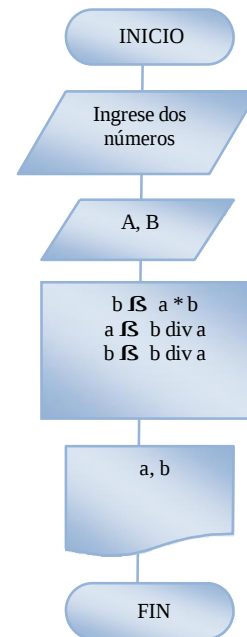


Ejercicio 3

Algoritmo para intercambiar los valores almacenados en a,b sin utilizar otra variable

```

Algoritmo Intercambio;
Variables
a,b  $\leftarrow$  entero;
Inicio
Escribir(' Ingrese dos números ');
Leer (a, b);
b  $\leftarrow$  a * b;
a  $\leftarrow$  b div a;
b  $\leftarrow$  b div a;
Imprimir ( a, b );
Fin_Alg
    
```

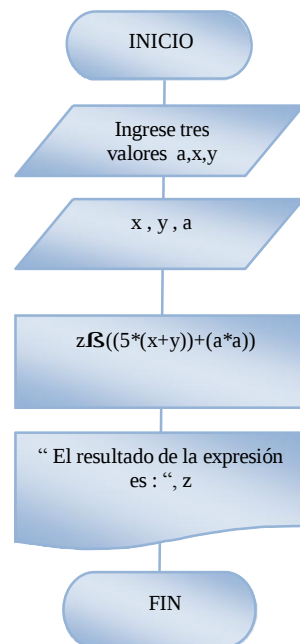


Ejercicio 4

Convertir la siguiente expresión aritmética a algoritmos: $\text{Expr} = 5 * (x + y) + a^2$

```

Algoritmo Expresión
Variables
z,x,y,a  $\leftarrow$  enteros;
Inicio
Escribir(" Ingrese tres valores de a,x,y ");
Leer ( x,y,a);
z  $\leftarrow$  ((5*(x+y)) + (a*a) );
Imprimir( " El resultado de la expresión es: ", z);
Fin-Alg
    
```



Ejercicio 5

Suponga que un individuo desea invertir su capital en un banco y desea saber cuanto dinero ganara después de un mes si el banco paga a razón de 2% mensual.

```

Algoritmo inversion;
Variables
Cap_inv, gan  $\leftarrow$  Real;
Inicio
Escribir("Ingrese su capital a invertir");
Leer ( cap_inv );
gan  $\leftarrow$  ( cap_inv * 0.02);
Imprimir (gan );
Fin

```

Ejercicio 6

Un alumno desea saber cual será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:

55% del promedio de sus tres calificaciones parciales.

30% de la calificación del examen final.

15% de la calificación de un trabajo final.

```

Algoritmo Calificacion;
Variables
Calif1, calif2, calif3  $\leftarrow$  entero;
prom, examenf, trabajof, promparcial, promexamenf, promtrabajof, califinal  $\leftarrow$  real;
Inicio
Escribir("Ingrese 3 calificaciones parciales, examen final y trabajo final");
Leer (Calif1, calif2, calif3, examenf, trabajof);
prom  $\leftarrow$  (Calif1, calif2, calif3)/3
promparcial  $\leftarrow$  (prom * 0.55);
promexamenf  $\leftarrow$  (examenf * 0.30);
promtrabajof  $\leftarrow$  (trabajof * 0.15);
califinal  $\leftarrow$  (promparcial + promexamenf + promtrabajof);
Imprimir (califinal );
Fin

```

Ejercicio 7

Un vendedor recibe un sueldo base mas un 10% extra por comisión de sus ventas, el vendedor desea saber cuanto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

```

Algoritmo inversion;
Variables
sueldob, ventas1, ventas2, ventas3, tot_ventas, comision, totalpago  $\leftarrow$  Real;
Inicio
Escribir("Ingrese su sueldo base y sus comisiones extras");
Leer (sueldob, ventas1, ventas2, ventas3);
tot_ventas  $\leftarrow$  ( ventas1 + ventas2 + ventas3 );

```

```
comision  $\mathbb{R}$  (tot_ventas * 0.10);
totalpago  $\mathbb{R}$  ( sueldob + comision )
Imprimir ( totalpago, commission );
Fin
```

Ejercicio 8

Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuanto deberá pagar finalmente por su compra.

```
Algoritmo tienda;
Variables
    tcompra, desc, tpago  $\mathbb{R}$  real;
Inicio
    Escribir("Ingrese la cantidad total de la compra ");
    Leer ( tcompra);
    desc  $\mathbb{R}$ ( tcompra * 0.15);
    tpago $\mathbb{R}$ ( tcompra – desc );
    Imprimir ( tpago);
Fin
```

Ejercicios Propuestos

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagramas de flujo aplicando las fases de resolución de un Algoritmo

1. Escribir un algoritmo que calcule y escriba el cuadrado de 25
2. Dada un cantidad en pesos, obtener la equivalencia en dólares, asumiendo que la unidad cambiaría es un dato desconocido.
3. Calcular el numero de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, si la formula es: $\text{num. pulsaciones} = (220 - \text{edad})/10$
4. Calcular el nuevo salario de un obrero si obtuvo un incremento del 25% sobre su salario anterior.
5. En un hospital existen tres áreas: Ginecología, Pediatría, Traumatología. El presupuesto anual del hospital se reparte conforme a la sig. tabla:

Área	Porcentaje del presupuesto
Ginecología	40%
Traumatología	30%
Pediatría	30%

6. Obtener la cantidad de dinero que recibirá cada área, para cualquier monto presupuestal.
7. El dueño de una tienda compra un artículo a un precio determinado. Obtener el precio en que lo debe vender para obtener una ganancia del 30%.

8. Todos los lunes, miércoles y viernes, una persona corre la misma ruta y cronometra los tiempos obtenidos. Determinar el tiempo promedio que la persona tarda en recorrer la ruta en una semana cualquiera.
9. Un alumno desea saber cual será su promedio general en las tres materias mas difíciles que cursa y cual será el promedio que obtendrá en cada una de ellas. Estas materias se evalúan como se muestra a continuación:

La calificación de Matemáticas se obtiene de la sig. manera:

Examen 90%

Promedio de tareas 10%

En esta materia se pidió un total de tres tareas.

La calificación de Física se obtiene de la sig. manera:

Examen 80%

Promedio de tareas 20%

En esta materia se pidió un total de dos tareas.

La calificación de Química se obtiene de la sig. manera:

Examen 85%

Promedio de tareas 15%

En esta materia se pidió un promedio de tres tareas.

3.3. Sentencias de Decision o Condicionales

Son aquellas que permiten decidir algo dependiendo del valor que asuma, a lo cual deberá ejecutarse una o varias acciones relacionadas con ese valor.

Las estructuras condicionales comparan una variable contra otro(s) valor(es), para que en base al resultado de esta comparación, se siga un curso de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Existen dos tipos básicos, las simples y las múltiples.

Las estructuras selectivas o de selección se clasifican en:

- § Estructura de selección simple (SI).
- § Estructura de selección doble (SI - SINO).
- § Estructura de selección múltiple (EN CASO - SEA)

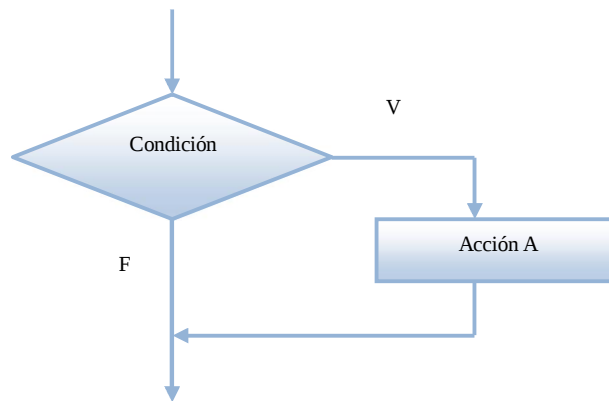
Estructura de Selección Simple SI

En la estructura de selección simple SI, evalúa una condición lógica y:

- § Si la condición es verdadera se ejecuta la acción A. La acción A puede ser una instrucción simple (una sola instrucción) o una instrucción compuesta (un conjunto de instrucción).

- § Si la condición es falsa, no se hace nada.
- § La condición puede ser elaborada mediante la utilización de operadores de relación así como operadores lógicos o booleanos

FLUJO DE CONTROL DE UNA SENTENCIA SI-ENTONCES



Nota:

En el caso de instrucciones compuestas, estas serán delimitadas por inicio y fin.

Pseudocódigo (acción simple)

SI(condición) entonces
instrucción A

Pseudocódigo (acción compuesta)

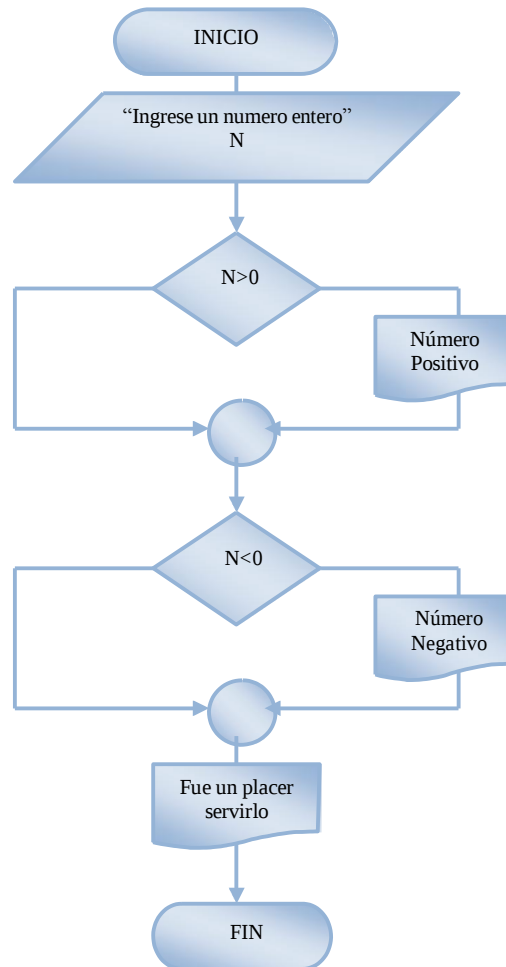
SI(condición) entonces
 inicio
 instrucción A1
 instrucción A2
 .
 .
 .
 instrucción An
fin_si

Ejemplo

Algoritmo para determinar si un número entero ingresado por teclado es positivo o negativo e imprimir el mensaje "Fue un placer Servirlo".

```

Algoritmo Números;
Variables
N  $\rightarrow$  Entero ;
Inicio
Escribir("Ingrese un número entero")
Leer( n );
Si n>0 Entonces
    Imprimir( ' Numero positivo ' );
Si n<0 Entonces
    Imprimir(' Numero negativo');
Imprimir(' Fue un placer servirlo');
Fin.
    
```



Estructura de Selección Doble SI - SINO

La estructura de selección doble SI - SINO evalúa una condición lógica y:

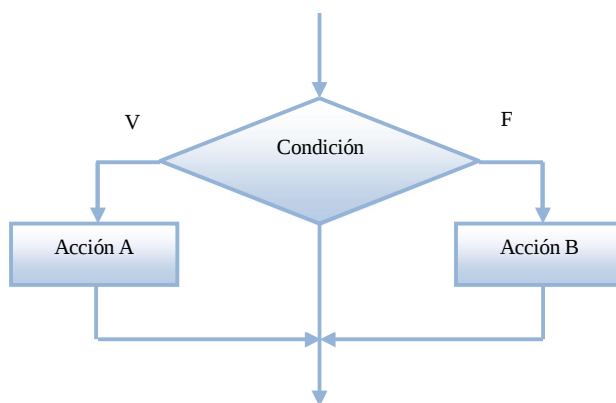
- § Si la condición es verdadera, ejecuta la acción A.
- § Si la condición es falsa, ejecuta la acción B.

Tanto la acción A como la acción B pueden ser acciones simples (una sola acción) o acciones compuestas (un conjunto de acciones).

Nota:

En el caso de acciones compuestas, estas serán delimitadas por inicio y fin.

Diagrama de Flujo Selección Doble



Pseudocódigo (acciones simples)

```

SI( condición ) entonces
    acción A
caso contrario
    acción B
  
```

Pseudocódigo (acciones compuestas)

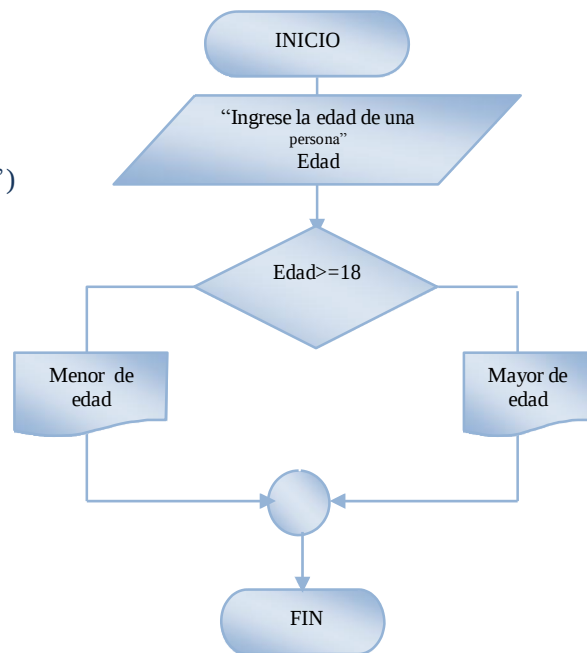
```

SI( condición ) entonces
    inicio
        acción A1
        acción A2
        .
        .
        .
        acción An
    fin_si
CASO CONTRARIO
    inicio
        acción B1
        acción B2
        .
        .
        .
        acción Bn
    fin_caso_contrario
  
```

Por ejemplo, si se desea saber si una persona es mayor o menor de edad, podemos escribir:

```

Algoritmo Edades;
Variables
edad  $\rightarrow$  Entero ;
Inicio
Escribir("Ingrese la edad de una persona")
Leer( edad );
SI( edad >= 18 ) entonces
    IMPRIMIR ("Mayor de edad")
CASO CONTRARIO
    IMPRIMIR ("Menor de edad");
Fin_si
Fin_alg.
    
```

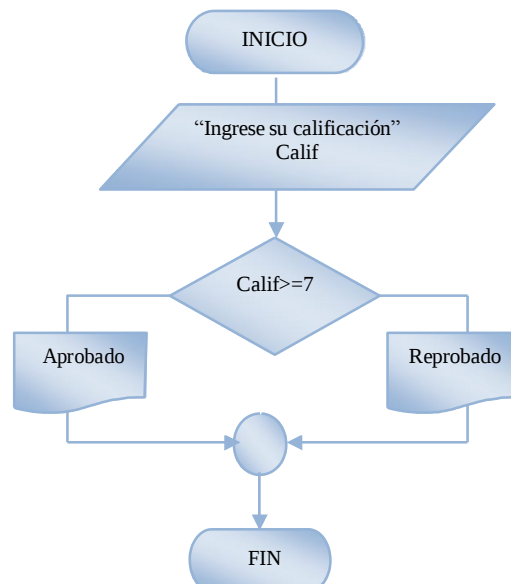


Ejemplo

Un estudiante desea saber si ha aprobado la asignatura de literatura

```

Algoritmo Estudiante;
Variables
calif  $\rightarrow$  entero;
Inicio
Escribir("Ingrese su calificación ");
Leer( calif );
Si calif >= 7 Entonces
    Escribir('Aprobado ')
CASO CONTRARIO
    Escribir('Reprobado ');
Fin_si
Fin_alg.
    
```



3.4. Sentencias de Selección Anidadas SI

Se dice que una estructura SI (o SI-SINO) esta anidada cuando esta contenida dentro de otra estructura SI o dentro de otra estructura SI-SINO. No existe límite en cuanto al nivel de anidamiento.

Por ejemplo, una estructura SI con tres niveles de anidamiento tendría el siguiente formato:

SI(condición C1) entonces

```

{
    Inicio
    acción A1
    SI( condición C2 ) entonces
    {
        inicio
        acción A2
        condición C3 ) entonces
        {
            acción A3
            fin_si(c2)
        }
    }
    Fin_si(c1).
}
    
```

En general, el anidamiento podría evitarse usando el operador lógico "y". Así, la anterior selección SI anidada puede descomponerse en tres estructuras de selección simple no anidadas consecutivas, como se muestra a continuación.

```

SI( condición C1 )
    acción A1
SI( condición C1 y condición C2 )
    acción A2
SI( condición C1 y condición C2 y condición C3 )
    acción A3
    
```

Ejemplo

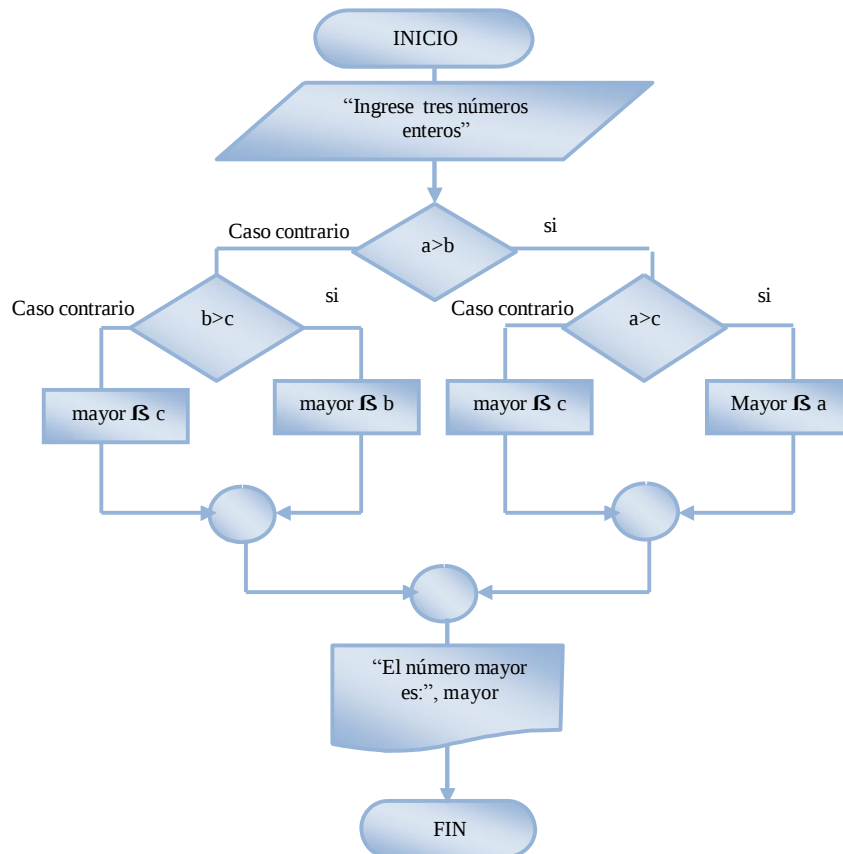
Dados tres números enteros. Determinar cuál es el mayor.

```

Algoritmo mayor;
Variables
    mayor, a, b, c ES enteros;
Inicio
    Escribir ('Ingrese tres números enteros ');
    Leer (a, b, c );
    SI a > b ENTONCES
        inicio
        Si a > c Entonces
            Mayor ES a
        Caso Contrario
            mayor ES c
        Fin_Si
    CASO CONTARIO
        inicio
    
```

```

Si b > c Entonces
    mayor ← b
Caso Contrario
    mayor ← c;
Fin_caso_contrario.
Escribir ('El número mayor es:', mayor);
Fin_Si
    
```



Estructura de Selección Múltiple

La estructura de selección múltiple SEGUN permite elegir una la elección de o más acciones acción de entre una lista de varias opciones, usando para ello una variable denominada selector. El selector se compara con una lista de constantes enteras o de carácter C1, C2, ..., Cn para cada una de las cuales hay una acción A1, A2, ..., An y:

- § Si el selector coincide con una constante de la lista, se ejecuta la acción correspondiente a dicha constante.
- § Si el selector no coincide con ninguna constante de la lista, se ejecuta la acción Df correspondiente al CASO CONTRARIO, si es que existe.

Las acciones A1, A2, A3, ..., An pueden ser acciones simples(una sola acción) o acciones compuestas (un conjunto de acciones).

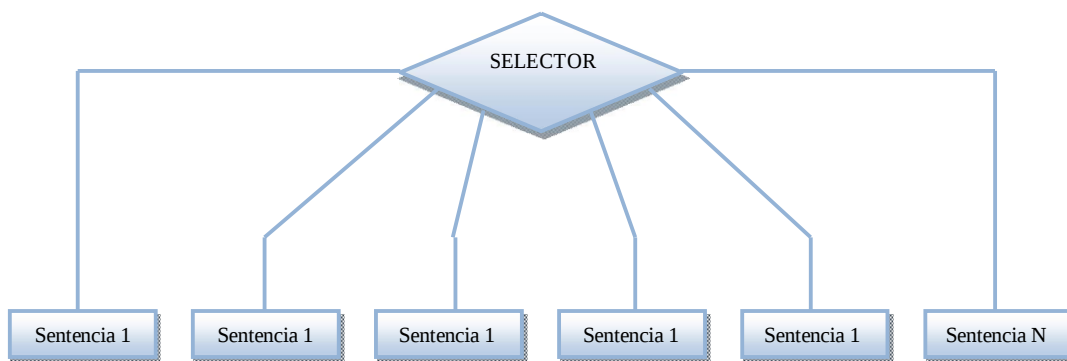
En la Figura se muestra el pseudocódigo de la estructura de selección múltiple.

Nota:

En el caso de acciones compuestas, estas no necesitan estar encerradas entre llaves.

```

SEGUN ( selector )HACER
  INICIO
    CASO C1 : acción A1
    CASO C2 : acción A2
    CASO C3 : acción A3
    CASO Cn : acción An
    CASO CONTARIO
      acción Df
  FIN_SEGUN_SEA
  
```



Ejemplo

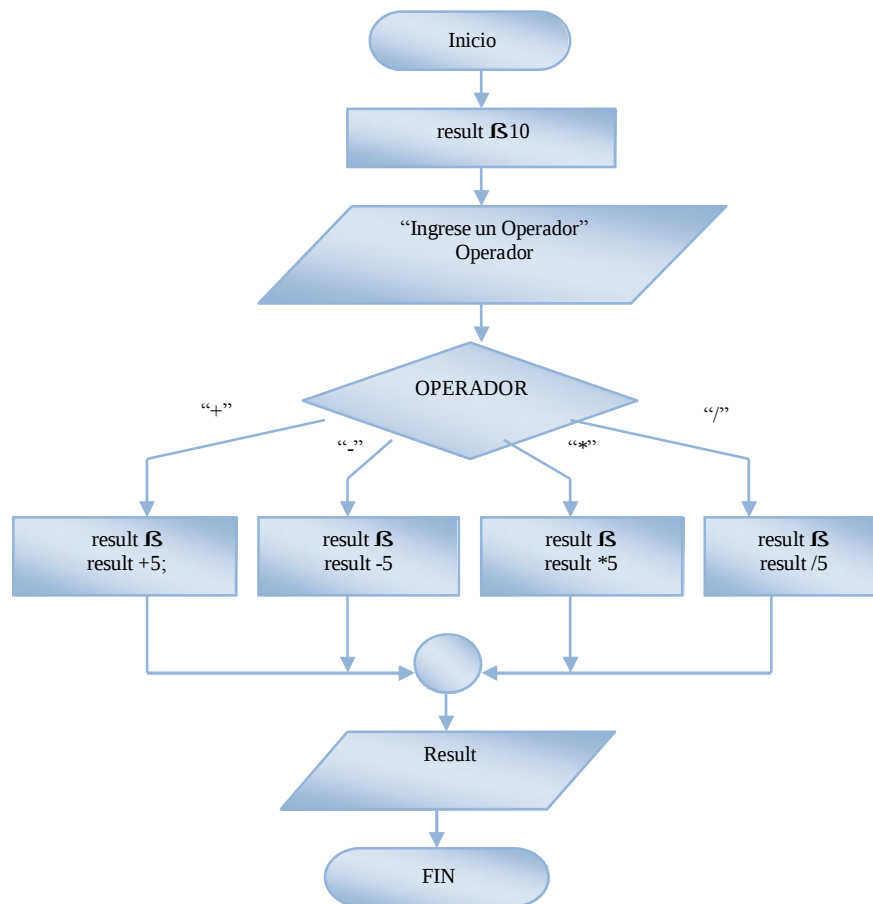
Elabore un Algoritmo que en la variable result=10 realicen las respectivas operaciones. El algoritmo debe leer un carácter y tomar una de las siguientes determinaciones:

- Si es '+' debe sumar la variable result mas 5
- Si es '-' debe restar la variable result menos 5
- Si es '*' debe multiplicar la variable result por 5
- Si es '/' debe dividir la variable para 5

```

Algoritmo Determinacion;
Variables
  Operador ⚡ Carácter;
  result ⚡ Entero;
Inicio
  result ⚡ 10;
  Escribir(' Ingrese un Operador ');
  Leer( operador );
  SEGÚN SEA operador HACER
    '+': result ⚡ result + 5;
    '-': result ⚡ result - 5;
    '*': result ⚡ result * 5;
    '/': result ⚡ result / 5 ;
  
```

```
Fin_Caso;
ESCRIBIR ( result );
Fin_Alg
```



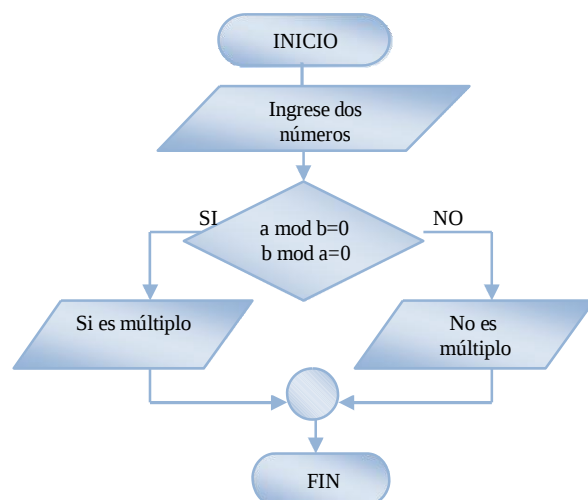
Ejercicios Resueltos

Ejercicio 1

Realizar un algoritmo para determinar si un número ingresado por teclado es múltiplo de otro.

```

Algoritmo múltiplo;
Variables
a, b ∈ entero;
Inicio
Escribir ('Ingrese dos números enteros ');
Leer( a,b);
Si (( a mod b )=0 ) ó (( b mod a)=0)
Entonces
    Escribir (' Si es múltiplo ')
Caso Contario
    Escribir ('No es múltiplo ');
Fin_Si.
Fin_Alg
  
```

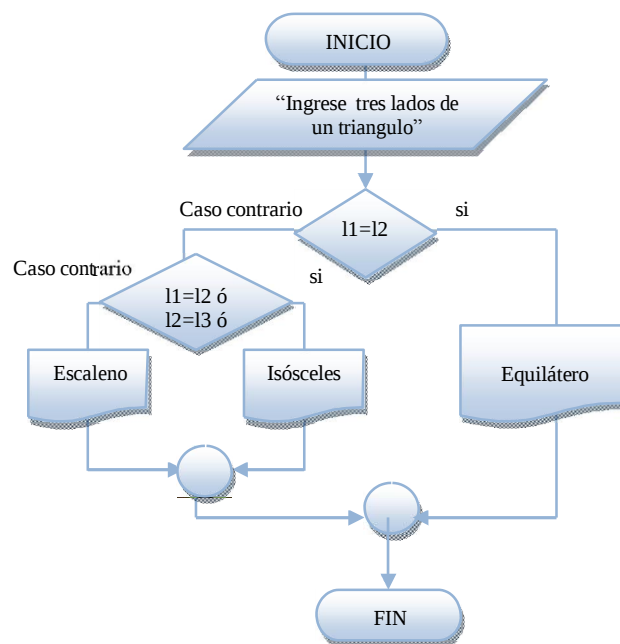


Ejercicio 2

Hacer un algoritmo que pida los tres lados de un triángulo, y diga si el Triángulo es Equilátero, Isósceles ó Escaleno.

```

Algoritmo triángulo;
Variables
L1,L2,L3 ∈ ENTEROS;
Inicio
Escribir(“ Ingrese tres lados de un triángulo “ );
Leer (L1,L2,L3);
Si L1=L2 y L2=L3 Entonces
    Escriba ( “ Equilátero”)
Caso Contrario
    Si L1=L2 o L2=L3 o L1=L3 Entonces
        Escriba(“ Isósceles”)
    Caso Contrario
        Escribir(“ Escaleno”)
    Fin_Si
Fin_Si
Fin_Algoritmo
    
```



Ejercicio 3

Diseñe un algoritmo que determine si un número es negativo, positivo o cero.

```

Algoritmo ( Formato 1 )
INICIO
REAL n
LEER n
SI( n > 0 )
    IMPRIMIR "Positivo"
    
```

```

CASO CONTRARIO
SI( n < 0 )
IMPRIMIR "Negativo"
CASO CONTRARIO
IMPRIMIR "Cero"
FIN

```

```

Algoritmo ( Formato 2 )
INICIO
REAL n
LEER n
SI( n > 0 )
    IMPRIMIR "Positivo"
CASO CONTRARIO
    SI( n < 0 )
        IMPRIMIR "Negativo"
CASO CONTRARIO
    IMPRIMIR "Cero"
FIN

```

Ejercicio 4

Unas camisas se venden a \$ 10 dólares cada una si se compran más de tres, y a \$ 12 en los demás casos. Escriba un algoritmo que lea un número de entrada de camisas compradas e imprima el costo Total.

```

Algoritmo Camisas ;
Variables
Cantidad IS Entero;
Costo IS Real;
Inicio
Escribir(" Ingrese El Número De Camisas Compradas);
Leer(Cantidad);
Si Cantidad > 3 Then
    Costo IS Cantidad*10;
Caso Contrario
    Costo IS Cantidad*12;
Fin_Si
Escribir(Cantidad, ' Camisas Cuestan $ ', Costo);
Fin_Alg.

```

Ejercicio 5

Escriba un programa que reciba el peso de una carta en onzas como entrada e imprima el costo del porte calculado según la regla siguiente:

La primera onza cuesta \$.25 (dólares)
Cada onza adicional cuesta \$.04.

```

Algoritmo onza;
Variables
Onzas IS Entero;
Precio IS Real;
Inicio
Escribir(" ingrese el pezo en onzas");
Leer(onzas);
Si (onzas= 1) entonces

```

```

    precio  $\leftarrow$  (onzas*25);
Caso contrario
    precio  $\leftarrow$  (((onzas-1)*0.4)+ 25);
Fin_si
Escribir (“ El precio total es ;”, precio);
    Fin_Alg.

```

Ejercicio 6

En la Ponduka State University, los veteranos sólo pagan \$30 (dólares) por asignatura mientras que los demás (regulares) pagan \$ 50 por asignatura . Escriba un algoritmo en el que el usuario introduce los datos del estudiante (Vet o Reg) y el número de asignaturas . La salida debe indicar si el estudiante es de la categoría veterano o regular e indicar el número de materias y los costos de la colegiatura.

```

Algoritmo Universidad;
Variables
a,b,c  $\leftarrow$  caracter;
mat,pag  $\leftarrow$  Entero;
Inicio
Escribir('INGRESE LOS DATOS DEL ESTUDIANTE (VET(v) o REG(r))');
Leer(c);
Escribir('INGRESE EL NUMERO DE MATERIAS');
Leer(mat);
Si (c = 'v') Entonces
    Pag  $\leftarrow$  mat*30;
    Escribir(' Tipo estudiante : Veterano');
    Escribir(' Número de materias:', mat);
    Escribir('PAGO DE COLEGIATURA:',PAG);
Fin_Si
Si (c= 'r')Entonces
    Pag  $\leftarrow$  mat*50;
    Escribir('Tipo estudiante: Regular');
    Escribir('Número de materias:', mat);
    Escribir('PAGO DE COLEGIATURA:',PAG);
Fin_Si
Fin_Alg

```

Ejercicio 7

Diseñe un algoritmo que lea tres longitudes y determine si forman o no un triángulo. Si es un triángulo determine de que tipo de triángulo se trata entre: equilátero (si tiene tres lados iguales), isósceles (si tiene dos lados iguales) o escaleno (si tiene tres lados desiguales). Considere que para formar un triángulo se requiere que: "el lado mayor sea menor que la suma de los otros dos lados.

```

Algoritmo longitudes;
Variables
L1, L2, L3, suma, mayor  $\leftarrow$  Enteros;
INICIO
Escribir (“ Ingrese las longitudes de un triangulo”);
LEER( L1, L2, L3);
mayor = L1 // Determina el lado mayor
SI( L2 > mayor ) entonces
    mayor  $\leftarrow$  L2;

```

```

SI( L3 > mayor ) entonces
    mayor ← L3;
suma ← ( L1+L2+L3-mayor); // Determina la suma de los lados a excepción del lado mayor
SI( mayor < suma ) entonces // Determina de que tipo de triángulo se trata
    Inicio
        SI( ( L1 = L2 ) y ( L2 = L3 ) ) entonces
            IMPRIMIR ("Triángulo equilátero")
        caso contrario
            Si ( ( L1 = L2 ) o ( L1 = L3 ) o ( L2 = L3 ) ) entonces
                IMPRIMIR ("Triángulo isósceles")
            Caso contrario
                IMPRIMIR ("Triángulo escaleno");
        Fin_si
    Caso contrario
        IMPRIMIR ("No es un triángulo");
FIN.

```

Ejercicio 8

Diseñe un algoritmo que lea tres números y los imprima de mayor a menor y de menor a mayor

```

Algoritmo orden;
Variables
n1, n2, n3, mayor, menor, medio ← enteros;
INICIO
    Escribir(" Ingrese tres números");
    LEER (n1, n2, n3);
    menor ← n1 // Determina el menor
    SI( n2 < menor ) entonces
        menor ← n2
    SI( n3 < menor ) entonces
        menor ← n3;
    mayor ← n1 // Determina el mayor
    SI( n2 > mayor ) entonces
        mayor ← n2;
    SI( n3 > mayor ) entonces
        mayor ← n3;
    medio = (n1+n2+n3-mayor-menor); // Determina el número medio
    IMPRIMIR (menor, medio, mayor); // Imprime en orden ascendente
    IMPRIMIR (mayor, medio, menor); // Imprime en orden descendente
FIN

```

Ejercicio 9

Imprimir el valor de los números romanos.

```

Algoritmo Num_Rom;
VARIABLES
Ch ← Caracter;
INICIO
    Leer(ch);
    CASO QUE Ch SEA
        'I' : Escribir(' Numero Uno');

```

```
'V': Escribir(' Numero Cinco');
'X' : Escribir(' Numero Diez' );
'L' : Escribir(' Numero cincuenta');
'C' : Escribir(' Numero Cien' );
'D' : Escribir(' Numero 500' );
'M' : Escribir(' Numero Mil' )
Caso contrario
  Escribir(' No es numero Romano' );
Fin_Caso
Fin_Alg
```

Ejercicio 10

Escribir una sentencia CASE que dado un selector entero llamado nombre, imprima el primer nombre si nombre =1, el primer Apellido si Nombre =2 y el segundo Apellido si Nombre es igual =3

```
Algoritmo Nombre;
variables
nombre ISEntero;
Inicio
Leer (nombre );
CASO QUE nombre SEA
  1 : Escribir('Mary');
  2 : Escribir('Lynn');
  3 : Escribir('Smith ');
Fin_Caso;
Fin_Alg.
```

Ejercicios Propuestos

Ejercicios Individuales

Desarrolle los siguientes problemas de sentencias Condicionales a un lenguaje algorítmico con su respectiva prueba escritorio y diagrama de Flujo

1. Dado tres variables enteras num1,num2,num3, encuentre el mayor entre estos tres números y almacene en una variable entera llamada max.
2. Escriba un Algoritmo que lea dos números enteros como entrada y escriba el mensaje " Signos Opuestos" solo si uno de los enteros es positivo y el otro negativo.
3. Escriba un Algoritmo que reciba como entrada dos enteros positivos distintos y escriba la diferencia entre el número mayor y el menor, asegúrese de que su programa escriba 6 tanto cuando la entrada es 9 15 como cuando la entrada es 15 9
4. Elaborar un Algoritmo para invertir una cifra almacenada en una variable A de tal manera si ingresa 834 debe darle como salida 438 , el dato ingresado debe estar en un rango de 1 y 999 .

5. Dados tres números ingresados por teclado, elaborar un Algoritmo para almacenar en una sola variable los tres números e imprimirlos a pantalla.
6. En una Universidad Estatal, los cargos por colegiatura son de \$50 (dólares) por materia, con un cargo máximo de \$750 independientemente del número de asignaturas tomadas. Así un estudiante que cursa 12 materias pagaría \$600, mientras que toma 21 pagaría el cargo máximo de \$750. Escriba un Algoritmo en el que la entrada es el número de materias y la colegiatura es la salida.
7. Elaborar un Algoritmo que calcule su edad actual en (dd-mm-aa).
8. Suponiendo que se ingresa una vocal por teclado, realice un algoritmo para determinar si es abierta o cerrada.
9. Se desea ingresar un número por teclado correspondiente a una calificación se desea saber su equivalencia:
19 – 20 : Sobresaliente
16 - 18 : Muy Buena
14 – 15 : Buena
12 – 13 : Regular
11 ó menos : Insuficiente
10. Elaborar un programa utilizando un menú :
N: Nombre del cliente
D: Dirección
T: Teléfono
C: Ciudad
F: Fin
11. El Algoritmo debe permitir elegir una opción, la cual se debe asignar a una variable de tipo carácter. El usuario puede introducir la letra mayúscula o minúscula.
12. Elabore un Algoritmo que lea dos variables enteras a y b. El programa debe leer un carácter y tomar una de las siguientes determinaciones:
Si es '+' debe sumar las dos variables
Si es '-' debe restar las dos variables
Si es '*' debe multiplicar las dos variables
Si es '/' debe dividir las dos variables
Si es '%' debe obtener el residuo de la división.

Ejercicios Grupales

Desarrolle los siguientes problemas de sentencias Condicionales a un lenguaje algorítmico y a un diagrama de flujo con su respectiva prueba escritorio

1. Calcular el total que una persona debe pagar en un llantera, si el precio de cada llanta es de \$800 si se compran menos de 5 llantas y de \$700 si se compran 5 o mas.
2. En un supermercado se hace una promoción, mediante la cual el cliente obtiene un descuento dependiendo de un numero que se escoge al azar. Si el numero escogido es menor que 74 el descuento es del 15% sobre el total

de la compra, si es mayor o igual a 74 el descuento es del 20%. Obtener cuanto dinero se le descuenta.

3. Calcular el número de pulsaciones que debe tener una persona por cada 10 segundos de ejercicio aeróbico; la fórmula que se aplica cuando el sexo es femenino es:

$$\text{num. pulsaciones} = (220 - \text{edad})/10$$

y si el sexo es masculino:

$$\text{num. pulsaciones} = (210 - \text{edad})/10$$

4. En una escuela la colegiatura de los alumnos se determina según el número de materias que cursan. El costo de todas las materias es el mismo.

Se ha establecido un programa para estimular a los alumnos, el cual consiste en lo siguiente: si el promedio obtenido por un alumno en el último periodo es mayor o igual que 9, se le hará un descuento del 30% sobre la colegiatura y no se le cobrará IVA; si el promedio obtenido es menor que 9 deberá pagar la colegiatura completa, la cual incluye el 10% de IVA. Obtener cuanto debe pagar un alumno.

5. El gobierno ha establecido el programa SAR (Sistema de Ahorro para el Retiro) que consiste en que los dueños de la empresa deben obligatoriamente depositar en una cuenta bancaria un porcentaje del salario de los trabajadores; adicionalmente los trabajadores pueden solicitar a la empresa que deposite directamente una cuota fija o un porcentaje de su salario en la cuenta del SAR, la cual le será descontada de su pago. Un trabajador que ha decidido aportar a su cuenta del SAR desea saber la cantidad total de dinero que estará depositado a esa cuenta cada mes, y el pago mensual que recibirá.

El gobierno del estado de México desea reforestar un bosque que mide determinado número de hectáreas. Si la superficie del terreno excede a 1 millón de metros cuadrados, entonces decidirá sembrar de la sig. manera:

Porcentaje de la superficie del bosque	Tipo de árbol
70%	pino
20%	oyamel
10%	cedro

Si la superficie del terreno es menor o igual a un millón de metros cuadrados, entonces decidirá sembrar de la sig. manera:

Porcentaje de la superficie del bosque	Tipo de árbol
50%	pino
30%	oyamel
20%	cedro

6. El gobierno desea saber el número de pinos, oyameles y cedros que tendrá que sembrar en el bosque, si se sabe que en 10 metros cuadrados caben 8 pinos, en 15 metros cuadrados caben 15 oyameles y en 18 metros

cuadrados caben 10 cedros. También se sabe que una hectárea equivale a 10 mil metros cuadrados.

7. Una fábrica ha sido sometida a un programa de control de contaminación para lo cual se efectúa una revisión de los puntos IMECA generados por la fábrica. El programa de control de contaminación consiste en medir los puntos IMECA que emite la fábrica en cinco días de una semana y si el promedio es superior a los 170 puntos entonces tendrá la sanción de parar su producción por una semana y una multa del 50% de las ganancias diarias cuando no se detiene la producción. Si el promedio obtenido de puntos IMECA es de 170 o menor entonces no tendrá ni sanción ni multa. El dueño de la fábrica desea saber cuanto dinero perderá después de ser sometido a la revisión.

8. En un juego de preguntas a las que se responde “Si” o “No” gana quien responda correctamente las tres preguntas. Si se responde mal a cualquiera de ellas ya no se pregunta la siguiente y termina el juego. Las preguntas son:

Colon descubrió América?

La independencia de México fue en el año 1810?

The Doors fue un grupo de rock Americano?

9. Una frutería ofrece las manzanas con descuento según la siguiente tabla:

NUM. DE KILOS COMPRADOS	% DESCUENTO
0 - 2	0%
2.01 - 5	10%
5.01 - 10	15%
10.01 en adelante	20%

Determinar cuanto pagará una persona que compre manzanas en esa frutería.

10. Una institución educativa estableció un programa para estimular a los alumnos con buen rendimiento académico y que consiste en lo siguiente:

- § Si el promedio es de 9.5 o mas y el alumno es de preparatoria, entonces este podrá cursar 55 unidades y se le hará un 25% de descuento.
- § Si el promedio es mayor o igual a 9 pero menor que 9.5 y el alumno es de preparatoria, entonces este podrá cursar 50 unidades y se le hará un 10% de descuento.
- § Si el promedio es mayor que 7 y menor que 9 y el alumno es de preparatoria, este podrá cursar 50 unidades y no tendrá ningún descuento.
- § Si el promedio es de 7 o menor, el numero de materias reprobadas es de 0 a 3 y el alumno es de preparatoria, entonces podrá cursar 45 unidades y no tendrá descuento.

- § Si el promedio es de 7 o menor, el numero de materias reprobadas es de 4 o mas y el alumno es de preparatoria, entonces podrá cursar 40 unidades y no tendrá ningún descuento.
- § Si el promedio es mayor o igual a 9.5 y el alumno es de profesional, entonces podrá cursar 55 unidades y se le hará un 20% de descuento.
- § Si el promedio es menor de 9.5 y el alumno es de profesional, entonces podrá cursar 55 unidades y no tendrá descuento.

Obtener el total que tendrá que pagar un alumno si la colegiatura para alumnos de profesional es de \$300 por cada cinco unidades y para alumnos de preparatoria es de \$180 por cada cinco unidades.

11. Usando la sentencia Caso que , elabore un Algoritmo que lea un carácter y escriba lo siguiente:
 - § Si el carácter representa un número, el mensaje ' número '.
 - § Si el carácter es una letra de la 'a' a la 'z' letra minúscula.
 - § Si el carácter es una letra de la 'A' a la 'Z' letra mayúscula.
 - § De otro modo escriba 'Carácter no válido '.
12. Usando la sentencia CASO QUE elabore un Algoritmo que lea un carácter numérico y dependiendo del carácter debe realizar lo siguiente:
 - § Si el carácter es '5' debe escribir el número 5
 - § Si el carácter está comprendido entre '6' y '9 ' es una variable se debe almacenar su valor numérico más el sucesor del valor numérico.
 - § Si el carácter está comprendido entre '6 ' y '9 ' es una variable se debe almacenar su valor numérico menos el anterior del valor numérico.

3.5. Sentencias de Bucles o Lazos Repetitivos.

Se llaman problemas repetitivos o cíclicos a aquellos en cuya solución es necesario utilizar un mismo conjunto de acciones que se puedan ejecutar una cantidad específica de veces. Esta cantidad puede ser fija (previamente determinada por el programador) o puede ser variable (estar en función de algún dato dentro del programa). Los ciclos se clasifican en:

- § PARA / DESDE/ HASTA / HAGA.
- § MIENTRAS/ HACER.
- § REPETIR / HASTA

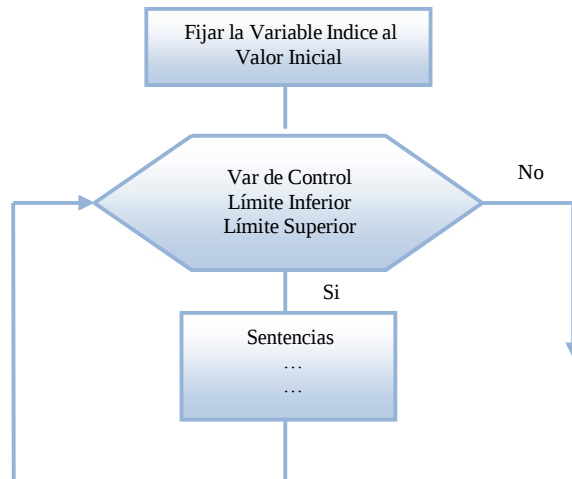
Sentencia Para / Desde / Hasta / Haga

El cuerpo de un ciclo PARA es la línea o grupos de líneas que se van a ejecutar un número específico de veces. Y DE MODO AUTOMÁTICO CONTROLA EL NÚMERO DE ITERACIONES O PASOS A TRAVÉS DEL CUERPO DEL BUCLE

PARA Variable **ÍS** Valor Inicial HASTA valor Final HAGA

```
Sentencia1;
Sentencia2;
.
.
SentenciaN;

Fin_Para
```

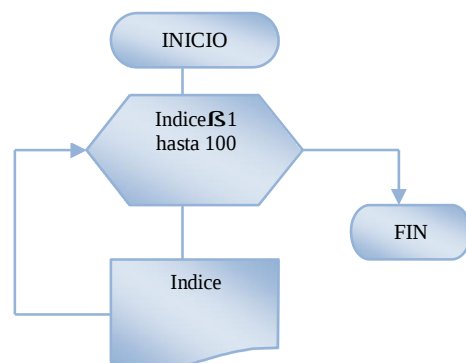


Al ejecutarse una sentencia PARA por primera vez, el valor inicial se asigna a variable que se denomina variable de control, y a continuación se ejecuta la sentencia del interior del bucle hasta que se cumpla el valor del límite superior.

Ejemplo

Escriba un Algoritmo para imprimir los números del 1 al 100

```
Algoritmo suma;
Variables
Indice ÍSEntero;
Inicio
Para ir desde Indice ÍS1 hasta 100 hacer
    Escribir(Indice);
Fin_Para
Fin_Alg.
```

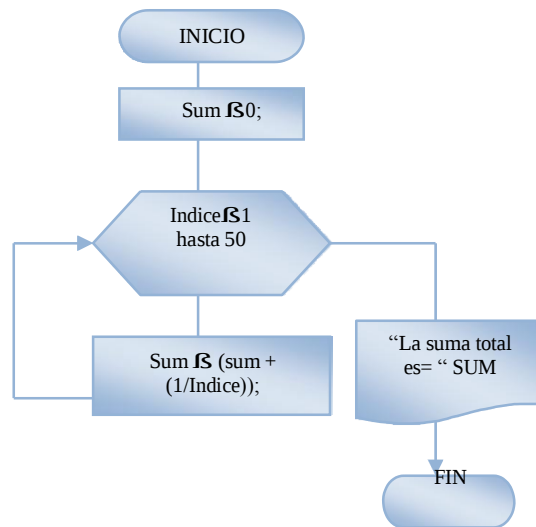


Ejemplo

Escriba un Algoritmo para calcular la suma de $1 + 1/2 + \dots + 1/50$

```

Algoritmo suma;
Variables
  Indice  $\mathbb{N}$  Entero;
  Sum  $\mathbb{R}$  Real;
Inicio
  Sum  $\mathbb{N}$  0;
Para ir desde Indice  $\mathbb{N}$ 1 hasta 50 hacer
  Sum  $\mathbb{R}$  (sum + (1/Indice));
Fin_Para
Escribir('LA SUMA TOTAL ES = ',SUM)
Fin_Alg.
  
```



Ejemplo

Algoritmo para Sumar los números del 1 al 100.

```

Algoritmo Suma;
Variables
  Suma, Indice  $\mathbb{N}$  Enteros;
Inicio
  Suma  $\mathbb{N}$  0;
Para ir desde Indice  $\mathbb{N}$ 1 hasta 100 hacer
  Suma  $\mathbb{N}$  Suma + Indice;
Escribir('La suma Total es :', Suma );
Fin_Para.
Fin_Alg
  
```

Bucles Anidados

La estructura de bucles pueden anidarse internamente, es decir se puede situar un bucle en el interior de otro bucle, Al igual que sucede con los bucles MIENTRAS \ REPEAT-UNTIL es posible poner un bucle dentro de otro bucle, cuando de anidan bucles se debe tener en cuidado que el bucle interior esté completamente dentro del bucle exterior. si los bucles se cruzan no serán válidos.

Ejemplos

```

{ Para I  $\mathbb{N}$  Val_In Hasta n Haga
  { Para I  $\mathbb{N}$  Val_In Hasta n Haga
    { Sentencia1
    { Sentencia2
    { Fin_Para;
  { Fin_Para.
}
  
```

```

{
  Para I ES Val_Ini Hasta val_Final Haga
  {
    Para J ES Val_Ini Hasta val_Final Haga
    {
      PARA K ES Val_Ini Hasta val_Final Haga
      {
        Sentencia1;
        Sentencia2;
      }
      Fin_Para
      Sentencia3;
    }
    Fin_Para
    Sentencia4;
  }
  Fin_Para

```

Ejemplo

Elaborar un algoritmo para obtener la tabla popular de multiplicar de los escolares.

```

Algoritmo Tabla_Multip ;
I, j, Producto ES Enteros;
Inicio
Para Ir desde I ES 1 hasta 9 Haga
  Escribir (" Tabla del", I);
  Para Ir desde J ES 1 hasta 10 Haga
    Producto ES I * J;
    Escribir (I, " * ", J, "=", Producto);
  Fin_Para
Fin_Para
Fin_Alg

```

Ejercicios Resueltos

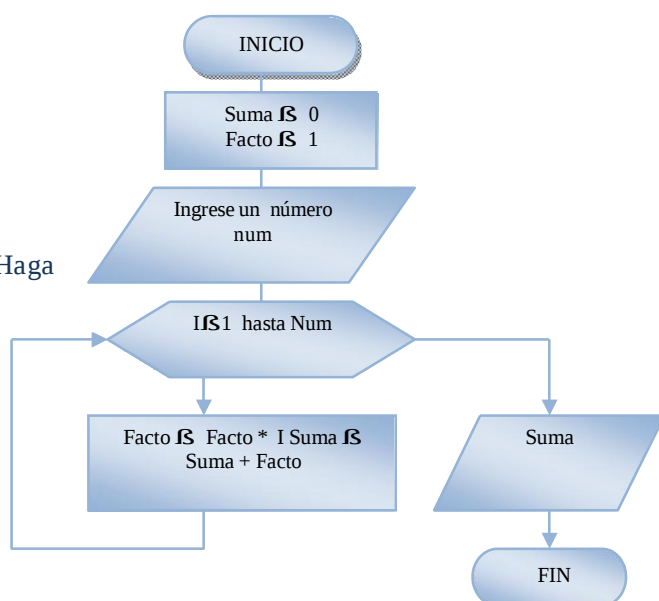
Ejercicio 1

Elaborar un diagrama de Flujo para calcular la sumatoria de la siguiente expresión $S = 1! + 2! + 3! + 4! + \dots + N!$

```

Algoritmo Factorial ;
Variables
I, Num, Facto, Suma ES Enteros;
Inicio
Suma ES 0;
Facto ES 1;
Escribir (" Ingrese un Número ");
Leer (Num);
Para Ir desde I ES 1 hasta Num Haga
  Facto ES Facto * I;
  Suma ES Suma + Facto;
Fin_Para
Escribir ( Suma );
Fin_Alg

```



Ejercicio 2

Algoritmo que permite imprimir las letras del abecedario (el código ASCII de la letra 'A' es 65).

```

Algoritmo Abecedario;
Variables
  I IS Entero;
  A IS Character;
Inicio
  Para Ir desde Indice IS 1 hasta 25 Haga
    A IS (chr ( 65+i ));
    Escribir ( ' Las letras del Abecedario son : ' , A );
  Fin_para
Fin_Alg

```

Ejercicio 3

Leer 500 números enteros y obtener cuantos son positivos

```

Algoritmo numero;
Variables
  Num ,pos,i IS Enteros;
Inicio
  Pos IS 0;
  Para Ir desde Indice IS 1 hasta 500 Haga
    Escribir ( ' Ingrese un número ');
    Leer( num );
    Si num > 0 Entonces
      Pos IS Pos + 1;
    Fin_Si
    Escribir ( 'Los números positivos son', Pos );
  Fin_Para
Fin_Alg.

```

Ejercicio 4

Diseñar un algoritmo para elevar (a^b). Luego imprimir el resultado

Ejemplo: Entrada 2 , 4 Salida 16
 Entrada 4 , 3 Salida 64

```

Algoritmo exponente;
Variables
  Base , Exp, result IS Enteros;
Inicio
  Result IS 1;
  Escribir ( ' Ingrese la Base y el Exponente ');
  Leer (Base, Exp);
  Para Ir desde Indice IS 1 hasta Exp Haga
    Result IS Result * Base;
    Escribir ( ' El resultado es : ' , Result );
  Fin_Para
Fin_Alg

```

Ejercicio 5

Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Diseño Estructurado de Algoritmos

```

Algoritmo promedio;
Variables
Sum, calif, c Enteros;
Prom Real;
Inicio
Sum 0
Para Ir desde c = 1 hasta 7 Haga
    Escribir ("Ingrese su calificación ", c);
    Leer (calif);
    Sum Sum (sum + calif);
Fin-para
prom Sum (sum / 7);
Imprimir (prom);
Fin.
    
```

Ejercicio 6

Leer 10 números e imprimir solamente los números positivos

```

Algoritmo positivos;
Variables
n, num Enteros;
Inicio
Para Ir desde n = 1 hasta 10 Haga
    Escribir (' Ingrese un número entero');
    Leer ( num);
    Si (num > 0) entonces
        Imprimir( num );
    fin-si
Fin-para
Fin.
    
```

Ejercicio 7

Leer 20 números e imprimir cuantos son positivos, cuantos negativos y cuantos neutros.

```

Algoritmo números;
Variables
cn, cp, cneg, num, x Enteros;
Inicio
cn = 0; cp
= 0; cneg
= 0;
Para Ir desde x = 1 hasta 20 Haga
    Escribir (' Ingrese un número entero');
    Leer (num);
    Sin ( num = 0) entonces
        cn cn + 1
    Caso contrario
        Si num > 0 entonces
            cp cp + 1
        Si num < 0 entonces
            cneg cneg + 1
    fin-si
Fin-para
Fin.
    
```

```

        Caso contrario
            cneg = cneg + 1;
    Fin-si
Fin-si
Fin-para
Imprimir( cn, cp, cneg);
Fin.

```

Ejercicio 8

Leer 15 números negativos y convertirlos a positivos e imprimir dichos números.

```

Algoritmo conversion;
Variables
num,x,pos ∈ Enteros;
Inicio
Para Ir desde x = 1 hasta 15 Haga
    Leer ( num );
    pos = num * -1;
    Imprimir (num, pos);
Fin-para
Fin.

```

Ejercicio 9

Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto.

```

Algoritmo tabla;
Variables
num,x,resul ∈ Enteros;
Inicio
Escribir ( ' Ingrese el número de la tabla de multiplicar a generarse' );
Leer (num);
Para Ir desde x = 1 hasta 12 Haga
    resul = (num * x);
    Imprimir (num, " * ", x, " = ", resul);
Fin-para
fin.

```

Ejercicio 10

Suponga que se tiene un conjunto de calificaciones de un grupo de 40 alumnos. Realizar un algoritmo para calcular la calificación media y la calificación mas baja de todo el grupo.

```

Algoritmo calificación;
Variables
sum, baja, calif, a ∈ Enteros;
media ∈ real;
Inicio
sum = 0;
baja = 9999;

```

```

Para Ir desde a = 1 hasta 40 Haga
  Escribir ( ' Ingrese calificación" );
  Leer( calif);
  sum  $\leftarrow$  (sum + calif)
  Si (calif < baja) entonces
    baja  $\leftarrow$  calif
  fin-si
Fin-para
media  $\leftarrow$  sum / 2
Imprimir(media, baja);
Fin

```

Ejercicios Propuestos

Ejercicios Individuales

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagrama de Flujo

1. Calcular la suma de los cuadrados de los 100 primeros números naturales.
2. Calcular la suma de cuatro números introducidos por teclado.
3. Calcular la suma de n números introducidos por teclado.
4. Calcular e imprimir los cuadrados del 20 al 50
5. Calcular la media aritmética de n números ingresados por teclado
6. Diseñar un algoritmo para imprimir los números impares en el rango del 1 al 100
7. Calcular la suma de los números enteros del 1 al 100
8. Imprimir todos los números pares entre 2 y 1000 inclusive
9. Leer 100 números. Determinar la media de los números positivos y la media de los números negativos
10. Elaborar un diagrama de flujo que calcule todas las ordenadas pares de la función $Y=f(x)=x^3+1$ el algoritmo debe imprimir la abscisa y la ordenada para los valores comprendidos entre 0 y 30

Ejercicios Grupales

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagrama de Flujo

1. Realizar un programa que lea 50 números e imprima el número mayor .
Ejemplo sea la serie : 2,3,1,5,8,4,2,6,3,5,6,7,5,7,8. La salida será : El mayor es 8
2. Se lanza una pelota hacia arriba con una velocidad inicial V_0 de 96 m/s la altura de la pelota sobre el nivel del suelo luego de t segundos se encuentra en h(t). Se sabe que $h(t) = V_0 \cdot t - 5t^2$ (h está dado en metros y t en segundos). Escriba un programa para producir una tabla que presente la altura de la pelota para cada valor de t de 1 a 8 segundos.
3. Escriba un programa interactivo para calificar a un grupo de 10 alumnos .el programa debe leer el nombre y sus tres calificaciones para cada

alumno, debe calcular el promedio para cada uno y después determinar si el alumno es aprobado o reprobado se requiere un promedio de al menos 60 para aprobar. La primera parte de la salida debe ser el nombre del alumno ,tres calificaciones, promedio y un mensaje de aprobado o reprobado . La segunda parte debe dar el numero de alumnos que aprobaron y el numero de los que obtuvieron un promedio de al menos 80

4. Escriba un programa pascal utilizando el ciclo FOR que permita imprimir las letras del abecedario en forma descendente desde la 'Z' .. ' A',(el código ASCII de la letra 'A' es 65).
5. Supongamos que un jardinero tenga 30m lineales de material (alambre de púa) para cercar un jardín rectangular, el área del jardín cumple la siguiente fórmula $X*(15-X)$. Escribir un programa para determinar el valor de X que produzca la máxima área probando todos los valores enteros positivos entre 10 y 15 , la salida debe ser :

Valor de x	Area
10	50
11	44
12	.
13	.
14	.

El área máxima es :

6. 5! (que se lee 'factorial 5') es igual al producto 5.4.3.2.1 Escriba un programa que calcule su factorial Cuando al correr este programa el usuario introduce 5 la salida debe ser: factorial 5 es 120
7. Una persona debe realizar un muestreo con 50 personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en su zona habitacional. Se determinan las categorías con base en la sig. tabla:

CATEGORIA	EDAD
Niños	0 - 12
Jóvenes	13 - 29
Adultos	30 - 59
Viejos	60 en adelante

8. Al cerrar un expendio de naranjas, 15 clientes que aun no han pagado recibirán un 15% de descuento si compran mas de 10 kilos. Determinar cuanto pagara cada cliente y cuanto percibirá la tienda por esas compras.
9. En un centro de verificación de automóviles se desea saber el promedio de puntos contaminantes de los primeros 25 automóviles que lleguen. Asimismo se desea saber los puntos contaminantes del carro que menos contamina y del que mas contamina.
10. Un entrenador le ha propuesto a un atleta recorrer una ruta de cinco kilómetros durante 10 días, para determinar si es apto para la prueba de 5

Kilómetros o debe buscar otra especialidad. Para considerarlo apto debe cumplir por lo menos una de las siguientes condiciones:

- Que en ninguna de las pruebas haga un tiempo mayor a 16 minutos.
- Que al menos en una de las pruebas realice un tiempo mayor a 16 minutos.
- Que su promedio de tiempos sea menor o igual a 15 minutos.

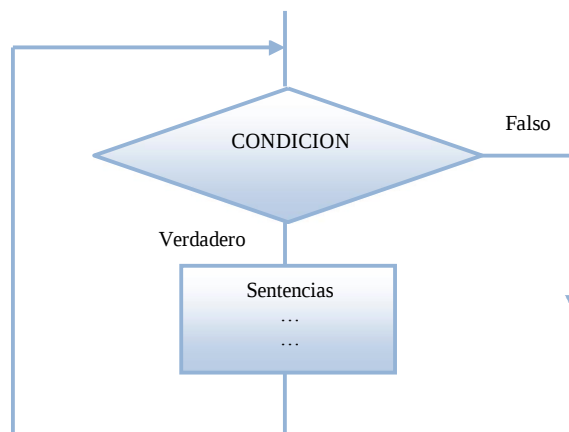
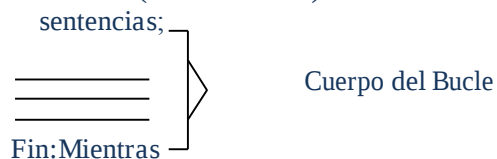
3.6. Sentencia Mientras Hacer

Un bucle o lazo mientras es un segmento de un algoritmo cuyas instrucciones se repiten un número determinado de veces mientras la condición sea verdadera. Se debe establecer mecanismo para determinar las tareas repetitivas. Este mecanismo es una condición que puede ser verdadera o falsa y que se comprueba una vez a cada paso o iteración del bucle (Total de instrucciones que se repiten en el bucle)

Un bucle consta de tres partes

- § Decisión
- § Cuerpo del Bucle
- § Salida del bucle

MIENTRAS (CONDICION) HACER



La condición (expresión lógica) se evalúa antes y después de cada ejecución de la expresión lógica, si la condición es verdadera se ejecuta el bucle, y si es falsa, el control pasa a la sentencia siguiente. Mientras la condición sea verdadera el bucle se ejecutará, esto significa que el bucle se ejecutará indefinidamente a menos que "ALGO "

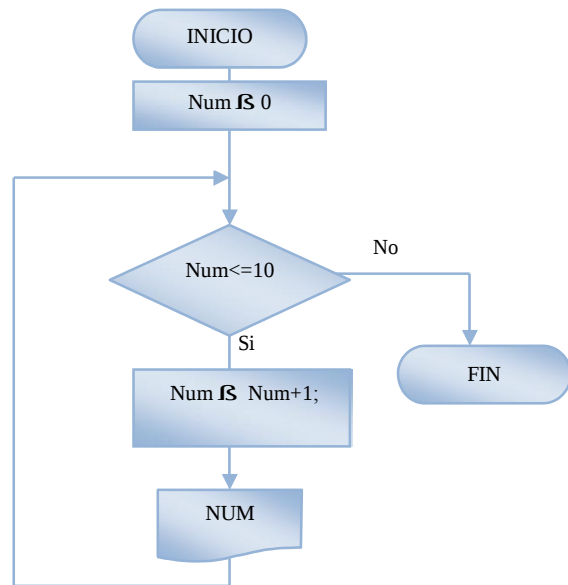
en el interior del bucle modifique la condición haciendo que su valor pase a falso, si la expresión nunca cambia el valor entonces estamos hablando de un ciclo infinito.

Ejemplo

imprimir los números del 1 al 10

```

Algoritmo Números ;
Variables
num IS Entero;
Inicio
Num IS 0;
Mientras num <= 10 Haga
    Num IS ( Num+ 1);
    Imprimir (Num);
Fin_Mientras;
Fin.
    
```



Ejemplo

Sumar los números del 1 al 100.

```

Algoritmo Suma ;
Variables
suma, num IS Entero;
Inicio
Suma IS 0;
Num IS 0;
Mientras num < 100 Haga
    Num IS Num+ 1;
    Suma IS Suma+ Num;
Fin_Mientras;
Imprimir (' La suma total es:' ,Suma );
Fin.
    
```

Ejemplo

Algoritmo para imprimir los cuadrados de los números del 1 al 10

```

Algoritmo Números ;
Variables
num IS Entero;
Inicio
Cont IS 0;
Num IS 0;
Mientras Cont < = 10 Haga
    Num IS cont * cont ;
    Cont IS Cont+ 1;
    Imprimir (cont, Num);
Fin_Mientras;
Fin.

```

Contadores

Los procesos repetitivos necesitan normalmente contar los sucesos o acciones internas del bucle. Una forma de controlar un bucle es mediante un contador. Un contador es una variable cuyo valor se incrementa o decrementa en una cantidad constante en cada iteración. Un bloque controlado por contador consta de tres partes :

- § Inicialización de la variable de control del bucle
- § Comprobación del valor de la variable de control del bucle
- § Incremento del valor de la variable de control del bucle.

Ejemplo. Pseudocódigo para repetir 50 veces e imprimir su valor

```

Algoritmo Números ;
Variables
contador IS Entero;
Inicio
Contador IS 1;                { Inicialización del contador }
Mientras contador >= 50 Hacer  { El bucle se ejecuta 50 veces }
    Imprimir ( Contador )      { Salida del 1,2,3,4,.. 50 }
    Contador := Contador + 1;   { contador se incrementa + 1 }
Fin_Mientras

```

Acumulador

Un acumulador o totalizador es una variable cuya misión es almacenar cantidades variables resultantes de sumas sucesivas. Realiza la misma función que un contador con diferencia de que el incremento o decremento de cada suma es variable en lugar de constante como es el caso del contador. Un bloque controlado por acumulador consta:

- § Inicialización de la variable de control del bucle en cero
- § Asignación de la variable a la misma variable mas el valor de la variable de control del bucle.

Ejemplo

Pseudocódigo para sumar los 10 primeros números

```

Algoritmo Números ;
Variables
Sum,cont ES Entero;
Inicio
Cont ES 0;
Suma ES 0;
Mientras cont <= 10 Hacer
    Suma ES (suma+numero);
    numero ES ( numero +1);
Fin_Mientras
Imprimir( Suma );
Fin_Alg
    
```

Bucles Infinitos

Un peligro de un ciclo MIENTRAS es que la condición nunca se satisfaga, los bucles infinitos no intencionados se deben evitar normalmente .

Ejemplo

```

contador:= 10;
Mientras contador <> 20 Hacer
    Leer ( numero );           { Ciclo Infinito }
    suma:= suma + numero ;
    contador := contador + 3;
Fin_Mientras
    
```

Ejercicios Resueltos

Ejercicio 1

Hacer un algoritmo para imprimir los números pares del 2 al 10

```

Algoritmo Pares;
Variables
Par ESEntero;
Inicio
Par ES 2;
Mientras par <= 10 Hacer
    Imprimir( par);
    Par ES Par + 2;
Fin_Mientras
Fin_Alg
    
```

Ejercicio 2

Realice un Algoritmo en que el usuario introduzca enteros positivos entre 1 y 5. El programa debe informar al usuario cuando haya escrito un entero que hace que la suma

de los números dados pase de 21. Además el mensaje MAS DE 21 se debe imprimir la suma y el último entero introducido.

```

Algoritmo ent_pos;
Variables
    sum,num ESEntero;
Inicio
    Sum ES 0;
Mientras sum< = 21 Haga
    Escribir('Ingrese un numero entero entre 1..5');
    Leer(num);
    Si ((num > 0) y (num< = 5 )) Entonces
        sum:= sum+ num
    Caso Contrario
        Escribir(' Error Ingrese otro numero');
    Fin_si
    Imprimir('MAS DE 21 ');
    Imprimir('La Suma es : ',sum);
    Imprimir('El Numero es : ',num);
Fin.

```

Ejercicio 3

Las potencias de 2 son 1,2,4,8,16,32,64, etc. Para obtener la siguiente potencia de 2 se multiplica la anterior por 2. Usando un ciclo MIENTRAS escriba un Programa que imprima la primera potencia de 2 que pase más de 1000

```

Algoritmo Potencia;
variables
    pot ESEntero;
Inicio
    Pot ES 1;
Mientras pot< = 1000 Haga
    Pot ES Pot * 2;
    Imprimir('La primera potencia mayor que 1000 es',pot);
Fin.

```

Ejercicio 4

Escriba un algoritmo para calcular la suma de: $1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/50$

```

Algoritmo Suma;
Variables
    x, y ESEntero;
    sum, cont ES Real;
Inicio
    Suma ES 0;
    Cont ES 1;
Mientras cont < = 50 Haga
    Suma ES suma + 1/cont;
    Cont ES cont+ 1;
Fin_Mientras
Escribir('La suma es : ', suma );
Fin_Alg

```

Ejercicio 5

Escriba un Algoritmo para calcular el valor máximo de una serie de 10 números ingresados por teclado.

```

Algoritmo Maximo;
Variables
N, max ,cont Entero;
Inicio
Leer (n)
Cont 1;
Max 0;
Mientras cont < 10 Haga
    Cont cont+ 1;
    Leer(n)
    Si n> max entonces
        Max n;
    Fin_Mientras
Escribir ('EL mayor es : ', Max );
Fin_Alg

```

Ejercicios Propuestos

Ejercicios Individuales

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagrama de Flujo

1. Obtener el promedio de calificaciones de un grupo de n alumnos.
2. Calcular el promedio de edades de hombres, mujeres y de todo un grupo de alumnos.
3. Encontrar el menor valor de un conjunto de n números dados.
4. Encontrar el mayor valor de un conjunto de n números dados.
5. Determinar cuantos hombres y cuantas mujeres se encuentran en un grupo de n personas, suponiendo que los datos son extraídos alumno por alumno.
6. Realice un Algoritmo que dado un cierto número de monedas de 5,10,20,50 sucres, imprima el total de la suma.
7. Determinar la media de una lista de números positivos terminado con un número negativo después del último numero válido.
8. Calcular e imprimir la suma y el producto de los números pares comprendidos entre 20 y 400 ambos inclusive

9. Leer 100 números. Determinar la media de los números positivos y la media de los números negativos
10. Averiguar si dados dos números leídos del teclado, uno es divisor del otro.

Ejercicios Grupales

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagrama de Flujo

1. Una persona desea invertir su dinero en un banco, el cual le otorga un 2% de interés. Cual será la cantidad de dinero que esta persona tendrá al cabo de un año si la ganancia de cada mes es reinvertida?.
2. Una compañía de seguros tiene contratados a n vendedores. Cada uno hace tres ventas a la semana. Su política de pagos es que un vendedor recibe un sueldo base, y un 10% extra por comisiones de sus ventas. El gerente de su compañía desea saber cuanto dinero obtendrá en la semana cada vendedor por concepto de comisiones por las tres ventas realizadas, y cuanto tomando en cuenta su sueldo base y sus comisiones.
3. En una empresa se requiere calcular el salario semanal de cada uno de los n obreros que laboran en ella. El salario se obtiene de la sig. forma:
 Si el obrero trabaja 40 horas o menos se le paga \$20 por hora
 Si trabaja mas de 40 horas se le paga \$20 por cada una de las primeras 40 horas y \$25 por cada hora extra.
4. El Depto. de Seguridad Publica y Transito del D.F. desea saber, de los n autos que entran a la ciudad de México, cuantos entran con calcomanía de cada color. Conociendo el ultimo dígito de la placa de cada automóvil se puede determinar el color de la calcomanía utilizando la sig. relación:

DÍGITO	COLOR
1 o 2	amarilla
3 o 4	rosa
5 o 6	roja
7 o 8	verde
9 o 0	azul

5. En un supermercado un cajero captura los precios de los artículos que los clientes compran e indica a cada cliente cual es el monto de lo que deben pagar. Al final del día le indica a su supervisor cuanto fue lo que cobro en total a todos los clientes que pasaron por su caja.
6. Cinco miembros de un club contra la obesidad desean saber cuanto han bajado o subido de peso desde la ultima vez que se reunieron. Para esto se debe realizar un ritual de pesaje en donde cada uno se pesa en diez básculas

distintas para así tener el promedio mas exacto de su peso. Si existe diferencia positiva entre este promedio de peso y el peso de la ultima vez que se reunieron, significa que subieron de peso. Pero si la diferencia es negativa, significa que bajaron. Lo que el problema requiere es que por cada persona se imprima un letrero que diga: “SUBIO” o “BAJO” y la cantidad de kilos que subió o bajo de peso.

7. Se desea obtener el promedio de g grupos que están en un mismo año escolar; siendo que cada grupo puede tener n alumnos que cada alumno puede llevar m materias y que en todas las materias se promedian tres calificaciones para obtener el promedio de la materia. Lo que se desea desplegar es el promedio de los grupos, el promedio de cada grupo y el promedio de cada alumno.
8. Escriba un Algoritmo que cuente el número de cifras en un entero de entrada (hasta 32.000) así para una entrada de 5837 la salida debe ser:
4 cifras
Asegúrese de que el programa funcione para los números 0, 1 y 9
para el número 0 la salida debe ser:
1 cifra
9. Escribir un programa que pida al usuario escribir un entero diferente de cero y después imprima su recíproco. Use un ciclo Mientras para forzar al usuario a volver a escribir el valor cuando escribe 0 por error.
10. Escriba un programa para evaluar la función $y = 4x^2 - 16x + 15$, donde x toma los valores de 1 a 2 en pasos de 1 . Para cada x la salida debe dar el valor de y y el mensaje POSITIVO o el mensaje NO POSITIVO. El formato de la salida debe ser :

Valor de X	Valor de Y	
1.0	3.00	Positivo
1.1	2.24	Postivo
.	.	.
.	.	.
.	.	.
2.0	-1.00	No Positivo

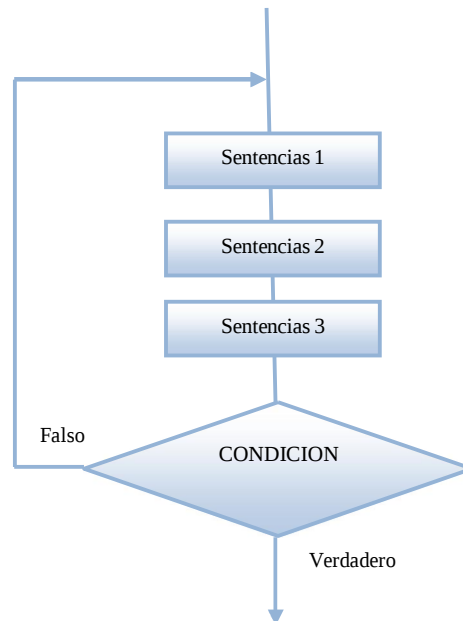
3.7. Sentencia Repetir \Hasta

Un bucle o lazo REPETIR / HASTA es un segmento de un algoritmo cuyas instrucciones se repiten un número determinado de veces mientras la condición sea FALSA, cuando la expresión se haga Verdadera, sale del lazo.

Así es como se debe escribir una sentencia Repetir.

```

REPETIR
    Sentencia 1;
    Sentencia 2;
    .
    .
    SentenciaN;
HASTA (CONDICIÓN)
    
```



Nota:

Debido a que HASTA delimita la sentencia, no es necesario poner Inicio y Fin.

El bloque de instrucciones se ejecuta al menos una vez y luego se evalúa la condición. Además si la expresión lógica o condición es falsa, se vuelve a Repetir el bucle y se ejecutan todas las sentencias y se vuelve a chequear la condición. Si la expresión lógica es Verdadera, se sale del ciclo y se ejecutan la siguiente Instrucciones.

Ejemplo

Algoritmo para sumar los números del 1 al 10 .

```

Algoritmo suma ;
Variables
suma, numero IS Entero;
Inicio
Suma IS 0;
Numero IS 0;
REPETIR
    Numero IS Numero+1; Suma
IS Suma + Numero; HASTA
Numero = 10 ; Escribir( 'El
total es:',Suma ); Fin_Alg
    
```

El bucle se repite hasta que el valor de variable `Numero` se iguala a 10, lo que sucederá después de 10 ejecuciones del cuerpo del bucle

Ejemplo

Algoritmo para imprimir los números del 1 al 100

```

Algoritmo numeros ;
Variables
numero  $\mathbb{N}$  Entero;
Inicio
Numero  $\mathbb{N}$  1;
REPETIR
Imprimir(Numero);
Numero  $\mathbb{N}$  Numero+1;
HASTA Numero = 101;
Fin_Alg

```

Ejercicios Resueltos

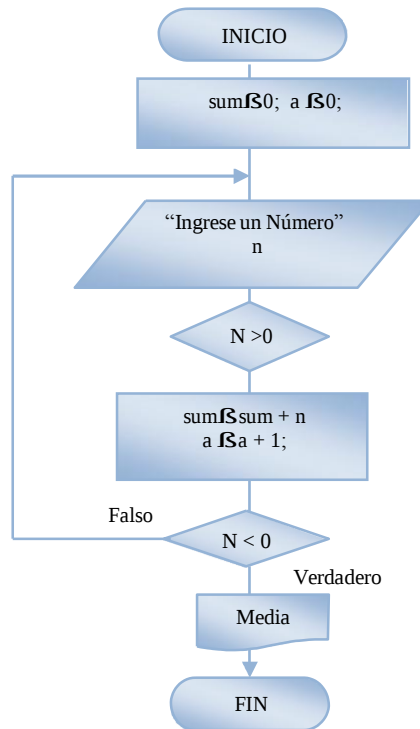
Ejercicio 1

Determinar la media de una lista de números positivos terminada con un número negativo después del último número válido

```

Algoritmo num_Pos;
variables
sum, a, n  $\mathbb{N}$ Enteros;
media  $\mathbb{R}$ real ;
Inicio
sum  $\mathbb{N}$ 0; a
 $\mathbb{N}$ 0;
REPETIR
Escribir('Ingrese un número ');
Leer( n );
Si n>0 Entonces
    sum  $\mathbb{N}$ sum + n;
    a  $\mathbb{N}$ a + 1;
Fin_Si
HASTA n< 0;
Imprimir (Media  $\mathbb{R}$ sum/a);
Fin_Alg

```



Ejercicio 2

Se desea leer por teclado un número comprendido entre 1 y 50 (inclusive), y se desea visualizar si el número es par o impar.

```

Algoritmo numero;
Variables
  N ∈ Entero;
Inicio
  Escribir(' Ingrese un número ');
  Leer( n );
  REPETIR
    Si ( n mod 2 ) = 0 Entonces
      Escribir('Número par');
    Fin_Si
    Si ( n mod 2 ) < > 0 Entonces
      Escribir('Número impar');
    Fin_Si
  HASTA ( n >= 1 ) y ( n <= 50 );
Fin_alg
  
```

Ejercicio 3

Algoritmo para hallar el promedio de una lista de n números

```

Algoritmo media;
Variables
  N, cont ∈ Entero;
  X, suma, media ∈ Real;
Inicio
  Cont ← 1;
  Suma ← 0;
  Leer ( n )
  
```

```

REPETIR
Leer (x)
Suma  $\mathbf{\$}$  suma + x
Cont  $\mathbf{\$}$  cont + 1
HASTA cont > n
Media suma/n;
Escribir ( ' La media es ', media );
Fin_Algo

```

Ejercicio 4

Algoritmo para obtener la suma de los números pares hasta 100 inclusive

$S = 2 + 4 + 6 + 8, \dots, +100$

```

Algoritmo pares;
Variables
suma, Numero  $\mathbf{\$}$  Reales;
Inicio
Suma  $\mathbf{\$}$  0;
Numero  $\mathbf{\$}$  4;
REPETIR
Suma  $\mathbf{\$}$  suma + Numero
Numero  $\mathbf{\$}$  Numero + 1
HASTA Numero > 100
Escribir ( ' La suma es : ', suma );
Fin_Algo

```

Ejercicio 5

Escriba un programa que cuente el número de cifras en un entero de entrada (hasta 32.000) así para una entrada de 5837 la salida debe ser: 4 cifras. Asegúrese de que el programa funcione para los números 0,1 y 10 . Para el número 0 la salida debe ser: 1 cifra

```

Algoritmo cifras;
variables
a  $\mathbf{\$}$  enteros;
Inicio
Repetir
Escribir('Ingrese un número desde 1.. a 32000 ');
Leer (a );
Hasta ( a >= 0 ) y ( a <= 32000 );
Si ((a >= 0) y (a < 10)) entonces
Imprima ('1 cifra ');
Si ((a >= 10) and (a <= 99)) entonces
Imprima ('2 cifras ');
Si ((a >= 100) and (a <= 999)) entonces
Imprima ('3 cifras ');
Si ((a >= 1000) and (a <= 9999)) entonces
Imprima ('4 cifras ');
Si ((a >= 10000) and (a <= 32000)) entonces
Imprima ('5 cifra ');
Fin.

```

Ejercicios Propuestos

Ejercicios Grupales:

Desarrolle los siguientes problemas a un lenguaje algorítmico y diagrama de Flujo

1. Determinar la cantidad semanal de dinero que recibirá cada uno de los n obreros de una empresa. Se sabe que cuando las horas que trabajo un obrero exceden de 40, el resto se convierte en horas extras que se pagan al doble de una hora normal, cuando no exceden de 8; cuando las horas extras exceden de 8 se pagan las primeras 8 al doble de lo que se paga por una hora normal y el resto al triple.
2. Una persona que va de compras a la tienda “Enano, S.A.”, decide llevar un control sobre lo que va comprando, para saber la cantidad de dinero que tendrá que pagar al llegar a la caja. La tienda tiene una promoción del 20% de descuento sobre aquellos artículos cuya etiqueta sea roja. Determinar la cantidad de dinero que esta persona deberá pagar.
3. Un censador recopila ciertos datos aplicando encuestas para el ultimo Censo Nacional de Población y Vivienda. Desea obtener de todas las personas que alcance a encuestar en un día, que porcentaje tiene estudios de primaria, secundaria, carrera técnica, estudios profesionales y estudios de posgrado.
4. Un jefe de casilla desea determinar cuantas personas de cada una de las secciones que componen su zona asisten el día de las votaciones. Las secciones son: norte, sur y centro. También desea determinar cual es la sección con mayor numero de votantes.
5. Un negocio de copias tiene un limite de producción diaria de 10 000 copias si el tipo de impresión es offset y de 50 000 si el tipo es estándar. Si hay una solicitud de un el empleado tiene que verificar que las copias pendientes hasta el momento y las copias solicitadas no excedan del limite de producción. Si el limite de producción se excediera el trabajo solicitado no podría ser aceptado. El empleado necesita llevar un buen control de las copias solicitadas hasta el momento para decidir en forma rápida si los trabajos que se soliciten en el día se deben aceptar o no.
6. Calcular la suma siguiente: $100 + 98 + 96 + 94 + \dots + 0$ en este orden
7. Leer por cada alumno de Diseño estructurado de algoritmos su numero de control y su calificación en cada una de las 5 unidades de la materia. Al final que escriba el numero de control del alumno que obtuvo mayor promedio. Suponga que los alumnos tienen diferentes promedios.

8. El profesor de una materia desea conocer la cantidad de sus alumnos que no tienen derecho al examen de nivelación. Diseñe un algoritmo que lea las calificaciones obtenidas en las 5 unidades por cada uno de los 40 alumnos y escriba la cantidad de ellos que no tienen derecho al examen de nivelación.
9. Suponga que tiene usted una tienda y desea registrar las ventas en su computadora. Diseñe un algoritmo que lea por cada cliente, el monto total de su compra. Al final del día que escriba la cantidad total de ventas y el numero de clientes atendidos.
10. En un supermercado una ama de casa pone en su carrito los artículos que va tomando de los estantes. La señora quiere asegurarse de que el cajero le cobre bien lo que ella ha comprado, por lo que cada vez que toma un articulo anota su precio junto con la cantidad de artículos iguales que ha tomado y determina cuanto dinero gastara en ese articulo; a esto le suma lo que ira gastando en los demás artículos, hasta que decide que ya tomo todo lo que necesitaba. Ayúdale a esta señora a obtener el total de sus compras.

2.11. Elección Estructuras Repetitiva Adecuada

La elección adecuada de estructuras repetitivas es una de las partes más importantes en el diseño de un algoritmo. La repetición de sentencias en el interior del bucle requiere definir las condiciones de terminación del bucle con sumo cuidado para evitar un bucle incorrecto o infinito.

En un bucle Repetir Hasta , la secuencia de sentencias se ejecutará al menos una vez, mientras que en bucle Mientras no siempre se ejecutará el proceso del bucle y en ocasiones (condición falsa) no se ejecutará. En la practica, el bucle Mientras se utiliza con más frecuencia que el bucle REPETIR.

El PARA se reserva para situaciones en que el número de repeticiones se puede predecir y enumerar fácilmente antes que comience el bucle. Un bucle for se puede reescribir fácilmente (normalmente con un bucle MIENTRAS) utilizando otras estructuras.

Cuando no se conocen con anticipación cuántas veces se tiene que ejecutar un bucle, el uso de las estructuras MIENTRAS es el mas indicado, incluso aun cuando el bucle está controlado por un simple incremento o decremento de un contador entero.

Es frecuente confundir estructuras SI- ENTONCES con MIENTRAS, debido al hecho que muchos algoritmos tienen estructuras que implican bucles dentro de los cuáles se realizan estructuras de decisión secundarias.

En conclusión podemos decir que para seleccionar un bucle correcto se debe tener en cuenta:

1. Si el número de interacciones se conoce por adelantado, se debe utilizar PARA
2. Si el número de interacciones es indeterminado y el bucle no se debe ejecutar cuando la condición sea falsa la primera vez, se debe utilizar MIENTRAS
3. Si el número de interacciones es indeterminada y el bucle se debe ejecutar siempre al menos una vez, utilizar REPETIR HASTA
4. Si el número de interacciones es indeterminada y debe cumplir una cierta condición (Decisión) y el bucle se debe ejecutar solo una vez, utilizar SI-ENTONCES ó CASO QUE/HAGA

Autoevaluación Final

Diseñe un algoritmo y flujograma con las sentencias mas óptimas que se considere usted

1. Elaborar un Algoritmo que lea un carácter y almacene en una variable entera el número correspondiente en la tabla ASCII. El programa debe escribir alguno de los siguientes mensajes:
 ' MAYOR ' Si la suma de los dígitos que componen el número almacenado es mayor a 10
 ' IGUAL ' Si la suma de los dígitos que componen el número almacenado es igual a 10
 ' MENOR ' Si la suma de los dígitos que componen el número almacenado es menor a 10
 Si al carácter leído le corresponde un número ASCII menor a 33 o mayor a 126, se debe escribir el mensaje : 'Carácter de Control '
 Independientemente del mensaje escrito, se debe escribir al final de la ejecución del programa, el mensaje 'FINALIZAMOS '.
2. Calcular todas las ordenadas impares de la función $Y=f(x)=x^3+2$. El algoritmo debe imprimir la abscisa y la ordenada para los valores comprendidos entre 1 y 50
3. Escriba un algoritmo que calcule la suma de los cuadrados de los 100 enteros es decir la adición $S=1^2+2^2+3^2+4^2+\dots+100^2$
4. Escriba un algoritmo que lea un dato n y calcule los siguientes valores n^n y $n!$

5. Escriba un algoritmo que lea los datos a , n y r y calcule la Sumatoria de:

$$\text{Suma} = ar^{n-1} \quad \text{y} \quad \text{Producto} = N! * 2^n$$

6. Suponga que tiene una tienda y desea registrar sus ventas por medio de una computadora. Diseñe un pseudocódigo que lea por cada cliente:

- a).- el monto de la venta,
- b).- calcule e imprima el IVA ,
- c).-calcule e imprima el total a pagar,
- d).- lea la cantidad con que paga el cliente,
- e).-calcule e imprime el cambio.

Al final del día deberá imprimir la cantidad de dinero que debe haber en la caja.

7. Se tiene un conjunto de 1,000 tarjetas cada una contiene la información del censo para una persona:

- 1.- Número de censo,
- 2.- Sexo
- 3.- Edad
- 4.- Estado civil (a.- soltero, b. Casado, c. Viudo, d. Divorciado)

Diseñe un pseudocódigo estructurado que lea todos estos datos, e imprima el número de censo de todas las jóvenes solteras que estén entre 16 y 21 años.

8. Diseñe un pseudocódigo que lea el valor de un ángulo expresado en radianes y calcule e imprima el valor del seno de dicho ángulo. Se leerá también el número de términos de la serie. $\text{SEN}(X) = X - (X^3 / 3!) + (X^5 / 5!) - (X^7 / 7!) + \dots$

9. Calcule la suma de los términos de la serie FIBONACCI cuyos valores se encuentran entre 100 y 10,000.

CAPITULO 4

Arreglos, Vectores y Matrices

- 4.1 Arreglos
- 4.2 Sentencias Secuenciales
- 4.3 Sentencias de Decisión o Secuenciales
- 4.4 Sentencias de Selección Anidada

4.1. Arreglos

Un Arreglo es una estructura de datos que almacena bajo el mismo nombre (variable) a una colección de datos del mismo tipo.

Los arreglos se caracterizan por:

- § Almacenan los elementos en posiciones contiguas de memoria
- § Tienen un mismo nombre de variable que representa a todos los elementos. Para hacer referencia a esos elementos es necesario utilizar un índice que especifica el lugar que ocupa cada elemento dentro del archivo.

Tipo de Arreglos	{	Unidimensionales (Vectores)
		Bidimensionales (Matrices)

4.2. Vectores

Es un arreglo de “N” elementos organizados en una dimensión donde “N” recibe el nombre de longitud o tamaño del vector. Para hacer referencia a un elemento del vector se usa el nombre del mismo, seguido del índice (entre corchetes), el cual indica una posición en particular del vector. Por ejemplo:

Vec[x]

Donde:

Vec..... Nombre del arreglo
 x..... Numero de datos que constituyen el arreglo

Representación gráfica de un vector

Vec[1]	7
Vec[2]	8
Vec[3]	9
Vec[4]	1
	0

Llenado de un Vector

Hacer para I = 1 a 10

Leer vec[I]

Fin-para

Hacer mientras I <= 10

Leer vec[I]

Fin-mientras

I=1

Repetir

Leer vec[I]

I = I + 1

Hasta-que I > 10

Ejercicios Propuestos

1. Calcular el promedio de 50 valores almacenados en un vector. Determinar además cuantos son mayores que el promedio, imprimir el promedio, el numero de datos mayores que el promedio y una lista de valores mayores que el promedio.
2. Llenar dos vectores A y B de 45 elementos cada uno, sumar el elemento uno del vector A con el elemento uno del vector B y así sucesivamente hasta 45, almacenar el resultado en un vector C, e imprimir el vector resultante.
3. Llenar un vector de 20 elementos, imprimir la posición y el valor del elemento mayor almacenado en el vector. Suponga que todos los elementos del vector son diferentes.
4. Almacenar 500 números en un vector, elevar al cuadrado cada valor almacenado en el vector, almacenar el resultado en otro vector. Imprimir el vector original y el vector resultante.
5. Almacenar 300 números en un vector, imprimir cuantos son ceros, cuantos son negativos, cuantos positivos. Imprimir además la suma de los negativos y la suma de los positivos.

6. Almacenar 150 números en un vector, almacenarlos en otro vector en orden inverso al vector original e imprimir el vector resultante.
7. Se tienen almacenados en la memoria dos vectores M y N de cien elementos cada uno. Hacer un algoritmo que escriba la palabra “Iguales” si ambos vectores son iguales y “Diferentes” si no lo son.
8. Serán iguales cuando en la misma posición de ambos vectores se tenga el mismo valor para todos los elementos.
9. Se tiene el vector A con 100 elementos almacenados. Diseñe un algoritmo que escriba “SI” si el vector esta ordenado ascendentemente o “NO” si el vector no esta ordenado
10. Diseñe un algoritmo que lea un numero cualquiera y lo busque en el vector X, el cual tiene almacenados 80 elementos. Escribir la posición donde se encuentra almacenado el numero en el vector o el mensaje “NO” si no lo encuentra. Búsqueda secuencial.
11. Diseñe un algoritmo que lea dos vectores A y B de 20 elementos cada uno y multiplique el primer elemento de A con el ultimo elemento de B y luego el segundo elemento de A por el diecinueveavo elemento de B y así sucesivamente hasta llegar al veinteavo elemento de A por el primer elemento de B. El resultado de la multiplicación almacenarlo en un vector C.
12. Diseñe un algoritmo que almacene en un vector llamado FIB[100] los 100 primeros números de la serie fibonacci.

5.3. Matriz

Es un arreglo de $M * N$ elementos organizados en dos dimensiones donde “M” es el numero de filas o reglones y “N” el numero de columnas.

Para representar una matriz se necesita un nombre de matriz se necesita un nombre de matriz acompañado de dos índices.

Mat [R,C]

Donde R indica el renglón y C indica la columna, donde se encuentra almacenado el dato.

Representación gráfica de una matriz

Mat [R,C]

Fila o
Renglon
[R]

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

Columna
[C]

Llenado de una matriz

```

Por renglones
Hacer para R = 1 a 5
  Hacer para C = 1 a 5
    Leer Mat [R,C]
  Fin-para
Fin-para
  
```

```

Por columnas
Hacer para C = 1 a 5
  Hacer para R = 1 a 5
    Leer Mat [R,C]
  Fin-para
Fin-para
  
```

Nota

Para hacer el llenado de una matriz se deben de usar dos variables para los índices y se utilizan 2 ciclos uno para los renglones y otro para las columnas; a estos ciclos se les llama ciclos anidados (un ciclo dentro de otro ciclo).

Ejercicios Propuestos

1. Hacer un algoritmo que almacene números en una matriz de 5 * 6. Imprimir la suma de los números almacenados en la matriz.
2. Hacer un algoritmo que llene una matriz de 10 * 10 y determine la posición [renglón ,columna] del numero mayor almacenado en la matriz. Los números son diferentes.
3. Hacer un algoritmo que llene una matriz de 7 * 7. Calcular la suma de cada renglón y almacenarla en un vector, la suma de cada columna y almacenarla en otro vector.

4. Hacer un algoritmo que llene una matriz de $20 * 20$. Sumar las columnas e imprimir que columna tuvo la máxima suma y la suma de esa columna.
5. Hacer un algoritmo que llene una matriz de $5 * 5$ y que almacene la diagonal principal en un vector. Imprimir el vector resultante.
6. Hacer un algoritmo que llene una matriz de $10 * 10$ y que almacene en la diagonal principal unos y en las demás posiciones ceros.
7. Hacer un algoritmo que llene una matriz de $6 * 8$ y que almacene toda la matriz en un vector. Imprimir el vector resultante.
8. Hacer un algoritmo que llene una matriz de $8 * 8$, que almacene la suma de los renglones y la suma de las columnas en un vector. Imprimir el vector resultante.
9. Hacer un algoritmo que llene una matriz de $5 * 6$ y que imprima cuantos de los números almacenados son ceros, cuantos son positivos y cuantos son negativos.
10. Diseñe un pseudocódigo que escriba el numero de la hilera cuya suma sea mayor que las demás hileras. Suponga que todas las hileras suman diferente cantidad.

Bibliografía

1. JEAN Paul Tremblay, B. Bunt Richard; "Introducción a la ciencias de las computadoras (enfoque algoritmico)" Mc Graw Hill
2. JOYANES Aguilar Luis; "Metodología de la programación" Mc Graw Hill
3. JOYANES Aguilar Luis; "Problemas de metodología de la programación" Mc Graw Hill
4. CORREA Uribe Guillermo; "Desarrollo de algoritmos y sus aplicaciones en Basic, Pascal y C (3ª. Edición)" Mc Graw Hill
5. Levine Guillermo; "Introducción a la computación y a la programación estructurada" Mc Graw Hill
6. JOYANES Aguilar Luis; "Fundamentos de programación, algoritmos y estructura de datos" Mc Graw Hill
7. JOYANES Aguilar Luis, Luis Rodríguez Baena y Matilde Fernández Azuela; "Fundamentos de programación, libro de problemas" Mc Graw Hill
8. Bores Rosario, Rosales Roman; "Computación. Metodología, lógica computacional y programación" Mc Graw Hill
9. LOZANO Letvin; "Diagramación y programación estructurada y libre" Mc Graw Hill
10. LOPEZ Roman Leobardo; "Programación estructurada (enfoque algoritmico)" Computec

Anotaciones

Anotaciones

Derechos Reservados



Todos los derechos de este material se encuentran reservados, esto implica sanciones contra copia, plagio total o parcial fotocopiado y xerografiado.

Esta obra se terminó de editar en Abril del 2007
Matriz Original Microsoft Word 2003
LIC. DIEGO ROLANDO REALPE B.
LIC. WASHINGTON FIERRO
Copyright © 2007