

¡Aprende a programar!
Soluciones en pseudocódigo



Programación

FACULTAD DE INFORMÁTICA
● UAS ●

MC. Gerardo Gálvez Gámez
MC. Rogelio Prieto Alvarado
MC. José de Jesús Uriarte Adrián

Agosto de 2013



ÍNDICE

CAPITULO I. Componentes de un sistema computacional	1
Sistema Computacional.....	1
Computadora	1
Dispositivos de Entrada.....	1
Dispositivos de Salida	2
Unidad Central de Procesamiento CPU.....	2
Memoria Principal	2
Hardware	4
Software.....	4
Sistema Operativo.....	4
Tipos de Interfaz.....	5
CAPITULO II. Conceptos de Programación	6
Lenguaje	6
Lenguaje de programación	6
Programa	7
Algoritmo.....	7
Ejemplo de algoritmo	8
Lenguajes Algorítmicos	8
Práctica y actividad para el alumno.....	8
CAPITULO III. Entidades primitivas para el desarrollo de algoritmos	9
Tipos de Datos	9
Variables.....	12
Constantes	13
Identificadores	14
Expresiones	15
Operadores y Operandos.....	15
Operador de Asignación	17
Actividades Prácticas	18



CAPITULO IV. Metodología para programación	21
Importancia de la aplicación de técnicas y herramientas metodológicas.....	21
Etapas en la solución de un problema.....	21
Problema	22
Análisis del problema.....	23
Construcción del algoritmo en Pseudocódigo	24
Verificación (prueba y depuración).....	25
Codificación.....	25
Documentación	25
Mantenimiento.....	26
Análisis de algoritmos	26
CAPITULO V. Estructuras Algorítmicas Básicas.....	27
Estructuras Básicas.....	27
Estructura Secuencial	27
Instrucciones de Entrada/Salida.....	28
Caso práctico utilizando estructura secuencial	28
Problemas prácticos de estructura secuencial.....	29
Estructuras de selección (condicionales)	31
Estructura de selección simple.....	31
Caso Práctico de estructura de selección simple.....	31
ANEXO A - Código ASCII.....	34
ANEXO B – Instrucciones de Entrega – Tareas	35
ANEXO 01 – TAREA Control de Lectura	36
ANEXO 02 – TAREA Resolución de Problemas Lógicos	37
ANEXO 03 – TAREA Elaboración de Algoritmos.....	38
ANEXO 04 – TAREA Conceptos, Tipos de Datos y Expresiones.....	39
ANEXO 05 – TAREA Elaboración de Algoritmos en pseudocódigo (i)	42
ANEXO 06 – TAREA Elaboración de Algoritmos en pseudocódigo (ii)	43
ANEXO 07 – TAREA Elaboración de Algoritmos en pseudocódigo (iii)	44
BIBLIOGRAFIA.....	45



CAPITULO I.

Componentes de un sistema computacional

Objetivo: El alumno definirá conceptos básicos y será capaz de distinguir los componentes esenciales de las Tecnologías de la Información y Comunicación.

Sistema Computacional

Un Sistema Computacional consiste en un conjunto de componentes electrónicos y electromecánicos interconectados que almacenan y transforman símbolos en base a las instrucciones especificadas en los componentes software del mismo sistema.

Un sistema computacional encierra cuatro componentes esenciales:

- Dispositivos de Entrada
- Dispositivos de salida
- Unidad Central de Procesamiento
- Memoria Principal y Secundaria

Computadora

Es un conjunto de dispositivos electrónicos que nos permiten procesar y almacenar grandes volúmenes de información a alta velocidad, para resolver problemas de tipo administrativo, técnico, de investigación, etc. de forma rápida y precisa. Debido a esto la computadora se convierte en una herramienta muy útil para cualquier persona, empresa o institución.



Una Computadora debe ser capaz de recibir, a través de sus dispositivos de entrada, ciertos datos e instrucciones para manipular éstos. Una vez que los datos e instrucciones son ingresados, la computadora debe ser capaz de almacenarlos internamente en su memoria principal y luego, procesar los datos en base a las instrucciones suministradas utilizando su(s) procesador(es). Finalmente, el producto resultante del procesamiento de los datos es entregado al usuario u otros sistemas a través de los dispositivos de salida.

Dispositivos de Entrada

Los Dispositivos de Entrada son aquéllos que permiten introducir la información a la computadora. Estos dispositivos convierten la información en secuencias propias de bits, capaces de ser interpretadas por el procesador. Algunos ejemplos de estos dispositivos son: el mouse, el teclado, el micrófono, la webcam, el escáner, etc.





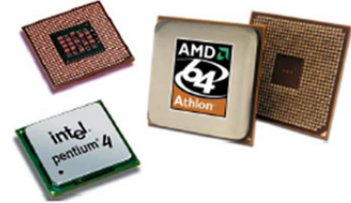
Dispositivos de Salida

Los Dispositivos de Salida son aquellos que envían la información del interior de la computadora al exterior, mostrándola al usuario. Es todo lo que sale de la computadora y puede aparecer en una variedad de formas; como caracteres, como imágenes, como páginas impresas, etc. Algunos ejemplos de estos dispositivos son: la impresora, el monitor, las bocinas, etc.



Unidad Central de Procesamiento CPU

Todos los sistemas computacionales poseen un procesador llamado CPU. Cada procesador es capaz de interpretar y ejecutar instrucciones y es el responsable de las operaciones básicas de la computadora. Constituye la parte operacional más importante de todo sistema computacional.



CPU es la abreviación de Central Process Unit. La CPU es el cerebro de la computadora. Es la encargada de realizar operaciones aritméticas y lógicas, coordinar las actividades de la computadora y determinar que operaciones se deben realizar y en que orden; así mismo controlar todo el proceso de la computadora.

En las computadoras personales, la CPU es contenida en un solo chip llamado microprocesador.

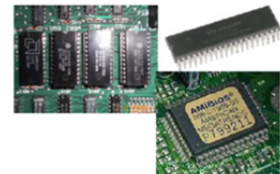
Memoria Principal

La Memoria Principal es un área de almacenamiento interno de la computadora, de acceso rápido, donde se almacenan las instrucciones y los datos que la CPU necesita para ejecutar alguna tarea.

En las computadoras personales son utilizados dos tipos de memorias: **Memoria ROM** y **Memoria RAM**. A continuación se explican cada una de ellas.

Memoria ROM

La memoria ROM viene de las siglas en inglés Read Only Memory, que significa memoria de sólo lectura, es decir, el procesador puede leer la información contenida en ella, mas no puede escribir datos en ella. Esta memoria almacena programas grabados por el fabricante.

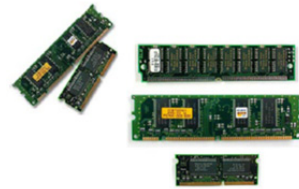


En la memoria ROM residen los programas de arranque de la máquina que le indican al microprocesador qué hacer y cómo evaluar el hardware existente (del que dispone).



Memoria RAM

La memoria RAM viene de las siglas en inglés Random Access Memory, que significa memoria de acceso aleatorio. Ésta es una memoria de acceso rápido donde se encuentran almacenados temporalmente los datos e instrucciones correspondientes a un programa dado que se encuentra en ejecución. Al contrario de la Memoria ROM, ésta memoria permite que se realicen operaciones de escritura y lectura.



La capacidad de la RAM afecta la forma en que se ejecutan los programas y la cantidad de datos que pueden procesarse. Cuanto más fácil de usar sea un programa, tanta más RAM se necesitará, generalmente.

LA RAM es una memoria volátil, su contenido se pierde cuando la computadora se desconecta, excepto en algunas que están provistas de baterías específicamente orientadas a mantener el contenido de la RAM.

Memoria Secundaria

La Memoria Secundaria es un conjunto de componentes electrónicos que almacenan programas y datos de forma permanente. Esta memoria tiene mayor capacidad que la memoria principal, pero es más lenta.



Algunos ejemplos de este tipo de memoria son: Disco Duro, CD, CR-R, CD-RW, Disquette, Memoria Flash, DVD, Disco Zip, etc.

El propósito de estos dispositivos es el de almacenar datos que la computadora no esté usando. El almacenamiento tiene tres ventajas sobre la memoria:

1. Hay más espacio en almacenamiento que en memoria.
2. El almacenamiento retiene su contenido cuando se apaga la computadora.
3. El almacenamiento es más barato que la memoria.

Unidades de Medidas de las Memorias

Bit = Unidad mínima de almacenamiento en una computadora. Cada bit puede representar un valor verdadero (1) o un valor falso(0).
Byte Agrupación de 8 bits, para formar un carácter.

Kilobyte (K)=	1024 Bytes
Megabyte (MG)=	1024 Kilobytes
Gigabyte (GB)=	1024 Megabytes
Terabyte (TB)=	1024 Gigabytes
Petabyte(PB)=	1024 Terabytes
Exabyte (EB)=	1024 Petabytes
Zettabyte (ZB)	1024 Exabyte
Yottabyte (YB)	1024 Zettabyte



Hardware

Se refiere a todos los componentes físicos que intervienen en un sistema computacional. Es todo lo tangible. Algunos ejemplos de hardware son: el disco duro, los disquetes, impresoras, mouse, chips, teclado, monitores, etc.



Software

Se refiere a las instrucciones, datos o programas con los cuales opera la computadora. Todo lo que puede ser almacenado electrónicamente es software. Es todo lo intangible.

Los libros proveen una excelente analogía entre hardware y software. Las páginas y la tinta son el hardware, mientras que las palabras, frases, párrafos y el significado total corresponden al software.

Una computadora sin software es como un libro lleno de hojas vacías. Se necesita del software para hacer la computadora útil, así como se necesitan palabras para hacer un libro significativo.

El software se divide en dos categorías:

- Software de Sistema.- Es el programa que administra y controla la operación del hardware mediante el Sistema Operativo y los programas de utilidad.
- Software de Aplicación.- Es un programa o grupo de programas diseñados para usuarios finales, incluyen programas que manejan bases de datos, procesadores de palabras, hojas de cálculo, gráficos, comunicaciones, etc.



Sistema Operativo

El Sistema Operativo es el programa más importante en un sistema computacional. Los sistemas operativos ejecutan tareas básicas tales como reconocer la entrada de información desde el teclado o cualquier otro dispositivo de entrada, mandar la salida en una pantalla o dispositivo de salida, no perder de vista archivos y carpetas en el disco y controlar dispositivos periféricos tales como unidades de disco e impresoras.



Para sistemas grandes, el Sistema Operativo tiene aún más responsabilidad y poder, se cerciora que los diferentes programas y usuarios que están trabajando al mismo tiempo no interfieran entre si.



El Sistema Operativo es también responsable de la seguridad, verificando que usuarios no autorizados puedan acceder al sistema.

Clasificación de Sistema Operativo

- Multi-usuario.- Permite que dos o más usuarios puedan ejecutar programas al mismo tiempo. Algunos sistemas operativos permiten cientos o aún miles de usuarios.
- Multi-procesamiento.- Permite ejecutar un programa en más de un CPU.
- Multi-tarea.- Permite ejecutar más de un programa. El Sistema Operativo proporciona una plataforma en el cual, el software de aplicación pueda operar. Existen diferentes tipos de



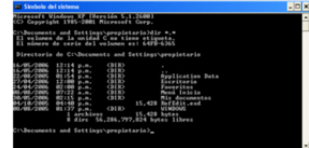
sistemas operativos que pueden ser instaladas en computadoras personales; actualmente los de mayor difusión son: Windows, Unix y Linux.

Tipos de Interfaz

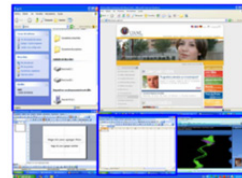
Interfaz es el medio a través del cual dos sistemas independientes que no hablan el mismo lenguaje se puedan comunicar. Es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Existen diferentes tipos de interfaces:

- Interfaz de Línea de Comandos.- Es una manera de interactuar con la computadora en el cual los comandos son introducidos como textos (secuencia de caracteres) a través del teclado, y la salida es recibida también como texto.



- Interfaz Gráfica de Usuario (GUI).- Es una manera de interactuar con la computadora a través de la manipulación directa de imágenes gráficas. GUI despliega elementos visuales tales como íconos, ventanas, menús en los cuales el usuario solamente va seleccionando dando un clic o doble clic, dependiendo de sus necesidades.





CAPITULO II.

Conceptos de Programación

Objetivo: El Alumno conocerá e identificará los conceptos básicos de programación y los aplicará en la creación de algoritmos.

Lenguaje

Es una serie de símbolos que sirven para transmitir uno o más mensajes (ideas) entre dos entidades diferentes. A la transmisión de mensajes se le conoce comúnmente como comunicación.

La comunicación es un proceso complejo que requiere una serie de reglas simples, pero indispensables para poderse llevar a cabo. Las dos principales son las siguientes:

1. Los mensajes deben correr en un sentido a la vez.
2. Debe forzosamente existir 4 elementos: Emisor, Receptor, Medio de Comunicación y Mensaje.

Tipos de lenguaje:

- **El lenguaje humano** se basa en la capacidad de los seres humanos para comunicarse por medio de signos. Principalmente lo hacemos utilizando el signo lingüístico. Aun así, hay diversos tipos de lenguaje. El lenguaje humano puede estudiarse en cuanto a su desarrollo desde dos puntos de vista complementarios: la ontogenia, que remite al proceso de adquisición del lenguaje por el ser humano, y la filogenia
- **El lenguaje animal** se basa en el uso de señales sonoras, visuales y olfativas a modo de signos para referirse a un referente o un significado diferente de dichas señales. Dentro del lenguaje animal están los gritos de alarma, el lenguaje de las abejas, etc.
- **Los lenguajes formales** son construcciones artificiales humanas, que se usan en matemática y otras disciplinas formales, incluyendo lenguajes de programación. Estas construcciones tienen estructuras internas que comparten con el lenguaje humano natural, por lo que pueden ser en parte analizados con los mismos conceptos que éste.

Lenguaje de programación

Es un conjunto de símbolos, caracteres y reglas (programas) que les permiten a las personas comunicarse con la computadora.

Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, cálculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación. Se clasifican en:

- **Lenguaje Maquina:** Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje maquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 o 1).
- **Lenguaje de Bajo Nivel (Ensamblador):** En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Los lenguajes de programación de alto nivel (C#, C++, Visual Basic, Java, etc.) son aquellos en los que las instrucciones o sentencias a la computadora



son escritas con palabras similares a los lenguajes humanos (en general en inglés), lo que facilita la escritura y comprensión del programa.

Programa

Un programa, o también llamado programa informático, programa de computación o programa de computadora, es simplemente un conjunto de instrucciones para una computadora. Las computadoras necesitan de los programas para funcionar, y un programa no hace nada a menos que sus instrucciones sean ejecutadas por el procesador. Un programa se puede referir tanto a un programa ejecutable como a su código fuente, el cual es transformado en un ejecutable cuando es compilado.

Generalmente el código fuente de los programas es escrito por profesionales conocidos como programadores. El código fuente es escrito en un lenguaje de programación que sigue uno de los siguientes dos paradigmas: imperativo o declarativo. El código fuente puede ser convertido en una imagen ejecutable por un compilador. Cuando se pide que el programa sea ejecutado, el procesador ejecuta el programa instrucción por instrucción, hasta que el programa termina.

De acuerdo a sus funciones, los programas pueden ser clasificados en software de sistema y software de aplicación. Pueden ejecutarse muchos programas de forma simultánea en una misma computadora, a lo cual se le llama multitarea.

Algoritmo

Un algoritmo (del latín, dixit algorithmus y éste a su vez del matemático persa al-Jwarizmi) es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

En la vida cotidiana se emplean algoritmos en multitud de ocasiones para resolver diversos problemas. Algunos ejemplos se encuentran en los instructivos (manuales de usuario), los cuales muestran algoritmos para usar el aparato en cuestión o inclusive en las instrucciones que recibe un trabajador por parte de su patrón. También existen ejemplos de índole matemática, como el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para calcular el máximo común divisor de dos enteros positivos, o el método de Gauss para resolver un Sistema lineal de ecuaciones.

Existen dos tipos de algoritmos que son:

- **Cualitativos:** Son aquellos en los que se describen los pasos utilizando palabras. Como ejemplo podemos decir que la utilización de un directorio (Búsqueda de un teléfono). Para poder buscar un teléfono en un directorio, se debe conocer el algoritmo que se va a utilizar, es decir la forma en que están codificados los nombres de las personas, para así lograr encontrarlos y localizar el número telefónico correspondiente.
- **Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. Como ejemplo podemos citar los pasos para resolver una ecuación de segundo grado, los pasos lógicos y secuenciales para obtener una nomina, entre otros.



Ejemplo de algoritmo

Objetivo: Algoritmo para cambiar un foco quemado:

Pasos a Seguir:

1. Situar escalera bajo el foco quemado.
2. Elegir un foco de reemplazo (de la misma potencia que el anterior).
3. Subir por la escalera hasta alcanzar el foco.
4. Girar el foco contra las manecillas del reloj hasta que esté suelto.
5. Ubicar el foco nuevo en el mismo lugar que el anterior.
6. Enroscar en el sentido de las manecillas del reloj hasta que quede apretado.
7. Bajar de la escalera.

Lenguajes Algorítmicos

Son una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Los tipos de Lenguajes Algorítmicos son:

Gráficos: Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).

No Gráficos: Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

Pseudocódigo	Diagrama de Flujo (Flowchart)
<p>Su estructura es similar a la de un programa. Es un punto intermedio entre el lenguaje natural y el lenguaje de alto nivel:</p> <p>Algoritmo SuperficieCircunferencia</p> <p>INICIO</p> <p>CONST PI = 3.1416</p> <p>ENTERO Radio, Superficie</p> <p>LEER Radio</p> <p>Superficie = PI*Radio*Radio</p> <p>IMPRIMIR "La superficie de la circunferencia de radio ",Radio, "es igual a ", Superficie</p> <p>FIN</p>	<p>Indica el orden en que deben ejecutarse las acciones. Ofrece mucha claridad y facilidad para diseñar el algoritmo pero a la hora de realizar tareas de actualización dificulta el proceso.</p> <pre> graph TD INICIO([INICIO]) --> LEER[/LEER (Radio)/] LEER --> CALC[Superficie=Radio*Radio*3.14] CALC --> IMPRIMIR[IMPRIMIR "La superficie de la circunferencia de radio ",Radio, "es igual a ", Superficie] IMPRIMIR --> FIN([FIN]) </pre>

Práctica y actividad para el alumno

1. Realizar la elaboración, presentación y defensa, de al menos dos algoritmos utilizados para solucionar problemas, basados en el ejemplo del foco quemado.



CAPITULO III.

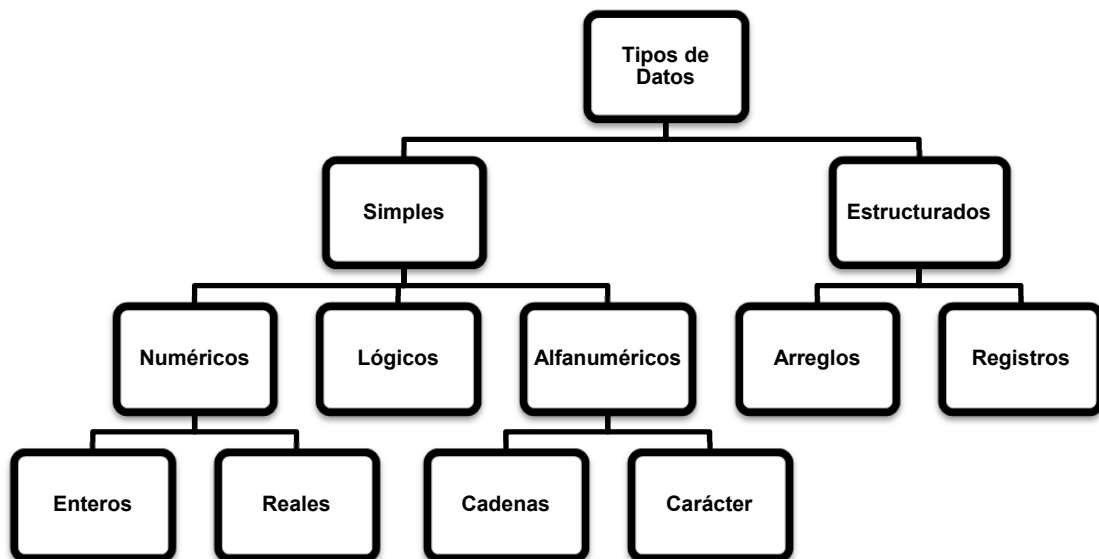
Entidades primitivas para el desarrollo de algoritmos

Objetivo: El alumno conocerá las reglas para desarrollar habilidades que le permitan cambiar fórmulas matemáticas a expresiones válidas para la computadora, además de diferenciar y aplicar correctamente la utilización de constantes, variables, tipos de datos, entre otros elementos que son fundamentales para la construcción de algoritmos.

Tipos de Datos

Para guardar datos internamente y representar cualquier situación en la computadora, es necesario que los datos tengan un tipo de dato asociado con ellos. Mediante un dato se puede representar un simple carácter, tal como 'g', el nombre de una persona, tal como "Geraldynne", un valor entero, tal como 5, etc.

Los tipos de datos determinan la naturaleza del conjunto de valores que puede tomar una variable y se clasifican en:



Tipos de Datos Simples

La principal característica de los datos simples es que hacen referencia a un único valor a la vez, es decir, una variable almacena un único elemento (valor). Dentro de este grupo se encuentran:

- **Datos Numéricos:** Permiten representar valores escalares de forma numérica, así como también, realizar operaciones aritméticas comunes tales como: multiplicar dividir, sumar y restar. En el ámbito computacional, este tipo incluye a:



ENTEROS Son los números que no contienen componentes fraccionarios y, por tanto, no incluyen el punto decimal; pueden ser positivos o negativos.

Ejemplo:

- 23, -345, 9, o -9.
- Número de autos vendidos en un día
- Número de pacientes con cita para un día determinado
- Número de días trabajador por un empleado

REALES Son los números que contienen una parte fraccionaria y, por tanto, incluyen el punto decimal; pueden ser positivos o negativos.

Ejemplo:

- 3.14,-45.7, 6.7 o -6.7
 - El sueldo de un trabajador
 - El precio de venta de un producto
 - Cantidad de queso a comprar
- **Datos Alfanuméricos:** Este tipo de dato puede almacenar letras (A-Z, a-z), dígitos (0-9) y caracteres especiales (\$, #, _, entre otros¹). Este tipo incluye a:

CADENAS Es una secuencia de caracteres alfanuméricos, que permiten representar valores descriptivos, esto incluye a:

- Nombres de personas o de empresas
- Dirección de personas o empresas
- Descripción de artículos
- Nombre de artículos
- Nombre de un país
- La marca de una computadora, entre otros.

Este tipo de datos se representan encerrados entre comillas (""). Por ejemplo:

- "Facultad de Informática"
- "2008"
- "Culiacán Rosales Sinaloa"
- "Nissan 200sx 1.6lts"

Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir, no es posible hacer operaciones aritméticas con ellos. Por ejemplo:

- Número telefónico de personas o empresas
- El código postal de una dirección
- El número de cuenta de una alumno de la UAS
- El número de cuenta de una tarjeta de crédito

¹ Para ver la lista completa del código ASCII, ir a ANEXO A



CARÁCTER Es un tipo de que permite almacenar sólo un símbolo alfanumérico, es decir, almacena *un sólo carácter* válido para la computadora según el código ASCII. Normalmente se encierra entre apostrofes ('). Almacenan un solo carácter que puede ser una mayúscula o minúscula, un dígito o caracteres especiales (como el signo @,&! entre otros).

Por ejemplo:

- 'a', 'B', '1', '#'
 - El género de una persona ('F' para Femenino o 'M' para Masculino)
 - El estado civil de una persona ('S' para Soltero o 'C' para Casado)
 - El tipo de gobierno ('M' para Municipal, 'E' para Estatal o 'F' para Federal).
- **Datos Lógicos:** Son aquellos que sólo pueden almacenar uno de dos posibles valores False (falso) y True (verdadero), ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).

Ejemplos:

Comparación	Resultado (dato lógico)
¿5 es mayor que 4?	True
¿"Rogelio" es igual a "Rogelio"?	True
¿"Informatica" es igual a "Informática"?	False
¿"Ana" es igual a "Fernanda"?	False

Las **palabras reservadas** son aquellas palabras que tienen un significado especial en el pseudocódigo o lenguaje de programación que se utilice. No se pueden utilizar con otro propósito.

Las palabras reservadas, que se utilizarán, en este material, para representar a cada tipo de dato descrito anteriormente son:

Tipo de dato	Palabra reservada para su representación
Numérico Entero	ENTERO
Numérico Real	REAL
Lógico	BOOLEAN
Alfanumérico (más de un carácter)	CADENA
Alfanumérico (un sólo carácter)	CARACTER

Tipos de Datos Estructurados

Los datos estructurados son aquellos que se caracterizan por el hecho de que con un sólo identificador, se hace referencia a un grupo de componentes que pueden ser a su vez un dato simple o estructurado, entre estos se encuentran:



Arreglos Se forma por un conjunto de elementos de un mismo tipo de dato simple. Pueden almacenar a más de un elemento (valor) a la vez, bajo un mismo identificador, que se diferencian por la posición que tienen cada elemento dentro del arreglo de datos.

Registros. Es un grupo de elementos de datos agrupados bajo un nombre. Estos elementos de datos, conocidos como miembros, pueden tener diferentes tipos.

Variables

Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa.

Las variables sirven para representar y manejar datos. Todo dato que vaya a ser introducido a la computadora o que vaya a ser generado, calculado a partir de otros datos para obtener algún resultado debe identificarse y manejarse en forma de variable.

Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

La sintaxis² para su definición es:

```
TipoDato Identificador ó TipoDato Identificador = Valor
```

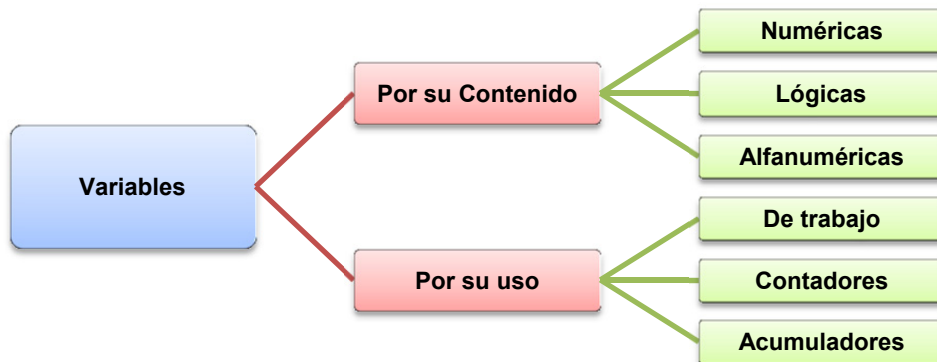
Donde :

TipoDato	Indica el tipo de dato que tendrá la variable.
Identificador	Es el nombre de la variable.
Valor	Es el valor que toma la variable, en caso de inicializarla (opcional).

Ejemplo:

- CADENA NombreAlumno
- ENTERO TotalVentas = 0
- ENTERO TotalVentas

Las variables se clasifican en:



² Sintaxis es la forma correcta (orden) de escribir una instrucción en un pseudocódigo o lenguaje de programación.



La clasificación de las **variables por su contenido** ya se ha explicado en la sección “Tipo de Datos” al inicio de este capítulo.

Las variables por su uso son las siguientes:

Variabes de Trabajo. Son aquellas variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.

Ejemplo:

- `Suma = Calificacion1 + Calificacion2 + Calificacion3`
- `Promedio = Suma / 3`

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

Ejemplo:

- `TotalAlumnosOjosAzules=0`, y su cumplimiento en cada ocasión
- `TotalAlumnosOjosAzules = TotalAlumnosOjosAzules + 1`

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.

Ejemplo:

- `TotalVenta=0`, y su cumplimiento en cada ocasión
- `TotalVenta = TotalVenta + PrecioProducto`

Constantes

Una constante es un dato de tipo numérico, lógico o alfanumérico, almacenado en la memoria de la computadora, que no cambia durante la ejecución del programa, es decir, son valores fijos, en consecuencia invariables.

La sintaxis utilizada para su definición es:

```
CONST TipoDato Identificador = Valor
```

Donde:

CONST	Identifica la acción que se va a realizar, la cual es definición de constantes simbólicas
TipoDato	Indica el tipo de Dato que tendrá la constante.
Identificador	Es el nombre de la constante
Valor	Es el valor que toma la constante

Ejemplo:

- `CONST REAL PI = 3.1416`
- `CONST CADENA NombreFacultad = "Facultad de Informática"`
- `CONST BOOLEAN Resultado=True`
- `CONST ENTERO Porcentaje = 30`

NOTA: Una definición incorrecta sería omitir el valor que se debe asignar a la constante, por ejemplo: `CONST ENTERO Porcentaje`



Identificadores

Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria de la computadora, que nos permite acceder a su contenido. Además, los identificadores representan los datos de un programa (constantes, variables, tipos de datos, entre otros) que van a ser calculados o generados a partir de otros datos para obtener algún resultado.

Ejemplo:

- NombreAlumno
- NumeroHoras
- Calificacion

Las reglas y recomendaciones para nombrar identificadores son:

Reglas	Recomendaciones
Debe comenzar con una letra (A a Z, mayúsculas o minúsculas).	Evite poner todas las letras en mayúsculas o minúsculas
Los demás caracteres pueden ser letras (a, b, c, ... ,z), el signo de subrayado (_) o dígitos (0, 1, 2, ... ,9).	Evite empezar con un signo de subrayado Evite el uso de abreviaturas
No deben contener espacios en blanco.	Use PascalCasing para nombres con varias palabras: inicial de cada palabra en mayúsculas sin espacio entre palabras. Debe definirse de tal manera que indique lo que está representando.

Ejemplos:

Identificador	Resultado	Cumple con las reglas y recomendaciones
NombreAlumno	Correcto	SI
1Telefono	Incorrecto	No, debido a que inicia con número
SuelTrab	Incorrecto	No, debido a que es una abreviación
Respuesta12	Correcto	SI
IVA	Correcto	SI, debido a que es una abreviación estandarizada
Calificación Final	Incorrecto	No, debido a que incluye un espacio en blanco y acento

Lectura opcional: ¿Es importante usar un estándar para elegir identificadores de variables?

Vídeo del tema en <http://www.milibreta.com.mx/programacion>



Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales, quienes permiten modificar la apariencia de los datos y generar información, es decir, procesar los datos que entran como materia prima, para convertirlos en información que saldrá de la computadora.

La sintaxis para representar una expresión es:

`Variable = expresión`

Donde:

Variable es el nombre de la variable en la que se asignara el resultado de la expresión.

Expresión es un valor constante, variable o una expresión algebraica, lógica o relacional, el cual se le asigna a la variable.

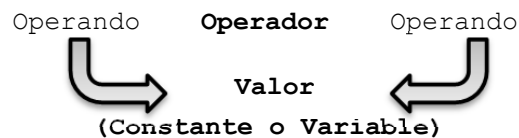
= es el símbolo que indica la asignación del valor de la expresión a la variable.

Por ejemplo:

Tipo de Expresión	Ejemplo
Aritmética	$X1 = A + (B + 3) / C$
Relacional	$X2 = A > B$
Lógica	$X3 = (A > 5) \text{ AND } (B < 10)$

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 y operandos, según sea el tipo de dato que manipulan, y debe tener el siguiente formato:



¿Cómo saber qué tipo de expresión es?

- Aritméticas, si sus operadores son aritméticos y el resultado de la evaluación es numérico.
- Lógicas, si sus operadores son lógicos y el resultado de la evaluación es BOOLEAN (True o False).
- Relacionales, si sus operadores son relacionales y el resultado de la evaluación es BOOLEAN (True o False).
- Mixta, si se combinan los tipos de expresiones anteriores para su construcción.

Operadores y Operandos

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.

Tipos de Operadores:



- Aritméticos
- Lógicos, y
- Relacionales

Operandos son números constantes o variables. El resultado de una operación aritmética será un número.

Operadores Aritméticos

Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes). El resultado de una operación aritmética será un dato tipo numérico.

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales, para formar expresiones aritméticas en la realización de cálculos. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real. Los operadores aritméticos para las operaciones básicas son:

Símbolo	Operación
*	Multiplicación
/	División
MOD	Módulo (residuo de la división entera)
+	Suma
-	Resta

También, se encuentran los operadores aritméticos unarios, cuyo objetivo es afectar a un sólo operando como lo es el cambio de signo, estos símbolos son los utilizados en las operaciones de suma y resta.

Ejemplo:

- $X1 = -A$
- $X2 = +B \quad x1 = (A + -B) * (-C + -D)$

Funciones Matemáticas

Cada lenguaje de programación tiene su conjunto de funciones matemáticas predefinidas, las cuales se ejecutan haciendo referencia a su nombre. Algunas necesitan, para arrojar un resultado, que se suministre información adicional (parámetros o argumentos). Las funciones matemáticas más utilizadas son:

Función matemática	Significado
POTENCIA POW (Base, Exponente)	Devuelve el resultado de elevar la Base al Exponente indicado.
RAÍZ CUADRADA SQRT (Valor o Dato de Entrada)	Devuelve la raíz cuadrada de su entrada

Ejemplo:

- $X1 = \text{POW} (10, 2)$
- $X2 = 5 * 3.1416 * \text{POW}(\text{Radio}, 2)$

Operadores Relacionales



Sirven para la elaboración de expresiones lógicas mediante las cuales se pueden plantear condiciones o comparaciones y que dan como resultado un valor booleano (True, False).

Los operadores relacionales son:

Símbolo	Operación
>	Mayor que
<	Menor que
> =	Mayor o igual que
< =	Menor o igual que
< >	Diferente
==	Igual (igualdad lógica)

Las expresiones lógicas simples se forman relacionando variables, constantes, subexpresiones aritméticas, mediante algunos de los operadores relacionales.

Operadores Lógicos

Estos operadores se utilizan para formar expresiones lógicas complejas. Se establecen relaciones entre valores lógicos y sus operandos pueden ser resultado de una expresión relacional. Los operadores lógicos son:

Operador	Significado
AND	Conjunción
OR	Disyunción
NOT	Negación

La tabla de verdad de los operadores lógicos es:

Expresion1	Expresion2	AND	OR	NOT Expresion1
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

Operador de Asignación

Se utiliza para asignar valores o expresiones a una localidad de memoria (variable). Es una operación destructiva debido a que si la variable tenía asignado un valor anteriormente, éste se destruye, conservando ahora el nuevo valor.

Su sintaxis es:

```
Variable = expresión o valor
```

Donde:



Variable Es el nombre o identificador de la variable o constante que recibirá el valor.

Expresión puede ser el resultado de una expresión aritmética, lógica, relacional o mixta, la asignación del valor de una variable o constante.

Ejemplo:

- $X = A + (B - 5)$
- $TotalVenta = 100$
- $TotalVenta = Precio * 2$
- $Aprobado = (Calificacion \geq 6) \text{ AND } (Calificacion \leq 10)$
- $Resultado2 = Resultado$

Prioridad de los Operadores en General

La computadora al evaluar una expresión aritmética, lógica, relacional o mixta, primero la analiza completamente, y va evaluando de izquierda a derecha cada componente de acuerdo al orden de precedencia que tienen los operadores, el cual es el siguiente:

Nivel de prioridad	Operadores
1	() asociación
2	Operaciones unarias (cambio de signo)
3	*, /, MOD, NOT
4	+, -, AND
5	>, <, >=, <=, <>, ==, OR

Por lo anterior, al elaborar una expresión, debemos respetar la jerarquía del orden de aplicación, ya que:

- Todas las expresiones entre paréntesis se evalúan primero.
- Las expresiones con paréntesis anidados se evalúan de dentro a fuera, el paréntesis más interno se evalúa primero.
- Los operadores en una misma expresión con igual nivel de prioridad se evalúan de izquierda a derecha.
- Si al menos uno de los operandos de una expresión es una función matemática, primero se evalúa la función y después el operador.

Actividades Prácticas

Especifique el tipo de dato que ocuparía para almacenar en la computadora los siguientes datos:

- El código Postal
- El total de frutas de un árbol
- El género (sexo) de una persona
- El teléfono de una empresa
- Estado civil de una persona
- El pago por hora de un obrero
- El precio de venta unitario de un producto



- Si una persona cuenta con tarjeta de crédito del banco X
- El número de placas de un automóvil
- El consumo en Kilowatts de una casa habitación

De la siguiente lista de Expresiones, indique la causa porque son incorrectas, en caso de serlo:

- $X1 = A + B * C -$
- $X2 = (A + B) * C)$
- $X3 = (A+) B * C$
- $X4 = A + -B * C + 5$
- $X5 = A+ * -B / 2$
- $X6 = Parcial1 + Parcial2 + Parcial3 / 3$

Indique el orden en la evaluación y el resultado que obtendría la computadora, de las siguientes expresiones, además indique el tipo de dato del resultado:

2. $X1 = 7 + 5 - 6$
3. $X2 = 7 - 5 + 6$
4. $X3 = 9 + 7 * 8 - 36 / 5$
5. $X3 = 7 * 8 * (160 \text{ MOD } 3) / 5 * 13 - 28$
6. $X5 = (3 * 4) + (4 - 2) / (9 * (5 - 3))$

Para las siguientes expresiones tome en cuenta los valores de las siguientes variables:
A= 5 y B=16.

7. $X1 = (A * 2) > (B + 6 * 2)$
8. $X2 = (A * 5 + 45.78 / 4) <= (B * 3 / A)$
9. $X3 = \text{NOT}(15 >= 7 * 2) \text{ OR } (43 - 8 * 2 / 4 <> 3 * 2 / 2)$

Convertir las siguientes formulas a expresiones válidas por la computadora

$$X1 = 3xy^2 - 2x^2y$$

$$X2 = \frac{6+18}{2}$$

$$X3 = \frac{\frac{a}{b} + b}{x - \frac{y}{z}}$$

$$X4 = \sqrt{a^2 + b^2} \quad \text{utilizar la función SQRT()}$$



$$X5 = \left(\frac{a+b}{c+d} \right)^3 \quad \text{utilizar la función POW(Base, Exponente)}$$

$$X6 = 4x + 5y / 2z^2$$

$$X7 = \pi \bullet r^2$$

$$X8 = \pi r^2 h$$

$$X9 = \frac{x^2}{a^2} - \frac{y^2}{b^2}$$

$$X10 = 5x^5 + 7x^4 - 5x^3 + 3x^2 - 8x + 4$$



CAPITULO IV.

Metodología para programación

Objetivo: El alumno:

Conocerá la importancia de aplicar una metodología adecuada en el proceso de la enseñanza de la programación de computadoras.

Conocerá una metodología aplicada para enseñar a programar computadoras en forma estructurada y desarrollará habilidades que podrá utilizar en cada una de sus etapas.

Importancia de la aplicación de técnicas y herramientas metodológicas.

¿Cómo enseñar a programar computadoras?, ¿Cómo enseñar a desarrollar la lógica necesaria para desarrollar problemas independientes de algún lenguaje de programación?, ¿Podemos enseñar a resolver un problema?, ¿Podemos enseñar a analizar el mismo? Y ¿Podemos enseñar a pensar?

La computadora es una máquina que puede realizar procesos para darnos resultados, sin que tengamos la noción exacta de las operaciones que realiza para llegar a esos resultados.

También podemos diseñar soluciones a la medida, de problemas específicos que se nos presenten. Más aún, si estos involucran operaciones matemáticas complejas y/o repetitivas, o requieren del manejo de un volumen muy grande de datos.

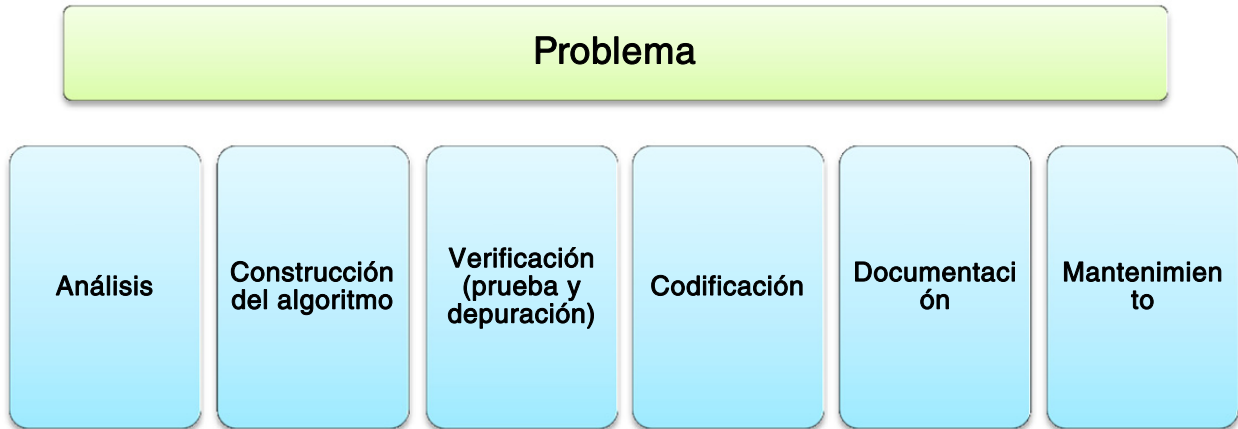
El diseño de soluciones a la medida de nuestros problemas, requiere como en otras disciplinas una metodología que nos enseñe de manera gradual, la forma de llegar a estas soluciones.

La aplicación de técnicas y herramientas metodológicas permiten enseñar a programar computadoras utilizando un tipo de algoritmo, es decir, sin emplear la computadora. Además, permiten desarrollar las capacidades mentales que una persona debe tener para programar computadoras y sienta las bases de disciplina y buena estructura.

Programar es una actividad intelectual que requiere: Mucha creatividad, Capacidad de abstracción, Capacidad de análisis y síntesis. Se desarrollan ejercitando la mente en forma apropiada.

Etapas en la solución de un problema

Tomando como referencia a diversos autores como son: Osvaldo Cairo, Metodología de la programación, Leonardo López Román, Metodología de la programación orientada a objetos, L. Joyanes Aguilar, Algoritmos, Estructuras de Datos y Objetos, entre otros, se obtienen las siguientes etapas a seguir para buscar la solución de un problema:



Problema

Etimológicamente, la palabra problema deriva del griego proballein y significa “algo lanzado hacia delante”. Un problema es un asunto o un conjunto de cuestiones que se plantean para ser resueltas. También puede definirse como planteamiento de una situación cuya respuesta desconocida debe obtenerse a través de métodos científicos.

La naturaleza de los problemas varía con el ámbito o con el contexto donde están planteados; así, existen problemas: Matemáticos, Químicos, Filosóficos, etc.

Se consideran aquellos problemas cuya solución se puede calcular utilizando una serie de reglas introducidas en una computadora. Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa.

Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se de del todo, no tiene mucho caso continuar con la siguiente etapa.

A través del tiempo, la humanidad ha utilizado diversas estrategias generales para resolver problemas. Schunk (1997), Woolfolk (1999) y otros, destacan los siguientes métodos o estrategias de tipo general:

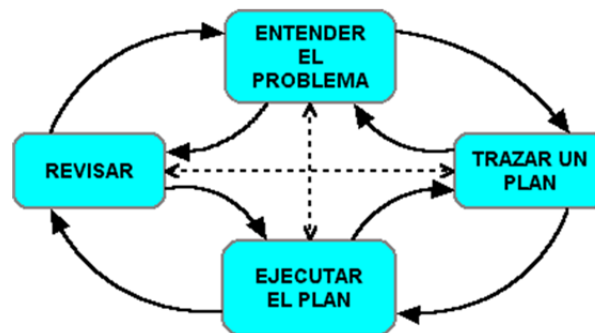
1. **Ensayo y error:** Consiste en actuar hasta que algo funcione. Puede tomar mucho tiempo y no es seguro que se llegue a una solución. Es una estrategia apropiada cuando las soluciones posibles son pocas y se pueden probar todas, empezando por la que ofrece mayor probabilidad de resolver el problema. Por ejemplo, un foco que no prende: revisar el foco, verificar la corriente eléctrica, verificar el interruptor.
2. **Heurística:** Se basa en la utilización de reglas empíricas para llegar a una solución. El método heurístico conocido como “IDEAL”, formulado por Bransford y Stein (1984), incluye cinco pasos:
 - Identificar el problema;
 - definir y presentar el problema;
 - explorar las estrategias viables;
 - avanzar en las estrategias; y
 - lograr la solución y volver para evaluar los efectos de las actividades (Bransford & Stein, 1984).



3. **Algoritmos:** Consiste en aplicar adecuadamente una serie de pasos detallados que aseguran una solución correcta. Por lo general, cada algoritmo es específico de un dominio del conocimiento. La programación de computadores se apoya en este método.
4. **Lluvia de ideas:** Consiste en formular soluciones viables a un problema. El modelo propuesto por Mayer (1992) plantea: definir el problema; generar muchas soluciones (sin evaluarlas); decidir los criterios para estimar las soluciones generadas; y emplear esos criterios para seleccionar la mejor solución. Requiere que no se emitan juicios con respecto a las posibles soluciones hasta que se terminen de formularlas.
5. **Pensamiento lateral:** Se apoya en el pensamiento creativo, formulado por Edwar de Bono, el cual difiere completamente del pensamiento lineal (lógico). El pensamiento lateral requiere que se exploren y consideren la mayor cantidad posible de alternativas para solucionar un problema. Su importancia radica en permitir que se explore (escuche y acepte puntos de vista diferentes, busque alternativas); avive (promueva el uso de la fantasía y del humor); libere (use la discontinuidad y escape de ideas preestablecidas); y contrarreste la rigidez (vea las cosas desde diferentes ángulos y evite dogmatismos).

Como se puede apreciar, hay muchas estrategias para solucionar problemas. Según Polya (1957), cuando se resuelven problemas, intervienen cuatro operaciones mentales:

1. Entender el problema
2. Trazar un plan
3. Ejecutar el plan (resolver)
4. Revisar



Análisis del problema

Esta etapa se inicia una vez que se ha comprendido la tarea que se desea de la computadora haga, en ella se define los siguientes elementos:

- Precisar los resultados esperados.- Es la información que se desea producir (**datos o información de salida**) debe preguntarse:
 - ¿Qué información me solicitan?
 - ¿Qué formato debe tener esta información?
- Identificar los datos disponibles (**Datos Conocidos**).- Se debe preguntarse:
 - ¿Qué información es importante?
 - ¿Qué información no es relevante?
 - ¿Cuáles son los datos de entrada? (conocidos)
 - ¿Cuál es la incógnita?
 - ¿Qué información me falta para resolver el problema? (datos desconocidos)



- ¿Puedo agrupar los datos en categorías?
- **Los Datos de Entrada.**-Son los datos no conocidos y requeridos para la solución del problema.
- **Establecer el proceso** (métodos y fórmulas que se necesitan para procesar los datos y obtener la salida).- Consiste en determinar los procesos que permiten llegar a los resultados esperados a partir de los datos disponibles. Se debe preguntarse:
 1. ¿Qué procesos necesito?
 2. ¿Qué fórmulas debo emplear?
 3. ¿Cómo afectan las condiciones a los procesos?
 4. ¿Qué debo hacer?
 5. ¿Cuál es el orden de lo que debo hacer?
- Determinar cuáles son las **restricciones establecidas.**- Consiste en determinar aquello que está permitido o prohibido hacer y/o utilizar para llegar a una solución. Debe preguntarse:
 - ¿Qué condiciones me plantea el problema?
 - ¿Qué está prohibido hacer y/o utilizar?
 - ¿Qué está permitido hacer y/o utilizar?
 - ¿Cuáles datos puedo considerar fijos (constantes) para simplificar el problema?
 - ¿Cuáles datos son variables?
 - ¿Cuáles datos debo calcular?

Una recomendación muy practica 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.

Construcción del algoritmo en Pseudocódigo

Luego de analizar detalladamente el problema hasta entenderlo completamente, se procede a diseñar un algoritmo (trazar un plan) que lo resuelva por medio de pasos sucesivos y organizados en secuencia lógica.

El algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico. Las características de un buen algoritmo son:

- Debe tener un punto particular de inicio
- Debe ser definido, no debe permitir dobles interpretaciones
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema
- Debe ser finito en tamaño y tiempo de ejecución

Considere que el algoritmo que se construye como pseudocódigo, no es un lenguaje formal y puede variar de un programador a otro, es decir, no hay una estructura semántica ni arquitectura estándar, por lo que es recomendado buscar una homogenización en un equipo de desarrollo de software. Por lo anterior se propone la utilización de la siguiente estructura:



Estructura de un programa en pseudocódigo

Objetivo:

Programador:

Fecha:

INICIO

Declaración de Nuevos Tipos de Datos

Definición de Constantes y Variables

//Instrucciones

Lecturas de Datos de Entrada

Procesamiento de los Datos

Impresión de Resultados

FIN

Verificación (prueba y depuración)

Esta etapa es el proceso de identificar y eliminar errores, para dar paso a una solución sin errores. Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. Resulta una tarea tan creativa como el mismo desarrollo de la solución, por ello se debe considerar con el mismo interés y entusiasmo.

Se debe realizar la verificación de lo que sucede tanto en la memoria de la computadora, su comportamiento ante la realización de cada instrucción, así como lo que se muestra en la pantalla de la misma.

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.

Documentación

Es la guía o comunicación escrita en sus variadas formas, ya sea en enunciados, procedimientos, dibujos o diagramas. A menudo un programa escrito por una persona, es usado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).



La documentación se divide en:

- **Documentación Interna:** Son los comentarios o mensaje que se añaden al código fuente para hacer más claro el entendimiento de un proceso.
- **Documentación Externa:** Se define en un documento escrito los siguientes puntos:
 - Descripción del Problema
 - Nombre del Autor
 - Algoritmo (diagrama de flujo o pseudocódigo)
 - Diccionario de Datos
 - Código Fuente (programa)
- **Manual del Usuario:** Describe paso a paso la manera como funciona el programa, con el fin de que el usuario obtenga el resultado deseado.

Mantenimiento

Se lleva a cabo después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa este correctamente documentado.

Análisis de algoritmos

Como medida de la eficiencia de un algoritmo, se suelen estudiar los recursos (memoria y tiempo) que consume el algoritmo. El análisis de algoritmos se ha desarrollado para obtener valores que de alguna forma indiquen (o especifiquen) la evolución del gasto de tiempo y memoria en función del tamaño de los valores de entrada.

El análisis y estudio de los algoritmos es una disciplina de las ciencias de la computación y, en la mayoría de los casos, su estudio es completamente abstracto sin usar ningún tipo de lenguaje de programación ni cualquier otra implementación; por eso, en ese sentido, comparte las características de las disciplinas matemáticas. Así, el análisis de los algoritmos se centra en los principios básicos del algoritmo, no en los de la implementación particular. Una forma de plasmar (o algunas veces "codificar") un algoritmo es escribirlo en pseudocódigo o utilizar un lenguaje muy simple tal como Lógico, cuyos códigos pueden estar en el idioma del programador.

Algunos escritores restringen la definición de algoritmo a procedimientos que deben acabar en algún momento, mientras que otros consideran procedimientos que podrían ejecutarse eternamente sin pararse, suponiendo el caso en el que existiera algún dispositivo físico que fuera capaz de funcionar eternamente. En este último caso, la finalización con éxito del algoritmo no se podría definir como la terminación de éste con una salida satisfactoria, sino que el éxito estaría definido en función de las secuencias de salidas dadas durante un periodo de vida de la ejecución del algoritmo. Por ejemplo, un algoritmo que verifica que hay más ceros que unos en una secuencia binaria infinita debe ejecutarse siempre para que pueda devolver un valor útil. Si se implementa correctamente, el valor devuelto por el algoritmo será válido, hasta que evalúe el siguiente dígito binario. De esta forma, mientras evalúa la siguiente secuencia podrán leerse dos tipos de señales: una señal positiva (en el caso de que el número de ceros sea mayor que el de unos) y una negativa en caso contrario. Finalmente, la salida de este algoritmo se define como la devolución de valores exclusivamente positivos si hay más ceros que unos en la secuencia y, en cualquier otro caso, devolverá una mezcla de señales positivas y negativas.



CAPITULO V.

Estructuras Algorítmicas Básicas

Objetivo.- Conocerá las diferentes estructuras algorítmicas selectivas como componentes básicos de los programas y aplicará la combinación de ellas para el desarrollo de algoritmos más complejos.

Estructuras Básicas

Un Algoritmo está compuesto por instrucciones de diferentes tipos, organizadas secuencialmente, en forma de estructuras de control. De estas estructuras, las más comunes son las siguientes:

1. Secuencial
2. Condicional (decisión, selección)
3. Iterativa (repetición)

Una estructura de control se define como un esquema que permite representar ideas de manera simplificada y que bajo condiciones normales, es constante (Trejos, 1999).

El uso del diseño descendente en los programas, la ejecución de operaciones secuenciales, la utilización de ciclos repetitivos y, la toma de decisiones y alternativas de proceso, ofrecen amplias posibilidades para resolver problemas mediante la construcción de procedimientos (Castellanos & Ferreyra, 2000b).

Estructura Secuencial

Una estructura se define como un esquema con cierta distribución y orden que permite representar una idea de forma simplificada y que bajo ciertas condiciones es constante (Trejos, 1999).

La estructura de control secuencial es la más sencilla. También se la conoce como estructura lineal. Se compone de instrucciones que deben ejecutarse en forma consecutiva, una tras otra, siguiendo una línea de flujo. Solamente los problemas muy sencillos pueden resolverse haciendo uso únicamente de esta estructura. Normalmente, la estructura secuencial hace parte de soluciones a problemas complejos en las que se la utiliza mezclada con estructuras iterativas (repetir varias veces un conjunto de instrucciones) y condicionales (tomar decisiones).

El cuerpo de la estructura de un programa en pseudocódigo que implemente la estructura secuencia es la siguiente:

```
INICIO
    //definición, declaración e inicialización de variables y
    constantes
    Acción 1 (Instrucciones)
    Acción 2
    Acción 3

    Acción N
FIN
```



Instrucciones de Entrada/Salida

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

```
LEER A, B
```

Donde: "A" y "B" son las variables que recibirán los valores

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

```
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.

Caso práctico utilizando estructura secuencial

Paso 1.- Definición del Problema: La Huerta Mucha Fruta cuenta con un sembrado de N árboles de Mangos, de los cuales la mitad son mangos *Finos* y el resto mangos *Corrientes*. Cada planta fina produce 8500 frutas. Elabore un Algoritmo que determine el total de mangos finos producidos.

Paso 2.- Análisis del problema:

- **Información de Salida**
 - TotalProduccion
- **Datos Conocidos**
 - ProduccionPlantaFina=8500
- **Datos de Entrada (No Conocidos)**
 - NumeroArboles
- **Proceso**
 - $TotalProduccion = (NumeroArboles - (NumeroArboles \text{ MOD } 2) / 2) * ProduccionPlantaFina$

Paso 3.- Construcción del Algoritmo

```
//Objetivo: Determinar el Total de Producción de Mangos Finos
//Programador: Gerardo Gálvez Gámez
//Fecha: 30/agosto/2007
INICIO
  //Definición de Variables y Constantes
  CONST ENTERO ProduccionPlantaFina=8500
  ENTERO NumeroArboles, TotalProduccion
  //Lecturas de Datos de Entrada
  IMPRIMIR "Teclee el Número de Árboles en la Huerta:"
  LEER NumeroArboles
  //Procesamiento de los Datos
  TotalProduccion=(NumeroArboles-(NumeroArboles MOD 2) / 2 ) * ProduccionPlantaFina
  //Impresión de Resultados
  IMPRIMIR "Total de Mangos Finos Producidos: ", TotalProduccion
FIN
```



Paso 4.- Verificación (prueba y depuración)

Pantalla de la computadora	Memoria de la computadora
<p>Teclee el Número de Árboles en la Huerta: <u>10</u></p> <p>Total de Mangos Finos Producidos: <u>42500</u></p>	<p>ProduccionPlantaFina 3F5000AC</p> <p style="text-align: center;">8500</p> <p>NumeroArboles 5F5000AC</p> <p style="text-align: center;">10</p> <p>TotalProduccion 7F5000AC</p> <p style="text-align: center;">42500</p>

Problemas prácticos de estructura secuencial

1. Diseñar un algoritmo pseudocódigo, para hallar el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.
2. Esteban está ahorrando para comprar una patineta que vale 155 pesos. Su papá le ha dado una mensualidad de 50 pesos durante 7 semanas. Por lavar el auto de su tío tres veces recibió 8 pesos. Su hermano ganó 10 pesos por hacer los mandados de su mamá y 4 por sacar a pasear el perro. Elaborar un pseudocódigo que ¿Determinar cuánto le falta a Esteban para comprar la patineta?
3. La Huerta mucha fruta cuenta con un sembrado de N arboles de Mangos de los cuales la mitad son mangos finos el resto mango corrientes. Las plantas finas producen una cantidad de N mangos mientras que las plantas corrientes producen el X% menos que las finas. Cada mango fino tiene un precio de D dólares y el centenar de mangos corrientes es 80% más barato. Elabore un pseudocódigo que determinar el total en pesos por la venta de mangos finos y corrientes.
4. En un juego de combate entre un ángel y un demonio, la Habilidad de Guiar de un ángel se determina mediante la utilización de la siguiente fórmula:

$$\sqrt{\frac{\text{bondad}^2 - \text{maldad}^2 - \text{inteligencia}}{\text{fe}^2 - \text{maldad}^2}}$$

Elabore un pseudocódigo que encuentre el valor de la habilidad.

5. Elabore un algoritmo y calcule el sueldo quincenal de un trabajador, dado su sueldo por hora, los días trabajados y las horas extras trabajadas, la jornada diaria es de 8 horas y las horas extras se pagan el doble que una hora normal.
6. Una carnicería ofrece al público tres tipos de productos: *Carne gruesa*, *delgada* y *hueso*, la *carne gruesa* tiene un determinado precio por kilo, la *carne delgada* es 15% más barata que la gruesa y el *hueso* tiene un precio que equivale al 17% del precio de la carne delgada. Elabore un algoritmo que determine cuanto pagara Ana por comprar N Kilos de Hueso.



7. Elabore un algoritmo que permita determina el total a pagar por consumo de luz a cualquier persona. La compañía de luz efectúa el cobro en base a la siguiente tabla:

140 KWH o menos	\$3.20 por KWH
Los siguientes 170 KWH	\$1.20 por KWH
Exceso sobre 310	\$0.80 por KWH

8. Elabore un algoritmo que calcule los importes por concepto de renta, el impuesto y el costo total de una llamada telefónica de larga distancia de acuerdo a lo siguiente:
- Los 3 primeros minutos tienen un determinado costo.
 - Los minutos que sobrepasen a 3 y no excedan a 10 su costo se incrementa a un 12% y
 - Los minutos que excedan a 10 su precio se incrementa en un 18% sobre el precio de los minutos iniciales.
 - El tiempo mínimo que se cobra por una llamada es de 3 minutos.
 - Solo se cobran minutos completos (es decir si la llamada fue de 4.2 minutos se le cobran 5, si fue de 1.3 se le cobran 2),
 - A cada llamada se le cobra un % de renta sobre lo que resulte se le cobra un % de impuestos.
9. Tres personas deciden invertir su dinero para fundar una empresa. Cada una de ellas invierte una cantidad distinta. Elabore un algoritmo que obtener e imprima el porcentaje que cada quien invierte con respecto a la cantidad total invertida.
10. Elabore un pseudocódigo que permita a un alumno conocer 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 y 15% de la calificación de un trabajo final.
11. El cajero de un banco necesita un programa que le facilite la entrega de una determinada cantidad de dinero, es decir, que le indique la manera de fraccionar dicha cantidad utilizando billetes de: 1000, 500, 200,100, 50, 20 y monedas de 10, 5,1. El objetivo del programa es utilizar la menor cantidad de monedas y billetes de bajo valor.

Por ejemplo la manera más eficaz de fraccionar la cantidad 32567 es:

32 billete(s) de 1000
1 billete(s) de 500
0 billete(s) de 200
0 billete(s) de 100
1 billete(s) de 50
0 billete(s) de 20
1 moneda(s) de 10
1 moneda(s) de 5
2 moneda(s) de 1



Estructuras de selección (condicionales)

Permiten la posibilidad de escoger qué grupos de instrucciones se desea ejecutar en un momento dado. Entre ellas se encuentran:

- Simples
- Dobles
- Anidadas

Estructura de selección simple

Son conocidas como estructuras de tomas de decisiones y su sintaxis es la siguiente:



```
SI expresión-booleana ENTONCES  
instrucciones (acciones a realizar si la  
expresión booleana es evaluada como  
verdadera)
```

```
FIN_SI
```

Caso Práctico de estructura de selección simple

Paso 1: Definición del problema

La utilidad que un trabajador recibe en el reparto anual de utilidades, se le asigna como un porcentaje de su salario mensual, que depende de su antigüedad en la empresa de acuerdo con lo siguiente:

más de 1 año 5.6%

Desarrolle un pseudocódigo el cual determine e imprima en pantalla: el salario mensual y la utilidad.



Paso 2: Análisis del problema

Datos Conocidos

PorcentajeUtilidad=5.6

Datos no Conocidos

SalarioMensual

AtiguedadLaboral

Datos de Salida

SalarioMensual

UtilidadAnual

Proceso

SI AntiguedadLaboral<=1 hacer que UtilidadAnual=0

SI AntiguedadLaboral>1 hacer que UtilidadAnual=SalarioMensual*(PorcentajeUtilidad/100)

Paso 3: Algoritmo en Pseudocódigo

Programador: Gerardo GálvezFecha: 2/agosto/2009Objetivo: Determinar la utilidad laboral de un trabajador correspondiente a su antigüedad laboral, mas de 1 año 5.6%**INICIO**

```
//Definición de variables y constantes
CONST REAL PorcentajeUtilidad=5.6
REAL AntiguedadLaboral, SalarioMensual
REAL UtilidadAnual

//Lectura de datos no conocidos
IMPRIMIR "Teclee la antigüedad laboral del Trabajador: "
LEER AntiguedadLaboral
IMPRIMIR "Teclee el Salario Mensual:$ "
LEER SalarioMensual

//Proceso, determinar la utilidad anual del trabajador
SI AntiguedadLaboral <= 1 ENTONCES
    UtilidadAnual=0
FIN_SI
Si AntiguedadLaboral > 1 ENTONCES
    UtilidadAnual=SalarioMensual*(PorcentajeUtilidad/100)
FIN_SI

//Imprimir la utilidad del trabajador
IMPRIMIR "Utilidad Anual:$ ", UtilidadAnual
```

FIN



Paso 4: Verificación del algoritmo

Pantalla de la computadora

Teclee la antigüedad laboral del Trabajador: 2
 Teclee el Salario Mensual: 3000
 Utilidad Anual:\$ 168

Memoria de la computadora

PorcentajeUtilidad	3F5000AC
	5.6
SalarioMensual	5F5000AC
	3000
UtilidadAnual	7F5000AC
	168
AntiguedadLaboral	7F7000AC
	2



ANEXO A - Código ASCII

Código ASCII. Siglas en inglés de American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información).

Referencia: Microsoft Developer Network. <http://bit.ly/14zvBy4>

Caracteres ASCII de control		Caracteres ASCII imprimibles				ASCII extendido (Página de código 437)									
00	NULL (carácter nulo)	32	espacio	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH (inicio encabezado)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	õ
02	STX (inicio texto)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
03	ETX (fin de texto)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	Ò
04	EOT (fin transmisión)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö
05	ENQ (consulta)	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	Õ
06	ACK (reconocimiento)	38	&	70	F	102	f	134	á	166	ª	198	‡	230	µ
07	BEL (timbre)	39	'	71	G	103	g	135	ç	167	º	199	Ä	231	þ
08	BS (retroceso)	40	(72	H	104	h	136	ê	168	¿	200	ℒ	232	ß
09	HT (tab horizontal)	41)	73	I	105	i	137	ë	169	®	201	ℓ	233	Ú
10	LF (nueva línea)	42	*	74	J	106	j	138	è	170	¬	202	ℝ	234	Ù
11	VT (tab vertical)	43	+	75	K	107	k	139	ì	171	½	203	℥	235	Û
12	FF (nueva página)	44	,	76	L	108	l	140	í	172	¼	204	℥	236	ý
13	CR (retorno de carro)	45	-	77	M	109	m	141	ì	173	⅓	205	≡	237	ÿ
14	SO (desplaza afuera)	46	.	78	N	110	n	142	Ë	174	«	206	≠	238	—
15	SI (desplaza adentro)	47	/	79	O	111	o	143	Ä	175	»	207	⊞	239	·
16	DLE (esc.vínculo datos)	48	0	80	P	112	p	144	É	176	⋮	208	⊞	240	≡
17	DC1 (control disp. 1)	49	1	81	Q	113	q	145	æ	177	⋮	209	⊞	241	±
18	DC2 (control disp. 2)	50	2	82	R	114	r	146	Æ	178	⋮	210	⊞	242	—
19	DC3 (control disp. 3)	51	3	83	S	115	s	147	ø	179		211	⊞	243	¾
20	DC4 (control disp. 4)	52	4	84	T	116	t	148	ö	180	†	212	⊞	244	¶
21	NAK (conf. negativa)	53	5	85	U	117	u	149	ò	181	‡	213	⊞	245	§
22	SYN (inactividad sínc)	54	6	86	V	118	v	150	ù	182	Ä	214	⊞	246	÷
23	ETB (fin bloque trans)	55	7	87	W	119	w	151	ù	183	À	215	⊞	247	˙
24	CAN (cancelar)	56	8	88	X	120	x	152	ÿ	184	©	216	⊞	248	˚
25	EM (fin del medio)	57	9	89	Y	121	y	153	Ö	185	ª	217	⊞	249	ˆ
26	SUB (sustitución)	58	:	90	Z	122	z	154	Ü	186	ª	218	⊞	250	˜
27	ESC (escape)	59	;	91	[123	{	155	ø	187	ª	219	⊞	251	₁
28	FS (sep. archivos)	60	<	92	\	124		156	£	188	ª	220	⊞	252	₃
29	GS (sep. grupos)	61	=	93]	125	}	157	Ø	189	¢	221	⊞	253	₂
30	RS (sep. registros)	62	>	94	^	126	~	158	x	190	¥	222	⊞	254	■
31	US (sep. unidades)	63	?	95	-			159	f	191	₯	223	⊞	255	nbsp
127	DEL (suprimir)														

de uso frecuente (idioma español)	vocales con acento (español acento agudo)	vocales con diéresis	símbolos matemáticos	comillas, llaves paréntesis
ñ alt + 164	á alt + 160	ä alt + 132	½ alt + 171	" alt + 34
Ñ alt + 165	é alt + 130	ë alt + 137	¼ alt + 172	' alt + 39
@ alt + 64	í alt + 161	ï alt + 139	¾ alt + 243	(alt + 40
¿ alt + 168	ó alt + 162	ö alt + 148	¹ alt + 251) alt + 41
? alt + 63	ú alt + 163	ü alt + 129	³ alt + 252	[alt + 91
¡ alt + 173	Á alt + 181	Ä alt + 142	² alt + 253] alt + 93
! alt + 33	É alt + 144	Ë alt + 211	f alt + 159	{ alt + 123
: alt + 58	Í alt + 214	Ï alt + 216	± alt + 241	} alt + 125
/ alt + 47	Ó alt + 224	Ö alt + 153	x alt + 158	« alt + 174
\ alt + 92	Ú alt + 233	Ü alt + 154	÷ alt + 246	» alt + 175



ANEXO B – Instrucciones de Entrega – Tareas

Las deberán ser entregadas cumpliendo con las siguientes indicaciones:

1. Deberá entregar estos ejercicios resueltos en su totalidad.
2. Deberá utilizar hoja blanca.
3. Deberá contener una hoja de presentación con los siguientes datos:
 - a. Nombre de la universidad, nombre de la facultad
 - b. Nombre de la materia, tema, nombre del instructor
 - c. Nombre del alumno, grupo
 - d. Fecha, lugar.
4. Las hojas deberán ser entregadas grapadas.
5. Su trabajo deberá ser manuscrito (a mano).
6. El contenido deberá ser escrito conforme a las reglas ortográficas y gramaticales del español.
7. Se considerará la limpieza y presentación del trabajo entregado.
8. Únicamente se aceptarán la entrega en la fecha establecida para cada tarea.



ANEXO 01 – TAREA Control de Lectura

Objetivo

El alumno definirá conceptos básicos y será capaz de distinguir los componentes esenciales de las Tecnologías de la Información y Comunicación.

Fecha de Asignación

Día lunes 12/agosto/2013.

Fecha de Entrega

Día martes 13 de agosto al inicio de clase.

Instrucciones

Deberá dar Lectura al Capítulo 1 (completo) y Capítulo 2 (solo los primeros 3 temas).

Por única ocasión **no habrá entrega por escrito**, deberá **participar activamente en la discusión** inicial durante la sesión 2 del curso.



ANEXO 02 – TAREA Resolución de Problemas Lógicos

Objetivo

El alumno planteará la solución de problemas que exigen razonamiento lógico.

Fecha de Asignación

Día lunes 12/agosto/2013.

Fecha de Entrega

Día viernes 16/agosto/2013 al inicio de clase.

Instrucciones

La entrega de esta tarea será por escrito cumpliendo con las “Instrucciones de Entrega” descritas en el “ANEXO B – Instrucciones de Entrega – Tareas”.

Problemas Lógicos

1. Un caracol sube por una pared vertical de 5 metros de altura. Durante el día sube 3 metros, pero durante la noche se queda dormido y resbala 2 metros. ¿En cuántos días subirá la pared? Elabore una descripción detallada y lógica de la solución del problema.
2. Un avión cubrió la distancia que separa a Culiacán, Sinaloa y Morelia, Michoacán en una hora y 20 minutos, sin embargo al volar de regreso recorrió esta distancia en 80 minutos. ¿Cómo se explica esto? Elabore una descripción detallada y lógica de la explicación que dé solución del problema, antes descrito.
3. Yendo yo para CU, me encontré con mi primo Juan, quien venía con su esposa y sus 3 hijos, cada hijo con dos mascotas, y cada mascota con 4 garrapatas. ¿Cuántos iban para CU?



ANEXO 03 – TAREA Elaboración de Algoritmos

Objetivo

El alumno demostrará la habilidad alcanzada en clases, para proponer solución algorítmica a diversos problemas.

Fecha de Asignación

Día lunes 12/agosto/2013.

Fecha de Entrega

Día viernes 16/agosto/2013 al inicio de clase.

Instrucciones

La entrega de esta tarea será por escrito cumpliendo con las “Instrucciones de Entrega” descritas en el “ANEXO B – Instrucciones de Entrega – Tareas”.

Los algoritmos deberán ser resueltos según la metodología vista en clase que incluye:

- a) Análisis del problema
- b) Construcción del algoritmo en pseudocódigo.
- c) Verificación (prueba y depuración)

Descripción de problemas

1. Diseñar un algoritmo que permita calcular y dar a conocer el total que una persona debe pagar por la compra de un Mouse Óptico. Se sabe que el cliente tiene que pagar el 16% por concepto de IVA.
2. Diseñar un algoritmo que permita calcular y dar a conocer el total que una persona debe pagar por la compra de un Mouse Óptico. Se sabe que el cliente recibe un Porcentaje por concepto de descuento.
3. Suponga que una persona desea invertir su capital en un banco y desea saber cuánto dinero ganará después de un mes si el banco paga a una razón de 2% mensual. Elabore un algoritmo que dé a conocer cuánto dinero ganara la persona.



ANEXO 04 – TAREA Conceptos, Tipos de Datos y Expresiones

Objetivo

El alumno reforzará los conceptos de algoritmo y sus requisitos. Identificará el tipo de dato para cada información. Demostrará la habilidad desarrollada para convertir expresiones matemáticas a expresiones válidas por la computadora.

Fecha de Asignación

Día martes 13/agosto/2013.

Fecha de Entrega

Día viernes 16/agosto/2013 al inicio de clase.

CONCEPTOS

1. Defina el concepto de algoritmo.
2. Defina Requisitos para un algoritmo según Donald Knuth: Precisión, Carácter Finito, Eficacia.
3. ¿Un algoritmo siempre tiene una solución única? Explique.
4. ¿Cuál es la diferencia entre una Constante y una Variable?

TIPOS DE DATOS

5. ¿Qué tipos de datos existen?
6. Escriba el concepto.

_____ Son los números que en su contenido incluyen el punto decimal; pueden ser positivos o negativos.

_____ Son los números que no contienen componentes fraccionarios; pueden ser positivos o negativos.

7. Escriba el tipo de dato correspondiente y especifique el identificador a utilizar.

Información	Tipo de Dato	Identificador
Nombre de persona		
La talla del calzado de una persona		
Cantidad de vacas que están en un corral		
Cantidad de computadoras del centro de cómputo		
Velocidad de transferencia, al descargar un archivo de Internet		
Salario quincenal de un trabajador		
“Universidad Autónoma de Sinaloa”		



Información	Tipo de Dato	Identificador
's'		
"No"		
3.14159...		
Cantidad de Habitantes de Culiacán		
Cantidad de alumnos en Facultad de Informática		
Calificación obtenida en la materia: Algoritmia		
Tu Nombre Completo		
Tu Apellido		
Número de Cuenta Alumno-UAS		
El código postal de tu casa.		
80010 es el código postal de la facultad.		
7161361 es el teléfono de la facultad		
rogelioprieto@gmail.com		
El nombre de la colonia donde vives.		
El total de frutas de un árbol		
El género (sexo) de una persona		
El teléfono de una empresa		
Estado civil de una persona		
El pago por hora de un obrero		
El precio de venta unitario de un producto		
Si una persona cuenta con tarjeta de crédito del banco X		
El número de placas de un automóvil		
El consumo en Kilowatts de una casa habitación		

EXPRESIONES

8. De la siguiente lista de expresiones, indique la causa porque son incorrectas, en caso de serlo:

- $X1 = -A + -X * Y /$
- $X2 = (A + B) / (* C)$
- $X3 = (A / X) * B * C -$
- $X4 = X * -Y - A \# 5$
- $X5 = +A * -B \% 2$
- $X6 = [Parcial1 - Parcial2) + Parcial3]$



9. Indique el orden en la evaluación y el resultado que obtendría la computadora, de las siguientes expresiones, además indique el tipo de dato del resultado:

- $X1 = 13 * 4 / 2$
- $X2 = 13 / 4 * 2$
- $X3 = 4 + 2 * 3 - 13 \text{ MOD } 5$
- $X4 = (7 - 4 + (16 \text{ MOD } 3)) / 2 * 3 - 2$
- $X5 = ((13 / 2) - (2 + 1)) / (9 * (5 + 1))$
- Para las siguientes expresiones tome en cuenta los valores de las siguientes variables: $A = 5$ y $B = 16$
- $X6 = (A * 7 + B / 2) >= (B + 6)$
- $X7 = (A * 5 - 19.78 / 4) < (B * 13 / A)$
- $X8 = \text{NOT}(5 > A + 2) \text{ OR } (34 + B - 2 / 4 <> 3 * 21 - A)$

10. Convertir las siguientes formulas a expresiones válidas por la computadora.

$$X1 = \frac{dy}{dx} - 2xy$$

$$X2 = (-b - \sqrt{b^2 - 4ac})/2a$$

$$X3 = \left(\frac{d^2y}{dx^3}\right)^4 + 2\frac{dy}{dx}$$

$$X4 = \frac{n(n-1)x^2}{2}$$

$$X5 = a^2 + b^2 - c^2$$

$$X6 = \pi r^2 + \frac{n(n-1)x^2}{2}$$

$$X7 = \frac{\pi}{2} + \frac{d}{dx} - \frac{d^2y}{dx}$$

$$X8 = \frac{\sqrt{a^2+b^2}}{\sqrt{a^2b^2x^2+(a^2-b^2)}}$$



ANEXO 05 – TAREA Elaboración de Algoritmos en pseudocódigo (i)

Objetivo

El alumno demostrará la habilidad alcanzada en clases, para analizar, construir y probar pseudocódigos de diversos problemas, utilizando procedimientos de solución secuenciales.

Fecha de Asignación

Día jueves 15/agosto/2013.

Fecha de Entrega

Día viernes 16/agosto/2013 al inicio de clase.

Instrucciones

La entrega de esta tarea será por escrito cumpliendo con las “Instrucciones de Entrega” descritas en el “ANEXO B – Instrucciones de Entrega – Tareas”.

Los algoritmos deberán ser resueltos según la metodología vista en clase que incluye:

- a) Análisis del problema
- b) Construcción del algoritmo en pseudocódigo.
- c) Verificación (prueba y depuración)

Descripción de problemas

1.- Elabore un algoritmo que solicite el valor de una variable numérica, la cual represente el radio de un círculo y que calcule y dé a conocer su área.

2.- Haga un algoritmo que lea una variable numérica X y que calcule e imprima el valor de Z, suponiendo que:

$$Z = \sqrt{(x + 7)^2 + 4}$$



ANEXO 06 – TAREA Elaboración de Algoritmos en pseudocódigo (ii)

Objetivo

El alumno demostrará la habilidad alcanzada en clases, para analizar, construir y probar pseudocódigos de diversos problemas, utilizando procedimientos de solución secuenciales y/o de selecciones simples, según aplique en cada caso.

Fecha de Asignación

Día viernes 16/agosto/2013.

Fecha de Entrega

Día lunes 19/agosto/2013 al inicio de clase.

Instrucciones

La entrega de esta tarea será por escrito cumpliendo con las “Instrucciones de Entrega” descritas en el “ANEXO B – Instrucciones de Entrega – Tareas”.

Los algoritmos deberán ser resueltos según la metodología vista en clase que incluye:

- a) Análisis del problema
- b) Construcción del algoritmo en pseudocódigo.
- c) Verificación (prueba y depuración)

Descripción de problemas

1. Elabore un algoritmo que lea dos variables numéricas que representen respectivamente la velocidad y el tiempo que dura el recorrido de un automóvil. El algoritmo debe calcular e imprimir la distancia que recorre dicho automóvil. Se sabe que la velocidad a la que viaja es la misma durante todo el recorrido, además, el viaje se realizó el día 14 de febrero del 2013 y estaba lloviendo.

2. Un alumno desea saber cuál será su calificación final en la materia de programación. 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.

Elabore un algoritmo que calcule e imprima la calificación final del alumno.

3. Construya un algoritmo que solicite como datos de entrada, el precio anterior y el precio actual de un artículo y que calcule e imprima el porcentaje de incremento que tuvo el artículo. Suponga que el precio actual es mayor que el anterior.

4. En cierta llantera de esta ciudad, por la compra de N llantas se hace un descuento de X%, cada llanta tiene un precio de \$899.99, sin embargo, si el cliente compra 4 llantas sólo pagara lo correspondiente a 3 llantas.

Las llantas son de diferentes marcas como: Michelin, Firestone, Good Year, Toyo, Tornel, entre otras.

La empresa tiene un horario de atención al cliente de 9:00 am a 9:00 pm. y la secretaria Paty, siempre llega 30 minutos tarde.

5. Hacer un algoritmo, que lea dos variables (X, Y) y que determine si se satisface la siguiente ecuación. $3X + 5Y \geq 4$.



ANEXO 07 – TAREA Elaboración de Algoritmos en pseudocódigo (iii)

Objetivo

El alumno demostrará la habilidad alcanzada en clases, para analizar, construir y probar pseudocódigos de diversos problemas, utilizando procedimientos de solución secuenciales y/o de selecciones simples, según aplique en cada caso.

Fecha de Asignación

Día lunes 19/agosto/2013.

Fecha de Entrega

Día miércoles 21/agosto/2013 antes de presentar examen.

Instrucciones

La entrega de esta tarea será por escrito cumpliendo con las “Instrucciones de Entrega” descritas en el “ANEXO B – Instrucciones de Entrega – Tareas”.

Los algoritmos deberán ser resueltos según la metodología vista en clase que incluye:

- Análisis del problema
- Construcción del algoritmo en pseudocódigo.
- Verificación (prueba y depuración)

Descripción de problemas

1.- Elabore un algoritmo que lea un número entero positivo y determine, si tiene 1, 2, 3, 4, 5 ó más de 5 cifras, imprimiendo lo que corresponda. Algunos Ejemplos son:

Entrada(Número)	Salida (Total de cifras)
5	1
15	2
567	3
5678	4
45678	5
1234567	Más de 5

2.- Elabore un algoritmo que determine el día que será mañana, para lograrlo deberá leer el día, el mes y el año actual. Considere que no existen años bisiestos. Ejemplos:

Entrada	Salida (Total de cifras)
Día =22 Mes=11 Año =2012	Mañana será 23 -11-2012
Día =30 Mes=11 Año =2012	Mañana será 01 -12-2012
Día =31 Mes=12 Año =2012	Mañana será 01 -01-2013



BIBLIOGRAFIA

LÓPEZ ROMÁN, LEOBARDO (2006) Metodología de la Programación Orientada a Objetos. Edit Alfaomega, México D.F. Primera Edición.

CAIRO BATTISTUTTI, OSVALDO. (2006) Metodología de la Programación, Algoritmos, Diagramas de Flujo y Programas. Edit Alfaomega, México D.F Tercera edición.

LÓPEZ ROMÁN, LEOBARDO. (2003) Programación Estructurada, un enfoque algorítmico. Edit Alfaomega, México D.F. Segunda Edición.

Microsoft. Manual del Lenguaje de Programación C# (ESPAÑOL SPANISH).doc