



Conceptos importantes

Repaso



Procedimiento de creación de un programa

■ Hacerles gráfico y explicar. Pág. 95.

■ Ciclo de desarrollo y actividades

■ Análisis. (Qué se quiere?)

- Analizar el caso real
- Algoritmo

■ Diseño. (Cómo automatizar lo que quiere?)

- Algoritmo Lenguaje Natural
- Pseudocódigo o Miniespecificación
- Diagrama de Flujo
- Pruebas de escritorio

■ Codificación. (Cómo se escribe en lenguaje de programación?)

- Codificación de algoritmos
- Compilación

■ Pruebas e implementación. (Se cumple con el diseño?)

- Pruebas e implementación

**Curso
3071**



Principios en la elaboración de enunciados

- **Relacionados con los datos de entrada.**
- **Relacionados con el proceso**
- **Relacionados con los datos de salida.**



Principios en la elaboración de enunciados

■ Relacionados con los datos de entrada.

- **Economía de datos:** proveer datos suficientes y necesarios.
- **Autosuficiencia de datos:** datos calculados no deben ser datos de entrada.
- **Compatibilidad de los datos:** los datos proporcionados deben ser compatibles y fáciles de manejar
- **Conocimiento preestablecido:** cuando hay datos que ya son aceptados como verdad y no cambian, evite preguntarlos, ya que forman parte del proceso. Ej. 60 min.



Principios en la elaboración de enunciados

■ Relacionados con el proceso:

- **Consistencia del resultado:** iguales datos de entrada, iguales datos de salida.
- **Eficiencia:** mayor cantidad de resultado, con menos datos de entrada.
- **Eficacia:** el proceso debe hacer exactamente lo que se requiere.
- **Simplicidad:** compacto y sencillo mejor a largo y complejo.
- **Independencia:** evitar dependencia con elementos externos.
- **Reutilización:** utilizar procesos existentes.
- **Rastreabilidad:** operaciones rastreables y auditables.



Principios en la elaboración de enunciados

■ Relacionados con los datos de salida.

- **Cantidad y forma:** generados en cantidad y forma requeridos por el usuario.
- **Exactitud:** tener grado de exactitud requerido por el usuario.
- **Dominio de salida:** los datos pueden tener diferente dominio que los datos de entrada.





Algoritmos



Algoritmos

- **Real Academia Española: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.**
- **En programación: conjunto ordenado y finito de asignaciones, procesos, cálculos y decisiones que permiten a un programa satisfacer una unidad de funcionalidad dada.**
- **Dar un ejemplo pequeño.**



Algoritmos ... Cont.

- 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.



Algoritmos ... Cont

- **Preciso**: implica el orden de realización de cada uno de los pasos
- **Definido**: si se sigue dos veces, se obtiene el mismo resultado.
- **Finito**: Tiene un número determinado de pasos, implica que tiene un fin,



Algoritmos ... Cont

- 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



Medios de expresión de un algoritmo

- pseudocódigo o miniespecificación
- diagrama de flujo
- lenguaje de programación



Ejemplo

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



Ejemplo-ALGORITMO:

Inicio

PASO 1. Dormir

PASO 2. Mientras no suene el despertador y se levante vuelve al **PASO 1**.

PASO 3. Una vez despierto, mirar la hora.

PASO 4. Bañarse.

PASO 5. Vestirse.

PASO 6. Si no tiene suficiente tiempo pasar al **paso 8**

PASO 7. Desayunar.

PASO 8. Cepillarse los dientes.

PASO 9. Despedirse de la mamá y el papá

PASO 10. Si no tiene suficiente tiempo pasar al **paso 12**

PASO 11. Caminar a la parada y pase al **paso 13**

PASO 12. Correr a la parada.

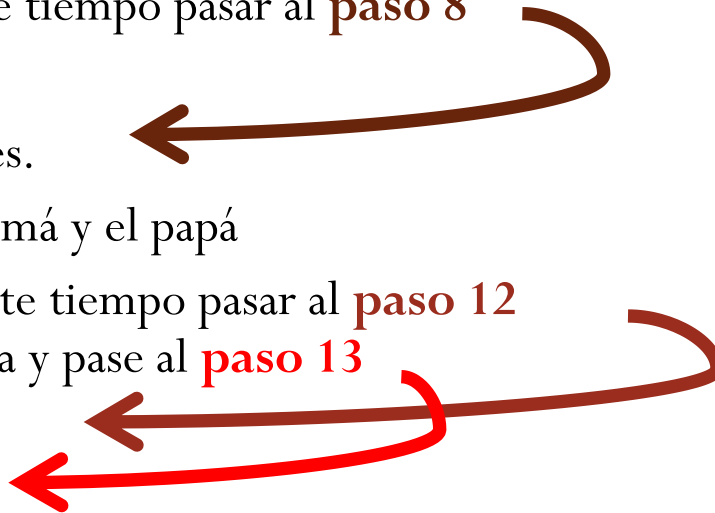
PASO 13. Esperar el bus

PASO 14. Tomar el bus.

PASO 15. Bajarse al llegar a la universidad.

PASO 16. Entrar a la universidad.

Fin





Algoritmos y pseudocódigo

LP07



Pseudocódigo o Miniespecificación

- **Representación abstracta y simbólica entendible por el ser humano, que muestra al máximo detalle las operaciones que un programa debe hacer con los datos**
- **Características:**
 - **Tiene inicio y fin**
 - **Independencia del lenguaje de programación**
 - **Cuadro básico de instrucciones.**
 - **Se describe a través de representaciones simbólicas**
 - **Referencia a los datos**



■ **Importancia de algoritmos:**

- **Definen actividades que se tienen que hacer para alcanzar un objetivo.**
- **Dan el orden de las actividades**
- **Dan alcance general de las tareas.**

■ **Algoritmos - Pseudocódigo:**

- **Primera parte-DECLARACIÓN:** Poner en evidencia los datos manejados en cada una de las actividades por desarrollar, así como el dominio de estos datos.
- **Objetivo:** hacer del conocimiento del desarrollador los datos que estarán involucrados en el programa, así como sus respectivos dominios.



Propiedades de los datos o variables

- **Tipo de dato.**
- **Dominio:** conjunto de valores válidos.
- **Opcionalidad:** allow null o no.
- **Valor:** expresión almacenada como dato en un momento determinado.



Dominios

- **Dominio:** conjunto de valores válidos para un dato
- **Dominio de tipo:** restricciones de valores por tipo de dato.
- **Dominio de regla de negocio:** restricciones de valores originada por una regla específica del negocio.
- **Dominio de relación:** restricción de valores originada por la relación que el dato debe mantener con otros datos. BD.



Operaciones a nivel dato

■ Declaración

Ejemplos:

estudiante : Entero[0-100]

promedioPonderado : Entero [0-100]

contador : Entero[1-10]

acumulaNotas : Entero[1-10]

letra : Carácter

nombreAlumno : Cadena [40]

notas : Arreglo [10] Entero[0-100]



Operaciones a nivel dato

- **Cambio de estado (valor en el tiempo – operación de asignación)**

Ejemplo:

contador = contador + 1

promedioPonderado = acumulaNotas / 10



Operaciones a nivel dato

■ Muestra de datos (mostrar en interfaz)

Ejemplo:

Imprimir “Nota final de la materia: ”

Leer notas

Imprimir “El promedio ponderado del
estudiante “ + **estudiante** + “ es: ” +
promedioPonderado



Mecanismos auxiliares

- **Contadores:** sumar 1 a una variable
- **Acumuladores:** $x = x + 20$
- **Anidamiento:** acción de que una estructura de decisión y control forme parte del código controlado de otra estructura.
- **Pruebas de escritorio:** pruebas manuales que se encargan de visualizar el comportamiento de los estados de las variables en el transcurso de la ejecución de un programa. Recorrer todos los caminos posibles del algoritmo o del Diagrama de Flujo
- **Arreglos:** variables que permiten asociar, a través de un mismo nombre de variable, varias posiciones de memoria. Súbíndice. Elemento



EJEMPLO COMPLETO

- **Demos un vistazo a un ejemplo completo antes de continuar.....**

1. **Problema**
2. **Algoritmo**
3. **Pseudocódigo**



Ejemplo

- **Hacer un programa que lea las calificaciones de un alumno en 10 asignaturas, las almacene en un vector y calcule e imprima su promedio ponderado.**
- **Plantee su solución utilizando arreglos**



Algoritmo Principal-*Lenguaje Natural*

1. Inicio.
2. Solicitar el Nombre del alumno.
3. Solicita Nombre de la materia
4. Solicita Nota de Asignatura
5. Guarda la notas de la asignatura en un arreglo.
6. Si no es la nota solicitada $\neq 10$, pasar al paso 4
7. Posicionarse al inicio del arreglo de notas
8. Leer nota del arreglo
9. Acumular valor de nota en Sumatoria de Notas
10. Avanzar una posición en el arreglo
11. Si no es la nota solicitada $\neq 10$, pasar al paso 8
12. Dividir Sumatoria de notas entre 10
13. Mostrar el promedio ponderado.
14. Fin.



Pseudocódigo

INICIO:

estudiante, Cadena (40)[{A-Z}, {a-z}, {BS}])

notas: Arreglo [10] Entero [0-100]

promedioPonderado: Entero[0-100]

contador:Entero [1-10]

acumulaNotas: Entero[0-1000]

contador = 1

Imprimir "Nombre del estudiante: "

Leer estudiante

DO

Imprimir "Nota final de la materia: "

Leer [notascontador]

 contador = contador + 1

WHILE contador >= 10

contador = 1

acumularNotas = 0

DO

 acumulaNotas = acumulaNotas + notas [contador]

 contador = contador + 1

WHILE contador >= 10

promedioPonderado=acumulaNotas/10

Imprimir "El promedio ponderado del estudiante " + estudiante + " es: "+promedioPonderado

FIN:





Estructura de Control

SELECCIÓN



Operaciones a nivel dato

■ Estructuras de decisión

■ Condicional simple (SI)

General:

Si <CONDICIÓN> **entonces**

Acción

Fin Si

Ejemplo:

IF contador = 10 **THEN**

IMPRIMIR “Terminar”

ENDIF



Operaciones a nivel dato

■ Estructuras de decisión

■ Condicional compuesta (SI – SINO)

General:

SI <CONDICIÓN> **ENTONCES**

Acción

SINO

Acción (Cuando condición es Falsa)

FIN SI

Ejemplo:

IF contador = 10 **THEN**

IMPRIMIR “Terminar”

ELSE

IMPRIMIR “Continuar”

ENDIF



EJERCICIO

Elaborar un algoritmo que lea el tamaño de un ángulo en grados e imprimir si es:

- Agudo si es menor a 90°
- Recto si es igual a 90°
- Obtuso si es mayor que 90° pero menor que 180°
- Llano si es igual a 180°
- Concavo si es mayor que 180° pero menor que 360°

Utilizar IF-THEN.

(Primero hágalo usted...después compare la solución)



Algoritmo TIPO ANGULO

Método **principal**

Declaraciones

Variables

angulo: Real

Imprimir "Escriba el ángulo:"

Leer angulo

IF angulo < 90 **THEN**

Imprimir "Agudo"

ENDIF

IF angulo = 90 **THEN**

Imprimir "Recto"

ENDIF

IF (angulo > 90) **AND** (angulo < 180) **THEN**

Imprimir "Obtuso"

ENDIF

IF angulo = 180 **THEN**

Imprimir "Llano"

ENDIF

IF (angulo > 180) **AND** (angulo < 360) **THEN**

Imprimir "Concavo"

ENDIF

Fin Método **principal**



EJERCICIO

Elaborar un algoritmo que permita leer el tamaño de dos ángulos A y B; e imprima:

- si son iguales y que tipo de ángulo son
- si son diferentes y que tipo de ángulo es cada uno

Utilizar IF-THEN-ELSE e IF-THEN.

(Primero hágalo usted...después compare la solución)



Algoritmo ANGULOS IGUALES O DIFERENTES

Método principal

Declaraciones

Variables

anguloA, anguloB: Real

Imprimir "Ingrese el primer ángulo"

Leer AnguloA

Imprimir "Ingrese el primer ángulo"

Leer AnguloB

IF anguloA = anguloB THEN

Imprimir "LOS ANGULOS SON IGUALES"

IF anguloA < 90 THEN

Imprimir "SON AGUDOS"

ENDIF

IF anguloA = 90 THEN

Imprimir "SON RECTOS"

ENDIF

IF

(anguloA > 90) AND (anguloA < 180) THEN

Imprimir "SON OBTUSOS"

ENDIF

IF anguloA = 180 THEN

Imprimir "SON LLANOS"

ENDIF

IF

(anguloA > 180) AND (anguloA < 360) THEN

Imprimir "SON CONCAVOS"

ENDIF

ELSE

Imprimir "LOS ANGULOS SON DIFERENTES"

IF anguloA < 90 THEN

Imprimir "EL ANGULO A ES AGUDO"

ENDIF

IF anguloA = 90 THEN

Imprimir "EL ANGULO A ES RECTO"

ENDIF

IF (anguloA > 90) AND (anguloA < 180) THEN

Imprimir "EL ANGULO A ES OBTUSO"

ENDIF

IF anguloA = 180 THEN

Imprimir "EL ANGULO A ES LLANO"

ENDIF

IF (anguloA > 180) AND (anguloA < 360) THEN

Imprimir "EL ANGULO A ES CONCAVO"

ENDIF

IF anguloB < 90 THEN

Imprimir "EL ANGULO B ES AGUDO"

ENDIF

IF anguloB = 90 THEN

Imprimir "EL ANGULO B ES RECTO"

ENDIF

IF (anguloB > 90) AND (anguloB < 180) THEN

Imprimir "EL ANGULO B ES OBTUSO"

ENDIF

IF anguloB = 180 THEN

Imprimir "EL ANGULO B ES LLANO"

ENDIF

IF (anguloB > 180) AND (anguloB < 360) THEN

Imprimir "EL ANGULO B ES CONCAVO"

ENDIF

ENDIF

Fin Método principal





Estructura de Control

DO ... WHILE



Operaciones a nivel dato

■ Estructuras de repetición

■ DO ... WHILE

General:

<INICIALIZA CONDICIÓN>

DO

Acción O Acciones

<CAMBIA CONDICIÓN>

WHILE <CONDICIÓN>

Ejemplo:

contador = 1

DO

IMPRIMIR "Terminar"

contador = contador + 1

WHILE contador = 10 **THEN**



EJERCICIO

Elaborar un algoritmo que proporcione el siguiente reporte:

Nombre	Calificaciones finales			Calif. final	Obser- vacion
	Cal.1	Cal.2	Cal.3		
XXXXXXXXXXXXXXXXXXXX	99.99	99.99	99.99	99.99	Aprobado
XXXXXXXXXXXXXXXXXXXX	99.99	99.99	99.99	99.99	Reprobado
: : : :	:				
: : : :	:				
XXXXXXXXXXXXXXXXXXXX	99.99	99.99	99.99	99.99	Reprobado
Total Aprobados: 999					
Total Reprobados: 999					



A partir de que se tienen varios alumnos. Por cada alumno se tiene el NOMBRE, calificación 1, 2, y 3, y que la calificación 1 vale el 20%, la 2 el 30% y la 3 el 50%.

Deberá imprimir la observación APROBADO si la calificación final es mayor o igual a 60, en caso contrario imprimirá REPROBADO.

Al final se imprime el total de alumnos aprobados y el total de reprobados, es decir, debe contar la cantidad de alumnos que obtuvieron 60 o más de calificación; y la cantidad de alumnos que su calificación fue menor a 60.

(Primero hágalo usted...después compare la solución)



Algoritmo ALUMNOS APROBADOS REPROBADOS

Método principal

Declaraciones

Variables

nombreAlum: Cadena

totAprobados, totReprobados: Entero

calif1, calif2, calif3, caliFinal: Real

observacion: Cadena

desea: Carácter

Imprimir encabezado

totAprobados = 0

totReprobados = 0

desea = ""

DO

Solicitar Nombre, calificación 1, 2 y 3

Leer nombreAlum, calif1, calif2, calif3

caliFinal =
 $(\text{calif1} * 0.2) + (\text{calif2} * 0.3) + (\text{calif3} * 0.5)$

IF caliFinal \geq 60 THEN

totAprobados = totAprobados + 1

observacion = "APROBADO"

ELSE

totReprobados = totReprobados + 1

observacion = "REPROBADO"

ENDIF

Imprimir nombreAlum, calif1, calif2,
calif3, caliFinal, observacion

Imprimir "¿Desea procesar otro alumno
(S/N)?"

Leer desea

WHILE desea = "S"

Imprimir totAprobados, totReprobados

Fin Método principal





Estructura de Control

FOR



Operaciones a nivel dato

■ Estructuras de repetición

■ FOR

General:

FOR <VALOR INICIAL>; <CONDICIÓN>; <INCREMENTO>

Acción O Acciones

ENDFOR

Ejemplo:

FOR contador = 0; contador = 10; contador ++

IMPRIMIR “Terminar”

ENDFOR

EJERCICIO

Elaborar un algoritmo que pregunte a cuántos números se desea calcular el factorial; lea la cantidad en N. A continuación, debe leer un número e imprimir su factorial, luego leer otro, y así hasta leer los N números.

(Primero hágalo usted...después compare la solución)



Algoritmo FACTORIALES DE N NUMEROS

Método principal

Declaraciones

Variables

n, i, j, fact, num: Entero

Imprimir “Escriba la cantidad de números a calcular factorial”

Leer n

FOR j=1; j<=n; j++

 Solicitar Número

 Leer num

 IF num = 0 THEN

 fact = 1

 ELSE

 fact = 1

 FOR i=num; i>=1; i--

 fact = fact * i

 ENDFOR

 ENDIF

 Imprimir fact

ENDFOR

Fin Método principal





Estructura de Control

WHILE



Operaciones a nivel dato

■ Estructuras de repetición

■ DO ... WHILE

General:

<INICIALIZA CONDICIÓN>

WHILE <CONDICIÓN>

Acción O Acciones

<CAMBIA CONDICIÓN>

ENDWHILE

Ejemplo:

contador = 1

WHILE contador = 10

IMPRIMIR "Terminar"

contador = contador + 1

ENDWHILE



EJERCICIO

Una compañía manufacturera fabrica el producto A. Para fabricar una unidad de dicho producto se requieren los siguientes materiales:

Material 1: 3 unidades

Material 2: 4 unidades

Material 3: 1 unidades

Material 4: 2 unidades

Material 5: 3 unidades

Material 6: 2 unidades

Elaborar un algoritmo que lea pedidos del producto A, en cada pedido se tiene el dato cantidad de unidades del producto A; cuando termine de leer los pedidos, imprimir:



Listado de materiales requeridos

Material	Total de unidades
1	999
2	999
3	999
4	999
5	999
6	999

(Primero hágalo usted...después compare la solución)



Algoritmo MATERIAL REQUERIDO

Método principal

Declaraciones

Variables

resp: Carácter

material1, material2, material3,
material4, material5, material6,
cantidad, totProd: Entero

totProd = 0

Imprimir “¿HAY PEDIDO (S/N)?”

Leer resp

WHILE resp = “S”

Imprimir “Escriba Cantidad pedida”

Leer cantidad

totProd = totProd + cantidad

Imprimir “¿HAY OTRO PEDIDO
(S/N)?”

Leer resp

ENDWHILE

material1 = totProd * 3

material2 = totProd * 4

material3 = totProd * 1

material4 = totProd * 2

material5 = totProd * 3

material6 = totProd * 2

Imprimir encabezado

Imprimir 1, material1,
material2, material3,
material4, material5,
material6

Fin Método principal



PRACTICA

Un carpintero necesita un programa que calcule el precio de cualquier escritorio que pida un cliente, basándose en los siguientes datos: longitud y anchura del escritorio, tipo de madera y número de cajones. El precio se calcula como sigue:

- El precio de todos los escritorios es como mínimo de \$220
- Si la superficie es mayor de 4000 cm², añade \$50
- Si la madera es “caoba” añade \$150, y si es “roble” añade \$135
- Por cada cajón que tenga el escritorio añade \$30
- Puede hacer n cálculos hasta que decida salir del programa.

Realice el algoritmo y escríbalo en pseudocódigo





GRACIAS

