

Unidad 2

METODOLOGÍA DE LA PROGRAMACIÓN ALGORITMOS Y ESTRUCTURAS DE DATOS

Objetivos

- Identificación de la estructura interna de la memoria y su relación con los datos.
- Identificación de distintos tipos de memorias y buses y su relación con los datos.
- Identificación de los diferentes tipos de datos.
- Diferenciación de los distintos códigos binarios.
- Identificación de los objetos elementales de un programa.
- Familiarización con las formas de expresar los algoritmos.
- Utilización de los objetos básicos que sirven para guardar los datos de un programa.

Contenidos

- La memoria central del ordenador.
- Los Datos y su representación.
- Identificadores.
- Operadores.
- Variables.
 - Declaración de variables
 - Tipos de Datos de las variables
 - Inicialización de las variables
- Constantes.
 - Literales
 - Constantes simbólicas
 - Tipos de datos de las constantes
- Expresiones.
- Algoritmos.
 - Diseño descendente y algoritmos
 - Algoritmo, instrucción y programa.
 - Características de los algoritmos
 - Estructuras de programación
 - Formas de expresar los algoritmos: pseudocódigo
 - Estructura general de un algoritmo escrito en pseudocódigo
- Instrucciones de entrada.
- Instrucciones de salida.

3

Introducción

La información circula internamente representada mediante el sistema binario con los dígitos 0 y 1.

- **Bit: Binary digit**
- Lógica positiva:
 - 0 ausencia de tensión
 - 1 presencia de tensión
- Lógica negativa
 - 0 presencia de tensión
 - 1 ausencia de tensión

4

La memoria central del ordenador

- Memoria ROM (Read Only Memoria)
 - ❑ Memoria de sólo lectura
 - ❑ No volátil
 - ❑ Adecuadas para almacenar los programas básicos de control del hardware
 - ❑ No puede ser escrita, ya viene grabada de fábrica
- Memoria RAM (Random Access Memory)
 - ❑ Memoria de acceso aleatorio
 - ❑ Puede leerse y escribirse
 - ❑ Es volátil
 - ❑ Memoria de trabajo, desde la que se ejecutan los programas.

5

Estructura interna de la memoria RAM

- La memoria puede verse como un plano dividido en filas que guardan el contenido almacenado y en columnas, de forma que cada casilla contiene un dígito binario.
- Cada fila se diferencia de otra por la situación que ocupa dentro del plano, su dirección.
- Las filas se conocen como celdas.

Siendo:

n líneas del bus de direcciones

M líneas del bus de datos

La capacidad es:

$$C = 2^n * M$$



6

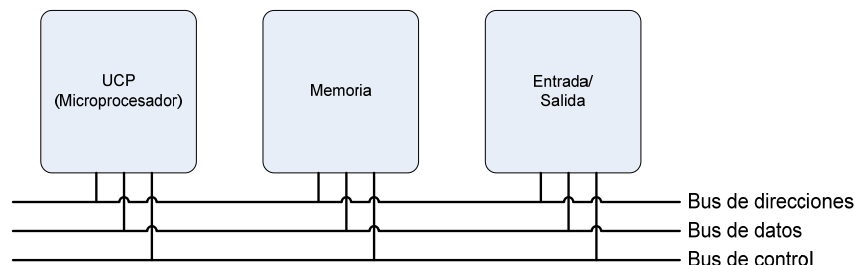
Unidades de capacidad de memoria RAM

- **Byte u octeto.** 1 byte = 8 bits. El byte es la unidad más pequeña que se puede manejar.
- **Kilobyte (KB).** 1KB = 2^{10} bytes.
- **Megabyte (MB).** 1MB = 2^{10} Kilobytes = 2^{20} bytes.
- **Gigabyte (GB).** 1GB = 2^{10} Megabytes = 2^{20} KB = 2^{30} bytes.
- **Terabyte (TB).** 1TB = 2^{10} GB = 2^{20} MB = 2^{30} KB = 2^{40} bytes.
- PB (Petabyte), EB (Exabyte), ZB (Zettabyte), YB (Yottabyte)

7

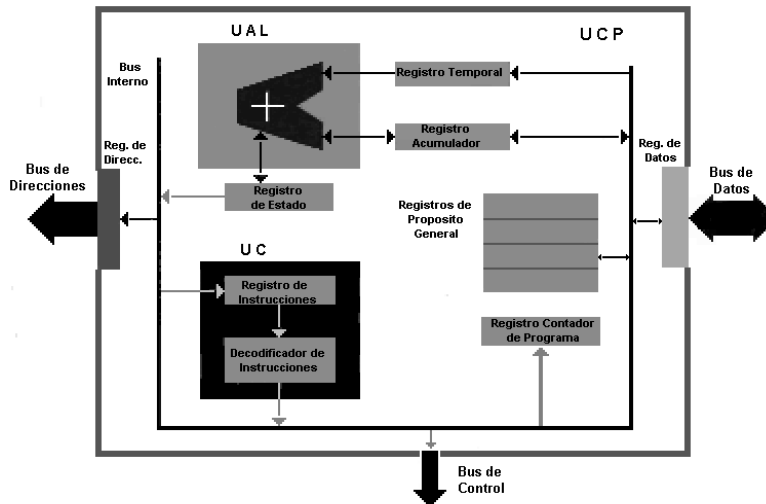
Buses del sistema

- **Bus de datos (DB):** transmite en paralelo en ambos sentidos. Determina las columnas de la memoria y debe corresponderse con el número de bits de datos del procesador.
- **Bus de direcciones (AB):** lleva la dirección de memoria donde se lee o escribe. Es un bus que transmite en un único sentido, del procesador a la memoria.
- **Bus de control (CB):** transmite instrucciones que controlan el flujo de información entre las unidades del sistema.



8

Esquema básico de un micro



9

Los Datos y su Representación

- Dato: Símbolo físico que sirve para registrar hechos, transacciones, acontecimientos,...Ej. 18, Carmen.
- Información: conlleva procesamiento de los datos para que sean útiles y significativos para quien los recibe. Ej. 18 años, 18 euros, Ópera Carmen, mi vecina Carmen
- Tipo de datos: Forma general que toma un conjunto de datos con similares características.
 - El tipo de dato determina su representación interna y las operaciones que podrán realizarse sobre él. Ej. 18 años, 18 euros
- Datos numéricos
 - Enteros
 - Reales
- Datos lógicos
- Datos alfanuméricos
 - Carácter
 - Cadenas

10

Identificadores

- Todos deben ser declarados antes de ser usados.
- Normalmente con un nombre que recuerde a su uso.
Ej. nombreAlumno, precio, notas, IVA.
- Comenzarán por una letra, un subrayado (`_`) o un dólar (`$`)
- Se hace distinción entre mayúsculas y minúsculas. (Antes \neq antes)
- Seguiremos las siguientes normas:
 - CONSTANTES en mayúsculas
 - variables y métodos en minúsculas
 - Clase con la primera letra capital
 - No usar tildes ni espacios en blanco.

Comentarios

- Son notas o comentarios que se añaden a las líneas de código.
- Los utilizaremos de forma generalizada
- Tipos:
 - multilinea, (`/* */`)
 - hasta el final de la línea (`//`)
 - de documentación (`/** */`)

11

Operadores

- Símbolos utilizados para representar los enlaces existentes entre los distintos elementos de una expresión.
- Se usan en programación para construir expresiones válidas.
- Tipos:
 - Aritméticos
 - Relacionales
 - Lógicos o booleanos
 - De asignación
 - Precedencia

12

Operadores aritméticos

binarios

SÍMBOLO	OPERACIÓN
+	Suma
-	Resta
*	Multiplicación
/	División. Si los dividendos son enteros el resultado será entero.
% ó MOD	Resto de la división entera o módulo
↑ ^ **	Potencia (los tres símbolos suelen usarse) En java no existe este operador y utilizaremos la función Pow()

unarios

SÍMBOLO	OPERACIÓN
-	Cambio de signo
++ --	Incremento y decremento

13

Relacionales, booleanos o lógicos y asignación

Relacionales

SÍMBOLO	OPERACIÓN
<	Menor que
==	Igual que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
!= ó <>	Distinto a

Ej. 'h' > 'H' es Falso ('h' **NO** es posterior o igual a 'H') y 5 < > 5 es Falso

Lógicos

SÍMBOLO	OPERACIÓN
&&	Producto lógico (AND)
	Suma lógica (OR)
!	Negación (NOT)

Ej. NO estoy leyendo es Falso

Asignación

SÍMBOLO	OPERACIÓN
= ó ←	Asignación

Ej. edad =35
numero = 16.56
LETRA = 'A'

14

Orden de evaluación de los operadores



Descripción	Operador
Postfijos	Expr++ Expr--
Unarios prefijos	++ Expr -- Expr - Expr + Expr ! Expr
Aritméticos	* / %
Aritméticos	+ -
Desplazamiento	<< >> >>>
Relacionales	< <= > >=
Comparación	== !=
AND lógico	&&
OR lógico	
Asignación	= += -= *= /= %=

◆ En primer lugar los paréntesis, comenzando siempre por los más internos.

◆ Ante igual precedencia, se evalúan de izquierda a derecha.

Ej. $10 > (4 * 2)$ es Verdadero; $(10 > (4 * 2)) \text{ O } ('a' == 'A')$ es V
 $((5+2.5)*(3-5))*(-1)+5$ Resultado 20

15

VARIABLES

- Una variable es una posición de memoria que se referencia con un identificador conocido como *nombre de la variable* y donde se almacena el valor de un dato que puede variar durante la ejecución del programa.
- Deben ser declaradas antes de usadas, escribiendo tipo de datos que almacenará e identificador, según el siguiente formato

□ **Tipo nombresVariables [= <valor inicial>];**

TIPOS de datos de las VARIABLES:

- Numéricas: enteras y reales.
- Alfanuméricas: carácter y cadena.
- Booleanas.

Ejemplos:

enteras edad = 35, numero.
 reales precio = 0.0, IVA = 11.5
 carácter estadoCivil = ' S ', díaSemana.
 cadenas nombreAlumno.
 booleanas ENCONTRADO = Falso.
 numero = 0
 nombreAlumno = " "

INICIALIZACIÓN DE VARIABLES:

- Al declararlas (conveniente).
- En cualquier momento del algoritmo mediante la asignación.

16

Constantes

- Es el nombre simbólico de una posición de memoria donde se almacena el valor de un dato que no puede cambiar durante la ejecución del programa.
- Declaración de una constante en Java:
`final <tipo> <nombre de la constante> = <valor asociado>;`
Ej. `final float PI = 3.14`
A partir de la declaración puede usarse el símbolo
`area = PI * (r ^ 2)`
- Tipos de datos de las constantes: Se asocian al tipo de dato que almacenan.

Cast o conversiones

- Podremos cambiar el tipo de una variable, valor o a otro tipo distinto.
- Cuando el cast es por ensanchamiento no hay que explicitarlo pero cuando es por estrechamiento es obligatorio especificarlo.

17

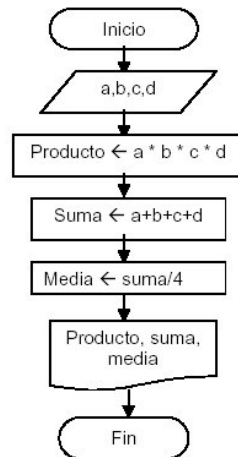
Estructuras de programación

- Lineal o secuencial.
Pelar patatas
Cortarlas a cuadraditos
.....
- Repetitiva o cíclica.
Repetir
Dejar que cuaje
Dar vuelta
Hasta que esté a nuestro gusto
- Alternativa.
Si tenemos de todo
Pelar patatas
.....
Si no
Mensaje: "Debe comprar todos los ingredientes y utensilios"
Finsi

18

ALGORITMOS: diagramas de flujo

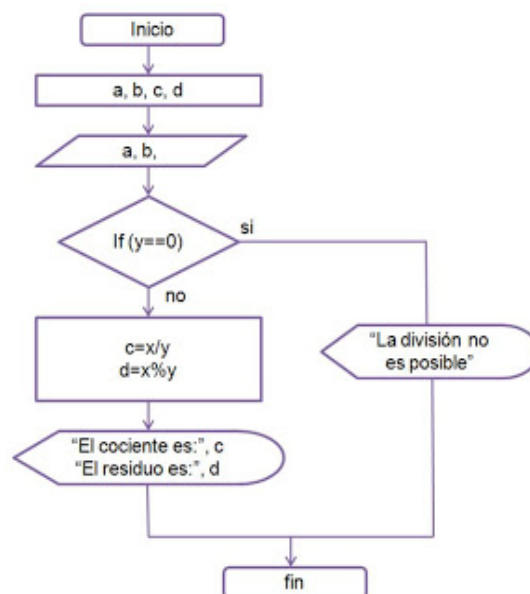
Ej: Calcular suma, producto y media de cuatro números presentando en pantalla los resultados.



19

Algoritmos: diagramas de flujo

EJERCICIO:
Hacer un diagrama de flujo:
Resolución de potencia por
productos sucesivos.



20

Algoritmo: Cambiar rueda al coche I

Suponemos que:

Disponemos de todos los utensilios: coche, rueda, llave y gato
Sabemos: aflojar tornillo, apretar tornillo,...

Inicio

Inmovilizar el coche

Introducir el gato

Repetir

Levantar el coche accionando manivela del gato

Mientras que la rueda roce el suelo.

Repetir

Aflojar un tornillo de la rueda

Mientras queden tornillos por aflojar

Quitar los tornillos

Quitar la rueda

Poner rueda de repuesto

Repetir

Apretar un tornillo de la rueda

Mientras queden tornillos por apretar

Repetir

Bajar el coche accionando manivela del gato

Mientras que el gato soporte peso.

Fin

21

Algoritmos y diseño modular

Algoritmo: Cambiar rueda al coche II

Suponemos que:

No sabemos si disponemos o no de los utensilios

Sabemos: aflojar tornillo, apretar tornillo,...

Inicio

Si no disponemos de los utensilios

realizar el módulo <Contactar con seguro>

Si disponemos de los utensilios

realizar el módulo <Cambiar rueda al coche I>

Fin

< Contactar con seguro >

inicio

Repetir

Llamar por teléfono al seguro

Mientras no consigamos contactar

Dar parte del siniestro

Fin

<Cambiar rueda al coche I> ---está realizado en diapositiva anterior

22

Diseño descendente

- Filosofía de diseño: **“Divide y vencerás”**. Dividir el problema, **módulo o programa principal**, en otros más pequeños llamados **subprogramas** y estos a su vez en otros más pequeños y así sucesivamente retrasando los detalles tanto como sea posible.
- Técnica a aplicar: **programación estructurada**
- Algoritmos:
 - Descripción de datos: constantes, variables y otros objetos
 - Descripción de instrucciones
- Características:
 - Eficientes
 - Legibles
 - Modificables
 - Modulares
 - Único punto de entrada, único punto de salida

23

ALGORITMOS: tablas de decisión

Muchos procesos de toma de decisiones pueden ser tratados por medio de **tablas de decisión**, en las que se representan los elementos característicos de estos problemas:

- Los diferentes **estados** que puede presentar la naturaleza: e_1, e_2, \dots, e_n .
- Las **acciones** o **alternativas** entre las que seleccionará el que decide: a_1, a_2, \dots, a_m .
- Las **consecuencias** o **resultados** x_{ij} de la elección de la alternativa a_i cuando la naturaleza presenta el estado.

Ejem. Toma de decisión sobre el importe a pagar estudiando varias posibilidades:

CONDICIONES	1	2	3	4
¿Paga contado?	S	S	N	N
¿Compra > \$ 50000?	S	N	S	N
ACCIONES				
Calcular descuento 5% s/importe compra	X	X		
Calcular bonificación 7% s/importe compra	X		X	
Calcular importe neto de la factura.	X	X	X	X

24

ALGORITMOS: tablas de decisión

Acabamos de echar cinco huevos en un tazón con la intención de hacer una tortilla. Disponemos, además, de un sexto huevo del que no conocemos su estado, aunque es de esperar que en caso de encontrarse en buen estado y no ser utilizado, se estropeará. Se presentan tres posibles alternativas:

- Romper el huevo dentro del tazón donde se encuentran los cinco anteriores.
- Romperlo en otro tazón diferente.
- Tirarlo directamente.

Alternativas	Estado del 6º huevo	
	Bueno (e_1)	Malo (e_2)
Romperlo dentro del tazón (a_1)	Tortilla de 6 huevos	5 huevos desperdiciados y no hay tortilla
Romperlo en otro tazón (a_2)	Tortilla de 6 huevos y un tazón más que lavar	Tortilla de 5 huevos y un tazón más que lavar
Tirarlo (a_3)	Tortilla de 5 huevos y un huevo bueno desperdiciado	Tortilla de 5 huevos

VALORACIÓN DE LOS RESULTADOS

Aunque los resultados x_{ij} no son necesariamente números (como ocurre en el ejemplo anterior), supondremos que el decisor puede valorarlos numéricamente, es decir, se asumirá la existencia de una función $V(\cdot)$ con valores reales tal que:

$V(x_{ij}) > V(x_{ik})$ si y sólo si el decisor prefiere el resultado x_{ij} al resultado x_{ik}

Realizaremos un proceso de valoración en el que se asignasen números a cada una de los resultados, dando lugar a una posible tabla como la que sigue:

EJERCICIO:

Hacer tabla de decisión:
Resolución de potencia por
productos sucesivos.

	e_1	e_2
a_1	10	0
a_2	8	6
a_3	5	7

Por motivos de simplicidad identificaremos cada resultado con su valoración numérica. Así, x_{ij} hará referencia tanto al propio resultado como al valor asignado por el decisor.

25

Algoritmos: pseudo-código

EJERCICIO:

Hacer un pseudocódigo:
Resolución de potencia por
productos sucesivos.

ENTORNO:

VARIABLES:

entero NUM1, NUM2, suma
carácter respuesta

PROGRAMA PRINCIPAL:

<Inicio>

//¿Desea realizar la suma?

Escribir (¿Quiere realizar la operación de sumar?)
Leer (respuesta)

//Fin Desea....

mientras(respuesta = != 'S')

//Obtener datos

Escribir("Introduzca dos números enteros")
Leer (NUM1, NUM2)

//Fin Obtener datos

//Realizar Suma

suma = NUM1 + NUM2

//Fin Realizar Suma

//Mostrar resultado...

Escribir("El resultado de la suma es: ", suma)

//Fin Mostrar....

//¿Desea realizar la suma?

Escribir (¿Quiere realizar la operación de sumar?)
Leer (respuesta)

//Fin Desea....

Fin_mientras

Fin_PP

26

Instrucciones de entrada

■ Leer (nombresVariables)

Ejemplos: Dadas las siguientes declaraciones:

enteras edad, numero.
real precio, IVA, totalPagar = 0.
carácter estadoCivil, diaSemana.
cadena nombreAlumno.

Instrucción	Datos de entrada	Contenido tras la lectura
leer (edad, IVA)	35 11.5	edad — 35, IVA — 11.5
leer (precio)	25 25	precio — 25 25
leer (estadoCivil, numero)	S 3	estadoCivil — 'S', numero — 3
leer (nombreAlumno)	Juan Vila	nombreAlumno — "Juan Vila"
leer (diaSemana)	L	diaSemana — 'L'

Instrucciones de salida

■ Escribir (mensajes , nombresVariables)

Instrucciones	Salida (lo que se imprime en pantalla)
escribir ("Cálculos realizados")	Cálculos realizados
totalPagar = precio +(precio * IVA)/100	totalPagar= 28.15375
escribir ("Pagará por el artículo", totalPagar)	Pagará por el artículo 28.15375
escribir ("El iva aplicado es:", IVA)	El IVA aplicado es: 11.5