

Estructuras y Tipos de Datos



Definiciones Básicas

TEMA I - Definiciones Básicas

CONTENIDO

- Nociones Preliminares
- Ciclo De Desarrollo De Programas
- Datos
- Tipo Abstracto De Datos
- Tipos de Datos
- Estructura de Datos



Nociones Preliminares

DATO

*Hecho, acción, punto, referencia que por si sólo no reviste ninguna relevancia, a menos que se interrelacione con otros datos o hechos que procesados producen **información**.*

Ejemplo:

12 → ???

12 + mes → información (diciembre)

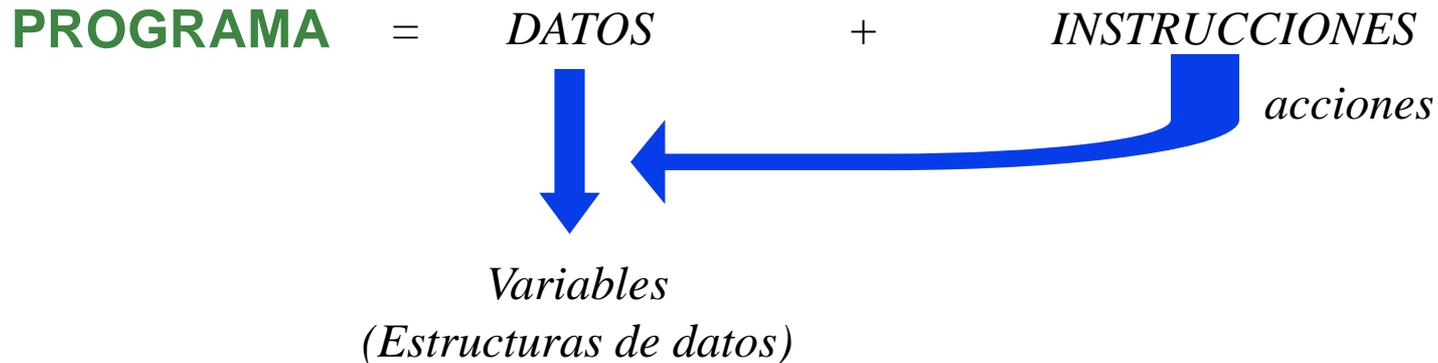
INFORMACION

Conjunto de datos relacionados entre sí y procesados



Obtendremos información después que los datos son procesados

Nociones Preliminares



VARIABLE

Es la definición de un espacio físico en la memoria del computador, con capacidad de almacenar datos; de acuerdo al tipo de dato de la variable, ésta tendrá diferente estructura y tamaño, ya que su representación será diferente

Nociones Preliminares

PROGRAMA

Para escribirlo

ALGORITMOS

ALGORITMO

Es una secuencia finita de pasos, cada una de los cuales tiene un significado preciso y puede “ejecutarse” en un tiempo finito. Además un algoritmo es independiente de los valores de entrada y debe tener fin.

INFORMAL

*Escrito en Leng.
Natural*

PSEUDO-CODIGO

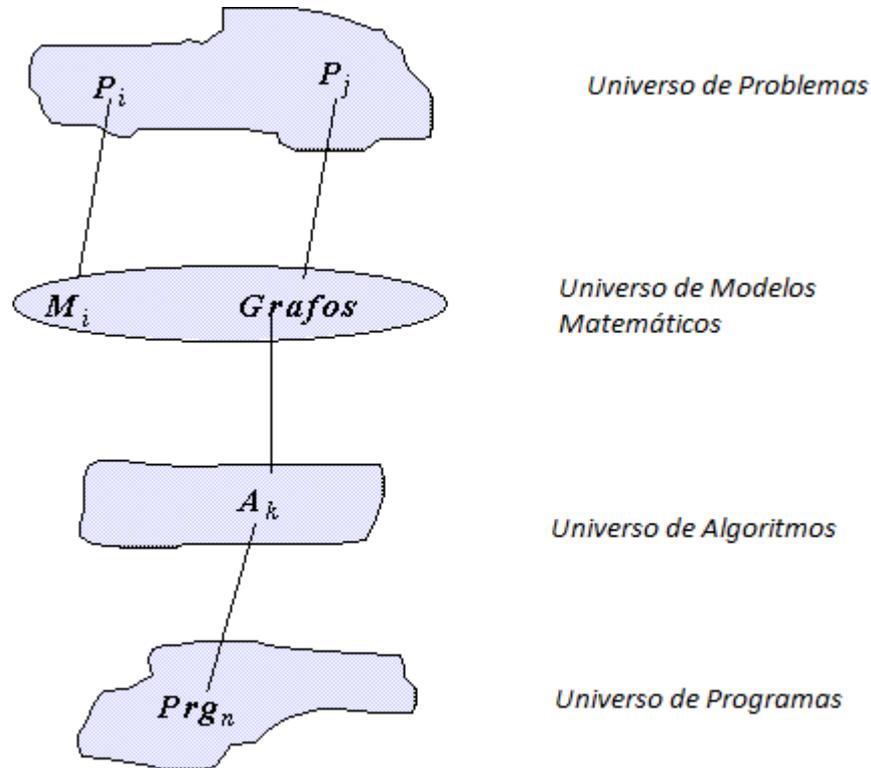
*Mezcla leng. Natural con formalidad y reglas
sintácticas de leng. de prog.*

PROGRAMA

*Escrito en Leng. de
Prog.*

Nociones Preliminares

PROCESO DE ABSTRACCION



Lógica que seguimos al tratar de dar solución a los problemas del mundo real, utilizando la ciencia de la computación

Nociones Preliminares

MODELO

Es una representación simplificada de la realidad. Es una abstracción y simplificación de un problema real, que de manera ideal incorpora los elementos y relaciones esenciales del problema real (BIERMAN, BONINI, HAUSMAN: 1994)

CLASIFICACION DE ACUERDO A SU REPRESENTACION

Icónicos

Analógicos

Simbólicos

Nociones Preliminares

Modelos Icónicos

Son aquellos que representan algunos aspectos de un Sistema, en forma visual o gráfica (maqueta)

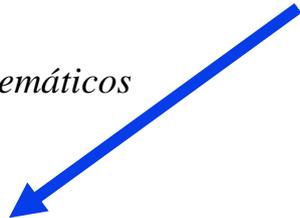
Modelos Analógicos

Utilizan un conjunto de características para representar otro conjunto de características que posee el objeto de interés.

Modelos Simbólicos

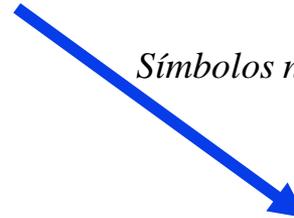
Emplean símbolos para representar las propiedades de la entidad estudiada

Símbolos matemáticos



Modelos Matemáticos

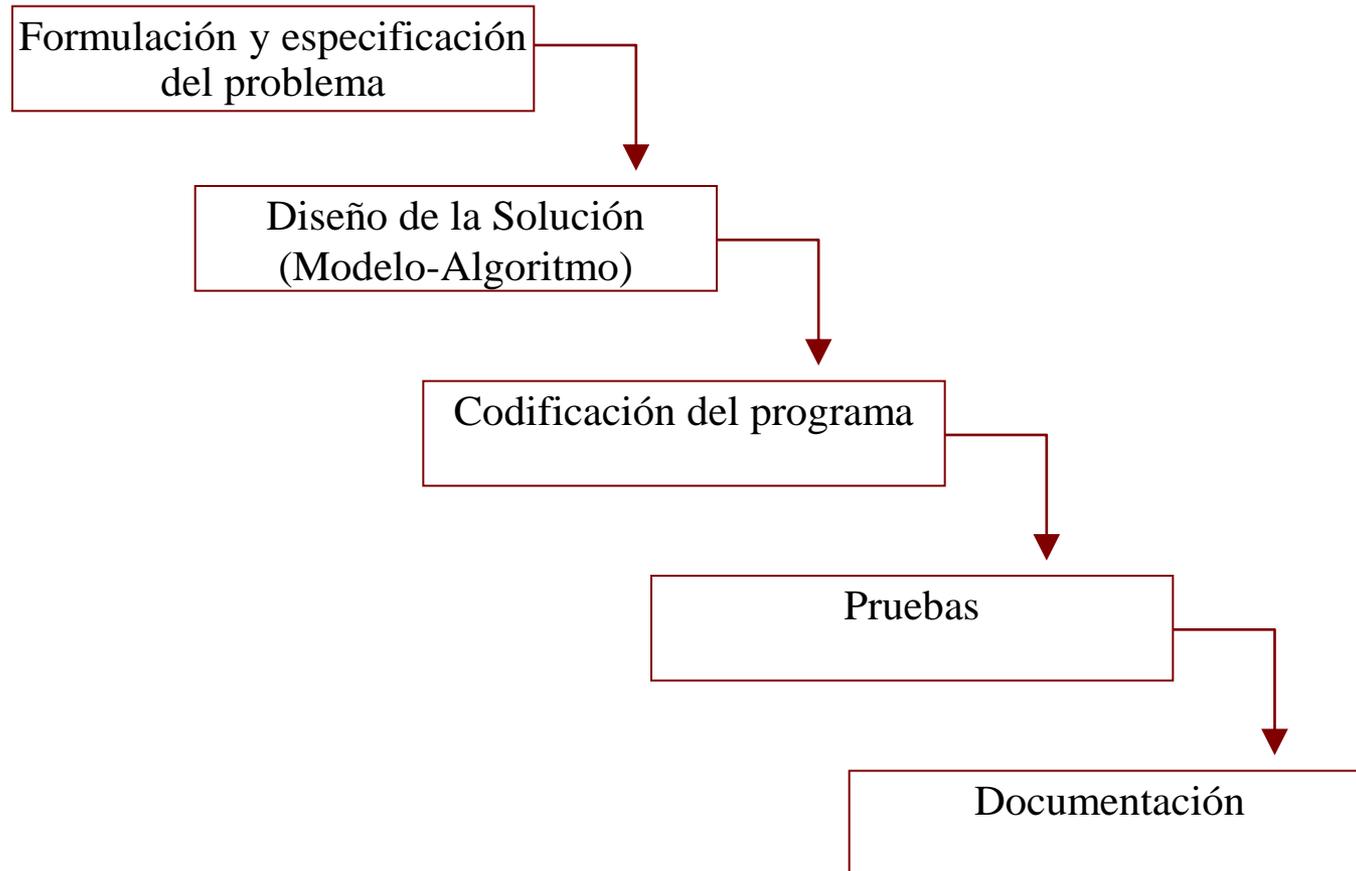
Símbolos no matemáticos



Modelos Gráficos

Ciclo de Desarrollo de Programas

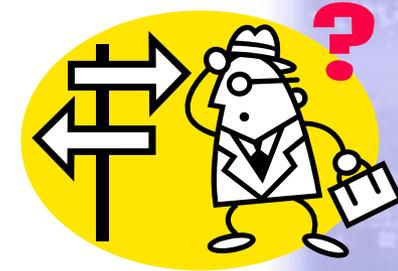
MODELO EN CASCADA



Ciclo de Desarrollo de Programas

1.- Formulación y especificación del problema

- *Análisis del Sistema*
- *Análisis de los requisitos del programa (QUE)*



Lista de Requerimientos

2.- Diseño de la Solución

- *Se concentra en el **COMO***
- *Se enfoca básicamente en*



la estructura de los datos

la arquitectura del programa

*la caracterización de la interfaz
(diseño de pantallas, reportes, etc.)*

el detalle procedimental (algoritmos)

Ciclo de Desarrollo de Programas

3.- Implementación (Codificación)

Se toma el diseño y se traduce a un lenguaje de programación



programa de computación o software



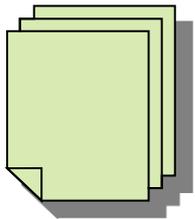
4.- Pruebas

- *Se concentra en la lógica interna del software*
- *Procura asegurar que el software*



- * *Cubra los requisitos*
- * *Produzca los resultados esperados*

5.- Documentación



Se realiza durante todo el desarrollo del programa



Interna



Externa

Diseño de la Solución

... Mas del Diseño

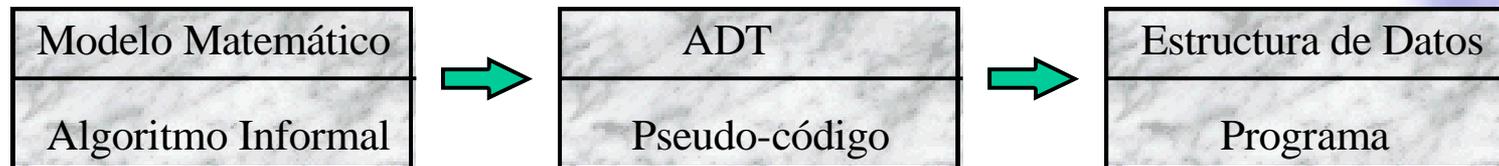
Se debe prestar toda la atención posible al diseño de la solución, en esta etapa se define la forma en que se representará la solución



Se analizan, entre otras cosas:

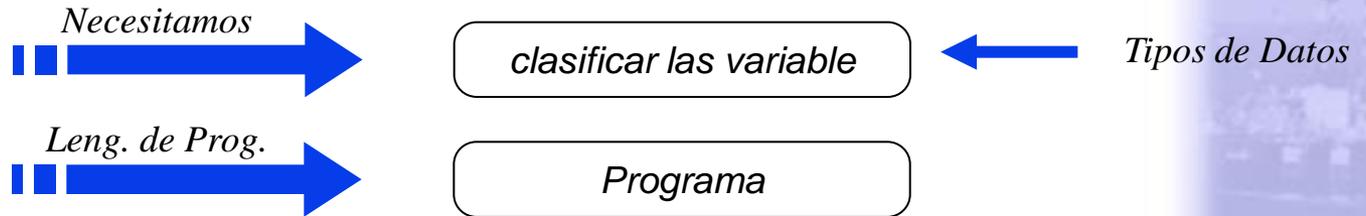
- Los datos, su definición e interrelación*
- La forma como estarán organizados*

UN MODELO SENCILLO PARA EL DISEÑO DE ESTRUCTURAS DE DATOS



Datos

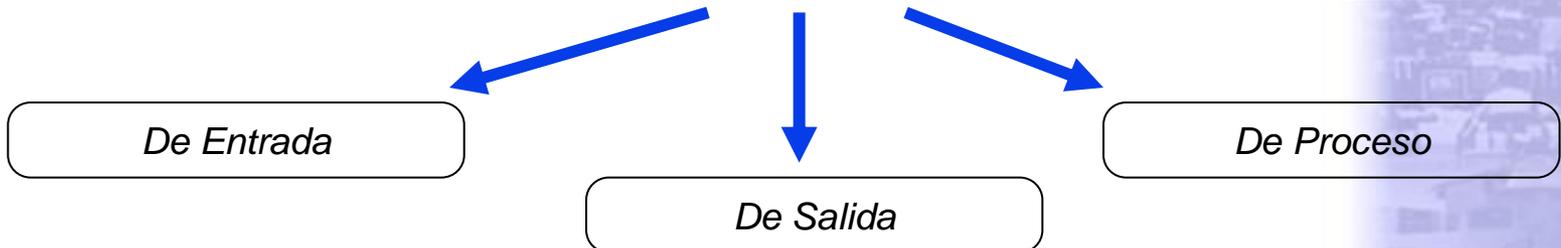
Algoritmos



Los **programas**, son formulaciones concretas de algoritmos abstractos basados en ciertas representaciones y "estructuras de datos". Están constituidos básicamente de **instrucciones** y **datos**

Datos

Los Datos, teóricamente ante todo, son abstracciones de fenómenos reales y se formulan de preferencia como estructuras abstractas que no necesariamente se elaboran en lenguajes comunes de programación



Identificados los datos del problema, deben ser representados en el prog. como variables de memoria

Tipo Abstracto de Datos

Introducción

Los tipos básicos de datos que ofrecen los lenguajes de programación no son suficientes para representar la gran variedad de soluciones de los problemas planteados



*El Programador define
sus propios T.D.*

Tipo Abstracto de Datos

Concepto parecido al de un procedimiento, está formado por datos e instrucciones los cuales están interrelacionados y encapsulados.

Tipo Abstracto de Datos

Definición

Un Tipo Abstracto de Datos (TAD) como un modelo matemático con varias operaciones definidas sobre el modelo

Ejemplo

Conjunto de enteros.

Modelo: *Podemos modelar un conjunto por extensión o por compresión*

Operaciones:

Unión(A,B): tomamos los elementos comunes y no comunes de A y de B

Intersección(A,B): Tomamos los elementos comunes de A y B.

Diferencia(A,B): Tomamos los elementos que están en A y no están en B.

El programador define en el T.A.D. solo aquellas operaciones que necesita en el programa.

Tipo Abstracto de Datos

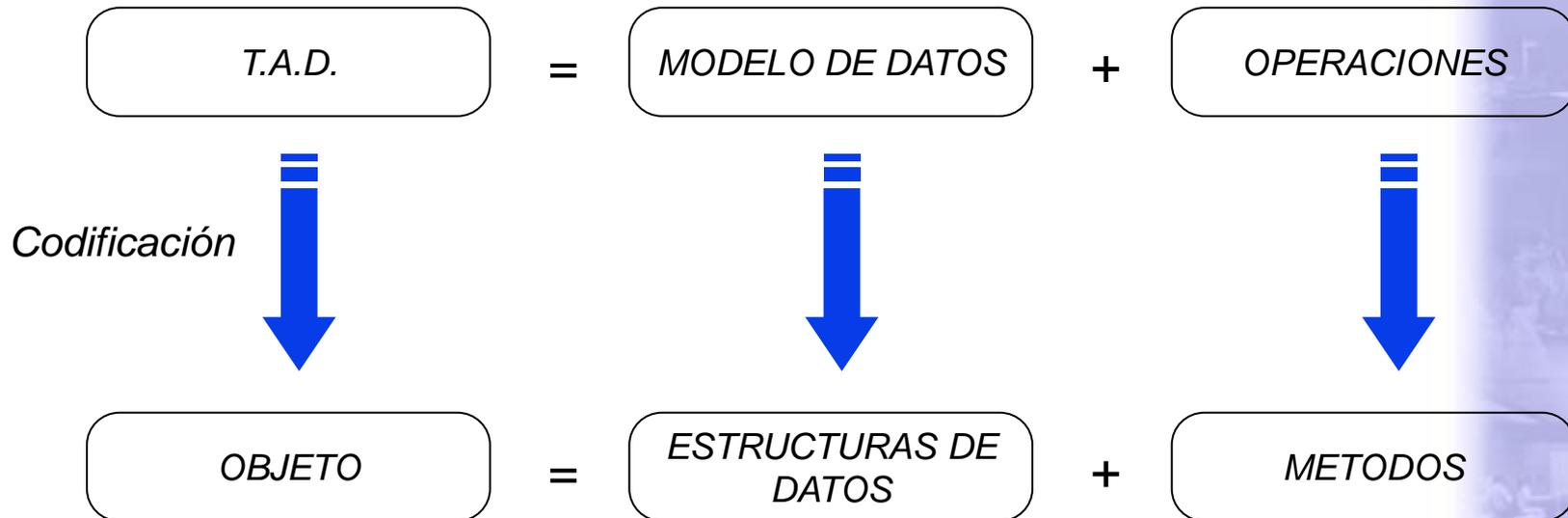
Características

- **Generalización de tipos primitivos:** el programador define sus propias operaciones.
- **Encapsulamiento de subtarefas en un lugar del TAD:** los procedimientos y funciones así como los datos se ubican físicamente en un lugar específico del programa y el acceso a los datos se realiza solo por las operaciones definidas en el TAD, es decir, los datos no pueden ser tocados por procedimientos o funciones que estén fuera del TAD

Definido el Tipo Abstracto de Datos, se requiere que se implemente o codifique

Tipo Abstracto de Datos

Implementación



Tipos de Datos

Definición

El tipo de dato de una variable de un programa es el conjunto de posibles valores que una variable de ese tipo puede tomar

Ejemplo

```
Var    B: boolean;  
       C: char;
```

Los tipos de las variables B y C son lógicos y caracteres respectivamente que



B: sólo puede tomar valores TRUE y FALSE.

C: sólo puede tomar valores carácter del juego de caracteres definidos en la tabla de símbolos o caracteres manejados por el lenguaje de programación.

Tipos de Datos

... Cont

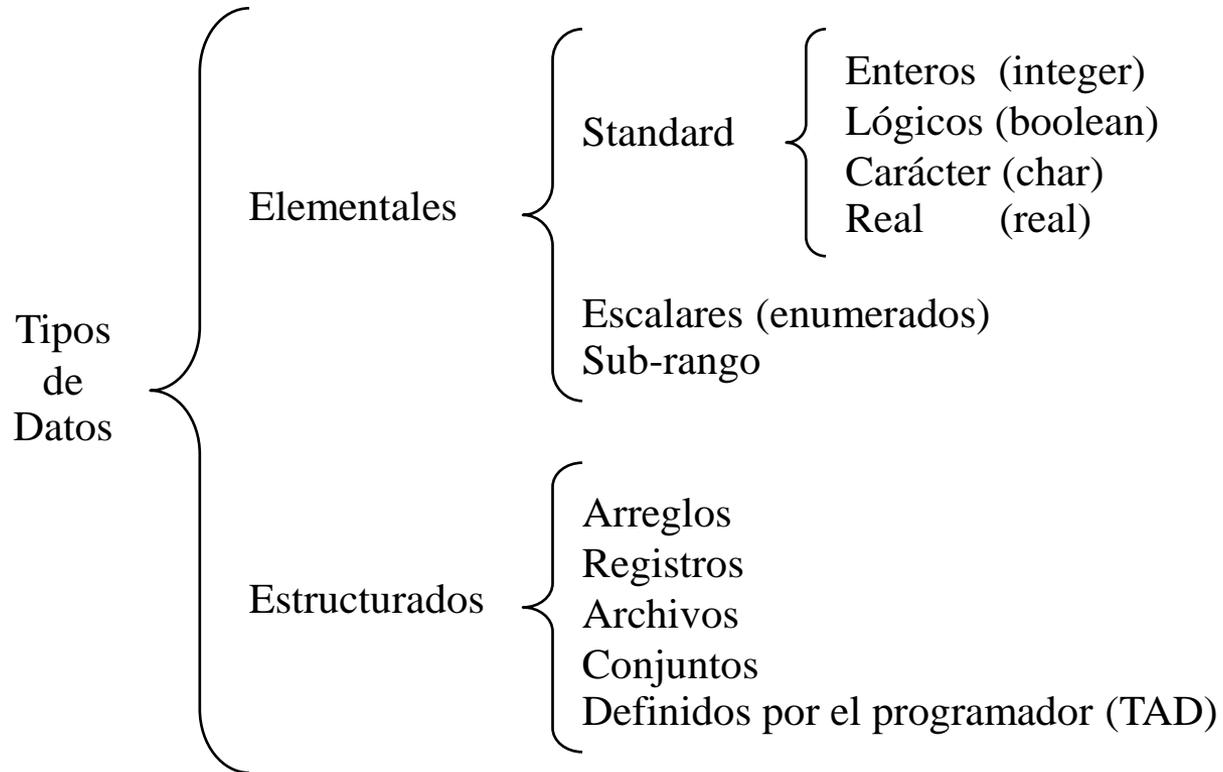
El T.D. no define físicamente la variable ni le asigna valor alguno, lo que define son los valores que esta puede tomar

El número de valores que pertenecen a un tipo T se conoce como la cardinalidad del tipo, esta sirve de base para determinar la cantidad de espacio físico que se necesita para almacenar una variable de ese tipo.

Tipos de Datos

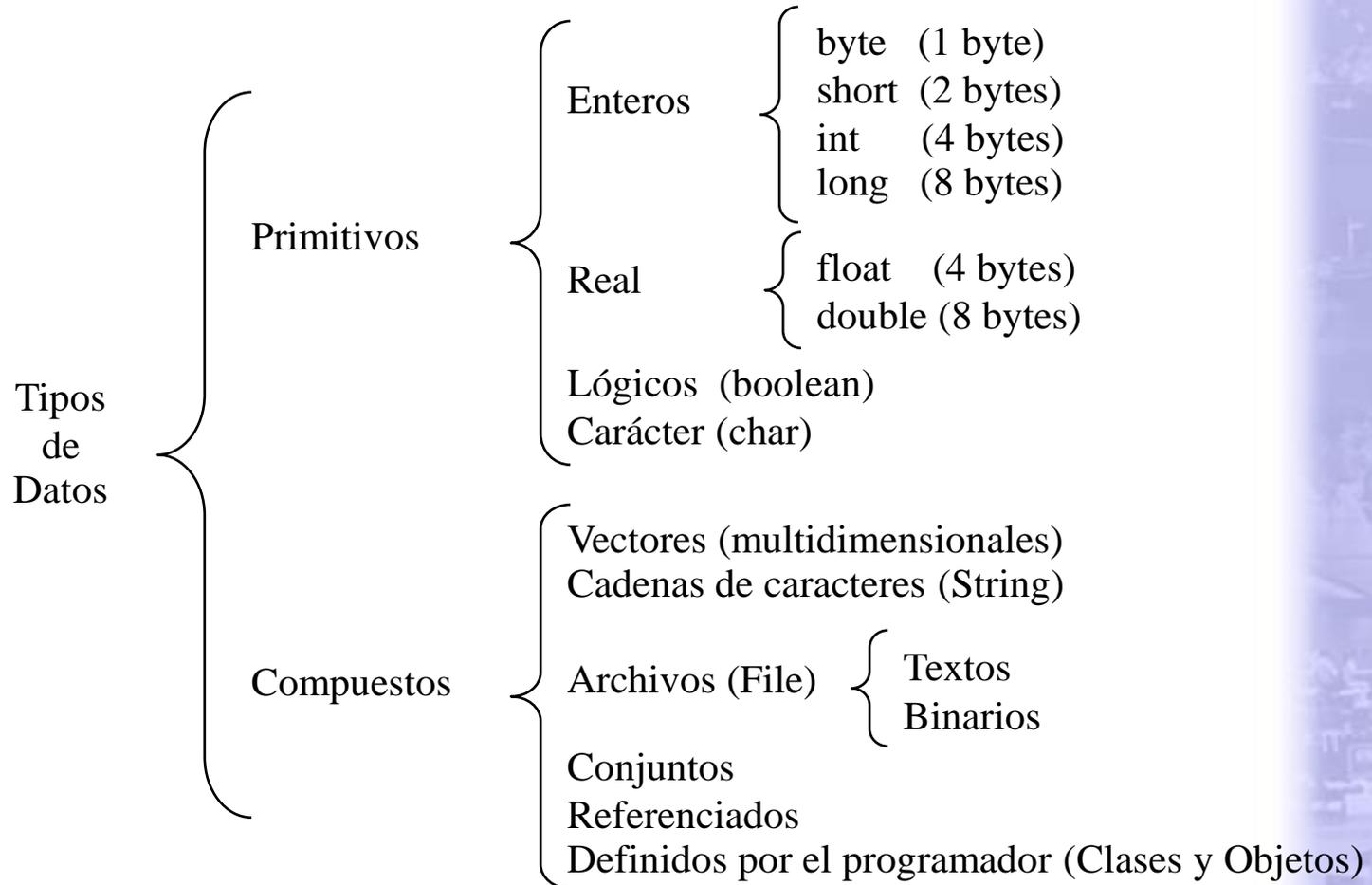
Clasificación (Base Pascal)

Los tipos básicos de datos dependen de los lenguajes de programación y varían de uno a otro



Tipos de Datos

Clasificación (Base Java)



Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Carácter (Char)

- Surge de la necesidad de manejar constantes y variables cuyos valores sean caracteres aislados.
- Variable declarada tipo char solo puede tomar valores que son caracteres de nuestro juego de caracteres
- Los dos conjuntos de caracteres más usados son:

ASCII (o ISO)

compuesto por 128 caracteres

ASCII extendido

compuesto por 256 caracteres

EBCDIC

con 256 fue diseñado por IBM.

- La Representación Interna es binaria.
- La memoria ocupada por una variable tipo char es de 1 byte (8 bits)

Ejemplo

CONST	blanco = " ";
	por = "***";
VAR	letra: char;

Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Lógico (Boolean)

- *Surgen de la necesidad de manejar variables que puedan asumir solo 2 estados*
- *La Representación Interna es binaria.*
- *La memoria ocupada es de 1 byte, aunque de ser posible podría representarse en 1 bit.*

Ejemplo

VAR

varlogica: boolean

Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Entero (Integer)

- Surgen de la necesidad de manejar variables cuyos valores posibles sean enteros
- La Representación Interna es binaria.
- La memoria ocupada es de es de 2 bytes
- Computador tiene memoria finita, entonces se representa un subconjunto del conjunto Z . Por lo general se utilizan 2 bytes para su representación lo que nos permite definir los enteros entre -32.765 y 32767 aproximadamente
- Algunos lenguajes de programación ofrecen representación en 1, 2, 4 y 8 bytes

Ejemplo

VAR

varentera: integer

Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Real (real)

- Se utilizan para representar variables que almacenan valores reales
- La representación interna es punto flotante
- La memoria ocupada varía dependiendo del compilador y de la arquitectura de la máquina
- La representación punto flotante utiliza tres elementos para definir un número: base, exponente y mantisa (es similar a la notación científica),

$$N = \text{mantisa} * (\text{Base})^{(\text{exponente})}$$

Ejemplo

2832	=	$2832 * 10^0$
0,2832	=	$2832 * 10^{-4}$
2832000	=	$2832 * 10^3$
-28,32	=	$-2832 * 10^{-2}$

Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Real (real) ... Cont

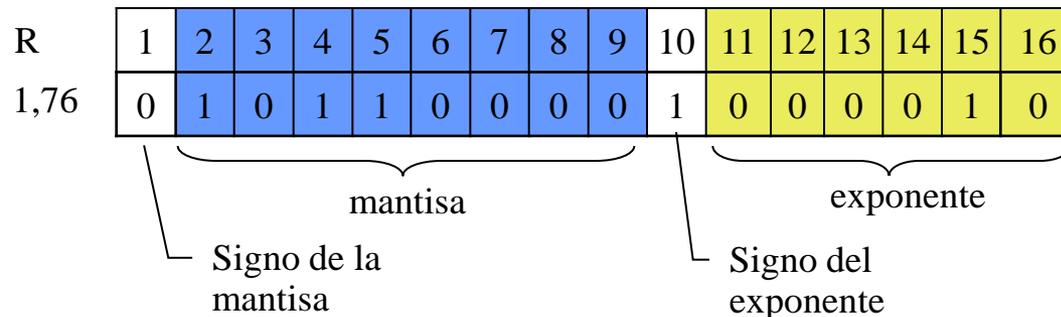
- Internamente para su representación se disponen unos bits para la mantisa, otros para el exponente, además de 1 bit para el signo de la mantisa y 1 bit para el signo del exponente.

Ejemplo

Vamos a almacenar 1,76 y se utilizan 16 bits para representar números reales, entonces

Número real: $1,76 = 176 * 10^{-2}$

Representación interna



Tipos de Datos

Tipos de Datos Elementales – Tipos Standard

Tipo Real (real) ... Cont

Características:

Magnitud: está referida a cuan grande es el número que podemos representar (por ejemplo 10^{75})

Precisión: consiste en la cantidad de dígitos que podemos representar del número

Dos tipos de Errores:

Error representacional: se produce cuando no podemos representar todos los dígitos de un número real en la variable

Error computacional: ocurre cuando se opera con números de magnitudes muy disímiles originando que se pierdan algunos datos intermedios al no poseer el computador capacidad de memoria suficiente para manejarlos simultáneamente

Tipos de Datos

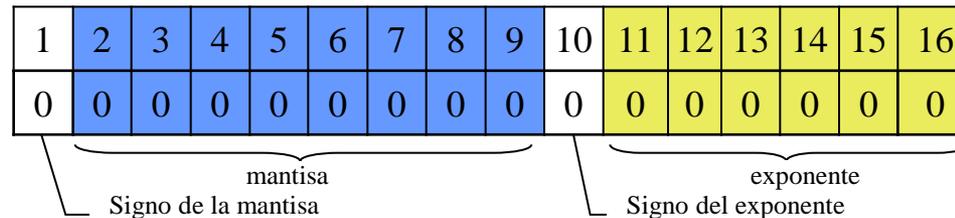
Tipos de Datos Elementales – Tipos Standard

Tipo Real (real) ... Cont

Ejemplos

Error representacional:

Supongamos la siguiente representación interna de los números reales



Queremos almacenar el 15,56

$$\text{Punto flotante}(15,56) = 1556 * 10^{-2}$$

La mantisa tiene 8 bits

*por lo tanto podemos representar el 15 y el 16 pero no el 15,56
porque 1556 no se puede escribir con 8 bits*

Tipos de Datos

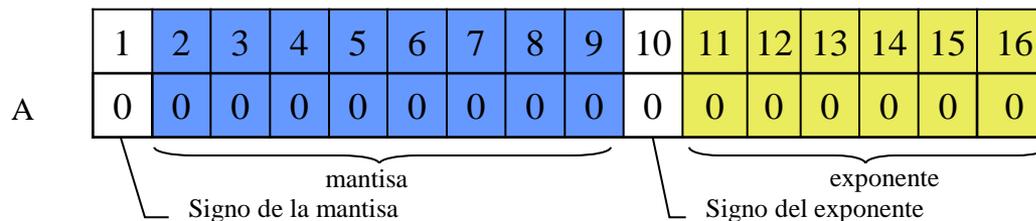
Tipos de Datos Elementales – Tipos Standard

Tipo Real (real) ... Cont

Ejemplos

Error computacional:

Supongamos la siguiente representación interna de los números reales



Queremos realizar la siguiente operación: $2 + 0,76 = 2,76$

Punto flotante(2) = $2 * 10^0$

Punto flotante(0,76) = $76 * 10^{-2}$

La mantisa tiene 8 bits,

por lo tanto podemos representar el 2 y el 76 pero no el 2,76

porque Punto flotante(2,76) = $276 * 10^{-2}$

y 276 no se puede escribir con 8 bits

Tipos de Datos

Tipos de Datos Elementales – Tipos Escalares

- *Un tipo escalar es un identificador de tipo al cual se le define un conjunto de identificadores que indican los valores que pueden tomar una variable de ese tipo*
- *Los únicos valores que se le pueden asignar a una variable de este tipo son los especificados en la lista de identificadores*
- *El conjunto de identificadores se puede utilizar como constantes en el programa*
- *La representación interna del tipo escalar es binaria, similar a los enteros*

Ejemplo

```
TYPE moneda = (lira, bolívar, peso, peseta, dollar)
           dia = (Lunes, martes, Miercoles, Jueves, Viernes, Sabado, Domingo)

VAR a : moneda;      d : dia;

    a := bolívar;
```

- *Son tipos enumerados, se comportan como los enteros*

Tipos de Datos

Tipos de Datos Elementales – Tipos Subrango

- Nos permiten definir variables cuyos valores están dentro de ciertos límites.

Ejemplo

```
CONST lim_inf = 1000
      lim_sup = 2000;

TYPE  V = lim_inf .. lim_sup
      Var  Mes: 1 .. 12;
      Año: V;
```

- Los límites inferior y superior del subrango deben ser constantes
- Se comportan igual a los enteros. Su representación interna es binaria

Los tipos enteros, carácter, escalares, subrango y lógicos se consideran también tipos ordinales, porque cada valor de un tipo ordinal tiene un único predecesor y un único sucesor.

Tipos de Datos

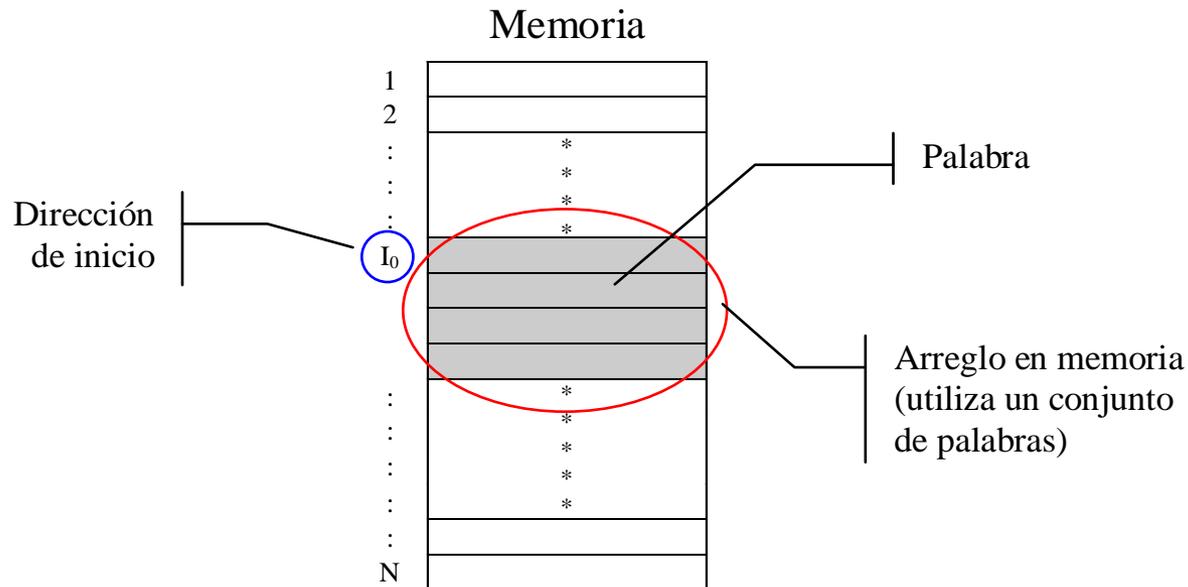
Tipos de Datos Estructurados - Arreglo

Es un tipo estructurado conformado por un conjunto de componentes todas del mismo tipo, llamado tipo base T .

Organización secuencial y contigua en memoria.

Acceso aleatorio a través de un índice que indica el componente que se desea acceder en el arreglo

Representación Interna: Mapeo del arreglo (abstracto) con componentes de tipo T en el almacenamiento, que es un arreglo con componentes de tipo palabra.



Tipos de Datos

Tipos de Datos Estructurados - Arreglo

El tamaño del arreglo depende del tamaño del tipo base



*Memoria ocupada = $n*s$*

donde

n : número de componentes

s : número de palabras que ocupa un componente (tamaño del tipo base)

Dirección i del j -ésimo componente

$$i = i_0 + j*s$$

donde

i_0 : dirección de inicio del arreglo

j : componente que se desea acceder.

s : número de palabras que ocupa un componente

Tipos de Datos

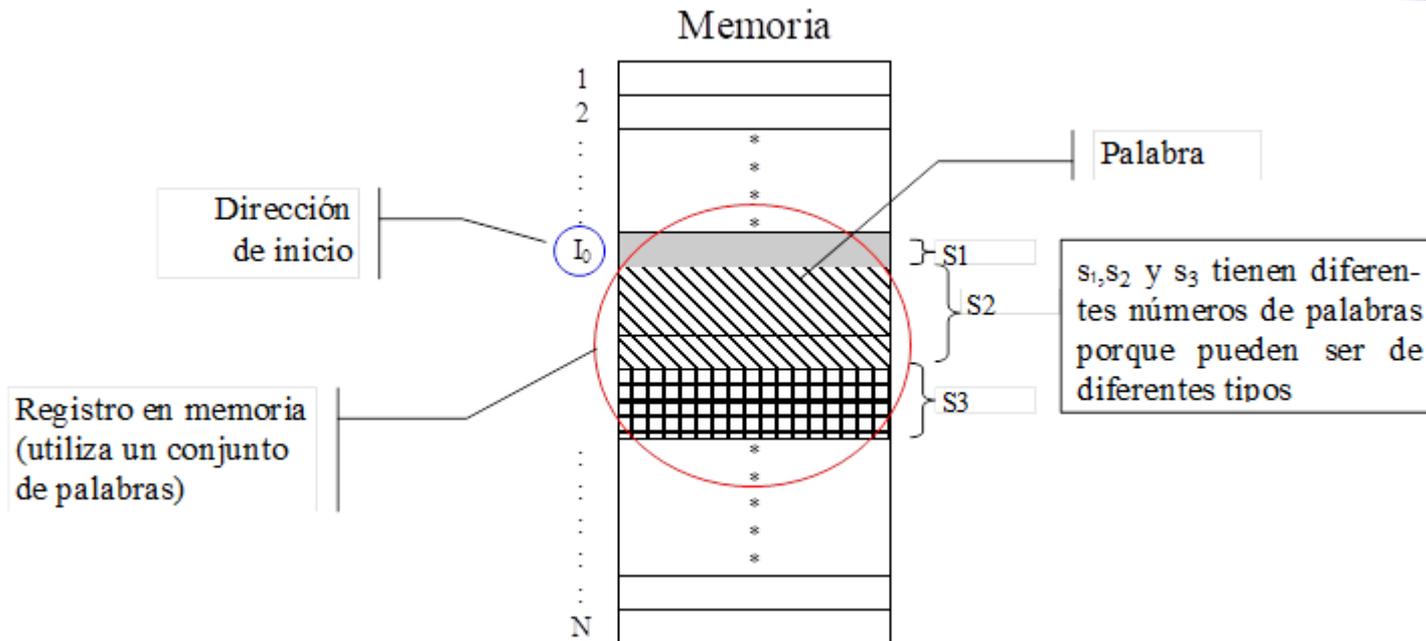
Tipos de Datos Estructurados - Registro

Es un tipo estructurado conformado por el producto cartesiano de varios tipos. Los componentes suelen llamarse campos del registro.

Organización de las componentes del registro es contigua en memoria

Acceso se realiza enunciando la variable tipo registro mas el campo que se desea acceder.

Representación Interna: se mapean en memoria por la simple yuxtaposición de sus componentes.



Tipos de Datos

Tipos de Datos Estructurados - Registro

El espacio ocupado en memoria se obtendría sumando el tamaño de todos los componentes que tiene el registro.



Memoria ocupada = $s_1 + s_2 + \dots + s_n$

donde

*n : número de componentes
 s : número de palabras que ocupa un componente (tamaño del tipo base)*

La dirección k_i del componente o campo s_i

$$k_i = s_1 + s_2 + \dots + s_{i-1}$$

donde

*s_j : tamaño del j -ésimo componente $1 \leq j \leq i-1$
 i : componente que se desea acceder.
 k_i : dirección relativa al origen o inicio del registro.*

Tipos de Datos

Tipos de Datos Estructurados - Conjunto

Es una colección de elementos todas del mismo tipo T , constituyendo T el tipo base del conjunto.

Acceso mediante la inspección de sus elementos a través de la relación de pertenencia.

El tipo base de un conjunto debe ser elemental.

Se representan por medio de una función característica $C(S)$.



Arreglo de valores lógicos, cuyo i -ésimo componente especifica la presencia o ausencia del valor i en el conjunto, es decir, significa " i pertenece o no al conjunto S ".

$$C(s_i) \equiv (i \text{ está en } S) \\ i \text{ in } S \quad (\text{in es el operador de pertenencia de Pascal})$$

El tamaño del arreglo está determinado por la cardinalidad del tipo base

Tipos de Datos

Tipos de Datos Estructurados - Conjunto

Ejemplo

Sea $0..9$ el universo de un conjunto S , (los valores posibles están entre 0 y 9)

Si $S = \{3, 4, 7\}$, la función característica $C(S)$ de S sería

$$C(S) = (F \ F \ F \ T \ T \ F \ F \ T \ F \ F) \\ \quad \quad \quad 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$$

Esto lo representaremos como un arreglo lógico, con un tipo base $0..9$, de la siguiente manera

$C(S).$ =

0	1	2	3	4	5	6	7	8	9
F	F	F	T	T	F	F	T	F	F

Si realizamos $C(1) = 1 \text{ está en } S = F$
 $C(4) = 4 \text{ está en } S = V$

En memoria la secuencia de valores lógicos se representa como una cadena de bits. Esto permite realizar operaciones básicas mediante prueba de bits muy eficientemente.

Se recomienda usar un tipo conjunto sólo en caso de tipos base pequeños

Estructuras de Datos

Estructura

Definición 1: *Distribución y orden de las partes.*

Definición 2: *Armadura que sustenta alguna cosa.*

Estructura de Datos

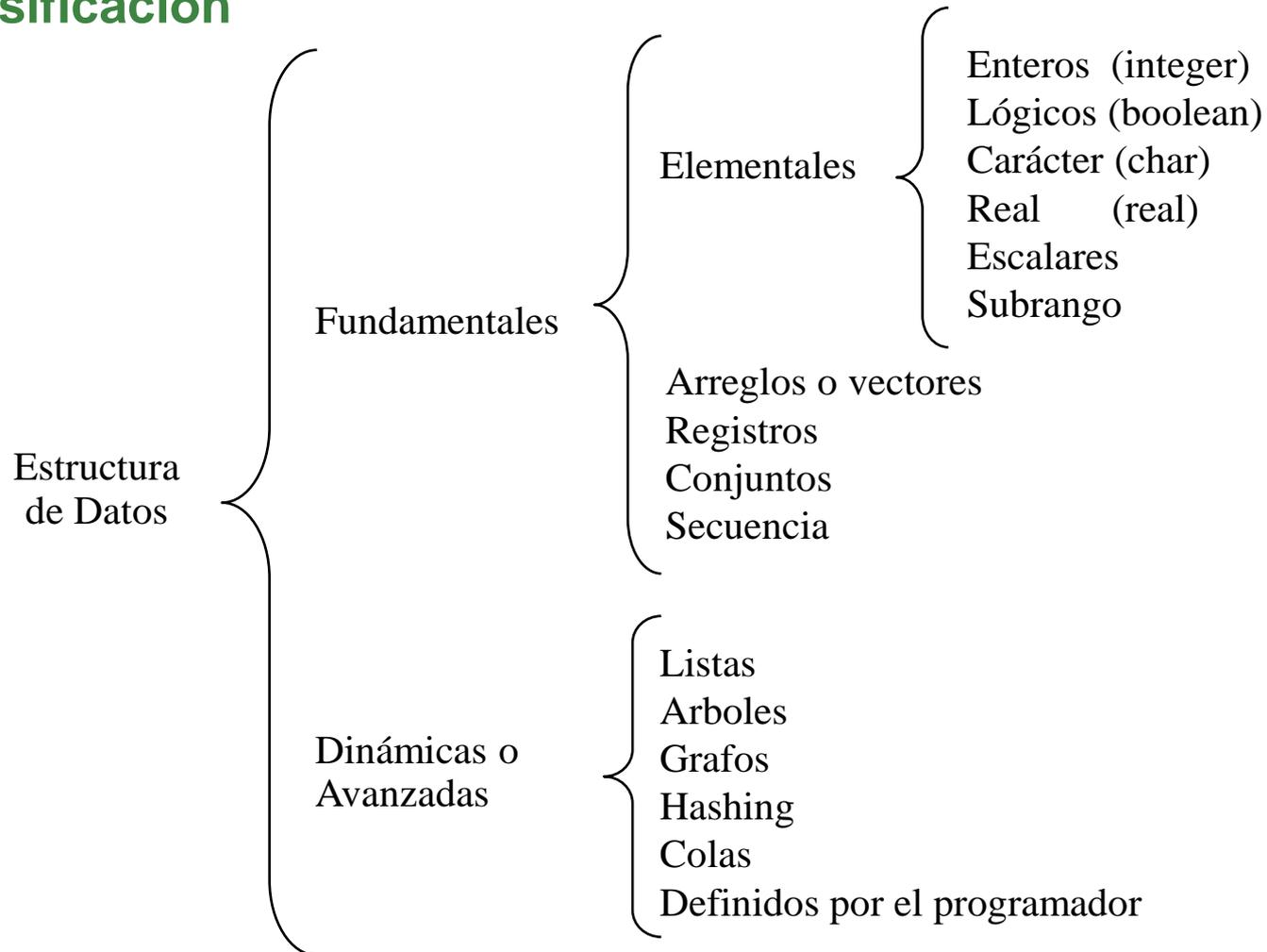
Definición Previa: *representación física de una variable.*

Definición: *la representación física de un modelo matemático mediante el uso de variables que pueden ser de diferentes tipos.*



Estructuras de Datos

Clasificación



Estructuras de Datos

Estructura de Datos Fundamentales

Son aquellas de tamaño fijo y no se modifican durante la ejecución del programa.

Las estructuras fundamentales son la base para construir estructuras más complejas en un lenguaje de programación, conocidas como estructuras avanzadas.

Estructura de Datos Fundamentales - Elementales

Estudiadas basado en la representación de los tipos de datos

Nos enfocaremos en el resto de las estructuras de datos



Estructuras de Datos

Estructuras de Datos Fundamentales - Arreglo

Es una estructura fundamental formada por un número fijo de componentes (definido cuando se edita el programa) todas del mismo tipo.

Permite crear un conjunto de variables que van a estar relacionadas bajo un mismo nombre y que son todas del mismo tipo, por eso se dice que es una estructura homogénea.

Ejemplo usando PASCAL

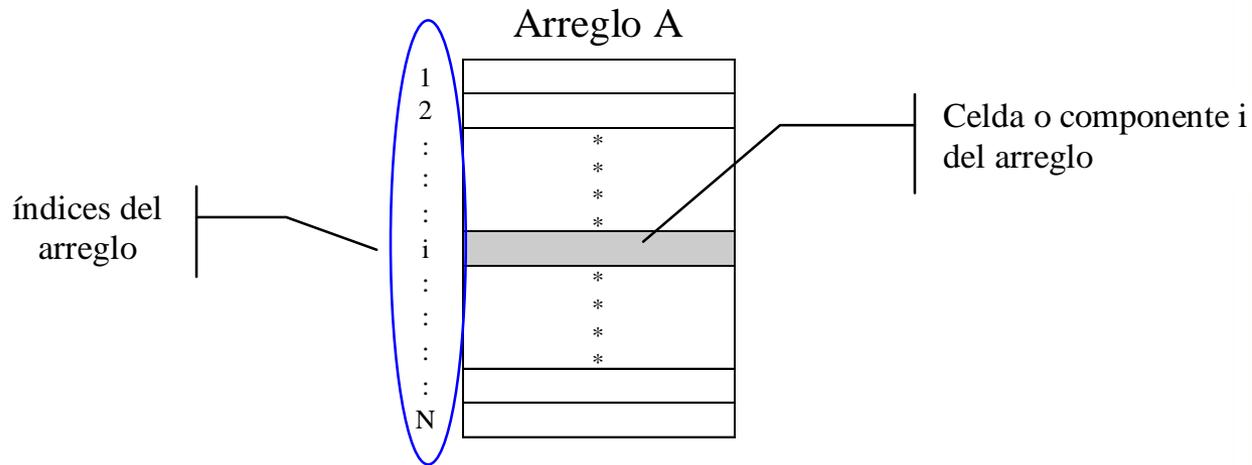
VAR a: array[lim_inf ... lim_sup] of T (T es el tipo base del arreglo)

lim_inf y lim_sup definen el número de componentes del arreglo, estos deben ser constantes y pertenecer a un tipo ordinal.

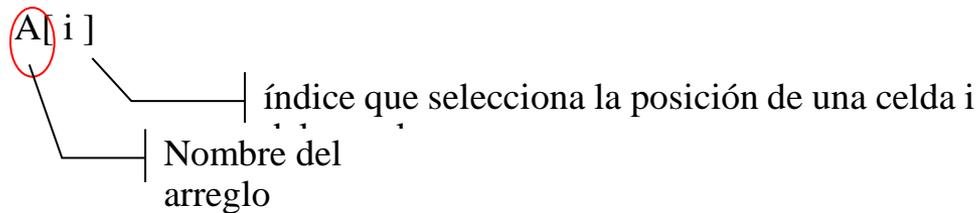
Estructuras de Datos

Estructuras de Datos Fundamentales - Arreglo

Acceso al arreglo es aleatorio,



Acceso



i debe ser de tipo ordinal

Estructuras de Datos

Estructuras de Datos Fundamentales - Arreglo

Ejemplo

VAR a: array[1 .. 6] of integer;

a[1] := 10; a[2] := 15; a[3] := 20
a[4] := 300; a[5] := 4; a[6] := 33

	Arreglo a
1	10
2	15
3	20
4	300
5	4
6	33

Estructuras de Datos

Estructuras de Datos Fundamentales - Arreglo

Arreglo Multidimensional

El Tipo Base T puede ser un arreglo



Formamos arreglo de Arreglos

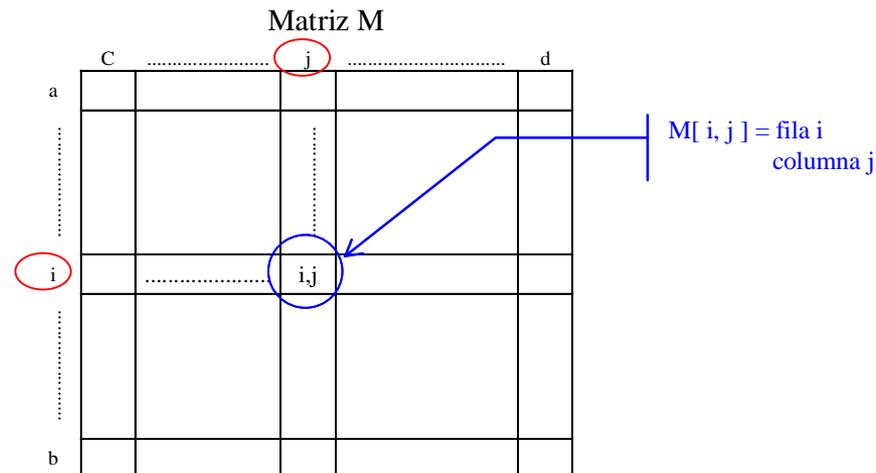
Ejemplo: $\text{Var } M: \text{array}[a .. b] \text{ of array } [c .. d] \text{ of } T1;$

Acceso: $M[i][j]$ j -ésimo componente del i -ésimo componente de M

Se puede abreviar definiendo una matriz

$\text{Var } M: \text{array}[a .. b, c .. d] \text{ of } T1;$

Acceso: $M[i, j]$: i -ésimo componente en la j -ésima columna de M



Estructuras de Datos

Estructuras de Datos Fundamentales - Registro

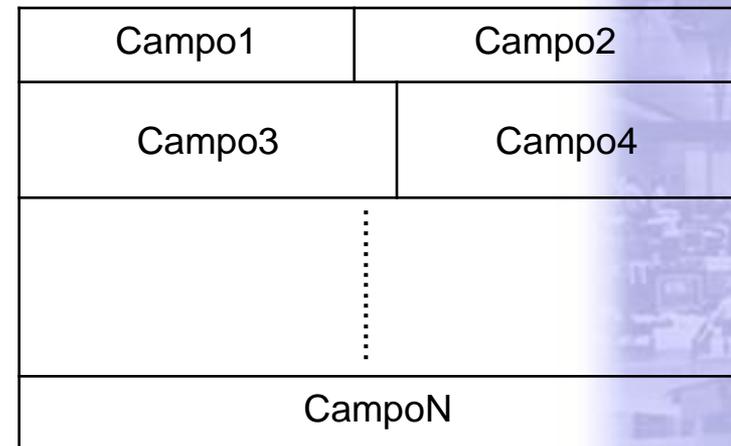
Es la estructura de datos más flexible ya que permite yuxtaponer elementos de cualquier tipo, incluso estructurados

Es una estructura fundamental formada por componentes que pueden ser de diferentes tipos. Consiste en un número fijo de componentes llamados campos, los componentes no están limitados a ser del mismo tipo y no se pueden acceder directamente.

Ejemplo usando PASCAL

```
VAR R : RECORD  
    Campo1 : T1;  
    Campo2 : T2;  
    Campo3 : T3;  
    Campo4 : T4;  
    :  
    :  
    CampoN : Tn;  
END;
```

Representación



Estructuras de Datos

Estructuras de Datos Fundamentales - Registro

Ejemplo

```
TYPE Fecha = RECORD
    Día : 1..31;
    Mes: 1..12;
    Año : 1900 .. 2100
END;

Persona = RECORD
    Nombre: string[80];
    Fecha_nac: Fecha;
    Sexo : (M, F);
    Edo_civil : (S,C,V,D)
END;

VAR P: Persona;
```

Para hacer *Fecha = 24/02/95*

```
P.Fecha_nac.día := 24;
P.Fecha_nac.mes := 02;
P.Fecha_nac.año := 1995;
```

Estructuras de Datos

Estructuras de Datos Fundamentales - Conjunto

PASCAL permite definir estructuras de datos tipo conjunto, como una estructura fundamental.

```
TYPE   Conjunto = Set of T;  
VAR    C: Conjunto;
```

Los posibles valores de la variable C es el conjunto de elementos del tipo T.

*El acceso a una estructura tipo conjunto se hace en PASCAL con el operador **in***

Ejemplo:

```
TYPE  Primario = (amarillo, azul, rojo);  
      Color = Set of Primario  
  
VAR   H1, H2: Color;  
      H1 := [rojo];  
      H2 := [ ];          /* vacío */  
      H2 := H2 + [ succ(amarillo) ];
```

Estructuras de Datos

Estructuras de Datos Fundamentales - Conjunto

Operaciones:

- +** : *Unión*
- *** : *Intersección*
- : *Diferencia de conjuntos*

Operadores relacionales:

- =, >, <** : *de igualdad y desigualdad*
- <=, >=** : *de inclusión de conjuntos*
- in** : *de pertenencia*

Ejemplos:

1) *If A in B* $A \in B$

2) *If op in ['a', 'b', 'c', 'd']*

en vez de op='a' or op='b' or op='c' or op='d'

3) *VAR op: Set of 0..7;*
Suma : boolean;

op := [3];
Suma := [2, 3] <= op (op está incluido en [2, 3] *)*

Estructuras de Datos

Estructura de Datos Dinámicas

Conocidas también como estructuras avanzadas, son aquellas que su tamaño se modifica durante la ejecución del programa.

Los componentes de las estructuras dinámicas se definen bajo un tipo base T , pero el número de componentes de la estructura puede variar durante la ejecución del programa.

Entre las más conocidas tenemos: listas, colas, grafos, árboles y hashing.

