



Estructuras fundamentales

- Registros*
- Arreglos y matrices*
- Conjuntos*
- Tipos abstractos de datos*



Programación III.

Objetivos

1. *Conocer y aplicar el concepto de estructuras, arreglos y conjuntos.*
2. *Construir tipos abstractos de datos.*



Estructuras y registros

- Un registro es una colección de datos bajo una misma estructura, que puede ser de diferente tipo. Cada elemento recibe el nombre de campo o miembro.
- Declaración

```
typedef struct Estructura {  
    int Valor;  
    struct OtraEstructura *otra;  
} TNode;
```

Estructuras y registros

□ Acceso a los miembros de una estructura

Ligadura estática

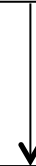
Estructura ejemplo;



ejemplo.valor;

Ligadura dinámica

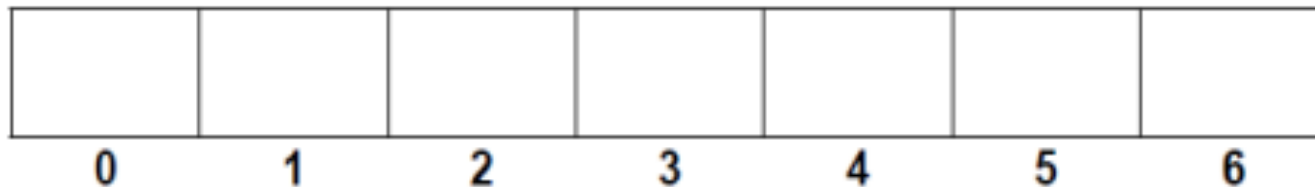
Estructura *ejemplo;



Ejemplo->valor;

Arreglos

- Es una estructura de datos, o mas técnicamente un espacio de memoria que permite almacenar una colección de elementos homogéneos (del mismo tipo)



- Dependiendo del tipo de datos con que se reserve el arreglo, así será la cantidad de bytes que se asignen a cada casilla, y en consecuencia los desplazamientos de los operadores aritmeticos +,-

Arreglos

- Sea la declaración:
`int Vector[4];`
- El compilador reservará 16 bytes de memoria, para controlar los desplazamientos sobre el vector, y cada valor estará formado por 4 direcciones de memoria.

Indice	Valor	Dirección
0		0x28ff30
		0x28ff31
		0x28ff32
		0x28ff33
1		0x28ff34
		0x28ff35
		0x28ff36
		0x28ff37
2		0x28ff38
		0x28ff39
		0x28ff40
		0x28ff41
3		0x28ff42
		0x28ff43
		0x28ff44
		0x28ff45

Arreglos

- ❑ Dicho de otra forma un arreglo en realidad es una secuencia de direcciones agrupadas secuencialmente.
- ❑ Razón por la cual, se puede asignar de manera directa a una variable de tipo puntero.
- ❑ Es una estructura de datos estática ya que las dimensiones deben ser definidas en tiempo de compilación.
- ❑ Las matrices son arreglos bidimensionales que se tratan exactamente igual.

Conjuntos

- ❑ Los conjuntos son una de las estructuras básicas de la matemática, y por tanto de la informática.
- ❑ Se revisarán algunas de las operaciones de la teoría de conjuntos que tienen implicaciones en el análisis de las estructuras de datos, como : unión, intersección y pertenencia.
- ❑ Las principales implementaciones son: arrays de bits, arrays y listas enlazadas

Conjuntos

Mediante arrays de bits

0	1	2	3	4
0	1	1	0	0

- ❑ En realidad no se declarará un vector como tal, sino un tipo de datos definido por el lenguaje entre 8 y 64 bit de almacenamiento.
- ❑ Podría declararse una variable entera (int), con un almacenamiento de 32 bits.

Conjuntos

Mediante arrays de bits

- Sea el caso

`int arreglo = 65;`

- Al convertir ese valor decimal a binario se obtiene :
1000001, y ese es el valor que entiende como
arreglo de bits, a continuación la representación

11111111 -> $U = \{0, 1, 2, 3, 4, 5, 6, 7\}$

11110001 -> $U = \{0, 4, 5, 6, 7\}$

01010101 -> $U = \{0, 2, 4, 6\}$

00000000 -> $U = \text{vacío}$

Conjuntos

Mediante arrays de bits

- ❑ En el ejemplo anterior todos los bits puestos a uno representarán al conjunto $U = \{0,1,2,3,4,5,6,7\}$.
- ❑ Si todos los bits están puestos a cero, representará al conjunto vacío.
- ❑ El bit mas significativo señalará al elemento de mayor valor, el bit menos significativo al de menor valor (bit mas significativo a la izquierda)

Conjuntos

Unión

- Se realiza mediante la operación OR inclusivo.
Ejemplo (con 8 bits en lugar de 32)

```
11001100 -> A = {2, 3, 6, 7}
Or 10010100 -> B = {2, 4, 7}
-----
11011100 -> C = {2, 3, 4, 6, 7}
```

- Operador utilizado |

Conjuntos

Intersección

□ Se realiza mediante la operación AND. Ejemplo

```
      11001100 -> A = {2,3,6,7}
And 10010100 -> B = {2,4,7}
-----
      10000100 -> C = {2,7}
```

Operador utilizado &

Conjuntos

Diferencia

- Para obtener $C = A - B$, se invierten todos los bits de B y se hace un AND entre A y B negado

10011101 -> A = {0, 2, 3, 4, 7}

10110010 -> B = {1, 4, 5, 7}

B negado: 01001101 -> B(negado) = {0, 2, 3, 6}

```
      10011101
And 01001101
-----
      00001101 -> C = {0, 2, 3}
```

Operador utilizado (AND)& (NEGACION)~

Conjuntos

Diferencia simétrica

- Para obtener $C = (A-B) \text{ unión } (B-A)$, se realiza mediante la operación de OR exclusivo (XOR) o aplicando las primitivas anteriores

```
11110000 -> A = {4,5,6,7}
Xor 00011110 -> B = {1,2,3,4}
-----
11101110 -> C = {1,2,3,5,6,7}
```

Operador utilizado \wedge

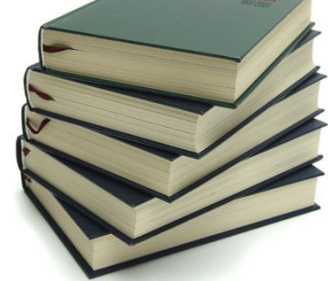
Tipos abstractos de datos

- ❑ Definición: es un conjunto de valores y de operaciones definidos mediante una especificación independiente de cualquier representación.
- ❑ TAD = Valores + Operaciones
- ❑ La manipulación de un TAD solo depende de su especificación, **nunca** de su implementación.
- ❑ Dada una especificación de TAD hay muchas implementaciones válidas
- ❑ Ejemplos : clases



```
int main()  
{  
    return 0;  
}
```





Bibliografía

- ❑ Análisis de algoritmos: complejidad

<http://www.lab.dit.upm.es/~lprg/material/apuntes/o/index.html#s2>

- ❑ Arreglos y matrices

<http://dis.unal.edu.co/~programacion/book/modulo3.pdf>

- ❑ Programación en c++. Algoritmos estructuras de datos y objetos. Luis Joyanes Aguilar