

# ESTRUCTURAS DE DATOS TIPO **ARREGLOS**

# Arreglos

---

Las estructuras de datos pueden clasificarse en lineales y no lineales.

Se dice que una estructura es lineal si sus elementos forman una secuencia, que generalmente se almacenan en posiciones consecutivas de memoria.

Estas estructuras lineales reciben el nombre de arreglos (*arrays*) y pueden ser:

- Unidimensionales (también llamados vectores)
- Bidimensionales (matrices o tablas)
- Multidimensionales (tres o más dimensiones)

# Arreglos de una dimensión

---

Los arreglos unidimensionales se definen como un conjunto *finito* y *ordenado* de elementos *homogéneos*.

- ***Finito***: almacena un número determinado de elementos ( $n$ ).
- ***Ordenado***: los elementos tienen un orden; existe un primer elemento, seguido de un segundo elemento; el quinto está precedido por el cuarto y seguido del sexto elemento.
- ***Homogéneos***: todos los elementos deben ser del mismo tipo: enteros, reales, ...

Los elementos de un arreglo se referencian a través de un conjunto de *índices* constituido por números consecutivos.

# Arreglos de una dimensión

---

Los elementos de un arreglo unidimensional podemos denotarlos, o bien por medio de notación subindicada

$$A_1, A_2, A_3, \dots, A_n,$$

por medio de notaciones parentizadas que utilizan algunos lenguajes

$$A(1), A(2), A(3), \dots, A(n)$$

o bien, utilizando corchetes, usuales también en algunos lenguajes

$$A[1], A[2], A[3], \dots, A[n]$$

que será la que utilizaremos en este curso.

# Arreglos de una dimensión

---

Independientemente de la notación utilizada, el número  $K$  en  $A[K]$ , recibe el nombre de índice y  $A[K]$  el de variable subindicada. Nos referiremos entonces, al elemento  $K$ -ésimo elemento del arreglo, cuya posición real depende del valor de  $K$ .

	A[1]	A [2]	A [3]	...			A [n]	
Arreglo A	19	26	23	14	29	18	17	13
	1	2	3	4	5	6	7	8

Obsérvese que la utilización de índices permite referenciar cada elemento del arreglo por su posición relativa en  $A$ .

# Arreglos de una dimensión

---

## **EJEMPLOS:**

Sea **DATOS** un arreglo de seis elementos, donde cada elemento es un número entero tal que

**DATOS[1] = 247**

**DATOS[2] = 56**

**DATOS[3] = 429**

**DATOS[4] = 135**

**DATOS[5] = 87**

**DATOS[6] = 156**

Algunas veces denotaremos un arreglo escribiendo simplemente

**DATOS: 247, 56, 429, 135, 87, 156**

# Arreglos de una dimensión

---

El número  $n$  de elementos recibe el nombre de *longitud* o tamaño del arreglo. Si no se indica lo contrario, asumiremos que el conjunto de índices consiste en el conjunto de enteros  $1, 2, \dots, n$ .

En general, la *longitud* o número de elementos del arreglo, también llamada número total de componentes (NTC), puede obtenerse del conjunto de valores de índice, a través de la fórmula:

$$\text{Longitud} = LS - LI + 1, \quad \text{o bien,} \quad \text{NTC} = LS - LI + 1$$

donde,  $LS$  es el índice de mayor valor, llamado *límite superior* del arreglo y,  $LI$  es el índice de menor valor, llamado *límite inferior* del arreglo.

Nótese que  $\text{longitud} = LS$  cuando  $LI=1$ .

# Arreglos de una dimensión

---

## EJEMPLOS:

Gráficamente, el arreglo **DATOS** suele dibujarse como se muestra en las figuras, vertical u horizontalmente:

Límite Inferior (LI)	→	1	247
<b>DATOS</b>		2	56
		3	429
		4	135
		5	87
Límite Superior (LS)	→	6	156

<b>DATOS</b>	247	56	429	135	87	156
	1	2	3	4	5	6

$$L = LS \rightarrow L = 6, \text{ o bien}$$

$$\begin{aligned} NTC &= LS - LI + 1 \\ &= 6 - 1 + 1 = 6 \end{aligned}$$



# Arreglos de una dimensión

---

## Declaración de arreglos unidimensionales

Algunos lenguajes de programación reservan el espacio destinado a los arreglos, de manera *estática*; es decir, durante la fase de compilación; por tanto la longitud del arreglo no puede variarse durante la ejecución.

Por el contrario, otros lenguajes permiten leer un entero  $n$  y después declarar el arreglo con  $n$  elementos.

Se dice que tales lenguajes reservan memoria *dinámicamente*.

# Arreglos de una dimensión

---

## **Declaración de arreglos unidimensionales**

Cada lenguaje de programación tiene sus reglas particulares para declarar un arreglo. Cada declaración debe incluir de forma explícita o implícita, tres cosas

1. El nombre del arreglo
2. El tipo de datos del arreglo
3. El conjunto de índices del mismo.

# Arreglos de una dimensión

---

## Declaración de arreglos unidimensionales

Utilizaremos la siguiente sentencia para declarar un arreglo unidimensional

**ident\_arreglo:** arreglo [**Límlnf..LímSup**] **de tipo**

Con los valores **Límlnf** y **LímSup** se declara el tipo de los índices, así como el número de elementos que tendrá el arreglo (NTC).

Con **tipo** se declara el tipo de datos para todos los componentes del arreglo unidimensional.

En general un índice puede ser cualquier valor ordinal: carácter, entero (+/-), enumerado, etc., pero la mayoría de los lenguajes usados actualmente, solo permiten valores enteros.

# Arreglos de una dimensión

---

## Representación de arreglos unidimensionales en memoria

Sea,

$$Loc(A[K]) = \text{dirección en memoria del elemento } K\text{-ésimo del arreglo } A$$

Como se ha comentado anteriormente, los elementos de un arreglo se almacenan en posiciones de memoria sucesivas. Debido a esto el computador no necesita contabilizar las posiciones de todos los elementos de  $A$ , sino que solo necesita conocer la dirección del primer elemento de  $A$ . Esta dirección la indicaremos como

$$dirBase(A)$$

Y recibe el nombre de *dirección base del arreglo*  $A$ .

# Arreglos de una dimensión

---

## Representación de arreglos unidimensionales en memoria

Utilizando la dirección base, el computador calcula la dirección de los demás elementos de la forma siguiente

$$Loc(A[K]) = dirBase(A) + w (K-LI)$$

Donde  $w$  representa el número de bytes que utiliza cada elemento del arreglo en memoria y que varía según el tipo de dato especificado.

Nombre	Tipo	Tamaño
Boolean	Lógico	1 bit
Char	Carácter	16 bits
Byte	Numero entero	8 bits
Short	Numero entero	16 bits
Int	Numero entero	32 bit
Long	Numero entero	64 bits
Float	Numero real	32 bits
Double	Numero real	64 bits

# Arreglos de una dimensión

## EJEMPLO:

Sea el arreglo **DATOS** del ejemplo anterior. La  $Loc(DATOS[4])$ , asumiendo que la  $dirBase(DATOS) = 2300$  y que  $w = 2$  (enteros cortos) será:

Límite Inferior (LI) →

	1	247
DATOS	2	56
	3	429
	4	135
	5	87
Límite Superior (LS) →	6	156

Considerando que  $Loc(A[K]) = dirBase(A) + w (K-LI)$   
Entonces,

$$\begin{aligned} Loc(DATOS[4]) &= 2300 + 2 (4-1) \\ &= 2300 + 6 \\ &= 2306 \end{aligned}$$

1	2	3	4	5	6
247	56	429	135	87	156
2300	2302	2304	2306	2308	2310

# Arreglos de una dimensión

---

## **EJEMPLO:**

Una compañía de automóviles utiliza un arreglo AUTO para almacenar el número de coches vendidos anualmente desde el año 1984 hasta el año 2019.

En este caso se consideró conveniente, utilizar el año como índice del arreglo con el fin de facilitar las búsquedas, quedando declarado de la siguiente manera:

**AUTO : arreglo [1984..2019] de enteros**

donde,

***NTC(AUTO) = 2019 – 1984 + 1 = 36 elementos***

# Arreglos de una dimensión

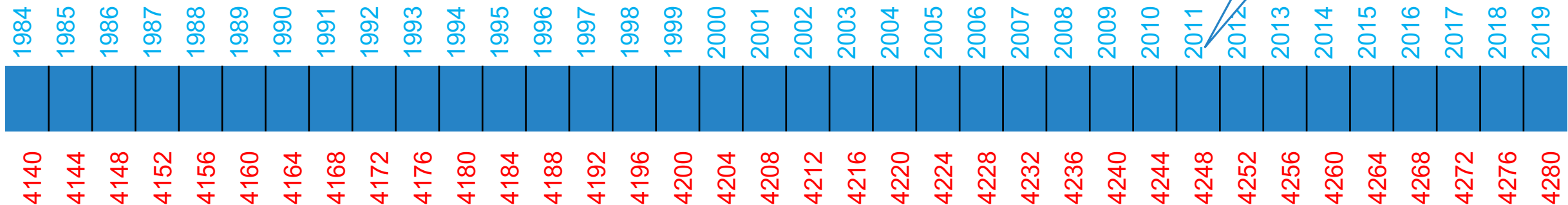
## EJEMPLO (CONT.):

Obtenga, la  $LOC(AUTO[2011])$ , asumiendo que la  $dirBase(AUTO) = 4140$  y que  $w = 4$  (enteros).

Considerando que  $Loc(A[K]) = dirBase(A) + w (K-LI)$   
Entonces,

$$\begin{aligned} Loc(AUTO[2011]) &= 4140 + 4 (2011-1984) \\ &= 4140 + 4 (27) \\ &= 4140 + 108 = \boxed{4248} \end{aligned}$$

**AUTO**





# Arreglos de una dimensión

---

## **PRÁCTICA:**

Asumiendo la tabla **AUTO** anterior y que la *dirBase*(**AUTO**) = 4140 y que  $w = 4$  (enteros), como se dijo, obtenga:

- (a) *LOC*(**AUTO**[1997]),
- (b) *LOC*(**AUTO**[1984]),
- (c) *LOC*(**AUTO**[2015]),
- (d) *LOC*(**AUTO**[1982]),
- (e) *LOC*(**AUTO**[2019]).

# Operaciones con Arreglos de una dimensión

---

Las operaciones básicas que se suelen realizar sobre un arreglo, son las siguientes:

## **Lectura:**

El proceso de lectura consiste en leer y asignar un valor a cada elemento del arreglo. Generalmente se usará un ciclo para leer todos los elementos del arreglo y la variable de control del ciclo será usada como índice para el almacenamiento. A esta operación se le suele llamar *carga del arreglo*.

Para I=1, 20, 1 repetir

Leer A[I]

Fin-para-I

# Operaciones con Arreglos de una dimensión

---

## Escritura:

Este proceso es similar al de lectura. Consiste en desplegar o imprimir el valor de cada elemento del arreglo. De igual manera se usará un ciclo y la variable de control del ciclo será usada como índice.

Se asume, que cada elemento se imprime en una nueva línea de salida.

A esta operación se le suele llamar *vaciado del arreglo*.

Para I=1, 20, 1 repetir

    Escribir A[I]

Fin-para-I

# Operaciones con Arreglos de una dimensión

## Asignación:

En general, los lenguajes no permiten la asignación directa de un determinado valor a todo el arreglo, sino que se debe asignar el valor deseado a cada componente.

Como en las operaciones anteriores, se usa un ciclo para realizar esta acción.

También podemos utilizar la asignación directa o la implícita, cuando los valores de cada elemento sean diferentes.

Para I=1, 20, 1 repetir

$A[I] = 0$

Fin-para-I

DATOS[1] = 247

DATOS[2] = 56

DATOS[3] = 429

DATOS[4] = 135

DATOS[5] = 87

DATOS[6] = 156

Asignación directa

Asignación implícita

DATOS: [247, 56, 429, 135, 87, 156]

# Operaciones con Arreglos de una dimensión

---

Otras operaciones que se suelen realizar habitualmente sobre una estructura lineal, sea un arreglo o una lista enlazada, son las siguientes:

1. **Recorrido:** Procesamiento de cada elemento del arreglo
2. **Búsqueda:** Recorrer el arreglo en búsqueda de un elemento en particular
3. **Actualización:** Se aplica en arreglos ordenados y desordenados
  - a. **Inserción:** Adición de un nuevo elemento al arreglo
  - b. **Borrado:** Eliminación de un elemento del arreglo
  - c. **Modificación:** Modificación de un elemento del arreglo
4. **Ordenación:** Organizar los elementos del arreglo en orden ascendente o descendente
5. **Mezcla:** Combinar dos arreglos en uno solo