Unidad 6

ESTRUCTURAS ESTÁTICAS

ARRAYS

Prof. Nicolás Battaglia

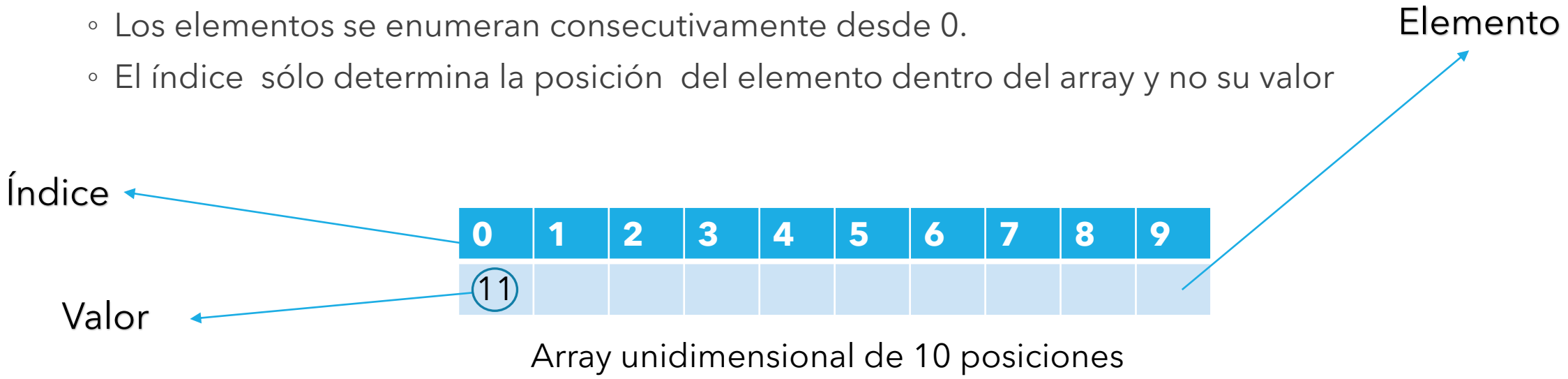
Arrays

DECLARAR, INICIALIZAR Y RECORRER UN ARRAY

Estructuras estáticas - Arrays

¿Qué son los arrays?

- Estructuras de memoria que poseen un tamaño conocido de ocurrencias.
- Variables organizadas en una secuencia de datos del mismo tipo.
- Se organizan en elementos y cada posición es accedida por un índice.
- Los elementos se enumeran consecutivamente desde 0.
- El índice sólo determina la posición del elemento dentro del array y no su valor



Estructuras estáticas - Arrays

¿Cómo se utilizan?

- Se declaran de la misma manera que cualquier variable en C, indicándole su tamaño.
- El índice puede ser referenciado de diferentes maneras para poder cargar o recuperar el valor almacenado en determinada posición.
- Deben inicializarse con un valor por default, generalmente en 0.

```c
int miArray[50];  //Declaro un array de 50 posiciones
int valor = miArray[15];  //acceso a la posicion 15

int x=10;
int valor2=miArray[x]; //acceso a la poisicón x.

printf("El valor de la posición 8 es %d",miArray[7]); //muestro el valor de la posición 8.
```

Estructuras estáticas - Arrays

Funcionamiento

- Al intentar **acceder a una posición** del array, se debe siempre asegurar que el **índice** utilizado este dentro del **rango definido**.
- C **no comprueba** que los **índices** del array estén dentro del **rango definido**, esto podrá dar un **error en tiempo de ejecución**.
- Se pueden **recorrer utilizando un ciclo FOR** aprovechando su índice, como índice del array.

```
int miArray[10];

int valor = miArray[15];
```

→ Declaro un array de 10 posiciones pero intento acceder a la posición 15 →

ERROR!

La posición no existe

Estructuras estáticas - Arrays

Funcionamiento

- El operador **sizeof** permite conocer el **tamaño** de un array.
- Se pueden **declarar y completar de manera simultánea**, de tal manera que se determine automáticamente su tamaño

```
int miArray[50];
printf("Tamaño: %d\n", (int)( sizeof(miArray) / sizeof(miArray[0]) ));
printf("Bytes: %d\n", (int)( sizeof(miArray)));
```

Calculo la cantidad de posiciones del array

Calculo el tamaño

```
int miArrayInt[]={15,2,19,41,13,18};
char miArrayChar[]={'h','o','l','a'};
```

Declaro un array de enteros de 6 posiciones

Declaro un array de char de 4 posiciones

Estructuras estáticas - Arrays

Arrays multidimensionales

- Cada **posición del array está compuesta por otro array**, tienen más de un índice.
- El array de **2 dimensiones** se representa con una **matriz**.

	0	**1**	**2**	**3**	**4**	**5**
0	0,0	0,1	0,2	0,3	0,4	0,5
1	1,0	1,1	1,2	1,3	1,4	1,5
2	2,0	2,1	2,2	2,3	2,4	2,5
3	3,0	3,1	3,2	3,3	3,4	3,5
4	4,0	4,1	4,2	4,3	4,4	4,5
5	5,0	5,1	5,2	5,3	5,4	5,5

Estructuras estáticas - Arrays

Arrays multidimensionales

- Cada **posición del array está compuesta por otro array**, tienen más de un índice.
- El array de **2 dimensiones** se representa con una **matriz**.

```
int matriz[5][5];
int valor=matriz[4][2];
```

→

	0	1	2	3	4
0	0,0	0,1	0,2	0,3	0,4
1	1,0	1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4
3	3,0	3,1	3,2	3,3	3,4
4	4,0	4,1	4,2	4,3	4,4
5	5,0	5,1	5,2	5,3	5,4

→

	0	1	2	3	4
0	100	121	33	31	31
1	34	21	0	44	91
2	21	45	53	11	10
3	41	76	65	45	95
4	43	55	**81**	39	19
5	82	4	3	19	67

→ 81