

ESTRUCTURAS DE DATOS

Arreglos **En Lenguaje C++**

Definición

Un arreglo, o también llamado array, es una estructura unidimensional de almacenamiento de datos que permite que en una sola variable se pueda almacenar mas de un dato.

Justificación

Por ejemplo, pensemos en la situación donde es necesario almacenar la edad de 3 estudiantes. Una posible solución sería declarar tres variables de tipo entero, y en cada una asignar la edad de los estudiantes, así:

```
int edad_1 = 17;
```

```
int edad_2 = 20;
```

```
int edad_3 = 19;
```

La anterior solución es válida, siempre y cuando la cantidad de estudiantes no sea mayor.

Por ejemplo, que sucedería si necesitáramos almacenar la edad de 60 estudiantes. Declarar 60 variables sería muy poco práctico.

En estos casos lo mejor será utilizar un arreglo.

Justificación

La solución al problema anterior sería, declarar un arreglo de tipo entero de tamaño 60, así:

```
int edades[60];
```

Esto permite que en una sola variable se pueda almacenar esa gran cantidad de datos.

Declaración de un Arreglo (1)

- Sintaxis en C #1:

Esta sintaxis es utilizada cuando sabemos cuantos elementos tendrá el arreglo

tipo_dato nom_arreglo[tamaño];

Declaración del arreglo

Asignación de memoria

Una sola instrucción

Por ejemplo:

int edades[30];

double notas[30];

char genero[30];

bool estado[30];

Declaración de un Arreglo (2)

Esta sintaxis es utilizada cuando **no** sabemos cuantos elementos tendrá el arreglo

- Sintaxis en C # 2:

```
tipo_dato *nom_arreglo;
```

Declaración del arreglo

```
nom_arreglo = new tipo_dato[tamaño];
```

Asignación de memoria

Dos instrucciones:

**1ra: declarar el arreglo
2da: asignar memoria**

Esta sintaxis permite declarar el arreglo de un tamaño que se desconoce, lo cual da flexibilidad, permitiendo por ejemplo que el usuario ingrese el tamaño del arreglo deseado.

Declaración de un Arreglo (3)

Por ejemplo:

```
int *edades;  
double *notas;  
char *genero;  
bool *estado;
```

Declaración de los arreglos

```
int tam;  
cout<<"Ingrese la cantidad de estudiantes: ";  
cin>>tam;
```

```
edades = new int[tam];  
notas = new double[tam];  
genero = new char[tam];  
estado = new bool[tam];
```

Asignación de memoria

Representación grafica (1)

Un arreglo se puede representar gráficamente como una serie de *cajones* en los cuales se almacena un dato. Estos *cajones* están seguidos uno de otro, y se identifican mediante un index.

El index o posición es un valor de tipo entero, que comienza desde 0 hasta el tamaño del arreglo -1

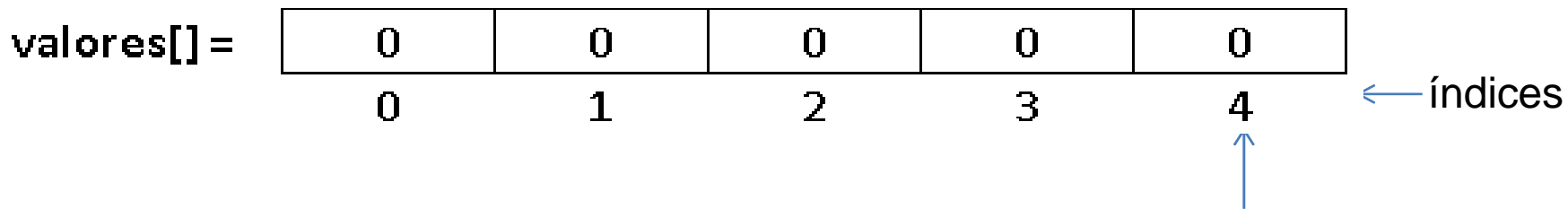
Representación grafica (2)

Por ejemplo:

```
int valores[5];
```

gráficamente el arreglo se vería así:

Al declarar un arreglo todas sus posiciones quedan asignadas con cero



Ultimo elemento = tamaño - 1
= 5 - 1
= 4

Asignación de un arreglo

Para almacenar datos en un arreglo, solo basta con *llamar* al arreglo e indicar en que posición del mismo se desea guardar datos.

Por ejemplo;

valores[0] = 18;

valores[1] = 20;

valores[4] = 17;

gráficamente el arreglo se vería así:

valores[] =	18	20	0	0	17
	0	1	2	3	4

Recorrido de un arreglo (1)

Para recorrer un arreglo, ya bien sea para leerlo, asignarlo o imprimirlo, es necesario utilizar un ciclo. Por lo general se utiliza el bucle `for`.

El ciclo permite controlar el índice que se recorre en el arreglo.

Recorrido de un arreglo (2)

Por ejemplo: asignar un arreglo de tamaño 10 con los primeros números pares desde el 2.

```
int pares[10]; //usando la sintaxis # 1
int n = 2; //variable para calcular los
pares
```

```
for(int i = 0; i < 10; i++){
    pares[i] = n;
    n+=2;
}
```

Recorrido de un arreglo (3)

Por ejemplo: imprimir por pantalla el arreglo anterior.

```
for(int i = 0; i < 10; i++){  
    cout<<pares[i]<<"\n";  
}
```

Salida

**2
4
6
8
10
12
14
16
18
20**