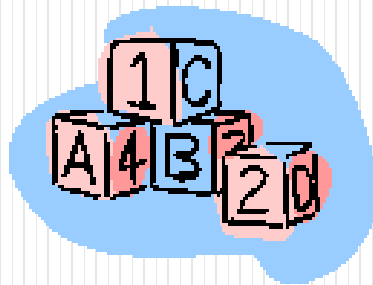
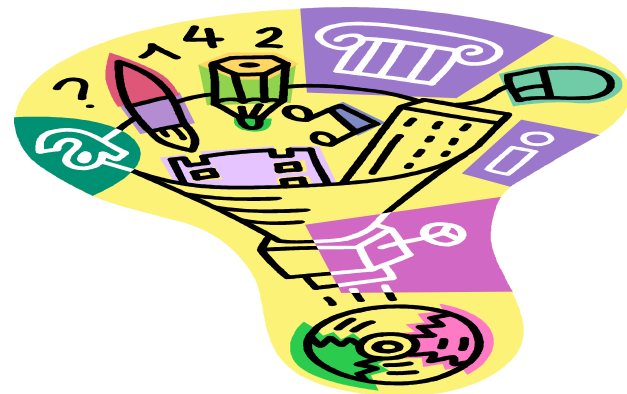


Tipos de Arreglos



Tipos de Arreglos

- **Arreglo:** Es una secuencia de datos del mismo tipo.
- El tipo de elementos almacenados en el arreglo puede ser de cualquier tipo de dato de C#:
 - int,
 - float,
 - char
 - estructuras, etc.



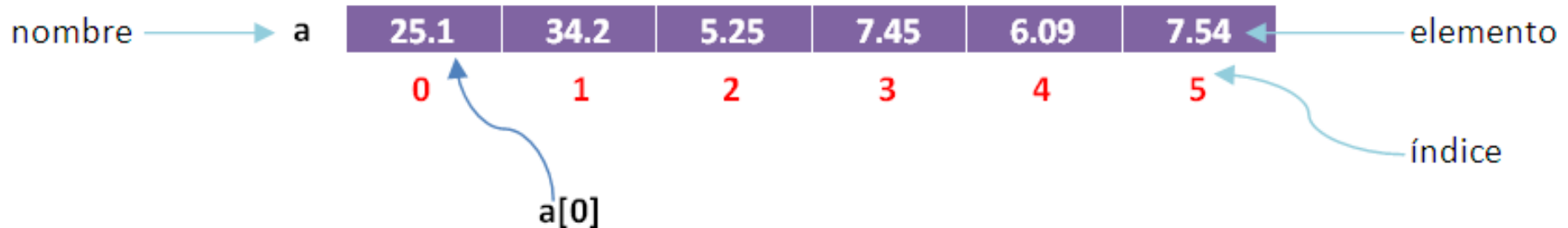
Tipos de Arreglos

- Hay dos tipos de arreglos:
 - Unidimensionales
 - Multidimensionales
- **Ejemplos de arreglos:**
 - Arreglo que contenga las calificaciones de los alumnos de una clase.
 - Arreglo que guarde las temperaturas leídas, etc.

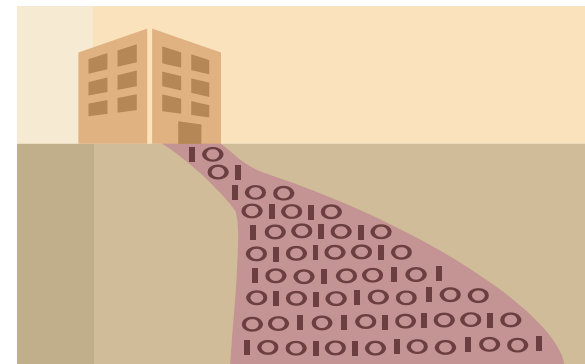


Tipos de Arreglos

- **Arreglo:**



- a[0] contiene 25.1
- a[1] contiene 34.2
- a[2] contiene 5.25
- a[3] contiene 7.45
- a[4] contiene 6.09
- a[5] contiene 7.54



Tipos de Arreglos

- **Limites del arreglo:**

- Límite inferior

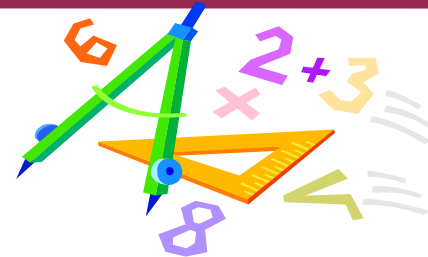
0

- Límite superior

$n - 1$, donde n es el número de elementos del arreglo.

- **Declaración de un arreglo:**

tipo [] nombre_arreglo = new int[numero_de elementos];



Tipos de Arreglos

- **Ejemplo:**

- `int [] números= new int[10];`
- `int [] edad=new int[5];`
- `float [] salarios= new float[25];`



- Para acceder a cada elemento del arreglo se utiliza su índice.

- **Ejemplo:**

```
Console.WriteLine("{0}", numeros[4]);
```

Tipos de Arreglos

- Para leer un valor de entrada y acceder al elemento 3 del arreglo números:

```
numeros[2]=int.Parse(Console.ReadLine());
```

- También se puede acceder a los arreglos utilizando variables o fórmulas.

- **Ejemplo:**

```
ventas[total + 5];
```

```
bonos[i];
```

```
salario[mes[i]*5];
```



Tipos de Arreglos

- **Inicialización de un arreglo:** se pueden inicializar los elementos de un arreglo de las siguientes maneras:

- **Ejemplo:**

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int [] numeros= {10,20,30,40,50,60};
```

```
int []n = {3, 4, 5};
```

```
char []c = {'L','u','i','s'};
```

```
int [] precios = new int[3];
```

```
precios[0]= 10;
```

```
precios[1]= 20;
```

```
precios[2]= 30;
```

```
Console.WriteLine("Imprimir algun elemento del vector accediendo por su  
indice");
```

```
Console.WriteLine("numeros[2]={0}", numeros[2]);
```

```
Console.WriteLine("n[0]={0}", n[0]);
```

```
Console.WriteLine("c[3]={0}", c[3]);
```

```
}
```

```
}
```

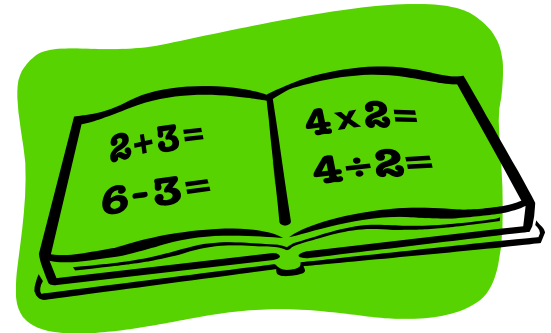


Tipos de Arreglos

- **Ejemplo:** Programa que lee datos del teclado y los asigna en un arreglo.

```
class Program
{
    static void Main(string[] args)
    {
        int[] cuenta = new int[10];
        int total = 0;

        for (int i = 0; i < 10; i++)
        {
            Console.WriteLine("Introduzca un numero");
            cuenta[i] = int.Parse(Console.ReadLine());
        }
        Console.WriteLine("Listado de numeros:");
        for (int i = 0; i < 10; i++)
        {
            Console.WriteLine("cuenta[{0}]: {1}", i, cuenta[i]);
            total += cuenta[i];
        }
        Console.WriteLine("La suma de los numeros es: {0}", total);
    }
}
```

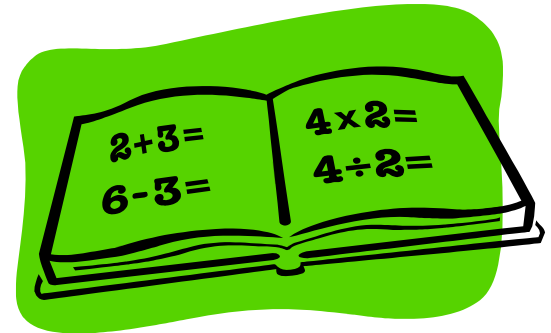


Tipos de Arreglos

- **FOR EACH:** Se utiliza para iterar los elementos de un vector o cualquier colección enumerable, procesa los elementos del 0 al último.

```
class Program
{
    static void Main(string[] args)
    {
        int[] cuenta = new int[10];
        int total = 0;

        for (int i = 0; i < 10; i++)
        {
            Console.WriteLine("Introduzca un numero");
            cuenta[i] = int.Parse(Console.ReadLine());
        }
        Console.WriteLine("Listado de numeros:");
        foreach (int i in cuenta)
        {
            Console.WriteLine("cuenta[]: {0}", i);
            total += i;
        }
        Console.WriteLine("La suma de los numeros es: {0}", total);
    }
}
```



Tipos de Arreglos

- **Ejercicio:**

Realizar un programa que utilice un vector de 11 elementos, el vector deberá ser llenado con valores aleatorios del 1 al 9 y deberá preguntar el número a buscar.

Ejemplo:

Número a buscar: 3

Se encontró el 3 en el índice 0

Se encontró el 3 en el índice 3

Se encontró el 3 en el índice 9



3
9
1
3
4
2
9
8
7
3
1

Tipos de Arreglos

- **Ejercicio:**

Crear el programa de un estacionamiento, utilizar un vector de 10 elementos para simular los cajones del estacionamiento, inicializar el vector con ceros para indicar que el cajón está libre y para indicar que está ocupado asignar un 1.

Nota: los cajones serán asignados aleatoriamente, cuando el estacionamiento esté lleno indicarlo con un mensaje.

