# Arreglos

Un arreglo (Array) es un conjunto de cosas similares.

Un arreglo es también conocido como una matriz, la cual almacena una colección de datos del mismo tipo (int, float, double, String, etc.) en un solo lugar, con ello se evita declarar una infinidad de variables, del mismo tipo.

Cada elemento del arreglo tiene un número al que está asociado, llamado "índice numérico" (numeric index), que permite acceder a él.

Los arreglos comienzan a partir del índice cero y pueden ser manipulados con varios métodos.

## Declaración y creación de un arreglo

Cuando vayamos a declarar un arreglo en C# debemos colocar los corchetes después del tipo de dato.

Para declarar un arreglo seguimos la siguiente sintaxis:

tipoDato[] nombreArreglo;

**tipoDato:** Se utiliza para especificar el tipo de dato del elemento del arreglo.

**[ ]:** Especifica el rango del arreglo, es decir, su longitud.

**nombreArreglo:** Es el nombre del arreglo que queremos declarar.

Es necesario inicializar a la matriz para crear una instancia a la matriz y poder asignar valores a la misma. Para crear la instancia de la matriz utilizamos la palabra reservada new.

double[] balance = new double[10];

La longitud se establece cuando se crea el arreglo. Después de la creación, su longitud es fija.

// Los arreglos "numeros" y "animales" ambos son declarados e inicializados con valores.

int[] numeros = { 1, 4, -3, 6, 3 };

string[] animales = { "gato", "perro", "oso", "tigre" };

## Acceder a los elementos de un arreglo

Accedes a un elemento del arreglo haciendo referencia al número de índice.

Console.WriteLine(animales[0]);

// Salida gato

## Cambiar un elemento de un arreglo

Para cambiar el valor de un elemento específico, haces referencia al número de índice y haces una asignación con el nuevo valor.

animales[0] = "lobo";

## Longitud de un arreglo

Para saber cuántos elementos tiene un arreglo, se usa la propiedad Length.

Console.WriteLine(animales.Length);

// Salida 4

## Bucle foreach

La instrucción foreach proporciona una forma sencilla y limpia de iterar a través de los elementos de una matriz.

La instrucción foreach procesa los elementos en orden de índice creciente, comenzando con el índice 0 y terminando con el índice Length - 1:

int[] numeros = { 1, 4, -3, 6, 3 };

foreach (int i in numeros)

{

Console.Write("{0} ", i);

}

// Salida 1 4 -3 6 3

## Búsqueda de arreglos

### Array.IndexOf

Devuelve el índice que el elemento ocupa dentro del arreglo, o -1 en caso de que no exista. Su sintaxis es:

int index = Array.IndexOf(arreglo, valorBuscar);

### Array.LastIndexOf

Devuelve el índice de la última aparición de un valor en una matriz Array unidimensional o en una parte de la matriz Array.

int lastIndex = Array.LastIndexOf(numeros, 3);

### Array.FindAll

Recupera todos los elementos que coinciden con las condiciones definidas por el predicado especificado. Ejemplo:

int[] numerosMenores = Array.FindAll(numeros, numero => numero < 100);

### Array.Find

Busca un elemento que coincida con las condiciones definidas por el predicado especificado y devuelve la primera aparición en toda la matriz Array.

int primerNumeroMenorEncontrado = Array.Find(numeros, numero => numero < 10);

### Array.FindLast

Busca un elemento que coincida con las condiciones definidas por el predicado especificado y devuelve la última aparición en toda la matriz Array.

int ultimoNumeroMenorEncontrado = Array.FindLast(numeros, numero => numero < 10);

### Array.FindIndex

Busca un elemento que cumpla las condiciones definidas por el predicado especificado y devuelve el índice de base cero de la primera aparición en un objeto Array o en una parte del mismo.

int primerIndexEncontrado = Array.FindIndex(numeros, numero => numero < 10);

### Array.FindLastIndex

Busca un elemento que cumpla las condiciones definidas por el predicado especificado y devuelve el índice de base cero de la última aparición en un objeto Array o en una parte del mismo.

int ultimoIndexEncontrado = Array.FindLastIndex(numeros, numero => numero < 10);

## Ordenamiento de arreglos

### Sort

Existen varios métodos de Array disponibles, por ejemplo Sort(), que ordena un arreglo alfabéticamente o en orden ascendente:

Array.Sort(animales);

foreach (string i in animales)

{

Console.WriteLine(i);

}

### Reverse

Si queremos hacer un ordenamiento descendente tendríamos primero que ordenar el array con Sort, y después aplicar un Reverse:

Array.Reverse(animales);

foreach (string i in animales)

{

Console.WriteLine(i);

}

### Comparison

Por comparación también se puede ordenar arreglos:

Array.Sort<int>(numeros, new Comparison<int>((i1, i2) => i1.CompareTo(i2)));

De este modo, el ordenamiento sería descendente.

Array.Sort<int>(numeros, new Comparison<int>((i1, i2) => i2.CompareTo(i1)));

### Linq

Podemos aprovechar el potente poder de cómputo que nos provee LINQ.

using System.Linq;

Ordenación de manera ascendente

animales = animales.OrderBy(x => x).ToArray();

De manera descendente

animales = animales. animales = animales.OrderByDescending(x => x).ToArray();

## Operaciones de arreglos

Con el paquete/namespace System.Linq podemos acceder a más métodos útiles de arreglos.

using System.Linq;

### Max

Devuelve el mayor valor.

int[] numeros = { 1, 4, -3, 6, 3 };

Console.WriteLine(numeros.Max());

// Salida 6

### Min

Devuelve el valor más pequeño.

Console.WriteLine(numeros.Min());

// Salida -3

### Sum

Devuelve la suma de los elementos.

Console.WriteLine(numeros.Sum());

// Salida 11

### Count

Devuelve el número de elementos de una secuencia.

Console.WriteLine("Conteo de números: {0}", numeros.Count());

// Salida Conteo de números: 5

# Bibliografía

**Arreglos**

[**https://sites.google.com/site/programacionnetc/arreglos**](https://sites.google.com/site/programacionnetc/arreglos)

[**https://www.latindevelopers.com/articulo/arreglos-en-csharp/**](https://www.latindevelopers.com/articulo/arreglos-en-csharp/)

[**https://www.w3schools.com/cs/cs\_arrays.php**](https://www.w3schools.com/cs/cs_arrays.php)

[**https://bravedeveloper.com/2022/02/18/como-ordenar-arreglos-en-c/**](https://bravedeveloper.com/2022/02/18/como-ordenar-arreglos-en-c/)

[**https://bravedeveloper.com/2021/03/16/aprende-linq-en-unos-minutos-y-aplicalo-en-tus-proyectos-con-c/**](https://bravedeveloper.com/2021/03/16/aprende-linq-en-unos-minutos-y-aplicalo-en-tus-proyectos-con-c/)

[**https://docs.microsoft.com/es-es/dotnet/api/system.array.indexof?view=net-6.0**](https://docs.microsoft.com/es-es/dotnet/api/system.array.indexof?view=net-6.0)

[**https://docs.microsoft.com/es-es/dotnet/api/system.linq.enumerable.count?view=net-6.0**](https://docs.microsoft.com/es-es/dotnet/api/system.linq.enumerable.count?view=net-6.0)