

Niñas
PRO (gramas
doras)



Arreglos y Vectores



Temario

- ★ Definición de Arreglo
- ★ Definición de Vector
- ★ Ejemplos
- ★ Ejemplos con Ciclo

Arreglos

¿Qué es un arreglo?

Un grupo o colección finita y ordenada de elementos.

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

Permite almacenar un conjunto de datos de un mismo tipo.

Arreglos en C++

Si conoces el tamaño de tu arreglo pero no sabes exactamente qué datos contiene, puedes declararlo así.

Tipo		Tamaño
int	edades	[4];
float	estaturas	[3];
bool	estados	[5];
string	nombres	[2];

Arreglos en C++

Si conoces tus datos puedes ponerlos en el arreglo directamente.

Tipo

```
int edades[] = {45, 21, 33, 1};
```

```
float estaturas[] = {1.50, 1.67, 1.32};
```

```
bool estados[] = {true, true, false, true, false};
```

```
string nombres[] = {"María", "Julián"};
```

Operaciones con Arreglos

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

Creamos un arreglo de números enteros de la siguiente forma:

```
int numeros [] = {-123, 49, 0, 36, 42, 33, 90};
```


Operaciones con Arreglos

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

Podemos acceder a un elemento i del arreglo de la siguiente forma:

```
cout << numeros[2]; // imprime 0
```

Vectores

¿Qué es un vector?

Es otra estructura de datos, pero que también nos sirve para almacenar datos de un mismo tipo.

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

Tenemos que incluir la biblioteca vector: `#include <vector>`

Vectores en C++

Para declarar un vector no es necesario que conozcas su tamaño.

Tipo

```
vector <int> edades;
```

```
vector <float> estaturas;
```

```
vector <bool> estados;
```

```
vector <string> nombres;
```

Vectores en C++

Si conoces tus datos puedes ponerlos directamente.

Tipo

```
vector <int> edades = {45, 21, 33, 1};
```

```
vector <float>estaturas = {1.50, 1.67, 1.32};
```

```
vector <bool> estados = {true, true, false, true};
```

```
vector <string> nombres = {"María", "Julián"};
```

Operaciones con Vectores

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

Podemos acceder a un elemento i del vector de las siguientes formas:

```
cout << numeros[2]; // Entrega 0  
cout << numeros.at(2); // Entrega 0
```

Operaciones con Vectores

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

front

Nos entrega el
primer elemento

back

Nos entrega el
último elemento

Operaciones con Vectores

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

`pop_back`

Elimina el último
elemento

`push_back`

Agrega un
elemento al final

Operaciones con Vectores

-123	49	0	36	42	33	90
0	1	2	3	4	5	6

empty

Retorna true si el
vector está vacío

size

Entrega el
tamaño del vector

Actividad

Indica qué imprime el siguiente programa.

```
using namespace std;

int main ()
{
    vector <int> notas = {43, 42, 67, 55};

    cout << notas.size() << endl;
    cout << notas.front() << endl;
    cout << notas.back() << endl;

}
```

Actividad

Indica qué imprime el siguiente programa.

```
(1) 4  
(2) 43  
(3) 55
```



Actividad

Indica qué imprime el siguiente programa.

```
using namespace std;

int main ()
{
    vector<float> alturas = {1.43, 1.42, 1.67, 1.55};

    alturas.pop_back();
    cout << alturas.size() << endl;

    alturas.push_back(1.80);
    cout << alturas.size() << endl;
}
```

Actividad

Indica qué imprime el siguiente programa.

```
(1) 3  
(2) 4
```



Actividad

Realiza la siguiente actividad de vectores con ciclos.

Escribe un programa que recibe 6 palabras y luego las imprime en orden inverso.



Actividad

Realiza la siguiente actividad de vectores con ciclos.

```
#include <iostream>
using namespace std;

int main()
{
    string palabras[6]; // Declaramos el arreglo

    // Guardamos los datos
    for (int i = 0; i < 6; i++)
    {
        cin >> palabras[i];
    }

    // Imprimimos los datos
    for (int i = 5; i >= 0; i--) {
        cout << palabras[i] << endl;
    }

    return 0;
}
```

¿Preguntas?

Línea Gráfica

Lorena Gonzalez - Diseñadora
@soygonzalez tambien

Íconos de Freepik, licenciados bajo Creative Commons BY 3.0.
<https://www.flaticon.com/authors/freepik>