

Arreglos

November 19, 2019

1 Arreglos en C++

1.1 Que aprenderemos?

- Que es un arreglo?
- Arreglos fijos vs dinamicos
- Ejemplos de uso

1.2 Que es un arreglo?

- int
- float
- char
- string
- otros arreglos

1.3 Arreglos fijos

```
tipo nombre_variable[cantidad];  
tipo nombre_variable[cantidad] = {<elementos>...};
```

```
[2]: #include<iostream>  
using namespace std;  
  
int pares[5] = {0, 2, 4, 6, 8};  
for(int i = 0; i < 5; i++){  
    cout << pares[i] << endl;  
}
```

0
2
4
6
8

```
[3]: int numeros[5] = {10, 20};  
for(int i = 0; i < 5; i++)  
    cout << numeros[i] << " ";
```

10 20 0 0 0

```
[4]: char animal[] = {'o','s','o'};
    for(int i = 0; i < 3; i++)
        cout << animal[i];
```

oso

1.4 Ingresar valores a un arreglo

```
[5]: float alturas[4];
    for(int i = 0; i < 4; i++)
        cin >> alturas[i]; // ingresar valores

    cout << endl;

    for(int j = 0; j < 4; j++)
        cout << alturas[j] << " "; //imprimir el contenido del arreglo
```

1.5
2.5
3.4
1.25

1.5 2.5 3.4 1.25

1.5 Arreglos dinamicos

```
vector<tipo> nombre_variable;
vector<tipo> nombre_variable = {...};
```

```
[6]: #include<vector>
    #include<string>

    vector<string> lista;
    cout << lista.size() << endl;
    lista.push_back("Andres");
    lista.push_back("Beatriz");
    lista.push_back("Filiberto");
    cout << lista.size() << endl;
```

0
3

```
[7]: for(int i = 0; i < lista.size(); i++)
    cout << lista[i] << endl;
```

Andres
Beatriz
Filiberto

```
[8]: cout << lista[0][0] << endl;
      for(int i = 0; i < lista.size(); i++){
          for(int j = 0; j < lista[i].length(); j++){
              cout << lista[i][j] << " ";
          }
      }
```

A
A n d r e s B e a t r i z F i l i b e r t o

```
[9]: vector<int> multiplos_tres;
      for(int i = 3; i <= 30; ){
          multiplos_tres.push_back(i);
          i+=3;
      }
      cout << multiplos_tres[0] << endl;
      cout << multiplos_tres[4] << endl;
      cout << multiplos_tres[9] << endl;
```

3
15
30

1.6 Insertar un elemento en una posicion

```
[10]: cout << lista[0] << endl;
        lista.insert(lista.begin(), "Denisse");
        cout << lista[0] << endl;
        cout << lista[1] << endl;
        lista.insert(lista.begin() + 1, "Juan");
        cout << lista[1] << endl;
```

Andres
Denisse
Andres
Juan

```
[11]: cout << lista[4] << endl;
        lista.insert(lista.end(), "Claudia");
        cout << lista[5] << endl;
```

Filiberto
Claudia

1.7 Eliminar elementos de una posicion

```
[12]: cout << multiplos_tres[0] << endl;
      multiplos_tres.erase(multiplos_tres.begin());
      cout << multiplos_tres[0] << endl;
```

3
6

1.8 Eliminar elementos

```
[15]: int i = multiplos_tres.size();
      cout << i << endl;
      while(multiplos_tres.empty() == false){ //mientras el vector no este vacio
          cout << multiplos_tres[i] << " ";
          multiplos_tres.pop_back(); //eliminar el ultimo elemento
          i--;
      }
```

9
30 30 27 24 21 18 15 12 9

1.9 Fijo vs Dinamico

1.10 Fijo

1.10.1 Ventajas

- Es mas sencillo de usar
- No requiere incluir libreria

1.10.2 Desventajas

- Para operaciones mas complejas es complicado de usar
- No podemos modificar el tamaño

1.11 Dinamico

1.11.1 Ventajas

- Se puede realizar operaciones complejas con gran facilidad
- Podemos hacer variar el tamaño

1.12 Desventajas

- Requiere usar una libreria
- Algunas de sus funciones requieren conceptos mas complejos

1.13 Tips

1.14 Cuando usar arreglos

```
[19]: vector<int> v = {1,2,3};  
for(int i : v) // visualizar los elementos de v  
    cout << i << " ";  
  
cout << endl;  
  
for(auto j : v)  
    cout << j << " ";
```

1 2 3

1 2 3

1.15 Actividades

- Dado un grupo **n** de numeros, buscar aquellos que son multiplos de **k** y mostrarlos en una linea
- Dado un grupo de **n** nombres, buscar aquellos que inicien con una letra solicitada