



Grado en Ingeniería Información

PROGRAMACIÓN I

Sesión 7

Curso 2022-2023

Marta N. Gómez (mgomezper@nebrija.es)

Tipos de datos estructurados.

- `std::string`
- `std::array`
- **`std::vector`**
- `std::set`



ÍNDICE

Tipo de datos **vector** (`std::vector`)

- El tipo **vector** es una secuencia ordenada de datos homogéneos, es decir, del mismo tipo.
- Su **declaración** sólo necesita que se indique el **tipo de dato** del se guardará en el vector.

`std::vector<tipo> var_vector;`

- Su tamaño inicial será 0 (al declararlo) y modificará su tamaño según se vayan añadiendo o eliminando elementos en él.
- Necesita incluir la biblioteca de C++ **vector**:

`#include <vector>`

Tipo de datos **vector** (`std::vector`)

- Declaración del tipo **vector**:

```
std::vector<float> vector1;
```

Se crea un vector de números decimales **vacío**.

- Inicialización del tipo **vector**:
 - Indicando sus valores entre llaves y separados por comas.

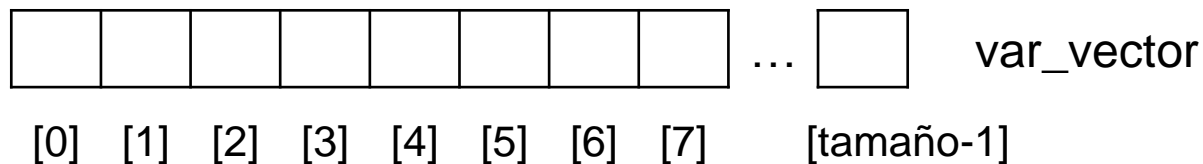
```
std::vector<int> un_vector{1,2,3,4,5,9,8,7,6};
```

- Utilizando el operador **=**

```
std::vector<char> vocales ={'a','e','i','o','u'};
```

Tipo de datos **vector** (`std::vector`)

- El tipo **vector** es un tipo de datos contenedor.
- Los elementos de un vector son accesibles a través de su **índice** (posición).
- El primer elemento tiene el **índice 0** y el último el **índice (tamaño - 1)**.



Tipo de datos **vector** (std::vector)

- El tipo **vector** es una **clase**:
 - Sus variables son **objetos**.
 - Contiene **miembros**: **atributos** (variables) y **métodos** (funciones).
 - El operador **punto** (.) permite el acceso a los **miembros** de la clase:

nom_var.**miembro**

Tipo de datos vector (std::vector)

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> vector1 = {1, 2, 3, 4, 5};

    cout << "\n\n\tEL VECTOR ES\n\n";
    cout << "\t( ";
    for (int dato:vector1)
    {
        cout << dato << " ";
    }
    cout << ")\n";

    cout << "\n\n\t";
    return 0;
}
```

EL VECTOR ES

(1 2 3 4 5)

Press <RETURN> to close this window...

Tipo de datos vector (std::vector) - Funciones

at Permite el acceso a una posición determinada del *vector*.

```
#include <iostream>
#include <vector>
```

```
using namespace std;
```

```
int main()
{
```

```
    vector<int> vector1 = {2, 4, 8, 0, 6};
```

```
    // Acceso a un determinado elemento
```

```
    cout << "\n\n\tvector(2): " << vector1.at(2) << endl;
```

```
    // Modificación del valor de un elemento
```

```
    vector1.at(0) = 20;
```

```
    cout << "\n\n\tvector(0): " << vector1.at(0) << endl;
```

```
vector(2): 8
```

```
vector(0): 20
```

```
Press <RETURN> to close this window...
```


Tipo de datos vector (std::vector) - Funciones



Operador [] Permite el acceso a una posición determinada del *vector*.

```
#include <vector>
```

```
using namespace std;
```

```
int main()  
{
```

```
    vector<int> vector1 = {2, 4, 8, 0, 6};
```

```
    // Acceso a un determinado elemento
```

```
    cout << "\n\n\tvector(2): " << vector1[2] << endl;
```

```
    // Modificar el valor de un elemento
```

```
    vector1[0] = 20;
```

```
    cout << "\n\n\tvector(0): " << vector1[0] << endl;
```

```
vector(2): 8
```

```
vector(0): 20
```

```
Press <RETURN> to close this window...
```

Tipo de datos vector (std::vector) - Funciones

front Devuelve el **primer elemento** del *vector*.

back Devuelve el **último elemento** del *vector*.

```
int main()
{
    vector<int> vector1 = {2, 4, 8, 0, 6};
    vector<char> vocales = {'a', 'e', 'A', 'E'};

    // Acceso al primer elemento
    cout << "\n\n\tvector(0): " << vector1.front() << endl;

    // Modificar el valor del primer elemento
    vector1.front() = 20;
    cout << "\n\n\tvector(0): " << vector1.front() << endl;

    // Acceso al último elemento
    cout << "\n\n\tvocales(3): " << vocales.back() << endl;

    // Modificar el valor del último elemento
    vocales.back() = 'O';
    cout << "\n\n\tvocales(4): " << vocales.back() << endl;
```

```
vector(0): 2
```

```
vector(0): 20
```

```
vocales(3): E
```

```
vocales(4): O
```

```
Press <RETURN> to close
```

Tipo de datos vector (std::vector) - Funciones

size Devuelve el número de elementos que tiene el *vector*.

```
#include <iostream>
#include <vector>

using namespace std;
```

```
int main()
{
```

```
    vector<int> vec1 = {2, 4, 8, 0, 6};
    vector<char> vocales;
```

```
    cout << "\n\n\tEl numero de elementos de vec1 es: " << vec1.size() << endl;
    cout << "\n\n\tEl numero de elementos de vocales es: " << vocales.size() << endl;
```

```
El numero de elementos de vec1 es: 5
```

```
El numero de elementos de vocales es: 0
```

```
Press <RETURN> to close this window...
```

clear Elimina todos los elementos de un *vector*, dejando su tamaño a 0.

```
#include <iostream>
#include <vector>

using namespace std;
```

```
int main()
{
```

```
    vector<int> vec1 = {2, 4, 8, 0, 6, 9, 1};
```

```
    cout << "\n\n\tEl numero de elementos de vec1 es: " << vec1.size() << endl;
```

```
    vec1.clear();
```

```
    cout << "\n\n\tAhora el numero de elementos de vec1 es: " << vec1.size() << endl;
```

```
El numero de elementos de vec1 es: 7
```

```
Ahora el numero de elementos de vec1 es: 0
```

```
Press <RETURN> to close this window...
```

push_back Añade un elemento al final del *vector*.

```
#include <iostream>
#include <vector>

using namespace std;
```

```
int main()
{
```

```
    vector<char> vocales{'a','e','A','E'};
```

```
    cout << "\n\n\tEl numero de vocales es: " << vocales.size() << endl;
    vocales.push_back('I');
    cout << "\n\n\tAhora el numero de vocales es: " << vocales.size() << endl;
    cout << "\n\n\tLa ultima vocal es: " << vocales.back();
```

```
El numero de vocales es: 4
```

```
Ahora el numero de vocales es: 5
```

```
La ultima vocal es: I
```

```
Press <RETURN> to close this window...
```

pop_back Elimina el último elemento del *vector*.

```
#include <iostream>
#include <vector>

using namespace std;
```

```
int main()
{
```

```
    vector<char> vocales{'a','e','A','E'};
```

```
    cout << "\n\n\tEl numero de vocales es: " << vocales.size() << endl;
```

```
    vocales.pop_back();
```

```
    cout << "\n\n\tAhora el numero de vocales es: " << vocales.size() << endl;
```

```
    cout << "\n\n\tLa ultima vocal es: " << vocales.back();
```

```
El numero de vocales es: 4
```

```
Ahora el numero de vocales es: 3
```

```
La ultima vocal es: A
```

```
Press <RETURN> to close this window...
```

Tipo de datos **vector** (std::vector)

- El tipo de elementos de un **vector** pueden ser cualquiera:

int, char, string, bool, struct, array, vector.

```
vector <vector<tipo, tam2>, tam1> var_vector;
```

Tipo de datos

- Ejemplo

```
vector<vector<int>> vec1;
```

```
vector<vector<int>> vec2 {{0 1}, {2}, {3, 4, 5}};
```

Tipo de datos vector (std::vector)

```
#include <iostream>
#include <vector>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    vector<vector<int>> vec1 {{1, 1, 0}, {0, 2}, {2, 1, 3}};
```

```
    cout << "\n\n\tDATOS DEL VECTOR 1";
```

```
    cout << "\n\t===== \n\n";
```

```
    for(unsigned int a=0;a<vec1.size();a++){
```

```
        for(unsigned int b=0;b<vec1[a].size();b++){
```

```
            cout << "\t" << vec1[a][b];
```

```
        }
```

```
DATOS DEL VECTOR 1
```

```
=====
```

```
1      1      0      0      2      2      1      3
```

```
Press <RETURN> to close this window...
```


Tipo de datos vector (std::vector)

```
vector<vector<int>> vec2;  
vector<int> unafila;
```

```
int fil;  
int col;
```

```
cout << "\n\n\n\n\tIndique el numero de filas: ";  
cin >> fil;  
cout << "\n\n\tIndique el numero de columnas: ";  
cin >> col;
```

```
cout << "\n\n\n\n\tDATOS DEL VECTOR 2";  
cout << "\n\t=====
```

```
for(int a = 0; a < fil; a++)  
{  
    vec2.push_back(unafila);  
    for(int b = 0; b < col; b++)  
    {  
        vec2.at(a).push_back(a*b); // Forma 1  
        // vec2[a].push_back(a*b); // Forma 2  
    }  
}
```

```
for(int a = 0; a < fil; a++)  
{  
    cout << endl << endl;  
    for(int b = 0; b < col; b++)  
    {  
        cout << "\t" << vec2[a][b] << " ";  
    }  
}
```

Indique el numero de filas: 3

Indique el numero de columnas: 4

DATOS DEL VECTOR 2
=====

0	0	0	0
---	---	---	---

0	1	2	3
---	---	---	---

0	2	4	6
---	---	---	---

Press <RETURN> to close this window...

Tipo de datos vector (std::vector)

```
vector<vector<int>> vec2;  
vector<int> unafila;  
int fil;  
int col;  
  
cout << "\n\n\n\n\tIndique el numero de filas: ";  
cin >> fil;  
cout << "\n\n\tIndique el numero de columnas: ";  
cin >> col;  
  
cout << "\n\n\n\n\tDATOS DEL VECTOR 2";  
cout << "\n\t=====";  
for(int a = 0; a < fil; a++)  
{  
    vec2.push_back(unafila);  
    for(int b = 0; b < col; b++)  
    { vec2.at(a).push_back(a*b); // Forma 1  
      // vec2[a].push_back(a*b); // Forma 2  
    }  
}  
  
for(vector<int> &filaV:vec2)  
{ cout << endl << endl;  
  for(auto dato:filaV)  
  {  
      cout << "\t" << dato << " ";  
  }  
}
```

Indique el numero de filas: 4

Indique el numero de columnas: 3

DATOS DEL VECTOR 2
=====

0	0	0
---	---	---

0	1	2
---	---	---

0	2	4
---	---	---

0	3	6
---	---	---

Press <RETURN> to close this window...