Grado en Ingeniería Información

PROGRAMACIÓN I

Sesión 6

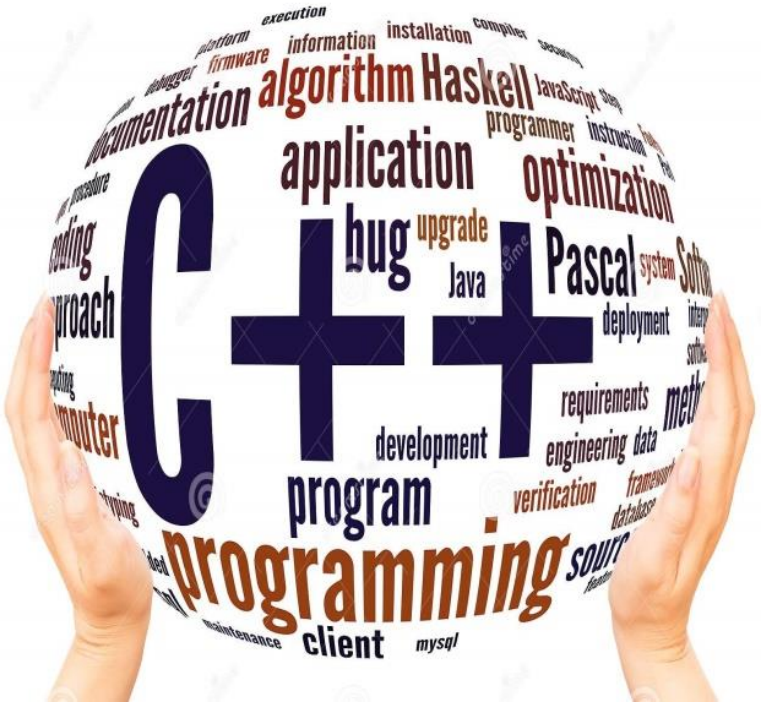
Curso 2022-2023

Marta N. Gómez (mgomezper@nebrija.es)

UNIVERSIDAD NEBRIJA

Tipos de datos estructurados.

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



ÍNDICE

Bucle *for(:)*

- Los tipos de datos: ***string, array, vector*** están formados por una serie de elementos de un tipo de datos determinado (*tipo_elem*) y son ***tipos de datos iterables*** (*tipo_iterable*).
- ***Iterar*** significa recorrer **elemento a elemento** dicho tipo de datos.

10	20	30	40	50	60	70	80	90

I	N	F	O	R	M	A	T	I	C	A

Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
Lectura	Natacion	Ingles	Pintura	Cocina	Patinaje	Relax
18:00	16:00	18:00	10:00	10:00	12:00	0:00

Bucle *for(:)*

- Estructura: ***for (elem:variable)***

```
std::tipo_iterable  etiqueta_tipoIter;
for (tipo_elem   etiqueta_tipoElem: etiqueta_tipoIter) {
  //  cuerpo del bucle  }
```

tipo_iterable: será un tipo ***string, vector, array*** que se declare.

etiqueta_tipoIter: identificador de la variable del tipo de dato iterable, estructura iterable.

tipo_elem: tiene que ser del mismo tipo de datos que los elementos contenidos en la estructura iterable.

etiqueta_tipoElem: identificador de la variable del tipo de dato contenido en la estructura iterable. Tomará el valor de los diferentes elementos contenidos en la estructura iterable.

Bucle *for(:)*

- Ejemplo de ***for (elem:variable)***

```
std::string  nombre {"MARTA"};
std::cout << "\n\n\tLas letras del nombre son ";
for (char   letra: nombre) {
    std::cout << letra << '-';  }
```

[0]	[1]	[2]	[3]	[4]
M	A	R	T	A

```
        Las letras del nombre son M-A-R-T-A-

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

Bucle *for(:)*

- Hay que diferenciar entre la variable de tipo iterable (***etiqueta_tipoIter***) y el valor de los datos que contiene (***etiqueta_tipoElem***).
- Aunque se modifique el valor de los datos de la variable iterable, no se actualiza su contenido

Bucle *for(:)*

```
#include <string>

using namespace std;

int main()
{
    string nombre{"MARTA"};

    cout << "\n\n\tLas letras del nombre son ";

    int cont {1};
    for (char letra:nombre)
    {    cout << "\n\tletra" << cont << " - " << letra;
         letra = '*';
         cout << " ahora letra" << cont << " - " << letra;
         cont++;
    }

    cout << "\n\n\tAhora las letras del nombre son ";

    cont=1;
    for (char letra:nombre)
    {    cout << "\n\tletra" << cont << " - " << letra;
         cont++;
    }
    cout << endl << endl;
```

```
        Las letras del nombre son
        letra1 - M ahora letra1 - *
        letra2 - A ahora letra2 - *
        letra3 - R ahora letra3 - *
        letra4 - T ahora letra4 - *
        letra5 - A ahora letra5 - *

        Ahora las letras del nombre son
        letra1 - M
        letra2 - A
        letra3 - R
        letra4 - T
        letra5 - A


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

Bucle *for(:)*

- Se puede interrumpir la ejecución del bucle ***for*** con la sentencia ***break;***

```
int main()
{
    string nombre{"MARTA"};
    int cont {1};

    for (char letra:nombre)
    {   cout << "\n\tletra" << cont << " - " << letra;
        if (letra == 'R') break;
        cont++;
    }

    cout << "\n\n\tFIN DEL PROCESO";
```

```
        letra1 - M
        letra2 - A
        letra3 - R

        FIN DEL PROCESO

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

Bucle *for(:)*

- Se puede cortocircuitar la ejecución del bucle ***for*** con la sentencia ***continue;***
- Se recorre el objeto iterable, pero cuando se cumplan ciertas condiciones se pasa a la siguiente iteración.

```
int main()
{
    string nombre{"MARTA"};
    int cont {1};

    for (char letra:nombre)
    {
        if (letra == 'A') continue;
        cout << "\n\tletra" << cont << " - " << letra;
        cont++;
    }

    cout << "\n\n\tFIN DEL PROCESO";
```

```
        letra1 - M
        letra2 - R
        letra3 - T

        FIN DEL PROCESO

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

Tipo de datos array (std::array)

UNIVERSIDAD NEBRIJA

- El tipo ***array*** es una secuencia ordenada de datos homogéneos, es decir, del mismo tipo.
- Su declaración necesita que se indique un **tamaño fijo** (número de datos que incluirá) y el **tipo de dato** de los mismos.

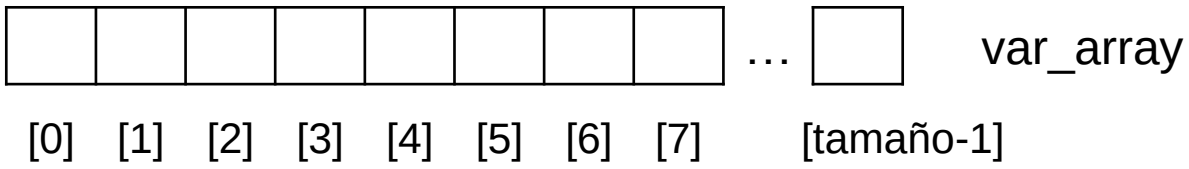
std::array<tipo, tamaño> var_array;

- Necesita incluir la biblioteca de C++ ***array:***

#include <array>

Tipo de datos array (std::array)

- El tipo ***array*** es un tipo de datos contenedor.
- Los elementos de un array 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 array (std::array)

- El tipo ***array*** 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 array (std::array)

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

```
std::array<int,10> un_array{1,2,3,4,5,9,8,7,6};
```

  - Utilizando el operador **=**

```
std::array<char,5> vocales ={'a','e','i','o','u'};
```

Tipo de datos array (std::array)

```cpp
#include <iostream>
#include <array>


using namespace std;

int main()
{
    std::array<int,10> un_array{1,2,3,4,5,9,8,7,6};

    std::array<char,5> vocales={'a','e','i','o','u'};

    cout << "\n\n\tLAS VOCALES SON: ";
    for (char letra: vocales)
        cout << letra << "-";

    cout << "\n\n\tLOS NUMEROS SON: ";
    for (int num: un_array)
        cout << num << "-";

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

```
LAS VOCALES SON: a-e-i-o-u-

LOS NUMEROS SON: 1-2-3-4-5-9-8-7-6-0-

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

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

UNIVERSIDAD NEBRIJA

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

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

```
El numero de la posicion 3 es: 4

El numero ahora es: 40

La vocal de la posicion 2 es: i

La vocal ahora es: I

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

```cpp
#include <iostream>
#include <array>


using namespace std;

int main()
{
    std::array<int,10> un_array{1,2,3,4,5,9,8,7,6};

    std::array<char,5> vocales={'a','e','i','o','u'};

    cout << "\n\n\tEl numero de la posicion 3 es: " << un_array.at(3) << endl;

    // Modificar el valor de una posicion
    un_array.at(3) = un_array.at(3) * 10;
    cout << "\n\n\tEl numero ahora es: " << un_array.at(3) << endl;

    cout << "\n\n\tLa vocal de la posicion 2 es: " << vocales[2] << endl;

    // Modificar el valor de una posicion
    vocales[2] = 'I';
    cout << "\n\n\tLa vocal ahora es: " <<  vocales[2] << endl;
```

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

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

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

```
El primer numero es: 1

El primer numero ahora es: 10

La ultima vocal es: u

La ultima vocal ahora es: U

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

```cpp
#include <iostream>
#include <array>


using namespace std;

int main()
{
    std::array<int,10> un_array{1,2,3,4,5,9,8,7,6};

    std::array<char,5> vocales={'a','e','i','o','u'};

    cout << "\n\n\tEl primer numero es: " << un_array.front() << endl;

    // Modificar el valor de la primera posicion
    un_array.front() = 10;
    cout << "\n\n\tEl primer numero ahora es: " << un_array.front() << endl;

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

    // Modificar el valor de la última posicion
    vocales.back() = 'U';
    cout << "\n\n\tLa ultima vocal ahora es: " <<  vocales.back() << endl;
```

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

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

```
El numero de elementos del array es: 10

El numero de vocales es: 5

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

```cpp
#include <iostream>
#include <array>


using namespace std;

int main()
{
    array<int,10> un_array{1,2,3,4,5,9,8,7,6};

    array<char,5> vocales={'a','e','i','o','u'};

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

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

fill Rellena el ***array*** con un valor determinado.

```
array<char,5> vocales={'a','e','i','o','u'};

vocales.fill('A'); // todos los elementos son 'A'

cout << "\n\n\tEL ARRAY VOCALES CONTIENE: ";
    for (char letra: vocales)
        cout << letra << "-";
```

```
EL ARRAY VOCALES CONTIENE: A-A-A-A-A-

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

UNIVERSIDAD NEBRIJA

Tipo de datos array (std::array)

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

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

array <array<tipo, tam2>, tam1> var_array;

Tipo de datos

```
#include <iostream>
#include <array>


using namespace std;

int main()
{
    array<array<int,2>,3> una_matriz;

    una_matriz.fill({0,0});
    cout << "\n\n\tLa MATRIZ ES\n\n";

    for (int fil{0};fil < 3; fil++)
      { cout << "\t| ";
        for (int col{0};col < 2; col++)
         cout << una_matriz[fil][col] << " ";
        cout << "|\n";
      }
```

```
La MATRIZ ES

| 0 0 |
| 0 0 |
| 0 0 |


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

Tipo de datos array (std::array)

UNIVERSIDAD NEBRIJA

```
int main()
{
    array<array<int,2>,3> una_matriz;

    cout << "\n\n\tDATOS DE LA MATRIZ\n\n";
    for (int fila{0};fila < 3; fila++)
           for (int col{0};col < 2; col++)
           { cout << "\n\tuna_matriz [" << fila << ", " << col <<"] = ";
             cin >> una_matriz[fila][col];  }

    cout << "\n\n\tLA MATRIZ ES\n\n";
    for (int fila{0};fila < 3; fila++)
      { cout << "\t| ";
        for (int col{0};col < 2; col++)
             cout << una_matriz[fila][col] << " ";
        cout << "|\n";
      }
```

```
DATOS DE LA MATRIZ


una_matriz [0, 0] = 1

una_matriz [0, 1] = 2

una_matriz [1, 0] = 3

una_matriz [1, 1] = 4

una_matriz [2, 0] = 5

una_matriz [2, 1] = 6


LA MATRIZ ES

| 1 2 |
| 3 4 |
| 5 6 |


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

Tipo de datos array (std::array)

UNIVERSIDAD NEBRIJA

```
#include <iostream>
#include <array>

using namespace std;

int main()
{
    array<array<int,2>,3> una_matriz;

    cout << "\n\n\tDATOS DE LA MATRIZ\n\n";
    for (int fila{0}; fila < 3; fila++)
        for (int col{0}; col < 2; col++)
         { cout << "\n\tuna_matriz [" << fila << "
           cin >> una_matriz[fila][col];  }

    cout << "\n\n\tLA MATRIZ ES\n\n";
    for (array<int,2> fila:una_matriz)
    {  cout << "\t| ";
        for (int dato:fila)
         { cout << dato << " ";  }
       cout << "|\n";
    }
```

```
DATOS DE LA MATRIZ


una_matriz [0, 0] = 1

una_matriz [0, 1] = 2

una_matriz [1, 0] = 3

una_matriz [1, 1] = 4

una_matriz [2, 0] = 5

una_matriz [2, 1] = 6


LA MATRIZ ES

| 1 2 |
| 3 4 |
| 5 6 |


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

Estructuras de datos (*struct*)

- Los tipos de datos simples (***int, float, char,*** etc.) permiten crear nuevos tipos de datos.
- Las estructuras de datos agrupan, bajo el mismo ***nombre***, **elementos** del **mismo** o **distinto tipo de datos** relacionados entre sí.
- El acceso a cada uno de sus **elementos/campos** se hace a través del **operador punto** (**.**).
- **Sintaxis** del tipo de datos ***struct***:

```
struct nombre_estructura
{       tipo1    campo1;
        tipo2    campo2;
        …
        tipoN    campoN;
};
```

Declaración, inicialización, asignación y acceso a las variables

UNIVERSIDAD NEBRIJA

```
#include <iostream>
#include <array>

using namespace std;


struct Persona{
    int edad;
    float altura;
    float peso;
  };

int main()
{
    // Declaracion de variables de tipo Persona

    Persona carla, maria;

    // Asociacion de valores a los miembros de las variables de tipo Persona
    carla.edad = 26;
    carla.altura = 168.3;
    carla.peso = 52.6;

    maria.edad = 26;
    maria.altura = 172.5;
    maria.peso = 55;

    // Acceso a los miembros de las variables de tipo Persona
    cout << "\n\n\tDatos de Carla.- " << "     edad: " << carla.edad
         << "     altura: " << carla.altura << "     peso: "  << carla.peso << endl;

    cout << "\n\n\tDatos de Maria.- " << "     edad: " << maria.edad
         << "     altura: " << maria.altura << "     peso: "  << maria.peso << endl;

    cout << "\n\n\t";

    return 0;
}
```

```
Datos de Carla.-      edad: 26     altura: 168.3     peso: 52.6

Datos de Maria.-      edad: 26     altura: 172.5     peso: 55

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

Declaración, inicialización y acceso de variables

```
#include <iostream>
#include <array>

using namespace std;


struct Persona{
    int edad;
    float altura;
    float peso;
  };

int main()
{
    // Declaración e inicialización de variables de tipo Persona

    Persona luis{
        30,
        194.5,
        77.8
      };

    // Acceso a los miembros de la variable de tipo Persona
    cout << "\n\n\tDatos de Luis.- " << "     edad: " << luis.edad
         << "     altura: " << luis.altura << "     peso: "  << luis.peso << endl;

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

```
Datos de Luis.-      edad: 30     altura: 194.5     peso: 77.8

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

// No recomendable

Estructuras anidadas

```cpp
// Crea un nuevo tipo de datos llamado Equipo
  struct Equipo{
    Persona jugador1, jugador2, jugador3;
    unsigned int juegos_ganados;
    unsigned int juegos_perdidos;
    unsigned int juegos_empatados;
  };

int main()
{
    // Declaracion de variables de tipo Persona
    Persona carla, maria, juan;

    // Asociacion de valores a los miembros de las variables de tipo Persona
    carla.edad = 26;  carla.altura = 168.3;  carla.peso = 52.6;

    maria.edad = 26;  maria.altura = 172.5;  maria.peso = 55;

    juan.edad = 24;   juan.altura = 174;     juan.peso = 66.4;

    // Declaración de una variable de tipo Equipo
    Equipo mi_equipo{
      carla, maria,
      {44, 185.5, 78.5},  //juan,
      10,
      3,
      5
    };

    maria.edad = maria.edad + 1;

    // Acceso a los miembros de las variables de tipo Persona
    cout << "\n\n\tEdad de Maria: " << maria.edad << "\n";
    cout << "\n\n\tEdad de Juan: " << juan.edad << "\n";

    cout << "\n\n\tEdad de Carla (jugadora 1): "<< mi_equipo.jugador1.edad << "\n";
    cout << "\n\n\tEdad de María (jugadora 2): "<< mi_equipo.jugador2.edad << "\n";
    cout << "\n\n\tEdad del jugador3 del equipo: "<< mi_equipo.jugador3.edad << "\n";

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

// La ejecución muestra como se modifica
// el valor de la variable **maria**, pero no
// cambia el valor de la variable **mi_equipo**

```
Edad de Maria: 27

Edad de Juan: 24

Edad de Carla (jugadora 1): 26

Edad de Maria (jugadora 2): 26

Edad del jugador3 del equipo: 44

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

Estructuras anidadas

```cpp
#include <iostream>
#include <array>

using namespace std;

  struct Persona{
    int edad;
    float altura;
    float peso;
  };

// Crea un nuevo tipo de datos llamado Otro_Equipo
   struct Otro_Equipo{
      array <Persona, 5> quinteto; // quinteto es un array de 5 elementos de tipo Persona

      unsigned int juegos_ganados;
      unsigned int juegos_perdidos;
      unsigned int juegos_empatados;
    };


int main()
{
    // Declaracion de variables de tipo Persona
    Persona carla, maria, juan, luis;

    // Asociacion de valores a los miembros de las variables de tipo Persona
    maria.edad = 26;  maria.altura = 172.5;  maria.peso = 55;
    juan.edad = 24;   juan.altura = 174;     juan.peso = 66.4;
    luis.edad = 38;   luis.altura = 184.5;   luis.peso = 79.4;
    carla.edad = 26;  carla.altura = 168.3;  carla.peso = 52.6;

    Otro_Equipo mi_equipo {
       {maria, juan, luis, carla, maria},
       5,
       3,
       4
     };

    cout << "\n\n\tEdad del jugador3 del equipo: "<< mi_equipo.quinteto.at(2).edad << "\n";

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

```
Edad del jugador3 del equipo: 38


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

de datos (*struct*)

Estructuras de datos (*struct*) - Estructuras anidadas

```
#include <iostream>
#include <array>

using namespace std;

  struct Persona{
    int edad;
    float altura;
    float peso;
  };

// Crea un nuevo tipo de datos llamado Otro_Equipo
   struct Otro_Equipo{
      array <Persona, 5> quinteto; // quinteto es un array de 5
      unsigned int juegos_ganados;
      unsigned int juegos_perdidos;
      unsigned int juegos_empatados;
    };

   int main()
   {   // Declaracion de variables de tipo Persona
       Persona carla, maria, juan, luis;
        Otro_Equipo mi_equipo;

       // Asociacion de valores a los miembros de las variables de tipo Persona
       maria.edad = 26;  maria.altura = 172.5;  maria.peso = 55;
       juan.edad = 24;   juan.altura = 174;     juan.peso = 66.4;
       luis.edad = 38;   luis.altura = 184.5;   luis.peso = 79.4;
       carla.edad = 26;  carla.altura = 168.3;  carla.peso = 52.6;

       mi_equipo.quinteto.at(0) = maria;
       mi_equipo.quinteto.at(1) = juan;
       mi_equipo.quinteto.at(2) = luis;
       mi_equipo.quinteto.at(3) = carla;
       mi_equipo.quinteto.at(4).edad = 22;  mi_equipo.quinteto.back().altura = 190;
       mi_equipo.quinteto.at(mi_equipo.quinteto.size() - 1).peso = 82;
       mi_equipo.juegos_ganados = 6; mi_equipo.juegos_perdidos = 0;
       mi_equipo.juegos_empatados = 2;

       cout << "\n\n\tEdad del jugador5 del equipo: "<< mi_equipo.quinteto.at(4).edad << "\n";
       cout << "\n\n\tAltura del jugador5 del equipo: "<< mi_equipo.quinteto.back().altura << "\n";
       cout << "\n\n\tPeso del jugador5 del equipo: "<< mi_equipo.quinteto.at(4).peso << "\n";
       cout << "\n\n\tNumero de partidos gandos por el equipo: "<< mi_equipo.juegos_ganados << "\n";

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

```
Edad del jugador5 del equipo: 22

Altura del jugador5 del equipo: 190

Peso del jugador5 del equipo: 82

Numero de partidos gandos por el equipo: 6

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