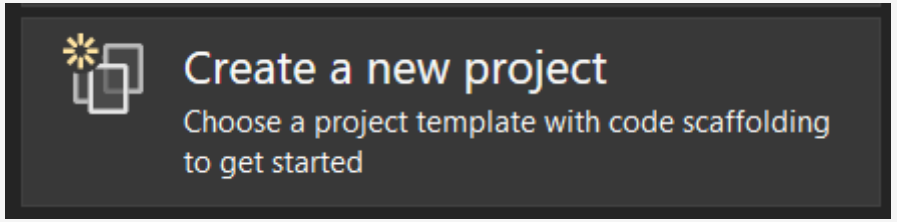
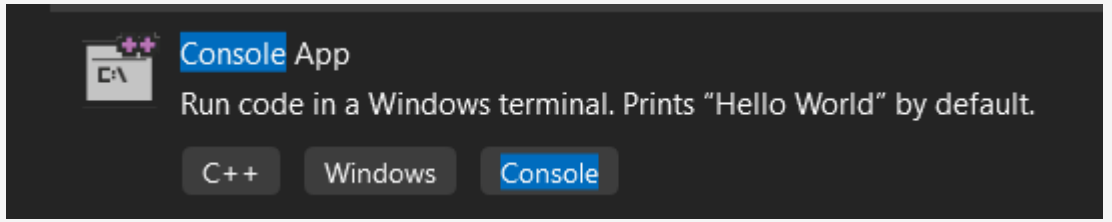
Práctica 02. Vectores en C++

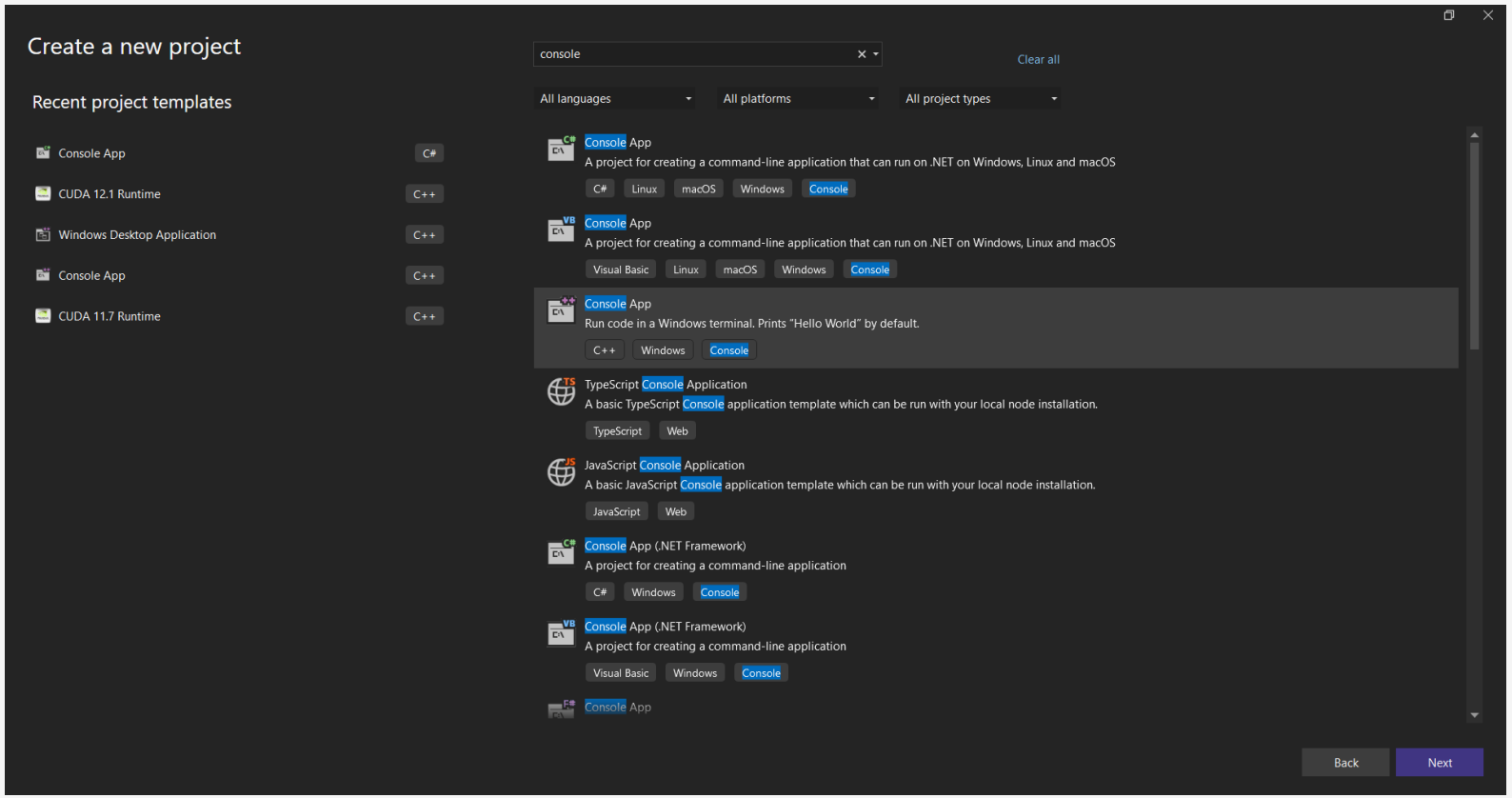
Vectores unidimensionales en C++

U.A.Q. Fac. de Informática
Dra. Sandra Luz Canchola Magdaleno
Correo: sandra.canchola@uaq.mx
Dra. Reyna Moreno Beltrán
Correo: reyna.moreno@uaq.mx

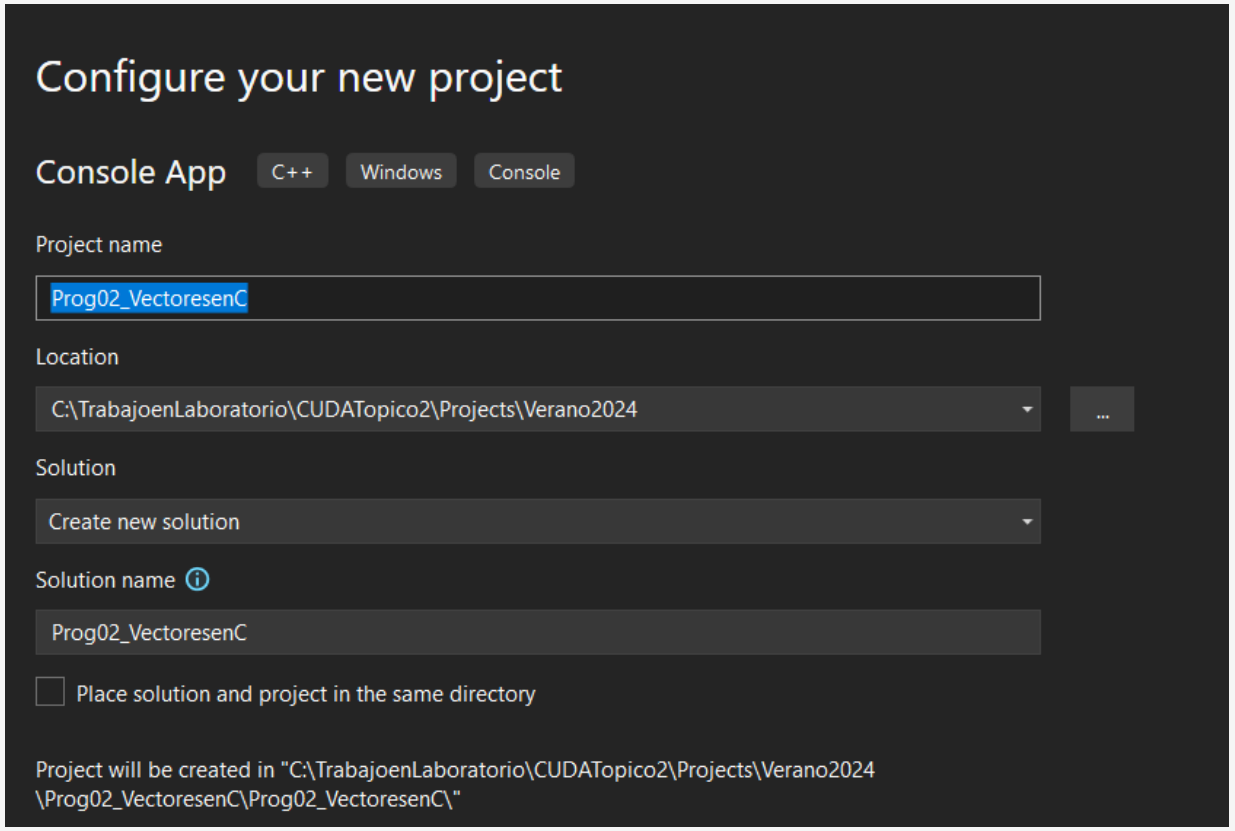
Crear un C++ Console App







Crear un C++ Console App



Comandos

Definir un vector

```
Type vectName[size];

Ejemplo:
float array1[5];
```

Definir un vector como apuntador

```
Type* pointerName;
pointerName = (Type *)malloc(size * sizeof(Type));

Ejemplo:
float *array2;
array2 = (float *)malloc(numRen * sizeof(float));
```

Comandos

Acceder a un elemento del vector

```
vectName[i]=valor;

Ejemplo:
array1[0]=15;
```

Acceder a un elemento del vector

```
*(pointerName + step)=valor;

Ejemplo:
*(array1 + step)=100;
```

Conocer la dirección de memoria de un vector

```
&vectName
&vectName[0]

Ejemplo:
&array2[i]
```

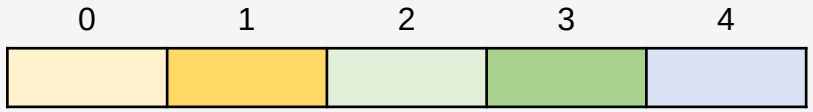
Liberar la memoria reservada con el comando malloc

```
free(pointerName);

Ejemplo:
free(array2);
```

Elementos de vectores

Vector 1 x 5



Índice de elementos

`vector[i]`

`donde:` $i \in [0 \quad n-1]$

Memoria

CPU (Host)

A01	numReng	5				
A05						
A10	array1				...	
A15	array2				...	
A20						
B01						
B15						
C01	array1Ptr	A10				
C05						
C20						
C30						
E07						
E10						

Código

```
////////////////////////////////////////////////
//              Program 02 Arrays             //
// Autor: Sandra Luz Canchola Magdaleno         //
// Email: sandracanchola@yahoo.com    y         //
//        sandra.canchola@uaq.mx                //
////////////////////////////////////////////////

#include <iostream>
#include <time.h>
using namespace std;
const int numRen = 5;

int main(int argc, char* argv[])
{
            float array1[numRen];
            float *array2;
            float *array1Ptr;

            array1Ptr = &array1[0];
            array2 = (float *)malloc(numRen * sizeof(float));
            srand((unsigned)time(NULL));

            for (int i = 0; i < numRen; i++)
            {
                        array1[i] = ((float)rand() / (float)RAND_MAX) * 100;
                        array2[i] = ((float)rand() / (float)RAND_MAX) * 100;
            }

            cout << "================================================\n";
            cout << "The memory addresses of the variables are:\n";
            cout << "A ) array1 is: " << &array1[0] << "\n";
            cout << "A') array1 is: " << &array1 << "\n";
            cout << "B ) array2 is: " << &array2 << "\n";
```

Código

```
cout << "================================================\n";
cout << "The contents of the pointer variables are:\n";
cout << "A) array1Ptr is: " << array1Ptr << "\n";
cout << "B) array2 is: " << array2 << "\n";

cout << "================================================\n";
cout << "The memory addresses of the elements of the arrays are:\n";
for (int i = 0; i < numRen; ++i)
{
          cout << i + 1 << ") array1[" << i << "]: " << &array1[i];
          cout << " -- array2[" << i << "]: " << &array2[i] << "\n";
}

cout << "================================================\n";
cout << "The content of the arrays are:\n";
for (int i = 0; i < numRen; ++i)
{
          cout << i + 1 << ") array1[" << i << "]: " << array1[i];
          cout << " -- array2[" << i << "]: " << array2[i] << "\n";
}

cout << "================================================\n";
cout << "The content of the arrays using index are:\n";
for (int i = 0; i < numRen; ++i)
{
          array1[i] = array1[i] + 100;
          array2[i] = array2[i] + 100;
          cout << i + 1 << ") array1[" << i << "]: " << array1[i];
          cout << " -- array2[" << i << "]: " << array2[i] << "\n";
}
```

Código

```
                cout << "=================================================\n";
                cout << "The content of the arrays using pointers are:\n";
                for (int i = 0; i < numRen; ++i)
                {
                        *(array1 + i) = *(array1 + i) + 20;
                        *(array1Ptr + i) = *(array1Ptr + i) + 20;
                        *(array2 + i) = *(array2 + i) + 20;
                        cout << i + 1 << ") array1[" << i << "]: " << array1[i];
                        cout << " -- array2[" << i << "]: " << array2[i] << "\n";
                }

                cout << "=================================================\n";

                free(array2);
                system("pause");
                return 0;
        }
```

Corrida

C:\TrabajoenLaboratorio\CUD

```
================================================
The memory addresses of the variables are:
A ) array1 is: 00000079D994F858
A') array1 is: 00000079D994F858
B ) array2 is: 00000079D994F888
================================================
The contents of the pointer variables are:
A) array1Ptr is: 00000079D994F858
B) array2 is: 000001A1F7A7E3C0
================================================
The memory addresses of the elements of the arrays are:
1) array1[0]: 00000079D994F858 -- array2[0]: 000001A1F7A7E3C0
2) array1[1]: 00000079D994F85C -- array2[1]: 000001A1F7A7E3C4
3) array1[2]: 00000079D994F860 -- array2[2]: 000001A1F7A7E3C8
4) array1[3]: 00000079D994F864 -- array2[3]: 000001A1F7A7E3CC
5) array1[4]: 00000079D994F868 -- array2[4]: 000001A1F7A7E3D0
================================================
The content of the arrays are:
1) array1[0]: 48.6587 -- array2[0]: 48.1368
2) array1[1]: 21.0395 -- array2[1]: 77.1325
3) array1[2]: 68.0044 -- array2[2]: 17.307
4) array1[3]: 55.5467 -- array2[3]: 76.6625
5) array1[4]: 72.9667 -- array2[4]: 53.6882
================================================
The content of the arrays using index are:
1) array1[0]: 148.659 -- array2[0]: 148.137
2) array1[1]: 121.039 -- array2[1]: 177.132
3) array1[2]: 168.004 -- array2[2]: 117.307
4) array1[3]: 155.547 -- array2[3]: 176.663
5) array1[4]: 172.967 -- array2[4]: 153.688
================================================
The content of the arrays using pointers are:
1) array1[0]: 188.659 -- array2[0]: 168.137
2) array1[1]: 161.039 -- array2[1]: 197.132
3) array1[2]: 208.004 -- array2[2]: 137.307
4) array1[3]: 195.547 -- array2[3]: 196.663
5) array1[4]: 212.967 -- array2[4]: 173.688
```

Bibliografía

- Herbert Schildt. **C++: The Complete Reference**, 4ta Edición. Edit. McGraw-Hill/Osborne, 2003. ISBN 0-07-222680-3.
- Marc Gregoire. **Professional C++.** 5ta. Edición. Edit. Wiley and sons, 2021. ISBN: 978-1-119-69540-0.
- Bjarne Stroustrup. **The C++ Programming Language**, 4ta. Edición. Addison Wesley, 2013. ISBN 978-0-321-56384-2
- Página www.cplusplus.com

Gracias por su atención

DRA.
Sandra Luz
CANCHOLA
MAGDALENO

U.A.Q. Fac. de Informática
Campus Juriquilla

Dra. Sandra Luz Canchola Magdaleno
sandra.canchola@uaq.mx
Cel. 442-1369270

Dra. Reyna Moreno Beltrán
reyna.moreno@uaq.mx