

PROGRAMACIÓN

Guía de estudio 10:

Ingresando en la Matrix

Matrices en C#

Nivel: Secundario - Modalidad Educación Técnico-Profesional.

Ciclo: Segundo ciclo.

Especialidades: Informática, Programación.

Introducción

Tal vez alguno de ustedes conozca qué es una matriz. Para el que no, ¡No se preocupen! Vamos a ver un poco de matrices (en programación, no en matemáticas) en la siguiente guía. Las matrices, en programación, también son conocidos como arreglos multidimensionales (tal vez ya te das una idea de por dónde va la cosa). ¡Veamos cómo se usan!

¿Qué estamos aprendiendo? Qué son las matrices, como crear una matriz en C# y cómo utilizarlas.

Recursos

Video nro 22 del curso:

<https://youtu.be/i5q35hPzhJ0>



Video nro 23 del curso: https://youtu.be/8FmCsRhUFIA	
--	---

Secuencia didáctica

En los siguientes dos videos vas a poder encontrar la información necesaria para poder resolver los ejercicios de matrices planteados a continuación. Es importante tener bien en claro todo lo referente a arreglos para poder entenderlo, ya que las matrices se basan en ellos. Por lo tanto, si no lo hiciste, te recomendamos repasar el capítulo 9.

https://youtu.be/i5q35hPzhJ0	
https://youtu.be/8FmCsRhUFIA	

NOTA: Aunque no se menciona en los videos, se puede crear una matriz de la siguiente manera:

```
int[,] matriz = new int[2, 3]{
    {1, 2, 3},
    {4, 5, 6}
};
```

De esta forma se crea una matriz de 6 posiciones (2X3) con la siguiente estructura:

Indice	0	1	2
0	1	2	3
1	4	5	6

Donde $matriz[0,0]$ es el 1, $matriz[0,1]$ es el 2, $matriz[0,2]$ es el 3, $matriz[1,0]$ es el 5, $matriz[1,1]$ el 5 y $matriz[1,2]$ el 6.

Comenzamos a sospechar que la estructura for nos viene de maravillas, ¿no?

Actividades

Después de mirar el video del link o del QR, te invitamos a que reflexiones con las siguientes preguntas.

No olvides escribir tus respuestas, te facilitarán realizar trabajos prácticos y repasar los temas que hayas estudiado con esta guía.

- 1) ¿Por qué crees que a las matrices (en programación) también se le llaman arreglos multidimensionales?
- 2) Hacer una matriz con números cualesquiera y obtener y mostrar:
 - a) La suma de todas las posiciones de la matriz.
 - b) La suma de las filas.
 - c) La suma de las columnas.

Utilizá la estructura for para recorrerla.

- 3) Hacé una función la cual reciba un número entero. Ese número será el tamaño de una matriz cuadrada que se deberá crear dentro de la función (si el número es 4, entonces se creará una matriz de 4x4). Luego, la matriz deberá ser llenada con números consecutivos (en una matriz 2X2: la posición $[0,0] = 0$, la posición $[0,1] = 2$, la posición $[1,0] = 3$ y la posición $[1,1] = 4$). por último se deberá devolver la matriz completa.

Nota: ver el tema de devolver matrices en "Para seguir aprendiendo".

- 4) Creá un código que llene con "1" la diagonal de izquierda-arriba a derecha-abajo de una matriz cuadrada (de tamaño $n \times n$).
- 5) Hacé un programa en el cual se le pida al usuario ingresar una cantidad de personas a registrar. Por cada persona registrada se le deberá pedir:
 - a) la edad de esa persona.
 - b) si está casada (1=si, 0=no)
 - c) cantidad de años trabajando

Cada fila de una matriz representará a una persona. Luego se le preguntará al usuario los datos de que persona quiere ver.

- a) Si el número es mayor o igual 0 y menor o igual a la cantidad de personas registradas: mostrar los datos de esa persona y volver a preguntar un número.
- b) Si el número es mayor a la cantidad de personas registradas: Mostrar un mensaje de error y volver a preguntar un número.
- c) Si el número es menor a 0: terminar el programa.

Para seguir aprendiendo...

¿Y si queremos "fabricar" una matriz en una función y usarla en el main? Bueno, es totalmente posible. Es tan simple como devolver una matriz (en donde dice *return*) ¡ojo que también hay que decir que tipo de dato devuelve la función!

Aquí te damos un ejemplo básico:

```
static int[ , ] funcion()
{
    int[,] result = new int[2, 2] {
        { 1, 1 },
        { 1, 1 }
    };
    return result;
}
```