Clase 3

lunes, 20 de marzo de 2023 15:32

Matrices int[,] matriz = new int[,]

{{1,2,3,4

- - - -

- - - -

- - - - }};

Vector de vectores

```
int[][] tabla = new int[2][];
tabla[0] = new int[2];
tabla[1] = new int[3];
```

0	0	
0	0	0

=

```
int[][] tabla = { new int[2], new int[3] };
```

```
tabla[1][2] = 8;
```

0	0	
0	0	8

Vector de vectores es más flexible q' una matriz.

Var

Sirve p. declarar variables sin q' lenguaje q' es.

Ej var i = 2 → No se puede declarar sin inicializar.

permite hacer Tipos anónimos → Compilador crea un tipo

```
var x = new { Nombre = "Juan", Edad = 28 };
var y = new { Alto = 12.4, Ancho = 11, Largo = 20 };
Console.WriteLine("Nombre de x: " + x.Nombre);
Console.WriteLine("Ancho de y: " + y.Ancho);
```

Dynamic (puede ser objeto)

puede cambiar durante ejecución

Dynamic no verifica antes de correr → no refleja cambios

→ problemas en ejecución

String de formato compuesto

```
string.Format("Hola {0} y {1}", a, b)
```

```
$"Hola {a} y {b}", a, b
```

```
string marca = "Ford";
int modelo = 2000;
Console.WriteLine($"Es un {marca,7} año {modelo}");
Console.WriteLine($"Es un {"Nissan",7} año {2020}");
```

Alineación derecha
Completa con blancos a izquierda

Pueda dar formato a texto

Boque try: Aquí dentro se controla la ocurrencia de excepciones

Cláusulas catch: Esta sección contiene manejadores para el caso de producirse excepciones en el bloque try

Bloque finally: Contiene código que se ejecuta siempre, se haya producido o no alguna excepción

```
try
{
  statements
}
catch( ... )
{
   statements
}
catch( ... )
{
   statements
}
catch ...
finally
{
   statements
}
```

Esta sección es requerida

Al menos una de estas secciones debe estar presente