

東京' TOKIOTA

Lenguaje C#

Sesión 1





Jose Manuel Corral

Cloud & Development Lead Consultant

jose.corral@Tokiota.com

 • @jmanuelcorral

NET Core



- Framework open-source para Windows, Linux y macOS.
- Está formado por CoreCLR una re-implementación del Runtime del CLR Clásico.
- Portable, Modular, y su principal baza es el Rendimiento.
- Desarrolla en Lenguaje C#

Pros y Cons

- Rendimiento.
 - Portabilidad.
 - Mejoras Testing.
 - Open-source.
 - Comunidad
- Compatibilidad Legacy.
 - Plataforma cambia “demasiado”
 - Providers de terceros aún no están listos.
 - Madurez “Enterprise”

Empezar a trabajar

- Sólo necesitamos el SDK: <https://www.microsoft.com/net/download>
- Editor de Código: <https://code.visualstudio.com/>
- Abrir una línea de comandos: cmd o <https://conemu.github.io/>

```
$ dotnet new console  
$ dotnet restore  
$ dotnet run
```

CLI (Command Line Interface)

- Portable.
- No es necesario un IDE.
- Más ligero.
- Combinar para automatizar y realizar scripts:

<https://github.com/Tokiota/NLayerCoreKickstart>

Lenguaje C# (C Sharp)

- Orientado a objetos
- Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.
- Detalles : https://es.wikipedia.org/wiki/C_Sharp

Tipos

- Enteros: byte, sbyte, short, ushort, int, uint, long, ulong
- Coma flotante: float, double, decimal
- Caracteres: char
- Lógicos: bool
- Cadenas: string

Agrupables en tipos de valor y tipos de referencia.

Orientación a Objetos

Gracias a

- Interfaces
- Clases
- Objetos

Conceptos

- Herencia
- Polimorfismo
- Encapsulación

Características de Lenguajes modernos

- Sentencias de control.
- Interrupción de flujo y Excepciones.
- Operadores.
- Genéricos.
- Métodos.
- Async y await.
- Lambdas.

Ejercicio



Ejercicio

- Pedir un número n al usuario, e imprimir la tabla del 1 al 10, de n , como por ejemplo:

Digite un numero: 5

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

Mucha suerte ;)

Ejercicio

```
$ mkdir multiplicaciones
```

```
$ dotnet new sln -n "multi"
```

```
$ dotnet new console -n "multi.console" -o "src/multi.console"
```

```
$ dotnet sln multi.sln add "src/multi.console/multi.console.csproj"
```

```
$ code .
```

Ejercicio

- Posible Implementación Pachanguera

```
int n;  
Console.Write("Número: ");  
n = int.Parse(Console.ReadLine());  
for(int i=1; i<11; i++)  
{  
    Console.WriteLine(n + " X " + i + " = " + n * i);  
}  
Console.ReadKey();
```

Añadiendo un proyecto de Test

```
$ dotnet new xunit -n "multi.tests" -o "tests/multi.tests"
```

```
$ dotnet sln multi.sln add "tests/multi.tests/multi.tests.csproj"
```

```
$ code .
```

```
$ dotnet watch -p ./tests/multi.tests/ test
```

東京' TOKIOTA