

# Introducción a la programación con C#

---

POO: Programación orientada a objetos



UTN.BA  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

# Un poco de repaso...

- ¿Cuál es la diferencia entre .Net Framework, .Net Core y .NET 5?
- ¿Qué es un objeto? ¿Cuál es la diferencia con una clase?
- ¿Cuáles son los 4 conceptos de POO?
- ¿Porqué son tan importante de aplicarlos? ¿qué beneficios nos aportan?

# Clases

La clase es el “molde” del objeto que queremos modelar.

```
1 public class Auto
2 {
3     private int _cambioActual;
4
5     public string Color { get; set; }
6
7     public Auto()
8     {
9
10    }
11
12    public void Encender()
13    {
14
15    }
16 }
```

# Class 'object'

```
1 using System;
2
3 public class Object
4 {
5     public Object ();
6
7     public virtual bool Equals (object? obj);
8
9     public static bool Equals (object? objA, object? objB);
10
11     public virtual int GetHashCode ();
12
13     public Type GetType ();
14
15     protected object MemberwiseClone ();
16
17     public static bool ReferenceEquals (object? objA, object? objB);
18
19     public virtual string? ToString ();
20 }
```

# ¿Qué vamos a ver hoy?

- La clase **List**
- La clase **Dictionary**
- Cómo iterar una lista
- Cómo aplicar filtros a lista
- Cómo ordenar una lista
- Seleccionar una **Property** dentro de una lista de objetos
- Cómo agrupar objetos dentro de una lista
- Cómo seleccionar objetos distintos de una lista

# Collections

Las encontramos bajo el namespace **System.Collections.Generic**

Contiene interfaces y clases que definen colecciones genéricas, que nos permiten crear colecciones fuertemente tipadas.

[Documentación](#)

# List<T>

```
1 using System.Collections.Generic;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         List<string> lista = new List<string>();
8
9         lista.Add("Elemento 1");
10        lista.Add("Elemento 2");
11        lista.Add("Elemento 3");
12        lista.Add(33);
13    }
14 }
```



Compilation error (line 12, col 13): Argument 1: cannot convert from 'int' to 'string'

# Dictionary<T, U>

```
1 using System.Collections.Generic;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         Dictionary<int, string> lista = new Dictionary<int, string>();
8
9         lista.Add(1, "Elemento 1");
10        lista.Add(2, "Elemento 2");
11        lista.Add(3, "Elemento 3");
12    }
13 }
```



¡¡A trabajar!!

