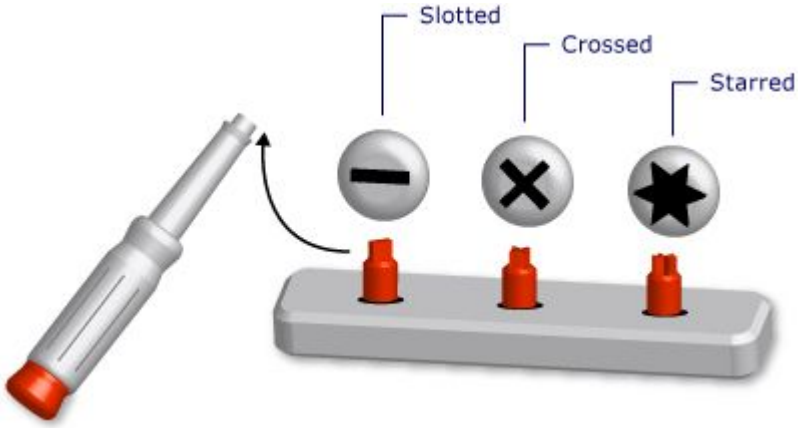
Clase 12
Generics

¿Qué son los GENERICS?

Elementos que se adaptan para realizar la misma funcionalidad para una variedad de tipos de datos.

Sin necesidad de definir una versión para cada tipo.



Clases Genéricas

```
List<string> paises = new List<string>();

paises.Add("España");
paises.Add("Rusia");

paises.RemoveAt("España");

List<Jugador> jugadores = new List<Jugador>();

Jugador j1 = new Jugador(1,"A");
jugadores.Add(j1);
Jugador j2 = new Jugador(3,"B");
jugadores.Add(j2);

jugadores.Remove(j1);
jugadores.Remove(j2);
```

```
public class List<T>
    : IList<T>, System.Collections.IList, IReadOnlyList<T>
{
    private const int _defaultCapacity = 4;

    private T[] _items;
    [ContractPublicPropertyName("Count")]
    private int _size;
    private int _version;

    static readonly T[]  _emptyArray = new T[0];

    public List()
    {
        _items = _emptyArray;
    }

    public void Add(T item)
    {
        if (_size == _items.Length)
        {
            EnsureCapacity(_size + 1);
        }
        _items[_size++] = item;
        _version++;
    }

    //...
}
```

Beneficios de usar Generics

SEGURIDAD DE TIPO

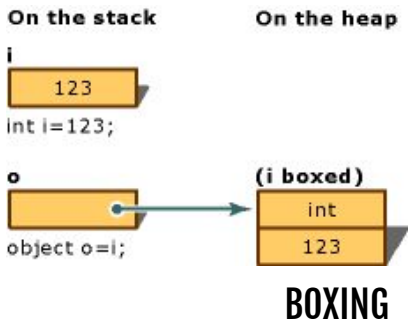
Previene posibles errores

REUTILIZACION DE CODIGO

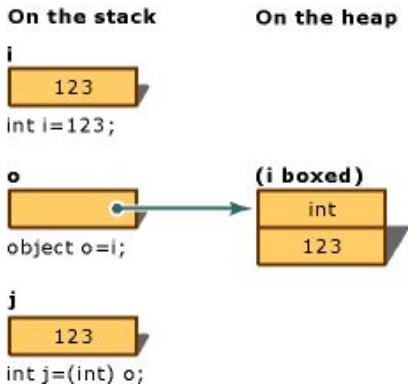
Misma tarea con tipos distintos

Generics VS No-Generics

Eficiencia. Evita boxing y Unboxing



BOXING



UNBOXING

Métodos Genéricos

```
1  Array arrayStr = Array.Empty<string>();
2
3  Array arrayJugadores = Array.Empty<Jugadores>();
4
```

RESTRICCIONES

where T : struct	El argumento de tipo debe ser un tipo de valor.
where T : class	El argumento de tipo debe ser un tipo de referencia.
where T : unmanaged	No puede ser tipo de referencia. Implica struct
where T : new()	El argumento de tipo debe tener un constructor público sin parámetros.
where T : <nombre de la clase base>	El argumento de tipo debe ser de la clase especificada o derivada
where T : <nombre de interfaz>	El argumento de tipo debe implementar la interfaz especificada
where T : U	El argumento de tipo **T** debe ser del tipo **U** o derivados