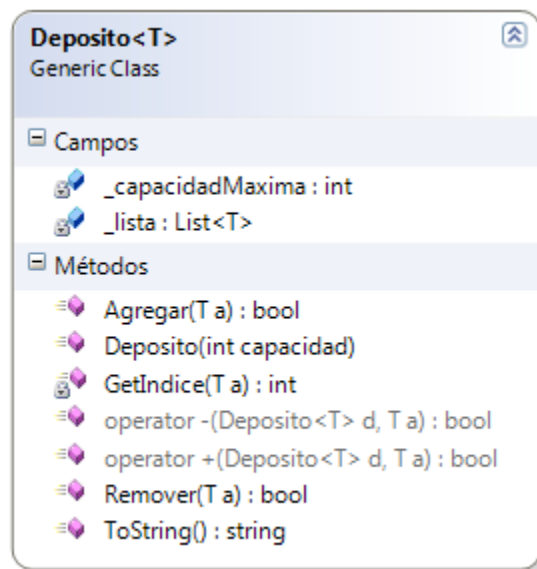


1.- Agregar clase al proyecto *Class library (Entidades)*

Agregar al proyecto de la *parte 2*, la *clase genérica*, detallada en el siguiente diagrama de clase.



**Deposito<T>:** Es una clase diseñada para poder almacenar (en una lista genérica de tipo *T*) una cierta cantidad de *objetos T*. Dicha clase tiene la funcionalidad de agregar, remover y listar *objetos T*.

- Sobrecarga operador adición:
  - Retorna **true**, si pudo agregar el *objeto T* al depósito de *T*, **false**, caso contrario.

Para poder agregar un *objeto T* a la lista genérica hay que tener en cuenta que la capacidad máxima del depósito no puede ser superada.
- Método privado y de instancia `GetIndice`:
  - Retorna el valor del índice en el cual se encuentra el *objeto T* pasada como parámetro. Se debe recorrer la lista genérica y retornar el índice de la primera ocurrencia, -1 si no se encuentra en la lista.
- Sobrecarga operador sustracción:
  - Retorna **true**, si pudo remover el *objeto T* del depósito de *T*, **false**, caso contrario.

Para poder remover un *objeto T* a la lista genérica hay que invocar al método *GetIndice*.
- Método de instancia `Agregar`:
  - Está asociado al operador `+`.
- Método de instancia `Remover`:

- Está asociado al operador -.
- Polimorfismo:
  - ToString. Retorna una cadena conteniendo la información del depósito *T* (capacidad y todo el detalle de los *objetos T* que contiene).

## 2.- Agregar al proyecto *Console Application (TestEntidades)*

Comentar las líneas del main de la **parte 2**, y reemplazarlas por las siguientes:

```
static void Main(string[] args)
{
    Cocina c1 = new Cocina(111, 12300, false);
    Cocina c2 = new Cocina(112, 15000, true);
    Cocina c3 = new Cocina(113, 5600, false);

    Auto a1 = new Auto("Rojo", "Ferrari");
    Auto a2 = new Auto("Amarillo", "Porsche");
    Auto a3 = new Auto("Negro", "BMW");
    Auto a4 = new Auto("Verde", "Ford");

    Deposito<Cocina> dc = new Deposito<Cocina>(5);
    Deposito<Auto> da = new Deposito<Auto>(3);

    dc.Agregar(c1);
    dc.Agregar(c2);
    if (!(dc + c3))
    {
        Console.WriteLine("No se pudo agregar el item!!!");
    }

    if ((da + a1))
    {
        Console.WriteLine("Se ha agregado el item!!!");
    }

    da.Agregar(a2);
    da.Agregar(a3);

    if (!da.Agregar(a4))
    {
        Console.WriteLine("No se pudo agregar el item!!!");
    }

    Console.WriteLine(dc);
    Console.WriteLine(da);

    dc.Remove(c2);
    if (!(dc - c2))
    {
        Console.WriteLine("No se pudo remover el item!!!");
    }
}
```

```

        da.Remove(a2);
        if (!(da - a4))
        {
            Console.WriteLine("No se pudo remover el item!!!");
        }

        Console.WriteLine(dc);
        Console.WriteLine(da);

        Console.ReadLine();
    }

```

El resultado es:

```

Capacidad máxima: 5
Listado de Cocina:
Código: 111 - Precio: 12300 - Es industrial? False
Código: 112 - Precio: 15000 - Es industrial? True
Código: 113 - Precio: 5600 - Es industrial? False

Capacidad máxima: 3
Listado de Auto:
Marca: Ferrari - Color: Rojo
Marca: Porsche - Color: Amarillo
Marca: BMW - Color: Negro

No se pudo remover el item!!!
No se pudo remover el item!!!
Capacidad máxima: 5
Listado de Cocina:
Código: 111 - Precio: 12300 - Es industrial? False
Código: 113 - Precio: 5600 - Es industrial? False

Capacidad máxima: 3
Listado de Auto:
Marca: Ferrari - Color: Rojo
Marca: BMW - Color: Negro

```