

## Problema 1 Lista mínima

La interfaz `ProblemaLista.IListaMinima<T>`, compilada en la dll que se le suministra y cuyo código listamos a continuación, representa una lista con solo la funcionalidad indispensable.

```
namespace ProblemaLista
{
    public interface IListaMinima<T> : System.Collections.IEnumerable
    {
        T this[int pos] { get; }
        void Insert(T newElem, int pos);
        void Sort(IComparer<T> comparer);
        int Count { get; }
    }
}
```

A continuación explicamos el comportamiento deseado para cada miembro de `IListaMinima`:

Miembro	Comportamiento
<code>T this[int pos] { get; }</code>	Indizador ( <i>indexer</i> ) que permite leer el elemento de la lista que se encuentra en la posición indicada por <code>pos</code> . Si el valor de <code>pos</code> no está entre <code>0</code> y <code>Count - 1</code> , se debe lanzar una excepción de tipo <code>IndexOutOfRangeException</code> .
<code>void Insert(T newElem, int pos);</code>	Inserta el elemento <code>newElem</code> en la posición <code>pos</code> de la lista. El valor del parámetro <code>pos</code> debe estar entre <code>0</code> y <code>Count</code> (incluyendo el caso <code>Count == 0</code> ), de lo contrario, se debe lanzar una excepción de tipo <code>IndexOutOfRangeException</code> . Tras la inserción, la posición de los elementos que se encontraban a partir del índice <code>pos</code> se incrementa en 1. Note que entonces si la lista está vacía se puede insertar en la posición <code>0</code> y que en general siempre se puede insertar en la posición <code>Count</code> (que sería poner al final de la lista)
<code>void Sort(IComparer&lt;T&gt; comparer);</code>	Ordena la lista de menor a mayor de acuerdo con el criterio de comparación establecido por el parámetro <code>comparer</code> .

<code>int Count { get; }</code>	Devuelve la cantidad actual de elementos en la lista.
---------------------------------	-------------------------------------------------------

Debe implementar una biblioteca de clases de nombre `ProblemaLista.dll` (le recomendamos nombrar al proyecto `ProblemaLista`, sin la terminación `dll`) donde defina una clase `ProblemaLista.ListaMinima` que implemente `IListaMinima` de acuerdo con la funcionalidad antes explicada para sus miembros. El código de su clase debe ser semejante al siguiente:

```
namespace ProblemaLista
{
    public class ListaMinima<T> : IListaMinima<T>
    {
        //Aquí van los miembros de la clase
    }
}
```

Note que como `IListaMinima<T>` hereda de `IEnumerable`, la clase que implemente `IListaMinima<T>` tendrá también que implementar el método `GetEnumerator` que deberá dar un mecanismo de iteración sobre los valores en la lista en el orden en que se encuentran en ella.

## Aclaraciones

- No se podrá utilizar *arrays*, ni ninguna clase de los *namespaces* `System.Collections` ni `System.Collections.Generic` ni ninguna otra de la plataforma .NET que represente una colección o estructura de datos cualquiera. Es decir, todo debe programarlo usted con enlaces. Note que sí puede usar las interfaces `IEnumerable`, `IEnumerator`, etc porque no son clases.
- NOTA Una violación de lo permitido le anula su respuesta por lo que si tiene alguna duda sobre si la aclaración anterior se aplica a determinada clase, **no se arriesgue, consulte a los profesores.**