

Clase 14

Viernes, 6 Octubre 2023

[Indice](#)

Funciones para manejar listas y arrays

- **FUNCIÓN:** Crear una función que calcule la media de los valores de un array

```
,
public static double GetMedian(double[] array)
{
    if (array == null || array.Length < 0)
        return null;

    double aux = 0.0;

    for(int i = 0; i<array.Length; i++)
    {
        aux += array[i];
    }

    return aux / array.Length;
},
```

- Es **MUY IMPORTANTE** validar los parámetros de entrada de la función
- **FUNCION:** Crear una función que calcule la media de los valores de un array, solo si superan el valor dado llamado 'Threshold'

```
,
public static double GetThresholdMedian(double[] array, double Threshold)
{
    if (array == null || array.Length < 0)
        return null;

    double aux = 0.0;
    int count = 0;

    for(int i = 0; i<array.Length; i++)
    {
        if(array[i] > Threshold)
        {
            aux += array[i];
            count++;
        }
    }
    return aux / count;
},
```

- **FUNCION:** Crear una función que devuelva el número de veces que se repite el número de mayor valor.

```
,
public static int GetMaxNumberRepeated(List<int> list)
{
    if (list == null || list.Count < 0)
        return null;

    int result = 0;
    int maxNumber;
    maxNumber = GetMajor(list);

    for (int i = 0; i<list.Count; i++)
    {
        if (list[i] == maxNumber)
            result++;
    }
}
```

```

        return result;
    }
}

```

- El proceso para construir una función consiste en **SEPARAR EL PROBLEMA en PEQUEÑOS FRAGMENTOS FUNCIONALES**
- **FUNCION:** Crear una función que devuelva la lista que se pasa por parametro en orden inverso.

```

,
public static List<int> GetReverseList(List<int> list)
{
    if (list == null || list.Count < 0)
        return null;

    List<int> result = new List<int>();

    for (int i = list.Count; i >= 0; i--)
    {
        result.Add(list[i]);
    }
    return result;
}
,

```

- **FUNCION:** Crear una función que imprima los valores de una lista de enteros en pantalla

```

,
public static void PrintList(List<int> list)
{
    if (list == null || list.Count < 0)
        return null;

    for(int i = 0; i < list.Count; i++)
    {
        Console.WriteLine(list[i]);
    }
}
,

```

- **FUNCION:** Crear una función que imprima los valores de un array de enteros en pantalla

```

,
public static void PrintList(int[] array)
{
    if (array == null || array.Length < 0)
        return null;

    for(int i = 0; i < array.Length; i++)
    {
        Console.WriteLine(array[i]);
    }
}
,

```

- **FUNCION:** Crear una función que añada un valor dado por parametros al final de un array de enteros.

```

,
public static void AddValueToArray(int[] array, int number)
{
    if(array == null || array.Length == 0)
        return null;

    int[] result = new int[array.Length + 1];

    for(int i = 0; i < array.Length; i++)
    {

```

```
        result[i] = array[i];  
    }  
    result[array.Length] = number;  
}  
.
```