

# Clase 13

Jueves, 5 Octubre 2023

---

[Indice](#)

## Binary Search

- **FUNCIÓN:** Consiste en una función que se le pasa una lista o arrays de enteros **ordenada** y te devuelve 'true or false' si el número se encuentra en dicha lista
- Funciona con Listas o Arrays.
- Solo funciona **si la lista está ordenada**.
- El algoritmo es el siguiente:
  1. Busco el punto medio de la lista.
  2. Comparo el valor dado con el punto medio. Si son iguales devuelvo 'true'.
  3. Si el número es *mayor*: Desplazo el **valor mínimo** una posición por delante del punto medio.  
Si el número es *menor*: Desplazo el **valor máximo** una posición por delante del punto medio.  
Vuelvo a calcular el punto medio.
  4. Este algoritmo se repite hasta que encuentro el parámetro dado o el **valor mínimo** es igual o mayor que el **valor máximo**.
- **EJEMPLO**

```
1.- Lista dada: [0,2,4,6,8]
Parametro dado: 6 Es el número que queremos encontrar
Posiciones: [0,1,2,3,4]
Posición media: (0 + 4) / 2 = 2
Valor en posición [2] = 4
```

```
2.- Comparamos valores: 6 = 4? false
```

```
3.- Desplazamos el valor minimo: Posición media + 1 = 3
```

```
4.- Repetimos el algoritmo.
```

```
1'.- Calculamos de nuevo el valor medio: (3 + 4) / 2 = 3
```

```
2'.- Comparamos valores: 6 = 6? 'true'
```

```
El número que buscábamos sí está en la lista.
```

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

    int minPosition = 0;
    int maxPosition = list.Count - 1;
    int midPosition;

    while(minPosition ≤ maxPosition)
    {
        midPosition = (minPosition + maxPosition) / 2;

        if(midPosition == number)
            return true;
        if(number > midPosition)
            minPosition = midPosition + 1;
        else
            maxPosition = midPosition - 1;
    }
    return false;
}
,
```