Búsqueda Binaria

¿Qué es la búsqueda binaria?

- Algoritmo eficiente para buscar en **listas ordenadas**
- Divide el espacio de búsqueda a la mitad en cada iteración
- Complejidad: **O(log n)**

Requisitos

- La lista debe estar **ordenada**
- Saber el valor que queremos encontrar (el objetivo)

Ejemplo simple

```
datos = [1, 3, 5, 7, 9, 11, 13]
objetivo = 9
```

Queremos saber si `9` está en la lista y en qué posición.

Lógica del algoritmo

```
low = 0
high = len(datos) - 1

while low <= high:
    mid = (low + high) // 2
    if datos[mid] == objetivo:
        return mid
    elif datos[mid] < objetivo:
        low = mid + 1
    else:
        high = mid - 1
```

Paso a paso (buscar 9)

```
Lista:     [1, 3, 5, 7, 9, 11, 13]
Índices:    0  1  2  3  4   5   6

1. mid = (0 + 6) // 2 = 3 → datos[3] = 7 → menor → derecha
2. mid = (4 + 6) // 2 = 5 → datos[5] = 11 → mayor → izquierda
3. mid = (4 + 4) // 2 = 4 → datos[4] = 9 → ✅ ¡Encontrado!
```

Código completo

```python
def busqueda_binaria(lista, objetivo):
    low, high = 0, len(lista) - 1
    while low <= high:
        mid = (low + high) // 2
        if lista[mid] == objetivo:
            return mid
        elif lista[mid] < objetivo:
            low = mid + 1
        else:
            high = mid - 1
    return -1  #o encontrado
```

Resumen

- Solo funciona en **listas ordenadas**
- Divide el rango de búsqueda a la mitad
- Muy eficiente: **O(log n)**
- Ideal para grandes volúmenes de datos