



# Algoritmo de Búsqueda Lineal

Recorriendo elemento por elemento hasta encontrar el objetivo

# Definición y funcionamiento

- Recorrer elementos uno por uno hasta hallar el objetivo
- Iniciar en índice 0 y comparar con objetivo
- Si coincide, devolver posición; si no, avanzar



# Complejidad del algoritmo

 Mejor caso

**O(1)**

Cuando el primer elemento coincide

 Peor caso

**O(N)**

Cuando el objetivo está al final o no está

 Caso promedio

**O(N)**

Asumiendo distribución uniforme del objetivo

 Complejidad espacial

**O(1)** Usa solo contadores/índices constantes

# Ventajas y desventajas



## Ventajas

- + Funciona con datos no ordenados
- + Implementación simple
- + No requiere memoria adicional



## Desventajas

- Ineficiente para conjuntos grandes
- Más comparaciones en promedio
- Escala pobre frente a métodos avanzados

# Casos de uso



## Listas pequeñas

Donde el costo de ordenar no se justifica



## Datos cambiantes

Listas no ordenadas o datos que cambian frecuentemente



## Listas enlazadas

Donde el acceso aleatorio no está disponible



## Situaciones adicionales

- Búsquedas ocasionales con baja frecuencia de consultas

# Comparación con otros algoritmos



## Búsqueda Lineal



## Búsqueda Binaria

### ⌚ Complejidad

$O(N)$

$O(\log N)$

### ❖ Ordenamiento



No requiere



Requiere ( $O(N \log N)$ )

### ⌚ Uso ideal

Datos pequeños o  
volátiles

Colecciones grandes con  
muchas consultas