

# Comparativa: Búsqueda Lineal vs Búsqueda Binaria

Aspecto	Búsqueda Lineal	Búsqueda Binaria
Definición	Recorre cada elemento del arreglo uno por uno.	Divide el arreglo ordenado y elimina la mitad en cada paso.
Complejidad	O(n)	O(log n)
Condiciones	Funciona en cualquier arreglo (ordenado o no).	Requiere un arreglo ordenado.
Implementación	Simple y fácil de implementar.	Más compleja de implementar que la búsqueda lineal.
Mejor Caso	O(1) (si el elemento buscado es el primero).	O(1) (si el elemento está en el medio).
Peor Caso	O(n) (si el elemento buscado es el último o no está).	O(log n) (si el elemento está al final o no está).
Uso Típico	Arreglos pequeños o no ordenados.	Arreglos grandes y ordenados.

## Ejemplo de Código en C

### Búsqueda Lineal

```
#include <stdio.h>

int busquedaLineal(int arr[], int n, int x) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == x)
            return i;
    }
    return -1;
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 4;

    int resultado = busquedaLineal(arr, n, x);
    printf("Búsqueda Lineal: %s en el índice %d\n", resultado != -1 ?
"Encontrado" : "No encontrado", resultado);

    return 0;
}
```

## Búsqueda Binaria

```
#include <stdio.h>

int busquedaBinaria(int arr[], int izquierda, int derecha, int x) {
    while (izquierda <= derecha) {
        int medio = izquierda + (derecha - izquierda) / 2;

        if (arr[medio] == x)
            return medio;
        if (arr[medio] < x)
            izquierda = medio + 1;
        else
            derecha = medio - 1;
    }
    return -1;
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 4;

    int resultado = busquedaBinaria(arr, 0, n - 1, x);
    printf("Búsqueda Binaria: %s en el índice %d\n", resultado != -1 ?
    "Encontrado" : "No encontrado", resultado);

    return 0;
}
```