

SilverTech

+talento +tecnología +empleabilidad

Socio estratégico

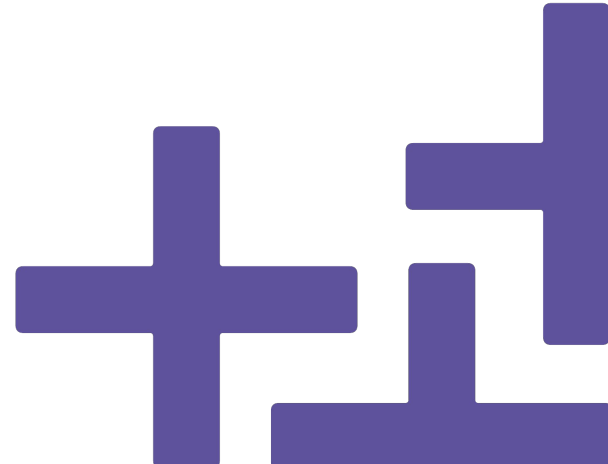


Impulsan



Cobol – Clase 19

Algoritmo de búsqueda



Reglas de la clase



Micrófonos apagados



Consultas al final de la clase



Consultas por chat



Cronograma

Primera Parte

18:30
a
19:25

Break

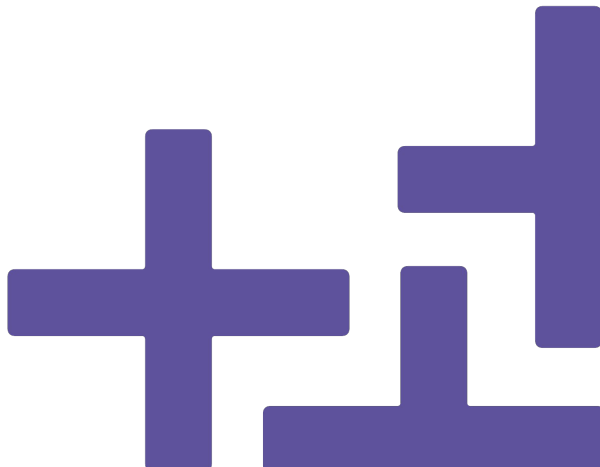
19:25
a
19:35

Segunda Parte

19:35
a
20:30

¿Qué veremos hoy?

- + Algoritmo de búsqueda secuencial
- + Algoritmo de búsqueda Binaria
- + Ejercicios



Métodos de Búsqueda



Los procesos de búsqueda involucran recorrer un arreglo completo con el fin de encontrar un elemento.



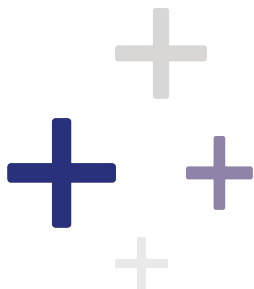
Métodos de Búsqueda

Para buscar el menor o mayor elemento de un arreglo, podemos usar la estrategia, de suponer que el primero o el último es el menor (mayor), para luego ir comparando con cada uno de los elementos, e ir actualizando el menor (mayor). A esto se le llama **Búsqueda Lineal**.

Métodos de búsqueda

01.

Búsqueda
Secuencial



02.

Búsqueda
Binaria





Búsqueda Secuencial

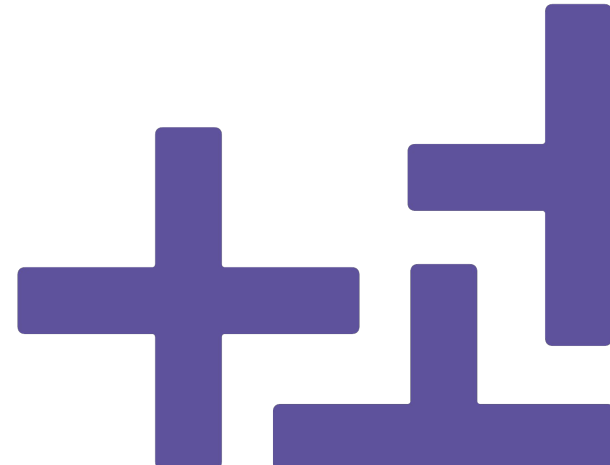
- + Consiste en recorrer un arreglo de forma secuencial e incremental, y comparando cada elemento contra el elemento buscado, hasta encontrarlo.
- + En este caso de búsqueda se asume que los elementos están en cualquier orden. En el peor de los casos deben hacerse n operaciones de comparación. Una búsqueda más eficiente puede hacerse sobre un arreglo **ordenado**

A	B	C	D	E	F	G
0	1	2	3	4	5	6

D
3

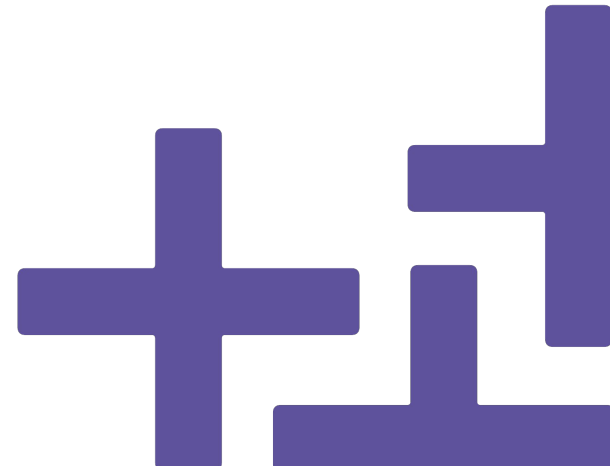
Búsqueda secuencial Menor

```
MOVE WS-VEC-A(0) TO WS-MENOR
PERFORM VARYING WS-I FROM 1 BY 1 UNTIL WS-I > WS-CANT
  IF WS-VEC-A(WS-I) < WS-MENOR THEN
    MOVE WS-VEC-A(WS-I) TO WS-MENOR
  END-IF
END-PERFORM
```



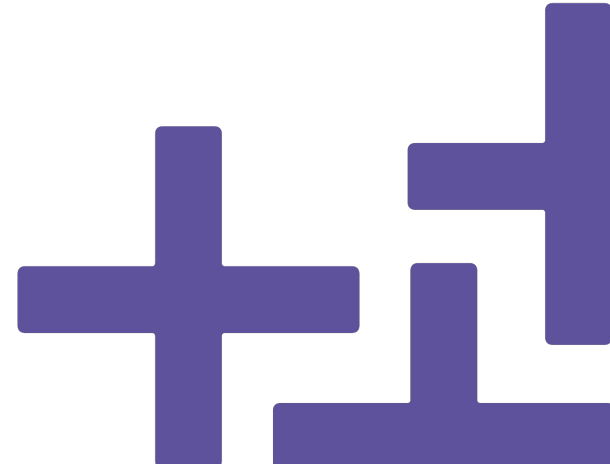
Búsqueda secuencial Menor

```
MOVE WS-VEC-A(0) TO WS-MAYOR
PERFORM VARYING WS-I FROM 1 BY 1 UNTIL WS-I > WS-CANT
  IF WS-VEC-A(WS-I) > WS-MAYOR THEN
    MOVE WS-VEC-A(WS-I) TO WS-MAYOR
  END-IF
END-PERFORM
```



Búsqueda secuencial Elemento

```
SET SW-ENCONTRAR-NO TO TRUE
PERFORM VARYING WS-I FROM 1 BY 1 UNTIL WS-I > WS-CANT
    OR SW-ENCONTRAR-SI
    IF WS-VEC-A (WS-I) EQUAL WS-ELEMENTO THEN
        SET SW-ENCONTRAR-SI TO TRUE
    END-IF
END-PERFORM
```



Instrucción SEARCH

01 WS-TB.

02 WS-VECTOR PIC 9(02) OCCURS 10 TIMES
INDEXED BY WS-I.

SEARCH WS-VECTOR

AT END

SET SW-ENCONTRAR-NO TO TRUE

WHERE WS-VECTOR(WS-I) = WS-ENCONTRADO

SET SW-ENCONTRAR-SI TO TRUE

END-SEARCH

Eficiencia y Complejidad



Mejor CASO

El elemento buscado se encuentra en la primer posición

Cantidad de interacciones : 1



Peor CASO

El elemento buscado se encuentra en la última posición

Cantidad de interacciones : N



CASO Promedio

El elemento buscado se encuentra en la mitad de arreglo.

Cantidad de interacciones : $N/2$



Búsqueda Binaria

- + La Búsqueda Binaria, compara si el valor buscado está en la mitad superior o inferior. En la que esté, subdivido nuevamente, y así sucesivamente hasta encontrar el valor.
- + Como precondition antes de realizar una búsqueda binaria, el arreglo deberá estar ordenado.
- + Optimiza el tiempo de búsqueda.

SilverTech
+talento +tecnología +empleabilidad

18

Tamaño arreglo: 7
Primer indice: 0
Mitad: 3
Ultimo indice: 6

18 \geq 8?

Tamaño arreglo: 3
Primer indice: 4
Mitad: 5
Ultimo indice: 6

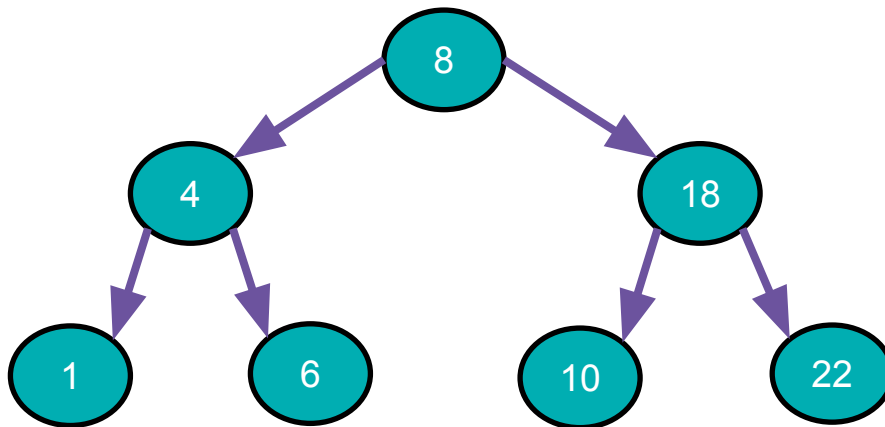
18 >= 18?



Búsqueda Binaria

Elemento a buscar

18





CÓDIGO



```
MOVE 1      TO WS-COMIENZO
MOVE WS-TAM TO WS-FIN
SET SW-ENCENTRO-NO TO TRUE

PERFORM UNTIL WS-COMIENZO > WS-FIN
    OR SW-ENCENTRO-SI TO TRUE

    ADD WS-COMIENZO TO WS-FIN GIVING WS-MITAD
    DIVIDE WS-MITAD BY 2      GIVING WS-MITAD

    IF WS-VECTOR(WS-MITAD) = WS-ELEMENTO THEN
        SET SW-ENCENTRO-SI TO TRUE
        IF WS-VECTOR(WS-MITAD) > WS-ELEMENTO THEN
            MOVE WS-MITAD TO WS-FIN
        ELSE
            MOVE WS-MITAD TO WS-COMIENZO
        END-IF
    END-IF
END-PERFORM.

IF WS-VECTOR(WS-FIN) EQUAL WS-ELEMENTO THEN
    DISPLAY 'Elemento encontrado'
ELSE
    DISPLAY 'Elemento no encontrado'
END-IF.
```

Instrucción SEARCH ALL

01 WS-TB.

02 WS-VECTOR PIC 9(02) OCCURS 10 TIMES
ASCENDING WS-VECTOR
INDEXED BY WS-I.

```
SEARCH ALL WS-VECTOR  
AT END  
    SET SW-ENCONTRAR-NO TO TRUE  
WHERE  WS-VECTOR(WS-I) = WS-ENCONTRADO  
    SET SW-ENCONTRAR-SI TO TRUE  
END-SEARCH
```

Eficiencia y Complejidad

+ Mejor CASO

El elemento buscado se encuentra en el centro

Cantidad de interacciones : 1

+ Peor CASO

El elemento buscado se encuentra en la primera o última posición

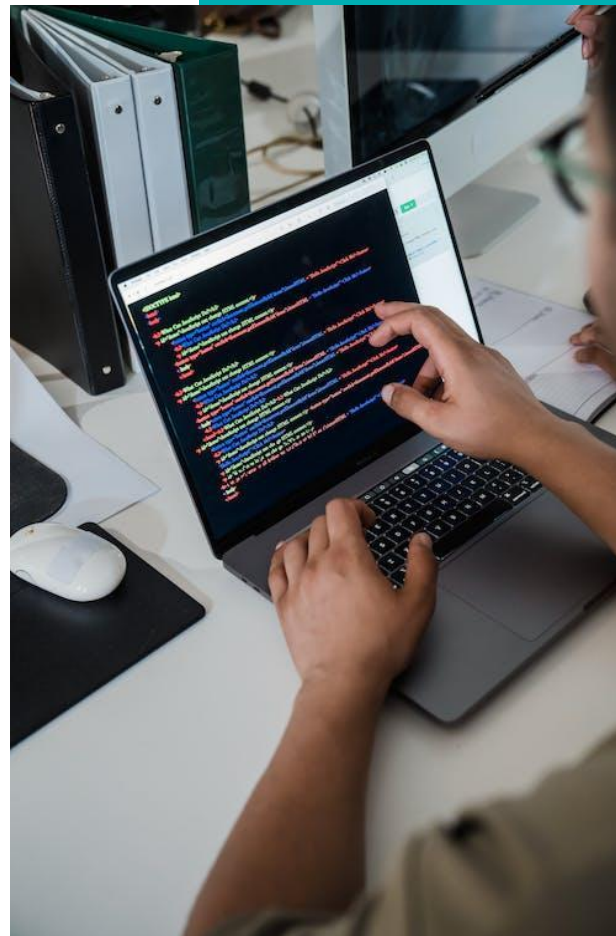
Cantidad de interacciones : $\log_2(n)$

+ CASO Promedio

Cantidad de interacciones : $\log_2(n/2)$

Ejercicio 1 a

- Dado un programa que ordenar el archivo de empleados por número de legajo.
- Se necesita desarrollar un nuevo programa de búsqueda para buscar por Número de Legajo del empleado.
- Analizar qué método de búsqueda conviene aplicar en este problema.



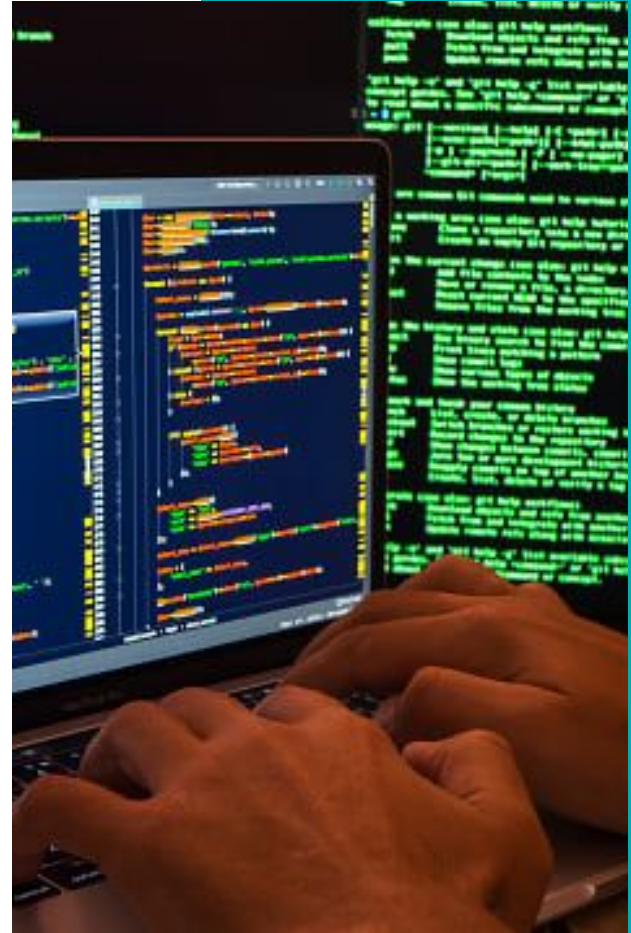
RESOLUCIÓN 1a





Ejercicio 1 a

- Agregar una función, para realizar una búsqueda por código de estado.
- El programa deberá devolver todos los empleados que se encuentren en ese estado.
- Analizar qué método de búsqueda conviene aplicar en este problema.



RESOLUCIÓN 1b





Comunicación

Foro de consultas TEC:

<https://campus.soysilverttech.org>

Mails de consulta TEC:

consultasCOBOL@soysilverttech.org



GRACIAS

Silver**Tech**
+talento +tecnología +empleabilidad



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

SilverTech

+talento +tecnología +empleabilidad