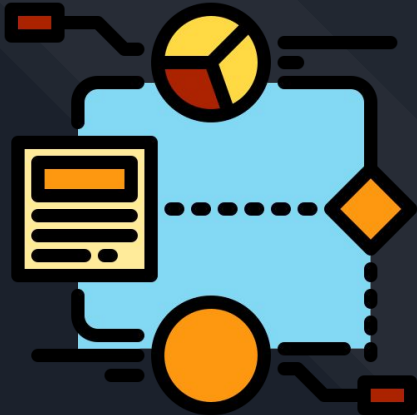


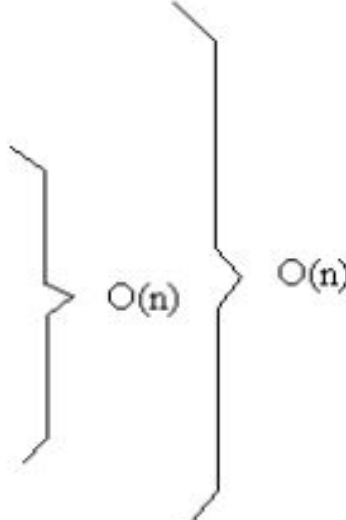
# Análisis de algoritmos / búsqueda y ordenamiento



Alumnos: Federico Almuina y Milton Alvarez

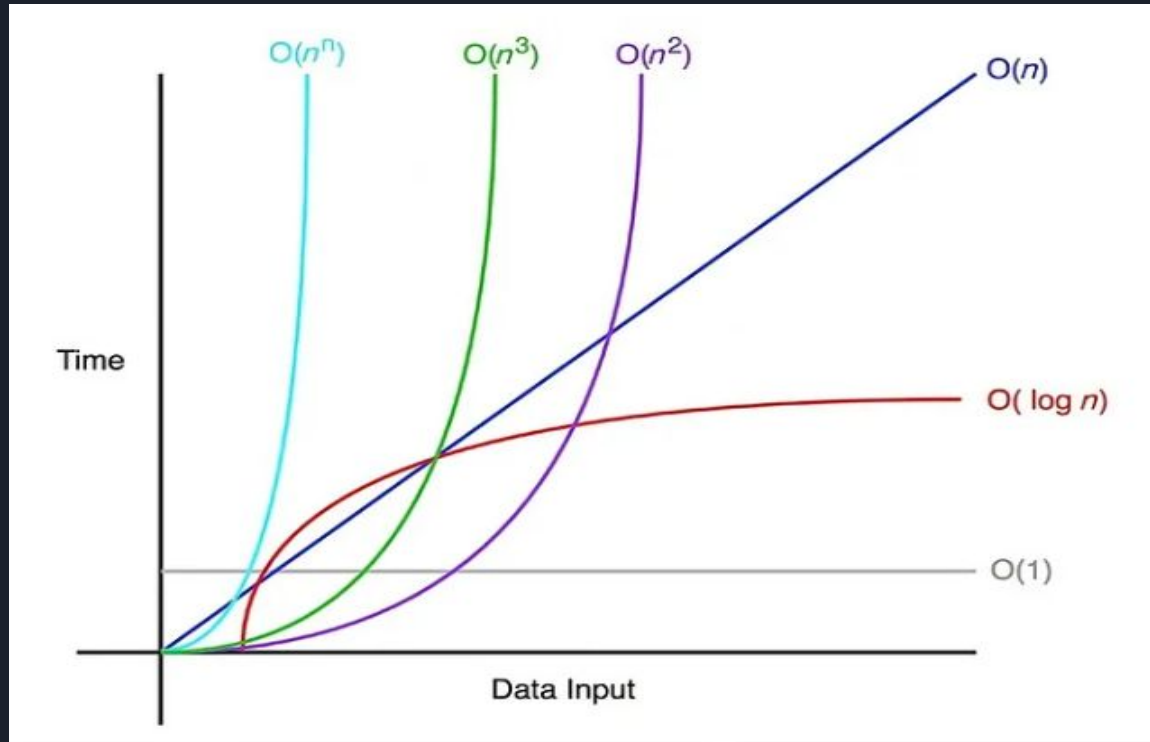
# Análisis de algoritmos

```
Desde  $i \leftarrow 1$  hasta  $n-1$  hacer
   $\min j \leftarrow i$   $O(1)$ 
   $\min x \leftarrow T[i]$   $O(1)$ 
  desde  $j \leftarrow i+1$  hasta  $n$  hacer
    si  $T[j] < \min x$  entonces
       $\min j \leftarrow j$   $O(1)$ 
       $\min x \leftarrow x$   $O(1)$ 
    fin si
     $T[\min j] \leftarrow T[i]$   $O(1)$ 
     $T[i] \leftarrow \min x$   $O(1)$ 
  fin desde
fin Desde
```

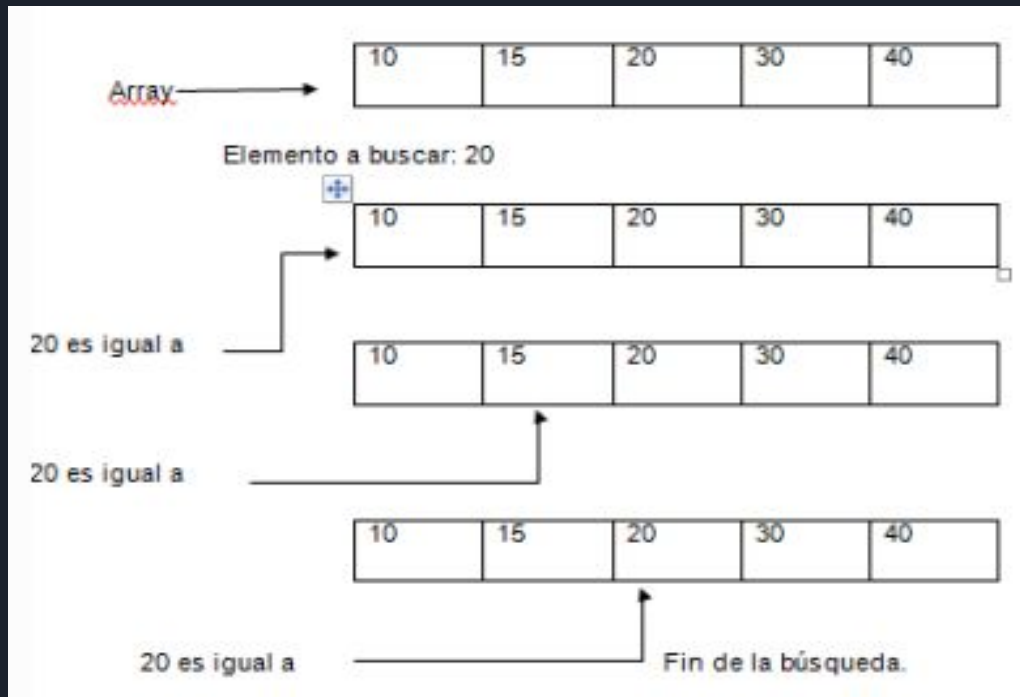


$O(n)$   $O(n)$

# Analisis teorico: Big O

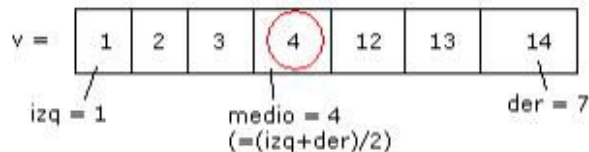


# Búsqueda Lineal

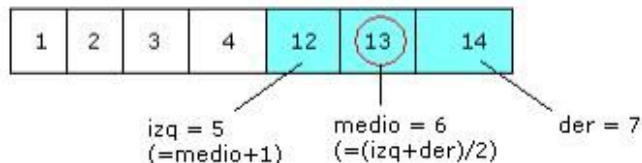


# Búsqueda Binaria

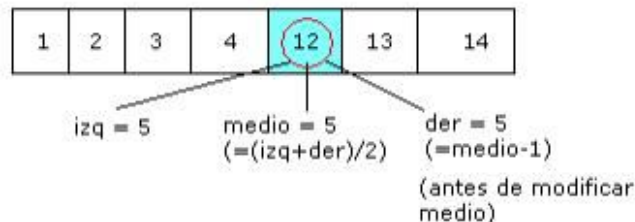
Elemento buscado: 12



$12 > v[medio] = 4$   
Cogemos el  
subvector derecho



$12 < v[medio] = 13$   
Cogemos el subvector  
izquierdo

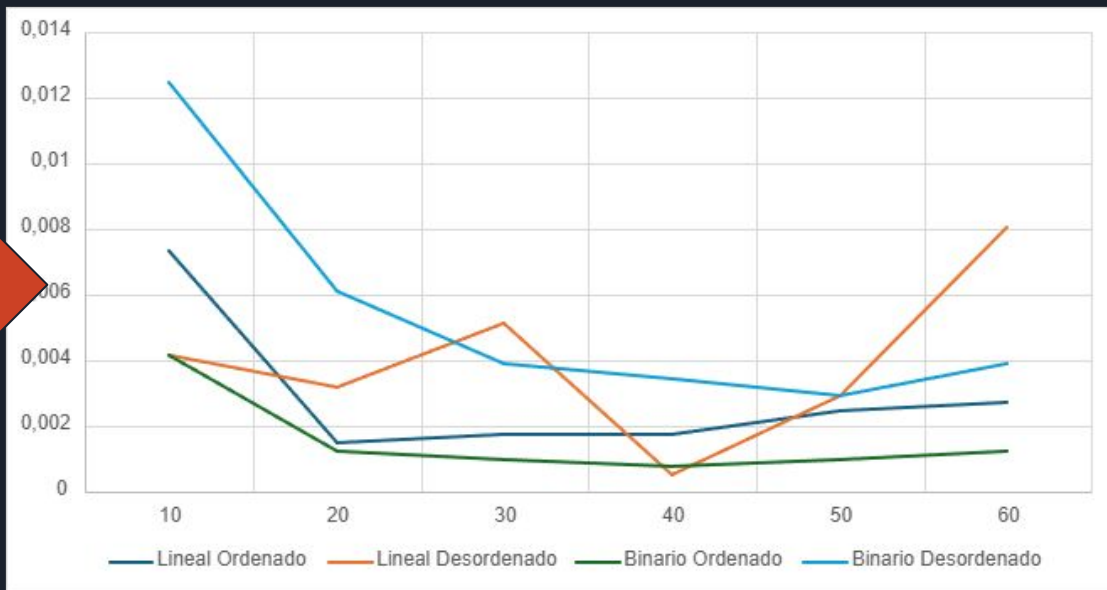


$12 = v[medio] = 12$   
ENCONTRADO

# Análisis empírico

```
projecto.py > ...
124 lineal_desordenada = []
125 binaria_ordenada = []
126 binaria_desordenada = []
127
128 # Clasificamos los resultados
129 for r in resultados:
130     if r["metodo"] == "lineal" and r["orden"] == "ordenada":
131         lineal_ordenada.append(r)
132     elif r["metodo"] == "lineal" and r["orden"] == "desordenada":
133         lineal_desordenada.append(r)
134     elif r["metodo"] == "binaria" and r["orden"] == "ordenada":
135         binaria_ordenada.append(r)
136     elif r["metodo"] == "binaria" and r["orden"] == "desordenada":
137         binaria_desordenada.append(r)
138
139 print("- *10, \"BUSQUEDA LINEAL EN LISTA ORDENADA\", \"-\" * 10)
140 for i in lineal_ordenada:
141     print(i)
142
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Hilton\Desktop\Rpyecto.py> python3 & C:\Users\Hilton\AppData\Local\Programs\Python\
python3\python.exe p1/proyecto.py
Ingrese el numero que desea buscar: 10
variables a escanear: [10, 20, 30, 40, 50, 60]
----- BUSQUEDA LINEAL EN LISTA ORDENADA -----
{'numero_buscado': 10, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.010009765625}
{'numero_buscado': 20, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.002197265625}
{'numero_buscado': 30, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.001708984375}
{'numero_buscado': 40, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.002685546875}
{'numero_buscado': 50, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.002685546875}
{'numero_buscado': 60, 'metodo': 'lineal', 'orden': 'ordenada', 'tiempo': 0.0029296875}
----- BUSQUEDA LINEAL EN LISTA DESORDENADA -----
```





# Conclusión del trabajo

- La búsqueda lineal es simple y funciona sobre cualquier tipo de lista, pero puede ser ineficiente con grandes volúmenes de datos.
- La búsqueda binaria es mucho más rápida, pero solo se puede aplicar sobre listas ordenadas.
- El rendimiento de un algoritmo depende del contexto y la estructura de los datos.
- Analizar y elegir el método adecuado permite optimizar tiempo y recursos.