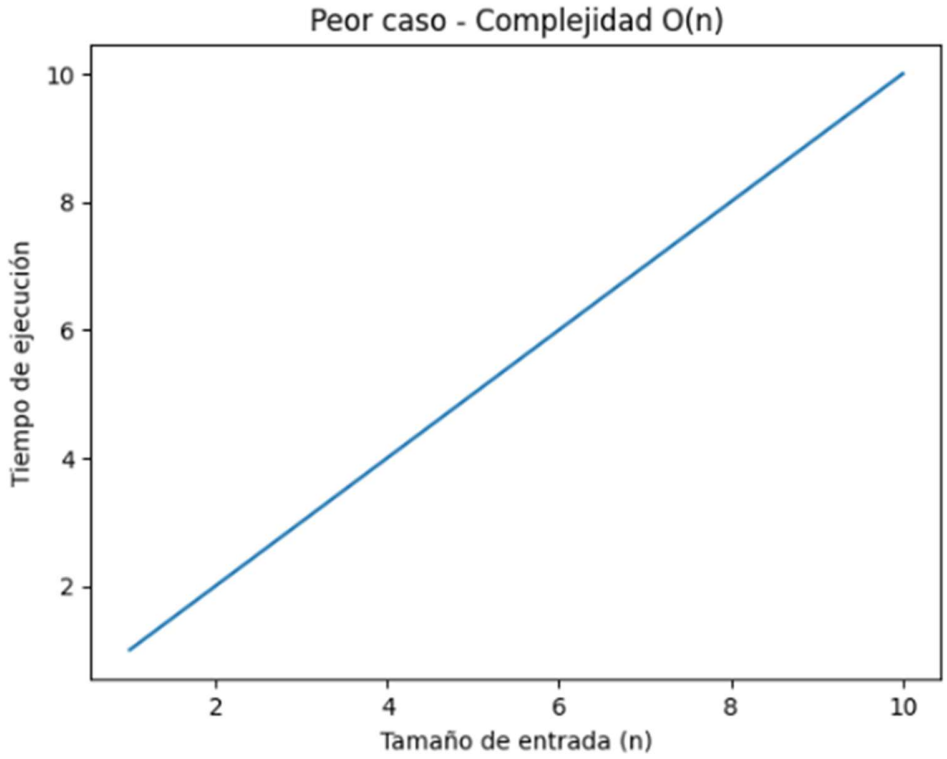


Problema de las diapositivas semana -3

1. Dado un arreglo N entero, cuyos valores van en decreciente y luego incrementado, encontrar el número menor del arreglo.

```
FUNCION buscarMinimo(arreglo, inicio, fin)
  SI inicio == fin
    RETORNAR arreglo[inicio]
  medio ← (inicio + fin) / 2

  SI arreglo[medio] < arreglo[medio - 1] Y arreglo[medio] < arreglo[medio + 1]
    RETORNAR arreglo[medio]

  SI arreglo[medio] > arreglo[medio - 1]
    RETORNAR buscarMinimo(arreglo, inicio, medio - 1)
```

SINO

RETORNAR buscarMinimo(arreglo, medio + 1, fin)

FIN FUNCION

2. Numero faltante en el arreglo N-1

```
FUNCION numeroFaltante(arreglo, N)
  sumaEsperada ← N * (N + 1) / 2
  sumaReal ← 0

  PARA cada elemento EN arreglo
    sumaReal ← sumaReal + elemento

  RETORNAR sumaEsperada - sumaReal
FIN FUNCION
```

3. Calcular a^n divide y conquista

```
FUNCION potencia(a, n)
  SI n == 0
    RETORNAR 1

  mitad ← potencia(a, n / 2)

  SI n ES PAR
    RETORNAR mitad * mitad-
  SINO
    RETORNAR a * mitad * mitad
FIN FUNCION
```

4. Aplicación de mergesort para ordenar algoritmo

```
FUNCION mezclar(izquierda, derecha)
  resultado ← lista vacía

  MIENTRAS izquierda y derecha NO estén vacías
    SI izquierda[0] < derecha[0]
```

```
        agregar izquierda[0] a resultado
      SINO
        agregar derecha[0] a resultado

    agregar elementos restantes
    RETORNAR resultado
  FIN FUNCION
```

5. Dado un numero x encontrar el menor n...

```
  FUNCION contarBits(numero)
    contador ← 0

    MIENTRAS numero > 0
      contador ← contador + (numero MOD 2)
      numero ← numero / 2

    RETORNAR contador
  FIN FUNCION
```

Laboratorio:

Requerimientos funcionales

H001- El sistema debe permitir **ingresar una lista de estudiantes con sus respectivas notas**.

H002- El sistema debe **calcular la suma total de las notas** utilizando el enfoque de **divide y conquista**.

H003- El sistema debe **calcular la cantidad total de estudiantes** usando recursión.

H004- El sistema debe **calcular el promedio del grupo** dividiendo la suma total entre la cantidad de estudiantes.

H005- El sistema debe **identificar y mostrar los estudiantes aprobados** (nota ≥ 3.0).

H006- El sistema debe **mostrar el promedio final del grupo** en pantalla.

H007- El sistema debe **mostrar el nombre y la nota de cada estudiante aprobado**.

Requerimientos no funcionales

El sistema debe ejecutarse correctamente en un computador personal.

El algoritmo debe usar el paradigma **Divide y Conquista**.

El código debe ser **claro, modular y fácil de entender**.

El tiempo de ejecución debe ser eficiente para listas pequeñas y medianas de estudiantes.

El sistema debe mostrar los resultados de forma clara para el usuario.

Historia de usuarios:

Como profesor
Quiero calcular el promedio de las notas del grupo
Para evaluar el rendimiento general de los estudiantes

Criterios de aceptación:

- El sistema recibe una lista de notas.
- El sistema usa divide y conquista para sumar las notas.
- El sistema muestra el promedio correctamente calculado.

Como profesor
Quiero ver qué estudiantes aprobaron
Para saber quiénes cumplieron con la nota mínima requerida

Criterios de aceptación:

- El sistema evalúa cada nota.
- Se muestran solo los estudiantes con nota mayor o igual a 3.0.
- Se muestra el nombre y la nota del estudiante aprobado.

Como profesor
Quiero que el sistema muestre los resultados en pantalla
Para interpretar fácilmente la información

Criterios de aceptación:

- El sistema muestra el promedio del grupo.
- El sistema muestra una lista clara de aprobados.
- Los resultados son legibles y ordenados.

Análisis de complejidad:

Cost	Times
C1	1
C2	1

C3	1
C4	1
C5	1
C6	1
C7	1
C8	1
C9	N+1
C10	1

$c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8 + c9 = c$

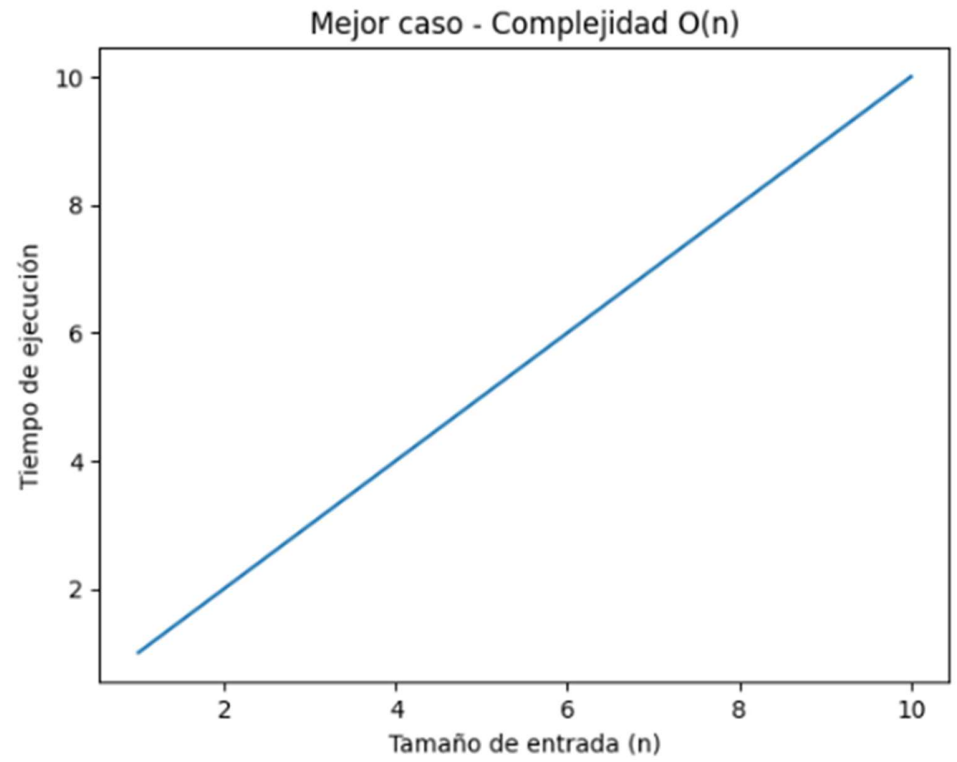
$T(n) = 2T\left(\frac{2}{n}\right) + c$

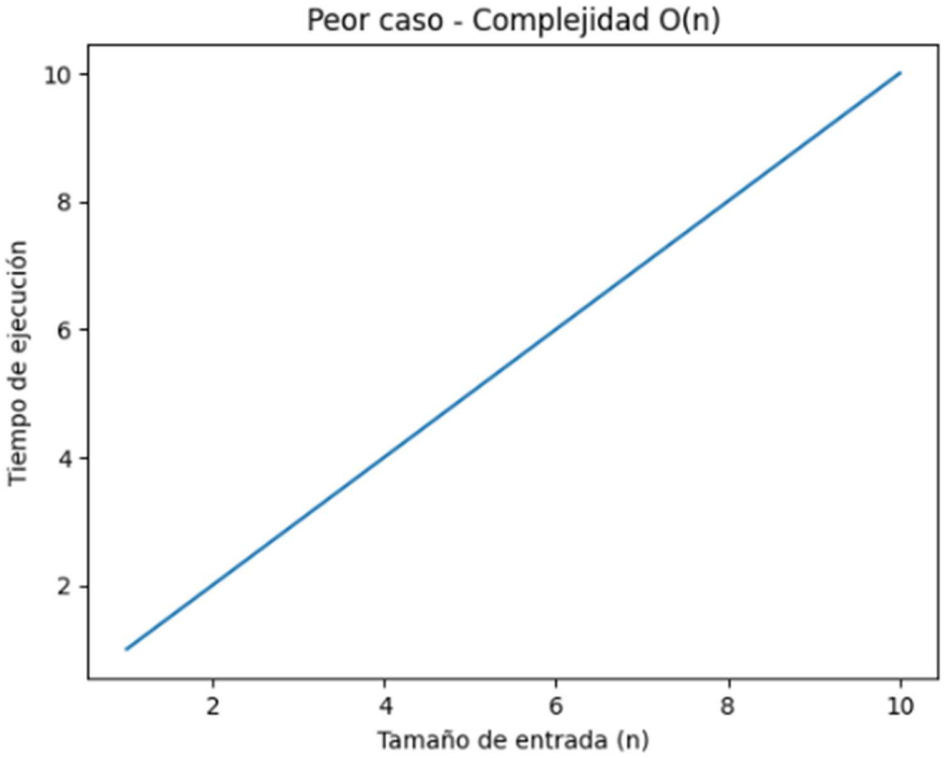
$T(n) = \mathrm{n} \log_2 2 + c$

$T(n) = n \log n = n$

$T(n) = O(n)$

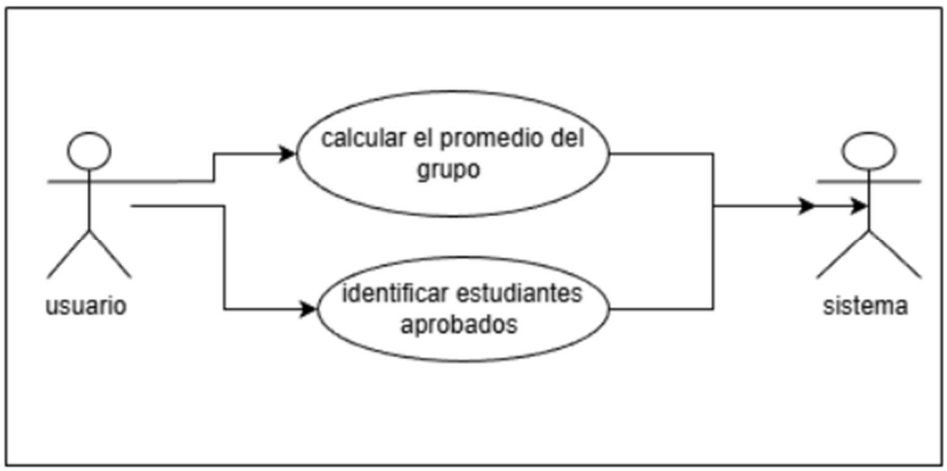
Complejidad es de O(n) - complejidad Big -O

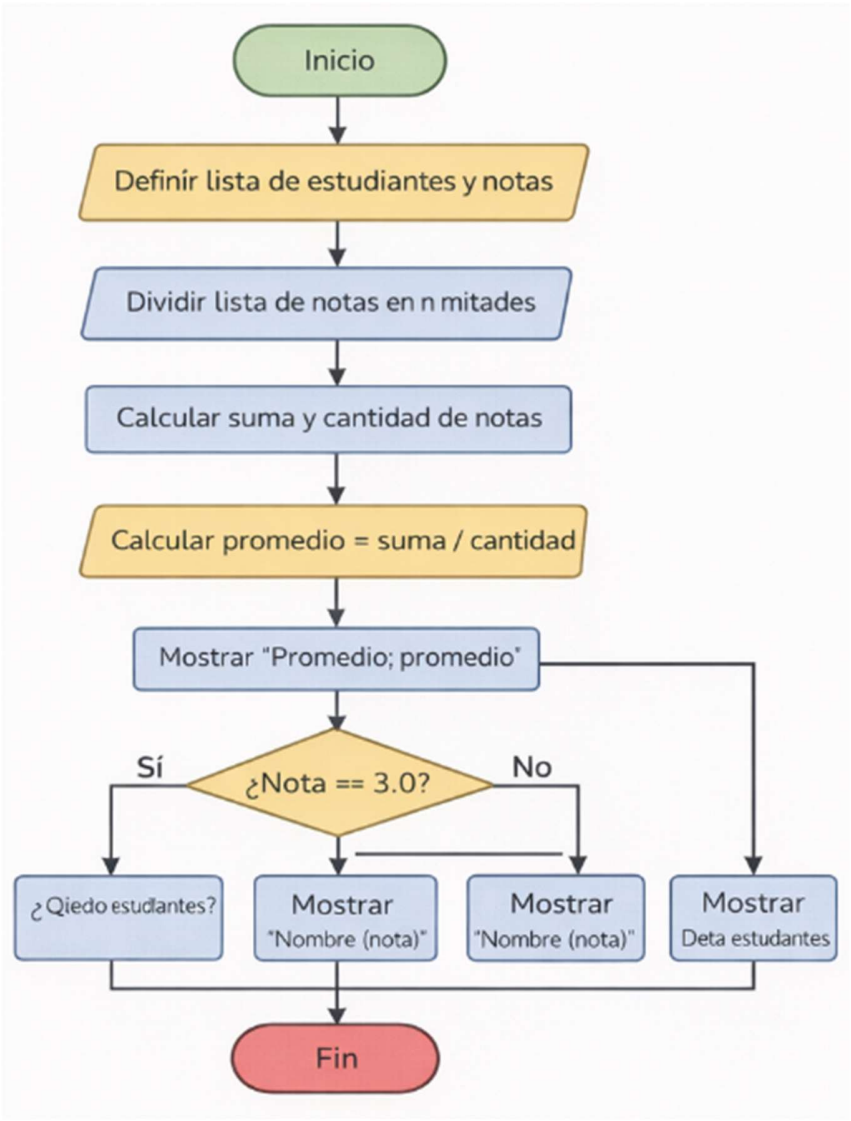

Mejor caso - Complejidad O(n)
Tiempo de ejecución
Tamaño de entrada (n)
2
4
6
8
10


Peor caso - Complejidad O(n)
Tiempo de ejecución
Tamaño de entrada (n)
2
4
6
8
10

Análisis

No tiene tanta complejidad, y son lo mismo, porque el uso de divide y conquistaras es una forma de hacer que se optimice mejor el código y que el costo veces se reduzca memorable mente. Además, también siempre será eficiente por que solo recorre los datos una sola vez.


calcular el promedio del grupo
identificar estudiantes aprobados
usuario
sistema


Inicio
Definir lista de estudiantes y notas
Dividir lista de notas en n mitades
Calcular suma y cantidad de notas
Calcular promedio = suma / cantidad
Mostrar "Promedio; promedio"
Sí
¿Nota == 3.0?
No
¿Qiedo esudiantes?
Mostrar
"Nombre (nota)"
Mostrar
"Nombre (nota)"
Mostrar
Deta estudantes
Fin