



---

**INSTITUTO POLITÉCNICO NACIONAL**  
Unidad Profesional Interdisciplinaria de Ingeniería  
Campus Zacatecas



## **PRACTICA 1**

**ALUMNO:** Luis Eduardo Espino Gtz

18 de Septiembre del 2023

# 1 Introducción

En esta practica hablaremos sobre el metodo de ordenamiento burbuja y sus respectivas mejoras en el tiempo de proceso, asi como algunos ejemplos y comparaciones entre burbuja simple y burbuja optimizada.

# 2 Desarrollo

## ” METODO BURBUJA ”

El ordenamiento de burbuja es un algoritmo que te permite ordenar valores de un arreglo. Funciona revisando cada elemento con su adyacente. Si ambos elementos no están ordenados, se procede a intercambiarlos, si por el contrario los elementos ya estaban ordenados se dejan tal como estaban. Este proceso sigue para cada elemento del arreglo hasta que quede completamente ordenado. el pseudocodigo es el siguiente:

```
Procedimiento MetodoBurbujaSimple ()  
  Inicio Procedimiento  
    entero tempo;  
    desde (entero a = 1; a < tamaño vector; a++)  
      desde (entero b = tamaño vector - 1; b >= a; b--)  
        Inicio for  
          si (vector [b - 1] > vector[b]) entonces  
            inicio del if  
              tempo = vector [b - 1];  
              vector [b - 1] = vector[b];  
              vector[b] = tempo;  
            fin del if  
          Fin de For  
        Fin Procedimiento
```

Figure 1: Pseudocodigo burbuja simple.

## ” MEJOR CASO ”

El mejor caso para este metodo es  $O(n)$ , Esto quiere decir que el tiempo necesario para ejecutar la función, es función directa y lineal del número de elementos que le pasemos.

Como primer ejemplo le introducimos al arreglo 10 numeros de 1 cifra cada uno, podemos observar que con esa entrada no es nada dificil para el programa, ya que su tiempo es 0.0, el caso mas ideal para este metodo es ingresar los numeros de forma ordenada

```

***** ORDENAMIENTO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opcion: 1
Ingrese el numero del elementos a ordenar: 10
numero 1: 1
numero 2: 2
numero 3: 3
numero 4: 4
numero 5: 5
numero 6: 6
numero 7: 7
numero 8: 8
numero 9: 9
numero 10: 10
***** METODO BURBUJA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
tiempo de ejecucion de burbuja: 0.0
Presione una tecla para continuar . . .

```

Figure 2: Ejemplo 1 burbuja.

En el segundo ejemplo tomaremos una entrada de 10 numeros pero ahora introducidos de esta forma, solo modificamos los ultimos 3 numeros, sigue siendo un caso ideal.

```

***** ORDENAMIENTO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opcion: 1
Ingrese el numero del elementos a ordenar: 10
numero 1: 4
numero 2: 5
numero 3: 6
numero 4: 7
numero 5: 8
numero 6: 9
numero 7: 10
numero 8: 1
numero 9: 2
numero 10: 3
***** METODO BURBUJA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
tiempo de ejecucion de burbuja: 0.0
Presione una tecla para continuar . . .

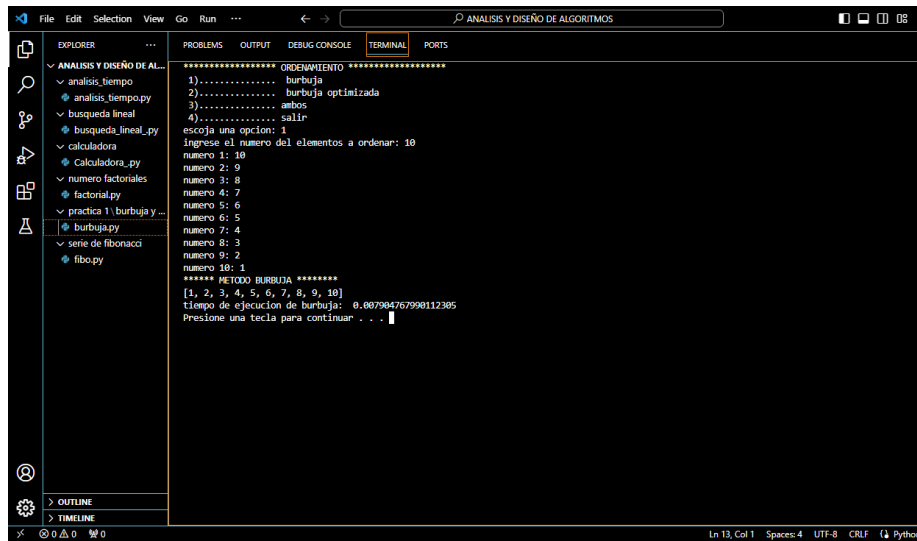
```

Figure 3: Ejemplo 2 burbuja.

## ” PEOR CASO ”

El peor caso de este metodo es  $O(n^2)$ , ya que esta complejidad necesitan realizar una iteración por todos los elementos en cada uno de los elementos a procesar. Si tuviese que hacer la iteración más de una vez serían de complejidad  $O(n^3)$ ,  $O(n^4)$ , etc... pero se trata de casos muy raros y poco optimizados.

Como primer ejemplo introducimos 10 numeros en orden desendente, podemos observar que tarda mas tiempo, ya que tiene que acomodar todos los numeros.



```
File Edit Selection View Go Run ... ANALISIS Y DISEÑO DE ALGORITMOS
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ANALISIS Y DISEÑO DE AL...
  analisis_tiempo.py
  busqueda_lineal
  busqueda_lineal.py
  calculadora
  Calculadora.py
  numero factoriales
  factorial.py
  practica 1\ Burbuja y ...
  burbuja.py
  serie de fibonacci
  fibo.py

***** ORDENAMIENTO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opcion: 1
Ingrese el numero del elementos a ordenar: 10
numero 1: 10
numero 2: 9
numero 3: 8
numero 4: 7
numero 5: 6
numero 6: 5
numero 7: 4
numero 8: 3
numero 9: 2
numero 10: 1
***** METODO BURBUJA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
tiempo de ejecucion de burbuja: 0.007904767990112305
Presione una tecla para continuar . . .
```

Figure 4: Ejemplo 3 burbuja.

Como segundo ejemplo tenemos de igual forma 10 numeros de entrada, esta vez con numeros grandes al principio esto porque el programa tarda mas en moverlos (se tienen que mover la mayoría).

```

***** OCHO EJEMPLOS *****
1)..... burbuja
2)..... burbuja optimizada
3)..... seleccion
4)..... salir
Escriba una opcion: 1
Ingrese el numero de elementos a ordenar: 10
numero 1: 10
numero 2: 9
numero 3: 8
numero 4: 7
numero 5: 6
numero 6: 10
numero 7: 10
numero 8: 7
numero 9: 6
numero 10: 5
***** METODO BURBUJA *****
[5, 4, 5, 7, 2, 6, 9, 10, 10, 10]
Tiempo de ejecucion de burbuja: 8.8
Presione una tecla para continuar . . .

```

Figure 5: Ejemplo 4 burbuja.

## ” CASO PROMEDIO ”

En el caso promedio tenemos la complejidad de igual forma de  $O(n^2)$ , además aparece en bucles o ciclos doblemente anidados. Si  $n$  se duplica, el tiempo de ejecución aumenta cuatro veces.

Como primer ejemplo introducimos 10 numeros, los cuales algunos ya estaran su posición y otros no.

```

***** OCHO EJEMPLOS *****
1)..... burbuja
2)..... burbuja optimizada
3)..... seleccion
4)..... salir
Escriba una opcion: 1
Ingrese el numero de elementos a ordenar: 10
numero 1: 1
numero 2: 10
numero 3: 3
numero 4: 9
numero 5: 5
numero 6: 8
numero 7: 7
numero 8: 6
numero 9: 9
numero 10: 10
***** METODO BURBUJA *****
[1, 1, 5, 6, 7, 8, 9, 9, 10, 10]
Tiempo de ejecucion de burbuja: 8.8
Presione una tecla para continuar . . .

```

Figure 6: Ejemplo 5 burbuja.

## ” METODO BURBUJA OPTIMIZADA ”

Este método recorre todo el arreglo comparando cada uno de los elementos con el elemento siguiente e intercambiándolo de ser necesario. Al finalizar la iteración el elemento mayor queda ubicado en la última posición, mientras los elementos menores ascienden una posición. La mejora es que, como al final de cada iteración el elemento mayor queda situado en su posición, ya no es necesario volverlo a comparar con ningún otro número, reduciendo así el número de comparaciones por iteración.

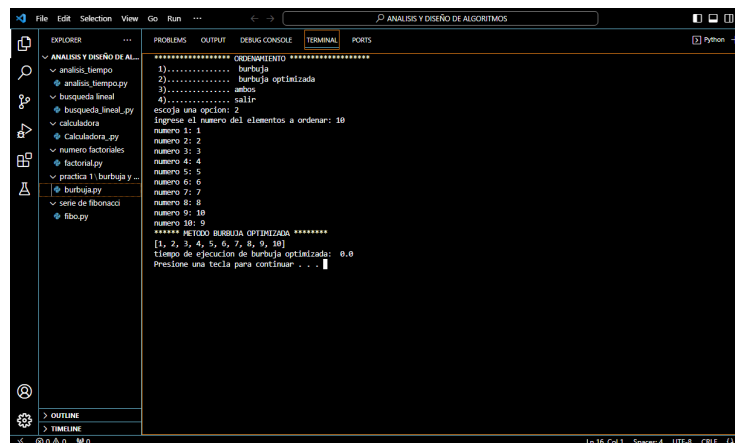
```
Procedimiento MetodoBurbujaMejorada ()
Inicio procedimiento
    entero bandera= 1, AUX = 0;
    entero N = tamaño vector;
    // ordena de mayor a menor
    desde (entero paso = 0; paso < tamaño vector - 1 Y bandera == 1; paso++)
        /* si en el paso anterior no hubo cambios se detiene ciclo */
        Inicio desde externo
            bandera = 0;
            desde (entero j = 0; j < N - paso - 1; j++)
                // las comparaciones van disminuyendo a medida que se efectúan los pasos
                Inicio desde interno
                    Si (vector[j] < vector [j + 1])
                        Inicia condición
                            bandera = 1; /* indica si se han realizados cambios o no */
                            AUX = vector[j];
                            vector[j] = vector [j + 1];
                            vector [j + 1] = AUX;
                        Termina Condición
                    Fin desde interno
                Fin desde externo
            Fin procedimiento
```

Figure 7: Pseudocódigo burbuja optimizada.

## ” MEJOR CASO ”

El mejor caso para este metodo es  $O(n)$ , Esto quiere decir que el tiempo necesario para ejecutar la función, es función directa y lineal del número de elementos que le pasemos.

Como primer ejemplo le introducimos al arreglo 10 numeros de 1 cifra cada uno, podemos observar que con esa entrada no es nada dificil para el programa, ya que su tiempo es 0.0, el caso mas ideal para este metodo es ingresar los numeros de forma ordenada, solamente modificando los 2 ultimos numeros.



```
File Edit Selection View Go Run ... ANALISIS Y DISEÑO DE ALGORITMOS
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ANÁLISIS Y DISEÑO DE AL...
  análisis_tiempo.py
  busqueda_lineal.py
  calculadora.py
  calculadora.py
  numero_factoriales.py
  factorial.py
  practica 1 burbuja y ...
  burbuja.py
  serie de fibonacci.py
  fibo.py

***** CREANDO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... seleccion
4)..... salir
escija una opcion: 2
Ingrese el numero del elementos a ordenar: 10
numero 1: 1
numero 2: 2
numero 3: 3
numero 4: 4
numero 5: 5
numero 6: 6
numero 7: 7
numero 8: 8
numero 9: 10
numero 10: 9
***** METODO BURBUJA OPTIMIZADA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Tiempo de ejecución de burbuja optimizada: 0.0
Presione una tecla para continuar . . .
```

Figure 8: Ejemplo 1 burbuja optimizada.

En el segundo ejemplo tomaremos una entrada de 10 numeros pero ahora introducidos de esta forma, solo modificamos los ultimos 3 numeros, sigue siendo un caso ideal.

```

***** CREANCIENTO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... seleccion
4)..... seleccion
escoge una opcion: 2
Escribe el numero del elemento a ordenar: 10
numero 1: 4
numero 2: 5
numero 3: 6
numero 4: 7
numero 5: 8
numero 6: 9
numero 7: 10
numero 8: 1
numero 9: 2
numero 10: 3
***** METODO BURBUJA OPTIMIZADA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
tiempo de ejecucion de burbuja optimizada: 0.0
Presione una tecla para continuar . . .

```

Figure 9: Ejemplo 2 burbuja optimizada.

## ” PEOR CASO ”

El peor caso de este metodo es  $O(n^2)$ , ya que esta complejidad necesitan realizar una iteración por todos los elementos en cada uno de los elementos a procesar. Si tuviese que hacer la iteración más de una vez serían de complejidad  $O(n^3)$ ,  $O(n^4)$ , etc... pero se trata de casos muy raros y poco optimizados.



Como primer ejemplo introducimos 10 numeros en orden desendente, podemos observar que tarda mas tiempo, ya que tiene que acomodar todos los numeros.

```

File Edit Selection View Go Run ... ANALISIS Y DISEÑO DE ALGORITMOS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
***** CHECKEANDO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opcion: 2
ingrese el numero del elementos a ordenar: 10
numero 1: 10
numero 2: 9
numero 3: 8
numero 4: 7
numero 5: 6
numero 6: 5
numero 7: 4
numero 8: 3
numero 9: 2
numero 10: 1
***** METODO BURBUJA OPTIMIZADA *****
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
tiempo de ejecucion de burbuja optimizada: 0.0
Presione una tecla para continuar . . .

```

Figure 10: Ejemplo 3 burbuja optimizada.

Como segundo ejemplo tenemos de igual forma 10 numeros de entrada, esta vez con numeros grandes al principio esto porque el programa tarda mas en moverlos (se tienen que mover la mayoría).

```

File Edit Selection View Go Run ... ANALISIS Y DISEÑO DE ALGORITMOS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
***** CHECKEANDO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opcion: 2
ingrese el numero del elementos a ordenar: 10
numero 1: 10
numero 2: 9
numero 3: 8
numero 4: 10
numero 5: 9
numero 6: 8
numero 7: 10
numero 8: 9
numero 9: 8
numero 10: 10
***** METODO BURBUJA OPTIMIZADA *****
[8, 8, 9, 9, 9, 9, 10, 10, 10, 10]
tiempo de ejecucion de burbuja optimizada: 0.0009980201721101400
Presione una tecla para continuar . . .

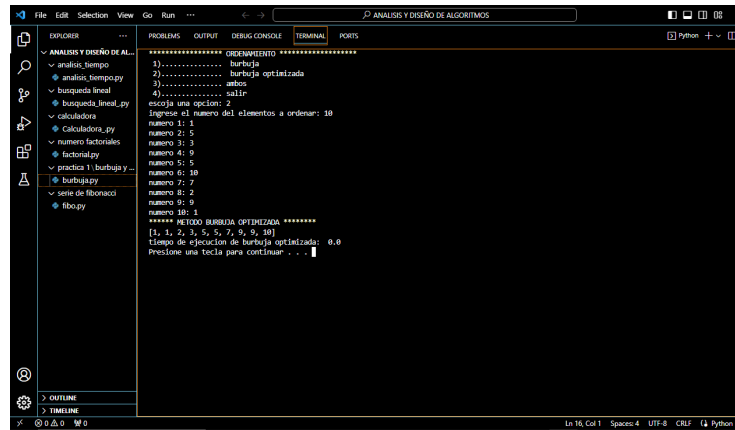
```

Figure 11: Ejemplo 4 burbuja optimizada.

## ” CASO PROMEDIO ”

En el caso promedio tenemos la complejidad de igual forma de  $O(n^2)$ , además aparece en bucles o ciclos doblemente anidados. Si  $n$  se duplica, el tiempo de ejecución aumenta cuatro veces.

Como primer ejemplo introducimos 10 números, los cuales algunos ya estarán su posición y otros no.



```
File Edit Selection View Go Run ... ANALISIS Y DISEÑO DE ALGORITMOS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
EXPLORE
ANÁLISIS Y DISEÑO DE AL...
  análisis_bu...
  análisis_bu...
  búsqueda lineal
  búsqueda lineal.py
  calculadora
  calculadora.py
  numero factoriales
  factorial.py
  practica 1 burbuja y...
  burbuja.py
  serie de fibonacci
  fibo.py
***** ORDENAMIENTO *****
1)..... burbuja
2)..... burbuja optimizada
3)..... ambos
4)..... salir
escoja una opción: 2
Ingrese el número del elemento a ordenar: 10
numero 1: 1
numero 2: 5
numero 3: 3
numero 4: 9
numero 5: 5
numero 6: 10
numero 7: 7
numero 8: 2
numero 9: 9
numero 10: 1
***** METODO BURBUJA OPTIMIZADA *****
[1, 1, 2, 3, 5, 5, 7, 9, 9, 10]
tiempo de ejecución de burbuja optimizada: 0.0
Presione una tecla para continuar ...
```

Figure 12: Ejemplo 5 burbuja optimizada.

## 3 Conclusiones

Como conclusiones podemos decir que el metodo de burbuja optimizada hace un mejor trabajo en comparacion al metodo normal, pero no es demasiada diferencia, ya que la forma en como lo ordena es bastante similar, solo varia en algunas partes en sus codigos de programacion.

La forma en como se introducen los numeros tiene que ver bastante, ya que varia en su tiempo de ordenamiento, entre menos numeros se tengan que mover de pocision, es mejor en el rendimiento del programa.

Ademas es importante destacar que el metodo burbuja es uno de los algoritmos de ordenamiento más sencillos aunque no es el más eficiente, ya que su simplicidad lo convierte en un algoritmo ideal para practicar programación.

Aunque existen mejores algoritmos de ordenamiento, este metodo es un algoritmo suficientemente bueno como para ser un primer acercamiento al estudio de los métodos de ordenamiento.

## 4 Referencias

<https://dcodingames.com/ordenamiento-de-datos-por-el-metodo-burbuja/>  
<https://juncotic.com/ordenamiento-de-burbuja-algoritmos-de-ordenamiento/>  
<https://tutospoo.jimdofree.com/tutoriales-java/m>  
<https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx>