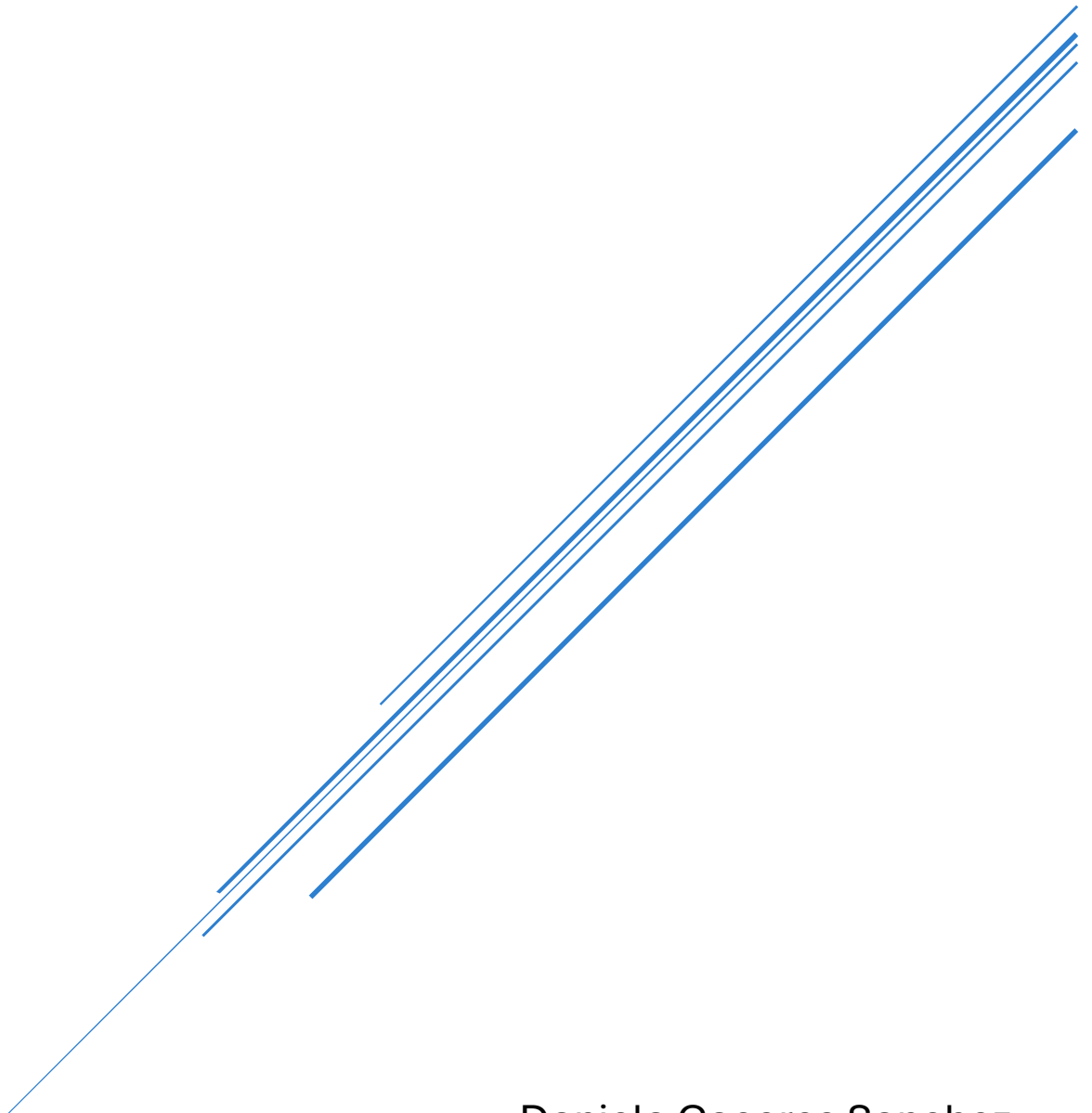


PARCIAL II

Algoritmos II



Daniela Caceres Sanchez
Ing. De sistemas

Implementación creativa de Algoritmos de ordenamiento

Para esta actividad se utiliza diferentes algoritmos para determinar cual es mas eficiente, midiendo el tiempo de ejecución.

Para ello usamos las estructuras unidimensional y bidimensional, con 3 algoritmos diferentes, los cuales son:

- Insercion: que lo use para ordenar el numero de reservas en orden descendente.
- Selección: lo use para ordenar por año en orden ascendente.
- Burbuja: para ordenar por título en orden alfabético.

Para evaluar su desempeño en los tiempos de ejecución recurrí a usar el `console.time()` que mide el tiempo em milisegundos dado en el taller y el `console.timeEnd()`, que pausa el tiempo.

Para organizar la lista de libros inicie creando la constante con el ejemplo dado, continúe usando las variables vistas en clase midiendo el tiempo de demora en ejecutar cada código

■ Arreglo unidimensional

```
// se define el arreglo de libros como objetos con titulo, año y reserva
let libros = [
  {titulo: "El Principito", autor: "Antoine de Saint-Exupéry", año: 1943,
  reservas: 150},
  {titulo: "1984", autor: "George Orwell", año: 1949, reservas: 200},
  {titulo: "Cien Años de Soledad", autor: "Gabriel García Márquez", año: 1967,
  reservas: 300},
  {titulo: "Don Quijote de la Mancha", autor: "Miguel de Cervantes", año: 1605,
  reservas: 50},
  {titulo: "La Odisea", autor: "Homero", año: -800, reservas: 120}
```

- Se inicia el código con el algoritmo de selección: El cual se organiza el año de manera ascendente.

```
//Algoritmos de seleccion para ordenar por año
function ordenarPorAño(libros) {
  let n = libros.length;
  for (let i = 0; i < n - 1; i++) {
    let minIndex = i;
    for (let j = i + 1; j < n; j++) {
      if (libros[j].año < libros[minIndex].año) {
        minIndex = j;
      }
    }
  }
  // Intercambio de elementos
```

```

        if (minIndex !== i) {
            [libros[i], libros[minIndex]] = [libros[minIndex],
libros[i]];
        }
    }
    return libros;
}
//Llamo la funcion para ordenar los libros por titulo
console.time("OrdenamientoPorAño");
console.log("\nOrden por año:", (ordenarPorAño(libros)));
console.timeEnd("OrdenamientoPorAño");

```



- Luego se organiza con el método de inserción organizando las reservas por orden descendente

```

// Algoritmo de insercion por reserva
function ordenarporReserva(libros) {
    for (let i = 1; i < libros.length; i++) {
        let currentBook = libros[i];
        let j = i - 1;
        while (j >= 0 && libros[j].reservas < currentBook.reservas) {
            libros[j + 1] = libros[j]; //Mueve los elementos hacia la derecha
            j--;
        }
        libros[j + 1] = currentBook; //coloca el libro en posicion correcta
    }
    return libros;
}

//realizo el llamado
console.time("OrdenamientoPorReserva");
console.log("\nOrden por Reserva:", ordenarporReserva(libros));
console.timeEnd("OrdenamientoPorReserva");

```

The screenshot shows a web browser's developer console with the 'Consola' tab active. The console displays the output of a sorting operation. At the top, it says 'Orden por Reserva:'. Below this, there is a list of 5 objects, each representing a book with its title, author, year, and reservation count. The objects are sorted in descending order of reservation count. The first object is {titulo: 'Cien Años de Soledad', autor: 'Gabriel García Márquez', año: 1967, reservas: 300}. The second is {titulo: '1984', autor: 'George Orwell', año: 1949, reservas: 200}. The third is {titulo: 'El Principito', autor: 'Antoine de Saint-Exupéry', año: 1943, reservas: 150}. The fourth is {titulo: 'La Odisea', autor: 'Homero', año: -800, reservas: 120}. The fifth is {titulo: 'Don Quijote de la Mancha', autor: 'Miguel de Cervantes', año: 1605, reservas: 50}. Below the list, it shows 'length: 5' and '[[Prototype]]: Array(0)'. At the bottom, it shows the execution time: 'OrdenamientoPorReserva: 0.759033203125 ms'.

```
Orden por Reserva: (5) [{"titulo": "Cien Años de Soledad", "autor": "Gabriel García Márquez", "año": 1967, "reservas": 300}, {"titulo": "1984", "autor": "George Orwell", "año": 1949, "reservas": 200}, {"titulo": "El Principito", "autor": "Antoine de Saint-Exupéry", "año": 1943, "reservas": 150}, {"titulo": "La Odisea", "autor": "Homero", "año": -800, "reservas": 120}, {"titulo": "Don Quijote de la Mancha", "autor": "Miguel de Cervantes", "año": 1605, "reservas": 50}]
length: 5
[[Prototype]]: Array(0)

OrdenamientoPorReserva: 0.759033203125 ms
```

- Por ultimo se realiza el método de burbuja, organizándolo por título por orden alfabético ascendente.

```
//Algoritmo de burbuja por titulo
function OrdenarPorTitulo(libros) {
  for(let i = onabort; i < libros.length - 1; i++) {
    for(let j = 0; j < libros.length - i - 1; j++) {
      if(libros[j].titulo > libros[j + 1]. titulo) {
        //intercambia los libros
        let temp = libros[j];
        libros[j] = libros[j + 1];
        libros[j + 1]= temp;
      }
    }
  }
  return libros;
}

console.time("OrdenamientoPorTitulo");
console.log("\nOrden por Titulo:", OrdenarPorTitulo(libros));
console.timeEnd("OrdenamientoPorTitulo");
```

The screenshot shows a web browser's developer console with the 'Consola' tab active. The console displays the output of a sorting operation. At the top, it says 'Orden por Titulo:'. Below this, there is a list of 5 objects, each representing a book with its title, author, year, and reservation count. The objects are sorted in ascending order of title. The first object is {titulo: '1984', autor: 'George Orwell', año: 1949, reservas: 200}. The second is {titulo: 'Cien Años de Soledad', autor: 'Gabriel García Márquez', año: 1967, reservas: 300}. The third is {titulo: 'Don Quijote de la Mancha', autor: 'Miguel de Cervantes', año: 1605, reservas: 50}. The fourth is {titulo: 'El Principito', autor: 'Antoine de Saint-Exupéry', año: 1943, reservas: 150}. The fifth is {titulo: 'La Odisea', autor: 'Homero', año: -800, reservas: 120}. Below the list, it shows 'length: 5' and '[[Prototype]]: Array(0)'. At the bottom, it shows the execution time: 'OrdenamientoPorTitulo: 0.89501953125 ms'.

```
Orden por Titulo: (5) [{"titulo": "1984", "autor": "George Orwell", "año": 1949, "reservas": 200}, {"titulo": "Cien Años de Soledad", "autor": "Gabriel García Márquez", "año": 1967, "reservas": 300}, {"titulo": "Don Quijote de la Mancha", "autor": "Miguel de Cervantes", "año": 1605, "reservas": 50}, {"titulo": "El Principito", "autor": "Antoine de Saint-Exupéry", "año": 1943, "reservas": 150}, {"titulo": "La Odisea", "autor": "Homero", "año": -800, "reservas": 120}]
length: 5
[[Prototype]]: Array(0)

OrdenamientoPorTitulo: 0.89501953125 ms
```

■ Arreglo bidimensional

```
// se define el arreglo de libros como objetos con título, año y reserva
let libros = [
  [{título: "El Principito", autor: "Antoine de Saint-Exupéry", año: 1943,
  reservas: 150},],
  [{título: "1984", autor: "George Orwell", año: 1949, reservas: 200},],
  [{título: "Cien Años de Soledad", autor: "Gabriel García Márquez", año: 1967,
  reservas: 300},],
  [{título: "Don Quijote de la Mancha", autor: "Miguel de Cervantes", año: 1605,
  reservas: 50},],
  [{título: "La Odisea", autor: "Homero", año: -800, reservas: 120}],
];
```

- Se realiza el mismo método que en el unidimensional usando los métodos
- Método de **burbuja** se organiza de forma alfabética.

```
// Algoritmo de burbuja para ordenar por título
function OrdenarPorTitulo(libros) {
  for (let i = 0; i < libros.length - 1; i++) {
    for (let j = 0; j < libros.length - i - 1; j++) {
      if (libros[j][0].título.localeCompare(libros[j +
1][0].título) > 0) {
        // Intercambia los libros
        let temp = libros[j];
        libros[j] = libros[j + 1];
        libros[j + 1] = temp;
      }
    }
  }
  return libros;
}

// Llamada a la función
console.time("OrdenamientoPorTitulo");
console.log("\nOrden por Título:", OrdenarPorTitulo(libros));
console.timeEnd("OrdenamientoPorTitulo");
```



- Luego realizo el arreglo con el método de **selección** organizando el año en orden ascendente

```
//Algoritmos de seleccion para ordenar por año
function ordenarPorAño(libros) {
    let n = libros.length;
    for (let i = 0; i < n - 1; i++) {
        let minIndex = i;
        for (let j = i + 1; j < n; j++) {
            if (libros[j][0].año < libros[minIndex][0].año) {
                minIndex = j;
            }
        }
        // Intercambio de elementos
        if (minIndex !== i) {
            [libros[i], libros[minIndex]] = [libros[minIndex],
libros[i]];
        }
    }
    return libros;
}

//Llamo la funcion para ordenar los libros por titulo
console.time("OrdenamientoPorAño");
console.log("\nOrden por año:", (ordenarPorAño(libros)));
console.timeEnd("OrdenamientoPorAño");
```



- Se usa el algoritmo de **insercion** para organizar las reservas de manera descendente

```
// Algoritmo de insercion por reserva
function ordenarPorReserva(libros) {
    for (let i = 1; i < libros.length; i++) {
        let currentBook = libros[i][0]; // Accede al objeto dentro del
        array bidimensional
        let j = i - 1;

        while (j >= 0 && libros[j][0].reservas < currentBook.reservas) {
            libros[j + 1] = libros[j]; // Mueve los elementos hacia la
            derecha
            j--;
        }

        libros[j + 1] = [currentBook]; // Coloca el libro en la posición
        correcta, manteniendo la estructura bidimensional
    }
    return libros;
}

//realizo el llamado
console.time("OrdenamientoPorReserva");
console.log("\nOrden por Reserva:", ordenarPorReserva(libros));
console.timeEnd("OrdenamientoPorReserva");
```

```
Orden por Reserva: (5) [Array(1), Array(1), Array(1), Array(1), Array(1)]
  0: Array(1)
    0: {titulo: 'Cien Años de Soledad', autor: 'Gabriel García Márquez', año: 1967, reservas: 300}
    length: 1
    [[Prototype]]: Array(0)
  1: Array(1)
    0: {titulo: '1984', autor: 'George Orwell', año: 1949, reservas: 200}
    length: 1
    [[Prototype]]: Array(0)
  2: Array(1)
    0: {titulo: 'El Principito', autor: 'Antoine de Saint-Exupéry', año: 1943, reservas: 150}
    length: 1
    [[Prototype]]: Array(0)
  3: Array(1)
    0: {titulo: 'La Odisea', autor: 'Homero', año: -800, reservas: 120}
    length: 1
    [[Prototype]]: Array(0)
  4: Array(1)
    0: {titulo: 'Don Quijote de la Mancha', autor: 'Miguel de Cervantes', año: 1605, reservas: 50}
    length: 1
    [[Prototype]]: Array(0)
    length: 5
    [[Prototype]]: Array(0)

OrdenamientoPorReserva: 0.7548828125 ms
```

Comparación de velocidades en con el método unidimensional

Análisis de cada algoritmo:

El de selección es más rápido en listas cortas, pero se vuelve lento con datos o listas más largas. El de inserción cuando la lista está parcialmente organizada y el de burbuja es mas lento por su complejidad.

Tabla de comparación tiempo de ejecución		
Criterio	Algoritmo usado	Tiempo de ejecución (ms)
Título de libro	Burbuja	0.89501953125
Año	Inserción	0.94580078125
Reservas	Selección	0.759033203125

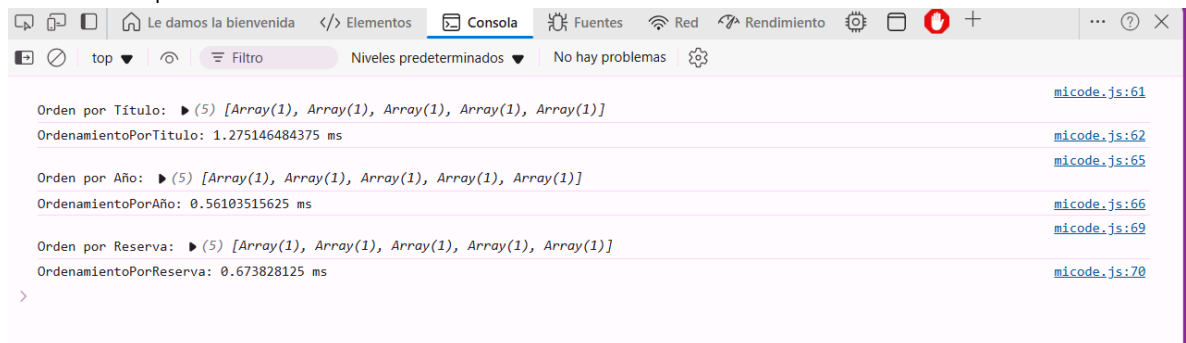
Análisis comparativo: Ya que hago uso de 5 libros en el array, el algoritmo de selección es un poco más rápido que el de inserción y burbuja, al usar el de selección para reservas le tomo aprox 0.759ms para ordenar de forma ascendente, al usar el de burbuja para organizar por titulo le tomo 0.895ms en orden alfabético y el de inserción al ordenar por año en orden descendente le tomo 0.946ms siendo el más demorado, aunque teniendo en cuenta que hay pocos elementos el tiempo de ejecución puede variar

Comparación de velocidades en con el método bidimensional

Tabla comparativa tiempo de ejecución		
Criterio	Algoritmo usado	Tiempo de ejecución
Título	Burbuja	0.498046875
Año	Selección	0.7548828125
Reserva	Inserción	0.741943359375

Análisis comparativo: En este se realiza con la misma cantidad de libros con el cual el método de burbuja fue más rápido en cargar ordenando por título teniendo 0.498ms, seguido el de inserción que le tomo 0.742ms en ordenar y por último el de selección que le tomo 0.755ms en ordenar, pero este resultado puede cambiar a medida que los elementos aumenten.

- Realice una prueba con el código que más me agrado usar el cual fue de inserción y obtuve una respuesta distinta al organizar los libros con el array bidimensional obteniendo tiempos distintos al ordenar el título, año y reserva, al ordenar cada uno le tomo un tiempo distinto a pesar de que es el mismo array, siendo el título al que mas tiempo le tomo tiempo en ordenar 1.275ms, el de reserva le tomo 0.673ms y el que menos tiempo le tomo en ordenar fue el de año 0.561ms.



```
Le damos la bienvenida </> Elementos Consola Fuentes Red Rendimiento
top Filtro Niveles predeterminados No hay problemas

Orden por Título: ▶ (5) [Array(1), Array(1), Array(1), Array(1), Array(1)] micode.js:61
OrdenamientoPorTitulo: 1.275146484375 ms micode.js:62

Orden por Año: ▶ (5) [Array(1), Array(1), Array(1), Array(1), Array(1)] micode.js:65
OrdenamientoPorAño: 0.56103515625 ms micode.js:66

Orden por Reserva: ▶ (5) [Array(1), Array(1), Array(1), Array(1), Array(1)] micode.js:69
OrdenamientoPorReserva: 0.673828125 ms micode.js:70
```