

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II

**Arreglos (Arrays): Búsquedas y
Ordenamientos**

Nombre: Jhonatan Benjamin Mamani Céspedes

CUI: 20232188

Link del repositorio GitHub: <https://github.com/JBenjamin01/fp2-24b.git>

Ejercicio 1: Implementar el algoritmo de Búsqueda Lineal en java.

```
1 public int linearSearch(int[] array, int target) {
2     for (int i = 0; i < array.length; i++) {
3         if (array[i] == target) {
4             return i; // Devuelve el índice del elemento encontrado
5         }
6     }
7     return -1; // Devuelve -1 si el elemento no se encuentra
8 }
```

Ejercicio 2: Implementar el algoritmo de Búsqueda Binaria en java.

```
1 public int binarySearch(int[] array, int target) {
2     int left = 0;
3     int right = array.length - 1;
4
5     while (left <= right) {
6         int mid = left + (right - left) / 2; // Evita desbordamiento
7
8         // Compara el elemento del medio con el objetivo
9         if (array[mid] == target) {
10             return mid; // Devuelve el índice del elemento encontrado
11         }
12         if (array[mid] < target) {
13             left = mid + 1; // Busca en la mitad derecha
14         } else {
15             right = mid - 1; // Busca en la mitad izquierda
16         }
17     }
18     return -1; // Devuelve -1 si el elemento no se encuentra
19 }
```

Ejercicio 3: Implementar el algoritmo de Ordenamiento Burbuja en java.

```
1  public void bubbleSort(int[] array) {
2      int n = array.length;
3      boolean swapped;
4
5      for (int i = 0; i < n - 1; i++) {
6          swapped = false;
7
8          for (int j = 0; j < n - 1 - i; j++) {
9              if (array[j] > array[j + 1]) {
10                 // Intercambiar array[j] y array[j + 1]
11                 int temp = array[j];
12                 array[j] = array[j + 1];
13                 array[j + 1] = temp;
14                 swapped = true;
15             }
16         }
17
18         // Si no hubo intercambios, el arreglo ya está ordenado
19         if (!swapped) {
20             break;
21         }
22     }
23 }
```

Ejercicio 4: Implementar el algoritmo de Ordenamiento por Selection en java.

```
1  public void selectionSort(int[] array) {
2      int n = array.length;
3
4      for (int i = 0; i < n - 1; i++) {
5          // Encontrar el índice del elemento mínimo en el resto del arreglo
6          int minIndex = i;
7          for (int j = i + 1; j < n; j++) {
8              if (array[j] < array[minIndex]) {
9                  minIndex = j; // Actualizar el índice del mínimo
10             }
11         }
12         // Intercambiar el elemento mínimo encontrado con el elemento en la posición i
13         int temp = array[minIndex];
14         array[minIndex] = array[i];
15         array[i] = temp;
16     }
17 }
```

Ejercicio 5: Implementar el algoritmo de Ordenamiento Insertion en java.

```
1  public void insertionSort(int[] array) {
2      for (int i = 1; i < array.length; i++) {
3          int key = array[i];
4          int j = i - 1;
5
6          // Mover elementos de array[0..i-1] que son mayores que key
7          // a una posición adelante de su posición actual
8          while (j >= 0 && array[j] > key) {
9              array[j + 1] = array[j];
10             j--;
11         }
12         array[j + 1] = key;
13     }
14 }
```