

**ESCUELA CIENCIAS EXACTAS E INGENIERÍA**  
**PROGRAMA CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL**  
**ASIGNATURA ESTRUCTURAS DE DATOS LINEALES**  
**PRIMERA ACTIVIDAD CORTE #3 ALGORITMOS DE ORDENAMIENTO BÚSQUEDA**

### Propósito:

El propósito de esta actividad es evaluar las competencias para comprender e implementación los fundamentos de algoritmos de ordenamiento y búsqueda, utilizando estructuras de datos lineales dinámicas y estructuras de datos estáticas para la solución a problemas de ordenamiento y búsqueda de datos.

### Ejercicio #1 Lista Productos Vendidos Totalizada

Se tiene una lista encadenada con datos de productos vendidos a los clientes. Los datos de una venta son: código de producto, cantidad vendida y precio de venta. Un pedido incluye uno o varios productos y un producto se encuentra en diferentes ventas, los datos se encuentran en una lista encadenada en orden de llegada. Tenga en cuenta que el código de un producto se encuentra en diferentes ventas.

Desarrollo un TDA que implemente las siguientes funcionalidades.

**Función poblarDatos(lista).** La función asignar los datos de las ventas a la lista de entrada. Los datos cada venta son aleatorios. El código del producto debe ser un numero aleatorio de 4 dígitos y se debe repetir en diferentes ventas, los datos cantidad y precio son aleatorios.

**Función mostrarDatos(lista).** La función muestra por consola los datos de las ventas contenidos en la lista de entrada o la lista de salida.

**Función sumaVentas(lita,lista).** La función totaliza la cantidad y promedia el precio de venta por cada producto repetido vendido. Durante el proceso la función debe adicionar a la lista de salida un dato por cada producto repetido, con el código del producto, total cantidades vendidas y promedio del precio de venta.

**Función main() o test.** La función permite probar el TDA, declara la lista de entrada y salida y llama a las funciones: poblarDatos(lista), mostrarDatos(lista), sumaVentas(lita,lista), mostrarDatos(lista).

**Datos de entrada:** Lista encadenada de objetos de la clase Producto.

código	1020	1010	1050	1030	1020	1010	1020	1030	1050
cantidad	10	8	8	7	10	5	5	2	1
precio	180	55	10	15	180	60	180	20	10

**Datos de Salida.** Lista Encadenada de objetos de clase producto.

código	1010	1020	1030	1050
cantidad	13	25	9	9
precio	57,5	180	17,5	10

### Restricciones:

1. No se requiere implementar **Map, Hash Table, Set**.
2. No se requiere implementar programación funcional **Expresión lambda**
3. Al finalizar el proceso la lista con los datos de entrada debe estar vacía.

**ESCUELA CIENCIAS EXACTAS E INGENIERÍA**  
**PROGRAMA CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL**  
**ASIGNATURA ESTRUCTURAS DE DATOS LINEALES**  
**PRIMERA ACTIVIDAD CORTE #3 ALGORITMOS DE ORDENAMIENTO BÚSQUEDA**

## Ejercicio #2 Nómina de la Empresa Paga lo gusto.

La nómina de la empresa Paga lo gusto, está registrada en dos arreglos unidimensionales y un arreglo bidimensional. En el primer arreglo unidimensional se guarda el número de identificación y en el segundo el nombre del empleado, en el arreglo bidimensional se guardan los siguientes datos: sueldo básico, deducciones y neto a pagar, el dato neto a pagar se calcula así:  $\text{neto a pagar} = \text{sueldo básico} - \text{deducciones}$ .

Desarrolle un TDA que implemente las siguientes funcionalidades.

**Función poblarDatos(arreglo,arreglo).** La función debe poblar los arreglos con datos aleatorios.

**Función ordenarPorIdentificación(arreglo,arreglo).** La función debe ordenar los arreglos por el número de identificación, Implemente el algoritmo de ordenamiento por selección.

**Función int buscarEmpleado(arreglo,arreglo, entero).** La función busca un empleado por el número de la identificación y si lo encuentra retorna la posición, Si no lo encuentra retorna -1. La función main() o test muestre en la consola los datos nombre, sueldo básico, deducciones y neto a pagar o "Empleado no encontrado". Para buscar el empleado por número identificación implemente el algoritmo de búsqueda binaria.

**Función ordenarPorNombre(arreglo,arreglo).** La función debe ordenar los arreglos por el nombre del empleado. Implemente el algoritmo de ordenamiento QuicSort.

**Función mostrarDatos(arreglo,arreglo).** La función muestra por consola los datos de todos los empleados con la siguiente información, identificación, nombre, sueldo básico, deducciones y neto a pagar. Ordenado por nombre.

**Función main() o test.** La función permite probar el TDA, declara los arreglos y llama a las funciones: poblarDatos(arreglo,arreglo), ordenarPorIdentificación(arreglo,arreglo), int buscarEmpleado(arreglo,arreglo, entero), ordenarPorNombre(arreglo,arreglo), mostrarDatos(arreglo,arreglo).

1040	1010	1050	1020	1035
------	------	------	------	------

Juanita	Pachita	Albita	Pedrito	Damaris
---------	---------	--------	---------	---------

1600000	2000000	3500000	1600000	2000000
100000	200000	500000	100000	200000
1500000	1800000	3000000	1500000	1800000