

Propósito:

El propósito de esta actividad es evaluar las competencias del estudiante, en la selección de algoritmos de ordenamiento y búsqueda que permitan resolver problemas en los que se guardan los datos en estructuras de datos lineales estáticas y estructuras de datos lineales dinámicas.

Criterios de Evaluación.

1. Diseño e implementación de la solución.
2. Selección y modificación del algoritmo de ordenamiento y/o búsqueda.
3. Los datos asignados a los elementos de los arreglos deben ser aleatorios.
4. El algoritmo de búsqueda a implementar es búsqueda binaria.
5. Eficiencia del módulo, como se comporta con la entrada de datos y la salida de información.

Ejercicio 1. Notas Estudiantes.

Un docente maneja la lista de estudiantes y calificaciones mediante arreglos unidimensionales, en un arreglo de tipo entero guarda los códigos, en uno de tipo cadena, los nombres; y en un arreglo bidimensional de cuatro columnas guarda las notas: corte 1, corte 2 y corte 3 y la definitiva. La definitiva se calcula así: definitiva = (parcial1*30% + parcial2*30% + parcial3*30%).

El docente requiere la siguiente información según estos criterios:

1. Buscar un estudiante por el código y mostrar el nombre, las tres notas y la definitiva.
2. Una lista con los datos del estudiante, las tres notas y la nota definitiva, ordenado descendente por nota definitiva.

Ejercicio 2. Nómina 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: **neto a pagar = sueldo básico – deducciones**.

Se requiere un programa que permita mostrar la siguiente información:

1. Buscar un empleado por el número de identificación y mostrar el nombre y los siguientes datos: sueldo básico, deducciones y neto a pagar.
2. Una lista con los datos de los empleados, sueldo básico, deducciones y neto a pagar ordenado por nombre.

Ejercicio # 3 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 organizados en la lista encadenada por orden de llegada. Tenga en cuenta que el código de un producto se encuentra en diferentes ventas.

Cree un TDA que implemente las siguientes funcionalidades:

1. Declarar la lista encadenada de ventas de entrada y salida.
2. Función para asignar datos a la lista. Datos de producto vendido en orden de llegada.
 - a. Código del producto debe ser un numero aleatorio de 4 dígitos y se debe repetir en diferentes ventas.
 - b. Los datos cantidad y precio deben ser aleatorios.
3. Función para mostrar por consola los datos contenidos en cada elemento de la lista de entrada.
4. Función para sumar las cantidad y promediar el precio de venta por cada producto vendido. La función debe retornar una nueva lista totalizada por producto. to
5. Función para mostrar por consola los datos de la lista de salida y el total general de productos vendidos y el total de las ventas.

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.