ENTREGA DE PARCIAL 2

ESTRUTURA DE DATOS

OSCAR VELANDIA SALGADO

UNICIENCIA

Moto

```
// Clase Moto para muestras de inventario de motos
public class Moto {
    // Atributos de la clase Moto declaracion
    private String marca;
    private int kilometraje;
    private String color;
    private String tipo;

    // Constructor de la clase marcas
    public Moto(String marca, int kilometraje, String color, String tipo) {
        this.marca = marca;
        this.kilometraje = kilometraje;
        this.color = color;
        this.tipo = tipo;
    }

    // Getters y setters atributos clase Moto
    // ...

    // String para representar la clase moto
    @Override
    public String toString() {
        // ...
    }
}
```

Comoarador

```
// Clase contiene comparadores estáticos para la clase Moto
public class ComparadorMoto {
```

```
    // Comparador por marca de las motos
    public static Comparator<Moto> porMarca() {
        return Comparator.comparing(Moto::getMarca);
    }

    // comparar por kilometraje de moto
    public static Comparator<Moto> porKilometraje() {
        return Comparator.comparingInt(Moto::getKilometraje);
    }

    // Comparador por color
    public static Comparator<Moto> porColor() {
        return Comparator.comparing(Moto::getColor);
    }

    // Comparador por tipo
    public static Comparator<Moto> porTipo() {
        return Comparator.comparing(Moto::getTipo);
    }
}
```

Nodo

```
// Clase Nodo para representar un nodo en un árbol binario de búsqueda
public class Nodo {
    Moto moto; // almacena la moto en el nodo
    Nodo izquierdo; // almacena la referencia al nodo hijo
    Nodo derecho; // almacena la referencia al nodo hijo derecho

    // Constructor de la clase Nodo
    public Nodo(Moto moto) {
```

```
        this.moto = moto;
    }
}

ArbolBinario
// representa un árbol binario de búsqueda los vehículos ( Moto )
public class ArbolBinarioBusqueda {
    private Nodo raiz; // almacena nodo raíz del árbol
    private Comparator<Moto> comparador; // ordena los nodos en el árbol

    // Constructor de la clase ArbolBinarioBusqueda
    public ArbolBinarioBusqueda(Comparator<Moto> comparador) {
        this.comparador = comparador;
    }

    // Método para insertar una moto en el árbol
    public void insertar(Moto moto) {
        raiz = insertarRecursivo(raiz, moto);
    }

    // Método recursivo para insertar una moto en el árbol
    private Nodo insertarRecursivo(Nodo actual, Moto moto) {
        // ...
    }

    // recorre el árbol en inorden y aplicar una acción a cada moto
    public void inorden(Consumer<Moto> action) {
        inordenRecursivo(raiz, action);
    }
```

```
// Método para recorrer el árbol en orden y aplicar una acción a cada vehiculo
private void inorden
```