PARQUEMOVIL.JAVA

```
import javax.swing.*;

public class ParqueMovil {
   private static Flota flota;

   public static void main(String[] args) {
      flota = new Flota();

      if (JOptionPane.showConfirmDialog(null,
            "¿Quieres inicializar la flota con 5 vehiculos?",
            "Inicialización", 0) == 0) {
         flota.flotaInicial();
         flota = Flota.ordenarMatriculas(flota);
      }

      JOptionPane.showMessageDialog(null, flota.getCadenaFlota());
      Menu opcionMenu;
      do {
         int opcion = JOptionPane.showOptionDialog(null,
               "Elija una opción",
               "Menu",
               0,
               -1,
               null,
               Menu.values(),
               Menu.Alta);

         opcionMenu = Menu.values()[opcion];
         Vehiculo[] desplegable;
         Vehiculo vehiculo;

         switch (opcionMenu) {
            case Alta:
               vehiculo = Flota.introducirVehiculo();
               if (vehiculo == null) {
                  JOptionPane.showMessageDialog(null, "Fallo al introducir algun dato");
               } else if (flota.comprobarMatricula(vehiculo)) {
                  JOptionPane.showMessageDialog(null, "El vehiculo ya existe");
               } else {
                  flota.altaVehiculo(vehiculo);
               }
               break;
            case Baja:
               desplegable = flota.menuDesplegable();
               if (desplegable.length > 0) {
                  vehiculo = (Vehiculo) JOptionPane.showInputDialog(null,
                        "Elija un vehiculo a dar de baja",
                        "Lista disponible",
                        3,
                        null,
```

```
                    desplegable, desplegable[0]);

                flota.bajaVehiculo(vehiculo);
            } else {
                JOptionPane.showMessageDialog(null, "No hay ningún vehiculo");
            }
            break;
        case Datos:
            flota = Flota.ordenarMatriculas(flota);
            desplegable = flota.menuDesplegable();

            if (desplegable.length > 0) {
                vehiculo = (Vehiculo) JOptionPane.showInputDialog(null,
                    "Elije un vehiculo a mostrar",
                    "Lista disponible",
                    3,
                    null,
                    desplegable,
                    desplegable[0]);
                if (vehiculo != null) {
                    String cadena = String.format("%10s %10s %13s %14s",
                        "Matricula", "Marca", "Carga", "KM");
                    cadena += "\n" + vehiculo.getDatosVehiculo();
                    JOptionPane.showMessageDialog(null, cadena);
                }
            } else {
                JOptionPane.showMessageDialog(null, "No hay ningún vehiculo");
            }
            break;
        case Listado:
            flota = Flota.ordenarMatriculas(flota);
            JOptionPane.showMessageDialog(null, flota.getCadenaFlota());
            break;
        default:
            break;
      }
    } while (opcionMenu != Menu.Salir);
  }
}
```

MARCAS.JAVA

```
public enum Marcas {
  FORD("Ford"),
  MERCEDES("Mercedes"),
  PEGASO("Pegaso");

  private String cadena;

  Marcas(String marca) {
```

```
        this.cadena = marca;
    }
}
```

MENU.JAVA

```
public enum Menu {
    Alta("Alta de vehiculo"),
    Baja("Baja de vehiculo"),
    Datos("Datos de un vehiculo"),
    Listado("Listado de vehiculos"),
    Salir("Salir");

    private final String cadena;

    Menu(String opcion) {
        this.cadena = opcion;
    }
}
```

VEHICULO.JAVA

```
public class Vehiculo {
    private int kilometraje;
    private Marcas marca;
    private int cargaMaxima;
    private String matricula;

    public int getKilometraje() {
        return kilometraje;
    }

    public Marcas getMarca() {
        return marca;
    }

    public int getCargaMaxima() {
        return cargaMaxima;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public Vehiculo(int kilometros, Marcas marca, int carga, String matricula) {
```

```
        this.kilometraje = kilometros;
        this.marca = marca;
        this.cargaMaxima = carga;
        this.matricula = matricula;
    }

    public Vehiculo(Marcas marca, int carga, String matricula) {
        this(0, marca, carga, matricula);
    }

    @Override
    public String toString() {
        return this.getMatricula();
    }

    public String getDatosVehiculo() {
        return String.format("%10s %10s %10dKg %8dkm",
                this.getMatricula(),
                this.getMarca(),
                this.getCargaMaxima(),
                this.getKilometraje());
    }

}
```

FLOTA.JAVA

```
import javax.swing.*;

public class Flota {
    private static final int VEHICULOS_MAXIMOS = 50;
    private Vehiculo[] vehiculos = new Vehiculo[VEHICULOS_MAXIMOS];
    private static int cantidadVehiculos = 0;

    public boolean altaVehiculo(Vehiculo vehiculo) {
        if (cantidadVehiculos == VEHICULOS_MAXIMOS) {
            return false;
        } else {
            this.vehiculos[cantidadVehiculos] = vehiculo;
            this.cantidadVehiculos++;
            return true;
        }
    }

    public int posicionVehiculo(Vehiculo vehiculoABuscar) {
        int posicion = -1;
        for (int i = 0; i < this.cantidadVehiculos; i++) {
            if (this.vehiculos[i].getMatricula().equals(vehiculoABuscar.getMatricula())) {
                posicion = i;
            }
```

```
        }
        return posicion;
    }

    public boolean bajaVehiculo(Vehiculo vehiculo) {
        int posicionABorrar = this.posicionVehiculo(vehiculo);

        if (posicionVehiculo(vehiculo) != -1) {
            for (int i = posicionABorrar; i < cantidadVehiculos; i++) {
                this.vehiculos[i] = this.vehiculos[i + 1];
            }
            this.cantidadVehiculos--;
            return true;
        } else {
            return false;
        }
    }

    public void flotaInicial() {
        this.altaVehiculo(new Vehiculo(300, Marcas.FORD, 3000, "4352-GHY"));
        this.altaVehiculo(new Vehiculo(Marcas.MERCEDES, 5000, "5432-BGC"));
        this.altaVehiculo(new Vehiculo(Marcas.MERCEDES, 3500, "1234-NKU"));
        this.altaVehiculo(new Vehiculo(150, Marcas.PEGASO, 3000, "7865-RTW"));
        this.altaVehiculo(new Vehiculo(Marcas.FORD, 2000, "9874-HYQ"));
    }

    public String getCadenaFlota() {
        String cadena = "";
        if (this.cantidadVehiculos == 0) {
            return "No hay ningún vehiculo";
        } else {
            cadena = cadena + String.format("%10s %10s %10s %14s", "Matricula", "Marca", "Carga",
"KM");
            cadena += "\n";
            for (int i = 0; i < this.cantidadVehiculos; i++) {
                cadena = cadena + this.vehiculos[i].getDatosVehiculo() + "\n";
            }

            return cadena;
        }
    }

    public Vehiculo[] menuDesplegable() {
        Vehiculo[] salida = new Vehiculo[this.cantidadVehiculos];

        for (int i = 0; i < this.cantidadVehiculos; i++) {
            salida[i] = this.vehiculos[i];
        }

        return salida;
    }
```

```
    public static Vehiculo introducirVehiculo() {
        String matricula = JOptionPane.showInputDialog(null, "Matricula");
        if (matricula == null) {
            return null;
        } else {
            Marcas marca = (Marcas) JOptionPane.showInputDialog((null),
                    "Marca",
                    "Marca",
                    0,
                    null,
                    Marcas.values(),
                    Marcas.FORD);

            if (marca == null) {
                return null;
            } else {
                String StringCarga = JOptionPane.showInputDialog(null, "Carga Máxima");
                if (StringCarga == null) {
                    return null;
                } else {
                    int carga = Integer.parseInt(StringCarga);
                    Vehiculo vehiculo = new Vehiculo(marca, carga, matricula);
                    return vehiculo;
                }
            }
        }
    }

    public boolean comprobarMatricula(Vehiculo vehiculo) {
        for (int i = 0; i < cantidadVehiculos; i++) {
            if (this.vehiculos[i].getMatricula().equals(vehiculo.getMatricula())) {
                return true;
            }
        }
        return false;

    }

    public static Flota ordenarMatriculas(Flota flota) {
        Vehiculo aux;

        for (int i = 0; i < cantidadVehiculos - 1; i++) {
            for (int j = 0; j < cantidadVehiculos - i - 1; j++) {
                if (flota.vehiculos[j].getMatricula().compareTo(flota.vehiculos[j + 1].getMatricula()) > 0)
{
                    aux = flota.vehiculos[j];
                    flota.vehiculos[j] = flota.vehiculos[j + 1];
                    flota.vehiculos[j + 1] = aux;
                }
            }
        }
        return flota;
```

```
    }
}
```