



**CICLO DAM
MÓDULO DE CJ**

Tarea N.º 2

**Alumno:
Francisco Hueso Aguilar
02644675C**

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2020 Fomento Ocupacional FOC SL todos los derechos reservados.

Contenido

1.	Documentos que se adjuntan a este informe.....	2
2.	Funcionamiento y desarrollo de la aplicación “gestión de vehículos”	2
2.1	Funcionamiento de la aplicación	2
2.2	Desarrollo de la aplicación	5

1. Documentos que se adjuntan a este informe

A continuación, se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Proyecto NetBeans – Java gestion_vehiculos

2. Funcionamiento y desarrollo de la aplicación “gestión de vehículos”

2.1 Funcionamiento de la aplicación

La aplicación se desarrolla en una única ventana dónde se representan varias escenas generadas utilizando la biblioteca JavaFX y el editor de escenas Scene Builder.

Para poner en marcha la aplicación basta con hacer doble click sobre el archivo gestion_vehiculos.jar, dentro de la carpeta dist del proyecto.

La escena principal está creada en un GridPane de dos columnas. En la columna izquierda se encuentran los botones “Coche”, “Moto” y “Bici” los cuales desencadenan los eventos que visualizan, en la columna de la derecha, los StackPane con la vista correspondiente a cada uno de estos tres botones.

El Stack Pane Coche es el que se visualizará por defecto, al iniciar la aplicación, en la escena principal.

Este panel ofrece la posibilidad al usuario de gestionar varios coches. Recoge los datos del vehículo y su mantenimiento.

Los botones (iguales para todos los paneles), ponen en marcha los eventos de dar de alta un vehículo, darlo de baja, modificarlo, buscar un vehículo por matrícula o ID, retroceder a un vehículo anterior, avanzar a un vehículo posterior y limpiar los campos para dejar la interfaz limpia de datos.

La gestión de todos estos eventos incluye un completo control de errores para evitar que la aplicación se cuelgue.

Gestion de vehículos

Gestión de vehículos

by FHueso

Matrícula
1457DLH

Marca
Opel

Modelo
Zafira

Color
gris

Cilindrada CC
2200

Potencia CV
125

Año de fabricación
2005

Fecha vto. seguro
23/05/2022

Fecha prox. ITV
29/05/2022

☒ Diesel ☐ Gasolina

Mantenimiento

01/08/2016	220000	Hecho por mí
		Aceite motor, tacos de motor
09/01/2017	231000 km	Opel Gil motor
		Valvolina cambio.
20/05/2018	241600 km	Fewbert
		Filtro habitáculo y refrigerante.
23/12/2019	257000 km	Lendacars
		Pastillas freno x 4. Brembo
03/06/2020	280.800 km	Hecho por mí
		Filtro gasoil, líquido frenos, aceite y filtro
31/08/2021	285.000 km	Hecho por mí
		Aceite motor y filtro

<< Buscar X >> - Modificar +

Imagen: Escena principal + Stack Pane Coche

Al ejecutar la aplicación, y si hay registros en la base de datos, automáticamente cada StackPane mostrará su primer vehículo (orden ascendente por matrícula o ID).

El funcionamiento del panel Moto es igual al del panel Coche. Únicamente varía en los distintos atributos que guarda cada tipo de vehículo. La misma observación es aplicable al panel Bici.

Los botones de la escena principal cuentan con una pequeña animación que resalta el efecto visual al ser presionados.

by FHueso

Gestión de vehículos

Matrícula
6879CWS

Marca
Yamaha

Modelo
Fz6-s

Color
blanco

Cilindrada CC
600

Potencia CV
98

Año de fabricación
2006

Fecha vto. seguro
18/08/2022

Fecha prox. ITV
21/05/2022

☐ A1 ☒ A1 ☐ A2

<< **Buscar** X >>

- **Modificar** +

Mantenimiento

15/08/2017
70000 km
Hecho por mí
Aceite motor y neumáticos metzeler

15/08/2018
80000 km
Hecho por mí
Aceite motor y bujías

01/08/2019
90000 km
Hecho por mí
Aceite motor, líquido de frenos

15/08/2020
100000km
Hecho por mí
Aceite motor y refrigerante

01/08/2021
110000 km
Hecho por mí
Aceite motor y filtro, bujías

15/03/2022
115000 km
Motos IQ
Cable embrague y cable acelerador

Imagen: Escena principal + Stack Pane Moto

by FHueso

Gestión de vehículos

Marca
Rock Rider

Modelo
MTB 26

Color
naranja

Año de compra
2022

Identificador
DECAT1

Tamaño
☐ Adulto ☒ Cadete ☐ Niño

Tipo
☒ Montaña ☐ Carretera ☐ Paseo

<< **Buscar** X >>

- **Modificar** +

Rutas

01/04/2022
10 km
Ruta La Fortuna - Leganés

08/04/2022
30 km
Ruta La Fortuna - Alcorcón

Imagen: Escena principal + Stack Pane Bici

2.2 Desarrollo de la aplicación

La permanencia de los objetos generados en la aplicación se consigue gracias al sistema gestor de bases de datos orientadas a objetos Db4o.

Las clases Coche, Moto y Bici heredan de la superclase Vehiculo. Todos los objetos generados, sean de la clase que sean, son guardados en una única base de datos, la cuál almacena objetos de tipo Vehiculo.

Los eventos generados en cada panel, haciendo uso del polimorfismo, distinguen de cada objeto Vehiculo, el que corresponde a la subclase que se necesite en cada momento.

Todo el proyecto cuenta con comentarios bajo la nomenclatura de JavaDoc. La API generada se encuentra en la ruta dist/javadoc del proyecto.

El proyecto ha sido versionando utilizando la herramienta Git, y está subido al repositorio https://github.com/FHuesoGitHub/gestionVehiculos_NetBeans. Este repositorio se hará público a mediados del mes de junio.

A continuación, se procede a comentar brevemente el contenido de cada paquete del proyecto.

- **Paquete controllerBici**

Contiene la clase vistaBiciController.java, la cual gestiona la vista vistaBici.fxml

```
/**
 * FXML Controller class
 *
 * Esta clase maneja los objetos de la vista bici y sus eventos.
 *
 * @author Francisco Hueso
 * @version 02/2022
 */
public class VistaBiciController implements Initializable {

    @FXML
    private Button botonAntBici, botonBuscarBici, botonSigBici, botonBajaBici, botonModBici, botonAltaBici, botonLimpiarBici;
    @FXML
    private TextField textMarca, textModelo, textColor, textAnioCompra, textId;
    @FXML
    private TextArea areaBici;
    @FXML
    private RadioButton radioAdultoBici, radioCadeteBici, radioNinioBici, radioMontaniaBici, radioCarrtereaBici, radioPaseoBici;

    private String marca = "", modelo = "", color = "", annio = "", id = "", area = "";
    private tmn tamano;
    private tp tipo;
}
```

Imagen: Atributos clase VistaBiciController

- **Paquete controllerCoche**

Contiene la clase vistaCocheController.java, la cual gestiona la vista vistaCoche.fxml

```
public void action(ActionEvent e) {

    if (e.getSource().equals(botonAltaCoche)) {
        altaCoche();
    }

    if (e.getSource().equals(botonBajaCoche)) {
        bajaCoche();
    }

    if (e.getSource().equals(botonLimpiarCoche)) {
        limpiarCampos();
    }

    if (e.getSource().equals(botonBuscarCoche)) {
        buscarCoche();
    }

    if (e.getSource().equals(botonModCoche)) {
        modificarCoche();
    }

    if (e.getSource().equals(botonAntCoche)) {
        anteriorCoche();
    }

    if (e.getSource().equals(botonSigCoche)) {
        siguienteCoche();
    }

}
```

Imagen: Método action clase VistaCocheController.java

- **Paquete controllerMain**

Contiene la clase vistaPrincipalController.java, la cual gestiona la vista vistaPrincipal.fxml


```

@FXML
public void action(ActionEvent e) {

    /*
    Dependiendo del botón pulsado, se muestra la vista correspondiente
    en el stack pane de la vista principal. Estas vistas fueron cargadas previamente,
    en el método initialize.
    */
    if (e.getSource().equals(botonCoche)) {

        StackPaneMoto.setVisible(false);
        StackPaneBici.setVisible(false);
        StackPaneCoche.setVisible(true);
    }

    if (e.getSource().equals(botonMoto)) {

        StackPaneMoto.setVisible(true);
        StackPaneBici.setVisible(false);
        StackPaneCoche.setVisible(false);
    }

    if (e.getSource().equals(botonBici)) {

        StackPaneMoto.setVisible(false);
        StackPaneBici.setVisible(true);
        StackPaneCoche.setVisible(false);
    }
}

```

Imagen: Método action clase VistaPrincipalController

○ Paquete controllerMoto

Contiene la clase vistaMotoController.java, la cual gestiona la vista vistaMoto.fxml

```

@Override
public void initialize(URL url, ResourceBundle rb) {

    //Al cargar la GUI, se mostrará el primer vehículo en la base de datos (Moto).
    Object object[] = new DBHandler().toArray();

    for (Object o : object) {

        if (o instanceof Moto) {
            mostrarMoto((Moto) o);
            break;
        }
    }
}

```

Imagen: Método initialize clase VistaMotoController. Java

○ Paquete dB

Contiene la clase DBHandler.java, la cual gestiona todas las operaciones con la base de datos.

En este paquete también se encuentra la base de datos. Es el archivo vehiculo.yap.

```
private void crearConexion() {

    //Ruta relativa al fichero de la BD
    final File path = new File("DB");
    final File db4o = new File(File.separator + "vehiculo.yap");

    try {

        if (!path.exists()) { //Si el directorio no existe, se crea
            path.mkdir();
        }

        //El método openFile abre un OobjectContainer en la BD para uso local
        db = Db4oEmbedded.openFile(Db4oEmbedded.newConfiguration(), path.getPath() + db4o.getPath());
    } catch (Db4oException e) {

        String mensaje = "Se arrojó el siguiente error abriendo la base de datos: "
            + "\n " + e;
        JOptionPane.showMessageDialog(null, mensaje, "Error en la base de datos", JOptionPane.ERROR_MESSAGE);
    }
}
```

Imagen: Método crearConexion clase DBHandler

○ Paquete enumerations

Contiene los tipos enumerados cmb.java, crnt.java, tmn.java y tp.java. Se utilizan para asignar los valores de algunos de los atributos de los objetos generados.

```
/**
 * Tipo de combustible
 *
 * @author FHueso
 */
public enum cmb {

    DIESEL, GASOLINA;
}
```

Imagen Enumeration cmb.java

○ Paquete images

Contiene todos los archivos de imagen necesarios para la construcción de la GUI; incluido el icono de la aplicación.

○ Paquete main

Contiene la clase Gestion_vehiculos.java. Se trata de la clase main y es la encargada de poner en marcha la aplicación, pasando el control a la vista VistaPrincipal.fxml y su clase controlador.

```
@Override
public void start(Stage stage) throws IOException {

    //Cargador para la vista principal.
    Parent root = FXMLLoader.load(getClass().getResource("/viewMain/VistaPrincipal.fxml"));
    Scene scene = new Scene(root);

    //Icono ventana
    stage.getIcons().add(new Image("/images/fondo_coche.jpg"));

    stage.setTitle("Gestion de vehículos");
    stage.setScene(scene);
    stage.show();

    //La ventana no se podrá redimensionar.
    stage.setResizable(false);
}
```

Imagen: Método start clase Gestion_Vehiculos

○ Paquete model

Contiene las clases encargadas de modelar los objetos que se guardarán en la base de datos.

```
public Bici(tmn tamaño, tp tipo, String marca, String modelo, String color, String annio, String id, String txtArea) {

    super(marca, modelo, color, annio, id, txtArea);

    this.tamaño = tamaño;
    this.tipo = tipo;
}
```

Imagen: Constructor clase Bici

- **Paquete viewBici**

Contiene el archivo VistaBici.fxml. Este archivo con estructura XML es generado por Scen Builder al crear una escena. Sus componentes y eventos son gestionados por la clase VistaBiciContoller.java

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Instituto FOC
Grado superior DAM
Proyecto creado para la certificación Cimientos de Java, incluido en el módulo de programación. -->
<!-- Este diseño corresponde a la vista bici. Ha sido generado con Scene Builder -->
```

Imagen: Instrucción de procesamiento archivo XML VistaBici.fxml

- **Paquete viewCoche**

Contiene el archivo VistaCoche.fxml. Este archivo con estructura XML es generado por Scen Builder al crear una escena. Sus componentes y eventos son gestionados por la clase VistaCocheContoller.java

```
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>
```

Imagen: Sección import archivo VistaCoche.fxml

- **Paquete viewMain**

Contiene el archivo VistaMain.fxml. Este archivo con estructura XML es generado por Scen Builder al crear una escena. Sus componentes y eventos son gestionados por la clase VistaMainContoller.java

```
<VBox alignment="CENTER" prefHeight="93.0" prefWidth="252.0">
  <children>
    <Label alignment="CENTER" prefHeight="94.0" prefWidth="251.0" text="Gestión de vehiculos" textAlignment="CENTER" textFill="WHITE">
      <font>
        <Font name="System Bold" size="24.0" />
      </font>
    </Label>
  </children>
</VBox>
```

Imagen: Etiquetas panel VBox con etiqueta label archivo VistaMain.fxml

- **Paquete viewMoto**

Contiene el archivo VistaMoto.fxml. Este archivo con estructura XML es generado por Scen Builder al crear una escena. Sus componentes y eventos son gestionados por la clase VistaMotoContoller.java

```
<RadioButton fx:id="radioAMoto" mnemonicParsing="false" style="-fx-background-color: white;" text="A">
  <HBox.margin>
    <Insets right="15.0" top="15.0" />
  </HBox.margin>
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
  <toggleGroup>
    <ToggleGroup fx:id="groupCarnet" />
  </toggleGroup>
</RadioButton>
```

Imagen: Etiquetas RadioButton archivo VistaMain.fxml