

Projecte – TruckDB (Oracle SQL Edition)



Nom i Cognoms: Radostin Valeriev Ivanov

Curs i Grup: 1r DAM A / 2023-24

Mòdul i Professor: MP03 – Programació / Santi Sabaté Sanz

Índex

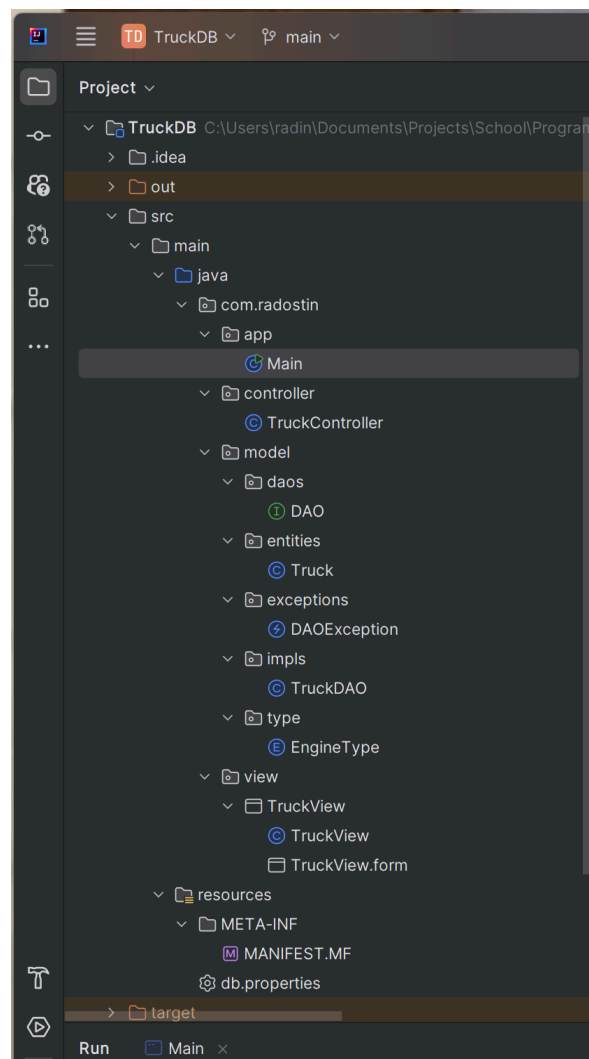
Introducció	3
La estructura dels paquets i fitxers de l'aplicació.	4
El paquet “app” i les seves classes	5
El paquet “controller” i la seva classe de controlador	7
El paquet “model” i les seves classes	7
El paquet “view” i la seva finestra	7
Conclusións	8

Introducció

- Aquest projecte tracta sobre una aplicació, molt pareguda a una base de dades sobre Camions i Conductors. Per això, fem utilitzar el patró de arquitectura MVC, que fer tot en un.

La estructura dels paquets i fitxers del projecte.

- Com ha de ser, amb el patró MVC ampliat que el anterior. Els paquets extra que hem posat dintre de model són especials per aquest projecte:
 - **daos** - Dintre tindrem la interfície que tractarà sobre Data Access Object.
 - **entities** - Dintre tindrem el nostre fitxer POJO, el de sempre, però canviat per a les necessitats per la BDD.
 - **exceptions** - Dintre tindrem totes les excepcions per al programa
 - **impls** - Dintre tindrem el tractament de totes les dades per a que es envien a la BDD.
 - **type** - El nostre tipo de enumeració sobre tipos de motors, que servirà per al dropdown de SwingUI que es anomena (JComboBox).



El paquet “app” i les seves classes

- En la classe “Main” posarem el nostre fitxer Main, que s'utilitza per a poder executar, visualitzar i utilitzar l'aplicació.
- Dintre especificarem el fitxer DAO, Vista i Controlador, per a que a l'execució ens pot funcionar correctament.

```
Main.java x
1  package com.radostin.app;
2
3  import com.radostin.controller.TruckController;
4  import com.radostin.model.exceptions.DAOException;
5  import com.radostin.model.impls.TruckDAO;
6  import com.radostin.view.TruckView;
7
8  public class Main {  Radostin Ivanov
9      public static void main(String[] args) {  Radostin Ivanov
10         try {
11             TruckDAO truckDAO = new TruckDAO();
12             TruckView truckView = new TruckView();
13             new TruckController(truckDAO, truckView);
14
15             truckView.setVisible(true);
16         } catch (DAOException e) {
17             e.printStackTrace();
18         }
19     }
20 }
```

El paquet “controller” i la seva classe de controlador

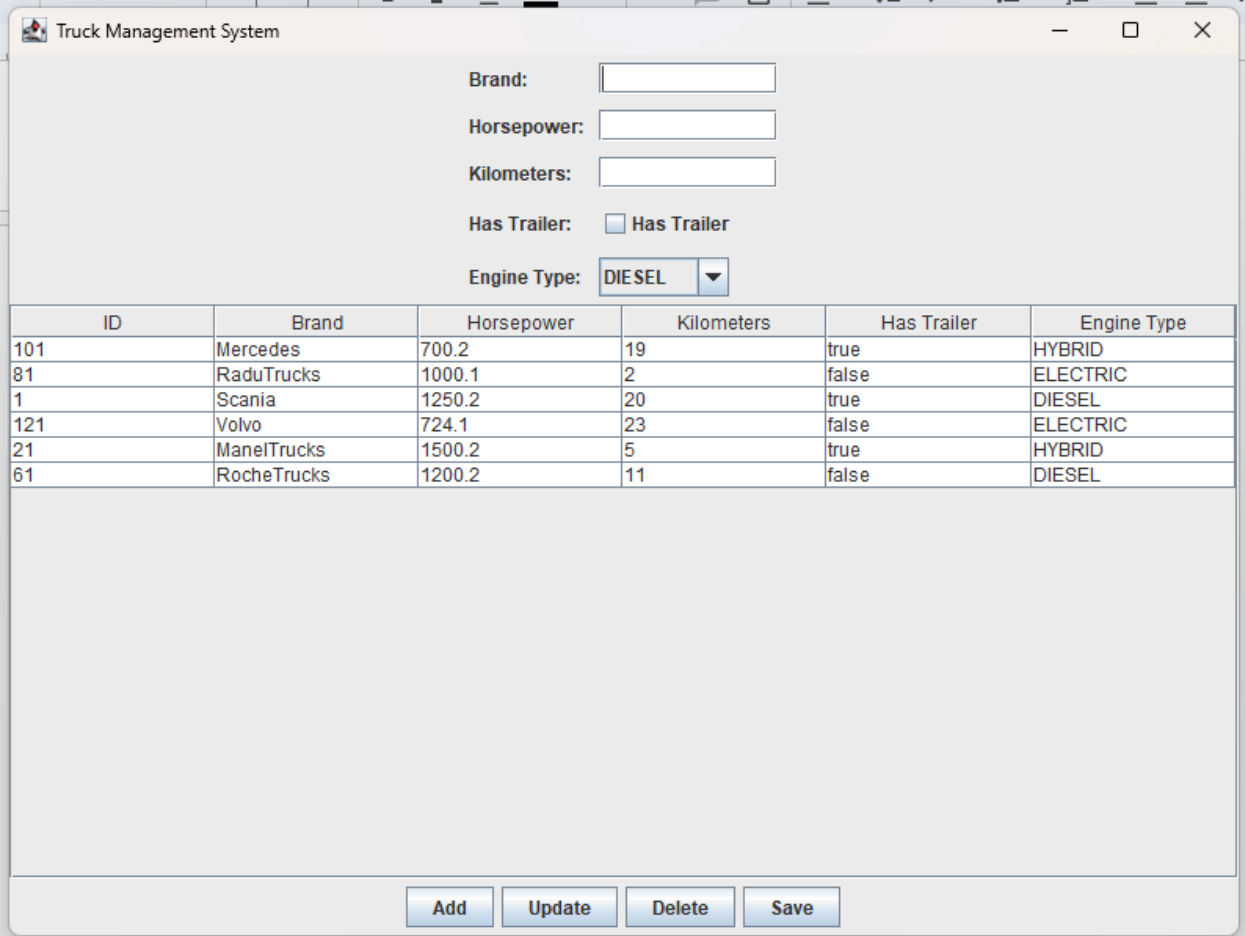
- El fitxer de controlador ens serveix per a:
 - Tractament de la expressió regular
 - Lògiques per insertar, modificar, borrar i guardar les dades, cridant funcions de PL/SQL per la BDD.
 - Funcions (void) que tracta per els objectes dels camps que introduïm dades.

El paquet “model” i les seves sub paquets necessaris.

- El paquet daos:
 - Dins tenim la interfície DAO que tracta sobre els excepcions per al fitxer TruckDAO per si algo ens anirà malament
- El paquet entities:
 - Dins tenim el nostre POJO de Trucks
- El paquet exceptions:
 - Dins tenim el tractament d'excepcions personalitzades per al fitxer TruckDAO si alguna cosa no ens va bé.
- El paquet impls:
 - Dins tenim tots els funcions per al SQL quan afegim, modifiquem, borrem o guardem objectes.
- El paquet type:
 - Dins tenim el tipo de enumeració que ens tracta sobre el dropdown per seleccionar el tipo de motor amb JComboBox.

El paquet “view” i la seva finestra

- El fitxer “view” és l'arxiu que:
 - Dissenya i mostra tota la vista
 - Els camps
 - Els botons
 - Tractament dels camps i els botons



The screenshot shows a window titled "Truck Management System". It contains a form with the following fields:

- Brand:
- Horsepower:
- Kilometers:
- Has Trailer: ☐ Has Trailer
- Engine Type:

Below the form is a table with the following data:

ID	Brand	Horsepower	Kilometers	Has Trailer	Engine Type
101	Mercedes	700.2	19	true	HYBRID
81	RaduTrucks	1000.1	2	false	ELECTRIC
1	Scania	1250.2	20	true	DIESEL
121	Volvo	724.1	23	false	ELECTRIC
21	ManelTrucks	1500.2	5	true	HYBRID
61	RocheTrucks	1200.2	11	false	DIESEL

At the bottom of the window are four buttons: Add, Update, Delete, and Save.

SQL

Creació de la taula

```
CREATE TABLE trucks (  
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    brand VARCHAR2(100) NOT NULL,  
    horsepower NUMBER NOT NULL,  
    kilometers NUMBER NOT NULL,  
    hasTrailer NUMBER(1) NOT NULL,  
    engineType VARCHAR2(50) NOT NULL  
);
```

Procedures per PL/SQL

Per insertar

```
CREATE OR REPLACE PROCEDURE insert_truck(  
    p_brand IN VARCHAR2,  
    p_horsepower IN NUMBER,  
    p_kilometers IN NUMBER,  
    p_hasTrailer IN NUMBER, -- Assuming 1 for true, 0 for false  
    p_engineType IN VARCHAR2  
) IS  
BEGIN  
    INSERT INTO trucks (brand, horsepower, kilometers, hasTrailer, engineType)  
    VALUES (p_brand, p_horsepower, p_kilometers, p_hasTrailer, p_engineType);  
END;
```


Per modificar

```
CREATE OR REPLACE PROCEDURE update_truck(  
    p_id IN NUMBER,  
    p_brand IN VARCHAR2,  
    p_horsepower IN NUMBER,  
    p_kilometers IN NUMBER,  
    p_hasTrailer IN NUMBER,  
    p_engineType IN VARCHAR2  
) IS  
BEGIN  
    UPDATE trucks  
    SET brand = p_brand,  
        horsepower = p_horsepower,  
        kilometers = p_kilometers,  
        hasTrailer = p_hasTrailer,  
        engineType = p_engineType  
    WHERE id = p_id;  
END;
```

Per borrar

```
CREATE OR REPLACE PROCEDURE delete_truck(  
    p_id IN NUMBER  
) IS  
BEGIN  
    DELETE FROM trucks  
    WHERE id = p_id;  
END;
```

Conclusións

- Aquest projecte era complicat com l'anterior i me ha costat temps de descobrir, però hem après ja últimament de com fer-ho a una base de dades Oracle local o remota que és molt més guapo.