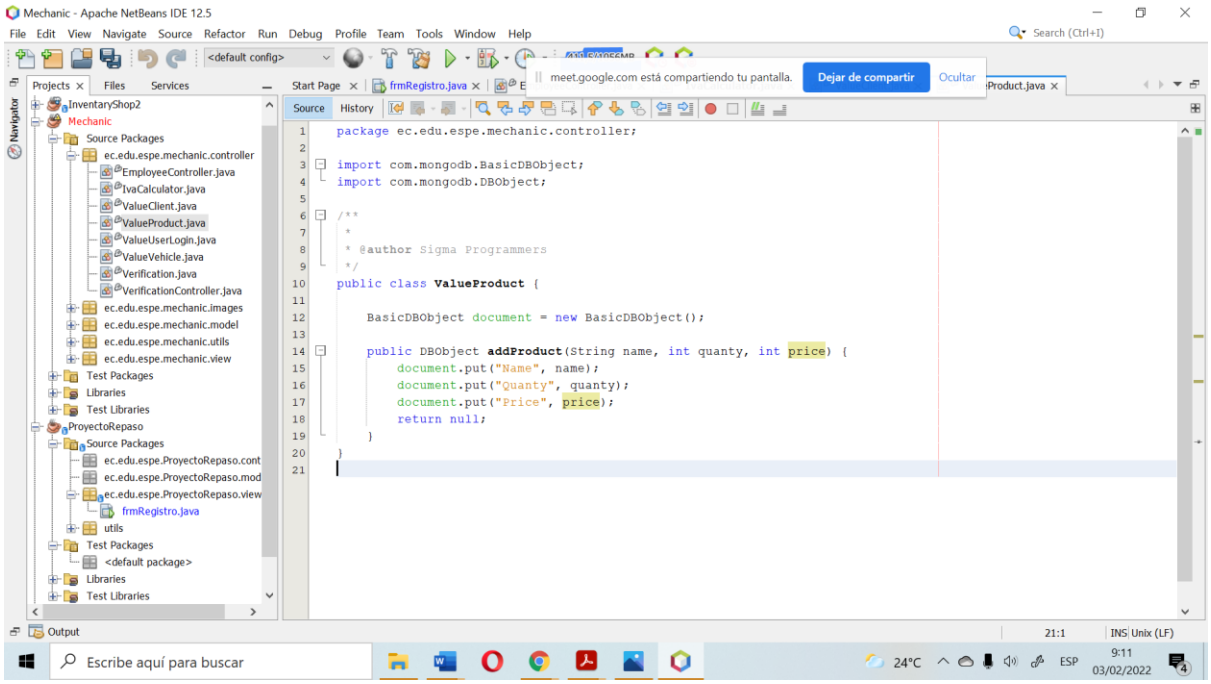
# WS32 - GROUPS

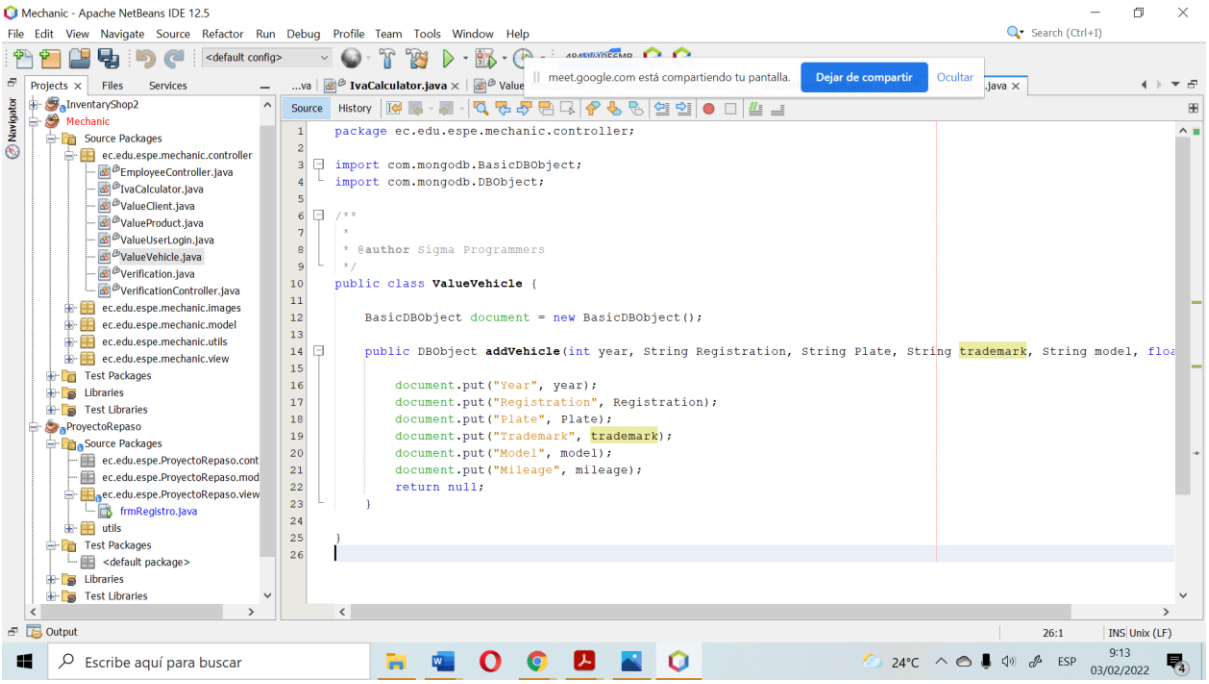
## DATE: 03th february 2021

**TEAM 5:**
Alvarez Michelle
Correa Kerly
Eivar Jaime
Garcia Mayerly
Teca Camila
Teran Melanie

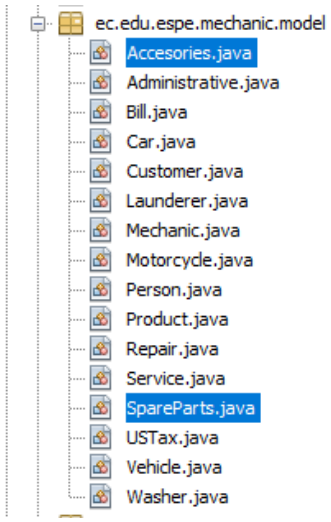
```
BasicDBObject document = new BasicDBObject();

public DBObject addVehicle(int Name, String LastName, String Phone, String Code) {
```

The name is declared as a int, it should be String - Bad quality code






Incorrect class names. Classes are written in the singular:

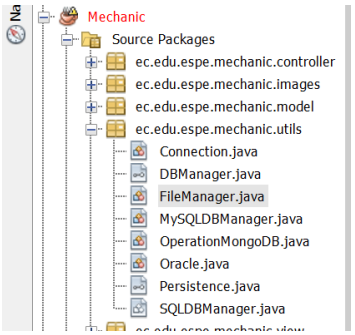



They are not names for the buttons:

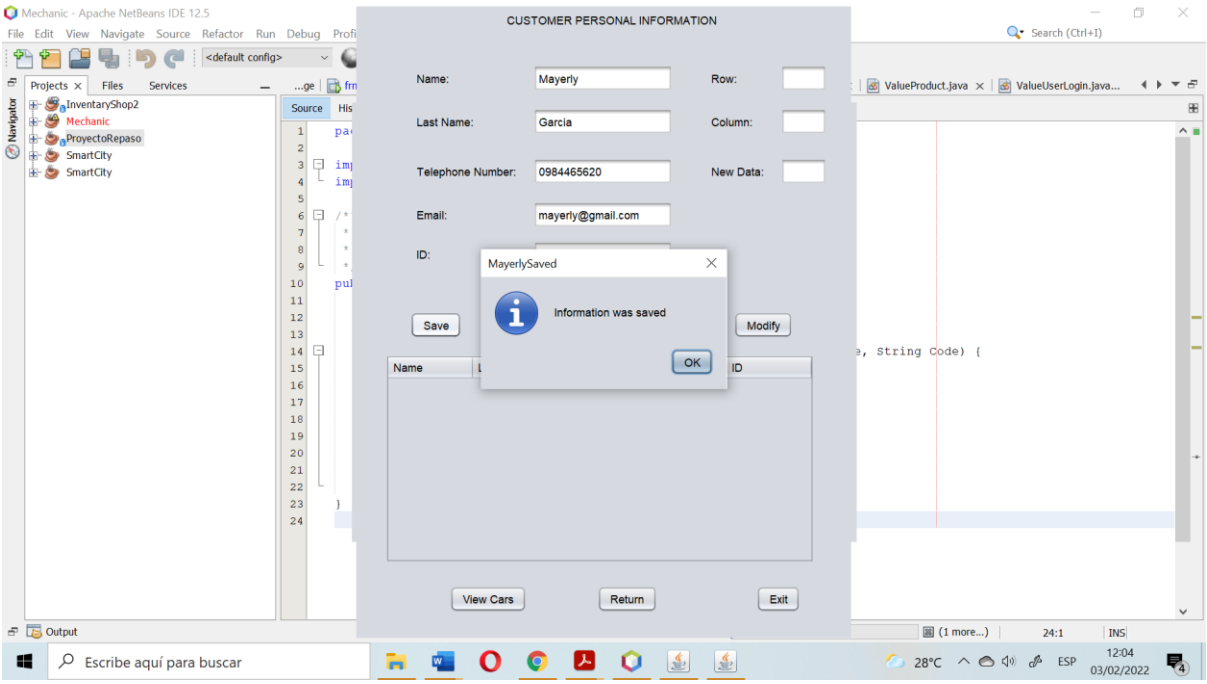
```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
        // TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

        FrmEmployers frmEmployers = new FrmEmployers();
        frmEmployers.setVisible(true);
        dispose();

// TODO add your handling code here:
    }
```

```java
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    FrmEmployers frmEmployers = new FrmEmployers();
    frmEmployers.setVisible(true);
    dispose();
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
    // TODO add your handling code here:
}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

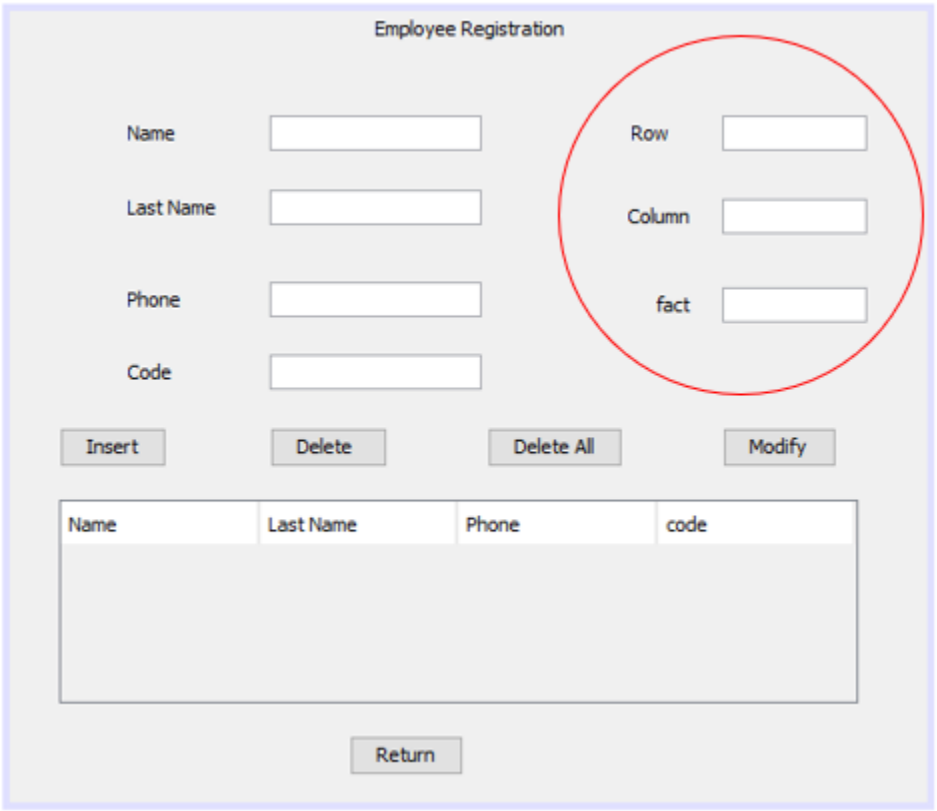
In the utils package there should be one class to the database, but no more than two classes connecting to the same database. (**Single Responsibility** )




**The Interface segregation principle** is broken because the clients of the program should know what data is entered and which methods should actually be used.
Added or saved data is not displayed in the table




**Single Responsibility Principle** broken because has not relation with the data of the class:




**Single Responsibility Principle** broken because the "add vehicle" object is not related to the "EmployeeController" class.

```java
package ec.edu.espe.mechanic.controller;

import com.mongodb.BasicDBObject;
import com.mongodb.DBObject;

/**
 *
 * @author Sigma Programmers
 */
public class EmployeeController {

    BasicDBObject document = new BasicDBObject();

    public DBObject addVehicle(int Name, String LastName, String Phone, String Code) {

        document.put("Name", Name);
        document.put("LastName", LastName);
        document.put("Phone", Phone);
        document.put("Code", Code);

        return null;
    }
}
```

Bad declaration of the person Method of the Person class, missing to declare id

```java

    public Person(int id, String name, String lastName, String plate) {
        this.id = id;
        this.name = name;
        this.lastName = lastName;
        this.plate = plate;
    }

```

```
45
        Person person = new Person(name,lastName,plate);
47
48      String saveData = person.toString();
49
```

Bad names for button functions, FrmBill does not perform any function and comments not deleted

```
Source  Design  History
300     private void txtNameActionPerformed(java.awt.event.ActionEvent evt) {
301         // TODO add your handling code here:
302     }
303
304     private void discountActionPerformed(java.awt.event.ActionEvent evt) {
305         // TODO add your handling code here:
306     }
307
308     private void otherPriceKeyTyped(java.awt.event.KeyEvent evt) {
309         // TODO add your handling code here:
310         char valid = evt.getKeyChar();
311
312         if (Character.isLetter(valid)) {
313             getToolkit().beep();
314
315             evt.consume();
316
317             JOptionPane.showMessageDialog(rootPane, "Input ONLY numbers");
318         }
319     }
320
321     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
322         System.exit(0);
323         // TODO add your handling code here:
324     }
325
326     private void btnReturnActionPerformed(java.awt.event.ActionEvent evt) {
327         FrmEmployers frmEmployers = new FrmEmployers();
328         frmEmployers.setVisible(true);
329         dispose();
330
331         // TODO add your handling code here:
332     }
```

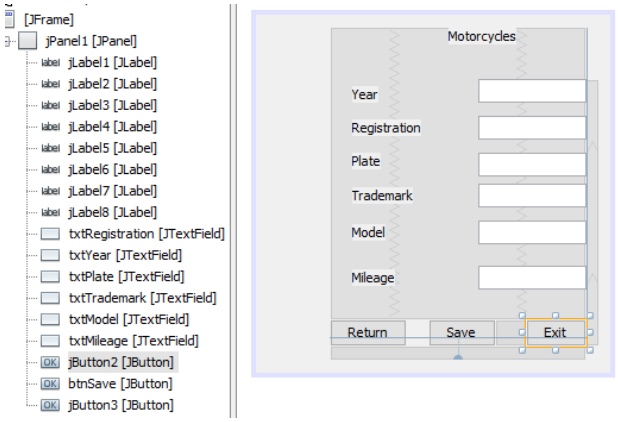
FrmCar.java × | FrmCarManage.java × | FrmCustomerRecord.java × | FrmCustumerManage.java × | Frn

Design History

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ec.edu.espe.mechanic.view;
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ec.edu.espe.mechanic.view;
```

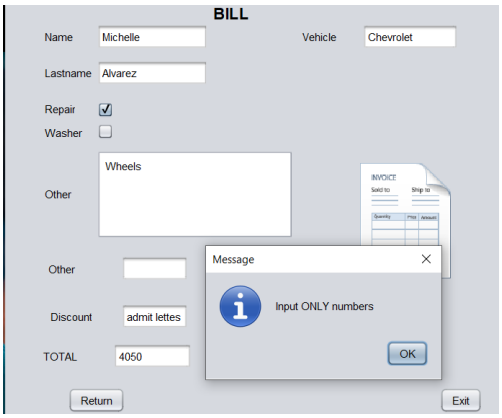
Unnecessary comments (FrmCarManage and FrmWorkManage )



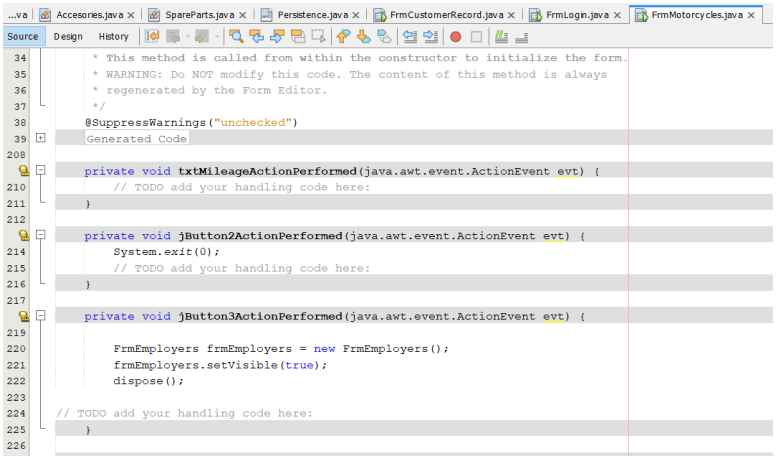

Bad names for Return and Exit buttons (FrmMotorcycles)

```
public class FrmWorkerRecord extends javax.swing.JFrame {

    DefaultTableModel modelo;
```

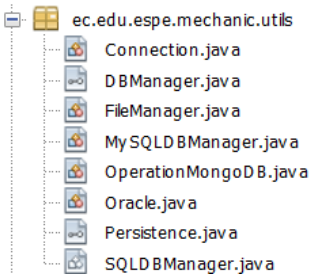
Table name in spanish (FrmWorkerRecord)




The FrmBill does not follow the **Liskov Substitution Principle** because it does not represent a window or public methods into the program. The are two fieds call “Other” and in one of them it is not clear which is the information that you have to righ. The field discount and TOTAL are not clear. They must be calculated into the program.




In the FrmMotorcycle the are some buttons without name so it is difficult to understand how the program works. **Unclean code.**




Each class should be responsible for only one part of the system's functionality. We can see there are many classes that allow the connection to the database, so it is not clear which class do a specific job. So in this part of the program does not follow the **Single Responsibility Principle.**