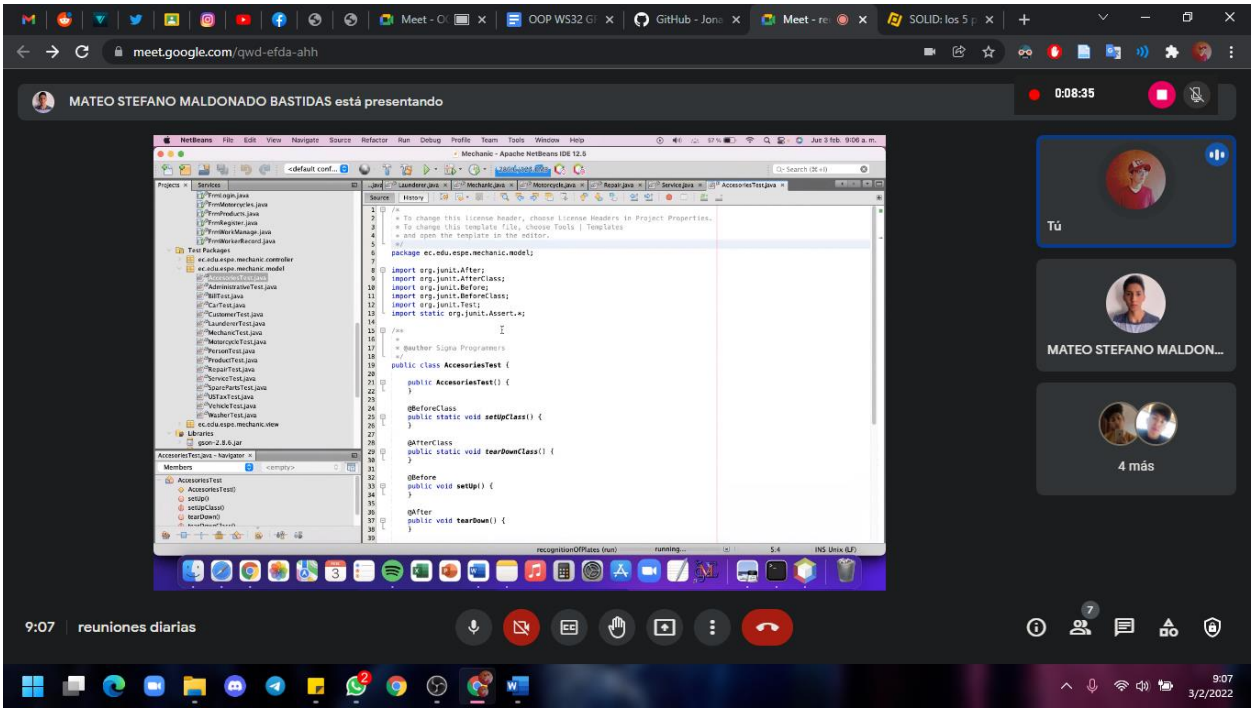
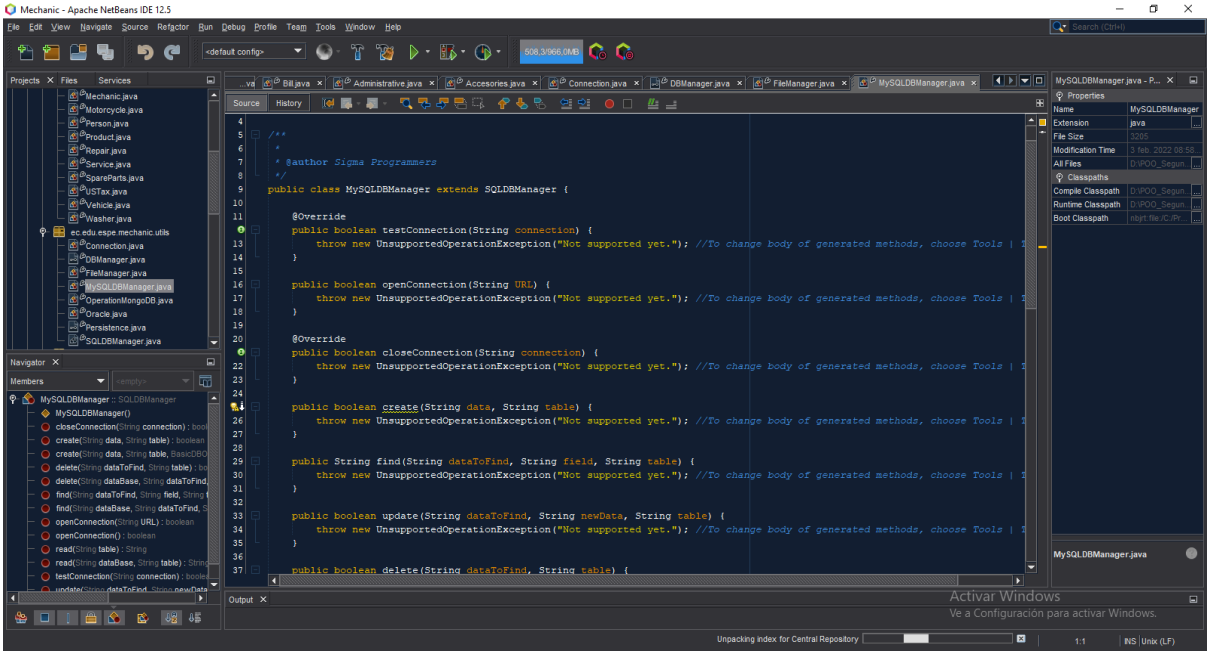
WS32 - SOFTWARE PRINCIPLES GROUPS

DATE: 03th february 2021

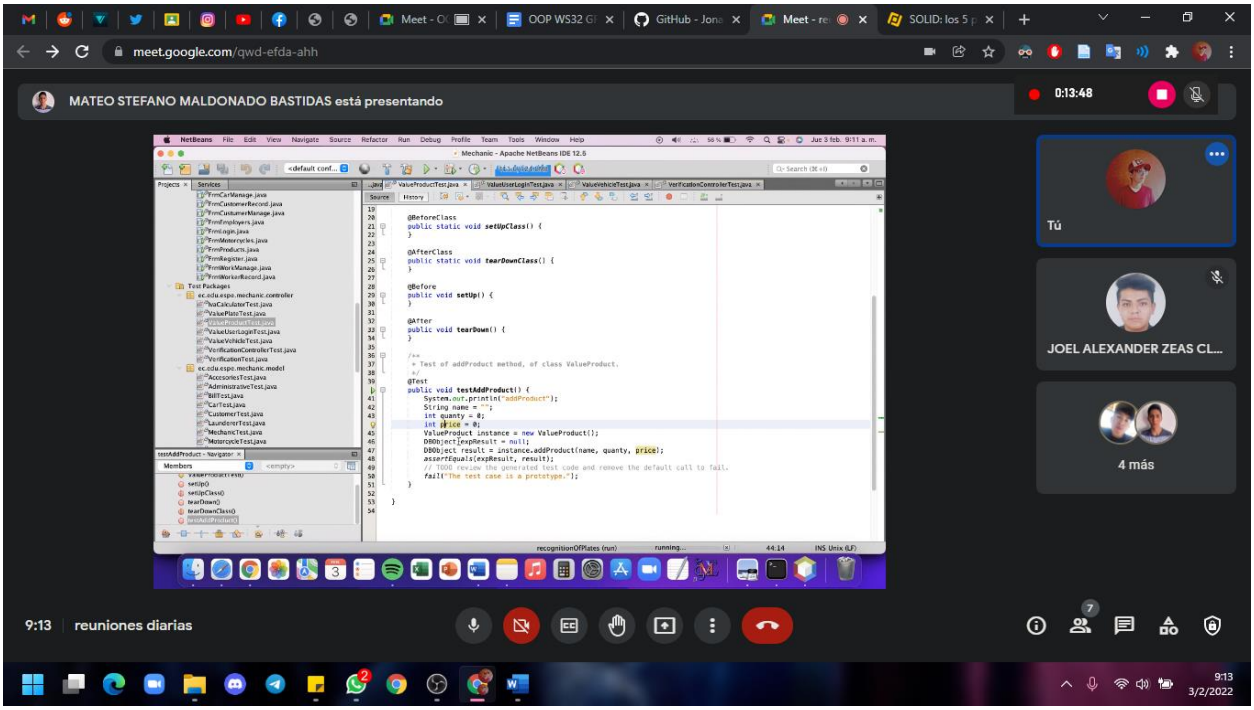
TEAM 1:
Salma Villegas
Alexander Ruano
Daniel Lincango
Mateo Maldonado
Leandro Quinga
Joel Zeas

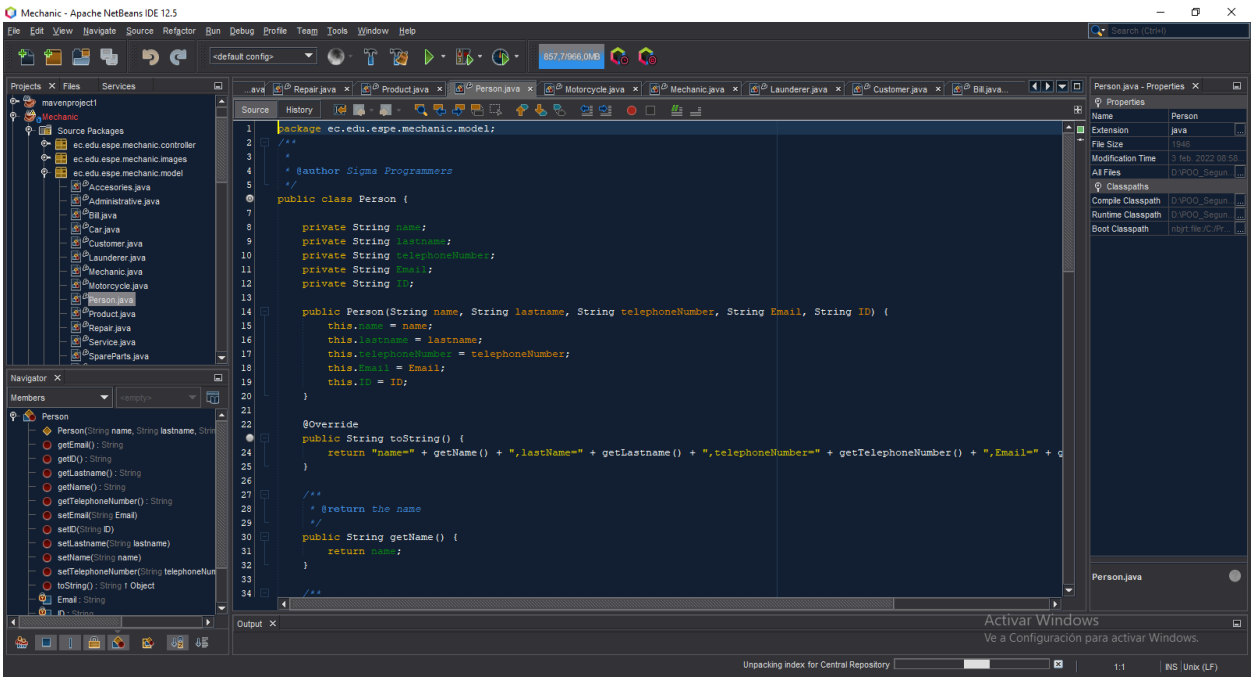


Innecesary comments, unclean code

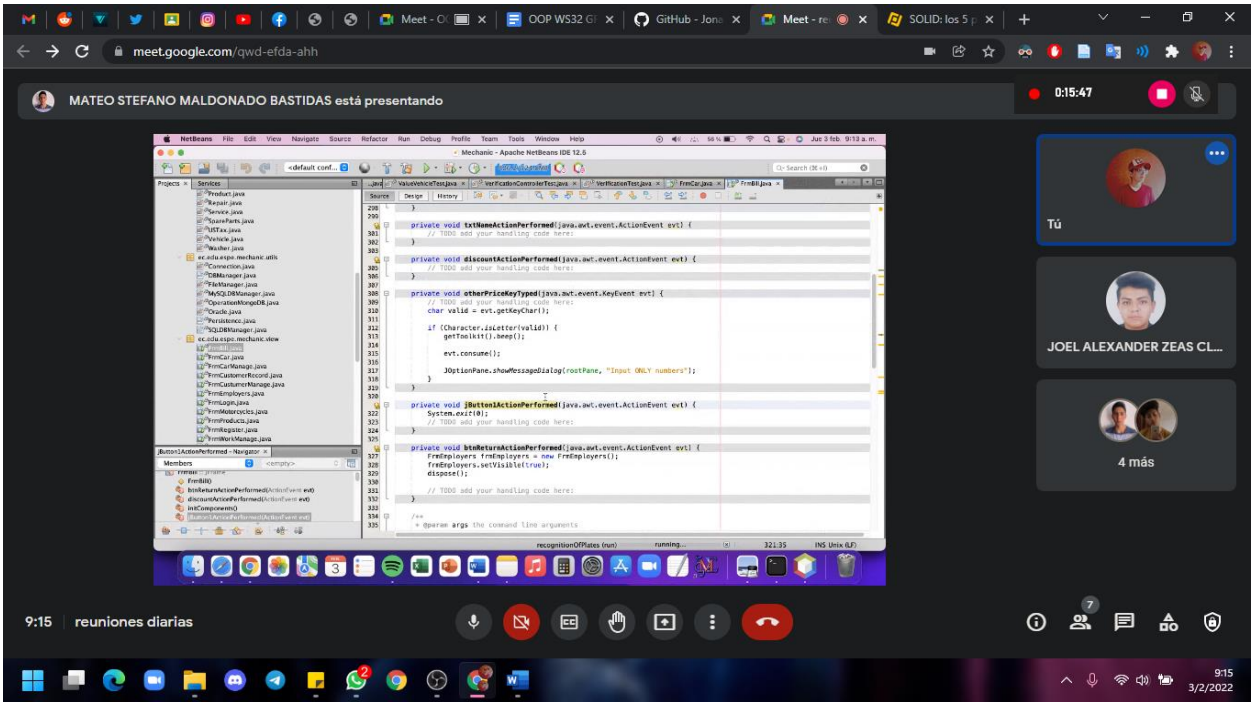


Unclean Code

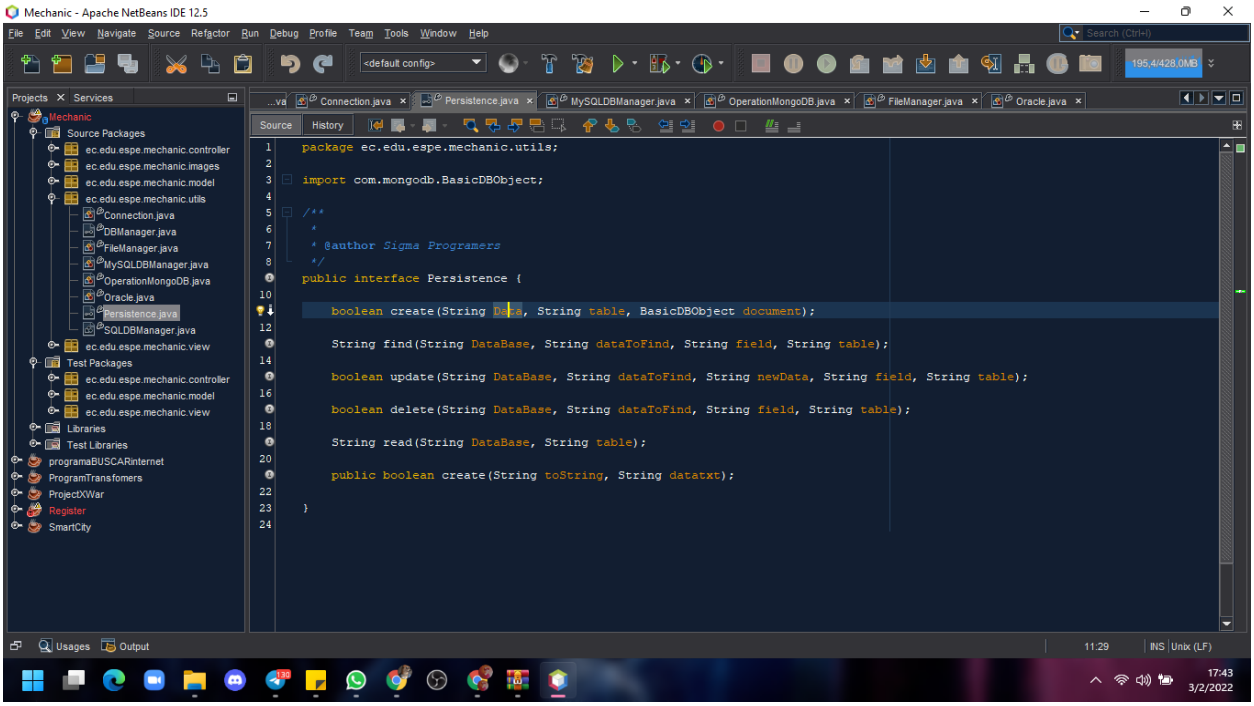




do not use the format Camelcase

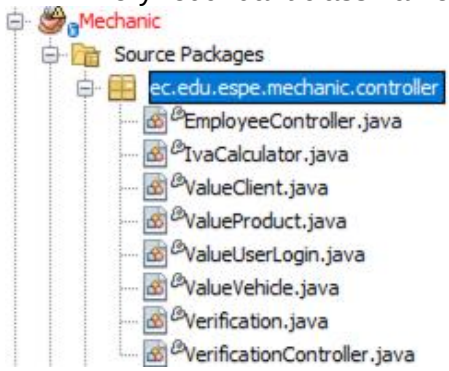


Refactor jbutton



Don't use camelcase

- Very redundant class names, difficult to differentiate



- **Single ResponsibilityPrinciples:** there is no relationship between the “document” variable and the “addVehicle” class. It is also a very general term.

```
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;
}
```

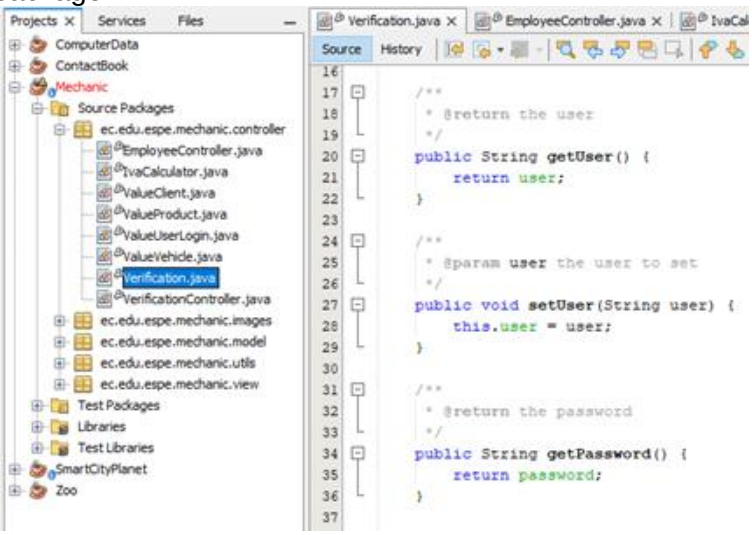
- Use an else variable that replaces the return and returns an actual data

```java
public class ValueClient {

    public boolean create(Customer customer) {
        boolean created = false;
        String personData;

        Persistence persistence;

        persistence = new FileManager();

        if (persistence.create(customer.toString(), "Data.txt")) {
            JOptionPane.showMessageDialog(null, customer + "was saved");
        }
        return created;
    }
}
```

- Use lower case for variables

```java
public DBObject addVehicle(int year, String Registration, String Plate, String trademark, String model, float mileage) {

    document.put("Year", year);
    document.put("Registration", Registration);
    document.put("Plate", Plate);
```

- Getters and setters must be programmed in the “.model” package, no in the “.controller” package



- **Open/Closed Principle:** Very general variable and method names, also the user validation programming is usually done in the .view package and it must be open to extend it

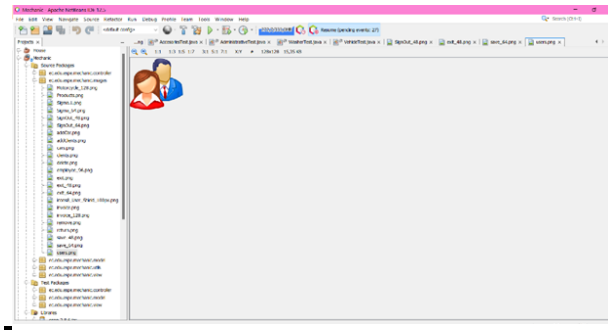
```java
public boolean login(Verification verification, String user, String password) {

    boolean logged = false;
    String readLine;
    Persistence persistence;
    persistence = new FileManager();
    readLine = persistence.read("", "Users.json");

    Gson gson = new Gson();

    verification = gson.fromJson(readLine, Verification.class);

    if (user.isEmpty() || password.isEmpty()) {
        JOptionPane.showMessageDialog(null, "FILL ALL THE FIELDS");
    } else if (user.equals(verification.getUser()) == false && password.equals(verification.getPassword()) == false) {
```

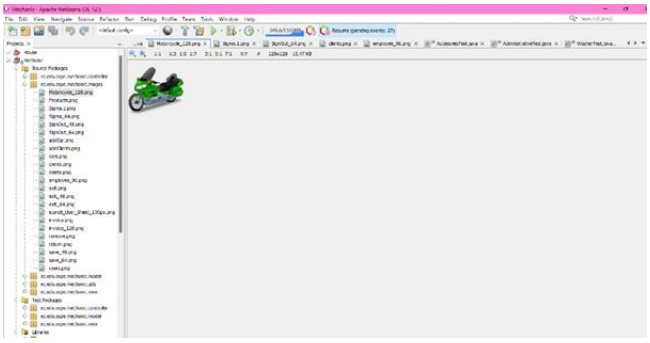
Source Packages

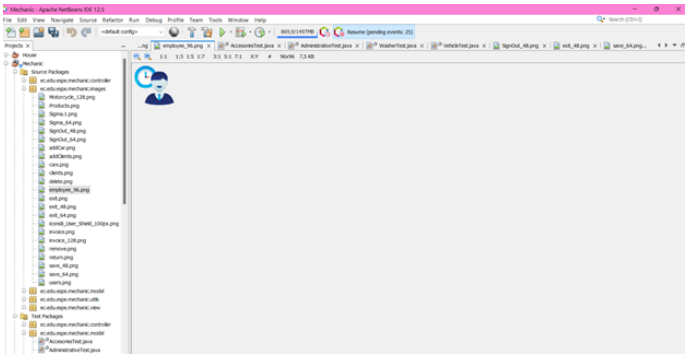
Package: ec.edu.espe.mechanic.images

Nothing to mention, each class has only one image.

Used Principles: Single Responsibility Principle (each class have only one image)







Test Packages

Package: ec.edu.espe.mechanic.model

Unnecessary comments, unclean code,

Used Principles: Single Responsibility Principle (This package is being used for testing only, so each class has a unique reason)