

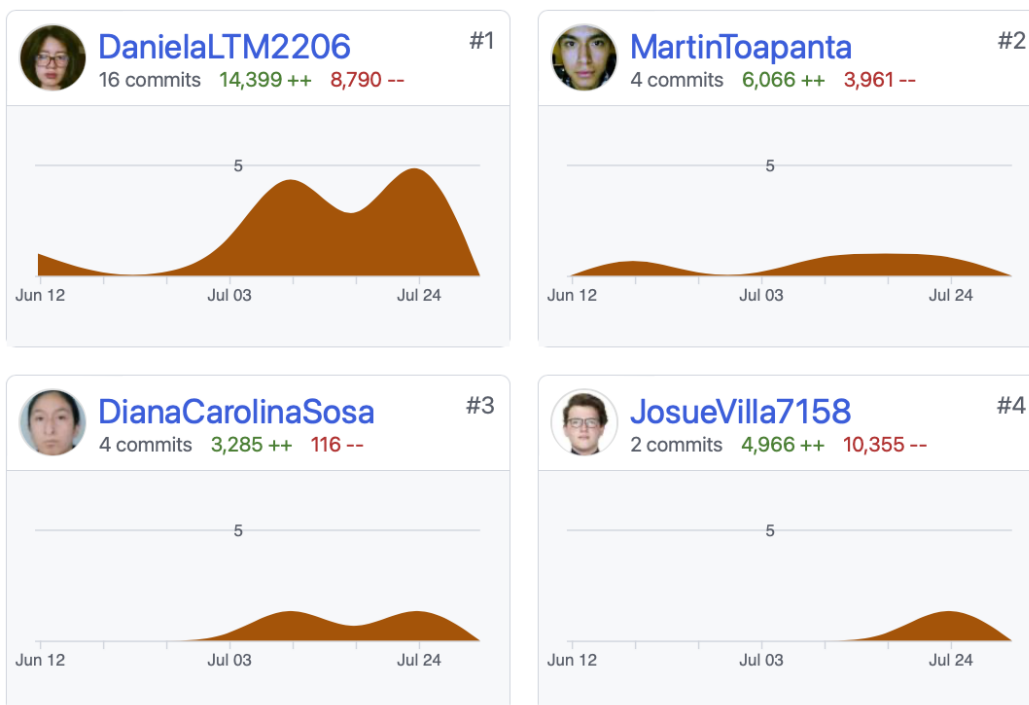
**TEAM 07 MyWayCode**  
**BettaCoders(Luis Burbano)**

**INSPECTOR: TEAM 01**

**Git Hub:** <https://github.com/DanielaLTM2206/TWC-Store-System-.git>

**Proyect:** TWC-Store-System-

1. Sosa Romero Diana Carolina 70%
2. Tituaña Moreno Daniela Lissette 100%
3. Toapanta Vásquez Martin Honorio 80%
4. Villavicencio Campoverde Bolivar Josue 70%



**Rubric:**

Requirements	10/10
UML Class Diagram	7.8/10
Clean Code	9.6/10
Db (MongoDb Atlas)	9.5 /10
GUI	7/10
Tables	5/10
Forms	22/30
UnitTest	0.2/10
<hr/>	
TOTAL	71.1/100

Proof:

## Requirements (Features/items)

---

### Requerimientos:

- El sistema debe contar con una sección de ingreso de usuarios y personal.
- El sistema tendrá la opción de agregar un nuevo usuario cuando sea necesario.
- El sistema mostrará el tiempo de vida de cada producto cuando se registren, teniendo así un producto perfecto para la venta.
- El sistema debe contar con el registro y ubicación de cada producto.
- El sistema debe tener un registro constante de cada venta realizada.
- El sistema debe aceptar devoluciones de ciertos productos por mal estado o por caducidad del producto.
- El sistema debe mostrar la cantidad para que sepa la cantidad necesaria para reponerlo.
- El sistema debe mostrar los horarios de mayor venta para una mejor organización dentro del local.
- El sistema debe mostrar una lista con información detallada sobre los usuarios del producto.
- El sistema debe mostrar información detallada sobre los productos.

- A. Login
- B. Add user
- C. Registration product
- D. Accept refund of product
- E. Stock of product
- F. Show the quantity to replenish the product
- G. Show horary
- H. List users
- I. List products

### UML Class Diagram

- Classes -2
- Attributes -2
- Methods -2
- Relations -2
- Abstract Classes/Interfaces 2
- Abstract methods 2
- Polymorphism (overridden methods) 2
- Inheritance 2
- MVC -> packages model (POJO classes), view(GUI), controller (Abstract C./Concrete C./Interfaces) 2
  - (getters and setters 9(-.01), labels(-0.1), panels(-0.1), no dependencies in the view package (0.1)

## GUI

### Splash Window

#### Authentication

2

(it should work)

#### Application (Menu)

2

(requirements)

(-0.2)

#### Forms

6

(requirements)

Navigability -

0.2

## Tables:

data come from DB

5

print data to printer

5

## Forms

at least, there must be 6 different forms, every form is worth 5 points

- Login Form
- 5 Forms for Information (CRUD operations, 4 business rules)

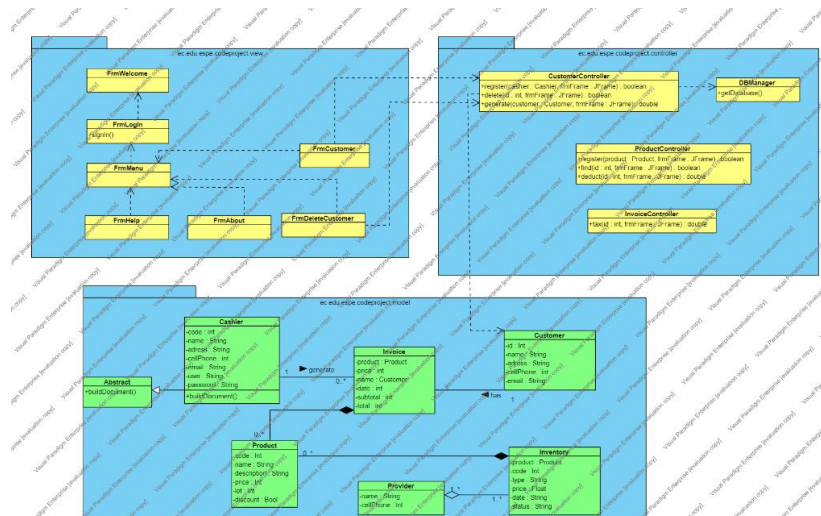
Every form

- Every input (JTextField) is validated
- There are different types of inputs (not everything is a JTextField) -1 for each JTextField
- Buttons' events, no code is here for CRUD operations or business rules, i.e., these functions will call some controller method, and will use some model class.

every mistake: -1

less than 5 forms -> /50%

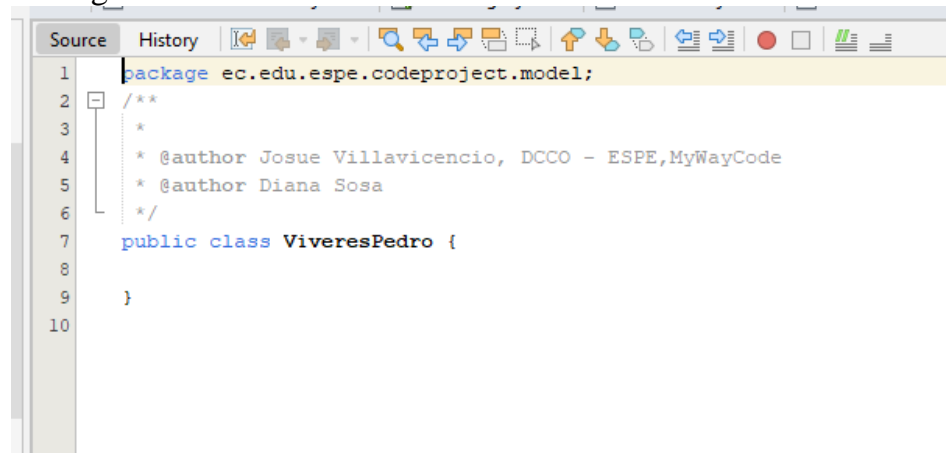
## Unit Test



## Clean Code

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    String Username="oop";  
    String Password="1234";  
  
    String Pass=new String(this.Password.getPassword());  
  
    if(txtUsername.getText().equals(Username)&&Pass.equals(Password)){
```

## Wrong name of button variables



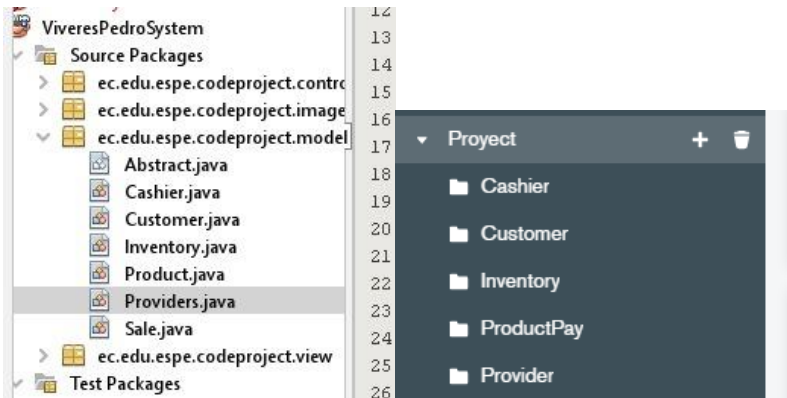
## classes created without using

```
250  
251  
252  
253 //          if (txtID.getText().length() >=20){  
254 //              lblIDError.setText("Max 10 digits");  
255 //              evt.consume();  
256 //          }  
257
```

## unused code

```
251  
252  
253 //          if (txtID.getText().length() >=20){  
254 //              lblIDError.setText("Max 10 digits");  
255 //              evt.consume();  
256 //          }  
257
```

## MONGO



## Authentication(With mongoDB not with burned data)

```

141     }
142
143     private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
144         String username="oop";
145         String password="1234";
146
147         String Pass=new String(this.Password.getPassword());
148
149         if(txtUsername.getText().equals(username)&&Pass.equals(password)) {
150
151             FrmMenu EP=new FrmMenu();
152             EP.setVisible(true);
153             dispose();
154
155         }
156         else{
157             JOptionPane.showMessageDialog(this,"Incorrect Username / Password");
158         }
159     }
160
161

```

```

Document doc;
    doc= createDBObject(cashier);
MongoDatabase userDB = DBManager.getDatabase();
MongoCollection<Document> col = userDB.getCollection("Cashier");

```

Provider

Id:

Full Name:

Company:

Cellphone:

Email: