“UNIVERSIDAD DE LAS FUERZAS ARMADAS”

ESPE

SOFTWARE ENGINEERING

OBJECT - ORIENTED PROGRAMMING

TOPIC:

WORKSHOP 30 - Project to review

TEACHER:

PHD. EDISON LASCANO

NRC:

4680

DATE:

July 25, 2022

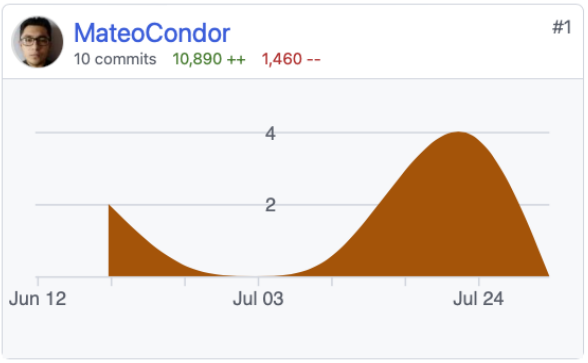
SEDE MATRIZ SANGOLQUÍ

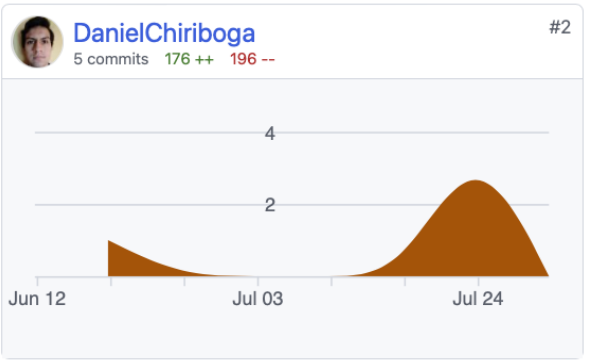
QUITO-ECUADOR

2022

TEAM 02 DAMAGE INSPECTOR: TEAM 03 Syntax Error (Granda Carlos)
GITHUB https://github.com/MateoCondor/PointOfSaleSystem
Project: Bakery point of sale system

1. Chavez Escudero Genaro David 0%
2. Chicaiza Ortiz Frank Sebastian 0%
3. Chiriboga Zurita Daniel Isai 80%
4. Condor Sosa Mateo Javier 100%





Rubric:

1. Requirements	9.5/10
2. UML Class Diagram	8.8/10
3. Clean Code	9.8/10
4. Db (MongoDb Atlas)	9.5/10
5. GUI	10/10
6. Tables	5/10
7. Forms	19 /30
8. Unit Tests	5/10

—----------

TOTAL:	76.6/100
GitHub:	90/100

Proof

1. Requirements (Features/items)
   -Cost
   -Payment
   -Worker
   -Product
   -Provider
   -Cash Register

REQUERIMIENTOS PARA EL SISTEMA DE PUNTO DE VENTA DE LA PANADERIA POMPEI

La panadería POMPEI ha registrado todos sus movimientos en los Libros de contabilidad de su negocio, pero no ha sido suficiente para llevar un control de calidad respecto de las finanzas y sus productos.

De este modo se necesita crear un sistema de punto de venta adecuado a la panadería para optimizar el tiempo con los siguientes requerimientos:

El usuario necesita que se registren los trabajadores de su negocio para que ellos manipulen el sistema y todo tenga un registro.

El usuario necesita que el sistema permita la creación de los diferentes productos de su negocio, para lo cual se establecerán parámetros según se cobre o no IVA. El usuario necesita clasificar sus productos de acuerdo a la Marca y Categoría, para una mejor búsqueda en el sistema.

El usuario necesita crear una sección para registrar la información el Cliente, ya que con esa información puede adecuar el servicio según las necesidades el Cliente

El usuario necesita una sección en el sistema que le permita registrar las ventas que se hacen a diario en el negocio, cada una con su respectivo número de factura, productos comprados y valor total a pagar, toda esta información es importante que se guarde al instante y de haber algún inconveniente con el registro poder anularlo.

El usuario necesita una sección en el sistema que le permita registrar la información de los Proveedores del negocio, ya que es de vital importancia esos datos para el abastecimiento del negocio, además el usuario a veces adeuda con el Proveedor por lo cual se necesita registrar el valor pagado y el valor adeudado al Proveedor.

El usuario necesita una sección en el sistema que le permita registrar los Pagos y Gastos del sistema, tales como Pagos al Proveedor, Gastos tales como servicios básicos, etc.

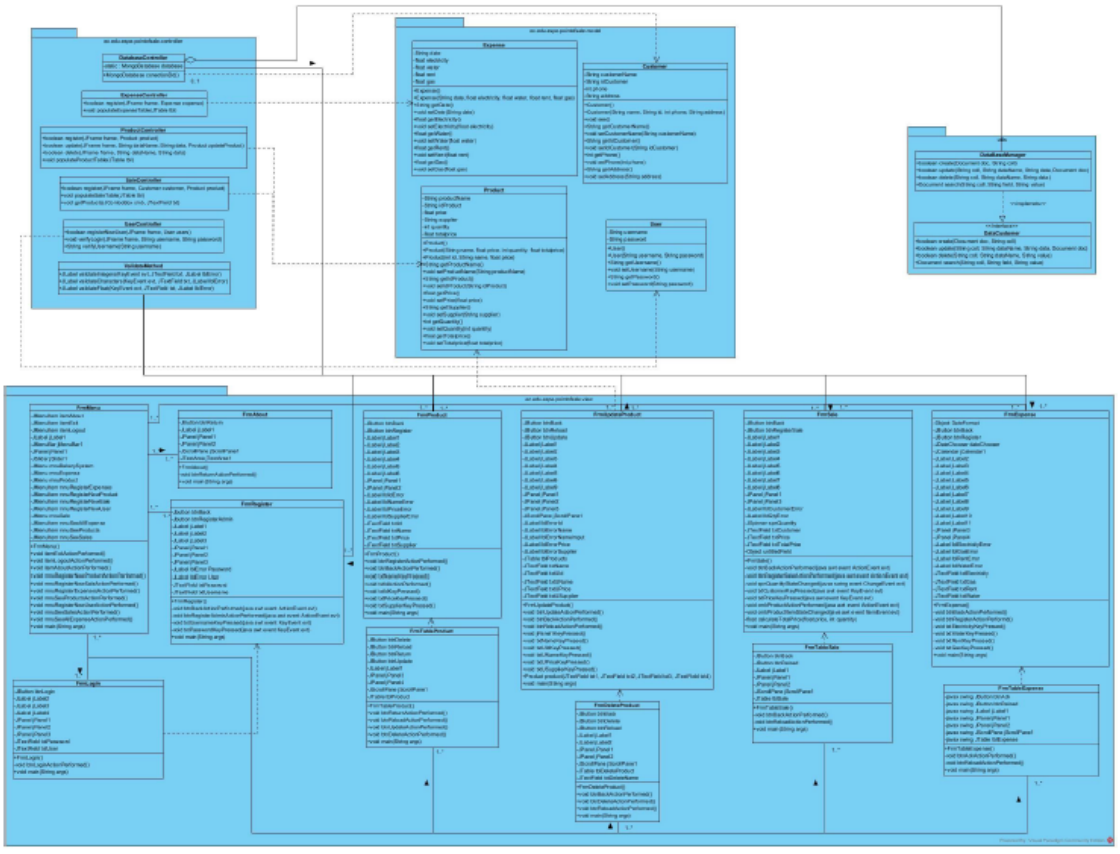
El usuario necesita una sección para la Caja del sistema, donde se registrarán la cantidad de dinero de los ingresos del negocio, los retiros que se hacen durante el día y los Pagos y Gastos del sistema.

El usuario necesita que cada Usuario cumpla una función indicada en el sistema, por lo cual se necesita privar de ciertas funciones a los Trabajadores, solo el Administrador/es tendrá acceso a todas las secciones del sistema

El usuario necesita una sección de Reportes de cada una de las secciones antes mencionadas ya que necesita la información a la mano de los Usuarios, Proveedores de las Compras y Ventas diarias por fechas y productos de acuerdo al stock,etc.

2.UML Class Diagram

- Classes
- Attributes
- Methods
- Abstract Classes/Interfaces 2/2
- Abstract methods 2/2
- Polymorphism (overridden methods) 2/2
- Inheritance 1 /2
- MVC-> packages model(POJO classes), view(GUI), controller(Abstract C./Concrete C./Interfaces) 1.8 /2



Observations

- Missing Inheritance
- In MVC have labels and panels

3.Clean Code

Pitfalls/Code Smells (0.1)

```java
public int getPhone() {
    return phone;
}

/**
 * @param phone the phone to set
 */
public void setPhone(int phone) {
    this.phone = phone;
}
```

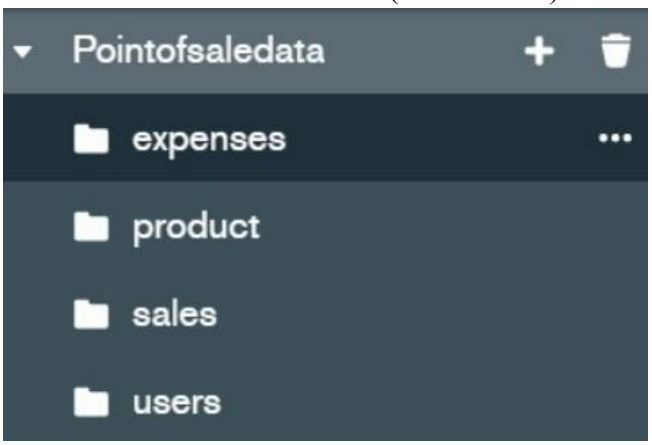
```java
public float getGas() {
    return gas;
}

/**
 * @param gas the gas to set
 */
public void setGas(float gas) {
    this.gas = gas;
}
```

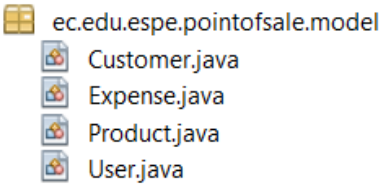
4. Db (MongoDb Atlas)

Database Name

Collection vs Model classes

attributes (collections) vs attributes (classes)



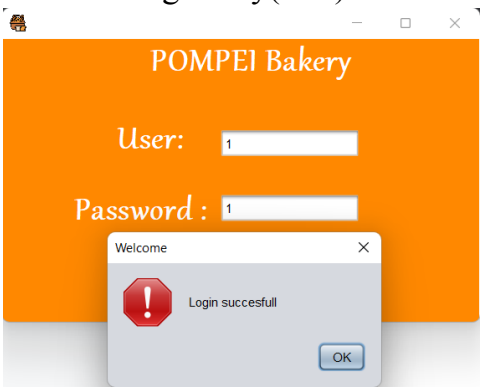


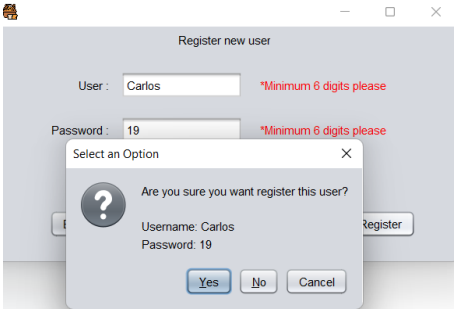
Observation:

You have changed a class

5. GUI

Splash Window
Authentication 2/2 (it should work)
Application (Menu) 2/2 (requirements) (-0.2)
Forms 6/6 (requirements)
Navigability(-0.2)





6. Tables

Date come from DB 5/5

```
_id: ObjectId('62de949ec0b4cb16731b4ecd')
idProduct: "190622"
productName: "break"
price: 12
supplier: "morales"
```

```
_id: ObjectId('62e13ff0646b4e2a90c74e82')
idProduct: "001"
productName: "break"
price: 12
supplier: "Fmorales"
```

Poduct list

Id	Name	Price	Supplier
1	pan de reyes	1.5	juan
2	cachito	0.15	esteban
3	biscocho	0.25	jose
3	Tres leches	1.3	Pompei Bakery
190622	break	12.0	morales
001	break	12.0	Fmorales

Print date to printer 0/5

7. Forms

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

-Login Form

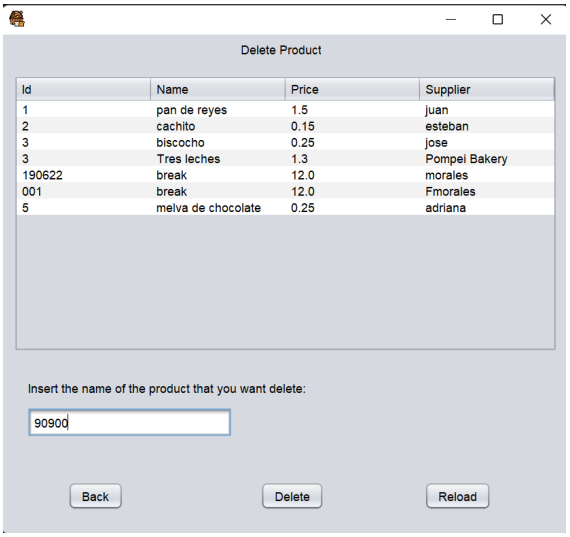
-5 Forms for Information (CRUD operations, 4 business rules)

Every Form

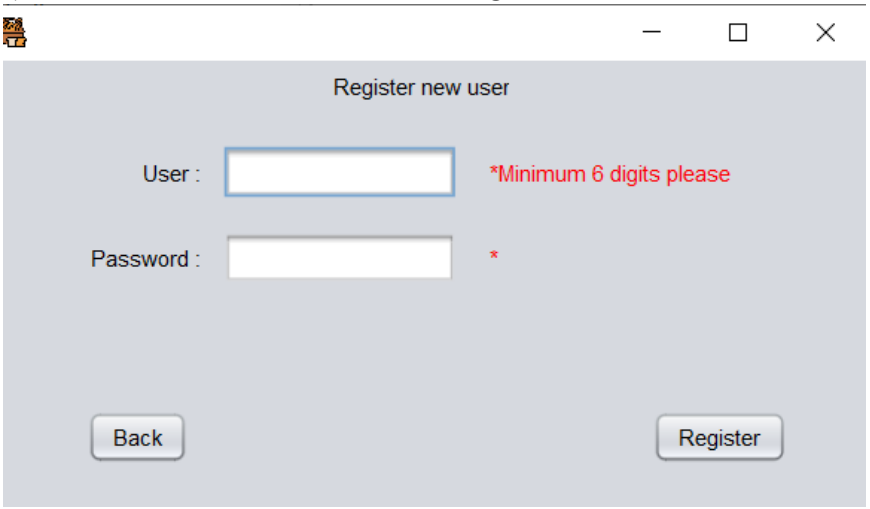
- Every input (JTextField) is validated
- There are different types of inputs (not everything is a JTextFields) -1 for each JTextField
- Buttons' events, no code is here for CRUD operations or business rules, i.e., these functions will call some controller method, and will use some model class.

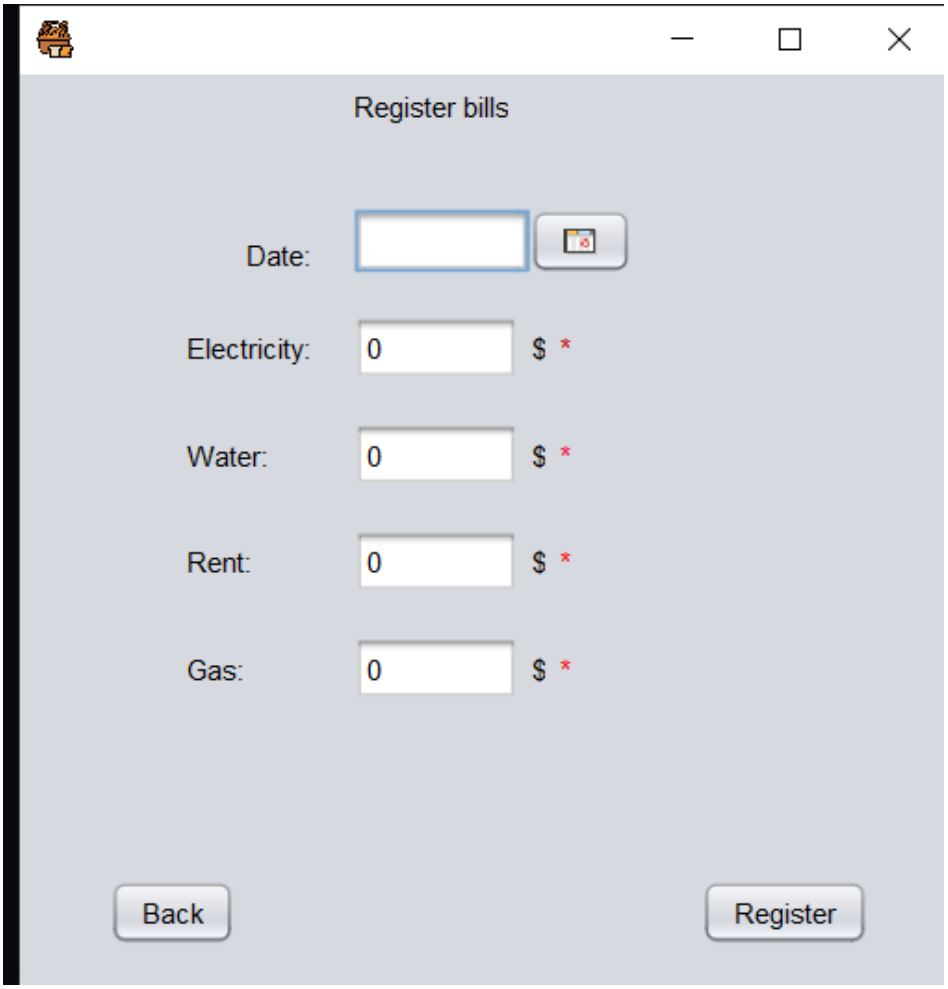
every mistake: -1
less than 5 forms -> /50%

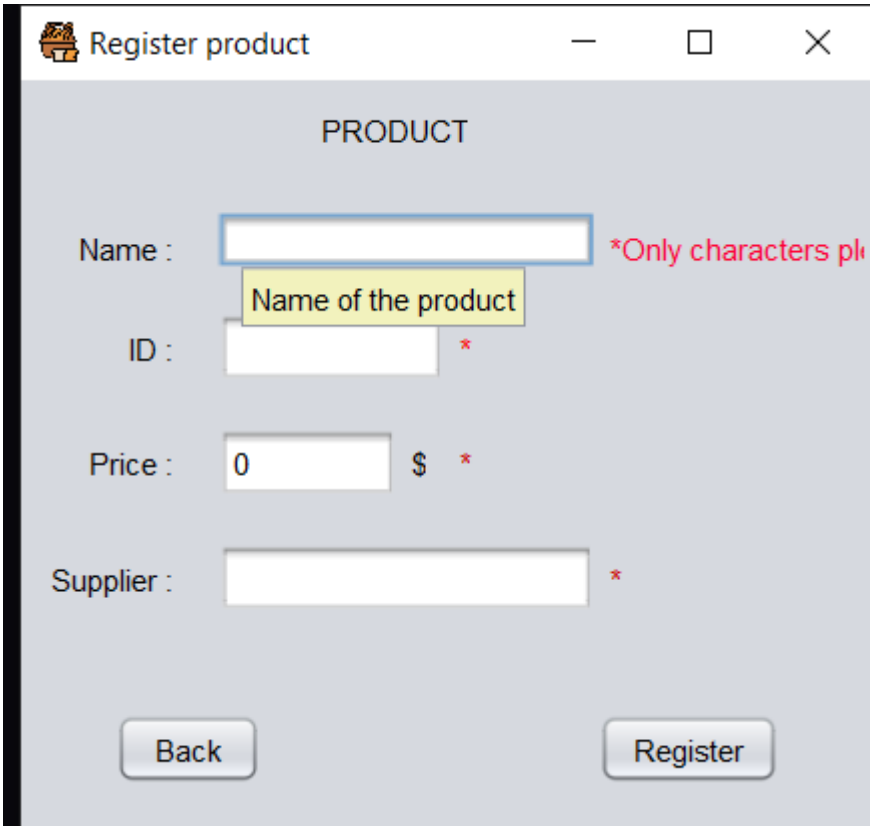
a) Enter the name of the product without validating
b) find the name of the product if you put letters



- Buttons' events, no code is here for CRUD operations or business rules, i.e., these functions will call some controller methods, and will use some model class.
  a)the buttons of the crud are missing




Register bills
Date:
Electricity:
0
$ *
Water:
0
$ *
Rent:
0
$ *
Gas:
0
$ *
Back
Register


Register product
PRODUCT
Name :
*Only characters pl
Name of the product
ID :
*
Price :
0
$ *
Supplier :
*
Back
Register

8. Unit Test

At least 100 unit tests.

- review that de unit test are no empty
- there must be at least one unit test that is failing.
- If all unit tests are OK, then it is half the grade. It means that the tests are not good enough
- every unit test is worth 0.1

Username