UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE

Object-oriented programming

NRC: 9642

U2 Workshop N° 40

Topic: Final Project Inspection

Ing. Jorge Edison Lascano. PhD

Rubric:
Project GitHub **/25 (Individual Contribution)**
Project Review (Inspection) **/25**
- **Requirements and modeling (updated and design patterns)** **/5**
- **Principles,CleanCode, Pitfalls** **/5**
- **Functional Requirements user test (No CRUD operations)** **/5**
- **MongoDB** **/5**
- **Design patterns code** **/5**

Project Maintenance **/25 (Time effort)**
- **new user / user type** **/10**
- **report screen** **/10**
- **print out** **/5**

Project Letter **/25 (Letter of acceptance or review)**

Evidence:
Project GitHub
Project Review (Inspection)
- **Requirements and modeling (updated and design patterns)**
- **Principles,CleanCode, Pitfalls**
- **Functional Requirements user test (No CRUD operations)** **7:25**
- **MongoDB**
- **Design patterns code**

Project Maintenance
Project Letter

Team 1: **Jsons** **Leader:** **Yeshua Chiliquinga**
Inspector: **KillChain** **Leader:** **Anabel Davila**
Project: **Odontoapp**
GitHub: **https://github.com/YeshuaChiliquingaAmaya/Jsons**
MongoDB Compass:
"mongodb+srv://RBenavides:RBenavides@cluster0.js2ve9m.mongodb.net/"

Grade: **PR: /25, PM: /25, Letter: /25 → 59.5/75**

1. **BENAVIDES MACIAS RUBEN DARIO**
2. **CALERO ARGUELLO ADONNY MATEO**
3. **CARRERA QUINDE PABLO ESTEBAN**
4. **CHILIQUINGA AMAYA YESHUA AMADOR**

Rubric:

Project Review (Inspection) **17.5/25**

Requirements and modeling (updated and design patterns) **5/5**

Principles,CleanCode, Pitfalls **3/5**

Functional Requirements user test (No CRUD operations) **3+/5**

MongoDB 1.5+2.5= **4/5**

Design patterns code **2.5/5**

Project Maintenance **17/25 (Time effort)**

new user / user type	**7/10**
report screen	**10/10**
print out	**0/5**

Project GitHub **/25 (Individual Contribution)**

Project Letter **25/25 (Letter of acceptance or review)**

Evidence:

Project GitHub

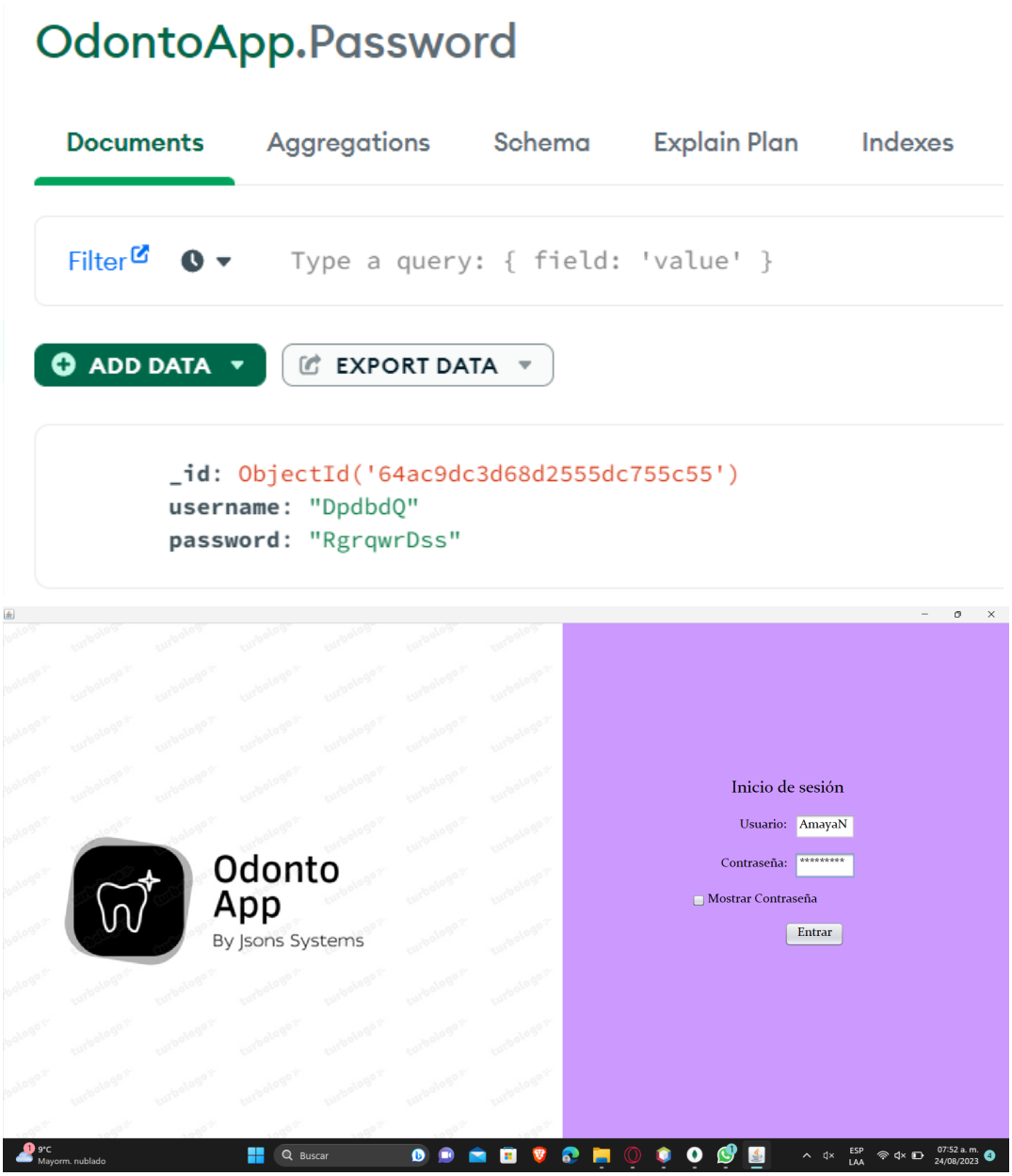
Project Review (Inspection)

Requirements and modeling (updated and design patterns)

- The system will have a login and password.
- **The system will be able to purchase services for a determined patient and show the corresponding bill of what he/she bought.**
- **The system will be able to print this information physically, and or save on a digital file e.g pdf.**
- The system will be able to schedule appointments for a determined patient.

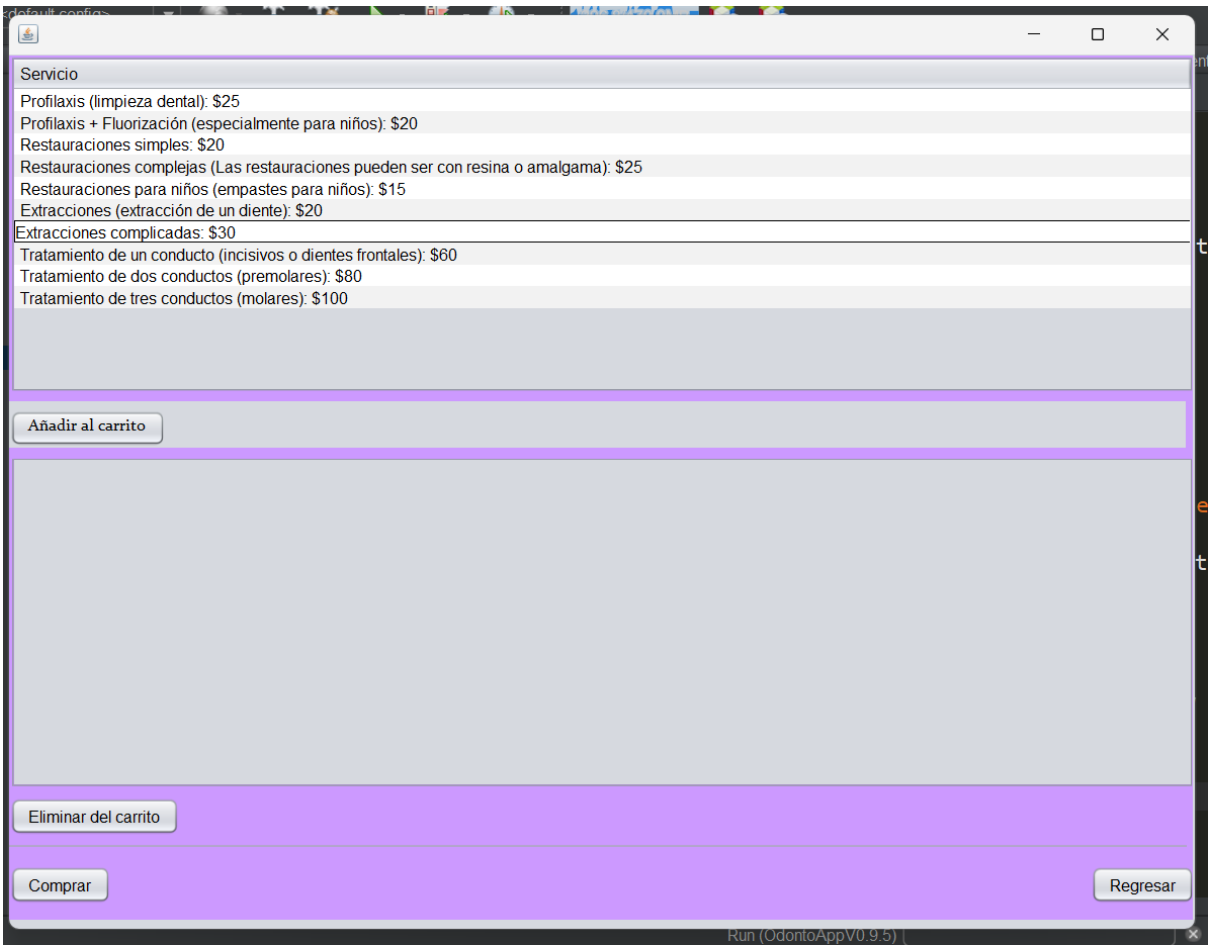
Principles,CleanCode, Pitfalls

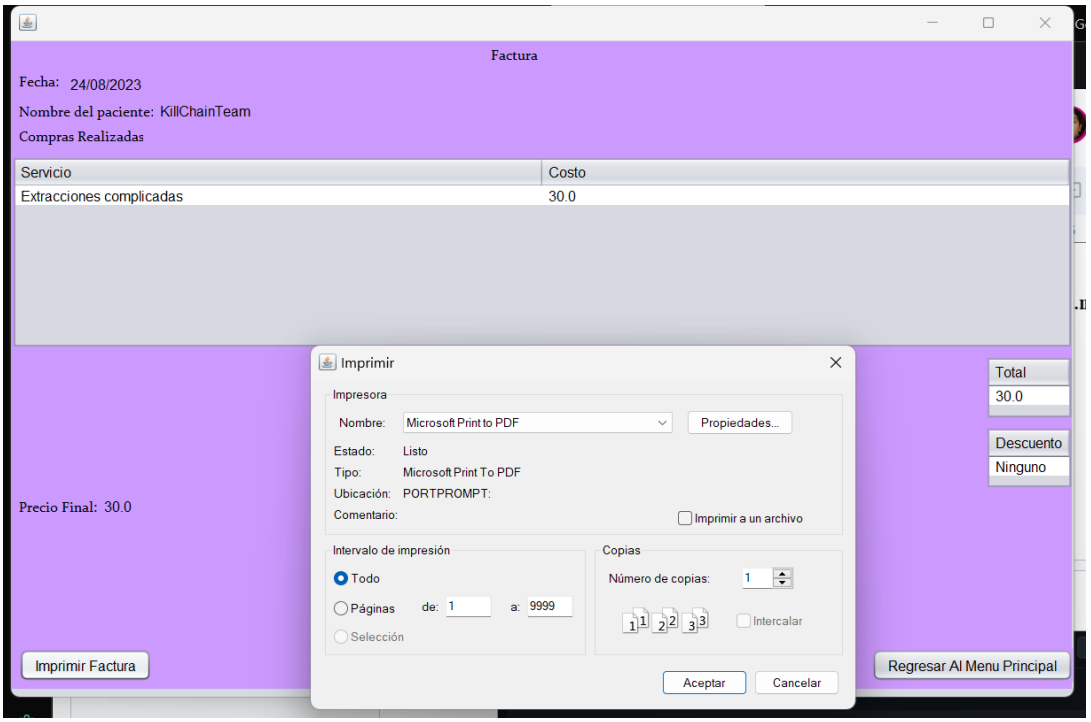
Functional Requirements user test (No CRUD operations)


OdontoApp.Password
Documents
Aggregations
Schema
Explain Plan
Indexes
Filter
Type a query: { field: 'value' }
ADD DATA
EXPORT DATA
_id: ObjectId('64ac9dc3d68d2555dc755c55')
username: "DpdbdQ"
password: "RgrqwrDss"

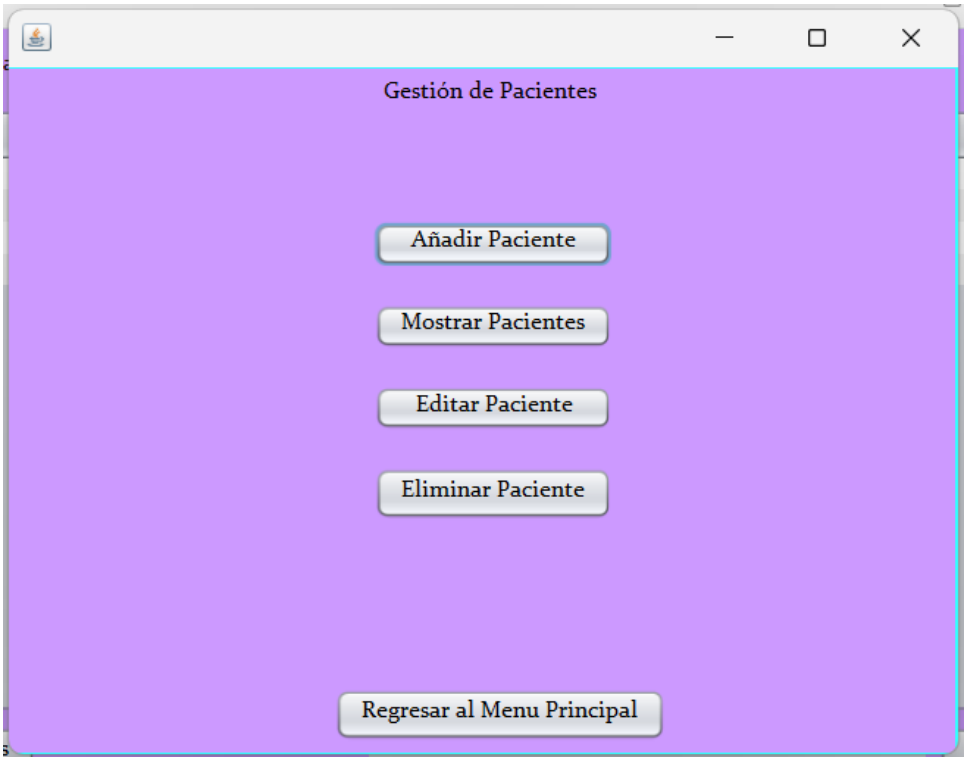
Odonto
App
By Jsons Systems
Inicio de sesión
Usuario:
AmayaN
Contraseña:
****
Mostrar Contraseña
Entrar

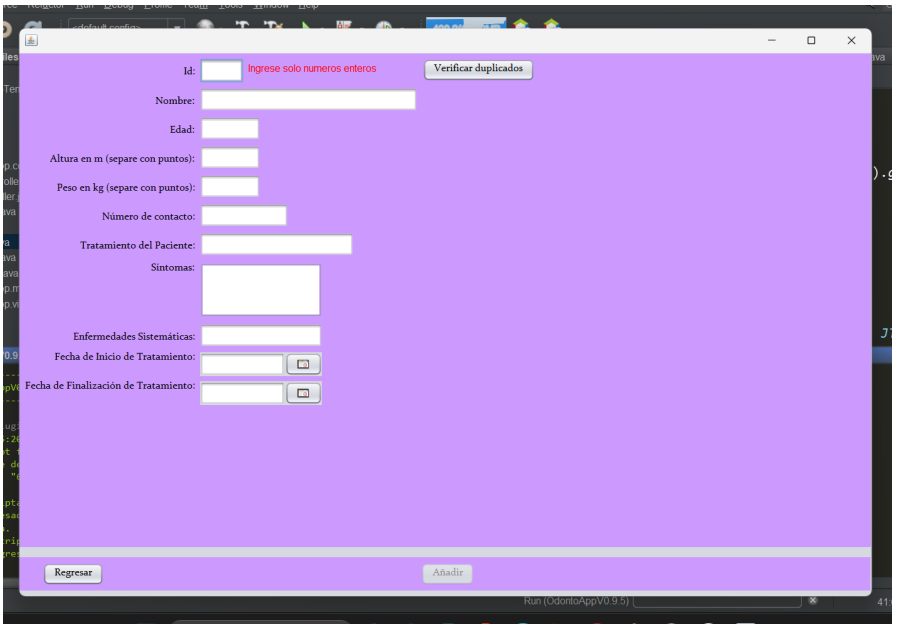
Login





Acquire Services with bill able to print


Gestión de Pacientes
Añadir Paciente
Mostrar Pacientes
Editar Paciente
Eliminar Paciente
Regresar al Menu Principal


Id:
Ingrese solo numeros enteros
Verificar duplicados
Nombre:
Edad:
Altura en m (separe con puntos):
Peso en kg (separe con puntos):
Número de contacto:
Tratamiento del Paciente:
Síntomas:
Enfermedades Sistemáticas:
Fecha de Inicio de Tratamiento:
Fecha de Finalización de Tratamiento:
Regresar
Añadir

```
_id: ObjectId('64e74e64a03e5f184a778477')
clinicalHistory: Object
  id: 4
  name: "KillChainTeam"
  age: 20
  weight: 75
  height: 1.9
  diseaseSymptoms: "Si"
  cellphone: "0968727003"
  systemicDiseases: "No"
  patientTreatment: "Cuidado general"
  treatmentDateStart: "Aug 24, 2023, 12:00:00 AM"
  treatmentEndDate: "Aug 27, 2023, 7:33:26 AM"
acquiredServices: Array (empty)
appointments: Array (empty)
```


CRUD is working

MongoDB


```
_id: ObjectId('64e74e64a03e5f184a778477')
clinicalHistory: Object
  id: 4
  name: "KillChainTeam"
  age: 20
  weight: 75
  height: 1.9
  diseaseSymptoms: "Si"
  cellphone: "0968727003"
  systemicDiseases: "No"
  patientTreatment: "Cuidado general"
  treatmentDateStart: "Aug 24, 2023, 12:00:00 AM"
  treatmentEndDate: "Aug 27, 2023, 7:33:26 AM"
acquiredServices: Array (empty)
appointments: Array (empty)
```


Schedule an appointment update the appointments in DB:

```
_id: ObjectId('64e74e64a03e5f184a778477')
clinicalHistory: Object
    id: 4
    name: "KillChainTeam"
    age: 20
    weight: 75
    height: 1.9
    diseaseSymptoms: "Si"
    cellphone: "0968727003"
    systemicDiseases: "No"
    patientTreatment: "Cuidado general"
    treatmentDateStart: "Aug 24, 2023, 12:00:00 AM"
    treatmentEndDate: "Aug 27, 2023, 7:33:26 AM"
acquiredServices: Array
appointments: Array
  0: Object
      date: "Aug 25, 2023, 12:00:00 AM"
      hour: "10:00 am"
      service: "Profilaxis (limpieza dental): $25"
```


The acquired service request is not updated in the database


```
    _id: ObjectId('64e74e64a03e5f184a778477')
  clinicalHistory: Object
  acquiredServices: Array (empty)
  appointments: Array (1)
    0: Object
        date: "Aug 25, 2023, 12:00:00 AM"
        hour: "10:00 am"
        service: "Profilaxis (limpieza dental): $25"
```


Design patterns code

```
private static LoginController uniqueInstance;

public static LoginController getInstance(JTextField txtUsername, JTextField txtPas
    if(uniqueInstance == null){
        uniqueInstance = new LoginController(txtUsername, txtPassword);
    }
    return uniqueInstance;
}

private LoginController(JTextField txtUsername, JTextField txtPassword) {
    this.txtUsername = txtUsername;
    this.txtPassword = txtPassword;
}
```

Just 1 Design Pattern, Singleton in LoginController

Project Maintenance

Project Letter

Team 2: KillChainTeam Leader: Anabel Dávila.
Inspector: Leader: Carlos Jaya
Project: EVSUStore
GitHub: https://github.com/JuDav-093/ESPE23-KillChainTeam
MongoDB Compass: mongodb+srv://jcobena:jcobena@cluster0.mhpiaao.mongodb.net/
Rubric:
Project GitHub /25 (Individual Contribution)
Project Review (Inspection) 19.5/25
Requirements and modeling (updated and design patterns) 4/5
Principles,CleanCode, Pitfalls 4.5/5
Functional Requirements user test (No CRUD operations) 4.5/5
MongoDB 4/5
Design patterns code 2.5/5
Project Maintenance 10/25 (Time effort)
new user / user type 5/10
report screen 5/10
print out 0/5

Project Letter 0/25 (Letter of acceptance or review)

Evidence:
Project GitHub
Project Review (Inspection)
Requirements and modeling (updated and design patterns)
Principles,CleanCode, Pitfalls
Functional Requirements user test (No CRUD operations)
MongoDB
Design patterns code

Grade: PR: /25, PM: /25, Letter: /25 →29.5 /75

5. **CHIPE ZAMBRANO PAMELA NAOMI**
6. **COBEÑA ZAMBRANO JOAN OSWALDO**
7. **DAVILA MOLINA ANABEL**
8. **ESPIN ANDRADE ANDRES ALEXANDER**

Evidence:

Project Review:

Requirements and modeling (updated and design patterns) 4/5

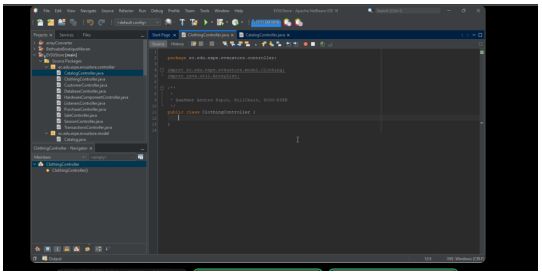
```
public synchronized static DatabaseController getInstance (){
    if (instance != null){

    }
    else {
        instance = new DatabaseController();
    }

    return instance;
}
```

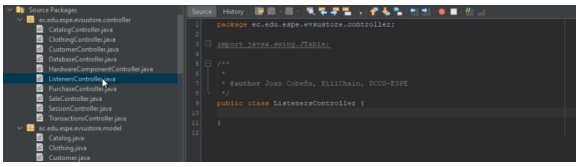
```java
public interface Mapeable {
    public HashMap<Object, Object> getData();
}
```

- Inventory management: add, remove and update products.
- Sales Tracking: record sales and generate reports.
- Customer Information Management: Store and update customer information.
- Vendor Recommendation: Suggest vendors based on component compatibility and quality.
- Algorithm to generate compatibility according to the components you buy.
- Implement a payment system.
- Implement a text generator to automate the creation of reviews and thus improve the business.
- You should handle different classes that contain various unique objects, since managing a store requires you to specify which item is for sale.
- Different types of data are formed since the project tightened it in the same way, the encapsulation of functions will be essential to have a better control of the code.

*** The UML Diagram is not update***

Principles,CleanCode, Pitfalls /5

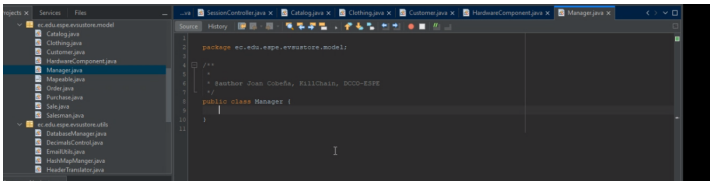




```
private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    int option=JOptionPane.YES_NO_OPTION;
    int election=JOptionPane.showConfirmDialog(parentComponent:this,message:"¿DESEA SALIR?",title: "SALIR",optionsType: opt
    if(election==0){
        System.exit(status: 0);
    }
}
```

LazyClass

Bad implementation of ({}) CleanCodeNOT just one line of code inside the if sentence



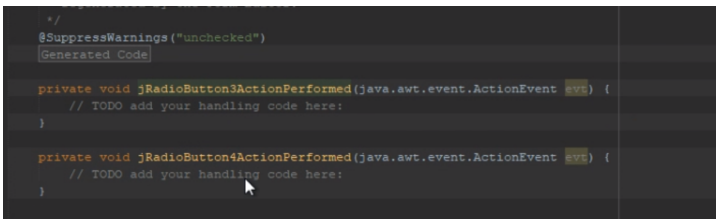
```
}

private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    int option=JOptionPane.YES_NO_OPTION;
    int election=JOptionPane.showConfirmDialog(parentComponent:this,message:"¿DESEA SAL
    if(election==0){
        System.exit(status: 0);
    }
}
```

Button contain code

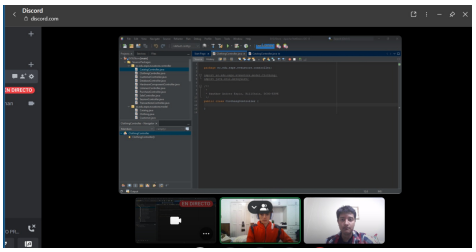
```
private void btnCreteUserActionPerformed(java.awt.event.ActionEvent evt) {
    String name = txtName.getText().trim();
    String lastName = txtLastName.getText().trim();
    String username = txtUserName.getText().trim();
    String password = new String(value: txtPassword.getPassword()).trim();


    boolean createUserSuccess = sessionController.createUser(name, lastName, username, password);
    if (createUserSuccess) {
        JOptionPane.showMessageDialog(parentComponent:this, message:"Usuario creado exitosamente");
        FrmLogin frmLogin = new FrmLogin();
        frmLogin.setVisible(b: true);
        dispose();
    } else {
        JOptionPane.showMessageDialog(parentComponent:this, message:"Error al crear el usuario. Nombre de usuario en
    }
}
```

```
private void btnCreteUserActionPerformed(java.awt.event.ActionEvent evt) {
    String name = txtName.getText().trim();
    String lastName = txtLastName.getText().trim();
    String username = txtUserName.getText().trim();
    String password = new String(value: txtPassword.getPassword()).trim();


    boolean createUserSuccess = sessionController.createUser(name, lastName, username, password);
    if (createUserSuccess) {
        JOptionPane.showMessageDialog(parentComponent:this, message:"Usuario creado exitosamente");
        FrmLogin frmLogin = new FrmLogin();
        frmLogin.setVisible(b: true);
        dispose();
    } else {
        JOptionPane.showMessageDialog(parentComponent:this, message:"Error al crear el usuario. Nombre de usua
    }
}
```

Default Name of Button



Evidence Meet

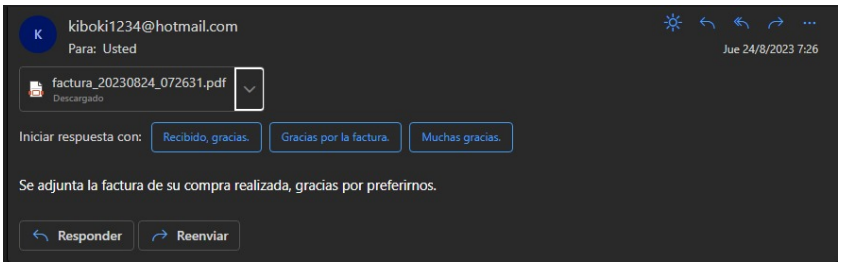


Functional Requirements user test (No CRUD operations) 4.5/5

Functional Requirements

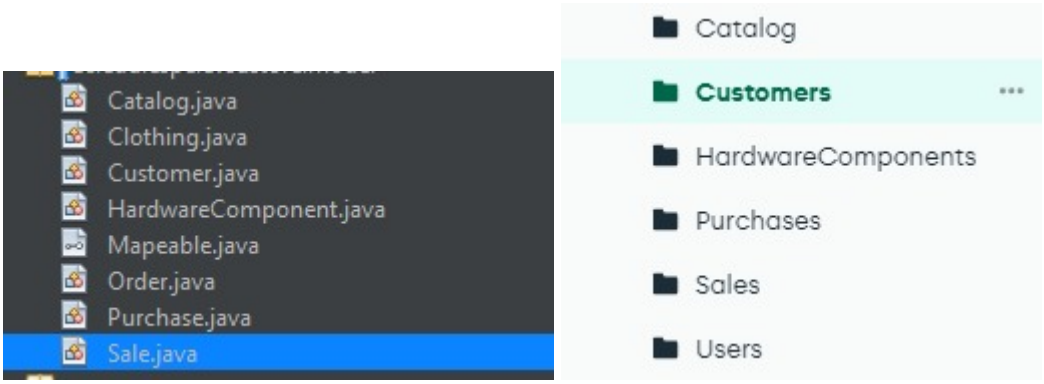
- Inventory management: add, remove and update products.
- **Sales Tracking: record sales and generate reports.**
- **Customer Information Management: Store and update customer information.**
- Vendor Recommendation: Suggest vendors based on component compatibility

Evidence:




```
_id: ObjectId('64e74c6c1ecbc6790faa15c3')
lastName: "FloresMaya"
phoneNumber: "0983372690"
name: "ALejandroCaetano"
idCardNumber: "1724621212"
location: "Quito"
id: 4
email: "caetano.flores@hotmail.com"
```


MongoDB /5



Clothing Class is not implemented in the MongoDB.

Design patterns code /5

```
public synchronized static DatabaseController getInstance (){
    if (instance != null){

    }
    else {
        instance = new DatabaseController();
    }

    return instance;
}
```

Singleton Pattern in the Connection.

Team 3: Leader: Carlos Jaya
Inspector: Team 4:Code Crafters Leader: Josue Marin
Project:
GitHub: https://github.com/cajaya1/MGOOP-SOFTWAREJUNIORS
MongoDB Compass:
mongodb+srv://FCaetano:FCaetano@cluster0.erktrrv.mongodb.net/
Grade: PR: /25, PM: /25, Letter: /25 →47 /75

9. **FLORES MAYA ALEJANDRO CAETANO**
10. **GUAMAN ALCIVAR JORDAN ALEXANDER**

11. JAYA HERRERA CARLOS ANDRES
12. JUMBO SALCEDO BRYAN ALEXANDER

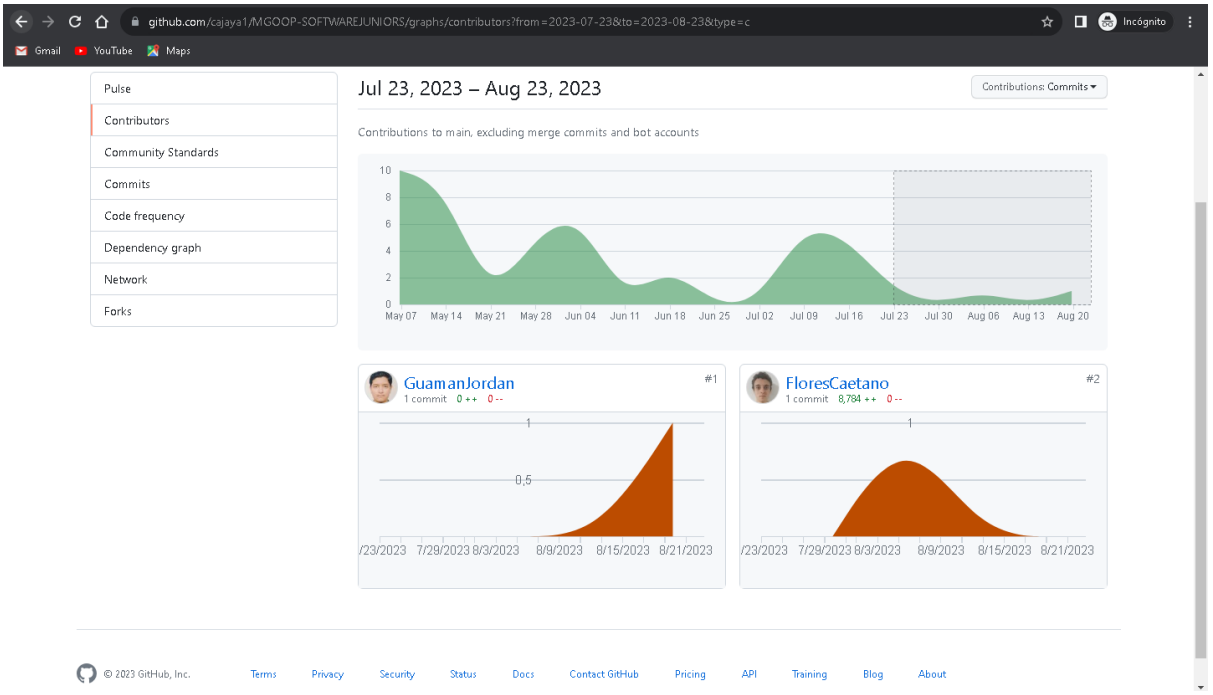
Rubric:

Project Review (Inspection)	19.5/25
Requirements and modeling (updated and design patterns)	**3.5/5**
Principles,CleanCode, Pitfalls	4/5
Functional Requirements user test (No CRUD operations)	4/5
MongoDB	4/5
Design patterns code	4/5
Project Maintenance	16/25 (Time efort)
new user / user type	**1/10**
report screen	**10/10**
print out	**5/5**
Project GitHub	/25 (Individual
TOTAL:	
Contribution)	

13. FLORES MAYA ALEJANDRO CAETANO 25
14. GUAMAN ALCIVAR JORDAN ALEXANDER 25
15. JAYA HERRERA CARLOS ANDRES 0
16. JUMBO SALCEDO BRYAN ALEXANDER 0

Project Letter **12/25 (Letter of acceptance or review)**

Evidence:
Project GitHub



Project Review (Inspection)

Requirements and modeling (updated and design patterns)

3.2. Functional Requirements

Product Registration: The system will allow you to register dresses (by specific object).

Product Elimination: The system will allow you to eliminate a product by searching for its name.

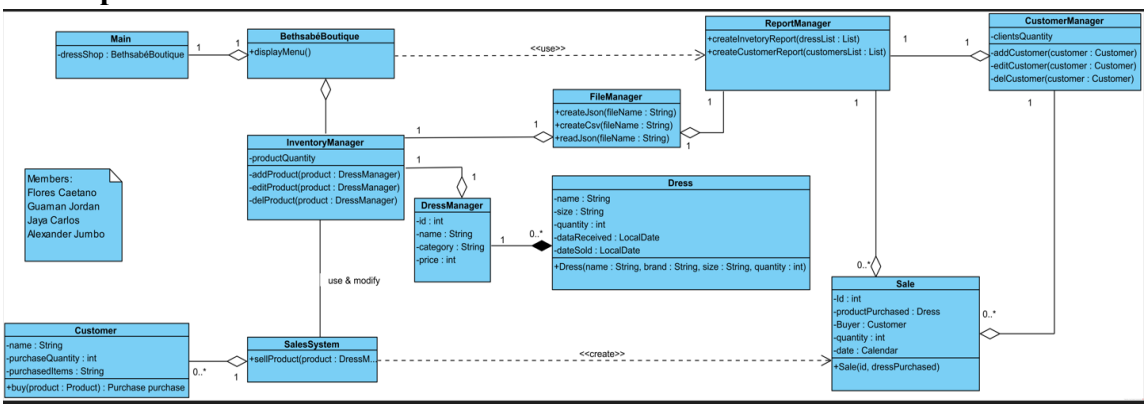
Product Edition: The system will allow you to edit the details of the dress. The system will allow manual registration of customers to the database.

Product printing(generate): Through a menu option it is possible to obtain a document external to the console in which the dataentered by the console is located.

Product Sale: If at least one dress was previously added to inventory, it can be sold, with the current billing application for sale to the public.

Product printing(console): The existing dresses prior to the generation of the file are displayed in the console.

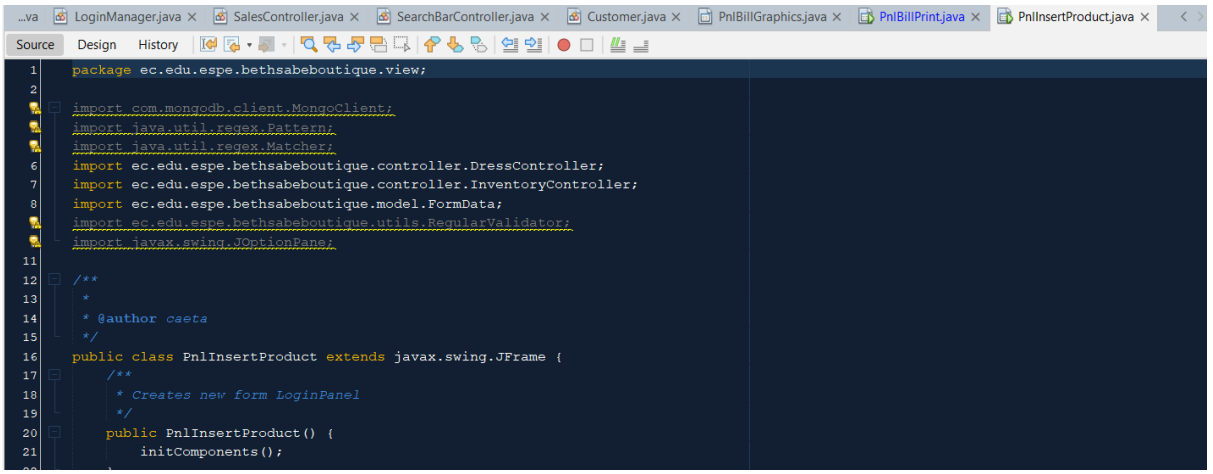
don´t Update UML ⅋



Principles,CleanCode, Pitfalls

```java
public boolean checkProductAlreadyInCart(int id, MongoCollection collection) { //If is on cart adds 1 to quantity
    Document filter = new Document(key:"id", value: id);
    FindIterable<Document> documents = collection.find(bson: filter);
    int quantity = 0;
    for (Document document : documents) {
        quantity = document.getInteger(key:"quantity");
    }
    Document update = new Document(key:"$set", new Document(key:"quantity", quantity+1));
    if(quantity != 0) {
        collection.updateOne(bson: filter, bson1: update);
        return true;
    }else {
        return false;
    }
}

public FindIterable<Document> retrieveDocuments(String collectionName) {
    collection = database.getCollection(string:collectionName);
    Document filter = new Document(); // Filtro vacio para obtener todos los documentos
    FindIterable<Document> documents = collection.find(bson: filter);
    return documents;
}

public float calculateTotal() {
```

Comments in Spanish and unnecessary

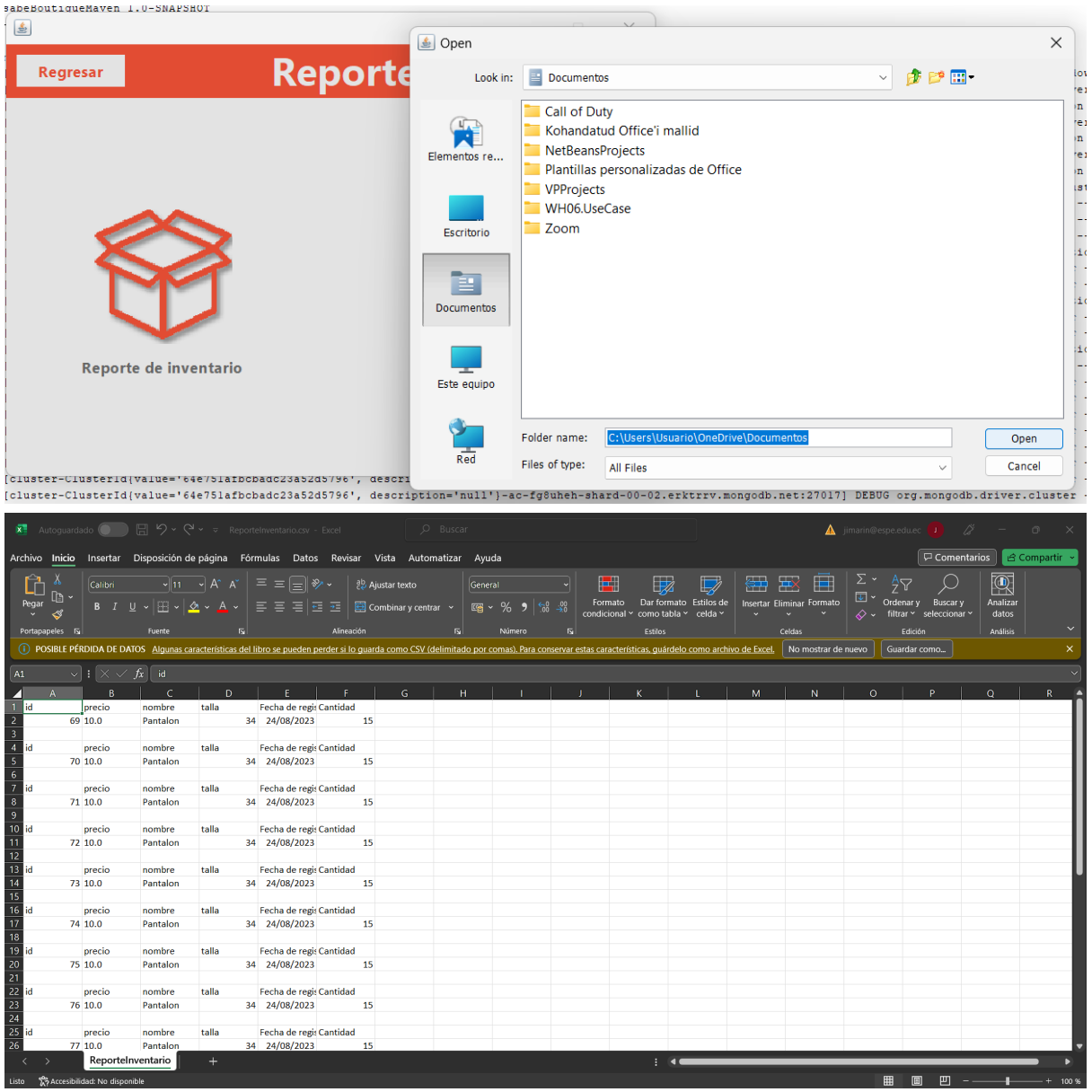



```java
package ec.edu.espe.bethsabeboutique.view;

import com.mongodb.client.MongoClient;
import java.util.regex.Pattern;
import java.util.regex.Matcher;
import ec.edu.espe.bethsabeboutique.controller.DressController;
import ec.edu.espe.bethsabeboutique.controller.InventoryController;
import ec.edu.espe.bethsabeboutique.model.FormData;
import ec.edu.espe.bethsabeboutique.utils.RegularValidator;
import javax.swing.JOptionPane;

/**
 *
 * @author caeta
 */
public class PnlInsertProduct extends javax.swing.JFrame {
    /**
     * Creates new form LoginPanel
     */
    public PnlInsertProduct() {
        initComponents();
```

Import libraries unnecessary because not uses

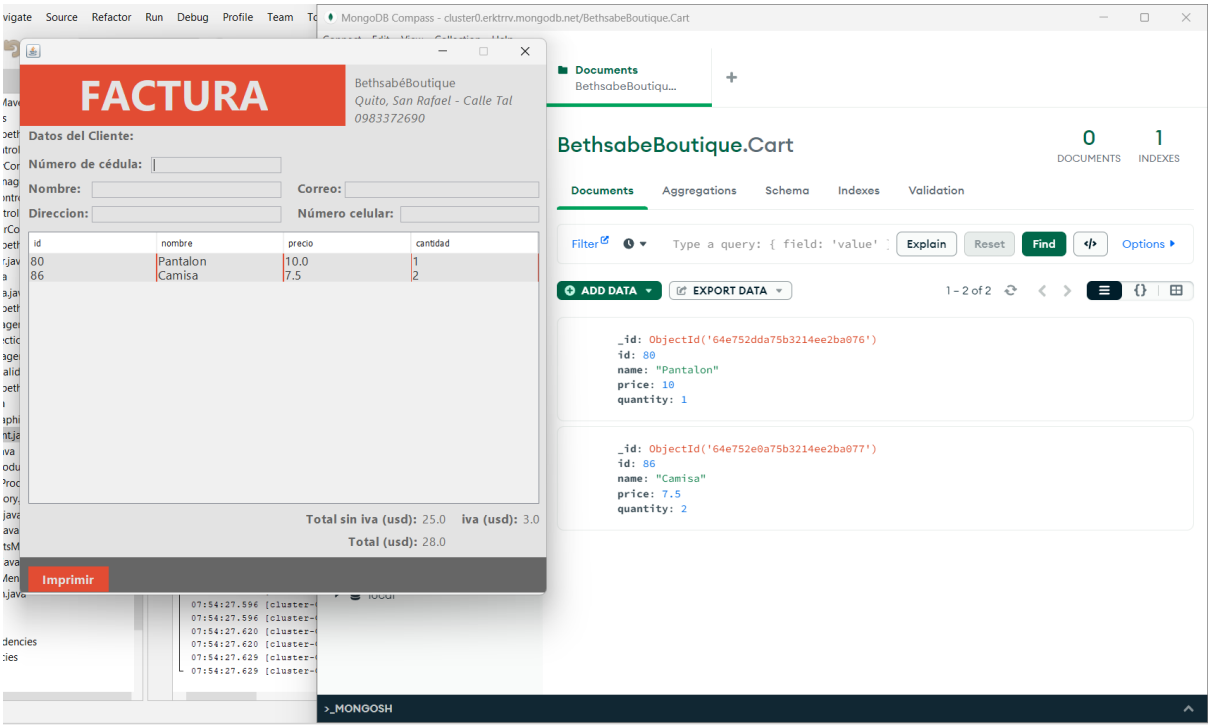
Functional Requirements user test (No CRUD operations)

Product printing(console): The existing dresses prior to the generation of the file are displayed in the console.



Generate report in excel

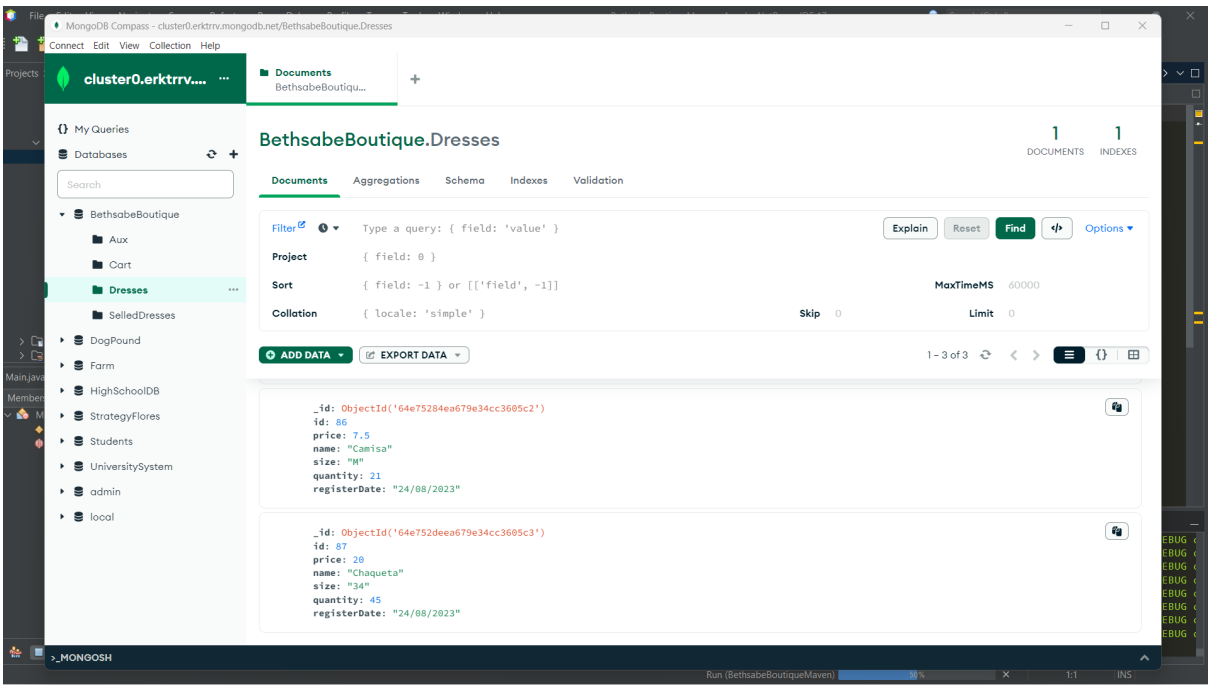
Product Sale: If at least one dress was previously added to inventory, it can be sold, with the current billing application for sale to the public.



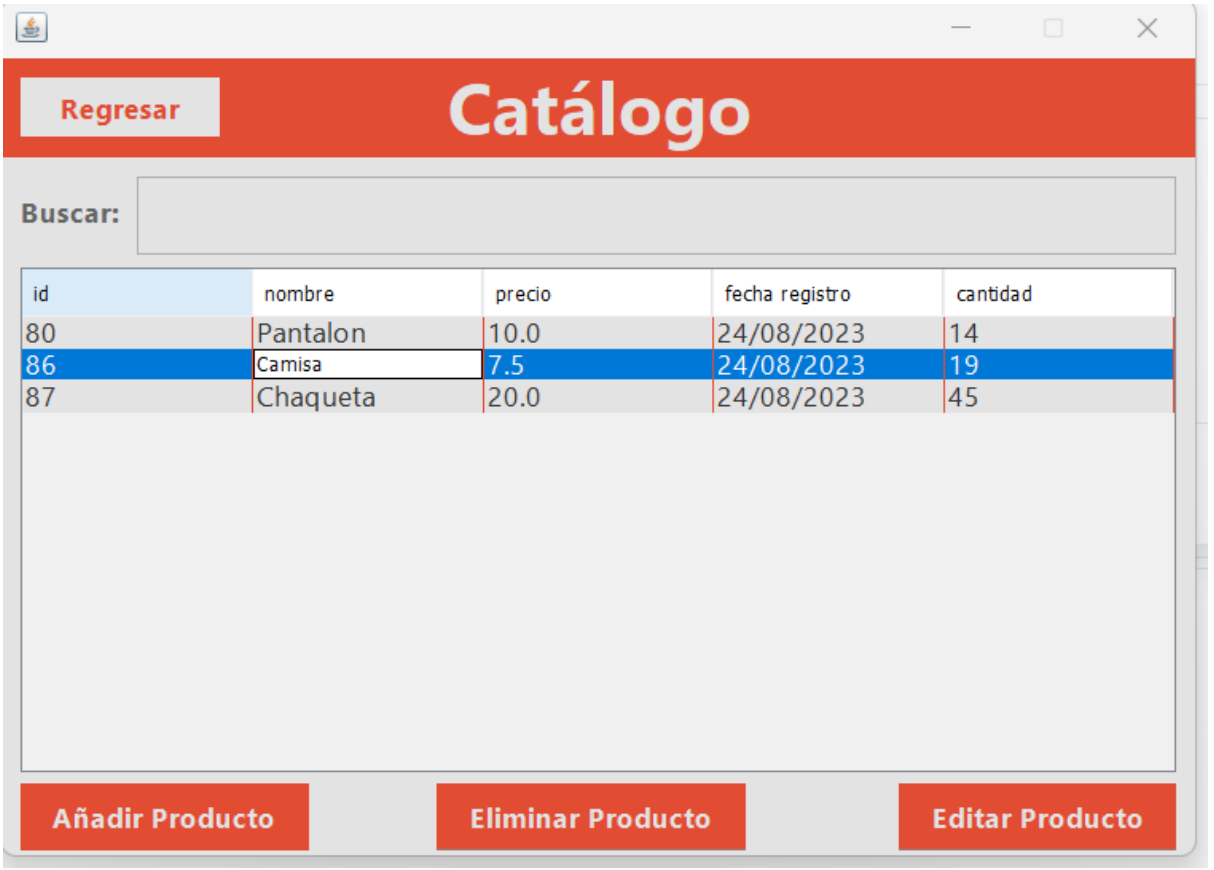
The shop yes sell and generate pdf and show the MongoDB

MongoDB
CRUD

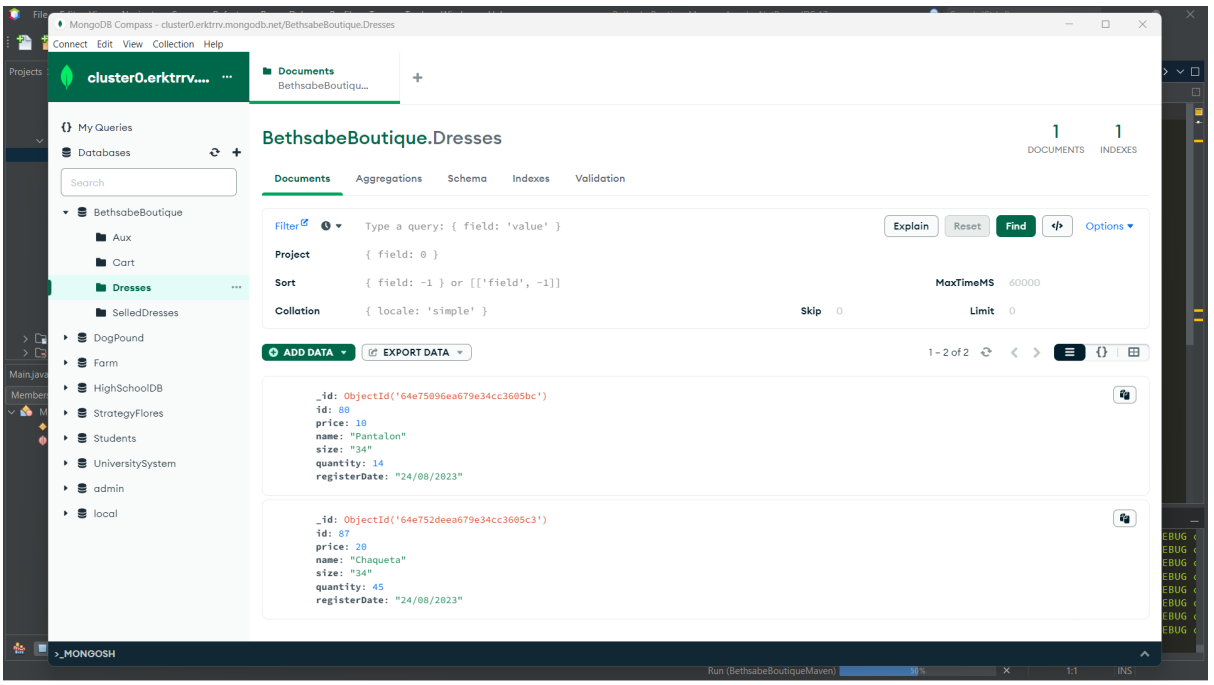
Create


MongoDB Compass - cluster0.erktrrv.mongodb.net/BethsabeBoutique.Dresses
Connect Edit View Collection Help
cluster0.erktrrv....
Documents
BethsabeBoutiqu...
My Queries
Databases
Search
BethsabeBoutique
Aux
Cart
Dresses
SelledDresses
DogPound
Farm
HighSchoolDB
StrategyFlores
Students
UniversitySystem
admin
local
BethsabeBoutique.Dresses
1 DOCUMENTS
1 INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' }
Explain
Reset
Find
Options
Project
{ field: 0 }
Sort
{ field: -1 } or [['field', -1]]
MaxTimeMS
60000
Collation
{ locale: 'simple' }
Skip
0
Limit
0
ADD DATA
EXPORT DATA
1 - 3 of 3
_id: ObjectId('64e75284ea679e34cc3605c2')
id: 86
price: 7.5
name: "Camisa"
size: "M"
quantity: 21
registerDate: "24/08/2023"
_id: ObjectId('64e752deea679e34cc3605c3')
id: 87
price: 20
name: "Chaqueta"
size: "34"
quantity: 45
registerDate: "24/08/2023"
>_MONGOSH

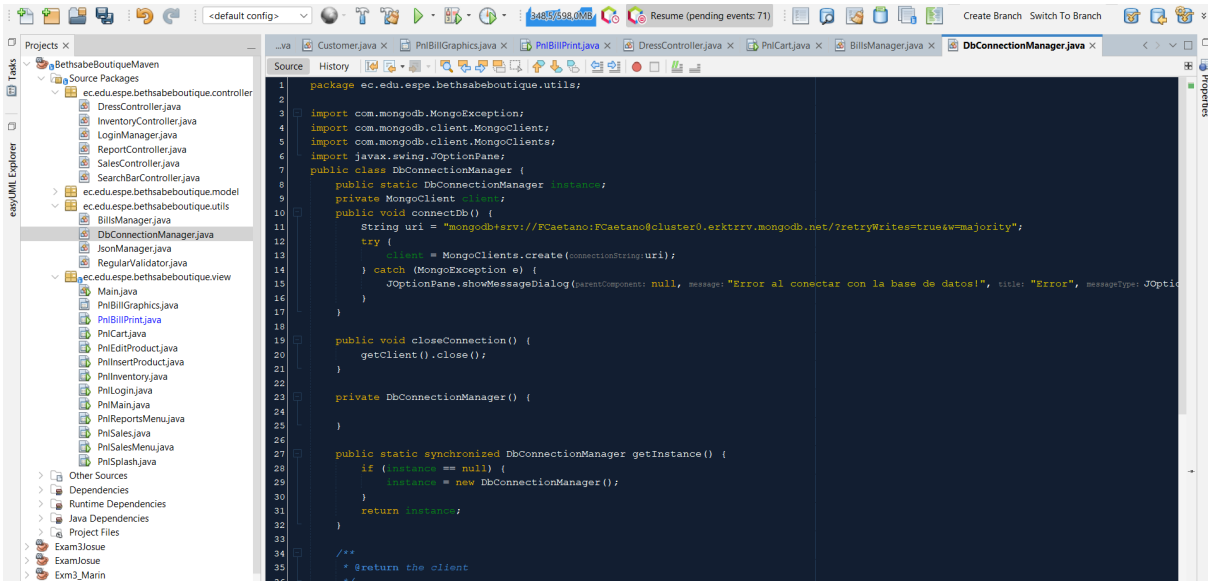
READ


Regresar
Catálogo
Buscar:
id
nombre
precio
fecha registro
cantidad
80
Pantalon
10.0
24/08/2023
14
86
Camisa
7.5
24/08/2023
19
87
Chaqueta
20.0
24/08/2023
45
Añadir Producto
Eliminar Producto
Editar Producto

UPDATE



Design patterns code



singleton

Project Maintenance
Project Letter
Team 4: Code Crafther Leader: Josue Marin
Inspector: Team 5 The FAMSE Leader: Anthony Morales
Project: Hardware Store
GitHub: https://github.com/marinjosue/ESPE2305-CodeCrafters.git
Compass: mongodb+srv://josuemarin:josuemarin@cluster0.lntjz9j.mongodb.net/

Grade: PR: /25, PM: /25, Letter: /25 → 54/75

17. MANOBANDA CHABLA JEFFREY ALEXANDER

18. **MARCILLO MORA JHORDY PAUL**
19. **MARIN ALQUINGA JOSUE ISAAC**
20. **MARISCAL OÑA MESIAS ORLANDO**

Rubric:

Project GitHub **/25 (Individual Contribution)**

Project Review (Inspection) **19/25**

- **Requirements and modeling (updated and design patterns)** **3.5/5**
  - No Dependencies
- **Principles, CleanCode, Pitfalls** **4/5**
- **Functional Requirements user test (No CRUD operations)** **5/5**
  Discount Product, Add to cart
- **MongoDB** **4/5**
- **Design patterns code** **2.5/5**

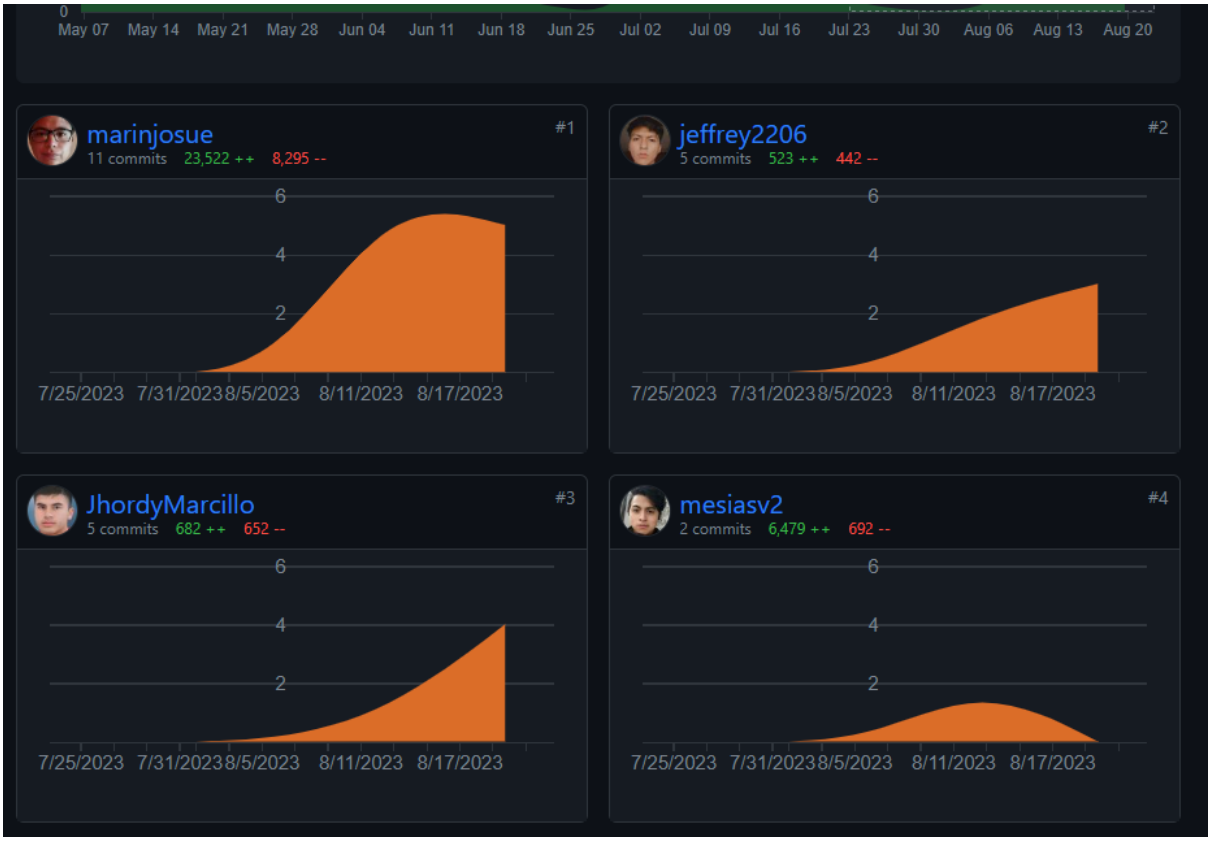
Project Maintenance **20/25 (Time effort)**

- **new user / user type** **5/10**
- **report screen** **10/10**
- **print out** **5/5**

Project Letter **15/25 (Letter of acceptance or review)**
missing contact information

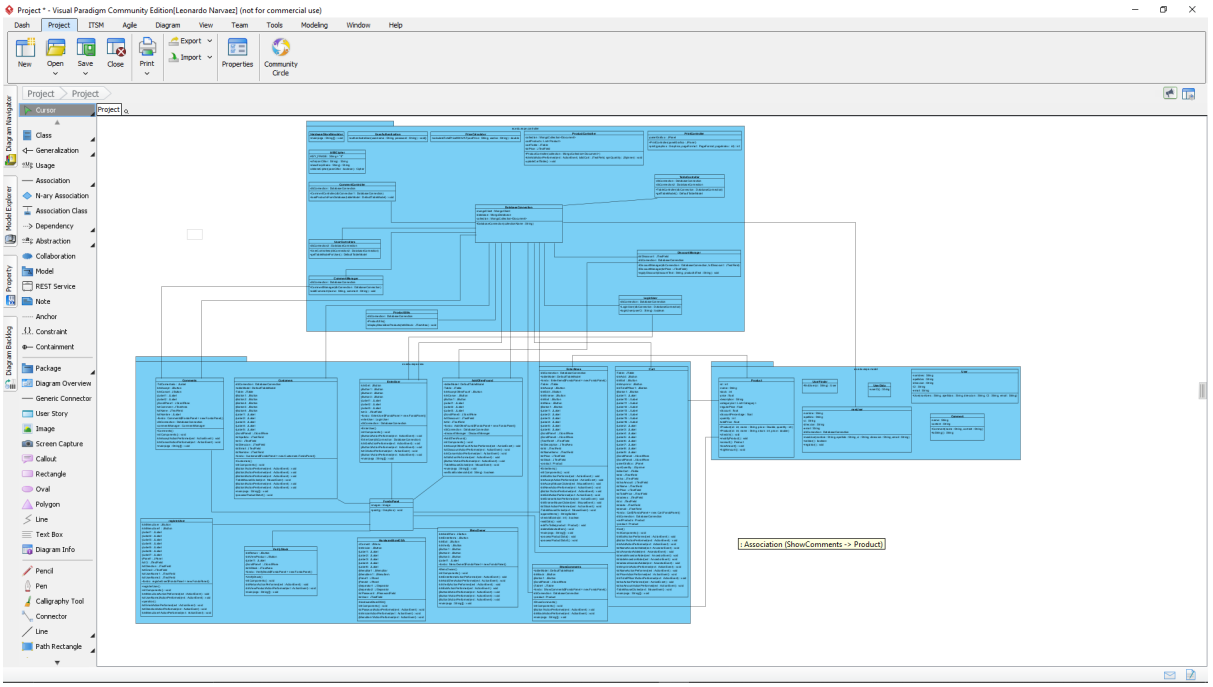
Evidence:

Project GitHub



Project Review (Inspection)

Requirements and modeling (updated and design patterns)



3.3 Functional requirements

OWNER

3.3.1 Functional requirement 1

User registration: The application should allow users to register and create an account in order to access all the functionalities of the application, such as viewing products, making purchases, managing their profile, among others.

3.3.2 Functional requirement 2

Product search: The application should allow owner to search for specific products using different search criteria, such as name, brand, Id, category, price, among others.

3.3.3 Functional requirement 3

Add products to shopping cart: The application must allow the owner to add products to the cart to generate a purchase invoice.

3.3.4 Functional requirement 4

Product offer: The owner of a hardware business requires a system that allows him to manage promotions and offers effectively. The software must offer the ability to create and schedule

special promotions, set percentage or dollar discounts, and define start and end dates for each promotion.

3.3.5 Functional requirement 5

User profile management: the application must allow the owner to manage a profile for customers where they will include the information to generate the invoice data.

3.3.6 Functional requirement 6

User profile management: The application should allow users to manage their profile, including updating their personal information, managing their purchase history.

3.3.7 Functional requirement 7

Add or remove a product to the catalog: The software should allow the owner to add a new product to the system, providing details such as name, description, category, price, and quantity in stock. If the product is available in inventory, the corresponding information must be recorded. In the event that there is no stock of the product, the system must allow the owner to record this situation and establish a notification to replenish the inventory. In addition, the owner must have the ability to remove a product from the catalog, when necessary, automatically updating the information related to the inventory.

3.3.8 Functional requirement 8

Modify product data: The system will have a function to enter the product catalog in the database. Certified users can add new information about each product, such as name, description, prices, and categories. Before allowing entry, you will be confirmed that the essential fields have been completed.

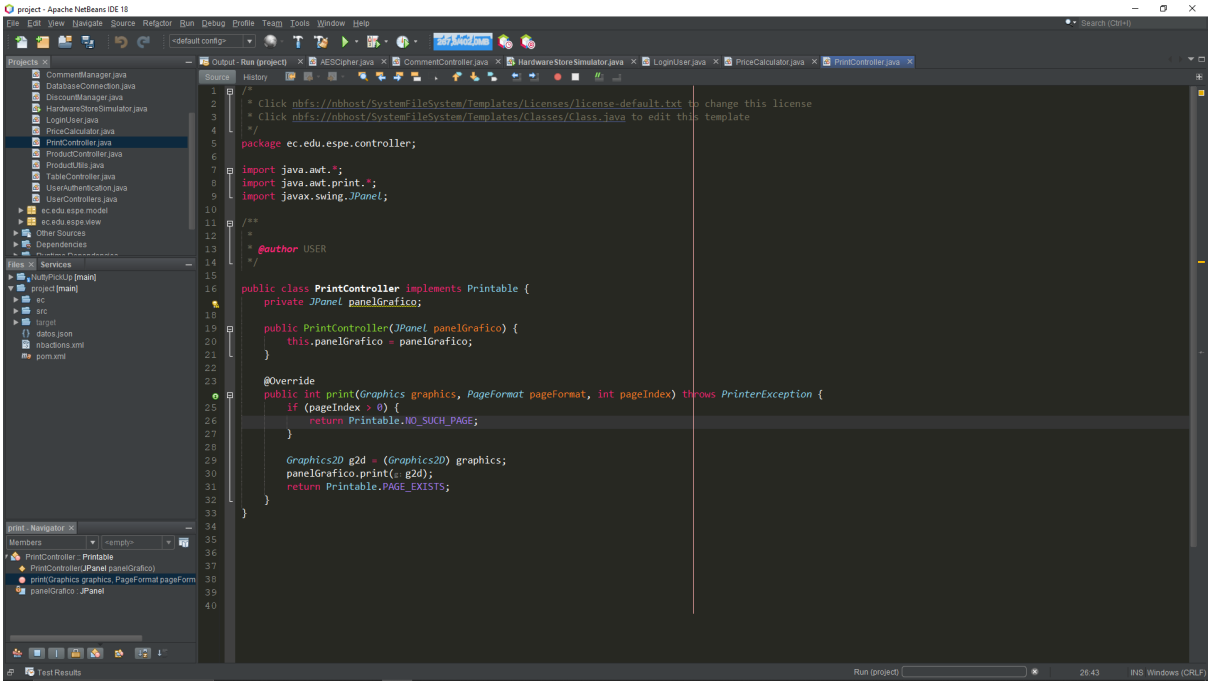
3.3.9 Functional requirement 9

Modify product data in the catalog: The system will allow the modification of the data of the existing products in the catalogue. Authorized users will be able to access and edit product information in a timely manner. Modifications can be made in fields such as name, description, price and category. The system will keep a record of all changes made, including the date and the user responsible for each change.

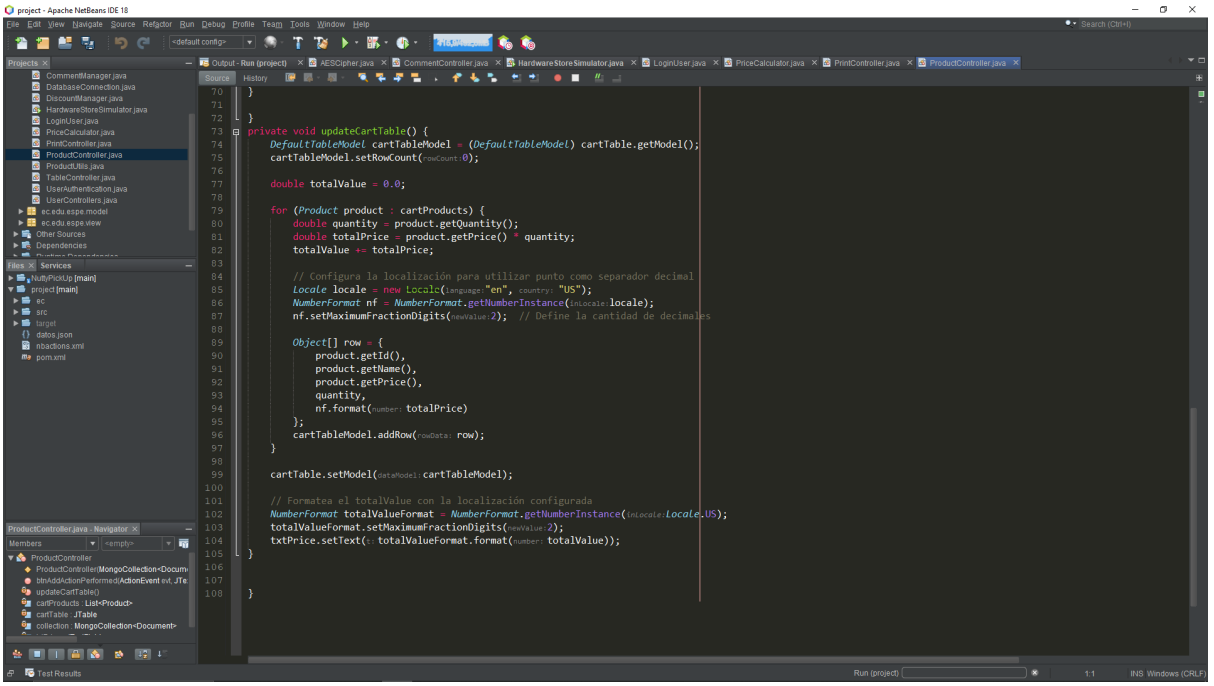
3.3.10 Functional requirements 10

You can add comments about the store in general to get feedback to continually improve the user experience. Users will be able to leave comments and opinions about the store, the products, the customer service, and any other relevant aspect. These comments will be visible to other users who visit the online store.

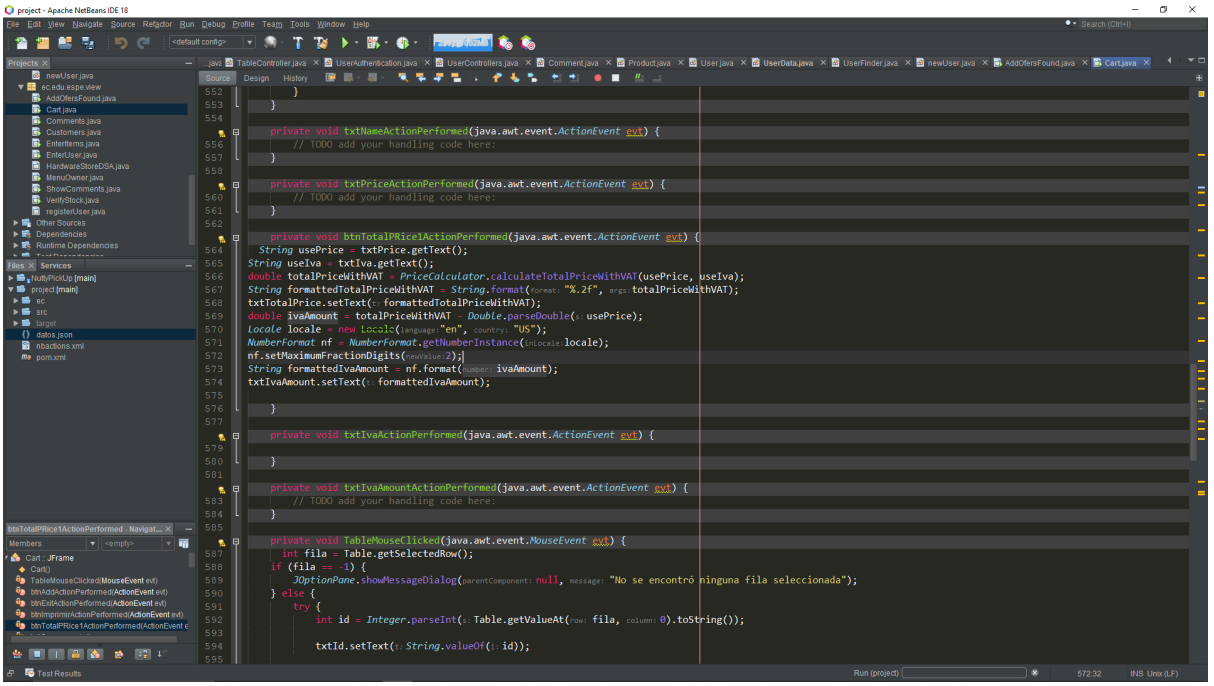
Principles,CleanCode, Pitfalls



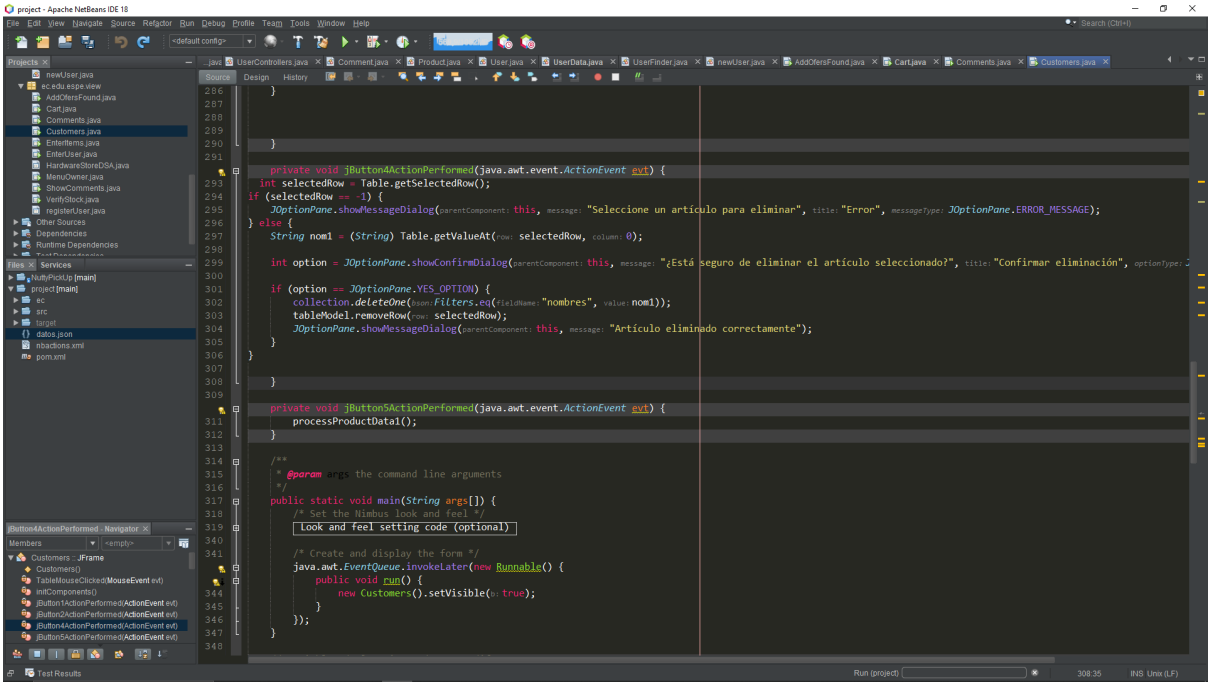
Variables in Spanish



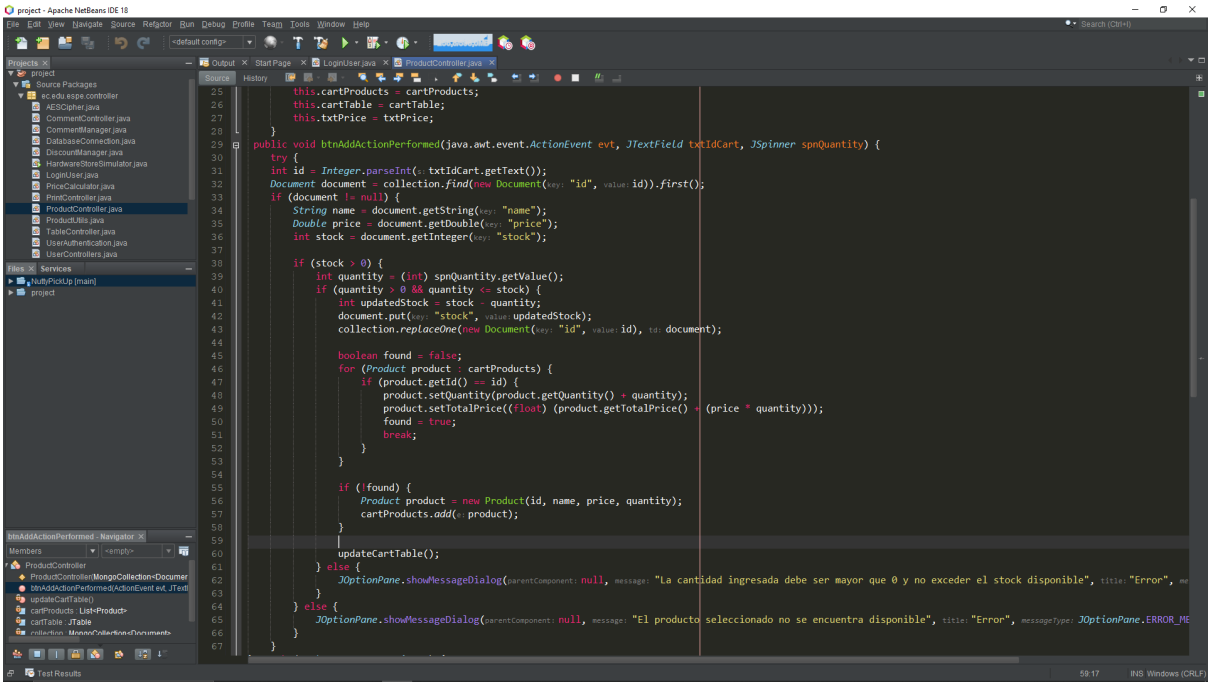
Comments in spanish.



Does not follow clean code standards.



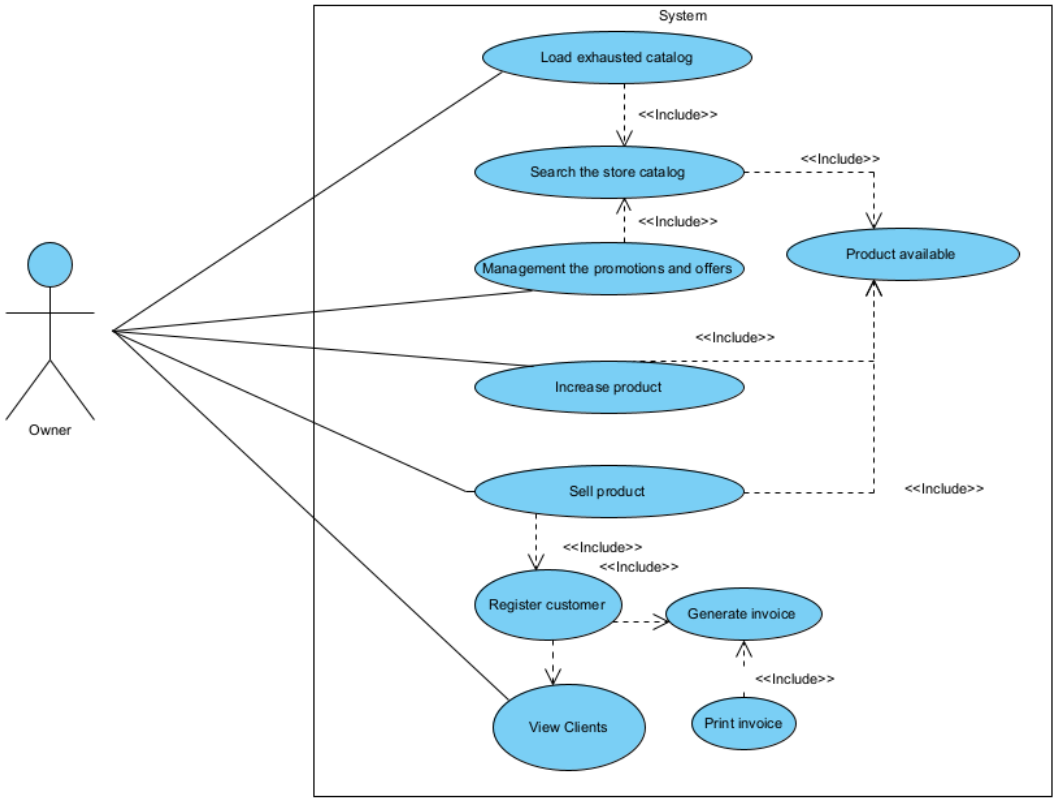
A lot of code in manny buttons



```
            this.cartProducts = cartProducts;
            this.cartTable = cartTable;
            this.txtPrice = txtPrice;
        }
    public void btnAddActionPerformed(java.awt.event.ActionEvent evt, JTextField txtIdCart, JSpinner spnQuantity) {
        try {
        int id = Integer.parseInt(s: txtIdCart.getText());
        Document document = collection.find(new Document(key: "id", value:id)).first();
        if (document != null) {
            String name = document.getString(key: "name");
            Double price = document.getDouble(key: "price");
            int stock = document.getInteger(key: "stock");

            if (stock > 0) {
                int quantity = (int) spnQuantity.getValue();
                if (quantity > 0 && quantity <= stock) {
                    int updatedStock = stock - quantity;
                    document.put(key: "stock", value:updatedStock);
                    collection.replaceOne(new Document(key: "id", value:id), td: document);

                    boolean found = false;
                    for (Product product : cartProducts) {
                        if (product.getId() == id) {
                            product.setQuantity(product.getQuantity() + quantity);
                            product.setTotalPrice((float) (product.getTotalPrice() + (price * quantity)));
                            found = true;
                            break;
                        }
                    }

                    if (!found) {
                        Product product = new Product(id, name, price, quantity);
                        cartProducts.add(e: product);
                    }

                    updateCartTable();
                } else {
                    JOptionPane.showMessageDialog(parentComponent: null, message: "La cantidad ingresada debe ser mayor que 0 y no exceder el stock disponible", title: "Error",
                }
            } else {
                JOptionPane.showMessageDialog(parentComponent: null, message: "El producto seleccionado no se encuentra disponible", title: "Error", messageType: JOptionPane.ERROR_ME
            }
        }
```

Some variables like spnQuantity could be more descriptive.

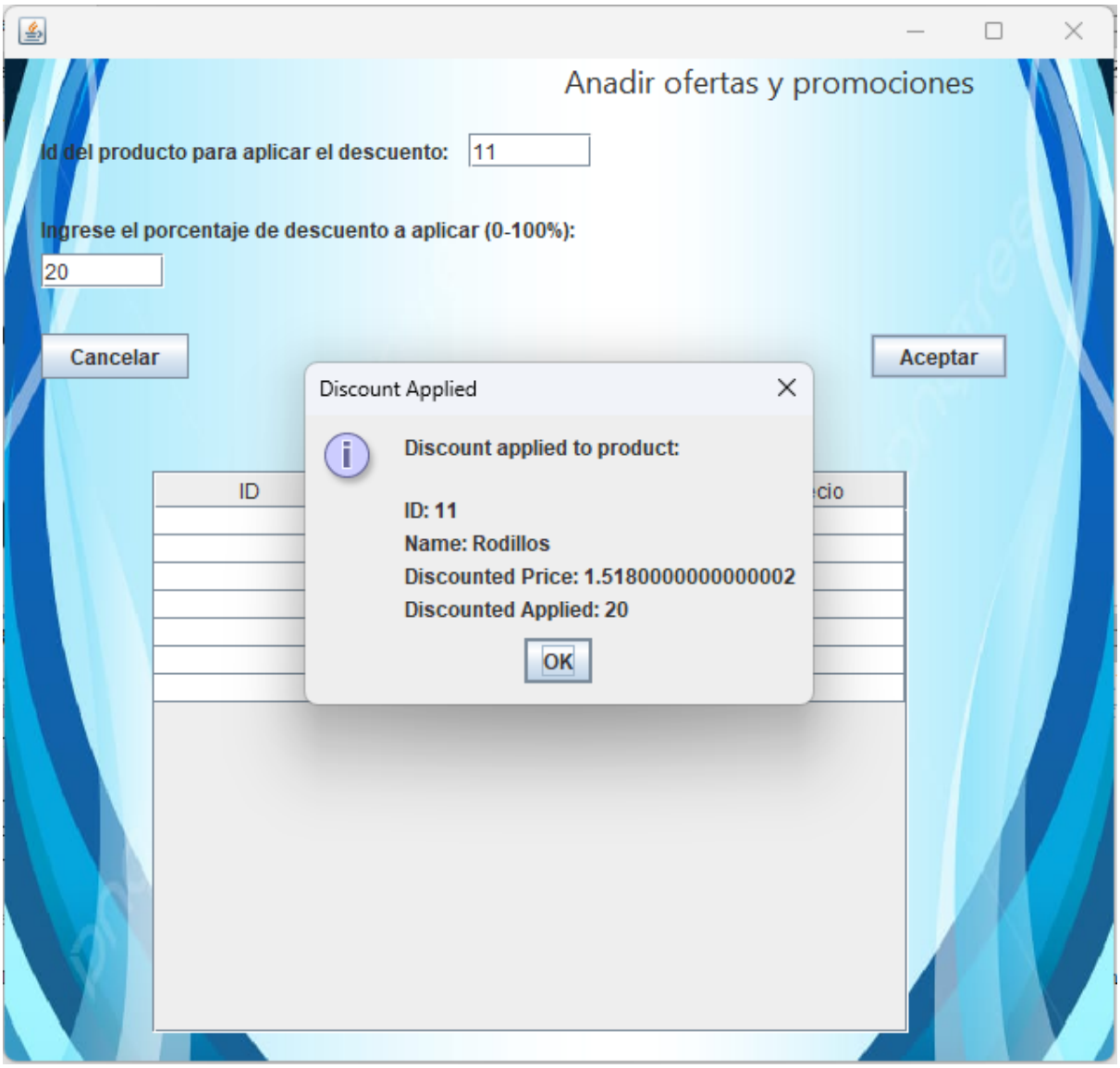
Use case diagram



Functional Requirements user test (No CRUD operations)

3.3.4 Functional requirement 4

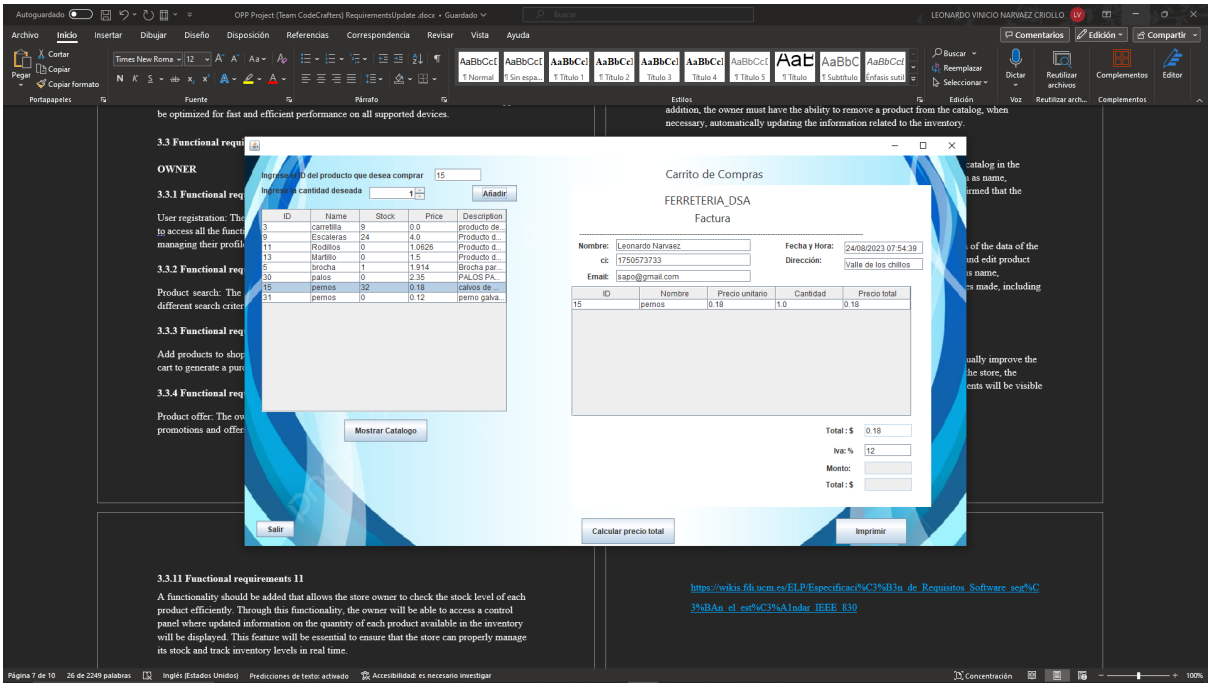
Product offer: The owner of a hardware business requires a system that allows him to manage promotions and offers effectively. The software must offer the ability to create and schedule special promotions, set percentage or dollar discounts, and define start and end dates for each promotion.

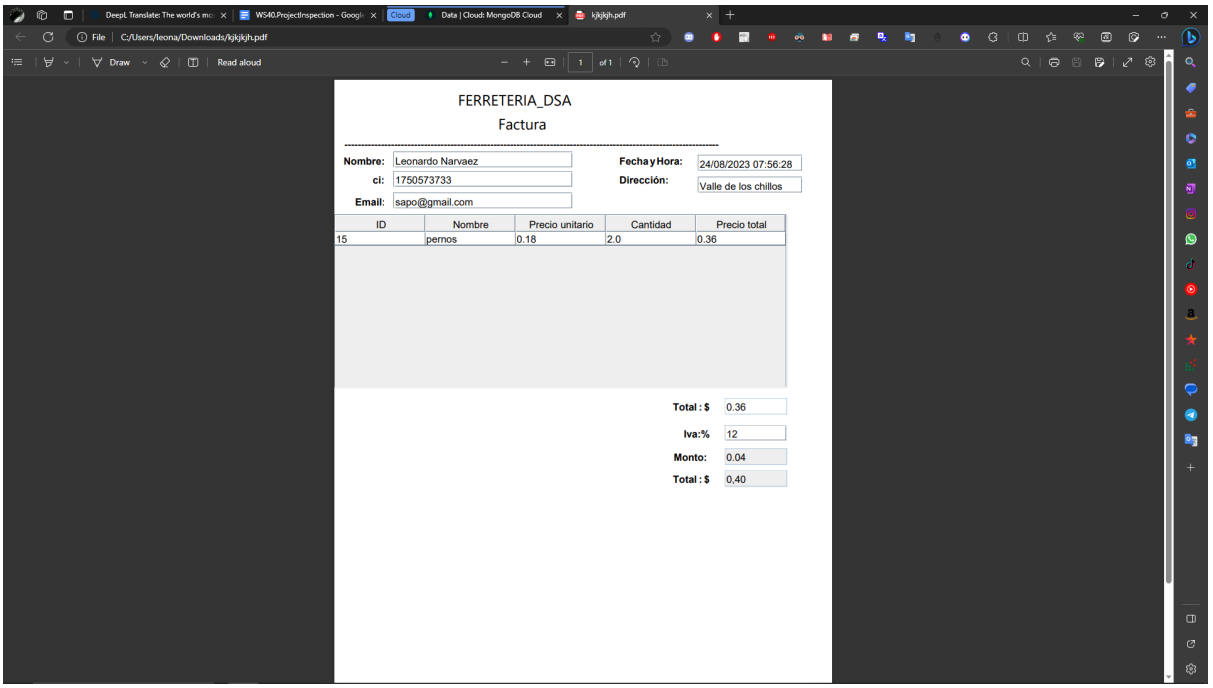


MongoDB

3.3.3 Functional requirement 3

Add products to shopping cart: The application must allow the owner to add products to the cart to generate a purchase invoice.





MongoDB

Design patterns code

Project Maintenance
Project Letter

Team 5: The FAMSE **Leader: Anthony Morales**

Inspector: Team 6 Code Warriors **Leader:** Edison Verdesoto
Project: NPU System
GitHub: **https://github.com/LeoNarvaez2503/The_FAMSE.git**
Compass:
mongodb+srv://anthonymorales:**anthonymorales@cluster0.nngqbpt.mongodb.net/**
Grade: **PR: /25, PM: /25, Letter: /25 →54 /75**

21. **MORALES CARVAJAL ANTHONY ALAIN**
22. **MORAN PALOMO LENIN DAVID**
23. **NARVAEZ CRIOLLO LEONARDO VINICIO**

Rubric:
Project Review (Inspection) **16/25**
Requirements and modeling (updated and design patterns) **4.5/5**
Principles,CleanCode, Pitfalls **2.5/5**
troubles: variables name, no indented, no Single responsibility, no vertical alignment
Functional Requirements user test (No CRUD operations) **5/5**
Sales proccesing , report generator
MongoDB **4/5**
Missing Login in the BD and Users in the code.
Design patterns code **0/5**
No pattern design
Project Maintenance **22/25** (Time effort)
new user / user type **8/10**
report screen **9/10**
print out **5/5**

Project GitHub **25/25** (Individual Contribution)
Project Letter **15/25** (Letter of acceptance or review)

missing contact information

Evidence:
Project GitHub

Project Review (Inspection)

Requirements and modeling (updated and design patterns)

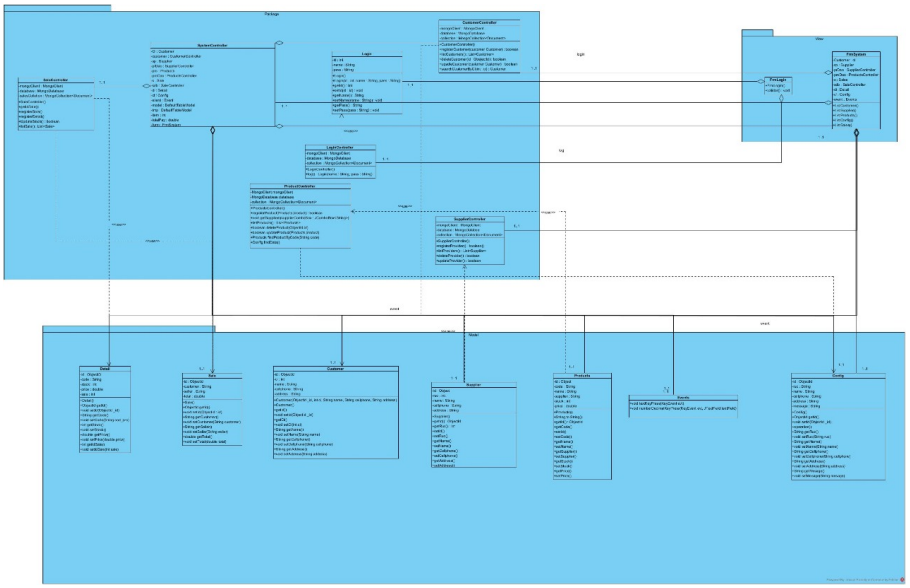
Requirement Number	R1
Name of Requirement	Product management
Type	Requirement / Restriction
Requirement Priority	High/Essential / Medium/Desired / Low/Optional

Requirement Number	R2
Name of Requirement	Price control
Type	Requirement / Restriction
Requirement Priority	High/Essential / Medium/Desired / Low/Optional

Requirement Number	R3
Name of Requirement	Stock management
Type	Requirement / Restriction
Requirement Priority	High/Essential / Medium/Desired / Low/Optional

Requirement Number	R4
Name of Requirement	Sales processing
Type	Requirement / Restriction
Requirement Priority	High/Essential / Medium/Desired / Low/Optional

Requirement Number	R5
Name of Requirement	Report generator
Type	Requirement / Restriction
Requirement Priority	High/Essential / Medium/Desired / Low/Optional



Principles,CleanCode, Pitfalls

```
        Object[] ob = new Object[6];
        for(int i = 0; i < ListarCl.size(); i++){
            ob[0] = ListarCl.get(index: i).getId();
            ob[1] =  ListarCl.get(index: i).getCi();
            ob[2] = ListarCl.get(index: i).getName();
            ob[3] = ListarCl.get(index: i).getCellphone();
            ob[4] = ListarCl.get(index: i).getAddress();
            modelo.addRow(rowData: ob);
        }
        TableCustomer.setModel(dataModel:modelo);
    }
```

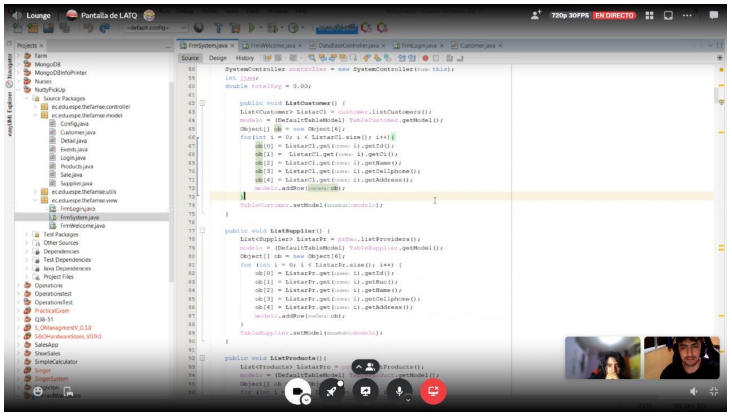
```
public class FrmSystem extends javax.swing.JFrame {
    Customer cl = new Customer();
    CustomerController customer = CustomerController.getInstance();
    Supplier sp = new Supplier();
    SupplierController prDao =  SupplierController.getInstance();
    Products pro = new Products();
    ProductsController proDao = ProductsController.getInstance();
    Sale s = new Sale();
    SaleController sdb = SaleController.getInstance();
    Detail di = new Detail();
    Config cf = new Config();
    Events event = new Events();
    DefaultTableModel modelo = new DefaultTableModel();
    DefaultTableModel tmp = new DefaultTableModel();
    SystemController controller = new SystemController(form: this);
```

```
  public boolean deleteCustomer(ObjectId id) {
      try {
          Bson filter = Filters.eq(fieldName:"_id", value: id);
          DeleteResult result = collection.deleteOne(bson: filter);
          return result.getDeletedCount() > 0;
      } catch (MongoException e) {
          System.out.println(x: e.toString());
          return false;
      }
  }
```

```java
public boolean deleteCustomer(ObjectId id) {
    try {
        Bson filter = Filters.eq(fieldName:"_id", value: id);
        DeleteResult result = collection.deleteOne(bson: filter);
        return result.getDeletedCount() > 0;
    } catch (MongoException e) {
        System.out.println(x: e.toString());
        return false;
    }
}

public boolean updateCustomer(Customer customer) {
    try {
        Document query = new Document(key:"_id", value: customer.getId());
        Document update = new Document(key:"$set", value: new Document(key:"Ci", value: customer.getCi())
                .append(key:"name", value: customer.getName())
                .append(key:"cellphone", value: customer.getCellphone())
                .append(key:"address", value: customer.getAddress()));

        collection.updateOne(bson: query, bson1: update);

        return true;
    } catch (MongoException e) {
        System.out.println(x: e.toString());
        return false;
    }
}

public Customer searchCustomerByCI(int ci) {
    Document query = new Document(key:"ci", value: ci);
    Document result = collection.find(bson: query).first();

    Customer customer = new Customer();

    if (result != null) {
```

```java
public Customer(ObjectId _id,
    this._id = _id;
```

meet evidence



Functional Requirements user test (No CRUD operations)

NUTTY PICK UP

Ruc: 1726063223
Nombre: Laura Jacome
Telefono: 0888888
Dirección: Quitumbe
Razon: Tienda De Frutos Secos

Factura:64e74bc70352
9c096f7954e0
Fecha: 24-23-2023

Datos de los Clientes

Ci/Ruc:	Nombre:	Telefono:	Dirección:
501365845	Leonel Tipan	0992314893	Centro Historico

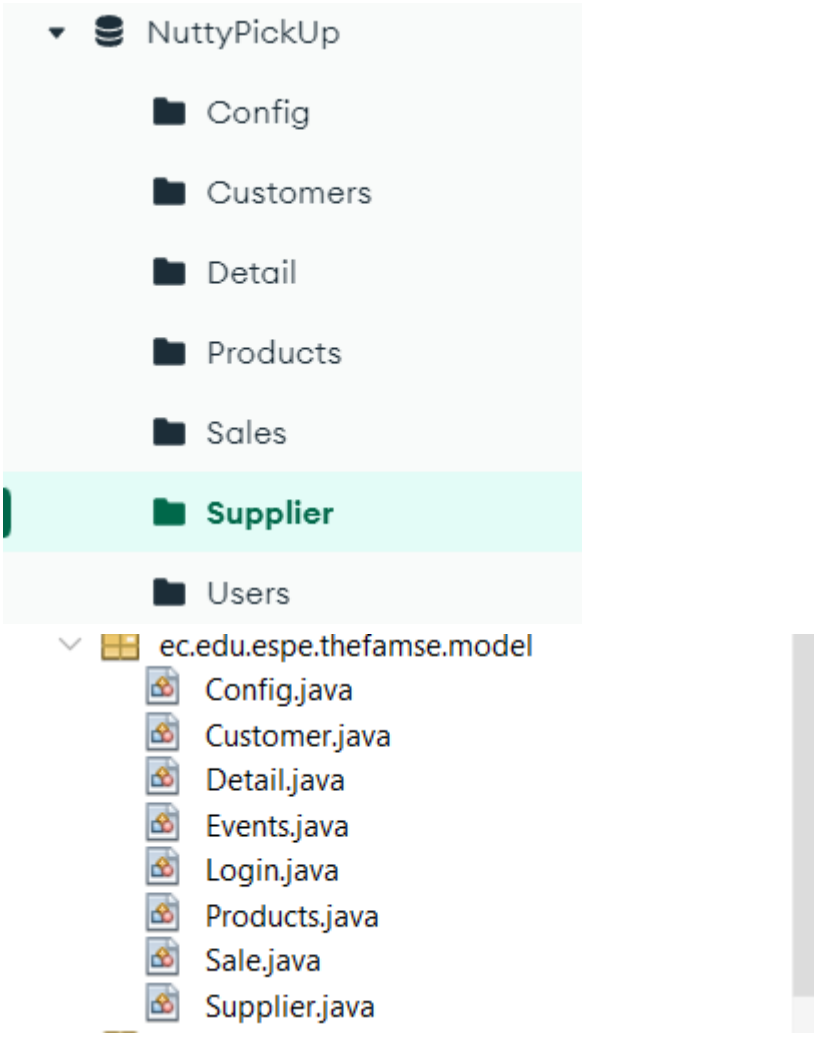
Cantidad:	Descripción:	Precio U:	Precio T:
5	Nuez	5.0	5.0
4	Peras	9.8	9.8
5	Pistacho	7.0	7.0

Total a Pagar: 21.8

Cancelación y Firma

Gracias por su Compra

MongoDB



Design patterns code

Project Maintenance
Project Letter

n

Team 6: Code Warriors **Leader:** Edison Verdesoto

24. **QUISHPE LLUMIQUINGA JOHN MATEO**
25. **TIPAN QUINTO LEONEL ALEJANDRO**
26. **VERDESOTO SEGOVIA EDISON DAMIAN**

Inspector: Team 1 Jsons **Leader:** Yeshua Amador
Project: S&O HardwareStore
GitHub: https://github.com/EDVerdesoto/T06_SandO-System.git

MongoDB Compass:
mongodb+srv://latipan06:latipan06@cluster0.kawghe4.mongodb.net/?retryWrites=true&w=majority
Grade: PR: /25, PM: /25, Letter: /25 →44 /75

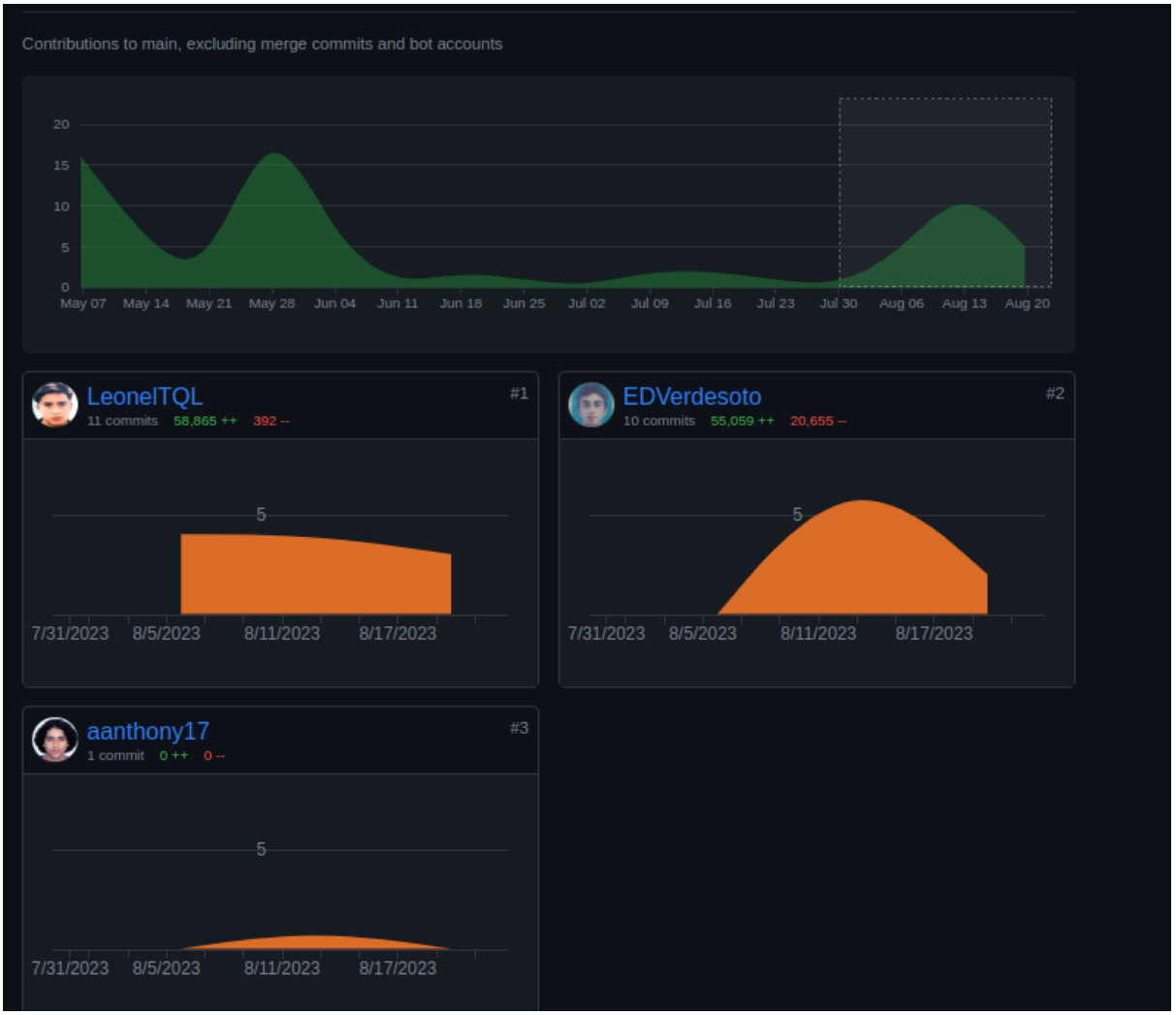
Rubric:
Project Review (Inspection) 17.5/25
Requirements and modeling (updated and design patterns) 4/5

Principles,CleanCode, Pitfalls 4/5
Functional Requirements user test (No CRUD operations) 4.5/5
MongoDB 5/5
Design patterns code 0/5
Project Maintenance 10/25 (Time effort)
new user / user type 0/10
report screen 5/10
print out 5/5

Project GitHub /25 (Individual Contribution)

Project Letter 15/25 (Letter of acceptance or review) missing contact information

Evidence:
Project GitHub



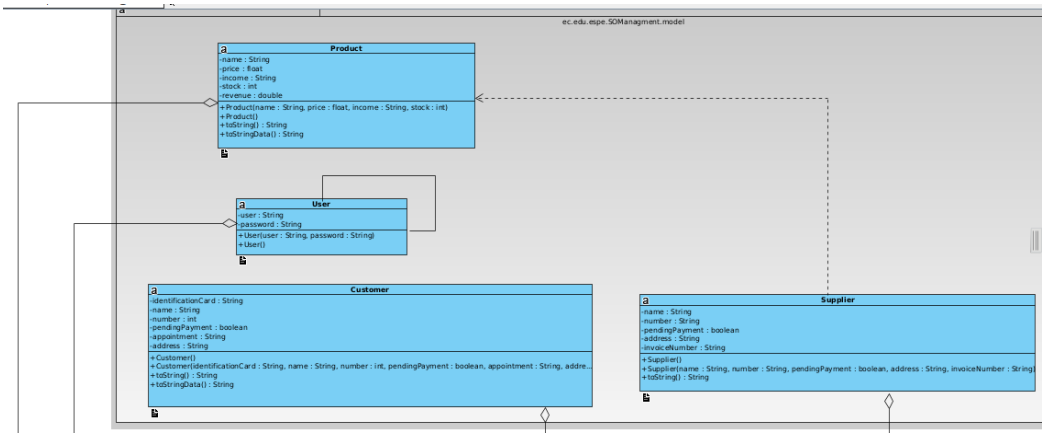
Project Review (Inspection)

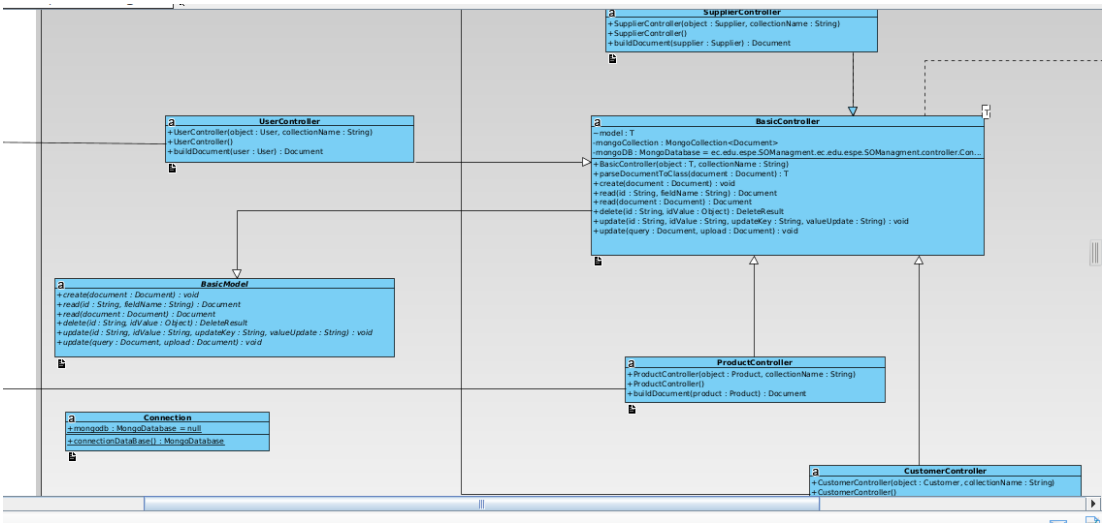
Requirements and modeling (updated and design patterns)

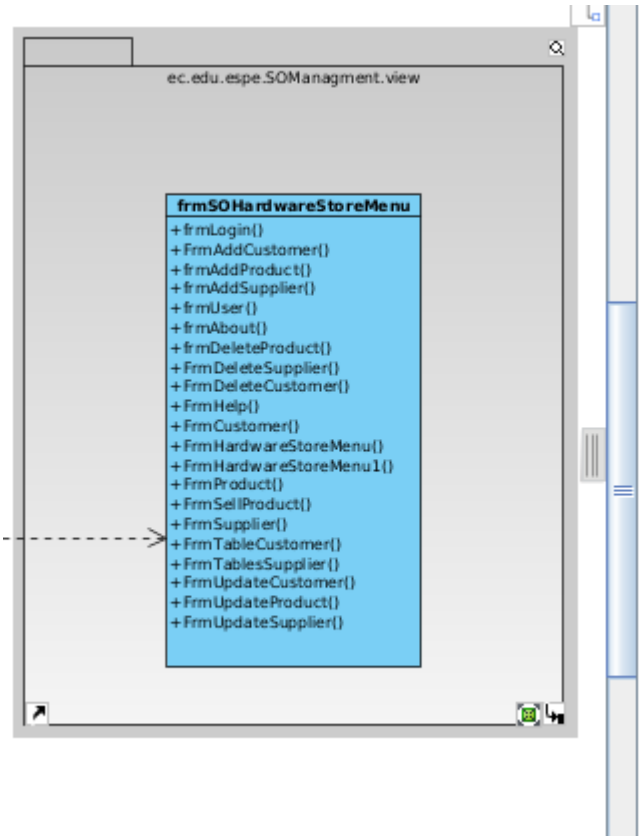
3.2. Functions

The software shall have the following functions:

- **Inventory management:**
- Add new inventory items: The user shall be able to add new items to the inventory database.
- Delete inventory items: The user shall be able to delete items for the inventory database.
- Update inventory items: The user shall be able to update the information of existing items in the inventory database.
- View inventory status: The user shall be able to view the current status of the inventory, including the quantity in stock and the location of the inventory.
- Calculate taxes and profits: The software shall automatically calculate the IVA and 30% of the products to determine the store's profit.
- User management: The program must allow more than one user.
- Sell Products: Update the BD by selling products but don't show the bill.








```java
 * @author Edison Verdesoto, Code Warriors, DCCO-ESPE
 */
public class Decimals {
    public static Double roundToTwoTenths(Double amountToRound){
        Double amountRounded;
        int dollars = (int) (amountToRound/1) ;
        Double cents = (amountToRound%1)*100;

        if(cents == 0){
            amountRounded = amountToRound;
        }
        else {
            cents = (double) Math.round(a: cents);
            amountRounded = dollars + (cents/100);
        }

        return amountRounded;
```

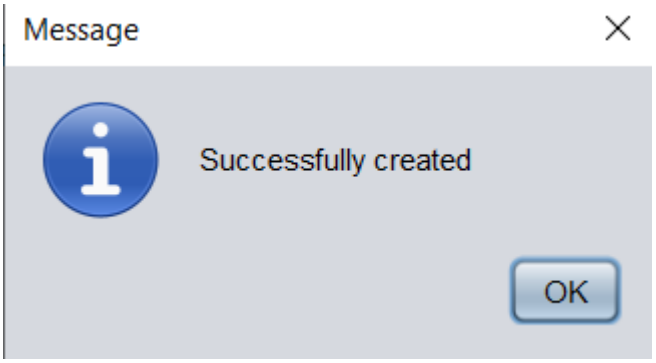
Principles,CleanCode, Pitfalls

```java
public static boolean validateID(String identificationCard) {

    int suma = 0;

    if (identificationCard.length() != 10) {

        return false;
    }

    for (int i = 0; i < 9; i++) {
        int digit = Integer.parseInt(identificationCard.charAt(index: i) + "");
        if (i % 2 == 0) {
            digit = digit * 2;
            if (digit > 9) {
                digit -= 9;
            }
```

```java
    String identificationCard;
String name;
int number;
boolean pendigPayment;
boolean validateId = true;
String appointment;
String address;

identificationCard = txtId.getText();
name = txtName.getText();
number = Integer.parseInt(s: txtNumber.getText());
pendigPayment = false;
appointment = formDate.format(date:txtDateService.getDate());
address = txtAddress.getText();

validateId = IdentificationCardController.validateID(identificationCard);
```

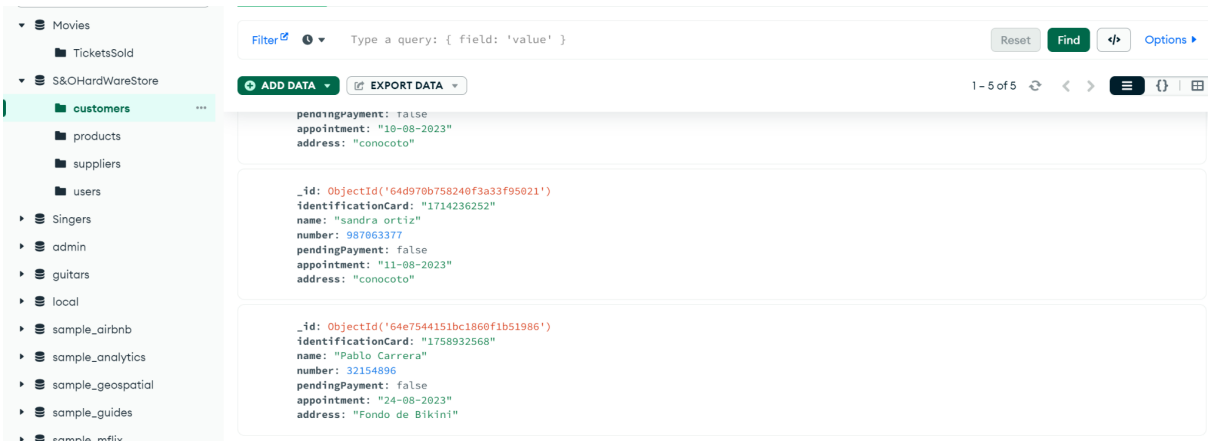
Functional Requirements user test (No CRUD operations)

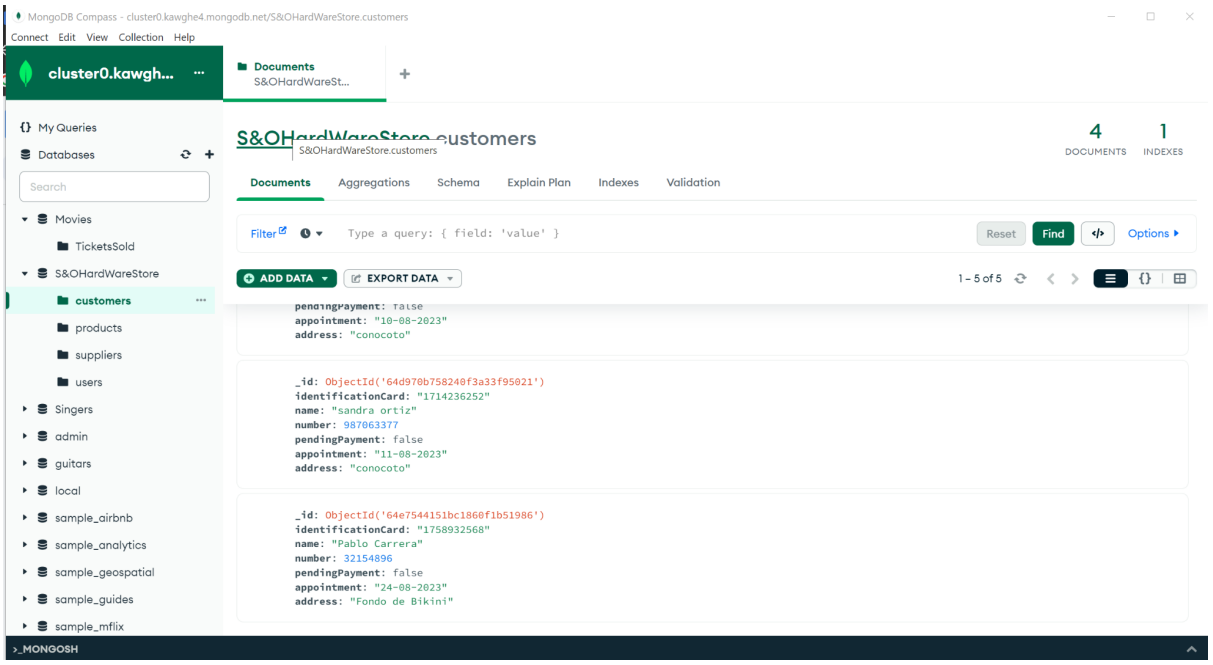
–Second functional requirement: PVP with IVA and 30% of the initial price

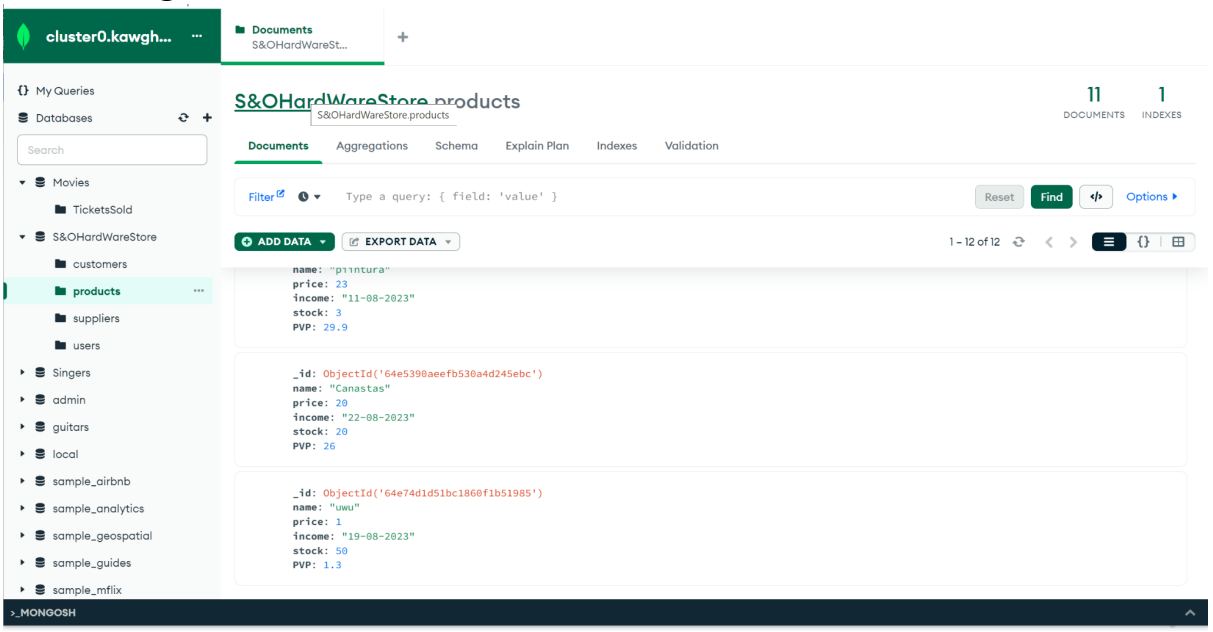


Nombre	Fecha de ingreso	Costo	Stock	Precio de venta
Brochas lizas	11-07-2023	5.0	29	6.5
Serrucho	16-07-2023	30.0	20	39.0
Pintura en spray	14-07-2023	20.0	17	26.0
Cenemto de cont...	17-07-2023	4.0	64	5.2
Tornillo en cruz	12-08-2023	14.0	18	18.2
Tornillo plano	25-07-2023	50.0	89	65.0
Destornillador	25-07-2023	20.5	83	26.65
Martillo	25-07-2023	20.5	89	26.65
Cinta de embalaje	25-07-2023	20.5	23	26.65
piintura	11-08-2023	23.0	3	29.9
Canastas	22-08-2023	20.0	20	26.0
uwu	19-08-2023	1.0	0	1.3
Tijeras	24-08-2023	15.0	10	19.5

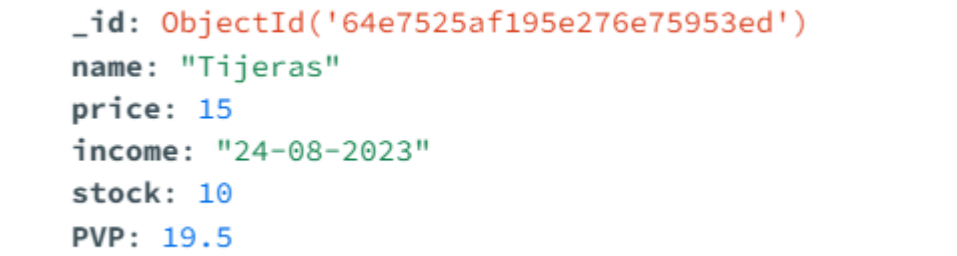
Creating Users Modified:


Movies
TicketsSold
S&OHardWareStore
customers
products
suppliers
users
Singers
admin
guitars
local
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
Filter
Type a query: { field: 'value' }
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 5 of 5
pendingPayment: false
appointment: "10-08-2023"
address: "conocoto"
_id: ObjectId('64d970b758240f3a33f95021')
identificationCard: "1714236252"
name: "sandra ortiz"
number: 987063377
pendingPayment: false
appointment: "11-08-2023"
address: "conocoto"
_id: ObjectId('64e7544151bc1860f1b51986')
identificationCard: "1758932568"
name: "Pablo Carrera"
number: 32154896
pendingPayment: false
appointment: "24-08-2023"
address: "Fondo de Bikini"


MongoDB Compass - cluster0.kawghe4.mongodb.net/S&OHardWareStore.customers
Connect Edit View Collection Help
cluster0.kawgh...
Documents
S&OHardWareSt...
My Queries
Databases
Search
S&OHardWareStore.customers
4 DOCUMENTS
1 INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Movies
TicketsSold
S&OHardWareStore
customers
products
suppliers
users
Singers
admin
guitars
local
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
sample_mflix
Filter
Type a query: { field: 'value' }
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 5 of 5
pendingPayment: false
appointment: "10-08-2023"
address: "conocoto"
_id: ObjectId('64d970b758240f3a33f95021')
identificationCard: "1714236252"
name: "sandra ortiz"
number: 987063377
pendingPayment: false
appointment: "11-08-2023"
address: "conocoto"
_id: ObjectId('64e7544151bc1860f1b51986')
identificationCard: "1758932568"
name: "Pablo Carrera"
number: 32154896
pendingPayment: false
appointment: "24-08-2023"
address: "Fondo de Bikini"
>_MONGOSH


cluster0.kawgh...
Documents
S&OHardWareSt...
My Queries
Databases
Search
S&OHardWareStore.products
11 DOCUMENTS
1 INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Movies
TicketsSold
S&OHardWareStore
customers
products
suppliers
users
Singers
admin
guitars
local
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
sample_mflix
Filter
Type a query: { field: 'value' }
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 12 of 12
name: "pintura"
price: 23
income: "11-08-2023"
stock: 3
PVP: 29.9
_id: ObjectId('64e5390aeefb530a4d245ebc')
name: "Canastas"
price: 20
income: "22-08-2023"
stock: 20
PVP: 26
_id: ObjectId('64e74d1d51bc1860f1b51985')
name: "uwu"
price: 1
income: "19-08-2023"
stock: 50
PVP: 1.3
>_MONGOSH

-Second functional requirement updated in database



-Third functional requirement updated in database
CRUD modifications in MongoDB

Design patterns code 0/5

Project Maintenance
Project Letter