UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Object-oriented programming

NRC: 9642

Workshop N° 12

Topic: Project Review

Team 6:Code Warriors
Project's name: S&O Hardware Store
Leader: Edison Damian Verdesoto Segovia
GitHub: https://github.com/EDVerdesoto/T06_SandO-System.git
Inspector: Team 1 , Yeshua Chiliquinga

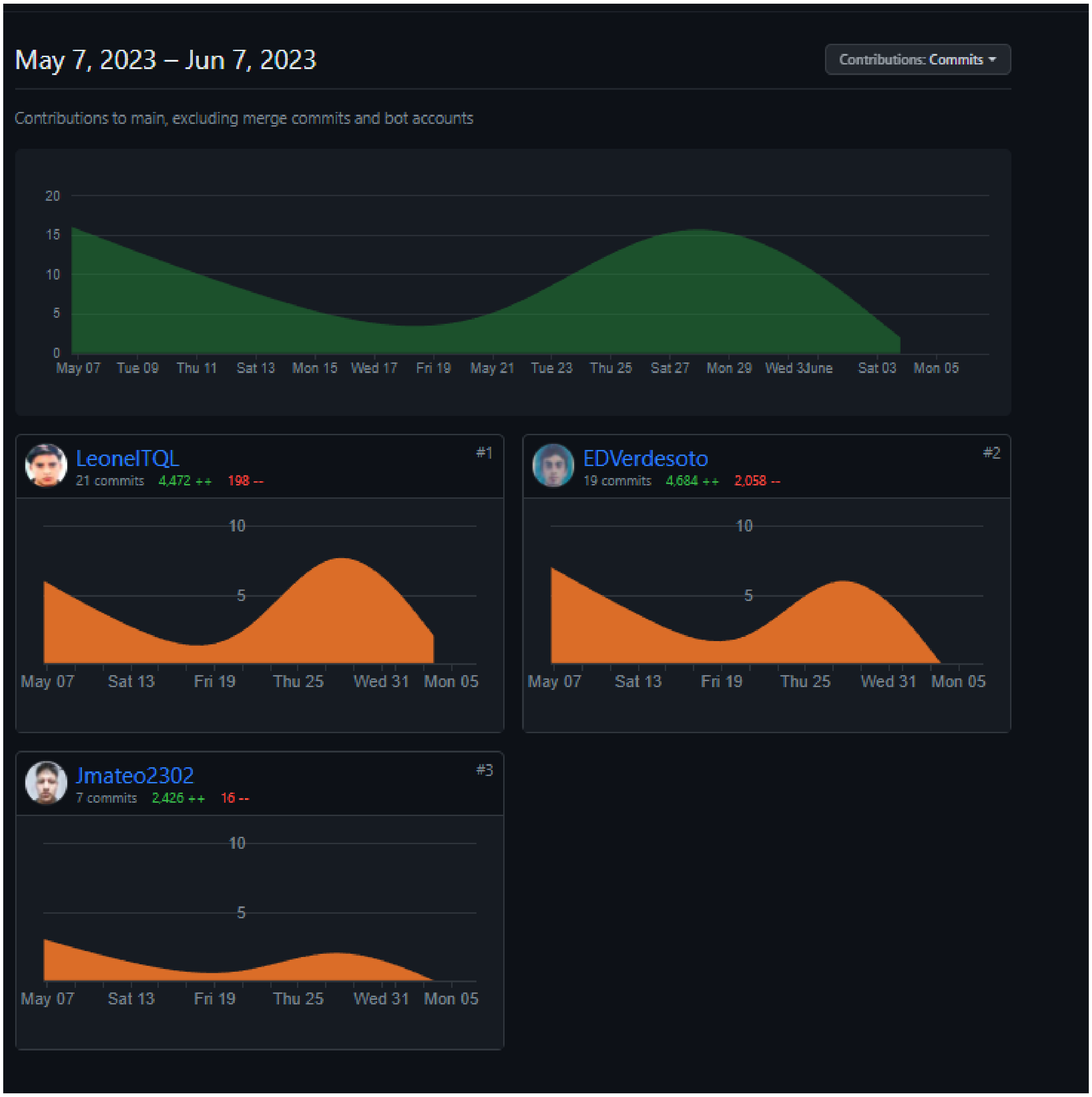
1. JOHN MATEO QUISHPE LLUMIQUINGA
2. LEONEL ALEJANDRO TIPAN QUINTO
3. EDISON DAMIAN VERDESOTO SEGOVIA
   Presentation: 8.5/10

Rubric

Presentation
0.- GitHub /100
1.- Interview and Presentation 10/10 (All the members appear in the video)
2.- Project definition 9/10 (In the presentation the team said that the program should save data in csv/excel files)
3.- Software Requirements 7.5/10 (The requirements do not follow the format of numbering IEEE-830 in the functional requirements and there are no images of the Class Diagram nor the Use Case Diagram)
4.- Use Case 10/10(Everything is following the specifications)
5.- Class Diagram 10/10(The diagram is according with the Requirements and the use case)
6.- Data files 1/10(The program does not generate the json file)
7.- Execution 9/10(The program executes very well but it has one bug detected)
8-10.- Clean Code 24/30 (The class name “ToSellItem” is incorrect, because it is a verb and there is no code [it is empty]; after reviewing the requirements this class wasn’t found, also there are some attributes names that are verbs like enterDate, request, sellPercentage, and a method that is not a verb “supplierContacts”. Finally there is a Scanner sc that is not used.)
TOTAL 80.5/100

Evidence

GitHub



May 7, 2023 – Jun 7, 2023

Contributions: Commits

Contributions to main, excluding merge commits and bot accounts

LeonelTQL #1
21 commits 4,472 ++ 198 --

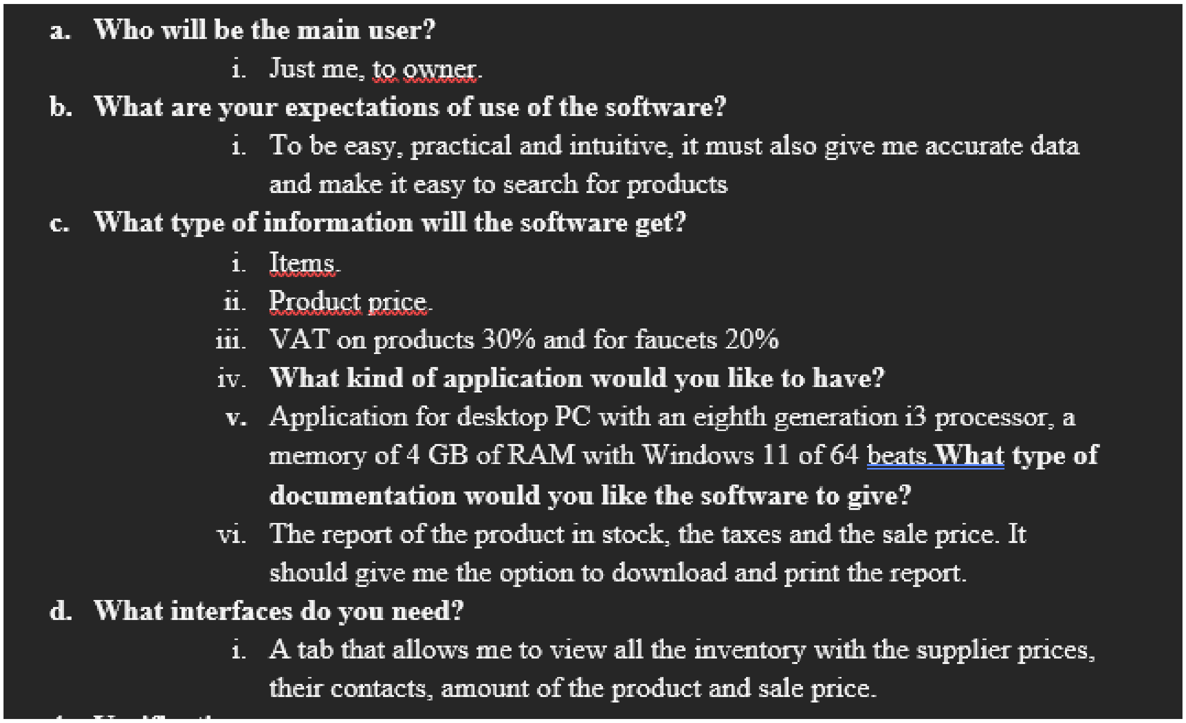
EDVerdesoto #2
19 commits 4,684 ++ 2,058 --

Jmateo2302 #3
7 commits 2,426 ++ 16 --

1.- Interview and Presentation

https://www.youtube.com/watch?v=4yvGGJaW0A4&t=4s&ab_channel=DamianVerdesoto

2.- Project definition



a. **Who will be the main user?**
   i. Just me, to owner.

b. **What are your expectations of use of the software?**
   i. To be easy, practical and intuitive, it must also give me accurate data and make it easy to search for products

c. **What type of information will the software get?**
   i. Items.
   ii. Product price.
   iii. VAT on products 30% and for faucets 20%
   iv. **What kind of application would you like to have?**
   v. Application for desktop PC with an eighth generation i3 processor, a memory of 4 GB of RAM with Windows 11 of 64 beats.**What type of documentation would you like the software to give?**
   vi. The report of the product in stock, the taxes and the sale price. It should give me the option to download and print the report.

d. **What interfaces do you need?**
   i. A tab that allows me to view all the inventory with the supplier prices, their contacts, amount of the product and sale price.

3.- Software Requirements

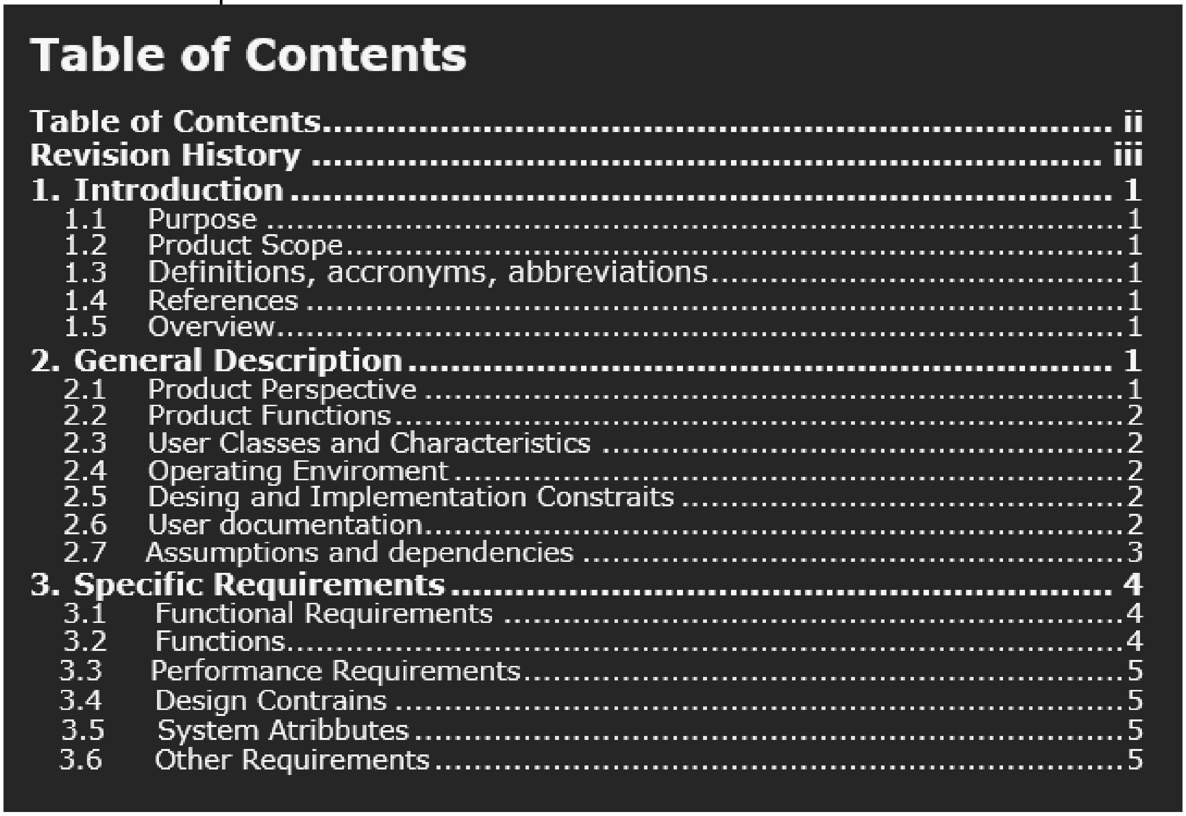
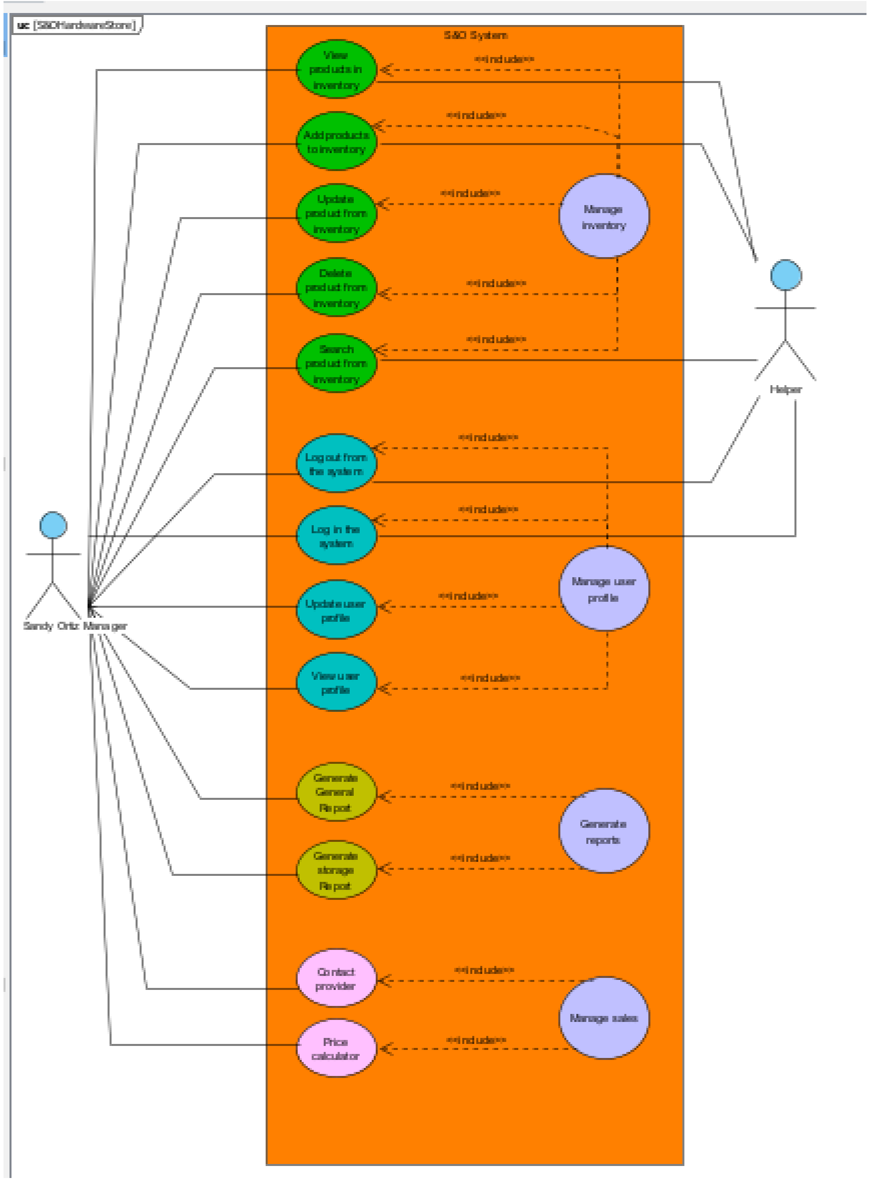
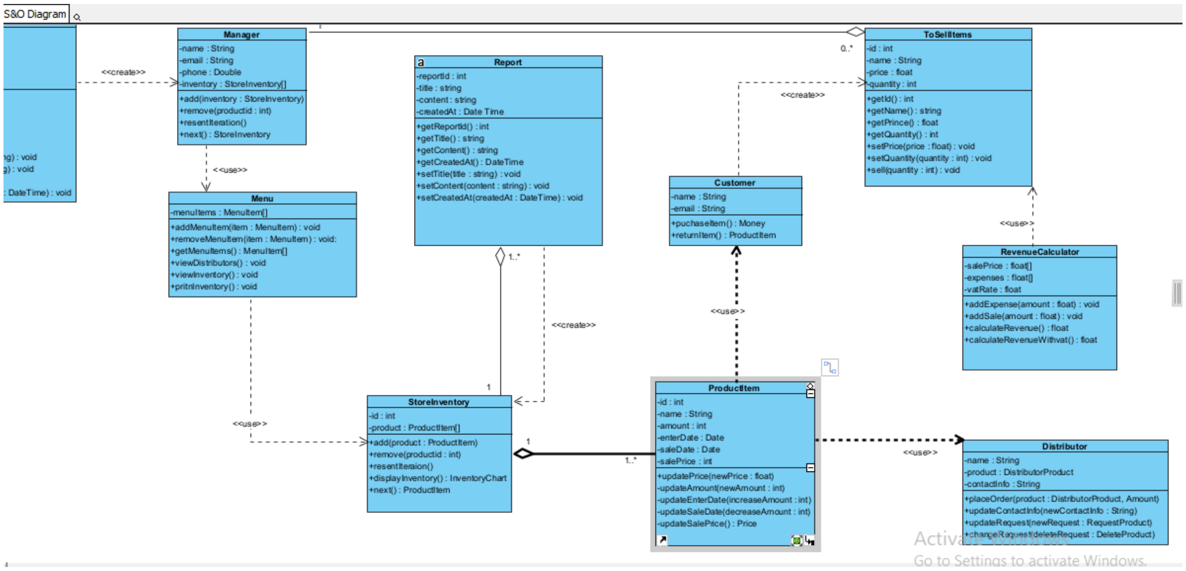


Table of Contents

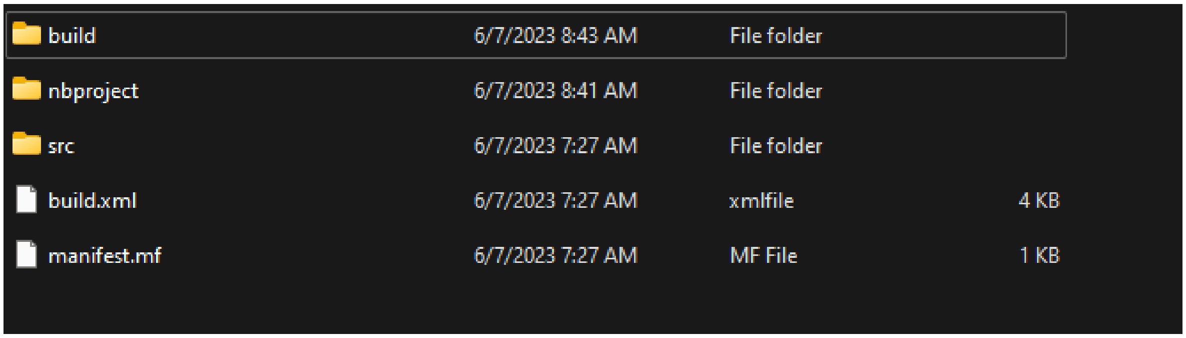
4.- Use Case



5.- Class Diagram

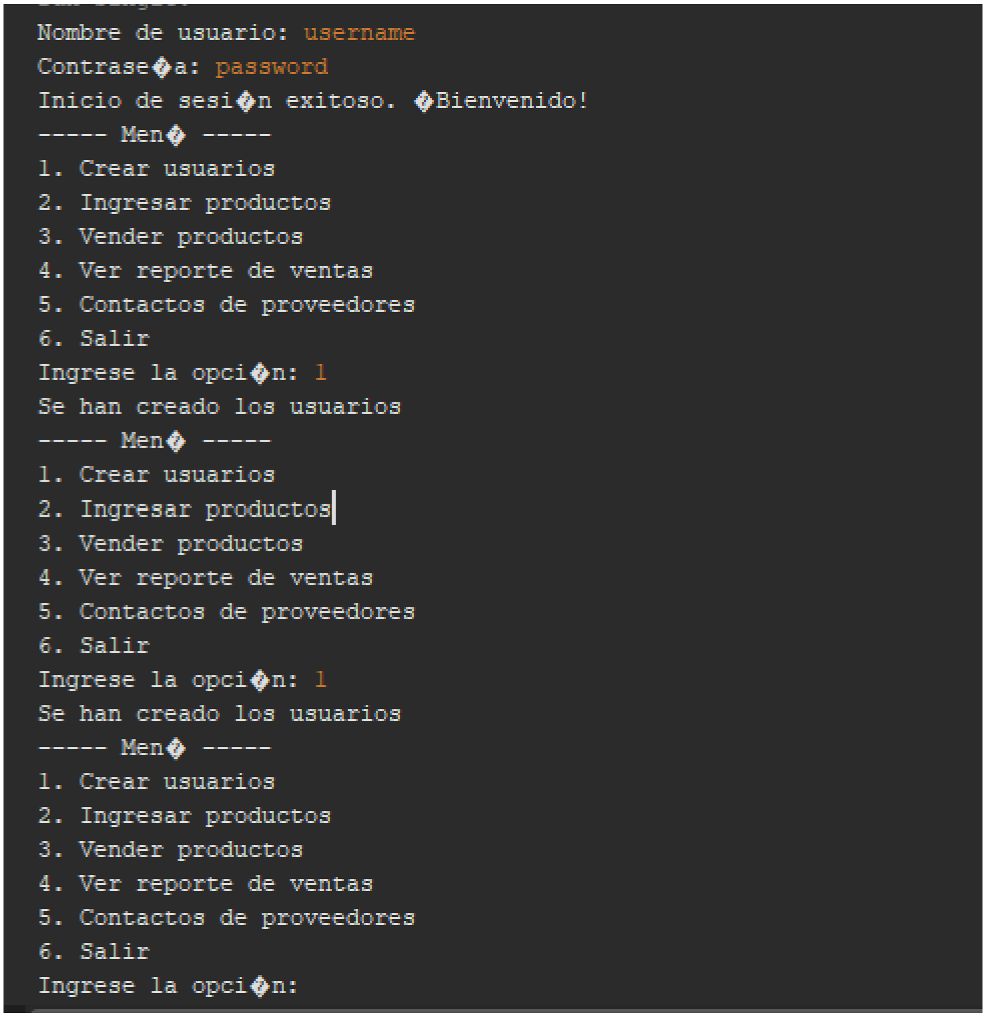

S&O Diagram
ng) : void
g) : void
: DateTime) : void
<<create>>
Manager
-name : String
-email : String
-phone : Double
-inventory : StoreInventory[]
+add(inventory : StoreInventory)
+remove(productid : int)
+resentIteration()
+next() : StoreInventory
<<use>>
Menu
-menuItems : MenuItem[]
+addMenuItem(item : MenuItem) : void
+removeMenuItem(item : MenuItem) : void:
+getMenuItems() : MenuItem[]
+viewDistributors() : void
+viewInventory() : void
+pritnInventory() : void
Report
-reportId : int
-title : string
-content : string
-createdAt : Date Time
+getReportId() : int
+getTitle() : string
+getContent() : string
+getCreatedAt() : DateTime
+setTitle(title : string) : void
+setContent(content : string) : void
+setCreatedAt(createdAt : DateTime) : void
1..*
<<create>>
1
0..*
ToSellItems
-id : int
-name : String
-price : float
-quantity : int
+getId() : int
+getName() : string
+getPrince() : float
+getQuantity() : int
+setPrice(price : float) : void
+setQuantity(quantity : int) : void
+sell(quantity : int) : void
<<create>>
Customer
-name : String
-email : String
+puchaseItem() : Money
+returnItem() : ProductItem
<<use>>
RevenueCalculator
-salePrice : float[]
-expenses : float[]
-vatRate : float
+addExpense(amount : float) : void
+addSale(amount : float) : void
+calculateRevenue() : float
+calculateRevenueWithvat() : float
<<use>>
<<use>>
StoreInventory
-id : int
-product : ProductItem[]
+add(product : ProductItem)
+remove(productid : int)
+resentIteraion()
+displayInventory() : InventoryChart
+next() : ProductItem
1
1..*
ProductItem
-id : int
-name : String
-amount : int
-enterDate : Date
-saleDate : Date
-salePrice : int
+updatePrice(newPrice : float)
-updateAmount(newAmount : int)
-updateEnterDate(increaseAmount : int)
-updateSaleDate(decreaseAmount : int)
-updateSalePrice() : Price
<<use>>
Distributor
-name : String
-product : DistributorProduct
-contactInfo : String
+placeOrder(product : DistributorProduct, Amount)
+updateContactInfo(newContactInfo : String)
+updateRequest(newRequest : RequestProduct)
+changeRequest(deleteRequest : DeleteProduct)

6.- Data files



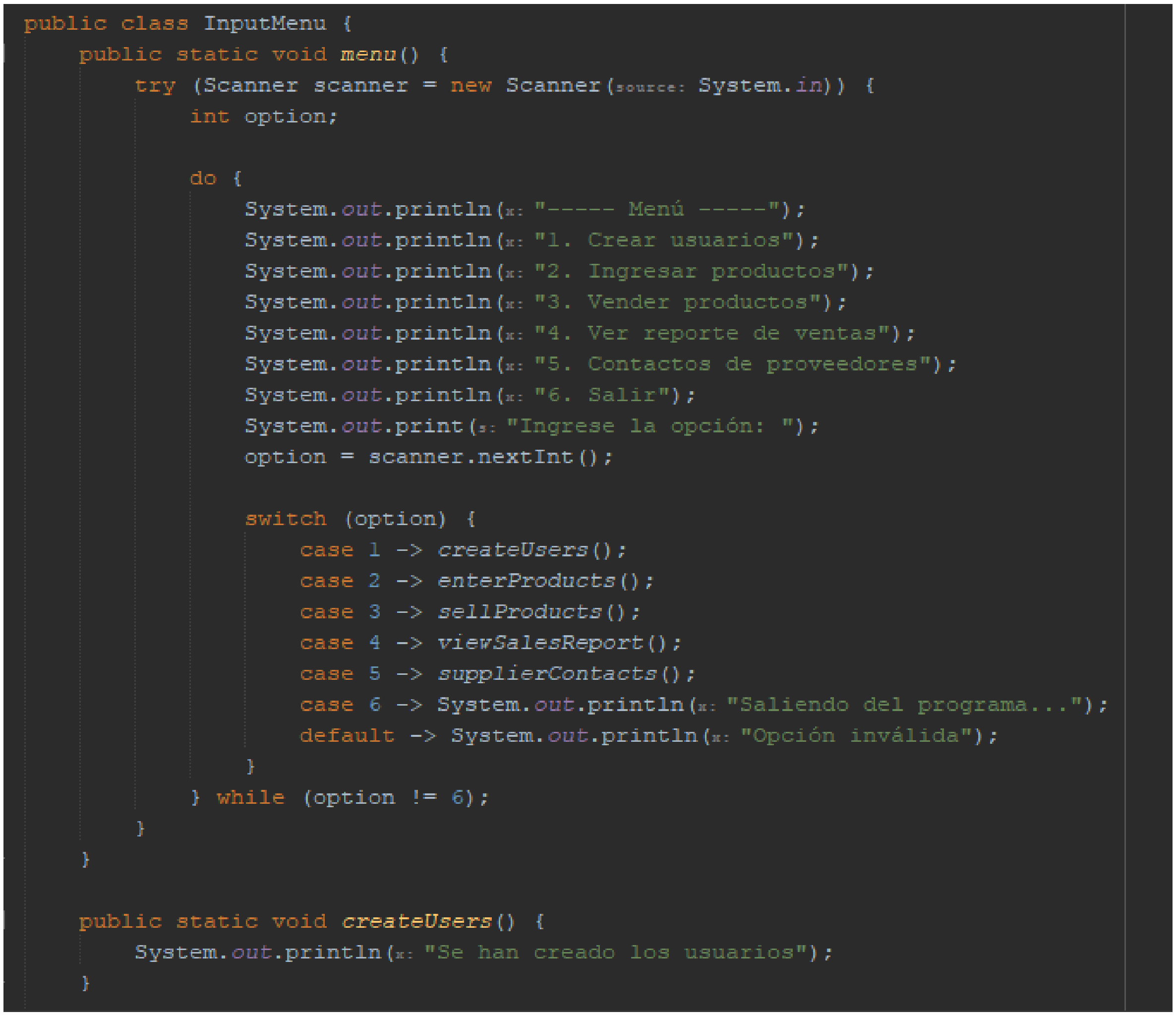
```
build          6/7/2023 8:43 AM    File folder
nbproject      6/7/2023 8:41 AM    File folder
src            6/7/2023 7:27 AM    File folder
build.xml      6/7/2023 7:27 AM    xmlfile          4 KB
manifest.mf    6/7/2023 7:27 AM    MF File          1 KB
```

7.- Execution

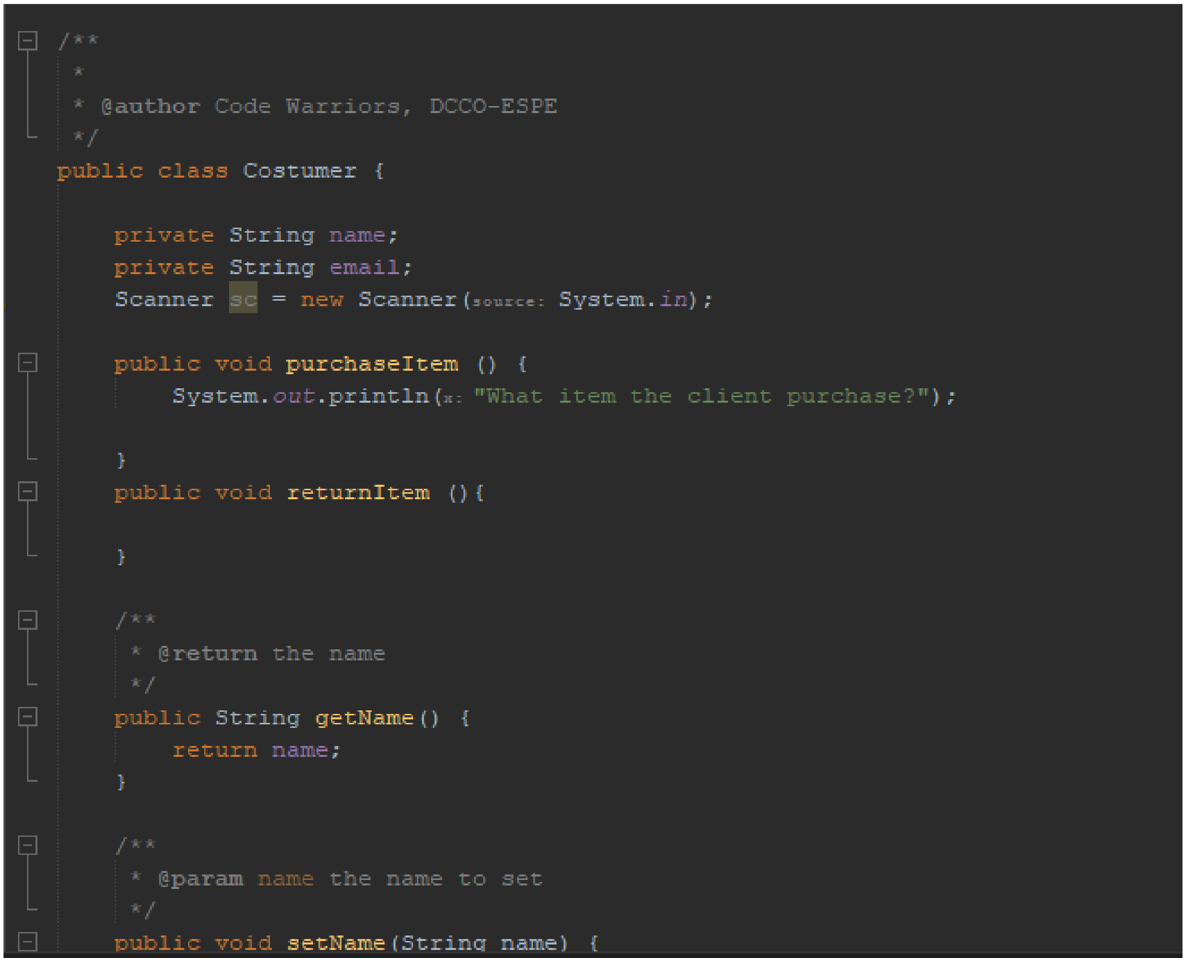


```
Nombre de usuario: username
Contrase�a: password
Inicio de sesi�n exitoso. �Bienvenido!
----- Men� -----
1. Crear usuarios
2. Ingresar productos
3. Vender productos
4. Ver reporte de ventas
5. Contactos de proveedores
6. Salir
Ingrese la opci�n: 1
Se han creado los usuarios
----- Men� -----
1. Crear usuarios
2. Ingresar productos
3. Vender productos
4. Ver reporte de ventas
5. Contactos de proveedores
6. Salir
Ingrese la opci�n: 1
Se han creado los usuarios
----- Men� -----
1. Crear usuarios
2. Ingresar productos
3. Vender productos
4. Ver reporte de ventas
5. Contactos de proveedores
6. Salir
Ingrese la opci�n:
```

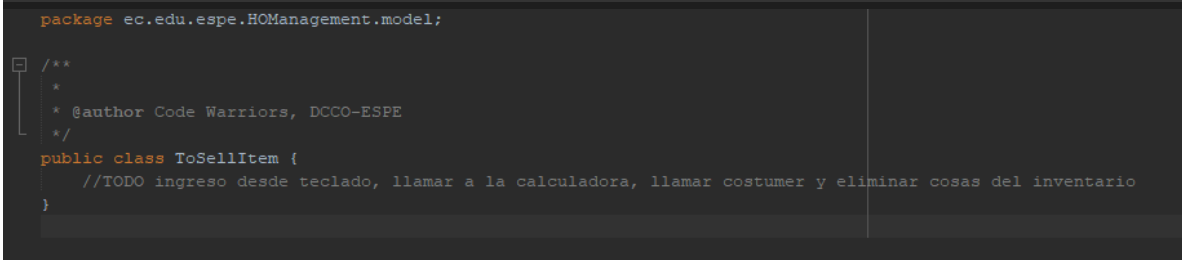
8-10.- Clean Code



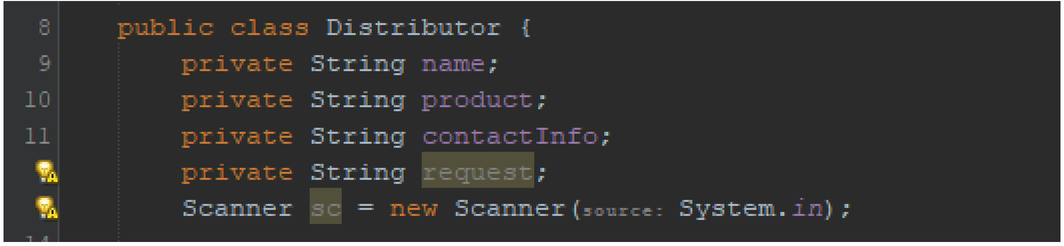
```
public class InputMenu {
    public static void menu() {
        try (Scanner scanner = new Scanner(source: System.in)) {
            int option;

            do {
                System.out.println(x: "----- Menú -----");
                System.out.println(x: "1. Crear usuarios");
                System.out.println(x: "2. Ingresar productos");
                System.out.println(x: "3. Vender productos");
                System.out.println(x: "4. Ver reporte de ventas");
                System.out.println(x: "5. Contactos de proveedores");
                System.out.println(x: "6. Salir");
                System.out.print(s: "Ingrese la opción: ");
                option = scanner.nextInt();

                switch (option) {
                    case 1 -> createUsers();
                    case 2 -> enterProducts();
                    case 3 -> sellProducts();
                    case 4 -> viewSalesReport();
                    case 5 -> supplierContacts();
                    case 6 -> System.out.println(x: "Saliendo del programa...");
                    default -> System.out.println(x: "Opción inválida");
                }
            } while (option != 6);
        }
    }

    public static void createUsers() {
        System.out.println(x: "Se han creado los usuarios");
    }
```



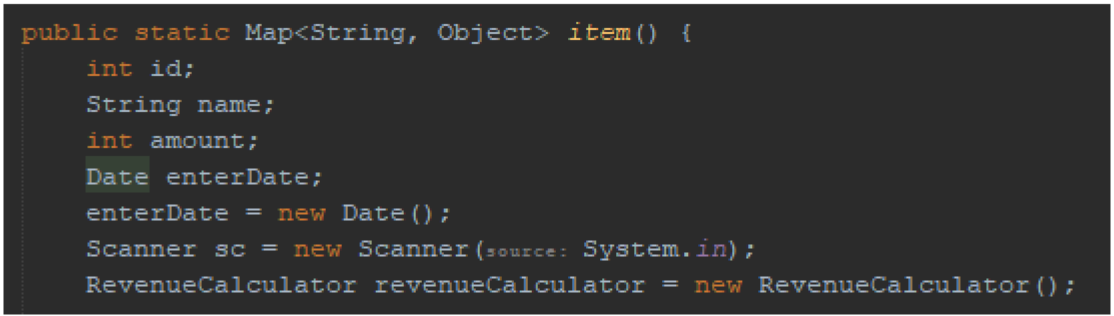
```java
/**
 *
 * @author Code Warriors, DCCO-ESPE
 */
public class Costumer {

    private String name;
    private String email;
    Scanner sc = new Scanner(source: System.in);

    public void purchaseItem () {
        System.out.println(x: "What item the client purchase?");

    }
    public void returnItem (){

    }

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }

    /**
     * @param name the name to set
     */
    public void setName(String name) {
```



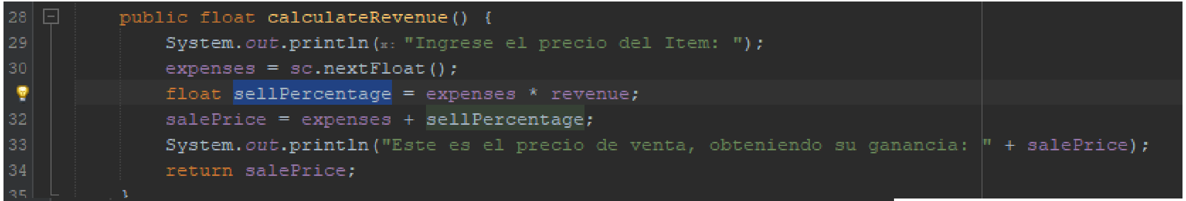
```java
package ec.edu.espe.HOManagement.model;

/**
 *
 * @author Code Warriors, DCCO-ESPE
 */
public class ToSellItem {
    //TODO ingreso desde teclado, llamar a la calculadora, llamar costumer y eliminar cosas del inventario
}
```



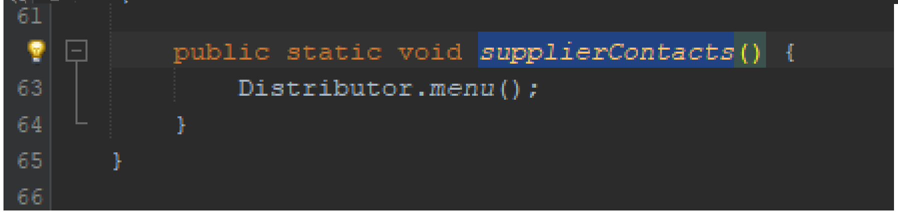
```java
public class Distributor {
    private String name;
    private String product;
    private String contactInfo;
    private String request;
    Scanner sc = new Scanner(source: System.in);
```



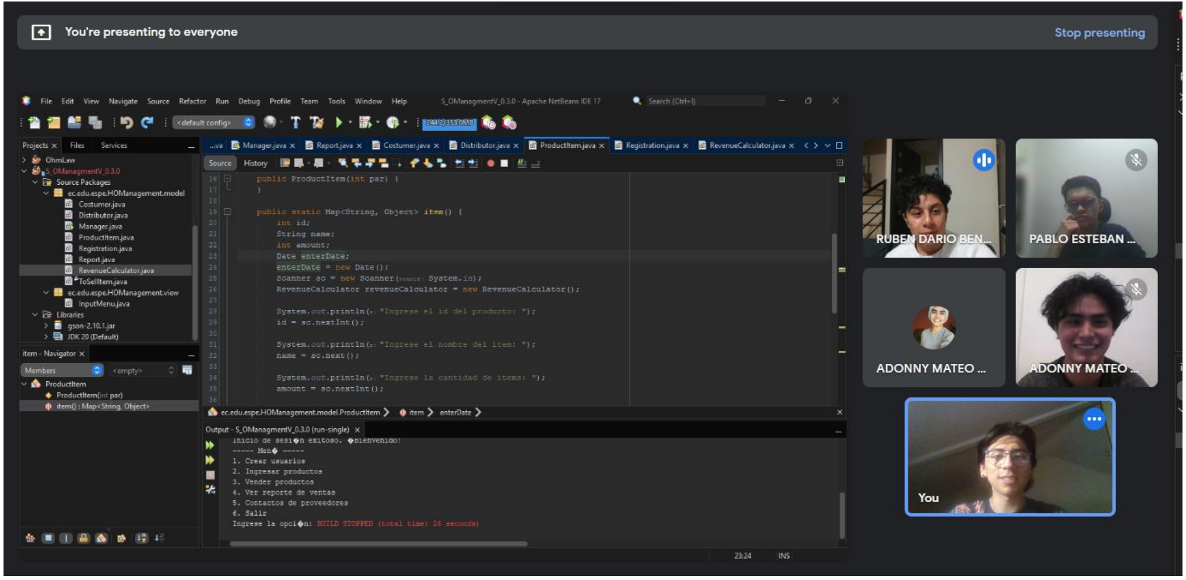
```java
public static Map<String, Object> item() {
    int id;
    String name;
    int amount;
    Date enterDate;
    enterDate = new Date();
    Scanner sc = new Scanner(source: System.in);
    RevenueCalculator revenueCalculator = new RevenueCalculator();
```



```java
    public float calculateRevenue() {
        System.out.println(x: "Ingrese el precio del Item: ");
        expenses = sc.nextFloat();
        float sellPercentage = expenses * revenue;
        salePrice = expenses + sellPercentage;
        System.out.println("Este es el precio de venta, obteniendo su ganancia: " + salePrice);
        return salePrice;
    }
```



```java
    public static void supplierContacts() {
        Distributor.menu();
    }
}
```



```java
    public ProductItem(int par) {
    }

    public static Map<String, Object> item() {
        int id;
        String name;
        int amount;
        Date enterDate;
        enterDate = new Date();
        Scanner sc = new Scanner(source: System.in);
        RevenueCalculator revenueCalculator = new RevenueCalculator();

        System.out.println(x: "Ingrese el id del producto: ");
        id = sc.nextInt();

        System.out.println(x: "Ingrese el nombre del item: ");
        name = sc.next();

        System.out.println(x: "Ingrese la cantidad de items: ");
        amount = sc.nextInt();
```

```
Inicio de sesi�n exitoso. �Bienvenido!
----- Men� -----
1. Crear usuarios
2. Ingresar productos
3. Vender productos
4. Ver reporte de ventas
5. Contactos de proveedores
6. Salir
Ingrese la opci�n: BUILD STOPPED (total time: 26 seconds)
```

Evidence Notes:

Report of Inspection

Team: 1 Leader: Yeshua Amador Chiliquinga Amaya

Name: Jsons

Team Inspected: 6

Team Inspected Name: Code Warriors

0. Github

Screenshot of activity in the repository

1.- Televideo

Checked the video and if all the team participated there. Result: 10

2.- Project Definition

We checked if it is coincident with the requirementes

Result: 9, because it was not proved that the information was saved in Excel during the presentation.

3.- Requirements

We reviewed that the document complies with the I-EEE 830 standard of SRS and all of its parts. Result: 7.5, because the requirements do not follow the format of numbering IEEE 830 in the functional requirements, Also, there is no class and Use case diagrams images

4.- Use Case

The Use Case diagram has to be coincident with the functional requirements.

Result: 10, because everything is coincident with the requirements.

5.- Class Diagram

The class diagram has to be coincident with the functional requirements

Result: 10, because everything is coincident with the requirements

6.- Data Files

About Jsons, these or this has to be consistent and working well. Result:

Because, jsons are not being created

7.- Execution

Review that code is working well according to the functional requirements.

Result: 9 Because has an error founded

Review if code follows a clean code format according to the good practices we saw during the unit 4

Results: The class name "ToSellItem" is incorrect and there is no code (it is empty); after reviewing the requirements this class wasn't found, also there are some attributes names that are verbs like enter Date, request, sellPercentage, and a method that is not a verb "SupplierContacts". Finally there is a Scanner so that is not used