|  |
| --- |
| DISCON 2013 |
| Especificació Memòria Codi i Test Errors |
| Versió 1.5 |

|  |
| --- |
|  |

# Historial de Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versió** | **Descripció** | **Autor** |
| 27/02/2013 | 0.0 | Esquelet Projecte | TOTS |
| 10/04/2013 | 1.0 | Primera revisió d’errors | TOTS |
| 12/04/2013 | 1.1 | Revisió i correcció dels errors | TOTS |
| 15/04/2013 | 1.2 | Creació dels Test | TOTS |
| 20/05/2013 | 1.3 | Modificació del text | TOTS |
| 26/05/2013 | 1.4 | Modificació projecte | TOTS |
| 27/05/2013 | 1.5 | Retocs finals | TOTS |

# Índex

[1- Historial de Revisions 2](#_Toc357462118)

[2- Índex 3](#_Toc357462119)

[3- Introducció 4](#_Toc357462120)

[4- Errors modificats del codi 5](#_Toc357462121)

[Classe Catàleg 5](#_Toc357462122)

[Classe Client 5](#_Toc357462123)

[Classe Compra 7](#_Toc357462124)

[Classe Constantes 7](#_Toc357462125)

[Classe GestionarCompres 8](#_Toc357462126)

[Classe LiniaCompra 8](#_Toc357462127)

[Classe Main 8](#_Toc357462128)

[Classe Productes 9](#_Toc357462129)

[Altres 9](#_Toc357462130)

[5- Test 11](#_Toc357462131)

[Classe ConstantesTests 11](#_Toc357462132)

[Classe TestCataleg 11](#_Toc357462133)

[Classe TestClient 12](#_Toc357462134)

[Classe TestProducte 12](#_Toc357462135)

[6- Conclusió 13](#_Toc357462136)

# Introducció

La companyia comercial Discon Market treballa des de fa 25 anys en la venta de productes a particulars i disposa d'un establiment de venda al públic. La filosofia de l'empresa es: "Tots els productes que tenim estan a disposició del client", es a dir, que tot el local comercial ha de ser dedicat a la venda, fins i tot el magatzem.

La venda es realitza en el mateix local els dies laborables de 10 : 00 a 14 : 00 i de 17:00 a 22 : 00.

La companyia vol disposar d'un sistema informàtic per millorar la gestió de l'empresa i continuar creixent.

La companyia esta dividida des del punt de vista organitzatiu en quatre departaments:

1. Gestió de vendes.

2. Gestió d'entregues.

3. Gestió logística.

4. Gestió contable.

# Errors modificats del codi

## Classe Catàleg

**Descripció general:**

Aquesta classe ens mostrarà el catàleg a partir d’un fitxer. A cada línia del fitxer tenim guardat la informació dels productes que tenim al catàleg.

**Canvis implementats:**

* El primer canvi realitzat ha sigut modificar l’ordre de les posicions de les variables “idProd” i el “nombreProd” en la llista on tenim guardat el catàleg. S’ha de canviar per poder tractar la informació del producte en altres funcions.

A més hem modificat les següents funcions:

* writeProductToTxt()

Aquesta funció modificarà la forma en la que inserim el nostre producte dintre del txt i també la forma en la que el tractem.

* readProductStr()

Llegeix el producte i el passa a String perquè el puguem tractar.

* GetProd()

Hem posat aquest mètode com a “públic” per tal de poder accedir al objecte “producte” a tot el programa.

**Report d’errors:**

El principal error al executar per primer cop, és que no teníem un document a on llegir i guardar els productes. Una vegada generat aquest document, podem accedir al catàleg i poder modificar l’stock.

## Classe Client

**Descripció general:**

A través de uns paràmetres obtindrem les dades de un determinat client, a més aquesta classe contindrà dades com pot esser el login i el compte bancari.

**Canvis implementats:**

Hem creat variables noves i la primera funció que hem modificat es:

* Inicialització de variables.
* Sets i Gets.

Aquestes modificacions es fan a partir de les noves variables que hem creat.

* Identificar-se

Sense aquest mètode no podem comprovar que el nostre client es el correcte

**Report d’errors:**

Al inserir N usuaris el sistema no els reconeix correctament.

## Classe Compra

**Descripció general:**

Classe que permetrà fer tots els passos oportuns per a poder definir una compra ja sigui mirant el carro o guardant les dades en un arxiu seleccionat.

**Canvis implementats:**

Hem modificat algunes funcions com:

* EfectuarCompra()
* RegistrarCompra()
* AfegirProducteCarro()

Perquè ens modifiqui el stock que hi ha dintre del nostre txt i a més les compres realitzades pel usuari.

**Report d’errors:**

Sense fitxer inicial el numero de compres no es realitzava.

Error en la comptabilitat del producte, per solucionar aquest error es van tenir que canviar alguns paràmetres dintre de la classe.

## Classe Constantes

**Descripció general:**

Classe que conté totes les rutes necessàries tant de fitxers com a variables publiques estàtiques i també variables publiques finals. Això ho fem per poder accedir a elles des de qualsevol part del nostre programa i poder utilitzar-les com si fos un arxiu de configuració.

**Canvis implementats:**

Per la realització d’aquesta classe hem fet:

|  |
| --- |
| **public** **class** Constantes {  **private** **static** **final** String *Ruta* = "src/org/uab/etse/es/ES00/";  **public** **static** **final** String *RutaClient* = *Ruta* + "clients.txt";  **public** **static** **final** String *RutaProductes* = *Ruta* + "productes.txt";  **public** **static** **final** String *RutaRegistros* = *Ruta* + "registro.txt";  } |

## Classe GestionarCompres

**Descripció general:**

Aquesta classe ens permet modificar i gestionar amb certa facilitat les compres efectuades per un client. A partir de llegir les dades del client i veure els productes que hi ha al catàleg.

**Canvis implementats:**

* Dins el mètode “RegistrarClient” hem canviat l’ordre de l’incrementador ”pos” del bucle principal del mètode. Amb això ens assegurem de recórrer tota la llista d’usuaris registrats.

**Report d’errors:**

Mal funcionament de la identificació del client.

## Classe LiniaCompra

**Descripció general:**

Classe que guarda la informació, el nombre d’unitats i el preu final d’un producte seleccionat per l’usuari.

**Canvis implementats:**

Hem implementat totes els mètodes de la classe, ya que no estaven fets.

* Creació de les funcions set() i get().

**Report d’errors:**

Creació dels nous mètodes per poder obtenir els preus,nvis en algun get() i set().

## Classe Main

**Descripció general:**

Classe principal on cridarem seqüencialment funcions del nostre codi per executar el nostre programa. A mes a mes hemimplementat una petita interfície en la que interactuarem amb un usuari que es podrà loguejar, registrar o consultar les nostres opcions.

**Canvis implementats:**

DesconectarClient() ho hem inserit dintre d’aquesta classe a la opció 6 perquè no ens doni errors

* Modificació de la variable “gestion”. Hem passat la variable de “private static” a “public static”.
* Modifiquem el tipus de tractat de la condició “col·leccions” per utilitzar arrays.
* Afegim dintre del nostre menú principal el cas 7 perquè ens permeti sortir de l’aplicació.

**Report d’errors:**

Error al loguejar i desloguejar

Error dintre de la compra de productes

Error al actualitzar els txt i comprovar les dades.

## Classe Productes

Descripció general:

A partir d’un arxiu txt anirem actualitzant i obtenint les dades que tenen els productes, com el seu nom, el stock, el preu entre altres funcions.

Canvis implementats:

Hem modificat la funció:

* UpdateProdTxt()

Canviem la posició dins de la funció “UpdateProdTxt()” del “Getcost()” per el “GetUnitats()”

Hem modificat la funció “IdProdStr” de la classe ”Producte” a la següent forma:

* ProdDades[1] -> ProdDades[0]

Així en retornarà l’ID del nostre producte en comptes del nom del nostre producte.

A més utilitzem el str perquè ens doni tota la línia del nostre producte i poder modificar-la.

Report d’errors:

Error en la contabilitat del producte i en el tractament de els dades.

## Altres

Per poder generar tot el nostre sistema hem creat 3 fitxers anomenats:

* Clients.txt
* Productes.txt
* Registro.txt

Aquests fitxers s’aniran actualitzant amb les dades noves que poden ser d’un usuari, d’un producte o un registre de compra.

Aquest es un dels primers errors que vam trobar al codi, sense aquests fitxers, el codi no executava res.

# Test

## Classe ConstantesTests

Descripció general:

Classe Test efectuada per comprovar que la classe Constantes no tenia cap error per això hem aplicat uns determinats algorismes.

Canvis implementats:

Aquest test lo que ens farà serà comprovar que hi ha una ruta o un objecte que existeix.

Report d’errors:

No retorna res.

## Classe TestCataleg

Descripció general:

Test basat en la classe catàleg que buscarà alguna vulnerabilitat o error dins del nostre codi.

Canvis implementats:

- Hem inicialitzar els vectors "names" i "description" com "Vector<String> i, el vector "prizes" com "Vector<Float>

- Hem influït els .txt necessaris per comprovar els test i hem creat la classe "ConstantesTests.java" on declarem les rutes dels arxius.

- Hem inicialitzat la variable "catalegFilename" amb la ruta que tenim guardada a la classe "ConstantesTests.java"

- Hem fet una crida a la funció "testAfegirProducte" a la funció principal del test.

- A la funció "testContainsProducte" hem iniciat l’iterador com "Iterator<Producte>".

- Implementem la funció “TestSetProducte”, que afegeix a la llista un nou producte amb el nom, la descripció i el preu que l’introduim.

- Modifiquem la funció “testEliminarProductes”, a la que afegim el paràmetre “String numero” per poder passar-li a la funció “eliminarProducto” dins de la classe “Cataleg”

Report d’errors:

No afegeix productes bé.

## Classe TestClient

Descripció general:

Test per veure si la classe client conte algun error.

Canvis implementats:

-El primer canvi que hem tingut que realitzar ha sigut el ”SetUp” on hem afegit tots els test de la classe ( “testDefaultClient” , “testFull”, “testModifiers”, “testIdentificar”, “testCompra”).

Report d’errors:

A la que hi ha N usuaris ens dona errors.

## Classe TestProducte

Descripció general:

Classe que obtindrà la comprovació que un producte es correcte o esta ben definit, sense cap error.

Canvis implementats:

Dins d’aquesta classe utilitzarem diferents mètodes per a comprovar que podem inserir i veure si els productes estan correctament inserits o son iguals.

Report d’errors:

No hi ha

# Conclusió

Una vegada realitzat la feina de implementació del codi de l anterior equip de treball, podem afirma que s'han modificat una gran quantitat de línies de codi i que s'han realitzat tests per confirma que s'ha fet una bona implementació i que tot el codi es sòlid de manera que passa els tests descrits per el client per acceptar el treball realitzat.