

TFC Seguretat Informàtica
Sistema de micropagaments PayWord.

Jordi Farré Ramon

ETIG

Antoni Martinez Balleste

18/6/2004

1. Resum

La realització d'aquest TFC correspon a la realització de una pagina web de venda de elements multimèdia, de manera que es puguin realitzar els pagaments a traves del sistema PayWord. Aquest sistema es ideal per pagaments de petites quantitats. El funcionament a grans trets es el següent.

Un client que vol comprar en una botiga amb sistema de pagaments PayWord necessita de una aplicació anomenada moneder, aquesta aplicació generarà el nombre de monedes que es necessitin las entregará al servidor de la botiga online en el moment de fer la compra.

Per a realitzar el sistema en qüestió m'he basat en la programació orientada a objectes, utilitzant el llenguatge Java, d'aquesta manera podia aprofitar components que formen part de java (concretament del J2EE) com poden ser Servlets i JSP's que crec que s'adiuen força a las característiques de la aplicació i que en general m'han facilitat força la feina. A mes el fet de haver fet la aplicació en Java fa que sigui una aplicació portable a altres sistemes operatius i a diferents servidors de aplicacions.

En general podem dir que he obtingut un producte, apte per a proves, però que en principi no es apte per producció, ja que he obviat temes com la necessitat de certificats digitals, implantar mòdul de connexió segura en el servidor o generar la persistència en una base de dades (que proporcionarà un millor rendiment que no pas la persistència en fitxers).

Index de continguts

1. Resum.....	2
2. Cos de la memòria.....	6
2.1.Estat de l'art.....	6
2.1.1.Introducció als pagaments electronics.....	6
2.1.2.Introducció als micropagaments i avantatges sobre altres tipus de pagaments....	7
2.1.3.Funcionament del sistema PayWord.....	9
2.2. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.....	10
2.3. Objectius del TFC.....	10
2.4. Enfocament i mètode seguit.....	11
2.5. Planificació del projecte.....	12
2.6. Productes obtinguts.....	14
2.7. Breu descripció de la resta de capítols de la memòria.....	14
3. Disseny.....	15
3.1. Casos d'us.....	16
3.1.1. Cas d'us numero 1: Afegir Client.....	16
3.1.2. Cas d'us numero 2: Consultar o Modificar Client.....	16
3.1.3. Cas d'us numero 3: Esborrar Client.....	17
3.1.4. Cas d'us numero 4: Afegir Element.....	17
3.1.5 Cas d'us numero 5: Esborrar Element.....	18
3.1.6. Cas d'us numero 6: Crear monedes.....	18
3.1.7. Cas d'us numero 7: Comprar element.....	18
3.1.8. Cas d'us numero 8: Actualitza arrel.....	19
3.2. Model de domini.....	20
3.3. Classes i Pàgines HTML.....	21
3.2. Diagrames classes.....	21
4. instal·lació.....	23
5. Manual de usuari i exemple de referència.....	25
5.1.Afegir un element a la botiga.....	25
5.2.Esborrar un element de la botiga.....	28
5.3.Crear monedes.....	30
5.4.Donar-se de alta com a nou client.....	31
5.5.Comprar un element de la botiga virtual.....	31
5.6.Actualitzar la arrel.....	34
5.7.Modificar dades de un client.....	35
5.8.Baixa de un client.....	36
6. Proves.....	38
6.1. Proves del cas d'us numero 1: Afegir Client.....	38
6.2. Proves del cas d'us numero 2: Modificar Client.....	38
6.3. Proves del cas d'us numero 3: Esborrar Client.....	39

6.4. Proves del cas d'us numero 5: Afegir Element.....	39
6.5. Proves del cas d'us numero 6: Esborrar Element.....	39
6.6. Proves del cas d'us numero 7: Crear Monedes.....	40
6.7. Proves del cas d'us numero 8: Comprar Element.....	40
6.8. Proves del cas d'us numero 9: Actualitzar Arrel.....	40
7. Conclusió.....	41
8. Bibliografia.....	42
9. Annex – Codi comentat.....	42

Index de figures

il·lustració 1: Planificació	14
Ilustración 2: Esquema general aplicació	15
il·lustració 3: Casos d'us (Actor Client)	19
il·lustració 4: Casos d'us (Actor Administrador)	20
il·lustració 5: Model de domini	20
il·lustració 6: Diagrama de classes: Classes que hereten de HttpServlet	21
il·lustració 7: Diagrama de classes: Classes que hereten de GestorDisc	22
il·lustració 8: Diagrama de Classes: Classe que hereta de JFrame, Classe que hereta de Thread	22
il·lustració 9: Diagrama de Classes: Classes que hereten de Jsp	23
il·lustració 10: Pagina principal sense cap element creat	26
il·lustració 11: Login del administrador	26
il·lustració 12: Afegir nou element	27
il·lustració 13: Confirmació nou element creat	27
Ilustración 14: Pagina principal amb un element creat	28
il·lustració 15: Esborrar un element	29
il·lustració 16: Confirmació element esborrat	29
il·lustració 17: Moneder sense cap moneda creada	30
il·lustració 18: Recordatori de renovació de la arrel	30
il·lustració 19: Moneder amb 1000 monedes creades	30
il·lustració 20: Alta nou client	31
il·lustració 21: Confirmació alta nou client	31
il·lustració 22: Pagina principal amb un element creat	32
il·lustració 23: Login client al comprar un element	33
il·lustració 24: Diàleg de descarrega de un fitxers	33
il·lustració 25: Moneder després de realitzar el pagament	34
il·lustració 26: Login client per actualitzar la arrel	34
il·lustració 27: Petició de confirmació de la renovació de la arrel	35
il·lustració 28: Confirmació de actualització de la arrel	35
il·lustració 29: Login client per modificar les seves dades	35
il·lustració 30: Modificació dades client	36
il·lustració 31: Confirmació dades personals modificades	36
il·lustració 32: Login client per realitzar la seva baixa	37
il·lustració 33: Petició de confirmació de la baixa del client	37
il·lustració 34: Confirmació baixa del client	37

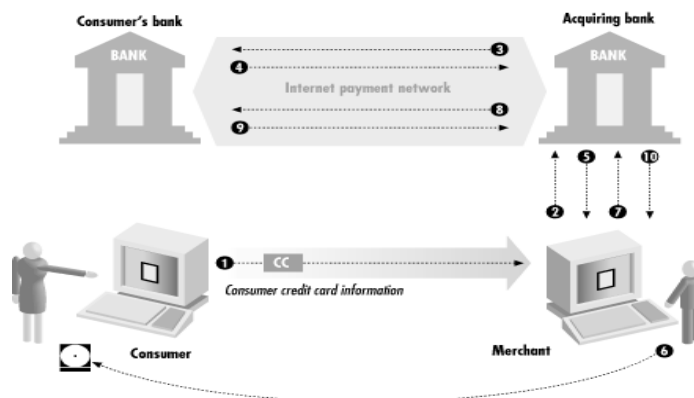
2. Cos de la memòria

2.1.Estat de l'art

2.1.1.Introducció als pagaments electrònics

Els sistemes de pagament emprats per fer les compres a través de internet, es podrien classificar en quatre categories:

1. Caixers electrònics: Es tracta de un sistema en el qual el client obre un compte en una entitat de internet. Aquesta entitat proporcionarà un codi al client que tindrà que utilitzar quan vulgui fer una compra.
2. Diners electrònics: Es tracta de diners generats de forma electrònica y que tenen una correspondència amb la moneda real, dintre dels diners electrònics podem fer dues classificacions:
 - 2.1.Diners online: Es necessari interactuar amb el banc per poder realitzar un pagament a una tercera entitat.
 - 2.2.Diners offline: Els diners estan en el ordinador del client i es poden utilitzar quan el client els necessiti, sense necessitat de contactar amb el banc. Aquests sistemes també permeten dipositar els diners en un compte i després utilitzar-los per comprar qualsevol cosa per internet.
3. Xecs electrònics: El client envia un e-mail al venedor autoritzant-lo a treu diners del seu compte, òbviament el e-mail enviat estarà encriptat i firmat digitalment.
4. Targetes de crèdit: El usuari introdueix el numero de la seva targeta de crèdit que viatjarà a través de la xarxa fins al comprador, el comprador farà les operacions necessàries amb alguna entitat bancària per tal de realitzar el cobrament, un possible esquema del funcionament de una pagament amb targetes de crèdit podria ser aquest:



De tots aquests sistemes segurament el mes emprat actualment es el de les targetes de crèdit.

2.1.2.Introducció als micropagaments i avantatges sobre altres tipus de pagaments

Que son els micropagaments? Els micropagaments son un sistema de pagament electrònic, indicat per realitzar pagaments de productes de un preu molt baix (com podria ser elements multimèdia, que es el cas que ens ocupa en aquest TFC, o cobrament per accés a continguts específics dins de una pagina web).

Tot i que el pagament per targeta es el mètode mes utilitzat actualment, presenta els següents inconvenients:

- Complexitat, alt nombre de transaccions per una sola operació (del client al venedor, del venedor al banc del venedor i del banc del venedor al banc del comprador i un cop confirmat el pagament tornarà a fer tot el circuit al revés).
- Comissions, el banc cobrarà comissions al venedor, que a vegades podrien ser superiors al preu del producte que ven (recordem que estem parlant de productes de un preu molt baix).
- Saturació del banc, en cas de que el banc del venedor estigui atenent moltes peticions de cobrament, podria saturar-se i endarrerir la operació o fins i tot no poder dur-la a terme.
- Seguretat, tot i que el numero de targeta de crèdit viatja a traves de un canal segur (SSL per exemple) aquest numero estarà en poder del venedor, que en podria fer un us indegut.

Davant d'aquests desavantatges dels pagaments per targeta de crèdit, els sistemes de micropagament tindrien les següents solucions (aquestes solucions estan basades en el sistema de micropagaments PayWord):

- Es redueix la complexitat, ja que per cada pagament només hi ha comunicació entre el client i el venedor, evitant així les comunicacions amb els bancs.
- No hi ha comissions o son mes reduïdes.
- S'evita la saturació amb el banc, ja que per verificar un pagament el venedor no te la necessitat de contactar amb el banc, ho pot fer ell mateix, d'aquesta manera tota la operació es fa molt mes ràpid.
- Es millora la seguretat, el venedor no disposa de cap numero de targeta del client, només rep una sèrie de monedes virtuals que no pot falsificar de cap manera.

Podem trobar gran quantitat de sistemes de micropagament:

- [Cartio Micropayments](#)
- [Clickshare](#)
- [CyberCash](#): CyberCoin
- [DigiCash](#)
- [E-Money](#)
- [Enition](#)
- [GC-Tech](#)
- [Internet Dollar](#)
- [Jalda](#)
- [Micropayments Transfer Protocol MPTP](#)
- [Millicent](#) by Compaq
- [NetBill](#) Carnegie Mellon University:
- [NetCheque](#) University of Southern California
- [NTSys](#)
- [NewGenPay MicroPayments](#)
- [OpenMarket](#)
- [Pay2See](#)
- [PayWord and MicroMint](#) by Rivest and Shamir
- [SOX](#) from Systemics
- [Trivnet](#)
- [The Ultimus Solution](#)
- [Wave Systems Corp](#)

De totes formes tots aquests sistemes de pagament no son massa utilitzats, en la compra de productes online es segueix utilitzant molt mes la targeta. Segurament la tecnologia dels micropagaments te mes futur per a cobrament de continguts en pagines web, ja que per aquest tipus de pagament serà impossible utilitzar targeta.

Ens centrarem en la explicació del funcionament del sistema de micropagaments PayWord, que es el que he utilitzat per la elaboració del TFC.

2.1.3.Funcionament del sistema PayWord

Una explicació esquemàtica del funcionament del sistema PayWord podria ser aquesta:

1. El client es registra amb una tercera entitat (anomenada Broker).
2. El client compra el broker un nombre determinat de monedes PayWord.
3. El broker autoritza al client per generar aquest nombre de monedes.
4. El moneder del client genera les monedes comprades.

5. El client es registra en una botiga virtual, li subministra a mes de les seves dades personals, la seva moneda arrel (posteriorment explicarem que es la moneda arrel).
6. El client realitza una compra i lliura al venedor el nombre de monedes necessàries.
7. El venedor verifica aquestes monedes a partir de la moneda arrel del client.
8. El venedor lliura l'element al client.
9. Posteriorment el venedor es posarà en contacte amb el broker per canviar les monedes obtingudes per diners reals.

Les monedes en qüestió no son mes que cadenes de hash unidireccionals (SHA1 o MD5), el que fa el moneder del client per genera aquestes monedes es:

1. Crear una primera moneda amb una funció hash undireccional a partir de un numero aleatori.
2. La segona moneda es crearà amb una funció hash unidireccional a partir de la primera moneda, la tercera igual però utilitzant la segona moneda i així successivament.
3. La ultima moneda generada es la arrel, que serà la que es proporciona al venedor en el moment de registrar-se en la botiga virtual.
4. Quan el client fa una compra, entrega las n ultimes monedes generades.

Un cop el venedor rep les monedes per realitzar la verificació fa el següent:

1. De les monedes rebudes agafa la ultima moneda, i a partir d'aquesta moneda genera un hash unidireccional. Si el hash generat coincideix amb la moneda arrel que te en la base de dades la ultima moneda rebuda serà correcte, per tant actualitzarà la arrel de la base de dades, posant en la base de dades la ultima moneda rebuda.
2. Amb la penúltima moneda rebuda farà la mateixa operació i així successivament amb totes les monedes rebudes.
3. Si totes les monedes son correctes el pagament es dona per bo.

El fet del utilitzar hash unidireccionals, permet que tot i que el venedor te la moneda arrel, no pugui generar monedes valides a partir d'aquesta moneda.

2.2. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC

El context del projecte es el de una botiga online de venda de productes multimèdia, el TFC es realitza per dotar a aquesta tenda de un sistema de pagaments PayWord. Aquests tipus de sistema de pagament, esta dissenyat per a ser utilitzat en pagaments de petites quantitats, per tant s'adiu totalment a les característiques de la tenda, ja que els productes multimèdia tenen un preu de venda baix. El sistema de pagaments PayWord proporciona un rendiment superior a altres sistemes basats en certificats (per exemple el pagament amb targeta a través de un canal segur).

A mes el fet de que no viatgen en cap moment dades de pagament a través de la xarxa, fa que aquest sistema sigui molt segur.

2.3. Objectius del TFC

El objectiu d'aquest TFC es crear un botiga virtual de venda de elements multimèdia, les compres en aquesta botiga es faran a través de un sistema de micropagaments, concretament el sistema de micropagaments PayWord.

Per a complir amb aquest objectiu haurem de crear el següents components:

- Una aplicació client anomenada moneder independent del servidor. Aquesta aplicació s'encarregarà de generar les monedes PayWord necessàries i fer els tractes amb el servidor de pagament en el moment de donar-se d'alta en la botiga virtual o quan realitzem una compra.
- Un servidor web que allotgi la botiga virtual, aquest servidor s'encarregarà de mostrar els elements disponibles per la venda i atindrà les peticions de compra dels clients.
- El mateix servidor web disposarà de opcions per donar d'alta un client en la botiga virtual, donar de baixa un client o modificar-ne les seves dades. També disposarà de una funcionalitat que permetrà donar d'alta els elements multimèdia dins del servidor, així com també permetre donar-los de baixa (aquesta opció només serà accessible per l'administrador de la web).
- Un servidor de pagaments que s'encarregarà d'obtenir les monedes necessàries del moneder del client, en verificarà la seva validesa i lliurarà l'element sol·licitat al client a través de una resposta http.

2.4. Enfocament i mètode seguit

Per la creació he emprat un enfocament orientat a objectes intentant seguir el patró MVC, per tal de seguir aquest patró en la part del servidor he emprat el següent:

- Classes en Java per al model, s'encarreguen de representar els objectes del model de domini.

- Servlets per al controlador, reben les peticions de la vista i fan les operacions oportunes amb el model de domini, un cop tractades les dades retornen una resposta a la vista corresponent.
- JSP i HTML per la vista, son la interfície gràfica del usuari.

Per a la persistència de les dades tant en la part client com en la part servidor, he emprat fitxers en lloc de una BBDD, per tal de simplificar la aplicació. Per tal de fer aquesta aplicació el mes independent possible del tipus de persistència, he intentat emprar el patró DAO, creant unes classes que s'encarreguen de gestionar la persistència de les diferents dades de la aplicació. D'aquesta manera, si posteriorment es volgués canviar a una persistència de dades basada en una BBDD, només caldria fer canvis en la classe que s'encarrega de gestionar la persistència i no tindríem que modificar tot el projecte.

Anem ara a definir les eines de programació que s'han emprat en el desenvolupament del TFC:

- Java 2 Standard Edition 1.4.2.
- Jakarta Tomcat 5 (Contenidor de Servlets i JSP, també servirà pàgines estàtiques HTML).
- Linux Mandrake 10.0 CE.
- Eclipse IDE 3.0 M8 + Plugin Lombok.
- NetBeans 3.6 per crear la interfície gràfica del moneder.
- Jakarta Commons FileUpload (emprat per pujar els elements multimèdia al servidor).
- OpenOffice (per generar tota la documentació, com aquesta memòria).
- Umbrello (per generar diagrames UML, emprat en la etapa de anàlisi i disseny).
- Quant plus, per dissenyar la pàgina web.
- JCE (Llibreria criptogràfica integrada dins del J2SE 1.4.2).

En un principi tenia pensat de utilitzar la combinació Tomcat + Apache, ho vaig descartar per simplificar el desenvolupament de la aplicació, si s'optes per posar l'aplicació en marxa es tindria que estudiar aquesta opció, ja que Apache es mes eficient servint pàgines HTML. També he descartat la opció de posar un mòdul SSL en el Tomcat, per no complicar-ne mes la administració, tot i que també seria una opció a tenir en compte en cas de una posada en producció d'aquesta aplicació.

Per tal de crear la aplicació client he emprat la llibreria gràfica integrada dins del J2SE Swing, una altre opció era utilitzar la llibreria SWT que en principi te millor rendiment que Swing, però donada la senzillesa de la aplicació client en quant a interfície gràfica es refereix i la pèrdua de portabilitat que es dona si fem SWT (ja que es dependent del sistema operatiu) em vaig decidir per emprar Swing.

La comunicació entre client (moneder) i servidor es farà a traves de sockets, de manera que

el servidor farà al client una petició de un nombre determinat de monedes a traves de una connexió amb sockets i el client li respondrà enviant-li les monedes que ha demanat.

L'enviament de l'element comprat es farà a traves del protocol http, tal com ens indica l'enunciat, aprofitant les característiques pròpies dels Servlets. La botiga virtual esta dissenyada de forma que només es pot comprar un element a la vegada, una opció a millorar seria la possibilitat de poder comprar mes de un element alhora (el servidor s'encarrega de comprimir tots els elements dins de un únic fitxer i després lliura aquest fitxers al client a traves de una resposta http), en principi he descartat aquesta opció, per tant tindrem aquesta restricció de només poder fer una compra alhora.

En quant als servidors, tot i que en els objectius de la assignatura ens trobem amb dos servidors diferents (Servidor web i Servidor de pagament), en la aplicació implementada finalment només he utilitzat un únic servidor que farà les dos funcions, tot i que amb algunes modificacions es podria posar perfectament en servidors independents.

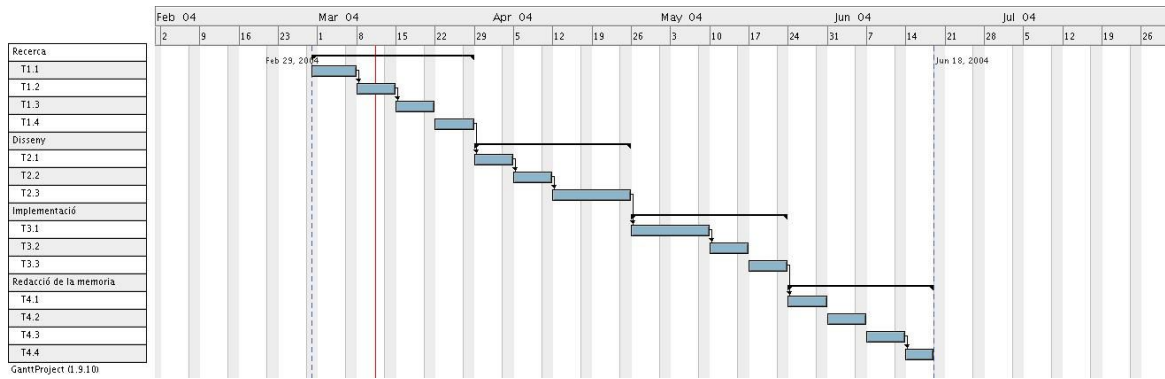
2.5. Planificació del projecte

La planificació del projecte segons el pla de treball es la següent:

<i>Setmana</i>	<i>Dates</i>	<i>Activitat</i>	<i>Esdeveniment</i>
1	29/2/2004 – 7/3/2004	(T1.1) Buscar informació general sobre el TFC.	Trobada presencial
2	8/3/2004 – 14/3/2004	(T1.2) Redacció del pla de treball	Lliurament del pla de treball – 11/3/2004
3	15/3/2004 – 21/3/2004	(T1.3) Buscar informació sobre: <ul style="list-style-type: none">• Funcionament de micropagaments PayWord.• Transmissió d'informació sobre HTTP.	
4	22/3/2004 – 28/3/2004	(T1.4) Buscar informació sobre: <ul style="list-style-type: none">• Transmissió d'informació sobre sockets.• Funcions de hash unidireccionals com SHA o MD5.	

<i>Setmana</i>	<i>Dates</i>	<i>Activitat</i>	<i>Esdeveniment</i>
5	29/3/2004 – 4/4/2004	(T2.1) Disseny del programa a nivell de processos, comunicació i interacció entre els diferents components del sistema.	
6	5/4/2004 – 11/4/2004	(T2.2) Disseny de la cartera i decisió de les seves funcionalitats bàsiques.	
7 i 8	12/4/2004 – 25/4/2004	(T2.3) Implementació del disseny realitzat. En cas d'error o de trobar noves funcionalitats repetirem les dues tasques anteriors.	Lliurament de la PAC2 – 13/4/2004
9 i 10	26/4/2004 – 9/5/2004	(T3.1) Implementació de la cartera i del servidor HTTP de pagament.	
11	10/5/2004 – 16/5/2004	(T3.2) Implementació del mòdul payword.	
12	17/5/2004 – 23/5/2004	(T3.3) Disseny de la web de la botiga.	Lliurament de la PAC3 – 17/5/2004
13	24/5/2004 – 30/5/2004	(T4.1) Elaboració de la memòria: <ul style="list-style-type: none"> • Objectiu del projecte. • Estat de l'art . • Eines de programació emprades. 	
14	31/5/2004 – 6/6/2004	(T4.2) Elaboració de la memòria: <ul style="list-style-type: none"> • Disseny del programa. 	
15	7/6/2004 – 13/6/2004	(T4.3) Elaboració de la memòria: <ul style="list-style-type: none"> • Proves, incidències i conclusions. • Apendix amb codi font, degudament comentat. 	Lliurament de la pre-memòria – 8/6/2004
16	14/6/2004 – 18/6/2004	(T4.4) Presentació del projecte.	Lliurament final TFC – 18/6/2004

Podem fer una representació d'aquesta planificació amb un diagrama de Gantt:



il·lustració 1: Planificació

2.6. Productes obtinguts

Després de la realització del TFC he obtingut dos productes ben diferenciats:

- Servidor: Conte la part de la aplicació que serveix les pàgines de la botiga virtual (servidor web) i la part que s'encarrega de gestionar les peticions de cobrament rebudes del servidor web, obtenir el pagament del moneder del client i verificar-lo (Servidor de pagament).
- Client: Conte únicament la aplicació moneder que s'encarrega de generar les monedes i fer les gestions amb el servidor de pagament quan es fa una compra.

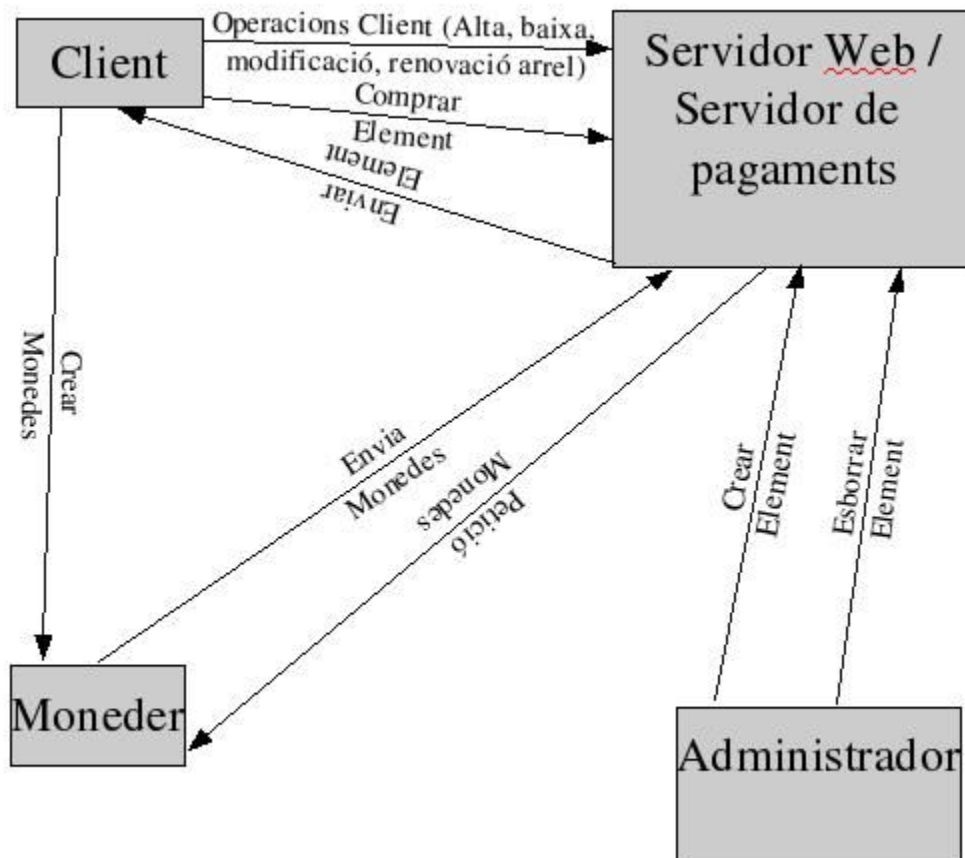
2.7. Breu descripció de la resta de capítols de la memòria

A continuació indicaré breument la resta de punts de que es compona la memòria:

- Disseny: Resum de les parts més importants del disseny de la aplicació.
- instal·lació: Senzilla explicació de com posar en funcionament la aplicació desenvolupada.
- Proves: Detall dels jocs de proves que s'han utilitzat per comprovar el funcionament de la aplicació.
- Conclusió: Breu opinió dels avantatges e inconvenient de un sistema PayWord, així com les coses que s'haurien de millorar en cas de utilitzar aquest sistema en una aplicació real.
- Bibliografia: Resum de llibres i pàgines web que he utilitzat per realitzar el TFC.
- Annex: Codi comentat al estil javadoc, amb comentaris addicionals en les parts més importants del codi.

3. Disseny

Primer de tot farem un petit esquema del funcionament general que es vol de la aplicació:



il·lustració 2: Esquema general aplicació

La idea es la de proporcionar a l'administrador una sèrie de funcionalitats que li permetin afegir i esborrar elements de la botiga. Així mateix el client a part de les funcionalitats obligatòries per poder realitzar una compra (alta i actualització de arrel) tindrà operacions de modificació de les seves dades i de baixa de la pagina web.

3.1. Casos d'us

3.1.1. Cas d'us numero 1: Afegir Client

Resum de la funcionalitat: Afegeix un client a la base de dades.

Actors: Client.

Casos d'us relacionats:

Precondició: El client no existeix a la base de dades.

Postcondició: El client existeix a la base de dades.

El client introdueix les seves dades personals (nom, cognoms, adreça, telèfon i e-mail), també introdueix un nom d'usuari i una contrasenya.

Alternatives de procés i excepcions:

- 1) Si el client ja existeix en la base de dades s'atura el procés.
- 2) Si el nom d'usuari ja existeix en la base de dades s'atura el procés.

3.1.2. Cas d'us numero 2: Consultar o Modificar Client

Resum de la funcionalitat: Consulta o modifica les dades de un client a la base de dades.

Actors: Client.

Casos d'us relacionats:

Precondició:

Postcondició:

El client introdueix el seu nom d'usuari i la seva contrasenya, el sistema li mostra les seves dades personals i el client pot procedir a la seva modificació.

Alternatives de procés i excepcions:

- 1) Si el client no existeix en la base de dades, s'haurà de donar de alta segons el cas d'us numero 1.

3.1.3. Cas d'us numero 3: Esborrar Client

Resum de la funcionalitat: Esborra un client de la base de dades.

Actors: Client.

Casos d'us relacionats:

Precondició: El client existeix en la base de dades.

Postcondició: El client no existeix en la base de dades.

El client introdueix el seu nom d'usuari i contrasenya i confirma la eliminació.

Alternatives de procés i excepcions:

- 2) Si el client no existeix en la base de dades, s'haurà de donar de alta segons el cas d'us numero 1.

3.1.4. Cas d'us numero 4: Afegir Element

Resum de la funcionalitat: Afegeix un element a la base de dades.

Actors: Administrador.

Casos d'us relacionats:

Precondició: L'element no existeix en la base de dades.

Postcondició: L'element existeix en la base de dades.

L'administrador introdueix el seu nom d'usuari i contrasenya, a continuació introdueix el nom del fitxer que es vol donar de alta, el tipus d'element i la seva descripció.

3.1.5 Cas d'us numero 5: Esborrar Element

Resum de la funcionalitat: Esborra un element de la base de dades.

Actors: Administrador.

Casos d'us relacionats:

Precondició: L'element existeix en la base de dades.

Postcondició: L'element no existeix en la base de dades.

L'administrador introdueix el seu nom d'usuari i contrasenya, a continuació indica l'element que vol donar de baixa i en confirma la seva eliminació.

3.1.6. Cas d'us numero 6: Crear monedes

Resum de la funcionalitat: Crea un nombre de monedes determinat.

Actors: Client.

Casos d'us relacionats: Actualitza arrel (8)

Precondició: El moneder te N monedes on $N \geq 0$.

Postcondició: El moneder te $N+M$ monedes, on M son les noves monedes creades.

El Client introdueix en el seu moneder el nombre de monedes a crear i en confirma la seva creació.

3.1.7. Cas d'us numero 7: Comprar element

Resum de la funcionalitat: Compra un element de la botiga online.

Actors: Client.

Casos d'us relacionats:

Precondició: L'element existeix en la botiga online.

Postcondició: El client ha obtingut l'element comprat.

El Client tria els elements que vol comprar i en confirma la compra, introduint el seu nom d'usuari i la seva contrasenya.

3.1.8. Cas d'us numero 8: Actualitza arrel

Resum de la funcionalitat: Actualitza la arrel en la base de dades.

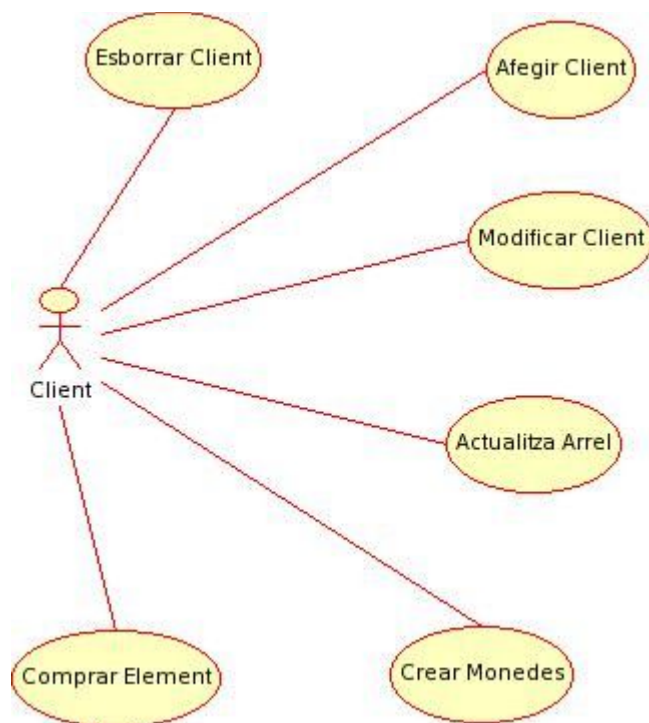
Actors: Client.

Casos d'us relacionats: Crear Monedes (6).

Precondició: El Client ha creat noves monedes.

Postcondició: L'arrel del Client s'ha actualitzat en la base de dades.

El Client introdueix el seu nom d'usuari i contrasenya i confirma la actualització de la seva arrel en la base de dades.



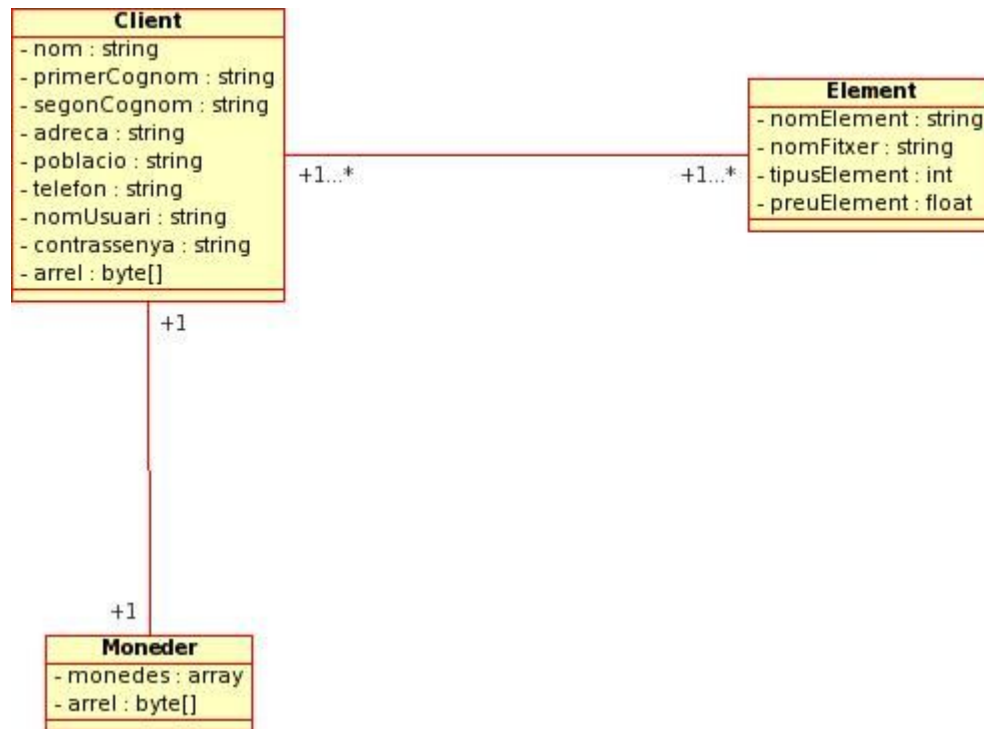
il·lustració 3: Casos d'us (Actor Client)



il·lustració 4: Casos d'us (Actor Administrador)

3.2. Model de domini

Segons les dades que ens proporciona l'enunciat podem obtenir el següent model de domini:

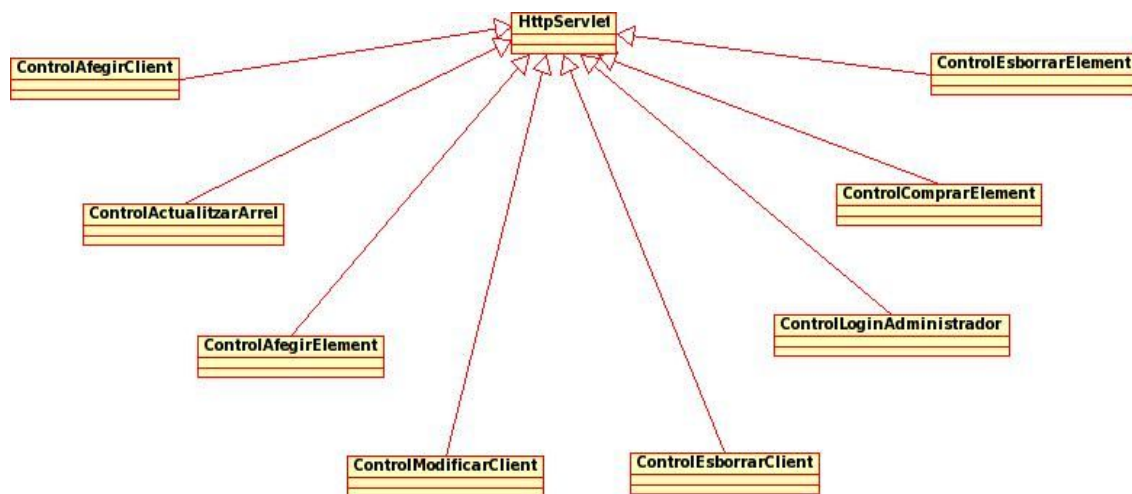


il·lustració 5: Model de domini

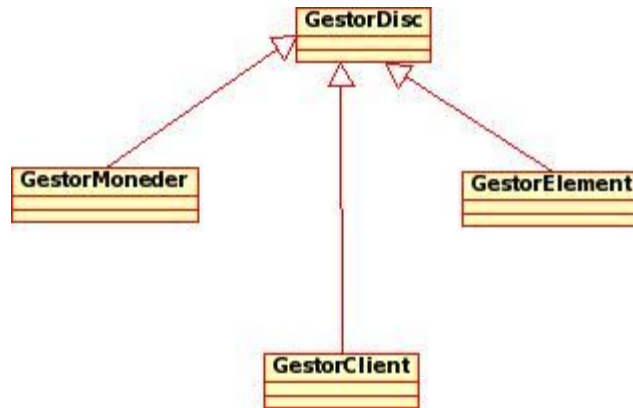
3.3. Classes i Pàgines HTML

Client	ControlAfegirClient	ControlAfegirElement
ControlComprarElement	ControlEsborrarElement	ControlEsborrarClient
ControlLoginAdministrador	ControlModificarClient	Element
GestorClient	GestorDisc	GestorElement
BarraNavegacio	Capçalera	VistaAfegirClient
VistaAfegirElement	VistaComprarElement	VistaEsborrarElement
VistaLoginAdministrador	VistaLoginClientBaixa	VistaLoginClientModificacio
VistaModificarClient	VistaLoginClientActualitzarArrel	ControlActualitzarArrel
ConnexioMoneder	ControlMoneder	GestorDisc
GestorMoneder	Moneder	VistaMoneder
ExcepcioClient	ExcepcioAdministrador	ExcepcioElement
PaginaError		

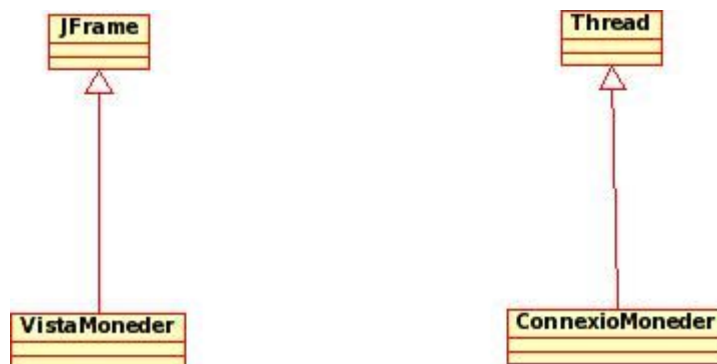
3.2. Diagrames classes



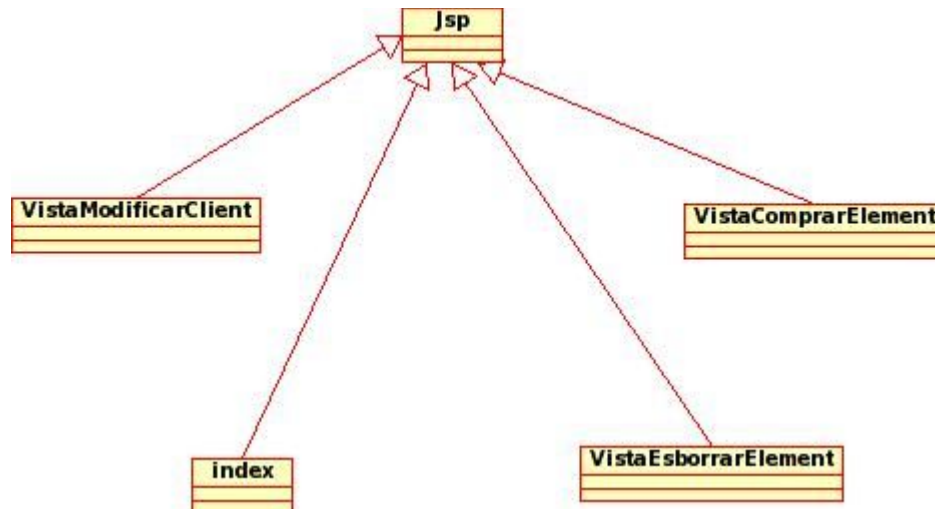
il·lustració 6: Diagrama de classes: Classes que hereten de HttpServlet



il·lustració 7: Diagrama de classes: Classes que hereten de GestorDisc



il·lustració 8: Diagrama de Classes: Classe que hereta de JFrame, Classe que hereta de Thread



il·lustració 9: Diagrama de Classes: Classes que hereten de Jsp

4. instal·lació

Per tal de poder posar en funcionament la aplicació necessitem tenir instal·lats els següents elements:

- Java 2 Standard Edition 1.4.2
- Jakarta Tomcat 5 (No ho he provat amb versions anteriors de Tomcat, suposo que funcionarà igualment però no ho puc assegurar).

La aplicació lliurada consta de tres elements:

- Fitxer client.jar on es troba la aplicació moneder.
- Fitxer botiga.war on es troba tota la part del servidor.
- Fitxer commons-fileupload-1.0.jar (Fitxer que correspon al projecte Jakarta Commons, emprat per pujar els elements multimèdia al servidor).

El procés per a realitzar la instal·lació de la aplicació serà l següent:

- 1) Posar en marxa el servidor Tomcat.
- 2) Copiar el fitxer botiga.war en el directori {Tomcat_home}/webapps
- 3) Copiar el fitxer commons-fileupload-1.0.jar dins del directori {Tomcat_home}/commons/lib
- 4) Copiar el fitxer client.jar en un directori a triar per l'usuari.

Un cop fet això podrem observar que el mateix servidor Tomcat ens ha creat un directori dintre de webapps, anomenat botiga. Si això es correcte haurem de editar el següent fitxer:

{Tomcat_home}/webapps/botiga/WEB-INF/web.xml

Aquest fitxer consta de una part que s'ha de modificar per el correcte funcionament de la aplicació, concretament es aquesta:

```
<context-param>
  <param-name>adminUser</param-name>
  <param-value>admin</param-value>
</context-param>
<context-param>
  <param-name>adminPassword</param-name>
  <param-value>admin</param-value>
</context-param>
<context-param>
  <param-name>fitxers</param-name>
  <param-value>/home/jordi/fitxers/</param-value>
</context-param>
<context-param>
  <param-name>elements</param-name>
  <param-value>/home/jordi/elements/</param-value>
</context-param>
<context-param>
  <param-name>clients</param-name>
  <param-value>/home/jordi/clients/</param-value>
</context-param>
```

Aquesta part del fitxer web.xml conte el següent:

- Nom de usuari de l'administrador.
- contrasenya de l'administrador.
- Ubicació on es guardaran els clients registrats.
- Ubicació on es guardaran els elements registrats.
- Ubicació on es guardaran els fitxers corresponents a cada element.

Per exemple si volguéssim canviar el directori on es guarden els clients per /jordi/clients ho deixariem d'aquesta manera:

```
<context-param>
  <param-name>clients</param-name>
  <param-value>/jordi/clients/</param-value>
</context-param>
```

En cas de que s'utilitzi windows la ruta haurà de ser similar a aquesta (C:\jordi\clients\ es

a dir amb dobles barres. És important que tots els directoris estiguin ja creats, ja que sinó la aplicació no funcionarà.

S'ha de tenir en compte que s'ha de posar la ruta del directori completa i que el directori ha de estar creat. Si no existeix el directori no funcionarà.

Un cop fet això ja podem provar la aplicació:

- Per posar en marxa el moneder, entrarem en el directori on hem copiat el fitxer client.jar i posarem en la línia de comandes java -jar client.jar. En cas de que l'executem des de windows serà suficient amb fer doble click sobre el fitxer client.jar.
- Per accedir a la botiga obrirem qualsevol navegador web (jo he utilitzat Mozilla) i posarem la adreça <http://localhost:8080/botiga>.

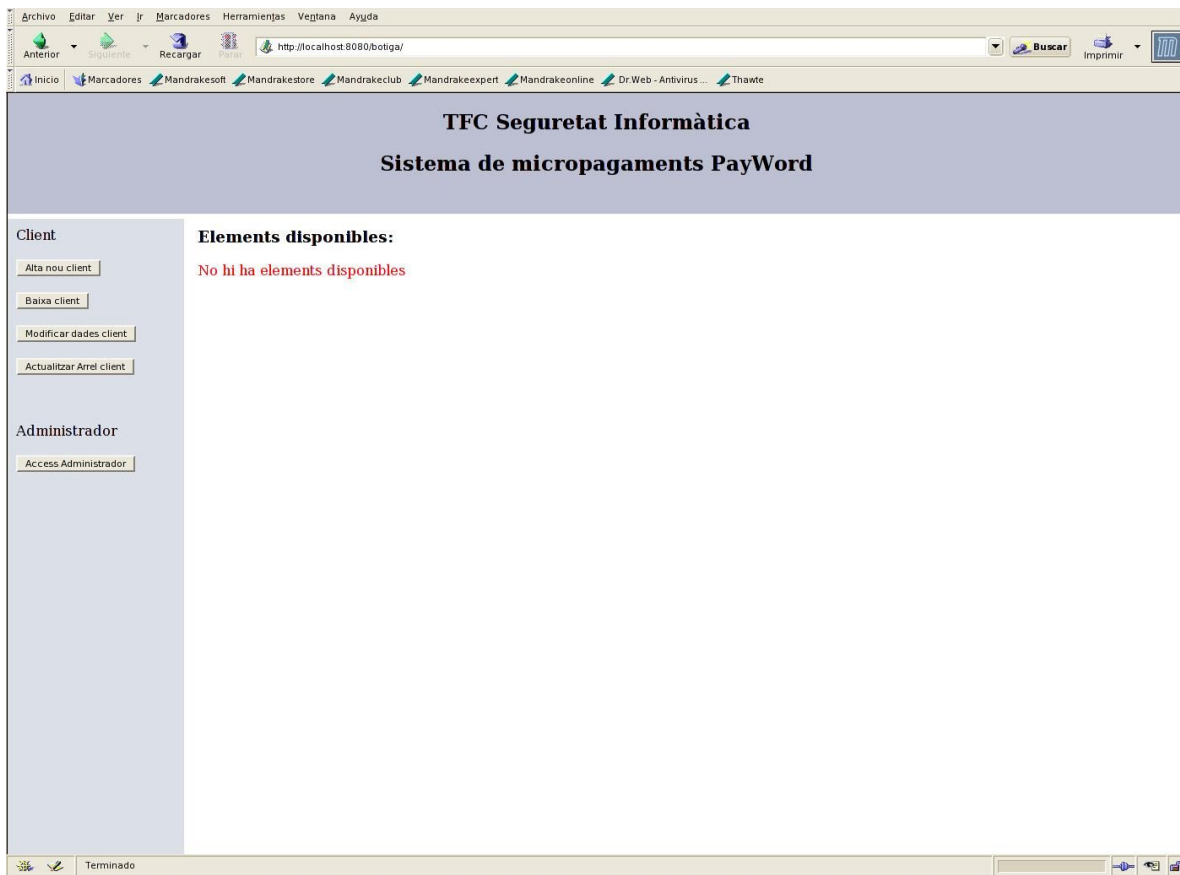
5. Manual de usuari i exemple de referència

A continuació posaré un exemple de funcionament de la aplicació que alhora servirà com a manual de referència per al usuari, per tal de poder dur a terme l'exemple que indiquem serà necessari el haver instal·lat amb antelació la aplicació, tot seguint els passos indicats en l'apartat corresponent.

El funcionament de la aplicació està dividit en dos parts, les funcionalitats que du a terme l'actor Client i les funcionalitats que du a terme l'actor Administrador. Començarem per les funcionalitats pròpies de l'administrador.

5.1. Afegir un element a la botiga

Aquesta funcionalitat es utilitza per l'administrador per donar d'alta un element multimèdia en el servidor de la botiga online. El primer pas doncs serà entrar en la pàgina de la botiga online, obrirem el nostre navegador web i posarem <http://localhost:8080/botiga> en la nostra barra d'adreces (localhost serà el host on està el servidor Tomcat, en aquest cas s'executa en local), obtindrem la següent resposta:



il·lustració 10: Pagina principal sense cap element creat

Com podem veure a la esquerra de la pantalla tenim un menú de navegació on estan totes les operacions que poden realitzar, tant els clients com l'administrador de la botiga. En el cos de la aplicació tenim els elements disponibles per vendre, en aquests moments no en tenim cap així que procedirem a afegir el primer element. Al prémer aquest boto l'aplicació ens demanarà el nom de usuari, contrassenya i la operació que volem realitzar, tal com podem veure en la següent captura:

Introdueix el teu nom d'usuari, contrassenya i la operació que vols realitzar:

Nom Usuari:

Contrassenya:

Tipus Element:

il·lustració 11: Login del administrador

Introduïrem el nom de usuari i la contrasenya del administrador (definita en web.xml), triarem la opció d'afegir element i premerem el botó de enviar consulta, obtindrem una pantalla com la de la següent captura:

Introdueix les dades del element que vols crear:



The form consists of four input fields and two buttons. The first field is labeled 'Nom Element:' and is a text input. The second field is labeled 'Nom Fitxer:' and is a text input with a button labeled 'Examinar...' to its right. The third field is labeled 'Tipus Element:' and is a dropdown menu with 'Musica' selected. The fourth field is labeled 'Preu Element:' and is a text input followed by the word 'Euros'. Below these fields is a button labeled 'Enviar Dades'.

il·lustració 12: Afegir nou element

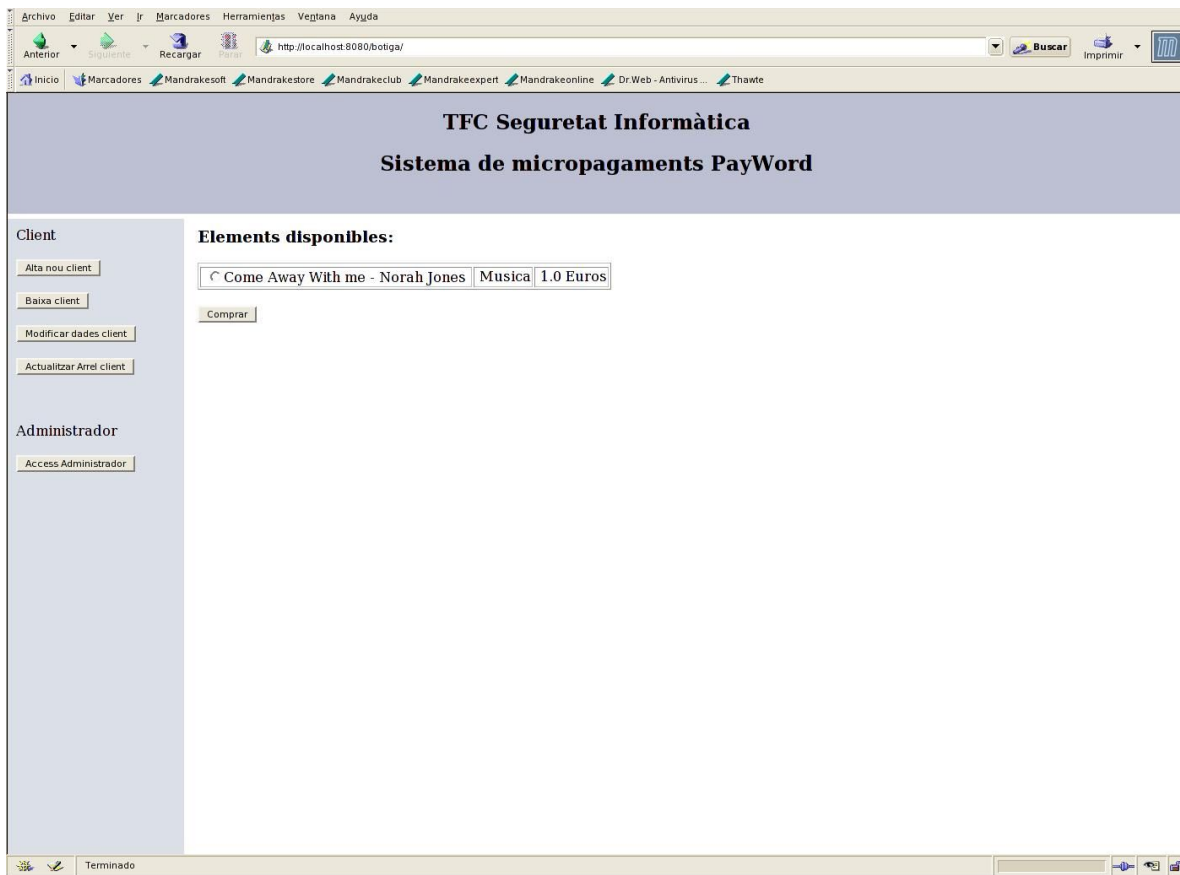
Primer de tot introduïrem el nom del element, aquest nom serà el que veuran els clients quan facin una compra, per tant tindrà que ser representatiu del element. A continuació premem el boto examinar que esta al costat del camp Nom Fitxer i se'ns obrirà un diàleg on podrem triar el fitxer al que correspon l'element que volem donar de alta, seguidament triem el tipus de element de que es tracta (musica, vídeo o imatge) i el preu en euros d'aquest element. Un cop esta tot completat premerem el boto de enviar dades, si tot ha funcionat correctament obtindrem un missatge com el de la següent captura:

El nou element s'ha creat correctament

[Tornar a la pagina inicial](#)

il·lustració 13: Confirmació nou element creat

Si volem podem comprovar que l'element s'ha afegit correctament si premem el link de tornar a la pagina inicial, obtindrem una pantalla com la de aquesta captura:



il·lustració 14: Pagina principal amb un element creat

Com podem veure el nou element s'ha afegit correctament i ja esta disponible per la seva venda. Repetirem la operació tantes vegades com elements vulguem afegir.

5.2.Esborrar un element de la botiga

Per esborrar un element el primer que farem es prémer el boto Accés Administrador del menú de navegació de la pagina. Igual que en el cas anterior introduïrem el nom de usuari i contrasenya, però aquest cop triarem la opció de esborrar element. Obtindrem una pantalla com la de la següent captura:

Selecciona l'element que vols esborrar:

<input type="radio"/> Come Away With me - Norah Jones
<input type="radio"/> The Corrs - Summer Sunshine

[Tornar a la pagina inicial](#)

il·lustració 15: Esborrar un element

Seleccionarem doncs l'element que volem esborrar i premerem el boto Esborrar Element, obtindrem doncs una pantalla com la de la següent captura:

Selecciona l'element que vols esborrar:

⌂ Come Away With me - Norah Jones

Esborrar Element

[Tornar a la pagina inicial](#)

il·lustració 16: Confirmació element esborrat

Com podem veure l'element s'ha esborrat correctament. Podrem repetir aquesta operació totes les vegades que sigui necessari.

A continuació detallarem les operacions que pot dur a terme un client que vol comprar en la nostra pagina web. Primer de tot hem de tenir en compte que per poder operar amb el sistema PayWord es necessari tenir en marxa la aplicació moneder, així doncs ens posarem en el directori on hem copiat el fitxer moneder.jar i introduïrem la següent comanda (en windows es suficient amb fer doble click sobre el fitxer jar):

```
java -jar Moneder.jar
```

La aplicació moneder es posarà amb marxa i podrem veure la seva interfície gràfica que tindrà un aspecte semblant a aquest:

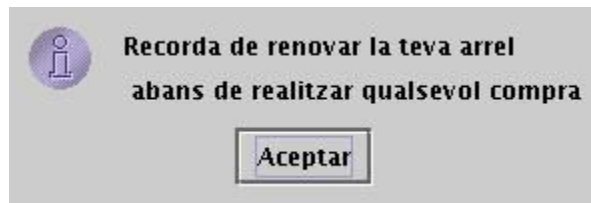


il·lustració 17: Moneder sense cap moneda creada

Com podem veure al iniciar la aplicació moneder per primera vegada no tindrem monedes disponibles, així doncs abans de realitzar cap compra el primer que haurem de fer es crear monedes per poder realitzar els pagaments.

5.3.Crear monedes

Introduïrem el nombre de monedes a crear, ho podem introduint-ho directament o amb les fletxes (compte, s'han d'introduir el nombre de monedes a crear, tenint en compte que una moneda PayWord es igual a un cèntim), suposem que volem crear 1000 monedes PayWord (10 euros), introduïm la quantitat i premem el boto de crear monedes, si tot ha anat correctament obtindrem el següent missatge:



il·lustració 18: Recordatori de renovació de la arrel

Simplement es tracta de un recordatori, ja que si no renovem la nostra arrel en la botiga virtual no podrem fer les nostres compres, premem el boto Acceptar i tornarem a la finestra principal de la aplicació moneder, on podrem veure com les monedes s'han creat correctament:



il·lustració 19: Moneder amb 1000 monedes creades

Doncs ara que ja tenim monedes disponibles ja podem anar a la botiga virtual i realitzar les nostres primeres compres. Hem de tenir en compte que no podem tancar la aplicació moneder si volem fer compres.

El primer que haurem de fer per poder realitzar una compra es donar-nos de alta com a clients.

5.4.Donar-se de alta com a nou client

Entrarem en la botiga virtual (<http://localhost:8080/botiga>) i triarem la opció de alta nou client, en el menú de navegació esquerra. Obtindrem una pantalla com la de la següent captura:

Introdueix les teves dades personals:

Nom: Primer Cognom:

Segon Cognom: Adreça:

Població: Telefon:

E-mail: Nom Usuari:

Contrassenya:

Nota: Per tal de poder donar-te de alta es necessari que la teva aplicació moneder estigui activa.

il·lustració 20: Alta nou client

Queda bastant clar doncs el que hem de fer, introduïrem les nostres dades personals i premerem el boto de enviar dades (totes les dades son obligatòries per poder donar-se de alta), si tot ha anat correctament obtindrem un missatge com el de la següent captura:

El usuari Jordi Farré ha estat donat de alta correctament

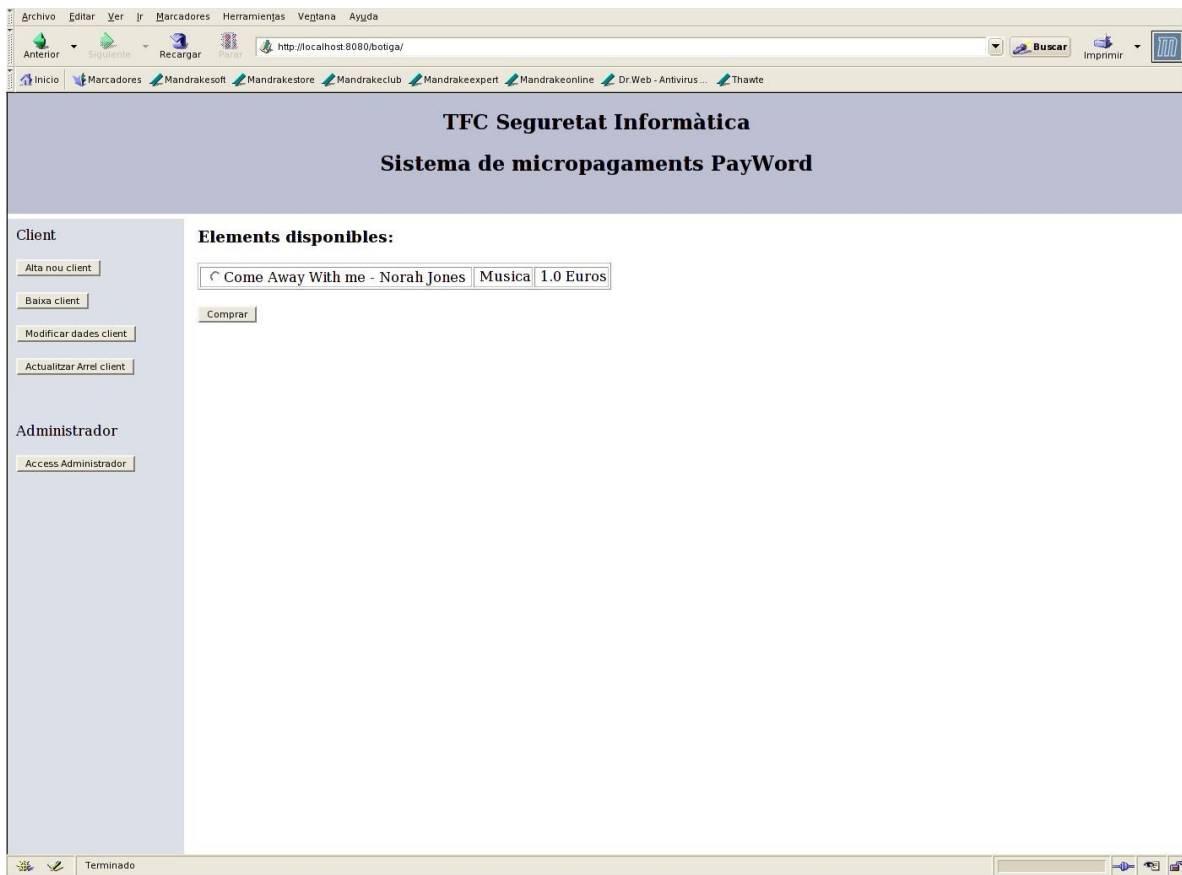
[Tornar a la pagina inicial](#)

il·lustració 21: Confirmació alta nou client

Un cop donats de alta ja podem realitzar la nostra primera compra, anem a descriure doncs el procés de compra.

5.5.Comprar un element de la botiga virtual

Anirem a la pagina principal de la botiga (<http://localhost:8080/botiga>), en la part central de la pagina podrem veure els elements disponibles per a una possible compra, recordem-ho:



il·lustració 22: Pagina principal amb un element creat

Seleccionarem doncs l'element que volem comprar i premerem el boto Comprar, obtindrem una pantalla com la de la següent captura:

Introdueix el teu nom d'usuari i contrassenya:

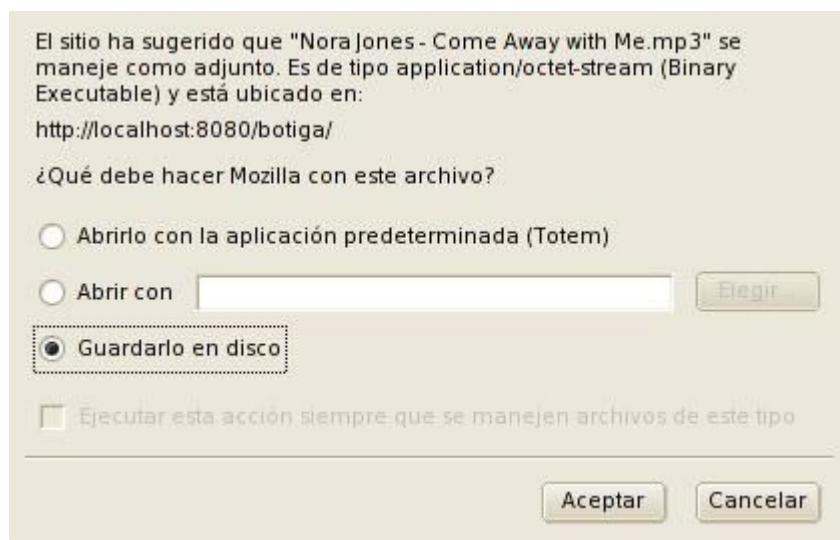
Nom Usuari:

Contrassenya:

Nota: Per tal de poder realitzar la teva compra es necessari que la teva aplicació moneder estigui activa.

il·lustració 23: Login client al comprar un element

Seguidament introduïrem el nom de usuari i la contrasenya que ens hem assignat en el moment de fer la alta i premerem el boto de Enviar Dades, si tot va correctament i el pagament es du a terme sense problemes, se'ns obrirà una finestra com aquesta:



il·lustració 24: Diàleg de descarrega de un fitxers

Triarem la opció de “Guardarlo en disco” i senyalarem on el volem guardar, al prémer el boto Guardar automàticament se'ns desarà l'element que hem comprat en la ubicació que hem triat. Si anem ara a la aplicació moneder, podrem veure que se'ns ha descomptat l'import de l'element, tal com podem veure en la següent captura:



il·lustració 25: Moneder després de realitzar el pagament

Fins aquí hem vist les operacions essencials per poder realitzar una compra en la nostra botiga virtual, però des de la mateixa botiga es poden realitzar altres operacions relacionades amb els clients. La primera que veurem serà la actualització de la arrel de un client.

5.6. Actualitzar la arrel

Sempre que creem monedes noves tindrem que actualitzar la arrel de la botiga virtual, a continuació detallem la manera de fer-ho.

Entrem a la pagina principal de la botiga virtual (<http://localhost:8080/botiga>) i triem la opció de Actualitzar arrel, entrarem en una pantalla similar a aquesta captura:

Introdueix el teu nom d'usuari i contrassenya:

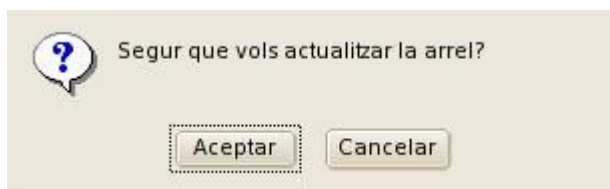
Nom Usuari:

Contrassenya:

Nota: Si no disposas de nom d'usuari i contrassenya prem [aquí](#)

il·lustració 26: Login client per actualitzar la arrel

Per tant introduïrem el nostre nom de usuari i contrasenya (assignats per nosaltres al realitzar l'alta en la botiga) i premerem el boto de enviar dades. Un cop fet això l'aplicació ens demanarà confirmació de la operació a traves de una finestra com aquesta:



il·lustració 27: Petició de confirmació de la renovació de la arrel

Si confirmem la operació i no hi ha cap problema per actualitzar la arrel, obtindrem un missatge de confirmació similar a aquest:

La arrel s'ha actualitzat correctament

[Tornar a la pagina inicial](#)

il·lustració 28: Confirmació de actualització de la arrel

Una altre operació que es pot realitzar, es la modificació de les dades de un client, a continuació detallarem el procés per fer aquestes modificacions

5.7.Modificar dades de un client

Entrarem com es habitual en la pagina principal de la botiga virtual (<http://localhost:8080/botiga>) i triarem la opció Modificar Dades client en el menú de navegació esquerra. Un cop fet això ens sortirà una pantalla similar a aquesta captura:

Introdueix el teu nom d'usuari i contrassenya:

Nom Usuari:

Contrassenya:

Nota: Si no disposas de nom d'usuari i contrassenya prem [aquí](#)

il·lustració 29: Login client per modificar les seves dades

Introduïrem el nom de usuari i la contrasenya i premerem el boto de enviar dades, un cop fet això la aplicació ens mostrarà les nostres dades, per tal de que las puguem modificar, en una pantalla similar a aquesta captura:

Modifica les teves dades personals:

Nom: Primer Cognom:

Segon Cognom: Adreça:

Població: Telefon:

E-mail: Nom Usuari:

Contrassenya:

il·lustració 30: Modificació dades client

Modificarem les dades que considerem necessari i premerem el boto de enviar dades, un cop fet això rebrem una confirmació de que la modificació s'ha dut a terme correctament, en una pantalla similar a aquesta:

El usuari Jordi Farré ha estat modificat correctament

[Tornar a la pagina inicial](#)

il·lustració 31: Confirmació dades personals modificades

Ja només ens queda una ultima operació que es pot realitzar a traves de la nostra botiga virtual, es la baixa de un client, detallem a continuació el procés a seguir.

5.8.Baixa de un client

Entrarem en la pagina principal de la botiga virtual (<http://localhost:8080/botiga>) i premerem el boto de Baixa Client, situat en el menú de navegació de la esquerra, un cop fet això obtindrem un pantalla similar a aquesta:

Introdueix el teu nom d'usuari i contrassenya:

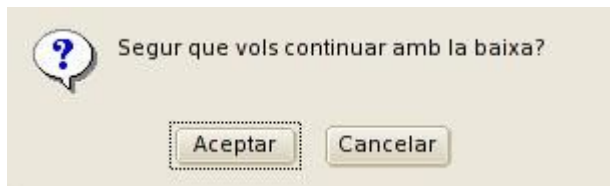
Nom Usuari:

Contrassenya:

Nota: Si no disposas de nom d'usuari i contrassenya prem [aquí](#)

il·lustració 32: Login client per realitzar la seva baixa

Introduïrem el nostre nom de usuari i contrasenya i premerem el boto de enviar dades, a continuació la aplicació ens demanarà confirmació de la operació a traves de una finestra com aquesta:



il·lustració 33: Petició de confirmació de la baixa del client

Si confirmem la baixa i no hi ha cap problema addicional, obtindrem un missatge de confirmació similar a aquest:

El usuari Jordi Farré ha estat donat de baixa correctament

[Tornar a la pagina inicial](#)

il·lustració 34: Confirmació baixa del client

6. Proves

A continuació adjunto una sèrie de taules amb els jocs de proves que he realitzat per cada cas d'us:

6.1. Proves del cas d'us numero 1: Afegir Client

Prova	Resposta del Sistema	Correcte
Donar d'alta un nou client omplint totes les dades i amb l'aplicació moneder en marxa	Client creat correctament	Si
Donar d'alta un nou client sense omplir algun dels camps del formulari	Error, el camp xxx es obligatori	Si
Donar d'alta un nou client amb l'aplicació moneder aturada	Error, no es pot connectar amb l'aplicació moneder	Si
Donar d'alta un nou client amb un nom d'usuari ja enregistrat en la base de dades.	Error, el nom d'usuari indicat ja esta registrat en la base de dades	Si

6.2. Proves del cas d'us numero 2: Modificar Client

Prova	Resposta del Sistema	Correcte
Introduir un nom d'usuari incorrecte	Error,L'usuari no esta registrat en la base de dades	Si
Introduir una contrasenya incorrecte	Error,La contrasenya introduïda no es correcte	Si
No introduir nom d'usuari o contrasenya	Error,El camp xxx es obligatori	Si
Introduir nom d'usuari i contrasenya correctes i modificar dades del usuari	Les dades s'han modificat correctament	Si

6.3. Proves del cas d'us numero 3: Esborrar Client

Prova	Resposta del Sistema	Correcte
Introduir un nom d'usuari incorrecte	Error,L'usuari no esta registrat en la base de dades	Si
Introduir una contrasenya incorrecte	Error,La contrasenya introduïda no es correcte	Si
No introduir nom d'usuari o contrasenya	Error,El camp xxx es obligatori	Si
Introduir nom d'usuari i contrasenya correctes	El client s'ha donat de baixa correctament	Si

6.4. Proves del cas d'us numero 5: Afegir Element

Prova	Resposta del Sistema	Correcte
Intentar afegir un element sense omplir tots els camps del formulari	Error,El camp xxx es obligatori	Si
Afegir un element introduint totes les dades en el formulari	El element s'ha creat correctament	Si

6.5. Proves del cas d'us numero 6: Esborrar Element

Prova	Resposta del Sistema	Correcte
Intentar esborrar un element sense introduir el nom d'usuari o la contrasenya	Error, el camp xxxx es obligatori	Si
Introduir un nom de usuari incorrecte	El nom d'usuari introduït es incorrecte	Si
Introduir una contrasenya incorrecte	La contrasenya introduïda es incorrecte	Si
Introduir tot correctament i seleccionar l'element a esborrar	L'element s'ha esborrat correctament	Si

6.6. Proves del cas d'us numero 7: Crear Monedes

Prova	Resposta del Sistema	Correcte
Introduir les monedes a crear	Les monedes s'han creat correctament	Si

6.7. Proves del cas d'us numero 8: Comprar Element

Prova	Resposta del Sistema	Correcte
Intentar comprar un element sense introduir el nom d'usuari o la contrasenya	Error, el camp xxxx es obligatori	Si
Introduir un nom de usuari incorrecte	El nom d'usuari introduït es incorrecte	Si
Introduir una contrasenya incorrecte	La contrasenya introduïda es incorrecte	Si
Introduir tot correctament i seleccionar l'element a comprar, amb l'aplicació moneder parada.	Error, No es pot connectar amb l'aplicació moneder	Si

Prova	Resposta del Sistema	Correcte
Introduir tot correctament i seleccionar l'element a comprar, amb l'aplicació moneder en marxa.	Em retorna l'element a través de una resposta http	Si
Introduir tot correctament i seleccionar l'element a comprar, l'import d'aquest element serà superior a les monedes de que disposem.	Error, no tens prou monedes per comprar aquest element.	Si

6.8. Proves del cas d'us numero 9: Actualitzar Arrel

Prova	Resposta del Sistema	Correcte
Introduir un nom d'usuari incorrecte	Error, L'usuari no està registrat en la base de dades	Si
Introduir una contrasenya incorrecte	Error, La contrasenya introduïda no és correcta	Si
No introduir nom d'usuari o contrasenya	Error, El camp xxx és obligatori	Si
Introduir nom d'usuari i contrasenya correctes	L'arrel s'ha actualitzat correctament.	Si

7. Conclusió

Un cop implementat el sistema de pagaments PayWord observem els següents beneficis:

- Molt bona opció per a pagaments petits. Altres tipus de pagament, com poden ser el pagament per targeta, graven una comissió sobre el preu del producte, de manera que en pagaments petits aquesta comissió pot no ser rentable, en el sistema PayWord aquesta comissió no existeix.
- Millor rendiment que altres opcions de pagament que treballen amb certificats.
- Molta seguretat, per la xarxa només viatgen monedes payword que no porten cap tipus de informació personal.

També ens trobarem amb una sèrie de problemes:

- El fet de funcionar amb una aplicació que s'ha d'instal·lar en el client. No tothom estarà disposat a instal·lar una aplicació per fer una compra.

- Es necessari trobar un Broker que controli la creació de monedes per part del client.
- S'ha de deixar obert el port 10000 per tal de que el servidor es pugui connectar amb el client, això pot representar un problema de seguretat.

En quant a la implementació que he realitzat del sistema de pagaments PayWord, he de destacar que es tracta de un sistema que funciona correctament, però en el que s'han obviat coses com:

- La necessitat de que la moneda arrel vagi acompanyada de un certificat bancari, s'ha eliminat en aquesta aplicació per simplificar el desenvolupament.
- La persistència tindria que anar a través de una base de dades i no per fitxers, ja que el rendiment de la aplicació se'n pot veure afectat. S'han utilitzat fitxers per simplificar l'aplicació i la seva posterior instal·lació.
- El enviament de dades personals tindria que anar a través de una connexió segura (SSL), no s'ha aplicat perquè es tracta de una aplicació de proves i el fet de posar aquest mòdul complica la instal·lació del servidor de aplicacions.
- S'haurien de fer proves de rendiment més exhaustives, amb algun programari especialitzat (Jmeter podria ser un), per tal de veure el rendiment de la aplicació amb quantitats de peticions grans.

En resum, totes aquests punts que no s'han aplicat en el desenvolupament, serien coses a millorar en el cas de voler posar una aplicació d'aquest tipus en marxa en una botiga real.

8. Bibliografia

- Ingenieria del software – Roger S. Pressman.
- Document Payword and Micromint de Ronald L. Rivest i Adi Shamir.
- Document sobre el sistema de pagaments Payword facilitat per el consultor.
- Tutorials de Servlets i JSP en la pagina oficial de Sun (<http://java.sun.com>).
- Llibre Core Servlets and JavaServer Pages de Marty Hall.
- Exemples de la pagina <http://www.javaalmanac.com>
- Tutorials de la pagina <http://www.javahispano.org>
- Tutorials i API del J2SE en la pagina oficial de Sun (<http://java.sun.com>).
- Documentació sobre el JCE en la pagina oficial de Sun (<http://java.sun.com>).
- <http://www.w3.org/ECommerce/Micropayments>

9. Annex – Codi comentat

Veure fitxer anomenat annex_codi. Entregat junt amb la memòria.