

Web Broker

Aplicació B2C broker on-line

Lluís Calvo Mayor

ETIS

Vicenç Font Sagrista

20/06/2011

Dedicatòria

A Soledad i Maria,

Gràcies per esser tant comprensives amb les incomputables hores que he dedicat a l'estudi de la carrera durant els darrers anys.

Agraïments

A tots els companys de la UOC que m'han ajudat en els meus dubtes, així com als companys de feina que desinteressadament m'han aconsellat en el desenvolupament d'aquest projecte.

Resum del Treball de Fi de Carrera

Per realitzar el treball de final de carrera he escollit l'àrea de J2EE. Com a professional del món del desenvolupament sempre he estat lligat a entorns Host, habitualment en màquines propietàries. El motiu d'escollir aquest paradigma ha estat que si bé he tingut interès en el món Web, mai he pogut desenvolupar cap projecte a nivell professional, i el TFC era una oportunitat per introduir-me en aquest paradigma.

El treball de fi de carrera, consisteix en l'especificació de requeriments, anàlisi, disseny i desenvolupament d'una aplicació de comerç electrònic, centrada en la comercialització de productes financers que cotitzen en mercats organitzats, és a dir, en la compra i venda d'accions i altres actius.

En totes les fases del d'anàlisi, disseny i desenvolupament s'ha utilitzat el paradigma de l'orientació a objectes, com no podria ser d'altra manera dins de l'àrea de J2EE.

L'objectiu a l'hora de realitzar el projecte, ha estat l'aprenentatge de les tecnologies més utilitzades avui en dia en els desenvolupament Web. Així s'ha escollit J2EE com a base, ja que en ell mateix s'integren les tecnologies més utilitzades avui en dia.

Finalment, en el desenvolupament del projecte no he pogut assolir totes les fites plantejades en un inici, ja que com ja he comentat al inici, pràcticament totes les tecnologies utilitzades en aquest projecte eren noves per mi, i la corba d'aprenentatge ha estat força més costosa del que pensava. Ara bé, he complert l'objectiu d'integrar totes les tecnologies que m'havia plantejat en un inici.

Índex de continguts

<i>Dedicatòria</i>	2
<i>Agraïments</i>	2
<i>Resum del Treball de Fi de Carrera</i>	3
<i>Índex de continguts</i>	4
<i>Índex de figures</i>	6
1.- Introducció	7
1.1.- Justificació del TFC	7
1.2.- Objectius del TFC	7
1.3.- Enfocament i mètode seguit	8
1.4.- Planificació del projecte	9
1.5.- Productes obtinguts	11
1.6.- Descripció de la resta de capítols	11
2.- Especificació i anàlisi de requeriments	12
2.1.- Introducció	12
2.2.- Descripció del projecte	12
2.3.- Mòduls funcionals de l'aplicació	13
2.4.- Especificacions funcionals	13
2.5.- Identificació dels actors	15
2.6.- Caso d'ús dels actors	16
2.6.- Requisits del programari i maquinari	30
2.7.- Seguretat	31
3.- Disseny	32
3.1.- Introducció	32
3.2.- J2EE	32
3.3.- Disseny arquitectònic	33
3.4.- Diagrames estàtics	36
3.5.- Disseny de la base de dades	38
4.- Implementació	39
4.1.- Requeriment del programari	39

4.2.- Eines de desenvolupament	39
5.- Valoració econòmica	40
6.- Conclusions	40
7.- Glossari	41
8.- Bibliografia i fonts d'informació	42
9.- Annexos	43
9.1.- Sentències DDL base de dades	43

Índex de figures

Il·lustració 1 - Planificació.....	10
Il·lustració 2 – Casos d'us Administrador	16
Il·lustració 3 – Casos d'us Client.....	23
Il·lustració 4 – Casos d'us Recerca Dades.....	29
Il·lustració 5 – Disseny arquitectònic.....	33
Il·lustració 6 – Patró MVC.....	35
Il·lustració 7 – Diagrama de classes d'entitat	37
Il·lustració 8 – Disseny de la base de dades	38

1.- Introducció

1.1.- Justificació del TFC

La justificació del treball de fi de carrera, és la posta en pràctica dels coneixements assolits durant el curs de la carrera, amb una visió molt pragmàtica. Dins d'aquest enfocament s'ha pretès abastar totes les fases del desenvolupament d'un projecte, des de la recollida de requeriments fins el desenvolupament i implementació.

Durant la fase d'estudi de la carrera, cada àmbit d'estudi es realitza de forma separada i independent, i el TFC ens serveix per integrar els coneixements assolits en diferents assignatures i fases del procés d'aprenentatge, obtenint una visió més global i integrada d'aquests coneixements.

El TFC també ens serveix per realitzar un enfocament molt més pràctic d'aquests coneixements, i enfrontar-nos a situacions molt més properes al món professional que ens trobarem al finalitzar els nostres estudis.

En uns estudis com els nostres, l'evolució de les tecnologies és constant i molt ràpida, fet per el qual molts cops estem estudiant tecnologies que en el món professional ja han estat millorades per altres tecnologies. Aquest TFC també ens permet acostar-nos a l'estat de l'art de l'àrea que hem triat com a tema del TFC.

1.2.- Objectius del TFC

Els objectius del TFC són realitzar l'anàlisi, planificació, disseny i desenvolupament d'una aplicació web B2C, utilitzant com a principal tecnologia J2EE, que és la base de les aplicacions web empresarials d'avui en dia.

A continuació fem un breu resum de les tecnologies, eines i conceptes que hem volgut integrar en el desenvolupament del projecte:

- Conceptes
 - Desenvolupament Front-end
 - Estructura d'aplicacions Web
 - Modelització de dades
 - Connectivitat de base de dades
 - ORM (Object-Relational Mapping)
 - Gestió de sessions
 - Logic de negoci transaccional

- Validació al client i validació al servidor
- Internacionalització
- Seguretat en aplicacions Web
- Patrons de disseny, entre altres MVC (Model-View-Controller) and Session Facade
- Technologies
 - HTML, CSS i JavaScript
 - Servlet i JSP (Java Server Pages)
 - EJB (Enterprise Java Beans)
 - JPA (Java Persistence API)
 - JSTL (JavaServer Pages Tag Library)
 - JDBC (Java Database Connectivity)
- Eines de desenvolupament
 - NetBeans IDE
 - GlassFish (Servidor d'aplicacions Java EE)
 - MySQL (RDBMS Relational DataBase Management Server)
 - MySQL WrokBench (Eina de disseny de Base de Dades)

1.3.- Enfocament i mètode seguit

He pretès realitzar les diferents etapes del projecte, al mateix temps que realitzava l'aprenentatge de les diferents tecnologies i eines de disseny i desenvolupament necessàries per realitzar el projecte.

La corba d'aprenentatge ha estat especialment costosa, més del previst inicial, degut la quantitat i complexitat de les tecnologies a aprendre, i a més a més, al fet de tenir-ho que fer de forma individual. Segurament hagués estat molt menys costos si aquest aprenentatge s'hagués realitzat dins d'un equip de treball; si bé el suport de companys de TFC i del consultor, així com companys de feina ha estat fonamental, no és el mateix que desenvolupar el projecte dins d'un equip.

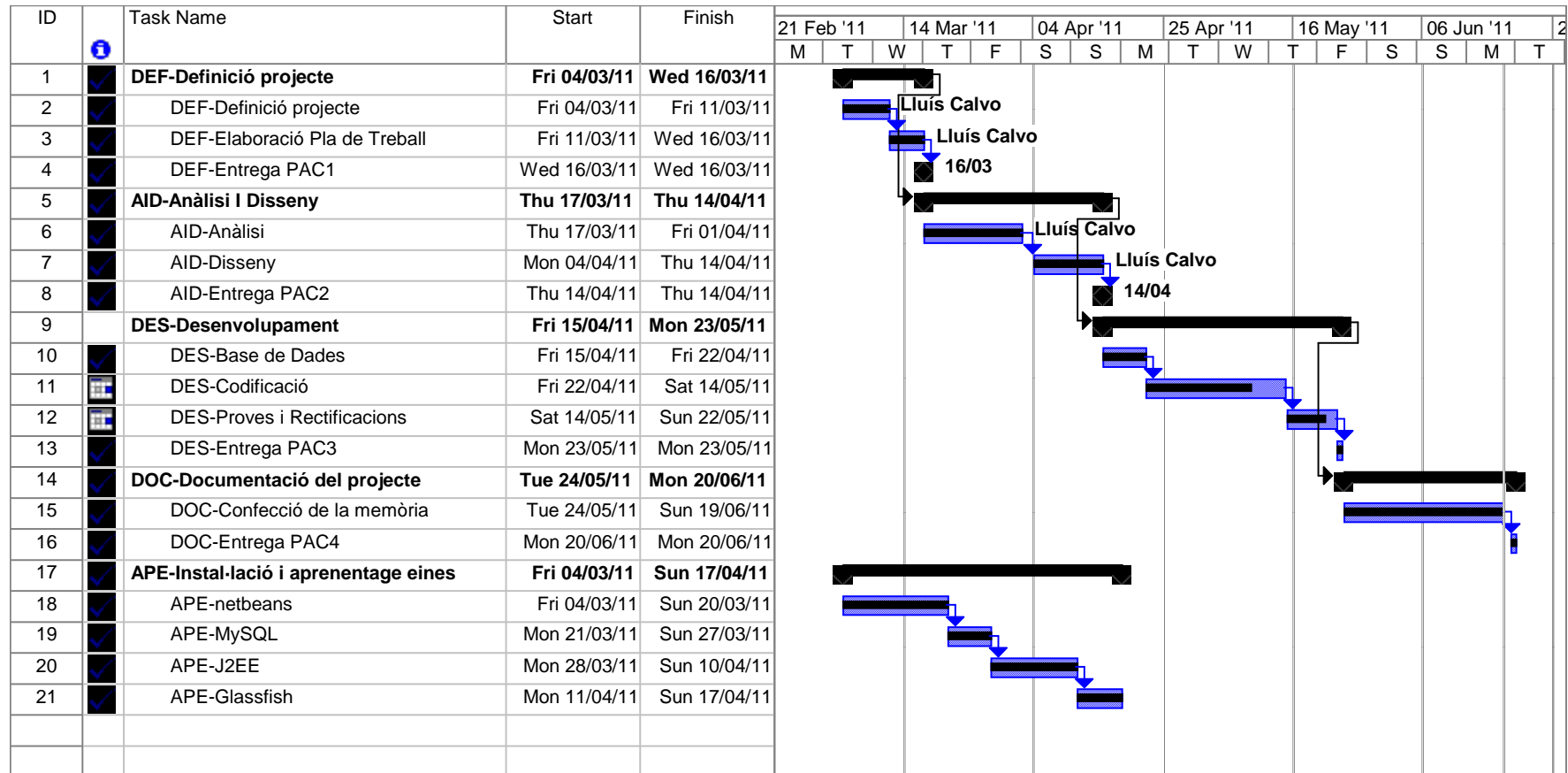
1.4.- Planificació del projecte

Per realitzar la planificació d'aquest projecte he tingut en compte que sóc un estudiant a temps parcial, i que les meves obligacions tant professionals com personals no em deixa gaire temps lliure per desenvolupar aquest projecte.

S'ha fet especialment complicada la tasca d'aprenentatge de totes les tecnologies i eines utilitzades tant el disseny com en el desenvolupament de l'aplicació, requerint finalment força més temps del previst.

Com a eina de planificació he utilitzar el Microsoft Project, degut a que és l'eina que estic habituat a utilitzar, i si bé hagués preferit utilitzar una eina de programari lliure, he optat per no afegir una nova eina més a aprendre.

Com es pot veure les tasques de desenvolupament no s'han assolit completament, restant algunes funcionalitats pendents d'implementar, encara que s'han après i utilitzat totes les tecnologies que volia utilitzar en un principi.



Il·lustració 1 - Planificació

1.5.- Productes obtinguts

El producte final obtingut és una aplicació J2EE plenament funcional, i que inclou els següents elements i documentació:

- Diagrames UML (fitxer zargo)
- Fitxer EAR: Que inclou l'aplicació Web amb les llibreries necessàries per el seu funcionament.
- Codi font de l'aplicació. Dins d'aquest apartat incloc tant el codi font java, com les jsp, fitxers xml de configuració, fitxer de propietats, etc.
- Script de creació de la base de dades
- Script de creació de les dades d'exemple
- Manual d'instal·lació de l'aplicació.

1.6.- Descripció de la resta de capítols

En la resta de capítols de la memòria realitzarem una descripció de les diferents fases del projecte, és a dir, de l'especificació i anàlisi de requeriments, el disseny i la implementació del projecte.

2.- Especificació i anàlisi de requeriments

2.1.- Introducció

En aquest capítol descriurem els requisits del sistema a desenvolupar i implementar. Descriurem les necessitats dels usuaris envers l'aplicació, i les funcionalitats finals que ha de satisfer el programari.

Aquest capítol realitzarà primerament una aproximació general, amb un primer apartat de descripció del projecte, i després anirem detallant el projecte utilitzant diagrames UML i altres.

2.2.- Descripció del projecte

L'escenari inicial és el d'un banc d'inversió que vol desenvolupar una aplicació Web per donar la possibilitat als seus clients d'operar als principals mercats financers, amb l'únic requisit d'un ordinador i una connexió a Internet.

Aquesta aplicació s'ha anomenat Web Broker, i des d'ella es preten que l'usuari pugui accedir a tota la informació dels mercats financers (Mercats, valors, cotitzacions), i si així ho desitja realitzar operatòria (Compra/Venda) des de la mateixa aplicació.

L'entitat financera preten ampliar el seu mercat potencial, ja que amb aquesta eina accedirà a un mercat global, ja que podrà tenir clients a qualsevol lloc del món. A més a més, també es preten baixar els costos a l'hora de que els seus clients operin als mercats de valors, ja que els clients podran comprar o vendre productes financers de forma totalment autònoma, sense tenir que anar a una sucursal de l'entitat o de forma telefònica per que un empleat de l'entitat registri la transacció al sistema. Degut a l'estalvi operatiu que pot suposar aquest nou canal de contractació d'instruments financers, s'ha decidit de implantar unes tarifes més baixes que als canals tradicionals (Sucursal i banca telefònica).

El client accedirà a l'aplicació mitjançant un navegador Web, i la primera acció a realitzar serà el de registre a l'aplicació, sumministrant totes les dades necessàries per a realitzar l'alta al sistema. Un cop registrat l'usuari i validades les dades per part de l'entitat financera, l'usuari ja tindrà un accés complet a l'aplicació. Les funcionalitats a oferir als clients, es poden separar en dos grans blocs, per una banda la de consulta, on el client podrà consultar els mercats, valors i les seves cotitzacions, i per una altra banda la gestió de les seves carteres de valors, que inclou la operativa de compra/venda de valors.

Un requeriment bàsic en l'aplicació serà la seguretat, ja que així ho requereix la pròpia natura de l'aplicació i les conseqüències que es poden derivar d'un accés no autoritzat. Per aquest motiu tota l'aplicació es realitzarà amb una connexió segura (SSL) que per una banda autèntiqui al client i per l'altra, garanteixi que la confidencialitat de la informació intercanviada .

Un altre dels requeriments principals de l'aplicació és el de la internacionalització, ja que per arribar a clients de tot el món, l'aplicació haurà d'estar disponible en l'idioma del client objectiu. En un inici l'aplicació estarà disponible en Català i Anglès, però estarà dissenyada de forma que incorporar una nova llengua sigui tant senzill com traduir els corresponents fitxers de propietats.

L'aplicació també donarà servei als usuaris interns de l'entitat financera, per que puguin realitzar tant la gestió de les dades dels clients, com realitzar el control i seguiment de la operativa que realitzen els mateixos, podent gestionar les ordres que han capturat els clients.

2.3.- Mòduls funcionals de l'aplicació

Primer realitzarem una descripció general de les funcionalitats, per fer-ho dividirem l'aplicació en diferents mòduls funcionals:

- **Administració:** Aquest mòdul inclou totes les funcionalitats relacionades amb el manteniment d'usuaris. Des d'aquest mòdul es realitzarà les altes, baixes, modificacions i consultes dels usuaris interns a l'empresa, és a dir els empleats. També es podran gestionar les dades dels clients, encara que en la majoria dels casos seran ells mateixos els gestionaran les seves dades. Des d'aquest mòdul també gestionarem les ordres de valors introduïdes per els usuaris.
- **Clients:** Des d'aquest mòdul els clients podran realitzar totes les gestions necessàries: Manteniment dades d'usuari, Gestió de carteres, Consultes d'informació financera dels mercats, Ingressos d'efectiu, Gestió d'ordres sobre actius financers.
- **Gestió informació financera:** Aquest mòdul serà l'encarregat de carregar en la base de dades de l'aplicació la informació financera necessària per que posteriorment els clients la puguin consultar. Aquesta informació serà bàsicament les cotitzacions (preus) dels diferents actius financers. En un futur s'anirà ampliant la informació financera que es cercarà, i s'inclouran notícies dels mercats financers i econòmiques.

2.4.- Especificacions funcionals

En aquest apartat realitzarem una descripció més detallada de les funcionalitats que han de proveir cadascun dels mòduls funcionals esmentats en l'apartat anterior.

Administració

Mitjançant aquest mòdul realitzarem el manteniment de les dades dels diferents usuaris del sistema. Aquest mòdul ens permetrà donar d'alta, de baixa, modificar i consultar les característiques dels usuaris. L'únic actor que participa en aquest mòdul és l'administrador. Cal fer notar que quan parlem d'usuaris en referim tant a usuaris interns (empleats), com a clients. En el cas dels clients, seran aquests els que realitzaran el manteniment de les seves pròpies dades, encara que en cas de necessitat d'Administrador podrà realitzar el manteniment de les dades dels clients.

Funcionalitats:

- Consulta usuaris
- Alta usuari
- Baixa usuari

- Modificació usuari
- Consulta ordres de clients segons el seu estat
- Execució d'ordres de clients

Clients

Aquest mòdul serà el que donarà servei als clients, i haurà de proveir a aquest de totes les funcionalitats necessàries.

Funcionalitats:

- Gestió de les dades d'usuari
- Gestió de carteres
- Consultes d'actius que componen les carteres
- Gestió d'ordres sobre actius financers.
- Ingressos d'efectiu
- Consulta d'índex borsaris
- Consulta llista valors d'un índex
- Compra/Venda de valors

Gestió informació financera

Aquest mòdul serà l'encarregat d'obtenir la informació financera d'altres fonts i carregar-la en la base de dades de l'aplicació. La informació més important a carregar, és la dels preus dels diferents actius financers. L'objectiu principal és que els nostres clients sempre tinguin la informació financera necessària, independentment de la font d'informació que s'utilitzi o de la disponibilitat d'aquesta; aquest és el motiu de que aquesta informació s'emmagatzemi a la nostra base de dades. La manera en que el nostre client vegi la informació ha de ser totalment independent de la nostra font.

Funcionalitats:

- Actualitzar la base de dades interna amb la informació financera dels valors.

2.5.- Identificació dels actors

Primerament identificarem els actors, i a després analitzarem els casos d'ús de cadascun dels actors. Els actors identificats són: Administrador, Client, Temporitzador recerca informació.

Administrador

- Validació usuari (Login)
- Consulta llista clients
- Gestió clients (Baixa, Consulta)
- Llista ordres
- Gestió ordres (Consulta, Bloqueig/Desbloqueig, Anul·lació)
- Llista moviments efectiu (Ingressos/Reintegraments)
- Consulta moviment efectiu

Client

Aquest actor correspon al client de l'entitat financera que utilitza els serveis de l'aplicació desenvolupada. El client gestionarà per si mateix les seves carteres de valors, realitzant el manteniment de les mateixes, i realitzarà la compra/venda d'actius financers de forma autònoma.

- Validació client (Login)
- Gestió client (Alta-Registre, Baixa, Modificació, Consulta)
- Llista carteres
- Gestió carteres (Alta, Baixa, Modificació, Consulta, Consulta composició)
- Gestió de posicions (Alta)
- Llista moviments d'efectiu d'una cartera
- Captura moviments d'efectiu (Ingrés/Reintegrament)
- Consulta d'ordres
- Gestió d'ordres (Alta, Anul·lació)

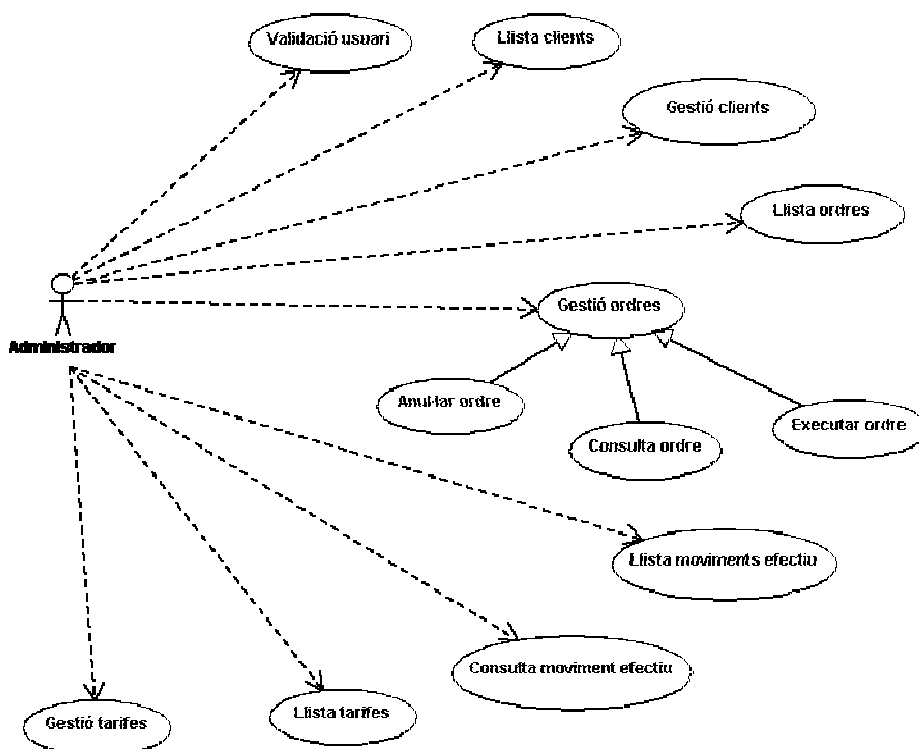
Temporitzador recerca informació

Aquest actor correspondrà a un temporitzador que s'activarà cada cert període de temps, i que anirà a buscar informació financera dels diferents valors, per emmagatzemar-la a la base de dades, i així poder-la servir als nostres clients.

- Actualització preus

2.6.- Caso d'ús dels actors

2.6.1.- Administrador



Il·lustració 2 – Casos d'us Administrador

Cas d'ús "Validació usuari"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat els usuaris interns, es validaran al sistema. Un cop introduïts el codi d'usuari i el mot de pas, aquest es validaran al sistema i s'iniciarà la sessió, amb els permisos adients a l'usuari.

Paper dins del treball de l'usuari: L'administrador i els usuaris interns, hauran de realitzar aquest cas d'ús per poder iniciar una sessió de treball al sistema.

Actor: Administrador

Casos d'ús relacionats: Tots els casos d'ús hi estan relacionats, ja que amb aquest s'obtenen els permisos per poder accedir a la resta de funcionalitats.

Precondició: L'usuari ha accedit a l'adreça URL interna d'administració de l'aplicació, i l'usuari no ha iniciat prèviament sessió a l'aplicació.

Postcondició: Un cop l'usuari ha introduït el nom d'usuari, el mot de pas i ha premut el botó acceptar, es poden produir dues situacions:

- Si el codi d'usuari i el mot de pas són correctes es mostrarà la pantalla principal de treball dels usuaris interns, amb accés a les diferents funcionalitats.
- Si l'usuari i/o el mot de pas no són correctes, es retornarà a la pàgina d'inici de sessió.

A la sessió de l'usuari s'han carregat tots els permisos i opcions segons el seu rol.

Descripció: Es presentarà a l'usuari una pàgina per introduir el codi d'usuari i el mot de pas, així com el botó acceptar. Un cop introduït el codi d'usuari i el mot de pas, és validaran les dades i si són correctes s'iniciarà la sessió, identificant el tipus d'usuari que ha iniciat la sessió, i segons aquest tipus se li assignaran els permisos adients. Al mateix temps es redirigirà el navegador a la pàgina principal de gestió per els usuaris interns, amb accés a les diverses funcionalitats.

Observacions:

Cas d'ús "Consulta llista de clients"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podrà consultar la llista de clients actius al sistema. Des d'aquesta funcionalitat podrem accedir a la baixa i la consulta detallada de clients.

Paper dins del treball de l'usuari: Aquest cas d'ús i els relacionats de manteniment de clients, s'utilitzaran solament en casos extraordinaris, ja que habitualment seran els mateixos clients qui realitzaran el manteniment de les seves dades.

Actor: Administrador.

Casos d'ús relacionats: Gestió de clients.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Post condició: L'usuari podrà accedir a les següents funcionalitats des d'aquesta pàgina:

- Consulta detallada de clients
- Baixa de clients

Descripció: Es mostrarà la llista de clients ordenats alfabèticament, des d'aquesta llista, seleccionant un client, podrem accedir a les funcionalitats de Consulta detallada i Baixa.

Comentaris: Les modificacions i les altes de clients les han de realitzar obligatòriament els mateixos clients, com a mesura de seguretat. Cap usuari del sistema podrà alterar les dades dels clients, si un usuari ha oblidat el seu mot de pas, haurà de contestar a les preguntes clau especificades per ell mateix quan es va donar d'alta, i si les respostes són correctes s'enviarà un correu electrònic a la seva bústia de correu.

Cas d'ús "Baixa de client"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podrà donar de baixa un client en l'aplicació.

Paper dins del treball de l'usuari: Aquest cas d'ús serà d'utilització extraordinària per part dels usuaris interns, ja que habitualment aquestes tasques les realitzarà el mateix usuari. Únicament es realitzarà una baixa de client, per que ho sol·liciti per escrit el mateix client o bé sota requeriments legals.

Actor: Administrador.

Casos d'ús relacionats: Consulta llista de clients i consulta detallada de client.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern. El client ha d'existir al sistema.

Postcondició: El client s'ha donat de baixa.

Descripció: Quan es seleccioni donar de baixa de client es mostraran totes les seves dades en la pàgina, i si es prem el botó acceptar es donarà de baixa el client al sistema. Les dades no es donaran de baixa físicament de la base de dades, sinó que es marcaran amb l'estat de baixa. Aquesta manera de procedir és imprescindible per motius legals, ja que ha de quedar un registre de totes les operacions realitzades per un client durant un període de temps determinat.

Cas d'ús "Consulta detallada de client"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podran consultar totes les dades relatives a un client.

Paper dins del treball de l'usuari: Aquest cas d'ús s'utilitzarà rarament per part dels usuaris interns.

Actor: Administrador.

Casos d'ús relacionats: Consulta llista de clients i Baixa de client.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern. El client que es vol consultar ha d'existir al sistema.

Postcondició: Les dades del client es visualitzaran en una pàgina de consulta.

Descripció: Quan es seleccioni la consulta detallada de les dades del client en la consulta de la llista de clients, es mostraran totes les seves dades en una pàgina on no es permetrà de modificar els camps. Al peu de la pàgina existirà el botó acceptar, i un cop l'usuari el premi es retornarà a la pàgina de la consulta de la llista de clients de l'aplicació.

Cas d'ús "Consulta llista d'ordres"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podrà consultar la llista de les ordres de compra/venda de valors que han realitzat els clients. Des d'aquesta funcionalitat podrem accedir a les opcions de Bloqueig/Desbloqueig i Anul·lació d'ordres.

Paper dins del treball de l'usuari: Aquest cas d'ús i els relacionats amb les ordres, s'utilitzaran com a eines de seguiment i validació de que el sistema funciona correctament i les ordres dels clients s'executen amb normalitat.

Actor: Administrador.

Casos d'ús relacionats: Consulta detallada d'una ordre, Bloqueig/Desbloqueig i anul·lació d'ordres.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Post condició: L'usuari podrà accedir a les següents funcionalitats des d'aquesta pàgina:

- Consulta detallada d'una ordre
- Bloqueig/Desbloqueig ordre
- Anul·lació ordre
- Execució ordre

Descripció: Es mostrarà la llista de les ordres que han introduït al sistema els clients, així com el seu estat. La consulta es podrà filtrar per client, mercat i valor; tots tres filtres seran independents.

Comentaris: Aquest cas d'ús s'utilitzarà amb diverses finalitats, per veure les ordres dels clients, per bloquejar/desbloquejar o anul·lar una ordre d'un client sota sol·licitud d'aquest, per verificar que les ordres dels clients s'estan executant correctament, etc.

Cas d'ús "Consulta detallada d'una ordre"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podran consultar totes les dades relatives a un client.

Paper dins del treball de l'usuari: Aquest cas d'ús i els relacionats amb les ordres, s'utilitzaran com a eines de seguiment i validació de que el sistema funciona correctament i les ordres dels clients s'executen amb normalitat.

Actor: Administrador.

Casos d'ús relacionats: Llista d'ordres, Bloqueig, Desbloqueig i anul·lació d'ordres.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Postcondició: Les dades del client es visualitzaran en una pàgina de consulta.

Descripció: Quan es seleccioni la consulta detallada d'una ordre, es visualitzaran totes les dades relatives a l'ordre en una pàgina on no es permetrà de modificar els camps. Al peu de la pàgina existirà el botó acceptar, i un cop l'usuari el premi es retornarà a la pàgina de la consulta de la llista de clients de l'aplicació.

Cas d'ús "Bloqueig/Desbloqueig i Anul·lació ordre"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat bloquejar, desbloqueig o anul·lar ordres.

Paper dins del treball de l'usuari: Aquest cas d'ús s'utilitzarà rarament per part dels administradors.

Actor: Administrador.

Casos d'ús relacionats: Consulta llista d'ordres i consulta detallada d'ordres.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern. L'ordre sobre la que es vol actuar ha d'existir al sistema, i no ha d'haver estat enviada al broker, és a dir, el seu estat ha de ser "Capturada"

Postcondició: L'estat de l'ordre passarà a estar.

Descripció: Quan es seleccioni l'opció de bloqueig, desbloqueig o anul·lació, es validarà que l'estat de l'ordre sigui capturada, ja que si ja s'ha enviat al mercat no es possible realitzar una anul·lació de la mateixa per mitjans informàtics.

Cas d'ús "Llista moviments efectiu"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podrà consultar la llista dels moviments d'efectiu que han realitzat els clients.

Paper dins del treball de l'usuari: Aquest cas d'ús i els relacionats amb les ordres, s'utilitzaran com a eines de seguiment i validació de que el sistema funciona correctament i que els moviments d'efectiu capturats per els clients es processen correctament.

Actor: Administrador.

Casos d'ús relacionats: Consulta detallada de moviments d'efectiu.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Post condició: Es visualitzarà la llista de moviments d'efectiu que han capturat els clients.

Descripció: Es mostrarà la llista de les ordres que han introduït al sistema els clients, així com el seu estat. La consulta es podrà filtrar per client, cartera, estat i rang de dates; tots aquests filtres seran independents uns dels altres.

Cas d'ús "Consulta detallada moviment efectiu"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podran consultar totes les dades relatives a un moviment d'efectiu.

Paper dins del treball de l'usuari: Aquest cas d'ús i els relacionats amb les ordres, s'utilitzaran com a eines de seguiment per part dels usuaris interns.

Actor: Administrador.

Casos d'ús relacionats: Llista de moviments d'efectiu.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Postcondició: Les dades del moviment es visualitzaran en una pàgina de consulta.

Descripció: Quan es seleccioni la consulta detallada d'un moviment d'efectiu, es visualitzaran totes les dades relatives al moviment en una pàgina on no es permetrà de modificar els camps. Al peu de la pàgina existirà el botó acceptar, i un cop l'usuari el premi es retornarà a la pàgina de la consulta de la llista de moviments d'efectiu.

Cas d'ús "Llista tarifes"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podrà consultar la llista de les tarifes a aplicar a les operacions de valors, i a les facturacions periòdiques.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels usuaris interns.

Actor: Administrador.

Casos d'ús relacionats: Gestió de tarifes (Alta, Baixa, Consulta, Modificació)

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Postcondició: Es visualitzarà la llista de tarifes que existeixen al sistema.

Descripció: Es mostrarà la llista de les tarifes que s'han donat d'alta al sistema. Des d'aquest cas d'ús, es podrà accedir a la gestió de tarifes: Alta, Baixa, Consulta i Modificació.

Cas d'ús "Gestió tarifes"

Resum de la funcionalitat: Mitjançant aquest cas d'ús, realitzarem el manteniment de les tarifes a aplicar en les operacions de compra/venda de valors i de les facturacions periòdiques.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels usuaris interns.

Actor: Administrador.

Casos d'ús relacionats: Llista de tarifes

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Postcondició: S'haurà realitzat la gestió corresponent de la tarifa: Alta, Baixa, Consulta o Modificació.

Descripció: Es tracta d'un formulari amb totes les dades de les tarifes. Existiran diferents tarifes per cada mercat de valors. Depenen de l'operació que estiguem fent el formulari es mostrarà en mode Alta (Buit), Baixa (Valors no modificables i botó de confirmació de baixa), Modificació (Valors modificables i botó d'acceptar) o Consulta (Valors no modificables i botó acceptar). Segons la operatòria i l'acció de l'usuari es realitzarà l'acció adient.

2.6.2.- Client



Il·lustració 3 – Casos d'us Client

Cas d'ús "Validació al sistema d'un client"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat és usuaris interns, es validaran al sistema. Un cop introduïts el codi d'usuari i el mot de pas, aquests es validaran al sistema i s'iniciarà la sessió.

Paper dins del treball de l'usuari: Aquest cas d'ús serà obligatori realitzar-lo, abans de qualsevol altre. No es permetrà treballar amb el sistema a un client no autenticat.

Actor: Client

Casos d'ús relacionats: Tots els casos d'ús hi estan relacionats, ja que mitjançant aquest s'obté la sessió necessària per realitzar qualsevol tipus d'operació.

Precondició: El client ha accedit a l'adreça URL pública de l'aplicació.

Postcondició: Un cop el client ha introduït el nom d'usuari, el mot de pas i ha premut el botó acceptar, es poden produir dues situacions:

- Si el nom d'usuari i el mot de pas són correctes es mostrarà la pantalla principal de treball dels clients, amb accés a les diferents funcionalitats.
- Si el nom d'usuari i/o el mot de pas no són correctes, es retornarà a la pàgina d'inici de sessió.

A la sessió de l'usuari s'han carregat tots els permisos i opcions segons el seu rol.

Descripció: Es presentarà al client una pàgina per introduir el codi d'usuari i el mot de pas, així com el botó acceptar. Un cop introduït el codi d'usuari i el mot de pas, és validaran les dades i si són correctes s'iniciarà la sessió, identificant el tipus d'usuari que ha iniciat la sessió, i segons aquest tipus se li assignaran els permisos adients. Al mateix temps es redirigirà el navegador a la pàgina principal de gestió de carteres.

Observacions: En la pàgina d'inici de sessió també habilitarem un accés directe a l'alta de clients, per que els nous clients es puguin donar d'alta a l'aplicació. També s'habilitarà un segon enllaç per els usuaris que han oblidat el mot de pas, aquest enllaç els portarà a la plana de recuperació del mot de pas.

Cas d'ús "Gestió client"

Resum de la funcionalitat: Mitjançant aquest cas d'ús, el client podrà realitzar la gestió de les seves dades personals i econòmiques.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb poca freqüència, essent obligatori en el registre de l'usuari.

Actor: Client

Casos d'ús relacionats: Validació al sistema.

Precondició: L'actor s'ha d'haver validat correctament al sistema i l'usuari que es vol gestionar a de coincidir amb l'usuari de la sessió.

Postcondició: S'haurà realitzat la gestió corresponent del client: Alta, Baixa, Consulta o Modificació.

Descripció: Es tracta d'un formulari amb totes les dades del client, tant les personals com les econòmiques. Depenen de l'operació que estiguem fent el formulari es mostrarà en mode Alta (Buit), Baixa (Valors no modificables i botó de confirmació de baixa), Modificació (Valors modificables i botó d'acceptar) o Consulta (Valors no modificables i botó acceptar). Segons la operatòria i l'acció de l'usuari es realitzarà l'acció adient.

Cas d'ús "Recuperació mot de pas"

Resum de la funcionalitat: Mitjançant aquest cas d'ús, el client podrà sol·licitar la generació i enviament d'un mot de pas temporal a la seva bústia de correu electrònic.

Paper dins del treball de l'usuari: Aquest cas d'ús s'utilitzarà únicament quan l'usuari hagi oblidat el seu mot de pas.

Actor: Client

Casos d'ús relacionats: Validació al sistema.

Precondició: L'actor ha d'haver seleccionat l'enllaç "He oblidat el mot de pas" de la pàgina de validació de l'aplicació.

Postcondició: Generació d'un nou mot de pas caducat i enviament del mateix a la bústia de correu electrònic del client.

Descripció: Quan el client faci clic a l'enllaç "He oblidat el mot de pas" de la pàgina de validació a l'aplicació, es redirigirà a un formulari on se li farà la pregunta secreta que ell mateix va seleccionar al donar-se d'alta, si la resposta és correcta, es generarà el nou mot de pas caducat i se li enviarà a la seva bústia de correu electrònic.

Cas d'ús "Consulta de la llista de carteres"

Resum de la funcionalitat: Aquest cas d'ús serà el principal de la gestió de carteres. Es mostrarà la llista de carteres que té el client que està interactuant amb el sistema, i des d'aquesta pàgina s'accedirà a la resta de funcionalitats de gestió de carteres i de captura de moviments d'efectiu.

Paper dins del treball de l'usuari: Aquest cas d'ús serà el punt de partida a la resta de casos d'ús de gestió de carteres i moviments d'efectiu.

Actor: Client

Casos d'ús relacionats: Gestió de carteres, Captura moviments d'efectiu.

Precondició: El client està registrat al sistema i s'ha autenticat correctament.

Postcondició: Es visualitzarà una llista de les carteres del client, visualitzant les dades bàsiques d'aquestes i habilitant enllaços per realitzar la gestió de les mateixes.

Descripció: Es mostrarà la llista de les seves carteres i les seves dades bàsiques. Per cada cartera, podrem realitzar les següents opcions: Alta cartera, Consulta dades generals de la cartera, Modificació dades generals de la cartera, Baixa cartera (Únicament si el saldo és 0), Consulta composició de la cartera, Consulta moviments d'efectiu de la cartera, Captura moviments d'efectiu de la cartera, Captura d'ordre de compra/venda de valors.

Observacions: Des d'aquesta pàgina, es podran accedir a gran part de la operativa que pot realitzar el client.

Cas d'ús "Gestió de carteres"

Resum de la funcionalitat: Mitjançant aquest cas d'ús, els clients realitzaran el manteniment de les seves carteres de valors.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb poca freqüència per part dels clients.

Actor: Client.

Casos d'ús relacionats: Llista de carteres, Llista moviments d'efectiu i Captura moviments d'efectiu.

Precondició: L'actor s'ha d'haver validat correctament al sistema i el seu rol ha de ser el d'administrador o usuari intern.

Postcondició: S'haurà realitzat la gestió corresponent de la notícia: Alta, Baixa, Consulta o Modificació.

Descripció: Es un formulari amb totes les dades d'una notícia. Depenen de l'operació que estiguem fent el formulari es mostrarà en mode Alta (Buit), Baixa (Valors no modificables i botó de confirmació de baixa), Modificació (Valors modificables i botó d'acceptar) o Consulta (Valors no modificables i botó acceptar). Segons la operatòria i l'acció de l'usuari es realitzarà l'acció adient.

Cas d'ús "Llista moviments efectiu d'una cartera"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podran consultar el moviments d'efectiu d'una cartera.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels clients.

Actor: Client.

Casos d'ús relacionats: Llista de carteres, Captura de moviments d'efectiu.

Precondició: L'actor s'ha d'haver validat correctament al sistema i ha seleccionat una cartera per consultar els seus moviments d'efectiu.

Postcondició: Es visualitzarà la llista de moviments d'efectiu de la cartera en ordre descendent per data.

Descripció: Es mostrarà la llista dels moviments d'efectiu de la cartera en qüestió, en ordre descendent per data. Es podran aplicar filtres per data, per poder acotar la consulta.

Cas d'ús "Captura moviments d'efectiu (Ingrés/Reintegrament)"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat capturarem ordre d'ingrés o reintegrament d'efectiu en una cartera de valors.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels clients.

Actor: Client.

Casos d'ús relacionats: Llista de carteres, Consulta de moviments d'efectiu.

Precondició: L'actor s'ha d'haver validat correctament al sistema i ha seleccionat una cartera per realitzar la captura del moviment d'efectiu.

Postcondició: Es poden donar dues possibilitats:

- Si l'ordre es pot executar bé, s'aplica el moviment al compte d'efectiu del client.
- Si l'ordre no es pot executar correctament, s'informa de l'error i es torna a la pàgina de captura, on el client podrà modificar les dades i tornar a intentar l'operació, o bé podrà cancel·lar l'operació.

Descripció: Es mostrarà un formulari al client, en aquest formulari s'haurà d'especificar si es tracta d'un ingrés o bé de un reintegrament. En qualsevol dels dos casos haurem d'escollir la modalitat de pagament/cobrament: Targeta de crèdit o Un compte d'una entitat bancària. Segons el tipus de pagament es demanà autorització.

Cas d'ús "Llista ordres"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat es podran consultar les ordres de compra/venda de valors d'una cartera.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels clients.

Actor: Client.

Casos d'ús relacionats: Llista de carteres, Captura d'ordres de compra/venda.

Precondició: L'actor s'ha d'haver validat correctament al sistema i ha seleccionat una cartera per consultar les seves ordres de compra/venda de valors.

Postcondició: Es visualitzarà la llista d'ordres de compra/venda de valors de la cartera en ordre descendent per data.

Descripció: Es mostrarà la llista d'ordres de compra/venda de valors, en ordre descendent per data, i mostrant les seves característiques principals: Valor, Si és una compra o una venda, si és limitada i a mercat, i en el cas de ser limitada, la limitació i la data caducitat de la mateixa.

Cas d'ús "Gestió d'ordres (Alta, Anul·lació)"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat realitzarem la captura (o Anul·lació) de les ordres de compra/venda de valors.

Paper dins del treball de l'usuari: Aquest cas d'ús és realitzarà amb certa freqüència per part dels clients.

Actor: Client.

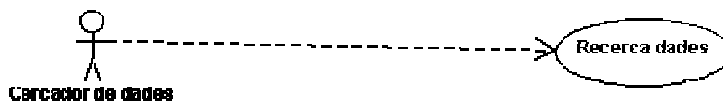
Casos d'ús relacionats: Llista de carteres, Gestió carteres (Consulta composició cartera).

Precondició: L'actor s'ha d'haver validat correctament al sistema i ha seleccionat una cartera per realitzar la captura de l'ordre de compra/venda.

Postcondició: L'ordre s'introdueix al sistema, i segueix el seu cicle de vida segons la seva natura: Compra/Venda, Limitada/Mercat, etc.

Descripció: Es mostrarà un formulari al client, en aquest formulari s'haurà d'especificar si es tracta d'una compra o d'una venda, de si és a mercat o limitada, i en aquest darrer cas s'haurà d'indicar el preu límit i la data caducitat.

2.6.3.- Temporitzador recerca informació



Il·lustració 4 – Casos d'us Recerca Dades

Cas d'ús "Actualització preus"

Resum de la funcionalitat: Mitjançant aquesta funcionalitat s'iniciarà el procés que realitza la càrrega de preus dels actius financers a la base de dades de l'aplicació.

Paper dins del treball de l'usuari: Aquest és el cas d'ús únic del Temporitzador de recerca d'informació.

Actor: Temporitzador de recerca d'informació.

Casos d'ús relacionats: Cap.

Precondició: S'ha produït l'esdeveniment de temps establert.

Postcondició: S'han actualitzat els preus a la base de dades..

Descripció: Aquest procés s'executarà en un interval de temps relativament petit (cada 5 minuts com a molt), i anirà a buscar tots els preus actualitzats dels actius a les fonts de dades establertes. Un cop obtinguda la informació, s'actualitzarà a la base de dades de l'aplicació per que els clients la puguin consultar.

2.6.- Requisits del programari i maquinari

2.6.1.- Requisits del programari

Clients

L'únic requisit de la banda del client, es que l'accés es realitzi amb un navegador web. Si bé en un principi el desenvolupament s'ha fet perquè funcioni amb Firefox 3.6, s'ha tingut en ment que funcioni per la resta dels principals navegadors web del mercat.

Servidors

De la banda de servidor necessitem dos components principals, un servidor d'aplicacions web J2EE, i un servidor de base de dades. A l'hora d'escollir el programari, hem tingut en compte com a requeriments principals, que siguin plataformes obertes i de codi lliure, i al mateix temps, que siguin programaris àmpliament utilitzats i d'una fiabilitat contrastada. L'objectiu és escollir productes fiables, escalables, multiplataforma i econòmics.

Com a servidor d'aplicacions web J2EE, hem escollit Glassfish 3.0.1, degut a que és gratuït i per altra banda és la implementació de referència del J2EE versió 6, que és la que hem utilitzat en els desenvolupament.

Com a servidor de base de dades hem escollit MySQL 5.1, per els mateixos motius, és a dir, és gratuït i àmpliament utilitzat.

En quan el sistema operatiu, tot el programari és multiplataforma, per tant, és independent del sistema operatiu. El desenvolupament s'ha realitzat sobre un sistema Windows, però la instal·lació en l'entorn de producció, es podria fer sobre un sistema Linux, per seguir la mateixa filosofia que la resta de programari.

2.6.2.- Requisits del maquinari

Clients

Els clients no tindran cap requisit de maquinari especial, ja que al ser l'accés des d'un navegador web, els requisits de maquinari són mínims

Servidor

En qual al requisits de maquinari de la part servidora, cal fer notar que els requeriments seran variables en funció del volum de transaccions a suportar; al ser una aplicació nova, que parteix de zero clients, començarem amb uns requisits mínims, però amb previsió de poder-los ampliar de forma senzilla i transparent.

Donat això, creiem que el més convenient és utilitzar la virtualització de servidors, utilitzant una instància per el servidor de base de dades, i una segona instància per el servidor d'aplicacions web. La virtualització ens permetrà d'assignar més recursos a cada una de les instàncies segons les necessitats que s'identifiquin al llarg del temps.

2.7.- Seguretat

2.7.1.- Seguretat d'accés

Degut a la pròpia natura de l'aplicació i als riscos en cas d'accés no autoritzat a l'aplicació, tots els accessos a l'aplicació es faran mitjançant una connexió segura (SSL – Secure Socket Layer), que ens garantirà l'autenticació del client, la confidencialitat de la informació intercanviada, i la no alteració de la mateixa. La implementació de la capa de seguretat la realitzarà el servidor d'aplicacions web.

Cal fer notar, que degut a la natura de l'activitat que es portarà a terme a l'aplicació, i que les autoritats reguladores exigeixen un control dels clients que es donin d'alta a l'aplicació (Consulta de fitxers de clients no desitjats dels reguladors), el procés d'alta dels clients es realitzarà de forma manual al servidor d'aplicacions, per garantir el control previ a l'alta.

2.7.2.- Seguretat de les dades

S'identifica la necessitat d'establir una política de còpies de seguretat per preveure cap tipus d'incidència. També s'ha de tenir present que s'hauran de conservar les dades de les transaccions realitzades per els clients, tant temps com les autoritats reguladores exigeixin.

3.- Disseny

3.1.- Introducció

En l'apartat de disseny realitzarem una proposta de solucions tècniques per donar resposta als requeriments especificats en l'apartat anterior. Explicarem totes les tecnologies utilitzades per donar implementar les funcionalitats especificades en els documents de requeriments, així com la manera d'integrar les diverses tecnologies.

3.2.- J2EE

J2EE (Java 2 Enterprise Edition) es una plataforma tecnològica, àmpliament utilitzada i que conté un conjunt de tecnologies coordinades que redueixen significativament el cost i la complexitat del desenvolupament, instal·lació i la gestió de aplicacions basades en el servidor. J2EE afegeix un conjunt de tecnologies a la plataforma estàndard de Java (SE), per facilitar la construcció d'aplicacions de servidor que siguin portables, robustes, segures i escalables.

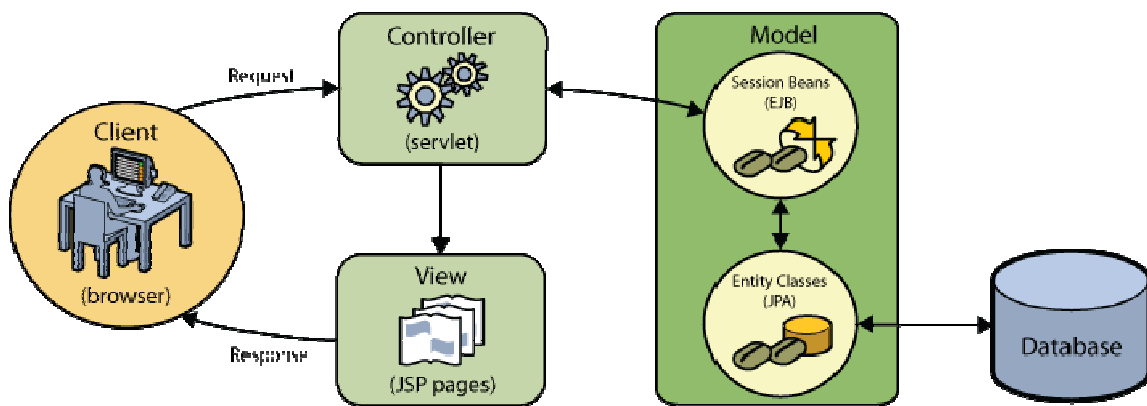
Algunes de les principals tecnologies del conjunt de J2EE són les següents:

- Enterprise Java Beans (EJB): Són components de servidor que són usats per encapsular la lògica de negoci d'una aplicació. La tecnologia EJB permet el desenvolupament ràpid i senzill d'aplicacions distribuïdes, transaccionals, segures i portables basades en la tecnologia Java. Nosaltres utilitzarem la versió 3.1 de EJB.
- Java Persistence API (MPA): És un marc de treball que permet als desenvolupadors gestionar les dades usant ORM (Object-Relational Mapping) en aplicacions construïdes sobre la tecnologia Java.

3.3.- Disseny arquitectònic

3.3.1.- Arquitectura de l'aplicació

En aquest apartat descriurem la arquitectura de l'aplicació. En el següent diagrama mostrem les quatre capes bàsiques en que dividim l'aplicació, a fi d'aconseguir una separació entre les diferents capes, que col·laboraran per assolir la funcionalitat requerida:



Il·lustració 5 – Disseny arquitectònic

Capa Client

Aquesta és la capa amb la que interactua l'usuari. N'hi ha de diversos tipus, diferenciant-se en les capacitats que tenen, dividint-se en clients lleugers (Poca capacitat) i clients pesats (Molta capacitat).

Els navegadors web, són segurament els clients més utilitzats avui en dia, i són els que utilitzarem en la nostra aplicació. El navegador web ens dona una gran independència del Sistema Operatiu de la màquina que s'utilitza com a client, i ens evita de instal·lar i mantenir una aplicació a la màquina de l'usuari final.

Capa web

Aquesta capa és l'encarregada de rebre les peticions dels clients, realitzar peticions al seu torn a la capa de lògica de negoci, i donar format a les dades per que es puguin presentar al client. Es podria dir que és com una component que d'una banda aïlla al client de la lògica de negoci, i de l'altra aïlla a la lògica de negoci de la presentació de les dades al client.

En aquesta capa és on s'implanta un dels patrons de disseny que més èxit han tingut en les aplicacions web, el patró MVC (Model-Vista-Controllador), i que descriurem amb més detall en un

dels següents apartats.

Capa lògica de negoci

En la capa de lògica de negoci, és on encapsulem les entitats i les regles de negoci pròpies a l'empresa i a la seva activitat. És en aquesta capa on estarà present amb tot el seu abast la tecnologia J2EE, i els seus components fonamentals, els EJB.

En aquesta aplicació utilitzarem principalment dos tipus d'EJB:

Beans d'entitat: Aquests són els encarregats de representar les dades del negoci, són el reflex de l'estructura de la base de dades, i són els encarregats d'aconseguir la persistència de les dades.

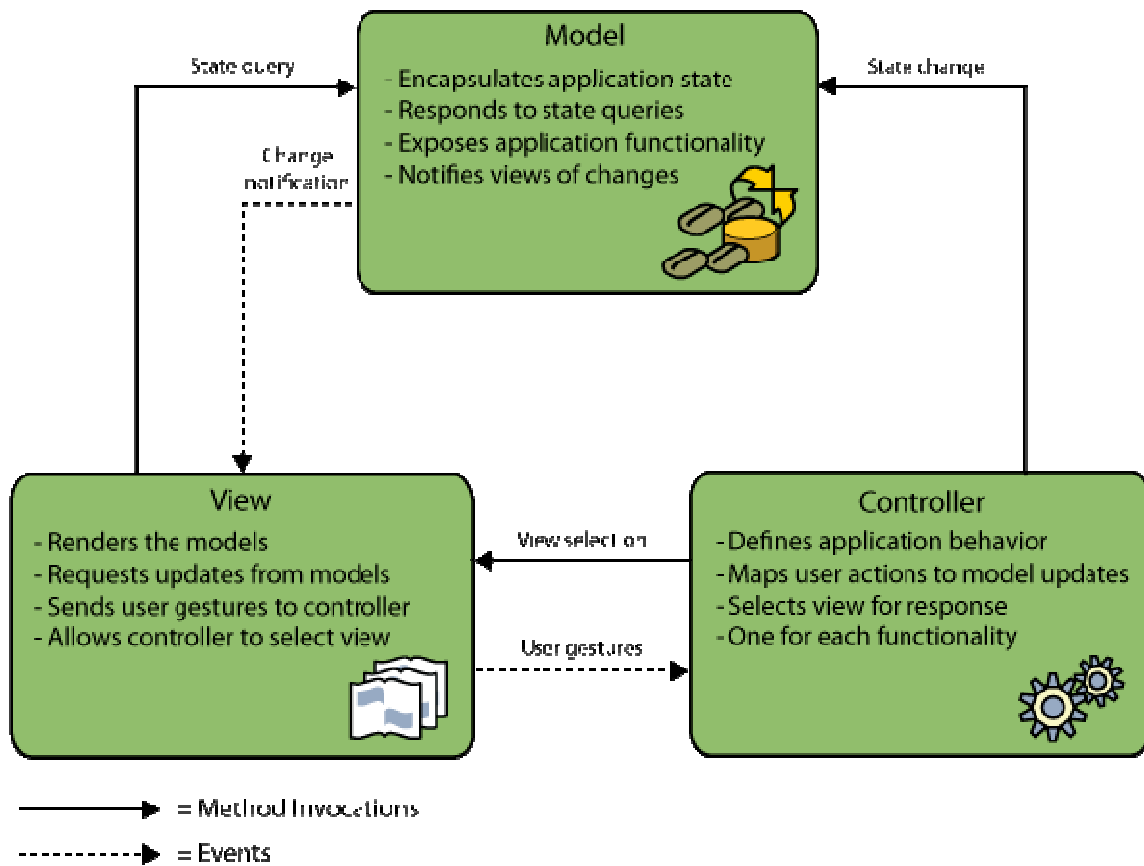
Beans de sessió: Són els encarregats d'implementar la lògica de negoci, les funcions necessàries per executar les accions que es desprenen de les peticions dels usuaris.

Capa EIS (Enterprise Information Systems)

En aquesta capa ens encarregaríem de encapsular l'accés a les diferents bases de dades, com altres sistemes d'informació heretats. D'aquesta manera s'aïllen les característiques específiques d'aquestes bases de dades o sistemes heretats, de la lògica de negoci. En el nostre cas, la capa EIS es tracta únicament de la base de dades.

3.3.2.- Patró Model-Vista-Contolador

El disseny arquitectònic de l'aplicació s'ha basat en el abastament utilitzat patró MVC (Model-View-Controller), que consisteix en dividir l'aplicació en tres capes que interaccionen entre elles.



Il·lustració 6 – Patró MVC

Model

Representa les dades i la lògica de negoci i és l'encarregada de gestionar l'accés i actualització de les dades. El model notifica a la visat quan les dades canvien, i permet a la vista realitzar consultes sobre el model.

Vista

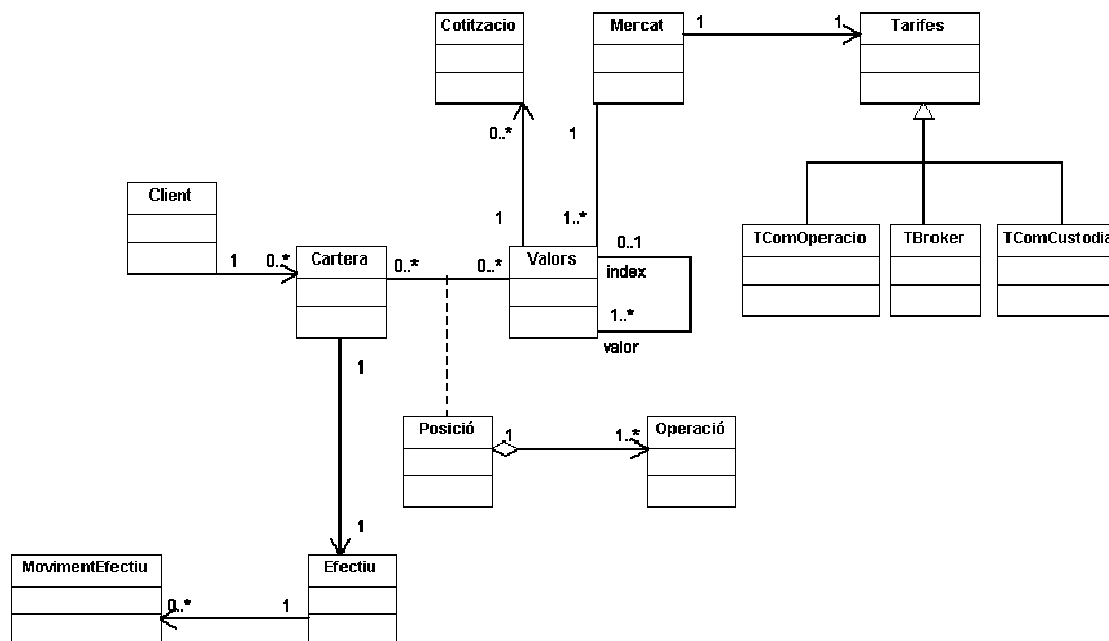
La vista presenta les dades del model a l'usuari. La vista recupera les dades del model, i especifica com aquestes dades s'han de presentar a l'usuari. Actualitza la presentació quan el model canvia. La vista també es l'encarregada de enviar les dades introduïdes per l'usuari a el controlador.

Controlador

El controlador defineix el comportament de l'aplicació. Gestiona les peticions rebudes de l'usuari, i selecciona les vistes a utilitzar per presentar la informació. Interpreta les entrades dels usuaris i les adreça cap a les accions que ha de realitzar el model. En una aplicació web com la que ens ocupa les entrades de l'usuari són les peticions HTTP GET i POST, amb les dades rebudes el controlador decideix quines accions s'han de realitzar i quina vista s'ha d'escollir.

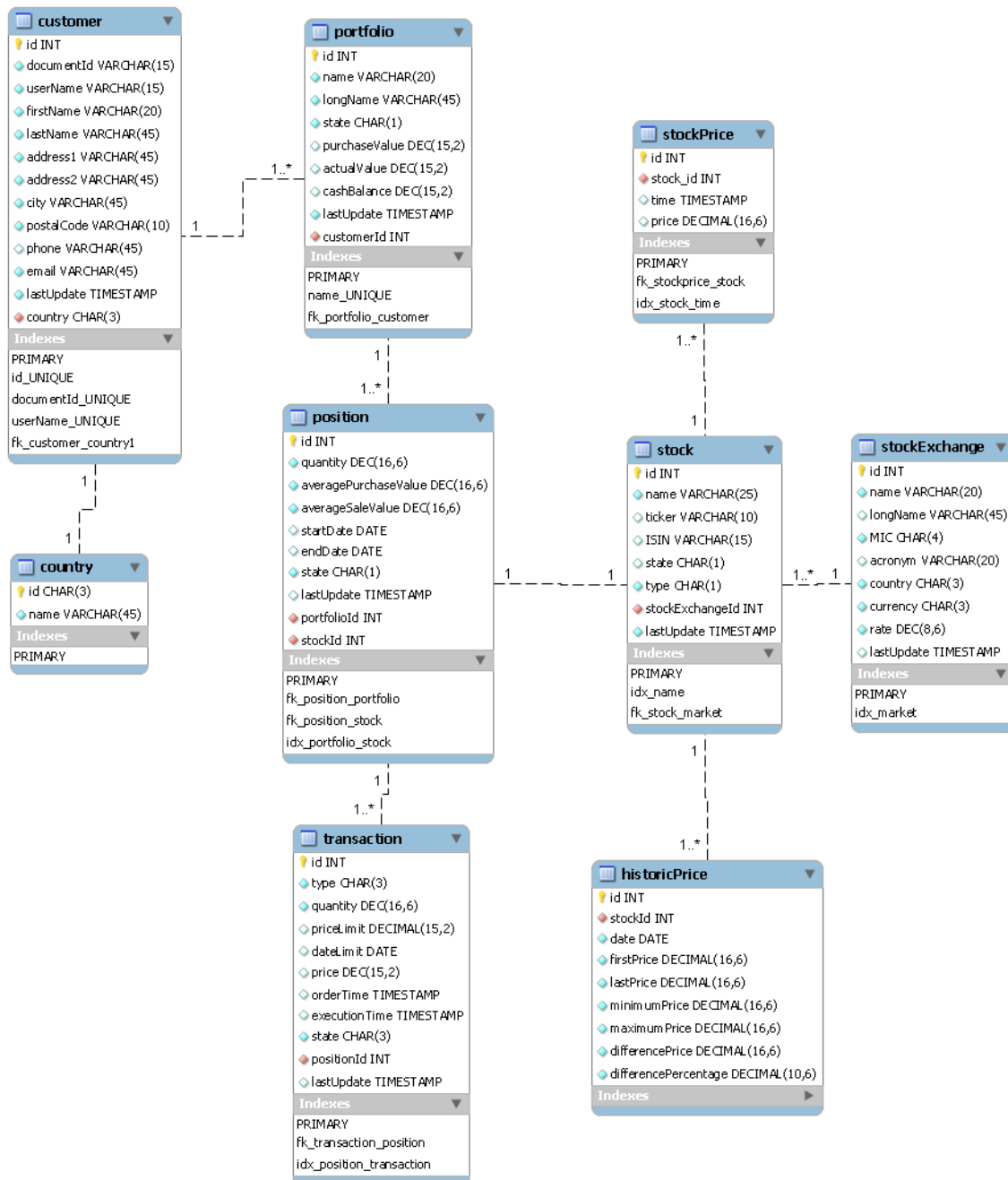
3.4.- Diagrames estàtics

3.4.1.- Diagrama de classes d'entitat



Il·lustració 7 – Diagrama de classes d'entitat

3.5.- Disseny de la base de dades



Il·lustració 8 – Disseny de la base de dades

4.- Implementació

4.1.- Requeriment del programari

A continuació detallarem els requisits de programari dels diferents elements de l'aplicació:

- Navegador Web: S'ha utilitzat com a navegador de referència el Firefox 3.6, ja que és reconegut com el que millor compleix els estàndards. De totes maneres l'aplicació hauria de funcionar amb qualsevol navegador que compleixi amb HTML 4 o superior.
- Java EE 6 SDK
- Glassfish 3.0.1: Servidor d'aplicacions que compleix amb les especificacions Java EE 6. De fet és la implementació de referència.
- MySQL 5.1: Servidor de base de dades

4.2.- Eines de desenvolupament

Les eines de programari que s'han utilitzat durant el procés de disseny i desenvolupament són les següents:

- Netbeans 6.9.1: IDE (Integrated Development Environment) utilitzar per realitzar el desenvolupament del codi i els fitxers de configuració.
- MySQL WorkBench 5.2.: Eina de disseny de base de dades de MySQL

5.- Valoració econòmica

No s'ha realitzat una valoració econòmica del projecte, val a dir però, que hem intentat utilitzar sempre eines de codi lliure i sense llicència, per reduir al màxim el cost del programari necessari, tant en la fase de desenvolupament del mateix, com en la fase de producció.

No s'ha realitzar cap valoració econòmica del maquinari necessari per la posta en producció del projecte.

6.- Conclusions

L'objectiu final del TFC es portar a la pràctica el màxim dels coneixements adquirits durant l'estudi de la carrera. L'elecció de J2EE com a àrea del TFC va ser motivada per que en la meua vida professional no he tingut oportunitat de treballar amb aquestes tecnologies, i tenia moltes ganes d'entrar en contacte amb elles.

No he pogut acomplir tots els objectius que m'havia fixat, i el motiu principal ha estat que la corba d'aprenentatge de totes les tecnologies i eines a utilitzar ha estat més costosa del que em pensava en un inici. He tingut que eliminar algun element dels que m'havia proposat, com integrar algun dels frameworks que implementen la part de la vista del model MVC. Havia pensat en integrar Spring, però degut a que ja anava endarrerit en la implementació d'altre tecnologies, i a la complexitat de Spring, he optat finalment per no utilitzar-lo.

De totes maneres, crec que puc estar moderadament satisfet, ja que abans d'iniciar el TFC, únicament coneixia aquestes tecnologies a nivell teòric, i mai havia treballat amb elles. Ara un cop realitzat el TFC, he de fer una valoració positiva de totes les tecnologies que he après, i sobretot d'haver fet un projecte de desenvolupament passant per totes les fases, cosa que és poc probable que m'hi trobi en la meua vida professional.

7.- Glossari

Administrador: Usuari empleat de la entitat financera que serà l'encarregat de realitzar la gestió de l'aplicació.

Client: Un usuari extern.

Cartera: Les posicions s'agrupen en carteres, és com una carpeta on posem totes les nostres posicions de valors, i de la qual es realitza una valoració econòmica del seu rendiment.

Contravalor: El valor d'una posició financera en una moneda donada

Instrument financer: Quelcom que es negocia a un mercat financer (Sinònim de valor)

Posició: La quantitat que es té d'un determinat valor. Se'n diu la posició d'un client en un valor.

Valor: Quelcom que es negocia a un mercat financer (Sinònim d'instrument financer)

8.- Bibliografia i fonts d'informació

The Java EE 5 Tutorial – Sun Microsystems

El lenguaje unificado de modelado – Booch – Rungaugh – Jacobson

La Biblia MySQL – Anaya

Disseny de bases de dades – Dolors Costal Costa (UOC)

Enginyeria del programari – UOC

Thinking in Java – Bruce Eckel

9.- Annexos

9.1.- Sentències DDL base de dades

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `webbroker` ;
CREATE SCHEMA IF NOT EXISTS `webbroker` ;
USE `webbroker` ;

-----
-- Table `webbroker`.`country`
-----
DROP TABLE IF EXISTS `webbroker`.`country` ;

CREATE TABLE IF NOT EXISTS `webbroker`.`country` (
  `id` CHAR(3) NOT NULL ,
  `name` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB,
COMMENT = 'Taula de països' ;

-----
-- Table `webbroker`.`customer`
-----
DROP TABLE IF EXISTS `webbroker`.`customer` ;

CREATE TABLE IF NOT EXISTS `webbroker`.`customer` (
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'idCustomer' ,
  `documentId` VARCHAR(15) NOT NULL ,
  `userName` VARCHAR(15) NOT NULL ,
  `firstName` VARCHAR(20) NOT NULL ,
  `lastName` VARCHAR(45) NOT NULL ,
  `address1` VARCHAR(45) NOT NULL ,
  `address2` VARCHAR(45) NOT NULL ,
  `city` VARCHAR(45) NOT NULL ,
  `postalCode` VARCHAR(10) NOT NULL ,
  `phone` VARCHAR(45) NULL ,
  `email` VARCHAR(45) NOT NULL ,
  `lastUpdate` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP COMMENT 'Darrera actualització' ,
  `country` CHAR(3) NOT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) ,
  UNIQUE INDEX `documentId_UNIQUE` (`documentId` ASC) ,
  UNIQUE INDEX `userName_UNIQUE` (`userName` ASC) ,
```

```

INDEX `fk_customer_country1` (`country` ASC) ,
CONSTRAINT `fk_customer_country1`
  FOREIGN KEY (`country` )
  REFERENCES `webbroker`.`country` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
COMMENT = 'Customer data' ;

-----
-- Table `webbroker`.`portfolio`
-----
DROP TABLE IF EXISTS `webbroker`.`portfolio` ;

CREATE TABLE IF NOT EXISTS `webbroker`.`portfolio` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(20) NOT NULL COMMENT 'Nom de la cartera' ,
  `longName` VARCHAR(45) NOT NULL ,
  `state` CHAR(1) NOT NULL COMMENT 'Estat de la cartera:\n0
  Operativa\nB Bloquejada\nA Anul·lada' ,
  `purchaseValue` DECIMAL(15,2) NULL DEFAULT 0 ,
  `actualValue` DECIMAL(15,2) NULL DEFAULT 0 ,
  `cashBalance` DECIMAL(15,2) NULL DEFAULT 0 ,
  `lastUpdate` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP ,
  `customerId` INT NOT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) ,
  INDEX `fk_portfolio_customer` (`customerId` ASC) ,
  CONSTRAINT `fk_portfolio_customer`
  FOREIGN KEY (`customerId` )
  REFERENCES `webbroker`.`customer` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB,
COMMENT = 'Stocks portfolio' ;

-----
-- Table `webbroker`.`stockExchange`
-----
DROP TABLE IF EXISTS `webbroker`.`stockExchange` ;

CREATE TABLE IF NOT EXISTS `webbroker`.`stockExchange` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(20) NOT NULL COMMENT 'Name' ,
  `longName` VARCHAR(45) NULL ,
  `MIC` CHAR(4) NOT NULL COMMENT 'MIC code of the market' ,
  `acronym` VARCHAR(20) NULL ,
  `country` CHAR(3) NOT NULL ,
  `currency` CHAR(3) NOT NULL ,
  `rate` DECIMAL(8,6) NOT NULL COMMENT 'Tarifa a aplicar a les
operacions' ,

```

```
`lastUpdate` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP ,  
PRIMARY KEY (`id`) ,  
INDEX `idx_market` (`name` ASC) )  
ENGINE = InnoDB,  
COMMENT = 'market' ;
```

```
-----  
-- Table `webbroker`.`stock`  
-----
```

```
DROP TABLE IF EXISTS `webbroker`.`stock` ;
```

```
CREATE TABLE IF NOT EXISTS `webbroker`.`stock` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(25) NOT NULL ,  
  `ticker` VARCHAR(10) NULL ,  
  `ISIN` VARCHAR(15) NULL ,  
  `state` CHAR(1) NULL ,  
  `type` CHAR(1) NOT NULL DEFAULT 'N' COMMENT 'Indicador índex' ,  
  `stockExchangeId` INT NOT NULL ,  
  `lastUpdate` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP ,  
  PRIMARY KEY (`id`) ,  
  INDEX `idx_name` (`name` ASC) ,  
  INDEX `fk_stock_market` (`stockExchangeId` ASC) ,  
  CONSTRAINT `fk_stock_market1`  
    FOREIGN KEY (`stockExchangeId` )  
    REFERENCES `webbroker`.`stockExchange` (`id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB,  
COMMENT = 'stock or index' ;
```

```
-----  
-- Table `webbroker`.`position`  
-----
```

```
DROP TABLE IF EXISTS `webbroker`.`position` ;
```

```
CREATE TABLE IF NOT EXISTS `webbroker`.`position` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `quantity` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Quantitat' ,  
  `averagePurchaseValue` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Preu  
de cost mig' ,  
  `averageSaleValue` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Preu de  
venda mig' ,  
  `startDate` DATE NULL COMMENT 'Data inici de la posició' ,  
  `endDate` DATE NULL COMMENT 'Data final de la posició' ,  
  `state` CHAR(1) NOT NULL DEFAULT 'V' COMMENT 'Estat' ,  
  `lastUpdate` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP ,  
  `portfolioId` INT NOT NULL ,  
  `stockId` INT NOT NULL ,
```

```

PRIMARY KEY (`id`) ,
INDEX `fk_position_portfolio` (`portfolioId` ASC) ,
INDEX `fk_position_stock` (`stockId` ASC) ,
INDEX `idx_portfolio_stock` (`portfolioId` ASC, `stockId` ASC) ,
CONSTRAINT `fk_position_portfolio`
  FOREIGN KEY (`portfolioId` )
  REFERENCES `webbroker`.`portfolio` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_position_stock`
  FOREIGN KEY (`stockId` )
  REFERENCES `webbroker`.`stock` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB,
COMMENT = 'stock position' ;

-----
-- Table `webbroker`.`transaction`
-----
DROP TABLE IF EXISTS `webbroker`.`transaction` ;

CREATE TABLE IF NOT EXISTS `webbroker`.`transaction` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `type` CHAR(3) NOT NULL COMMENT 'Operation type\nACT      Buy\nVCT
  Shell\nINC Increment\nDEC      Decrement' ,
  `quantity` DECIMAL(16,6) NOT NULL ,
  `priceLimit` DECIMAL(15,2) NULL ,
  `dateLimit` DATE NULL ,
  `price` DECIMAL(15,2) NULL ,
  `orderTime` TIMESTAMP NULL ,
  `executionTime` TIMESTAMP NULL ,
  `state` CHAR(3) NOT NULL ,
  `positionId` INT NOT NULL ,
  `lastUpdate`  TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_transaction_position` (`positionId` ASC) ,
  INDEX `idx_position_transaction` (`positionId` ASC, `orderTime` ASC) ,
  CONSTRAINT `fk_operation_position`
    FOREIGN KEY (`positionId` )
    REFERENCES `webbroker`.`position` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB,
COMMENT = 'Position\'s operations' ;

-----
-- Table `webbroker`.`stockPrice`
-----
DROP TABLE IF EXISTS `webbroker`.`stockPrice` ;

```

```
CREATE TABLE IF NOT EXISTS `webbroker`.`stockPrice` (  
  `id` INT NOT NULL ,  
  `stock_id` INT NOT NULL ,  
  `time` TIMESTAMP NULL ,  
  `price` DECIMAL(16,6) NULL ,  
  PRIMARY KEY (`id`) ,  
  INDEX `fk_stockprice_stock` (`stock_id` ASC) ,  
  INDEX `idx_stock_time` (`stock_id` ASC, `time` ASC) ,  
  CONSTRAINT `fk_stockprice_stock`  
    FOREIGN KEY (`stock_id` )  
    REFERENCES `webbroker`.`stock` (`id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB,  
COMMENT = 'stock\'s price' ;
```

```
-----  
-- Table `webbroker`.`historicPrice`  
-----
```

```
DROP TABLE IF EXISTS `webbroker`.`historicPrice` ;
```

```
CREATE TABLE IF NOT EXISTS `webbroker`.`historicPrice` (  
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'id' ,  
  `stockId` INT NOT NULL COMMENT 'stock id' ,  
  `date` DATE NOT NULL COMMENT 'Price date' ,  
  `firstPrice` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'First price' ,  
  `lastPrice` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Last price' ,  
  `minimumPrice` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Minimum price'  
,  
  `maximumPrice` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Maximum price'  
,  
  `differencePrice` DECIMAL(16,6) NOT NULL DEFAULT 0 COMMENT 'Diference  
between last price of current day and last price of last working day' ,  
  `differencePercentage` DECIMAL(10,6) NOT NULL DEFAULT 0 COMMENT  
'Percentage diference between last price of current day and last price of  
last working day' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `fk_historicPrice_stock` (`stockId` ASC) ,  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) ,  
  INDEX `stock_date` (`stockId` ASC, `date` ASC) ,  
  CONSTRAINT `fk_historicPrice_stock1`  
    FOREIGN KEY (`stockId` )  
    REFERENCES `webbroker`.`stock` (`id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB,  
COMMENT = 'Historic prices\nOne record per day' ;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```