



## TREBALL FINAL DE CARRERA

### *Projecte: Gestió Banc de Temps*

Desenvolupat en tecnologia Java JEE 5.

## MEMÒRIA

**Alumne:** Josep Lluís Fita López  
**Consultor:** Vicenç Font Sagrista

**Estudis:** Enginyeria Tècnica de Gestió  
**Curs:** 2010-2011 - 2n Semestre  
Juny 2011

Als meus familiars pel suport donat i paciència.

A la meva fillola Astrid, pels moments de felicitat donats.

## RESUM

Aquest document, correspon a la memòria del **Treball de Fi de Carrera (TFC)**, hi trobareu tota la documentació realitzada per l'anàlisi, disseny i implementació d'una aplicació informàtica que s'ha desenvolupat utilitzant la tecnologia **Java JEE 5**.

El projecte **TBank**, és una aplicació de gestió de temps, per a entitats que vulguin implementar un Banc de Temps. Està pensada per accedir-hi a través d'una intranet mitjançant un client *Standalone*. A grans trets, aquesta aplicació serveix per controlar i registrar els moviments dels serveis realitzats o demandats per part dels clients d'aquest tipus d'entitat sense ànim de lucre.

A més, aquest projecte té un repte afegit, el d'estudiar la tecnologia **JEE** (*Java Enterprise Edition*) i desenvolupar la nostra aplicació, tot això en un espai de temps curt. Això implica, que alhora de planificar el nostre projecte s'hagi de tenir en compte la corba d'aprenentatge.

**Nom de l'àrea del TFC:** J2EE

**Paraules clau:** *Java, JEE, Java Enterprise Edition, J2EE, EJB, Enterprise Java Beans, JPA, Jboss.*

**Llicència:** Per a la seva distribució, s'ha pensat en una llicència de **Reconeixement-No Comercial- Sense Obres derivades de Creative Commons**. Vol dir, que es pot copiar i distribuir públicament sense modificar-la sempre que es citi l'autor i l'obra, fer ús comercial. En aquesta direcció es pot consultar sobre la llicència. <http://es.creativecommons.org/licencia/>

## Índex de continguts

RESUM.....	3
ÍNDEX DE FIGURES.....	6
1. INTRODUCCIÓ.....	7
1.1. Justificació del TFC: punt de partida i aportació.....	7
1.2. Objectius del TFC.....	8
1.3. Enfocament i mètode seguit .....	8
1.4. Planificació del Projecte.....	9
1.5. Productes obtinguts.....	10
1.6. Descripció dels capítols de la memòria.....	10
2. DESCRIPCIÓ DEL PROJEKETE.....	11
2.1. Descripció.....	11
2.2. Composició del programari.....	12
2.3. Especificació de les funcionalitats.....	12
2.4. Entorn de seguretat.....	15
2.5. Funcionalitats proposades per a properes versions.....	15
2.6. Rols.....	15
3. ANÀLISI.....	16
3.1. Identificació dels actors.....	16
3.2. Diagrama de casos d'ús.....	17
3.3. Fitxes de casos d'ús.....	18
3.3.1. Fitxes de casos d'ús Subsistema Connexió.....	18
Descripció.....	18
Funcions implementades.....	18
Usuaris implicats.....	18
Cas d'us: Identificació.....	18
3.3.2. Fitxes de casos d'ús Subsistema Manteniment.....	19
Descripció.....	19
Funcions implementades.....	19
Usuaris implicats.....	19
Cas d'us: Gestió usuari del sistema.....	20
Cas d'us: Gestió àrees del coneixement.....	20
Cas d'us: Gestió compte client.....	21
3.3.3. Fitxes de casos d'ús Subsistema d'Intercanvi de serveis.....	22
Descripció.....	22

Funcions implementades.....	22
Usuaris implicats.....	22
Cas d'us: Gestió intercanvi de serveis.....	22
3.3.4. Fitxes de casos d'ús Subsistema de consultes general.....	23
Descripció.....	23
Funcions implementades.....	23
Usuaris implicats.....	23
Cas d'us: Gestió de consultes.....	23
3.4. Prototipus.....	24
3.4.1. Subsistema de connexió.....	24
3.4.2. Subsistema de manteniment.....	25
3.4.3. Subsistema d'intercanvi de serveis.....	26
3.4.4. Subsistema de consultes general.....	27
4. DISSENY.....	28
4.1. Fitxes CRC de les classes d'entitat.....	28
4.2. Diagrames UML.....	32
4.2.1. Diagrama de classes.....	32
4.2.2. Diagrames de col·laboració i seqüència.....	33
4.3. Persistència.....	35
4.3.1. Diagrama Entitat-Relació.....	35
4.3.2. Descripció dels atributs.....	36
4.4. Arquitectura de l'aplicació.....	37
5. CODIFICACIÓ.....	39
5.1. Decisions de disseny i codificació.....	39
5.2. Requeriments de software.....	40
5.3. Script de la Base de dades.....	40
5.4. Instal·lació de la Base de Dades.....	42
5.5. Instal·lació de Tbank en el servidor JBoss.....	42
5.6. Instal·lació de Tbank Client.....	44
6. CONCLUSIONS.....	46
GLOSSARI.....	47
BIBLIOGRAFIA.....	48
ANNEXES.....	49
Tour gràfic per Tbank Client.....	49

## ÍNDEX DE FIGURES

• Il·lustració 1: Logotip 'Tbank'.....	0
• Il·lustració 2: Calendari .....	8
• Il·lustració 3: Diagrama Gantt .....	9
• Il·lustració 4: Diagrama Casos d'ús .....	16
• Il·lustració 5: Pantalla de connexió .....	24
• Il·lustració 6: Pantalla de Gestió usuaris del sistema.....	25
• Il·lustració 7: Pantalla de Gestió àrea Coneixement.....	25
• Il·lustració 8: Pantalla de Gestió Compte client.....	26
• Il·lustració 9: Pantalla de Gestió de serveis.....	26
• Il·lustració 10: Pantalla de Consulta saldo.....	27
• Il·lustració 11: Diagrama de classes.....	32
• Il·lustració 12: Diagrama alta nou usuari del sistema.....	33
• Il·lustració 13: Diagrama alta nou compte client.....	33
• Il·lustració 14: Diagrama Inscripció a un servei.....	34
• Il·lustració 15: Diagrama Entitat-Relació.....	35
• Il·lustració 16: Diagrama de capes EJB.....	38
• Il·lustració 17: Pantalla servidor Jboss.....	43
• Il·lustració 18: Pantalla de Login Client.....	44
• Il·lustració 19: Pantalla principal TbankClient.....	45
• Il·lustració 20: Pantalla Login TbankClient.....	49
• Il·lustració 21: Pantalla Gestió Usuaris.....	49
• Il·lustració 22: Pantalla Gestió Àrea Coneixement.....	50
• Il·lustració 23: Pantalla Gestió Client.....	50
• Il·lustració 24: Pantalla Gestió Serveis.....	51
• Il·lustració 25: Pantalla Consulta Saldo.....	51

## 1. INTRODUCCIÓ

El meu **Treball Fi de Carrera** (TFC), consisteix en el desenvolupament d'una aplicació Java mitjançant la tecnologia J2EE. L'aplicació en si, és un programa de gestió de temps, pensat específicament per aquest tipus d'entitats.

### 1.1. Justificació del TFC: punt de partida i aportació

Actualment no treballa en el món de la informàtica, ja havia treballat temps enrere primer com a tècnic de *hardware* i després com a *webmaster*. Però actualment, per raons diverses, estic treballant en un altre tipus de feina que no té res a veure amb aquest món.

En vaig decidir per aquesta àrea, per que m'agrada molt la programació orientada a objectes mitjançant el llenguatge *Java*. Fa dos anys, vaig fer un curs de *C#*, té moltes similituds amb *Java*, però haig de dir que personalment en decanto per *Java*.

Haig de remarcar que desconeixia el ventall de tecnologies que ofereix *Java* amb la seva plataforma **J2EE**. Cosa que amb la realització d'aquest projecte, m'ha ajudat a ampliar una mica més els coneixements i penso que m'ajudarà de cares al futur. Per una banda a profunditzar més en aquesta tecnologia, i per l'altre a trobar feina relacionada amb el món de la informàtica.

Com ja he dit anteriorment, alhora de realitzar i planificar aquest projecte, he tingut en compte la meua corba d'aprenentatge sobre la tecnologia J2EE triada. A més, vaig buscar un projecte que alhora fos complet i a la vegada en permetés desenvolupar-lo en els terminis establerts. Per tant, vaig pensar que el projecte '**Tbank**', que com ja he esmentat, és un projecte per a la gestió d'un banc del temps, era una bona opció que s'ajustava al que estava cercant.

L'aportació d'aquest projecte, es proveir d'un software pensat per a la gestió d'un banc de temps, implementat per petits ajuntaments amb pocs recursos o associacions veïnals sense ànim de lucre, que vulguin oferir aquest servei als seus ciutadans.

## ***1.2. Objectius del TFC***

---

L'Objectiu principal del **Treball Fi de Carrera (TFC)**, es aprofundir en l'ús de la tecnologia *Java* a través de l'arquitectura de J2EE (*Java to Enterprise Edition*) en el qual s'ha de desenvolupar un projecte. En el nostre cas un programa de gestió per un banc de temps.

Per arribar a l'objectiu principal que ens marca la realització del **Treball Fi de Carrera (TFC)**, s'ha tingut que treballar amb rigorositat i sense demora en l'anàlisi, el disseny i la implementació de l'aplicació.

## ***1.3. Enfocament i mètode seguit***

---

La metodologia emprada per a la realització del Treball Fi de Carrera (TFC), ha sigut mitjançant el cicle de vida clàssic o cicle en cascada. Això vol dir, que per cada etapa que fem, s'ha d'haver fet la documentació que correspon a cada fase per poder anar avançant.

Les etapes que s'han seguit són les següents:

- Anàlisi prèvia
- Disseny
- Codificació
- Proves

A més, s'han hagut de realitzar algunes activitats paral·leles com:

1. Triar i avaluar la millor opció tecnològica que ens ofereix J2EE per al projecte.
2. Decidir l'entorn de desenvolupament, IDE, servidor d'aplicacions, gestor Base de Dades.
3. Formar-nos en l'opció tecnològica triada, en el nostre cas *Enterprise Java Beans* 3.



## 1.4. Planificació del Projecte

Per a la realització del TFC, que és l'aplicació de gestió d'un banc del temps, s'ha realitzat la següent planificació en relació amb el calendari lliurat per el consultor.

El calendari lliurat pel consultor és:

- PAC1 - Pla de treball: 4/3/2011 al 16/3/2011
- PAC2 - Disseny: 17/3/2011 al 14/4/2011
- PAC3 – Implementació: 15/4/2011 al 23/5/2011
- PAC4 – Memòria + Presentació: 24/5/2011 al 20/6/2011

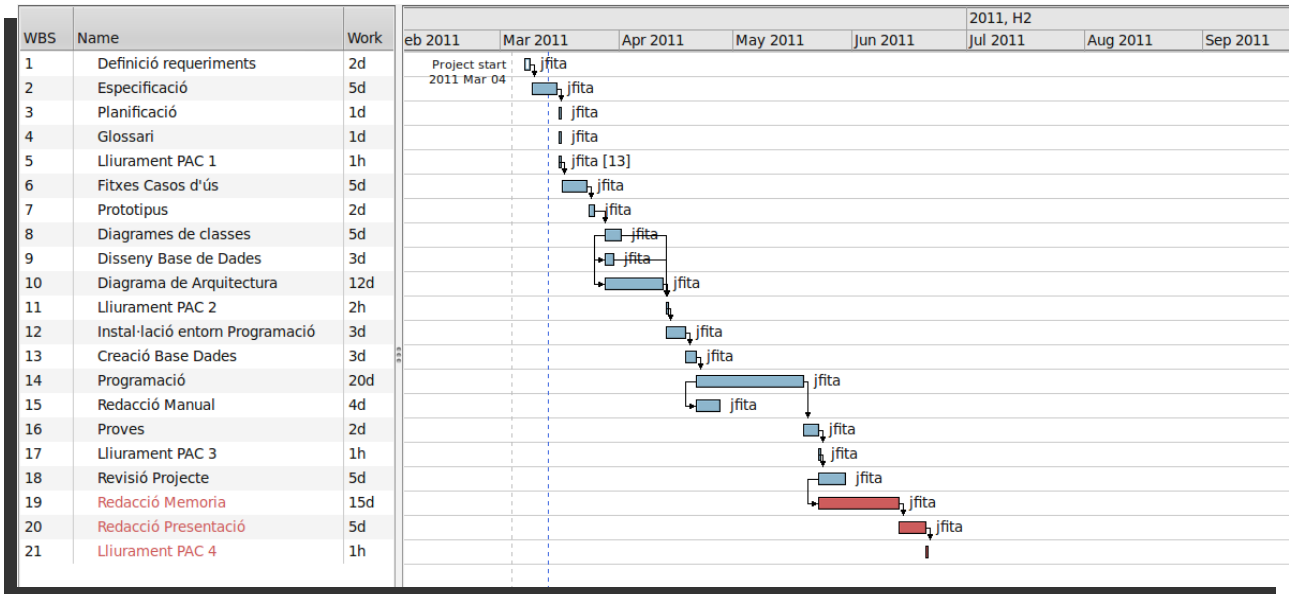
S'ha de tenir en compte que hi han factors que poden provocar retards en el lliurament de les fites, com és la falta d'experiència en aquest entorn de programació de J2EE o compromisos laborals.

Una vegada analitzat a fons el calendari, mostrem el quadre de tasques necessàries a realitzar per a l'elaboració del TFC.

Document	Fase	Tasca	Duració
<b>PAC 1</b>	Pla de Treball	Definició requeriments	2 dies
		Especificació	5 dies
		Planificació	1 dies
		Glosari	1 dia
<b>PAC 2</b>	Disseny	Realització de les fitxes de casos d'ús	5 dies
		Realizació prototipus	2 dies
		Diagrames de classes	5 dies
		Disseny Base Dades	3 dies
		Diagrama Arquitectura	12 dies
<b>PAC 3</b>	Implementació	Instalació Entorn de Treball	3 dies
		Creació de la base de dades	3 dies
		Programació	20 dies
		Proves	2 dies
		Redacció Manual	4 dies
<b>PAC 4</b>	Memòria + Presentació	Revisió del projecte	5 dies
		Redacció de la memòria	15 dies
		Realització presentació	5 dies

Il·lustració 2: Calendari

A continuació mostrem el **diagrama de GANTT** de tot el nostre projecte en el qual podem distingir les quatre fases.



Il·lustració 3: Diagrama Gantt

### 1.5. Productes obtinguts

Els productes obtinguts després de realitzar el Treball Fi de Carrera són:

- La memòria del projecte **TFC**.
- La presentació del projecte.
- L'aplicació de gestió de banc del temps '**Tbank**'.

### 1.6. Descripció dels capítols de la memòria

A continuació presentem els darrers capítols de que es compona la memòria:

- **Descripció del projecte:** En aquest capítol trobem la descripció i especificació de l'aplicació.
- **Anàlisi:** En aquest capítol trobem la descripció dels actors i els principals casos d'ús.
- **Disseny:** En aquest capítol trobem les fitxes CRC de les classes entitat, els diagrames UML, el disseny de la base de dades i l'arquitectura de l'aplicació.
- **Codificació:** En aquest capítol podem veure les modificacions fetes durant la implementació. Així com del software que es requereix.
- **Conclusions:** En aquest capítol parlem sobre les conclusions a que s'arriba un cop realitzat la part de codificació.

## 2. DESCRIPCIÓ DEL PROJEKETE

A continuació es dona una descripció introductoria del programari que es vol desenvolupar per al TFC.

### 2.1. Descripció

---

Un banc del temps emula el funcionament d'un banc, però allò que el fa moure no són els diners o els beneficis, sinó que és el temps dedicat als serveis i el temps dedicat a les persones.

En altres paraules, un usuari d'aquest tipus d'entitat, el que busca és un servei que li fa falta, per exemple 1 hora de classe d'Anglès a canvi de donar ell un servei, per exemple treure el gos a passejar, a una altre persona que ho requereixi. Cada servei donat o rebut val 1 hora que va a compte de l'usuari, si dona el servei guanya una hora i si el rep se li resta una hora del seu compte particular.

Normalment aquest tipus de bancs són impulsats pels ajuntaments amb l'ajuda de les xarxes associatives, per que puguin resoldre problemes de la vida quotidiana mitjançant l'intercanvi de temps sense ànim de lucre.

Alguns ajuntaments, sobretot els de les grans ciutats, disposen a dia d'avui dels suficients recursos monetaris per encarregar a qualsevol empresa consultora un estudi del software que necessiten per que li facin el programari que necessiten per aquesta finalitat a mida. Tot el contrari dels ajuntaments d'entorns d'àmbit rural, que no disposen de tants recursos monetaris, la majoria dels quals quan volen portar a terme una iniciativa de certa volada ho han de fer a través de subvencions, ja siguin concedides per les entitats regionals, estatals o supranacionals. En un entorn de crisi sistèmica com la que estem patint a dia d'avui, fa que aquest ens públics retallin subvencions. Per tant tota iniciativa que es vulgui portar a terme comptant amb els diners públics en aquests àmbits rurals queda tallada, i per tant només arriben a veure la llum aquelles que la gent participa voluntàriament.

En el nostre cas un banc del temps, és una iniciativa que de mica en mica s'anirà estenen pel territori i que ajudarà a la gent a resoldre petits problemes de la vida diària per vies alternatives. Els ajuntaments d'àmbit rural i amb pocs recursos si volen implementar aquest servei, s'ho hauran de gestionar tan bonament com poden, fulles d'excel, ect. Ja que no disposaran de les subvencions per cobrir les despeses per comprar o encarregar el programari de gestió per aquest afí.

La finalitat del projecte de gestió de banc de temps, es proveir als ajuntaments o entitats associatives sense ànim de lucre que ho desitgin de poder implementar un banc de temps amb un software gratuït per a la gestió d'aquest, com un servei més als seus ciutadans.

S'ha de pensar que la majoria de ciutadans d'aquest pobles són gent gran que no tenen o han tingut gaire contacte amb les TIC.

L'usuari del nostre software, és una persona que domina mínimament les TIC, com pot ser la secretaria de l'ajuntament o persona afí per aquesta tasca. Per tant, l'interfície del nostre programari serà dissenyada de forma clara i concisa amb una navegació bastant intuïtiva.

La nostra aplicació serà de tipus distribuïda, desplegada en un servidor d'aplicacions que inclougui un contenidor EJB i qualsevol que tingui accés i els permisos corresponents podrà entrar i fer anar l'aplicació des de el client, que serà l'interfície de l'usuari. S'ha pensat per al seu desenvolupament en els estàndars que hi han en el mercat i que són de llicència lliure, com és la tecnologia J2EE mitjançant els Enterprise Java Beans (EJB), en la seva versió 3.

El programari el compondran quatre subsistemes:

- Subsistema de connexió.
- Subsistema de manteniment
- Subsistema d'intercanvi de serveis.
- Subsistema de consultes general.

## ***2.2. Composició del programari***

---

L'aplicació es desenvoluparà mitjançant la programació Orientada a Objectes, fet que facilitarà que l'aplicació sigui extensible i, per tant, pugui créixer segons futures necessitats. El disseny de l'aplicació es farà tan modular com sigui possible per facilitar aquest creixement. A més la programació Orientada a Objectes permet la flexibilitat i re-usabilitat. El llenguatge a emprar per la implementació serà Java, ja que és tracta d'una aplicació Java EE (Java Enterprise Edition) el qual el servidor d'aplicacions pot assegurar una bona escalabilitat, seguretat, ect.

## ***2.3. Especificació de les funcionalitats***

---

### **2.3.1. - Subsistema connexió.**

El subsistema de connexió, és l'encarregat de validar l'usuari en el sistema quan aquest accedeixi al programari mitjançant el client de l'aplicació (client swing). Aquest li demanarà el ID d'usuari i contrasenya mitjançant la pantalla d'accés.

L'usuari l'introdueix el ID d'usuari, la contrasenya, i es valida en el sistema, si és correcte accedirà a la pantalla principal de l'aplicació, segons el seu perfil.

En cas contrari, l'aplicació mostra un missatge d'error i tornarà a demanar la validació d'aquest on haurà d'introduir correctament el nom d'usuari i contrasenya a la pantalla d'accés.

També permet el canvi de sessió d'usuari, que correspon al tancament de sessió d'aquest i per tant, torna a demanar el ID i la contrasenya.

Per últim direm que aquest subsistema permet tancar l'aplicació client a l'usuari.

- **2.3.2. - Subsistema de manteniment.**

El subsistema de manteniment, és on l'usuari administrador i l'usuari secretari tenen accés a les opcions d'administrar usuaris, administrar noves àrees de coneixement i administrar els comptes dels clients.

- 1. Gestió usuari del sistema.**

Per aquestes opcions, només els usuaris amb rol d'administrador, poden donar d'alta o baixa els usuaris del sistema. A més de poder modificar els existents si es necessari. Això sí després d'emplenar el corresponent formulari.

- 2. Gestió àrees de coneixement.**

Per aquestes opcions, els usuaris amb rol administrador i amb rol secretari poden accedir i donar d'alta o baixa les àrees de coneixement en la qual es classifiquen els tipus d'intercanvis. A més de poder modificar les existents si es dona el cas.

- 3. Gestió Compte d'usuari del Banc.**

Per aquestes opcions, els usuaris amb rol administrador i amb rol secretari poden accedir i donar d'alta o baixa els Comptes dels usuaris del Banc del temps. A més de poder modificar les existents si es dona el cas. S'ha de pensar que quan es dona d'alta un usuari del banc, l'usuari del sistema ha d'emplenar un formulari, en el qual es demanen les dades personals, així com l'àrea de servei que vol fer l'usuari del banc. Un cop s'ha premut el botó acceptar, es crea el compte d'usuari amb el saldo inicial d'una hora si aquest no existeix a la base de dades.

- **2.3.3. - Subsistema d'intercanvi de serveis.**

El subsistema d'intercanvis de serveis, és on l'usuari administrador i l'usuari secretari tenen accés a les opcions d'administrar els serveis que ofereixen els usuaris del banc del temps.

- 1. Llistat de Serveis.**

L'usuari Administrador o L'usuari secretari com a usuaris del sistema, quan seleccionen aquesta opció, accedeixen a la pantalla de llistat de serveis on tenen un *combobox* per seleccionar l'àrea de servei que volem llistar els usuaris que donen un servei en aquesta àrea. Un cop han seleccionat l'àrea de coneixement, prenen el botó acceptar i el sistema mostrarà el llistat de clients que presten servei en una àrea determinada per pantalla, a més mostrarà les dades de contacte de cada usuari.

- 2. Gestió Servei.**

L'usuari del sistema (Administrador/secretari), entraran a la pantalla de confirmar servei, en la qual ha d'emplenar un formulari amb les dades corresponents on demana el codi client (nif) que dona el servei, com el codi client (nif), que rep el servei. Si el client que ha de rebre el servei té saldo 0, a llavors no es podrà confirmar el servei.

L'usuari del sistema (Administrador/secretari), entraran a la pantalla de Gestió servei, quan hagin de canviar les dades del servei o donar-lo de baixa del sistema.

### **3. Pagament Servei.**

L'usuari del sistema (Administrador/secretari), entraran a la pantalla de pagament de servei, un cop que s'hagi realitzat el servei. El sistema demanarà el codi client que ha fet el servei i el codi de client que l'ha rebut un cop introduïdes les dades i confirmades, el sistema procedirà a augmentar el saldo de l'usuari que ha fet el servei i restar-lo del que l'ha rebut. A més guardarà el servei a l'històric de serveis, i el mostrarà per pantalla junt amb altres serveis realitzats.

- **2.3.4. - Subsistema de consultes general.**

El subsistema de consultes generals, és on l'usuari administrador i l'usuari secretari tenen accés a les opcions de consulta de les dades generals que té el sistema emmagatzemades a la base de dades del banc del temps.

#### **1. Consulta Saldo**

L'usuari del sistema (Administrador/secretari), pot consultar el saldo de temps d'un client qualsevol entrant a la pantalla de consulta saldo. Aquesta li demanarà el codi i un cop introduït el sistema li mostrarà el saldo d'hores per pantalla.

#### **2. Consulta Serveis fets**

L'usuari del sistema (Administrador/secretari), pot consultar el saldo de temps d'un client qualsevol entrant a la pantalla de consulta saldo. Aquesta li demanarà el codi i un cop introduït el sistema li mostrarà els serveis fets per pantalla.

#### **3. Consulta Serveis rebuts**

L'usuari del sistema (Administrador/secretari), pot consultar el saldo de temps d'un client qualsevol entrant a la pantalla de consulta saldo. Aquesta li demanarà el codi i un cop introduït el sistema li mostrarà els serveis rebuts per pantalla.

#### **4. Consulta històric de serveis**

L'usuari del sistema (Administrador/secretari), pot consultar l'històric de serveis simplement entrant en aquesta pantalla, l'usuari pot consultar els serveis realitzats per tipus de servei.

## ***2.4. Entorn de seguretat***

---

Els usuaris hauran de validar-se en el programari, i depenent del rol que tinguin assignat accediran a unes funcions o unes altres del programari.

## ***2.5. Funcionalitats proposades per a properes versions***

---

- **Valoració usuaris:** S'ha pensat per a properes versions que els usuaris del banc del temps puguin tenir l'opció de valorar el servei rebut, Mitjançant una escala de puntuacions del 1 al 10.
- **Consulta nif:** S'ha pensat per a properes versions que el subsistema de consultes generals disposi d'una cerca de nif d'usuari del banc del temps. Així que aquesta opció demanaria el cognom i donaria un llistat de nif i noms corresponents per pantalla.
- **Visualització del moviments de compte:** S'ha pensat per a la propera versió, la inclusió en l'apartat de consulta de saldo, la visualització dels moviments realitzats en el compte com a millora.

## ***2.6. Rols***

---

La nostra aplicació té definits 2 rols d'usuari, que seran els que interactuaran amb el sistema.

- **Administrador:** És el rol d'usuari que té tot els privilegis del sistema, té accés a totes les funcionalitats del sistema. El seu paper es la de vetllar pel correcte funcionament del sistema, en altres paraules administrar de forma eficient els recursos del sistema.
- **Secretari:** És el rol d'usuari que té accés a les funcionalitats del sistema, per tant interactuarà amb el sistema per tal d'administrar els recursos del banc del temps amb la màxima eficiència. Té accés a pràcticament totes les funcionalitats del sistema menys l'administració d'usuaris del sistema.

### 3. ANÀLISI

#### 3.1. Identificació dels actors

---

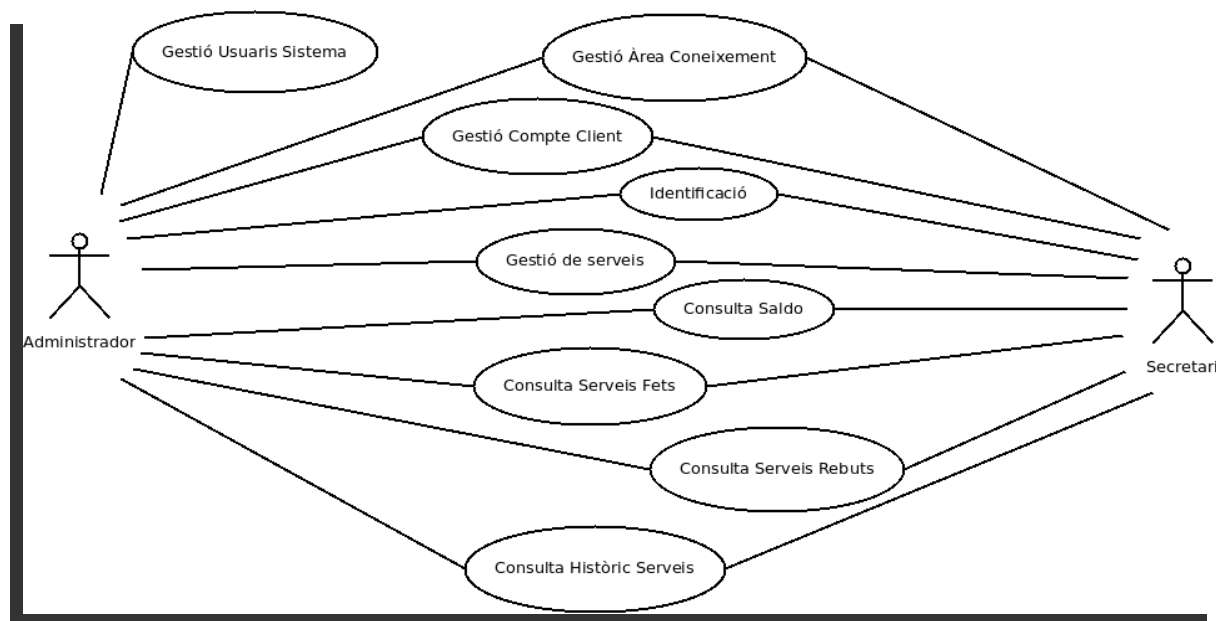
El programari de gestió banc de temps '**Tbank**', s'identifiquen dos actors que interactuen amb el sistema.

- **Administrador:** És la persona que vetlla pel correcte funcionament del sistema. Possiblement serà el tècnic informàtic de l'empresa de manteniment que tingui contractat l'ajuntament. La seva tasca consisteix en gestionar els usuaris del sistema, (alta, baixa, modificar usuaris). A més, també poden realitzar les tasques de secretaria arribat el moment, com és la gestió de comptes dels clients, la gestió dels serveis dels clients. Així com els llistats de consulta que disposa el sistema, com és consulta a l'històric de serveis, consulta de saldo dels clients, ect...
- **Secretari:** És la persona 'usuària' del sistema. Possiblement serà la secretaria o persona responsable de la gestió del banc del temps. La seva tasca consisteix en gestionar els comptes bancaris dels clients, així com la gestió dels serveis i realització de consultes (saldo, històric...).



### 3.2. Diagrama de casos d'ús

A continuació mostrem el diagrama de casos d'ús del programari de gestió banc de temps 'Tbank' per a una millor comprensió de la interacció dels actors amb el sistema.



Il·lustració 4: Diagrama Casos d'Ús

### 3.3. Fitxes de casos d'ús

---

A continuació mostrem les fitxes de casos d'ús principals del programari de gestió banc de temps 'Tbank' de cada subsistema.

#### 3.3.1. Fitxes de casos d'ús Subsistema Connexió

---

##### Descripció

---

Aquest és el subsistema que permet accedir a l'aplicació de gestió banc de temps 'TBank' a qualsevol persona que en sigui usuària.

##### Funcions implementades

---

###### 1.- Identificació

- Entrar en el sistema 'TBank'.

##### Usuaris implicats

---

- Tots els usuaris del sistema 'TBank'.

#### CAS D'US: IDENTIFICACIÓ

Funcionalitat:	Identifica l'actor que pretén interactuar amb el sistema 'TBank'.
Descripció:	El sistema 'TBank' mostra una pantalla demanant l'identificador d'usuari i la seva contrasenya. L'usuari introdueix aquestes dades, i si el sistema troba aquest usuari a la base de dades comprova que la contrasenya introduïda és la que li correspon. A continuació el sistema mostra per pantalla una interfície gràfica que permet l'usuari accedir a les funcionalitats que aquest especifica pel seu tipus d'usuari.
Actors:	Tots els usuaris que tinguin dret d'accés al sistema 'TBank'.
Paper de l'usuari:	És una condició necessària per poder entrar i fer servir el sistema.
Casos d'ús relacionats:	Cap.
Precondició:	L'actor existeix com a usuari a la BBDD i està identificat al sistema 'TBank'.
Postcondició:	L'usuari està identificat i pot accedir a totes les funcionalitats que el sistema proporciona al tipus d'usuari al qual pertany. En cas d'haver-se produït un error en la identificació el sistema 'TBank' visualitza una pantalla que informa d'aquesta circumstància.
Altres comentaris:	

### ***3.3.2. Fitxes de casos d'ús Subsistema Manteniment***

---

#### ***Descripció***

---

Aquest és el subsistema que permet gestionar els usuaris del sistema '**TBank**' a qualsevol persona que en sigui administrador. A més, aquest subsistema permet la gestió de les àrees de coneixement per part de les persones usuàries que tinguin perfil administrador o secretaria. Aquest subsistema inclou també la gestió dels comptes dels clients del banc del temps per part de les persones amb perfil administrador o secretari.

#### ***Funcions implementades***

---

##### 1.- Gestió usuaris del sistema.

- Crear un nou usuari del sistema.
- Cercar usuari del sistema
- Modificar usuari del sistema.
- Borrar usuari del sistema.

##### 2.- Gestió àrees del coneixement.

- Crear nova àrea de coneixement.
- Cercar nova àrea de coneixement.
- Modificar àrea de coneixement.
- Borrar àrea de coneixement.

##### 3.- Gestió compte client.

- Crear nou compte client.
- Cercar nou compte client.
- Modificar compte client.
- Borrar compte client.

#### ***Usuaris implicats***

---

##### 1.- Gestió usuaris del sistema.

- Tots els usuaris del sistema 'Tbank' amb perfil administrador.

##### 2.- Gestió àrees del coneixement.

- Tots els usuaris del sistema 'Tbank' amb perfil administrador o secretari.

##### 3.- Gestió compte client.

- Tots els usuaris del sistema 'Tbank' amb perfil administrador o secretari.

**CAS D'US: GESTIÓ USUARI DEL SISTEMA**

<b>Funcionalitat:</b>	Gestiona(alta / Cercar/ modifica / baixa) els usuaris del sistema 'Tbank'.
<b>Descripció:</b>	El sistema 'TBank' demana el codi únic d'usuari (DNI), les dades corresponents d'aquest, així com un password i el tipus d'usuari. L'actor introdueix les dades i les valida. El sistema consulta l'existència de l'usuari, si aquest no existeix finalitza el cas d'ús.
<b>Actors:</b>	Tots els usuaris que tinguin perfil administrador.
<b>Paper de l'usuari:</b>	Una de les funcionalitats que pot fer l'administrador.
<b>Casos d'ús relacionats:</b>	Identificació. Consulta usuari.
<b>Precondició:</b>	L'actor s'ha identificat al sistema i és de tipus Administrador. L'usuari no existeix.
<b>Postcondició:</b>	L'usuari existeix i consta com a donat d'alta en el sistema.
<b>Altres comentaris:</b>	En aquest cas d'ús només s'ha explicat el procés d'alta d'un nou usuari, per als altres processos de gestió el funcionament és el mateix.

**CAS D'US: GESTIÓ ÀREES DEL CONEIXEMENT**

<b>Funcionalitat:</b>	Gestiona(alta / Cercar/ modifica / baixa) les àrees del coneixement del sistema 'Tbank'.
<b>Descripció:</b>	El sistema 'TBank' demana el codi únic d'àrea. Les dades corresponents d'aquesta. L'actor introdueix les dades i les valida. El sistema consulta l'existència de l'àrea, si aquesta no existeix finalitza el cas d'ús.
<b>Actors:</b>	Tots els usuaris que tinguin perfil administrador o secretari.
<b>Paper de l'usuari:</b>	Una de les funcionalitats que pot fer l'administrador o el secretari.
<b>Casos d'ús relacionats:</b>	Identificació. Consulta àrea.
<b>Precondició:</b>	L'actor s'ha identificat al sistema i és de tipus Administrador o secretari. L'àrea de coneixement no existeix.
<b>Postcondició:</b>	L'àrea de coneixement existeix i consta com a donat d'alta en el sistema.
<b>Altres comentaris:</b>	En aquest cas d'ús només s'ha explicat el procés d'alta d'una nova àrea de coneixement, per als altres processos de gestió el funcionament és el mateix.

**CAS D'US: GESTIÓ COMPTE CLIENT**

<b>Funcionalitat:</b>	Gestiona(alta / Cercar/ modifica / baixa) els comptes d'usuari-client del sistema 'Tbank'.
<b>Descripció:</b>	El sistema 'TBank' demana el codi únic de compte (NIF). A més demana les dades corresponents del compte client. L'actor introdueix les dades i les valida. El sistema consulta l'existència del compte, si aquest no existeix finalitza el cas d'ús.
<b>Actors:</b>	Tots els usuaris que tinguin perfil administrador o secretari.
<b>Paper de l'usuari:</b>	Una de les funcionalitats que pot fer l'administrador o el secretari.
<b>Casos d'ús relacionats:</b>	Identificació. Consulta compte client.
<b>Precondició:</b>	L'actor s'ha identificat al sistema i és de tipus Administrador o secretari. El compte client no existeix.
<b>Postcondició:</b>	El compte client consta com a donat d'alta en el sistema amb saldo d'una hora.
<b>Altres comentaris:</b>	En aquest cas d'ús només s'ha explicat el procés d'alta d'un nou compte client, per als altres processos de gestió el funcionament és el mateix.

### 3.3.3. Fitxes de casos d'ús Subsistema d'Intercanvi de serveis

#### *Descripció*

Aquest és el subsistema que permet gestionar els serveis disponibles en el sistema que ofereixen els clients del banc de temps 'Tbank', per part de les persones amb perfil administrador o secretari.

#### *Funcions implementades*

1.- Gestió intercanvi de serveis.

- Llistar ofertes de serveis disponibles.
- Inscripció a un servei (confirmar servei).
- Cercar inscripció de servei.
- Modificar inscripció de servei.
- Borrar inscripció de servei.
- Pagament servei.

#### *Usuaris implicats*

1.- Gestió intercanvi de serveis.

- Tots els usuaris del sistema 'Tbank' amb perfil administrador o secretari.

#### **CAS D'US: GESTIÓ INTERCANVI DE SERVEIS**

<b>Funcionalitat:</b>	Gestiona(Llistar/ Inscripció/ Cercar/ Modificar/ Borrar/ Pagament) els serveis ofertats del sistema 'Tbank'.
<b>Descripció:</b>	El sistema 'TBank' demana el codi únic de compte client (nif) que oferta el servei, el codi de compte client (nif) que demanda el servei, així com les dades corresponents al servei. L'actor les entra i valida. El sistema consulta que existeixin els comptes clients, si tot va bé a llavors mira si existeix el servei donat d'alta en el sistema, si no hi ha cap problema el sistema mira que el saldo del client demandant no sigui 0. si tot va bé és finalitza el cas d'ús.
<b>Actors:</b>	Tots els usuaris que tinguin perfil administrador o secretari.
<b>Paper de l'usuari:</b>	Una de les funcionalitats que pot fer l'administrador o secretari.
<b>Casos d'ús relacionats:</b>	Identificació. Consulta usuari. Consulta servei.
<b>Precondició:</b>	L'actor s'ha identificat al sistema i és de tipus Administrador o secretari. L'inscripció de servei no existeix.
<b>Postcondició:</b>	L'inscripció existeix i consta com a donat d'alta en el sistema.
<b>Altres comentaris:</b>	En aquest cas d'ús només s'ha explicat el procés d'inscripció a un servei per part d'un client, per als altres processos de gestió el funcionament és similar.

### 3.3.4. Fitxes de casos d'ús Subsistema de consultes general

#### Descripció

Aquest és el subsistema que permet gestionar les diverses consultes de que disposa el sistema 'Tbank', per part de les persones amb perfil administrador o secretari.

#### Funcions implementades

1.- Gestió de consultes.

- Consulta de saldo.
- Consulta de serveis fets i rebuts.
- Consulta de serveis.
- Consulta històric.

#### Usuaris implicats

1.- Gestió de consultes.

- Tots els usuaris del sistema 'Tbank' amb perfil administrador o secretari.

#### CAS D'US: GESTIÓ DE CONSULTES

Funcionalitat:	Gestiona(Saldo/ serveis fets/ serveis rebuts/ històric) les consultes al sistema 'Tbank'.
Descripció:	El sistema 'TBank' demana el codi únic de compte client (nif) que es vol consultar el saldo. L'actor introdueix les dades i les valida. El sistema consulta l'existència del compte client, si aquest no existeix el sistema mostra un missatge a l'actor que el compte client introduït no existeix, si el compte client existeix, a llavors el sistema mostra les dades per pantalla i finalitza el cas d'ús.
Actors:	Tots els usuaris que tinguin perfil administrador o secretari.
Paper de l'usuari:	Una de les funcionalitats que pot fer l'administrador o secretari.
Casos d'ús relacionats:	Identificació. Consulta usuari.
Precondició:	L'actor s'ha identificat al sistema i és de tipus Administrador o secretari. El compte usuari-client existeix.
Postcondició:	Les dades del compte client + el saldo es mostrat per pantalla.
Altres comentaris:	En aquest cas d'ús nomès s'ha explicat el proces de consulta de saldo per part d'un client, per als altres processos de gestió el funcionament és similar.

### 3.4. Prototipus

---

El que volem mostrar és un prototipus de les pantalles del programari de gestió banc de temps 'Tbank', en principi no es veurà una presentació exhaustiva de totes les pantalles de l'aplicació. Sinó que veurem un aproximació visual de les pantalles de l'aplicació.

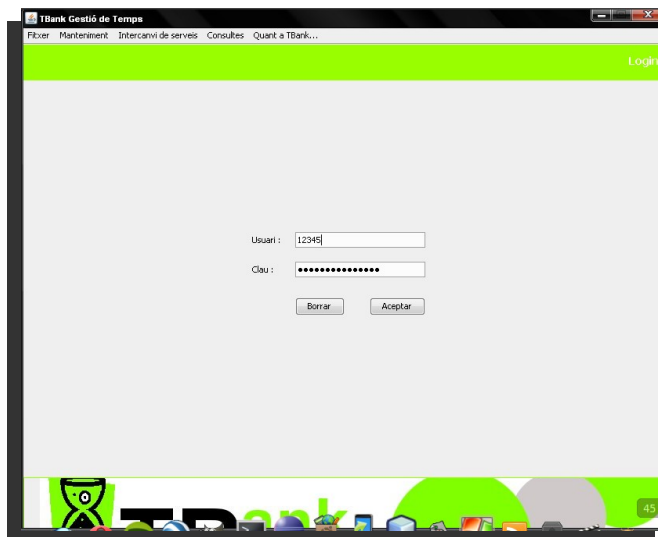
S'ha pensat en una interfície d'usuari senzilla i sobretot intuïtiva, per tal de facilitar la navegació i la corba d'aprenentatge del funcionament del nostre programari al futur usuari del sistema. Les pantalles poden sofrir alguna modificació en l'etapa de desenvolupament.

A continuació mostrem les principals pantalles de que es compona la nostra aplicació client.

#### 3.4.1. Subsistema de connexió

---

- **Pantalla de Connexió:**

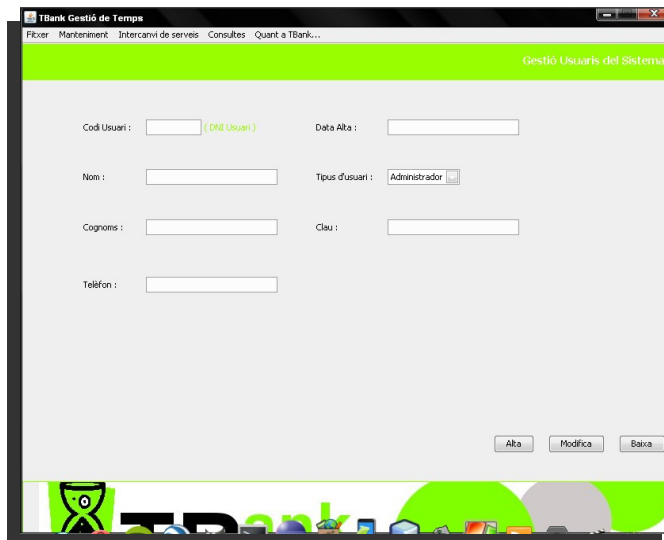


Il·lustració 5: Pantalla de Connexió



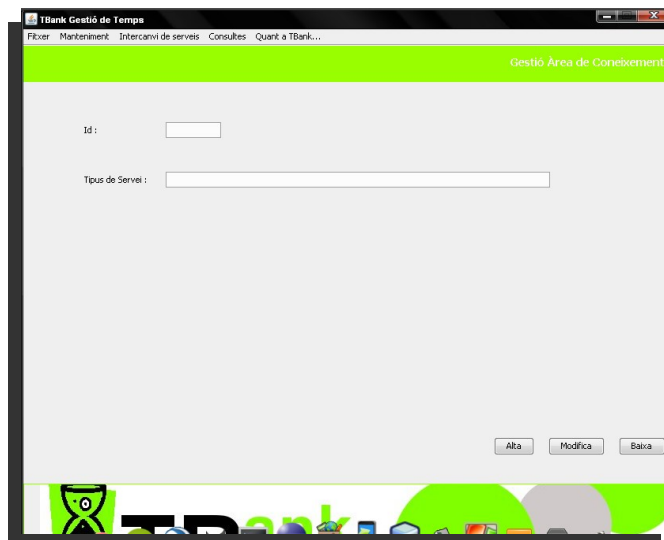
### 3.4.2. Subsistema de manteniment

- **Pantalla de Gestió usuaris del sistema:**



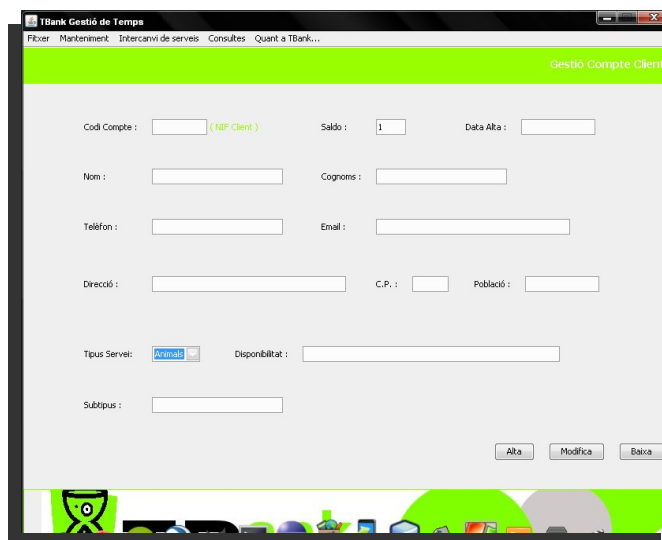
Il·lustració 6: Pantalla de Gestió usuaris del sistema

- **Pantalla de Gestió àrea de coneixement:**



Il·lustració 7: Pantalla de Gestió àrea coneixement

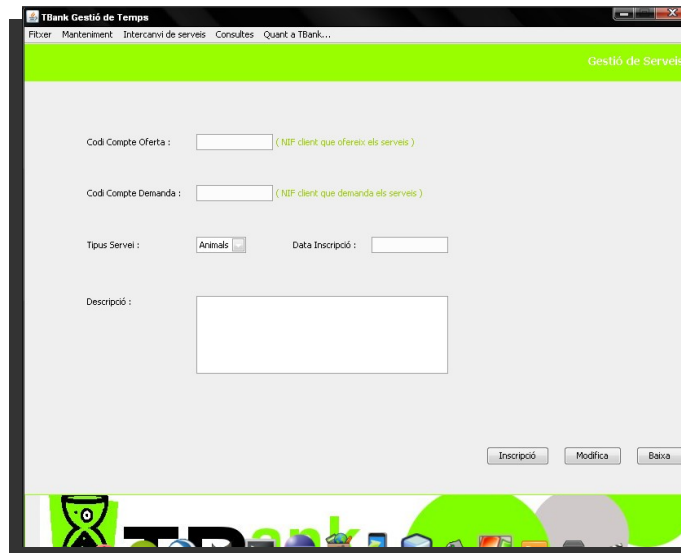
- **Pantalla de Gestió compte client:**



II-lustració 8: Pantalla de Gestió compte client

### 3.4.3. Subsistema d'intercanvi de serveis

- **Pantalla de Gestió de serveis:**

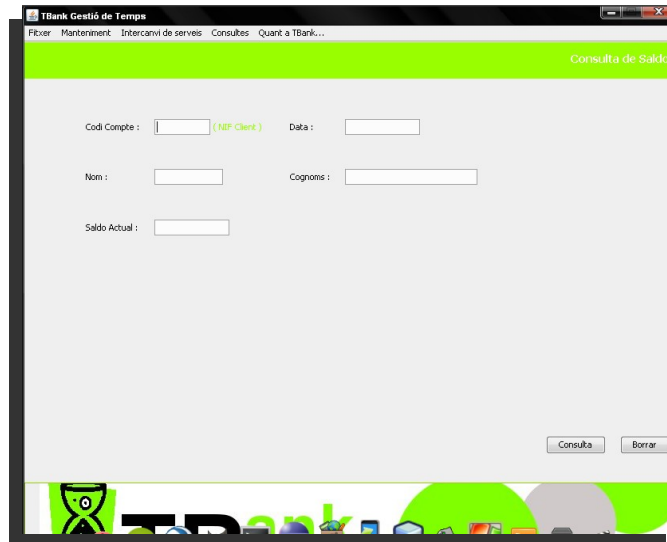


II-lustració 9: Pantalla de Gestió de serveis

### 3.4.4. Subsistema de consultes general

---

- **Pantalla de consulta de saldo:**



Il·lustració 10: Pantalla de consulta de saldo

## 4. DISSENY

### 4.1. Fitxes CRC de les classes d'entitat

A continuació mostrem les fitxes CRC de les principals classes d'entitat que s'utilitzen a l'aplicació.

Classe	Persona
Descripció de la classe	Atributs de totes les persones que intervenen a l'aplicació ( Usuaris i Clients)
Tipus de classe	Classe principal
Característiques de la classe	Abstracta
Responsabilitats	Col·laboracions
Modela un objecte Persona amb els seus atributs	
Constructors	
+Persona() +Persona(nom, cognoms, telèfon)	
Atributs	
-nom: String -cognoms: String -telèfon: Integer	
Mètodes	
+getNom() +setNom(String nom) +getCognoms() +setCognoms(string cognoms) +getTelefon() +setTelefon(Integer telefon)	

Classe	Usuari (hereva de persona)
Descripció de la classe	Identificació dels usuaris de l'aplicació
Tipus de classe	Classe principal
Característiques de la classe	Concreta, composta, persistent
Responsabilitats	Col·laboracions
Gestiona els usuaris(crea, modifica, borra, consulta).	
Constructors	
+Usuari() +Usuari(dni, clau) +Usuari(persona: objecte persona, dni, clau, dataAlta)	
Atributs	
-clau: String -dataAlta: Date -dni: Integer	

Mètodes
+getClau() +setClau(String clau) +getDataAlta() +setDataAlta(Date dataAlta) +getDni() +setDni(Integer dni)

Classe	Secretari (hereva de Usuari)*
Descripció de la classe	Identificació dels usuaris Secretaris
Tipus de classe	Classe principal
Característiques de la classe	Concreta,persistent
Responsabilitats	Col·laboracions
Identifica els usuaris secretaris	
Constructors	
Atributs	
Mètodes	

Classe	Administrador (hereva de Usuari)*
Descripció de la classe	Identificació dels usuaris Administradors
Tipus de classe	Classe principal
Característiques de la classe	Concreta,persistent
Responsabilitats	Col·laboracions
Identifica els usuaris administradors	
Constructors	
Atributs	
Mètodes	

\*. **Les classes administrador i Secretari** desapareixeràn ja que no tenen cap atribut. Per tant, la classe Usuari tindrà un nou atribut 'tipusUsuari', on indicarem si els objectes d'aquesta classe són del tipus de la subclasse (Administrador o Secretari).

<b>Classe</b>	<b>Client (hereva de persona)</b>
Descripció de la classe	Identificació dels clients de l'aplicació
Tipus de classe	Classe principal
Característiques de la classe	Concreta, composta, persistent
<b>Responsabilitats</b>	<b>Col·laboracions</b>
Gestiona els clients (crea, modifica, borra, consulta).	Compte, EnumServei
<b>Constructors</b>	
+Client() +Client(nif) +Client( persona: objecte persona, dataAlta, nif, saldo, tipuServei) +Client(persona: objecte persona, cp, dataAlta, direccio, disponibilitat, email, nif, saldo, subtipus, poblacio, tipuServei)	
<b>Atributs</b>	
-cp: Integer -dataAlta: Date -direccio: String -disponibilitat: String -email: String -nif: String -poblacio: String -saldo: Compte -subtipus: String -tipuServei: EnumServei	
<b>Mètodes</b>	
+getCp() +setCp(Integer cp) +getDataAlta() +setDataAlta(Date dataAlta) +getDireccio() +setDireccio(String direccio) +getDisponibilitat() +setDisponibilitat(String disponibilitat) +getEmail() +setEmail(String email) +getNif() +setNif(String nif) +getPoblacio() +setPoblacio(String poblacio) +getSaldo() +setSaldo(Objecte Compte) +getSubtipus() +setSubtipus(String subtipus) +getTipuServei() +setTipuServei(Objecte EnumServei)	

Classe	Compte
Descripció de la classe	Classe que representa a un compte
Tipus de classe	Classe principal
Característiques de la classe	Concreta
Responsabilitats	Col·laboracions
Mantenir el saldo	Client
Constructors	
+Compte(saldo)	
Atributs	
-saldo: Integer	
Mètodes	
+getSaldo() +setSaldo(Integer saldo)	

\***La classe compte**, no serà persistent, ja que el seu atribut passarà a la classe client, ja que en aquest tipus d'entitat un client només pot tenir un compte. Per tant un client no tindrà múltiples comptes.

Classe	Servei
Descripció de la classe	Classe que representa a un servei
Tipus de classe	Classe principal
Característiques de la classe	Concreta, composta, persistent
Responsabilitats	Col·laboracions
Gestiona els serveis(crea, modifica, borra, consulta).	Client, EnumServei
Constructors	
+Servei() +Servei(oferta, demanda) +Servei(oferta, demanda, dataInscripcio, descripcio, tipuServei)	
Atributs	
-oferta: Objecte Client -dataInscripcio: Date -demanda: Objecte Client -descripcio: String -tipuServei(EnumServei)	
Mètodes	
+getOferta() +setOferta(Objecte Client) +getDataInscripcio() +setDataInscripcio(Date dataInscripcio) +getDemanda() +setDemanada(Objecte Client) +getDescripcio() +setDescripcio(String descripcio) +getTipuServei() +setTipuServei(Objecte EnumServei)	

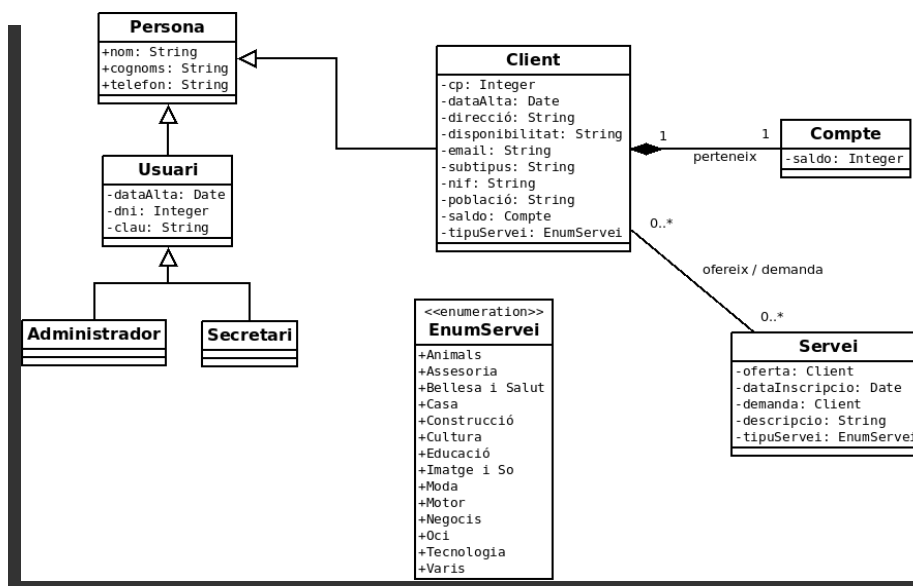
Classe	EnumServei
Descripció de la classe	Classe que representa a un tipus de servei
Tipus de classe	Enumeration
Característiques de la classe	Concreta, persistent
Responsabilitats	Col·laboracions
Gestionar el tipus de serveis	Client, servei
Constructors	
+EnumServei(String tipusServei) +EnumServei(String tipusServei, Integer Id)	
Atributs	
+tipusServei: String -Id: Integer	
Mètodes	
+getTipusServei() +setTipusServei(String tipusServei) +getId() +setTId(Integer id)	

## 4.2. Diagrames UML

A continuació mostrem el disseny de l'aplicació 'Tbank' mitjançant els principals diagrames UML, com els diagrames de classes, col·laboració i seqüència més significatius.

### 4.2.1. Diagrama de classes

Diagrama de classes on es mostren les classes d'entitat més rellevants amb els seus atributs.



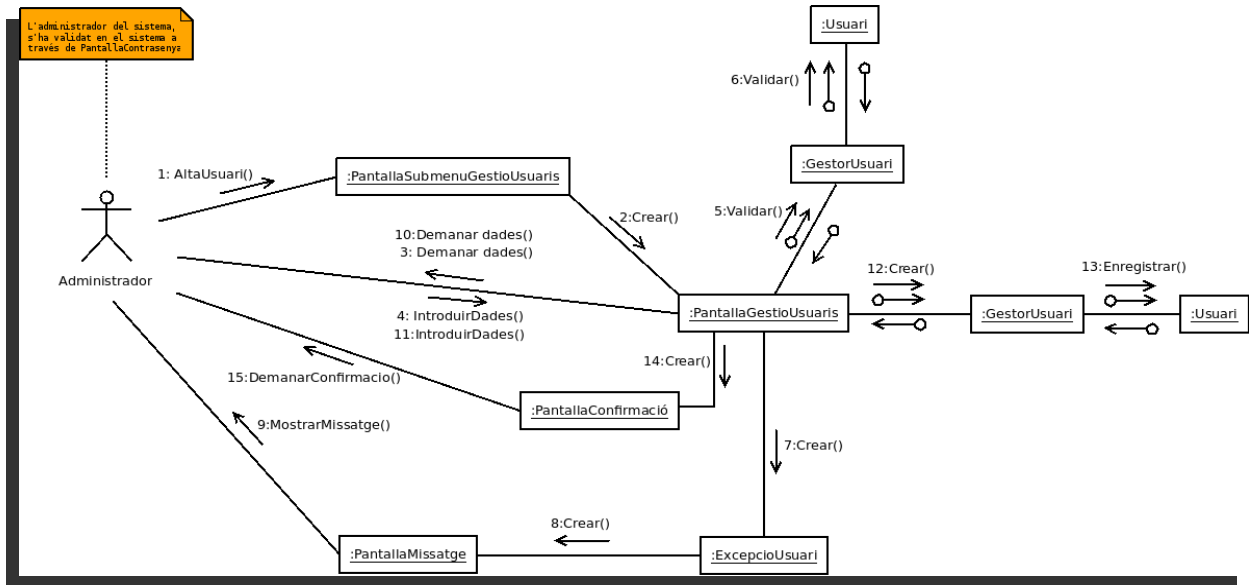
Il·lustració 11: Diagrama de classes



### 4.2.2. Diagrames de col·laboració i seqüència

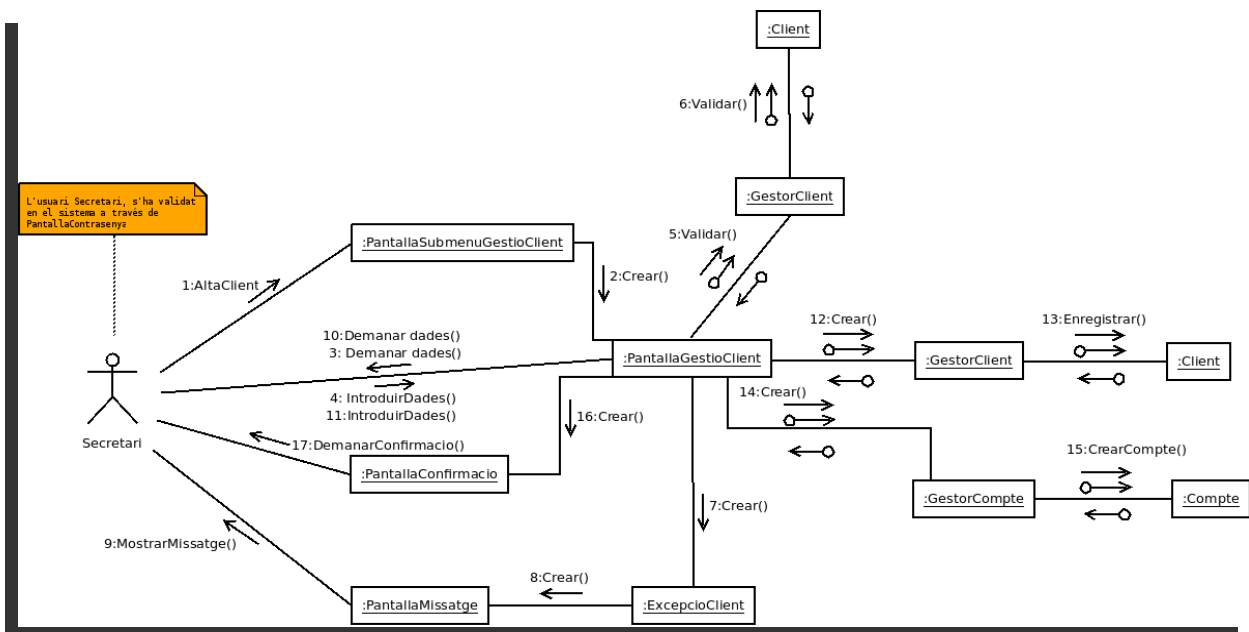
Diagrama de col·laboració i seqüència per alguns dels casos d'ús més rellevants de l'aplicació.

- Alta Nou Usuari del Sistema**



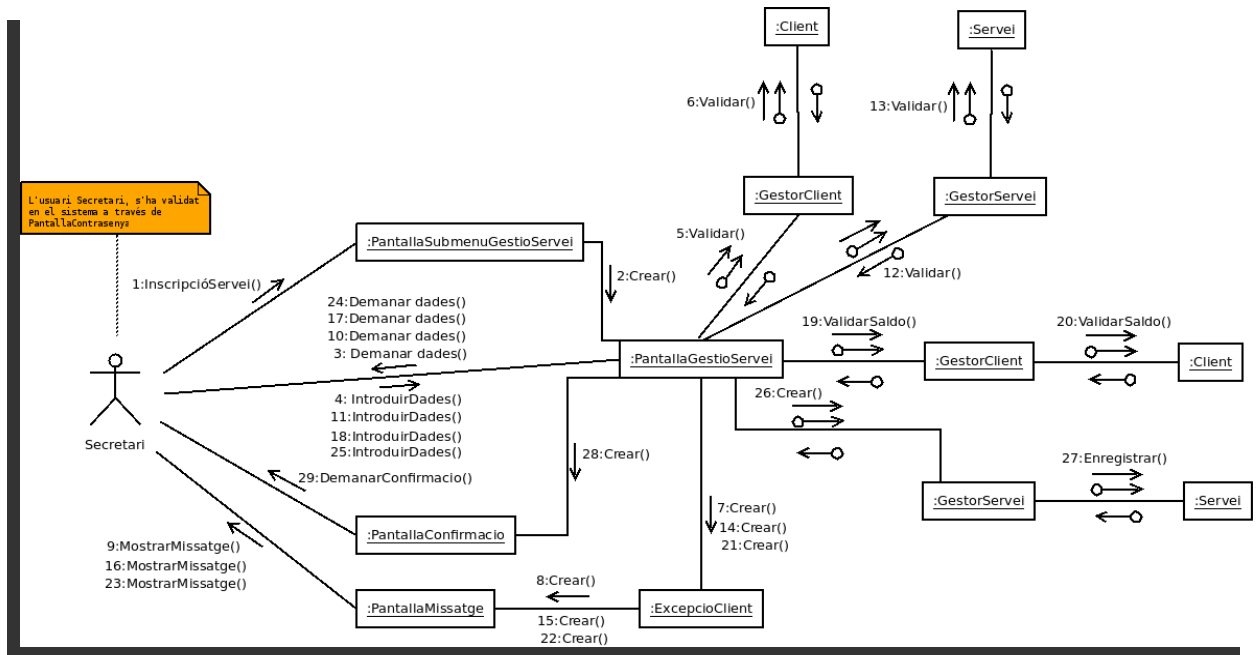
Il·lustració 12: Diagrama Alta Nou Usuari del Sistema

- Alta Nou Compte Client**



Il·lustració 13: Diagrama Alta Nou Compte Client

• **Inscripció a un Servei per part d'un client**



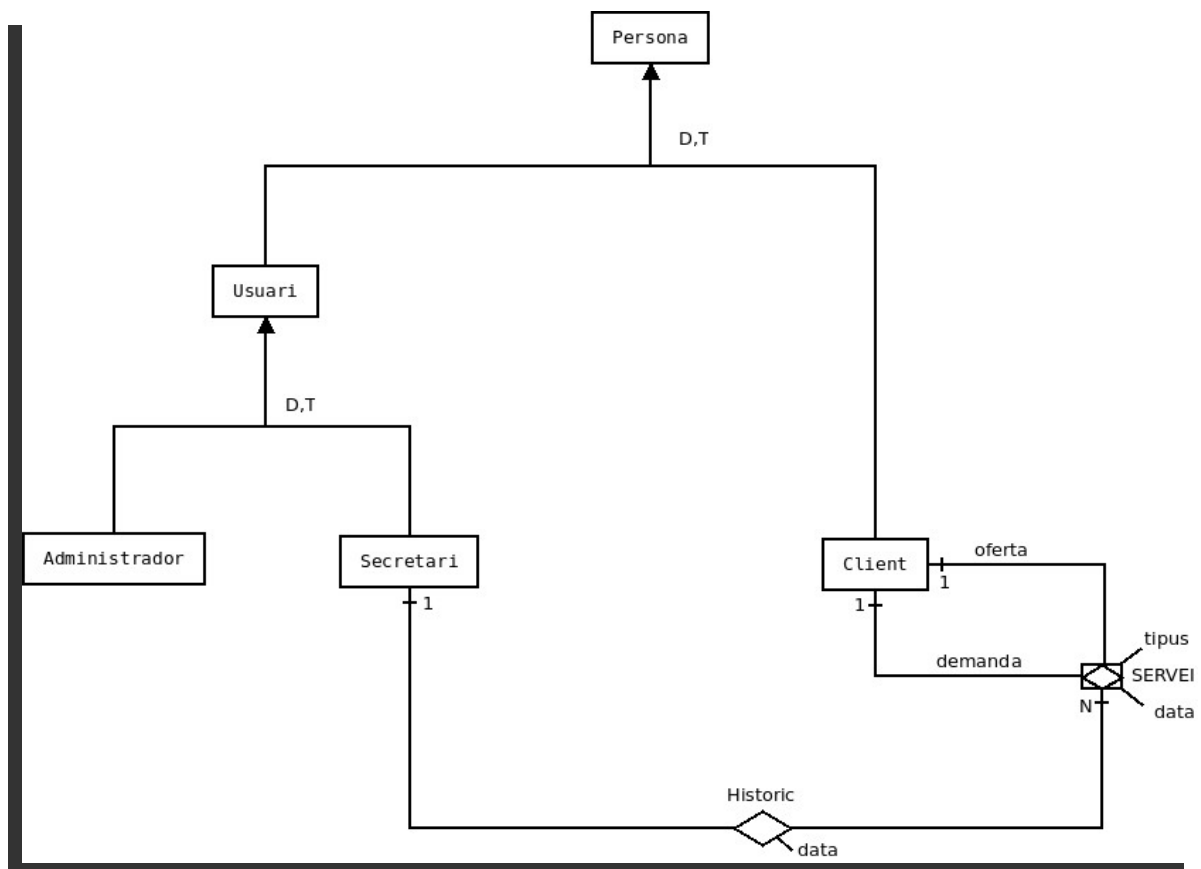
II-lustració 14: Diagrama Inscripció a un servei

### 4.3. Persistència

Les instàncies persistents, són aquelles de les quals es vol mantenir les dades en una base de dades. En el nostre cas farem servir la base de dades Postgres, però podria ser qualsevol altre com MySQL, Oracle... Ja que són compatibles amb el programari J2EE.

A continuació mostrarem el diagrama ER corresponent, per veure de quines classes guardem les dades de l'aplicació 'TBank'.

#### 4.3.1. Diagrama Entitat-Relació



Il·lustració 15: Diagrama Entitat-Relació

#### 4.3.2. Descripció dels atributs

---

**PERSONA**

codi-persona, nom, cognoms, telefon

**USUARI** (Subclasse de PERSONA)

codi-persona, dni, data-alta, clau

**CLIENT**(Subclasse de PERSONA)

codi-persona, nif, codi-postal, data-alta, direccio, disponibilitat, email, subtipus, poblacio, saldo, tipus-servei

**ADMINISTRADOR** (Subclasse de USUARI)

codi-persona

**SECRETARI** (Subclasse de USUARI)

codi-persona

**SERVEI** (Entitat Associativa)

codi-client-oferta, codi-client-demanda, tipus, data, descripcio

**HISTORIC**

Id, codi-client-oferta, codi-client-demanda, codi-secretari, tipus, data, descripcio, data-tancament

- La classe **Persona** no és persistent, per tant els seus atributs (nom, cognoms, telèfon) passen a formar part de la taula usuaris i clients.
- La taula **Historic**, es on es guarden els serveis un cop que s'ordena al sistema fer el pagament per part del secretari, es mouen a aquesta taula per tal de que la **taula servei** no sigui cada cop més pesada.
- L'atribut codi-usuari de la classe usuari, serà el DNI, l'atribut codi-client de les classe client, serà el NIF. Amb aquesta mesura es pretén diferenciar els usuaris dels clients amb més claretat.
- Les subclasse administrador i secretari, passen a ser un atribut tipus d'usuari.

## 4.4. Arquitectura de l'aplicació

---

J2EE, **Java Enterprise Edition**, és una plataforma de programació que té incloses varies API, amb la qual podem desenvolupar aplicacions distribuïdes de N capes i desplegar-les en un servidor d'aplicacions, en la qual aquesta ens proporciona escalabilitat, seguretat, ect... A més, un programador a partir d'aquesta tecnologia pot desenvolupar una aplicació empresarial escalable i portable entre plataformes. Hem decidit per al nostre projecte emprar la versió JEE 5.

Per al nostre desenvolupament hem decidit finalment, després de moltes divagacions, triar l'arquitectura **EJB** (*Enterprise JavaBeans*) en la seva versió 3.

Aquesta arquitectura de components de servidor ens permet entre altres coses:

- Escriure aplicacions escalables, fiables i segures.
- Facilitar la programació d'aplicacions distribuïdes.
- Portabilitat de les aplicacions entre diverses plataformes i servidors d'aplicacions.

Per tant podem dir que un **EJB** (*Enterprise JavaBeans*) de la versió 3, es una classe java (POJO – *Plain Old Java Object*) que s'executa dins un contenidor del servidor. Capa de negoci.

Aquests (POJO) contenen anotacions, mitjançant les anotacions podem declarar:

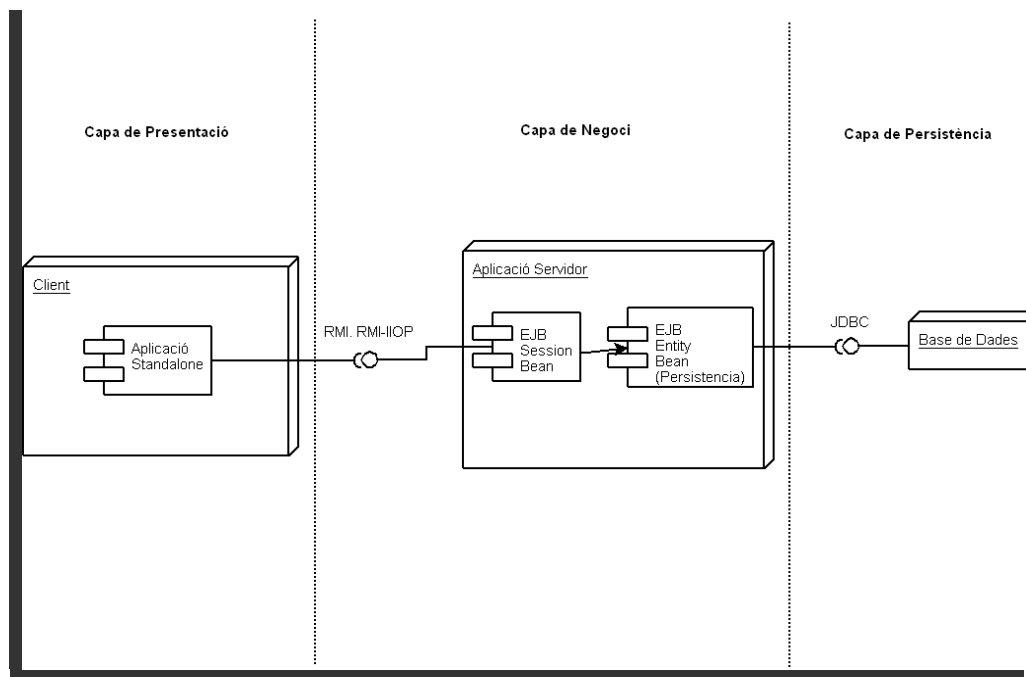
- Una interfície del Bean
- Mapeig O/R
- Referències a recursos

Algunes de les anotacions poden ser:

- `@Stateless`
- `@Statefull`
- `@Local`
- `@Remote`

Si un usuari vol accedir als serveis dels **EJB** de sessió (capa de negoci), ho ha de fer des de la capa de presentació, ho pot fer a través d'un framework de presentació, com per exemple **JSF** (*Java Server Faces*), o desde un **client Standalone**, una aplicació Java standard.

En el nostre cas, accedirem a la interfície (que és el que veu l'aplicació client) a través d'una aplicació **swing** de Java tal com podem veure en el gràfic següent.



Il·lustració 16: Diagrama de capes EJB

La interfície (capa negoci), ofereix al client una sèrie de serveis que resolen un cas d'ús concret.

En l'aplicació servidor, (capa de negoci), és com hem comentat anteriorment, on es despleguen els **EJB** (*Enterprise JavaBeans*), dels quals hi han tres tipus.

- *Session Bean*: Permeten accedir als serveis de manera síncrona.
- *Message-Driven Bean*: Permeten accedir als serveis de manera asíncrona.
- *Entity Bean*: Permeten accedir a la capa de persistència, en la versió 3 de EJB, s'accedeix a la capa de persistència mitjançant POJO.

Es defineixen dos tipus de *Session Bean*.

- Sense estat (*@Stateless*): El client fa una crida d'aquest sense cap ordre preestablert.
- Amb estat (*@Statefull*): La crida que fa el client condiciona les altres.

Un cop creat els nostres *Session Beans*, és despleguen en un servidor d'aplicacions per que puguin ser cridats per l'aplicació client, en el nostre cas el servidor d'aplicacions és **JBOSS**.

Un cop que el client interactua amb la interfície, aquest accedirà a les dades de persistència mitjançant el **Session Bean**, aquest últim interactua amb el **Entity Bean** per que mitjançant l'**API de persistència JPA** interactuï amb el gestor de base de dades **Postgres 8.4**, per tal d'obtenir les dades persistents.

## 5. CODIFICACIÓ

En aquesta secció, explicarem les decisions preses en quan a disseny i codificació que s'han efectuat en aquesta fase. Encara que no difereix gaire de l'original, s'han hagut de fer ja que eren necessàries i representaven una millora en quan al plantejament original.

A més, exposarem el *software* requerit per tal que la nostra aplicació funcioni correctament. Així com de proveir un petit manual d'instal·lació.

### 5.1. Decisions de disseny i codificació

---

Ara exposarem les raons de les modificacions realitzades, en quan a disseny i codificació de la nostra aplicació, la raó principal tal com hem dit anteriorment, és una millora en quan al plantejament de la solució original.

- **Millora del disseny del client:** S'ha procedit a millorar l'aspecte visual de la interfície del client, mitjançant l'ús de *Jpanels*, amb els quals agrupem en un mateix grup, els camps referents a un determinat grup de dades. A més, en les pantalles de gestió, s'ha procedit a redissenyar la botonera introduint un nou botó, el botó de cerca. La raó ha sigut el facilitar la navegació i la corba d'aprenentatge a l'usuari.
- **Nou camp a la taula de la BBDD HISTORIC:** Alhora de fer la implementació de la base de dades, vaig adonar-me que havia una petita errada en quan al disseny d'aquesta taula, corria el perill de matxucar els diferents moviments d'un mateix client cada cop que s'emmagatzemava un moviment d'aquest. La solució va ser introduir un camp 'Id' i auto-incrementable. D'aquesta manera es solucionava el problema plantejat.
- **Ús de Java Persistence API, JPA:** En un principi s'havia plantejat persistir les dades de la nostra aplicació, mitjançant els *Session Beans* i *JDBC*. Això implicava un parell de problemes, tenir una codificació dels *Session Beans* feixugues i unes consultes codificades en *SQL*. Això vol dir, que si es vol canviar de gestor de base de dades, hauríem d'adaptar el codi *SQL* al nou gestor. Utilitzant la **Java Persistence API-JPA**, a part de ser un *Standard* de *EJB3*, implica menys línies de programació i a més no tenir que reescriure les consultes *SQL*. Ja que *JPA* quan fa la consulta s'adapta automàticament al gestor de base de dades.
- **Executable per al Sistema Operatiu Windows:** S'ha cregut convenient per a la part client d'incloure un fitxer executable per Windows de la nostra aplicació. En principi només s'havia pensat per incloure un fitxer executable per Linux. Però d'aquesta manera incrementem el nombre d'usuaris.

## 5.2. Requeriments de software

---

L'aplicació de gestió del banc del temps, està pensada per córrer en diferents sistemes operatius que hi han actualment en el mercat. Però, ens centrarem concretament en el Linux de la distribució **Ubuntu 10.04**, que és on s'ha desenvolupat la nostra aplicació **TBANK**. A continuació mostrem el programari requerit per que la nostra aplicació funcioni correctament.

- **Sistema Operatiu: Linux Ubuntu 10.04.** <http://www.ubuntu.com/>. Sistema Operatiu basat en Linux. Es una distribució bastant estable.
- **Base de Dades: PostgreSQL 8.4,** <http://www.postgresql.org/> ja que és un gestor de base de dades relacional bastant fiable i gratuït. També s'aconsella instal·lar l'eina d'administració gràfica per gestionar el nostre gestor. **PgAdmin III.** <http://www.pgadmin.org/>.
- **Java Enterprise Edition:** Per un correcte funcionament de l'aplicació s'ha de tenir instal·lat en la part servidora JEE 5. <http://www.oracle.com/technetwork/java/javaee/tech/javaee5-jsp-135162.html>
- **Java Standard Edition 6:** Per al correcte funcionament de la part client de l'aplicació. S'ha de tenir instal·lat en la part de l'usuari. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- **JBoss:** És un servidor d'aplicacions JEE de codi obert implementat en java pur. Pot ser utilitzat en qualsevol Sistema Operatiu. La versió emprada, es la **JBoss-5.0.1.GA.** <http://www.jboss.org/>

A més per que el servidor **Jboss**, es pugui connectar a la base de dades, és necessita el **driver JDBC de Postgres 8.4** <http://jdbc.postgresql.org/download.html>. Es proporciona amb la distribució de la nostra aplicació.

S'ha de tenir present, que no explicarem com s'ha d'instal·lar el programari requerit, sinó que es parteix de la base que ja està instal·lat correctament en l'ordinador. S'ha de remarcar, que si es vol instal·lar en un Sistema Operatiu *Windows*, el programari requerit es el mateix.

## 5.3. Script de la Base de dades

---

La base de dades de la nostra aplicació, s'aconsella crear-la a través del **pgAdmin III**. Un cop creada la base de dades buida, s'importa o es copia les instruccions SQL per a la creació de les taules de la qual es compona la base de dades des de el fitxer que es lliura amb la distribució **DDL-TBANK.sql** i que es troba a la carpeta **DDBB**.

A continuació mostrem el contingut del fitxer **DDL-TBANK.sql**.



```

--TBank Software DataBase
--TFC 2011
-- DROP DATABASE "TBankDB";
-- CREATE DATABASE "TBankDB" WITH ENCODING='UTF8'OWNER=tbank CONNECTION LIMIT=-1;
-- Creació Domini d'usuari en Postgres

CREATE DOMAIN user_type
AS text
CHECK (VALUE IN ('Administrador', 'Secretari));

-- Creació taula Usuari
CREATE TABLE USUARI (
    codi_usuari INTEGER NOT NULL,
    nom character varying(30) NOT NULL,
    cognoms character varying(60) NOT NULL,
    telefon INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    tipus_usuari user_type,
    data_alta date NOT NULL,
    clau character varying(30) NOT NULL,
    PRIMARY KEY (codi_usuari)
);

-- Creació Taula Client
CREATE TABLE CLIENT (
    codi_client character varying(9) NOT NULL,
    nom character varying(30) NOT NULL,
    cognoms character varying(60) NOT NULL,
    telefon INTEGER NOT NULL,
    nif character varying(9) NOT NULL,
    saldo INTEGER NOT NULL,
    data_alta date NOT NULL,
    direccio character varying(60) NOT NULL,
    codi_postal INTEGER NOT NULL,
    poblacio character varying(30) NOT NULL,
    email character varying(60) NOT NULL,
    tipus_servei character varying(60) NOT NULL,
    subtipus character varying(60) NOT NULL,
    disponibilitat character varying(200) NOT NULL,
    PRIMARY KEY (codi_client)
);

-- Creació Taula Servei
CREATE TABLE SERVEI (
    codi_client_oferta character varying(9) NOT NULL,
    codi_client_demanda character varying(9) NOT NULL,
    tipus_servei character varying(60) NOT NULL,
    data_alta date NOT NULL,
    descripcio character varying(255) NOT NULL,
    PRIMARY KEY (codi_client_oferta),
    FOREIGN KEY (codi_client_oferta) REFERENCES CLIENT(codi_client),
    FOREIGN KEY (codi_client_demanda) REFERENCES CLIENT(codi_client)
);

-- Creació Taula Historic de Servei
CREATE TABLE HISTORIC (
    id serial,
    nif_client_oferta character varying(9) NOT NULL,
    nif_client_demanda character varying(9) NOT NULL,
    codi_secretari INTEGER NOT NULL,
    tipus_servei character varying(60) NOT NULL,
    data_alta date NOT NULL,
    data_tancament date NOT NULL,
    descripcio character varying(255) NOT NULL,
    PRIMARY KEY (id)
);

-- Creació Taula tipus servei
CREATE TABLE TIPUSERVEI (
    id serial,
    tipus_servei character varying(100) NOT NULL,
    PRIMARY KEY (id)
);

-- Dades per defecte de l'aplicació
-- Usuari administrador per defecte.
INSERT INTO USUARI VALUES (11111111, 'Nom1', 'Cognoms', 999999999, 11111111, 'Administrador', '1-01-2011', 'admin');

-- Creació tipus de servei de l'aplicació
INSERT INTO TIPUSERVEI VALUES (1, 'Animals');
INSERT INTO TIPUSERVEI VALUES (2, 'Assesoria');
INSERT INTO TIPUSERVEI VALUES (3, 'Belleza i Salut');
INSERT INTO TIPUSERVEI VALUES (4, 'Casa');
INSERT INTO TIPUSERVEI VALUES (5, 'Construcció');
INSERT INTO TIPUSERVEI VALUES (6, 'Cultura');
INSERT INTO TIPUSERVEI VALUES (7, 'Educació');
INSERT INTO TIPUSERVEI VALUES (8, 'Imatge i So');
INSERT INTO TIPUSERVEI VALUES (9, 'Moda');
INSERT INTO TIPUSERVEI VALUES (10, 'Motor');
INSERT INTO TIPUSERVEI VALUES (11, 'Negocis');
INSERT INTO TIPUSERVEI VALUES (12, 'Oci');
INSERT INTO TIPUSERVEI VALUES (13, 'Tecnologia');
INSERT INTO TIPUSERVEI VALUES (14, 'Varis');

```

## 5.4. Instal·lació de la Base de Dades

---

Un cop que hem acabat d'instal·lar el servidor de Base de dades **PostgreSQL versió 8.4** i l'eina d'administració **pgAdmin III**. Procedim a executar la segona fase, seguint els passos següents:

1. Obrim el pdAdmin III.
2. Creem un nou usuari a la base de dades o a l'usuari **postgres** afegim un *password*. El *password* per defecte que s'ha d'afegir és 'astrid10'.
3. Creem la base de dades de l'aplicació, **TBankDB**, amb els valors per defecte.
4. Un cop creada la base de dades, procedim a crear les taules, desde l'arxiu **DDL-TBANK.sql**.
5. Prenem el botó **SQL** de la l'administrador **pgAdmin III**.
6. Fem un *Copy & Paste* del fitxer DDL-TABNK.sql a la pantalla i prenem el botó **Execute Query**.
7. Ens crea les bases de dades i introdueix un Usuari Administrador per defecte de l'aplicatiu.
8. Un cop fet això ja podem tancar el pdAdmin III.

## 5.5. Instal·lació de Tbank en el servidor JBoss

---

Un cop instal·lat correctament el servidor d'aplicacions **JBoss** en el nostre ordinador o servidor. Explicarem com instal·lar la part servidora del nostre aplicatiu, **Tbank**.

- Obrir la carpeta **TBankEJB** i copiar el fitxer **TBankEJB.jar** en el servidor, dins la carpeta on hem instal·lat el servidor **JBoss**, en la ruta: `../server/default/deploy`
- Obrir la carpeta **TBankEJB** i copiar el fitxer **jboss-ds.xml** en el servidor, dins la carpeta on hem instal·lat el servidor **JBoss**, en la ruta: `../server/default/deploy`

Aquest arxiu conté els paràmetres de connexió del servidor d'aplicacions **JBoss** a la base de dades. Bé configurat amb un usuari de la base de dades per defecte. Si la base de dades fos en un altre ordinador o tingués un altre usuari o paraula clau, hauríem d'obrir-lo i modificar-lo.

A continuació mostrem el contingut del fitxer:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>tbank_postgres</jndi-name>
    <connection-url>jdbc:postgresql://localhost:5432/TBankDB</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>postgres</user-name>
    <password>astrid10</password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
  </local-tx-datasource>
</datasources>
```

1. Si el servidor de Base de Dades fos en un altre Servidor, hauríem de retocar la part de <connection-url>. Envers de *localhost* posar la IP on es troba i el port.
2. Si la base de dades ha sigut creada amb un altre propietari, hauríem de retocar <user-name>...</user-name> i posar l'usuari que toca.
3. Si la base de dades ha sigut creada amb un altre *password*, hauríem de retocar <password>...</password> i posar el *password* que toca.

Un cop fet això, procedirem a guardar-lo i instal·lar-lo en el servidor *JBoss*.

- Per últim copiarem el *driver* JDBC de Postgres que es troba a la carpeta BBDD i el copiarem en el servidor a la ruta: `../server/default/lib`

Un cop realitzats aquest passos, procedirem a arrencar el servidor d'aplicacions *JBoss*. Si tot ha anat bé el servidor mostrarà una pantalla com aquesta.

```

joe@joe-laptop: ~/jboss/bin
File Edit View Terminal Help
AreaBean/remote - EJB3.x Default Remote Business Interface
AreaBean/remote-com.tbank.coneixement.AreaRemote - EJB3.x Remote Business Interface
09:00:47,778 INFO [JndiSessionRegistrarBase] Binding the following Entries in Global JNDI:
    UsuariBean/remote - EJB3.x Default Remote Business Interface
    UsuariBean/remote-com.tbank.usuari.UsuariRemote - EJB3.x Remote Business Interface
09:00:47,787 INFO [JndiSessionRegistrarBase] Binding the following Entries in Global JNDI:
    HistoricBean/remote - EJB3.x Default Remote Business Interface
    HistoricBean/remote-com.tbank.historic.HistoricRemote - EJB3.x Remote Business Interface
09:00:47,870 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8080
09:00:47,915 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
09:00:47,927 INFO [ServerImpl] JBoss (Microcontainer) [5.0.1.GA (build: SVNTag=JBoss_5_0_1_GA date=200902231221)] Started in 52s:867ms

```

Il·lustració 17: Pantalla del Servidor JBoss

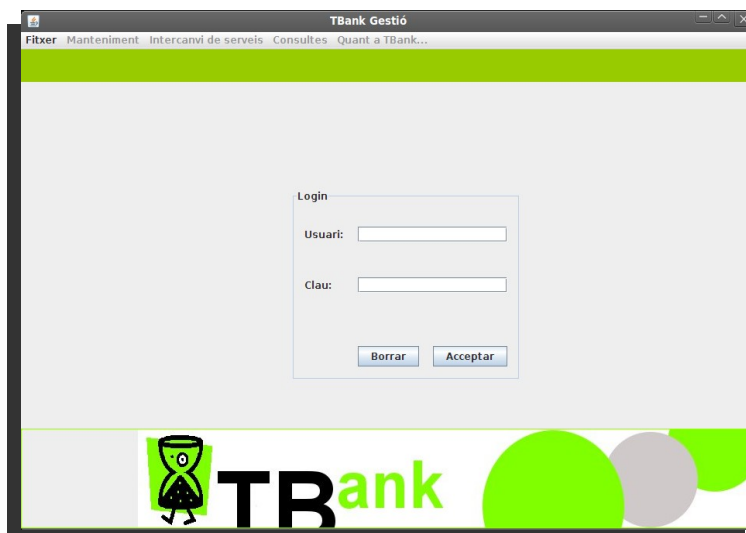
## 5.6. Instal·lació de Tbank Client

---

Ara donarem els passos per instal·lar la part del client de la nostra aplicació.

- Copiem de la distribució la carpeta **TBankClient** a la ruta que vulguem.
- Dins la carpeta **TBankClient** troben, el fitxer **jndi.properties**, doncs posarem la ruta d'aquest en el **CLASSPATH**. Reiniciarem l'ordinador per que els canvis en el **CLASSPATH** tinguin efecte.
- Si el servidor JBoss, és en una altre màquina, a llavors s'haurà d'obrir aquest fitxer i retocar la línia **java.naming.provider.url=localhost:1099**, envers de **localhost:1099**, posar la IP de la màquina i el port habilitat.
- Desde la lina de comandes, anar al directori **TbankClient** i executar la comanda **java -jar TbankClient.jar**. O executar el fitxer executable **TBankClient.sh**, en el cas del S.O. Windows, executar **TBankClient.bat**.

Haurà de sortir la pantalla de *login* següent:

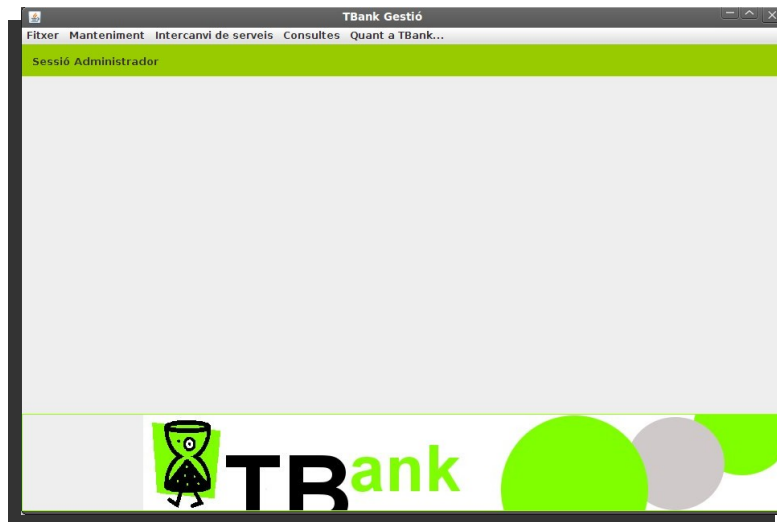


Il·lustració 18: Pantalla de Login Client

Un cop que hem posat en marxa l'aplicació introduïrem l'usuari administrador per defecte.

- **Usuari:** 11111111
- **Clau:** admin

Si s'ha introduït correctament, accedirem a la pantalla principal:



Il·lustració 19: Pantalla principal TBankClient

## 6. CONCLUSIONS

La realització d'aquest projecte de **gestió d'un Banc de Temps**, ha sigut una experiència enriquidora a nivell personal en tots els nivells, sobretot per aprendre i acostar-me al apassionant món de Java EE.

En un principi, quan t'acostes al món de Java EE, quedes bastant desorientat ja que hi ha una amalgama de *frameworks* i tecnologies. Per superar aquesta desorientació, vaig haver de fer una serie de passos. Un d'ells, explorar la millor tecnologia de J2EE afí pel desenvolupament de la meva aplicació. Per tant, això en van ajudar a fer una bona planificació.

Penso que una bona planificació es un pilar bàsic en tot projecte, ja que ens permet abordar-lo amb èxit fins al final i acomplir amb els terminis establerts amb el client. Si no fos així, acabaríem desbordats i el projecte fracassaria.

Per tant, com que el meu projecte ha sigut desenvolupat al 100%, crec que en el seu moment vaig fer una bona planificació, en el qual vaig tenir en compte dues coses molt importants d'aquest projecte, el temps de dedicació i la corba d'aprenentatge de la tecnologia escollida per a desenvolupar-lo.

En quan a la tecnologia escollida de J2EE, en un principi vaig pensar en **Java Server Faces**. Però finalment en vaig decantar per **Enterprise Java Beans - EJB 3**, Ja que la codificació d'aquesta és molt semblant a la tecnologia de Java d'aplicacions distribuïdes per **RMI**. Per tant vaig pensar que al ser similar, ajudaria a reduir la corba d'aprenentatge i d'aquesta manera acomplir amb els terminis establerts en la planificació.

Per acabar, penso que alhora d'abordar un projecte, sigui del tipus que sigui, es molt important tenir un bon coneixement de la tecnologia a emprar, així com fer una bona planificació, penso que aquests dos factors influeixen en gran mesura en el resultat final del projecte, sigui d'una qualitat acceptable o excel·lent.

## GLOSSARI

Glossari de termes relacionats amb el model de negoci de l'aplicació Gestió Banc del temps :

**Acceptar:** Confirmar una acció iniciada per un usuari.

**Administrador:** Rol d'Usuari del sistema que s'encarrega de la gestió dels usuaris (altes, baixes i modificacions).

**Base de Dades:** Conjunt d'informació de l'aplicació organitzada en taules i emmagatzemada pel sistema.

**Banc del temps:** Entitat sense ànim de lucre en la qual no hi han amos, la seva fita és donar solucions als problemes quotidians mitjançant els serveis que donen els usuaris vers altres usuaris del banc mitjançant serveis.

**Client:** Persona que fa ús dels serveis del banc del temps.

**Codi client:** NIF de l'usuari del banc del temps.

**Compte Client:** Compte d'un usuari determinat del banc del temps.

**Contrasenya:** Conjunt de caràcters que permeten l'accés al sistema.

**Funcionalitat:** Tasca o utilitat del programa.

**Nom usuari:** Nom amb què un usuari està identificat en el sistema.

**Pantalla:** Interfície gràfica del programa a la que accedeix un usuari i que li permet interactuar amb el sistema.

**Pagament:** En el cas del banc del temps, 1 servei=1 hora, es la transferència d'una hora de la compte de l'usuari que rep el servei a la compte de l'usuari que realitza el servei.

**Saldo:** En un banc de temps, es el nombre d'hores que acumula un usuari d'aquest en un moment determinat.

**Secretari:** Rol d'usuari que assumeix les tasques relacionades amb la gestió del banc del temps.

**Servei:** Moneda de canvi en un banc de temps 1 servei=1 hora, és la feina que es realitza per part d'una persona a una altre persona a canvi d'una hora de temps.

**Subsistema:** Part de l'aplicació amb funcionalitats comunes.

**Subsistema de connexió:** Subsistema que s'ocupa de la connexió al sistema.

**Subsistema de consultes generals:** Subsistema responsable de l'extracció de llistats a partir de les dades de l'aplicació.

**Subsistema d'intercanvi de serveis:** Subsistema utilitzat per gestionar els serveis realitzats pels usuaris del banc del temps.

**Subsistema de manteniment:** Subsistema responsable del manteniment de l'aplicació.

**Usuari:** Persona que fa ús de l'aplicació. Amb rol d'Administrador o Secretari.

**BIBLIOGRAFIA**

- **Administrador.** Construir aplicaciones EJB con Jboss.  
[www.programacion.com/articulo/construir\\_aplicaciones\\_ejb\\_con\\_jboss\\_lomboz\\_y\\_eclipse\\_267](http://www.programacion.com/articulo/construir_aplicaciones_ejb_con_jboss_lomboz_y_eclipse_267)
- **Bill Burke, Richard Monson (2001).** Enterprise JavaBeans. O'Reilly
- **Bill Burke, Richard Monson (2006).** Enterprise JavaBeans, 3.0. O'Reilly
- **Debu Panda, Reza Rahman, Derek Lane (2007).** EJB3 in Action. Manning
- **Felipe Lima Díaz (2010).** Manual Avanzado JAVA 6. ANAYA Multimedia
- **Francisco Javier Martines.** EJB 3.0: Resurrection.  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EJB3Resurrection>
- **George Reese (2001).** JDBC y JAVA. ANAYA Multimedia – O'Reilly
- **John Wiley.** Master EJB and J2EE. [www.ForoDeJava.com](http://www.ForoDeJava.com)
- **Jboss Community.** EJB3. [www.jboss.org/ejb3](http://www.jboss.org/ejb3)
- **Jboss Community.** Installation and Getting Started Guide . [www.jboss.org/](http://www.jboss.org/)
- **Rogers CadenHead, Laura Lemay (2007).** JAVA 6. ANAYA Multimedia
- **Oracle.** Enterprise JavaBeans. [www.oracle.com/technetwork/java/index-jsp-140203.html](http://www.oracle.com/technetwork/java/index-jsp-140203.html)
- **Ozito.** Curso JavaBeans . [www.ForoDeJava.com](http://www.ForoDeJava.com)
- **Sebastian Hennebrueder (2006).** First EJB 3 Tutorial.  
[www.laliluna.de/download/first-ejb3-tutorial-en.pdf](http://www.laliluna.de/download/first-ejb3-tutorial-en.pdf)
- **Wikipedia.** Enterprise JavaBeans. [es.wikipedia.org/wiki/Enterprise\\_JavaBeans](http://es.wikipedia.org/wiki/Enterprise_JavaBeans)



## ANNEXES

### Tour gràfic per Tbank Client

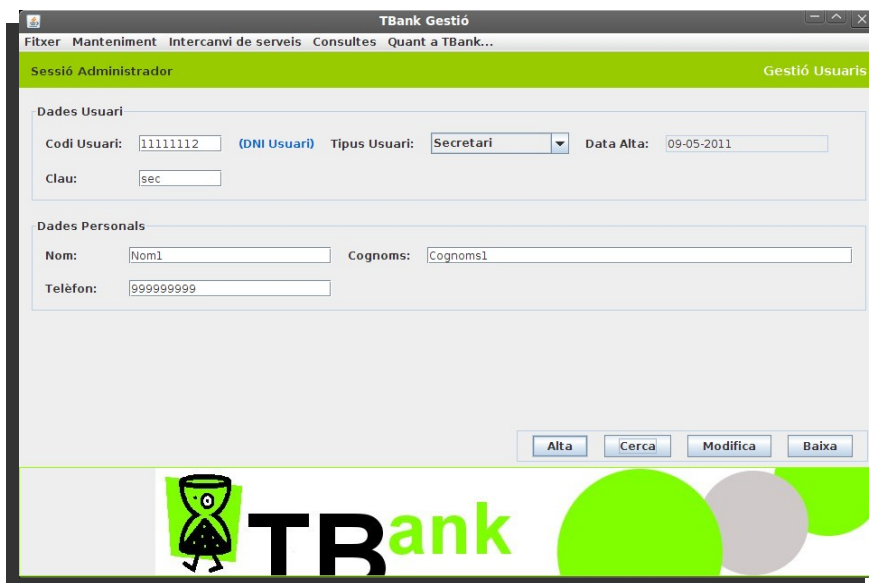
En aquesta secció podem veure les captures de les principals pantalles de l'aplicació de gestió TbankClient.

- Pantalla Login



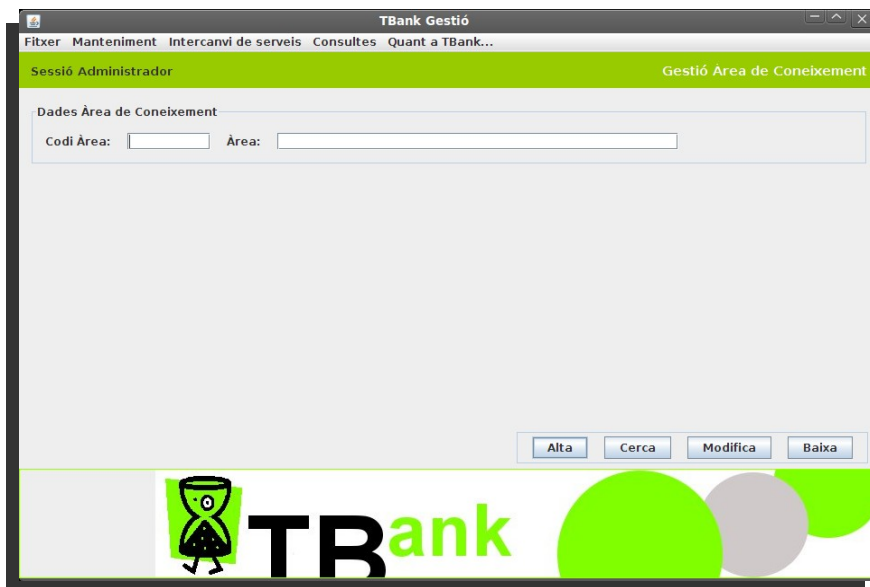
Il·lustració 20: Pantalla LoginTBankClient

- Pantalla Gestió Usuaris



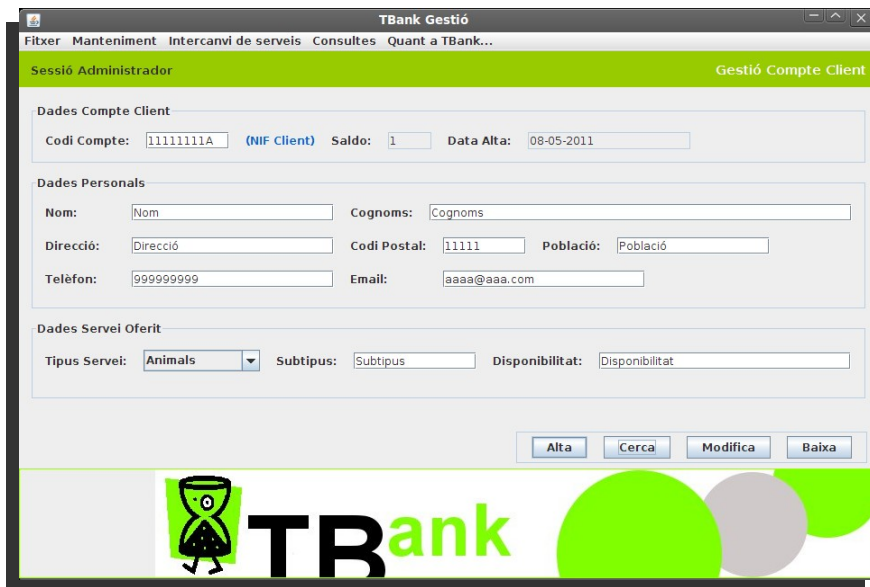
Il·lustració 21: Pantalla Gestió Usuaris

- **Pantalla Gestió Àrea de Coneixement**



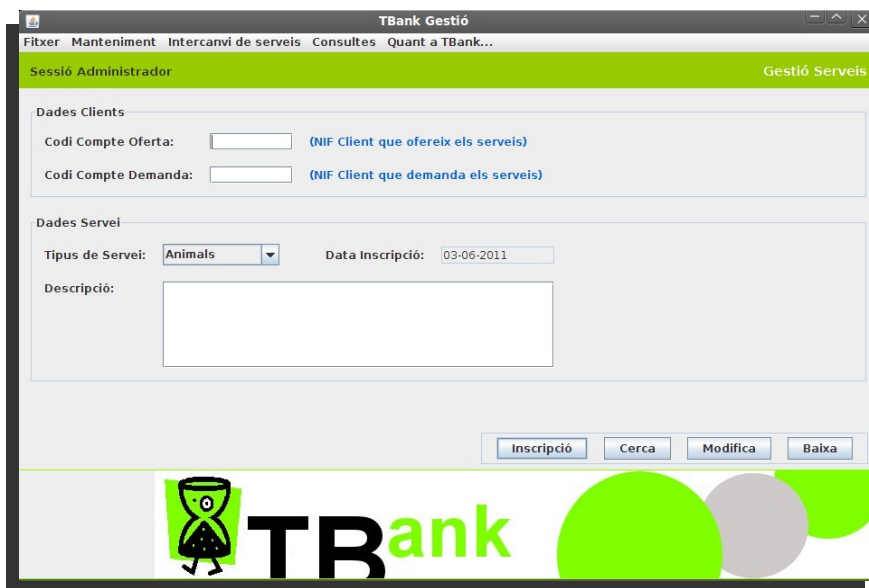
Il·lustració 22: Pantalla Gestió Àrea de Coneixement

- **Pantalla Gestió Compte Client**



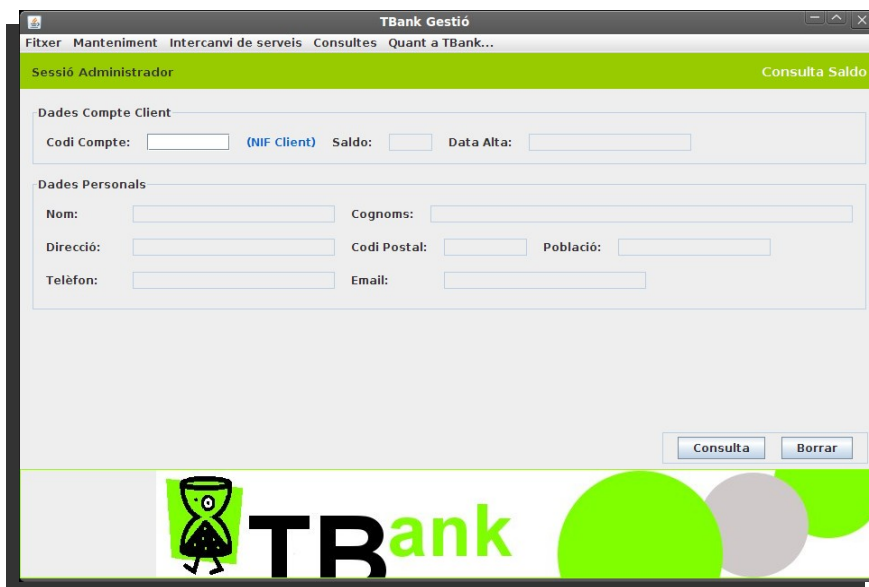
Il·lustració 23: Pantalla Gestió Compte Client

- **Pantalla Gestió Serveis**



Il·lustració 24: Pantalla Gestió Serveis

- **Pantalla Consulta Saldo**



Il·lustració 25: Pantalla Consulta Saldo