

# TFC: Gestor de l'economia domèstica

## Memòria

19/06/2011

Roger Artigas Soler

ETIS

Consultor: Salvador Campo Mazarico

**A l'Eulàlia per haver aguantat tants caps de setmana  
d'estudi i sacrifici**

**A l'Arnau, que m'ha donat l'energia en la recta final**

## Resum

El present document conté la memòria del Treball Final dels estudis d'Enginyeria Tècnica Informàtica de Sistemes. El projecte consisteix en el desenvolupament d'una plataforma web utilitzant la tecnologia J2EE.

Aquesta aplicació té com a objectiu permetre la gestió de l'economia domèstica als diferents usuaris, de forma que puguin registrar les despeses i ingressos de forma ordenada, classificar-les i poder posteriorment analitzar-les.

Per a dur a terme el projecte ha estat necessari seguir un enfoc metodològic en quant a les diferents fases: planificació, anàlisi, disseny i implantació, així com l'aplenentatge i ús de la tecnologia J2EE i dels diferents frameworks i llibreries de mercat que ajuden a estructurar l'aplicació en les diferents capes de l'arquitectura.

Aquest projecte està subjecte a llicència GPL:

Perpet copiar i distribuir el Programa (o un treball basat en ell, segons s'especifica en l'apartat 2, com a codi objecte o en format executable segons els termes dels apartats 1 i 2, suposat que a més compleixi una de les següents condicions:

1. Acompanyar-lo amb el codi font complet corresponent, en format electrònic, que ha de ser distribuït segons s'especifica en els apartats 1 i 2 d'aquesta Llicència en un medi habitualment utilitzat per a l'intercanvi de programes, o
2. Acompanyar-lo amb una oferta per escrit, vàlida durant almenys tres anys, de proporcionar a qualsevol tercera part una còpia completa en format electrònic del codi font corresponent, a un cost no major que el de realitzar físicament la distribució del font, que serà distribuït sota les condicions descrites en els apartats 1 i 2 anteriors, en un medi habitualment utilitzat per a l'intercanvi de programes, o
3. Acompanyar-lo amb la informació que vas rebre oferint distribuir el codi font corresponent. (Aquesta opció es permet només per a distribució no comercial i només si vostè va rebre el programa com a codi objecte o en format executable amb tal oferta, d'acord amb l'apartat 2 anterior).

# Índex

<b>1</b>	<b>Introducció</b>	<b>6</b>
1.1	Justificació i context	6
1.2	Objectius del TFC	6
1.3	Enfoc i mètode seguit	7
1.4	Planificació	8
1.5	Productes obtinguts	8
1.6	Descripció dels capítols de la memòria	8
<b>2</b>	<b>Cos de la Memòria</b>	<b>10</b>
2.1	Descripció del Projecte	10
2.2	Anàlisi i disseny de l'aplicació	12
2.3	Implementació	38
2.4	Conclusions i línies de futur	41
<b>3</b>	<b>Glossari</b>	<b>42</b>
<b>4</b>	<b>Bibliografia</b>	<b>43</b>
<b>5</b>	<b>Annexes</b>	<b>45</b>
5.1	Requeriments mínims	45
5.2	Manual d'instal·lació	45
5.3	Manual bàsic d'usuari	57

# 1 INTRODUCCIÓ

## 1.1 JUSTIFICACIÓ I CONTEXT

Els projectes finals representen l'acabament d'una etapa d'aprenentatge i l'inici d'una altra etapa on es comença a aplicar els coneixements adquirits. Aquest projecte no n'és una excepció: representa l'assoliment de coneixements complexos en l'àmbit de la informàtica amb una aplicació web com a resultat.

Tot i no tenir experiència previa en la tecnologia J2EE, m'ha atret el fet de ser una tecnologia emergent: estem veient com cada dia més i més aplicacions del món empresarial utilitzen entorns web. Entre les seves virtuts estan l'ús d'interfícies d'usuari accessibles per a tot nivell d'usuaris, i la possibilitat de treballar amb aplicacions ubicades a qualsevol servidor del món fent ús d'Internet... Totes aquestes característiques fan d'aquest un món apassionant.

Personalment he intentat posar tots els sentits i les hores que ha anat robant a la son per a utilitzar no només els coneixements adquirits al llarg de tots els anys d'estudi a la UOC, sinó també utilitzar les eines i mètodes per a fer una aprenentatge autodidacta i assolir els coneixements indispensables en la tecnologia J2EE per a poder portar a terme el projecte en una tecnologia que no és senzilla i que permet multitud d'alternatives.

Per a triar la temàtica de l'aplicació, vaig pensar en un problema senzill però amb un impacte potencial molt gran: la crisi financera actual ha causat que moltes famílies es trobin actualment en situacions límit, i moltes es troben sense eines que els ajudin a gestionar i planificar millor les seves despeses i ingressos. És així com decideixo iniciar un projecte que implementi una eina de comptabilitat domèstica que ajudi a tots els usuaris que es registrin a fer aquesta labor de classificació i anàlisi de la seva economia particular. Es busca facilitar el màxim l'ús i l'accés per a que els usuaris puguin fer-ne ús i treure'n profit. Resumint, es tracta de posar a disposició una eina que permeti planificar millor les economies particulars.

## 1.2 OBJECTIUS DEL TFC

L'objectiu final del TFC és la implantació d'una aplicació J2EE d'acord amb una planificació inicial, passant pel cicle de vida d'un projecte de creació de software, que contempla les fases d'anàlisi, disseny, construcció i proves, posada en producció, sense oblidar la documentació associada: la memòria i la presentació virtual.

Per a tal de portar-ho a terme, serà necessari definir una arquitectura tecnològica que ajudin a fer una separació de capes: patró Model – Vista – Controlador.

L'aplicació d'aquest patró facilita enormement la reutilització de codi, establir metodologia d'enginyeria del software que permetin un manteniment i escalabilitat funcional posterior de l'aplicació senzill i minimitzar els costos associats. Per a implementar-la serà necessari l'ús de frameworks de desenvolupament, la creació d'una Base de dades que asseguri la persistència de les dades que es van generant i tractant, un entorn que implementi la lògica de negoci i també l'interfície d'usuari en forma web, formant la capa de presentació.

### 1.3 ENFOC I MÈTODE SEGUIT

Pel desenvolupament del projecte s'ha seguit un enfocament metodològic similar al que es porta a terme en entorns professionals, on es segueixen una sèrie de passes o fases de forma ordenada que permeten minimitzar els riscos del projecte i augmentar les garanties d'èxit.

Les fases en que s'ha estructurat són:

- ◆ **Planificació:** consisteix en identificar les tasques necessàries a dur a terme en el projecte, dimensionar-les i planificar-les de forma realista. Com a resultat d'aquesta fase s'obté una descripció inicial de l'abast del projecte així com la llista d'activitats planificades amb data d'inici i finalització.
- ◆ **Anàlisi:** consisteix en analitzar funcionalment el problema que es resol en el projecte, identificar els casos d'ús i donar la solució funcional. Com a resultat s'obté el diagrama de casos d'ús, la descripció textual dels casos d'ús i un pla de proves que en finalitzar la construcció haurà de garantir el correcte funcionament del sistema.
- ◆ **Disseny:** es defineix com s'implementarà el projecte a nivell tècnic, l'arquitectura i el model d'implementació de la Base de dades. Per a fer-ho s'ha analitzat diferents opcions per a fer d'implementació: frameworks, servidors d'aplicacions, Bases de dades,... i s'ha realitzat la labor d'estudi i aprenentatge necessari pel seu ús en el projecte. El resultat és d'identificació de les classes principals, el disseny de la Base de dades i els diagrames de seqüència i d'activitat dels casos d'ús principals.
- ◆ **Construcció:** es programa i construeix cada una de les capes necessàries pel funcionament de la plataforma.
- ◆ **Proves:** S'executa el pla de proves definit en la fase d'anàlisi, es verifica que el resultat és l'esperat, tant el casos on s'espera l'èxit com un error i es corregeixen les incidències que s'identifiquin (el resultat difereix de l'esperat en el pla de proves).
- ◆ **Implantació:** Es segueix el manual d'instal·lació per a deixar l'entorn funcionant i operatiu.

## 1.4 PLANIFICACIÓ

La planificació del projecte s'adapta al calendari marcat per la UOC per a l'entrega del treball, i està dividit en diferents etapes, seguint la metodologia definida anteriorment.

Es defineixen en la planificació les següents activitats organitzades en 4 entregues:

- ◆ **PAC1:** Definició del projecte i planificació
- ◆ **PAC2:** Anàlisi i disseny de l'aplicació
- ◆ **PAC3:** Entrega parcial d'implementació
- ◆ **PAC4:** Entrega final. Aplicació completada, memòria i presentació virtual

A continuació es detalla la planificació detallada amb totes les activitats:

Tasques	Descripció	Inici	Fi	Març	Abril	Maig	Juny
1	<b>PAC 1 - Pla de treball</b>	01/03/2011	10/03/2011				
1.1	Pla - Definició dels objectius del TFC	01/03/2011	05/03/2011				
1.2	Pla - Definició i planificació de les tasques del projecte	06/03/2011	07/03/2011				
1.3	Redacció PAC1	06/03/2011	10/03/2011				
2	<b>PAC 2 - Anàlisis i Disseny</b>	11/03/2011	11/04/2011				
2.1	Anàlisi - Casos d'ús	11/03/2011	17/03/2011				
2.2	Anàlisi - Identificació dels actors	11/03/2011	17/03/2011				
2.3	Anàlisi - Definir model ER	18/03/2011	22/03/2011				
2.4	Anàlisi - Interfície gràfica	23/03/2011	23/03/2011				
2.5	Anàlisi - Definir l'índex de proves	24/03/2011	24/03/2011				
2.6	Disseny - Definició de l'arquitectura de l'aplicació	25/03/2011	27/03/2011				
2.7	Disseny - Comparació i selecció de l'entorn tècnic	25/03/2011	27/03/2011				
2.8	Disseny - Definició dels criteris i patrons	28/03/2011	02/04/2011				
2.9	Disseny - Lògica funcional	03/04/2011	05/04/2011				
2.10	Disseny - Model de persistència	06/04/2011	08/04/2011				
2.11	Disseny - Interfície gràfica	09/04/2011	10/04/2011				
2.12	Redacció PAC2	11/03/2011	11/04/2011				
3	<b>PAC 3 - Implementació</b>	12/04/2011	25/05/2011				
3.1	Impl - Preparació de l'entorn tecnològic	12/04/2011	15/04/2011				
3.2	Impl - Base de dades	16/04/2011	18/04/2011				
3.3	Impl - Mòdul de gestió d'usuaris	19/04/2011	26/04/2011				
3.4	Impl - Mòdul de configuració	27/04/2011	04/05/2011				
3.5	Impl - Mòdul d'entrada de moviments	05/05/2011	15/05/2011				
3.6	Impl - Mòdul de llistat	16/05/2011	23/05/2011				
3.7	Impl - Proves integrades	24/05/2011	24/05/2011				
3.8	Entrega PAC3	25/05/2011	25/05/2011				
4	<b>Entrega final</b>	26/05/2011	17/06/2011				
4.1	EF - Preparació del WAR	26/05/2011	27/05/2011				
4.2	EF - Redacció del manual de desplegament	28/05/2011	31/05/2011				
4.3	EF - Redacció del manual d'usuari	01/06/2011	04/06/2011				
4.4	EF - Redacció de la memòria	05/06/2011	10/06/2011				
4.5	EF - Presentació virtual	10/06/2011	17/06/2011				
5	<b>Formació</b>	11/03/2011	24/05/2011				

## 1.5 PRODUCTES OBTINGUTS

Els productes generats en el projecte són els que es detallen a continuació:

- ◆ El principal producte obtingut és l'aplicació desenvolupada en el projecte. S'entrega en format WAR i es pot desplegar en un servidor Tomcat (veure l'apartat "manual d'instal·lació").
- ◆ Memòria del projecte que explica el projecte en detall: objectius, planificació inicial i les diferents etapes de la metodologia seguida per a obtenir el resultat final.
- ◆ Presentació virtual que sintetitza el treball fet en el projecte

## 1.6 DESCRIPCIÓ DELS CAPÍTOLS DE LA MEMÒRIA

En el cos de la memòria es descriu el projecte. Està organitzat en els següents capítols:

- ◆ **Descripció del projecte:** Descripció de l'abast del projecte des d'un punt de vista de la seva funcionalitat.



- ◆ **Anàlisis i disseny de l'aplicació:** Descripció del funcionament dels casos d'ús identificats, així com els diagrames de seqüència i d'activitats dels casos d'ús més rellevants. Definició del model d'entitats i el disseny de la Base de dades que permet la persistència de les dades necessàries. Disseny de l'arquitectura tecnològica utilitzada en el projecte que permetrà la implementació dels casos d'ús i facilitin un cicle de vida del software equilibrat mitjançant una separació de capes ben marcada.
- ◆ **Implementació:** S'expliquen les decisions preses per a la implementació del projecte i el manual d'instal·lació.
- ◆ **Conclusions i línies de futur:** conclusions obtingudes del projecte i explicació de l'eventual evolució de l'aplicació construïda en el projecte.

## 2 COS DE LA MEMÒRIA

### 2.1 DESCRIPCIÓ DEL PROJECTE

#### 2.1.1 DESCRIPCIÓ DE L'APLICACIÓ

La temàtica que tractaré en el present TFC serà una aplicació J2EE que permeti portar un control sobre l'**economia domèstica** de qualsevol persona o família.

Per a fer-ho, els usuaris podran accedir a través d'Internet al portal on podran registrar i classificar informació sobre la seva economia domèstica (moviments o transaccions monetàries) de forma ordenada per a poder posteriorment fer anàlítica de la despesa per conceptes i naturaleses.

El portal es vertebrarà a través del moviment, dada principal de l'aplicació. El moviment representa una transacció monetària, és a dir qualsevol despesa o ingrés que l'usuari vulgui considerar.

Qualsevol moviment que l'usuari registri, quedarà classificat segons dos atributs diferents: el **concepte** de la despesa/ingrés, que representa el centre de cost sobre el qual l'usuari vol controlar les seves despeses (per exemple Cotxe, Pis, Oci, etc.), i la seva **naturalesa**, que defineix el tipus de despesa segons la seva realitat (per exemple Gasolina, Restaurant, Roba, Llibres, etc.)

L'aplicació permetrà que l'usuari defineixi els seus propis Conceptes i Naturaleses segons la seva conveniència, tot i que es preveu que pugui existir un pre-configurat amb els Conceptes i Naturaleses més habituals per estalviar feina als usuaris.

A més a més, degut a que certs Conceptes de despeses només són aplicables a moviments de certa Naturalesa, hi haurà la possibilitat de restringir quines Naturaleses són vàlides per a cada Concepte, i el sistema ho validarà.

Totes les dades que es registren en l'aplicació queden associades de forma unívoca a un únic usuari. Aquest usuari és l'únic propietari d'aquestes dades, i per tant l'únic que les pot modificar o visualitzar.

Es consideren dos tipus d'usuaris: el propietari de les dades (usuari) i un administrador que serà capaç d'administrar els usuaris per a tal de poder desbloquejar un usuari bloquejat (per sobrepassar el límit d'intents d'accés fallits), inicialitzar passwords o donar de baixa usuaris, però que en cap cas podrà visualitzar les dades personals d'altres usuaris.

El projecte podria ser més extens: podria incloure en l'abast moltes funcionalitats addicionals, com per exemple un mòdul de pressupostació que permeti fer un seguiment de Real vs. Pressupost, un mòdul d'anàlisis gràfic, un mòdul de simulació davant escenaris (per exemple per a prendre decisions sobre endeutament, canvi de situació laboral,...) i un llarg etcètera.

De totes maneres, l'abast del projecte el mantindré controlat per poder centrar-me completament en l'entrega d'una funcionalitat definida amb una estructura que permeti facilitar el manteniment futur de l'aplicació i escalabilitat funcional per a agilitzar la inclusió de nous mòduls.

La funcionalitat de l'aplicació permetrà:

- ◆ **Gestió d'usuaris:** per a utilitzar l'aplicació serà necessari estar registrat. Un usuari només tindrà accés a les seves dades. Des de la plataforma es podrà:
  - ◆ Autenticar usuaris amb Usuari i Password.

- ◆ Donar d'alta nous usuaris.
- ◆ **Configuració de l'entorn:** Cada moviment o transacció monetària estarà compost de certes dades rellevants, com la data, l'import,...Per a classificar el moviment de cara al seu anàlisi posterior, s'utilitzen tres figures: la **naturalesa** de la despesa/ingrés (descriu què és la despesa/ingrés) el **concepte** de control (que indica sobre quin concepte s'imputa la despesa/ingrés) i el **compte** que representa el compte bancari o fictici de control, per a controlar don surten on van els diners. L'aplicació ha de permetre que cada usuari es configuri les naturaleses, conceptes i comptes que vulgui utilitzar per a classificar els seus moviments. Possibilitat: plantilla de naturaleses i conceptes per defecte en l'alta d'usuari per a facilitar l'ús.
  - ◆ Gestió de Naturaleses (alta, modificació, baixa)
  - ◆ Gestió de Conceptes (alta, modificació, baixa)
  - ◆ Gestió de Comptes (alta, modificació, baixa)
- ◆ **Registre de moviments:** Per a donar d'alta un moviment, s'haurà d'entrar les dades rellevants: dates, imports, naturalesa, concepte,... Ha de permetre l'alta, modificació i baixa. El sistema realitzarà les validacions sobre la classificació per assegurar que les dades guardades són coherents.
- ◆ **Llistat de moviments:** Ha de permetre treure llistats classificats segons els conceptes que vulgui l'usuari de forma flexible per a permetre un anàlisi de les dades (per exemple treure tots els moviments de l'any d'una naturalesa concreta). Possibilitat: Baixada del llistat de moviments a fitxer Excel.
- ◆ **Mòdul d'anàlisi gràfic:** Introduir gràfiques que facilitin la comprensió i anàlisi de les dades.

Altres funcionalitats futures que es podrien anar afegint (no incloses en l'abast del TFC):

- ◆ **Mòdul de pressupostos:** Mòdul que permeti introduir un pressupost i poder fer el seguiment de les despeses/ingressos en referència al pressupost.
- ◆ **Mòdul d'anàlisi gràfic:** Extendre la funcionalitat gràfica amb noves opcions orientades a facilitar als usuaris la comprensió i anàlisi de les dades.
- ◆ **Mòdul de simulació:** Que permeti simular diferents escenaris hipotètics que permetin a l'usuari fer estudis de viabilitat i que facilitin la presa de decisions, com per exemple la viabilitat d'endeutar-se.

### 2.1.2 NOM DE L'APLICACIÓ

Per a identificar l'aplicació, li he posat el nom **Ecodom**, que inclou les dues paraules que millor representen l'objectiu del projecte: economia domèstica.

## 2.2 ANÀLISIS I DISSENY DE L'APLICACIÓ

Per a realitzar l'anàlisi de l'aplicació, seguiré la següent metodologia:

1. Identificar els actors externs que interaccionen amb l'aplicació
2. Identificar els casos d'ús, que representen les funcionalitats vistes des dels usuaris que hi interactuen. El resultat d'aquest procés és el diagrama de Casos d'ús.
3. Per a cada un dels casos d'ús:
  - a. Fer el seu diagrama de seqüència corresponent. Aquest exercici permetrà identificar les classes necessàries a implementar i definir una estructura modular, reutilitzable i escalable funcionalment en un futur.
  - b. Seleccionar els casos d'ús clau de l'aplicació, i representar el diagrama d'activitats, que representa el fluxe lògic de negoci que implementa.
4. Identificar les classes (diagrama de classes) que representi el model estàtic de dades de l'aplicació.

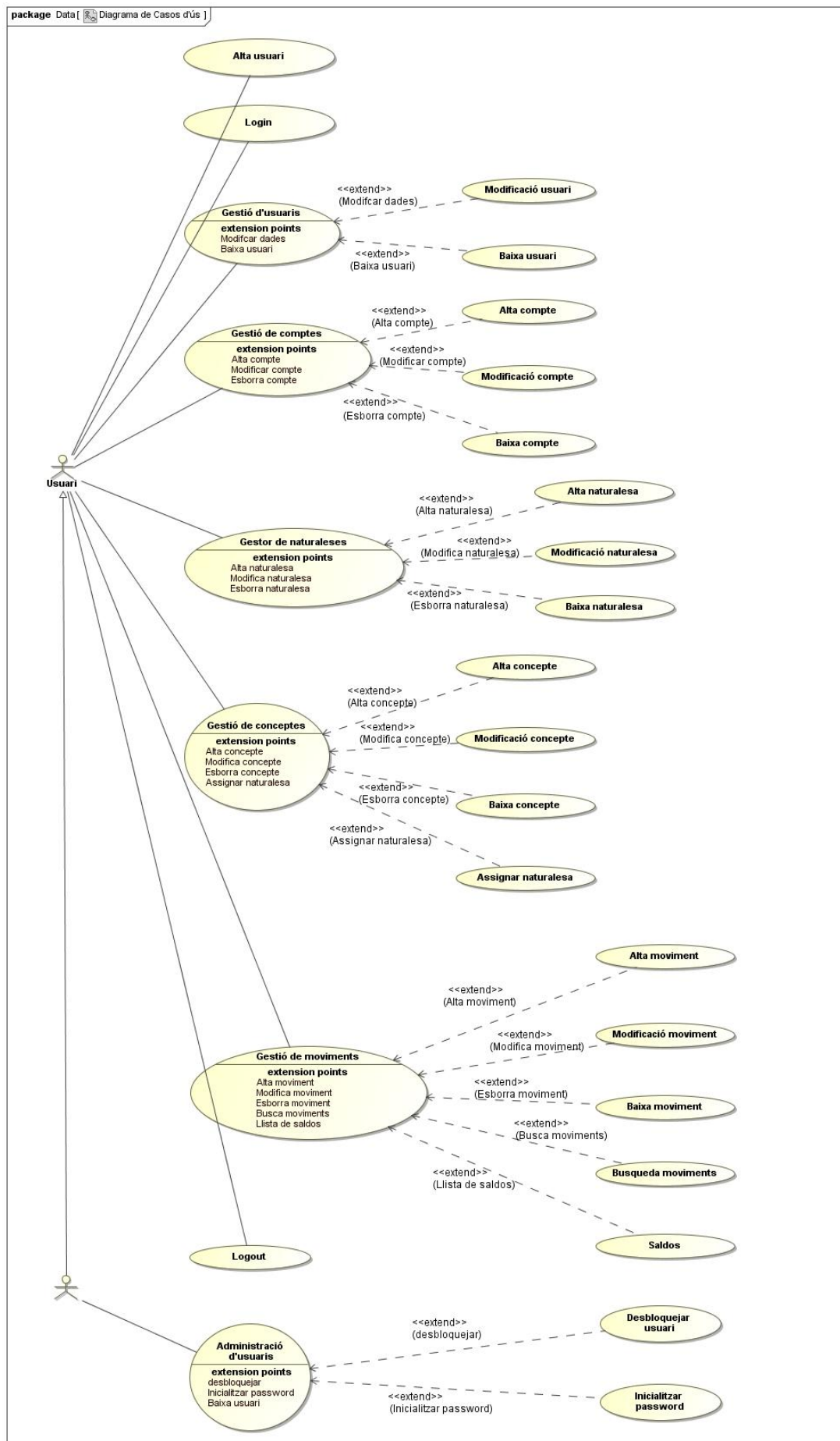
### 2.2.1 ACTORS

Es consideren dos actors diferenciats dins de l'aplicació:

- ◆ **Administrador:** usuari amb rol d'administrador. Podrà gestionar els seus moviments, com qualsevol usuari, però addicionalment podrà gestionar usuaris per a tal de desbloquejar-los (en cas d'haver sobrepassat el límit d'intents d'accés fallits),...
- ◆ **Usuari:** Representa el propi usuari de l'aplicació que fa de gestor de les seves propies dades. Per les característiques del procés, l'usuari és autosuficient en la gestió de les seves dades, i únicament requeriria l'intervenció de l'administrador per problemes amb l'accés a la plataforma.

### 2.2.2 CASOS D'ÚS

L'alcanç del projecte es limitarà als casos d'ús principals, per a garantir el compliment de les dates d'entrega, tot i que es preveu que en un futur s'hi podran incloure certs mòduls addicionals que dotarien de més funcionalitats.



### 2.2.2.1 ALTA USUARI

**Descripció:** Permet a un usuari no registrat donar-se d'alta al portal per a poder ser-ne usuari. Serà necessari entrar les següents dades:

- Nom
- Primer Cognom
- Segon Cognom
- DNI
- Password
- Repetició del password (per seguretat)

**Actors:** Usuari

**Casos d'ús relacionats:** Login, Logout, Modificació Usuari, Baixa Usuari

**Pre-condicions:** L'usuari no està registrat al portal.

**Post-condicions:** L'usuari està registrat al portal amb Usuari i Password que permet entrar (login) i tractar les seves dades privades.

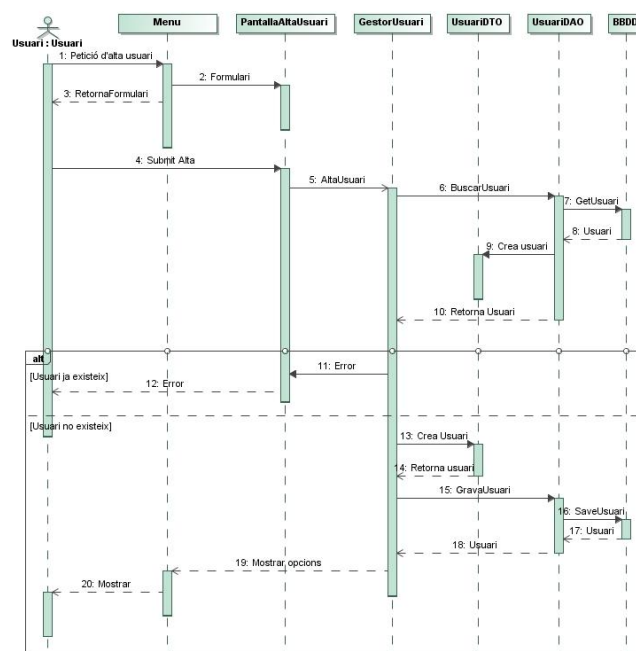
**Procés normal:**

1. L'usuari selecciona l'opció per registrar un nou usuari.
2. El sistema presenta el formulari per a que s'entren les dades rellevants (nom, cognoms, password,...)
3. L'usuari envia les dades omplertes (submit).
4. El sistema verifica que l'usuari no existeix i que els camps compleixen els criteris mínims (camps obligatoris,...)
5. El sistema registra el nou usuari i accedeix a la seva area de dades privades per a que pugui gestionar-les.

**Procés alternatiu:**

1. Si l'usuari ja existeix o les dades entrades no compleixen els criteris mínims (camps obligatoris,...) l'usuari rebrà un missatge explicatiu de l'error sobre la mateixa pantalla del formulari.

**Diagrama de seqüència:**



### 2.2.2.2 LOGIN

**Descripció:** Permet als usuaris autenticar-se per a tenir accés a les funcionalitats de la plataforma i tenir accés a les seves dades privades.

**Actors:** Usuari

**Casos d'ús relacionats:** Modificació Usuari, Baixa Usuari, Gestió de Comptes, Gestió de Conceptes, Gestió de Naturaleses, Gestió de Moviments

**Pre-condició:** L'usuari encara no s'ha autenticat al sistema i ha entrat a la pantalla inicial que li permetrà autenticar-se.

**Post-condició:** L'usuari s'ha autenticat i té accés a les seves dades privades.

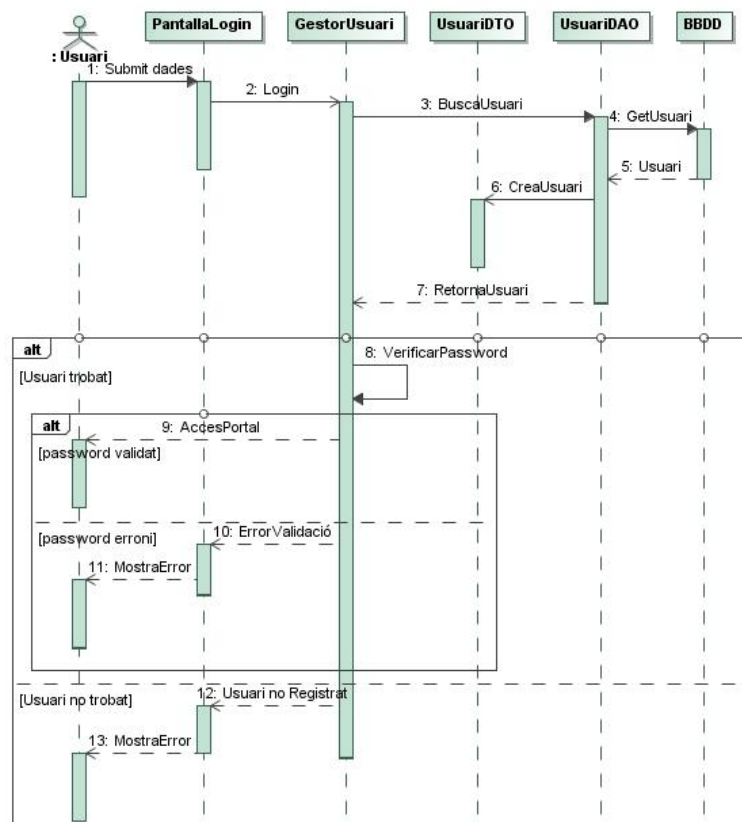
**Procés normal:**

1. El sistema demana les dades a l'usuari per a que s'autentifiqui (nom de l'usuari i password)
2. L'usuari entra les dades i accepta
3. El sistema comprova la validesa de l'Usuari-password i dona accés o el denega.
4. El sistema dóna accés al portal i presenta les diferents opcions a l'usuari.

**Procés alternatiu:**

1. En cas que les dades d'usuari-password no siguin validades, el sistema:
  - a. Mostrarà un missatge d'error
  - b. Retornarà a la pantalla anterior per a que pugui reintentar-ho
2. En cas de tres intents fallits, l'usuari s'ha de bloquejar.

**Diagrama de seqüència:**



### 2.2.2.3 LOGOUT

**Descripció:** Permet als usuaris tancar la sessió oberta en haver-se autenticat prèviament amb usuari i password.

**Actors:** Usuari

**Casos d'ús relacionats:** Login

**Pre-condició:** L'usuari s'ha autenticat al sistema

**Post-condició:** L'usuari ha tancat la sessió.

**Procés normal:**

1. L'usuari selecciona l'opció de sortir de la sessió.
2. El sistema comprova tanca la sessió i torna a la pantalla inicial.



#### 2.2.2.4 MODIFICACIÓ USUARI

**Descripció:** Un usuari modifica les seves dades guardades a la Base de dades (Nom, Cognom,...)

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió d'Usuaris

**Pre-condicions:** L'Usuari està creat a la Base de dades i l'usuari s'ha autenticat.

**Post-condicions:** S'ha modificat les dades de l'Usuari a la Base de dades.

**Procés normal:**

1. El sistema mostra les dades de l'usuari per a que es modifiquin.
2. L'usuari modifica la informació i accepta
3. L'Usuari es modifica a la Base de dades

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.5 BAIXA D'USUARI

**Descripció:** Un usuari dona de baixa el seu Usuari.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Gestió d'Usuaris

**Pre-condicions:** L'Usuari està creat a la Base de dades i s'ha autenticat.

**Post-condicions:** S'ha donat de baixa l'Usuari al sistema. Si l'executa l'actor Usuari, només podrà donar de baixa el seu propi usuari. En canvi, si l'executa l'actor Administrador, podrà donar de baixa qualsevol usuari.

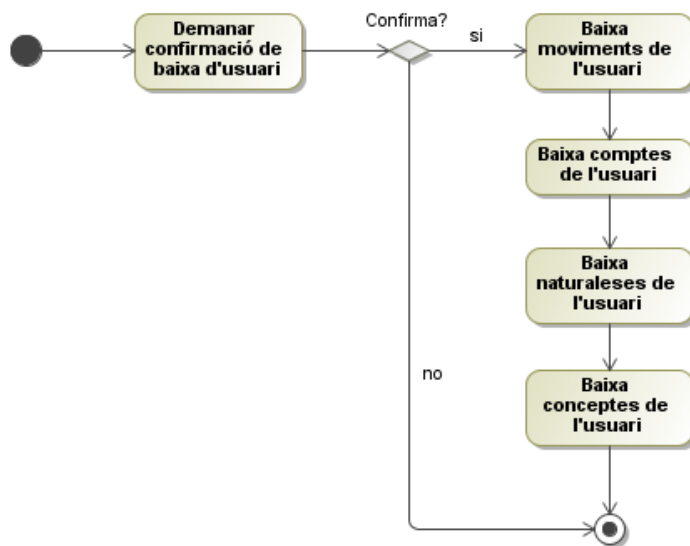
**Procés normal:**

1. El sistema mostra una pantalla per a confirmar la baixa de l'usuari i totes les dades associades.
2. L'usuari accepta
3. El sistema esborra totes les dades associades a l'usuari, incloent:
  - a. Moviments
  - b. Comptes
  - c. Conceptes
  - d. Naturaleses
  - e. Usuari

**Procés alternatiu:**

1. En cas de cancel·lar l'acció, es torna a la pantalla anterior.

**Diagrama d'activitats:**



#### 2.2.2.6 DESBLOQUEJAR USUARI

**Descripció:** L'administrador desbloqueja un usuari bloquejat.

**Actors:** Administrador

**Casos d'ús relacionats:** Login

**Pre-condicions:** L'Usuari està creat a la Base de dades i està bloquejat.

**Post-condicions:** L'usuari s'ha desbloquejat i permet tornar a entrar al sistema.

**Procés normal:**

1. L'administrador selecciona l'opció d'administrar usuaris
2. L'administrador selecciona l'usuari bloquejat
3. L'administrador desbloqueja l'usuari, que tornarà a disposar de 3 intents de Login.

**Procés alternatiu:**

#### 2.2.2.7 INICIALITZAR PASSWORD

**Descripció:** L'administrador modifica el password d'un Usuari.

**Actors:** Administrador

**Casos d'ús relacionats:** Login

**Pre-condicions:** L'Usuari està creat a la Base de Dades.

**Post-condicions:** El password de l'Usuari s'ha modificat manualment al valor indicat per l'administrador.

**Procés normal:**

1. L'administrador selecciona l'opció d'administrar els usuaris
2. L'administrador selecciona l'Usuari
3. L'administrador selecciona l'opció d'inicialitzar el password
4. Indica el password (i el confirma)

**Procés alternatiu:**

#### 2.2.2.8 ALTA DE COMPTE

**Descripció:** Un usuari accedeix a l'opció per a donar d'alta un Compte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Comptes

**Pre-condicions:** l'usuari està autèntificat

**Post-condicions:** S'ha donat d'alta un Compte nou al sistema assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra un formulari d'entrada de dades necessaries per a donar d'alta el Compte.
2. L'usuari completa la informació i accepta
3. El Compte es crea en el sistema associat a l'Usuari

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.9 MODIFICACIÓ DE COMPTE

**Descripció:** Un usuari accedeix a l'opció per a modificar un Compte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Comptes

**Pre-condicions:** El Compte existeix i l'usuari està autèntificat.

**Post-condicions:** S'ha modificat el Compte assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Comptes assignats a l'Usuari autèntificat.
2. L'usuari selecciona el Compte que vol modificar.
3. El sistema presenta la pantalla de modificació de Comptes (només permet modificar certes dades
4. L'usuari modifica les dades i accepta
5. El sistema guarda les modificacions a la Base de dades.

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.10 BAIXA DE COMPTE

**Descripció:** Un usuari accedeix a l'opció per a donar de baixa un Compte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Comptes

**Pre-condicions:** El Compte existeix i l'usuari està autèntificat.

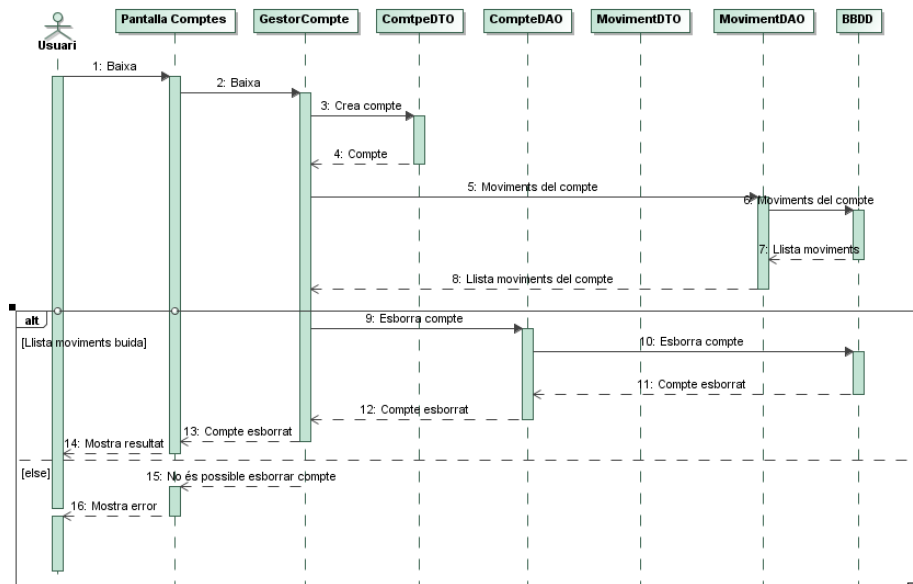
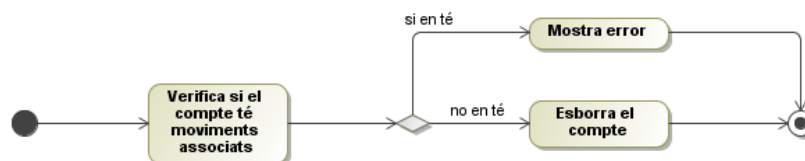
**Post-condicions:** S'ha esborrat el Compte assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Comptes assignats a l'Usuari autenticat.
2. L'usuari selecciona el Compte que vol esborrar.
3. El sistema verifica que no hi ha cap moviment amb aquest Compte.
4. El sistema esborra el Compte de la Base de dades.

**Procés alternatiu:**

1. Si hi ha Moviments amb el Compte que es vol esborrar:
  - a. Es mostra l'error a l'usuari indicant que existeixen moviments
  - b. Es torna a mostrar la pantalla anterior.

**Diagrama de seqüència:****Diagrama d'activitats:****2.2.2.11 ALTA DE NATURALES**

**Descripció:** Un usuari accedeix a l'opció per a donar d'alta una Naturala a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Naturaleses

**Pre-condicions:** l'usuari està autenticat

**Post-condicions:** S'ha donat d'alta una Naturala nova al sistema assignada a l'Usuari.

**Procés normal:**

1. El sistema mostra un formulari d'entrada de dades necessaries per a donar d'alta la Naturala.
2. L'usuari completa la informació i accepta
3. La Naturala es crea en el sistema
4. S'actualitzen les relacions d'aquesta nova Naturala amb tots els Conceptes del sistema (inicialment totes inactives).

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - c. Es mostra l'error a l'usuari
  - d. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.12 MODIFICACIÓ DE NATURESA

**Descripció:** Un usuari accedeix a l'opció per a modificar una Naturalesa a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Naturaleses

**Pre-condicions:** La Naturalesa existeix i l'usuari està autènticat.

**Post-condicions:** S'ha modificat la Naturalesa assignada a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Naturaleses assignades a l'Usuari autènticat.
2. L'usuari selecciona la Naturalesa que vol modificar.
3. El sistema presenta la pantalla de modificació de naturaleses (només permet modificar la descripció)
4. L'usuari modifica les dades i accepta
5. El sistema guarda les modificacions a la Base de dades.

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.13 BAIXA DE NATURESA

**Descripció:** Un usuari accedeix a l'opció per a donar de baixa una Naturalesa a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Naturaleses

**Pre-condicions:** La Naturalesa existeix i l'usuari està autènticat.

**Post-condicions:** S'ha esborrat la Naturalesa assignada a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Naturaleses assignades a l'Usuari autènticat.
2. L'usuari selecciona la Naturalesa que vol esborrar.
3. El sistema verifica que no hi ha cap moviment amb aquesta Naturalesa.
4. El sistema esborra la Naturalesa de la Base de dades.
5. El sistema esborra les assignacions de la Naturalesa a tots els Conceptes de l'usuari.

**Procés alternatiu:**

1. Si hi ha Moviments amb la Naturalesa que es vol esborrar:
  - a. Es mostra l'error a l'usuari indicant que existeixen moviments
  - b. Es torna a mostrar la pantalla anterior.

#### 2.2.2.14 ALTA DE CONCEPTE

**Descripció:** Un usuari accedeix a l'opció per a donar d'alta un Concepte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Conceptes

**Pre-condicions:** l'usuari està autenticat

**Post-condicions:** S'ha donat d'alta un Concepte nou al sistema assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra un formulari d'entrada de dades necessaries per a donar d'alta el Concepte.
2. L'usuari completa la informació i accepta
3. El Concepte es crea en el sistema
4. S'actualitzen les relacions d'aquest nou Concepte amb totes les Naturaleses del sistema (inicialment totes inactives).

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.15 MODIFICACIÓ DE CONCEPTE

**Descripció:** Un usuari accedeix a l'opció per a modificar un Concepte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Conceptes

**Pre-condicions:** El Concepte existeix i l'usuari està autenticat.

**Post-condicions:** S'ha modificat el Concepte assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Conceptes assignats a l'Usuari autenticat.
2. L'usuari selecciona el Concepte que vol modificar.
3. El sistema presenta la pantalla de modificació de Concepte (només permet modificar la descripció).
4. L'usuari modifica les dades i accepta
5. El sistema guarda les modificacions a la Base de dades.

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.16 BAIXA DE CONCEPTE

**Descripció:** Un usuari accedeix a l'opció per a donar de baixa un Concepte a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Conceptes

**Pre-condicions:** El Concepte existeix i l'usuari està autenticat.

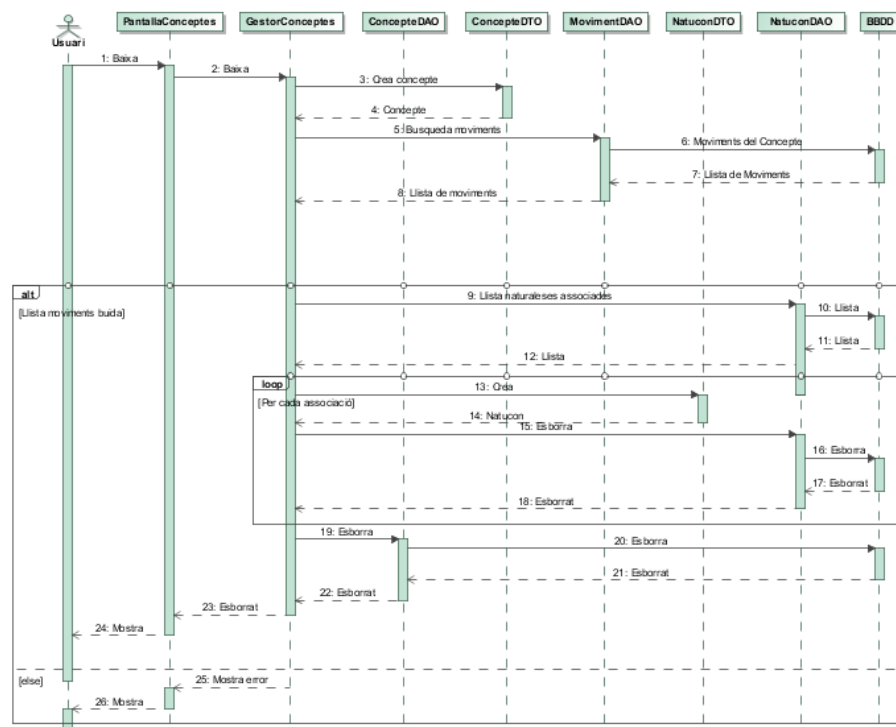
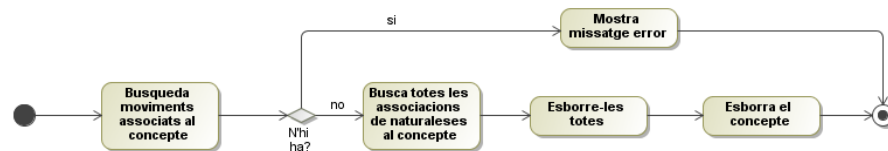
**Post-condicions:** S'ha esborrat el Concepte assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la llista de Conceptes assignats a l'Usuari autènticat.
2. L'usuari selecciona el Concepte que vol esborrar.
3. El sistema verifica que no hi ha cap moviment classificat amb aquest Concepte.
4. El sistema esborra l'assignació de naturaleses al concepte.
5. El sistema esborra el Concepte de la la Base de dades.

**Procés alternatiu:**

1. Si hi ha Moviments amb el Concepte que es vol esborrar:
  - a. Es mostra l'error a l'usuari indicant que existeixen moviments
  - b. Es torna a mostrar la pantalla anterior.

**Diagrama de seqüència:****Diagrama d'activitats:****2.2.2.17 ASSIGNAR NATURALES A CONCEPTE**

**Descripció:** Assignar les Naturaleses de despeses que es poden imputar al Concepte de cost.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Conceptes

**Pre-condicions:** El Concepte i la Naturalesa existeix i l'usuari està autènticat.

**Post-condicions:** S'ha activat/desactivat la relació entre el Concepte i la Naturalesa.

**Procés normal:**

1. Es selecciona el Concepte

2. El sistema mostra una llista de Naturaleses de l'Usuari.
3. L'usuari activa/desactiva l'assignació de cada Naturalesa al concepte..
  - a. Si s'està desactivant, s'ha de verificar que no hi hagi cap moviment amb la relació Concepte – Naturalesa a desactivar abans de procedir.
4. El sistema actualitza la relació Concepte-Naturalesa a la Base de dades.

**Procés alternatiu:**

1. Si hi ha Moviments amb la relació Concepte-Naturalesa a desactivar:
  - a. Es mostra l'error a l'usuari indicant que existeixen moviments
  - b. Es torna a mostrar la pantalla anterior.

#### 2.2.2.18 ALTA DE MOVIMENT

**Descripció:** Un usuari accedeix a l'opció per a donar d'alta un Moviment a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Moviments

**Pre-condicions:** Usuari autenticat

**Post-condicions:** S'ha donat d'alta un Moviment nou al sistema assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra un formulari d'entrada de dades necessaries per a donar d'alta el Concepte.
  - a. S'informa dels camps (import, comentaris,...)
  - b. Selecció del compte origen (valida que sigui un dels donats d'alta per l'Usuari)
  - c. Selecció del Concepte (valida que sigui un dels donats d'alta per l'Usuari)
  - d. Selecció de la Naturalesa (valida pel concepte seleccionat, és a dir, que estigui activa l'assignació de la Naturalesa al Concepte)
2. El sistema valida les dades introduïdes
3. El Moviment es dona d'alta a la Base de dades

**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

#### 2.2.2.19 MODIFICACIÓ DE MOVIMENT

**Descripció:** Un usuari accedeix a l'opció per a modificar un Moviment a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** de Moviments

**Pre-condicions:** El Moviment existeix i l'usuari està autenticat.

**Post-condicions:** S'ha modificat el Moviment assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la pantalla de búsqueda o selecció de moviments
2. L'usuari entra els criteris de búsqueda i accepta.
3. El sistema mostra la llista de Moviments assignats a l'Usuari autenticat que compleixen els criteris de búsqueda anteriors.
4. L'usuari selecciona el Moviment que vol modificar.
5. El sistema presenta la pantalla de modificació del Moviment.
6. L'usuari modifica les dades i accepta
7. El sistema guarda les modificacions a la Base de dades.



**Procés alternatiu:**

1. El sistema valida les dades entrades per l'usuari abans de guardar-les a la Base de dades. En cas que no sigui vàlida:
  - a. Es mostra l'error a l'usuari
  - b. Es torna a mostrar la pantalla de dades per a que es corregeixi

**2.2.2.20 BAIXA DE MOVIMENT**

**Descripció:** Un usuari accedeix a l'opció per a donar de baixa un Moviment a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Moviments

**Pre-condicions:** El Moviment existeix i l'usuari està autenticat.

**Post-condicions:** S'ha esborrat el Moviment assignat a l'Usuari.

**Procés normal:**

1. El sistema mostra la pantalla de búsqueda o selecció de moviments
2. L'usuari entra els criteris de búsqueda i accepta.
3. El sistema mostra la llista de Moviments assignats a l'Usuari autenticat que compleixen els criteris de búsqueda anteriors.
4. L'usuari selecciona el Moviment que vol esborrar.
5. El sistema esborra el Moviment de la Base de dades.

**Procés alternatiu:**

**2.2.2.21 LLISTA DE MOVIMENTS**

**Descripció:** Un usuari accedeix a l'opció per a llistar Moviments a la Base de dades del sistema.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Moviments

**Pre-condicions:** l'usuari està autenticat.

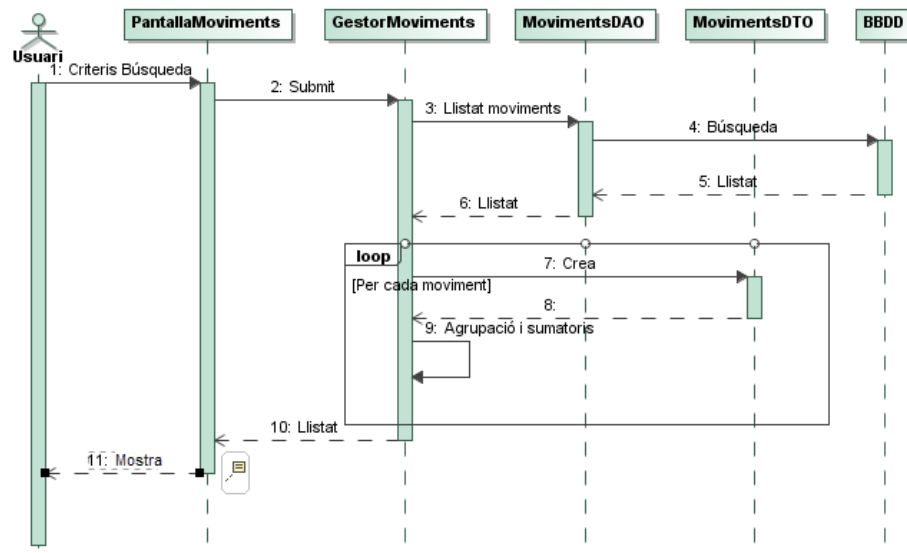
**Post-condicions:** S'ha llistat els Moviments assignats a l'Usuari.

**Procés normal:**

1. El sistema mostra la pantalla de búsqueda o selecció de moviments (Serà possible seleccionar per Compte, Naturalesa, Concepte de cost, data valor, data operació i import)
2. L'usuari entra els criteris de búsqueda i accepta.
3. El sistema mostra la llista de Moviments assignats a l'Usuari autenticat que compleixen els criteris de búsqueda anteriors.

**Procés alternatiu:**

**Diagrama de seqüència:**



#### 2.2.2.22 LLISTA DE SALDOS

**Descripció:** Un usuari accedeix a l'opció per a llistar Saldos.

**Actors:** Usuari

**Casos d'ús relacionats:** Gestió de Moviments

**Pre-condicions:** l'usuari està autenticat.

**Post-condicions:** S'ha calculat i llistat els Saldos de tots els Comptes assignats a l'Usuari a una data concreta.

**Procés normal:**

1. El sistema busca tots els comptes de l'usuari
2. El sistema busca tots els moviments de l'usuari per a cada compte
3. El sistema suma els imports dels moviments per a calcular el Saldo de cada Compte i el mostra per pantalla.

**Procés alternatiu:**

#### 2.2.2.23 BAIXADA DE LLISTAT A FITXER EXCEL

**Descripció:** L'usuari es baixa el llistat seleccionat en la pantalla a un fitxer Excel.

**Actors:** Usuari

**Casos d'ús relacionats:** Llista de Moviments, Llista de Saldos

**Pre-condicions:** L'usuari està autenticat i està visualitzant una llista.

**Post-condicions:** S'ha generat un fitxer Excel amb el contingut del llistat i s'ha guardat a la ruta indicada per l'usuari.

**Procés normal:**

1. L'usuari selecciona l'opció de baixar el llistat a un fitxer Excel des de la pantalla del llistat.
2. L'usuari indica la ruta on guardar el fitxer quan el sistema ho demana.
3. El fitxer es guarda.

**Procés alternatiu:**

#### 2.2.2.24 INFORME GRÀFIC DE SALDOS

**Descripció:** L'usuari visualitza gràficament els saldos que té disponibles en els diferents comptes definits.

**Actors:** Usuari

**Casos d'ús relacionats:** Llista de Saldos

**Pre-condicions:** L'usuari està autènticat.

**Post-condicions:** Es visualitza un informe gràfic de barres dels saldos de cada compte de l'usuari.

**Procés normal:**

1. L'usuari selecciona l'opció de visualitzar gràfic de saldos
2. El sistema calcula els saldos de cada compte.
3. El sistema visualitza per pantalla el gràfic de barres amb la informació recuperada.

**Procés alternatiu:**

#### 2.2.2.25 INFORME GRÀFIC DE DESPESES/INGRESSOS

**Descripció:** L'usuari visualitza gràficament els moviments en forma gràfica i la seva evolució temporal agrupats per la seva classificació de compte, concepte o naturalesa.

**Actors:** Usuari

**Casos d'ús relacionats:** Llista de Moviments

**Pre-condicions:** L'usuari està autènticat.

**Post-condicions:** Es visualitza un informe gràfic dels moviments agrupats pel concepte triat per l'usuari

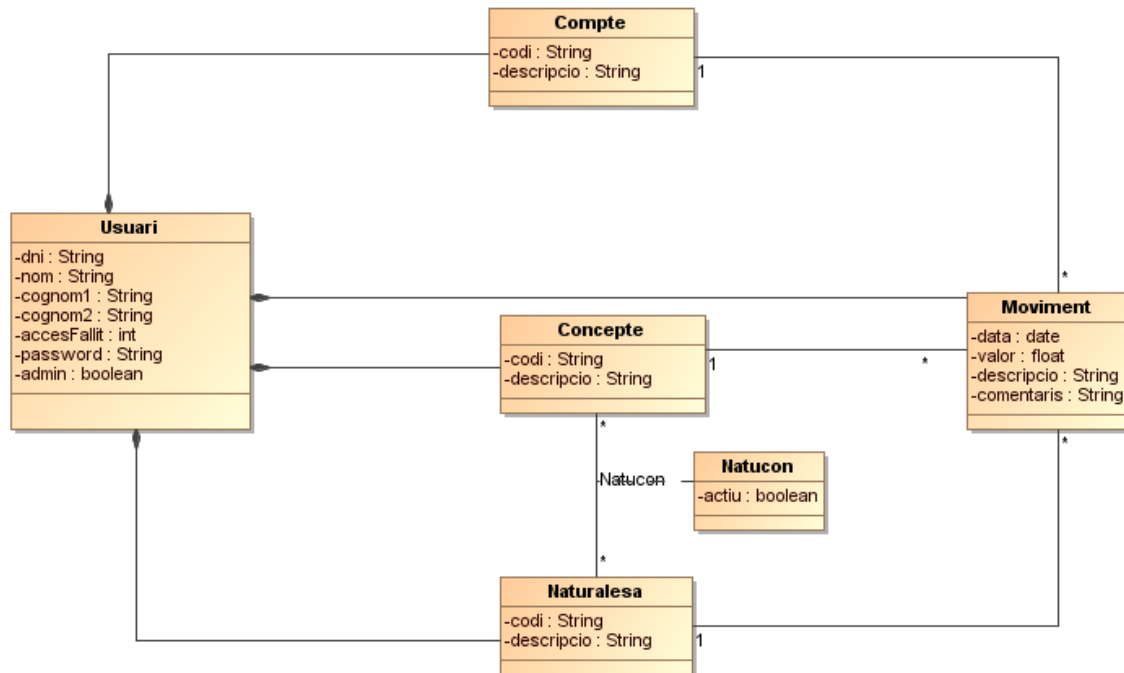
**Procés normal:**

1. L'usuari selecciona l'opció de visualitzar gràfic de moviments
2. L'usuari selecciona els criteris de selecció i agrupació
3. El sistema calcula les dades a mostrar
4. El sistema visualitza per pantalla el gràfic amb la informació recuperada.

**Procés alternatiu:**

### 2.2.3 DIAGRAMA DE CLASSES

En aquest apartat es mostra el model de Classes UML que representaran les entitats necessàries del projecte:



Comentaris sobre el model:

- Totes les dades estan associades a un Usuari, ja que li son específiques, inclús la definició de Conceptes i Naturaleses.
- L'entitat bàsica de tot el model és el Moviment, que registra tota la informació a gestionar. La resta d'entitats son classificadors i de suport/validació.
- Natucon és l'entitat que relaciona les associacions Concepte i Naturalesa, que defineixen les Naturaleses que són acceptades per imputar-se a un Concepte.
- Per programari es controlarà que:
  - un moviment haurà d'estar associat a dades d'un únic usuari, ja que el model no ho garanteix per si sol.
  - La naturalesa i el concepte amb que es classifica un moviment qualsevol està actiu en la relació Natucon

El disseny de les taules a implantar seria:

Usuari (id\_usuari, dni, nom, cognom1, cognom2, password, accesFallit, admin)

Compte (id\_usuari, id\_compte, codi, descripció)

Concepte (id\_usuari, id\_con, descripció)

Naturalesa (id\_usuari, id\_nat, descripció, tipus)

Natucon (id\_usuari, id\_nat, id\_con, actiu)

Moviment (id\_usuari, id\_mov, id\_compte, id\_nat, id\_con, data, descripcio, valor, comentaris)

En cursiva indico les claus foranies del model relacional.

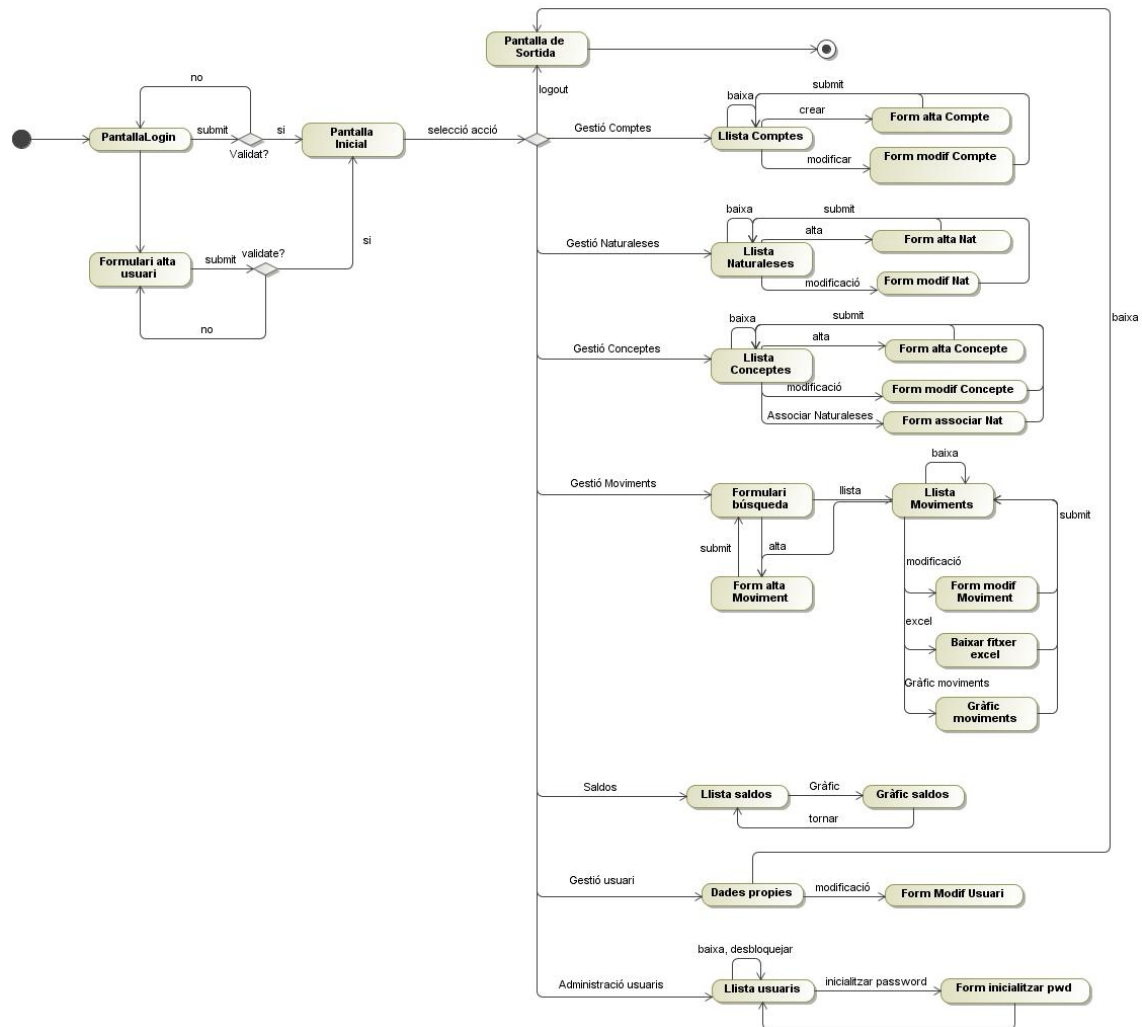
Consideracions:

S'ha utilitzat claus primàries sense significat (meaningless) per les següents raons:

- Independitzar la clau de les dades, de tal manera que no és necessari canviar la clau de en cas d'error en les dades significatives. Aquesta clau no s'utilitzarà per a fer búsquedes, en canvi es pot utilitzar per a comparar objectes de forma senzilla.
- S'utilitza la clàusula UNIQUE i NOT NULL per a definir claus alternatives.
- S'utilitza la clàusula FOREIGN KEY per a identificar les claus foranies en referència a altres taules.
- Els identificadors de les taules son del tipus integers (rang de 0 a 4.294.967.295) ja que considero que és suficient per la volumetria d'entrades podria requerir l'aplicatiu.

## 2.2.4 PROTOTIPUS DE L'INTERFÍCIE GRÀFICA

A continuació es mostra un diagrama de navegació de pantalles dins de l'aplicació:



Les pantalles es construïran mitjançant pàgines JSPs, que permeten generar el contingut de forma dinàmica en el servidor abans d'enviar-les al navegador.

A més a més, per a facilitar la composició de les diferents àrees de les pantalles (capçalera, peu, menú, cos,...) s'utilitza el framework de presentació TILES.

El projecte està provat en el navegador Mozilla Firefox 4.0.1, tot i que hauria de funcionar en la majoria, donat que es fan servir HTML estàndar.

La visualització està pensada per a pantalles amb una resolució mínima de 1024 x 600.

El prototipus de les possibles pantalles de l'aplicació (l'aparença pot canviar durant el procés de construcció):

L'aplicació tindrà una pantalla de benvinguda que permetrà que els usuaris s'autentifiquin o en cas de no estar registrats, que ho puguin fer.

Página principal - Windows Internet Explorer

C:\Users\Usuario\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO\PubWebPagePreview\pub4612.0\index.htm

Favoritos UOC - Universitat Oberta d... Página principal

**€ CODOM**

**ECODOM**

El portal per a gestionar les teves despeses

**Accés usuaris:**

Usuari

Password

---

En aquest portal podras gestionar totes les teves despeses, clasificar-les i portar-ne un control exhaustiu de cara a ajudar en la gestió de l'economia domèstica.

Si encara no n'ets usuari, crea un compte ara fent [click aquí](#).

Si es tracta d'un usuari que no té un compte i selecciona l'opció d'obrir-ne un, es mostrarà la següent pantalla per a que pugui completar les dades necessaries:

Página principal - Windows Internet Explorer

C:\Users\Usuario\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO\PubWebPagePreview\pub4612.0\index.htm

Favoritos UOC - Universitat Oberta d... Página principal

**€ CODOM**

**ECODOM**

El portal per a gestionar les teves despeses

**Creació d'un compte:**

Completi les dades següents:

Nom

Primer cognom

Segon Cognom

DNI

Password

Confirma el password

Quan un usuari s'autentifica, la primera pantalla que veurà per defecte serà la corresponent al **llistat de moviments**:

**ECODOM**  
El portal per a gestionar les teves despeses

**Selecció de moviments**

Compte

Concepte

Naturalesa

Data operació Desde  Fins

Data valor Desde  Fins

Import Desde  Fins

El **resultat de la selecció** de moviments es mostrarà en una pantalla nova que contingui una taula:

**ECODOM**  
El portal per a gestionar les teves despeses

**Moviments**

Compte	Data op.	Data valor	Descripció	Import	Concepte	Naturalesa	Comentaris

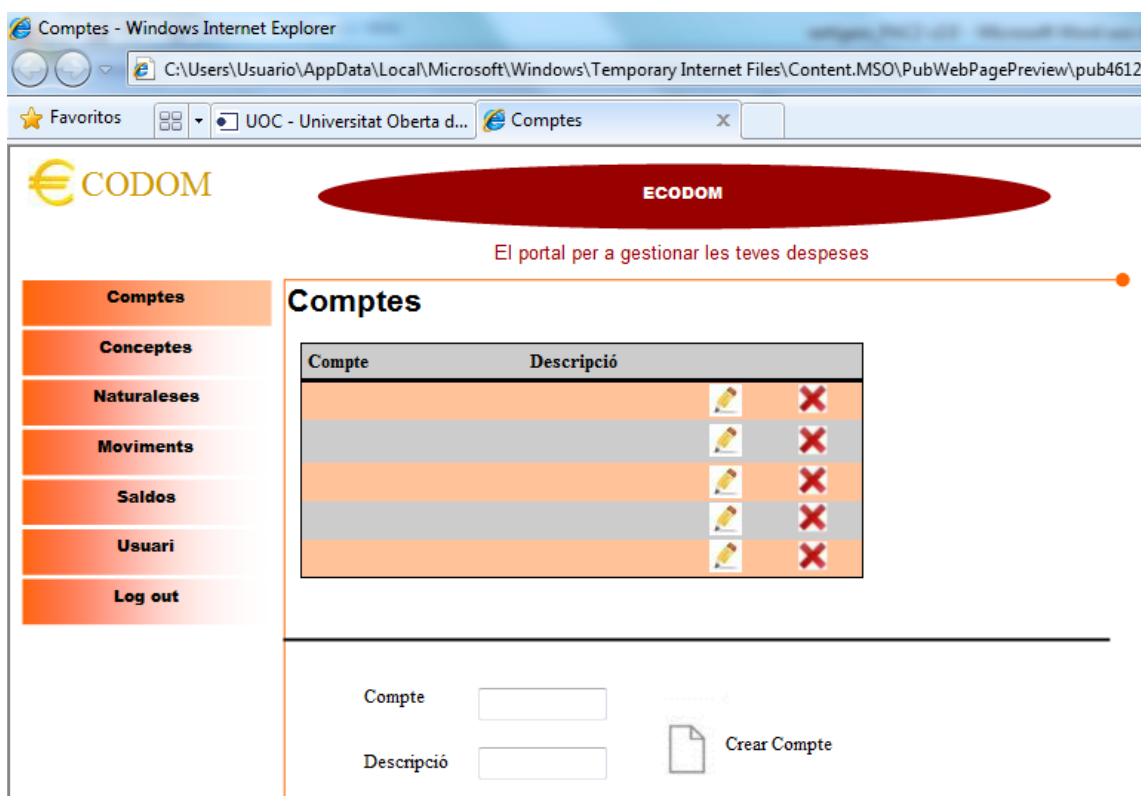
També podrà accedir al **llistat de saldos** de tots els seus comptes associats:



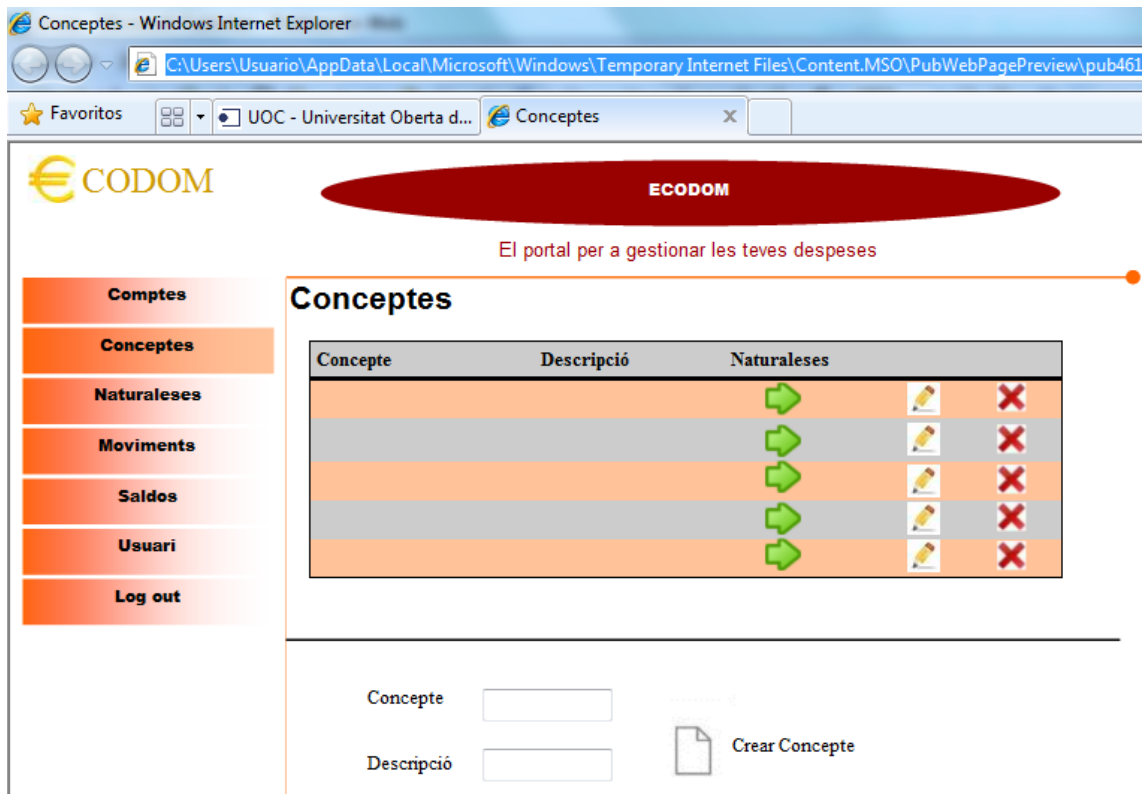


Les següents opcions de configuració:

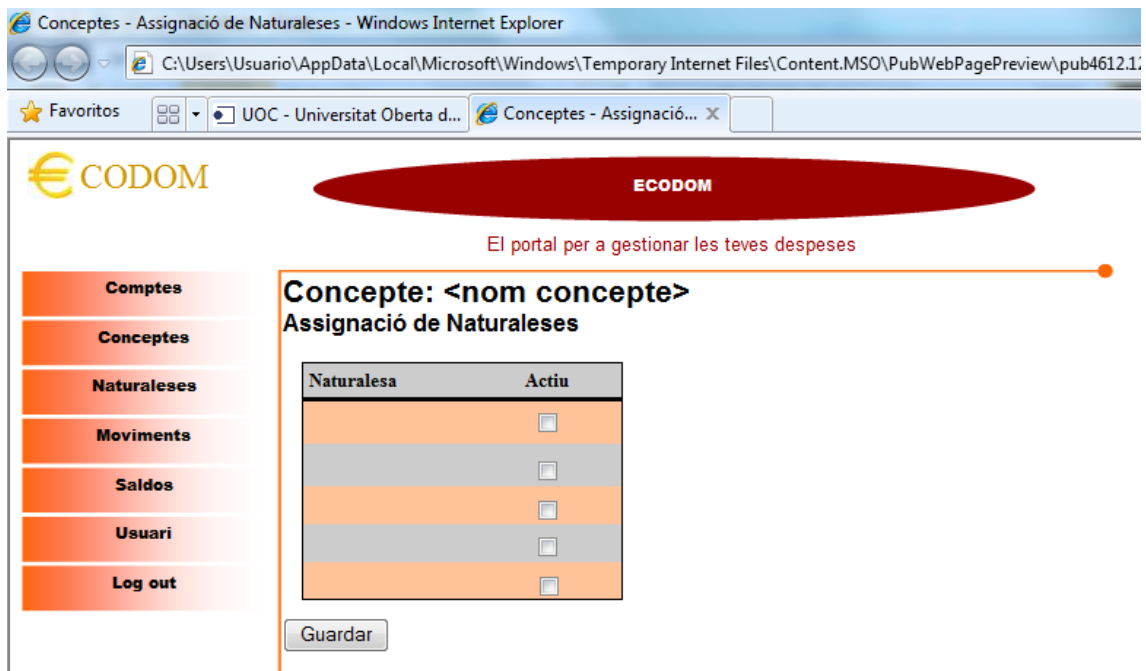
Gestió de **comptes**, on l'usuari podrà crear-ne de noves o modificar i esborrar les existents:



Gestió de **conceptes**, on l'usuari podrà crear-ne de noves o modificar i esborrar les existents:



Quan l'usuari seleccioni el botó de Naturaleses (fletxa verda) s'accedirà a la pantalla d'activació/desactivació de les naturaleses habilitades pel concepte triat:



Gestió de **naturaleses**, on l'usuari podrà crear-ne de noves o modificar i esborrar les existents:

Naturaleses - Windows Internet Explorer

C:\Users\Usuario\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO\PubWebPagePreview\pub4612...

Favoritos UOC - Universitat Oberta d... Naturaleses

**€ CODOM**

**ECODOM**

El portal per a gestionar les teves despeses

**Naturaleses**

Naturalesa	Descripció	Tipus

Naturalesa

Descripció

Crear Naturalesa

Gestió de l'usuari, on l'usuari podrà modificar les seves característiques (excepte en DNI, que és l'identificador). En cas d'esborrar el seu usuari, el sistema preguntarà si està segur de borrar totes les dades associades i haurà de confirmar-ho. Automaticament sortirà a la pantalla inicial (log out).

Usuari - Windows Internet Explorer

C:\Users\Usuario\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO\PubWebPagePreview\pub4612...

Favoritos UOC - Universitat Oberta d... Usuari

**€ CODOM**

**ECODOM**

El portal per a gestionar les teves despeses

**Usuari**

Nom	Primer Cognom	Segon Cognom	DNI

L'opció de sortir (log out) treu l'usuari de la zona privada i li mostra la pantalla d'inici per a tornar a fer el log in.

### 2.2.5 ARQUITECTURA DE L'APLICACIÓ

Pel desenvolupament del projecte s'utilitzarà la plataforma JEE seguint el paradigma del desenvolupament en tres capes: presentació, negoci i persistència de dades.

Aquesta arquitectura de separació en capes té grans avantatges respecte a la programació "anàrquica", ja que permet una millor estructuració dels programes, modularitzar, separar les funcions i com a conseqüència una major escalabilitat i facilitat en el manteniment. Els canvis necessaris es focalitzen en la capa afectada de forma que si, per exemple, es fes un canvi en la capa de presentació no afectaria a la capa de negoci.

La **capa de presentació** és la que s'ocupa de presentar i gestionar les pantalles que visualitza l'usuari de l'aplicació (recollint els events i les interaccions). Aquesta capa es comunica mitjançant un API amb la capa de negoci. Un bon disseny de la capa de presentació ajudarà molt a l'usabilitat i facilitat d'ús i per tant és part clau de l'èxit del projecte.

La **capa de negoci** és l'encarregada d'implementar la lògica del problema que resol el programa. Recull les dades que va rebent de la capa de presentació per a executar els algorismes corresponents i donar respostes a la capa de presentació.

Aquesta capa també es comunica amb la capa de persistència de dades, ja que per a aplicar certs algorismes sobre dades concretes ha de poder accedir-hi. Per a fer-ho utilitza una altra API de comunicació.

La **capa de persistència de dades** s'ocupa de l'emmagatzematge de dades per a assegurar la persistència en el temps de les mateixes. Aquesta capa interactua a través de l'API corresponent amb la capa de negoci segons les necessitats establertes per aquesta capa.



#### 2.2.5.1 CAPA DE PRESENTACIÓ

La capa de presentació és l'interfície a través de la qual els actors externs a la nostra aplicació interactuen amb ella.

En el nostre cas, serà l'encarregada de transmetre les peticions dels usuaris que fan des dels navegadors i comunicar els resultats d'aquestes peticions utilitzant diferents mecanismes establerts en la tecnologia web (imatges, text, menús, botons, checkboxes). Per a fer-ho, utilitzaré pàgines JSP i el framework de presentació Tiles par a ajudar a organitzar la navegabilitat entre les diferents pàgines.

Per a gestionar la interacció de la capa de Presentació amb la capa de Negoci (enviament d'events a la capa de Negoci per a que s'executi la lògica corresponent i retornar un resultat a la capa de presentació utilitzaré el framework Struts, que captura els events i els envia a través de les Actions a la capa de Negoci. Aquest framework implementa i segueix l'arquitectura Model-Vista-Controlador. Aquesta arquitectura encaixa amb el paradigma de la separació de l'aplicació en tres capes. Les pàgines JSPs implementaran les vistes, que es comunicaran amb les classes que encapsulen la lògica de negoci (Action), controlat pel fitxer de configuració struts-config.xml.

### 2.2.5.2 CAPA DE NEGOCI

Gràcies a que la capa de presentació es preocupa de la interacció amb l'usuari, la capa de negoci només es focalitza en la lògica sense preocupar-se de com es presentaran. Així mateix, també es despreocupa de com es guardaran les dades, ja que d'aquest tema se'n preocuparà la capa de persistència de dades. Les funcions d'aquesta capa és la de rebre les peticions enviades per la capa de presentació, executar la lògica de negoci (validacions, càlculs, estat) i obtenir i comunicar els resultats.

La capa de negoci és pròpia de cada aplicació (implementa els casos d'ús recollits en l'anàlisi), i per tant serà necessari desenvolupar-la completament per a resoldre el problema que es planteja solucionar amb el projecte.

### 2.2.5.3 CAPA DE PERSISTÈNCIA DE DADES

Gràcies a la capa de persistència de dades, la capa de negoci no necessita preocupar-se de com i on es guarden les dades. En el projecte s'utilitzarà una Base de dades per a guardar les dades que necessiten persistència. Aquesta base de dades és MySQL, àmpliament utilitzada i d'ús gratuït.

A més a més, també s'utilitzarà Hibernate com a framework, que permet configurar l'aplicació per a que la capa de Negoci tracti les dades persistents com si es tractés d'un objecte més. D'aquesta manera és completament transparent a l'aplicació quan està tractant amb dades guardades a la Base de dades, agilitzant i facilitant el desenvolupament.

## 2.2.6 ENTORN TECNOLÒGIC

El desenvolupament del projecte requereix l'ús d'un entorn tecnològic. A continuació es descriu el conjunt d'aplicacions utilitzats pel desenvolupament del projecte:

- **Servidor d'aplicacions:** servidor on s'executa l'aplicació J2EE. Pel projecte s'ha utilitzat Tomcat.
- **Servidor de Bases de dades:** servidor que gestiona la Base de dades que assegura la persistència de les dades. . Pel projecte s'ha utilitzat el servidor de BBDD MySQL 5.1.45.
- **Entorn IDE de desenvolupament:** entorn que facilita el desenvolupament d'aplicacions en tecnologia JEE, debugar,... . Pel projecte s'ha utilitzat Eclipse IDE for Java EE Developers 3.1.2.
- **JDK (Java Development Kit):** inclou la màquina virtual (JRE – Java Runtime Environment) i les eines necessàries pel programador, com el compilador de Java. . Pel projecte s'ha utilitzat la versió 6.
- **Framework de desenvolupament:** Framework que facilita el desenvolupament sobre les llibreries estàndard de Java. Ajuden per exemple a la programació seguint el paradigma de la separació en tres capes MVC (Model – Vista – Controlador). . Pel projecte s'ha utilitzat Struts 2.1.8.1.
- **Framework de persistència de dades:** Framework que facilita la interacció de l'aplicació amb la base de dades, tant per fer consultes com per a insertar-les o modificar-les. . Pel projecte s'ha utilitzat Hibernate 3.5.0-CR-2 release.
- **Connector de Base de dades:** Necessari per a accedir des de l'aplicació a la BBDD. . Pel projecte s'ha utilitzat el JDBC mysql-connector-java-5.1.10.
- **Entorn d'administració de BBDD:** Facilita la creació i administració de la Base de dades. Utilitzaré MySQL GUI Tools i Hibernate tools (plugin d'Eclipse que facilita la generació a partir d'una base de dades, dels POJOs (o DTOs) i dels fitxers de mapeig necessaris per Hibernate.
- **Hardware:** Ordinador que permeti l'execució dels servidors i la resta d'aplicacions de desenvolupament. Utilitzaré el meu portàtil, amb el Sistema Operatiu Windows 7 Home

Premium de 64 bits, de 4 GB de memòria RAM i un processador AMD Turion Dual-Core 2,30 GHz .

---

## 2.3 IMPLEMENTACIÓ

Durant la implementació del projecte ha estat necessari anar prenent decisions tècniques i prenent decisions per escollir entre les diferents alternatives per a anar assolint els diferents objectius.

A continuació detallo les decisions més rellevants preses en el projecte en cada una de les capes de l'aplicació:

### 2.3.1 CAPA DE PRESENTACIÓ

Les pantalles s'han construït mitjançant pàgines JSPs, que permeten generar el contingut de forma dinàmica en el servidor abans d'enviar-les al navegador.

Una dificultat important en un projecte d'aquestes característiques es gestionar les diferents arees de cada pantalla d'usuari, de forma que sigui senzill gestionar el contingut de les mateixes. Una opció era utilitzar navegació entre pàgines, però es presenta un gran problema: implica repetir a cada JSP cada una de les arees de visualització: la capçalera, el menú, el peu, el cos,... dificultant enormement el manteniment posterior: qualsevol canvi implica modificar un conjunt de JSPs.

Per a solucionar aquest problema vaig decidir utilitzar el **framework Tiles**, que permet fer una gestió dels continguts de cada pantalla utilitzant plantilles configurables. Aquesta elecció ha facilitat clarament l'implementació de les pantalles minimitzant la redundància i facilitant totes les modificacions.

Una altra objectiu del projecte era l'inclusió de **gràfiques** dins de les pàgines de visualització per a donar valor i facilitar la comprensió de les dades registrades a la plataforma. Amb aquest objectiu vaig provar d'utilitzar JASPER Reports com a eina per a generar aquestes gràfiques. Finalment, el fet de la falta de disponibilitat de temps va fer-me decidir a utilitzar una altra alternativa: google charts. Es tracta d'un servei publicat per Google que mitjançant l'enviament de paràmetres a través de paràmetres a la URL, retorna una imatge amb el gràfic demanat. Per a integrar-ho en el projecte s'usa una llibreria (charts4j) que genera les URLs de forma senzilla i agilitza la generació d'aquests gràfics. Per contra cal tenir en compte que únicament funcionarà sempre i quan existeixi una connexió a Internet que permeti fer ús del servei de Google Charts.

Per a facilitar l'usabilitat de l'aplicació, també utilitzo **components AJAX** amb autocompletables, que permeten actualitzar valors buscats validant les opcions contra la Base de dades sense necessitat de carregar tota la pàgina. Per a fer-ho utilitzo llistats JSON retornats pels Actions definits.

Un tema rellevant també en l'implementació de l'aplicació és el de la **seguretat**. L'aplicació ha de garantir que un usuari únicament pot accedir a les dades a les que té autorització. En aquest sentit he definit un control d'accés a través d'un interceptor específic pel projecte que per a cada request fet des d'un client a l'aplicació revisa si està accedint a una zona privada sense estar autenticat. El fet que l'interceptor estigui en l'stack de totes les possibles actions del projecte fa que centralitzi tot el control de forma neta i senzilla. També es controla que un usuari que no és administrador no pugui accedir a les opcions d'administrar usuaris de l'aplicació.

Adicionalment, s'ha ubicat totes les JSPs del projecte dins de la carpeta privada WEB-INF de tal manera que servidor no permetrà l'accés directe a la JSP des del navegador Web i d'aquesta manera prevenir

l'accés directe per URL i accedir-hi únicament a través del controlador que té el control dels accessos implementat.

El projecte està provat en el navegador Mozilla Firefox 4.0.1, tot i que hauria de funcionar en la majoria, donat que es fan servir HTML estàndar. Podria ser que la visualització canviï lleugerament d'un navegador a un altre.

La visualització està pensada per a pantalles amb una resolució mínima de 1024 x 600.

### 2.3.2 CAPA DE NEGOCI

En la capa de negoci es presenten també dificultats per a gestionar els diferents casos d'ús i com es comunica amb la capa de presentació.

Per a solucionar i facilitar aquesta tasca es poden utilitzar diferents frameworks de mercat. En el projecte he utilitzat el **framework Struts**, que captura els diferents events que es van generant des de la capa de presentació i els envia a través d'Actions a la capa de Negoci. Aquest framework implementa i segueix l'arquitectura Model-Vista-Controlador. A més a més aquesta arquitectura potencia el paradigma de la separació de l'aplicació en tres capes: les pàgines JSPs implementaran les vistes, que es comuniquen amb les classes que encapsulen la lògica de negoci (Action), controlat pel fitxer de configuració struts-config.xml.

Per a facilitar el registre i coneixement del que està executant el projecte també he optat per utilitzar un framework de registre de logs molt estès com és el **log4j**. Aquest framework permet registrar diferents nivells de logs de forma jeràrquica per a que es defineixi en el fitxer de propietats quin nivell de detall de logs és el desitjat en cada moment. També permet definir el destí dels logs que pot ser configurat (fitxer, consola de sortida estàndard,...).

Adicionalment, per a la generació de fitxers Excels també ha estat necessari l'ús de **la llibreria POI** per a reutilitzar les funcionalitats de generació de fitxers.

### 2.3.3 CAPA DE PERSISTÈNCIA

Aquesta capa, com a responsable d'assegurar la persistència de les dades necessàries per a l'aplicació, requereix d'una base de dades. En el projecte s'ha utilitzat la base de dades és **mysql**, àmpliament utilitzada i d'ús gratuït.

Les aplicacions que interactuen amb bases de dades, presenten una dificultat: com accedir a les dades de forma ordenada per a que existeixi una separació per capes reals? Aquest repte té un clar objectiu: fer que un canvi en la definició de la Base de Dades no afecti a la capa de negoci. En molts casos s'opta per la via ràpida i fer que cada classe de la capa de negoci accedeixi directament a la base de dades utilitzant directament llenguatge SQL. Tot i semblar el més ràpid i eficient, aquest mètode provoca no tenir separació real de capes el que implica que qualsevol canvi o manteniment posterior en la capa de persistència tindrà un efecte molt gran en la capa de negoci provocant majors costos de manteniment, poca transparència i especialització de funcions i en definitiva una arquitectura que dificultarà a la llarga l'evolució de l'aplicatiu.

A més a més, en Java es tracten Objectes de negoci, i les Bases de dades registren camps de dades. Un altre repte és simplificar la visió de les capes de negoci per a que accedeixin a les dades persistents en forma d'objectes i que sigui transparent com s'implementa a la Base de dades.

Per a resoldre aquests reptes existeix el **framework Hibernate**, que permet configurar l'aplicació per a que la capa de Negoci tracti les dades persistents com si es tractés d'un objecte més. D'aquesta manera és completament transparent a l'aplicació quan està tractant amb dades guardades a la Base de dades, agilitzant i facilitant el desenvolupament.

Per a assolir aquesta separació de capes, utilitzo el **patró DAO** – Data Access Object – i **DTO** – Data Transfer Object. El DTO (altament anomenat POJO) no és més que una classe Java que representa un registre d'una taula persistent, i el DAO és també una classe Java que encapsula tots els accessos a una taula concreta. Així, qualsevol accés a un registre de la Base de dades en el projecte es fa a través del DAO corresponent, que retorna els Objectes (DTO) seleccionats segons els criteris utilitzats.

Hibernate necessita la definició d'un fitxer de mapeig entre els diferents camps de la taula i els objectes del DTO corresponent. Aquesta relació es guarda en els fitxers hbm d'Hibernate en format XML. Per a generar-los, juntament amb les classes DTO, he utilitzat el plugin d'hibernate amb Eclipse per a fer ingenieria inversa de la BBDD i crear automàticament les classes DTO així com els fitxers de mapeig.

Per a fer el disseny de la Base de dades, he seguit els següents criteris enumerats a continuació:

- ◆ Ús de claus primàries seqüencials sense significat (*meaningless*) per a independitzar la clau de les dades, de tal manera que no és necessari canviar la clau de en cas d'error en les dades significatives. Aquesta clau no s'utilitzarà per a fer cerques, en canvi es pot utilitzar per a comparar objectes de forma senzilla.
- ◆ Ús de la clàusula UNIQUE i NOT NULL per a definir claus alternatives.
- ◆ Ús de la clàusula FOREIGN KEY per a identificar les claus foranies en referència a altres taules.
- ◆ Els identificadors de les taules son del tipus Integers (rang de 0 a 4.294.967.295) ja que considero que és suficient per la volumetria d'entrades podria requerir l'aplicatiu. A excepció del Moviment, ja que pot arribar a generar un nombre major d'entrades, que està definit com a Long.



## 2.4 CONCLUSIONS I LÍNIES DE FUTUR

Els objectius inicials del projecte eren diversos: per un costat la creació d'una aplicació en la plataforma J2EE utilitzant una arquitectura equilibrada que permetés una gestió i manteniment posterior senzill, i per un altre, l'aprenentatge d'una tecnologia desconeguda (un nou món!), que ha suposat un repte important per a demostrar que els coneixements apresos al llarg de les assignatures i cursos de la UOC finalment es conjuguen per a poder generar una aplicació complexa.

En quant al desenvolupament del projecte i la planificació, he pogut portar-lo a terme segons el previst inicialment, incloent el desenvolupament de les funcionalitats addicionals (a implementar opcionalment en cas de disposar de suficient temps). Aquest fet ha estat possible per haver estat rigorós en el compliment de les dates planificades, havent de treballar moltes vegades en horaris nocturns robant hores de son per a poder avançar.

A més a més, el projecte pot seguir creixent funcionalment. Sembla natural que el sistema evolucioni cap a l'incorporació de les següents funcionalitats/mòduls:

- ◆ **Pressupost:** un control senzill de pressupost, on es pugui definir despeses per partida i poder fer-ne un seguiment mensual de les mateixes. Aquesta seria una de les funcionalitats que podria ajudar més a molts dels usuaris potencials de l'aplicació.
- ◆ **Simulacions:** permetria a l'usuari evaluar un canvi en la seva situació econòmica, com un canvi de feina, quedar-se en l'atur, hipotecar-se o comprar-se un cotxe. Aquesta funcionalitat donaria resposta als usuaris que es fan preguntes com: Em puc permetre comprar un cotxe/pis? Què he de fer per a equilibrar la meua economia si m'he quedat a l'atur? Què he estalviar si vull comprar-me un pis?
- ◆ **Gestió d'actius:** Representar els actius valorats (pis, cotxe, parking,...) per a disposar d'una informació patrimonial completa de l'unitat familiar representada.

Des de la perspectiva d'haver finalitzat el projecte i estar a punt d'entregar-lo, puc assegurar que l'experiència ha estat molt bona. He hagut de navegar entre la infinitat d'alternatives, llibreries, frameworks, mòduls, patrons,... existents per a poder anar superant obstacle a obstacle, aprenent i decidint quina era la millor alternativa pel projecte en cada moment per obtenir com a resultat la present entrega.

En definitiva, el projecte ha requerit d'un esforç molt important d'aprenentatge i d'aplicar coneixements adquirits. En aquest sentit, destacar el fet que es demostra que la carrera no només m'ha servit per a aprendre "coneixements", sinó que m'ha donat eines suficients per a poder aprendre, decidir amb criteri suficient i poder seguir fent-ho en un futur per a aplicar-ho a aquesta o a altres àrees del món de la tecnologia existents o per venir. Gràcies UOC.

### 3 GLOSSARI

- **Moviment:** transacció monetària. Qualsevol despesa o ingrés que l'usuari consideri.
- **Compte:** Representa el compte on es realitzen els moviments d'ingrés o despesa. Podria tractar-se de comptes bancaris reals o lògics, i es poden utilitzar a conveniència de l'usuari pel seu control. Es podria utilitzar, per exemple per a diferenciar la persona responsable de la despesa, el compte bancari,...
- **Concepte** de despesa: Centre de cost sobre el que l'usuari vulgui controlar les seves despeses. Per exemple podrien ser conceptes de despesa el Cotxe, el Pis, Oci, etc.
- **Naturalesa:** Classificador dels moviments que defineix el tipus de despesa segons la seva realitat. Exemples de Naturalesa podrien ser Gasolina, Restaurant, Roba, Llibres, etc.

## 4 BIBLIOGRAFIA

**JSP** - Anaya Multimedia 2001

**Programación en Java 2 J2SE 1.4** - Anaya Multimedia 2003

Materials de la UOC: Apunts de l'assignatura Enginyeria del programari

Pàgines Internet:

Formació prèvia en tecnologia J2EE:

- <http://java.sun.com/j2se/1.5.0/docs/api/>
- <http://java.sun.com/javase/6/docs/api/>
- <http://www.programacion.com/java/tutorial/j2ee/>

IDE Eclipse

- <http://help.eclipse.org/galileo/index.jsp>

Servidor Tomcat:

- <http://tomcat.apache.org/tomcat-6.0-doc/index.html>

Framework Struts2:

- <http://struts.apache.org/2.1.8/index.html>
- <http://struts.apache.org/2.0.14/docs/tag-reference.html>

Base de dades MySQL

- <http://dev.mysql.com/doc/refman/5.1/en/index.html>

Framework Hibernate:

- ◆ <http://docs.jboss.org/hibernate/core/3.3/reference/en/html/>
- ◆ <http://www.programacion.com/java/tutorial/hibernate/>
- ◆ <http://www.agilogy.com/ca/disenyo-de-bases-de-datos-del-modelo-conceptual-al-modelo-logico/>
- ◆ <http://www.hibernate-training-guide.com/natural-keys.html>
- ◆ <http://www.javabeat.net/tips/106-three-ways-to-create-query-in-hibernate.html>
- ◆ <http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/queryhql.html#queryhql-from>
- ◆ <http://docs.jboss.org/hibernate/core/3.3/reference/en/html/objectstate.html>
- ◆ <http://www.vaannila.com/hibernate/hibernate-example/hibernate-tools-1.html>
- ◆ <http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/tutorial.html#tutorial-firstapp>
- ◆ <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernateTools>

Formació en tecnologia AJAX

- ◆ <http://www.benmccann.com/dev-blog/struts-2-ajax-tutorial-doj-autocompleter-example/>

- ◆ <http://viralpatel.net/blogs/2010/01/struts-2-ajax-tutorial-example-drop-down.html>

Capa de visualització:

- ◆ <http://struts.apache.org/1.x/struts-tiles/index.html>
- ◆ <http://code.google.com/intl/es-ES/apis/chart/>
- ◆ <http://www.w3schools.com/html>
- ◆ <http://www.w3schools.com/css>

## 5 ANNEXES

### 5.1 REQUERIMENTS MÍNIMS

El projecte ha estat provat per a funcionar en la següent infraestructura:

#### Servidor

- Servidor d'aplicacions: **Tomcat 6.0.26**
- Servidor de Base de dades: **mySQL 5.1.45**
- Tenir instal·lat el **JRE 6**
- Accés a Internet per a la generació de gràfiques amb el servei Google Charts

#### Client

- Navegador del client: **Mozilla Firefox 4.0.1**

### 5.2 MANUAL D'INSTAL·LACIÓ

Considerant que es compleixen els requisits mínims descrits anteriorment, per a la instal·lació de l'aplicació s'ha de procedir a executar els següents passos:

- ◆ Desplegament del WAR que empaqueta l'aplicació al servidor Tomcat.
- ◆ Configuració del fitxer Hibernate on es configura l'accés al servidor de la Base de dades
- ◆ Configuració del fitxer log4j.properties de l'aplicació per a definir el nivell de registres i el seu destí.
- ◆ Execució de l'script de creació i inicialització de la Base de dades
- ◆ Inici de l'ús de l'aplicació.

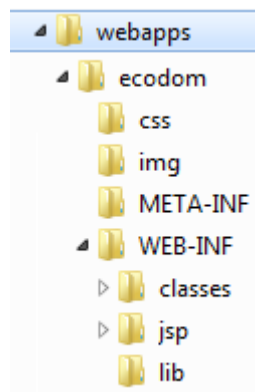
#### 5.2.1 DESPLEGAMENT DE L'APLICACIÓ AL SERVIDOR D'APLICACIONS TOMCAT

Per a realitzar el desplegament de l'aplicació es poden utilitzar diferents mètodes:

1. Utilitzant l'opció d'autodesplegament d'aplicacions. És necessari:
  - a. Assegurar-se que el Tomcat està configurat per a fer l'autodeploy en arrancar (opció `unpackWARs="true"` marcada en el fitxer de configuració `conf/server.xml`)
  - b. Guardar el fitxer **WAR** de la versió de l'aplicació a desplegar en el directori **webapps** d'aplicacions del Tomcat.
  - c. Arrancar el servidor Tomcat. Aquest pas provocarà el desplegament de l'aplicació en el servidor de forma automàtica.
2. Fer un deploy (desplegament) del WAR manualment utilitzant el Manager de Tomcat URL: <http://localhost:8080/manager/> utilitzant l'usuari i password definit per l'administrador del servidor i utilitzar l'opció "Archivo WAR a desplegar", seleccionant el WAR de la versió de l'aplicació a desplegar.

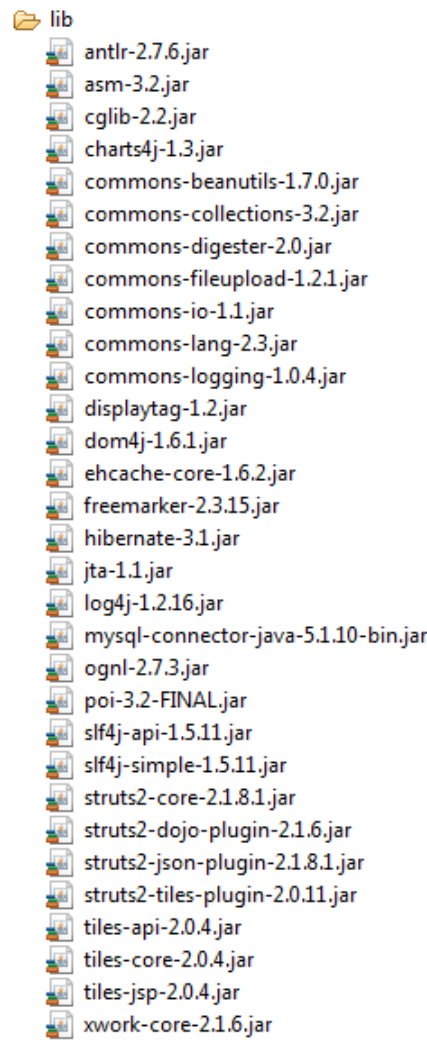
Qualsevol de les dues opcions executen el desplegament de l'aplicació en el servidor, generant les carpetes de l'aplicació dins de la carpeta base definida per les aplicacions, normalment /webapps.

Dins d'aquesta carpeta es despleguen totes les carpetes contingudes en el WAR i que segueixen els estàndards definits per a l'especificació de Servlets.



Carpetes:

- ◆ CSS: Fulles d'estil on es defineixen característiques de visualització de l'aplicació. Aquesta fulla es referència a totes les pàgines de l'aplicació, pel que modificar l'aparença és senzill i ràpid.
- ◆ IMG: Imatges utilitzades en l'aplicació.
- ◆ META-INF: conté el fitxer manifest
- ◆ WEB-INF: Conté el fitxer de configuració de l'aplicació (**web.xml**) i la configuració de les plantilles de visualització necessàries per l'ús del framework Tiles (**tiles.xml**)
  - ◆ CLASSES → Conté els fitxers de configuració d'hibernate on es defineix l'accés a la Base de dades (**hibernate.cnf**), el fitxer de configuració dels logs on es defineix el nivell de logs a registrar i el medi (**log4j.properties**) i el fitxer de configuració de Struts per a mapejar cada Action amb la classe corresponent a executar i definir les accions segons el resultat obtingut (**struts.xml**). Conté també els paquets i classes del projecte.
  - ◆ JSP → Conté les pàgines JSP que s'executaran en el Servidor per a generar les respostes al client en format HTML.
  - ◆ LIB → Conté les llibreries estàndards necessàries per al funcionament:



### 5.2.2 CONFIGURACIÓ HIBERNATE

En el fitxer WEB-INF/classes/hibernate.cfg.xml es defineixen les propietats bàsiques de l'accés de l'aplicació a la Base de dades, com és la URL de la Base de dades, l'usuari i password d'accés, el driver a utilitzar,... Pel funcionament de l'aplicació serà necessari revisar l'usuari i password d'accés a la Base de dades segons estigui configurat en el servidor MySQL:

```
<property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="hibernate.connection.password">xxxxx</property>
<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ecodom2</property>
<property name="hibernate.connection.username">root</property>
```

### 5.2.3 CONFIGURACIÓ DE LES PROPIETATS DEL LOG

En el fitxer WEB-INF/classes/log4j.properties es defineixen les propietats desitjades respecte al registre de logs. Modificant el paràmetre indicat, es canvia el nivell de detall de logs registrats segons la jerarquia de nivells de menys a més detall: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE, ALL.

```
### direct log messages to stdout ###
log4j.rootLogger=INFO, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} ECODOM %5p %c{1}:%L - %m%n
```

## 5.2.4 SCRIPT DE LA BBDD

Executar l'script següent per a la creació de la base de dades de l'aplicació.

Genera la base de dades ecodom2 i totes les seves taules. Addicionalment genera l'usuari administrador i genera la configuració de conceptes i naturaleses associats a l'usuari administrador, utilitzat posteriorment com a referència per a la creació de nous usuaris que utilitza els conceptes i naturaleses de l'administrador com a model:

```
-- Definició de paràmetres i guardat dels inicials--
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';

-- Versió 2.0 de la BBDD ecodom. --
CREATE DATABASE IF NOT EXISTS ecodom2;

USE ecodom2;

-- Definició de la taula usuari --
DROP TABLE IF EXISTS `ecodom2`.`usuari`;
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`usuari` (
  `id_usuari` INTEGER NOT NULL AUTO_INCREMENT,
  `dni` CHAR(9) NOT NULL,
  `nom` CHAR(30) NOT NULL,
  `cognom1` CHAR(30) NOT NULL,
  `cognom2` CHAR(30),
  `acces_fallit` INTEGER NOT NULL DEFAULT 0,
  `rol` CHAR(1) NOT NULL,
  `password` CHAR(30) NOT NULL,
  PRIMARY KEY (`id_usuari`),
  UNIQUE (`dni`)
) ENGINE = InnoDB;

-- Definició de la taula compte --
DROP TABLE IF EXISTS `ecodom2`.`compte`;
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`compte` (
  `id_compte` INTEGER NOT NULL AUTO_INCREMENT,
  `id_usuari` INTEGER NOT NULL,
  `descripcio` CHAR(45),
  PRIMARY KEY (`id_compte`),
  CONSTRAINT `fk_compte_id_usuari`
    FOREIGN KEY (`id_usuari`)
      REFERENCES `ecodom2`.`usuari` (`id_usuari`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
) ENGINE = InnoDB;

-- Definició de la taula naturalesa --
DROP TABLE IF EXISTS `ecodom2`.`naturalesa`;
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`naturalesa` (
  `id_naturalesa` INTEGER NOT NULL AUTO_INCREMENT,
  `id_usuari` INTEGER NOT NULL,
  `descripcio` CHAR(45),
  PRIMARY KEY (`id_naturalesa`),
  CONSTRAINT `fk_naturalesa_id_usuari`
    FOREIGN KEY (`id_usuari`)
      REFERENCES `ecodom2`.`usuari` (`id_usuari`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
) ENGINE = InnoDB;

-- Definició de la taula concepte --
DROP TABLE IF EXISTS `ecodom2`.`concepte`;
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`concepte` (
  `id_concepte` INTEGER NOT NULL AUTO_INCREMENT,
  `id_usuari` INTEGER NOT NULL,
  `descripcio` CHAR(45),
  PRIMARY KEY (`id_concepte`),
  CONSTRAINT `fk_concepte_id_usuari`
    FOREIGN KEY (`id_usuari`)
      REFERENCES `ecodom2`.`usuari` (`id_usuari`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
) ENGINE = InnoDB;

-- Definició de la taula natucon --
DROP TABLE IF EXISTS `ecodom2`.`natucon`;
```



```

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`natucon` (
  `id_naturalesa` INTEGER NOT NULL,
  `id_concepte` INTEGER NOT NULL,
  `id_usuari` INTEGER NOT NULL,
  `actiu` BIT,
  PRIMARY KEY (`id_naturalesa`, `id_concepte`),
  CONSTRAINT `fk_natucon_id_naturalesa`
    FOREIGN KEY (`id_naturalesa`)
      REFERENCES `ecodom2`.`naturalesa` (`id_naturalesa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_natucon_id_concepte`
    FOREIGN KEY (`id_concepte`)
      REFERENCES `ecodom2`.`concepte` (`id_concepte`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_natucon_id_usuari`
    FOREIGN KEY (`id_usuari`)
      REFERENCES `ecodom2`.`usuari` (`id_usuari`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB;

-- Definició de la taula moviment --
DROP TABLE IF EXISTS `ecodom2`.`moviment` ;
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `ecodom2`.`moviment` (
  `id_moviment` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `id_usuari` INTEGER NOT NULL,
  `id_naturalesa` INTEGER NOT NULL ,
  `id_concepte` INTEGER NOT NULL ,
  `id_compte` INTEGER NOT NULL ,
  `data` DATE NOT NULL ,
  `valor` FLOAT NOT NULL ,
  `descripcio` VARCHAR(100) NOT NULL ,
  `comentaris` VARCHAR(100) NULL ,
  PRIMARY KEY (`id_moviment`),
  CONSTRAINT `fk_moviment_id_usuari`
    FOREIGN KEY (`id_usuari`)
      REFERENCES `ecodom2`.`usuari` (`id_usuari`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_moviment_id_naturalesa`
    FOREIGN KEY (`id_naturalesa`)
      REFERENCES `ecodom2`.`naturalesa` (`id_naturalesa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_moviment_id_concepte`
    FOREIGN KEY (`id_concepte`)
      REFERENCES `ecodom2`.`concepte` (`id_concepte`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_moviment_id_compte`
    FOREIGN KEY (`id_compte`)
      REFERENCES `ecodom2`.`compte` (`id_compte`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION )
ENGINE = InnoDB;

-- Creació de l'usuari administrador
INSERT INTO usuari
  (dni, nom, cognom1, acces_fallit, rol, password) VALUES
  ('00000000T', 'administrador', 'ecodom', 0, 'A', 'administrador');

-- Alta de Naturaleses --

INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Roba');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Menjar');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Hipoteca');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Equipament');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Interessos');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Impostos');
INSERT INTO naturalesa
  (id_usuari, descripcio) VALUES
  (1, 'Manteniment');

```

```

INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Assegurança');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Col·legi');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Llibres');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Nòmina');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Restaurant');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Traspas');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Llum');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Telèfon Mòbil');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Telèfon fix');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Lloguer');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Gasolina');
INSERT INTO naturalesa
(id_usuari, descripcio) VALUES
(1, 'Parking');

-- Alta de Conceptes --

INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Pis');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Manutenció');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Finances');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Feina');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Estudis');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Cotxe');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Salut');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Transport');
INSERT INTO concepte
(id_usuari, descripcio) VALUES
(1, 'Oci');

-- Relació Conceptes - Naturaleses --

INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 1, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 2, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 3, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 4, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 5, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(1, 6, 1, 1);
INSERT INTO natucon

```

```

        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 7, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 8, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 9, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 10, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 11, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 12, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 13, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 14, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 15, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 16, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 17, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 18, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (1, 19, 1, 0);

INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 1, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 2, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 3, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 4, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 5, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 6, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 7, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 8, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 9, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 10, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 11, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 12, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 13, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 14, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 15, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (2, 16, 1, 1);

```

```
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(2, 17, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(2, 18, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(2, 19, 1, 0);

INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 1, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 2, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 3, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 4, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 5, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 6, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 7, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 8, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 9, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 10, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 11, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 12, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 13, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 14, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 15, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 16, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 17, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 18, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(3, 19, 1, 0);

INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 1, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 2, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 3, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 4, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 5, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 6, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
```

```
(4, 7, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 8, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 9, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 10, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 11, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 12, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 13, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 14, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 15, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 16, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 17, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 18, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(4, 19, 1, 0);

INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 1, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 2, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 3, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 4, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 5, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 6, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 7, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 8, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 9, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 10, 1, 1);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 11, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 12, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 13, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 14, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 15, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
(5, 16, 1, 0);
INSERT INTO natucon
(id_concepte, id_naturalesa, id_usuari, actiu) VALUES
```

```

        (5, 17, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (5, 18, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (5, 19, 1, 0);

INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 1, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 2, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 3, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 4, 1, 1);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 5, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 6, 1, 1);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 7, 1, 1);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 8, 1, 1);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 9, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 10, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 11, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 12, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 13, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 14, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 15, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 16, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 17, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 18, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (6, 19, 1, 1);

INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 1, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 2, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 3, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 4, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 5, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 6, 1, 0);
INSERT INTO natucon
    (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
    (7, 7, 1, 0);
INSERT INTO natucon

```

```

        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 8, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 9, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 10, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 11, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 12, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 13, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 14, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 15, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 16, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 17, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 18, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (7, 19, 1, 0);

INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 1, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 2, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 3, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 4, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 5, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 6, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 7, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 8, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 9, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 10, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 11, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 12, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 13, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 14, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 15, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 16, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 17, 1, 0);
INSERT INTO natucon

```

```

        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 18, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (8, 19, 1, 1);

INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 1, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 2, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 3, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 4, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 5, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 6, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 7, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 8, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 9, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 10, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 11, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 12, 1, 1);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 13, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 14, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 15, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 16, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 17, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 18, 1, 0);
INSERT INTO natucon
        (id_concepte, id_naturalesa, id_usuari, actiu) VALUES
        (9, 19, 1, 0);

-- Restauració dels paràmetres inicials --
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Executant aquest script, es genera un usuari administrador amb el següent dni (clau de l'usuari) i password:

- Usuari administrador: 00000000T
- Password: administrador

L'administrador podrà modificar el seu perfil el primer cop que entri a l'aplicació i canviar el password d'accés. Qualsevol usuari es pot donar d'alta des de la mateixa aplicació.



### 5.2.5 ACCEDIR A L'APLICACIÓ

En aquest punt, ja es pot accedir a l'aplicació mitjançant la URL: <http://srv:8080/ecodom>


On srv és la direcció del servidor on està allotjada l'aplicació

En cas d'executar-se des del mateix servidor, la URL podria ser: <http://localhost:8080/ecodom>

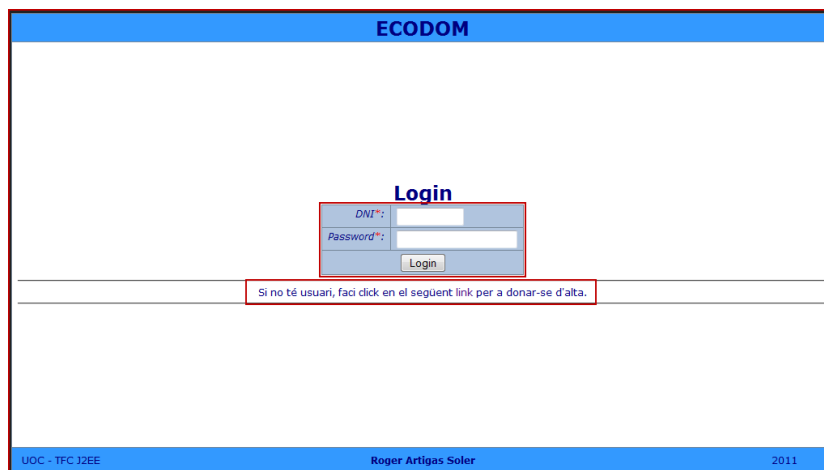
## 5.3 MANUAL BÀSIC D'USUARI

### 5.3.1 ACCÉS D'USUARIS A L'APLICACIÓ

Obrir el navegador i accedir a l'URL de l'aplicació:

 <http://localhost:8080/ecodom/loginForm>

Obtindrem la següent pantalla on podrem accedir si tenim usuari donat d'alta o podrem crear un compte nou.



La imatge mostra la pantalla de login d'ECODOM. A la part superior hi ha un capçalera blava amb el text "ECODOM". Al centre, hi ha un formulari titulat "Login" amb dos camps d'entrada: "DNI\*\*:" i "Password\*\*:". A sota dels camps hi ha un botó "Login". Just a sota del formulari, hi ha un text que diu: "Si no té usuari, faci click en el següent link per a donar-se d'alta." A la part inferior de la pantalla, hi ha una barra blava amb el text "UOC - TFC I2EE", "Roger Artigas Soler" i "2011".

### 5.3.2 PANTALLA PRINCIPAL

La pantalla principal està dividida en una àrea de menús i la zona de treball.

S'ha de seleccionar una opció del menú.



La imatge mostra la pantalla principal d'ECODOM. A la part superior hi ha un capçalera blava amb el text "ECODOM". A l'esquerra, hi ha un menú amb dos seccions: "Configuració" i "Moviments". La secció "Configuració" inclou: "Dades propies", "Comptes", "Naturaleses" i "Conceptes". La secció "Moviments" inclou: "Alta Moviment", "Busca Moviments", "Fixer de moviments", "Salvos" i "Anàlisi de moviments". A la dreta, hi ha una zona de treball amb el text "Benvingut a Ecodom" i "Escull una opció del menú." A la part superior esquerra, hi ha el text "Avui és 13/06/2011". A la part superior dreta, hi ha el text "Usuari: Roger1 Artigas" amb un icona de perfil. A la part inferior de la pantalla, hi ha una barra blava amb el text "UOC - TFC I2EE", "Roger Artigas Soler" i "2011".

### 5.3.3 CONFIGURACIÓ

Les opcions de configuració serveixen per actualitzar les dades personals i els classificadors dels moviments (comptes, naturaleses i conceptes). En donar d'alta l'usuari es carrega amb una pre-configuració – l'associada a l'usuari administrador – que posteriorment pot ser modificada segons les necessitats de cada usuari.

#### 5.3.3.1 DADES PRÒPIES

Accions:



Modifica dades de l'usuari

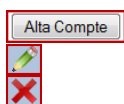
Modificar el password

Baixa de l'usuari

Esborra totes les seves dades associades

#### 5.3.3.2 COMPTES

Accions:



Crear compte

Modificar compte

Esborrar compte El sistema valida que no tingui moviments associats

## 5.3.3.3 NATURALESES

**ECODOM**

Avui és 13/06/2011 Usuari: Roger Artigas

**Naturaleses**

Configuri les naturaleses de les seves despeses/ingressos per a poder classificar els seus moviments i poder conèixer posteriorment en què es gasta els diners.  
Es tracta d'un classificador pels moviments que representa el QUÈ de la despesa o ingrés.

ID	Naturalesa	Eborra	Modifica
20	Roba	X	
21	Menjar	X	
22	Hipoteca	X	
23	Equipament	X	
24	Interessos	X	
25	Impostos	X	
26	Manteniment	X	
27	Assegurança	X	
28	Col·legi	X	
29	Llibres	X	
30	Nòmina	X	
31	Restaurant	X	
32	Traspas	X	
33	Llum	X	
34	Telèfon Mòbil	X	

**Configuració**

- Dades propies
- Comptes
- Naturaleses
- Conceptes

**Moviments**

- Alta Moviment
- Busca Moviments
- Fitxer de moviments
- Saldos
- Anàlisi de moviments

Accions:



Crear naturalesa

Modificar naturalesa

Esborrar naturalesa

El sistema valida que no tingui moviments associats

## 5.3.3.4 CONCEPTES

**ECODOM**

Avui és 13/06/2011 Usuari: Roger Artigas

**Conceptes**

Configuri els conceptes de les seves despeses/ingressos per a poder classificar els seus moviments i poder conèixer posteriorment perquè es gasta els diners.  
Es tracta d'un classificador pels moviments que representa el PERQUÈ de la despesa o ingrés.  
Haurà d'assignar les naturaleses permet utilitzar a cada concepte.

ID	Concepte	Eborra	Modifica	Naturaleses
10	Pis	X		
11	Manutenció	X		
12	Finances	X		
13	Feina	X		
14	Estudis	X		
15	Cotxe	X		
16	Salut	X		
17	Transport	X		
18	Oci	X		

Alta

UOC - TFC J2EE Roger Artigas Soler 2011

Accions:



Crear concepte

Modificar concepte

Esborrar concepte

El sistema valida que no tingui moviments associats



Associar naturaleses

## 5.3.3.5 RELACIÓ CONCEPTES - NATURALESES

**ECODOM**

Avui és 13/06/2011 Usuari: Roger Artigas

**Assigna Naturaleses**  
Per a cada concepte, identifiquei quines naturaleses són possibles. Es tracta d'una ajuda per a evitar errors en la classificació dels moviments.

Concepte: **Pis**

Concepte	Naturalesa	Actiu
Pis	Roba	✗
Pis	Menjar	✗
Pis	Hipoteca	✓
Pis	Equipament	✓
Pis	Interessos	✗
Pis	Impostos	✓
Pis	Manteniment	✓
Pis	Assegurança	✓
Pis	Col·legi	✗
Pis	Llibres	✗
Pis	Nòmina	✗
Pis	Restaurant	✗
Pis	Traspas	✗
Pis	Llum	✗

**Configuració**

- Dades propies
- Comptes
- Naturaleses
- Conceptes

**Moviments**

- Alta Moviment
- Busca Moviments
- Fitxer de moviments
- Salos
- Anàlisi de moviments

Accions:



Indica que no està activa. Pulsar per a activar la naturalesa



Indica que està activa. Pulsar per a desactivar la naturalesa El sistema valida que no tingui moviments associat amb la relació que es desactiva

## 5.3.4 MOVIMENTS

## 5.3.4.1 ALTA DE MOVIMENT

**ECODOM**

Avui és 13/06/2011 Usuari: Roger Artigas

**Alta Moviment**  
Dongui d'alta un moviment

Compte\*:

Concepte\*:

Naturalesa\*:

Data\*: 13/06/11

☒ Despesa ☐ Ingress

Import\*: 0,00

Descripció\*:

Comentaris:

UOC - TFC J2EE Roger Artigas Soler 2011

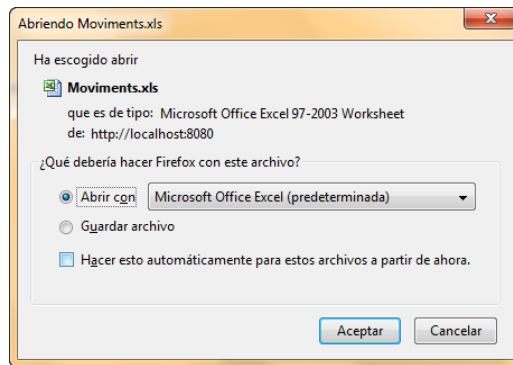
Informar els camps obligatoris. En el camp Naturalesa únicament apareixeran les naturaleses associades al concepte prèviament seleccionat.

## 5.3.4.2 BUSCA MOVIMENTS

Indicar els conceptes de cerca desitjats. Si es deixa en blanc, es busquen tots els moviments sense considerar el criteri. En pulsar Busca, es mostrarà el llistat de moviments que compleixen els criteris de cerca.

## 5.3.4.3 FITXER DE MOVIMENTS

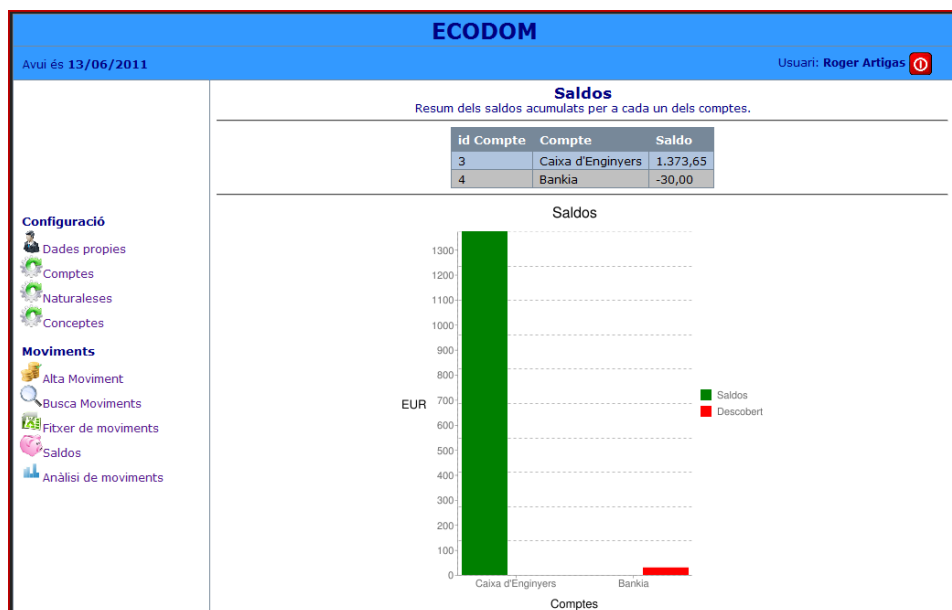
Indicar els conceptes de cerca desitjats. Si es deixa en blanc, es busquen tots els moviments sense considerar el criteri. En pulsar Genera fitxer, es generarà un fitxer en format Excel amb el llistat de moviments que compleixen els criteris de cerca.



L'usuari pot indicar l'opció desitjada.

#### 5.3.4.4 SALDOS

Es visualitzen els saldos acumulats de cada compte de l'usuari, tant si té saldo creditor com deutor:



#### 5.3.4.5 ANÀLISI DE MOVIMENTS

El sistema permet utilitzar diferents criteris d'anàlisi: per concepte i per naturalesa.

També permet seleccionar un subconjunt de moviments sobre els que es vol fer l'estudi utilitzant diferents criteris de selecció.

**ECODOM**

Avui és 13/06/2011 Usuari: Roger Artigas

**Configuració**

- Dades propies
- Comptes
- Naturaleses
- Conceptes

**Moviments**

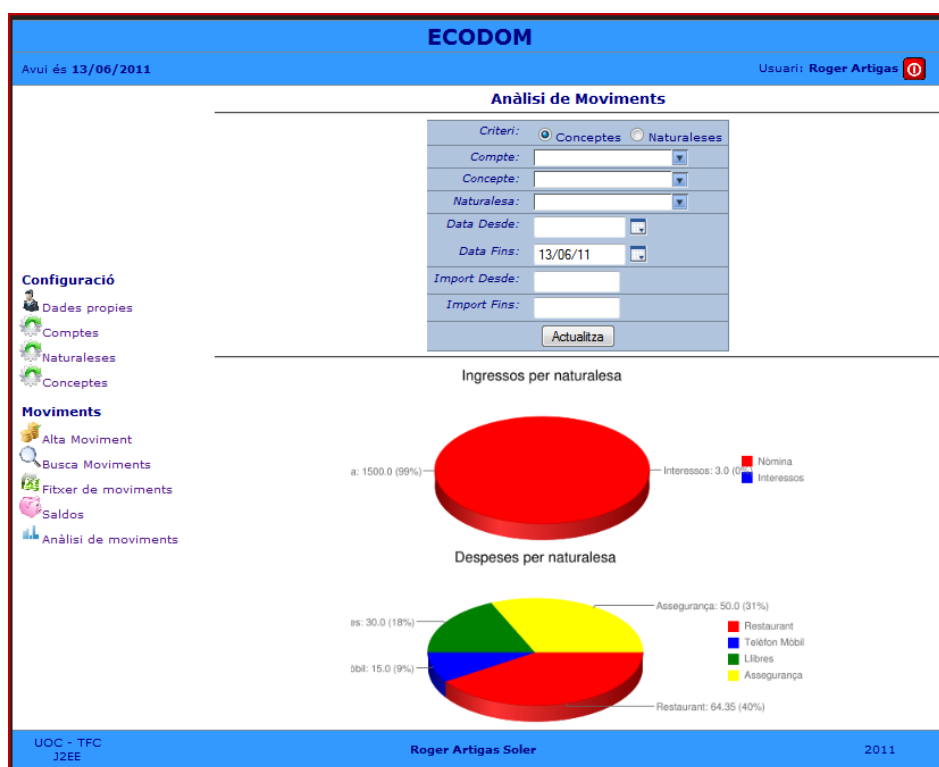
- Alta Moviment
- Busca Moviments
- Fitxer de moviments
- Salos
- Anàlisi de moviments

**Anàlisi de Moviments**

Criteri:	<input checked="" type="radio"/> Conceptes <input type="radio"/> Naturaleses
Compte:	<input type="text"/>
Concepte:	<input type="text"/>
Naturalesa:	<input type="text"/>
Data Desde:	<input type="text"/>
Data Fins:	13/06/11
Import Desde:	<input type="text"/>
Import Fins:	<input type="text"/>
<input type="button" value="Actualitza"/>	

UOC - TFC J2EE Roger Artigas Soler 2011

En pulsar Actualitza, es mostra el gràfic:



### 5.3.5 ADMINISTRACIÓ

Si l'usuari és administrador, tindrà accés en el menú a una opció addicional per a gestionar els usuaris:



### 5.3.5.1 USUARIS

Aquesta opció de menú dona accés a tots els usuaris de l'aplicació:

**ECODOM**

Avui és 13/06/2011

Usuari: administrador ecodom

**Configuració**

- Dades propies
- Comptes
- Naturaleses
- Conceptes

**Moviments**

- Alta Moviment
- Busca Moviments
- Fitxer de moviments
- Saldos
- Anàlisi de moviments

**Administració**

- Usuaris

**Usuaris donats d'alta**

ID	Nom	Primer cognom	Segon cognom	DNI	Bloq	Ini.pwd	Baixa
1	administrador	ecodom		00000000T	0		
2	Roger	Artigas	Soler	44023383A	0		
3	Eulàlia	Bori	Ribas	47715188D	1		
7	Andrés	Iniesta	Fuentealbilla	44023382W			
8	pepe	perez	perez	36950369H	0		

UOC - TFC 12EE

Roger Artigas Soler

2011

Accions:



Desbloquejar usuari

Només s'activa si l'usuari està bloquejat (ha superat els 3 intents de login amb errors). En cas contrari indica en numero d'intents d'accés incorrectes.



Inicialitzar password



Baixa de l'usuari

S'esborren totes les dades associades. Anar amb compte.