

# Treball Final de Carrera - .NET

Gestió de despeses personals



**Universitat Oberta  
de Catalunya**

**Xavier Miñana Garcia**  
13/01/2014  
Consultor: Jairo Sarrias Guzman

## Índex de continguts

1	INTRODUCCIÓ.....	4
2	DESCRIPCIÓ DEL PROJECTE .....	5
3	ANÀLISI I DISSENY.....	11
4	DESENVOLUPAMENT .....	37
5	AVALUACIÓ DE COSTOS .....	44
6	TREBALL FUTUR.....	45
7	CANVIS INCLOSOS EN AQUESTA ENTREGA.....	46
8	CANVIS RESPECTE L'ANÀLISI I DISSENY DE L'APLICACIÓ.....	47
9	CONCLUSIONS.....	49
10	BIBLIOGRAFIA .....	50

*A tots els familiars i amics.*

*En especial,*

*a la meva dona, per estar al meu costat, no només  
durant aquest temps, sinó des del primer dia*

*al meu fill, que no he pogut passar tantes estones  
jugant al teu costat com m'hagués agradat*

*i a la meva mare, que no has pogut veure com acaba  
aquest capítol però sempre m'has animat a acabar-lo*

*Gràcies!*

# 1 Introducció

Aquest document és un resum de totes les fases del desenvolupament del projecte de final de carrera, la proposta del qual ve donada per la Universitat Oberta de Catalunya (UOC) i en el present document es recull el resum de la planificació del projecte, un extracte de l'anàlisi i disseny del mateix i observacions i reflexions un cop finalitzat.

Aquest projecte neix d'una necessitat específica d'una empresa i pretén definir un sistema de gestió de despeses personals. L'objectiu d'aquesta empresa és posar fi als pagaments de despeses dels seus treballadors mitjançant la caixa per passar a fer-ho mitjançant la nòmina del mes en curs, per tant, necessiten un sistema en el qual es puguin registrar despeses i acceptar-les o denegar-les.

Per aconseguir aquest objectiu s'ha definit una plataforma que conté tres parts ben diferenciades.

- Un servei web WCF on estarà tota la lògica de l'aplicació i la capa d'accés a les dades.
- Una aplicació web ASP.NET que contindrà tota la funcionalitat especificada en aquest document i que farà crides al servei web per poder oferir aquesta funcionalitat.
- Una aplicació Windows Phone 7 on estarà disponible tota la funcionalitat exceptuant la part d'administració que únicament estarà disponible des de l'aplicació web.

A continuació es detallen les eines emprades en cadascuna de les fases d'anàlisi i disseny, acompanyades d'una breu explicació del perquè de tal elecció:

- *Diagrames de casos d'us* i *Model conceptual* han estat elaborats amb **StarUML** (<http://staruml.sourceforge.net/>), es tracta d'una eina bastant potent que permet fer tot tipus de diagrames UML, és senzilla d'utilitzar i és gratuïta.
- Els *Diagrames d'arquitectura d'aplicació*, *HW* i *SW* han estat elaborats amb **Microsoft Office Visio 2003**. Un clàssic per elaborar diagrames d'aquest tipus.
- Per al *Diagrama ER* de la base de dades s'ha optat per la utilització del propi **Entity Framework Designer**. La avantatge d'utilitzar aquest software és que ens permet crear la base de dades directament a partir del model, fer modificacions al model i passar-les a la base de dades, la creació de les classes del model i també ens permet la creació dels scripts de creació de la base de dades.
- Per al disseny de l'aplicació web com per a l'aplicació mòbil s'ha utilitzat **Mockflow** (<http://app.mockflow.com/>). Es tracta d'una eina cloud molt senzilla d'utilitzar i que no fa falta instal·lar res a l'ordinador per poder utilitzar-la. No és una eina gratuïta però la versió de proves és suficient per fer les pantalles que es necessiten.

## 2 Descripció del projecte

### 2.1 Justificació del projecte (idoneïtat)

#### 2.1.1 Per què el projecte?

Moltes de les tasques de gestió que es realitzen d'una manera molt senzilla quan una empresa és petita es poden convertir en un veritable problema d'organització i recursos quan l'empresa creix. Un exemple molt clar és el cas que ens ocupa i el causant d'aquest projecte, la gestió de les despeses de personals.

Una mala gestió de les despeses personals pot fer perdre diners a l'empresa, per exemple, despeses que es paguen més d'una vegada o despeses no justificades, i de cara al treballador pot provocar malestar, degut a despeses que ha pagat de la seva butxaca i es triga molt temps a abonar o fins i tot que no es paguen.

Tot plegat pot provocar una font de problemes que en aquest cas es pot evitar amb una eina de gestió.

#### 2.1.2 Descripció del projecte

Es vol disposar d'una eina que compleixi amb les següents especificacions donades per el client:

- El treballador que vulgui afegir una despesa a la seva nòmina haurà d'omplir una fulla de despesa mitjançant el programa, la fulla serà per al mes en curs i es podrà afegir tantes despeses com calgui, un cop pagada la fulla es podrà obrir una altre nova.
- Les fulles han de ser autoritzades pel cap del departament al que pertany el treballador abans de ser pagades.
- L'aplicació ha de permetre introduir el tipus de despesa (quilometratge, dieta, aparcament, taxi, autobús, peatge, etc.)
- S'ha de permetre gestionar el calendari de dies festius, per tal de distingir les dietes i pagar-les amb l'import corresponent.
- S'ha de permetre introduir un motiu i en cas de tenir un justificant (sobretot per a despeses grans) poder adjuntar un fitxer amb la factura.
- Un cop afegida la despesa s'ha de poder sol·licitar el pagament, un cop sol·licitat, el responsable d'aquesta persona hauria d'autoritzar o rebutjar la despesa, en cas de rebuig s'ha d'indicar el motiu. Un cop autoritzada s'ha de procedir al seu pagament de forma que la fulla de despesa quedarà tancada

## 2.2 Objectius del projecte

### 2.2.1 Generals

Oferir a l'empresa un sistema que permeti fer una gestió centralitzada i eficient de les despeses personals.

### 2.2.2 Específics

- Identificar quines són les funcionalitats que necessita el client
- Fer un estudi de la solució extraient les tasques a realitzar.
- Valorar la viabilitat de les tasques a realitzar.
- Realitzar una planificació de projecte el més realista possible.
- Fer l'anàlisi i disseny del producte basant-nos en la planificació del projecte.
- Desenvolupament del producte.
- Finalment lliurar una solució robusta que cobreixi les necessitats del client en els temps marcats per la planificació del projecte.

## 2.3 Funcionalitats a desenvolupar

- Model de dades i la capa de persistència de dades.
- Capa de negoci que cobreixi tots els requisits funcionals.
- Serveis web que ofereixin la funcionalitat de la capa de negoci a altres aplicacions.
- Aplicació web que faci ús dels serveis web.
- Aplicació mòbil que faci ús dels serveis web.

## 2.4 Resultats esperats

S'espera obtenir un producte de qualitat que cobreixi totes les expectatives del client i que compleixi amb totes les funcionalitats descrites en l'anàlisi i disseny.

El producte final anirà acompanyat d'un manual d'usuari que contindrà tota la informació necessària perquè tant administradors, treballadors i responsables siguin capaços de fer servir el producte.

La interfície d'usuari serà senzilla i agradable, de fàcil navegació entre les diferents opcions del sistema i amb ajuda contextual sobre cadascuna de les mateixes.

El producte final també anirà acompanyat d'un manual d'instal·lació que inclourà els passos a seguir i la configuració necessària per posar el producte en funcionament.

## 2.5 Productes obtinguts

Al llarg del projecte hi han hagut variacions, però els productes finals obtinguts s'han aproximat molt al que es va planificar en un principi.

Les variacions que hi ha hagut es poden consultar a l'apartat *8 Canvis respecte l'anàlisi i disseny de l'aplicació*.

Per tant, els productes obtinguts han estat:

- Una llibreria on es troba tota la capa d'accés a dades i la capa de negoci de l'aplicació
- Un servei web (WCF) on es proveeix de la funcionalitat de la capa de negoci a la resta d'aplicacions de la plataforma.
- Una aplicació web que fa ús dels serveis web, que conté tota la funcionalitat de la plataforma.
- Una aplicació Windows Phone que fa ús dels serveis web, que conté tota la funcionalitat de la plataforma exceptuant la part d'administració.

## 2.6 Planificació inicial vs planificació final

### 2.6.1 Relació d'activitats

- **Planificació del projecte** (selecció del projecte, definició del projecte, estimació, redacció i entrega de la planificació)
- **Anàlisi i disseny** (instal·lació de programari, estudi EntityFramework, definició del model de dades, disseny del model ER, definició serveis web, disseny IU Web i IU mòbil, redacció i entrega de l'anàlisi i disseny)
- **Desenvolupament** (BBDD, capa de negoci, probes BBDD, serveis web, probes serveis web, aplicació Web MVC, aplicació mòbil, probes aplicacions, redacció i entrega dels manuals d'ús i instal·lació del producte final )
- **Lliurament Projecte** (probes i últimes correccions, redacció de la memòria del projecte, edició del vídeo de presentació, entrega del projecte)

### 2.6.2 Estimació de temps

Les estimacions de temps es poden veure en el diagrama de gantt de l'apartat 2.6.4 i estan calculades segons el temps que es disposa per a dedicar a la realització de les activitats descrites en l'apartat 2.6.1 i la dificultat de les mateixes.

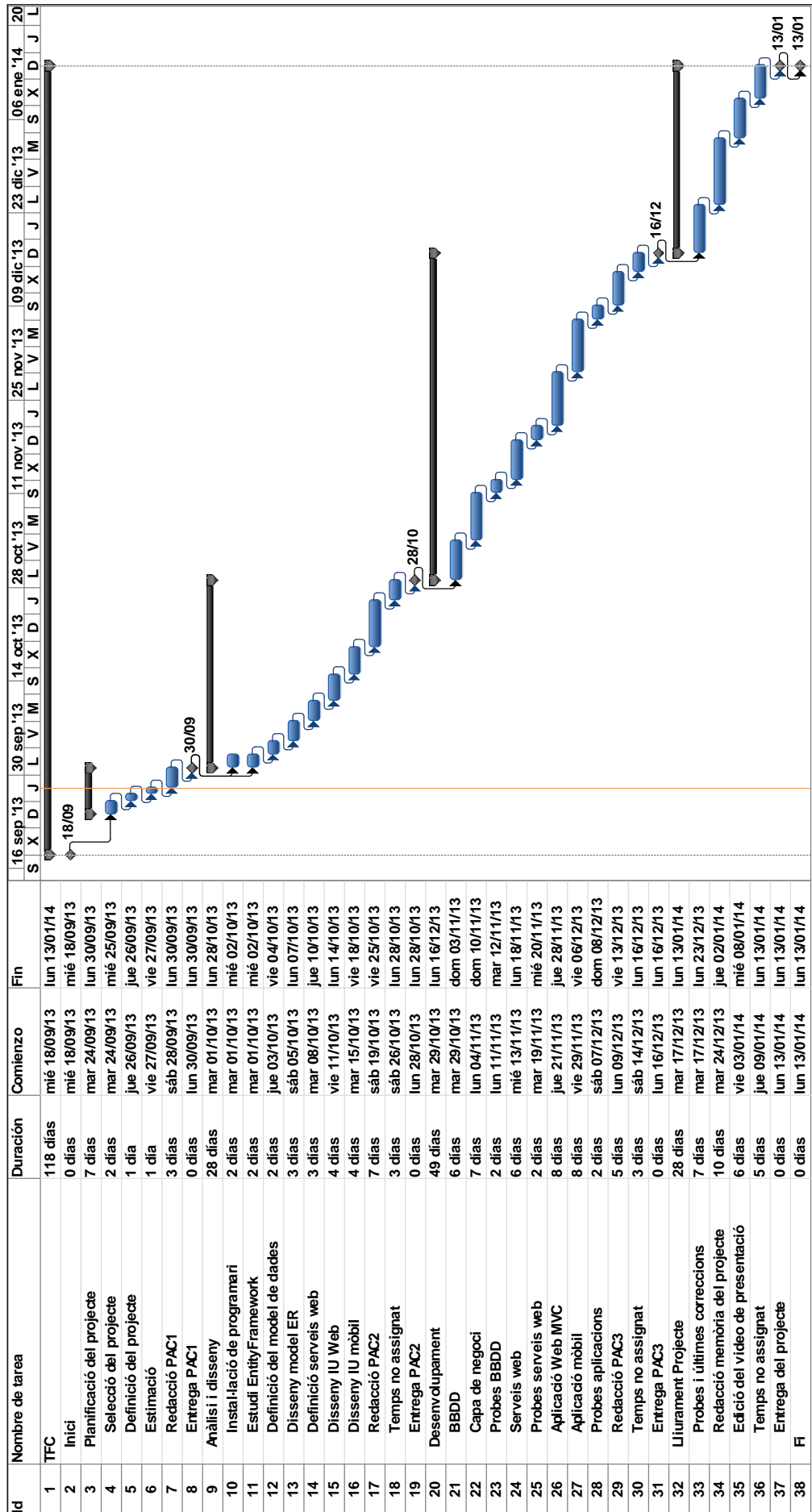
També s'ha deixat un marge de temps en cadascuna de les fites per a tenir un marge d'acció per imprevistos que puguin sorgir en la realització de les activitats.

### 2.6.3 Fites a complir

- **Planificació del projecte:** 30/09/2013
- **Anàlisis i disseny:** 28/10/2013
- **Desenvolupament:** 16/12/2013
- **Lliurament del projecte:** 13/01/2014
- **Defensa:** 20/01/2014.



## 2.6.4 Diagrama de Gantt



### 2.6.5 Temps dedicat

Totes les fites establertes a l'estimació inicial fins la data d'edició d'aquest document han estat finalitzades dins dels terminis establerts per aquestes, tot i que la distribució del temps dedicat a les subtasques d'aquestes fites ha variat en alguns casos.

El cas més notable ha estat el desenvolupament, degut a la tecnologia escollida per a la realització de l'aplicació web. A l'etapa de planificació, com a primera opció, es va escollir ASP.NET MVC com a tecnologia sobre la qual es basaria el desenvolupament d'aquesta aplicació. A la fase de disseny es va reconsiderar aquesta elecció i es va optar per ASP.NET Web Forms, ja que en un principi el desenvolupament amb aquesta tecnologia hauria de ser menys costós. La falta d'experiència amb Web Forms, juntament amb la resta de tecnologies escollides pel desenvolupament d'aquesta part del projecte, va provocar que utilitzés Web Forms de manera poc eficient produint uns costos de temps molt més elevats dels que es van calcular en un principi, gairebé el doble del temps assignat per aquesta tasca.

Els serveis web i la capa de negoci, tot i que s'han incorporat subtasques que en un principi no existien, s'han ajustat als temps estimats.

Per una altra banda el temps dedicat a l'aplicació per a Windows Phone va ser inferior a l'estimat inicialment, la falta de coneixements sobre aquesta tecnologia en el moment de realitzar la planificació em va fer sobreestimar els costos.

En total, es van dedicar més hores al desenvolupament del que es va estimar en un principi, en total unes 20h més de feina tenint en compte que es va pensar en una dedicació de 2h diàries al projecte.

## 3 Anàlisi i disseny

### 3.1 Requeriments

A continuació es fa un llistat dels requeriments funcionals i no funcionals del producte.

#### 3.1.1 Funcionals

- Un usuari podrà accedir a la plataforma com a treballador, responsable o administrador.
- L'usuari administrador serà capaç de donar d'alta o de baixa usuaris i modificar-los la paraula de pas.
- L'usuari administrador podrà assignar un responsable a un treballador. El responsable també és un usuari treballador donat d'alta prèviament.
- L'usuari administrador podrà visualitzar les fulles de despeses validades (amb l'objectiu d'abonar-les a la nòmina del més en curs).
- L'usuari administrador pot marcar una data límit a partir de la qual no es pot obrir ni validar més fulles de despeses pel més en curs.
- L'usuari administrador pot marcar el preu per unitat (ex: €/Km), valor màxim sense resguard i valor màxim amb resguard per cada tipus de despesa que existeix a l'aplicació i per dia festiu o laborable.
- L'usuari administrador pot especificar quins són els dies laborables i quins són els festius de cada més.
- L'usuari responsable podrà visualitzar tots els fulls de despeses en estat pendent de validar dels treballadors que té assignats.
- L'usuari responsable podrà validar o rebutjar cada full de despeses. En cas de rebuig ha d'indicar un motiu.
- L'usuari responsable també pot actuar com a usuari treballador.
- L'usuari treballador pot introduir i esborrar despeses al seu full de despeses actiu.
- L'usuari treballador ha de poder saber per què se li ha rebutjat una despesa.
- El treballador sol·licita el pagament d'un full de despeses quan ell vulgui, quedant el full pendent de validar.
- Les despeses estaran catalogades segons el seu tipus.
- Cada tipus de despesa tindrà uns requeriments establerts per l'administrador.
- Les despeses estaran indicades en un full de despeses.
- Cada despesa ha d'anar acompanyada d'un motiu.
- Cada despesa que superi el valor màxim sense resguard (dependrà del tipus de despesa) ha d'anar acompanyada d'un arxiu amb el rebut o factura.
- Un full de despeses pot tenir més d'una despesa.
- Només hi haurà un full de despeses actiu, per treballador, al mateix temps.
- Fins que no es valida un full de despeses no queda tancada.
- Un cop validat un full de despeses, les despeses seran abonades en la nòmina del més en curs de quan es va validar, sempre i quan la validació no excedeixi de la data límit, i automàticament s'obrirà un nou full.
- Tots els fulls que es validin després de la data límit que l'administrador hagi marcat passaran a formar part del següent més.

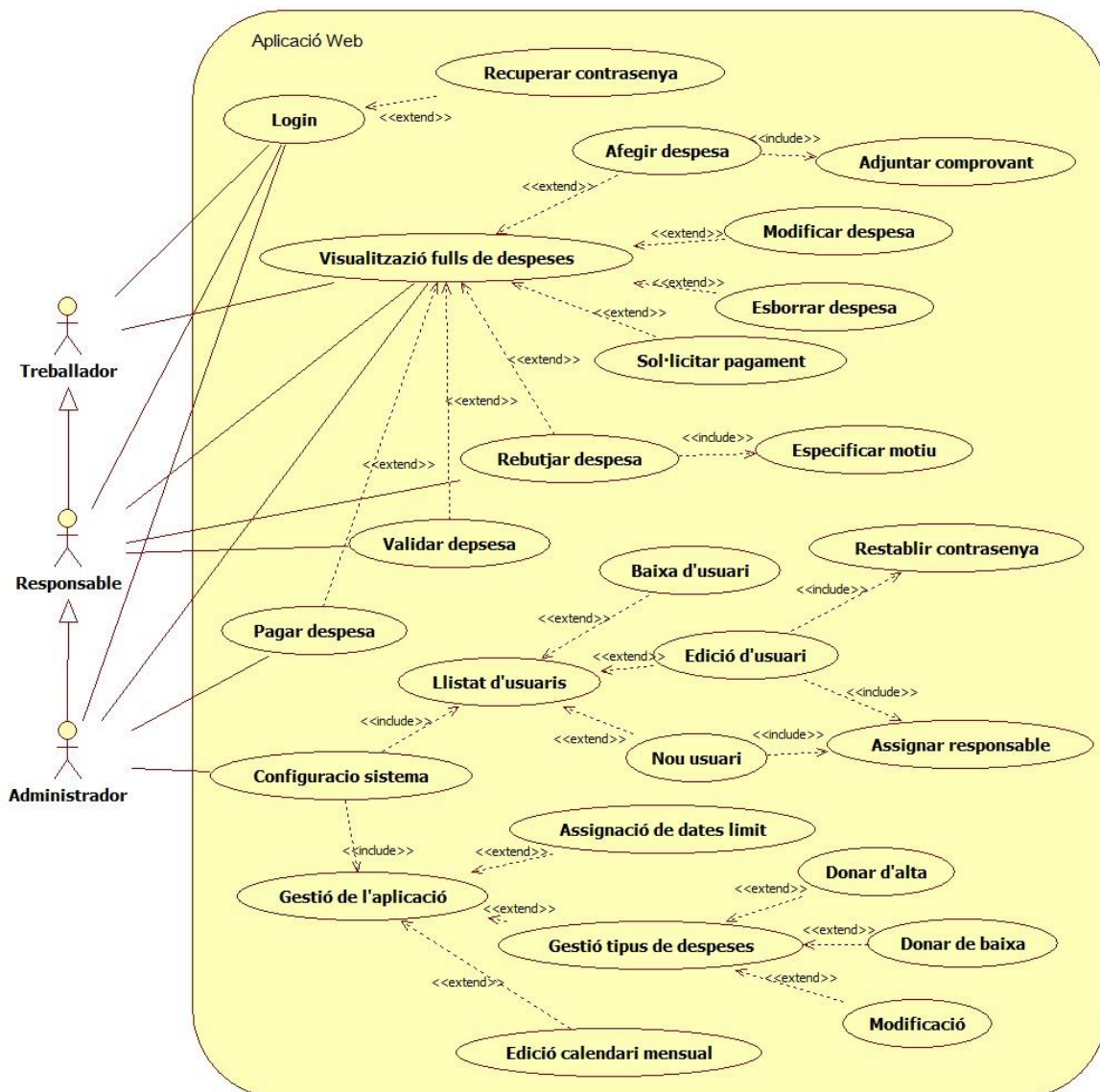
### 3.1.2 No funcionals

- Només el treballador, el seu responsable i l'administrador podran accedir a les despeses del propi treballador, assegurant la confidencialitat de les dades introduïdes.
- El sistema estarà operatiu les 24h del dia, exceptuant parades puntuals per manteniment, per tal que el treballador pugui introduir les despeses quan ell vulgui.
- L'accés estarà controlat per un sistema d'autenticació basat en usuari i paraula de pas.
- El sistema permetrà la connexió simultània de diferents usuaris.
- Es podrà accedir al sistema des de fora de la xarxa de l'empresa.
- El treballador podrà introduir les despeses mitjançant una aplicació web o una aplicació per a mòbils.

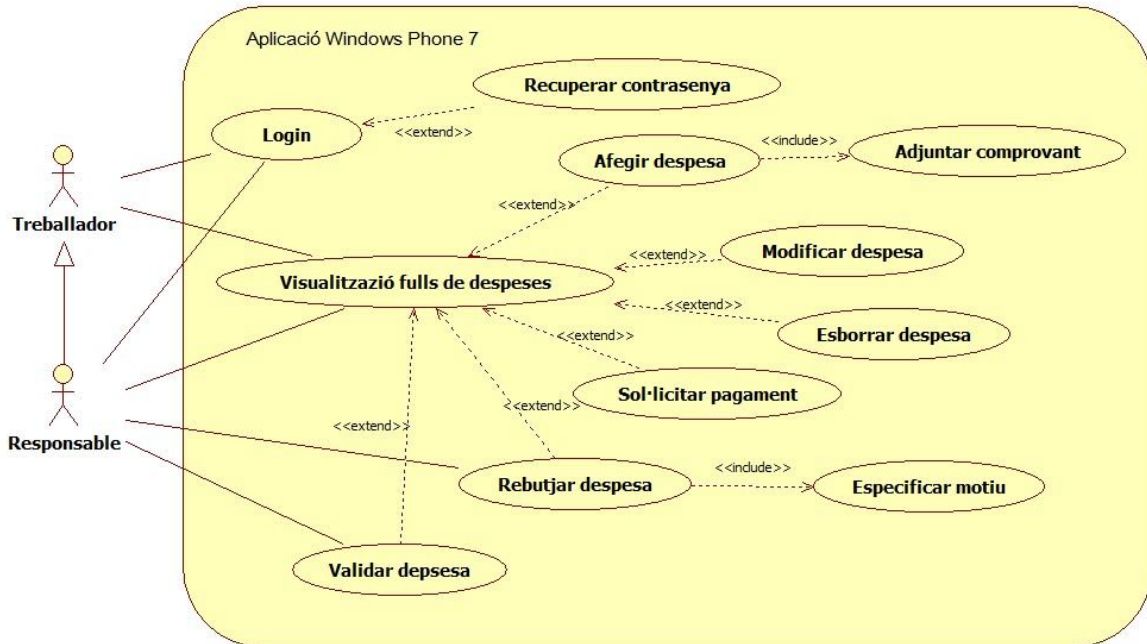
## 3.2 Diagrama de casos d'ús

L'arquitectura del sistema està dividida en tres parts, però només amb dues d'aquestes parts és on els usuaris interactuaran, que són l'aplicació web i l'aplicació per a mòbil. Els actors que s'han definit pel sistema són tres, que seran l'usuari administrador, l'usuari treballador i l'usuari responsable.

Pel que fa a l'aplicació web tots tres actors poden interactuar amb la plataforma. L'usuari administrador té les funcionalitats de gestió i administració de la plataforma, l'usuari treballador té les funcionalitats d'ús de la plataforma i l'usuari responsable, que també actua a la vegada com a usuari treballador, té funcionalitats d'ús que no té l'usuari treballador.

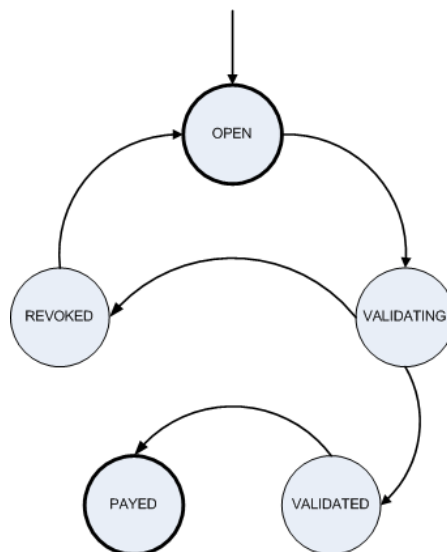


Per a l'aplicació mòbil l'actor administrador desapareix, i es queden els altres dos actors que tenen exactament les mateixes funcionalitats per a l'aplicació mòbil que per a l'aplicació web.



### 3.3 Diagrama d'estat

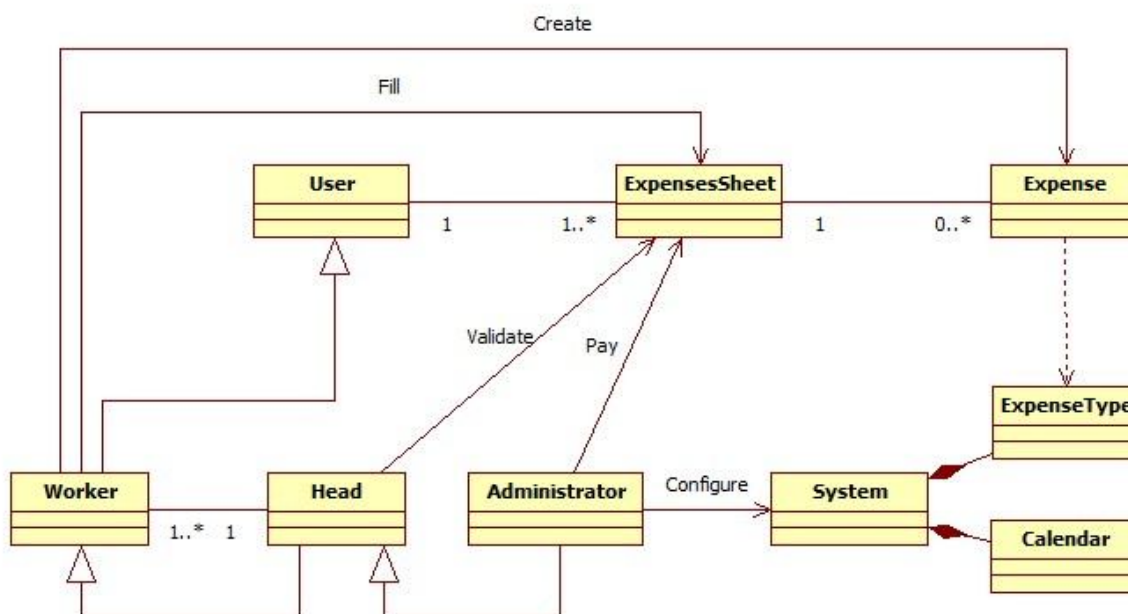
Un full de despeses passa per diferents estats abans d'arribar a l'estat final de pagat. En el següent esquema es pot veure d'una manera visual quins són aquests estats i com es pot arribar a cadascun d'ells



Només hi pot haver un full de despeses "actiu" per treballador en un moment donat, tenint en compte que "actiu" es refereix al fet que encara es poden fer modificacions en el seu contingut, i això correspon amb els estats *Open*, *Validating* i *Revoked*. Un cop un full a

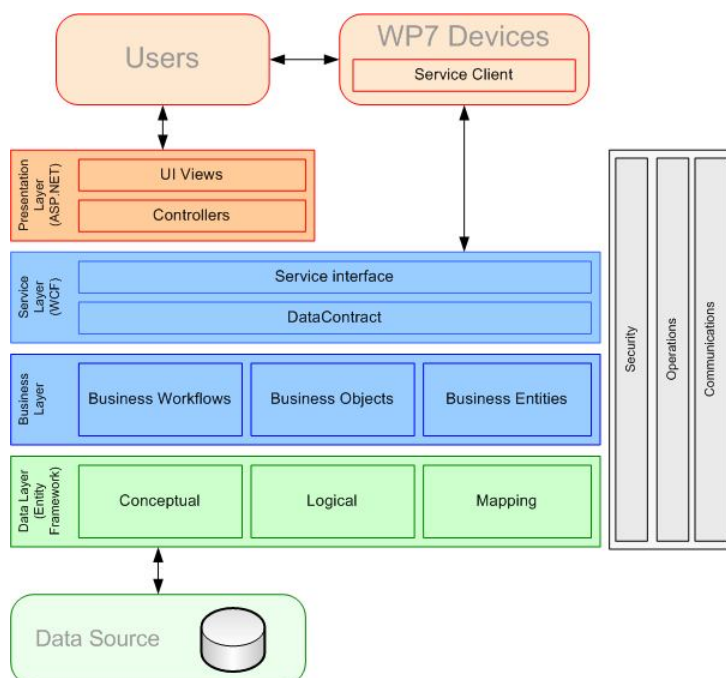
passat a l'estat *Validated* no es pot fer cap més modificació en el seu contingut i per tant s'ha de crear un full nou (en estat inicial *Open*) per tal que el treballador pugui anar afegint despeses.

### 3.4 Model conceptual

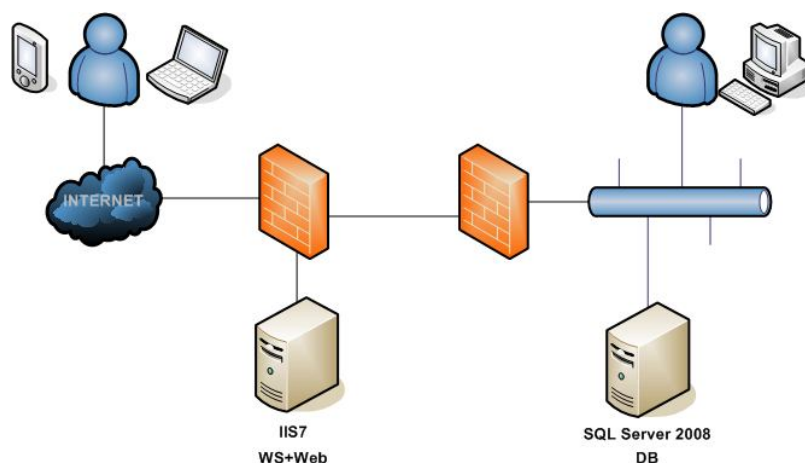


### 3.5 Diagrama d'arquitectura: d'aplicació / SW / HW

En el següent diagrama es pot veure quines són del diferents capes que conformen l'arquitectura de l'aplicació.

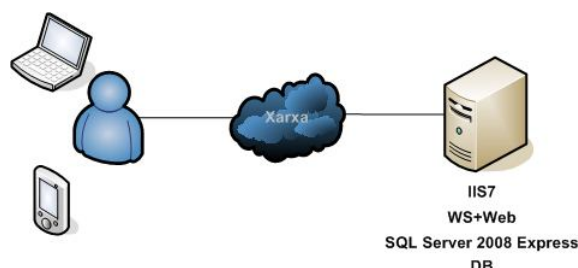


L'arquitectura HW del client es desconeguda de moment, així que es parteix d'una arquitectura típica per fer una primera aproximació a com quedarà en el client:



L'entorn ideal és el que es mostra al diagrama anterior, on tenim la base de dades a la zona segura i el servidor (IIS7) a la zona pública. En aquesta arquitectura tant els treballadors que es troben fora de la xarxa com els que es troben a dins poden accedir al servidor IIS7.

Per l'entorn de desenvolupament es disposa únicament d'un servidor on s'allotjaran tots els serveis, per tant, l'arquitectura és la següent.

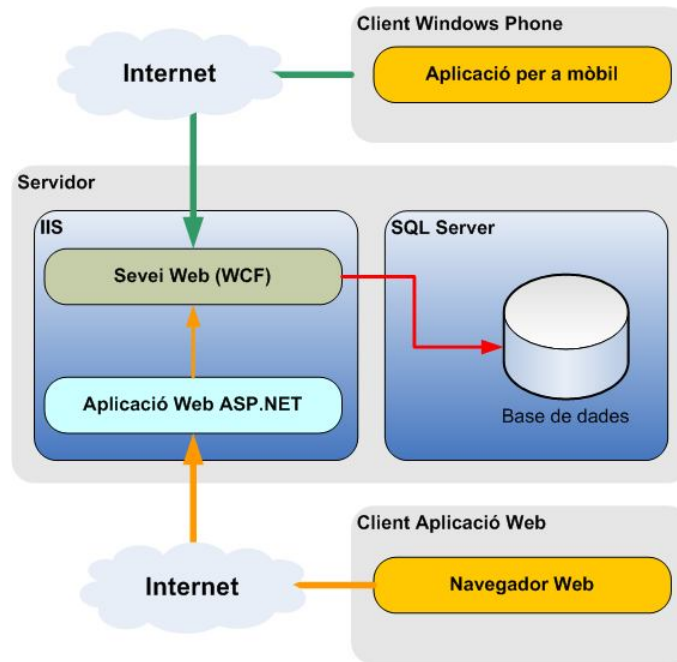


Es tindrà en compte l'arquitectura HW de desenvolupament com a referència a la resta del document. Tenim dues parts ben diferenciades dins de l'arquitectura de l'aplicació, que són la part servidor i la part client.

A la part servidor disposarem d'un servidor web (IIS) i un servidor de bases de dades (SQL Server). A la base de dades s'emmagatzemarà tota la informació necessària pel funcionament correcte de l'aplicació. El servidor web tindrà allotjades dues aplicacions: un servei web que contindrà tota la capa de negoci de l'aplicació i que s'encarregarà de l'accés a les dades i per una altra banda una aplicació web que farà ús dels serveis web, on els usuaris es connectaran, mitjançant un navegador web, per interactuar amb la plataforma. Als serveis web també es podrà accedir de manera remota mitjançant un dispositiu WP7, per tal d'oferir la funcionalitat requerida per l'aplicació als treballadors que, per motius de mobilitat, no disposin d'un ordinador amb connexió a Internet per a poder introduir les despeses.



A la part client ens trobem amb dues possibles situacions, clients que es connecten mitjançant un navegador des d'un ordinador i clients que es connecten mitjançant un dispositiu mòbil amb l'aplicació de gestió de despeses instal·lada.

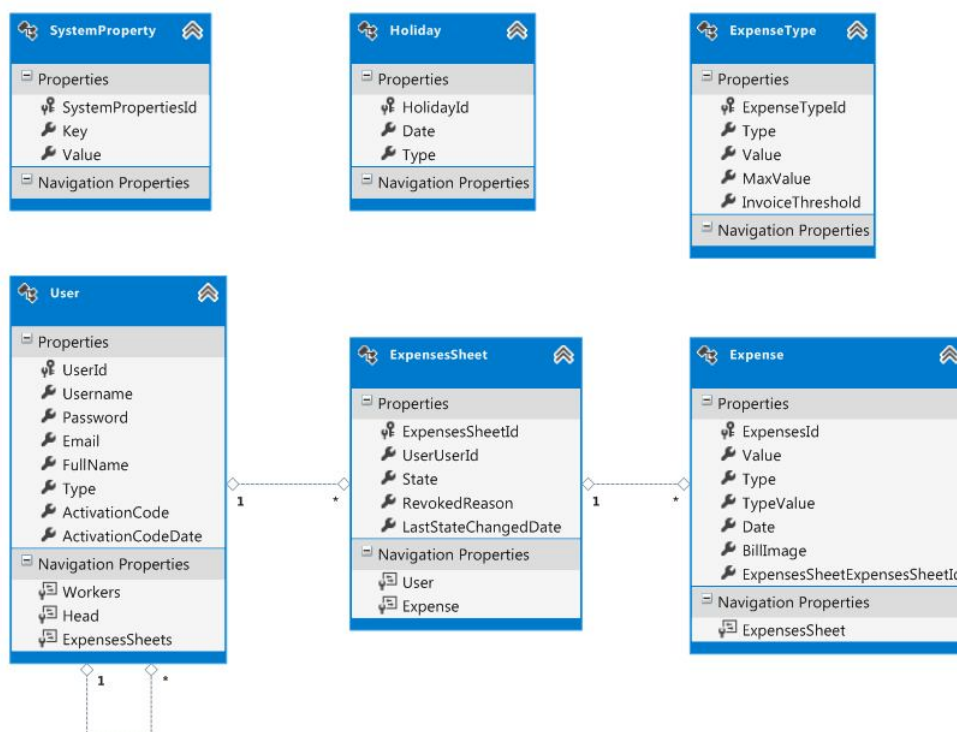


### 3.6 Disseny de la BD / Diagrama E-R

El fet de treballar amb Entity Framework ha permès la creació del diagrama ER a partir de la pròpia eina que incorpora el software (*Entity Framework Designer*) utilitzant la metodologia *Model-First*.

S'ha trobat a faltar la possibilitat de definir columnes amb valor únic (UNIQUE). El problema s'ha esmenat creant un script que s'ha d'executar seguidament del script que genera el *EF Designer* per crear la base de dades.

El model final és una base de dades molt senzilla però que cobreix totes les necessitats del projecte.



Les columnes amb valor UNIQUE són *User.Username*, *User.Email*, *User.ActivationCode* i *SystemProperties.Key*. D'aquesta manera ens assegurem que dos usuaris no poden tenir ni el mateix usuari ni el mateix correu electrònic ni el mateix codi d'activació i també que no hi ha dues propietats del sistema amb el mateix nom.

Els elements centrals de la base de dades són els usuaris i les despeses, que es relacionen mitjançant els fulls de despeses. Un usuari (*User*) pot tenir un o més fulls de despeses (*ExpensesSheet*), i un full de despeses pot tenir una o més despeses (*Expense*).

La taula *ExpenseType* és on l'administrador definirà els diferents tipus de despeses que es poden assignar a un full de despeses, i serveixen com a plantilla per crear una entrada a la taula *Expense*. El fet que no existeixi una relació entre *ExpenseType* i *Expense* és que un cop un full de despeses ha estat validat no es pot modificar. Les despeses que s'hagin inclòs en un full i que el seu tipus hagi estat modificat abans de ser validades es

visualitzaran de manera diferent en els llistats per tal d'avisar al responsable, o al propi treballador, que hi ha hagut un canvi.

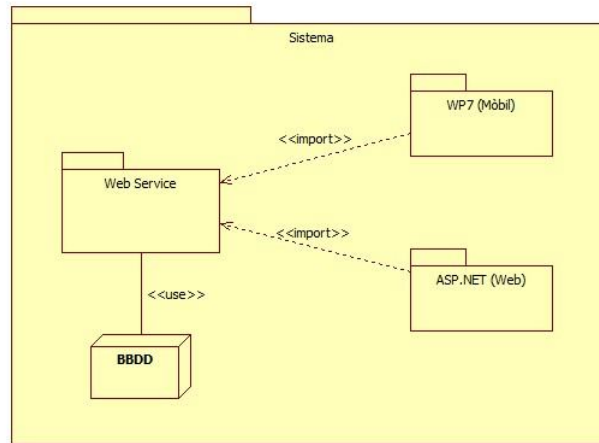
La taula *SystemProperty* tindrà la configuració del SMTP.

La taula *Holiday* contindrà les dates dels dies festius que marqui l'usuari Administrador.

Totes les taules del model tenen el seu identificador intern, que es genera automàticament en fer una inserció.

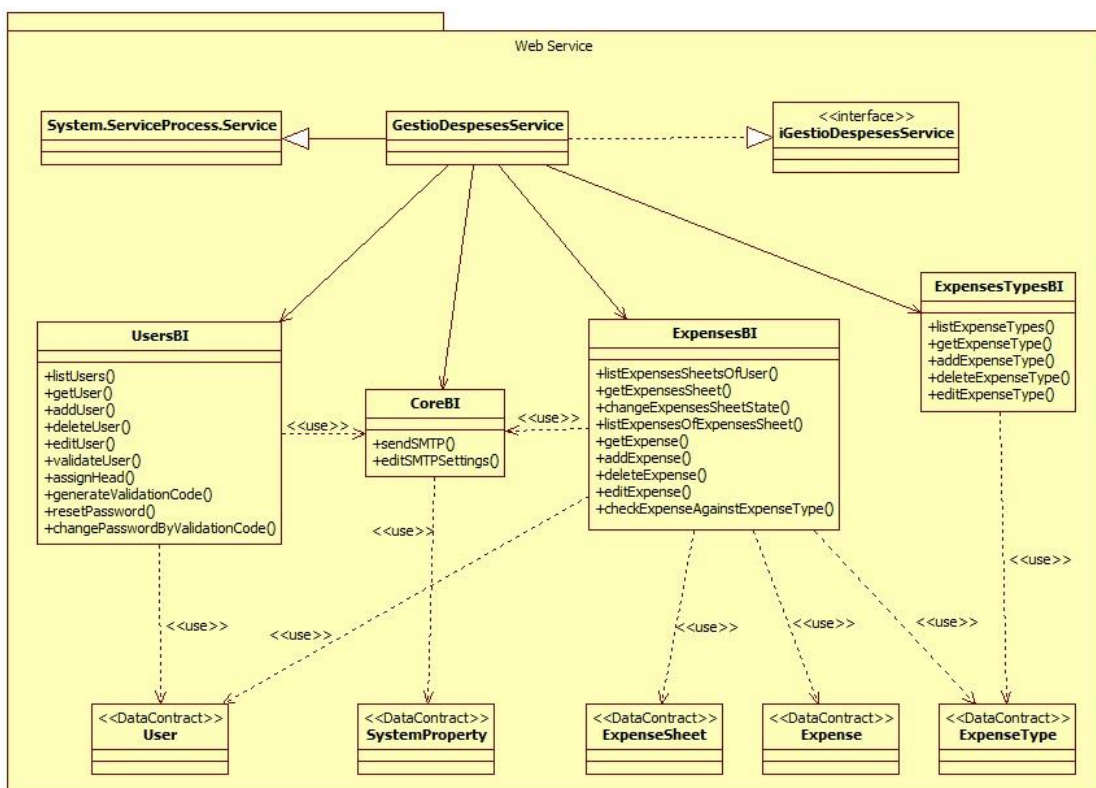
### 3.7 Model de classes

El model de classes o diagrama de classes de l'aplicació està compost per quatre paquets. El Web Service és el que conté tota la lògica de negoci i les classes d'accés a les dades (generades per EF). Les aplicacions client (WP7 i ASP.NET) faran ús del Web Service:



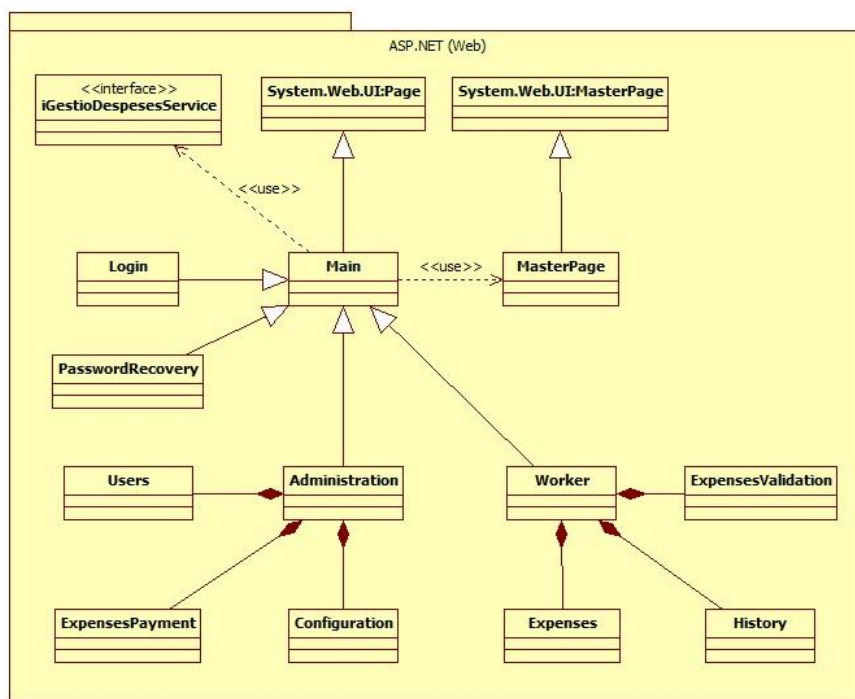
#### 3.7.1 Web Service (WCF)

El servei web, com ja s'ha comentat en diverses ocasions, és el nucli del sistema i conté tota la lògica de negoci i l'accés a les dades.



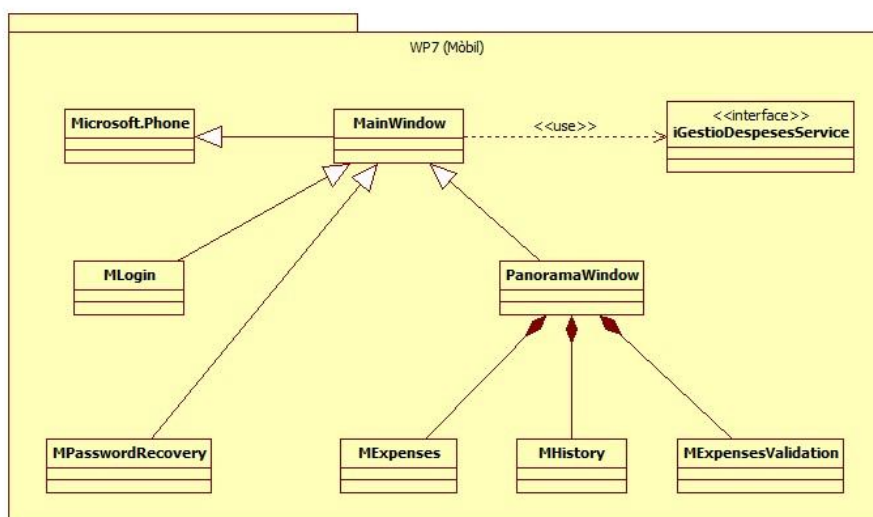
### 3.7.2 Aplicació Web (ASP.NET)

L'aplicació web és l'aplicació principal del sistema i que engloba tant les funcionalitats d'administració del sistema com les funcionalitats pels treballadors. Es tracta d'una aplicació ASP.NET que farà ús dels serveis web.



### 3.7.3 Aplicació Mòbil (WP7)

Igual que l'aplicació web, l'aplicació WP7 farà crides als serveis web. Aquesta aplicació no té incorporades les funcionalitats d'administració del sistema, només les dels treballadors.



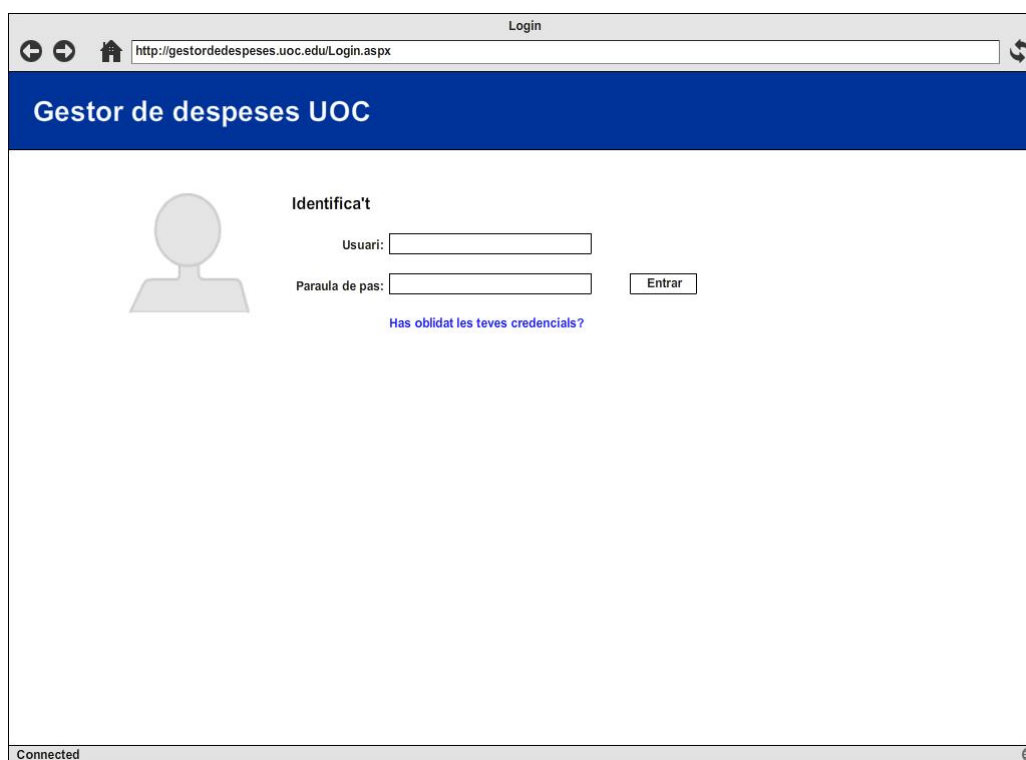
## 3.8 Disseny de la interfície d'usuari

### 3.8.1 Interfície Web

#### Inici, autenticació i recuperar credencials

En accedir a l'aplicació web el primer que es veu és la pàgina d'autenticació d'usuaris. En aquesta pàgina se sol·licita a l'usuari que introdueixi les seves credencials d'accés (usuari i paraula de pas) per poder continuar. En cas que l'usuari hagi oblidat la seva paraula de pas hi ha un enllaç que permet que l'usuari la canviï.

La pàgina d'autenticació és comú pels tres tipus d'usuari: Administrador, Treballador i Responsable.



Per motius de seguretat la paraula de pas no es guarda directament a la base de dades, sinó que es guarda un resum (MD5) de la mateixa, per tant, el sistema no pot enviar-li a l'usuari quina és la seva paraula de pas.

El procediment per restaurar les credencials és el següent:

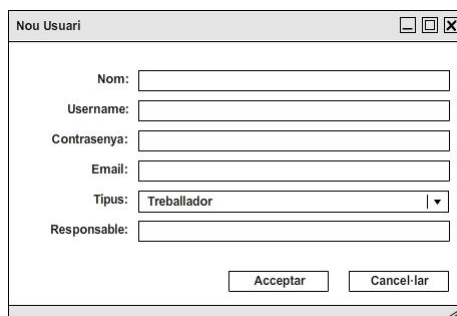
1. L'usuari ha d'introduir el seu usuari o correu electrònic i prem el botó *Validar*
2. L'aplicació envia a l'usuari un correu amb un codi d'activació.
3. Un cop l'usuari té el codi d'activació, l'introdueix al formulari que hi ha a la pàgina juntament amb la nova paraula de pas que vol fer servir.

## Administració d'usuaris

Només l'usuari administrador podrà accedir a la pàgina d'administració d'usuaris. En aquest punt l'usuari tindrà una llista amb els usuaris que hi ha donats d'alta a la plataforma on cada fila correspon amb un usuari i tindrem unes accions associades a cada fila, que seran *Editar*, *Esborrar*, *Canviar contrasenya* i *Assignar responsable*.

Usuari	Tipus	Username	Email	Accions
Administrador	Admin	admin	admin@example.com	
Usuari 1	Treballador	user1	user1@example.com	
Usuari 2	Treballador	user2	user2@example.com	
Usuari 3	Treballador	user3	user3@example.com	
Usuari 4	Responsable	user4	user4@example.com	
Usuari 5	Treballador	user5	user5@example.com	
Usuari 6	Treballador	user6	user6@example.com	
Usuari 7	Treballador	user7	user7@example.com	

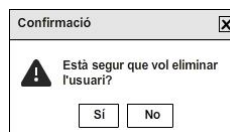
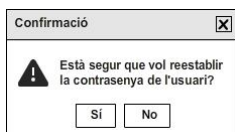
El botó de *Nou Usuari* ens mostrarà una finestra modal com la que tenim a continuació on es poden incloure les dades del nou usuari: *Nom*, *Usuari*, *Contrasenya*, *Email*, *Tipus d'usuari* i finalment *Responsable* de l'usuari. La finestra d'edició d'usuaris serà igual, amb la limitació que el camp *Usuari* no serà editable.



L'acció d'assignar responsable mostrarà també una finestra modal on s'ha d'introduir l'usuari que serà responsable.



Les accions de restablir contrasenya i d'esborrar usuari mostraran un missatge de confirmació de l'acció.



En restablir la contrasenya el sistema generarà una de nova automàticament que mostrarà a l'administrador per pantalla i que enviarà per correu a l'usuari. D'aquesta manera, si per algun motiu l'usuari no té accés al correu, podrà entrar al sistema amb la contrasenya que l'administrador li faciliti.

## Pagament de fulls validats

Només l'usuari administrador podrà accedir a la pàgina de pagament de despeses. Aquesta pàgina mostra un llistat amb totes les despeses que estan validades i pendents de pagament, de manera que quan l'administrador introdueix la quantitat al sistema de pagaments de la nòmina ja pot marcar com a pagat el full de despeses.



Les accions que pot fer l'usuari per cada full són visualitzar en detall el full de despeses i marcar com a pagada la despesa.

Administration  
http://gestordedespeses.uoc.edu/Administration.aspx

## Gestor de despeses UOC

Administrador  
Sortir

Usuaris Despeses Configuració

Despeses validades per pagar

Usuari	Quantitat	Accions
Usuari 1	350€	
Usuari 3	1127.32€	
Usuari 5	23.12€	
Usuari 7	25.12€	

Connected

L'acció de veure els detalls mostra una finestra modal amb un llistat de totes les despeses que hi ha al full de despeses i permet a l'usuari administrador marcar com a pagada la despesa des de la mateixa finestra de visualització.

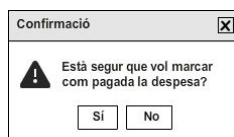
Detalls despesa

Concepte	Tipus	Data	Valor	Festiu	Tiquet	Total(€)
Parking	Variable	20/10/2013	13.83	Si	Si	15.21
Dieta Dinar	Variable	20/10/2013	9.8	Si	Si	10.78
Parking	Variable	22/10/2013	8.02	No	No	8.02
Dieta Dinar	Variable	22/10/2013	9.8	No	Si	9.8
Dieta Sopar	Fixe	25/10/2013	12	No	No	12
Kilometratge	Multiplicatiu	25/10/2013	75	No	No	14.25
Parking	Variable	27/10/2013	3.53	Si	Si	3.88
Dieta Dinar	Variable	27/10/2013	15.75	Si	Si	17.32
Dieta Sopar	Fixe	27/10/2013	12	Si	No	13.2

Despeses: 9  
Total: 104.46€

Pagar Tancar

En marcar com pagada una despesa apareix un missatge de confirmació de l'acció que s'està realitzant.



## Configuració del sistema

Només l'usuari administrador podrà accedir a la pàgina de configuració del sistema. En aquesta pàgina es pot configurar diverses coses.

En primer lloc es pot configurar el servidor SMTP, essencial per a poder enviar correus als usuaris del sistema.

Per una altra banda es pot configurar els dies festius mitjançant el calendari. Els dies festius fan que les despeses que s'originin dins d'aquests dies se'ls aplica un increment (per exemple 10%) per compensar. Al calendari també es podrà fixar la data límit per validar fulls de despeses.

Per últim es poden configurar els tipus de despeses que existeixen al sistema. En aquesta pàgina apareix un llistat on cada fila correspon amb un tipus de despesa donada d'alta.

Administration  
http://gestordedespeses.uoc.edu/Administration.aspx

**Gestor de despeses UOC**

Administrador [Sortir](#)

Usuaris Despeses **Configuració**

**Configuració del sistema**

SMTP

Host:  Port:   TLS

From:

Usuari:  Contrasenya:

Calendari

< August 2009 >

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Tipus de despeses

[Afegir despesa](#)

Concepte	Tipus	Valor-Valor per unitat(€)	Valor màxim	Tiquet a partir de	Accions
Parking	Variable	-	50	10	
Kilometratge	Multiplicatiu	0.19	-	10	
Dieta Dinar	Variable	-	24	10	
Dieta Sopar	Fixe	12	-	-	

Connected

En polsar el botó *Afegir despesa* apareix una finestra modal que permet a l'usuari configurar una nova despesa. En editar una despesa apareix la mateixa finestra amb les dades omplertes.

Nou Tipus de despesa

Nom:

Tipus:

Valor:

Valor màxim:

Tiquet a partir de:

En esborrar una despesa es demanarà confirmació de l'acció que es vol dur a terme.

Confirmació

⚠ Està segur que vol esborrar el tipus de despesa?

## Gestió de despeses

Només l'usuari Treballador podrà accedir a la pàgina de gestió de despeses. Recordem que l'usuari Responsable és també usuari Treballador, per tant, també podrà accedir a aquesta pàgina.

En un principi es mostra un llistat que correspon al full de despeses actiu de l'usuari. En aquest llistat l'usuari introdueix despeses mitjançant el botó *Afegir despesa*. Un cop l'usuari creu que el seu full de despeses és correcte prem el botó *Enviar a validar* per tal que el seu responsable el validi o rebutgi.

Les accions que es poden fer per a cada despesa són *Editar* i *Esborrar*.


Despeses

http://gestordedespeses.uoc.edu/Despeses.aspx

### Gestor de despeses UOC

Usuari 1  
[Sortir](#)

Despeses **Historic**

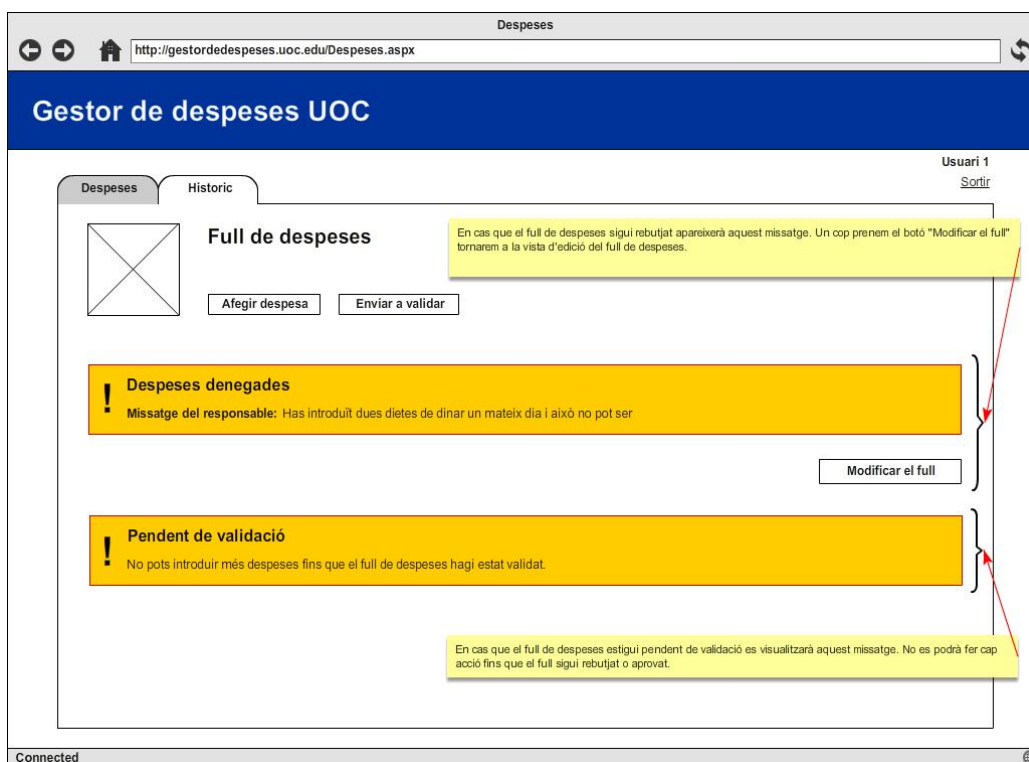
 **Full de despeses**

Concepte	Tipus	Data	Valor	Festiu	Tiquet	Total(€)	Accions
Parking	Variable	20/10/2013	13.83	Sí	Sí	15.21	
Dieta Dinar	Variable	20/10/2013	9.8	Sí	Sí	10.78	
Parking	Variable	22/10/2013	8.02	No	No	8.02	
Dieta Dinar	Variable	22/10/2013	9.8	No	Sí	9.8	
Dieta Sopar	Fixe	25/10/2013	12	No	No	12	
Kilometratge	Multiplicatiu	25/10/2013	75	No	No	14.25	
Parking	Variable	27/10/2013	3.53	Sí	Sí	3.88	
Dieta Dinar	Variable	27/10/2013	15.75	Sí	Sí	17.32	
Dieta Sopar	Fixe	27/10/2013	12	Sí	No	13.2	

Despeses: 9  
Total: 104.46€

Connected

Hi ha dues situacions en què en accedir a aquesta pàgina no podrem fer res. Aquestes situacions són quant s'ha enviat un full a validar, situació en la qual s'ha d'esperar a què el responsable validi o rebutgi el full de despeses, i per una altra banda quan l'usuari responsable rebutja un full, cas en el qual l'usuari veurà per pantalla el motiu pel qual se li ha rebutjat el full i se li donarà accés de nou perquè el pugui modificar.



En afegir una despesa apareix un formulari com el següent en el qual l'usuari introdueix les dades de la despesa que vol donar d'alta en el full de despeses.

Com en els casos anteriors, abans d'eliminar una despesa es demanarà confirmació de l'acció que es vol realitzar.

## Històric de despeses

Només l'usuari Treballador(i Responsable) podrà accedir a la pàgina d'històric de despeses. En aquesta pàgina es mostra un llistat de tots els fulls de despeses de l'usuari, l'estat en el qual es troben i quina és la quantitat de diners corresponent.

L'única acció que es pot realitzar sobre una despesa és veure-hi els detalls de la mateixa.

Despeses

Historic

Usuari 1  
Sortir

Historic de despeses

Estat	Quantitat	Data	Accions
Oberta	104.46€	27/10/2013	
Validada	1127.32€	14/09/2013	
Pagada	23.12€	28/07/2013	
Pagada	25.12€	15/07/2013	

Connected

En veure els detalls apareix un llistat amb totes les despeses que hi han donades d'alta al full de despeses corresponent.

Detalls despesa

Concepte	Tipus	Data	Valor	Festiu	Tiquet	Total(€)
Parking	Variable	20/10/2013	13.83	Sí	Sí	15.21
Dieta Dinar	Variable	20/10/2013	9.8	Sí	Sí	10.78
Parking	Variable	22/10/2013	8.02	No	No	8.02
Dieta Dinar	Variable	22/10/2013	9.8	No	Sí	9.8
Dieta Sopar	Fixe	25/10/2013	12	No	No	12
Kilometratge	Multiplicatiu	25/10/2013	75	No	No	14.25
Parking	Variable	27/10/2013	3.53	Sí	Sí	3.88
Dieta Dinar	Variable	27/10/2013	15.75	Sí	Sí	17.32
Dieta Sopar	Fixe	27/10/2013	12	Sí	No	13.2

Despeses: 9  
Total: 104.46€

Tancar

## Validació de despeses

Només l'usuari Responsable podrà accedir a la pàgina de Despeses per validar. En aquesta pàgina es mostra un llistat de les despeses dels usuaris que estan a càrrec de l'usuari responsable i que tenen el full de despeses pendent de validar.

Les accions que es poden realitzar sobre cada despesa són: *Veure detalls*, *Validar* i *Rebutjar*. En aquest últim cas l'usuari haurà d'introduir una raó per la qual el full de despeses ha estat rebutjat.

Usuari	Quantitat	Accions
Usuari 1	104.46€	
Usuari 3	1127.32€	
Usuari 5	23.12€	
Usuari 7	25.12€	

L'acció de veure els detalls d'una despesa obre una finestra modal on es pot veure un llistat amb els detalls de cada una de les despeses que conformen el full de despeses.

El botó de *Validar* valida el full de despeses des de la mateixa finestra per tal de no haver de tancar i validar en el llistat de fulls de despeses. El mateix passa amb el botó *Rebutjar*, que permet rebutjar el full de despeses sempre i quan s'hagi introduït un motiu pel qual es rebutja el full.

Concepte	Tipus	Data	Valor	Festiu	Tiquet	Total(€)
Parking	Variable	20/10/2013	13.83	Si	Si	15.21
Dieta Dinar	Variable	20/10/2013	9.8	Si	Si	10.78
Parking	Variable	22/10/2013	8.02	No	No	8.02
Dieta Dinar	Variable	22/10/2013	9.8	No	Si	9.8
Dieta Sopar	Fixe	25/10/2013	12	No	No	12
Kilometratge	Multiplicatiu	25/10/2013	75	No	No	14.25
Parking	Variable	27/10/2013	3.53	Si	Si	3.88
Dieta Dinar	Variable	27/10/2013	15.75	Si	Si	17.32
Dieta Sopar	Fixe	27/10/2013	12	Si	No	13.2

Despeses: 9  
Total: 104.46€

Motiu:

### 3.8.2 Interfície mòbil

La interfície mòbil té les mateixes funcionalitats que la interfície web exceptuant les tasques d'administració del sistema, com són la gestió d'usuaris, pagament de despeses i configuració del sistema.

#### Inici, autenticació i recuperar credencials

De la mateixa manera que en l'aplicació web el primer que veure'm després d'iniciar l'aplicació és la pantalla d'autenticació d'usuaris. En aquesta pantalla se sol·licita a l'usuari que introdueixi les seves credencials d'accés (usuari i paraula de pas) per poder continuar. En cas que l'usuari hagi oblidat la seva paraula de pas hi ha un enllaç que permet que l'usuari la canviï.

En aquest cas aquesta pantalla està adreçada només als usuaris Treballador i Responsable.

The image shows two mobile application screens for the 'Gestor despeses UOC' application. The left screen is the 'Login' screen, featuring a title 'Gestor despeses UOC' and 'Login:'. It has two input fields: 'Usuari:' and 'Contrasenya:'. Below the password field is a link that says 'Has oblidat la teva contrasenya?'. A 'Login' button is at the bottom right. The right screen is the 'Recuperar Credencials' screen, also titled 'Gestor despeses UOC'. It has an input field for 'Usuari o Email:' and a 'Validar' button. Below this is a yellow information box with an 'i' icon and the text: 'Se l'enviarà un codi d'activació a la teva bústia. Si ja tens el teu codi d'activació introdueix-lo al següent formulari juntament amb les noves credencials'. Below the box are three input fields: 'Codi:', 'Contrasenya:', and 'Confirmació:'. A 'Recuperar' button is at the bottom right. Both screens have a bottom navigation bar with a back arrow, a Windows logo, and a search icon.

El mètode de recuperació de credencials és el mateix que per a l'aplicació web:

1. L'usuari ha d'introduir el seu usuari o correu electrònic i prem el botó *Validar*
2. L'aplicació envia a l'usuari un correu amb un codi d'activació.
3. Un cop l'usuari té el codi d'activació, l'introdueix al formulari que hi ha a la pàgina juntament amb la nova paraula de pas que vol fer servir.

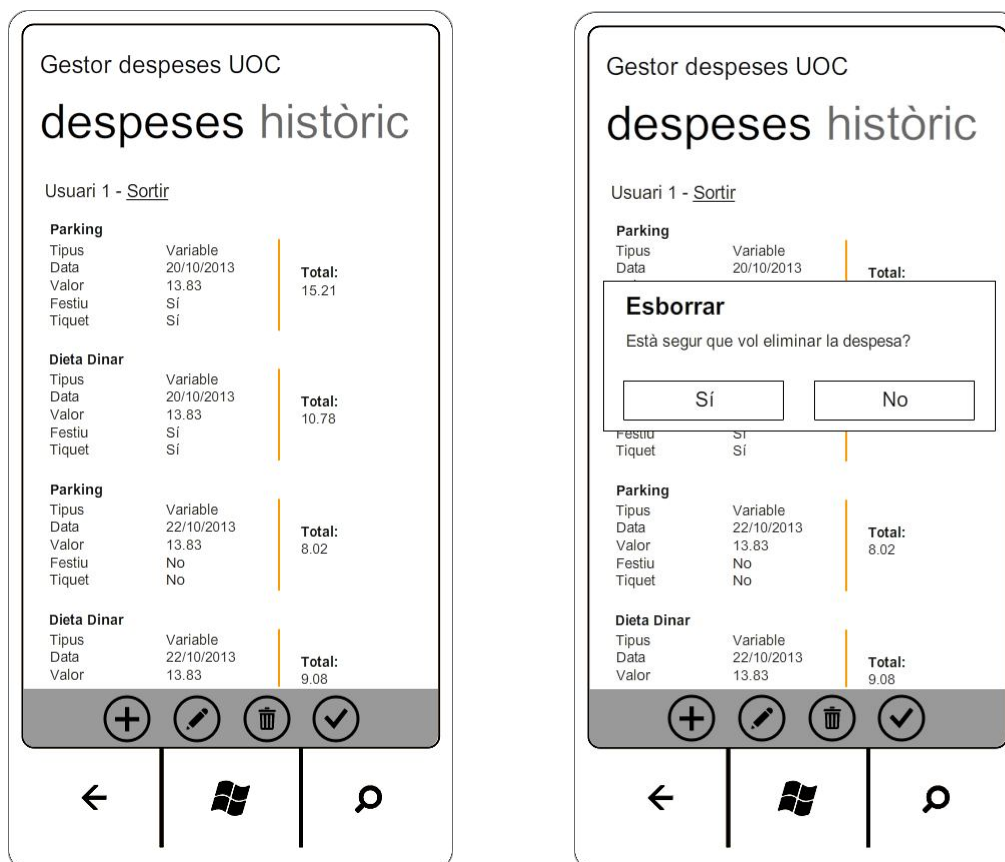


## Gestió de despeses

En aquest cas tenim una vista “panorama” on es pot passar per “Gestió de despeses”, “Històric de despeses” i “Validació de despeses”(aquest últim només disponible per l'usuari Responsable).

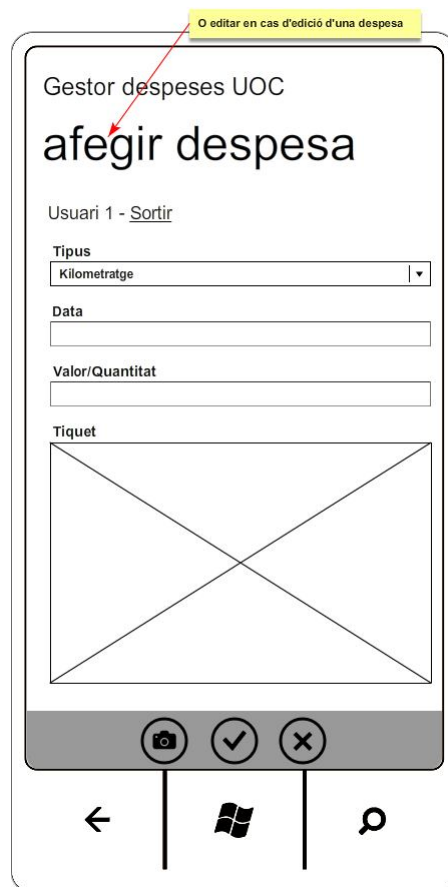
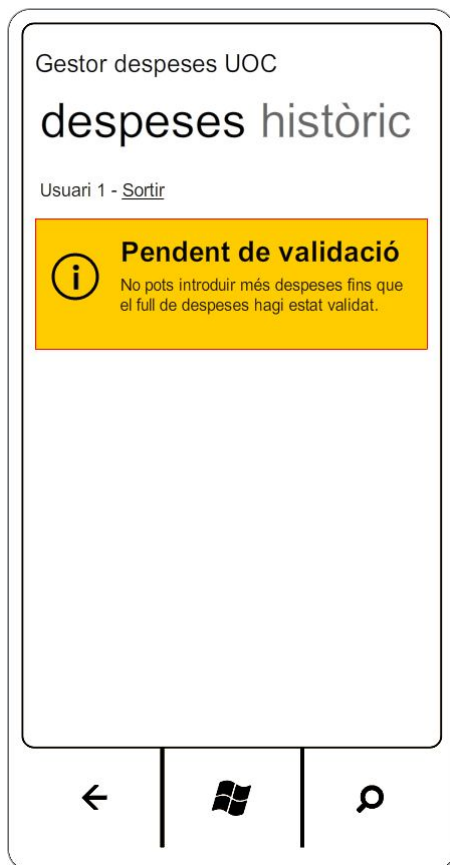
En un principi es mostra un llistat que correspon al full de despeses actiu de l'usuari. En aquest llistat l'usuari introdueix despeses mitjançant el botó situat a la barra inferior de la pantalla. Un cop l'usuari creu que el seu full de despeses és correcte prem el botó per tal que el seu responsable el validi o el rebutgi.


Seleccionant una despesa de la llista les accions que es poden fer per a cada despesa són *Editar* i *Esborrar* .



Totes les accions que modifiquin un estat mostraran un missatge de confirmació, tal com es pot veure en la pantalla d'esborrat d'una despesa.

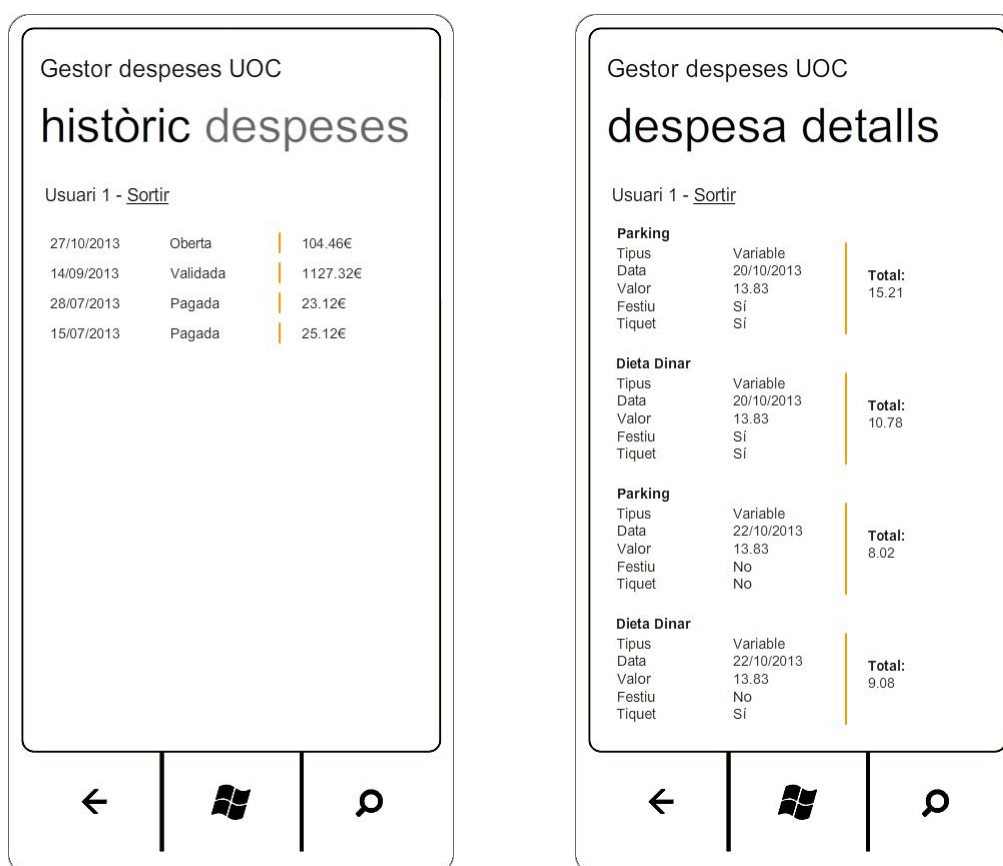
Hi ha dues situacions en què al accedir a aquesta pantalla no podrem fer res. Aquestes situacions són quan s'ha enviat un full a validar, situació en la qual s'ha d'esperar a què el responsable validi o rebutgi el full de despeses, i per una altra banda quan l'usuari responsable rebutja el full, cas en el qual l'usuari veurà per pantalla el motiu pel qual se li ha rebutjat el full i se li donarà accés de nou perquè el pugui modificar.



En afegir una despesa apareix un formulari com l'anterior en el qual l'usuari introdueix les dades de la despesa que vol donar d'alta en el full de despeses. El botó  permetrà introduir una imatge com a tiquet/comprovant de la despesa, i preguntarà a l'usuari si vol escollir una imatge de la seva galeria d'imatges o obrir l'aplicació de la càmera de fotos per capturar una imatge nova.

## Històric de despeses

Aquesta pantalla forma part de la vista panorama. En aquesta pantalla es mostra un llistat de tots els fulls de despeses de l'usuari, l'estat en el qual es troben i quina és la quantitat de diners corresponent.

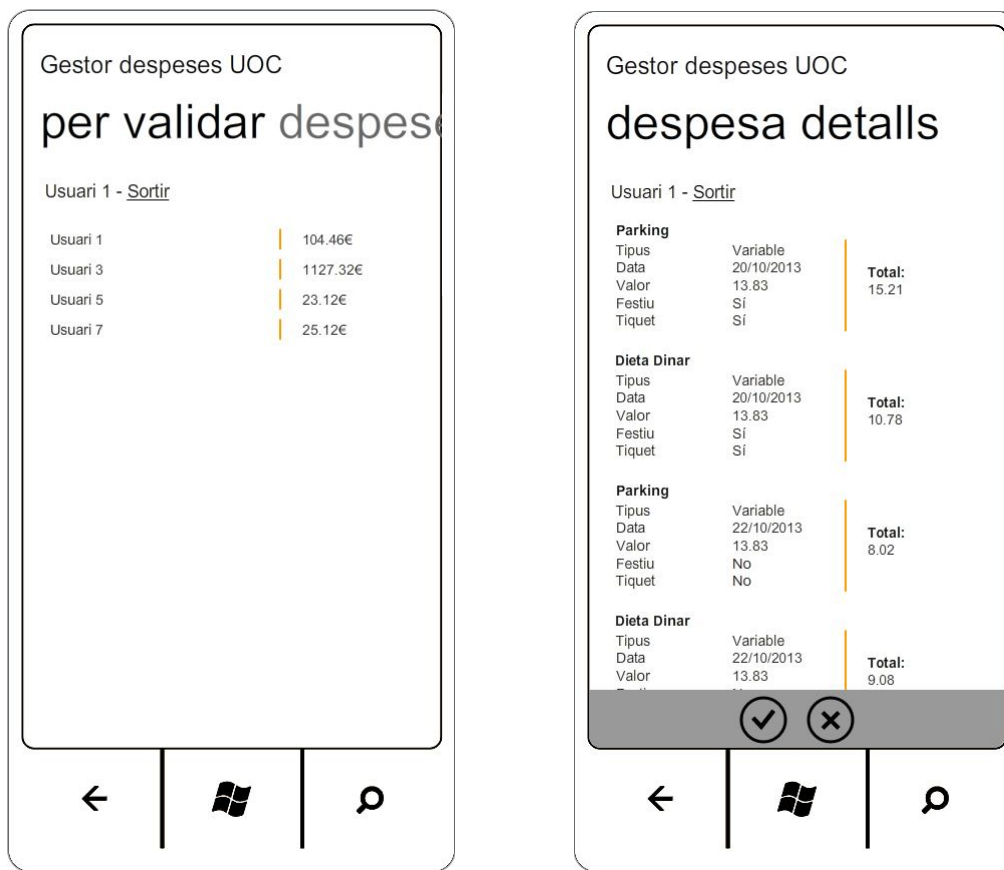


L'única acció que es pot realitzar sobre una despesa és veure-hi els detalls de la mateixa. Al veure els detalls apareix un llistat amb totes les despeses que hi han donades d'alta al full de despeses corresponent.

## Validació de despeses

Aquesta pantalla forma part de la vista panorama. Només l'usuari Responsable podrà accedir a la pantalla de Despeses per validar. En aquesta pantalla es mostra un llistat de

les despeses dels usuaris que estan a càrrec de l'usuari responsable i que tenen el full de despeses pendent de validar.



Només existeix una acció que es pot realitzar sobre el llistat de despeses per validar i és el de veure els detalls del full de despeses. Dins de la pantalla de detalls de la despesa hi ha les accions de *Validar* (✓) i *Rebutjar* (✗). En aquest últim cas apareixerà una finestra de confirmació on l'usuari ha d'introduir el motiu pel qual el full de despeses ha estat rebutjat.

## 4 Desenvolupament

### 4.1 Software utilitzat

Al projecte s'han utilitzat unes quantes tecnologies, a continuació es fa un llistat d'aquestes tecnologies agrupades segons per a que s'han fet servir.

A nivell general s'han fet servir les següents tecnologies:

- **C#** com a llenguatge de programació.
- **EntityFramework 5.0** com a ORM per modelar la base de dades i autogenerar les classes i tot el necessari per poder accedir a la mateixa.
- **Linq**, per a fer consultes sobre els objectes de la base de dades entre altres.
- **Internet Information Services 7.0**, com a servidor per als Serveis Web i per a l'aplicació web.
- **SQL Server 2008 Express** com a servidor de base de dades.
- **Visual Studio 2010 Express for Windows Phone** com a entorn de desenvolupament per a l'aplicació mòbil.
- **Visual Studio 2012 Professional** com a entorn de desenvolupament per a la resta de components.

Als serveis web s'ha fet servir **Windows Communication Foundation (WCF)**

A l'aplicació web s'han fet servir:

- **ASP.NET Web Forms**, com a model de programació.
- **AJAX** per a fer crides al servidor de l'aplicació web des de client.
- **JQuery** per simplificar l'ús de Javascript i la integració amb AJAX
- **JQuery UI** per afegir components i temes
- HTML, Javascript i CSS

A l'aplicació per a Windows Phone s'ha fet servir:

- **Windows Phone SDK 7.1**
- **Silverlight** per a Windows Phone (versió de Octubre 2011) per els inputs de tipus autocompletar i els inputs de tipus data.
- **Coding4fun**, per el control de *Loading* que apareix quan s'estan carregant dades del servei web, el *About* i pel dialog que conté un input de text per poder indicar un motiu en rebutjar una despesa.

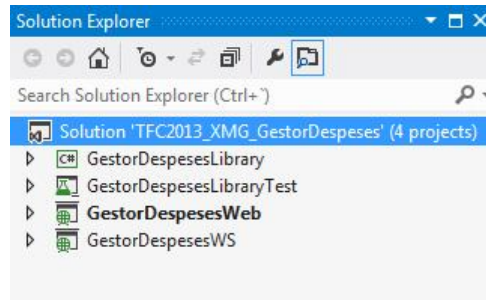
### 4.2 Aspectes tècnics

Per motius de limitació de hardware hi ha dues solucions de Visual Studio amb versions diferents, a continuació es descriu el contingut de cadascuna.

#### 4.2.1 Solució Visual Studio 2012

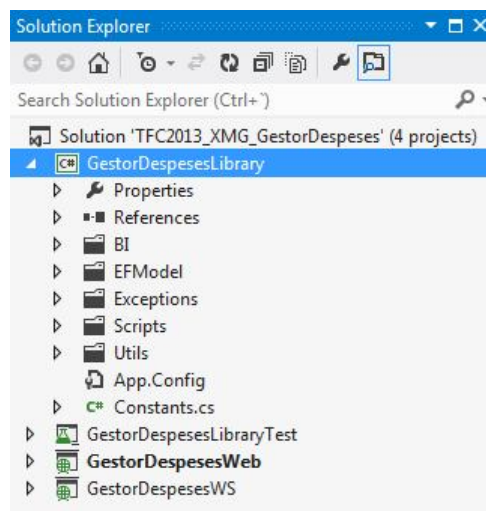
Aquesta solució inclou tot exceptuant l'aplicació per a Windows Phone que s'ha hagut de crear al *Visual Studio 2010 Express for Windows Phone*.

En aquesta solució hi ha inclosos el següents projectes:



## GestioDespesesLibrary

És un projecte que genera com a resultat una llibreria que conté tota la capa d'accés a dades i la capa de negoci de l'aplicació.



**BI** inclou una sèrie de classes i mètodes per poder dur a terme totes les tasques de negoci que requereix l'aplicació. Les classes són les següents:

- **CoreBI:** Conté mètodes per poder enviar correus electrònics i llegir i modificar la configuració del sistema.
- **ExpensesBI:** Conté mètodes per a gestionar despeses i fulls de despeses, com crear una despesa, modificar l'estat d'un full de despeses...
- **ExpensesTypesBI:** Conté els mètodes per a gestionar tipus de despeses, com crear un tipus de despesa, modificar un tipus de despesa...
- **UsersBI:** Conté mètodes per gestionar els usuaris, com crear usuari, esborrar usuari, modificar usuari...
- **WSSessionsBI:** Conté mètodes per a crear i controlar les sessions del usuari.

**EFModel** inclou tota la capa d'accés a dades. Per l'accés a les dades s'ha utilitzat Entity Framework com a ORM i s'ha utilitzat un mode de treball anomenat *Model First*, que consisteix en crear el model i després generar les classes necessàries a partir del model. Aquesta carpeta inclou el model i les classes generades.

**Exceptions** inclou dos tipus d'excepció que s'han creat que seran les úniques que la capa de negoci llençarà. Una és per errors controlats (s'intenta fer una acció que no es permet,

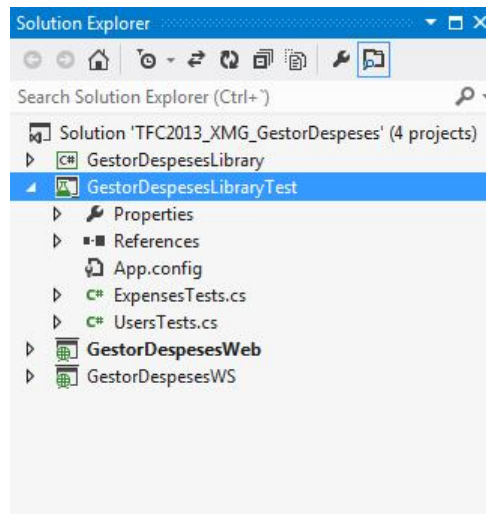
alguna limitació del model, ...) i l'altre és per quan hi ha un comportament no esperat a la capa de negoci.

**Scripts** inclou dos scripts sql, un per, una vegada generada la base de dades a partir de l'script que genera el Entity Framework, completar-la amb la creació de claus úniques per a certes columnes i l'altre per fer una pre-càrrega de dades a la base de dades.

**Utils** inclou una classe amb utilitats generals per la resta parts.

## GestioDespesesLibraryTest

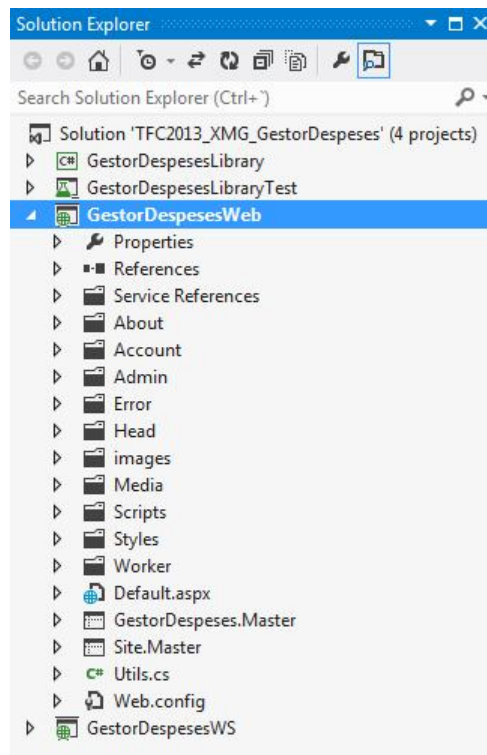
Aquest projecte conté probes unitàries del projecte *GestorDespesesLibrary*. És un projecte que està per terminar i només inclou els tests de la part corresponent a Usuaris.



## GestioDespesesWeb

Aquest projecte genera com a resultat una aplicació web que serveix per que els usuaris interactuïn amb el sistema.

Per a cada plana de l'aplicació existeix un fitxer .aspx (que es la part visual) i un fitxer .cs (que seria la part de controlador). Aquesta part de controlador gairebé la única tasca que fa és enviar peticions als serveis web per a realitzar les accions pertinents i processar la informació rebuda per mostrar-la a les vistes.



Hi ha 6 carpetes que corresponen al mapa de la web:

- **About:** Informació del producte.
- **Account:** Login, LogOff i recuperació de credencials.
- **Admin:** Gestió d'usuaris, pagament de fulls validats i configuració del sistema.
- **Error:** Pàgines d'error de l'aplicació.
- **Head:** Gestió dels fulls de despeses per validar.
- **Worker:** Gestió del full de despeses personal i històric de despeses.

Hi ha dues pàgines "Master":

- **Site.Master:** Pàgina màster global per totes les pàgines.
- **GestorDespeses.Master:** Pàgina màster per a aquelles pàgines on l'usuari ja hagi estat autenticat. Aquesta pàgina màster té com a pàgina màster *Site.Master*.

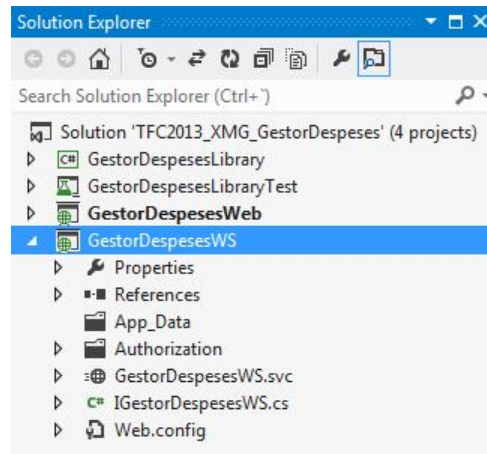
*Default.aspx* és la pàgina que és carrega per defecte al entrar en la web i té com a única funcionalitat redirigir al usuari a una pàgina o una altre dependent si esta autenticat o no i dependent del rol d'usuari que tingui.

*Service References* és una referència al servei web.

## GestioDespesesWS

Aquest projecte genera un servei web per exposar remotament tota la funcionalitat de la plataforma i que l'aplicació web i l'aplicació per a Windows Phone puguin realitzar les tasques corresponents.





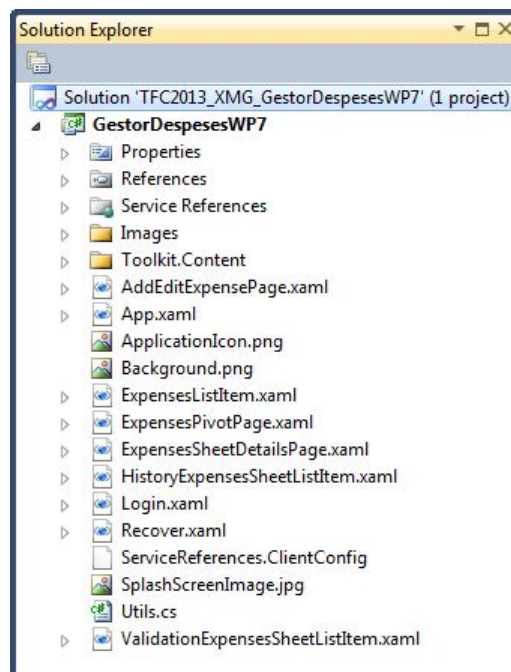
Aquest projecte té una dependència amb el projecte *GestorDespesesLibrary*.

Conté un mòdul d'autorització (que s'explica al apartat *Seguretat*), una classe de tipus interface on es troba tota la definició del servei web i una classe que implementa aquesta interface.

El servei només conté un enllaç (binding) de tipus *basicHttpBinding*, és un tipus d'enllaç sense autenticació ni xifrat de dades. En el apartat de *Seguretat* s'expliquen les mesures adoptades per dotar de seguretat a aquesta part del sistema.

#### 4.2.2 Solució Visual Studio 2010

Aquesta solució inclou un únic projecte que correspon a l'aplicació per a dispositius Windows Phone.



Per a cada pantalla de l'aplicació existeix un fitxer .xaml (que es la part visual) i un fitxer .cs (que seria la part de controlador). Aquesta part de controlador fa crides al servei web depenent de les tasques que hagi de fer i modifica la informació que es mostra a la part visual.

Aquest projecte té dues dependències externes que son:

- **Microsoft.Phone.Controls.Toolkit:** Llibreria de Silverlight per a Windows Phone (versió de Octubre 2011). En el projecte s'utilitza per els camps de tipus autocompletar i per els camps de tipus data.
- **Coding4Fun.Toolkit.Controls:** Llibreria que s'utilitza per el control de Loading que apareix quan s'estan carregant dades del servei web, el about i per el dialog que conté un textbox per poder rebutjar una despesa.

Tots dos toolkits necessiten que al projecte s'inclogui la carpeta *Toolkit.Content* que conté dues imatges per els botons.

A imatges hi ha les imatges dels botons de la barra d'aplicacions que apareix en l'aplicació.

*Service References* és una referència al servei web.

### 4.2.3 Seguretat

Un altre punt és la seguretat del sistema implementat i quines són les mesures de seguretat que s'han dut a terme per augmentar-la el màxim possible, tal com podem veure als següents apartats.

## Autenticació

Només existeix un endpoint on es poden connectar els clients del servei web, i té un binding de tipus *basicHttpBinding*. Aquest tipus de binding no incorpora cap tipus de seguretat, ni a nivell d'autenticació ni a nivell de xifratge de dades, per tant qualsevol amb connectivitat al servei podria fer crides als mètodes que el servei exposa sense cap tipus de restricció.

El servei web exposa diferents mètodes, hi ha uns quants que no requereixen autenticació (login i tots els referents a la part de recuperació de credencials) i per la resta si que es requereix.

Hi ha dues solucions per aquest problema:

1. Desenvolupar un servei web amb dos contractes i associar cadascun a un endpoint diferent. Un endpoint tindria un binding de tipus *basicHttpBinding* per les operacions de recuperació de credencials (login no faria falta perquè ja ho incorporaria el binding de l'altre endpoint). L'altre endpoint s'hauria d'escollir un dels tipus de seguretat que proporciona WCF per serveis web, per exemple *wsHttpBinding* que ja incorporaria la part d'autenticació, confiabilitat i seguretat de la missatgeria. Aquesta solució implica:
  - Generar un certificat i fer que els clients confiïn en aquest.
  - Desenvolupar un mòdul d'autenticació custom que faci l'autenticació dels usuaris.
  - Configurar dos endpoints i separar el desenvolupament en dos contractes.

- Escollir un tipus de context per el servei que suporti sessions i peticions concurrents.
2. Desenvolupar el nostre propi sistema d'autenticació i control de sessions. En el propi endpoint que ja disposem s'ha de generar un mètode per login i un altre per logout. El mètode de login accepta usuari i contrasenya i retorna com a resultat satisfactori un token de sessió que s'ha d'incorporar com a paràmetre en tots aquells mètodes que requereixen autenticació. El token té un temps de vida determinat que es reinicia cada cop que es rep una petició amb aquest token. L'únic que s'ha de tenir en conte és que la comunicació ha d'anar xifrada d'alguna manera, això ho aconseguim modificant el binding del endpoint, com per exemple modificant la propietat *BasicHttpSecurityMode* del nostre binding a *Transport*, que ens ofereix confidencialitat e integritat de la missatgeria. Aquesta solució implica:
- Generar un sistema de control de sessions.
  - Afegir al contracte els mètodes de login i logout.
  - Modificar la resta de mètodes del contracte per que acceptin un paràmetre nou que serà el token de sessió.
  - Modificar el mode de seguretat del binding.
  - Generar un certificat i fer que els clients confiïn en aquest.

Per raons de temps s'ha escollit la segona opció, a falta de modificar el binding per que la missatgeria vagi xifrada. En el seu moment vaig determinar que el cost d'utilitzar una o una altre és semblant però que en la segona, en cas no arribar a temps per l'entrega em costaria menys desfer que no pas la primera.

Per un projecte no acadèmic segurament hagués optat per la primera opció i així utilitzar les eines que ofereix WCF per aquests temes.

## Autorització

S'ha implementat un mòdul d'autorització al servei web per tal que, quan s'intenti cridar a una operació del contracte i a partir del token i dels paràmetres de la pròpia crida, es pugui determinar si es té autorització per realitzar la operació o no.

En concret hi ha 4 mètodes per autoritzar.

- **isAllowed:** amb el token i el tipus d'usuari que es requereix per fer la operació, determina si l'usuari del token es del tipus d'usuari específic.
- **isAllowed:** amb el token i un identificador d'usuari, determina si l'usuari del token té accés a les dades del usuari corresponent al identificador.
- **isAllowedExpenseSheet:** amb el token i un identificador de full de despeses, determina si l'usuari del token té accés a les dades del full de despeses corresponent al identificador.
- **isAllowedExpense:** amb el token i un identificador de despesa, determina si l'usuari te accés a les dades de la despesa corresponent al identificador.

## 5 Avaluació de costos

Per calcular el cost del projecte es tindrà en compte les hores invertides. Es faran dos càlculs, un corresponent al cost que tindria el projecte si s'hagués ajustat exactament a la planificació inicial i un altre amb les hores que s'han invertit realment, per poder veure la desviació que ha hagut.

### 5.1 Estimació

La planificació es va fer en base a dedicar 2h diàries al projecte, per tant:

Tasca	Dies
Planificació del projecte	7
Anàlisi i disseny	28
Desenvolupament	49
Lliurament del projecte	28
	<b>112</b>

Traient els dies de "Temps no assignat" que es reservaven per a emergències i descans ens queden 101 dies. Si la dedicació que es pretenia dedicar eren 2 hores/dia, tenim 202 hores de treball que es van estimar, per tant, calculant amb una mitja de 30 €/hora:

$202 \times 30 = 6060 \text{ €}$
----------------------------------

### 5.2 Real

En la fase de desenvolupament es van esgotar els dies de temps no assignat, i a més, es van dedicar més hores de les que es van calcular, unes 20 hores.

Al final he dedicat 23 hores més en finalitzar el projecte, per tant:

$225 \times 30 = 6750 \text{ €}$
----------------------------------

Hi ha hagut una desviació de 690 euros en el projecte.

## 6 Treball futur

Hi ha moltes coses que el producte podria incloure, noves funcionalitat o millorar les existents, canvis de disseny,...

Tasques que degut a la limitació de temps no s'han pogut incloure. S'ha de tenir en compte que s'han desenvolupat d'altres que no hi eren a la planificació però que s'han considerat més prioritàries.

- Modificar el mode de seguretat del binding a *Transport* per garantir la confidencialitat de les peticions.
- Afegir un nou endpoint amb un binding de tipus *netNamedPipeBinding* amb el mateix contracte al servei web, per tal que la web que s'allotja a la mateixa màquina es connecti al servei web mitjançant aquest canal. Per serveis que es troben en una mateixa màquina aquest tipus de binding és el més eficient.
- Que l'usuari pugui modificar les seves credencials quan està autenticat. Ara mateix la contrasenya només es pot canviar utilitzant la funcionalitat de recuperar credencials o demanant a l'administrador que la canviï.
- Afegir comprovacions bàsiques en els inputs dels formularis. Ara mateix no es comprova en client coses com que un camp no estigui buit, que un camp numèric sigui un número o que una direcció de correu tingui el format correcte, tot això es comprova a la part servidor però es poden estalviar crides al servidor inútils.
- Finalitzar el projecte de *GestorDespesesLibraryTest* afegint-hi els tests unitaris que cobreixin tota la funcionalitat de la llibreria.

Tasques que no hi eren a la planificació però que a l'hora de programar i provar el producte serien de gran utilitat.

- Que l'usuari pugui eliminar la imatge d'una despesa. Ara mateix es pot modificar la imatge d'una despesa però no es pot eliminar. Per fer-ho s'ha d'esborrar la despesa i crear-la de nou sense afegir-hi la imatge.
- Que un usuari de tipus Responsable (en validacions) i un usuari de tipus Administrador (en pagaments) puguin veure la imatge d'una despesa.
- Al visualitzar les despeses d'un full de despeses veure una columna que indiqui si la data correspon a un dia del cap de setmana o no.
- Al generar una despesa i seleccionar una data que es mostri si és festiu o cap de setmana i que es pugui veure l'increment que s'aplicarà en aquest cas.
- Afegir un sistema de notificacions a l'aplicació de Windows Phone per rebre alertes de quan un full de despeses ha estat validat, rebutjat o pagat. També per aquells usuaris de tipus Responsable notificar-los que un usuari al seu càrrec ha posat un full de despeses a validar.
- Fer configurables per l'administrador l'increment que s'aplica a una despesa si es fa en cap de setmana o en un dia festiu. Ja existeixen els valors a la taula de propietats del sistema però no son configurables.
- Fer configurables per l'administrador el missatge que s'envia per correu per enviar el codi de validació. Ja existeix el valor a la taula de propietats del sistema però no és configurable.
- Enviar alertes per correu als usuaris responsables quan tinguin fulls de despeses per validar i als administradors quan tinguin fulls de despeses per pagar, i que sigui una opció configurable per l'usuari.
- Fer el producte multiidioma.

- Fer que el mòdul d'autorització sigui més complert, afegint més condicions com per exemple si s'accedeix per lectura o per lectura i escriptura.
- Configurar el plug-in TableTools per poder exportar el contingut d'una taula a diferents formats o enviar a imprimir. Els components necessaris es troben dins de la solució al projecte *GestorDespesesWeb*, però les proves que s'han fet són funcionalment correctes però provoquen que es desquadrin les taules.

## 7 Canvis inclosos en aquesta entrega

Respecte a l'entrega corresponent a la PAC3 s'han fet modificacions, la majoria de les quals corresponen a bugs que s'han trobat a l'aplicació.

### 7.1 Errors corregits

Afegir un tipus de despesa amb valors opcionals vuits no funcionava correctament. No es contemplava al codi de la vista aquesta opció.

Modificar un tipus de despesa retornava un error. Hi havia dos errors en aquest bug, un que no es passava correctament l'identificador del tipus de despesa a modificar i l'altre, que el mètode que persisteix els canvis a la base de dades sempre retornava com a resposta que no s'havia pogut fer el canvi a les dades.

Al llistat d'usuaris no es refrescava la informació referent al tipus d'usuari, per exemple, si hi havia un usuari treballador i s'afegia un nou i es posava a càrrec del primer el tipus d'aquest no canviava a "Responsable" fins que no es recarregava la plana sencera.

Editar una despesa no funcionava correctament. A última hora es fa afegir una comprovació per veure si el fitxer que es puja al formulari correspon a una imatge vàlida o no, però no es va tenir en compte el cas de modificar la despesa que no es reenvia la imatge.

Afegir una despesa després d'haver modificat una altre editava aquesta última en lloc d'afegir una de nova. No es netejava l'identificador a l'afegir i el sistema interpretava que s'estava editant.

Afegir o editar un usuari no es feien les comprovacions bàsiques dels camps a la part servidor, podent afegir un usuari amb el username o el nom en blanc.

Esborra un usuari amb usuaris al càrrec no es podia. Ara els usuaris que tenen responsable assignat queden sense responsable assignat quant s'esborra aquest.

### 7.2 Noves característiques

S'ha afegit la opció de esborrar el tiquet d'una despesa sense la necessitat d'esborrar la despesa i afegir-la de nou

Als dialogs que contenen formularis (afegir o editar: despeses, tipus de despeses, usuaris i revocar un full de despeses a validació de despeses) s'ha modificat el comportament dels mateixos en produir-se un error. Abans, anés el procés bé o malament es tancava el dialog, ara es tanca en cas afirmatiu i es queda obert en cas d'error per que l'usuari no hagi de tornar a omplir el formulari complert si ha de fer una petita modificació.

S'ha afegit una limitació de longitud mínima que ha de tenir la contrasenya.

## 8 Canvis respecte l'anàlisi i disseny de l'aplicació

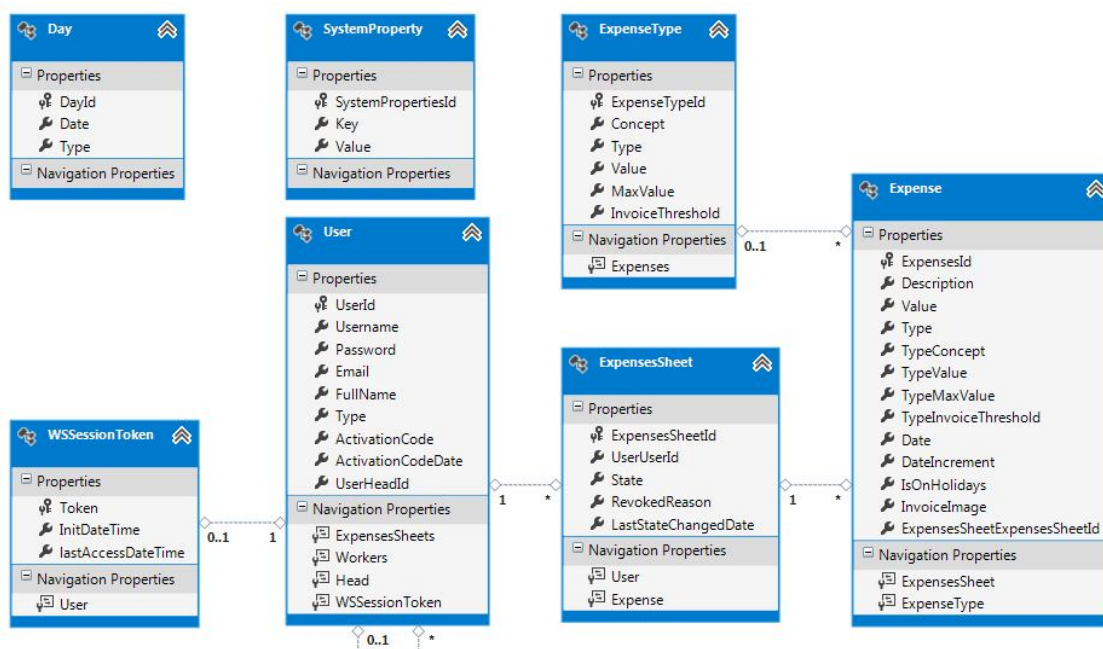
S'han realitzat alguns canvis sobre l'anàlisi i disseny que es va plantejar inicialment.

Visualment no s'han fet variacions destacables, s'ha intentat que el disseny final s'ajustés el màxim possible al disseny proposat.

En quan al funcionament, s'ha eliminat la opció a l'aplicació web de poder assignar un responsable directament, es fa mitjançant el propi formulari d'afegir/modificar un usuari. També s'ha modificat el formulari per afegir/modificar una despesa, fent que sigui dinàmic, dependent del tipus de despesa seleccionat es mostren uns camps o uns altres.

A l'aplicació mòbil s'ha descartat la vista panorama i s'ha escollit una vista pivot, que en el fons s'ajusta molt més al que en un principi es volia aconseguir.

On més canvis hi ha hagut és a la base de dades, com es pot veure en el disseny ER final d'aquesta:



A la taula *Expense* s'han afegit uns quants camps més que són necessaris per mantenir els valors de les despeses que ja estan afegides al sistema encara que es facin modificacions al seu tipus, també s'ha afegit una relació entre aquesta taula i la taula de *ExpenseType* per poder realitzar comprovacions a les despeses ja afegides de si hi han hagut modificacions al tipus de despesa del qual hereta abans que la despesa hagi estat validada.

També s'ha afegit una taula nova, *WSSessionToken*, per poder controlar les sessions dels usuaris. Aquesta funcionalitat no es va tenir en compte al disseny inicial i s'ha implementat per afegir seguretat al sistema. Quan es crida al mètode login dels serveis web amb unes credencials vàlides es crea una entrada en aquesta taula amb un token, amb un temps de vida limitat, associat al usuari corresponent.

Als serveis web, a part de l'autenticació i el control de sessions, s'ha afegit un mòdul d'autorització per controlar, a partir del token rebut i el mètode que s'està cridant, determinar si l'usuari té permís o no.



## 9 Conclusions

L'objectiu del treball final de carrera és el d'elaborar i presentar un treball integrador de totes les competències, tant tècniques com transversals, adquirides al llarg dels estudis de la titulació, per tal d'apropar als alumnes al món laboral.

La metodologia d'estudi que ofereix la Universitat Oberta de Catalunya permet que, gent que es troba incorporada actualment al món laboral (com és el meu cas), pugui compaginar la feina amb els estudis, de manera que l'objectiu d'apropar als alumnes al món laboral es converteix en el meu cas en un objectiu més aviat recordatori, que pot permetre a l'estudiant/treballador corregir mals hàbits adquirits al dia a dia al seu lloc de treball.

Tot i que la planificació inicial no s'ha ajustat al 100%, ha estat bastant encertada ja que en tota planificació hi ha un marge d'error ja que no es poden tenir controlades totes les variables que intervenen, que és el que m'ha passat a mi en el desenvolupament de l'aplicació Web, ja que he tingut bastants problemes amb aquesta part.

Per una altre banda, part de les tecnologies eren desconegudes per a mi, per tant ha estat una gran oportunitat el poder trastejar amb EntityFramework, WebForms i tot el referent a Windows Phone. M'he quedat molt satisfet de EntityFramework, és molt senzill d'utilitzar i agilitza molt la programació de la capa d'accés a dades. El mateix m'ha succeït amb l'aplicació per a Windows Phone, que m'he quedat sorprès de lo fàcil que és fer una aplicació per a aquests dispositius. En canvi, Web Forms a estat una font continua de problemes, ja tenia una mica d'experiència amb MVC i volia provar aquesta altre, però no he sabut encaixar totes les tecnologies que vaig escollir i he acabat per donar-me conte que l'elecció de MVC encaixava millor. Tot i això, no es una dada negativa, ja que, davant d'una altre situació en que tingui que triar entre una o l'altre, ara tinc criteris per fer-ho correctament.

Finalment crec que s'ha aconseguit un producte que compleix amb totes les necessitats descrites a l'enunciat i fàcil d'utilitzar per els usuaris.

Sobre l'assignatura en sí, crec que són molt importants treballs d'aquest tipus en una carrera tècnica com és aquesta, ja que, com ja he comentat, permet posar en pràctica molts dels coneixements adquirits al llarg de la carrera i apropar a l'alumne al món laboral, demostrant-li que tot els continguts que ha adquirit al llargs dels semestres, i que en molts casos no se sap ben bé quin sentit tenen, tenen una aplicació real.

## 10 Bibliografia

- “Cómo redactar fácilmente la bibliografía de tus trabajos”. <http://bibliotecabiologia.usal.es/tutoriales/Elaboracion-citas-bibliograficas.pdf> [Consulta: 30-09-2013]
- Foradada Angrill, Oriol. “Sistema de monitorització de xares”. 12-03-2012.
- Solà Aranda, Guillem. “Aplicació gestió de lloguer de vehicles”. 03-10.2011.
- Beneito Montagut, Roser. “Presentació de documents i elaboració de presentacions”. [http://materials.cv.uoc.edu/cdocent/HKZB5UG6XS130\\_6R5O43.pdf?ajax=true](http://materials.cv.uoc.edu/cdocent/HKZB5UG6XS130_6R5O43.pdf?ajax=true)
- Sáenz Higuera, Nita i Vidal Oltra, Rut. “Redacció de textos científicotècnics”
- “Requisito funcional”. [http://es.wikipedia.org/wiki/Requisito\\_funcional](http://es.wikipedia.org/wiki/Requisito_funcional) [Consulta: 28-09-2013]
- “Requisito no funcional”. [http://es.wikipedia.org/wiki/Requisito\\_no\\_funcional](http://es.wikipedia.org/wiki/Requisito_no_funcional) [Consulta: 28-09-2013]
- “Get Started with Entity Framework (EF)”. <http://msdn.microsoft.com/en-us/data/ee712907> [Consulta: 06-10-2013].
- “Getting started with developing for Windows Phone”. <http://msdn.microsoft.com/library/windowsphone/develop/ff402529%28v=vs.105%29.aspx> [Consulta: 06-10-2013]
- “Developing a Windows Phone Application from Start to Finish”. <http://msdn.microsoft.com/en-us/library/gg680270%28v=pandp.11%29.aspx> [Consulta: 07-10-2013]
- Armando Canchala. “UML, ejemplo sencillo sobre Modelado de un Proyecto”. <http://msdn.microsoft.com/es-es/library/bb972214.aspx> [Consulta: 13-10-2013].
- “Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios: Apéndices”. <http://msdn.microsoft.com/es-es/library/ms978371.aspx> [Consulta: 16-10-2013].
- “Introduction to ASP.NET and Web Forms”. <http://msdn.microsoft.com/en-us/library/ms973868.aspx>
- “ASP.NET Web Forms – Tutorial”. [http://www.w3schools.com/aspnet/aspnet\\_intro.asp](http://www.w3schools.com/aspnet/aspnet_intro.asp)
- “jQuery API Documentation”. <http://api.jquery.com/>
- “jQuery UI Support”. <http://jqueryui.com/support/>

- “Chapter 4: WCF Security Fundamentals”. <http://msdn.microsoft.com/en-us/library/ff650862.aspx> [Consulta: 04-11-2013]
- “DataTables Usage”. <http://www.datatables.net/usage/>
- “AutoCompleteBox for WP7 in depth”  
<http://www.geekchamp.com/articles/autocompletebox-for-wp7-in-depth>
- “WP7 DatePicker and TimePicker in depth | API and Customization”.  
<http://www.geekchamp.com/articles/wp7-datepicker-and-timepicker-in-depth--api-and-customization>
- “Coding4Fun Toolkit Documentation” <http://coding4fun.codeplex.com/documentation>