

TREBALL FI DE CARRERA

Gestió de despeses personals

Mònica González Montilla

Consultor:
Jairo Sarrias Guzman

[MEMÒRIA]

Gener 2014

Informació General

Títol	Treball Fi de Carrera
Tecnologia	Microsoft .NET
Projecte	Gestió de les despeses personals d'una empresa
Consultor	Jairo Sarrias Guzman
Alumne	Mònica González Montilla
Fase	Memòria

ÍNDEX

1. Introducció.....	4
2. Descripció del projecte.....	4
JUSTIFICACIÓ DEL PROJECTE (IDONEÏTAT).....	4
DESCRIPCIÓ DEL PROJECTE.....	5
OBJECTIUS DEL PROJECTE.....	5
REQUERIMENTS DE LA SOLUCIÓ.....	6
FUNCIONALITATS A IMPLEMENTAR.....	7
RESULTATS ESPERATS.....	7
PRODUCTES OBTINGUTS.....	7
PLANIFICACIÓ INICIAL VS PLANIFICACIÓ FINAL.....	8
ANÀLISI DE RISCOS.....	9
3. Anàlisi i Disseny.....	9
REQUERIMENTS FUNCIONALS.....	9
REQUERIMENTS NO FUNCIONALS.....	11
DIAGRAMA DE CASOS D'ÚS.....	11
MODEL CONCEPTUAL.....	25
DIAGRAMA D'ARQUITECTURA.....	27
DISSENY DE LA BBDD.....	29
DIAGRAMA DE CLASSES DE LA INTERFÍCIE GRÀFICA.....	33
DIAGRAMA D'ESTAT I DE DESPLEGAMENT.....	35
DISSENY DE LA INTERFÍCIE D'USUARI.....	36
4. Desenvolupament.....	38
SOFTWARE UTILITZAT.....	38
CAPES DE L'APLICACIÓ.....	39
CAPA DE DADES - ENTITY FRAMEWORK.....	39
CAPA LÒGICA DE NEGOCI - WCF.....	41
CAPA DE PRESENTACIÓ - ASP.NET WEB FORMS.....	47
5. Avaluació de costos.....	57
6. Treball futur.....	58
FÓRA DE L'ABAST.....	58
7. Conclusions.....	59
8. Bibliografia.....	59

1. Introducció

El present treball final de carrera es basa en la "Gestió de despeses personals" dels treballadors d'una empresa qualsevol.

Aquest document és la memòria del projecte final de carrera de l'Enginyeria tècnica d'informàtica de Gestió. Resumeix tot el treball realitzat durant el projecte i inclou tota la informació dels lliuraments realitzats.

El Treball Fi de Carrera pertany a l'àrea de Microsoft .NET i presenta la resolució a un problema del món real. En la solució realitzada es posen en pràctica els coneixements adquirits a d'altres assignatures de la carrera, adquirint també competències sobre la plataforma .NET de Microsoft (aprofundint en les tecnologies més importants com ASP.NET, Entity Framework, Web Services, WCF, WFP, LINQ to Entities...)

Per tant el TFC es centra no només en la posada en pràctica dels coneixements adquirits en les assignatures de l'ETIG sinó en la recerca, estudi i posada en pràctica de les tecnologies indicades.

L'abast del TFC comprèn:

- Pla de treball
- Anàlisi i disseny
- Implementació
- Memòria
- Presentació virtual
- Debat defensa del TFC realitzat

Les funcionalitats i planificació del TFC es descriuen en els apartats següents.

2. Descripció del projecte

JUSTIFICACIÓ DEL PROJECTE (IDONEÏTAT)

Perquè el projecte

Per gestionar d'una manera més eficient i poder portar el control de les despeses que realitzen els treballadors en una empresa, es decideix implementar un software amb totes les funcionalitats necessàries per donar solució al problema. Aquest software serà una Web que ens permetrà fer les gestions sense estar ubicat al centre de treball. El software serà a mida, per cobrir exactament les necessitats específiques de la empresa

Per un altre costat i des d'una vessant més personal la principal raó per la elecció d'un projecte .NET és la d'aprendre i aprofundir en la tecnologia .NET.

Amb l'atractiu afegit que aquest aprenentatge és eminentment pràctic.

DESCRIPCIÓ DEL PROJECTE

El punt de partida del TFC és la realització d'un projecte que gestioni les despeses personals de tots els treballadors d'una empresa.

El projecte es compondrà de:

INTERFÍCIE WEB

Es podrà accedir a través d'Internet i els:

- Treballadors podran gestionar les seves despeses que un cop autoritzades s'incorporaran a les seves nòmines mensuals.
- Autoritzadors podran gestionar i aprovar les despeses dels treballadors.
- Administradors veuran el panell de configuració des de el que podran gestionar els llistats de treballadors, els diccionaris de l'aplicació i el calendari de dies festius.

CAPA LÒGICA DE NEGOCI

Gestió de tota la lògica de negoci

CAPA DE DADES

Gestió de les dades de l'aplicació

Tots aquests components es detallen mes endavant.

OBJECTIUS DEL PROJECTE

GENERALS

L'objectiu principal del TFC és posar en pràctica els coneixements adquirits a d'altres assignatures de la carrera així com aprofundir en la tecnologia .NET de Microsoft.

Referent al projecte en concret, els objectius generals són:

- Realització d'una aplicació que gestioni les despeses dels treballadors d'una empresa fent servir tecnologia .NET
- Elaborar tots els documents previs abans de la seva implementació (així com una memòria i presentació final de projecte)

ESPECÍFICS

Les fases específiques del projecte són les següents:

- **Pla de Projecte**
Redactar el pla de projecte on s'especifica, a grans trets, la descripció del projecte, objectius, requeriments i funcionalitats de la solució així com l'organització del projecte, proposta d'activitats, cronograma, rols i valoració econòmica.
- **Anàlisis i Disseny**
En aquesta fase es realitza l'especificació, anàlisis dels requeriments funcionals i no funcionals. I en referència al disseny es realitza la especificació dels casos d'us, descripció dels subsistemes, diagrama de classes, model E-R, etc...
- **Implementació**
Implementació de la solució, analitzada i dissenyada en la fase anterior, fent servir tecnologia .NET i realització dels manual d'usuari, configuració i implementació.

TREBALL FI DE CARRERA

- **Memòria**
En la memòria final del projecte es recollirà un resum de tot el treball.
- **Presentació virtual**
En aquesta presentació es reflecteix els punts més importants del treball realitzat perquè posteriorment en un debat virtual es realitzi la defensa del mateix.

REQUERIMENTS DE LA SOLUCIÓ

FUNCIONALS

Els requeriments inicials, fets a la fase de planificació són:

Aplicació Web per a la gestió completa de les despeses del personal d'una empresa. Els punts més importants són:

- **Validació al sistema**
Aquest mòdul gestiona els accessos als sistema (login i canvi de clau).
- **Configuració del sistema**
Des d'aquest mòdul es pot realitzar el manteniment del llistat de treballadors de l'empresa (alta, edició i baixa), dels diccionaris (tipus de treballador, estat fulles i tipus de despesa).
- **Gestió despeses**
Els treballadors de l'empresa que vulguin afegir una despesa a la seva nòmina poden omplir un full de despesa a on es pot afegir tantes despeses com calguin. Les despeses, un cop aprovades s'associaran a la nòmina del mes en curs, no forma part de l'abast d'aquest projecte la preparació de nòmines.
- El treballador pot realitzar cerques sobre les seves fulles de despeses. Es permet l'edició o eliminació tant de les fulles com de les despeses sempre hi quan la fulla no hagi estat tancada.
- **Autoritzacions**
Els usuaris que disposin permisos per autoritzar fulles han de poder autoritzar, refusar o tancar fulles de despeses, així com realitzar cerques segons varis criteris.
- **Gestió Calendari**
Gestió del calendari de dies festius, per tal de distingir dietes i pagar-les amb l'import corresponent.

NO FUNCIONALS

Aconseguir que la web tingui un aspecte visual agradable i sigui fàcilment usable, perquè els usuaris es sentin còmodes evitant qualsevol rebuig inicial que certs usuaris no experts puguin sentir en el primer contacte amb la mateixa.

Per aquest motiu és de vital importància tenir en compte els següents punts:

- El sistema ha de ser atractiu i intuïtiu amagant els aspectes de major complexitat, per què l'usuari tingui la certesa que té el control sobre el sistema.
- La corba d'aprenentatge dels usuaris ha de ser el més curta possible i per tant haurien d'adquirir un complet domini del sistema en un termini curt de temps.

Per un altre costat el sistema és flexible, per permetre la incorporació de nous mòduls amb posterioritat i de forma senzilla.

FUNCIONALITATS A IMPLEMENTAR

Es defineixen les funcionalitats a implementar segons els requeriments proposats al punt anterior:

Validació al sistema.

- Gestió d'accés al sistema, en funció del tipus d'usuari es veuran unes opcions o altres.

Gestió Canvi de clau

Gestió de configuració

- Gestió dels Treballadors de l'empresa.
- Gestió dels diccionaris del sistema.

Gestió Despeses

- Obrir fulles de despeses a on es poden afegir, editar o eliminar línies amb cada una de les despeses a cobrar.
- Sol·licitar autorització sobre la fulla que en cas d'acceptació s'afegeix a la nòmina del mes en curs.

Gestió Autoritzacions fulles

- Cercar, visualitzar i canviar l'estat d'una fulla (autoritzar, refusar o tancar)

Gestió Calendari

- Gestió del calendari de dates festives.

Aquestes funcionalitats es desenvolupen fent servir tres capes (presentació, lògica i de dades) que es detallen mes endavant.

RESULTATS ESPERATS

Al final del procés de desenvolupament es pretén oferir una aplicació Web des de la que es puguin gestionar les despeses del personal d'una empresa i que ofereixi una interfície simple i flexible (tan externa com interna que permeti incorporar nous mòduls sense que afecti al disseny).

Per un altre costat s'espera adquirir i aprofundir en els coneixements obtinguts en altres assignatures de la carrera així com aprendre competències pròpies de la tecnologia .NET de Microsoft.

PRODUCTES OBTINGUTS

Els productes obtinguts són els següents:

- Aplicació Web per gestionar les despeses del personal d'una empresa. Codi font amb la solució anomenada 'sDespesesPersonals' que inclou tres projectes:
 - CapaDadesDespesesEF. Projecte realitzat amb Entity Framework
 - CapaDadesDespesesWCF. Projecte realitzat amb WCF
 - WebGestorDespeses. Projecte realitzat amb ASP.NET Web Forms
- Backup de la base de dades anomenat 'GestioDespeses.bak' que inclou una còpia de la base de dades inicial.
- Memòria amb una síntesis dels documents realitzats en fases anteriors: Planificació, Anàlisi i Disseny i guia d'implementació.
- Manual d'usuari i Guia de Configuració
- Presentació Virtual: PowerPoint i Vídeo per a la presentació del producte.

TREBALL FI DE CARRERA

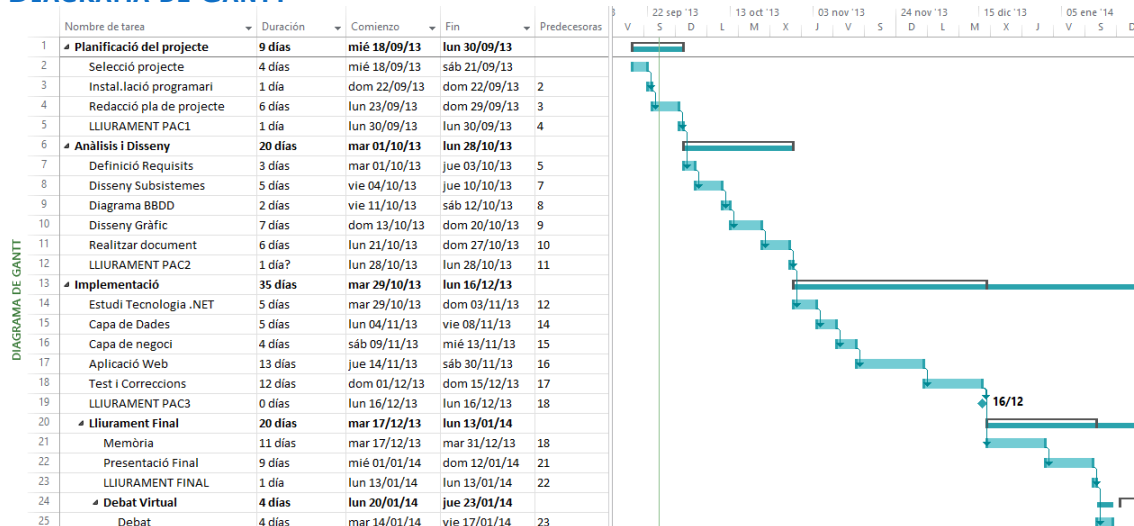
PLANIFICACIÓ INICIAL VS PLANIFICACIÓ FINAL

La planificació inicial és:

Id	Tasca	Duració	Inici	Fi	Precedent
Planificació del projecte					
1	Selecció del projecte	4	18/09/2013	21/09/2013	
2	Instal·lació del programari	1	22/09/2013	22/09/2013	1
3	Realitzar pla de projecte	6	23/09/2013	29/09/2013	2
4	LLIURAMENT PAC1	1	30/09/2013	30/09/2013	3
Anàlisis i Disseny					
5	Definició formal requisits	3	01/10/2013	03/10/2013	4
6	Disseny subsistemes	5	04/10/2013	10/10/2013	5
7	Diagrama de BBDD	2	11/10/2013	12/10/2013	6
8	Disseny Gràfic (UI)	7	13/10/2013	20/10/2013	7
9	Realitzar document PAC2	6	21/10/2013	27/10/2013	8
10	LLIURAMENT PAC2	1	28/10/2013	28/10/2013	9
Implementació					
11	Estudi Tecnologia .NET	5	29/10/2013	03/11/2013	10
12	Capa de dades	5	04/11/2013	08/11/2013	11
13	Capa de negoci	4	09/11/2013	13/11/2013	12
14	Aplicació web	13	14/11/2013	30/11/2013	13
15	Test i correccions	12	01/12/2013	15/12/2013	14
16	LLIURAMENT PAC3	1	16/12/2013	16/12/2013	15
Lliurament Final					
17	Memòria	11	17/12/2013	31/12/2013	16
18	Presentació final	9	01/01/2014	12/01/2014	17
19	LLIURAMENT FINAL	1	13/01/2013	13/01/2014	18
Debat Virtual					
20	Debat	4	20/01/2014	23/01/2014	19

La planificació Final no s'inclou ja que s'ha pogut seguir el ritme de la planificació inicial proposada.

DIAGRAMA DE GANTT



ANÀLISI DE RISCOS

Segons el PMBOK, un risc és un event o condició incerta que, si succeeix, té un efecte en almenys un dels objectius del projecte.

PLANIFICACIÓ

- La planificació inicial omet tasques que són necessàries per a la bona resolució de la solució.
- Planificació optimista en lloc de realista.
- El retard en una tasca/activitat produeix retards en cascada en les tasques dependents.
- Les àrees desconegudes del projecte poden requerir més temps de l'esperat en el disseny i implementació.

DISSENY I IMPLEMENTACIÓ

- Disseny massa senzill que no cobreixi les qüestions principals i sigui necessari tornar a dissenyar i implementar.
- Disseny massa complex que exigeixi tenir en compte complicacions innecessàries i faci improductiva la implementació.
- Un mal disseny que impliqui tornar a dissenyar i implementar.
- La utilització de metodologies desconegudes pot derivar en un període extra de formació amb el risc de tornar enrere per a corregir els errors inicials comesos en la metodologia.
- Els components desenvolupats per separat que no es puguin integrar de forma senzilla havent de tornar a dissenyar i repetir varis treballs.

PRODUCTE

- Els mòduls propensos a tenir errors igual necessitin més treball de comprovació, disseny i implementació.
- Utilitzar tecnologia desconeguda pugui allargar la planificació de forma imprevisible.
- El desenvolupament de funcions software incorrectes pugui requerir de tornar-los a dissenyar i implementar.
- El desenvolupament d'una interfície d'usuari inadequada pugui requerir de tornar-la a dissenyar i a implementar.

PERSONAL

- La formació en les tecnologies .NET no es completi a temps.
- Inexperiència tècnica en la plataforma .NET
- Malalties o urgències que provoquin algun retràs no desitjat.
- Pèrdua de dades o informació del treball realitzat.

3. Anàlisis i Disseny

S'inclouen a la memòria els requeriments realitzats a la fase d'anàlisis i disseny i s'incorpora, en cursiva, una explicació dels canvis que s'han fet en el desenvolupament.

REQUERIMENTS FUNCIONALS

Aquest projecte consta d'una aplicació Web realitzada en tecnologia ASP.NET. Estarà formada pels següents subsistemes:

- Gestió de validació al sistema.
- Gestió canvi de clau.
- Gestió de la configuració.
- Gestió dels llistats de treballadors de l'empresa.
- Gestió de les despeses personals.
- Gestió de les autoritzacions de les fulles de despesa.
- Gestió del calendari de dies festius.

L'aplicació consta de tres tipus d'usuaris: Administrador, Autoritzador i Treballador. Els treballadors sols tenen accés als subsistemes de despeses personals, validació de sistema, canvi de clau *i veure el calendari de dies festius*

Els Autoritzadors tenen accés a les mateixes funcionalitats que els treballadors i a més a les autoritzacions de fulles de despesa.

Els Administradors tenen accés a totes les funcionalitats del sistema.

Inicialment l'aplicació disposa d'un usuari administrador que no es pot eliminar del sistema.

Gestió Validació al sistema

Inicialment es mostra una pantalla on l'usuari ha d'introduir un login i clau.

Si l'usuari no existeix es mostra una pantalla d'error i en cas contrari permet l'entrada al sistema. Segons el tipus d'usuari que ha iniciat sessió s'habilitaran més o menys funcionalitats.

Gestió canvi de clau

Aquesta funcionalitat permet als usuaris canviar la clau d'accés al sistema.

L'administrador podrà canviar la clau de qualsevol usuari des de la funcionalitat dels treballadors.

Gestió Despeses Personals

Tots els treballadors de l'empresa tenen l'opció de crear fulles de despeses que s'associen automàticament a la nòmina del mes en curs.

Des d'aquest subsistema els usuaris poden donar d'alta, editar i eliminar fulles de despesa.

Es poden afegir tantes fulles de despeses com calgui però no es poden obrir altres *mentre hi hagi alguna fulla en estat pendent*.

A una fulla de despesa se li afegirà com a mínim una línia de despesa.

Si una fulla de despesa està en estat tancat no es pot eliminar ni editar.

Quan es doni d'alta una fulla amb una despesa se li assigna, per defecte, l'estat pendent i l'usuari podrà sol·licitar el pagament quan ho desitgi.

Els administradors no poden gestionar les despeses personals de la resta dels treballadors (fora de l'abast)

Gestió Autoritzacions Despeses Personals

Els usuaris Autoritzadors poden cercar fulles de despeses i aquelles que estiguin en estat 'sol·licitat pagament' les podran autoritzar, refusar o tancar.

Una fulla autoritzada també es podrà refusar o a la inversa, si l'usuari s'ha equivocat. Però les fulles pendents o tancades no les podrà canviar d'estat.

En el moment de canviar l'estat (autoritzar, denegar o tancar) d'una fulla s'ha d'introduir un motiu.

El usuaris Autoritzadors sols poden fer autoritzacions sobre els treballadors dels quals són caps o sobre tots els treballadors (Segons el tipus d'autorització escollida) Els Autoritzadors també poden tancar fulles.

TREBALL FI DE CARRERA

Gestió dels treballadors de l'empresa

Des d'aquest subsistema es poden realitzar les següents opcions: alta i cerca de treballadors de l'empresa.

La cerca es pot realitzar en base a diferents paràmetres i el resultat es mostra en una llista.

En seleccionar un element de la llista es podrà editar o eliminar (exceptuant l'usuari administrador inicial que no es pot eliminar).

L'opció d'eliminar comporta una baixa lògica de la base de dades.

Els tipus d'usuaris que es poden crear des d'aquí són: Administradors, Autoritzadors i Treballadors.

Sols els administradors tenen accés a aquesta funcionalitat.

Gestió de la configuració del sistema

La gestió dels diccionaris mestres sols la poden realitzar els administradors de l'aplicació.

La funcionalitat és la mateixa per a tots els diccionaris bàsics:

Es permet donar d'alta un element que automàticament es visualitza en una llista.

En aquesta mateixa llista es mostren tots els elements del diccionari i seleccionant un es pot editar o eliminar (sols si l'element no està relacionat amb algun registre d'alguna taula relaciona o *mentre no sigui l'últim element del diccionari ja que sinó l'aplicació no funcionaria correctament*)

Comentaris: *En general s'han respectat les funcionalitats estudiades a la fase d'anàlisi exceptuant algun petit canvi per millorar l'aplicació final.*

REQUERIMENTS NO FUNCIONALS

Són els mateixos que s'han definit en la planificació inicial del projecte. Veieu el punt Requeriments a implementar del punt anterior.

DIAGRAMA DE CASOS D'ÚS

ESPECIFICACIÓ GRÀFICA

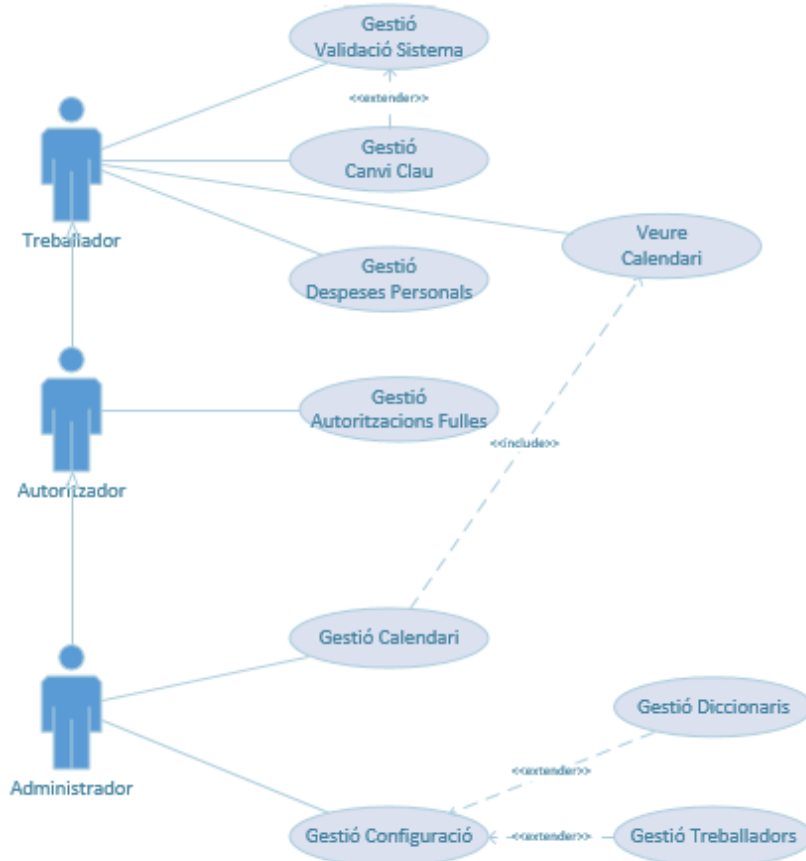
En el sistema hi han tres tipus bàsics de perfils: Administradors, Autoritzadors i Treballadors.



(Informació) Els casos d'ús que es mostren reflecteixen els canvis fets a la fase de desenvolupament. A la part inferior de cada imatge hi ha una petita explicació, en cursiva, dels canvis que s'han fet al desenvolupament.

TREBALL FI DE CARRERA

CAS D'ÚS AMB VISTA GLOBAL

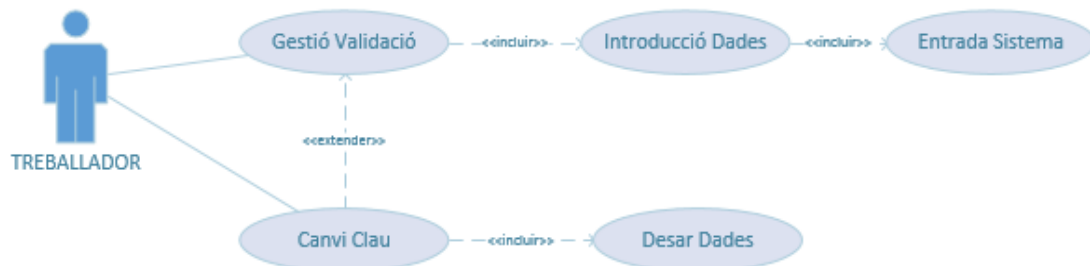


Canvis realitzats en el desenvolupament: Al diagrama anterior s'ha incorporat els cas d'us 'Veure Calendari' i 'Gestió Diccionaris' que no es va especificar a la fase d'anàlisi.

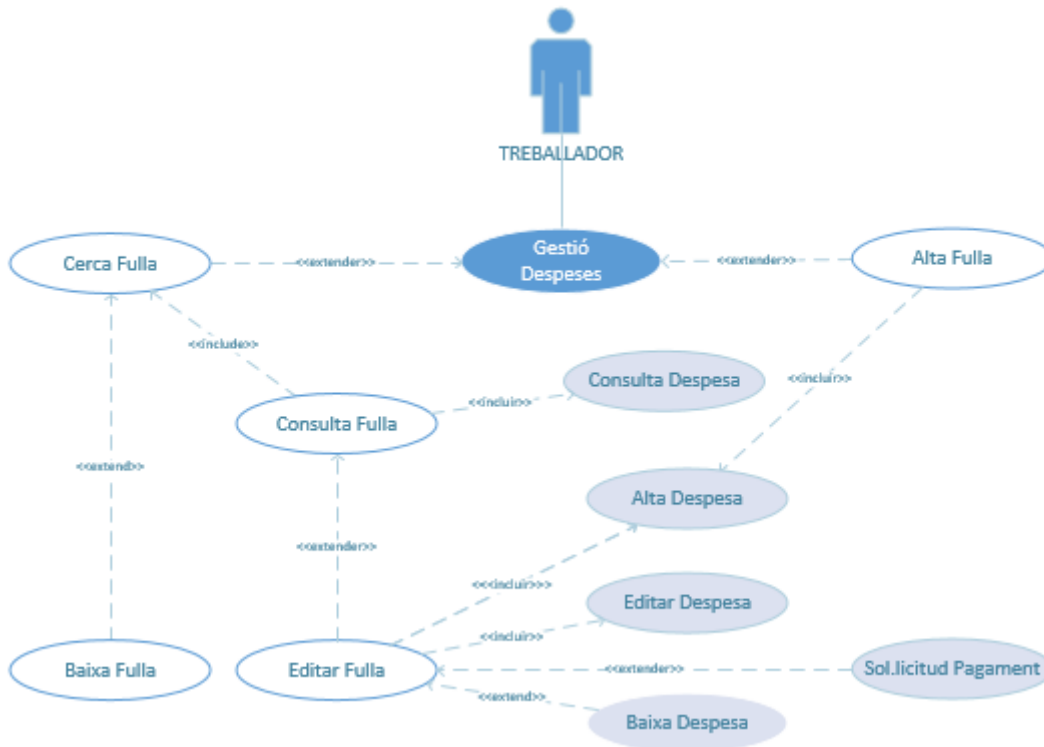
Veiem els subsistemes amb més detall:

GESTIÓ VALIDACIÓ I CANVI CLAU

GESTIÓ VALIDACIÓ I CANVI CLAU

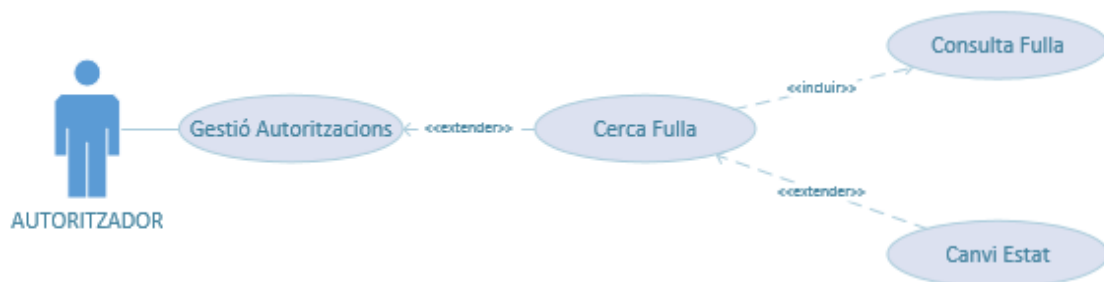


GESTIÓ DESPESES PERSONALS



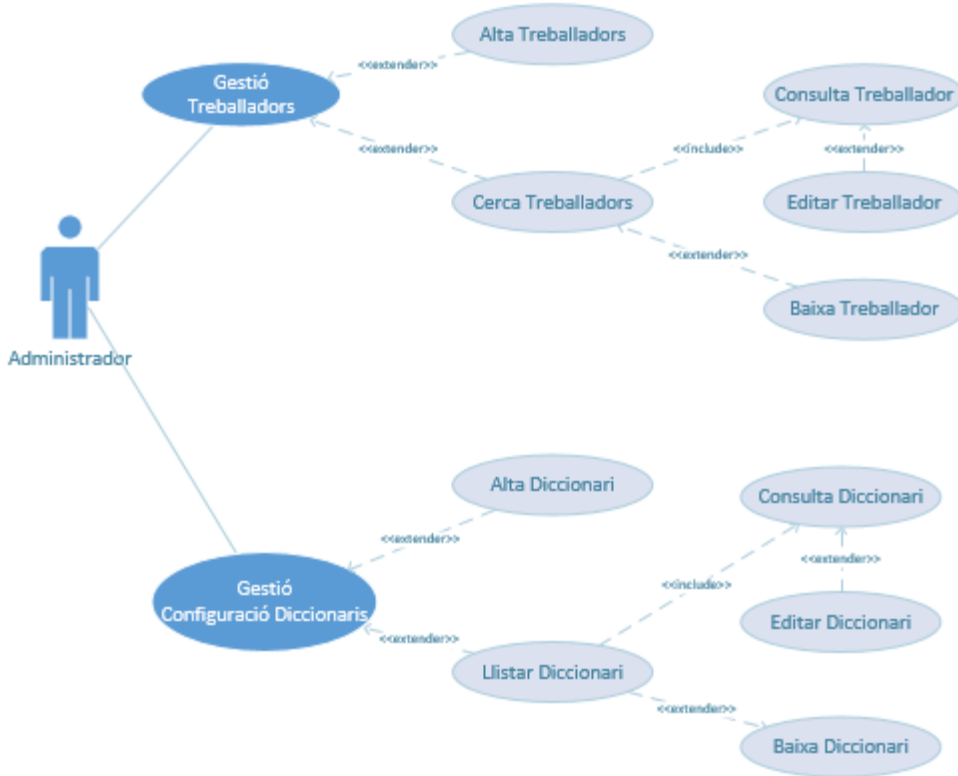
Canvis realitzats en el desenvolupament: S'ha reestructurat aquest cas d'ús per reflectir el que s'ha fet a la part d'implementació per evitar que l'ús de la Web no sigui massa complicat.

GESTIÓ AUTORIZACIONS



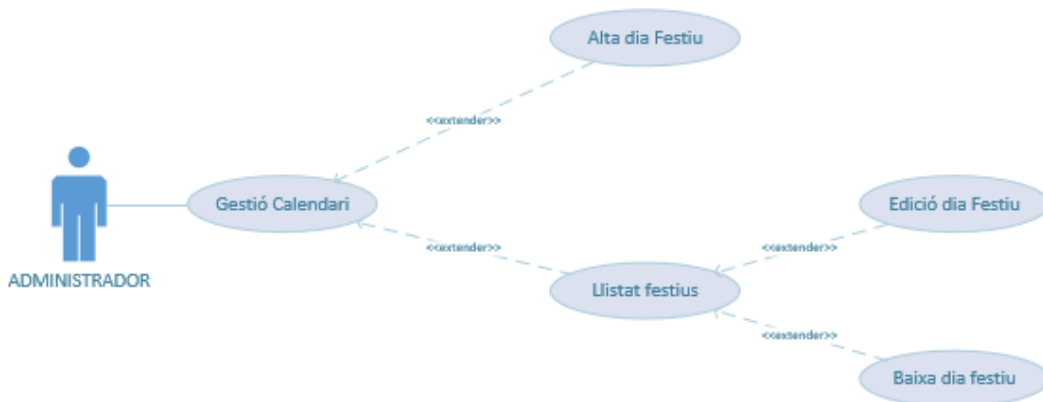
Canvis realitzats en el desenvolupament: En la part d'anàlisi des del cas d'ús 'consulta de fulla' hi havia un accés al cas d'ús 'canvi d'estat' però en el desenvolupament es pot navegar al cas d'ús 'Canvi d'Estat' des del 'Cerca de Fulla'.

GESTIÓ CONFIGURACIÓ I GESTIÓ TREBALLADORS



Canvis realitzats en el desenvolupament: El canvi que s'ha fet per millorar l'ús de la web és: des de els casos d'ús 'Consulta...' es pot navegar a 'Editar...' i no des de la cerca o la llista directament.

GESTIÓ CALENDARI



Canvis realitzats en el desenvolupament: S'ha afegit el cas d'ús 'Edició dia Festiu' per poder editar la descripció.

TREBALL FI DE CARRERA

ESPECIFICACIÓ TEXTUAL

Descripció textual sobre cada un dels casos d'ús dels subsistemes.

A les observacions s'afegeixen els canvis realitzats al desenvolupament

SUBSISTEMA DE VALIDACIÓ

Nom del Cas d'ús:	VALIDACIÓ AL SISTEMA
Descripció general	Entrada al sistema.
Actors	Tots
Casos d'ús relacionats	
Precondició	Ha d'existir l'usuari a la base de dades.
Postcondició	L'usuari es valida i entra al sistema
Descripció	Abans d'accedir al sistema es necessari introduir un login i clau. Si les dades introduïdes són correctes l'usuari entrarà al sistema i sinó visualitzarà un missatge d'error.
Regles de negoci	
Observacions	

SUBSISTEMA CANVI DE CLAU

Nom del Cas d'ús:	CANVI CLAU
Descripció general	Canviar la clau d'un usuari del sistema
Actors	Tots
Casos d'ús relacionats	
Precondició	L'usuari s'ha validat al sistema
Postcondició	L'usuari canvia la clau.
Descripció	L'usuari introdueix la clau vella i la nova dues vegades. Si les dades introduïdes son correctes el sistema canviarà la clau a l'usuari
Regles de negoci	<ul style="list-style-type: none"> La clau vella introduïda ha de coincidir amb la clau actual de l'usuari validat. Les dues claus noves introduïdes han de coincidir.
Observacions	

SUBSISTEMA DESPESES PERSONALS

Nom del Cas d'ús:	ALTA FULLA DESPESA
Descripció general	Els usuaris del sistema podran donar d'alta una fulla de despesa que anirà vinculada a la nòmina en curs
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Alta, edició, baixa línies de despeses.
Precondició	Sols es pot donar d'alta una fulla si no hi ha cap fulla pendent
Postcondició	Acabat el procés s'haurà creat una nova fulla al sistema vinculada a la nòmina en curs, amb estat PENDENT
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt

TREBALL FI DE CARRERA

	"Alta Fulla Despesa", introduirà les dades de la fulla i d'una despesa i guardarà o cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> • El camp ID és auto numèric i no està habilitat. • L'actor ha d'introduir una línia de despesa inicialment sinó no es grava la fulla. • No es podrà afegir una nova fulla mentre hi hagi alguna de pendent
Observacions	<p>Al desenvolupament s'han realitzat també les següents validacions abans de desar la fulla:</p> <ul style="list-style-type: none"> • Es comprova que els camps marcats amb (*) s'hagi introduït un valor. • Es comprova que el format de la data i el preu siguin correctes. • Es comprova que si han adjuntat un fitxer, l'extensió del fitxer adjunt sigui JPG, PNG, JPEG o GIF • Segons el tipus de despesa escollida: <ul style="list-style-type: none"> ▪ Sigui obligatori adjuntar un fitxer. ▪ Sigui necessari introduir un preu inferior o igual al preu màxim.

Nom del Cas d'ús:	SOL.LICITUD PAGAMENT FULLA DESPESA
Descripció general	Els usuaris del sistema podran sol·licitar el pagament d'una fulla de despesa
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta fulla.
Precondició	Sols es pot sol·licitar el pagament d'una fulla si el seu estat és PENDENT
Postcondició	Acabat el procés la fulla passa a estat 'SOL.LICITAT'
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Sol·licitar pagament Fulla Despesa".
Regles Negoci	<ul style="list-style-type: none"> • La fulla ha d'estar en estat PENDENT • La fulla ha de contenir almenys una despesa.
Observacions	Canvis realitzat al desenvolupament: Per sol·licitar el pagament sols cal accedir al punt 'Veure Detalls d'una Fulla' i si la fulla està en estat pendent apareix un botó per fer la sol·licitud.

Nom del Cas d'ús:	CERCA FULLA DESPESA
Descripció general	Els usuaris del sistema podran cercar fulles de despesa
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta Fulla
Precondició	Han d'existir fulles creades.
Postcondició	Acabat el procés s'haurà mostrat en un llistat les fulles que compleixen amb els paràmetres de cerca.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Cerca Fulla Despesa", introduirà els paràmetres de cerca i acceptarà o cancel·larà.

TREBALL FI DE CARRERA

Regles Negoci	<ul style="list-style-type: none"> Els usuaris sols poden cercar fulles de despeses pròpies (inclòs l'administrador)
Observacions	

Nom del Cas d'ús:	EDICIÓ FULLA DESPESA
Descripció general	Els usuaris del sistema podran editar una fulla de despesa.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta Fulla Alta, edició, baixa línies despesa
Precondició	La fulla no està en estat tancada.
Postcondició	Acabat el procés s'haurà editat una fulla del sistema. La fulla passa a estat pendent
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Edició Fulla Despesa" per a modificar les dades.
Regles Negoci	<ul style="list-style-type: none"> El camp ID no està habilitat. L'estat de la fulla no pot estar en estat 'Tancada' Un cop editada la fulla torna a estat 'Pendent'.
Observacions	

Nom del Cas d'ús:	BAIXA FULLA DESPESA
Descripció general	Els usuaris del sistema podran donar de baixa una fulla de despesa.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Cerca Fulla
Precondició	La fulla no està en estat tancada.
Postcondició	Acabat el procés s'haurà donat de baixa una fulla del sistema amb les línies de despesa associades.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Baixa Fulla Despesa".
Regles Negoci	<ul style="list-style-type: none"> L'estat de la fulla no pot estar a 'Tancada' Es tracta d'una baixa física de la base de dades
Observacions	

Nom del Cas d'ús:	CONSULTA FULLA DESPESA
Descripció general	Els usuaris del sistema podran consultar una fulla de despesa.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Edició Fulla
Precondició	La fulla ha d'existir
Postcondició	Acabat el procés s'haurà consultat una fulla del sistema.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Consulta Fulla Despesa".
Regles Negoci	
Observacions	

TREBALL FI DE CARRERA

LÍNIES DE DESPESA

Nom del Cas d'ús:	ALTA LÍNIA DE DESPESA
Descripció general	Els usuaris del sistema podran donar d'alta una línia de despesa que s'afegirà sobre la fulla de despesa actual.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta, edició i baixa de despesa . Edició fulla.
Precondició	Ha d'existir una fulla que no estigui en estat tancada.
Postcondició	Acabat el procés s'haurà creat una nova línia de despesa vinculada a una fulla. La fulla passa a estat pendent.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Fulla Despesa->Alta despesa", introduirà les dades i guardarà o cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> • El camp ID és auto numèric i no està habilitat. • La fulla passa a estat 'pendent' • La fulla no pot estar 'tancada'
Observacions	<p>Al desenvolupament s'han realitzat les següents validacions abans de desar la fulla:</p> <ul style="list-style-type: none"> • Es comprova que els camps marcats amb (*) s'hagi introduït un valor. • Es comprova que el format de la data i el preu siguin correctes. • Es comprova que si han adjuntat un fitxer, l'extensió del fitxer adjunt sigui JPG, PNG, JPEG o GIF • Segons el tipus de despesa escollida <ul style="list-style-type: none"> ▪ Serà obligatori adjuntar un fitxer. ▪ Serà necessari introduir un preu inferior o igual al preu màxim.

Nom del Cas d'ús	EDICIÓ LÍNIA DE DESPESA
Descripció general	Els usuaris del sistema podran editar una línia de despesa existent.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta, edició despesa. Edició Fulla.
Precondició	Ha d'existir una fulla donada d'alta que no estigui en estat tancada.
Postcondició	Acabat el procés s'haurà modificat una línia de despesa existent. La fulla passa a estat pendent.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Fulla Despesa ->Edició Despesa" , modificarà les dades, guardarà o cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> • El camp ID és auto numèric i no està habilitat. • La fulla no pot estar en estat 'tancada'

TREBALL FI DE CARRERA

Observacions	Al desenvolupament s'han realitzat les mateixes validacions, en desar la línia, que en l'alta.
---------------------	--

Nom del Cas d'ús	BAIXA LÍNIA DE DESPESA
Descripció general	Els usuaris del sistema podran eliminar una línia de despesa existent.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Cerca Despesa
Precondició	Ha d'existir una fulla donada d'alta que no estigui en estat tancada.
Postcondició	Acabat el procés s'haurà eliminat una línia de despesa existent.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Fulla Despesa ->Baixa Despesa" , eliminarà la despesa.
Regles Negoci	<ul style="list-style-type: none"> La fulla no pot estar en estat 'tancada'
Observacions	

Nom del Cas d'ús:	CONSULTA LÍNIA DE DESPESA
Descripció general	Els usuaris del sistema podran consultar una línia de despesa existent.
Actors	Administrador, Treballador, Autoritzador
Casos d'ús relacionats	Consulta, Edició Fulla.
Precondició	Ha d'existir una fulla donada d'alta.
Postcondició	Acabat el procés s'haurà consultat una línia de despesa existent.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Fulla Despesa ->Consulta Despesa".
Regles Negoci	
Observacions	

SUBSISTEMA AUTORITZACIONS FULLA DE DESPESA

Nom del Cas d'ús	AUTORITZAR FULLA DE DESPESA
Descripció general	Els Autoritzadors dels sistema podran canviar l'estat a les fulles de despesa.
Actors	Autoritzador
Casos d'ús relacionats	Cerca Despesa
Precondició	Les fulles no poden estar en estat 'Tancat'
Postcondició	Acabat el procés s'haurà modificat l'estat de una fulla de despesa.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Cerca Despesa". Introduirà els paràmetres de cerca i en acceptar es mostrarà en un llistat les fulles que compleixen els paràmetres.
Regles Negoci	<ul style="list-style-type: none"> Sols es permetrà canviar l'estat de la fulla si

	<p>està en estat 'Sol·licitat', 'Autoritzat' o 'Refusat'.</p> <ul style="list-style-type: none"> • Una fulla autoritzada també es pot refusar o a la inversa, si l'usuari s'ha equivocat.
Observacions	El usuari Autoritzadors sols podran fer autoritzacions sobre els treballadors dels qual són el CAP o sobre tots els treballadors (dependrà la configuració del treballador)

SUBSISTEMA DE CONFIGURACIÓ (DICIONARIS BÀSICS)

Com gairebé tots els diccionaris es tracten d'igual forma, es realitza una descripció textual genèrica.

Taules diccionaris: Departament, Estat Fulla, Tipus de Despesa i Tipus Treballador

Nom del Cas d'ús:	ALTA DICIONARI
Descripció general	L'administrador del sistema s'encarregarà d'omplir les dades necessàries per donar d'alta a un nou diccionari en el sistema.
Actors	Administrador
Casos d'ús relacionats	Edició diccionari Baixa diccionari.
Precondició	El registre del diccionari no ha d'estar donat d'alta prèviament.
Postcondició	Acabat el procés el registre del diccionari es trobarà a la base de dades del sistema.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Alta diccionari", introduirà les dades i les guardarà.
Regles Negoci	<ul style="list-style-type: none"> • El camp ID és auto numèric i no està habilitat.
Observacions	

Nom del Cas d'ús	EDICIÓ DICIONARI
Descripció general	L'administrador del sistema s'encarregarà de modificar les dades necessàries per editar un registre d'un diccionari en el sistema.
Actors	Administrador
Casos d'ús relacionats	Alta diccionari. Baixa diccionari.
Precondició	El registre del diccionari ha d'estar donat d'alta prèviament.
Postcondició	Un cop acabat el procés el registre s'ha modificat.
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al diccionari que vol editar, seleccionarà un registre de la llista i podrà modificar les seves dades
Regles Negoci	<ul style="list-style-type: none"> • Els camps ID no estan habilitats.

TREBALL FI DE CARRERA

Nom del Cas d'ús:	CONSULTA DICCIONARI
Descripció general	L'administrador del sistema s'encarregarà de consultar les dades del diccionari.
Actors	Administrador
Casos d'ús relacionats	Alta Diccionari. Baixa Diccionari. Edició Diccionari.
Precondició	El diccionari ha de contenir dades.
Postcondició	Es mostren les dades del registre del diccionari.
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al punt Diccionari per visualitzar les seves dades

Nom del Cas d'ús:	BAIXA DICCIONARI
Descripció general	L'administrador del sistema s'encarregarà d'eliminar el registre del sistema.
Actors	Administrador
Casos d'ús relacionats	Edició diccionari. Alta diccionari.
Precondició	El registre del diccionari ha d'estar donat d'alta prèviament.
Postcondició	Un cop acabat el procés el registre del diccionari no existirà a la base de dades.
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al diccionari, seleccionarà un registre de la llista i podrà donar-lo de baixa.
Regles Negoci	<ul style="list-style-type: none"> • Es tracta d'una baixa física de la base de dades. • Els registres relacionats amb altres taules no es podran modificar.
Observacions	En el desenvolupament també s'ha controlat que no es pugui eliminar un registre si aquest és l'únic a la taula.

SUBSISTEMA TREBALLADORS

Nom del Cas d'ús	ALTA TREBALLADOR
Descripció general	L'administrador del sistema s'encarregarà d'omplir les dades necessàries per donar d'alta un nou treballador en el sistema.
Actors	Administrador
Casos d'ús relacionats	Edició treballador. Baixa treballador. Consulta Treballador
Precondició	El treballador no ha d'estar donat d'alta prèviament.
Postcondició	Acabat el procés, l'usuari es trobarà a la base de dades del sistema.
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Alta treballador", introduirà les dades i les guardarà o

	cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> • El camp ID és auto numèric i no està habilitat. • El login i la clau (simultàniament) no es podran repetir amb cap usuari existent. • El login s'omplirà automàticament amb el NIF del treballador i no es pot modificar • La clau es generarà automàticament però es pot modificar.
Observacions	<p>En el desenvolupament es valida la següent informació abans de desar:</p> <ul style="list-style-type: none"> • Es comprova que s'hagi introduït valor a tots els camps obligatoris • Es comprova que el format del NIF sigui correcte • Es comprova que el NIF no existeixi (per a les altes) • Es comprova que si es marca 'És Cap' s'hagi introduït com a mínim un treballador com a subordinat • Es comprova que no s'hagi seleccionat a un treballador com a CAP i Subordinat al mateix temps.

Nom del Cas d'ús	EDICIÓ TREBALLADOR
Descripció general	L'administrador del sistema s'encarregarà de modificar les dades necessàries dels treballadors en el sistema.
Actors	Administrador
Casos d'ús relacionats	Alta Treballador. Baixa Treballador. Consulta Treballador
Precondició	El treballador ha d'estar donat d'alta prèviament.
Postcondició	El treballador ha estat modificat.
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al punt treballadors, cercarà i seleccionarà un treballador de la llista per a modificar les seves dades
Regles Negoci	<ul style="list-style-type: none"> • Els camps ID i login no estan habilitats. • El login i la clau (simultàniament) no es podran repetir amb cap treballador existent.
Observacions	<p>Canvis realitzat en el desenvolupament:</p> <ul style="list-style-type: none"> • En desar les dades s'han realitzat les mateixes validacions que en l'alta. • Un usuari marcat com a baixa no es pot editar. • Un usuari marcat com a baixa, és pot tornar a marca com a actiu.

TREBALL FI DE CARRERA

Nom del Cas d'ús	CONSULTA TREBALLADOR
Descripció general	L'administrador del sistema s'encarregarà de consultar les dades necessàries dels treballadors en el sistema.
Actors	Administrador
Casos d'ús relacionats	Alta Treballador. Baixa Treballador. Edició Treballador
Precondició	El treballador ha d'estar donat d'alta prèviament.
Postcondició	Es mostren les dades del treballador
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al punt treballadors, cercarà i seleccionarà un treballador de la llista per a visualitzar les seves dades
Regles Negoci	
Observacions	

Nom del Cas d'ús	BAIXA TREBALLADOR
Descripció general	L'administrador del sistema s'encarregarà d'eliminar treballador del sistema.
Actors	Administrador
Casos d'ús relacionats	Edició treballador. Alta treballador. Consulta treballador
Precondició	L'usuari ha d'estar donat d'alta prèviament.
Postcondició	Es marca l'usuari com a baixa omplint-se la data de baixa amb la data actual
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al punt treballadors, seleccionarà un treballador de la llista i podrà eliminar-lo.
Regles Negoci	<ul style="list-style-type: none"> Es tracta d'una baixa lògica de la base de dades. L'usuari Administrador central (el que ve per defecte en instal·lar l'aplicació) no es podrà eliminar.
Observacions	

SUBSISTEMA CALENDARI DIES FESTIUS

Nom del Cas d'ús:	ALTA DIA FESTIU
Descripció general	L'administrador del sistema s'encarregarà de donar d'alta els dies festius de l'any en curs.
Actors	Administrador
Casos d'ús relacionats	Consulta Dies Festius Baixa Dia Festiu Edició Dia Festiu
Precondició	
Postcondició	Acabat el procés s'ha afegit un dia festiu al Calendari
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt

TREBALL FI DE CARRERA

	"Alta dia festiu", introduirà una data i descripció. La guardarà o cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> El camp ID és auto numèric i no està habilitat.
Observacions	Canvis realitzat al desenvolupament: <ul style="list-style-type: none"> No es permet donar d'alta una data ja existent.

Nom del Cas d'ús:	EDICIÓ DIA FESTIU
Descripció general	L'administrador del sistema s'encarregarà d'editar els dies festius de l'any en curs.
Actors	Administrador
Casos d'ús relacionats	Consulta dies Festius Baixa dia Festiu Alta dia Festiu
Precondició	Ha d'existir prèviament el dia festiu
Postcondició	Acabat el procés s'ha editat la descripció del dia festiu
Descripció	Un cop l'actor s'ha validat al sistema accedirà al punt "Edició dia festiu", introduirà una nova descripció. La guardarà o cancel·larà.
Regles Negoci	<ul style="list-style-type: none"> El camp ID és auto numèric i no està habilitat.
Observacions	Canvis realitzat al desenvolupament: <ul style="list-style-type: none"> Aquest cas d'ús no es va especificar a la fase d'anàlisis.

Nom del Cas d'ús	CONSULTA CALENDARI
Descripció general	L'administrador del sistema s'encarregarà de consultar les dades del calendari dels dies festius
Actors	Administrador
Casos d'ús relacionats	Alta Dia Festiu. Edició Dia Festiu Baixa Dia Festiu.
Precondició	Ha d'existir dates creades.
Postcondició	Es mostren les dades del calendari
Descripció	Un cop l'administrador s'ha validat al sistema accedirà al punt 'Consulta Calendari' per visualitzar les seves dades
Regles Negoci	
Observacions	

Nom del Cas d'ús	BAIXA DIA FESTIU
Descripció general	L'administrador del sistema s'encarregarà d'eliminar un dia festiu del calendari.
Actors	Administrador
Casos d'ús relacionats	Alta Calendari. Consulta Calendari.
Precondició	El registre ha d'estar donat d'alta prèviament.
Postcondició	Un cop acabat el procés, el dia festiu no existirà al calendari

TREBALL FI DE CARRERA

Descripció	Un cop l'administrador s'ha validat al sistema accedirà al Calendari->Baixa, seleccionarà un dia festiu de la llista i l'eliminarà o cancel·larà.
Regles Negoci	Es tracta d'una baixa física de la base de dades.

MODEL CONCEPTUAL

DIAGRAMA E-R

Veiem les entitats i les relacions entre elles.

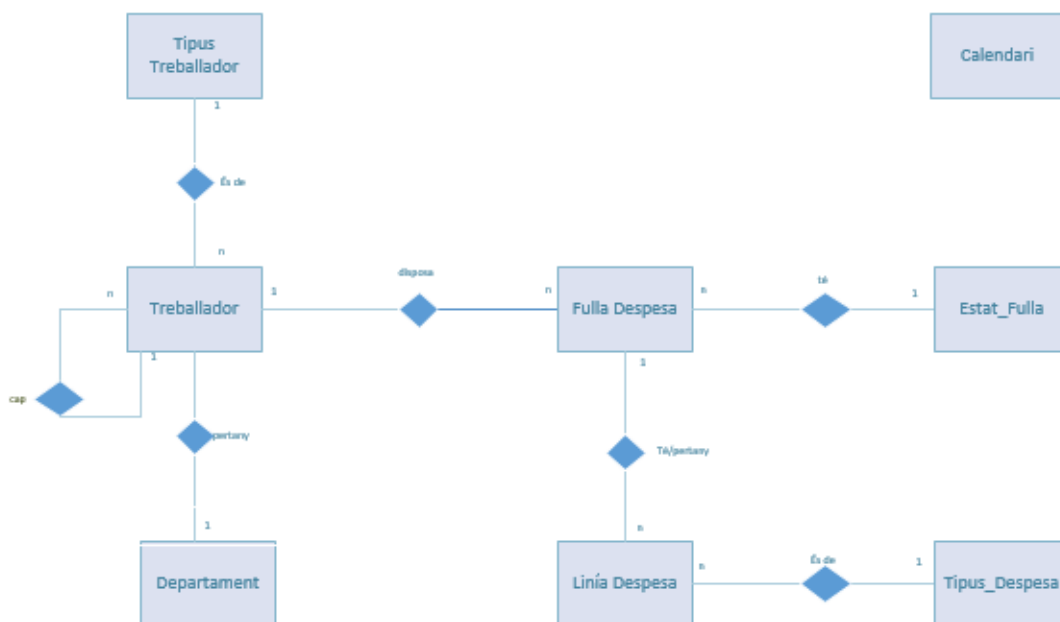
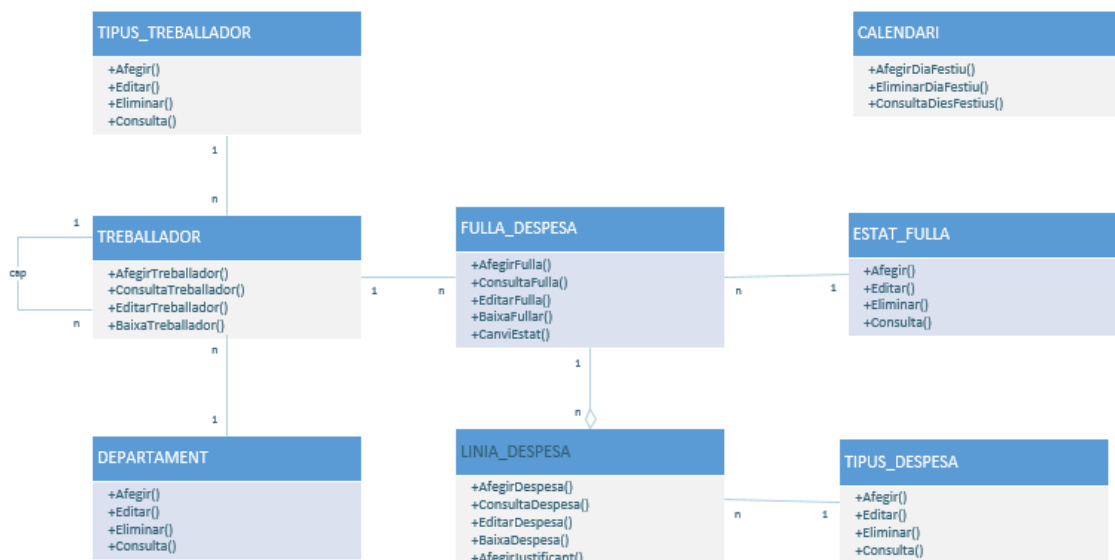


DIAGRAMA DE CLASSES UML



TREBALL FI DE CARRERA

Els atributs de les entitats que figuren al diagrama són els següents (les claus primàries són les subratllades i les claus foranes es marquen en cursiva):

TREBALLADOR (idTreballador, login, contrasenya, nomCompleto, nif, telèfon, es_admin, es_autoritzador, dataAlta, dataBaixa *idTipusTreballador*, *idDepartament*, *idTreballadorCap*)

ON {*idTipusTreballador*} referencia TIPUS_USUARI

ON {*idDepartament*} referencia DEPARTAMENT

ON {*idTreballadorCap*} referencia TREBALLADOR

FULLA_DESPESA (idFulla, data, descripció, *idTreballador*, *idEstat*, motiuEstat)

ON {*idTreballador*} referencia TREBALLADOR

ON {*idEstat*} referencia ESTAT_FULLA

LINIA_DESPESA (idDespesa, data, descripció, preu, justificant, *idFulla*, *idTipus*)

ON {*idFulla*} referencia FULLA_DESPESA

ON {*idTipus*} referencia TIPUS_DESPESA

CALENDARI (idCalendari, data, descripcio)

DICCIONARIS BÀSICS

TIPUS_TREBALLADOR (idTipusTreballador, descripció)

DEPARTAMENT (idDepartament, descripció)

ESTAT_FULLA (idEstat, descripció)

TIPUS_DESPESA (idTipus, descripcio, importMaxim, calJustificant)

JUSTIFICACIÓ DEL MODEL

A partir del disseny conceptual es detallen els aspectes de les entitats obtingudes així com les relacions existents:

- L'entitat TREBALLADOR té una relació binària (1:N) reflexiva sobre si mateixa ja que un treballador té un cap i un cap pot tenir sota la seva responsabilitat més d'un treballador.
- Les entitats TREBALLADOR i TIPUS_TREBALLADOR tenen una interrelació binària (1:N) ja que un treballador és d'un tipus de treballador. Per contra un tipus de treballador té associat molts treballadors.
- Les entitats TREBALLADOR i DEPARTAMENT tenen una interrelació binària (1:N) ja que, en aquest cas, un treballador és d'un departament però en un departament n'hi ha molts treballadors.
- Les entitats TREBALLADOR i FULLA_DESPESA tenen una interrelació binària (1:N) ja que un treballador pot crear moltes fulles de despesa però una fulla sols pot estar creada per un sol treballador.

TREBALL FI DE CARRERA

- Les entitats FULLA_DESPESA i LÍNIA_DESPESA tenen una interrelació binària (1:N) ja que una fulla pot estar formada per una o moltes despeses. Una Despesa, però, pertany a una sola fulla.
- Les entitats FULLA_DESPESA i ESTAT_FULLA tenen una interrelació binària (1:N) ja que una fulla pot estar en un sol estat però hi ha moltes fulles amb el mateix estat.
- Les entitats LÍNIA_DESPESA i TIPUS_DESPESA tenen una interrelació binària (1:N) ja que una despesa es d'un tipus però hi ha moltes despeses que tenen el mateix tipus.

DIAGRAMA D'ARQUITECTURA

COMPONENTS SOFTWARE

La solució de Gestió de Despeses del Personal consta de les següents parts:

Client Web

Es realitzarà amb la tecnologia ASP.NET que és un entorn d'aplicació Web per construir llocs Webs dinàmics.

Model de tres capes

Es farà servir un model de tres capes (Presentació, Negoci i Dades)
Aquesta arquitectura facilita l'ampliació de funcionalitats de forma senzilla.

- Capa de Presentació:
Aquesta capa fa referència a la interfície gràfica d'usuari.
- Capa de Negoci:
Aquesta capa gestiona tota la lògica de negoci i es comunica tant amb la capa de presentació com amb la capa de dades.
- Capa de Dades:
En aquesta capa es gestionarà l'accés als objectes de la base de dades.

COMPONENTS HARDWARE

- Sistema operatiu: Windows 8
- Eines de desenvolupament: Microsoft Visual Studio 2012
- Sistema Gestor de Base de dades: SQL Server 2012

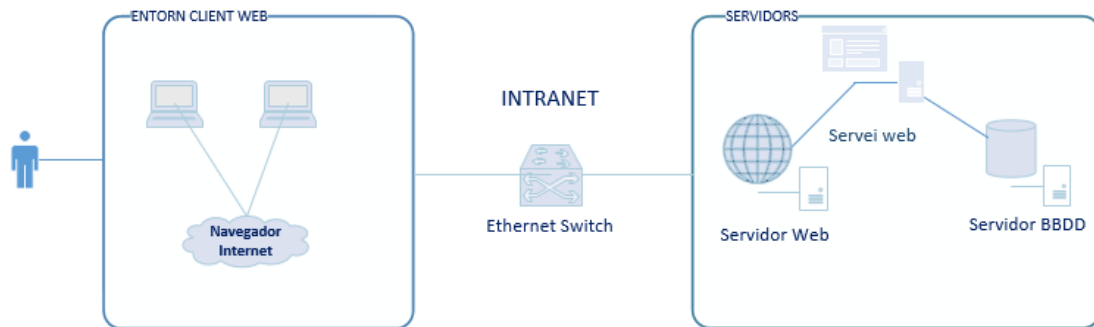
GESTOR DE BASE DE DADE

- Gestor de dades on s'emmagatzemaran totes les dades de l'aplicació.
- El motor de base de dades que es farà servir es MS SQL Server 2012.

TREBALL FI DE CARRERA

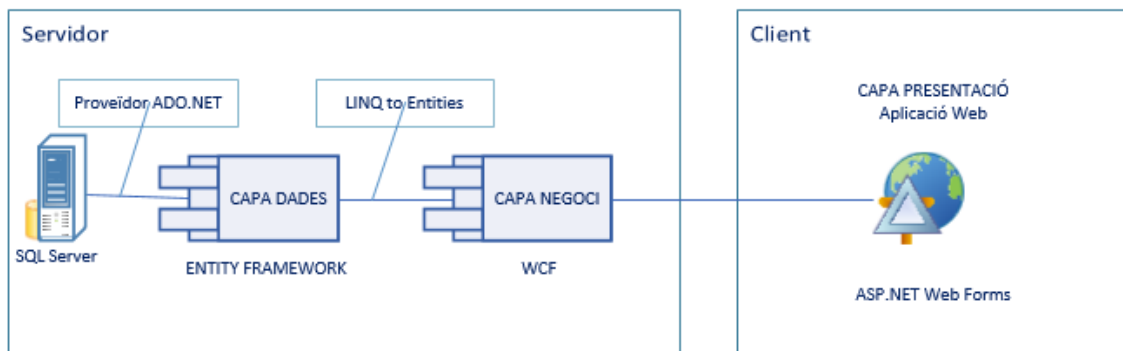
ARQUITECTURA FÍSICA DEL PROJECTE

Veiem l'arquitectura física que tindrà la nostra solució.



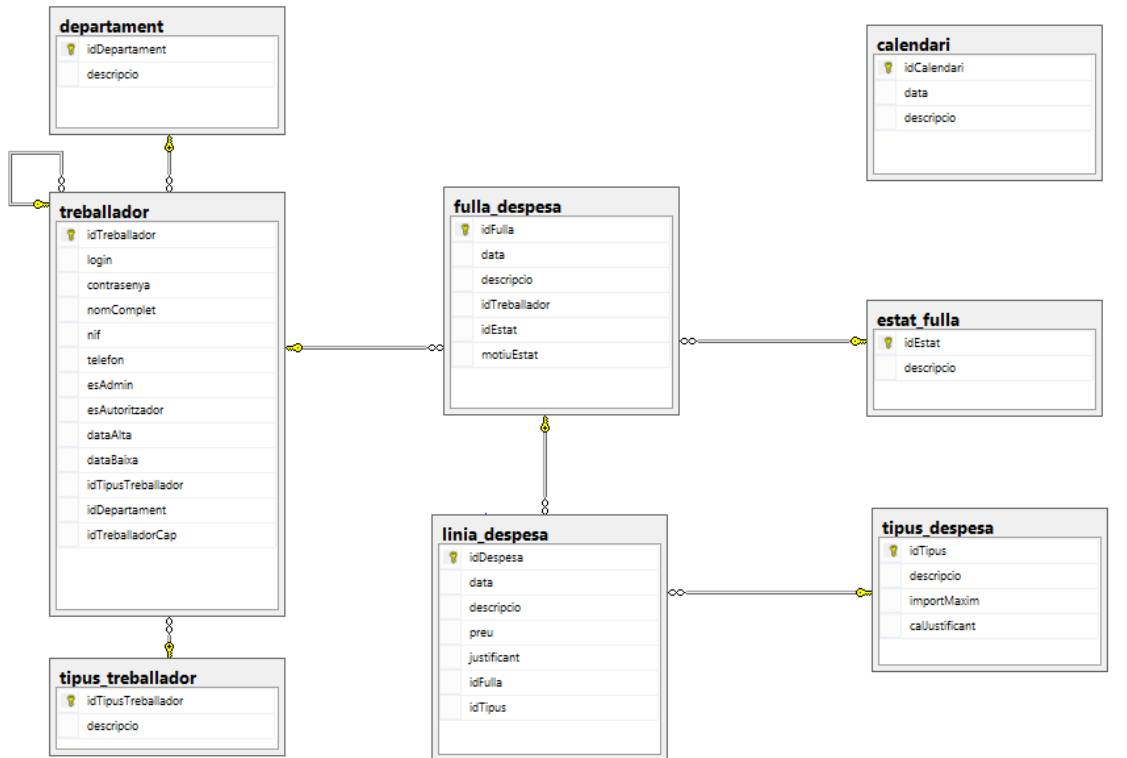
En aquest cas veiem com un client a través d'un navegador fa una petició al servidor web per veure, per exemple, una fulla de despesa, el servidor web fa una demanda de les dades a la BBDD a través del servei web que li retorna la informació.

ESQUEMA D'ARQUITECTURA LÒGICA



La capa de Negoci es comunica amb el model de dades conceptual a través del llenguatge LINQ així com la capa de dades es comunica amb el servidor SQL Server mitjançant el proveïdor ADO.NET.

DISSENY DE LA BBDD



DICCIONARI DE DADES

Veiem les descripcions detallades de les taules i camps.

TREBALLADOR

treballador *

	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idTreballador	int	No	
	login	varchar(50)	No	
	contrasenya	varchar(50)	No	
	nomCompleto	varchar(150)	No	
	nif	varchar(10)	No	
	telefon	varchar(10)	Sí	
	esAdmin	bit	No	((0))
	esAutoritzador	int	No	((0))
	dataAlta	datetime	No	
	dataBaixa	datetime	Sí	
	idTipusTreballador	int	No	
	idDepartament	int	No	
	idTreballadorCap	int	Sí	

TREBALL FI DE CARRERA

El camp 'esAutoritzador' pot contenir els següents valors: 0-NO, 1-Dels treballadors dels qual és cap, 2-De Tots els treballadors

Inicialment contindrà un usuari administrador amb drets a tota l'aplicació.

Veiem el contingut inicial de la taula:

TREBALLADOR					
idTreballador	nomComple	login	clau	esAdmin	esAutoritzador
1	Administrador Central	admin	12345	1	2

Aquest usuari NO ES PODRÀ ELIMINAR DEL SISTEMA.

TIPUS_TREBALLADOR

tipus_treballador				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idTipusTreballador	int	No	
	descripcio	varchar(250)	No	

Contingut Inicial:

TIPUS USUARI	
idTipusTreballador	descripció
1	Comercial
0	Cap projecte
3	Directiu
4	Gestor Nòmina

DEPARTAMENT

departament				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idDepartament	int	No	
	descripcio	varchar(250)	No	

Contingut Inicial:

DEPARTAMENT	
idDepartament	descripció
1	Comercial
2	Màrqueting
3	Projectes
4	Administració
5	Direcció

FULLA_DESPESA

fulla_despesa *				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idFulla	int	No	
	data	datetime	No	
	descripcio	varchar(200)	Sí	
	idTreballador	int	No	
	idEstat	int	No	
	motiuEstat	varchar(MAX)	Sí	

ESTAT_FULLA

estat_fulla				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idEstat	int	No	
	descripcio	varchar(250)	No	

Contingut Inicial

ESTAT_FULLA	
idFulla	descripció
1	Pendent
2	Sol·licitat
3	Autoritzat
4	Refusat
5	Tancat

LINIA_DESPESA

linia_despesa				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idDespesa	int	No	
	data	datetime	No	
	descripcio	varchar(MAX)	No	
	preu	int	No	
	justificant	varbinary(MAX)	Sí	
	idFulla	int	No	
	idTipus	int	No	

TIPUS_DESPESA

tipus_despesa				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
🔑	idTipus	int	No	
	descripcio	varchar(250)	No	
	importMaxim	varchar(250)	Sí	
	calJustificant	bit	No	((0))

Contingut Inicial

TIPUS_DESPESA			
idFulla	descripció	import Màxim	calJustificant
1	Dieta normal	10	0
2	Dieta Festius	20	0
3	Quilometratge		1
4	Bono Tiquet TMB (10 viatges)	14	0
5	Taxi		1
6	Peatge		1
7	Avió		1
8	Autobús		1
9	Metro		1
10	Aparcament		1
10	Altres		1

CALENDARI


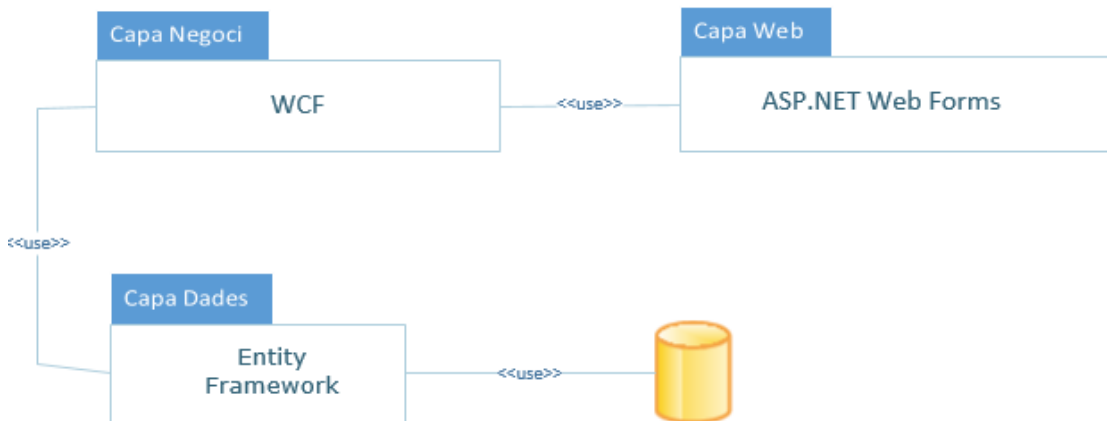
calendari				
	Nombre de columna	Tipo comprimido	Acepta valores NULL	Valor predeterminado
	idCalendari	int	No	
	data	datetime	No	
	descripcio	varchar(250)	No	

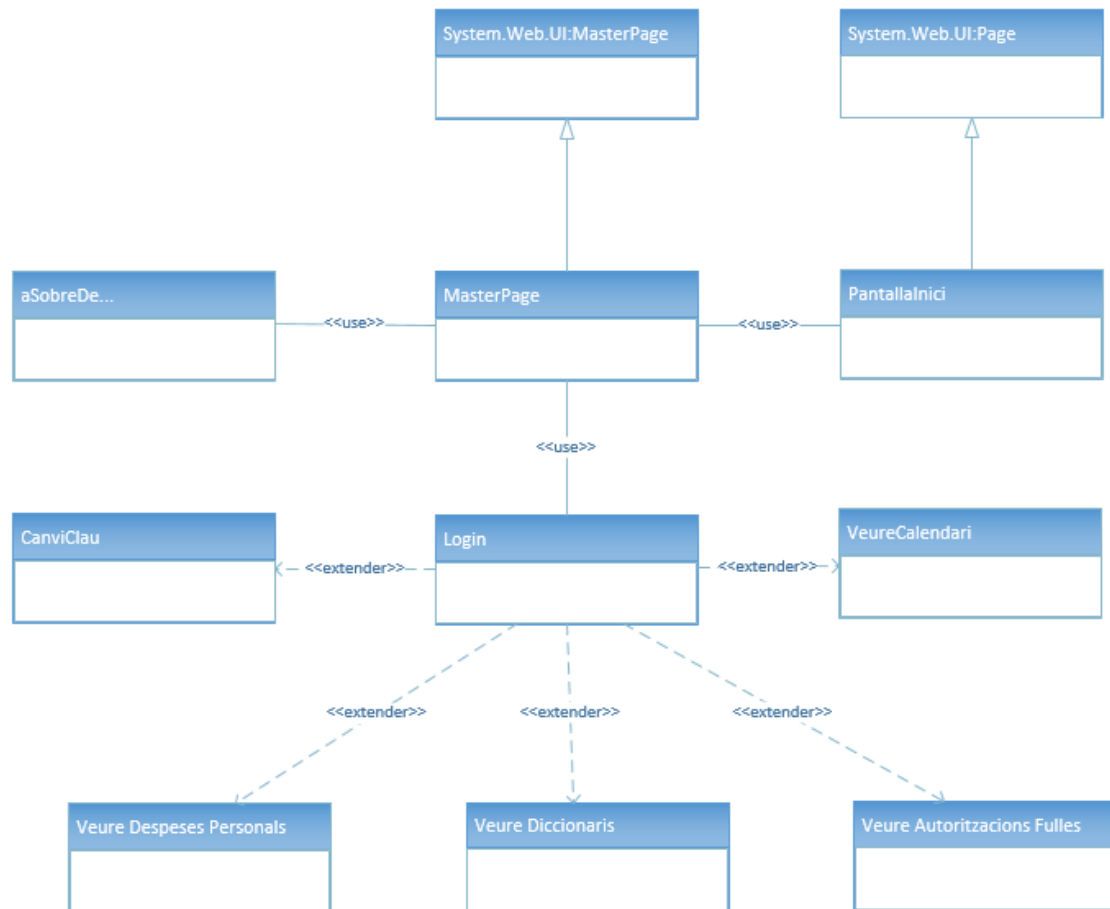
DIAGRAMA DE CLASSES DE LA INTERFÍCIE GRÀFICA

El diagrama de classes de l'aplicació es compon de tres paquets: Existeix un paquet bàsic d'accés a la base de dades al que només es pot accedir a través del paquet de Negoci que gestiona tota la lògica de l'aplicació. Aquest paquet és comunicat amb la Interfície gràfica de l'usuari. És a dir fa de pont entre el paquet web i el paquet de dades.



PACKAGE WEB

Veiem l'estructura Web referent a les pantalles de l'aplicació:



TREBALL FI DE CARRERA

DIAGRAMA D'ESTAT I DE DESPLEGAMENT

Veiem els diferents estats pels qual passa una fulla de despesa

DIAGRAMA D'ESTATS

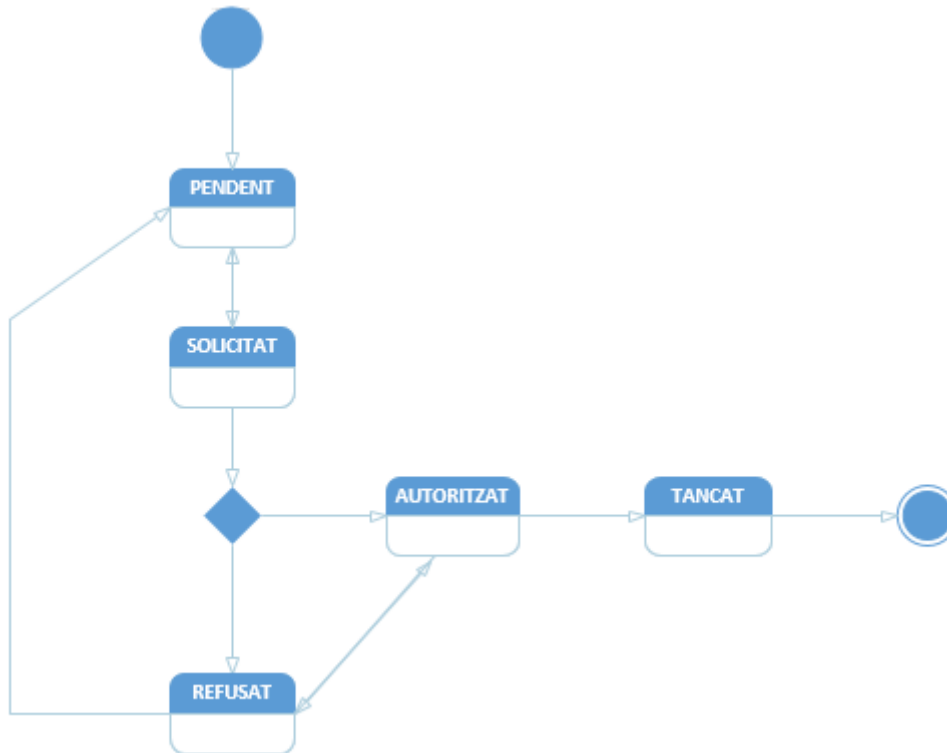
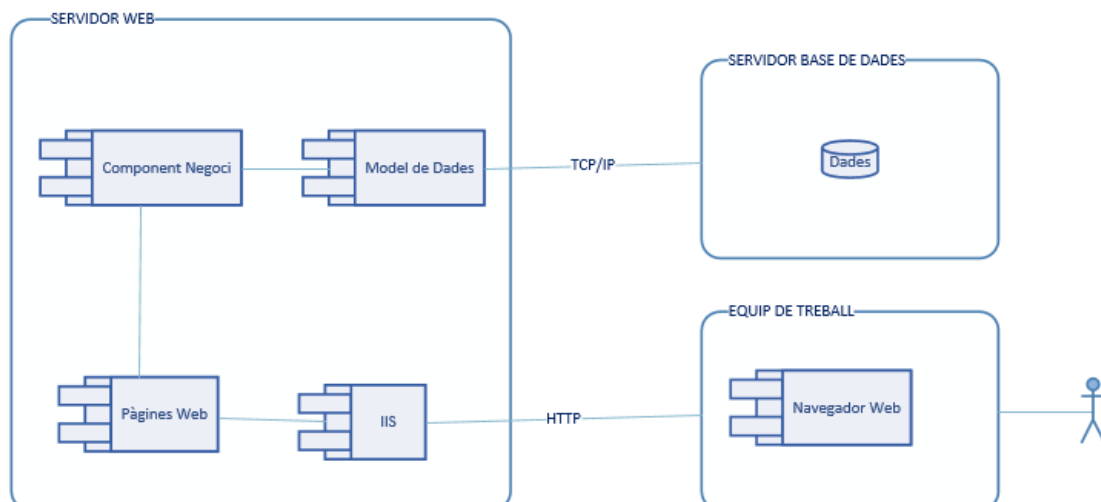


DIAGRAMA DE DESPLEGAMENT



TREBALL FI DE CARRERA

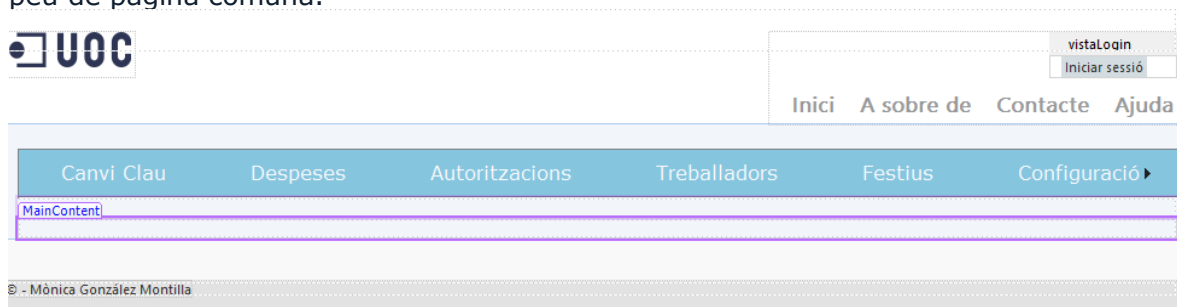
DISSENY DE LA INTERFÍCIE D'USUARI

Per no fer molt extensa aquesta memòria no s'afegeix el detallat de la interfície gràfica. Aquest disseny, però, es pot consultar al Manual d'usuari, que s'adjunta amb el producte lliurat (o també es pot localitzar des de l'opció AJUDA que es troba a la capçalera de la Web)

La plana d'inici que veurà l'usuari a l'entrar a la web és la següent:



Totes les pàgines inclouen la 'Master Page' que s'utilitza per afegir una capçalera i peu de pàgina comuna.



En general, per intentar que la Web sigui senzilla, usable i intuïtiva per a l'usuari, la majoria de les pàgines s'estructuren i funcionen de la mateixa manera:

- Inicialment quan s'accedeix a qualsevol pàgina, a la part superior es mostra un panell de Cerca de les Dades i el botó per Afegir un nou registre.

Cerca Fulles de Despesa









Any:

Estat:

No es mostra més informació per donar sensació de senzillesa.
No totes les pàgines tenen, però el panell de cerca (diccionaris per exemple)

- En realitzar una cerca es mostra en un llistat el resultat:

Llistat de Dades

ID	Descripció	Detalls	Eliminar
5	Direcció		
4	Administració		
3	Projectes		
2	Màrqueting		

1 2 3

El llistat únicament mostra la informació més important, per no sobrecarregar visualment, però disposa d'una comanda per veure els detalls o/i per eliminar l'element.

Si es decideix **eliminar** es demana a l'usuari confirmació prèviament.

La comanda de **veure detalls** amaga el llistat (perquè no hi hagi molta informació en pantalla i fer-ho així més llegible per l'usuari)

Detalls Tipus Despesa

Identificador: 1

Descripció (*):

Especificar import Màxim:

Cal que adjuntin un justificant:

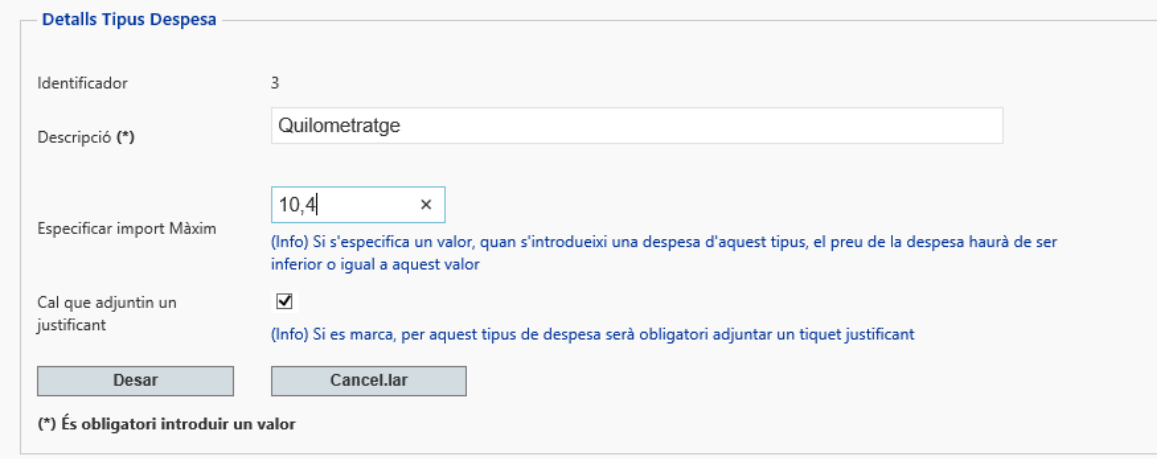
(*): És obligatori introduir un valor

TREBALL FI DE CARRERA

Sobre la imatge anterior:

La comanda **Cancel·lar** tanca el panell de detall i torna a mostrar la pantalla tal i com estava abans de mostrar aquest panell.

Perquè l'usuari no pugui cometre errors d'edició de dades, en mostrar els detalls, tots els controls estan deshabilitats i s'ha de polsar la comanda **'Editar'** perquè s'habilitin, veiem com queda:



Detalls Tipus Despesa

Identificador 3

Descripció (*) Quilometratge

Especificar import Màxim 10,4 x
(Info) Si s'especifica un valor, quan s'introdueixi una despesa d'aquest tipus, el preu de la despesa haurà de ser inferior o igual a aquest valor

Cal que adjuntin un justificant
(Info) Si es marca, per aquest tipus de despesa serà obligatori adjuntar un tiquet justificant

Desar Cancel·lar

(*) És obligatori introduir un valor

Tal i com es veu a la imatge, quan cal, es mostra informació addicional a l'usuari sobre aquells camps que necessiten una explicació (i així no hagin de consultar el manual d'ajuda)

La comanda **'Desar'** pregunta a l'usuari prèviament abans de desar i en cas d'acceptar es guarden els canvis a la BBDD.

La comanda **'Cancel·lar'** descarta qualsevol canvi i tanca el panell.

- La comanda **'Afegir'** obre el panell amb tots els controls habilitats perquè l'usuari pugui introduir la informació. El botó Desar demana confirmació i en cas de confirmació, si les dades són correctes, realitza l'alta de les dades.

4. Desenvolupament

SOFTWARE UTILITZAT

S'ha realitzat amb diferent tecnologies:

- **Entity Framework** és un conjunt de tecnologies ADO.NET que permeten el desenvolupament d'aplicacions de software orientat a les dades.
- **Windows Communication Foundation** és un marc de treball per a la creació d'aplicacions orientades a serveis. Amb WCF es possible enviar dades asíncrones d'un extrem de servei a un altre.
- **ASP.NET Web Forms** és un entorn d'aplicació Web per construir llocs dinàmics.

TREBALL FI DE CARRERA

CAPES DE L'APLICACIÓ

El projecte s'ha dividit en tres capes ben diferenciades per estructurar de forma robusta el codi font i per facilitar l'escalabilitat de forma ràpida i senzilla.

- **Capa de Dades:**
En aquesta capa es gestionarà l'accés als objectes de la base de dades. La tecnologia emprada és Entity Framework.
- **Capa de Negoci:**
Aquesta capa gestiona tota la lògica de negoci i es comunica tant amb la capa de presentació com amb la capa de dades. La tecnologia emprada és el servei WCF.
- **Capa de Presentació:**
Aquesta capa fa referència a la interfície gràfica d'usuari realitzada íntegrament amb AS.NET Web Forms.

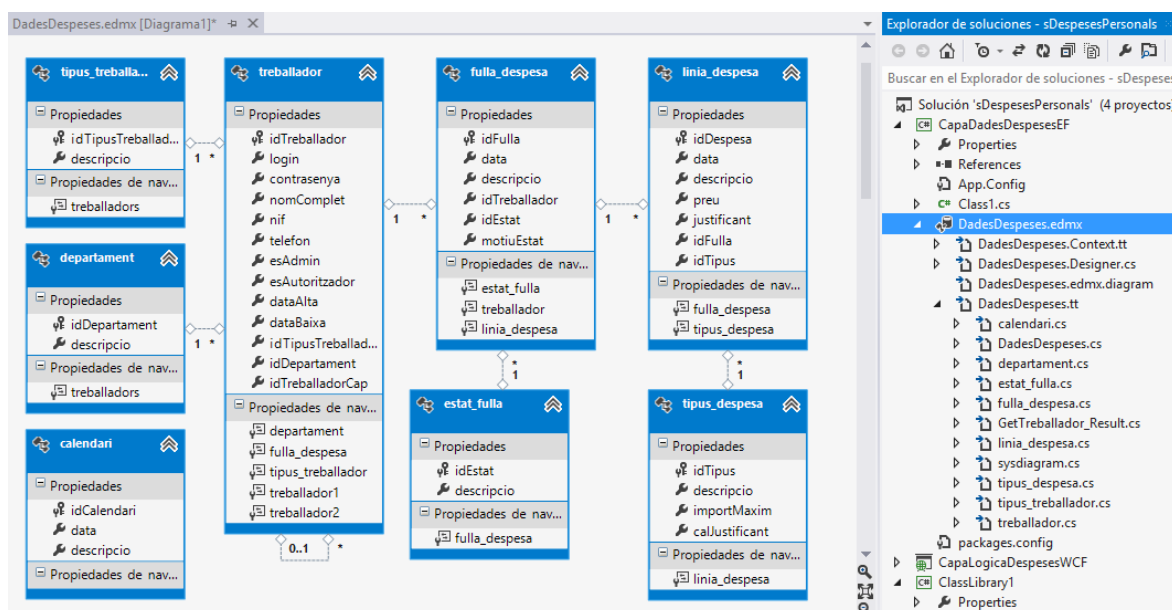
Veiem cada una de les capes de forma més detallada:

CAPA DE DADES - ENTITY FRAMEWORK

L'aplicació incorpora un projecte anomenat '**CapaDadesDespesaEF**' que com bé indica el nom s'utilitza per implementar la capa de dades fent servir la tecnologia ADO.NET Entity Framework.

S'ha afegit a l'aplicació un projecte de tipus 'Biblioteca de Classes' i en aquest projecte s'ha incorporat un nou element anomenat **ADO.NET Entity Data Model** que permet crear un model d'entitat a partir d'una base de dades. Aquesta opció permet especificar la connexió de la base de dades, la configuració del model i els objectes de dades que s'inclouran en el model.

Seguint l'assistent que ens apareix hem d'especificar la connexió a la base de dades de la qual volem crear el model conceptual i triar els objectes que volem incloure (taules, vistes, procediments emmagatzemats,..). Un cop fet això ens crearà un fitxer amb extensió. EDMX



TREBALL FI DE CARRERA

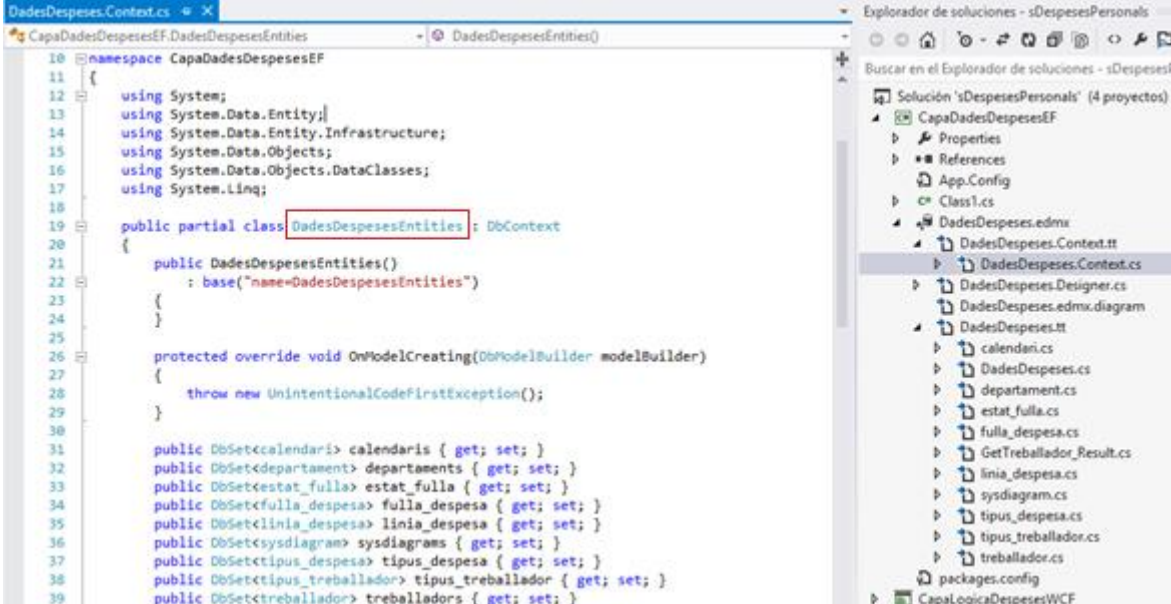
Tal i com podem veure a la imatge, l'arxiu .EDMX conté un model conceptual. Aquest fitxer defineix un conjunt de relacions entre entitats i taules.

Entity Framework usa aquest model conceptual per proporcionar una vista de les dades en objectes expressat com a tipus d'entitats i associacions.

Si despleguem el fitxer .EDMX veiem que ens ha generat un conjunt de classes que representen el model

Veiem el Fitxer **Context.cs**:

On tenim la classe **DadesDespesesEntities** que es farà servir en la capa lògica per a l'accés a les dades

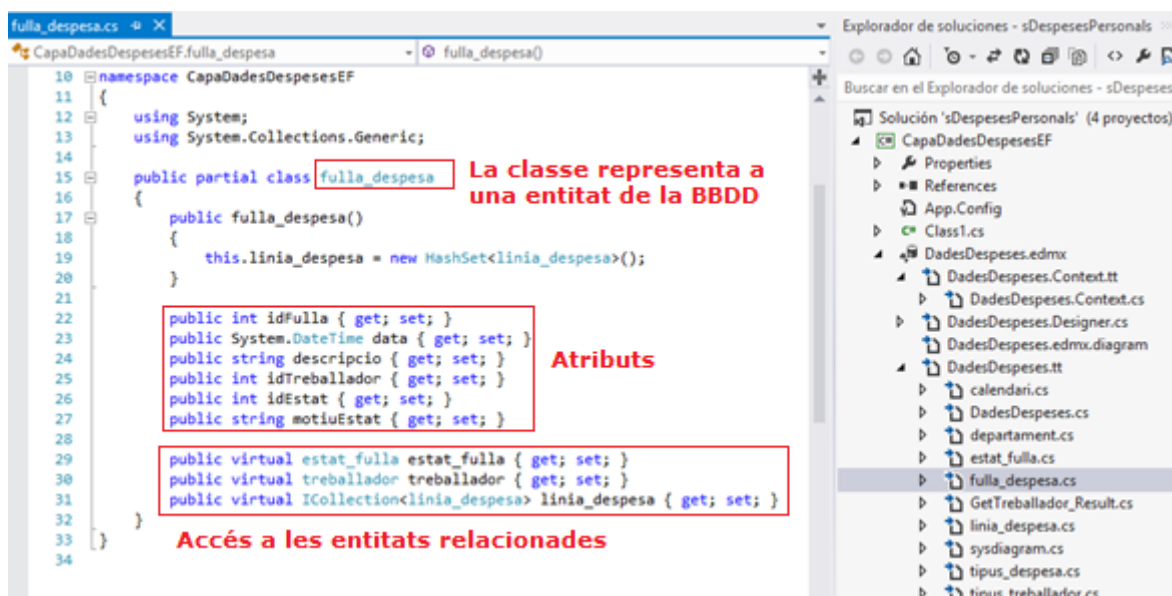


```

10 namespace CapaDadesDespesesEF
11 {
12     using System;
13     using System.Data.Entity;
14     using System.Data.Entity.Infrastructure;
15     using System.Data.Objects;
16     using System.Data.Objects.DataClasses;
17     using System.Linq;
18
19     public partial class DadesDespesesEntities : DbContext
20     {
21         public DadesDespesesEntities()
22             : base("name=DadesDespesesEntities")
23         {
24         }
25
26         protected override void OnModelCreating(DbModelBuilder modelBuilder)
27         {
28             throw new UnintentionalCodeFirstException();
29         }
30
31         public DbSet<calendari> calendaris { get; set; }
32         public DbSet<departament> departaments { get; set; }
33         public DbSet<estat_fulla> estat_fulla { get; set; }
34         public DbSet<fulla_despesa> fulla_despesa { get; set; }
35         public DbSet<linia_despesa> linia_despesa { get; set; }
36         public DbSet<sysdiagrama> sysdiagramas { get; set; }
37         public DbSet<tipus_despesa> tipus_despesa { get; set; }
38         public DbSet<tipus_treballador> tipus_treballador { get; set; }
39         public DbSet<treballador> treballadors { get; set; }

```

I per cada taula afegida ens ha creat un fitxer .cs on s'implementa una classe que representa una entitat de la base de dades i les seves relacions amb d'altres entitats.



L'avantatge és que sols ens hem de dedicar a programar les classes generades en lloc de pensar també en l'esquema d'emmagatzematge i en la forma d'obtenir accés als objectes de la base de dades per transformar-los en objectes de programació. Podem dir que l'organització de les classes generades reflexa l'organització de les taules i les relacions d'aquesta base de dades.

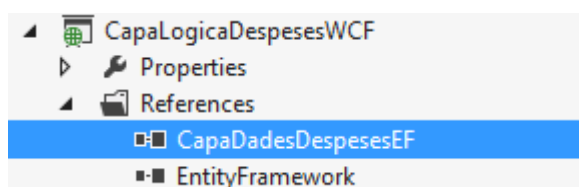
Per separar l'aplicació en tres capes, l'accés a la capa de dades es farà des de la capa de negoci com s'especifica en el punt següent:

CAPA Lògica DE NEGOCI - WCF

L'aplicació també incorpora un projecte anomenat '**CapaLogicaDespesaWCF**' que com bé indica el seu nom, s'utilitza per implementar la capa lògica de negoci fent servir tecnologia WCF com a servei Web.

Des d'aquesta capa s'accedeix a la capa de dades per consultar, afegir, editar i eliminar valors de les entitats de la base de dades fent servir, es clar, el model conceptual creat anteriorment, en concret es farà través de la classe **DadesDespesesEntities**

Per tant veiem que s'ha afegit en aquest projecte una referència al projecte anterior:



De fet aquesta capa s'utilitza com a pont entre la capa de les dades i la capa de presentació i és on s'aplica la lògica de negoci de la nostra aplicació.

TREBALL FI DE CARRERA

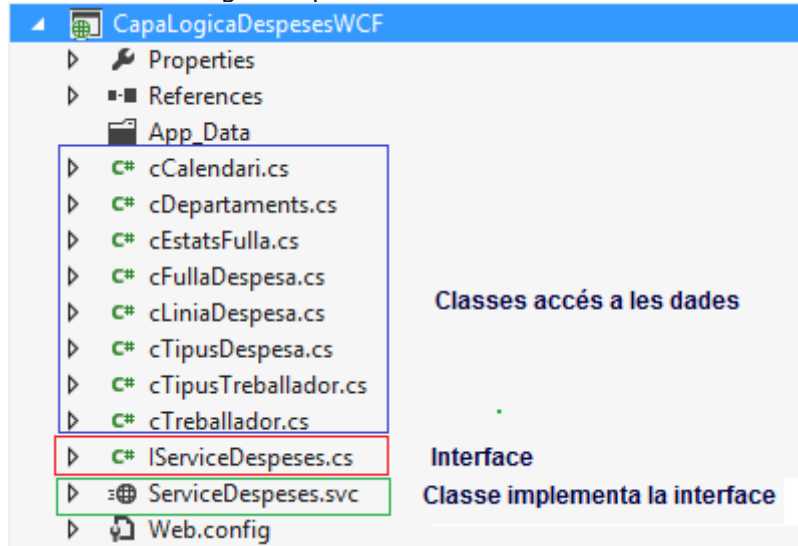
Com ja s'ha dit anteriorment, aquesta capa s'ha realitzat amb tecnologia Windows Communication Foundation.

El tipus de projecte és de tipus '**Aplicació de Serveis WCF**'. És un tipus d'aplicació que s'allotja dins el IIS com a servei, esperant sol·licituds de tots els seus clients (que en el nostre cas la capa de presentació serà el client).

El projecte es divideix en tres parts:

- Interface que defineix el contracte amb els tipus de dades i mètodes públics que seran accessibles des dels clients.
- Classes que implementen els mètodes definits a la Interface
- Classes addicionals que s'utilitzen per accedir al model de dades i que són invocades des de la classe que implementa la Interface (aquestes classes s'han creat per dividir)

Veiem una imatge d'aquesta estructura:



Detallem cada una de les parts

INTERFACE

En aquesta Interface **IServiceDespeses** hem definit el contracte, és a dir, els mètodes i tipus de dades que seran accessibles des dels clients.

```

namespace CapaLogicaDespesesWCF
{
    [ServiceContract]
    public interface IServiceDespeses
    {
        #region TREBALLADOR
        [OperationContract]
        Boolean loginTreballador(String login, String clau);

        [OperationContract]
        treballadorWCF getTreballador(String login, String clau);
    }
}

```

TREBALL FI DE CARRERA

Les etiquetes [ServiceContract] sobre la interface i [OperationContract] sobre cada mètode són importants per especificar el contracte públic que ofereix el servei.

També veiem com definir tipus de dades públics fent servir les etiquetes [DataContract] sobre el tipus classe i [DataMember] per a cada atribut:

```
[DataContract]
public class tipusDespesaWCF
{
    [DataMember]
    public int idTipus;
    [DataMember]
    public String descripcio;
    [DataMember]
    public Nullable<decimal> importMaxim;
    [DataMember]
    public Boolean calJustificant;
}
```

Sense aquestes etiquetes tots aquests mètodes i classes serien privats al servei WCF.

CLASSE QUE IMPLEMENTA LA INTERFACE

Aquesta classe anomenada **ServeiDespeses** ofereix la implementació per a cada un dels mètodes exposats en la interface anterior:

```
namespace CapaLogicaDespesesWCF
{
    public class ServiceDespeses : IServiceDespeses
    {
        #region TREBALLADOR

        public Boolean loginTreballador(String login, String clau)
        {
            try
            {
                return cTreballador.loginTreballador(login, clau);
            }
            catch (Exception e)
            {
                throw e;
            }
        }
    }
}
```

Com es pot veure a les imatges anteriors la implementació d'aquests mètodes és molt senzill ja que deleguen la feina a cada una de les classes addicionals que fan tota la feina d'accés a la capa de dades:

TREBALL FI DE CARRERA

```

▷ C# cCalendari.cs
▷ C# cDepartaments.cs
▷ C# cEstatsFulla.cs
▷ C# cFullaDespesa.cs
▷ C# cLiniaDespesa.cs
▷ C# cTipusDespesa.cs
▷ C# cTipusTreballador.cs
▷ C# cTreballador.cs

```

Veiem la seva implementació al següent apartat:

CLASSES ADDICIONALS D'ACCÉS AL MODEL DE DADES

Cada una d'aquestes classes accedeixen al model de dades tant per consultar com per actualitzar-lo.

Ho fan a través de la classe anomenada **DadesDespesesEntities** que s'ha generat en crear el model conceptual amb Entity Framework.

Per estructurar la informació de forma modular s'ha creat una classe que referencia a cada una de les entitats de la base de dades.

En aquesta petita guia, com la implementació de totes les classes és exactament igual, solament veurem una.

Primer de tot s'han d'importar els objectes de la capa de dades per, poder accedir al model conceptual i poder treballar així amb els seus objectes:

```

using CapaDadesDespesesEF;
using System.Data.Objects;
using System.Data.Objects.DataClasses;

```

Compte! Per fer més fàcil l'accés als mètodes d'aquestes classes es defineixen tots els atributs i mètodes de forma estàtica (i així s'evita crear una instància per a cada classe)

Fem una vista ràpida a la classe cTreballador:

```

public class cTreballador
{
    private static CapaDadesDespesesEF.DadesDespesesEntities entitats=new DadesDespesesEntities();

    METODES CONSULTA

    MÉTODES ACTUALITZACIÓ

    MÉTODES INICIALITZACIÓ-TANCAMENT

    UTILITATS
}

```

TREBALL FI DE CARRERA

Veiem com s'estructuren aquestes classes:

Primer de tot es crea una instància al objecte Entities de la capa de dades per poder accedir als objectes del model conceptual Entity Framework.

```
DadesDespesesEntities entitats=new DadesDespesesEntities();
```

I després es separa el codi en regions per diferenciar les diferents funcionalitats:

- Mètodes de consulta a la base de dades
- Mètodes d'actualització (alta, baixa i edició) sobre les classes que representen a les entitats de la base de dades
- Mètodes inicialització i tancament del model per refrescar les dades.
- Mètodes varis per ajudar i estructurar la resta de codi.

MÈTODES DE CONSULTA

Mètodes que accedeixen al model per realitzar consultes sobre les dades.

Es fa servir **LINQ to Entities** que proporciona capacitat per realitzar consultes integrades en llenguatge (LINQ) que permet escriure consultes contra el model conceptual de Entity Framework.

```
public static treballadorWCF getTreballador(String login, String clau)
{
    treballador t = new treballador();
    try
    {
        ActualitzarValorsBBDD();
        var Query = (from p in entitats.treballadors
                    where p.login == login && p.contrasenya == clau && !p.dataBaixa.HasValue
                    select p);
        t = Query.FirstOrDefault<treballador>();
        return ConvertTreballadorToTreballadorWCF(t);
    }
    catch (Exception e)
    {
        throw new Exception("getTreballador:" + e.Message);
    }
}
```

MÈTODES D'ACTUALITZACIÓ

Aquests mètodes es fan servir per donar d'alta, eliminar o editar registres de les entitats de la base de dades a través del model conceptual.

Veiem un **ALTA**

```
public static int afegirDepartament(departamentsWCF depa)
{
    int max = 0;
    try
    {
        departament t = new departament();
        if (!String.IsNullOrEmpty(depa.descripcion)) t.descripcion = depa.descripcion;

        entitats.departaments.Add(t);
        if (entitats.SaveChanges() > 0)
        {
            //Obtenir el ID de la nova dada introduïda
            max = (from p in entitats.departaments select p.idDepartament).Max();
        }

        return max;
    }
    catch (Exception e)
    {
        throw new Exception("afegirDepartament:" + e.Message);
    }
}
```

Veiem una **EDICIÓ**

```
public static bool editarDepartament(departamentsWCF depa)
{
    departament t;
    bool res = false;
    try
    {
        var Query = (from p in entitats.departaments where p.idDepartament == depa.idDepartament select p);
        t = Query.FirstOrDefault<departament>();
        if (!String.IsNullOrEmpty(depa.descripcion)) t.descripcion = depa.descripcion;
        entitats.SaveChanges();
        res = true;
        return res;
    }
    catch (Exception e)
    {
        throw new Exception("editarDepartament:" + e.Message);
    }
}
```

TREBALL FI DE CARRERA

Veiem una **ELIMINACIÓ**:

```
public static bool eliminarDepartament(int idDepartament)
{
    departament t;
    bool res = false;
    try
    {
        if (getCountDepartament() > 1)
        {
            var Query = (from p in entitats.departaments where p.idDepartament == idDepartament select p);
            t = Query.FirstOrDefault<departament>();
            entitats.departaments.Remove(t);
            entitats.SaveChanges();
            res = true;
        }
        return res;
    }
    catch (Exception e)
    {
        throw new Exception("eliminarDepartament:" + e.Message);
    }
}
}
```

MÈTODES INICIALITZACIÓ I TANCAMENT

Aquests mètodes tanquen i obren el model conceptual per refrescar la informació de la base de dades:

```
public static void Inicialitzar()
{
    if (entitats == null)
        entitats = new DadesDespesesEntities();
}

public static void Dispose()
{
    if (entitats != null)
    {
        entitats.Dispose();
        entitats = null;
    }
}
}
```

CAPA DE PRESENTACIÓ - ASP.NET WEB FORMS

L'aplicació incorpora, també, el projecte anomenat '**WebGestorDespeses**' que representa la capa de presentació fent servir tecnologia ASP.NET Web Forms.

Aquest projecte es comunica amb el Servei WCF per obtenir les dades de la base de dades i així poder presentar-la a l'usuari.

També valida la informació que introdueix l'usuari abans de comunicar-se amb el Servei per enviar-li la informació.

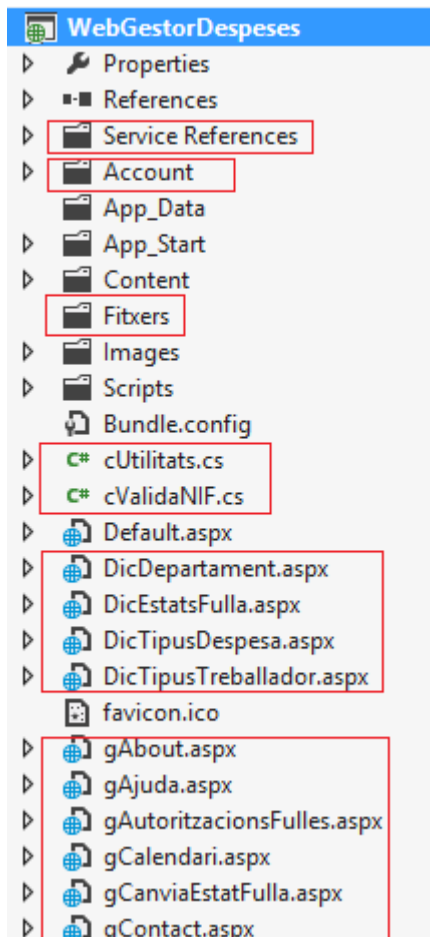
TREBALL FI DE CARRERA

El tipus de projecte creat és '**Aplicació ASP.NET Web Forms**'. Aquest projecte, per defecte, crea un esquelet d'aplicació i una estructura de carpetes per facilitar la feina al programador.

Posteriorment s'ha eliminat aquells elements de l'esquelet que no fan falta així com adaptat els elements existents a les funcionalitats pròpies de l'aplicació i també s'han afegit nous elements per implementar la nostra solució.

ESTRUCTURA DEL PROJECTE

L'estructura del projecte, bàsicament és la mateixa que l'esquelet creat automàticament pel Visual Studio.



Com ja s'ha dit, s'han afegit o eliminat elements (carpetes i fitxers) per donar solució a aquesta aplicació en concret.

Fem un repàs d'aquells elements més destacats:

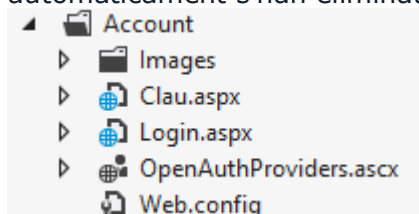
TREBALL FI DE CARRERA

CARPETA SERVICEREFERENCE

Conté una referència al Servei WCF perquè des d'aquest projecte es pugui fer crides als mètodes públics.

CARPETA ACCOUNT

Conté únicament les pàgines Login i Clau. La resta de fitxers generats automàticament s'han eliminat ja que no es fan servir per aquesta solució.



CARPETA FITXERS

S'ha creat aquesta carpeta per desar tots els fitxers que l'usuari adjunta en crear despeses. *És podrien haver desat al servidor de BBDD però s'ha considerat més adequat guardar-los aquí per la facilitat d'accés i per no sobrecarregar-lo.*

CLASSE CUTILITATS

S'ha creat aquesta classe per afegir tots aquells mètodes que són comuns a totes les pàgines i així evitar haver de repetir les mateixes funcionalitats en més d'una pàgina.

CLASSE CVALIDANIF

Aquesta classe s'utilitza per validar el NIF i d'altres documents.

S'ha baixat des del següent enllaç:

<http://alskare.wordpress.com/2009/07/21/validar-nif-nie-cif/>

PÀGINES ASPX

S'han creat tantes pàgines .aspx com la nostra solució ha necessitat.

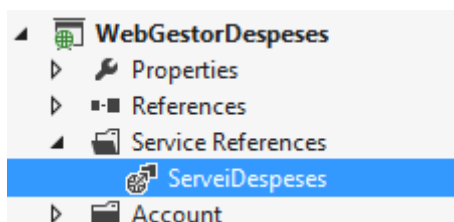
Per fer-ho similar al model de base de dades, s'ha creat una pàgina per cada entitat de la base de dades.

En algun cas s'ha creat alguna pàgina més per donar suport a la resta (és a dir mostrar informació en una nova pàgina com la vista dels fitxers adjunts, per exemple)

A les pàgines principals s'ha afegit el prefix 'g' i les pàgines que fan referència a diccionaris se'ls ha afegit el prefix 'DIC'

ACCÉS A LES DADES

Com ja s'ha explicat, per implementar l'accés a les dades es necessari fer-ho a través del Servei WCF i per això primer s'ha d'afegir una referència d'aquest servei al projecte:



TREBALL FI DE CARRERA

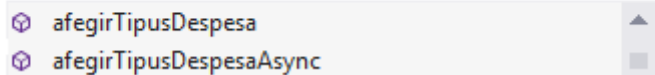
També s'ha d'importar la referència creada en aquelles pàgines on es desitgi accedir.

```
using System.Data.Objects;
using System.Data.Objects.DataClasses;
using WebGestorDespeses.ServeiDespeses;
```

I seguidament en el codi es crea una instància del client (del servei) i per poder accedir a tots els mètodes:

```
ServiceDespesesClient wcf = new ServiceDespesesClient();
```

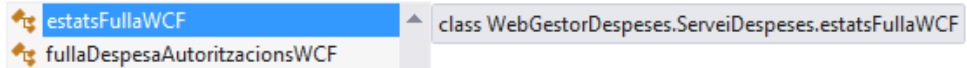
```
wcf.
```



```
afegirTipusDespesa
afegirTipusDespesaAsync
```

I tipus de dades públics:

```
ServeiDespeses.
```

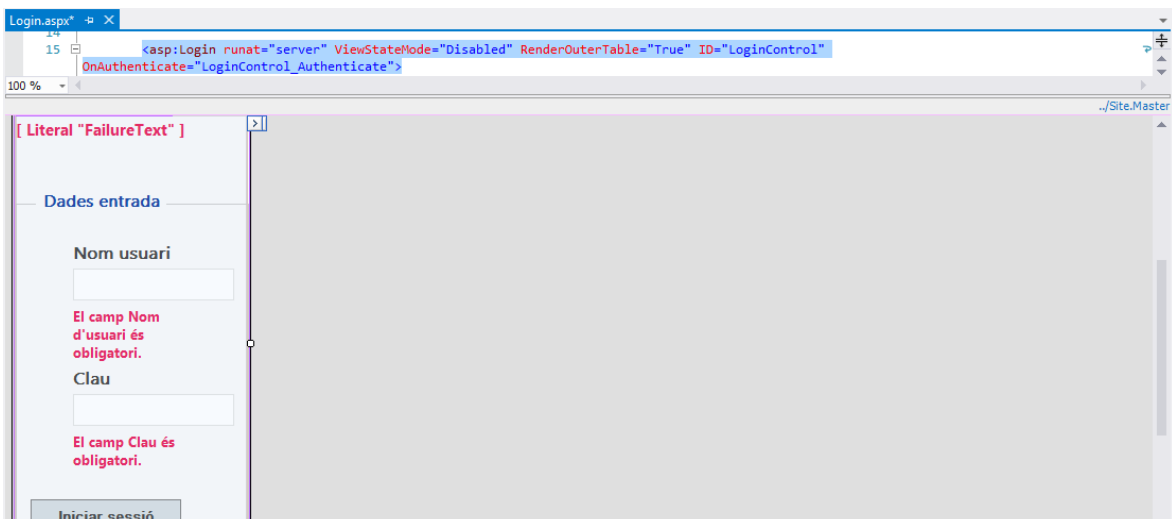


```
estatsFullaWCF
fullaDespesaAutoritzacionsWCF
```

class WebGestorDespeses.ServeiDespeses.estatsFullaWCF

PÀGINA DE LOGIN

La part Visual d'aquesta pàgina ja ve implementada a l'esquelet inicial de l'aplicació, que fa servir el control Login



Al pulsar la comanda 'Iniciar Sessió', s'ha afegit el codi específic per a la nostra solució:

Es comprova si existeix l'usuari a la base de dades que en cas positiu redirigeix a la pàgina d'inici o mostra missatge d'error.

```
protected void LoginControl_Authenticate(object sender, AuthenticateEventArgs e)
{
    try
    {
        ServiceDespesesClient wcf = new ServiceDespesesClient();

        if (wcf.LoginTreballador(LoginControl.UserName, LoginControl.Password))
        {
            //obtenir les dades del treballador loginat
            perfilTreballadorWCF perfil = wcf.getPerfilTreballador(LoginControl.UserName, LoginControl.Password);

            Session["perfil"] = perfil;
            FormsAuthentication.RedirectFromLoginPage(perfil.nomCompleu.Split(' ')[0], LoginControl.RememberMeSet);
        }
        else
        {
            LoginControl.FailureText = "No existeix aquest usuari; si us plau torna a intentar-ho";
        }
    }
    catch (Exception ex)
    {
        LoginControl.FailureText = "S'ha produït un error; torni a intentar-ho";
    }
}
```

Veiem que el codi és molt senzill si fem servir els mètodes definits a la capa de negoci (Servei WCF). *Sols hem de realitzar crides i analitzar el que retornen!!!*

ESTRUCTURACIÓ VISUAL DE LES PÀGINES

MASTER PAGE

S'ha aprofitat en aquest objecte per afegir el menú a la capçalera ja que és un element comú per a totes les pàgines web.

I des d'aquí aprofitant que tenim aquest element comú, depenent de l'usuari validat es mostra unes opcions de menú o d'altres:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (HttpContext.Current.User.Identity.IsAuthenticated)
    {
        if (Session["perfil"] != null)
        {
            this.mnuPrincipal.Visible = true;
            perfilTreballadorWCF perfil = (perfilTreballadorWCF)Session["perfil"];
            //Si es usuari normal
            if (perfil.esAdmin == false && perfil.esAutoritzador == 0)
            {
                if (mnuPrincipal.FindItem("Autoritzacions") != null)
                    mnuPrincipal.Items.Remove(mnuPrincipal.FindItem("Autoritzacions"));
            }
            //Si es autoritzador o usuari normal
            if (perfil.esAdmin == false)
            {
                if (mnuPrincipal.FindItem("Treballadors") != null)
                {
                    mnuPrincipal.Items.Remove(mnuPrincipal.FindItem("Treballadors"));
                    mnuPrincipal.Items.Remove(mnuPrincipal.FindItem("Configuració"));
                }
            }
        }
        else { TancarSessio(); }
    }
    else { this.mnuPrincipal.Visible = false; }
}
```

ESTRUCTURACIÓ CODI FONT EN LES PÀGINES

Per fer més llegible el codi font es subdivideix el codi en regions que tenen funcionalitats similars.

```
public partial class gFullaDespeses : Page
{
    perfilTreballadorWCF perfil;

    EVENT PAGINA

    FUNCIONS PROPIES

    EVENTS DELS CONTROLS

}
```

En aquest cas veiem que s'ha subdividit en tres parts:

- Events de la pròpia pàgina
- Events dels controls afegits a la pàgina
- Funcions pròpies creades amb l'objectiu de reutilitzar codi o per no recarregar massa el codi (la idea és intentar crear parts de codis que no siguin molt llargues i que no facin molt difícil la seva lectura).

CONTROLS DE VALIDACIÓ

Per a validar les dades que introdueixen els usuaris, s'han fet servir els controls de validació que inclou ASP.NET amb l'objectiu d'alliberar aquesta feina a nivell de codi i fer la lectura del codi més simple realitzant únicament aquelles validacions que els controls no poden fer.

VALIDAR QUE UN CAMP SIGUI OBLIGATORI

Control **RequiredFieldValidator** s'utilitza per a validar que l'usuari hagi introduït un valor al camp obligatoris.

```
<asp:RequiredFieldValidator ID="requireDescripcio" runat="server"
    ErrorMessage="Has d'introduir la descripció de la despesa" ControlToValidate="dtxtdescripcioDespesa"
    ForeColor="Red"></asp:RequiredFieldValidator>
```

VALIDAR L'ENTRADA D'UNA DATA

Control **CompareValidator** s'utilitza per a validar que el format de la data sigui correcte.

```
<asp:CompareValidator ID="CompareValidator2" runat="server" ErrorMessage="La data de la despesa és incorrecta"
    ControlToValidate="dtxtdata" ForeColor="Red" Type="Date" Operator="DataTypeCheck"></asp:CompareValidator>
```

VALIDAR LA ENTRADA D'UN PREU (VALOR DECIMAL)

Control **CompareValidator** s'utilitza per a validar que la dada introduïda sigui un número decimal major que el 0.

```
<asp:CompareValidator ID="CompareValidator1" runat="server" ErrorMessage="El preu és incorrecte"
    ControlToValidate="dtxtpreu" Type="Double" Operator="GreaterThan" ValueToCompare="0" ForeColor="Red"></asp:CompareValidator>
```

VALIDAR EL VALOR EL FORMAT DEL NIF

Control **RegularExpression** s'utilitza per validar el format del NIF.

```
<asp:RegularExpressionValidator ID="regularNIF" runat="server" ControlToValidate="txtNif" ForeColor="Red"
    ErrorMessage="El Format del NIF és incorrecte"
    ValidationExpression="^(X(-|\.)?0?\d{7}(-|\.)?[A-Z]|[A-Z](-|\.)?\d{7}(-|\.)?[0-9A-Z])\d{8}(-|\.)?[A-Z])$" >
</asp:RegularExpressionValidator>
</td>
```

CONTROL GRIDVIEW

Per mostrar la informació dels registres de les entitats, s'ha fet servir el control GridView ja que ofereix una sèrie de funcionalitats automàtiques molt apropiades per aquest tipus de tasca, evitant haver de fer la feina manualment (a nivell de codi).

Entre d'altres:

- Permet paginar els registres automàticament.
- Permet realitzar accions sobre cada un dels registres afegits (editar, eliminar, etc...)
- Carrega el contingut, de forma automàtica, amb el resultat d'una consulta del model de dades. El programadors sols ha de configurar quins camps vols que apareixin a la llista.

Perquè el control funcioni gairebé automàtic s'ha de configurar a nivell d'atributs:

```
<asp:GridView ID="gdGrid" runat="server" CellPadding="4" Width="100%" AutoGenerateColumns="False" DataKeyNames="idDepartament"
    ForeColor="#333333" AllowPaging="True" PageSize="4" BorderStyle="None" CaptionAlign="Top" EmptyDataText="No hi han dades
    introduïdes" OnPageIndexChanging="gdGrid_PageIndexChanging" OnRowCommand="gdGrid_RowCommand"
    OnRowDataBound="gdGrid_RowDataBound" OnRowDeleting="gdGrid_RowDeleting"
>
```

Els camps mes importants:

ID i runat: Identificador del control i 'runat' per poder accedir al mateix via codi

DataKeyNames: Indica quin es el camp clau dels registres que es volen carregar

AllowPaging i PageSize: Per indicar si es desitja que el control realitzi paginació i la quantitats d'elements a visualitzar.

EmptyDataText: Text que mostra si no hi ha informació que mostrar.

Configuració de les columnes que mostren els camps dels registres:

```
<Columns>
<asp:BoundField DataField="idDepartament" ReadOnly="true" HeaderText="ID" HeaderStyle-HorizontalAlign="Left" />
<asp:BoundField DataField="descripcio" ReadOnly="true" HeaderText="Descripció" />
<asp:TemplateField HeaderText="Details" InsertVisible="False" ShowHeader="False">
<ItemTemplate>
<asp:ImageButton ID="imageDetails" runat="server" CausesValidation="False" ImageUrl="~/Images/lupa.jpg"
    CommandArgument="<%# Eval("idDepartament") %%" CommandName="Details" AlternateText="Details" />
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Eliminar" InsertVisible="False" ShowHeader="False">
<ItemTemplate>
<asp:ImageButton ID="imageEliminar" runat="server" CausesValidation="False" ImageUrl="~/Images/eliminar.jpg"
    CommandArgument="<%# Eval("idDepartament") %%" CommandName="Delete" AlternateText="Eliminar" />
</ItemTemplate>
</asp:TemplateField>
</Columns>
```

L'atribut **DataField** del control **BoundField** és per especificar el nom del camp de l'entitat que és vol mostrar quan es carreguen les dades d'una columna.

TemplateField s'utilitza, en el nostre cas, per afegir dues comandes (veure detalls i eliminar) per a cada registre carregat al control. Quan l'usuari fa clic sobre una comanda (en el nostre cas hem definit una imatge) es crida a l'event `ROW_COMMAND` i s'envia com a paràmetre el camp definit a l'atribut **CommandArgument**

CARREGA DE DADES AL CONTROL

```
private void CarregarGrid()
{
    try
    {
        ServiceDespesesClient wcf = new ServiceDespesesClient();
        this.gdGrid.DataSource = wcf.getDepartaments();
        this.gdGrid.DataBind();
    }
    catch { lblComentaris.Text = "Ha succeït un error en carregar les dades.Consulti amb l'administrador!"; }
}
```

Veiem que aquesta tasca es molt senzilla sols s'ha d'assignar a l'atribut **DataSource** a un mètode públic del Servei WCF. El mètode ha de retornar una llista de registres que amb la crida a **DataBind** es vinculen automàticament al control.

PAGINAR

Si s'ha configurat l'atribut **AllowPaging=true** del control, perquè es pagini, es mostra a peu de pàgina números per canviar de vista als següents o anteriors registres:

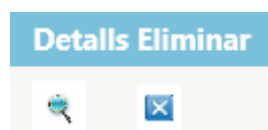


I en el moment que l'usuari faci clic sobre un número es dispara el següent event a on s'ha afegit codi per recarregar les dades al control.

```
protected void gdGrid_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    GridView gv = (GridView)sender;
    gv.PageIndex = e.NewPageIndex;
    CarregarGrid();
}
```

COMANDES 'ELIMINAR' I 'VEURE DETALLS'

Els registres del control poden tenir associats aquestes comandes:



Són objectes de tipus **TemplateField** que ja s'ha explicat anteriorment:

```
<asp:TemplateField HeaderText="Detalls" InsertVisible="False" ShowHeader="False">
  <ItemTemplate>
    <asp:ImageButton ID="imageDetalls" runat="server" CausesValidation="False" ImageUrl="~/Images/lupa.jpg"
      CommandArgument='<%# Eval("idDepartament") %>' CommandName="Detalls" AlternateText="Detalls" />
    </ItemTemplate>
  </asp:TemplateField>
```

En realitzar clic a sobre de la imatge s'executa l'event ROW_COMMAND que bàsicament realitza la següent funcionalitat (cada pàgina, però, pot realitzar feina específica segons funcionalitat establerta)

```
protected void gdGrid_RowCommand(object sender, GridViewCommandEventArgs e)
{
    int idDepartament = Convert.ToInt32(e.CommandArgument);

    if (e.CommandName == "Delete") //Si es vol eliminar
    {
        ServiceDespesesClient wcf = new ServiceDespesesClient();
        wcf.eliminarDepartament(idDepartament);
        CarregarGrid();
    }
    else if (e.CommandName == "Detalls") //Clic en Detalls
    {
        String redi= Request.Url.AbsolutePath + "?idDepartament=" + idDepartament ;
        Response.Redirect(redi);
    }
}
```

(Info) Depenent de la pàgina i complexitat pot haver validacions prèvies abans d'executar cada comanda.

COMANDA DESAR LES DADES

Dades Despesa

Tipus

Descripció (*)

Data (*)

Preu (*)

Fitxer Adjunt (**)

(Info)Per aquest tipus de despesa és obligatori adjuntar un justificant

(**)Serà obligatori segons el tipus de despesa escollida. (Info) Extensions vàlides: JPG, PNG,JPEG i GIF

(*) És obligatori introduir un valor

TREBALL FI DE CARRERA

Independent si és un alta o una edició de dades quan es realitza clic sobre la comanda desar es comprova que tots els camps introduïts siguin correctes (cada pantalla té les seves particularitats) i en aquest cas depenen si es una alta o edició es crida als mètodes públics del Servei WCF enviant-li les dades perquè les desi. Veiem una versió senzilla (sense comprovacions) per fer-nos una idea del funcionament:

```
protected void cmdDesar_Command(object sender, CommandEventArgs e)
{
    bool correcte = true;
    ServiceDespesesClient wcf = new ServiceDespesesClient();
    if (!Page.IsValid) lblError.Text = lblError.Text + "No es poden desar les dades. Revisi que hagi i
    else //Realitzar Validacions varies
    { if (correcte)
      { //omplir les dades amb el contingut del controls
        departamentsWCF depa = new departamentsWCF(); Assignar la informació a desar
        depa.descripcion = this.txtDescripcion.Text;
        if (string.IsNullOrEmpty(this.lblID.Text)) //Si és una alta
        { wcf.afegirDepartament(depa);
          //amagar-mostrar controls
          this.fieldAlta.Visible = false;
          this.cmdAddData.Visible = true;
          this.fieldLlistat.Visible = true;
        }
      }
      else //edicio
      {
        depa.idDepartament = int.Parse(this.lblID.Text); //assignar ID del registre a modificar
        wcf.editarDepartament(depa);
        //amagar-mostrar i habilitar controls
        HabilitarControls(false);
        this.cmdDesar.Visible = false;
        this.cmdEditar.Visible = true;
      }
      this.lblError.Text = "Les dades han estat desades correctament.";
      CarregarGrid();
    }
}
```

TRACTAMENT D'EXCEPCIONS

En tots els projectes s'ha tingut en compte els possibles errors que es poden ocasionar i per aquest motiu s'han tractat totes les excepcions per donar a l'usuari un missatge personalitzat en cas de succeir-se.

CAPA LÒGICA DE NEGOCI (SERVEI WCF)

Tots els mètodes afegeixen el try i catch per si succeeix algun error en accedir a les dades i poder retornar a l'usuari el missatge amb l'error a més d'indicar en quin mètode ha succeït.


```
public static List<estatsFullaWCF> getEstats()
{
    try
    {
        ActualitzarValorsBBDD();
        var Query = (from p in entitats.estat_fulla
                    select new estatsFullaWCF
                    {
                        idEstat=p.idEstat,
                        descripcio=p.descripcio
                    }
                    );
        return Query.ToList<estatsFullaWCF>();
    }
    catch (Exception e)
    {
        throw new Exception("getEstats:" + e.Message);
    }
}
```

CAPA DE PRESENTACIÓ

En aquells llocs que són sensibles a poder rebre una excepció, com quan s'accedeix al Servei WCF, es tracta el codi amb try i catch per detectar les excepcions que es puguin ocasionar. Des d'aquí no es mostra a l'usuari final el missatge d'error (ja que no es entenedor per ell) sinó que es mostra un missatge personalitzat perquè pugui derivar-lo a l'administrador (o programador perquè el solucioni).

5. Avaluació de costos

RECURSOS HUMANS

Existeixen tres rols ben definits per a la realització d'aquest projecte, a grans trets veiem les tasques assignades:

CAP DE PROJECTE

Realitza el pla de projecte, tasques de coordinació de l'equip i vetlla perquè el projecte finalitzi a la data convinguda amb una qualitat òptima i compleixi les expectatives del client.

ANALISTA

Realitza tasques de comunicació amb el client per a desenvolupar l'anàlisi, disseny i documentació del projecte.

PROGRAMADOR

Persona que implementa la solució en tecnologia .NET i testeja l'aplicació final.

VALORACIÓ ECONÒMICA ESTIMADA:

Perfil	Concepte	Preu hora	Jornades	Hores	Import	
Cap projecte	Planificació projecte	50€	10	20	1.000	
Cap Projecte	Seguiment projecte	50€	10	20	1.000	
Analista	Anàlisis i Disseny	35€	17	68	2.380	
Analista	Documentació	35€	20	80	2.800	
Programador	Implementació	25€	26	208	5.200	
Programador	Testeig	25€	13	104	2.600	
					12.180 €	Sense IVA

6. Treball futur**FÓRA DE L'ABAST**

En aquesta versió no es contempla les següents funcionalitats:

- Gestió mòdul de nòmines
- Gestió de l'administrador sobre les fulles i despeses de la resta de treballadors.
- Automatitzar el procés d'afegir els caps de setmana com a dies festius dins la funcionalitat del calendari.
- Recuperació de la contrasenya via e-mail.
- Imprimir dades.
- Mostrar informes o llistats segons diferents paràmetres
- Importar o exportar dades.
- Realització de còpies de seguretat de les dades.
- Possibilitat d'enviament d'e-mail als treballadors.
- Contemplar diferents tipus d'identificació com DNI o NIE a part del NIF
- Enviar un Mail a l'usuari quan s'autoritza o refusa una fulla.

7. Conclusions

Amb aquest projecte s'ha desenvolupat una aplicació de dificultat mitjana. El desenvolupament consta d'una sèrie de fites prèvies abans de la implementació de la solució ja que es una forma correcta de desenvolupar qualsevol software. Aquestes fites prèvies són la realització del pla de projecte, anàlisis, disseny i prototipus UI. Els coneixements pel desenvolupament d'aquestes fites s'han adquirit a d'altres assignatures de la carrera ETIG. Durant la implementació del software s'han adquirit competències de la plataforma .NET de Microsoft. I finalment s'ha realitzat la memòria, presentació i debat per a la defensa del projecte.

8. Bibliografia

- [1] Pla docent
- [2] Pla de projectes d'exemple aportats pel consultor.
- [3] Mòduls assignatura "Enginyeria del Software" de la carrera ETIG.
- [4] Mòduls assignatura "TDP" de la carrera ETIG.
- [5] Material del curs "Planificació de projectes informàtics" realitzat a la UOC.
- [6] Mòduls assignatura "Informàtica aplicada a la Gestió" de la carrera ETIG.
- [7] Mòduls assignatura "BBDD I" de la carrera ETIG.
- [8] Mòduls assignatura "BBDD II" de la carrera ETIG.
- [9] Anàlisis i disseny d'exemple aportats pel consultor.
- [10] Projectes d'anys anteriors de la biblioteca pública de la UOC.

ENLLAÇOS CONSULTATS

ASP.NET

<http://www.asp.net/>

<http://es.wikipedia.org/wiki/ASP.NET>

<http://www.maestrosdelweb.com/editorial/tutoria-desarrolloweb-asp-net/>

<http://msdn.microsoft.com/es-es/library/ywdtth2f.aspx>

ASP.NET Web Forms

<http://www.asp.net/web-forms>

Arquitectura de tres capas

<http://msdn.microsoft.com/es-es/asp.net/aa336583.aspx>

ADO.NET

<http://msdn.microsoft.com/es-es/library/e80y5yhx.aspx>

Entity Framework

<http://msdn.microsoft.com/es-es/library/bb399572.aspx>

<http://www.c-sharpcorner.com/uploadfile/ankithakur/getting-started-with-ado-net-entity-framework-in-net-3-5/>

Windows Communication Foundation

<http://msdn.microsoft.com/en-us/library/dd456779.aspx>

[http://msdn.microsoft.com/en-us/library/vstudio/ms735119\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/ms735119(v=vs.90).aspx)

<http://www.codeproject.com/Articles/537161/Implementing-a-WCF-Service-with-Entity-Framework-v>

<http://www.codeproject.com/Articles/162746/Concurrency-Control-of-WCF-Service-with-Entity-Fra>