

# **Memòria del projecte**

## **Gestor de projectes de desenvolupament de programari**

A la meva dona, Carme. Tot es mèrit seu.

## 1 Resum del projecte.

Es planteja la realització d'una petita aplicació de gestió de projectes informàtics, per a una empresa de desenvolupament de software, que actualment no disposa d'eines per controlar la informació relativa a les assignacions i documentació dels projectes.

Una empresa de desenvolupament de software desitja mantenir una petita gestió de la documentació dels seus projectes; així com de l'assignació del personal a cada projecte. Actualment la empresa no porta cap control sobre la documentació ni sobre l'assignació de personal, produint-se diversos problemes, entre els que es poden destacar:

- Desequilibris en el pes assignat als diferents recursos de la empresa.
- Descontrol en quant a la ubicació, localització i manteniment de la documentació dels diferents projectes.
- Problemes d'atenció a incidències i manteniment dels diferents projectes un cop implantats.

La empresa, a més, planteja la necessitat, per motius de seguretat, de que l'accés a les dades no el faci directament l'aplicació, i planteja la possibilitat de crear un sistema que ocultí la situació física de la base de dades a l'aplicació de gestió, així com la seva morfologia.

Per aconseguir aquest propòsit es proposa la creació d'una aplicació de formularis per donar solució als problemes de gestió descrits. A més, es planteja la creació d'un servei web que faci d'intermediari entre la base de dades i l'aplicació descrita, de forma que totes les peticions passin per aquest servei web.

Per últim, la intenció es la realització d'un treball en que es conjuguin molts dels coneixements adquirits en les diferents assignatures de l'ETIG; per tal de resoldre un problema de caire informàtic que podria ser exportable a la realitat.

Tanmateix, també representa una oportunitat de prendre contacte amb un nou llenguatge de programació, (C#), i algunes de les tecnologies que incorpora, (ADO, serveis web, winforms, etc).

Les seccions més rellevants d'aquest document són les corresponents a l'anàlisi funcional i disseny orgànic. D'aquests podem destacar els capítols següents:

De l'anàlisi funcional podem destacar les seccions [Descripció](#), [Cas d'ús general i tipologia](#) i [Diagrama general de col·laboració](#); on podem trobar la descripció de les funcionalitats i comportament general de l'aplicació.

De l'anàlisi orgànic podem destacar les seccions:

- A [Descripció del disseny de l'arquitectura](#), podem trobar descripció del disseny estructural que s'ha adoptat per aquest projecte
- A [Decisions respecte al disseny de l'arquitectura](#), es pot trobar una relació de les decisions sobre disseny orgànic més importants.
- A [Canvis sobre el disseny orgànic durant la implementació](#), podem trobar la relació dels principals canvis sobre el disseny orgànic respecte a l'anàlisi orgànic plantejat inicialment. I de com la implementació ha de saber adaptar-se als canvis de necessitats, i aplicar el redisseny de forma constant durant totes les fases de construcció del programari.

**Àrea de TFC:** Tecnologies .NET

**Paraules claus:** Gestió de projectes, .NET, C#, serveis web, Orientació objectes

## Índex

<b>1</b>	<b>RESUM DEL PROJECTE.....</b>	<b>3</b>
<b>2</b>	<b>INTRODUCCIÓ.....</b>	<b>7</b>
2.1	Objectiu.....	7
2.2	Antecedents.....	7
2.3	Descripció de les funcionalitats.....	7
2.3.1	Gestió de col·laboradors.....	9
2.3.2	Gestió de responsabilitats.....	9
2.3.3	Gestió de documents.....	9
2.3.4	Gestió de projectes.....	9
2.3.5	Emissió d'informes.....	10
2.4	Planificació temporal.....	10
2.5	Documentació del projecte.....	11
<b>3</b>	<b>ANÀLISI FUNCIONAL.....</b>	<b>12</b>
3.1	Descripció.....	12
3.1.1	Descripció de la part client.....	12
3.1.2	Descripció de la part servidor.....	14
3.2	Requeriments funcionals.....	15
3.3	Cas d'us general i tipologia.....	16
3.4	Diagrama general de col·laboració.....	17
3.5	Requeriments tècnics.....	17
3.5.1	Llenguatge i entorn de desenvolupament.....	17
3.5.2	SGBD:.....	18
3.6	Questions de seguretat.....	18
<b>4</b>	<b>DISSENY ORGÀNIC.....</b>	<b>18</b>
4.1	Descripció del disseny de l'arquitectura.....	18
4.1.1	Arquitectura externa.....	18
4.1.2	Arquitectura interna.....	19
4.2	Aprofitament de components de tercers i propis.....	22
4.3	Decisions respecte al disseny de l'arquitectura.....	22
4.3.1	Mes sobre l'arquitectura interna.....	22
4.3.2	Herència d'implementació.....	23
4.3.3	Herència d'interfaces.....	23
4.3.4	Sobre la base de dades.....	23
4.4	Disseny de classes.....	24
4.4.1	Diagrames de paquets, arquitectura i comunicació.....	24
4.4.2	Diagrames de disseny de classes.....	26
4.5	Disseny de la base de dades.....	30
4.5.1	Diagrama ER.....	30
4.5.2	Disseny lògic.....	31
4.6	Canvis sobre el disseny orgànic durant la implementació.....	31
4.6.1	Disseny d'interfaces.....	31
4.6.2	Entitats.....	32
4.6.3	Gestors dels formularis.....	33
4.6.4	Gestors del domini: Gestor d'informes.....	33
4.6.5	Classes d'excepció.....	34

<b>5</b>	<b>MANUAL DE L'USUARI</b> .....	<b>35</b>
5.1	Autenticació .....	35
5.2	Gestions .....	36
5.2.1	Gestió de descripcions de documents .....	37
5.2.2	Gestió de responsabilitats .....	38
5.2.3	Gestió de col·laboradors .....	38
5.2.4	Gestió de projectes .....	39
5.3	Informes .....	40
5.3.1	Fitxa de projecte .....	40
5.3.2	Càrregues per col·laborador .....	41
5.4	Altres opcions .....	41
5.4.1	A propòsit de .....	41
5.4.2	Ajuda .....	41
5.4.3	Sortir .....	42
<b>6</b>	<b>VALORACIÓ ECONÒMICA</b> .....	<b>42</b>
<b>7</b>	<b>CONCLUSIONS</b> .....	<b>43</b>
<b>8</b>	<b>POSSIBLES AMPLIACIONS I MILLORES</b> .....	<b>44</b>
8.1	Ampliacions de la part servidora, (SW_TFC) .....	44
8.2	Ampliacions de la part client, (GestioProjectes) .....	44
<b>9</b>	<b>GLOSSARI</b> .....	<b>45</b>
<b>10</b>	<b>BIBLIOGRAFIA</b> .....	<b>46</b>
10.1	Sobre productes .....	46
10.2	Sobre llenguatges i programació .....	46
10.3	Exemples de programari de gestió de projectes .....	47

## Índex d'il·lustracions

IL·LUSTRACIÓ 1: DIAGRAMA DE NEGOCI	8
IL·LUSTRACIÓ 2: CAS D'ÚS GENERAL	8
IL·LUSTRACIÓ 3: ESQUEMA DE PANTALLES	9
IL·LUSTRACIÓ 4: PLANIFICACIÓ DEL PROJECTE	11
IL·LUSTRACIÓ 5: VISTA GENERAL DE L'APLICACIÓ	12
IL·LUSTRACIÓ 6: VISTA DEL MENÚ DE L'APLICACIÓ	13
IL·LUSTRACIÓ 7: ÀREA DE TREBALL	14
IL·LUSTRACIÓ 8: CAS GENERAL DE COL·LABORACIÓ	17
IL·LUSTRACIÓ 9: ESQUEMA DE L'ARQUITECTURA D'IMPLEMENTACIÓ	19
IL·LUSTRACIÓ 10: MODEL VISTA CONTROLADOR	20
IL·LUSTRACIÓ 11: DIAGRAMA GENERAL D'INTERACCIÓ ENTRE CAPES	21
IL·LUSTRACIÓ 12: VISTA GENERAL DE RELACIÓ ENTRE CAPES	24
IL·LUSTRACIÓ 13: DIAGRAMA DE PAQUETS	25
IL·LUSTRACIÓ 14: CLASSES DEL SERVEI WEB	25
IL·LUSTRACIÓ 15: CLASSES D'ENTITAT	26
IL·LUSTRACIÓ 16: GESTORS DELS FORMULARIS	27
IL·LUSTRACIÓ 17: GESTORS DEL DOMINI	28
IL·LUSTRACIÓ 18: CLASSES D'EXCEPCIÓ	29
IL·LUSTRACIÓ 19: CLASSES DE PRESENTACIÓ	29
IL·LUSTRACIÓ 20: INTERFACES DE L'APLICACIÓ	30
IL·LUSTRACIÓ 21: DIAGRAMA E/R DE LA BASE DE DADES	30
IL·LUSTRACIÓ 22: DISSENY LÒGIC DE LA BASE DE DADES	31

## Índex de taules

TAULA 1: VALORACIÓ ECONÒMICA	43
------------------------------	----

## 2 Introducció

### 2.1 Objectiu.

L'objectiu concret es la realització d'una petita aplicació de gestió de projectes informàtics, per a una empresa de desenvolupament de software que actualment no disposa d'eines per controlar la informació relativa a les assignacions i documentació dels projectes.

### 2.2 Antecedents.

Una empresa de desenvolupament de software desitja mantenir una petita gestió de la documentació dels seus projectes; així com de l'assignació del personal a cada projecte. Actualment la empresa no porta cap control sobre la documentació ni sobre l'assignació de personal, produint-se diversos problemes, entre els que es poden destacar:

- Desequilibris en el pes assignat als diferents recursos de la empresa.
- Descontrol en quant a la ubicació, localització i manteniment de la documentació dels diferents projectes.
- Problemes d'atenció a incidències i manteniment dels diferents projectes un cop implantats.

La empresa, en la seva voluntat de constant millora, vol resoldre aquests problemes, (i d'altres no explicitats). I per això a encarregat el desenvolupament d'un programari que els doni solució.

### 2.3 Descripció de les funcionalitats.

L'aplicació ha de resoldre principalment dos aspectes:

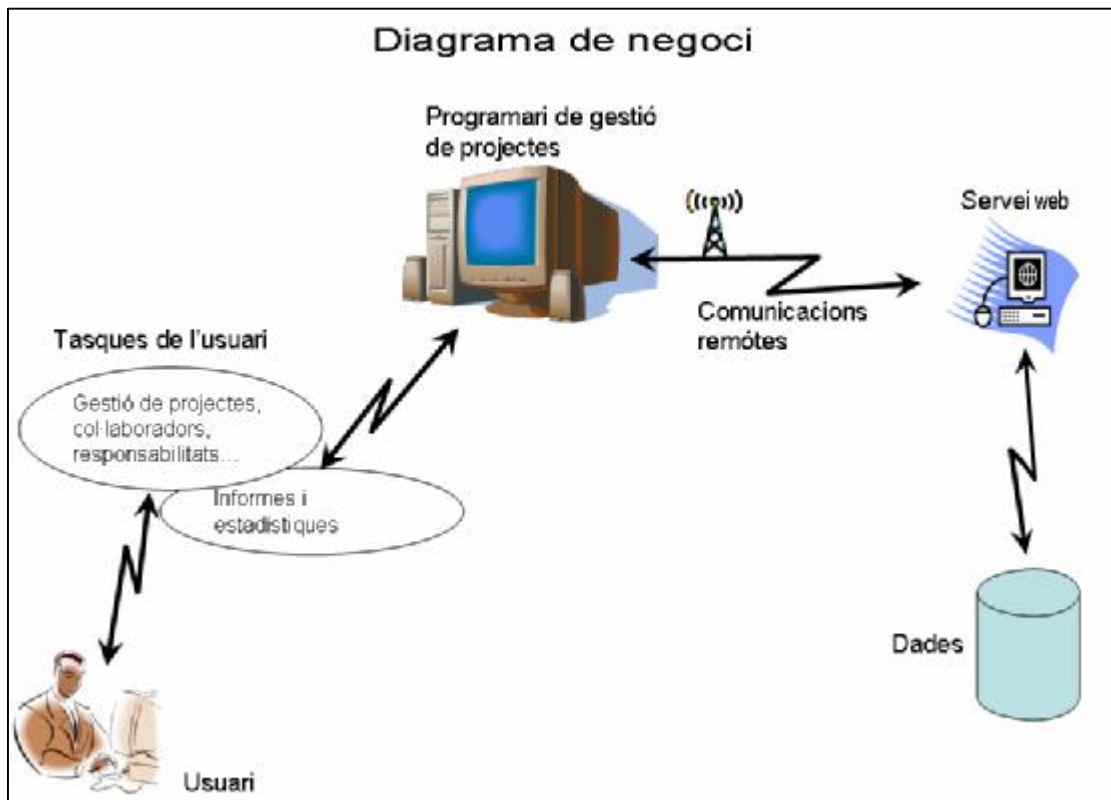
**1. Ha de reflectir l'assignació de responsabilitats de cada projecte.**

Ha de mostrar qui es el responsable de cada projecte. I quines persones han col·laborat en cada projecte, així com la relació, (resumida), de tasques que han fet.

**2. Ha de resumir i facilitar l'accés a tota la documentació generada per cada projecte.**

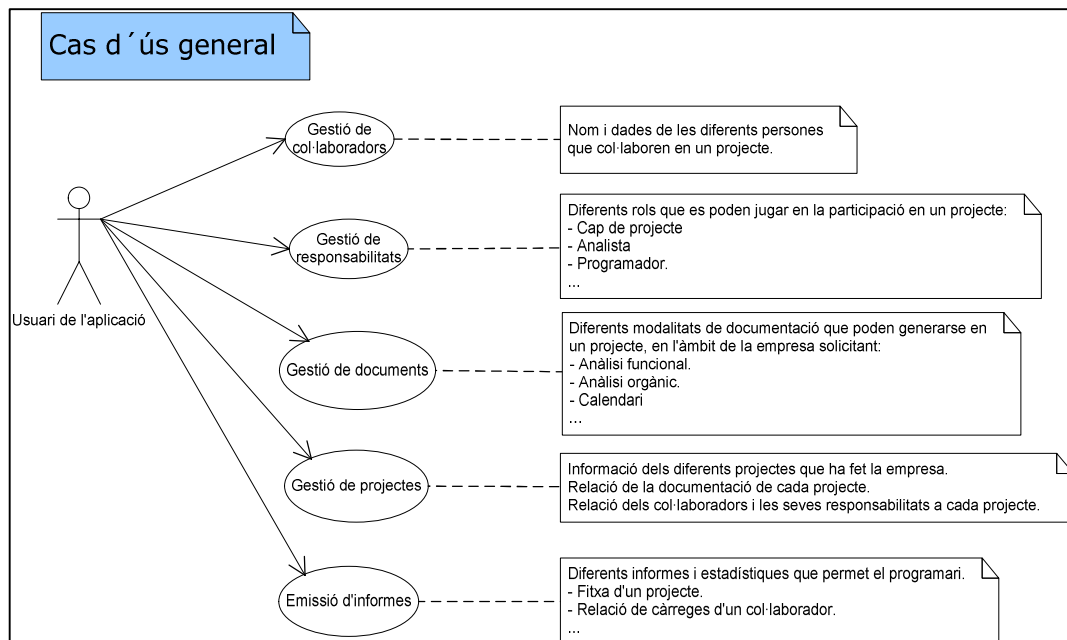
Cada projecte genera documentació diversa, (anàlisi funcional, anàlisi orgànic, calendaris, etc). El format d'aquesta documentació es també divers, (word, visio, project, etc). I cada document d'un projecte pot estar localitzat en ubicacions diverses. Es requereix, (mitjançant el programari que es proposa), que es pugui gestionar una llista dels diferents documents de que es compon cada projecte, així com de la seva ubicació. També molt important, pugui identificar-se el seu contingut sense necessitat d'obrir el document, així com un petit resum orientatiu, (si escau).

La composició del programari resultant queda reflectit en el següent gràfic de negoci:



**Il·lustració 1: Diagrama de negoci**

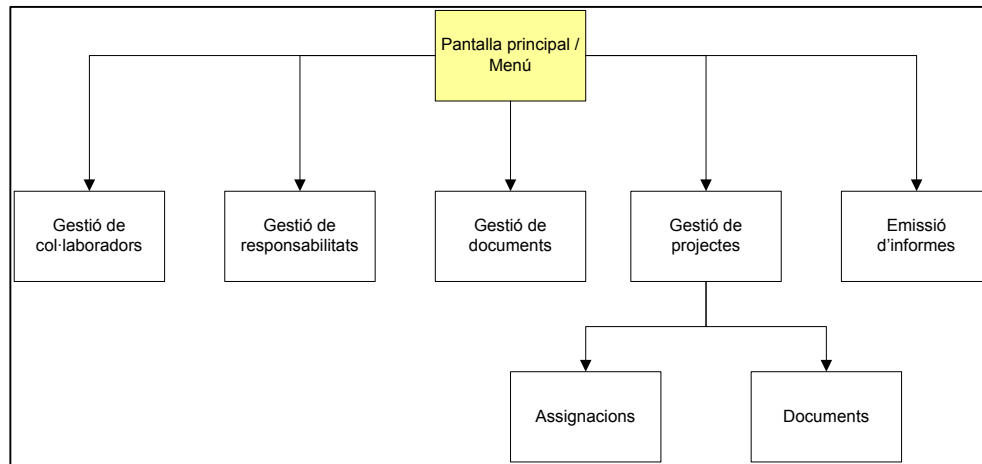
Les accions principals que pot realitzar un usuari de l'aplicació queda reflectit en el següent cas d'ús, (no detallat):



**Il·lustració 2: Cas d'ús general**



L'aplicació es realitzarà basant-se en formularis windows. I contindrà el següent esquema de pantalles:



**Il·lustració 3: Esquema de pantalles**

L'aplicació disposa de 5 mòduls principals:

### **2.3.1 Gestió de col·laboradors.**

L'usuari de l'aplicació podrà afegir, editar, esborrar, etc, dades de col·laboradors. Entre d'altres, hi hauran les següents dades:

- Nom i cognoms.
- Responsabilitat principal, (existent a la gestió de responsabilitats).
- Data d'alta del col·laborador.

### **2.3.2 Gestió de responsabilitats.**

En aquesta secció existirà la descripció de tots els rols que un col·laborador pot tenir en el decurs d'un projecte. Per exemple: analista, programador, administrador, etc. Així com el cost/hora estimat per a cada responsabilitat.

### **2.3.3 Gestió de documents.**

En aquesta secció l'usuari gestionarà l'índex de les diferents modalitats de documents que es poden generar en un projecte. Per exemple: anàlisi d'impacte, anàlisi orgànic, calendari, etc.

### **2.3.4 Gestió de projectes.**

En aquesta secció l'usuari de l'aplicació gestionarà els diferents projectes de la empresa, amb els següents subapartats:

#### **- Entrada d'assignacions.**

L'usuari entrarà els diferents col·laboradors del projecte, amb les següents dades:

- El rol que juguen en el projecte en curs. Un valor existent a la gestió de responsabilitats.
- El còmput d'hores dedicades al projecte en curs.

Si un col·laborador està relacionat amb un mateix projecte amb mes d'un rol, s'entrarà al sistema tantes vegades com rols tingui.

- **Entrada de documentació.**

L'usuari entrarà els diferents documents de que es compon el projecte. Indicant la seva localització, el seu contingut, (corresponent a un valor existent en la gestió de l'índex de documents), i un petit resum del mateix.

### **2.3.5 Emissió d'informes.**

En la emissió d'informes, l'usuari de l'aplicació podrà sol·licitar diversos llistats i estadístiques, entre les que cal destacar:

- **Fitxa de projecte.**

Imprimeix tota la informació d'un projecte determinat.

- **Relació de càrregues d'un col·laborador.**

Imprimeix les actuacions d'un col·laborador en tots els projectes. Així com el còmput d'hores dedicades.

## **2.4 Planificació temporal.**

La planificació ve determinada per una sèrie de dates claus, corresponents a les dates de lliurament de PACs i d'altres fites.

### **Lliurament del pla de treball, (13 de març).**

Definició del projecte, requisits tècnics i calendari.

### **Anàlisi funcional i primera versió del prototip, (3 d'abril).**

Anàlisi funcional, (basat en casos d'us), i primera versió del prototip, (disseny de pantalles).

### **Disseny orgànic i prototip definitiu, (18 d'abril).**

Disseny orgànic, (diagrama de classes, estructura de la BBDD, etc), i versió definitiva del prototip.

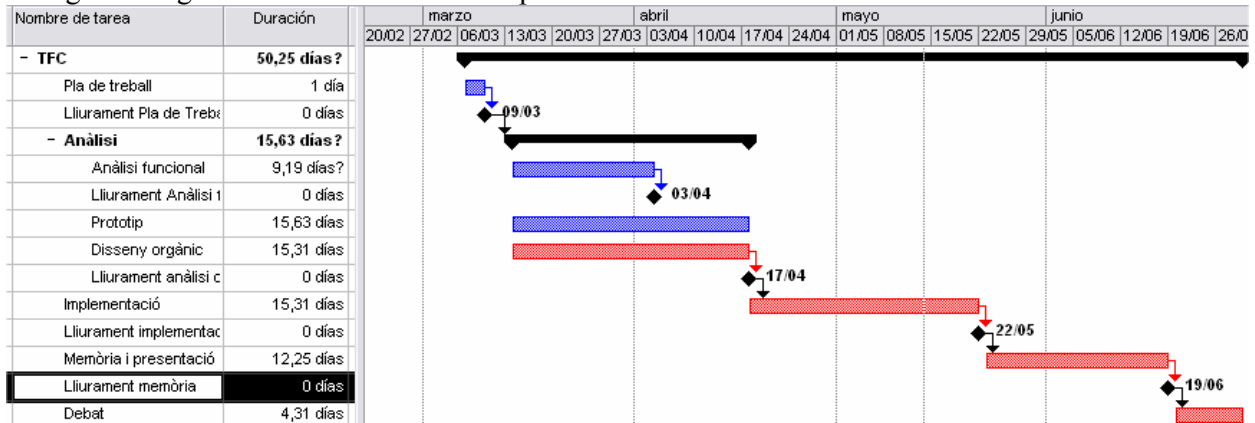
### **Implementació, (22 maig).**

Implementació en base als anàlisis anteriors. Entrega de una primera versió de la implementació.

### **Memòria i presentació virtual, (19 juny).**

Document definitiu d'avaluació i memòria. Entrega definitiva de la implementació. Entrega d'una presentació en Power Point. Inici del debat d'avaluació.

El següent diagrama de Gantt reflecteix la planificació establerta:



Il·lustració 4: Planificació del projecte

## 2.5 Documentació del projecte.

Al llarg del projecte s'han generat els documents següents:

- **Pla de treball**  
Descripció del projecte i planificació temporal.
- **Anàlisi funcional**  
Anàlisi de requeriments i especificació de les funcionalitats basades en casos d'ús.
- **Prototip**  
Implementació no funcional de l'aplicació.
- **Anàlisi orgànic**  
Disseny de l'arquitectura i de les classes de l'aplicació. Diagrames estàtics UML de classes i disseny lògic de la base de dades.
- **Implementació.**  
Fonts de les aplicacions de que es compon el projecte, i instal·lació d'aquests.
  - Part servei web, (SW\_TFC).
  - Part client, (GestioProjectes).
- **Manual d'instal·lació.**  
Manual per facilitar la instal·lació de les aplicacions de que es compon el projecte.
- **Manual de l'usuari.**  
Manual de l'usuari de l'aplicació client.
- **Presentació del projecte.**  
Presentació en Power Point on es fa èmfasi sobre els punts mes importants del desenvolupament del projecte.

### 3 Anàlisi funcional

Tota la informació completa de l'anàlisi funcional es troba al document d'anàlisi funcional del projecte.

Es proposa la realització d'un programari per l'automatització dels processos de gestió i control dels projectes informàtics d'una empresa. I més concretament de les assignacions de recursos a projectes i del control i gestió de la documentació d'aquests.

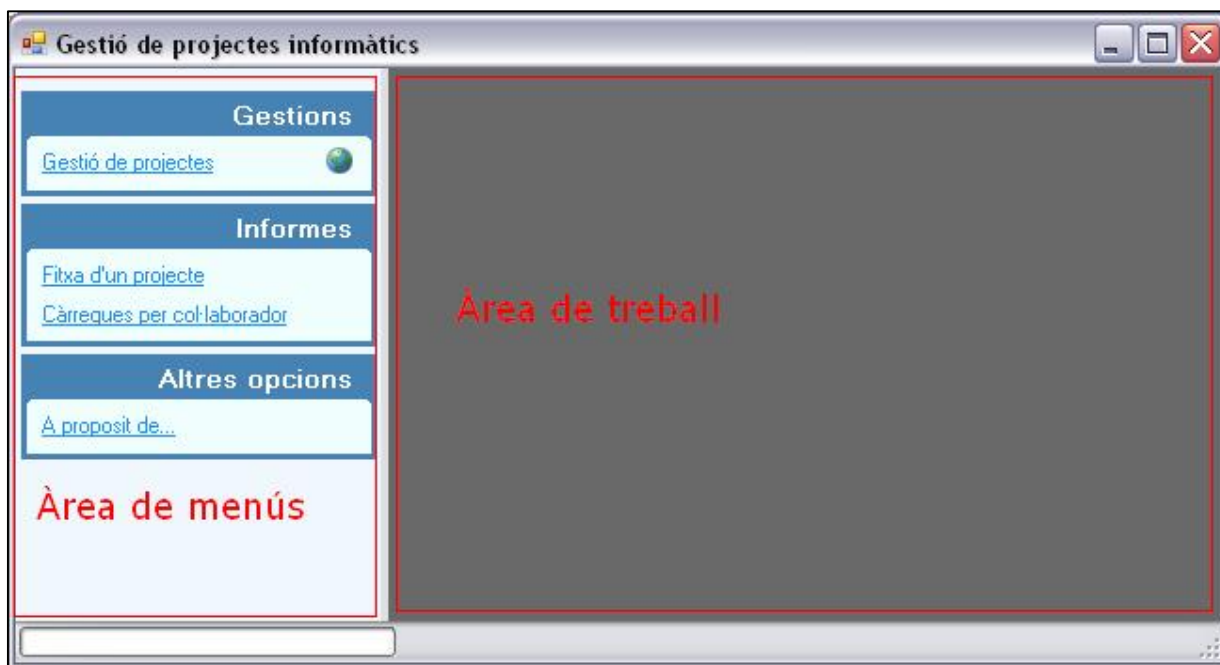
L'aplicació es divideix en dos mòduls que poden instal·lar-se en dos entorns diferents:

- **Part client:** On es gestionen les operacions que l'usuari realitza amb l'aplicació.
- **Part servidor:** Amb un servei web encarregat d'alimentar de dades a la part client.

#### 3.1 Descripció

##### 3.1.1 Descripció de la part client.

Es planteja un entorn en que, a partir d'una única finestra, es mostri a l'usuari les diferents opcions i gestions de l'aplicació. L'usuari, a partir d'un menú tipus "El Meu PC" de Windows® XP, podrà dirigir-se a l'opció de gestió que desitgi.



Il·lustració 5: Vista general de l'aplicació

L'usuari trobarà dos àrees a la pantalla. Un àrea de menús i un àrea de treball. Segons les opcions que l'usuari triï a l'àrea de menús, es mostrarà a l'àrea de treball una gestió o una altra.



Per anar a una opció de menú concreta, l'usuari passa el ratolí per sobre de la opció desitjada i, en fer clic a l'opció desitjada, a l'àrea de treball s'obrirà la gestió seleccionada. Paral·lelament, l'usuari pot encongir i maximitzar els diferents menús, fent que es mostrin només les opcions importants, o totes les opcions dels menús. Això ho aconsegueix fent doble-clic al títol del menú.

**Il·lustració 6: Vista del menú de l'aplicació**

L'àrea de menús es troba dividida en les següents seccions :

- **Gestions**

En aquesta secció l'usuari podrà escollir entre les principals gestions de que proveu l'aplicació. Que son:

- Gestió de projectes.
- Gestió de responsabilitats.
- Gestió de col·laboradors.
- Gestió de documents.

A l'àrea de treball s'obrirà la gestió triada en fer clic amb el ratolí sobre el menú.

Trobarà més informació d'aquestes gestions a la secció de [Requeriments funcionals](#).

- **Informes**

En aquest menú el sistema proposa els llistats i/o informes que es poden extreure amb aquesta aplicació. L'usuari fa clic sobre l'informe que desitja extreure, i l'aplicació el mostra a l'àrea de treball. Un cop mostrat l'informe, l'usuari el pot imprimir o fer diverses exportacions

En principi el sistema proposa dos informes:

- Fitxa d'un projecte.
- Càrreges per col·laborador.

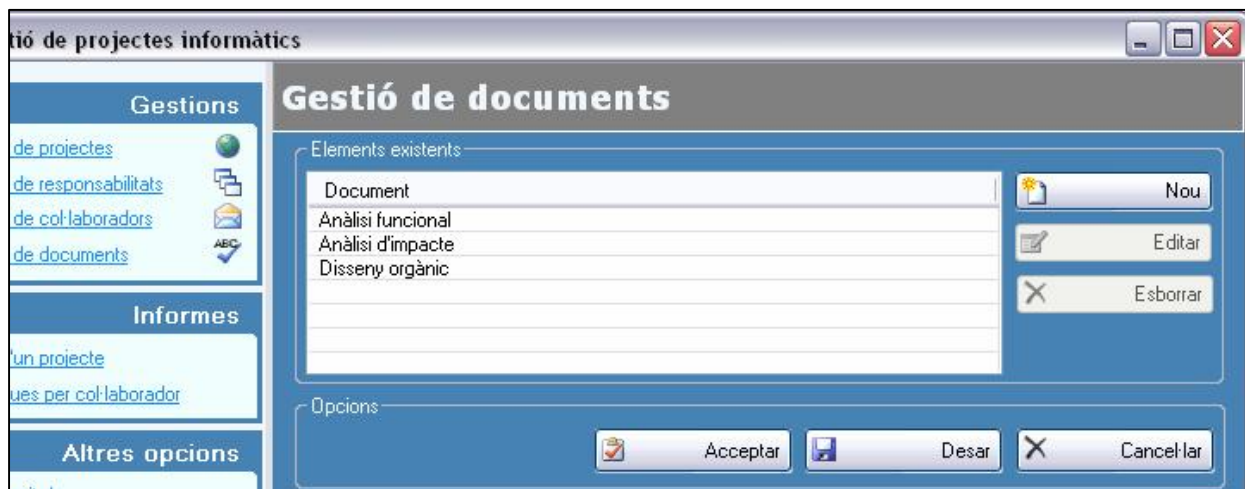
Però si el calendari ho permet, no es descarta la inclusió de més informes.

Trobarà més informació d'aquestes gestions a la secció de [Requeriments funcionals](#).

- **Altres opcions.**

A altres opcions l'usuari pot principalment obtenir ajuda o be sortir de l'aplicació.

L'àrea de treball es pot distingir per presentar una interfície homogènia en totes les gestions de l'aplicació.



II-lustració 7: Àrea de treball

En aquesta part comuna es poden distingir les següents parts:

- **Elements existents.**

Aquí es mostra una llista amb les dades rellevants de tots els registres de la gestió en curs. Es a dir: Si, com a la imatge de mostra, ens trobem a la secció de documents, se'ns mostrarà els noms de les descripcions de document que actualment estan donades d'alta en aquesta gestió. Fent clic sobre qualsevol document existent, i posteriorment fent clic sobre el botó **Editar**, editarem el registre seleccionat a la base de dades.

De la mateixa manera també podem crear un nou registre, o be esborrar un existent a la llista.

- **Opcions.**

Les opcions actuen sobre allò que estem editant/creant en aquell moment. Se'ns presenten tres botons on podem:

- Desar la informació editada fins aquell moment.
- Desar la informació editada fins aquell moment i sortir de la gestió.
- Sortir de la gestió sense desar cap canvi.

### 3.1.2 Descripció de la part servidor.

La part servidor no disposa inicialment d'interfície gràfica. Encara que si el calendari ho permet, està previst crear-ne una en ASP.NET, (C# o VB), per la gestió del servei web.

El servei web interconnecta la part client, (descrita anteriorment), amb la base de dades. I no te molta funcionalitat d'interfície. Com a informació essencial ha de mostrar si el servei web està engegat a aturat, i en quin estat es troba. Opcionalment també pot mostrar informació sobre l'estat de la base de dades. També pot realitzar altres funcions com:

- Gestionar la política de contrasenyes dels clients que fan servir el servei web.
- Establir rols d'accés a diferents clients. Bloquejar-ne l'ús o atorgar/retirar privilegis a altres.
- Mantenir un log de peticions i accessos a la base de dades.

### 3.2 Requeriments funcionals.

L'aplicació que es proposa consta dels següents subsistemes :

- **Gestió de documents.**

En aquesta secció l'usuari gestionarà l'índex de les diferents modalitats de documents que es poden generar en un projecte. Per exemple: anàlisi d'impacte, anàlisi orgànic, calendari, etc.

- **Gestió de col·laboradors.**

En aquesta secció l'usuari podrà afegir, editar, esborrar, etc, dades de col·laboradors dels projectes de l'empresa. Entre d'altres, hi hauran les següents dades:

- Nom i cognoms.
- Responsabilitat principal, (existent a la gestió de responsabilitats).
- Data d'alta del col·laborador.

- **Gestió de projectes.**

En aquesta secció l'usuari de l'aplicació gestionarà els diferents projectes de la empresa, amb els següents subapartats:

- Dades del projecte.

Aquí figuren les dades generals del projecte. Com per exemple: Nom del projecte, data d'inici del projecte, responsable principal del projecte, etc.

- Assignacions.

L'usuari entrarà els diferents col·laboradors del projecte, amb les següents dades:

- El rol que juguen en el projecte en curs, (un valor existent a la gestió de responsabilitats).
- El còmput d'hores dedicades al projecte en curs.

Si un col·laborador està relacionat amb un mateix projecte amb mes d'un rol, s'entrarà al sistema tantes vegades com rols tingui.

- Documentació.

L'usuari entrarà els diferents documents de que es compon el projecte. Indicant la seva localització, el seu contingut, (corresponent a un valor existent en la gestió de l'índex de documents), i un petit resum del mateix.

- **Gestió de responsabilitats.**

En aquesta secció existirà la descripció de tots els rols que un col·laborador pot tenir en el decurs d'un projecte. Per exemple: analista, programador, administrador, etc. Així com el cost/hora estimat per a cada responsabilitat; i d'una descripció orientativa de la responsabilitat.

- **Informe: Fitxa d'un projecte.**

La **Fitxa d'un projecte** es un informe en el que apareix tota la informació relativa a un projecte. Es correspon a tot allò que pot gestionar-se des de la opció de gestió de projectes:

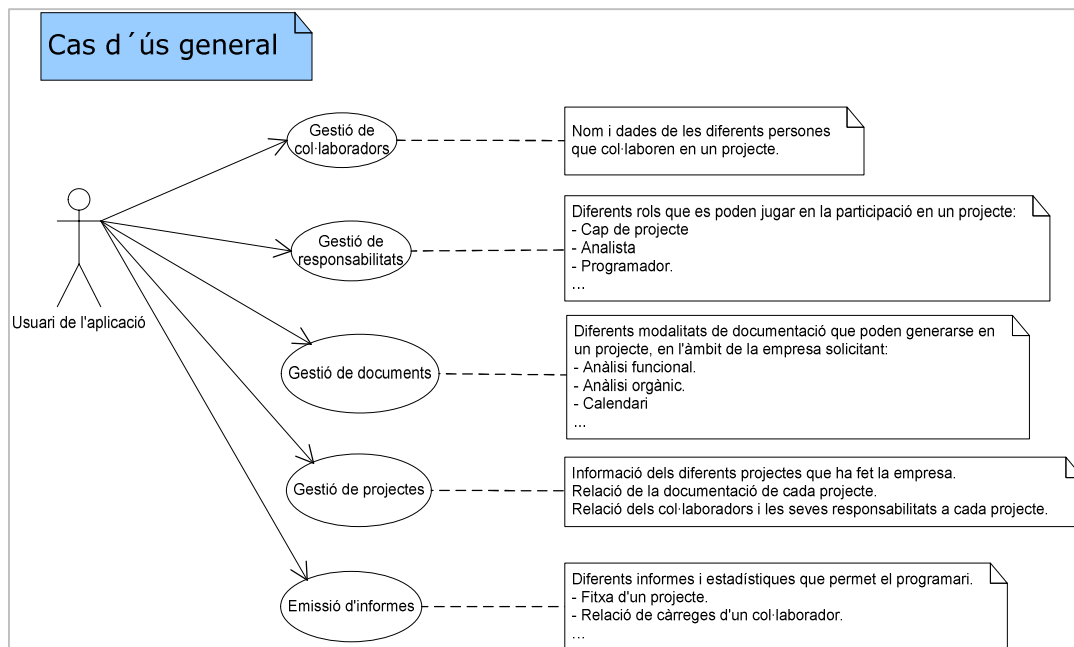
- Dades generals.
- Assignacions
- Documentació.

- **Informe: Càrregues per col·laborador.**

Per un col·laborador, apareix informació sobre tots els projectes en que ha estat implicat, així com el còmput d'hores dedicades a cada projecte. També apareix tota la informació disponible del col·laborador, corresponent a tot allò que pot gestionar-se des de l'opció de gestió de col·laboradors.

### 3.3 Cas d'ús general i tipologia.

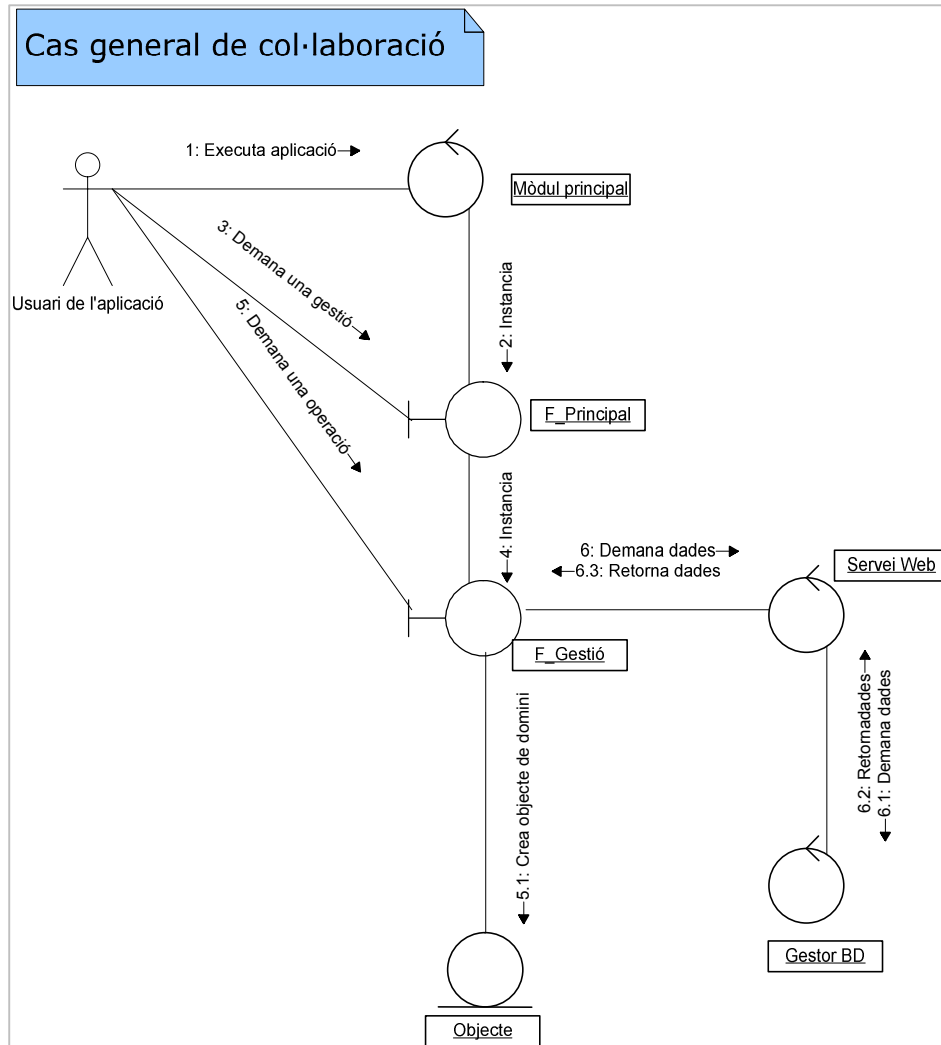
Les accions que l'usuari pot realitzar amb l'aplicació es defineixen de forma general en el següent cas d'ús :





### 3.4 Diagrama general de col·laboració.

La interacció de l'usuari amb el sistema es pot resumir en el següent diagrama de col·laboració general:



Il·lustració 8: Cas general de col·laboració

### 3.5 Requeriments tècnics.

#### 3.5.1 Llenguatge i entorn de desenvolupament:



L'aplicació es realitzarà en l'entorn Visual Studio.NET, i concretament en el llenguatge C#. Serà una aplicació windows estàndard, (winforms)

### 3.5.2 SGBD:



Les dades es contindran en un SGBD de la família SQL Server. I l'aplicació accedirà a les dades mitjançant ADO. Totes les operacions sobre dades es realitzaran a partir de procediments emmagatzemats al propi SGBD.

Com a característica especial, es planteja la possibilitat de que l'aplicació accedeixi a les dades de forma indirecta, emprant un servei web que atengui les peticions de l'aplicació. Per justificar aquesta intermediació, es planteja un escenari en que, per motius de seguretat, l'aplicació no veu, (ni tan sols coneix), l'origen de les dades. I per obtenir-les, fa servir una passarel·la que, mitjançant autenticació, decidirà si ha d'alimentar o no, (i amb quin detall), les dades demanades pel sol·licitant.

## 3.6 Qüestions de seguretat.

Les qüestions de seguretat son derivades principalment de qüestions de connexió a base de dades i al servei web. Els clients es connecten al servei web mitjançant nom d'usuari i contrasenya. No s'ha previst l'administració de perfils d'usuari; això vol dir que tots els clients tenen els mateixos privilegis. Per tant només es fa necessari un usuari i contrasenya per accedir al servei web.

El servei web també accedeix a la base de dades mitjançant nom d'usuari i contrasenya. Degut a que no son necessaris perfils d'usuari, el servei web pot fer servir el mateix nom d'usuari i contrasenya que els clients utilitzen per connectar-se a ell.

Però no es descarta, si el calendari ho permet, afegir polítiques de connexió al servei web mitjançant perfils d'usuaris. Creant usuaris que tenen accés a tota l'aplicació, altres que només tenen accés a informes. Altres que poden realitzar consultes però no poden fer canvis, etc.

## 4 Disseny orgànic.

Tota la informació complerta de l'anàlisi orgànic es troba al document de l'anàlisi orgànic del projecte.

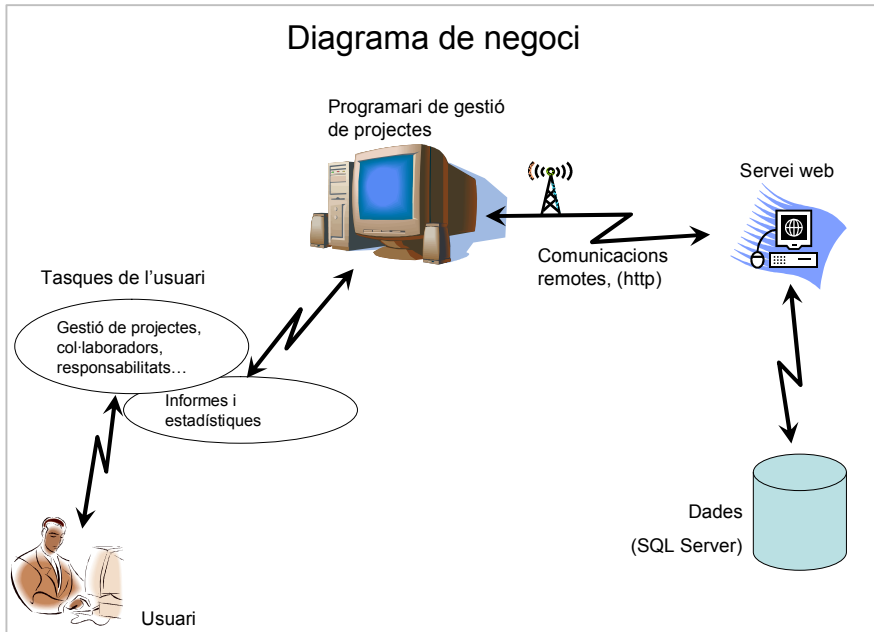
### 4.1 Descripció del disseny de l'arquitectura.

En la definició de l'arquitectura d'aquest projecte, i tenint en compte que es tracta de la creació o composició de diferents components de programari, hem de diferenciar primer la interacció d'aquests components i, tot seguit, la definició interna d'aquests. Per aquest motiu diferenciem entre arquitectura interna i externa.

#### 4.1.1 Arquitectura externa.

Aquest projecte està compost bàsicament de dos components:

- Part client.
- Servei web.

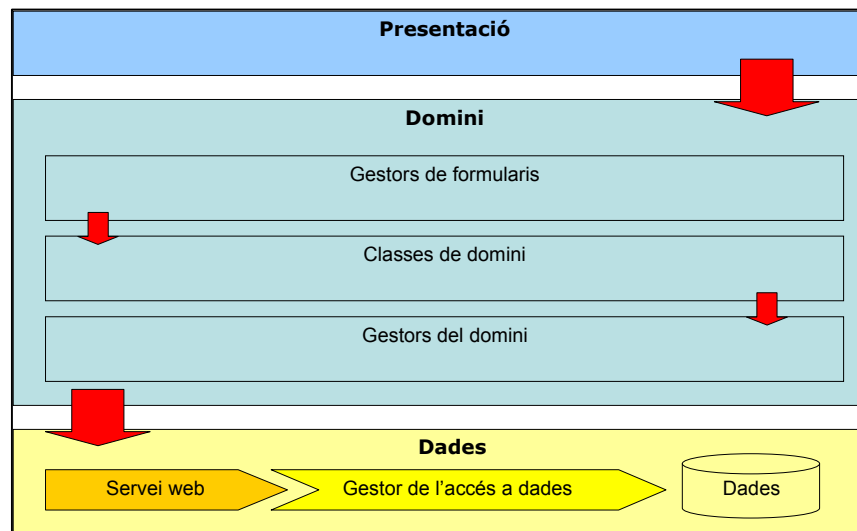


Des de la part client l'usuari envia peticions al servei web per llegir i desar la informació que l'aplicació és capaç de gestionar. El servei web, d'altra banda, contacta amb la base de dades per atendre les peticions que rep de la part client. L'arquitectura externa es pot definir gràficament amb el següent diagrama de negoci :

Òbviament, la part client és independent del servei web; és a dir, la part client pot accedir a aquest servei web o a qualsevol altre en el futur que pugui atendre les seves peticions. Per tant, la part client es desentén absolutament de les dades. Amb l'afegit que, gràcies a la intermediació del servei web, no només es desentén de la morfologia de la base de dades, sinó també de la seva localització física.

#### 4.1.2 Arquitectura interna.

Internament aquest projecte es basa en la implementació de diferents capes de programari, que s'especialitzen en la realització de diferents funcions, i que es comuniquen entre elles seguint una sèrie de normes: El següent diagrama mostra la disposició general de capes per una aplicació estàndard:



Il·lustració 9: Esquema de l'arquitectura d'implementació

En la secció [Disseny de classes](#) d'aquest document es pot consultar la disposició de capes, paquets y classes d'aquest projecte de forma concreta. A la secció [Altres decisions del disseny de classes](#) es pot consultar més informació sobre les decisions de disseny d'aquest projecte.

Bàsicament, les responsabilitats de cada una de les capes previstes al disseny de l'aplicació son les següents:

### Presentació

Aquesta capa es dedica única i exclusivament a la presentació de dades a l'usuari, i a la interacció d'aquest amb la interfície gràfica, (botons, desplegable, etc).

### Domini

Podem dividir la capa de domini en 3 capes especialitzades:

- **Gestors dels formularis.**

Aquesta capa fa funcions d'intermediació entre la capa de presentació i les classes del domini.

- **Classes de domini.**

Aquesta capa encapsula per una banda la lògica corporativa de l'aplicació, i per altra les dades que circulen en direcció a les capes de presentació i de dades.

- **Gestors del domini.**

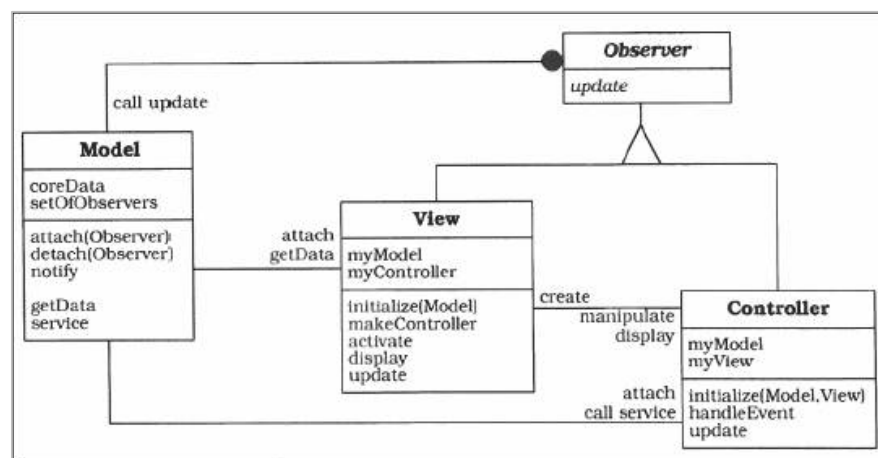
Aquesta capa fa d'intermediària entre les capes de classes del domini i la de dades.

### Dades

Aquesta capa està dedicada a l'alimentació de dades a les classes del domini. Com a cas particular en el nostre disseny, cal destacar la presència d'un servei web que farà d'intermediari entre la pròpia capa de dades i la capa de domini.

En la creació d'una aplicació d' $n$  capes s'han de tenir en consideració una sèrie de normes i convenis, que s'han de seguir per assegurar el perfecte funcionament de les diferents capes i de l'aplicació completa. Un conjunt d'aquestes normes parla de quina manera és possible realitzar la intercomunicació entre les diferents capes d'un projecte. Aquí entra en joc el disseny arquitectònic conegut com "Model vista controlador", (MVC). Aquest model estableix la existència de 3 entitats:

- Model
- Vista
- Controlador



II-lustració 10: Model Vista Controlador<sup>1</sup>

<sup>1</sup> Microsoft.com. Patrones y antipatrones II

Cada una d'aquestes entitats realitza un conjunt especialitzat d'accions, i es comunica amb les altres entitats de tal forma que el tot, (l'aplicació), funcioni correctament. Aquesta comunicació entre les diferents entitats es realitza de la forma següent:

### Model

Encapsula les dades i les funcionalitats. I és independent de qualsevol entitat d'entrada/sortida. En el nostre model de capes, el model correspon a les subcapes de *Classes de domini* i *Gestors de domini* de la capa de *Domini*.

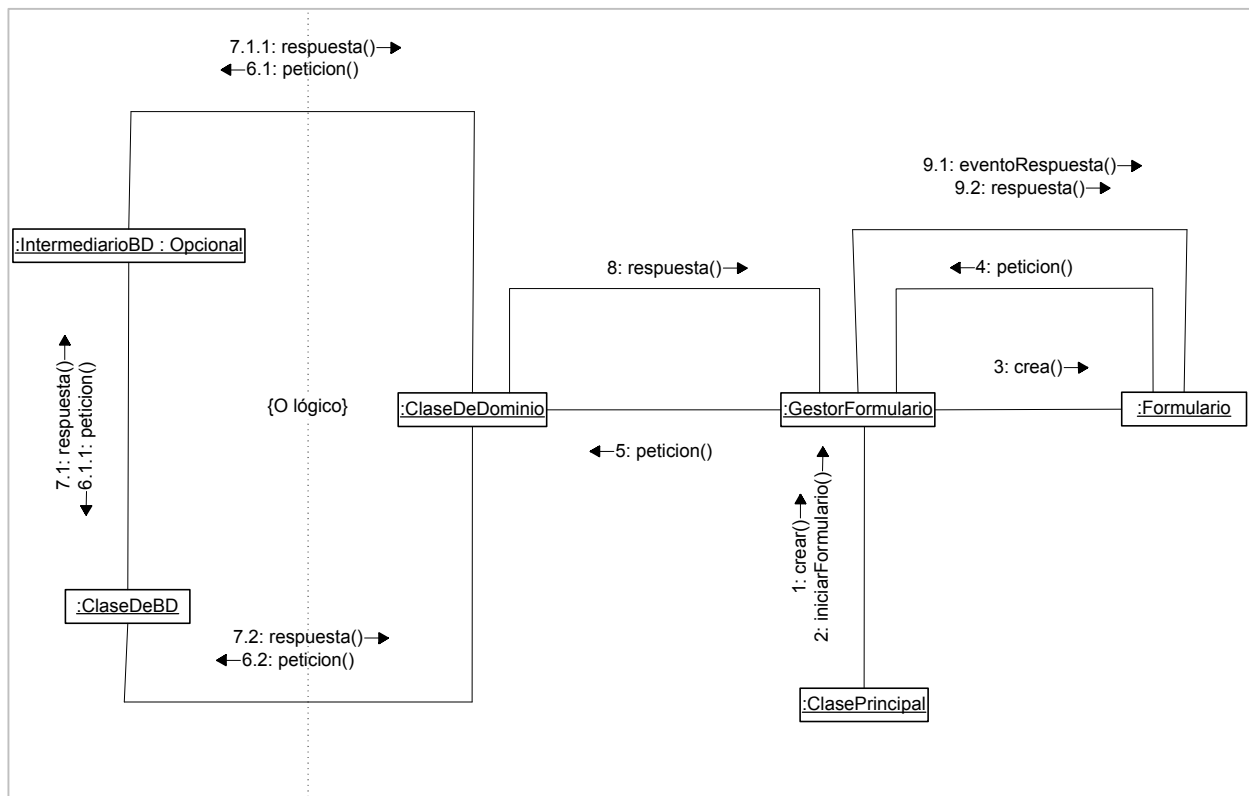
### Vista

Mostra la informació a l'usuari. No conté lògica d'aplicació. Obté les dades de la entitat model. En el nostre model de capes, la entitat es correspon amb la capa de *Presentació*.

### Controlador

Rep les entrades i provoca events que interconnecten les entitats de vista i model. En el nostre model de capes, la entitat Controlador es correspon amb la subcapa de *Gestors dels formularis* de la capa de *Domini*.

Gràficament, la comunicació entre les diferents classes d'una estructura en  $n$  capes, conjuntament amb la metodologia proposada al disseny MVC podria ser el següent:



Il·lustració 11: Diagrama general d'interacció entre capes

En aquesta representació, però, es planteja la possibilitat d'incloure o no una subcapa de *Gestors del domini* a la capa de domini. És a dir, es planteja la possibilitat de que la part Model del disseny MVC sigui competència única de les classes de domini

## 4.2 Aprofitament de components de tercers i propis.

A l'aplicació s'aprofiten un conjunt de components i eines de tercers per tal d'afavorir el rendiment i la viabilitat del desenvolupament. Entre d'altres s'han fet servir els components i eines següents:

- Informes amb l'eina Crystal reports.
- Visor d'arxius HTML per mostrar l'ajuda de l'aplicació.

Com a cas particular s'ha reaprofitat un component de desenvolupament intern. Es tracta d'un component gràfic que mostra un menú a l'estil dels menús de finestra de Windows XP. Aquest component va ser desenvolupat a l'any 2003 en Visual Basic .NET 2003.



## 4.3 Decisions respecte al disseny de l'arquitectura.

A continuació descriurem, en quan al disseny orgànic, algunes decisions d'implementació interessants d'aquest projecte.

### 4.3.1 Mes sobre l'arquitectura interna.

L'arquitectura interna, tal i com s'ha explicat a la secció [Descripció del disseny de l'arquitectura](#), s'ha dissenyat tenint com a base una implementació de **3 capes** i el model **MVC**. Però, per tal de garantir la màxima reutilització i encapsulació, hem subdividit la capa de domini en 3 subcapes, obtenint així una implementació  $n$ , (5), capes. Aquesta decisió de disseny es justifica de la forma següent:

#### - Capa de gestors del domini

Un gestor de domini es essencialment una classe que rep peticions d'una o diverses classes de domini. Aquestes peticions fan referència a interaccions amb la base de dades. El gestor de domini fa d'intermediari entre la classe de domini i les classes d'implementació de l'accés a les dades, que conformen la capa de dades.

Interposar una capa d'intermediació entre les classes de domini i la classe de dades té l'avantatge principal en que la classe de domini es desentén de les peculiaritats de la implementació de la capa de dades. De forma que seria relativament senzill substituir per a una classe de domini qualsevol, un gestor per un altre; i d'aquesta manera variar el comportament d'aquesta classe de domini en quan a l'accés a dades. A més, d'aquesta manera, un mateix gestor podria atendre les peticions de diverses classes de domini.

#### - Capa de gestors de presentació

De forma semblant a el que s'ha dit en referència a la capa de gestors de domini. La lògica de domini no es únicament lògica des del punt de vista de les classes de domini. Hi ha molta implementació de comportament que recau fora de les classes de domini i que, tradicionalment, han recaigut sobre els formularis.

La interposició d'una capa de gestors de formularis entre els formularis i les classes de domini garanteix la independència entre les capes de domini i de presentació; de forma que, en un cas hipotètic, podria substituir-se la capa de presentació, (per exemple, migrar d'una aplicació de formularis a una aplicació web), sense que es veies afectada la capa de domini.

### 4.3.2 Herència d'implementació.

En alguns casos s'ha optat per utilitzar mètodes d'herència d'implementació, (o herència en temps de disseny), per agilitzar la implementació mitjançant la reutilització de codi i disseny que proporciona aquesta metodologia. En alguns casos en que s'ha optat per aquesta metodologia, s'ha intentat en tot moment heretar de classes abstractes. El motiu es perquè d'aquesta forma es preveu que una reimplementació posterior de la classe pare pugui entorpir el correcte funcionament de les classes filles. Es a dir, tal com diu Erich Gamma<sup>2</sup>: “Les dependències d'implementació poden causar problemes al tractar de reutilitzar una subclasse. Quan qualsevol aspecte de la implementació heretada no sigui adequat per a nous dominis de problemes, la classe pare haurà de ser reescrita de nou o reemplaçada per una altra més adequada. Aquesta dependència limita la flexibilitat i la reutilització. Una solució a això es heretar només de classes abstractes, ja que aquestes generalment tenen poca o cap implementació.”

Un exemple d'això son els gestors del domini que hereten d'un gestor bàsic, (*GD*). O els formularis que hereten de diversos formularis bàsics, (*F\_UtilitatsHeretable* per als formularis de gestió, i *F\_Informe* per als formularis d'informe).

### 4.3.3 Herència d'interfaces

S'han definit de forma exhaustiva les interfaces que, a mode de contracte, defineixen el comportament i les funcionalitats de les diferents classes de la solució. En lloc de dissenyar en base a implementacions independents, s'han dissenyat els comportaments comuns dels objectes, i s'han implementat les classes en base a aquests comportaments. D'aquesta forma, qualsevol classe del mateix àmbit hauria d'implementar les mateixes funcionalitats, per tal que es pugues adaptar al sistema de forma correcta.

Un exemple d'això el representen les classes de domini que implementen la interface *IDomini*.

### 4.3.4 Sobre la base de dades

Tots els accessos a la base de dades es realitzen mitjançant crides a procediments emmagatzemats. No hi ha consultes dinàmiques en codi per accedir a les dades. La estructura optada en la construcció dels procediments emmagatzemats es la següent:

- **Càrrega de registres**

Cada classe de domini te associat un procediment emmagatzemat que realitza la càrrega de les dades. Els procediments emmagatzemats d'aquesta mena tenen com a paràmetre el/els ID de l'objecte de domini associat. Per regla general, si no s'indica ID en la càrrega de les dades, el procediment emmagatzemat retornarà una llista de tots els registres d'aquell domini que existeixin a la base de dades.

- **Alta/Modificació de registres**

L'alta o modificació de registres conformen els procediments emmagatzemats mes elaborats de l'aplicació. Per regla general, cada objecte de domini te associat un procediment emmagatzemat per aquestes tasques. Aquests procediments emmagatzemats no contenen una llarga llista de

<sup>2</sup> Erigh Gamma. Patrones de diseño. Addison Wesley

paràmetres per a cada camp de registre de l'objecte de domini, sinó que espera **una estructura XML** amb tota la informació del/dels objecte/s a salvar/modificar.

La elecció entre alta o modificació la fa de forma transparent el procediment emmagatzemat mateix; simplement determinant si l'ID del registre a salvar existeix o no a la base de dades.

#### - Esborrar registres

Per regla general hi ha un únic procediment emmagatzemat encarregat de la eliminació de registres de la base de dades. La selecció del domini es concreta mitjançant un paràmetre d'aquest procediment emmagatzemat.

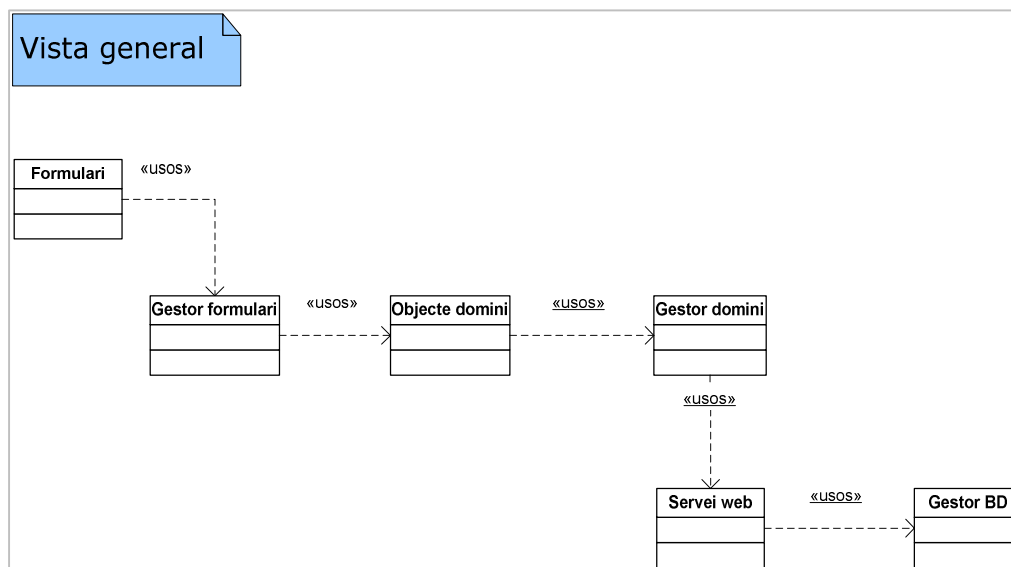
## 4.4 Disseny de classes.

### 4.4.1 Diagrames de paquets, arquitectura i comunicació.

Els següents diagrames mostren l'estructura de l'aplicació des del punt de vista de l'arquitectura de paquets, conjunts i comunicacions de dades.

#### 4.4.1.1 Vista general

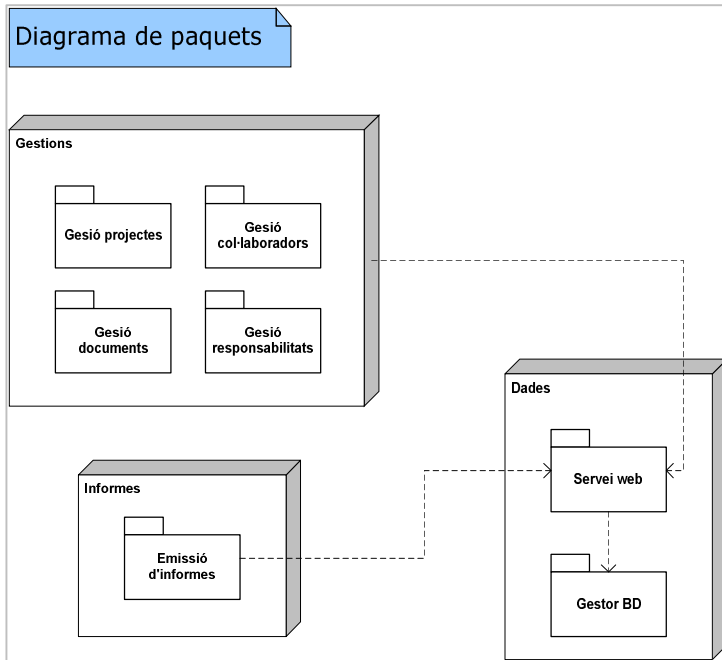
Aquest diagrama mostra la política general de comunicacions entre les diferents classes de que es compon l'aplicació. Aquest esquema de comunicacions està directament lligat a l'esquema d'arquitectura interna mostrat a la secció [Disseny de l'arquitectura](#).



Il·lustració 12: Vista general de relació entre capes



#### 4.4.1.2 Paquets

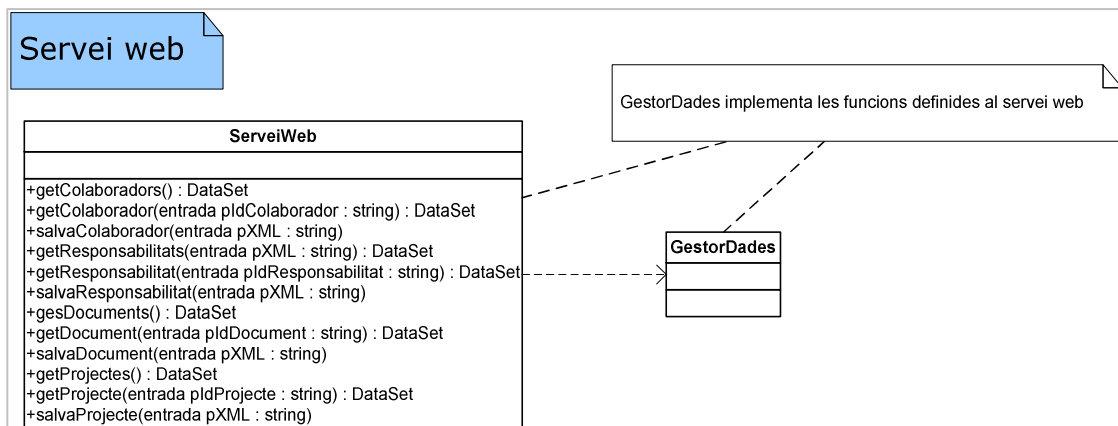


Vista general del domini de l'aplicació. Des del punt de vista dels diferents paquets de l'aplicació, es poden veure les intercomunicacions més importants.

Il·lustració 13: Diagrama de paquets

#### 4.4.1.3 Classes del servei web

El servei web es realitzarà, al igual que l'aplicació client, en Visual Studio .NET 2005 C#. Concretament es realitzarà com a projecte independent de l'aplicació client, (com a servei web). Al ser un projecte a part, i dedicat en exclusiva a la intercomunicació entre la part client i la base de dades, es mostra a part del diagrama de classes general.

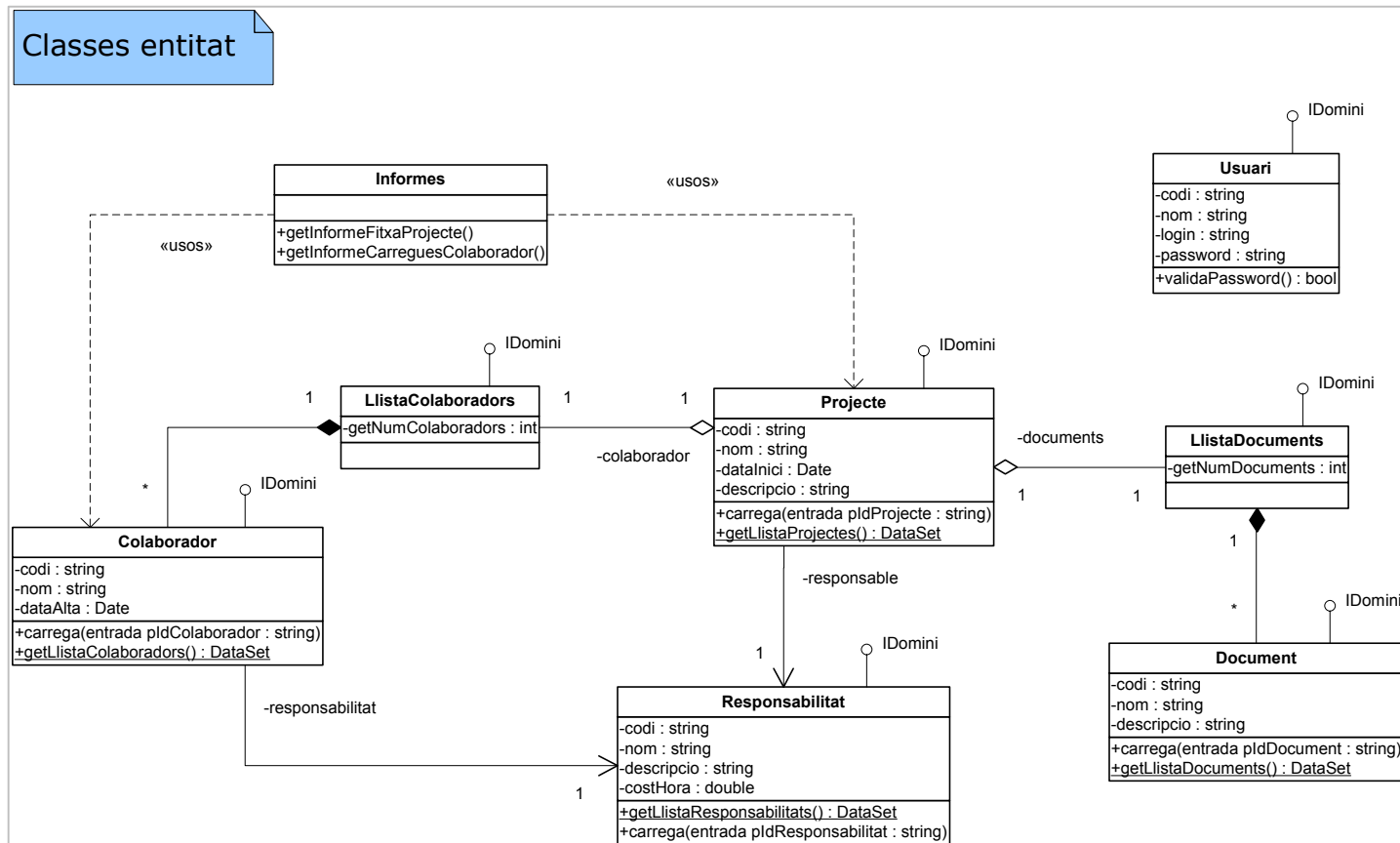


Il·lustració 14: Classes del servei web

## 4.4.2 Diagrames de disseny de classes.

### 4.4.2.1 Entitats

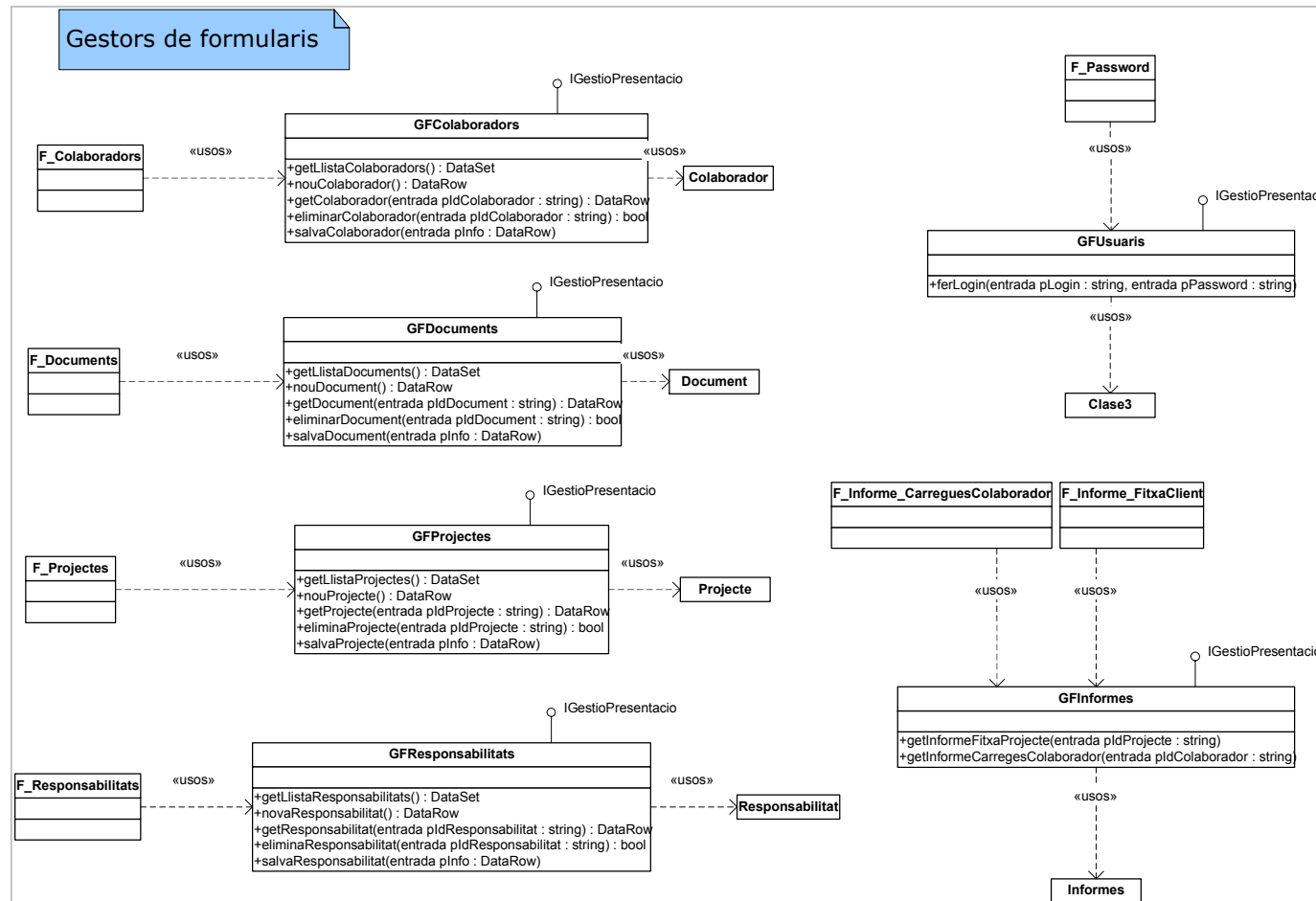
El diagrama de classes d'entitat de l'aplicació, on es poden veure les classes de domini i la seva intercomunicació.



Il·lustració 15: Classes d'entitat

#### 4.4.2.2 Gestors de formularis

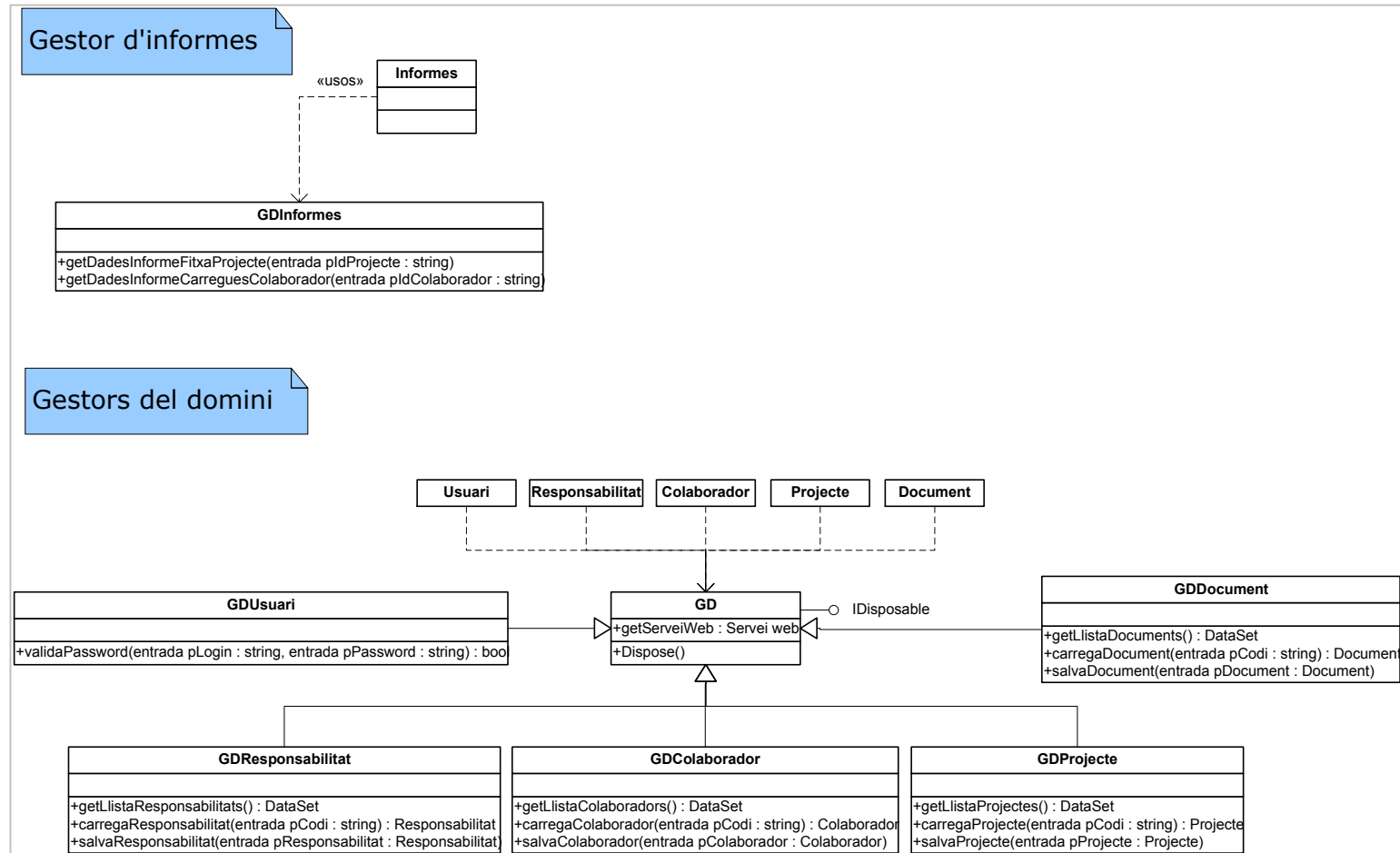
El diagrama de les classes dels gestors dels formularis. Aquestes classes fan d'intermediari entre els formularis de l'aplicació i les classes del domini.



Il·lustració 16: Gestors dels formularis

### 4.4.2.3 Gestors de domini

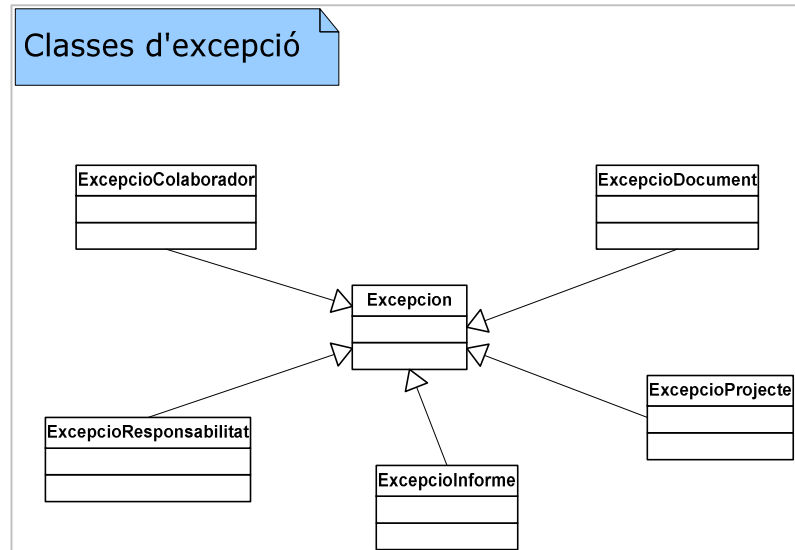
El diagrama de classes dels gestors del domini. Aquestes classes fan d'intermediari entre les classes del domini i el servei web.



Il·lustració 17: Gestors del domini

#### 4.4.2.4 Excepcions

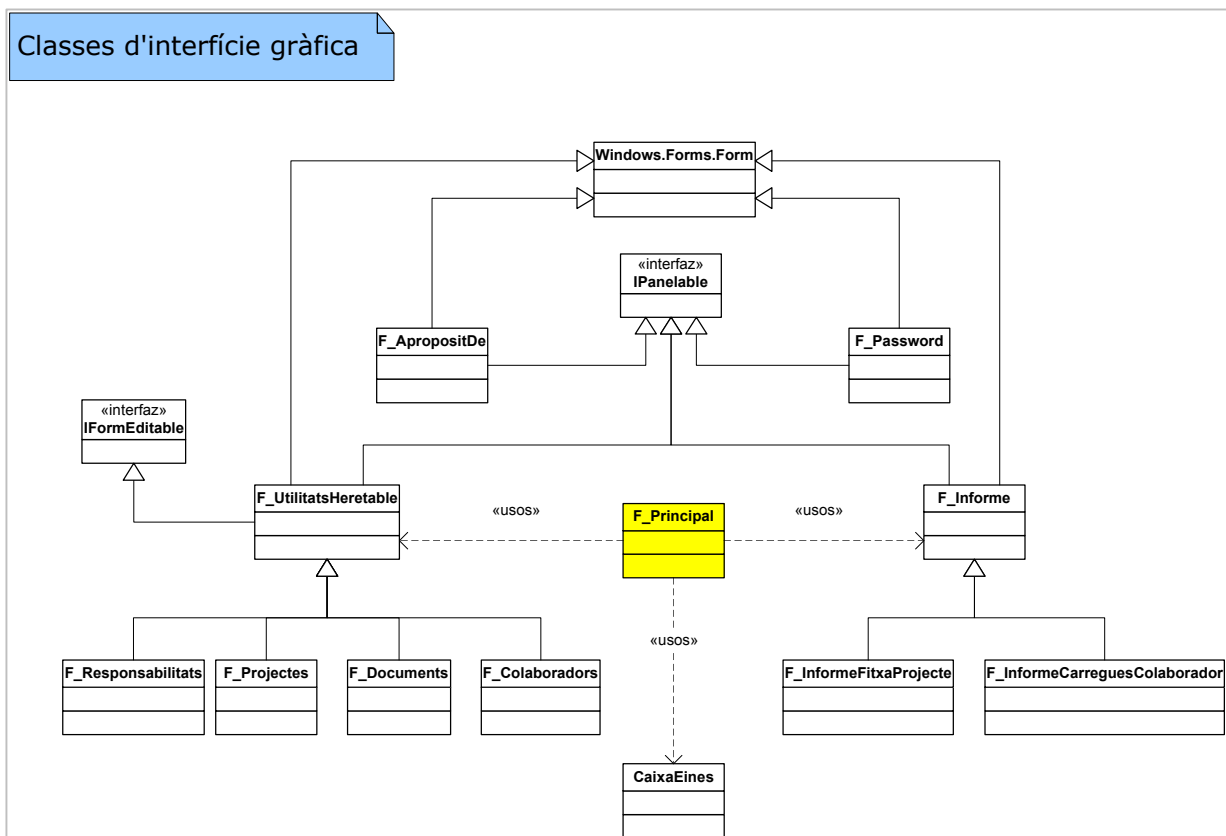
El següent diagrama mostra les excepcions de l'aplicació; on podem trobar una excepció personalitzada per cada unitat operativa de l'aplicació, (gestió de col·laboradors, responsabilitats, documents, projectes i emissió d'informes).



Il·lustració 18: Classes d'excepció

#### 4.4.2.5 Pantalles

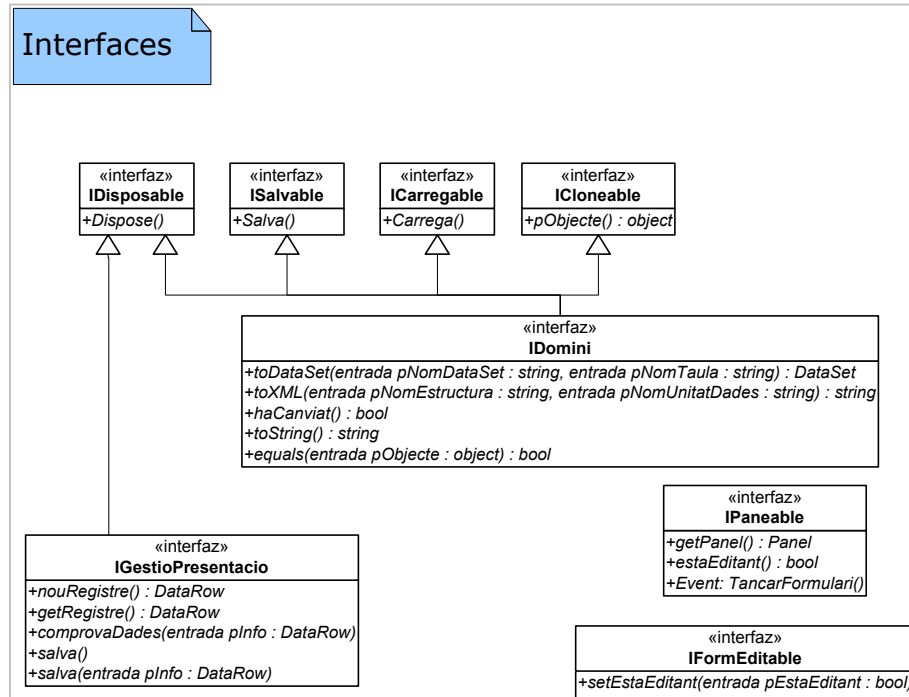
Les diferents pantalles, (formularis), de l'aplicació.



Il·lustració 19: Classes de presentació

#### 4.4.2.6 Interfaces

Les diferents interfaces de l'aplicació. Com a elements contractuals de les diferents classes del domini i gestors.

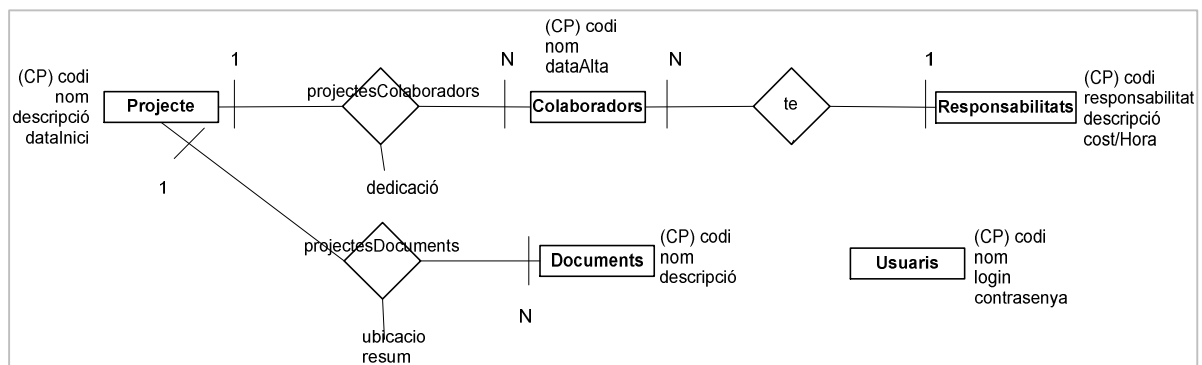


Il·lustració 20: Interfaces de l'aplicació

### 4.5 Disseny de la base de dades

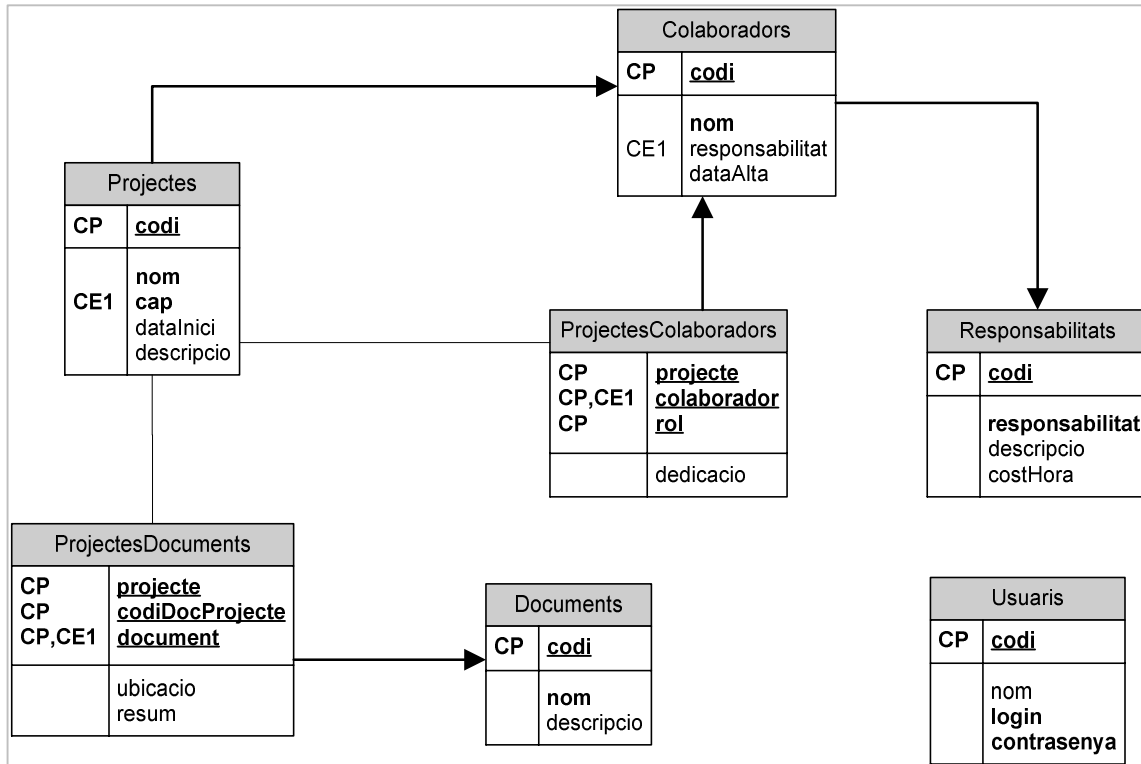
La base de dades, segons s'ha establert al pla de treball i, posteriorment, a l'anàlisi funcional, s'implementarà sobre un SGBD SQL Server. Aquesta base de dades s'instal·larà sobre un maquinari indeterminat, al que tindrà accés el servei Web previst en aquest projecte.

#### 4.5.1 Diagrama ER



Il·lustració 21: Diagrama E/R de la base de dades

#### 4.5.2 Disseny lògic.



Il·lustració 22: Disseny lògic de la base de dades

#### 4.6 Canvis sobre el disseny orgànic durant la implementació.

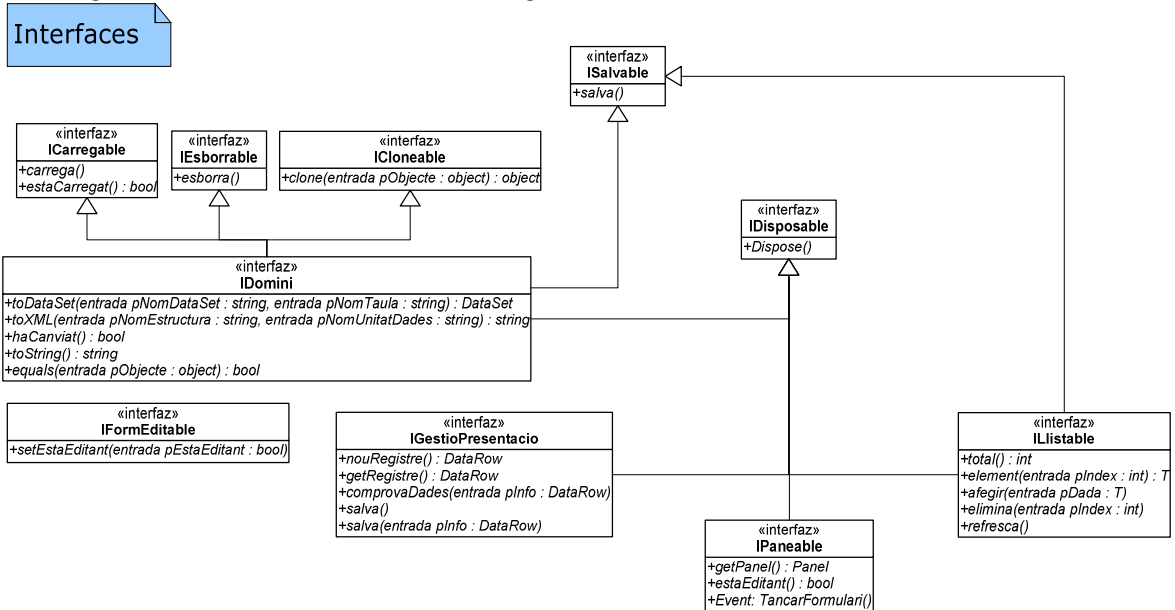
Durant la implementació, han sorgit necessitats noves i algunes incongruències amb el disseny de classes, proposat en primera instància a l'anàlisi orgànic. Per aquest fet s'han fet algunes modificacions sobre els diagrames que resumim a continuació:

##### 4.6.1 Disseny d'interfaces.

En el transcurs de la implementació va sorgir la necessitat de fer que els objectes del domini fossin *esborrables*. De la mateixa manera que son *salvables* o *carregables*. Així doncs es va incloure una nova interface amb aquesta funcionalitat. I es va vincular amb la interface general *IDomini*.

Per gestionar les llistes d'elements d'un projecte, (assignacions de col·laboradors i de documents), es van crear una sèrie de classes de llista. La funcionalitat d'aquestes classes de llista ve regida per la interface *ILlistable*.

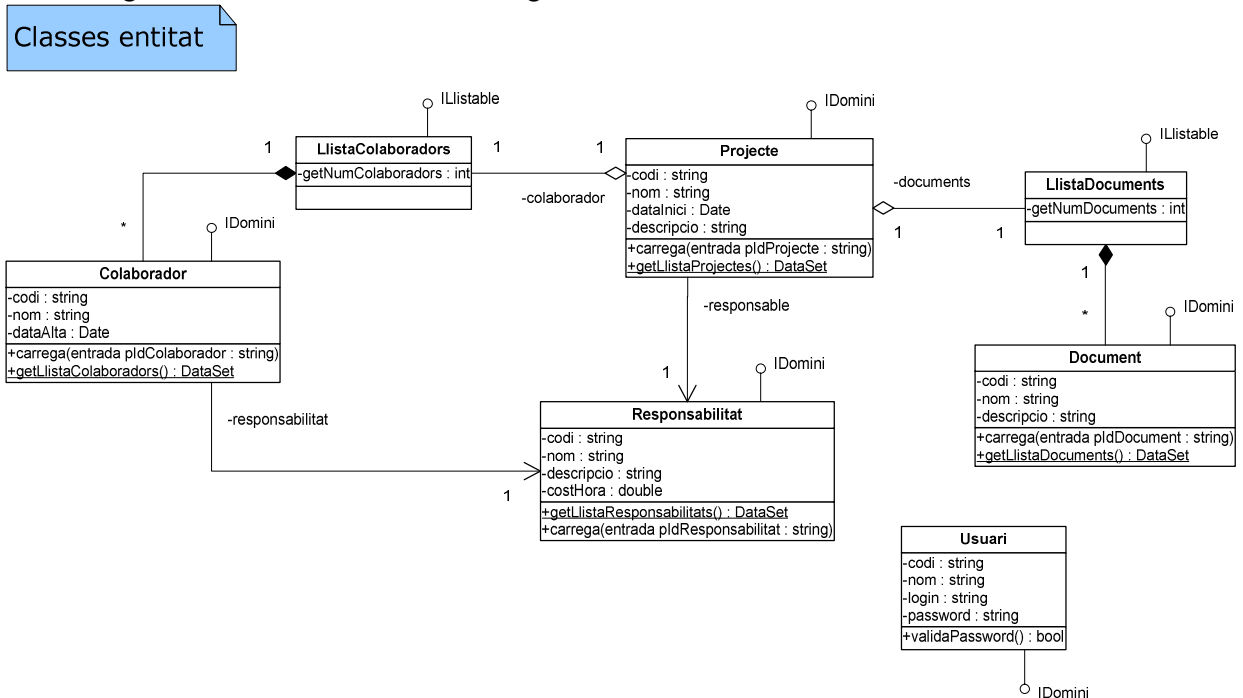
El nou diagrama de classes d'interface es el següent:



### 4.6.2 Entitats.

La classe *Informe* desapareix del diagrama, perquè es va optar per prescindir d'una classe de domini per mostrar informes, fent servir directament els gestors d'informes. Aquesta acció queda justificada pel fet que la classe mencionada no conté atributs, i els mètodes que te son tasques que es redirigeixen al gestor del domini corresponent.

El nou diagrama de classes d'entitat es el següent:



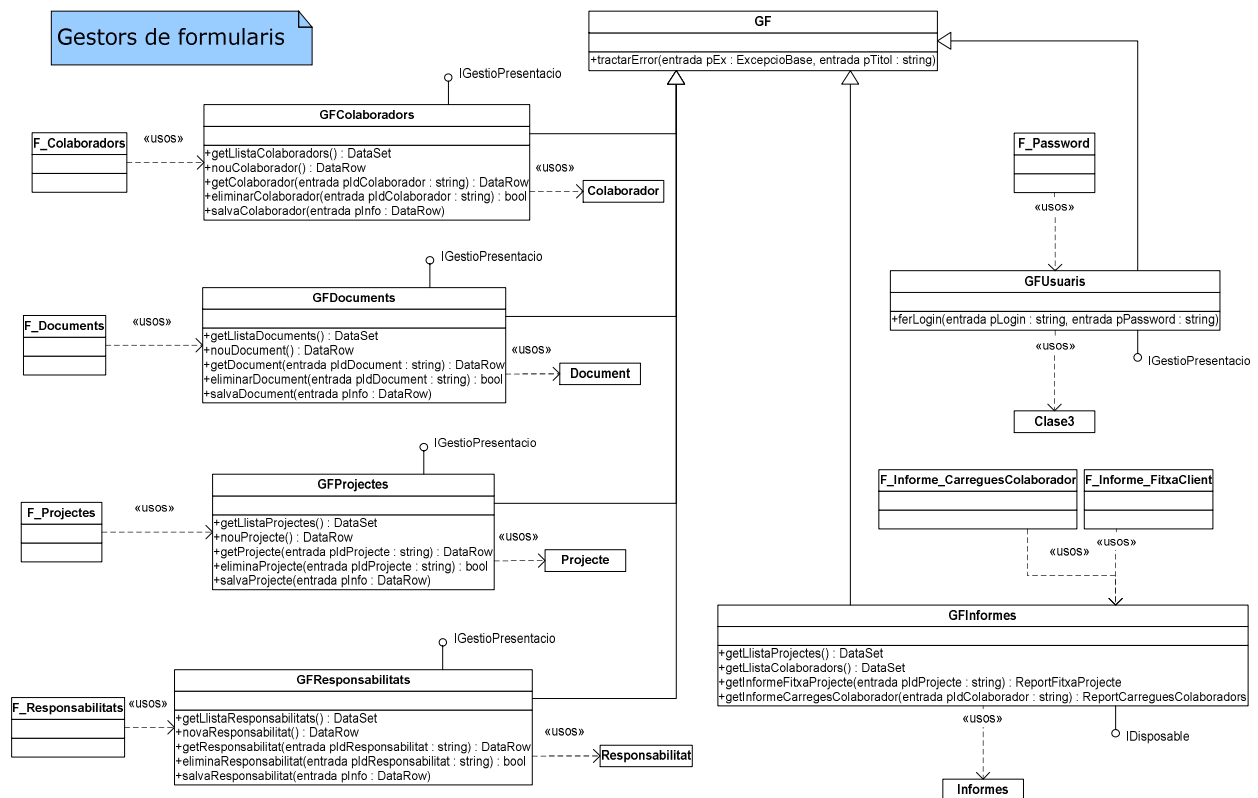


### 4.6.3 Gestors dels formularis.

Als gestors dels formularis se'ls ha afegit un gestor universal, (*GF*), del que hereten tots els gestors d'aquest tipus. Aquest gestor conté el tractament unificat d'errors de l'aplicació.

Al gestor d'informes, (*GFInformes*), se li ha afegit alguns mètodes no previstos inicialment a l'anàlisi orgànic.

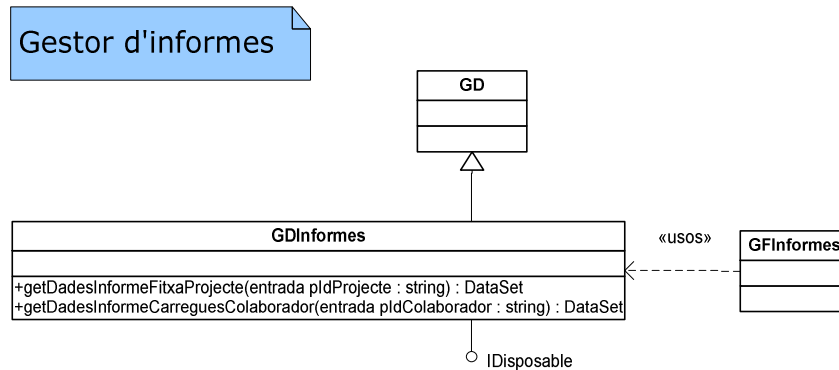
El nou diagrama de gestors de formularis es el següent:



### 4.6.4 Gestors del domini: Gestor d'informes.

S'ha mostrat gràficament la relació entre *GFInformes* i *GDInformes*; per indicar de forma més clara que el gestor de formulari d'informes, (*GFInformes*), es comunica directament amb el gestor de domini d'informes, (*GDInformes*), sense la presència d'una classe de domini.

El nou diagrama de gestor d'informes es el següent:

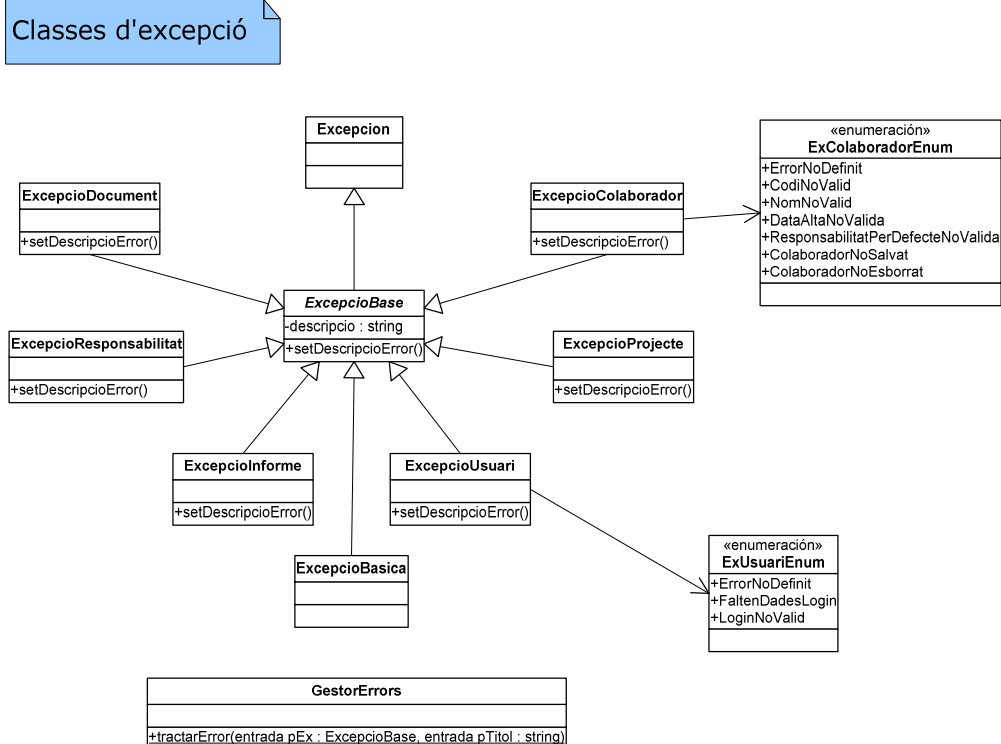


#### 4.6.5 Classes d'excepció.

Les classes d'excepció son regides per enumeracions que indiquen quin es l'error produït, on cada enumeració està vinculada a una classe d'excepció. Aquestes enumeracions no es van documentar a l'anàlisi orgànic.

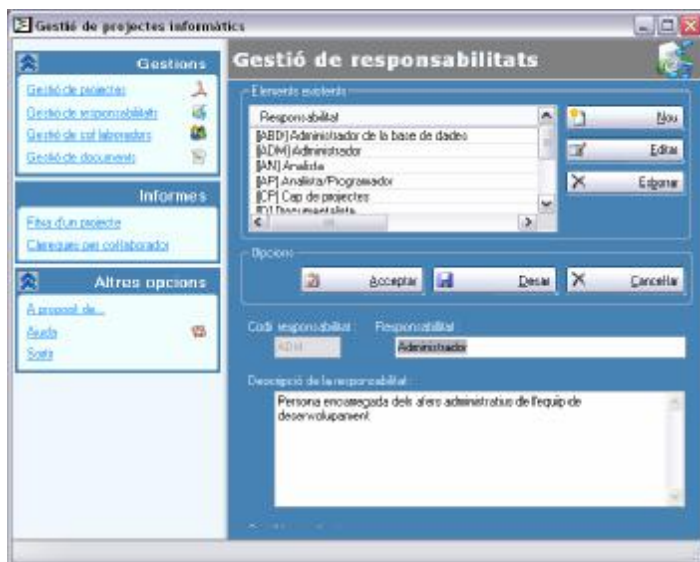
A més, sorgeix una excepció que es la base de tota la resta, (*ExcepcioBase*), amb funcionalitats i informacions bàsiques. També sorgeix una excepció nova, amb el nom *ExcepcioBasica*. Aquesta excepció es utilitzada quan no es vol entrar en detall sobre un error concret. També apareix la excepció *ExcepcioUsuari*.

El nou diagrama d'excepcions es el següent:



## 5 Manual de l'usuari.

Tal i com s'ha dit a la introducció, Gestió de projectes es una aplicació per al manteniment i control dels projectes informàtics d'una empresa de desenvolupament de programari. La presentació gràfica simula l'entorn *d'escriptori de windows*; aquesta circumstància el fa especialment senzill en l'ús per a qualsevol usuari que estigui habituat a manipular aplicacions en entorns Windows®.

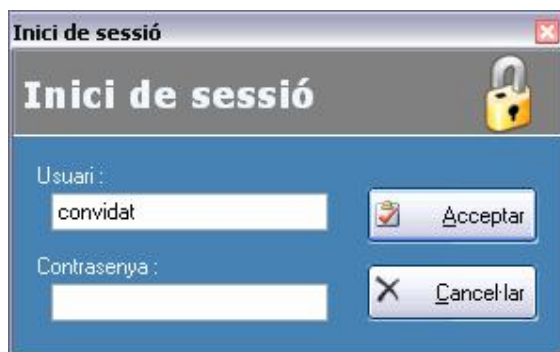


Destaquen dues àrees de treball. A l'àrea de l'esquerra es presenten les diferents opcions de l'aplicació. Aquesta àrea pot dividir-se en tres grups. Un grup d'opcions de gestió, (gestió de projectes, de responsabilitats, de col·laboradors i de descripcions de document). Un grup d'informes, (fitxa de projecte, càrregues per col·laboradors). I una àrea d'opcions diverses, (ajuda, a posòsit de i sortir).

A l'àrea de la dreta podem trobar la zona de treball. A cada opció seleccionada sobre el menú aquesta àrea canvia, mostrant la gestió o l'informe triat.

### 5.1 Autenticació

Quan s'inicia l'aplicació apareix una finestra en que es demana nom d'usuari i contrasenya:



Introdueixi un nom d'usuari i contrasenya vàlid per poder iniciar sessió a l'aplicació. Amb el nom d'usuari "convidat" i sense contrasenya, iniciarà una sessió com a convidat.

## 5.2 Gestions

A continuació es mostren les diferents gestions de l'aplicació. Les gestions poden diferenciar-se en dos grups:

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>- Configuracions de gestions             <ul style="list-style-type: none"> <li>- Gestió de descripcions de documents</li> <li>- Gestió de responsabilitats</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>- Gestions             <ul style="list-style-type: none"> <li>- Gestió de col·laboradors</li> <li>- Gestió de projectes.</li> </ul> </li> </ul> |
|---|--|

Les *Configuracions de gestions* son gestions que permeten completar la informació de les gestions de col·laboradors i de projectes.

El funcionament es idèntic per a totes les gestions. Es basa en tres àrees de treball:

- Elements existents
- Opcions
- Entrada de dades

Col·laborador	Responsabilitat	Data d'alta
[C01] Col·laborador 1 Analista	[AN] Analista	01/01/2006 0:00:00
[C02] Col·laborador 2 Analista	[AN] Analista	31/01/2006 0:00:00
[C03] Col·laborador 3 programador becari	[PJ] Programador junior	04/05/2006 0:00:00
[C04] Col·laborador 4 Programador	[PS] Programador senior	05/04/2006 0:00:00
[C05] Col·laborador 5 Programador	[PS] Programador senior	30/12/2005 0:00:00
[C06] Col·laborador 6 Analista/Programador	[AP] Analista/Programador	01/01/2006 0:00:00
[C07] Col·laborador 7 Analista/Programador	[AP] Analista/Programador	04/01/2006 0:00:00

Opcions: Acceptar, Desar, Cancel·lar

Codi col·laborador: C03    Nom i cognoms: Col·laborador 3 programador becari

Responsabilitat principal: [PJ] Programador junior

Data d'alta: 04/05/2006

A l'àrea d'**elements existents** es mostra la llista dels registres d'aquesta gestió existents a la base de dades:

Col·laborador	Responsabilitat	Data d'alta
[C01] Col·laborador 1 Analista	[AN] Analista	01/01/2006 0:00:00
[C02] Col·laborador 2 Analista	[AN] Analista	31/01/2006 0:00:00
[C03] Col·laborador 3 programador becari	[PJ] Programador junior	04/05/2006 0:00:00
[C04] Col·laborador 4 Programador	[PS] Programador senior	05/04/2006 0:00:00
[C05] Col·laborador 5 Programador	[PS] Programador senior	30/12/2005 0:00:00
[C06] Col·laborador 6 Analista/Programador	[AP] Analista/Programador	01/01/2006 0:00:00
[C07] Col·laborador 7 Analista/Programador	[AP] Analista/Programador	04/01/2006 0:00:00

Sobre aquesta llista podem:

- Fer doble-clic, (o bé seleccionar-lo i fer clic sobre *Editar*), per seleccionar-ne un i mostrar-lo a l'àrea d'entrada de dades, per editar-lo i desar-lo posteriorment.
- Seleccionar-ne un i fer clic sobre *Esborrar*, per esborrar l'element seleccionat de la base de dades. Amb aquesta operació l'element desapareix de la llista.
- Fer clic sobre *Nou* per entrar un nou registre en aquesta gestió.

A l'àrea d'**opcions** podem fer diverses accions de caire general:

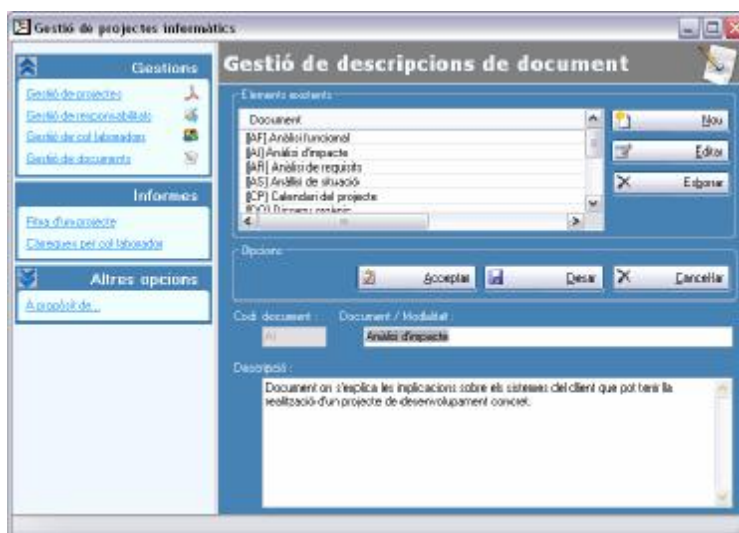


Podem:

- Acceptar  
Acceptarem l'entrada de dades actual, editant o afegint un nou registre a la base de dades. Posteriorment, la gestió es tanca.
- Desar  
Fem el mateix que amb *Acceptar*, però la gestió no es tancarà.
- Cancel·lar  
Descartem els possibles canvis que estiguéssim fent sobre l'àrea d'entrada de dades, i tanquem la gestió.

L'àrea d'**entrada de dades** es diferent a cada gestió. Allà podem editar un registre existent o donar d'alta un nou registre:

### 5.2.1 Gestió de descripcions de documents



La gestió de descripcions de documents permet identificar i explicar cada un dels documents que es pot generar arran d'un projecte informàtic.

Per a cada document incorporat en la gestió d'un projecte concret, (veure [Gestions/Gestió de projectes](#)), es necessari indicar de quin tipus es, (anàlisi funcional, anàlisi orgànic, etc). El tipus de document es defineix aquí.

La base de dades ja ve preparada amb la incorporació dels principals documents de gestió de projectes informàtics.

### 5.2.2 Gestió de responsabilitats

De igual forma que a la secció anterior, un col·laborador usualment exerceix un càrrec. I opcionalment, per un projecte concret, en pot exercir un altre. La descripció i tipologia d'aquests càrrecs es gestiona des de la gestió de responsabilitats.

Per a una responsabilitat concreta podem introduir una descripció del càrrec, i el cost per hora estimat.

La base de dades ja ve preparada amb la incorporació de les principals descripcions de responsabilitat, per a càrrecs en l'àmbit del desenvolupament i gestió de projectes informàtics.



### 5.2.3 Gestió de col·laboradors

La gestió de col·laboradors es una de les parts fonamentals d'aquesta aplicació. Aquí s'introdueixen totes les persones que, d'una forma o altra, col·laboren en el desenvolupament o gestió d'un projecte informàtic.

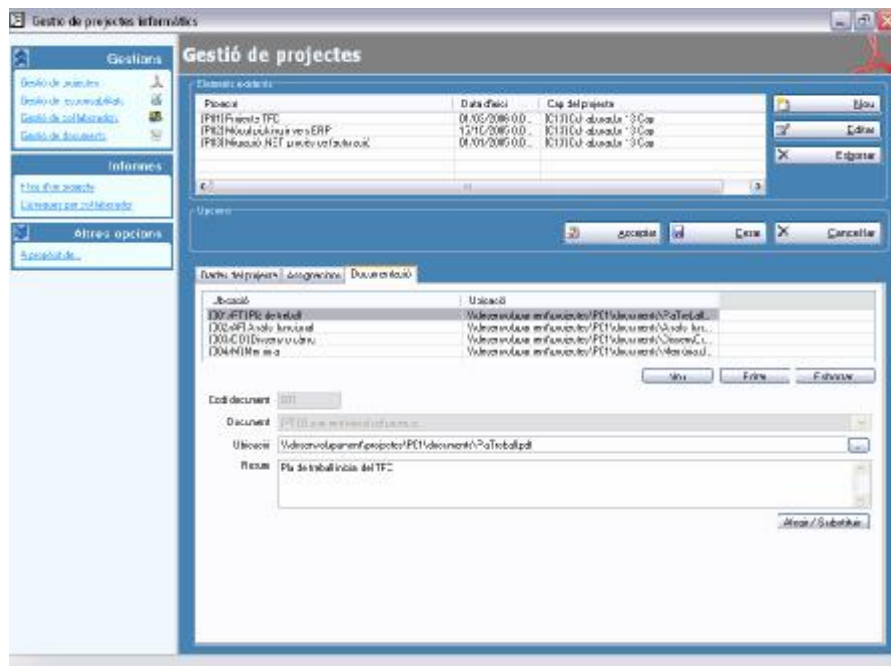
Per a un col·laborador concret podem introduir el seu càrrec per defecte, (veure [Gestions/Gestió de responsabilitats](#)), i la data d'incorporació d'aquest col·laborador.



Només els col·laboradors introduïts en aquesta gestió podran ser assignats a projectes, (veure [Gestions/Gestió de projectes](#)).

## 5.2.4 Gestió de projectes

La gestió de projectes es el manteniment que dona nom a aquesta aplicació. Aquí podem declarar els diferents projectes que la empresa desenvolupa. I, per a cada projecte, podem assignar recursos i documentació.



L'àrea d'entrada de dades d'aquesta gestió es compon de tres pestanyes:

- Dades del projecte
- Assignacions
- Documentació

A la pestanya de *Dades del projecte* trobem les dades generals del projecte, com són:

- Codi i nom del projecte.
- Data d'inici del projecte.
- Descripció del projecte.
- Cap assignat al projecte, (de la llista de col·laboradors).

A la pestanya de *Assignacions* declarem quins col·laboradors han estat assignats al projecte. Per a cada col·laborador que assignem a un projecte haurem d'indicar:

- Rol, (responsabilitat), que el col·laborador exerceix en aquest projecte. No té perquè ser el mateix càrrec que el definit per defecte per al col·laborador.
- Percentatge de dedicació, (valor del 0 al 100), que indica, de la seva dedicació total, quanta dedica a aquest projecte amb aquest rol.

Un col·laborador pot estar assignat diversos cops a un mateix projecte, sempre que el rol que exerceix a cada assignació sigui diferent. Així, un col·laborador pot exercir diversos càrrecs en un mateix projecte.

A la pestanya de *Documentació* indiquem els diferents documents que es van generant en aquest projecte, de quin tipus és, (modalitat), i on és físicament. Així com un resum del document que pugui informar a l'usuari del seu contingut sense necessitat d'obrir-lo.

## 5.3 Informes

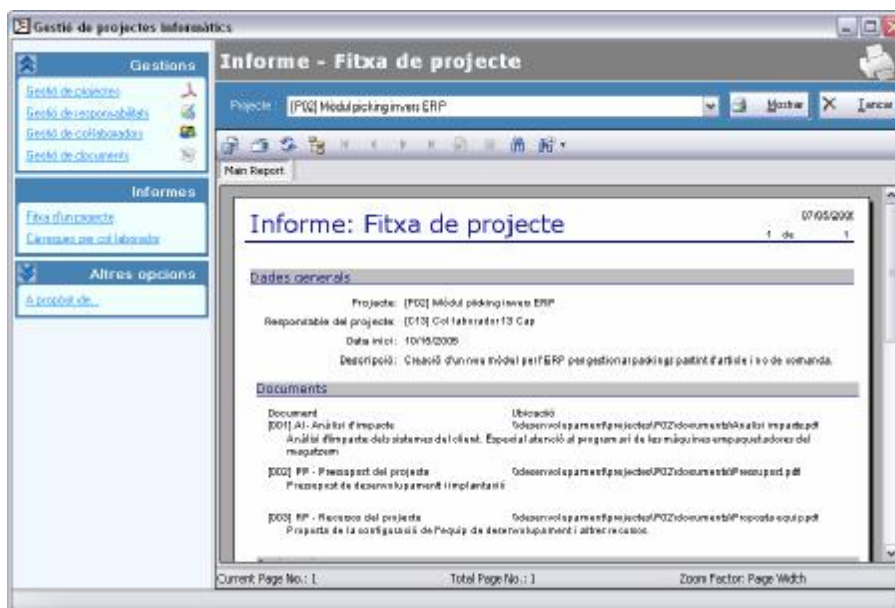
Aquesta aplicació extreu un conjunt d'informes que ajuden a l'usuari en les seves tasques d'administració de l'aplicació. Els informes que extreu l'aplicació són:

- Fitxa de projecte
- Càrregues per col·laborador

Els informes poden imprimir-se per qualsevol impressora accessible des de la màquina on s'executi l'informe; poden exportar-se als formats:

- Word, (.doc)
- Estàndard: Rich text format, (.rtf).
- Full de càlcul Excell, (.xls)
- Formar propietari Crystal Reports, (.rpt)
- Adobe Acrobat Reader, (.pdf)

### 5.3.1 Fitxa de projecte

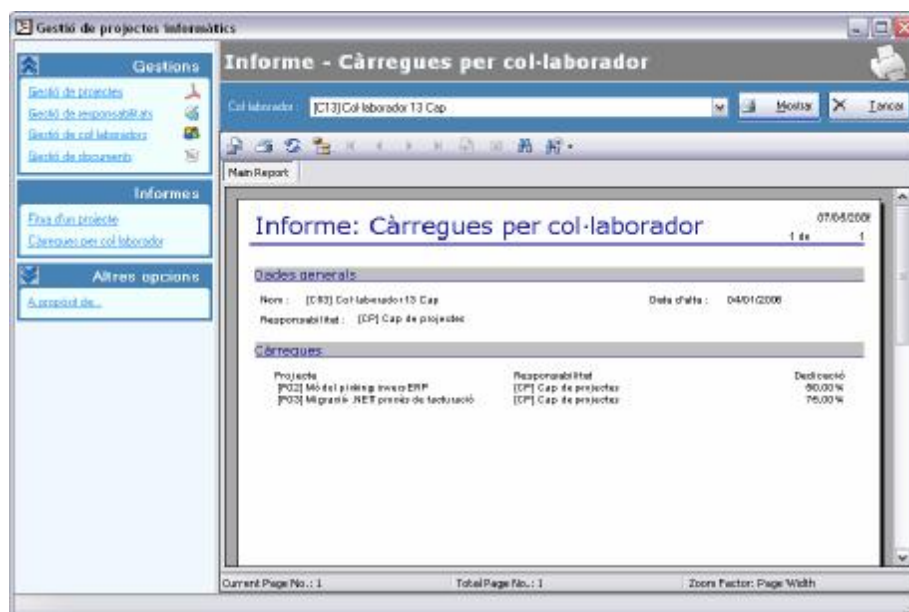


L'informe de *fitxa de projecte* extreu tota la informació disponible d'un projecte concret. L'informe mostra les dades generals del projecte, (data d'inici, descripció, cap de projectes assignat, etc.), els documents generats fins al moment per al projecte, i les assignacions de col·laboradors al projecte.



### 5.3.2 Càrregues per col·laborador

L'informe de càrregues per col·laborador extreu, per al col·laborador seleccionat, a quins projectes està assignat i en quin percentatge de dedicació es troba per a cada projecte.



### 5.4 Altres opcions

A continuació es mostren la resta d'opcions de l'aplicació.

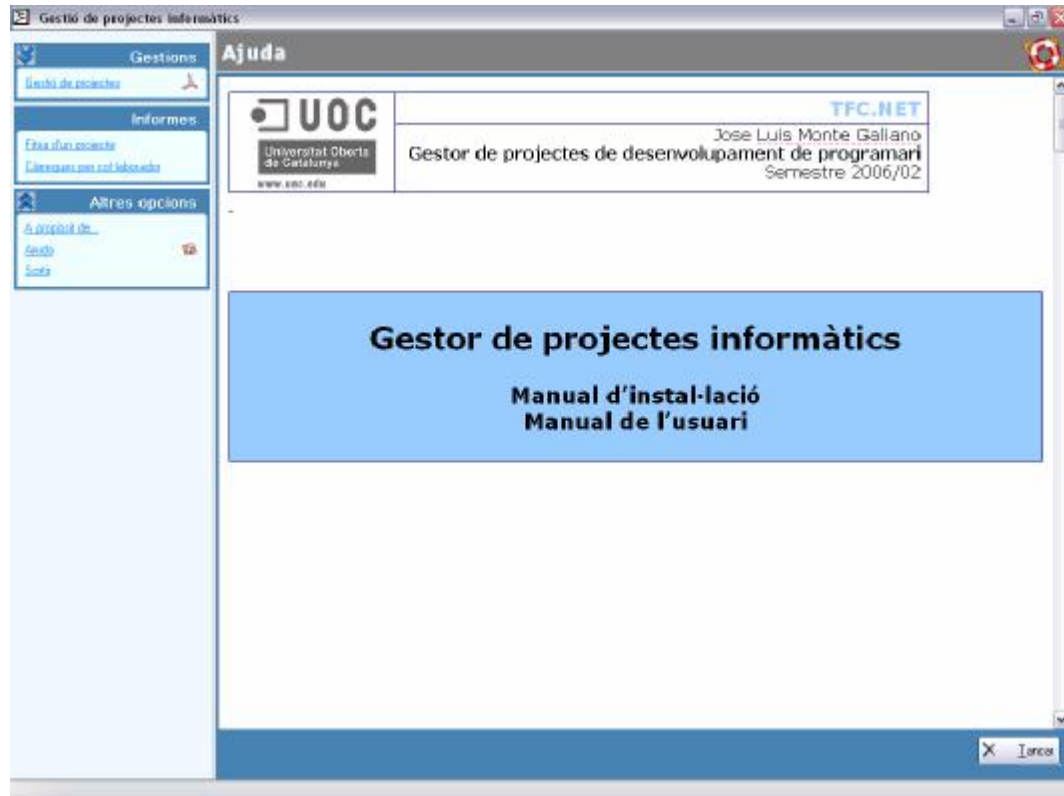
#### 5.4.1 A propòsit de...

Aquesta opció mostra un quadre informatiu de l'aplicació.



#### 5.4.2 Ajuda

Aquesta opció mostra l'ajuda de l'aplicació.



### 5.4.3 Sortir

Aquesta opció tanca l'aplicació.

## 6 Valoració econòmica.

Podem fer una estimació econòmica basant-nos en les variables següents:

- Cost/hora d'un analista: 60€
  - Cost/hora d'un programador: 40€
  - Jornada laboral: 8 hores
  - Dies treballats, (segons calendari d'estimació mostrat anteriorment): 50 dies
- A aquesta última dada s'ha d'afegir:
- Instal·lació: 1 dia
  - Integració: 5 dies
  - Assessorament a usuaris clau: 2 dies

El que fa un total de 58 dies.

El pressupost inicial seria de :

Actuació	Cost/hora	Dies	Total
Anàlisi	50€	5	2.000
Implementació	40€	45	14.400
Instal·lació	30€	1	240
Integració	30€	5	1.200
Assessorament	25€	2	400
<b>Total</b>		<b>58</b>	<b>18.240</b>

Taula 1: Valoració econòmica

## 7 Conclusions.

D'una banda s'han assolit algunes fites de caràcter tecnològic, com son:

- Prendre contacte amb un entorn tecnològic molt nou. Visual Studio 2005 i SQL Server 2005.
- Aprendre en profunditat un nou llenguatge de programació, (C#).
- Desenvolupar aplicacions fent servir el paradigma de la OO i el desenvolupament en  $n$  capes.
- Aprendre a crear i compartir un servei web.
- Aprendre en profunditat l'ús de les tecnologies d'accés a dades, (ADO.NET).

D'altra banda s'han assolit altres fites de caràcter menys tècnic però igualment professional, com son:

- Desenvolupar totes les fases de documentació d'un projecte informàtic
  - Pla de treball
  - Anàlisi funcional
  - Creació de prototips
  - Anàlisi orgànic
- S'ha après que un anàlisi funcional es acurat si i només si hi ha un profund coneixement del negoci i si l'anàlisi de requisits previ es correcte.
- S'ha après que un anàlisi orgànic es acurat si i només si els anàlisis anteriors també ho son.
- S'ha après que l'anàlisi orgànic depèn de les especificacions de l'anàlisi funcional, del grau de coneixements del negoci de l'equip de treball i de la experiència acumulada en projectes anteriors. Per tant, l'anàlisi orgànic generalment no es perfecte. I la implementació ha de saber adaptar-se a aquest fet.

Per últim, des del punt de vista acadèmic, s'han posat en pràctica tots els coneixements adquirits al llarg de la carrera d'ETIG; i el que és més important, s'ha après a interrelacionar i coordinar les diferents àrees de coneixement, per poder assolir el desenvolupament de totes les fases d'un projecte de programari.

## 8 Possibles ampliacions i millores.

Es proposen diverses possibilitats d'ampliació, deixant de banda els possibles requeriments que sorgeixin postimplantació :

### 8.1 Ampliacions de la part servidora, (SW\_TFC).

- **Panell de control del servei web.**  
Creació d'una interfície d'usuari que tingui la funcionalitat de panell de control per al servei web, que fa d'enllaç entre l'aplicació proposada i l'accés a les dades. Aquesta interfície es desenvoluparia en ASP.NET

### 8.2 Ampliacions de la part client, (GestioProjectes).

- **Gestió de planificació d'assignacions.**  
Dates d'inici i fi de les diferents assignacions de recursos a projectes. Percentatges de dedicació de recursos.
- **Indexació de cerques en documents assignats.**  
Creació d'un motor de cerques que permeti localitzar informacions tant a la base de dades, com als documents que han estat assignats a projectes. De forma que es relacionin diferents projectes a partir d'una cerca.
- **Gestió d'usuaris avançada.**  
Gestió i control dels usuaris de l'aplicació. Altes i baixes d'usuari; canvis de contrasenya; control de les activitats dels usuaris a l'aplicació.
- **Gestió d'incidències.**  
Mòdul per la gestió d'incidències derivades dels projectes que es gestionen des de l'aplicació.
- **Gestió d'evolutius.**  
Mòdul per la gestió de les actuacions d'ampliació de funcionalitats del programari de la empresa.
- **Calendari de projectes i control de l'estat dels projectes.**  
Control de les tasques que s'estan duent a terme per assolir cada projecte. En quina situació es troben aquestes tasques i qui les està fent.
- **Alertes**  
Gestió d'alertes al cap de projecte i als components dels projectes. Sobre quan comença i acaba cada tasca assignada. Així com alertar si alguna fita no s'acompleix segons el calendari.

## 9 Glossari.

**Capa de Dades:** Conjunt de classes en les que s'implementa tot el relatiu a l'accés a les dades, (seleccions, actualitzacions, etc), de peticions provinents de la capa de domini.

**Capa de Domini:** Conjunt de classes en les que s'implementa per una banda la gestió lògica de les dades, i per l'altra la operativa del negoci en que l'aplicació treballa.

**Capa de Presentació:** Conjunt de dades dedicades a mostrar la informació a l'usuari, i a interactuar amb ell.

**Capes:** Separació modular d'un projecte de desenvolupament de programari. On cada capa realitza unes funcions específiques, i on totes juntes conformen una solució informàtica.

**Classes del domini:** Un subconjunt de classes de la capa de domini especialitzades en la lògica corporativa concreta de la solució informàtica, així com contenir les dades que seran mostrades en la capa de presentació o seran enviades a la capa de dades.

**Contrasenya:** Clau d'accés necessària per poder accedir al servei web que alimenta de dades l'aplicació.

**Gestió de col·laboradors:** Gestió de les dades de les persones que poden ser assignades a projectes.

**Gestió de documents:** Gestió de descripcions de documents. Tot document assignat a un projecte, li ha de correspondre una descripció de document.

**Gestió de projectes:** Gestió base de l'aplicació. Magatzem dels projectes de la empresa. Cada projecte està compost d'unes dades descriptives, així com les assignacions de col·laboradors i documents.

**Gestió de responsabilitats:** Cada col·laborador té un rol que el caracteritza. La gestió de responsabilitats gestiona aquests rols.

**Gestors del domini:** Un subconjunt de classes de la capa de domini especialitzades en la comunicació de les classes del domini amb la capa d'accés a dades.

**Gestors de formularis:** Un subconjunt de classes de la capa de domini especialitzades en l'abastiment de dades a la capa de presentació per ser mostrades a l'usuari, així com en la recepció dels events produïts a la capa de presentació, producte de peticions i interaccions de l'usuari.

**Microsoft Framework .NET v.2.0:** Programari requerit que conforma la màquina virtual sobre la que s'executa l'aplicació.

**Microsoft Internet Information Server, (IIS):** Programari requerit sobre el que s'executa el servei web, necessari per al funcionament de l'aplicació.

**MVC – Model Vista Controlador:** Correspon al model de descomposició per capes més popular avui dia, on coexisteixen tres entitats anomenades Model, (gestió de les dades i les operacions de lògica corporativa), Vista, (la capa de presentació a l'usuari), i Controlador, (la capa de gestió dels events i peticions produïts a l'entitat Vista).

**Nom d'usuari:** Paraula d'accés necessària i que, juntament amb una contrasenya, permetran iniciar sessió a l'aplicació.

**Part client:** Espai d'execució de la part de l'aplicació en la que s'inclou la interfície de comunicació amb l'usuari.

**Part servidor:** Espai d'execució on s'executa el servei web i el gestor de dades de l'aplicació.

**Perfil d'usuari:** Conjunt d'informació que caracteritza un usuari de l'aplicació.

**SGBD:** Programari necessari on s'emmagatzemen les dades generades per l'aplicació.

**Servei web:** Part de l'aplicació que s'executa de forma independent i que, en aquest cas, s'encarrega d'alimentar de dades l'aplicació.

## 10 Bibliografia.

### 10.1 Sobre productes.

#### Requeriments d'instal·lació de Visual Studio 2005

Al CD d'instal·lació, (CD1), hi ha un arxiu amb aquesta informació.

#### Requeriments d'instal·lació de Ms. Sql Server 2005

<http://www.microsoft.com/spain/sql/productinfo/sysreqs/default.msp>

### 10.2 Sobre llenguatges i programació.

#### Profesional C# 2a edició.

Simon Robinson

Editorial Danypress

ISBN: 8493272027

[http://www.danyshop.com/shop/product.asp?catalog\\_name=Libros&product\\_id=3-00028&ProfesionalC&cookie%5Ftest=1&cookie%5Ftest=1](http://www.danyshop.com/shop/product.asp?catalog_name=Libros&product_id=3-00028&ProfesionalC&cookie%5Ftest=1&cookie%5Ftest=1)

#### Patrones de diseño

Erich Gamma, Richard Helm, Ralph Johnson i John Vlissides

Editorial Pearson / Addison Wesley

ISBN: 84-7829-059-1

**UML y patrones, (2a edició)**

Larman, Graig  
Pearson Educación  
ISBN: 8420534382

**MSDN Spain**

<http://www.microsoft.com/spain/>

**Visual Studio 2005**

<http://www.microsoft.com/spanish/msdn/vs2005/default.aspx>

**Patrones y antipatrones, (II):**

[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/mtj\\_3317.asp#M19](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/mtj_3317.asp#M19)

**Desarrollo de aplicaciones en n-capas con .NET**

<http://www.microsoft.com/spanish/msdn/spain/eventos/descargas/corporate/DES21DesarrollodeAplicacionesenNcapas.ppt>

**elGuille – Aplicaciones n-capas en VB.NET**

[http://www.elguille.info/colabora/NET2005/FernandoLuque\\_NCapas.htm](http://www.elguille.info/colabora/NET2005/FernandoLuque_NCapas.htm)

y

[http://www.elguille.info/colabora/NET2005/Tomasmm\\_3Capas2.htm](http://www.elguille.info/colabora/NET2005/Tomasmm_3Capas2.htm)

**Definició de MVC, (Model Vista Controlador):**

<http://es.wikipedia.org/wiki/MVC>

**Model View Controller History**

<http://c2.com/cgi/wiki?ModelViewControllerHistory>

## 10.3 Exemples de programari de gestió de projectes.

**KmKey**

Versió demostrativa d'un gestor de projectes.

[http://www.kmkey.com/productos/kmkey\\_project](http://www.kmkey.com/productos/kmkey_project)