

TFC.NET - Memòria

Narcís Patiño Gispert

Enginyeria Tècnica en Informàtica de Gestió

Consultor: Jordi Ceballos Villach

9 de gener de 2008

Índex

Índex	2
1. Introducció	5
1.1. Definició del projecte	5
1.2. Justificació del projecte	5
1.3. Descripció del projecte	6
1.4. Objectius	7
1.5. Planificació	7
1.5.1. Cicle de vida	7
1.5.2. Detall d'activitats	8
1.5.2.1. Pla de treball	8
1.5.2.2. Anàlisi de requeriments	8
1.5.2.3. Disseny	9
1.5.2.4. Implementació y proves	9
1.5.2.5. Finalització	10
1.5.3. Temporització del projecte	10
1.6. Programari utilitzat	13
1.7. Productes obtinguts	13
1.8. Estructura d'aquest document	13
2. Requeriments inicials	14
2.1. Escenari de partida	14
2.2. Usuaris	14
2.3. Requeriments funcionals	14
2.3.1. Funcionalitats del subsistema gestió d'àlbums	14
2.3.2. Funcionalitats del subsistema gestió imatges directori.	15
2.3.3. Funcionalitats del subsistema gestió imatges àlbum	15
2.4. Requeriments no funcionals	16
2.4.1. Requeriments d'interfície	16
3. Anàlisi del sistema	17
3.1. Diagrames de casos d'ús	17
3.1.1. Taula de casos d'ús	19
3.1.2. Descripció textual dels casos d'ús	21
GA02-Crear àlbum	21
GA03-Modificar àlbum	21
GA04-Esborrar àlbum	22
GA05-Llistar àlbums	23

GA06-Seleccionar àlbum	23
GI03-Modificar imatge de l'àlbum	24
GI04-Esborrar imatge de l'àlbum	25
GI05-Publicar imatges a Internet	25
GI06-Seleccionar imatge de l'àlbum	26
GD03-Modificar imatge directori	27
GD04-Esborrar imatge directori	27
GD05-Renombrar imatge directori	28
GD06-Seleccionar imatge directori	28
GD07-Copiar imatge directori	29
GD08-Moure imatge directori	30
GD09-Seleccionar directori	30
GD10-Enviar imatges a l'àlbum	31
C01-Afegir imatge cistella	32
C02-Treure imatge cistella	32
EI02-Exportar imatge	33
EI03-Imprimir Imatge	33
EI04-Següència d'imatges	34
EI05-Aplicar eina	34
3.2. Model del domini	36
4.Disseny	37
4.1. Arquitectura de l'aplicació	37
4.1.1. Capa de presentació	38
4.1.2. Capa de negoci	38
4.1.3. Capa de dades	38
4.1.4. Components	39
4.2. Arquitectura WPF	41
4.2.1. Element System	42
4.2.2. Visual System	42
4.2.3. Font System	42
4.2.4. Input/Event System	43
4.2.5. Property System	43
4.2.6. Message Transport System	44
4.2.7. Composition System	44
4.3. Decisions tecnològiques	45
4.4. Diagrama estàtic de disseny	45
4.5. Disseny de la persistència	47

4.5.1. Model relacional de la base de dades	47
4.5.2. Diagrama de la base de dades	47
4.6. Prototipus de la interfície d'usuari	48
5. Implementació	50
6. Conclusions	51
7. Glossari	52
8. Bibliografia	54

1. Introducció

1.1. Definició del projecte

Aquest Treball de Fi de Carrera es centra en l'àrea temàtica .NET, la qual consisteix en què l'estudiant s'introdueixi i aprofundeixi en aquest nou marc de desenvolupament de software, fent servir algunes de les seves tecnologies més importants: ADO.NET, WPF, Serveis Web, etc.

El projecte consisteix en desenvolupar una aplicació fent servir les tecnologies de Microsoft .NET. Es podrà escollir un dels dos escenaris d'aplicació següents:

- Desenvolupar una aplicació per a gestionar llistes jeràrquiques, que permeti anotar la llista de la compra, les tasques que tenim pendents de fer, etc. Un exemple d'aplicació d'aquest tipus és TreNotes, disponible a la següent pàgina Web: <http://www.fannsoftware.com/TreNotes.html>
- Desenvolupar un catalogador i/o editor d'imatges. Un exemple d'aplicació d'aquest tipus és Picasa i PicasaWeb, de Google.

D'aquests dos escenaris s'ha escollit el desenvolupament d'un catalogador i editor d'imatges.

No és possible realitzar cap treball de temàtica lliure. Les funcionalitats concretes a desenvolupar dins de l'escenari escollit, les proposarà l'estudiant al principi de curs i es pactaran amb el consultor corresponent. Les aplicacions proposades com a exemple (TreeNotes i Picasa) poden servir per a agafar idees, però l'estudiant té llibertat per a proposar quines característiques tindrà la seva aplicació, i com serà la interfície gràfica.

En quant a la interfície de l'aplicació, l'estudiant pot escollir una entre les diferents que ofereix la plataforma .NET (client WinForms, pàgina ASP.NET, client embedded, etc.), o crear-ne més d'una.

També pot resultar interessant fer servir altres tecnologies com Serveis Web, XML, .NET Remoting, etc.

L'aplicació haurà de fer ús de bases de dades relacionals SQL Server 2005, i per tant de la tecnologia ADO.NET. Independentment del projecte i tipus d'interfície escollit, serà obligatori fer servir en el desenvolupament del projecte alguna de les següents tecnologies:

- AJAX
- WPF (Windows Presentation Foundation) o WPF/E (WPF EveryWhere)
- Windows Mobile 6

Aspectes que es valoraran positivament són la diversitat de tecnologies emprades, una bona interfície d'usuari, la utilització de controls o mòduls externs (sempre i quan siguin "open-source"), la generació d'informes, gràfics, estadístiques, etc.

1.2. Justificació del projecte

Actualment, les imatges digitals estan cada cop més esteses tant en l'àmbit professional com el domèstic. A moltes llars es disposa de càmeres digitals o altres dispositius que generen gran quantitat d'imatges, que si no es cataloguen d'alguna manera, queden

perdues enmig d'un mar d'arxius.

Per una banda, en el mercat d'aplicacions informàtiques trobem els anomenats visors d'imatges, alguns dels quals contemplen la possibilitat de gestionar el que s'anomenen àlbums de fotos per a la seva catalogació. Aquests programes solen tenir una bona interfície d'usuari i el seu funcionament és senzill i intuïtiu però no solen aportar cap més funció que la catalogació.

Per altra banda, els programes d'edició fotogràfica com el Photoshop tenen un funcionament complex i un llarg aprenentatge d'ús tot i que sovint només ens calen utilitzar unes poques de les seves funcions.

Els usuaris demanen aplicacions que combinin tant la catalogació d'imatges com les funcions d'edició digital més freqüents, juntament amb una interfície d'usuari amb disseny modern, intuïtiva i de ràpid aprenentatge. Per omplir aquest buit, han sorgit programes com el Picasa de Google que combina tots aquests requeriments.

Per incrementar l'oferta d'aquest segment de programari, es pretén desenvolupar una aplicació d'aquest tipus, per catalogar i editar d'imatges dins del marc de la tecnologia .NET de Microsoft. El projecte contemplarà totes les fases del cicle de vida d'un projecte informàtic.

1.3. Descripció del projecte

Aquest treball té per finalitat l'anàlisi, disseny i implementació d'una aplicació de catalogació i edició d'imatges, sobre la plataforma de desenvolupament .NET de Microsoft i utilitzant diverses de les tecnologies que incorpora.

L'aplicació conté tres subsistemes relacionats entre ells:

1. **Subsistema gestió d'àlbums:** En certa manera és el nucli de l'aplicació, ja que els àlbums contendran imatges i per enviar imatges des d'un directori a un àlbum caldrà haver seleccionat prèviament l'àlbum corresponent. La informació dels àlbums es desa en una base de dades per a la seva gestió.
2. **Subsistema gestió imatges directori:** L'aplicació visualitza miniatures (thumbnails) de les imatges de les carpetes per on vagi navegant l'usuari. Disposa de diverses funcions de tractament de fitxers. També pot seleccionar imatges per enviar-les a l'àlbum seleccionat prèviament.
3. **Subsistema gestió imatges àlbum:** Des d'aquest subsistema es gestionen les imatges d'un àlbum seleccionat prèviament. El sistema visualitza miniatures (thumbnails) de les imatges de l'àlbum i permet realitzar diverses funcions. També es poden seleccionar imatges per publicar-les a Internet. Les imatges de l'àlbum es desen en una base de dades per a la seva gestió.

Els subsistemes de gestió d'imatges comparteixen la funcionalitat d'edició d'imatges. L'usuari pot realitzar diverses funcions de tractament d'imatges i posteriorment desar els canvis realitzats. També hi ha la possibilitat d'exportar les imatges a d'altres formats gràfics o imprimir-les.

1.4. Objectius

La realització d'aquest treball suposa un valor afegit en quant a adquisició de coneixements i experiència en un nou entorn de desenvolupament cada cop més present en el mercat laboral. Els objectius del qual són:

- L'objectiu principal del projecte és fer servir els coneixements adquirits al llarg de la carrera.
- Utilitzar algunes de les tecnologies més importants que proporciona la plataforma .NET de Microsoft per a desenvolupar una aplicació de dificultat mitjana.

1.5. Planificació

1.5.1. Cicle de vida

El projecte s'ha desenvolupat seguint una variació del cicle de vida clàssic de creació de programari, lleugerament adaptat als requeriments del TFC i està compost per aquestes etapes:

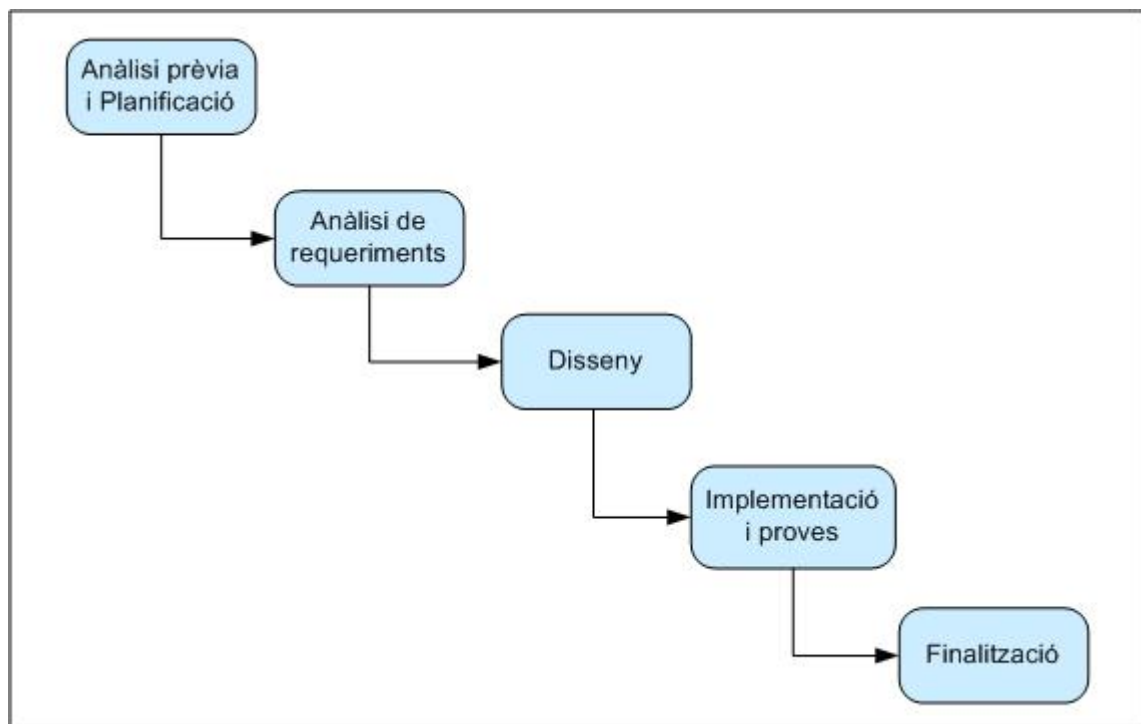


Figura 1: Cicle de vida en cascada

La documentació que s'ha generat en cada fase, ha sigut lliurada en els terminis establerts a cada fita del pla de treball i els seus continguts s'han resumit en aquesta memòria.

1.5.2. Detall d'activitats

Les següents subseccions recullen les principals activitats realitzades per a cada etapa del cicle de vida, amb la finalitat d'obtenir un producte que compleixi els requeriments establerts, amb la qualitat desitjada i dins del plaç donat.

1.5.2.1. Pla de treball

En aquesta etapa s'ha realitzat una descripció del projecte que es vol realitzar, així com un anàlisi de requeriments detallat de les funcionalitats de l'aplicació a desenvolupar. Al pla de treball s'han indicat també les tecnologies concretes que es faran servir, i la planificació estimada de les diferents tasques a realitzar per dur a terme el projecte.

Activitat	Descripció
Selecció del projecte	Selecció del projecte entre els dos escenaris possibles.
Preparació del projecte	Lectura de materials, descàrrega de software
Definició del projecte	Conceptualització del projecte a realitzar, a nivell de requeriments i tecnologies a utilitzar.
Planificació del projecte	Temporització del projecte conforme a la metodologia a utilitzar i a les fites de l'assignatura.
Elaboració del document	Elaboració del document corresponent al primer lliurament amb el material anterior.
Lliurament de la PEC1	Lliurament del document corresponent a aquesta etapa.

El consultor ha validat aquest document, puntualitzant-ne alguns detalls.

1.5.2.2. Anàlisi de requeriments

Durant aquesta fase s'ha realitzat un anàlisi de requeriments detallat de les funcionalitats de l'aplicació a desenvolupar.

Les activitats realitzades van ser:

Activitat	Descripció
Especificació de requeriments funcionals	Elaboració i documentació de casos d'ús.
Especificació de requeriments no funcionals	Identificació i documentació de requeriments no funcionals.

1.5.2.3. Disseny

Aquesta etapa defineix com cal que s'implementi el sistema per cobrir les necessitats i objectius plantejats en les fases anteriors, incloent aspectes sobre les tecnologies a utilitzar.

En aquesta fase caldrà realitzar l'especificació formal i el disseny de l'aplicació. No es demana un nivell de detall elevat, més aviat es demana que l'alumne demostrï que ha adquirit els coneixements necessaris per especificar i dissenyar un projecte abans de passar a la implementació.

Les activitats realitzades van ser:

Activitat	Descripció
Disseny arquitectura	Disseny de l'arquitectura de la solució
Disseny de classes	Creació del diagrama estàtic de disseny
Disseny base de dades	Definició de l'estructura de la base de dades
Maqueta gràfica (prototip)	Elaboració del prototipus no funcional de l'aplicació que bàsicament correspon a la interfície d'usuari.
Estudi d'alternatives tecnològiques	Estudi de diferents tecnologies disponibles per a la implementació de determinats components (patrons, persistència, etc.), sempre dins del marc actual.
Elaboració del document	Recopilació d'informació y preparació del document d'anàlisi i disseny de l'aplicació.
Lliurament PEC2	Lliurament de la PEC2.

1.5.2.4. Implementació y proves

Durant aquesta fase s'ha realitzat la implementació de l'aplicació, tenint en compte les especificacions definides fins al moment. També s'han realitzat les proves necessàries per testejar la correcció de la solució.

Les activitats realitzades són les següents:

Activitat	Descripció
Desenvolupament de l'aplicació	Implementació del sistema, que correspon als components següents: <ul style="list-style-type: none">• Interfície d'usuari• Base de dades• Components d'accés a dades• Entitats• Lògica de negoci• Proves
Preparació del paquet a	Elaboració del manual d'instal·lació i creació del

lliurar	paquet amb el producte.
Lliurament de la PEC3	Lliurament de la PEC3 del treball.

1.5.2.5. Finalització

En un projecte de programari aquesta etapa no existeix com a tal, tanmateix, en aquest cas s'ha utilitzat per elaborar la documentació final del treball de fi de carrera i per coincidir amb el lliurament final. Inclou les activitats següents:

Activitat	Descripció
Redacció memòria final	Recopilació de documents y elaboració de la memòria final.
Creació presentació virtual	Creació de la presentació virtual
Producte final	Finalització de la implementació i de la elaboració del manual d'instal·lació.
Lliurament final	Lliurament de documentació i del producte final.

1.5.3. Temporització del projecte

Aquesta és la temporització inicial prevista del projecte. Les dates més rellevants estan en negreta.

tasca	Durada	Inici	Fi	Explicació tasca
Inici projecte		18/09/2007	18/09/2007	
Estudi de l'entorn .NET i WPF	25 dies	18/09/2007	22/10/2007	Estudi de les tecnologies que componen l'entorn del projecte
Instal·lació del software	11 dies	18/09/2007	02/10/2007	Instal·lació del software necessari per al desenvolupament
Recopilació informació projecte	69 dies	18/09/2007	21/12/2007	Recopilació d'informació respecte a la memòria final
Pla de treball (PAC1)	11 dies	18/09/2007	02/10/2007	PAC1
Descripció projecte/objectius	2 dies	18/09/2007	19/09/2007	Descripció del projecte i objectius que es pretenen.
Requeriments funcionals	2 dies	20/09/2007	21/09/2007	Funcionalitats del projecte
Requeriments tècnics	2 dies	24/09/2007	25/09/2007	Tecnologia del projecte i software previst que s'utilitzarà
Planificació temporal (Gantt)	3 dies	26/09/2007	28/09/2007	Diagrama de Gantt i establir la temporització del projecte.

Document pla de treball	2 dies	01/10/2007	02/10/2007	Elaboració document a entregar amb tota la documentació anterior.
Entrega PAC1 (02/10/07)	fita	02/10/2007	02/10/2007	Entrega document pla de treball
Anàlisi i Disseny (PAC2)	14 dies	03/10/2007	22/10/2007	PAC2
Identificació de classes	2 dies	03/10/2007	04/10/2007	Identificació de les classes que intervenen en el projecte.
Anàlisi	3 dies	05/10/2007	09/10/2007	Diagrames de casos d'us.
Disseny de classes	2 dies	10/10/2007	11/10/2007	Diagrama de classes
Disseny arquitectura	2 dies	12/10/2007	15/10/2007	Disseny de l'arquitectura
Disseny base de dades	1 dies	16/10/2007	16/10/2007	Disseny de la persistència.
Maqueta gràfica (prototip)	2 dies	17/10/2007	18/10/2007	Realització del prototip de l'aplicació.
Documents a entregar	2 dies	19/10/2007	22/10/2007	Elaboració dels documents a entregar amb la documentació anterior.
Entrega PAC2 (22/10/07)	fita	22/10/2007	22/10/2007	Entrega dels documents i el prototip.
Implementació (PAC3)	44 dies	23/10/2007	21/12/2007	PAC3
Base de dades	1 dies	23/10/2007	23/10/2007	Creació i proves de la base de dades.
Interfície d'usuari	8 dies	24/10/2007	02/11/2007	Implementació en XAML de la interfície d'usuari.
Proves	2 dies	05/11/2007	06/11/2007	Proves interfície d'usuari
Part catalogador d'imatges	14 dies	07/11/2007	26/11/2007	Implementació part catalogador d'imatges en C#.
Proves	2 dies	27/11/2007	28/11/2007	Proves de la part implementada fins aquest moment.
Part edició d'imatges	14 dies	29/11/2007	18/12/2007	Implementació part edició d'imatges.
Proves	2 dies	19/12/2007	20/12/2007	Proves de tota l'aplicació.
Manual d'instal·lació	1 dies	21/12/2007	21/12/2007	Elaboració manual d'instal·lació.
Entrega PAC3 (21/12/07)	fita	21/12/2007	21/12/2007	Entrega de l'aplicació i del manual d'instal·lació.
Memòria i presentació (PPT)	14 dies	22/12/2007	09/01/2008	PAC4
Memòria	9 dies	22/12/2007	02/01/2008	Elaboració de la memòria de final de projecte.
Presentació virtual (PPT)	3 dies	03/01/2008	07/01/2008	Elaboració de la presentació virtual amb PowerPoint.
Implementació definitiva	2 dies	08/01/2008	09/01/2008	Últims retocs a l'aplicació.
Entrega final (09/01/08)	fita	09/01/2008	09/01/2008	Entrega final de la memòria, la presentació i l'aplicació definitiva.
Debat virtual	5 dies	21/01/2008	25/01/2008	
Fi projecte		25/01/2008	25/01/2008	

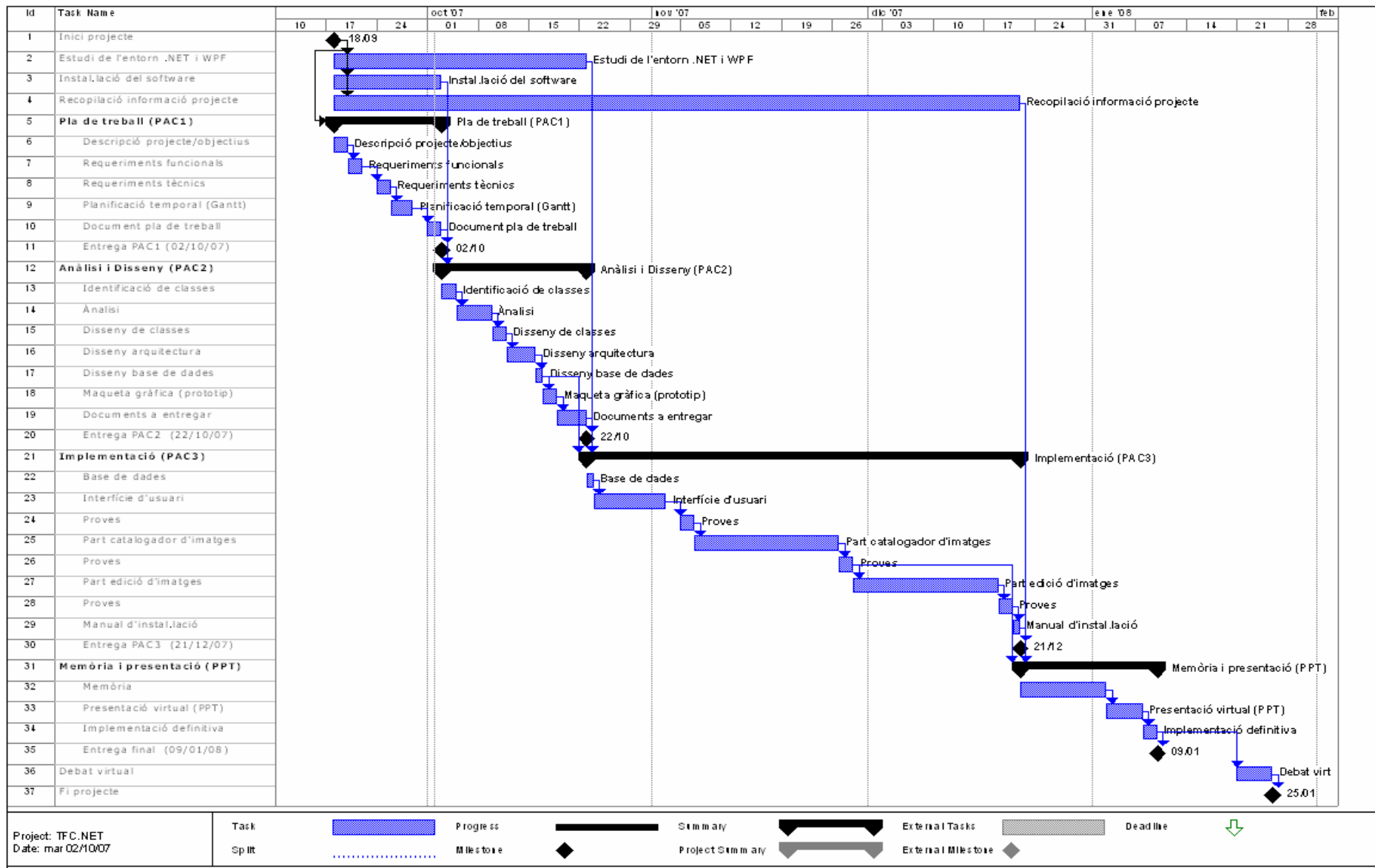


Figura 2: Diagrama de Gantt temporització del projecte

1.6. Programari utilitzat

El programari utilitzat per a la realització del projecte és el següent:

- MagicDraw UML per als diagrames UML.
- Microsoft Visual Studio 2008 beta 2 per a l'entorn de desenvolupament integrat de la plataforma .NET.
- Microsoft XAMLPad i Microsoft Expression Blend per al disseny de la interfície d'usuari i el prototipus en WPF i les diverses proves amb el llenguatge XAML.
- Microsoft SQL Server 2005 Express SP2 com a SGBD
- Microsoft SQL Server Management Studio per al disseny de la base de dades.
- Microsoft Project 2003 per a la planificació del projecte i el diagrama de Gantt.
- Microsoft Visio 2003 per als diagrames.
- Microsoft PowerPoint per a la presentació del projecte.
- Microsoft Word i Excel per a l'elaboració de la documentació.

1.7. Productes obtinguts

Durant la realització del projecte s'han lliurat els següents materials:

- Document del Pla de Treball.
- Documents d'Anàlisi i Disseny del sistema.
- Prototipus del sistema.
- Sistema **Photo.Net**, compostat pel projecte Projecte.NET
- Manual d'instal·lació del sistema.
- Memòria final del projecte.
- Presentació virtual.

1.8. Estructura d'aquest document

A continuació, aquest document recull els aspectes més importants de les diferents tasques realitzades. Es comença pels requeriments inicials, els quals descriuen l'escenari del projecte.

Seguidament es comenta la fase d'anàlisi, detallant els requeriments funcionals a través dels casos d'ús. També es descriuen aspectes no funcionals i un model del domini.

Segueix amb el disseny d'arquitectura del sistema, les tecnologies emprades i el diagrama estàtic de disseny.

En els últims apartats es comenta alguns detalls de la implementació, les conclusions finals del treball i per últim el glossari i els materials utilitzats més destacables.

2. Requeriments inicials

A continuació es descriu l'escenari sobre el que es desenvolupa aquest treball, així com els requeriments inicials necessaris.

2.1. Escenari de partida

A partir del que s'ha comentat en els apartats 1.2 i 1.3 de justificació i descripció del projecte els requeriments inicials d'aquest projecte són:

- La interfície d'usuari (IU) ha de ser rica, gràfica i agradable a l'usuari.
- La IU ha de tenir en compte els dispositius tàctils, òptics i altres.
- La IU ha de tenir un autoaprenentatge ràpid per part de l'usuari.
- La IU ha de ser usable permetent realitzar les accions amb pocs hits.
- L'aplicació ha de ser accessible, no només amb els colors, també cal utilitzar tooltips i altres mecanismes per a una bona accessibilitat procurant seguir les recomanacions del W3C.
- Ha de proporcionar eines de gestió de fitxers
- Ha de proporcionar eines d'edició d'imatges.
- Ha de permetre l'emmagatzematge de dades.

2.2. Usuaris

Cal fer notar que només hi ha un tipus d'usuari, donat que les restriccions de seguretat s'han considerat que són suficients amb les que proporciona el sistema on s'instal·larà l'aplicació, restringint els accessos a nivell de seguretat de base de dades, a nivell de seguretat de directori i de polítiques de sistema.

2.3. Requeriments funcionals

Es comenten els requeriments funcionals seguint la classificació en subsistemes establerta en l'apartat 1.3.

2.3.1. Funcionalitats del subsistema gestió d'àlbums

Aquest subsistema contempla aquests requeriments funcionals.

- Alta d'un àlbum: Permet introduir un nou àlbum al sistema omplint les dades corresponents.

- Baixa d'un àlbum: Permet esborrar un àlbum i les seves imatges relacionades del sistema.
- Modificació d'un àlbum: Permet modificar les dades d'un àlbum.
- Llistat d'àlbums: Permet un llistat per pantalla dels àlbums. Aquest llistat també es pot exportar a Excel o a PDF.
- Accés a la gestió d'imatges àlbum i directori: A través de les icones corresponents s'accedirà a aquestes finestres.
- Possibilitat de canviar a colors més accessibles: A través de la icona d'accessibilitat es canviaran els colors de l'aplicació.

2.3.2. Funcionalitats del subsistema gestió imatges directori.

Aquest subsistema contempla aquests requeriments funcionals. Tots ells s'executen al clicar el botó corresponent.

- Funcions de tractament de fitxers: esborrar, copiar, moure, canviar de directori.
- Possibilitat de modificar la imatge i guardar els canvis.
- Possibilitat d'afegir la imatge seleccionada a la cistella d'imatges.
- Possibilitat de treure la imatge seleccionada de la cistella d'imatges.
- Possibilitat d'enviar les imatges de la cistella a l'àlbum seleccionat prèviament.
- Presentació de les imatges del directori. Recorregut seqüencial per les imatges del directori actual.
- Exportació a altres formats gràfics.
- Impressió de la imatge seleccionada.
- Visualització metadades Exif associades a la imatge.
- Diverses funcions d'edició d'imatges.

2.3.3. Funcionalitats del subsistema gestió imatges àlbum

Aquest subsistema contempla aquests requeriments funcionals. Tots ells s'executen al clicar el botó corresponent.

- Funcions de tractament de fitxers: esborrar, copiar, moure, canviar de directori.
- Modificació d'una imatge.
- Baixa d'una imatge.
- Possibilitat d'afegir la imatge seleccionada a la cistella d'imatges.
- Possibilitat de treure la imatge seleccionada de la cistella d'imatges.
- Possibilitat d'enviar les imatges de la cistella a l'àlbum seleccionat prèviament.

- Presentació de les imatges del directori. Recorregut seqüencial per les imatges del directori actual.
- Exportació a altres formats gràfics.
- Impressió de la imatge seleccionada.
- Visualització metadades Exif associades a la imatge.
- Diverses funcions d'edició d'imatges.

2.4. Requeriments no funcionals

2.4.1. Requeriments d'interfície

- En aplicació dels requeriments inicials, la interfície es desenvoluparà procurant seguir les recomanacions d'usabilitat i accessibilitat.
- Per poder permetre una àmplia gamma de dispositius òptics, tàctils i altres a més dels tradicionals, el front-end es realitzarà bàsicament amb icones.
- Es desenvoluparà amb la tecnologia WPF i es dissenyarà amb el llenguatge enriquit XAML, el qual està pensat per a dissenyar interfícies d'usuari i que reuneix tots els requeriments necessaris.

3.Anàlisi del sistema

Aquest apartat mostra la fase d'anàlisi del sistema. Es mostren els casos d'ús més importants i a continuació la seva descripció formal detallada.

3.1. Diagrames de casos d'ús

Aquest diagrama mostra els casos d'ús agrupats en 3 subsistemes que corresponen a les seves funcionalitats principals i en els altres diagrames es descomponen detalladament amb una granularitat més alta. Aquestes 3 subsistemes en la fase d'implementació correspondran a les 3 finestres principals.

Només hi ha un tipus d'actor: **Usuari**.

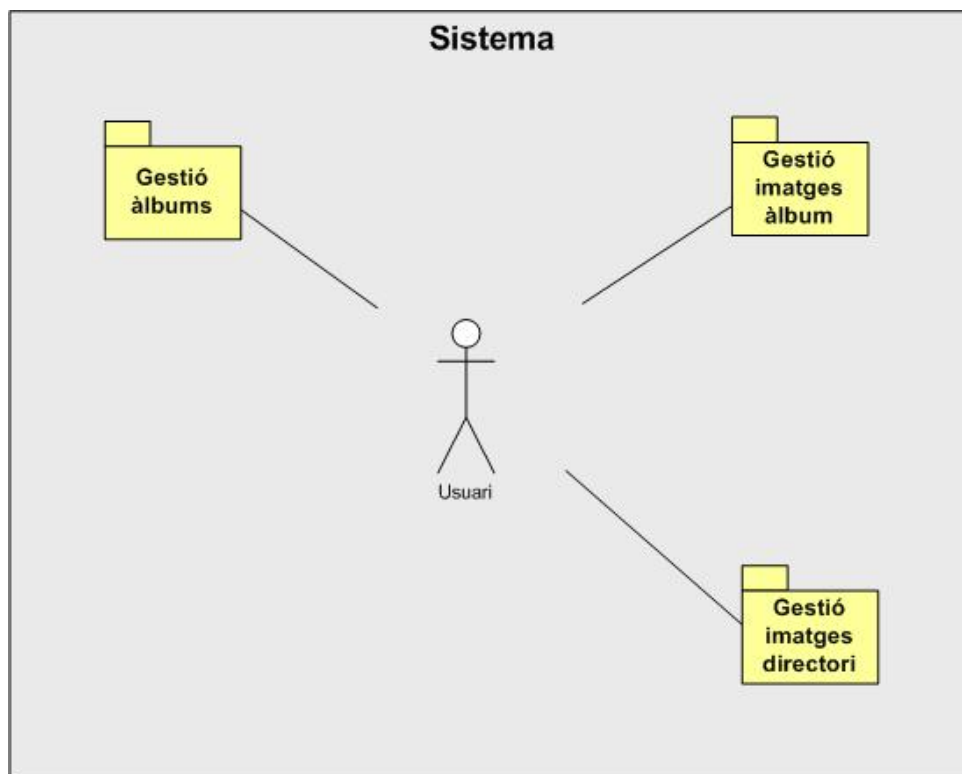


Figura 3: Sistema

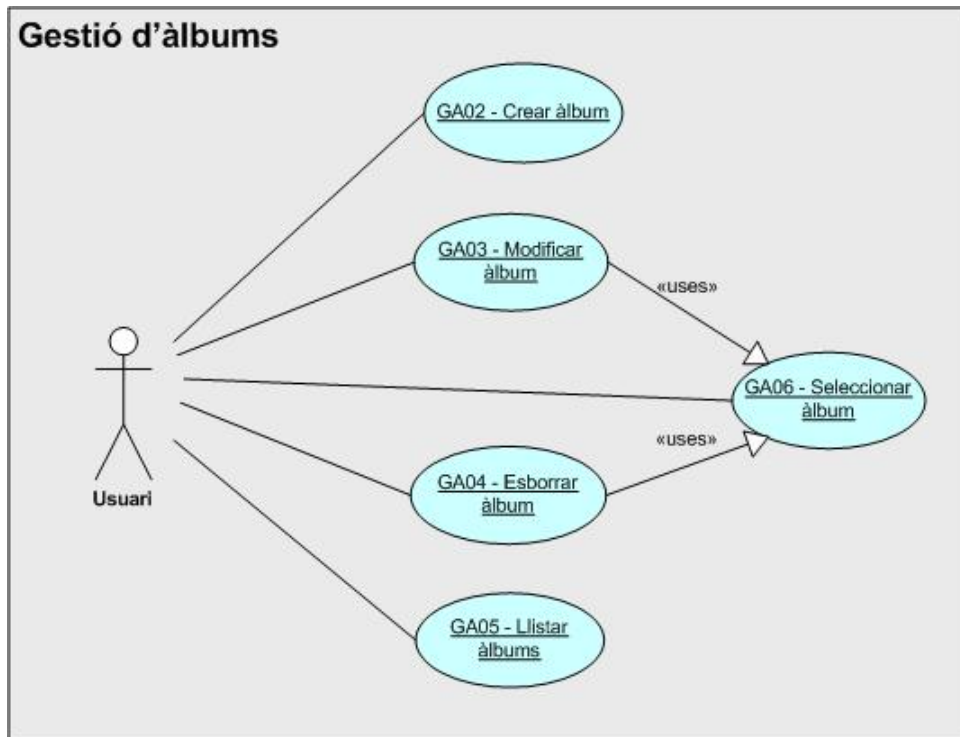


Figura 4: Gestió d'àlbums

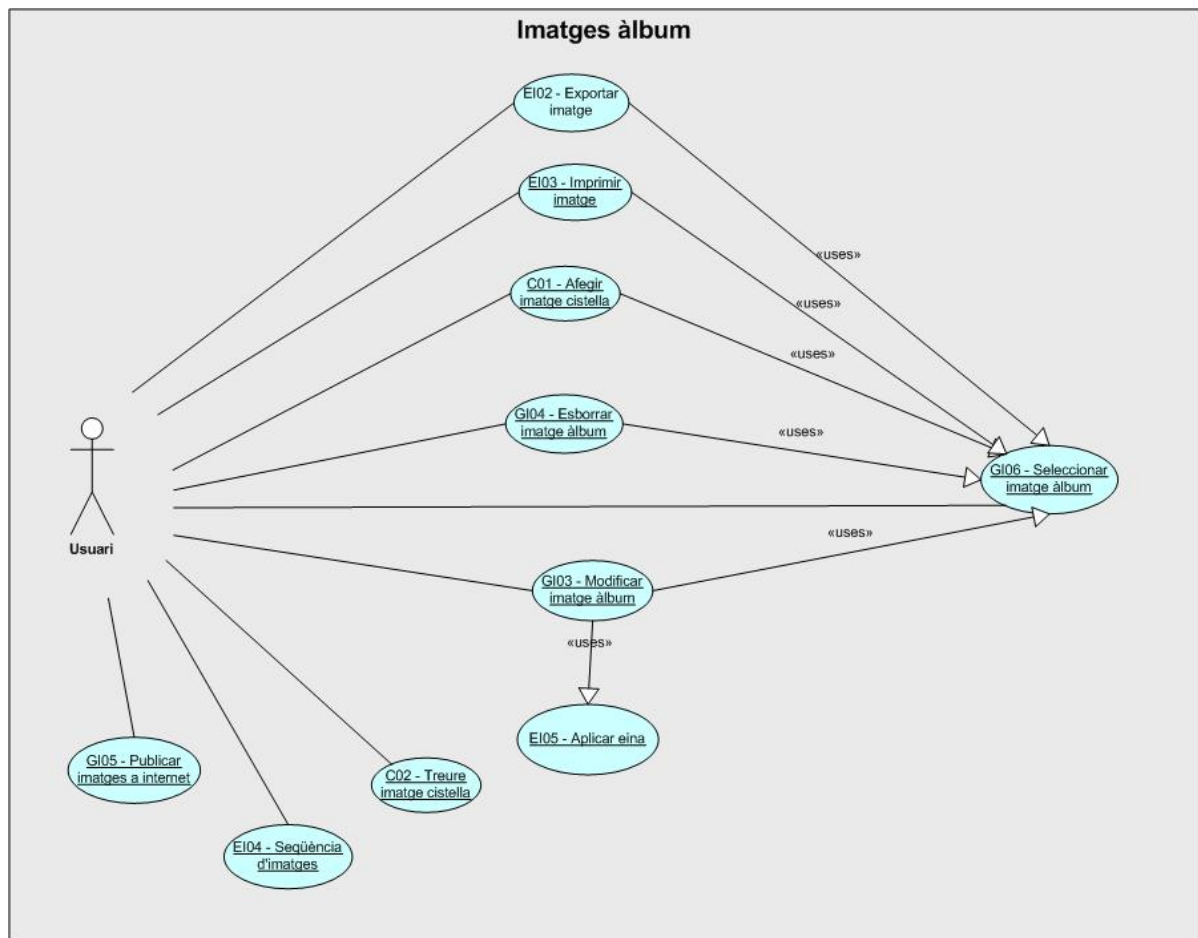


Figura 5: Gestió Imatges àlbum

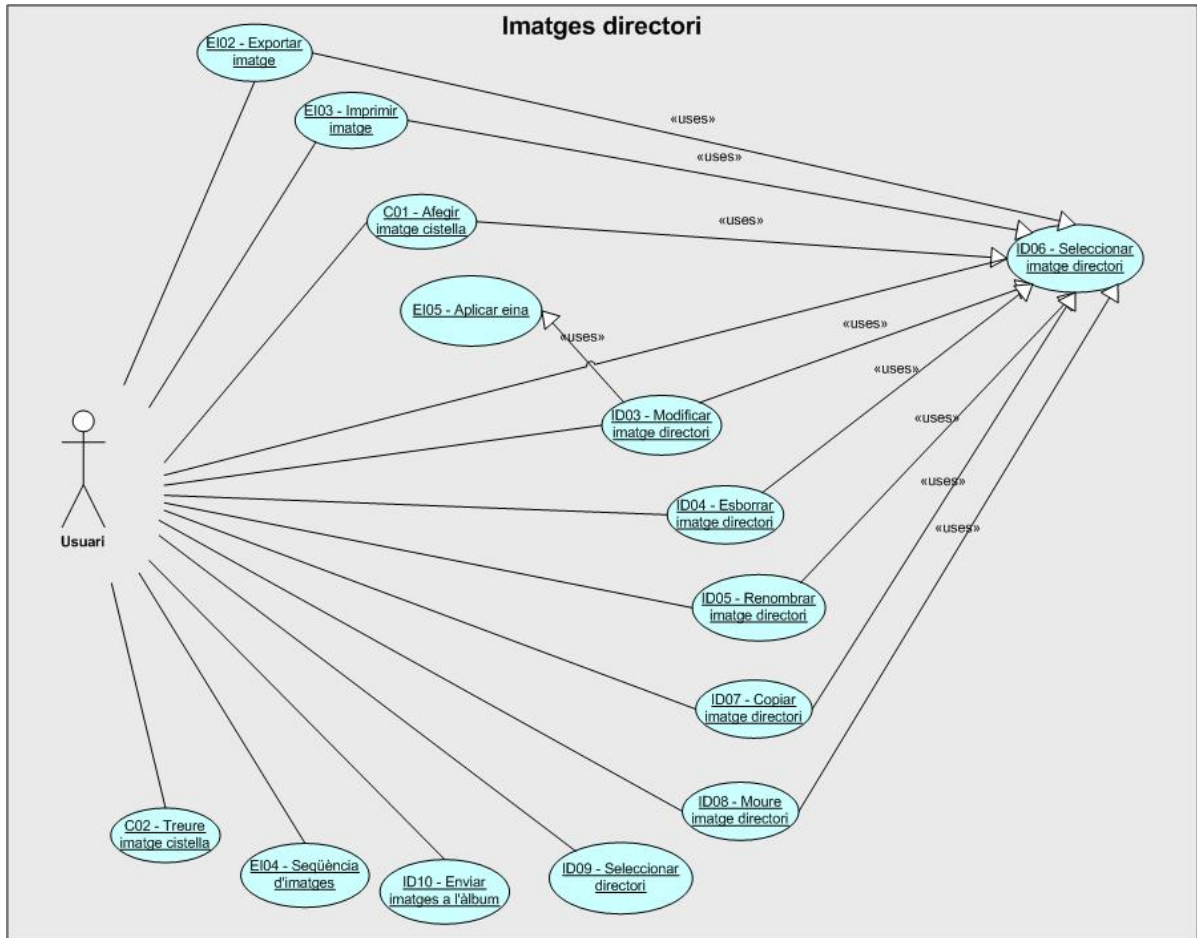


Figura 6: Gestió imatges directori

3.1.1. Taula de casos d'ús

En la taula següent es mostra una llista dels casos d'ús que intervenen en l'aplicació.

Codi	Descripció	Actor(s)
GA02	Crear àlbum	Usuari
GA03	Modificar àlbum	Usuari
GA04	Esborrar àlbum	Usuari
GA05	Llistar àlbums	Usuari
GA051	Llistar àlbums Impressora	Usuari
GA052	Llistar àlbums Excel	Usuari
GA053	Llistar àlbums PDF	Usuari
GA06	Seleccionar àlbum	Usuari

GI03	Modificar imatge de l'àlbum	Usuari
GI04	Esborrar imatge de l'àlbum	Usuari
GI05	Publicar imatges a Internet	Usuari
GI06	Seleccionar imatge de l'àlbum	Usuari
GD03	Modificar imatge directori	Usuari
GD04	Esborrar imatge directori	Usuari
GD05	Renombrar imatge directori	Usuari
GD06	Seleccionar imatge directori	Usuari
GD07	Copiar imatge directori	Usuari
GD08	Moure imatge directori	Usuari
GD09	Seleccionar directori	Usuari
GD10	Enviar imatges a l'àlbum	Usuari
C01	Afegir imatge cistella	Usuari
C02	Treure imatge cistella	Usuari
EI02	Exportar imatge	Usuari
EI03	Imprimir imatge	Usuari
EI04	Seqüència d'imatges	Usuari
EI05	Aplicar eina	Usuari
EI051	Aplicar eina rotació	Usuari
EI052	Aplicar eina transformació	Usuari
EI053	Aplicar eina blanc negre	Usuari
EI054	Aplicar eina marc	Usuari
EI055	Aplicar eina retallar	Usuari
EI056	Aplicar eina zoom	Usuari
EI057	Aplicar eina scroll	Usuari
EI058	Aplicar eina relleu	Usuari
EI059	Aplicar eina llapis	Usuari
EI060	Aplicar eina difuminar	Usuari

3.1.2. Descripció textual dels casos d'ús

GA02-Crear àlbum

Identificador	GA02
Nom	Crear àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari crea un nou àlbum.
Actor(s)	Usuari
Precondicions	Cap
Postcondicions	L'àlbum ha sigut enregistrat a la base de dades o el procés ha sigut cancel·lat.
Flux normal	<ol style="list-style-type: none">1. El cas d'ús comença quan l'actor selecciona l'opció de crear un nou àlbum.2. El sistema mostra l'entrada de dades de l'àlbum en un marc de la finestra de l'aplicació. Les dades són: (codiAlbum, titol, descripcio, dataCreacio).3. L'actor omple les dades.4. El sistema valida les dades. Si la validació és incorrecta torna al pas 3.5. El sistema assigna a codiAlbum i dataCreacio el seu valor de manera automàtica, codiAlbum és un enter autonumèric i dataCreacio la data actual.6. El sistema enregistra el nou àlbum en la base de dades i acaba el cas d'ús
Fluxos alternatius	<ol style="list-style-type: none">a. L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none">• Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

GA03-Modificar àlbum

Identificador	GA03
Nom	Modificar àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari modifica les dades d'un àlbum.
Actor(s)	Usuari

Precondicions	Cal un àlbum seleccionat amb el cas d'ús <u>GA06 - Seleccionar àlbum</u>
Postcondicions	Les dades de l'àlbum han sigut modificades, validades i enregistrades a la base de dades o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none"> 1. El sistema mostra en un marc de la finestra les dades de l'àlbum. 2. L'actor modifica les dades modificables de l'àlbum. Les dades modificables són: (títol, descripció) 3. El sistema valida les dades. Si no són validades correctament torna al pas 3. 4. El sistema enregistra els canvis en la base de dades i finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none"> a. L'actor pot cancel·lar el procés en el pas 2, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GA06 - Seleccionar àlbum</u>

GA04-Esborrar àlbum

Identificador	GA04
Nom	Esborrar àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor esborra un àlbum.
Actor(s)	Usuari
Precondicions	Cal un àlbum seleccionat amb el cas d'ús <u>GA06 - Seleccionar àlbum</u> .
Postcondicions	L'àlbum ha sigut esborrat, o bé s'ha cancel·lat el procés.
Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan l'actor selecciona el botó esborrar àlbum. 2. El sistema mostra a l'actor un formulari de confirmació. 3. L'actor confirma l'esborrat de l'àlbum 4. El sistema esborra l'àlbum de la base de dades i finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none"> a. L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si es produeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

Inclusions	Inclou a <u>GA06 - Seleccionar àlbum</u>
------------	--

GA05-Llistar àlbums

Identificador	GA05
Nom	Llistar àlbums
Autor	Narcís Patiño Gispert
Resum	El cas d'ús descriu com un actor llista els àlbums de la base de dades. És un cas d'ús genèric el qual és especialitzat per diversos casos d'ús.
Actor(s)	Usuari
Precondicions	Cal que existeixi almenys un àlbum a la base de dades.
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none">1. El cas d'ús comença quan l'actor selecciona l'opció de llistar àlbums.2. El sistema mostra el llistat en pantalla.3. Finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none">a. Des del pas 2 es pot seleccionar llistar els àlbums a impressora, Excel o a PDF.
Excepcions	<ul style="list-style-type: none">• Si es produeix algun error en l'accés o en la sortida de les dades, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Extensions	<u>GA051 – Llistar àlbums a Impressora</u> <u>GA052 – Llistar àlbums a Excel</u> <u>GA053 – Llistar àlbums a PDF</u>

GA06-Seleccionar àlbum

Identificador	GA06
Nom	Seleccionar àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor del sistema pot seleccionar i mostrar les dades d'un àlbum.
Actor(s)	Usuari
Precondicions	Cal que existeixi almenys un àlbum a la base de dades.
Postcondicions	Cap.

Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan un usuari selecciona un àlbum de la llista que mostra el sistema. 2. El sistema mostra les dades de l'àlbum en un marc de la pantalla de l'aplicació. Les dades visibles de l'àlbum són: (títol, descripció, dataCreacio). D'aquesta manera finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si es produeix algun error d'accés a la informació, es captura l'excepció, és mostra un missatge i finalitza el cas d'ús.

GI03-Modificar imatge de l'àlbum

Identificador	GI03
Nom	Modificar imatge de l'àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari modifica una imatge
Actor(s)	Usuari
Precondicions	Cal una imatge de l'àlbum seleccionada amb el cas d'ús <u>GI06 - Seleccionar imatge de àlbum.</u>
Postcondicions	Les dades de la imatge han sigut modificades, validades i enregistrades a la base de dades o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none"> 1. L'actor modifica les dades modificables de la imatge. Les dades modificables són: (títol, descripció, imatge). 2. El sistema valida els canvis. Si les dades no són validades correctament torna al pas 1. 3. L'actor envia petició de guardar canvis al sistema. 4. El sistema enregistra els canvis en la base de dades i finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none"> a. L'actor modifica la imatge a través del cas d'ús <u>EI05- Aplicar eina</u> b. L'actor torna al pas a o va al pas 3 per enviar petició d'enregistrar canvis. <p>L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.</p>
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	<p>Inclou a <u>GI06 - Seleccionar imatge de l'àlbum</u></p> <p>Inclou a <u>EI05 - Aplicar eina</u></p>

GI04-Esborrar imatge de l'àlbum

Identificador	GI04
Nom	Esborrar imatge de l'àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor esborra una imatge de àlbum.
Actor(s)	Usuari
Precondicions	Cal una imatge de l'àlbum seleccionada amb el cas d'ús <u>GI06 - Seleccionar imatge de àlbum.</u>
Postcondicions	L'àlbum ha sigut esborrat o el procés ha sigut cancel·lat.
Flux normal	<ol style="list-style-type: none">1. El cas d'ús comença quan l'actor selecciona l'opció d'esborrar la imatge de l'àlbum seleccionada prèviament.2. El sistema mostra a l'actor un formulari de confirmació.3. L'actor confirma l'esborrat de la imatge.4. El sistema esborra la imatge de l'àlbum i finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none">a. L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none">• Si es produeix algun error en l'eliminació, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GI06 - Seleccionar imatge de àlbum</u>

GI05-Publicar imatges a Internet

Identificador	GI05
Nom	Publicar imatges a Internet
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari publica les imatges seleccionades a Internet.
Actor(s)	Usuari
Precondicions	Cal que hi hagi almenys una imatge en la cistella d'imatges seleccionades.
Postcondicions	Les imatges han sigut publicades a Internet o el procés ha sigut cancel·lat.

Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan l'actor prem el botó de publicar imatges a Internet. (Aquest botó només està actiu si hi ha almenys una imatge a la cistella.) 2. El sistema mostra una finestra on l'usuari ha d'introduir l'adreça FTP on es publicaran les imatges. 3. El sistema envia a Internet les imatges que estan a la cistella d'imatges seleccionades a l'àlbum seleccionat. 4. El sistema buida la cistella d'imatges seleccionades i finalitza el cas d'ús.
Fluxos alternatius	Cap
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en la tramesa, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

GI06-Seleccionar imatge de l'àlbum

Identificador	GI06
Nom	Seleccionar imatge de l'àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor del sistema pot seleccionar una imatge i mostrar-la juntament amb les seves dades associades.
Actor(s)	Usuari
Precondicions	<p>Cal un àlbum seleccionat amb el cas d'ús <u>GA06 - Seleccionar àlbum</u></p> <p>Cal que en la base de dades existeixi almenys una imatge relacionada amb l'àlbum seleccionat.</p>
Postcondicions	Cap.
Flux normal	<ol style="list-style-type: none"> 1. El sistema mostra una llista de miniatures d'imatges (thumbnails) que corresponen a les imatges enregistrades a l'àlbum. 2. L'actor en selecciona una. 3. El sistema mostra la imatge i les seves dades associades (títol, descripció, dataAlta) en un marc de la pantalla de l'aplicació. D'aquesta manera finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

GD03-Modificar imatge directori

Identificador	GD03
Nom	Modificar imatge directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari modifica una imatge del directori seleccionat.
Actor(s)	Usuari
Precondicions	Cal una imatge del directori, seleccionada amb el cas d'ús <u>GD06 - Seleccionar imatge directori.</u>
Postcondicions	La imatge ha sigut modificada o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none">1. L'actor modifica la imatge a través del cas d'ús <u>EI05-Aplicar eina</u>2. L'actor torna al pas 1 o envia petició al sistema de guardar la imatge modificada.3. El sistema enregistra la imatge modificada i finalitza el cas d'ús.
Fluxos alternatius	<ol style="list-style-type: none">a. L'actor pot cancel·lar el procés en el pas 2, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none">• Si succeeix algun error al sobre escriure el fitxer, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u> Inclou a <u>EI05 – Aplicar eina</u>

GD04-Esborrar imatge directori

Identificador	GD04
Nom	Esborrar imatge directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari esborra una imatge del directori seleccionat.
Actor(s)	Usuari
Precondicions	Cal una imatge del directori, seleccionada amb el cas d'ús <u>GD06 - Seleccionar imatge directori.</u>
Postcondicions	La imatge ha sigut esborrada o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none">1. L'actor seleccionant l'opció d'esborrar imatge.

	<ol style="list-style-type: none"> 2. El sistema mostra una finestra de confirmació. 3. L'actor confirma l'esborrat. 4. El sistema esborra la imatge seleccionada i finalitza el cas d'ús.
Fluxos alternatius	<ul style="list-style-type: none"> o L'actor pot cancel·lar el procés en el pas 2, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error al esborrar el fitxer, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u>

GD05-Renombrar imatge directori

Identificador	GD05
Nom	Renombrar imatge directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari renombra una imatge del directori seleccionat.
Actor(s)	Usuari
Precondicions	Cal una imatge del directori, seleccionada amb el cas d'ús <u>GD06 - Seleccionar imatge directori</u> .
Postcondicions	La imatge ha sigut renombrada o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció de renombrar imatge. 2. El sistema sol·licita el nou nom. 3. L'actor entra el nou nom. 4. El sistema renombra la imatge si valida el nou nom i finalitza el cas d'ús. En cas contrari torna al pas 2.
Fluxos alternatius	<ul style="list-style-type: none"> o L'actor pot cancel·lar el procés en el pas 2, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error al renombrar el fitxer, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u>

GD06-Seleccionar imatge directori

Identificador	GD06
Nom	Seleccionar imatge del directori actual
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor del sistema pot seleccionar una imatge del directori actual i mostrar-la.
Actor(s)	Usuari
Precondicions	Cal que el directori actual contingui almenys una imatge.
Postcondicions	Cap.
Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan un usuari selecciona una imatge de la llista de thumbnails mostrats pel sistema. 2. El sistema mostra la imatge al marc de la finestra destinat a l'edició de la imatge i finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'accés a la imatge, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

GD07-Copiar imatge directori

Identificador	GD07
Nom	Copiar imatge directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari copia una imatge del directori seleccionat.
Actor(s)	Usuari
Precondicions	Cal una imatge del directori, seleccionada amb el cas d'ús <u>GD06 - Seleccionar imatge directori</u> .
Postcondicions	La imatge ha sigut copiada o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció de copiar imatge. 2. El sistema sol·licita el directori destí presentant una finestra de diàleg. 3. L'actor selecciona el directori destí. 4. El sistema copia la imatge al directori destí i finalitza el cas d'ús.
Fluxos alternatius	<ul style="list-style-type: none"> ○ L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error al copiar el fitxer, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u>
------------	---

GD08-Moure imatge directori

Identificador	GD08
Nom	Moure imatge directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari mou una imatge del directori seleccionat.
Actor(s)	Usuari
Precondicions	Cal una imatge del directori, seleccionada amb el cas d'ús <u>GD06 - Seleccionar imatge directori</u> .
Postcondicions	La imatge ha sigut moguda o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció de moure imatge. 2. El sistema sol·licita el directori destí presentant una finestra de diàleg. 3. L'actor selecciona el directori destí. 4. El sistema mou la imatge al directori destí i finalitza el cas d'ús.
Fluxos alternatius	<ul style="list-style-type: none"> ○ L'actor pot cancel·lar el procés en el pas 3, finalitzant així el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error al moure el fitxer, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u>

GD09-Seleccionar directori

Identificador	GD09
Nom	Seleccionar directori
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un actor del sistema pot seleccionar un directori d'un dispositiu donat.
Actor(s)	Usuari
Precondicions	Cap.
Postcondicions	Cap.
Flux normal	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció de seleccionar directori.

	<ol style="list-style-type: none"> 2. El sistema mostra una finestra amb les unitats i directoris per seleccionar. 3. L'actor selecciona el directori que serà el directori actual. Finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'accés al sistema de fitxers, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

GD10-Enviar imatges a l'àlbum

Identificador	GD10
Nom	Enviar imatges a l'àlbum
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari envia les imatges seleccionades a un àlbum.
Actor(s)	Usuari
Precondicions	<p>Cal un àlbum seleccionat prèviament amb el cas d'ús <u>GA06 - Seleccionar àlbum.</u></p> <p>Cal que hi hagi almenys una imatge en la cistella d'imatges seleccionades.</p>
Postcondicions	Les imatges han sigut enregistrades a la base de dades o el procés ha sigut cancel·lat.
Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan l'actor prem el botó d'enviar imatges a l'àlbum. (Aquest botó només està actiu si hi ha almenys una imatge a la cistella d'imatges seleccionades.) 2. El sistema registra les imatges que estan a la cistella d'imatges seleccionades a l'àlbum seleccionat. Per a cada imatge afegeix un registre a la taula imatge de la base de dades, el contingut del qual és: (codilmatge, títol, descripció, dataAlta). Els valors de codilmatge i dataAlta són generats automàticament pel sistema, codilmatge és un autonumèric i dataAlta la data actual. 3. El sistema buida la cistella d'imatges seleccionades i finalitza el cas d'ús.
Fluxos alternatius	Cap
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

C01-Afegir imatge cistella

Identificador	C01
Nom	Afegir imatge cistella
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari afegeix la imatge seleccionada del directori o de l'àlbum actual a la cistella d'imatges seleccionades.
Actor(s)	Usuari
Precondicions	Cal una imatge seleccionada prèviament.
Postcondicions	La imatge ha sigut afegida a la cistella o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none">1. L'actor selecciona l'opció d'afegir imatge a la cistella.2. El sistema afegeix la imatge seleccionada a la cistella. Finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u> o <u>GI06 – Seleccionar imatge àlbum</u>

C02-Treure imatge cistella

Identificador	C02
Nom	Treure imatge cistella
Autor	Narcís Patiño Gispert
Resum	El cas d'ús mostra com un usuari esborra una imatge de la cistella d'imatges seleccionades.
Actor(s)	Usuari
Precondicions	Cal que hi hagi almenys una imatge en la cistella d'imatges.
Postcondicions	La imatge ha sigut treta de la cistella o el procés ha sigut cancel·lat .
Flux normal	<ol style="list-style-type: none">1. L'actor selecciona una imatge de la cistella d'imatges.2. L'actor selecciona l'opció d'esborrar imatge de la cistella. (si no hi ha cap imatge a la cistella, aquesta opció està desactivada)3. El sistema esborra la imatge seleccionada de la cistella. Finalitza el cas d'ús.

EI02-Exportar imatge

Identificador	EI02
Nom	Exportar imatge
Autor	Narcís Patiño Gispert
Resum	El cas d'ús descriu com un usuari exporta una imatge.
Actor(s)	Usuari
Precondicions	Cal una imatge seleccionada prèviament.
Postcondicions	La imatge ha sigut exportada o bé s'ha cancel·lat el procés.
Flux normal	<ol style="list-style-type: none">1. Aquest cas d'ús comença quan l'actor selecciona el botó d'exportar imatge.2. El sistema mostra una finestra sol·licitant el nom de la imatge i en quin format es vol exportar (jpg,gif,bmp,png)3. L'actor entra les dades requerides.4. El sistema les valida. Si no són validades torna al pas 2.5. El sistema exporta la imatge i finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none">• Si succeeix algun error en l'exportació, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u> o <u>GI06 – Seleccionar imatge àlbum</u>

EI03-Imprimir Imatge

Identificador	EI03
Nom	Imprimir imatge
Autor	Narcís Patiño Gispert
Resum	El cas d'ús descriu com un usuari imprimeix una imatge.
Actor(s)	Usuari
Precondicions	Cal una imatge seleccionada prèviament.
Postcondicions	La imatge ha sigut impresa o bé s'ha cancel·lat el procés.
Flux normal	<ol style="list-style-type: none">1. Aquest cas d'ús comença quan l'actor selecciona el botó d'imprimir imatge.2. El sistema mostra una llista dels dispositius disponibles i demana a l'actor confirmació per imprimir.

	<ol style="list-style-type: none"> 3. L'actor confirma la impressió. 4. El sistema imprimeix la imatge i finalitza el cas d'ús.
Flux alternatiu	L'actor pot cancel·lar el procés en el pas 3 i finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en la impressió, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Inclou a <u>GD06 - Seleccionar imatge directori</u> o <u>GI06 – Seleccionar imatge àlbum</u>

EI04-Seqüència d'imatges

Identificador	GI05
Nom	Seqüència d'imatges
Autor	Narcís Patiño Gispert
Resum	El cas d'ús descriu una presentació que mostra per pantalla seqüencialment a les imatges que componen un àlbum o a un directori.
Actor(s)	Usuari
Precondicions	Cal un àlbum seleccionat o un directori seleccionat. Cal que almenys hi hagi una imatge a mostrar.
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. El cas d'ús comença quan l'actor selecciona l'opció corresponent a la seqüència d'imatges. 2. El sistema mostra la seqüència d'imatges per pantalla. 3. Finalitza el cas d'ús.
Fluxos alternatius	Cap
Excepcions	<ul style="list-style-type: none"> • Si succeeix algun error en l'enregistrament, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.

EI05-Aplicar eina

Identificador	EI05
Nom	Aplicar eina
Autor	Narcís Patiño Gispert

Resum	El cas d'ús descriu com un actor aplica una funcionalitat a una imatge. És un cas d'ús genèric el qual és especialitzat per diversos casos d'ús.
Actor(s)	Usuari
Precondicions	Cal una imatge prèviament seleccionada.
Postcondicions	S'ha aplicat l'eina o s'ha cancel·lat el procés.
Flux normal	<ol style="list-style-type: none">1. El cas d'ús comença quan l'actor selecciona l'opció corresponent a un dels casos d'ús especialitzats d'aplicar eina.2. El sistema mostra els paràmetres de l'eina.3. L'actor selecciona els paràmetres adequats de l'eina i l'aplica.4. El sistema aplica l'eina i finalitza el cas d'ús.
Fluxos alternatius	a. Si l'eina seleccionada no té paràmetres, el flux alternatiu és aquest: <ol style="list-style-type: none">1. El cas d'ús comença quan l'actor selecciona l'opció corresponent a un dels casos d'ús especialitzats d'aplicar eina.2. El sistema aplica l'eina i finalitza el cas d'ús.
Excepcions	<ul style="list-style-type: none">• Si es produeix algun error en l'accés o en la sortida de les dades, el sistema captura l'excepció, mostra un missatge i finalitza el cas d'ús.
Inclusions	Cap
Extensions	<u>EI051 Aplicar eina rotació</u> <u>EI052 Aplicar eina transformació</u> <u>EI053 Aplicar eina blanc negre</u> <u>EI054 Aplicar eina marc</u> <u>EI055 Aplicar eina retallar</u> <u>EI056 Aplicar eina zoom</u> <u>EI057 Aplicar eina scroll</u> <u>EI058 Aplicar eina relleu</u> <u>EI059 Aplicar eina llapis</u> <u>EI060 Aplicar eina difuminar</u>

3.2. Model del domini

Per especificar el context del programari de manera simplificada, es descriu a continuació el model del domini de l'aplicació, el qual com es pot veure és molt senzill.

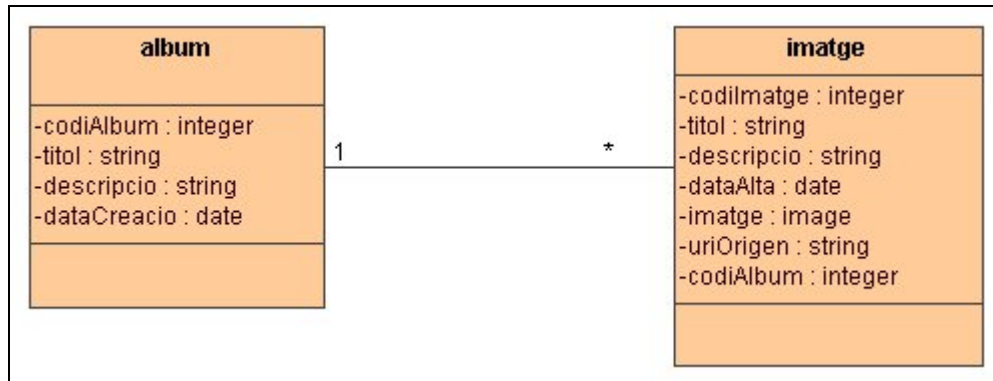


Figura 7: Model de domini

Els principals objectes detectats són:

- **album**, la funció d'aquest objecte és la d'agrupar diferents imatges sota un mateix concepte. Es pot classificar com a objecte del món real i del qual es mostren els atributs. Cada àlbum pot incloure una o més imatges.
- **imatge**, una imatge pertany a un àlbum. També es pot classificar com a objecte del món real i se'n mostren els atributs.

4. Disseny

4.1. Arquitectura de l'aplicació

S'ha decidit dissenyar l'aplicació seguint un model clàssic de tres capes:

- **Capa de presentació:** Aquesta capa mostra informació a l'usuari final i permet l'entrada de dades. La interfície d'usuari assumirà aquesta funcionalitat.
- **Capa de negoci:** Aquesta capa conté funcions de negoci que actuen en les dades de negoci.
- **Capa de dades:** La capa de dades guarda les dades de manera persistent en la base de dades. Aquesta capa és també coneguda com la capa de recurs (resource layer). El gestor de disc assumirà aquesta funcionalitat.

A continuació es comenta el funcionament de les diverses capes i components en l'entorn de l'aplicació, el .NET.

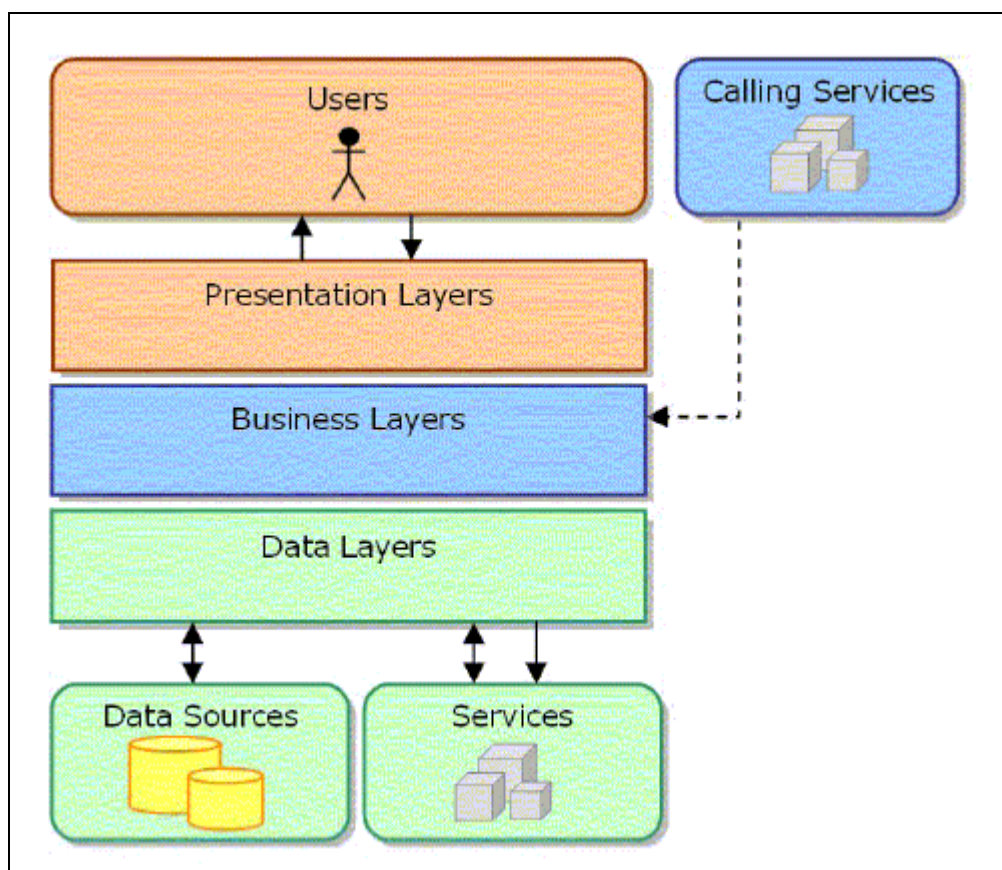


Figura 8: Capes de l'aplicació

4.1.1. Capa de presentació

La capa de presentació conté els components que són requerits per la interacció entre l'usuari i l'aplicació. Les capes de presentació més simples contenen components d'interfície d'usuari com Windows Forms o ASP.NET Web Forms. Per interaccions més complexes, es poden dissenyar components de processos d'usuari per organitzar la interfície d'usuari i controlar la interacció de l'usuari. Els components de procés d'usuari són especialment útils quan la interacció de l'usuari segueix un flux de passes, les quals es poden preveure. Per exemple quan s'utilitza un *wizard* per realitzar una tasca.

4.1.2. Capa de negoci

El nucli de l'aplicació és la funcionalitat de negoci que proporciona. Una aplicació realitza un procés de negoci que consisteix en una o més tasques. En els casos més simples, cada tasca pot ser encapsulada en un mètode d'un component .NET i cridada síncrona o asíncronament. Per negocis més complexos que requereixen múltiples passes i transaccions de llarga execució, l'aplicació necessita disposar d'alguna forma d'organitzar les tasques de negoci i l'estat d'enregistrament fins que el procés s'ha completat. En aquests escenaris, es pot utilitzar *BizTalk Server Orchestration* per definir el *workflow* per el procés de negoci.

Es pot dissenyar la lògica de les capes de negoci per ser utilitzades directament per els components de presentació o per ser encapsulada com un servei i cridada a través d'un servei d'interfície, el qual coordina la conversa asíncrona amb el servei que fa la crida i invoca el *BizTalk Server workflow* o components de negoci. El nucli de la lògica de negoci també se l'anomena **domini lògic**. Els components de negoci també poden fer peticions de serveis externs, i en aquest cas és necessari implementar agents de servei per manejar la conversa requerida.

4.1.3. Capa de dades

Pràcticament totes les aplicacions i serveis necessiten enregistrar i accedir algun tipus de dades. Quan es treballa amb dades s'ha de determinar:

- El magatzem de dades (*data store*) que s'està utilitzant.
- El disseny dels components utilitzats per accedir al *data store*.
- *El format de les dades passades entre components i el model de programació que requereix aquest format.*

L'aplicació o servei pot tenir un o mes orígens de dades (*data sources*), i aquests poden ser de diferents tipus. La lògica utilitzada per accedir dades en un *data source* serà encapsulada en *data access logic components*, els quals proporcionen mètodes per consultar i actualitzar dades. Les dades de l'aplicació necessiten relacionar-se amb entitats del món real que juguen un paper en el negoci. En alguns escenaris, es tenen components personalitzats que representen aquestes entitats, mentre en altres es pot triar de treballar amb dades utilitzant *datasets* ADO.NET o directament documents XML.

La major part de les aplicacions utilitzen bases de dades relacionals com a *data store* primari per les dades de l'aplicació. Altres inclouen Microsoft Exchange Server Web store, el sistema de fitxers o altres.

Quan l'aplicació recupera dades de la base de dades, ho pot fer utilitzant un format de dades com *DataSet* o *DataReader*. Les dades seran transferides entre les capes de l'aplicació i finalment seran tractades per els components. Es poden utilitzar diferents formats per recuperar, passar i operar amb les dades. Tanmateix, per millorar el rendiment i la mantenibilitat, cal tenir només un joc limitat de formats. Evitant també d'aquesta manera, la necessitat de capes de traducció addicionals i la necessitat d'aprendre API diferents.

4.1.4. Components

És important fer notar, que les capes són simplement agrupacions de components de software que componen l'aplicació o el servei. Les capes ajuden a diferenciar entre diferents tipus de tasques fetes per els components, fent més senzill dissenyar reusabilitat dins la solució. Cada capa lògica conté un cert nombre de tipus de components, agrupats en subcapes, on cada subcapa realitza un tipus específic de tasca.

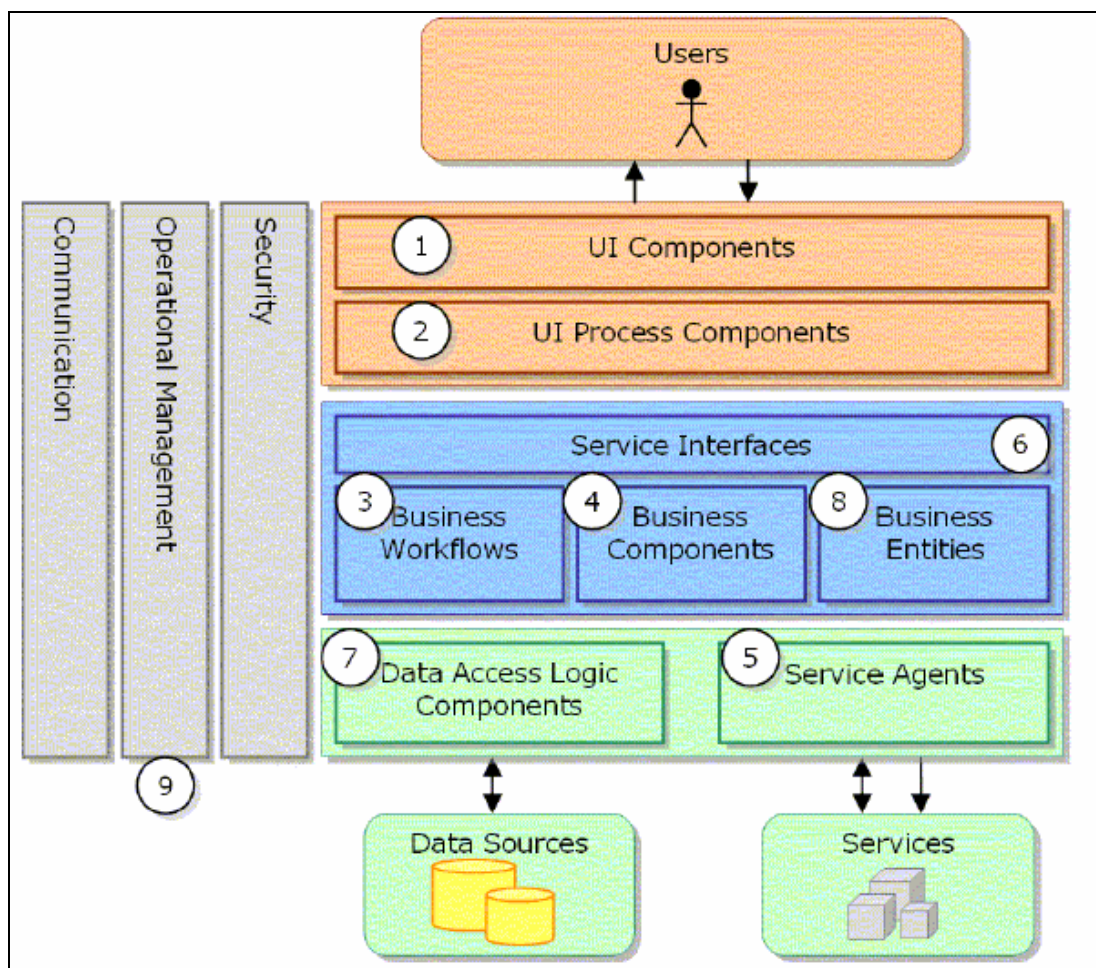


Figura 9: Components arquitectura de capes.

1. **Components d'interfície d'usuari (User Interface (UI) Components):** Són els components de la interfície d'usuari. La majoria de les aplicacions han de subministrar una manera per a què els usuaris puguin interactuar amb l'aplicació. En .NET les interfícies d'usuari s'implementen amb Windows Forms, WPF Forms, pàgines ASP.NET, controls i altres tecnologies que permetin renderitzar i formatar dades per als usuaris i adquirir i validar dades proporcionades per ells. En el **Photo.Net** els usuaris interactuen amb els botons i altres controls que proporcionen els formularis WPF.
2. **Components de procés d'usuari (User process components):** En molts casos una interacció amb el sistema segueix un procés previsible. Per ajudar a sincronitzar i organitzar aquestes interaccions, pot ser útil impulsar que s'utilitzin diferents components de procés d'usuari. D'aquesta manera l'estat de la lògica de *management* no és totalment codificada en els propis elements de la interfície d'usuari i el mateix motor d'interacció bàsica d'usuari pot ser reutilitzat per múltiples interfícies d'usuari. En el **Photo.Net**, alguns exemples de component de procés d'usuari són: quan es selecciona un àlbum la situació previsible seria mostrar les dades de l'àlbum i un altre situació previsible és quan es dona d'alta un àlbum, ja que a continuació es produeix la corresponent recollida d'informació.
3. **Fluxos de treball de negoci (Business workflows):** Després que les dades requerides són col·leccionades per un procés d'usuari, les dades poden ser utilitzades per dur a terme un procés de negoci. Els fluxos de treball (*workflow*) de negoci defineixen i coordinen processos de negoci organitzats, que engloben múltiples passes, els quals poden ser implementats utilitzant eines de direcció de processos de negoci com són Biztalk Server Orchestration. En el **Photo.Net**, un exemple de workflow de negoci són els diferents passos d'enviar les imatges de la cistella a l'àlbum corresponent.
4. **Components de negoci (Business components):** Sense considerar si un procés de negoci consta d'un sol pas o un *workflow* organitzat, l'aplicació requereix components que implementen regles de negoci i probablement porten a terme tasques de negoci. Els components de negoci implementen la lògica de negoci de l'aplicació.
5. **Agents de servei (Service agents):** Quan un component de negoci necessita utilitzar una funcionalitat proporcionada per un servei extern, és pot necessitar proporcionar algun codi per manejar la semàntica de comunicació amb el servei particular. Els agents de servei aïllen les rareses de cridar serveis diversos de l'aplicació i poden proveir serveis addicionals, com la correspondència bàsica entre el format de les dades exposades per el servei i el format que l'aplicació requereix. En el **Photo.Net**, els components de negoci podrien utilitzar un agent de servei per manejar la comunicació amb el servidor *FTP* al qual es poden publicar les imatges de l'àlbum.
6. **Interfícies de servei (Service interfaces):** Per exposar lògica de negoci com a servei, cal crear interfícies de servei que suportin els contractes de comunicació (comunicació basada en missatge, formats, protocols, seguretat, excepcions, etc) que els seus diferents consumidors requereixin. Les interfícies de servei també són anomenades **façanes de negoci (business facades)**.
7. **Components de la lògica d'accés a dades (Data access logic components):** La major part de les aplicacions i serveis necessitaran accedir a un *data store* en algun punt durant un procés de negoci. La funció d'aquests components és el d'abstreure la lògica necessària per accedir a les dades d'una capa diferent de la dels components de la lògica d'accés a dades. Al fer-ho es centralitza la funcionalitat d'accés a les dades, la qual simplifica la seva configuració i manteniment.

Cadascun d'aquests components, típicament proporciona mètodes per executar les operacions Create, Read, Update i Delete, relacionades amb una entitat de negoci específica. Aquests mètodes poden ser utilitzats per els processos de negoci. En el **Photo.Net**, aquests components s'utilitzen per exemple per actualitzar les dades de l'entitat àlbum.

8. **Components de l'entitat de negoci (Business entity components):** La major part de les aplicacions requereixen traspasar dades entre components.(Per exemple, en el **Photo.Net**, la llista d'àlbums ha de ser passada des dels components de la lògica d'accés a dades als components d'interfície d'usuari, per tal que la llista sigui mostrada als usuaris.) Les dades s'utilitzen per representar entitats de negoci del món real, com poden ser àlbums o imatges. Les entitats de negoci que s'utilitzen internament en l'aplicació normalment són estructures de dades, com DataSets, DataReaders, o streams XML. Tots ells poden ser implementats també a través de classes OO que representen les entitats del món real amb les que l'aplicació ha de treballar.
9. **Components de seguretat, management operacional i comunicació:** L'aplicació, probablement necessitarà utilitzar components pel control d'excepcions, per autoritzar usuaris per realitzar certes tasques, i per comunicar-se amb altres serveis i aplicacions. Aquesta és la tasca d'aquests components.

4.2. Arquitectura WPF

Aquesta aplicació utilitza la tecnologia de presentació WPF de .NET i en conseqüència queda condicionada en gran mesura per la seva arquitectura. Per aquest motiu s'ha dedicat un apartat a descriure l'arquitectura WPF.

L'anatomia de WPF consta de serveis no dirigits, de subsistemes dirigits, y una API dirigida disponible per al consum d'aplicacions de WPF, coneguda com la base de presentació.

Aquest diagrama mostra l'arquitectura general de WPF, amb els components més importants que l'integren. Seguidament s'enumeren les seves funcions.

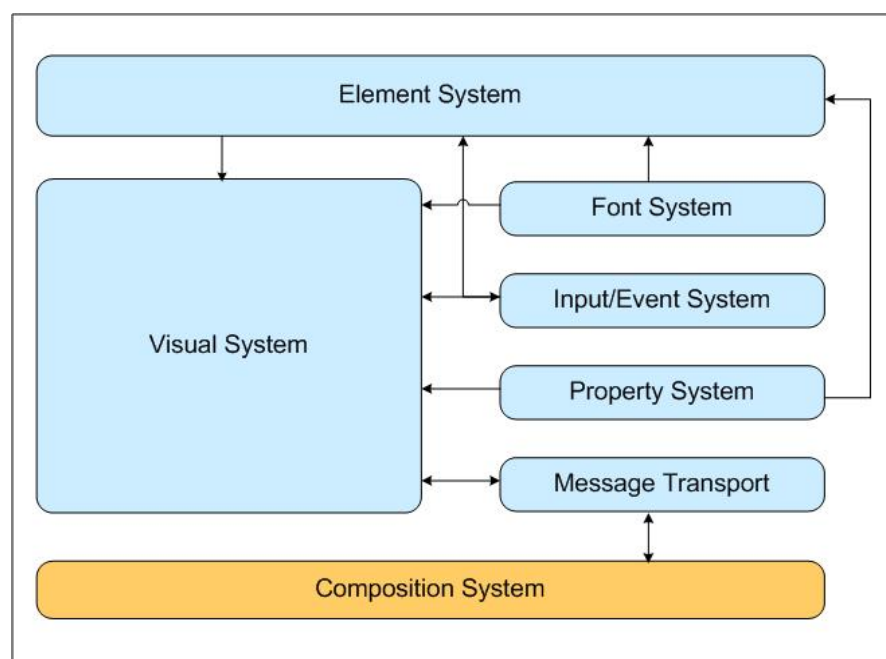


Figura 10: Arquitectura WPF

4.2.1. Element System

Representa la superfície de WPF amb la que els desenvolupadors interactuaran. Dins d'ell hi trobem els principals components de la interfície d'usuari (IU), tals com estils, disseny, controls, enllaç a dades (binding) i el disseny de text.

La major part dels components del **element system** deriven de la importantíssima classe `System.Windows.FrameworkElement`. Aquesta classe proporciona la funcionalitat bàsica per interactuar amb el nucli del servei de presentació de WPF i implementa mètodes de la seva superclasse, `System.Windows.UIElement`, la qual proporciona la funcionalitat per a tots els elements visuals dins de WPF per al disseny, successos i entrades.

Un concepte important en WPF és la noció dels arbres d'elements, els quals representen els elements visuals que componen una aplicació WPF. Existeixen dos tipus: l'arbre lògic i l'arbre visual.

L'arbre lògic (Logical Tree) és l'estructura jeràrquica que conté els elements exactes d'una aplicació WPF. Aquests elements s'anomenen elements d'aplicació.

L'arbre visual (Visual Tree) , és l'arbre lògic on de cadascun dels elements d'aplicació en pengen els seus elements de contingut visual. D'aquesta combinació d'elements d'aplicació i d'elements visuals s'anomena arbre visual.

4.2.2. Visual System

És el subsistema a través del qual les aplicacions accedeixen al nucli dels serveis de presentació de WPF. Aquest subsistema examina els components inclosos en l'aplicació (etiquetes, botons, text, gràfics 2D i 3D, animacions) i els comunicarà amb el **composition system** subjacent via **message transport** per generar en pantalla el resultat renderitzat.

Tot i que el concepte de **visual system** és clau en l'arquitectura de WPF, gran part de la feina pesada es fa en el **composition system** subjacent.

4.2.3. Font System

Aquest subsistema ha sigut totalment reescrit per WPF per proporcionar un motor de fonts i text, superior al dels sistemes previs disponibles en Windows. Els dos motors de fonts disponibles actualment, GDI i Uniscribe, tenen desavantatges importants que no els fan apropiats per WPF.

El nou **font system** proporciona un mecanisme únic per crear i guardar informació en catxé sobre fonts en WPF, incloent fonts TrueType i Adobe OpenType. La mètrica, glyphs, i camins que fan especial a un tipus de lletra són calculats per el **font system**, guardats en catxé i els fa disponibles pel seu ús en WPF. Aquest procés és costós i utilitzar aquesta tècnica d'enregistrament en catxé millora significativament el rendiment del disseny de text i renderització en WPF.

Cal fer notar que el **font system** és l'únic component dirigit que corre fora de procés i es comunica directament amb WPF a través de comunicació entre processos per compartir

fonts de la catxé.

Com es pot veure en la Figura 1, el **font system** interactua amb dos subsistemes principals, el **element system** i el **visual system**, cadascun dels qual per un propòsit diferent. El primer subsistema l'utilitza per l'organització i el disseny del text dins de la IU, això inclou paràgrafs, taules i blocs de text. El segon subsistema l'utilitza per a serveis de disseny de text dins d'una simple línia de text, com la direcció i espaiat.

4.2.4. Input/Event System

Aquest subsistema introdueix millores importants en l'entrada i interacció d'usuari respecte als sistemes previs disponibles en Windows, com Win32. Proporciona un flux lineal d'integració entre l'entrada d'usuari i l'arbre visual. A través de comandes, WPF proporciona un avançat mecanisme amb el qual una aplicació pot detectar i respondre a l'entrada d'usuari.

WPF assumeix la responsabilitat de gestionar els dispositius d'entrada i determinar l'acció apropiada en resposta dependent del tipus d'entrada rebut i els successos de l'aplicació. L'enrutament de successos (events), és un concepte molt important que cal entendre per desenvolupar aplicacions WPF. Aquest enrutament funciona de manera que els elements que formen l'arbre visual poden escoltar i respondre a altres elements a través de tunneling i bubbling.

Si l'arbre visual de l'aplicació és molt complex, l'enrutament d'events pot afectar seriosament el rendiment de l'aplicació. Per solucionar aquest problema WPF implementa manipuladors(handlers) de classe com a part del **input/event system**. Aquests manipuladors de classe proporcionen escoltes(listeners) d'event estàtics de qualsevol instància per respondre l'entrada d'usuari, tals com l'entrada de text des del teclat.

4.2.5. Property System

Aquest subsistema es essencial pel nucli de funcions relacionades amb dades dins de WPF. Comprèn els tres components següents:

- Notificació de canvi (Change Notificacion): És la base per a molts aspectes de WPF. Totes les dades dins de WPF son mostrades a través de propietats i per tant els canvis en aquestes propietats son integrals en la interacció amb els elements d'una aplicació WPF. La notificació de canvi recau sobre la interfície System.ComponentModel.INotifyPropertyChanged. És responsabilitat de cada classe que en deriva la de comunicar els canvis en les propietats a través de l'event PropertyChanged per tal de monitoritzar-los.
- Enregistrament (Storage): Com s'ha dit, les propietats són l'element principal de dades en WPF, les quals proporcionen el fonament que pot suportar el model declaratiu, i activen moltes de les característiques incloses en WPF, com els estils, enllaç a dades, animació i altres. Però aquest aparellament extensiu de propietats també té desavantatges, es pot produir una sobrecàrrega enorme d'emmagatzament perquè cada element pot contenir moltes propietats i això pot afectar negativament al rendiment en temps d'execució i de visualització. El sistema de dependència de propietat proporciona el suport per l'emmagatzematge de

propietat simplificat, utilitzant els mètodes ajudants, les classes poden accedir a les propietats fàcilment i prendre part en la notificació de canvi, binding, estils i les altres característiques disponibles en aquest sistema.

- Expressions: Les expressions proporcionen l'extensibilitat requerida per implementar el model declaratiu i són utilitzades exhaustivament en tot el WPF. Les característiques de WPF, estils, herència i enllaç a dades, depenen de les expressions per avaluar-se dinàmicament i moure dades pel sistema..

4.2.6. Message Transport System

Aquest subsistema és un component clau en l'arquitectura de WPF i la seva tasca és relacionar el **visual system** amb el **composition system**. Com ja s'ha comentat, el **sistema visual** proporciona una interfície dirigida a través de la qual els altres subsistemes dirigits dins de WPF proveeixen instruccions sobre quins elements han de ser representats sobre la pantalla. Tot i que el sistema visual proporciona aquesta funcionalitat, no la implementa ell mateix; una gran part d'aquesta feina la realitza el **composition system**.

El message transport system proveeix canals de comunicació entre els dos subsistemes esmentats. Aquests canals implementen un protocol remot de .NET que crea un mecanisme eficient a través del qual poden passar les estructures de dades. Utilitzar aquest protocol remot, permet que WPF transfereixi les estructures de dades a l'altre costat dels canals de comunicació, subministrant la possibilitat de descarregar la versió dels elements de la IU a una màquina completament diferent. Per tant, aquest concepte és primordial per als escenaris de clients terminals, tals com Escriptori Remot o Citrix MetaFrame, on el processament de gràfics d'aplicacions que s'executen remotament poden ser descarregats des del servidor al client terminal.

4.2.7. Composition System

Com ja s'ha comentat, aquest subsistema no dirigit proporciona realment la combustió interna de l'arquitectura WPF. Rep instruccions del **visual system** dirigit i les converteix en gràfics que apareixien en la pantalla.

El composition system proveeix la visualització dels elements aparentment nebulosos dins d'una aplicació WPF. Això inclou la funcionalitat de calcular píxels per un element especial, i genera configuracions de triangle per a elements visuals, que son enviats directament al GPU per la sortida de la visualització. Aquest procés de rendering basat en maquinari (a través de Direct3D) allibera la CPU la qual pot optimitzar altres operacions, millorant l'èxit en conjunt d'aplicacions WPF. La qualitat de gràfics en pantalla és significativament més alta gràcies al ús de maquinari basat en rendering.

4.3. Decisions tecnològiques

En aquest apartat s'enumeren les diferents opcions tecnològiques que s'han pres per al desenvolupament del projecte.

- El **llenguatge de desenvolupament** de l'aplicació és el C# 3.0. S'ha seleccionat per ser el llenguatge per al qual Microsoft va desenvolupar la plataforma .NET i principalment perquè la major part de la documentació que es troba relacionada amb WPF és precisament amb C#. Inicialment havia pensat implementar el projecte amb Visual Basic però degut als motius exposats vaig optar pel Csharp.
- El **Sistema Gestor de Base de Dades (SGBD)** contra el que funciona l'aplicació és el Microsoft SQL Server 2005 Express. Amb la versió Express funciona correctament i no cal dir que amb qualsevol versió superior també funciona (només caldria canviar la cadena de connexió amb la bd). L'elecció d'aquest SGBD era un dels requeriments inicials del projecte però en qualsevol cas era pràcticament obligat per la seva total integració amb la plataforma .NET ja que ADO.NET forma part de les diverses tecnologies que el formen.
- La **Interfície d'Usuari (IU)** està implementada en XAML, el llenguatge de disseny de IU que forma part de Windows Presentation Foundation (WPF). En aquest cas es va decidir aplicar la tecnologia WPF per la finalitat que Microsoft la va dissenyar, i que és precisament el disseny de IU.

4.4. Diagrama estàtic de disseny

En aquest apartat es mostra el diagrama estàtic de disseny i els seus aspectes més importants.

- L'entitat GestorDisc implementa el pont entre la capa de negoci i la capa de dades.
- Les entitats Principal i ImatgesAlbum contenen atributs de tipus GestorDisc, els quals fan la funció de GestorAlbum i GestorImatgesAlbum respectivament. Aquestes dues entitats tenen relació d'associació amb la classe GestorDisc.
- Amb l'entitat ImageFile es construeixen les llistes amb les que es mostren les miniatures de les imatges de directori i d'àlbum. Per aquest motiu s'estableix una relació d'associació amb les dues a través de les entitats ImatgesDirector i ImatgesAlbum.
- De la mateixa manera, les entitats ImatgesDirector i ImatgesAlbum mantenen relació d'associació amb l'entitat Basket amb la qual s'implementa la cistella d'imatges.

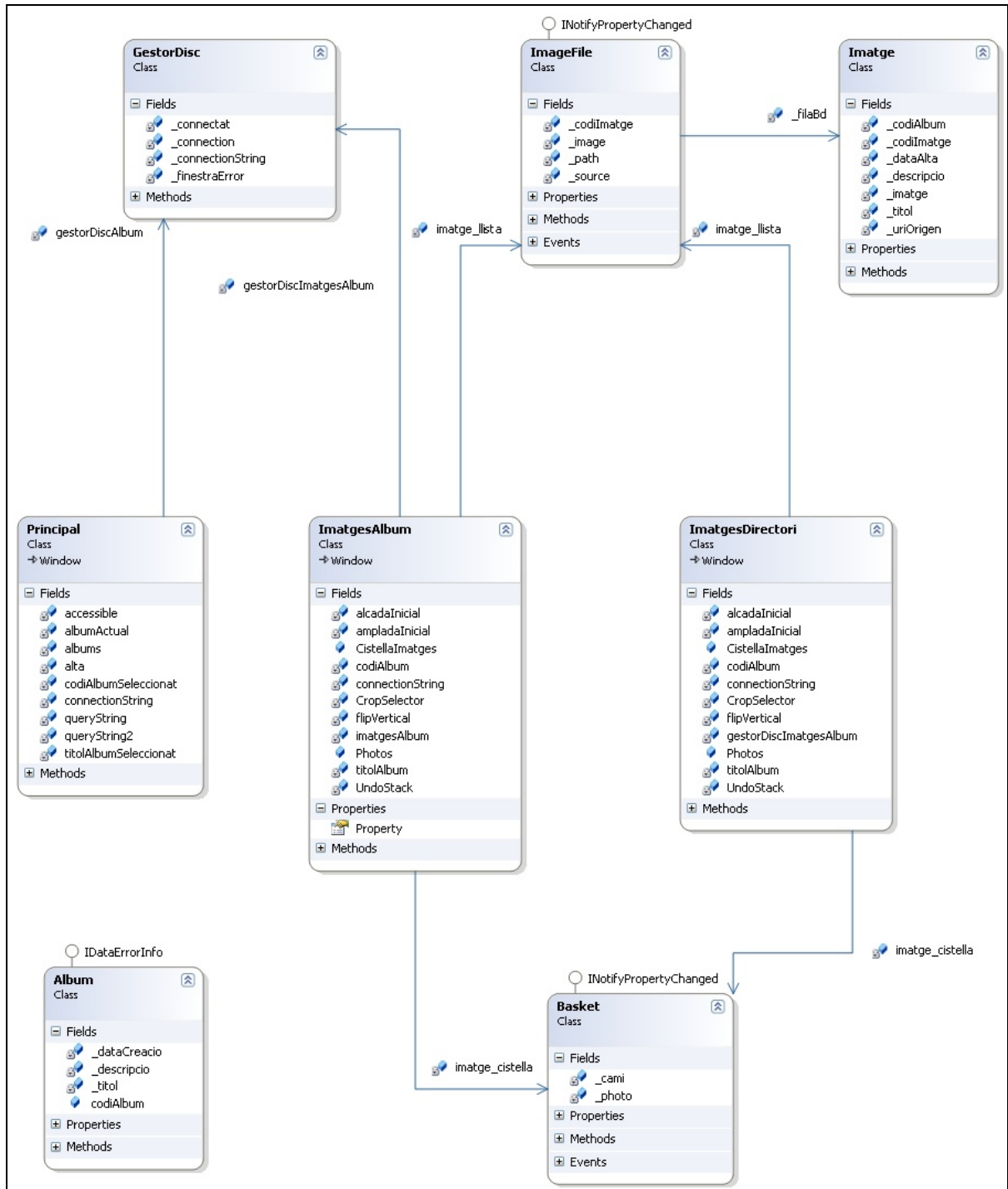


Figura 11: Diagrama estàtic de disseny

4.5. Disseny de la persistència

En aquest punt es mostra el disseny de les taules relacionals resultants. La clau primària de cada taula està formada per l'atribut subratllat. També s'indica la clau forània. La decisió de quin atribut és la clau primària d'una taula s'ha triat en funció de la semàntica dels atributs. Les claus primàries s'han implementat amb camps autonumèrics.

4.5.1. Model relacional de la base de dades

ALBUM(codiAlbum, titol, descripcio, dataCreacio)

IMATGE(codiimatge, titol, descripcio, dataAlta, imatge, uriOrigen, codiAlbum)

codiAlbum és clau forània cap a ALBUM

4.5.2. Diagrama de la base de dades

Aquest és el diagrama de la persistència.

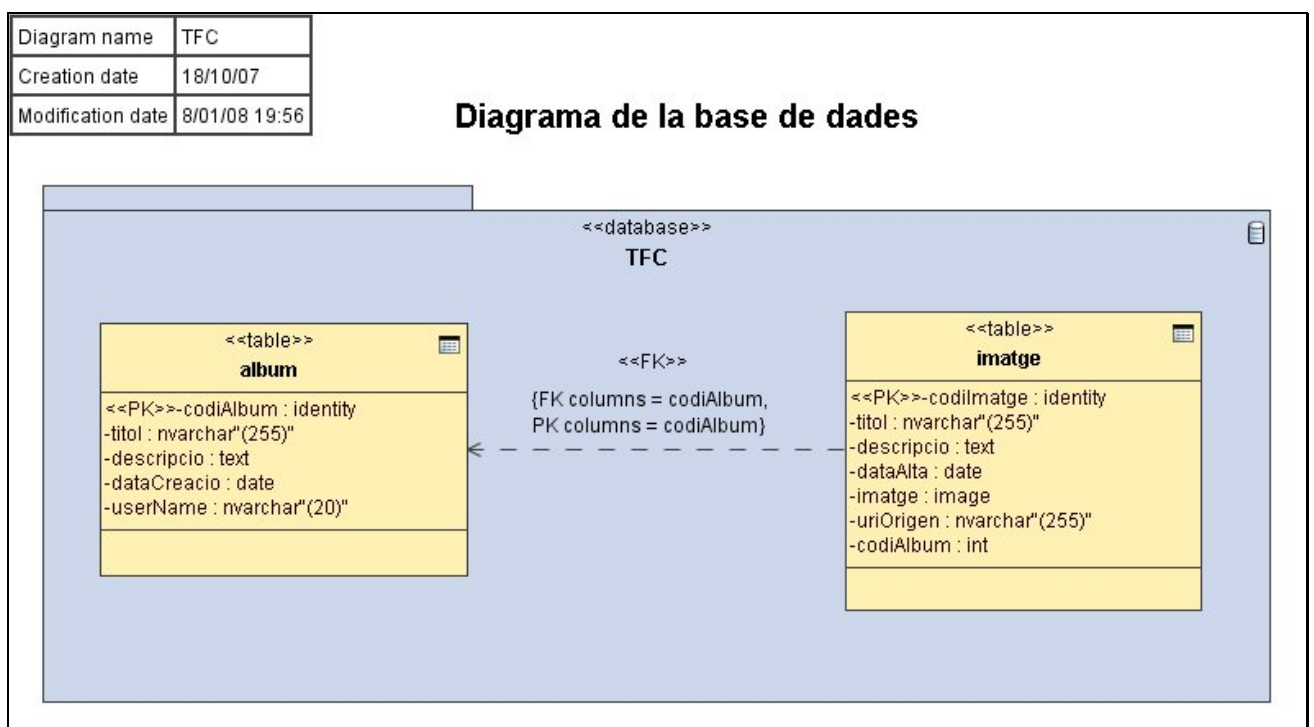


Figura 12: Diagrama de la base de dades

4.6. Prototipus de la interfície d'usuari

Durant la fase de disseny del sistema es va desenvolupar un prototipus de l'aplicació amb les finestres i components principals de la interfície d'usuari. Aquest prototipus, el qual està desenvolupat en XAML es va entregar en el lliurament corresponent i se'n mostren varies captures de pantalla a efectes informatius donat que el prototipatge és una fase prèvia a la implementació i ja no te validesa després d'aquesta.

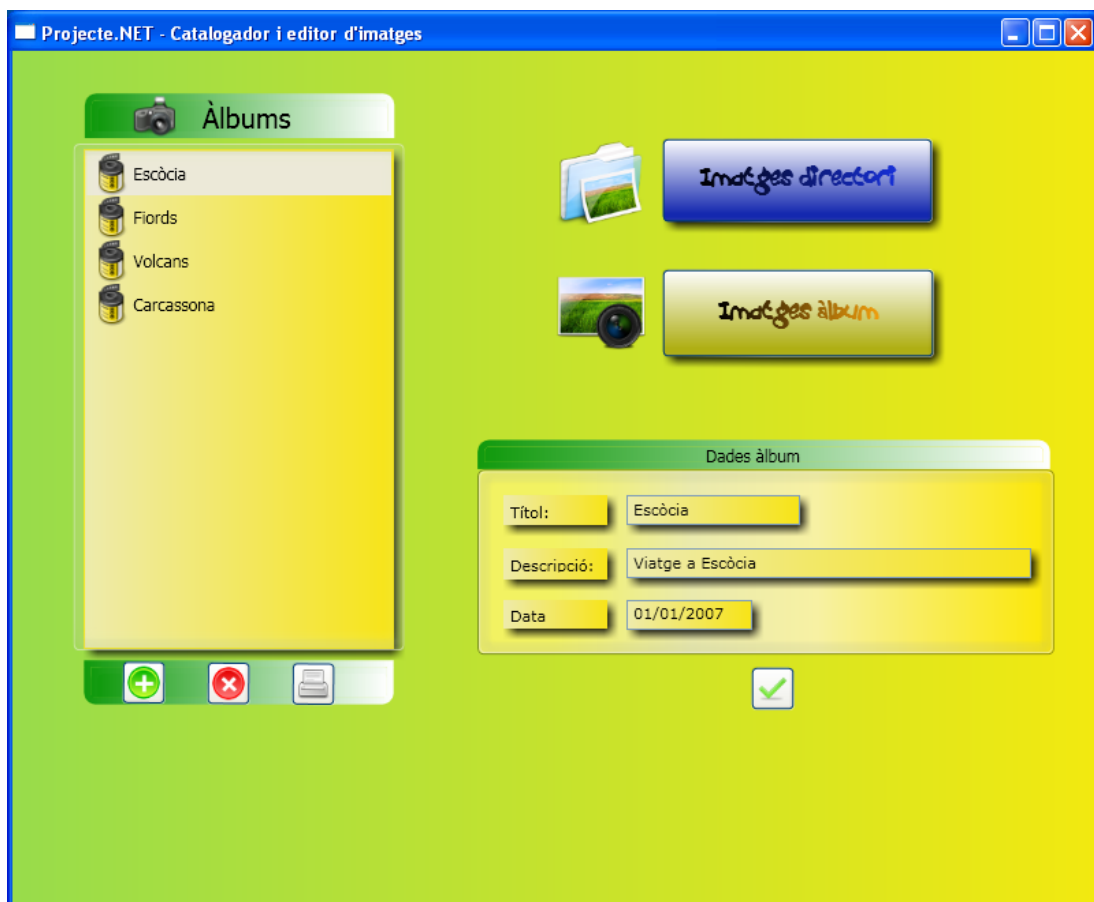


Figura 13: Pantalla principal del prototip.



Figura 14: Pantalla d'imatges directori del Prototip.



Figura 15: Pantalla d'imatges àlbum del Prototip.

5. Implementació

A continuació es vol fer constar els aspectes que més han incidit en la implementació i algunes decisions preses al respecte :

- La implementació ha sigut condicionada per la utilització de la tecnologia WPF. Aquesta tecnologia, que s'ha utilitzat a través del llenguatge XAML té una gran potència per presentar dades i per interactuar amb l'usuari però alhora té una gran complexitat pel que respecta a l'entrada de dades, validacions i en la interacció entre la capa de presentació i la capa de negoci. Aquesta dificultat i la manca de documentació ha alentit tota la fase d'implementació reduint el temps útil d'implementació, el qual de per si ja era bastant just.
- El treball amb imatges també ha condicionat en gran manera la implementació. Ha sigut complex adquirir la base necessària per treballar amb imatges per la multitud de conversions entre formats i concretament en el que respecta a les conversions necessàries per a l'aplicació de determinades eines.
- Un altre aspecte que ha incidit en la implementació i el rendiment de l'aplicació ha sigut la generació de la llista de miniatures (thumbnails) que de fet és el nucli de la gestió d'imatges. Les dues alternatives de carregar les imatges tenen els seus avantatges i inconvenients:

1. Carregar la imatge amb `BitmapFrame.Create(_source)` permet carregar les imatges molt ràpidament, per fer-ho s'utilitza la propietat `Thumbnail` que tenen les imatges, d'aquesta manera: `Image Source = "{Binding Image.Thumbnail}"`. Però els thumbnails només estan integrats en el format jpg. Per tant els altres formats es queden sense thumbnail en la llista d'imatges. L'altre inconvenient és que al carregar-los els arxius queden bloquejats i no hi ha manera de desbloquejar-los, això implica que els mètodes d'esborrar i de moure fitxer donen error perquè al estar bloquejat no es pot esborrar el fitxer.
2. Aquesta alternativa consisteix en convertir la imatge en un array de bytes abans de fer el `BitmapFrame.Create()`. D'aquesta manera els arxius no queden bloquejats i tots els formats tenen thumbnail (aquesta no s'extreu de la imatge, es crea de la manera: `Image Source = "{Binding Image}"`), però va molt més lent al carregar les imatges, fins i tot es pot quedar bloquejat si n'hi ha moltes.

S'ha optat per la solució 2 perquè aporta més avantatges tot i que és més lenta.

- Per recuperar les metadades Exif associades a la imatge succeix el mateix problema de bloqueig del fitxer. Per aquest motiu no s'ha pogut incloure aquesta informació com a atribut del tipus `ExifMetadata` de la classe `ImageFile`. S'ha optat per carregar les dades directament de la imatge en el moment que en què es demanen.
- El llenguatge de consultes LINQ que s'inclou en la versió 3.5 de la plataforma .NET és d'una gran potència ja que es poden utilitzar expressions del llenguatge per a l'accés a les dades, ja no estem restringits a SQL. Però té l'inconvenient que és complicat fer procediments genèrics perquè les consultes es construeixen de manera personalitzada, la qual cosa dificulta la reusabilitat. Per aquest motiu no s'ha pogut implementar la classe `GestorDisc` en LINQ, el qual només s'ha pogut aplicar en situacions concretes no genèriques.

6. Conclusions

- S'ha implementat un sistema catalogador d'imatges amb funcions d'edició utilitzant les tecnologies que proporciona la plataforma .NET de Microsoft, la qual cosa ha permès assolir els objectius exposats a l'inici d'aquesta memòria.
- L'arquitectura de tres capes de l'aplicació ha resultat vàlida per a la implementació dels components i s'integra totalment amb l'arquitectura de WPF i la de .NET amb la conseqüent optimització que això implica.
- S'han integrat amb èxit diferents tecnologies de Microsoft tot i que en el cas de WPF li queda molt camí per endavant si vol substituir totalment Windows Forms com a interfície d'usuari per a les aplicacions d'escriptori de Windows. El mateix succeeix amb LINQ si vol substituir els mètodes tradicionals SQL que segueixen vigents mentre el LINQ no evolucioni.
- La planificació, formació docent i experiència professional també han jugat un paper important en el desenvolupament del projecte. D'aquesta manera s'han pogut enfocar i planificar les diferents tasques des de diferents perspectives per assolir les diferents fites del treball amb les millors alternatives.
- Personalment estic molt satisfet d'aquest treball, pel tipus d'aplicació ja que el treball amb imatges es molt engrescador i per l'aprenentatge de les diverses tecnologies punteres que ofereix la plataforma .NET i que s'han treballat en el projecte. Crec que s'han assolit totalment els objectius del Treball de Fi de Carrera.

7. Glossari

- **Browse:** Mostra una llista de dades amb funcionalitat d'scroll i amb la possibilitat de seleccionar un element d'aquesta llista.
- **Bevel:** Efecte que aplica un marc als costats de la imatge.
- **Blur:** Efecte que difumina la imatge.
- **Crop:** Retallar una àrea de la imatge.
- **Emboss:** Efecte que aplica relleu a la imatge.
- **Flip:** Transformació que realitza un intercanvi d'eixos de la imatge. Aquest intercanvi pot ser tant en l'eix de les X, com en l'eix de les Y.
- **FTP:** File Transfer Protocol, protocol de transferència de fitxers.
- **IDE:** Integrated Development Environment, entorn integrat de desenvolupament d'aplicacions.
- **LINQ:** Language Integrated Query, llenguatge integrat de consultes que forma part de la plataforma .NET 3.5. LINQ suporta un model extensible molt ric que facilita la creació d'operadors molt eficients per a orígens de dades. Aquests orígens de dades poden ser objectes, XML i bases de dades.
- **Mirall (Mirror):** Efecte que mostra la imatge com si es veiés reflexada en l'aigua.
- **.NET:** Plataforma de desenvolupament de la casa Microsoft, integrada per diverses tecnologies.
- **Preview:** Presentació preliminar. Mostra en pantalla com quedarà un objecte abans d'aplicar-li un canvi o abans d'imprimir-lo.
- **Rotació:** Gira la imatge en un angle determinat.
- **Scroll:** Desplaçament d'un objecte que pot ser de manera horitzontal o vertical. En aplicació a una imatge és el desplaçament d'aquesta imatge. L'scroll també s'aplica a les llistes i als browse.
- **SGBD:** Sistema Gestor de Base de Dades. Software especialitzat en l'emmagatzament i recuperació de dades segons el model relacional.
- **Slide-show:** És una presentació que visualitza per pantalla una seqüència d'imatges, deixant entre imatge i imatge un interval de temps donat. En aquest projecte la seqüència d'imatges es pot fer tant en les imatges d'un directori com les d'un àlbum.
- **SQL:** Structured Query Language, llenguatge estructurat de consulta utilitzat per a la manipulació d'informació contra un SGBD.
- **SQL Server:** SGBD de Microsoft, el qual aglutina juntament amb Oracle la major part dels SGBD comercials implantats a nivell mundial.
- **Thumbnail:** És la miniatura d'una imatge. Són molt utilitzats en visualitzadors d'imatges perquè visualment és molt més ràpid per buscar una imatge que utilitzant imatges senceres. En aquest projecte s'utilitzen els thumbnails tant per visualitzar les imatges d'un directori com les d'un àlbum.
- **UML:** Unified Modeling Language, és un llenguatge per permet modelar, construir i documentar els elements que formen un sistema de programari orientat a objectes. S'ha convertit en l'estàndard de facto de la indústria.
- **URL:** Uniform Resource Locator, localitzador d'adreces a Internet.

- **Visual Studio:** IDE de la plataforma .NET de Microsoft. La versió actual és la 2008.
- **WPF:** Windows Presentation Foundation, tecnologia proporcionada per la plataforma .NET 3.0 i 3.5 per al disseny d'interfícies riques d'usuari.
- **XAML:** eXtensible Application Markup Language, Llenguatge extensible de marques, derivat de XML per al disseny d'interfícies d'usuari i que forma part de la tecnologia WPF.
- **XML:** eXtensible Markup Language, llenguatge extensible de marques a l'estil HTML. Actualment s'ha convertit en un estàndard de facto per a l'intercanvi d'informació entre diferents sistemes.
- **Zoom:** Ampliació o reducció en pantalla de la mida de la imatge.

8. Bibliografia

A banda de la completa informació i exemples inclosos en el Visual Studio, s'han consultat multitud de pàgines web de les quals he fet una selecció de les més destacables, juntament amb diversos materials consultats.

Origen informació	Descripció
Internet	C# Corner Web de desenvolupadors en .NET http://www.c-sharpcorner.com
Internet	The Code Project Web de desenvolupadors en .NET http://www.codeproject.com
Internet	El Gille Web sobre .Net i les seves tecnologies http://www.elquille.net
Internet	Longhorn Corner Web de desenvolupadors en .NET http://www.longhorncorner.com
Internet	Blogs Microsoft Developer Network Blogs de desenvolupadors de Microsoft http://blogs.msdn.com
Internet	Microsoft Developer Network Web de desenvolupadors de Microsoft http://msdn2.microsoft.com/es-es/default.aspx
Llibre	Advanced C# Programming Editorial: McGraw Hill Autor: Paul Kimmel
Llibre	Expert C# 2005 Business Objects Editorial: Apress Autor: Rockford Lhotka
Llibre	Visual C# 2005 Recipes - A Problem-Solution Approach Editorial: Apress Autors: Allen Jones i Matthew MacDonald
Llibre	Professional WPF Programming Editorial: Apress Autors: Chris Andrade, Shawn Livermore, Mike Meyers, Scott Van Vliet
Llibre	Materials de diverses assignatures de la UOC