

PHOTO STORE

Galeria virtual d'imatges

David Torné Llorens

Enginyeria tècnica en informàtica de gestió

Consultor: Jordi Ceballos Villach

Gener 2008

*Als meus pares, per tot el suport donat des de l'inici fins al final d'aquesta cursa
A la Laura, per aquesta última empenta*

Índex de continguts

INTRODUCCIÓ	6
DESCRIPCIÓ DEL PROJECTE	6
CONTEXT I JUSTIFICACIÓ DEL PROJECTE	6
OBJECTIUS	7
PLANIFICACIÓ TEMPORAL	7
ESTRUCTURACIÓ DEL PROJECTE	7
DETALL D'ACTIVITATS	7
DIAGRAMA DE GANNT	8
EINES UTILITZADES	9
TECNOLOGIES	9
PRODUCTES OBTINGUTS	11
ESTRUCTURA D'AQUEST DOCUMENT	12
ANÀLISI DEL SISTEMA	13
CASOS D'US D'USUARIS VISITANTS	13
CERCAR	13
VISUALITZAR	14
COMENTAR	15
LLEGIR COMENTARIS	15
PUNTUAR	16
ADVERTIR	16
CREAR USUARI	16
RANKING	17
CASOS D'US D'USUARIS REGISTRATS	18
RECUPERAR DADES D'USUARI	18
MANTENIMENT DEL COMPTE DE USUARI	19
MANTENIMENT D'ÀLBUMS	19
MANTENIMENT DE CATEGORIES	20
VALIDACIÓ EN EL SISTEMA	20
PUBLICACIÓ D'IMATGES	21
REGULAR COMENTARIS	22
REGULAR ADVERTÈNCIES	23
CLASSIFICACIÓ D'IMATGES	23
CONSULTA D'ESTADÍSTIQUES	23
HERÈNCIA ENTRE ELS VISITANTS I USUARIS REGISTRAT	24
CASOS D'US D'USUARIS ADMINISTRADORS	24
VALIDACIÓ EN EL SISTEMA	25
GESTIONAR LES DADES DEL COMPTE	25
GESTIONAR ADVERTÈNCIES	25
DESACTIVAR USUARIS	26
CONSULTA D'ESTADÍSTIQUES	27
DISSENY	28
ARQUITECTURA	28
ARQUITECTURA FÍSICA DE L'APLICACIÓ	28

ARQUITECTURA LÒGICA DEL SISTEMA	28
CLASSES ENTITAT	30
DIAGRAMA	30
DESCRIPCIÓ DE LES CLASSES ENTITAT	31
VISITANT, USUARI, NOVALIDAT, REGISTRAT, ADMINISTRADOR	31
ÀLBUM, CATEGORIA	32
IMATGE	32
CLASSIFICADOR	32
COMENTARI	32
PUNTUACIÓ	33
ADVERTÈNCIA	33
DIAGRAMA DE CLASSES CONNECTORES	33
DESCRIPCIÓ DE LES CLASSES CONNECTORES	33
DESCRIPCIÓ DE ELEMENTS COMUNS EN LES CLASSES CONNECTORES	33
DESCRIPCIÓ DE ELEMENTS PARTICULARS EN LES CLASSES CONNECTORES	36
CLASSES DE PROPÒSIT ESPECIAL	41
ESTADÍSTIQUES	41
BUSCADOR	41
DISSENY DE LA BASE DE DADES	43
DIAGRAMA ENTITAT/RELACIÓ	43
DESCRIPCIÓ D'ENTITATS I ATRIBUTS	43
TAULA USUARIS	44
TAULA ADVERTENCIA	44
TAULA PUNTUACIO	44
TAULA COMENTARIS	45
TAULA ÀLBUM	45
TAULA CATEGORIA	45
TAULA ASSIGNACIO	46
TAULA IMATGES	46
TAULA CLASSIMG	46
TAULA CLASSIFICADOR	46
IMPLEMENTACIÓ	47
SISTEMA DE GESTIÓ D'USUARIS	47
ROLS I RESTRICCIONS	47
PROCÈSSOS I ACCIONS VINCULATS AL SISTEMA MEMBERSHIP	47
ACCÉS A DADES	49
PHOTOSTORE DATA LAYER	49
CLASSES CONNECTORES	50
OBJECTDATASOURCE	50
PAGINACIÓ	51
IMPLEMENTACIÓ DE SERVEIS	52
FORMATAT DE LES IMATGES	53
UTILITZACIÓ DE LA TECNOLOGIA AJAX EN L'APLICACIÓ	55
AUGMENTAR LA INTERACTIVITAT DE L'APLICACIÓ	55
MILLORAR L'ERGONOMIA DE LES INTERFÍCIES	56
COMPLEMENTES ADDICIONALS AFEGITS A L'APLICACIÓ	57
GENERADOR DE GRÀFICS	57
DESCRIPCIÓ DEL PROCÉS	57
INCONVENIENTS	57

LÍNIES DE DESENVOLUPAMENT FUTUR **58**

CONCLUSIONS **58**

BIBLIOGRAFIA **59**

Introducció

Descripció del projecte

Context i justificació del projecte

El cercador **infocerca.cat** ha decidit oferir un nou servei als seus usuaris actuals, ampliant la seva presència a la xarxa amb un nou servei. El portal ha volgut crear una web que permeti publicar, als usuaris registrats en aquest servei, les seves fotografies en la xarxa. El gran atractiu del nou servei es disposar de les galeries d'imatges des de qualsevol lloc i a qualsevol hora.

El servei s'anomenarà Photo Store. Oferirà a l'usuari la possibilitat de crear la seva pròpia galeria virtual d'imatges, classificant-les en àlbums dividits en categories per oferir una catalogació molt més específica. Els àlbums els subdividim públics per a que qualsevol visitant de la web (registrat o no) pugui tenir la possibilitat d'accedir-hi, visualitzar i imprimir les fotos; o privats per a l'ús exclusiu de l'usuari que els publica.

Més enllà d'oferir un servei de les característiques que s'han descrit, el que realment es vol, es construir una comunitat web d'usuaris que comparteixin el seu interès per la fotografia. Per tal d'assolir aquest objectiu, es crearan un seguit de complements a les funcionalitats principals que ajudaran a fomentar aquest sentiment de comunitat, com ara un sistema de puntuació i classificacions de les imatges publicades, o la possibilitat d'afegir comentaris en les fotografies d'àlbums públics. Si hi ha una bona acceptació d'aquestes funcionalitats es poden complementar amb altres de similars en una segona fase.

L'equip de desenvolupament d'infocerca.cat, ha tingut en compte els problemes que pot generar una aplicació com aquesta tals com, la publicació d'imatges inadequades o la creació de comentaris ofensius en poden ser un exemple. Per tal de minimitzar aquests problemes es crearà un sistema d'advertència, al qual tindrà accés qualsevol visitant de la web. Una vegada tramitat l'advertiment, aquest serà gestionat, per l'equip d'administradors, que restringirà l'accés o eliminarà el contingut no desitjat.

En conclusió, volem oferir una eina senzilla i atractiva que en permeti una ràpida popularització. A la vegada esperem augmentar el nombre d'usuaris del nostre portal infocerca.cat a través de la creació d'aquesta comunitat virtual d'usuaris.

Objectius

L'objectiu principal del projecte és la creació d'un entorn intuïtiu i senzill d'utilitzar per als usuaris. Un entorn utilitzable, amb una base de coneixements d'informàtica a nivell d'usuari i sense tenir que rebre cap mena de formació.

A nivell tècnic – i també econòmic – es pretén desenvolupar una web senzilla d'administrar i fàcil de mantenir. Això implica, per una part, desenvolupar un entorn d'administració que simplifiqui al màxim les tasques a realitzar pels supervisors. Per l'altra banda i en referència al desenvolupament, ens proposem utilitzar una arquitectura en capes (presentació – negoci - dades), que permeti encapsular el codi de tal manera que minimitzi els costos de correcció, modificació i ampliació del programari.

Tecnològicament, tenim com a objectiu, la utilització de la tecnologia .NET, a través del entorn Visual Studio, de la forma més òptima possible consolidant els coneixements en l'entorn, el llenguatge escollit, així com la tecnologia Ajax, utilitzada per potenciar la interactivitat del portal.

Planificació temporal

Estructuració del projecte

Estructurem el desenvolupament del projecte en un esquema de cicle de vida clàssic. Degut a les limitacions de temps hem hagut d'adaptar el cicle en cascada limitant-lo ha les etapes de:

1. Anàlisi previ: Pla de treball.
2. Disseny: On s'han unit les fases d'anàlisi de requisits i disseny de l'aplicació.
3. Implementació: Implementació de l'aplicació, proves i documentació¹.
4. Documentació: Considerem la memòria com la documentació del projecte.

Detall d'activitats

A continuació es detallen les tasques realitzades en cadascunes de les fases:

1. Fase d'anàlisi previ
 - a. Elecció del projecte a realitzar.
 - b. Valoració de la tecnologies a utilitzar.

¹ Documentació entregada amb l'aplicació. Manual d'instal·lació i guia d'ús.

- c. Obtenció del software i complements necessaris.
 - d. Realització de la temporització del treball.
 - e. Realització del pla de treball.
2. Disseny
- a. Realitzar l'anàlisi funcional de la aplicació.
 - b. Realitzar el disseny de l'aplicació.
 - c. Creació del pilot de l'aplicació.
3. Implementació
- a. Creació de la base de dades.
 - b. Implementació de l'estructura de classes i Serveis
 - c. Disseny i creació de les interfícies d'usuari.
 - d. Proves unitàries
 - e. Redacció del manual d'instal·lació.
 - f. Creació del paquet d'instal·lació.
4. Documentació
- a. Redacció de la memòria.
 - b. Creació de la puntuació.

Diagrama de Gantt

A continuació es presenta la planificació temporal del projecte en forma de diagrama de Gantt. En ell s'exposen les tasques a realitzar, la seva duració, les dates d'inici i final i la tasca precedent. Les subtasques d'una tasca apareixen tabulades.

Nº Tasca	Denominació ² de la tasca	Duració	Inici	Final	Tasca Precedent
1	Inici projecte	1 dies	20/09/2007	20/09/2007	
2	Pla de treball	12 dies	21/09/2007	02/10/2007	1
3	Escollir el projecte a realitzar	1 dies	21/09/2007	21/09/2007	1
4	Redacció prèvia d'especificacions	4 dies	22/09/2007	25/09/2007	3
5	Redacció del pla de treball	7 dies	26/09/2007	02/10/2007	4
6	Disseny de l'aplicació	19 dies	03/10/2007	21/10/2007	2
7	Document d'anàlisi	6 dies	03/10/2007	08/10/2007	
8	Document de disseny	8 dies	09/10/2007	16/10/2007	7
9	Disseny de l'estructura de classes	5 dies	09/10/2007	13/10/2007	7
10	Disseny del diagrama ER	1 dies	14/10/2007	14/10/2007	9
11	Disseny dels diagrames dinàmics	2 dies	15/10/2007	16/10/2007	10
12	Disseny de la maqueta gràfica	5 dies	17/10/2007	21/10/2007	11
13	Implementació³	61 dies	22/10/2007	21/12/2007	12
14	Sistema de creació d'usuaris i validació	5 dies	22/10/2007	26/10/2007	12
15	Sistema de classificació d'imatges (Àlbums i	7 dies	27/10/2007	02/11/2007	14

² La denominació de les tasques és suficientment explícita per descriure'n el contingut.

³ En l'apartat d'implementació, cada sistema correspon a cadascuna de les funcionalitats a implementar.

	categories)				
16	Sistema de publicació d'imatges	9 dies	03/11/2007	11/11/2007	15
17	Sistema de cerca d'imatges	7 dies	12/11/2007	18/11/2007	16
18	Sistema de visualització d'imatges	7 dies	19/11/2007	25/11/2007	17
19	Sistema de comentaris i votació	7 dies	26/11/2007	02/12/2007	18
20	Sistema d'administració	10 dies	03/12/2007	12/12/2007	19
21	Redactar ajuda	4 dies	13/12/2007	16/12/2007	20
22	Probes	5 dies	17/12/2007	21/12/2007	21
23	Lliurament final	19 dies	22/12/2007	09/01/2008	22
24	Redacció de la memòria	12 dies	22/12/2007	02/01/2008	22
25	Creació de la presentació	7 dies	03/01/2008	09/01/2008	24

Eines utilitzades

Per a la creació del projecte s'han utilitzat les eines que s'enumeren a continuació. Aquestes són presentades en grups, segons l'aspecte del desenvolupament que han ajudat a realitzar:

Eines base per el desenvolupament de l'aplicació:

- Microsoft Visual Studio 2005
- Microsoft Sql Server 2005 Express Edition
- Microsoft Sql Server Management Studio Express Edition

Complements del disseny de l'aplicació i construcció del pilot:

- Adobe Fireworks 8.0: Edició de les imatges.
- Adobe DreamWeaver 8.0: Editor Web utilitzat per a la creació del pilot.

Eines utilitzades com a suport a la planificació i a la documentació del projecte:

- Microsoft Word: Documentació del projecte
- Microsoft Visio: Disseny del conjunt de diagrames necessaris per la construcció de l'aplicació.
- Microsoft Project: Eina de suport per a la creació de la planificació de l'aplicació.
- Microsoft Powerpoint: Eina usada per a la realització de la presentació.

Tecnologies

A continuació es descriuen les tecnologies utilitzades en el desenvolupament del projecte:

Microsoft .NET

L'arquitectura .NET és la tecnologia desenvolupada per Microsoft per tal de simplificar els processos de creació de software independentment de l'entorn d'execució. Ja sigui per entorns d'escriptori, web o dispositius mòbils, Microsoft ens

proporciona Visual Studio, un entorn de desenvolupament únic per tots aquests àmbits.

VB.NET

Per implementar l'aplicació escollim VB.NET com a llenguatge ja que es considera l'evolució dins de la plataforma .NET del llenguatge Visual Basic clàssic, amb el qual estem habituats a treballar i en coneixem la sintaxi. És tracta d'un llenguatge senzill i fàcil que ens permetrà codificar l'aplicació d'una forma senzilla, clara i intuïtiva.

ASP.NET

Tecnologia creada per Microsoft per a la creació de pàgines web dins el seu entorn .NET. Proporciona una gran font de recursos per a enriquir els dissenys amb controls ActiveX. Aquests controls són declarats dins el codi HTML de la pàgina mitjançant tags que seran executats pel servidor al rebre la sol·licitud del navegador. Una vegada executat el control es genera el codi HTML necessari per a que pugui ser mostrat pel navegador.

Accés a dades

Per a l'accés a dades utilitzarem la tecnologia ADO.NET. Per simplificar-ne l'ús, en definirem els mètodes i funcionalitats a través d'un DataSet. Aquesta col·lecció de funcionalitats s'encapsulen en un conjunt de classes de visual basic, que anomenem connectores, que composaran la capa de dades i seran cridades des de la interfície d'usuari per tal d'operar amb la informació emmagatzemada. Finalment, l'emmagatzematge de dades es farà en una base de dades Microsoft SQL Server 2005.

AJAX (Asynchronous Java Script And XML)

AJAX és una combinació de les tecnologies XHTML, amb les fulles d'estils en cascada (css). Model de documents amb objectes (DOM), accedint i modificant els objectes del document amb un llenguatge de programació del tipus Script, sent el més utilitzat és JavaScript. L'objecte XMLHttpRequest, utilitzat per a realitzar l'intercanvi d'informació amb el servidor. Finalment s'utilitza XML com a format per rebre les dades del servidor.

L'objectiu d'aquesta tecnologia és dota les aplicacions d'una major interactivitat. En una web convencional, cada cop que hi ha una petició de dades al servidor i aquest respon, el client (el navegador web) es veu obligat a actualitzar la pàgina per

mostrar els resultats rebuts. AJAX és una metodologia per l'obtenció d'aplicacions Web interactives mitjançant l'establiment d'un sistema capaç de mantenir una comunicació asíncrona amb el servidor, de tal manera que sigui possible actualitzar la interfície web sense necessitat de recarregar-la. Per dur a terme tot aquest procés és clau el paper de l'objecte XMLHttpRequest. Aquesta interfície és la base per realitzar peticions HTTP i HTTPS al servidor.

En el nostre cas, l'ús d'aquesta tecnologia és fonamental. Pretenem construir una aplicació similar a les aplicacions d'escriptori utilitzades com a un àlbum o classificador de fotografies, però en un marc de treball web. Per tant la interactivitat és un factor clau.

Amb la intenció d'obtenir aquest complement d'usabilitat i ergonomia farem ús de les extensions ASP.NET 2.0 AJAX Extensions. Aquest complement ens donarà un conjunt de recursos per millorar la qualitat de la nostra aplicació mitjançant una col·lecció de llibreries i controls.

Un altre aspecte important és el disseny i la capacitat de l'aplicació d'oferir una aparença dinàmica i atractiva que faciliti la feina a l'usuari. Amb la intenció d'obtenir un plus funcional i de potenciar el treball de l'equip de desenvolupament, incorporarem al projecte AJAX Control Toolkit. Aquest paquet està compost d'un conjunt de controls basats en tecnologia AJAX i d'extensions de controls base oferts per ASP.NET potenciant les seves funcionalitats amb utilitats complementaries.

Serveis Web

Els serveis web són un conjunt de protocols que permeten l'intercanvi de dades entre les aplicacions, independentment sobre quina plataforma o llenguatge s'hagin desenvolupat.

La nostra aplicació farà ús d'aquest recurs com a complement en l'utilització de determinats controls del AJAX Control Toolkit. Aquests controls requereixen l'accés al flux de dades per mitja d'un servei web.

Productes obtinguts

A continuació s'enumera el conjunt de productes obtinguts en el moment de finalitzar el projecte. Aquí s'inclouen el software i documentació enviat en l'entrega final i la entrega en cadascuna de les fites de control del desenvolupament del projecte.

1. Pla de treball
2. Document d'anàlisi de requisits.

3. Document de disseny del sistema.
4. Paquet PhotoStore compostat per
 - a. Aplicació Web: Composada pel conjunt d'interfícies d'usuari. Objectes i eines d'accés a dades i lògica de negoci.
 - b. Estructura de bases de dades necessàries per l'emmagatzematge de la informació gestionada per l'aplicació.
 - c. Manual d'instal·lació i guia d'ús de l'aplicació.
5. Memòria final.
6. Presentació multimèdia del producte.

Estructura d'aquest document

Aquest document és una compilació de les tasques realitzades en l'execució en la planificació i execució d'aquest projectes. En els següents apartats s'exposaran els requisits de la aplicació (Capítol dedicat a l'anàlisi). Una descripció de l'entorn d'execució, l'arquitectura, i el disseny de l'aplicació (Capítol dedicat al disseny). En el capítol dedicat a la implementació és comentaran les decisions comentades en quan a estructuració de l'aplicació, els recursos utilitzats per a generar les interfícies d'usuari. Finalment es mostrarà com interactuen les classes connectores, creades per accedir a les dades de l'aplicació i la capa de dades generada a partir del format XSD, recurs propi de Visual Studio.

La part final de la memòria recopila les conclusions del projecte. En aquestes conclusions es fa esment a les variacions de la planificació inicial respecte al producte final i les motivacions que han dut a realitzar-les. Aquests comentaris es complimentaran amb una breu exposició de com l'execució del projecte, amb l'ús de la tecnologia AJAX, a canviat la percepció inicial que es tenia de les aplicacions en entorn web.

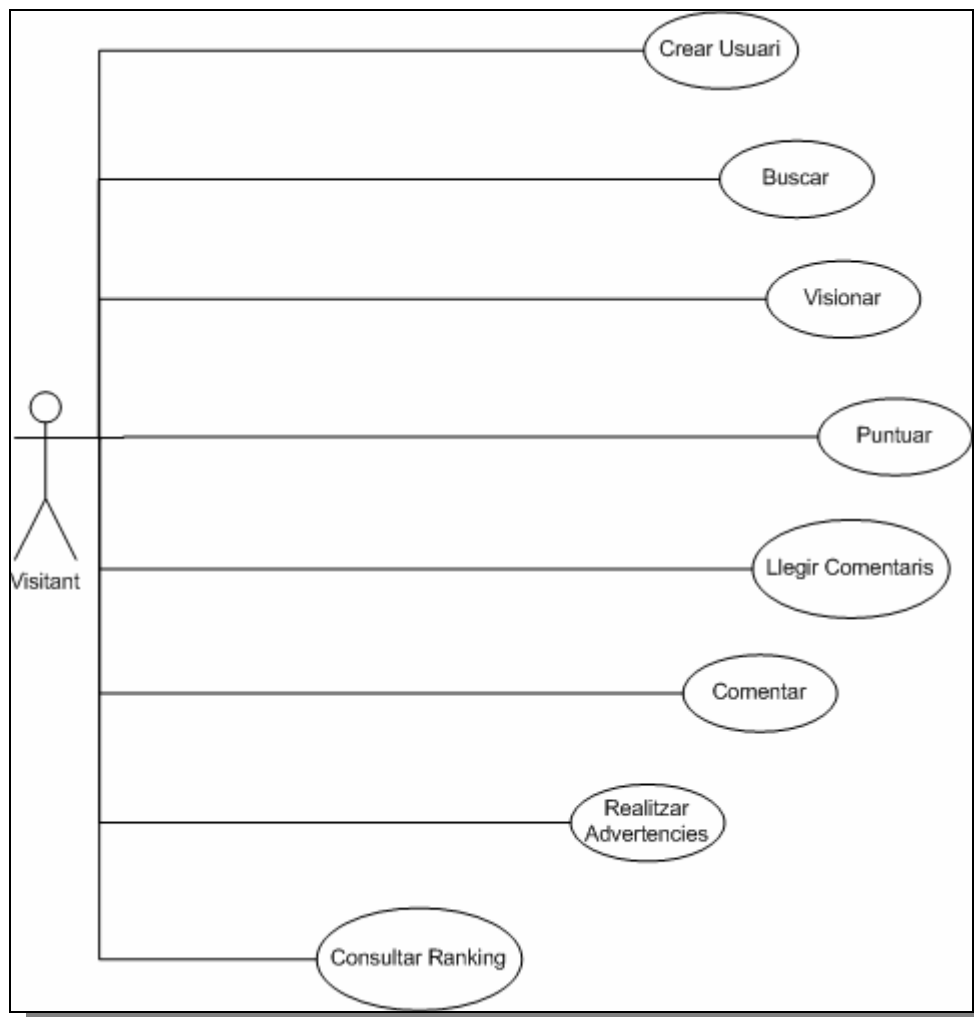
Conjuntament a la conclusió s'adjuntarà una bibliografia de totes les publicacions i llocs web d'on s'ha obtingut informació.

Anàlisi del sistema

A continuació és presenten els casos d'ús i una descripció de cadascun d'ells. Aquests es divideixen segons l'actor⁴ que interactua amb el cas d'ús.

Casos d'ús d'usuaris visitants

L'actor Visitant és aquell que fa ús del sistema sense haver-s'hi validat. Les funcionalitats a les que té accés es descriuen en els casos d'ús següents:



(Figura 1) Diagrama de casos d'ús dels usuaris no validats

Cercar

Per tal d'accedir a les imatges d'una forma fàcil i directa, es permet al visitant realitzar una cerca per diversos conceptes. En la pàgina principal o la pàgina de visualització d'imatges, trobarem una secció que ens permet entrar un conjunt de

⁴ Com és comenta al llarg del document comptem amb els perfils d'usuari visitant, usuari registrat i administrador

paraules clau, separades per un espai, i una llista desplegable que ens permetrà cercar per diferents criteris. L'aplicació generarà una col·lecció d'imatges coincidents amb els criteris de la cerca que es posaran a disposició de l'usuari per a que les pugui visualitzar.

Es podran realitzar cerques per els següents criteris:

1. Usuari (Nom d'usuari): Nom amb que l'usuari es valida al sistema.
2. Nom persona (Nom natural del usuari): Nom de la persona que utilitza el servei i que s'entra en el perfil d'usuari, un cop donat d'alta.
3. Àlbum: Nom del àlbum.
4. Categoria: Nom de la categoria a la que s'assigna la imatge dins l'àlbum
5. Classificadors (Tags): Conjunt de paraules classificatòries que s'assignen a la imatge.

La cerca per categoria i per classificadors, ens retornarà la col·lecció d'imatges que coincideixin amb criteris de la cerca. Pel contrari si cerquem usuari o nom persona, ens retornarà una pàgina de resultats amb tots els usuaris trobats que coincideixin amb els criteris de cerca. A partir d'aquí podrem seleccionar el que desitgem i visualitzarem la seva col·lecció d'àlbums i de cadascun d'ells, fent un clic damunt, les imatges que el componen.

La cerca d'àlbums permet mostrar la col·lecció d'àlbums que coincideixen amb el filtre aplicat a la cerca. Fent clic damunt, visualitzem les imatges.

En cas de que la cerca no retorni resultats se'ns presenta una pantalla advertint de la incidència i se'ns convida a tornar a la pantalla principal mitjançant un vincle. Totes les cerques només retornarà imatges publicades en àlbums públics⁵.

Visualitzar

Una vegada realitzada la cerca es retornaran un conjunt d'imatges que l'usuari podrà visualitzar en diferents formats:

1. Galeria: Les imatges són mostrades en conjunt, en una mida més reduïda que la de l'àrea total de la imatge, i estandarditzat per tal d'oferir una presentació similar al d'una pàgina d'àlbum de fotos. Apareixen distribuïdes en una taula, en files i columnes, on cada imatge ocupa una cel·la. Conjuntament en la imatge es presenta el nom d'usuari que la publicat i el nom del àlbum i la categoria a la que pertany.
2. Detall: Es presenta un llistat/Informe, mostrant una miniatura de la imatge i diverses dades d'aquesta. En aquest llistat, la informació serà presentada en fileres, on en primer lloc es mostrarà la imatge i seguidament dades

⁵ L'usuari registrat que publiqui les fotos pot classificar l'àlbum com a públic, obert a tothom o privat, restringint-ne la visualització.

d'interès com la data de publicació, el nom del àlbum i la categoria a la que pertany i la seva descripció.

3. Visió complerta: En les dues anteriors presentacions podem seleccionar qualsevol imatge i aquesta es presentarà individualment, en la seva mida original. Podrem visualitzar les altres imatges del grup, mitjançant els botons anterior, següent, primer i últim. En aquest mode de visualització serà on se'ns permeti puntuar, comentar i enviar advertiments als administradors. Qualsevol d'aquestes tasques es realitzaran fent clic al link corresponent al marge esquerra de la pantalla. Mitjançant el link amb la llegenda [Llegir comentaris](#)⁶ és permetrà visualitzar els comentaris fets a la imatges per altres usuaris. En cas de no disposar de comentaris l'opció ja no apareixerà.
4. Slideshow: Aquesta presentació, mostra les imatges una per una, carregant-les de forma automàtica, deixant un interval de temps abans de carregar la següent. S'oferirà l'opció de parar la reproducció automàtica de les imatges i visualitzar-les de forma manual, passant a la següent o a l'anterior mitjançant un clic als botons que s'assigna aquesta funció. En qualsevol dels dos modes de treball la imatge es presenta en una nova finestra en un tamany de pantalla complerta.

En el cas de les cerques per usuari, nom persóna o nom del àlbum, l'única visualització que se'ns oferirà serà la presentació de galeria.

Comentar

Les imatges públiques podran ser comentades per qualsevol usuari, registrat en el sistema o no, accedint a la imatge mitjançant el mode de presentació visió complerta i fent clic al vincle corresponent en el marge esquerra de la imatge.

Al fer clic s'obrirà un panell emergent i modal, on ens permetrà entrar un text de 255 caràcters. Al guardar el comentari, guardarà el text i retornarà a la imatge.

En el cas que es tracti d'un usuari registrat que hagi iniciat sessió, quedarà guardat el nom d'usuari.

Perquè el comentari sigui visible als altres usuaris, s'haurà d'haver revisat per l'usuari propietari dins l'entorn d'usuaris registrats.

Llegir comentaris

Dins del mode visió complerta, trobem l'opció veure comentaris de la imatge. Aquesta opció només apareix en el cas que la imatge tingui comentaris ja revisats per l'usuari propietari de la imatge. Si és aquest el cas, al fer clic en el link

⁶ A la dreta i entre parèntesis s'oferirà el nombre de comentaris totals dels que disposa la imatge.

corresponent, s'obrirà una nova finestra amb el llistat complet dels comentaris, paginats de 10 en 10.

Puntuar

Un dels atractius del sistema Photostore és el seu sistema de rankings. Els usuaris o visitants podran valorar les imatges que estiguin veient. En el mode de presentació de visió completa, disposem d'un apartat en el marge superior dret de la imatge, on apareixen dues fileres de cinc estrelles. La primera fila, serà utilitzada per valorar la imatge. Presentarà la mateixa renglera de cinc icones, en les quals podrem clicar per assignar una puntuació. Al fer clic en una o altra icona s'establirà una puntuació de 1 a 5. El valor es calcula d'esquerra a dreta, com més a la dreta esta la icona més alta serà la puntuació. Sota les icones apareixeran el nombre de vots rebuts.

La segona filera, mostrarà la valoració actual. Les cinc icones amb forma d'estel, apareixeran plenes de color fins al punt que marqui la mitja de les valoracions fetes fins al moment i no se'n podrà editar el valor. A la part inferior apareix la mitja numèrica del nombre de vots rebuts.

Advertir

No existeix cap control en la publicació d'imatges per part dels usuaris registrats, per tant, és relativament senzill introduir imatges amb un contingut inapropiat dins del sistema. Per tal d'intentar apartar les imatges amb contingut ofensiu de les galeries públiques dels usuaris, s'implementarà un sistema d'avis i supervisió de les imatges.

Un visitant qualsevol que visualitzi una col·lecció d'imatges, ja sigui com a resultat d'una cerca en el sistema o entrant en els àlbums d'un usuari en concret, podrà enviar una advertència als administradors del sistema. Per fer-ho s'haurà de mostrar la imatge en el mode visió completa. Al marge esquerra de la pantalla es trobarà el link [Advertencia](#). al fer click es farà visible un panell en la part superior de la pàgina, on podrem escollir el motiu pel qual es realitza l'avis (Violència / Nuesa / Sexe explícit / Ofensiu / Altres) i podrem entrar una breu descripció de per que el considerem ofensiu. Al fer clic en el botó acceptar s'enviarà l'avis als administradors del sistema i al usuari propietari de la imatge perquè prenguin les mesures que creguin convenients.

Crear usuari

Un usuari visitant es podrà donar de alta en el sistema fent click a la imatge/vincle col·locada a la pàgina principal amb aquesta finalitat. El vincle ens dona accés a l'apartat de registre d'usuaris, on es realitzarà la primera fase del procés. En ell

donarem d'alta el nostre compte d'usuari, complimentant les dades referents al nom d'usuari, mot de pas, correu electrònic, així com la pregunta i la resposta per recuperar les dades del compte. Una vegada completades les dades i fent clic al botó acceptar, ens validarà el compte. El nom d'usuari escollit ha de ser únic, si ja existeix, no ens permetrà continuar endavant fins que entrem un nom d'usuari no escollit.

Una vegada creat el compte, entrarem automàticament al entorn d'usuari, on se'ns obligarà a complimentar les dades personals de l'usuari. Fins que aquestes no siguin complimentades no es podrà començar a utilitzar el sistema. Per obligar a l'usuari a fer-ho, és la única opció que es deixarà activa⁷.

Ranking

El sistema ofereix la possibilitat de consultar classificacions de les imatges i àlbums. Els usuaris que no estiguin registrats en el sistema tindran accés a un conjunt bàsic de rankings:

1. Les imatges més valorades pels visitants.
2. Les imatges més visitades pels visitants.
3. Les imatges amb la millor ràtio votació dels visitants
4. Les imatges més valorades pels usuaris registrats.
5. Les imatges amb la millor ràtio votació dels usuaris
6. Les imatges més visitades pels usuaris registrats.

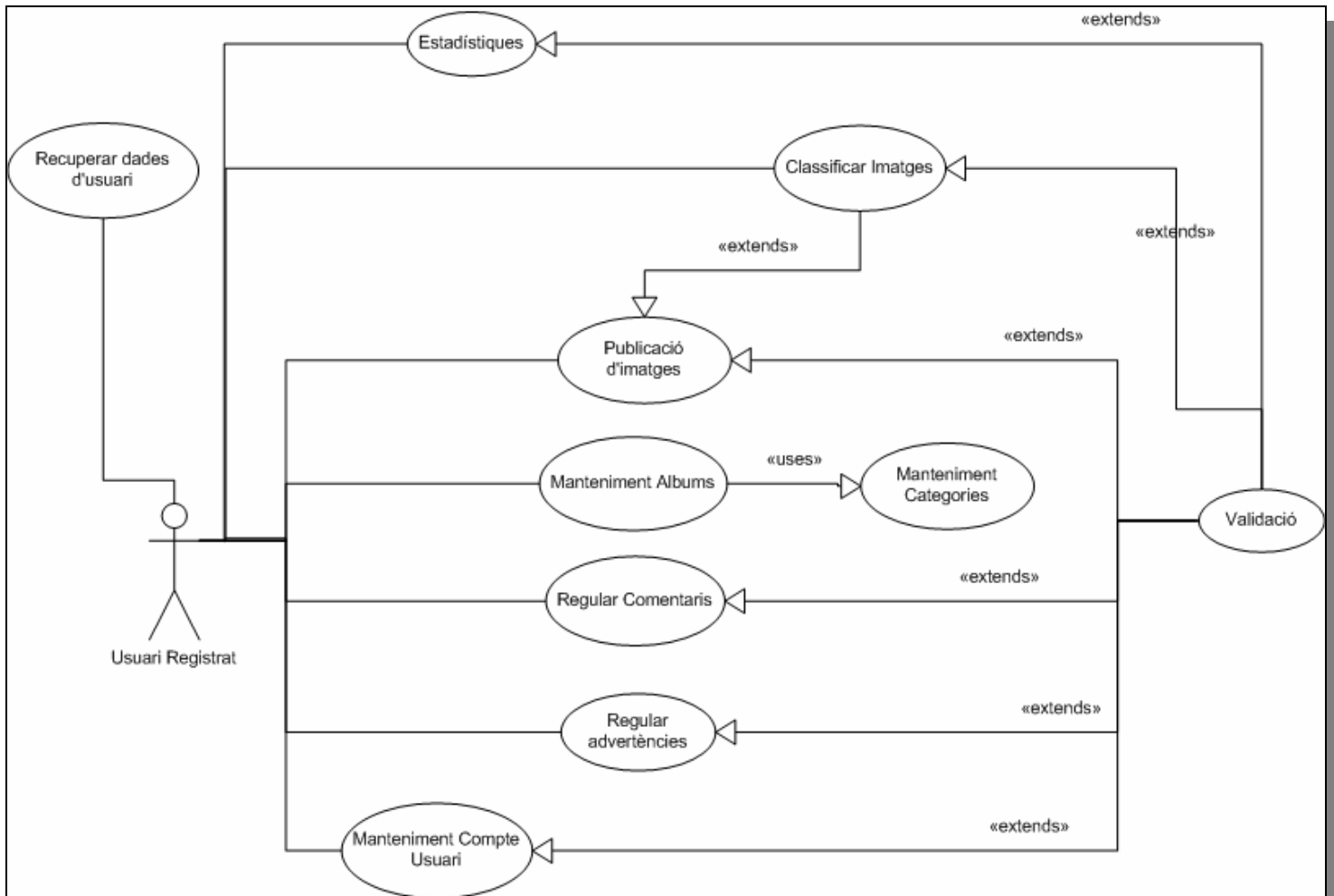
Podrem realitzar consultes sobre qualsevol d'aquestes categories aplicant un filtre per data.

L'apartat oferirà la possibilitat de visualitzar les imatges i d'accedir als àlbums i categories a les que pertanyen, així com a la galeria del usuari que les ha publicat.

⁷ Es deshabilitaran els vincles de la capçalera, l'únic que restarà actiu serà el de finalitzar sessió.

Casos d'ús d'usuaris registrats

L'actor usuari registrat, és aquell usuari que s'ha registrat en el sistema i que s'ha validat per tenir accés als casos d'ús reservats aquest actor. L'usuari registrat pot fer ús dels casos d'ús del usuari visitant⁸.



(Figura 2) Diagrama de casos d'ús dels usuaris registrats.

Recuperar dades d'usuari

Un usuari pot perdre o oblidar-se de les dades que li permetin entrar en l'entorn privat. En la pantalla principal trobem un link que ens permetrà entrar en aquest l'apartat. Se'ns demanarà que entrem el nostre nom d'usuari, seguidament se'ns farà la pregunta que varem entrar en el nostre perfil i introduïrem la resposta. Si la resposta és correcta se'ns permetrà entrar un altre password, que a partir d'aquest moment serà el que haurem d'utilitzar per validar-nos en el sistema.

⁸ Entre l'usuari registrat i l'usuari visitant hi ha una relació d'herència. No es reflexa en el diagrama de casos d'ús, ja que hem decidit que aquestos es presentin per separat en el document.

Manteniment del compte de usuari

La secció compte d'usuari ens mostra la informació del nostre compte i ens permet modificar-la. Ens permet modificar les dades personals del usuari i les del compte.

L'única dada que no ens permeten modificar és el nom d'usuari.

L'usuari pot vincular una imatge al seu perfil. Per defecte apareixerà una imatge genèrica per tots els usuaris. Al costat un el link [Edita la teva imatge](#), al fer-hi clic ens portarà a la pantalla de publicació d'imatge a partir d'on podrem guardar la imatge. El procés es descriu en el cas d'ús publicació d'imatges. Al tornar al perfil d'usuari se'ns carregarà una miniatura de la fotografia de 150x120 píxels. Comptem amb un link per [eliminar](#) la imatge i restablir la imatge per defecte.

Els processos per modificar el password i la pregunta de seguretat són més complexos que una simple edició del camp. Accedim a la pantalla d'edició de dades de seguretat a través de la pantalla de compte d'usuari. Per modificar el mot de pas, hem de teclejar el mot de pas anterior, i el nou mot de pas dos cops, el segon per confirmar-lo.

Per la seva part, per canviar la pregunta de seguretat hem d'anar seguint els passos que ens marca un assistent. En primer lloc, visualitzem l'actual pregunta. En segon lloc introduïm el password d'usuari i finalment introduïm una nova pregunta i resposta.

Manteniment d'àlbums

L'usuari registrat podrà donar d'alta àlbums i posteriorment modificar les seves dades. En la barra d'opcions del entorn d'usuaris, trobem l'opció galeria. Dins d'aquesta pantalla se'ns presenta un llistat de tots els àlbums donats d'alta. Les dades que apareixen són el títol, la descripció, la data de creació i si és públic o no. Cadascun dels registres estarà acompanyat de dos botóns, un per seleccionar – il·lustrat amb una fletxa verda- i un altre per eliminar, amb una creu vermella.

Al cap damunt de la secció tindrem una barra amb dos links:

- Mode de treball: Marca el mode de treball de la pantalla. A la dreta tenim la llegenda del mode actual. Els modes de treball poden ser Consulta i Edició. En el mode consulta, al fer clic en el botó de selecció, es carregarà la pantalla del àlbum, descrita a cas d'ús manteniment de categories. Al fer clic en el link Mode de treball s'obrirà una secció per editar dades, sobre el llistat d'àlbums donats d'alta i es passarà al mode d'edició. En aquest mode de treball, al seleccionar un registre, les dades es carreguen en la secció d'edició i es podran modificar. Els canvis es guardaran o es rebutgen amb els botóns guardar/cancel·lar. Una vegada carregat un registre no se'n pot carregar cap més fins que es premi un dels botóns.

- Nou: En mode edició, permet crear un registre nou. Buida els camps i al guardar el registre apareixerà en el llistat de la part inferior de la pantalla.

Manteniment de categories

El manteniment de categories equivaldria a la pàgina individual del àlbum. En ella crearem i administrarem les categories en que és distribuïran les imatges. Una categoria és simplement una subdivisió classificatòria de les fotos.

Al cap damunt de la pàgina apareix el títol i la descripció del àlbum. Aquestes dades no són editables. A continuació trobem una barra de divisió amb les opcions:

- Mode de treball: Estableix el mode de treball de la pantalla. Consulta o Edició. En el mode consulta seleccionarem la categoria del llistat i se'ns carregaran les imatges existents. En el mode edició se'ns carregarà el descriptor de la categoria en la casella de text, que és troba al costat de les opcions nou i guardar, i ens permetrà modificar-ne el valor.
- Guardar: Guarda el descriptor de la categoria, una vegada aquest s'ha editat.
- Nou: Buida la caixa de text per que es pugui crear una nova categoria.
- Classificar⁵: Ens permet assignar classificadors (tags) a totes les imatges de la categoria. Els classificadors introduïts prèviament es esborraran i només és guardant els entrats des d'aquest apartat.

El cos de la pàgina es compon d'un llistat de totes les categories a la part esquerra, on cadascun dels registres està acompanyat d'un botó per seleccionar i un altre per eliminar. A la part dreta es mostraran les imatges assignades a la categoria carregada.

Fent clic sobre la imatge desitjada, aquesta es mostrarà en la presentació visualització completa. En aquesta presentació se'ns permet editar la descripció i la data de publicació. A la part superior de la imatge se'ns presenten una sèrie d'ícones que permetran realitzar les següents funcions:

- Guardar: Guarda els canvis en la data de publicació i la descripció.
- Eliminar: Elimina la imatge. Es demana confirmació.
- Establir la imatge com a coberta del àlbum.
- Establir la imatge com a coberta de la categoria.
- Classificar: S'introdueixen els classificadors particulars per aquesta imatge.

Validació en el sistema

En la pàgina principal de l'aplicació trobem una secció per la validació de l'usuari. Entrem el nom, i el mot de pas i validem les dades, si són correctes entrem a l'entorn reservat a usuaris registrats. Segons si és un usuari client del servei o un usuari administrador, ens donarà pas a un entorn personalitzat segons el perfil.

En el cas que la validació no sigui correcta es mostrarà un missatge d'error i ens donarà la possibilitat de tornar a entrar les dades.

En el panel de la validació hi ha la opció Recordar l'usuari al fer clic quedarà marcada i la pròxima vegada que entri a la pàgina principal, enlloc de la secció de validació trobarem un panell amb el nom d'usuari i el link sortir per tancar la sessió.

Publicació d'imatges

La funcionalitat de publicació pot ser carregada des de dues ubicacions. La primera des del menú principals del entorn d'usuari i des del manteniment del compte d'usuari, fent clic en el link edita imatge.

En el cas de ser cridada des de la pantalla del compte d'usuari es carregarà una versió simplificada de la pàgina. En ella es disposarà el control de cerca de la imatge en el sistema de fitxer, una casella de text on es guarda l'encaminament del arxiu físic i un botó amb la llegenda Examinar. La pantalla es completa amb dos botons Afegir i tornar. En aquesta modalitat l'operativa és la següent:

1. Fem clic en el botó examinar, s'obre un quadre de diàleg que ens permet navegar pel sistema de fitxers del nostre PC per tal de seleccionar la imatge⁹ desitjada. Una vegada seleccionada apareixerà l'encaminament d'on es ubicada en el nostre sistema local.
2. Premem el botó Afegir i la imatge es guardarà en el nostre perfil. Al completar-se correctament el procés, s'oculta el botó afegir. Si per alguna circumstància imprevista no es pogués guardar se'ns notificaria amb un missatge.
3. Fem clic en el botó tornar i se'ns torna al carregar el perfil, amb la imatge seleccionada.

El segon mode de treball, utilitzat per pujar fotografies als àlbums del usuari i que és crida per mitja de la barra d'opcions del entorn, presentarà una interfície més complexa. Comptem amb el control de cerca d'imatges, descrit anteriorment, amb el comandament Afegir tot just després del comandament Examinar. Una casella d'opció amb el text Pujar totes les imatges del directori. Dos camps selectors per l'àlbum i la categoria on s'han de pujar les imatges, un llistat de les imatges a pujar i el botó acceptar per iniciar el procés. L'operativa de l'opció en aquest mode de treball és la següent:

1. Fem clic en el botó examinar i amb el quadre de diàleg seleccionem la imatge desitjada.

⁹ El sistema només accepta imatges dels formats, jpg, gif, png i bmp

2. Premem el botó afegir i es crea una entrada en una llista – en la part inferior de la pantalla - amb una casella de selecció a l'esquerra i la ruta física de la imatge dins del nostre PC.
3. En el cas que seleccionem la opció Pujar totes les imatges del directori i premem afegir, es carregaran totes les imatges del directori, dels formats admésos pel sistema. Al ser un sistema que permet la selecció múltiple, podrem repetir totes les vegades que vulguem aquests passos.
4. Mitjançant el selectors situats en la part inferior del selector de fitxers escollirem el àlbum i la categoria al que pujar les imatges. El selector es compon d'una casella de text i un comandament a la dreta de la casella, que al fer-hi clic mostrarà una pantalla emergent amb una llista amb tots els àlbums/categories disponibles. Al seleccionar el registre i prémer acceptar a la pantalla emergent col·locarà el descriptor en la casella de text corresponent.
5. Al clicar Acceptar s'iniciarà el procés de publicació de les imatges. Apareixerà una control de progrés que ens anirà informant, de les imatges publicades fins al moment.

Al capdamunt de la llista comptem amb les opcions seleccionar/desseleccionar tot, per facilitar la selecció de les referències existents en la llista. Per altra banda l'opció Eliminar eliminarà les referències seleccionades de la llista.

Si el nombre de registres en les llistes d'albarans i categories és molt elevada, fent clic en un ítem de la llista i teclejant l'inici del nom que cerquem, ens col·locarà damunt del que major coincidència tingui.

Pot ser que al intentar realitzar una selecció d'àlbum/categoria, no disposem del registre desitjat. Al visualitzar la pantalla emergent per realitzar la selecció comptarem amb una casella de text - sota la llista - on entrar el títol d'un nou registre i donar-lo d'alta per mitja del vincle afegir que hi ha a la dreta de la casella.

Regular comentaris

L'usuari haurà de revisar els comentaris fets per altres usuaris – registrats o no – en les imatges que exhibeix. En l'entorn privat d'usuaris, trobem la secció administració, en ella apareixen un resum dels comentaris nous no revisats des de la última visita. Des d'aquesta secció podrem gestionar els comentaris no revisats. Es mostraran ordenats per data, de més recent a menys. Encapçalant cadascun dels registres es mostrarà la imatge a la que pertanyen. Juntament a cada registre comptarem amb un botó per donar-lo per vist i un botó per eliminar-lo.

Regular advertències

En la secció administració trobem l'apartat de revisió d'avísos. En ell podrem visualitzar la col·lecció d'imatges que són objecte d'avís per part dels altres usuaris o visitants del sistema. En aquest cas se'ns presentaran una col·lecció d'imatges amb el tipus d'advertència realitzada, la data i el comentari per part del usuari emissor, en el cas que aquest l'hagi entrat. Els avisos es presentaran inicialment en una llista. Visualitzarem la imatge i el text entrat per l'usuari emissor. Hi haurà una casella on l'usuari propietari podrà entrar una justificació si així ho desitja i donar per vist l'avís. En el llistat d'avísos només apareixeran els no revisats per l'usuari.

Classificació d'imatges

Una vegada publicades les imatges podrem assignar-les-hi classificadors o tags. Aquests classificadors són paraules que permeten descriure el contingut de la imatge, per posteriorment fer-ne més senzilla la cerca. La classificació pot fer-se de dues formés:

- Globalment per categoria. En el cas d'us manteniment de categories és menciona l'existència de l'opció classificar en la barra d'eines de la pàgina. Al fer-hi clic apareixerà un panel amb una caixa de text on podrem escriure els diferents classificadors que es volen assignar a totes les imatges d'aquesta categoria o àlbum. Els classificadors s'escriuran separats per comés. Se'ns oferirà l'opció de cerca i selecció dels tags anteriorment entrats per altres imatges, a través de un botó amb tres punts al costat de la caixa de text. Una vegada entrats els tags premem el botó acceptar i es guardaran les dades. En aquest cas, si hi havia tags assignats anteriorment a les imatges de la categoria es sobreescriran amb els que entrem des d'aquesta opció.
- Individualment per imatge: Accedint individualment a la imatge, en el mode de presentació visió complerta, trobem el link [Classificar](#) que ens obrirà el mateix panel que mostrem en l'opció classificar per àlbum i realitzem el mateix procés. La diferencia està en que el classificador només s'aplicarà a la imatge actual.

Consulta d'estadístiques

En l'apartat d'estadístiques se'ns mostraran dades de analítiques sobre la valoració de les imatges publicades. En un segon apartat és mostra l'activitat en quan a comentaris, puntuacions i advertències rebudes pels àlbums.

En la pantalla on visualitzarem un ranking de les imatges millor valorades. Ordenades per la valoració mitja de puntuació i nombre de vots. El ranking es pot

ordenar pels diversos conceptes que mostra la llista. Per fer-ho hem de fer clic a la capçalera de la columna.

En l'apartat d'estadístiques d'àlbums és presentarà un llistat dels àlbums públics. Al fer clic en la coberta, la imatge que apareix en el marge esquerra es carregaran les estadístiques. Es totalitzaran els comentaris, advertències i vots. Sota les totalitzacions es mostrarà el detall de cadascun dels conceptes.

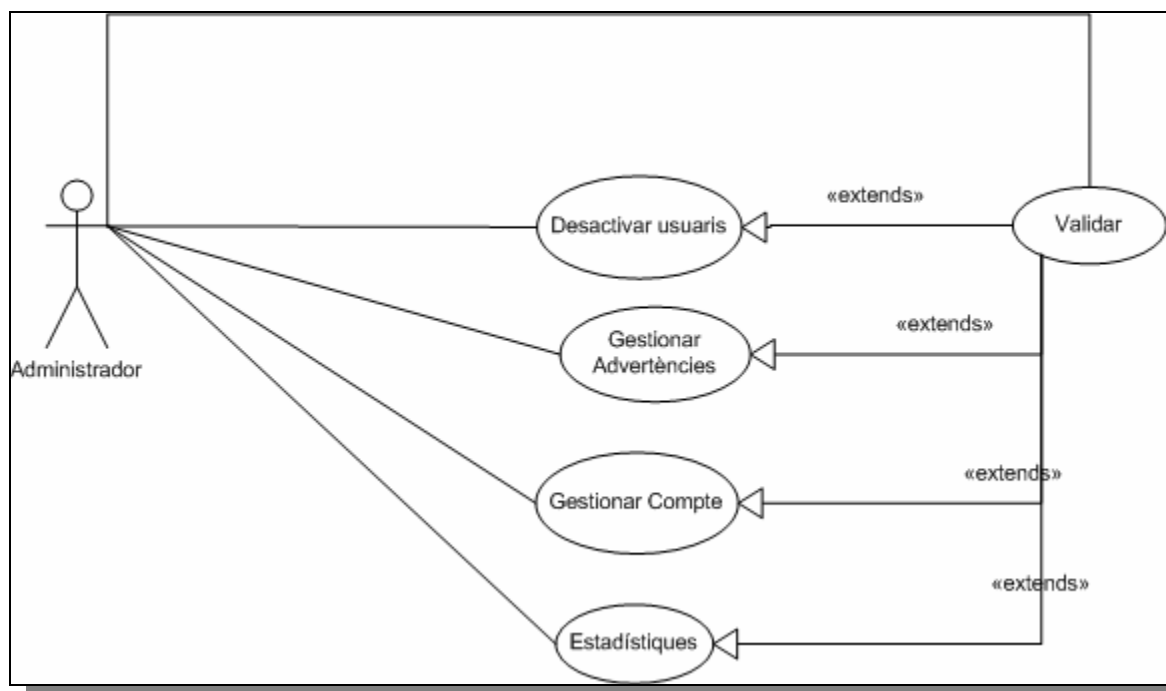
Tant les estadístiques com el ranking es podran filtrar mitjançant un filtre per dates que hi ha a la capçalera de la pàgina. Al cercar és recarregà el ranking i les Estadístiques.

Herència entre els visitants i usuaris registrat

Tot i no ser reflexat per els gràfics de la secció de casos d'ús, existeix una herència del actor Visitant per part del actor Usuaris registrat. Un usuari registrat podrà realitzar totes les accions que realitza un visitant. En l'entorn privat, reservat als usuaris registrats, s'habilitarà un vincle que al clicar-lo ens permetrà retornar a l'entorn d'ús públic sense tancar la sessió. Podrem realitzar totes les accions estan validats.

Casos d'ús d'usuaris administradors

L'actor administrador és un usuari registrat en el sistema amb el perfil d'administrador. Te un entorn privat diferent al d'usuari registrat. Són donats d'alta des de fora del sistema, per l'equip administrador.



(figura 3) Diagrama de casos d'ús dels usuaris administradors.

Validació en el sistema

Veure apartat anterior. Casos d'ús d'usuaris registrats.

En el cas dels administradors, al accedir a l'entorn privat es carregarà la pàgina del compte d'usuari

Gestionar les dades del compte

La pàgina de gestió del compte d'usuari serà similar a la del compte d'usuari registrat descrita en l'apartat anterior, pertanyent als usuaris registrats o usuaris clients. Consta d'una fitxa on l'administrador podrà editar les dades del seu compte. Essencialment són les mateixes de les que disposa l'usuari registrat. En el cas de l'administrador però, no comptem amb una descripció personal, ni amb la possibilitat d'assignar una fotografia a la seva fitxa, ja que aquestes no tindrien cap utilitat en especial. Per conèixer la seva operativa veure el cas d'ús manteniment del compte d'usuari, de l'apartat anterior.

Un usuari administrador no pot donar-se d'alta al sistema ell mateix. Ha de ser creat per l'equip d'administració des de fora de l'entorn web. L'administrador té l'obligació d'entrar les seves dades personals i com l'usuari client, s'obligarà a entrar les seves dades, abans de començar a treballar amb el sistema.

En quan a modificar les dades de seguretat del seu compte, se li permet modificar el mot de pas i la pregunta de seguretat. L'operativa d'aquest apartat és exactament la mateixa que per els usuaris registrats.

Gestionar advertències

L'administrador tindrà accés a totes les advertències generades pels usuaris que visitin el nostre portal i que no s'hagin processat anteriorment per un administrador. Aquestes es presentaran en forma de llista ordenada per data d'emissió de l'avís de més a menys recent, tot i que serà ordenable per diferents criteris.

Cadascun dels registres presenta el tipus d'advertència, la data, la descripció i si s'ha revisat per l'usuari. Encapçalant cada fila hi trobem un selector que ens permet seleccionar l'advertència i passar a una pantalla on visualitzarem la imatge que és objecte de judici, la descripció, la resposta de l'usuari, si aquest l'ha entrat i on disposarem dels comandaments Eliminar i Vist, utilitzats per que l'administrador atengui l'avís. Les seves funcionalitats són les següents:

- Eliminar dona per bona l'advertència i elimina la fotografia. Si hi ha més advertències sobre aquesta fotografia se'ls imputarà la mateixa resposta per part de l'administrador, retornant el nombre d'avisos atesos.

- Vist la desestima i continua endavant. Si hi ha més advertències sobre la imatge revisada ens preguntarem si volem aplicar-hi la mateixa resposta. Si diem que sí, es donaran per ateses.

Una vegada processada la imatge, aquesta continua emmagatzemada a la base de dades amb la referència al administrador que la validat. Es tornarà a la pantalla inicial, on es recarregarà el llistat d'avisos actualitzant-lo amb les modificacions realitzades per qualsevol dels administradors que hi hagi treballant en aquest moment.

En previsió de que el sistema hagi de gestionar un gran nombre d'avisos, en el llistat principal s'inclourà un filtre per varis criteris, per interval de dates, per avisos revisats per l'usuari o no, i si aquests avisos han tingut resposta per part de l'usuari propietari de la imatge.

Es pot donar el cas que hi hagi varis administradors processant avisos, en aquest cas, quan un administrador esculli una advertència per processar-la tots els avisos que estiguin disponibles per aquesta imatge quedaran assignats al administrador, assegurant així que no hi hagi conflictes a l'hora d'atendre avisos entre varis administradors a la vegada.

Desactivar Usuaris

La pantalla de desactivació d'usuari tindrà la funcionalitat d'apartar del sistema els usuaris que en facin un ús incorrecte. Aquesta decisió es prendrà en funció del nombre d'avisos que s'hagin rebut sobre l'usuari.

Al entrar a la pàgina, visualitzarem un conjunt de filtres que ens permetran obtenir un llistat del nombre d'usuaris i dels seus avisos rebuts. Aquests filtres són els següents:

- Per nom d'usuari.
- Per interval de dates de creació del usuari.
- Per interval de dates de l'última de validació del usuari.

Al genera el llistat, es presentarà el conjunt de registres amb les següents dades:

- Fotografia del usuari.
- Nom usuari.
- Mail de l'usuari.
- Data de creació del compte d'usuari.
- Data de l'últim login en el sistema.

Precedint cada registre tindrem un botó selector que ens carregarà una nova pàgina detallant les dades bàsiques del usuari, la seva fotografia i el nombre d'avisos processats i el d'imatges que s'han eliminat de la seva galeria. Comptarem amb dos

comandaments que ens permetran cancel·lar l'operació i tornar a la pantalla principal per visualitzar el llistat o el botó desactivar, que desactivarà el compte i ocultarà tot el contingut publicat per l'usuari. Aquesta acció declararà com ha privats tots la seva galeria d'imatges. El llistat d'usuaris, també comptarà amb el botó desactivar al costat del selector. Abans de dur a terme l'operació demanarà confirmació.

Al marge superior esquerra de la pantalla, trobem un link que marca sobre quins usuaris és realitza la cerca, per defecte apareix el text "Usuaris actius", al fer clic passa al mode "Usuaris Inactius" i serà sobre aquests tipus d'usuaris sobre els que es realitzaran les cerques.

Consulta d'estadístiques

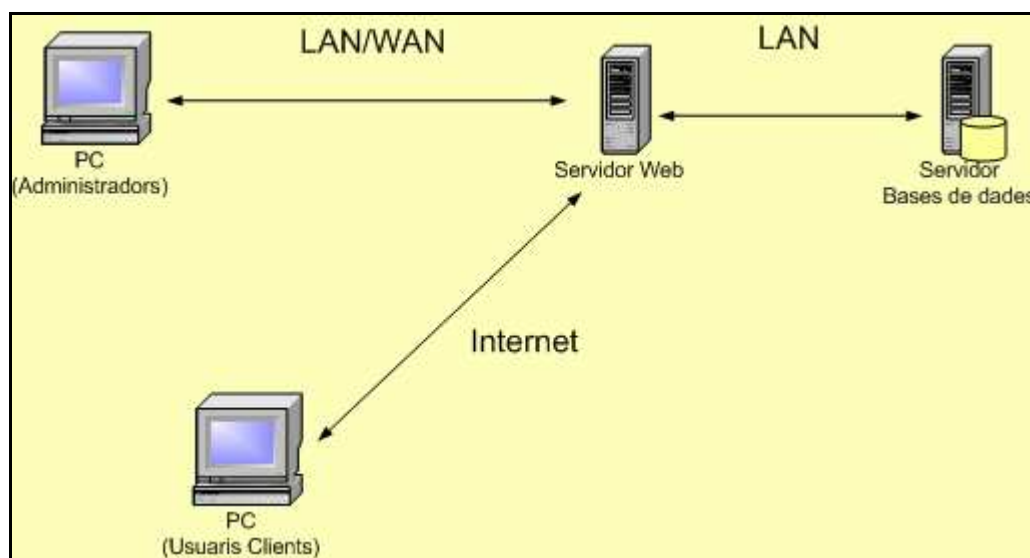
L'apartat d'estadístiques pels usuaris administradors estarà orientat a verificar d'utilització de l'entorn per part dels usuaris clients, oferint dades representatives sobre la creació de comptes d'usuari, publicacions d'imatges, així com creacions d'àlbums, publicació de comentaris i ús del sistema de puntuació. La informació es podrà filtrar entre dues dates i el resultat es presentarà en un resum totalitzat en pantalla o en forma de gràfica. En la mateixa pantalla es presentarà el detall de l'activitat diària per a cadascun dels conceptes anteriorment mencionats.

Disseny

Arquitectura

Arquitectura física de l'aplicació

S'ha escollit un àmbit físic per la implantació del projecte el més reduït i simplificat possible. El següent diagrama presenta la distribució dels diferents elements del sistema:



(figura 4) El diagrama que és presenta en el gràfic correspondria a una implantació ideal del sistema.

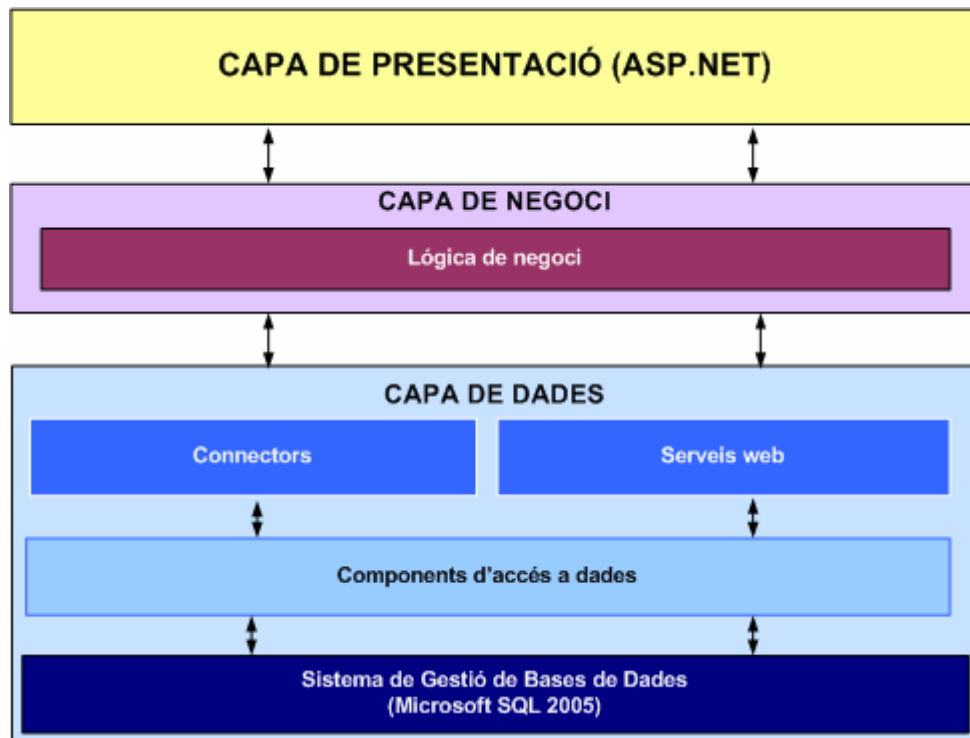
Comptarem amb dos servidors. Un servidor web per hostatjar i administrar tot l'entorn web. S'utilitzarà el servidor IIS de Microsoft. Per altra banda, comptem amb un servidor on s'implantarà l'entorn Microsoft SQL Server i on emplaçarem les bases de dades de la nostra aplicació. Emplaçar per separat els elements de programa i dades millorarà el rendiment i el manteniment del sistema.

Una altra característica que ens mostra el sistema és la possibilitat que s'atorga als usuaris administració d'accedir a través de la LAN corporativa per accedir al sistema. Aquesta opció permetrà un accés més ràpid i un treball més eficient.

Els administradors també podran accedir al sistema via web.

Arquitectura lògica del sistema

L'aplicació estarà estructurada en una arquitectura per capes (n-tier). Les dades, el negoci i les interfícies de presentació seran definides per un conjunt de classes relacionades entre si, però clarament separades en quant a funcionalitat. Aquest mètode de treball ens proporcionarà una encapsulació destinada a facilitar la implementació i el manteniment.



(figura 5) Esquema descriptor de les diferents capes de l'aplicació i la seva interacció

Com s'observa en el diagrama d'estructura de l'aplicació, ens trobem davant de tres blocs:

Presentació: Conjunt d'interfícies que permetran la interacció de l'usuari amb l'aplicació. Les anomenades classes frontera seran pàgines ASP que es carregaran en el navegador.

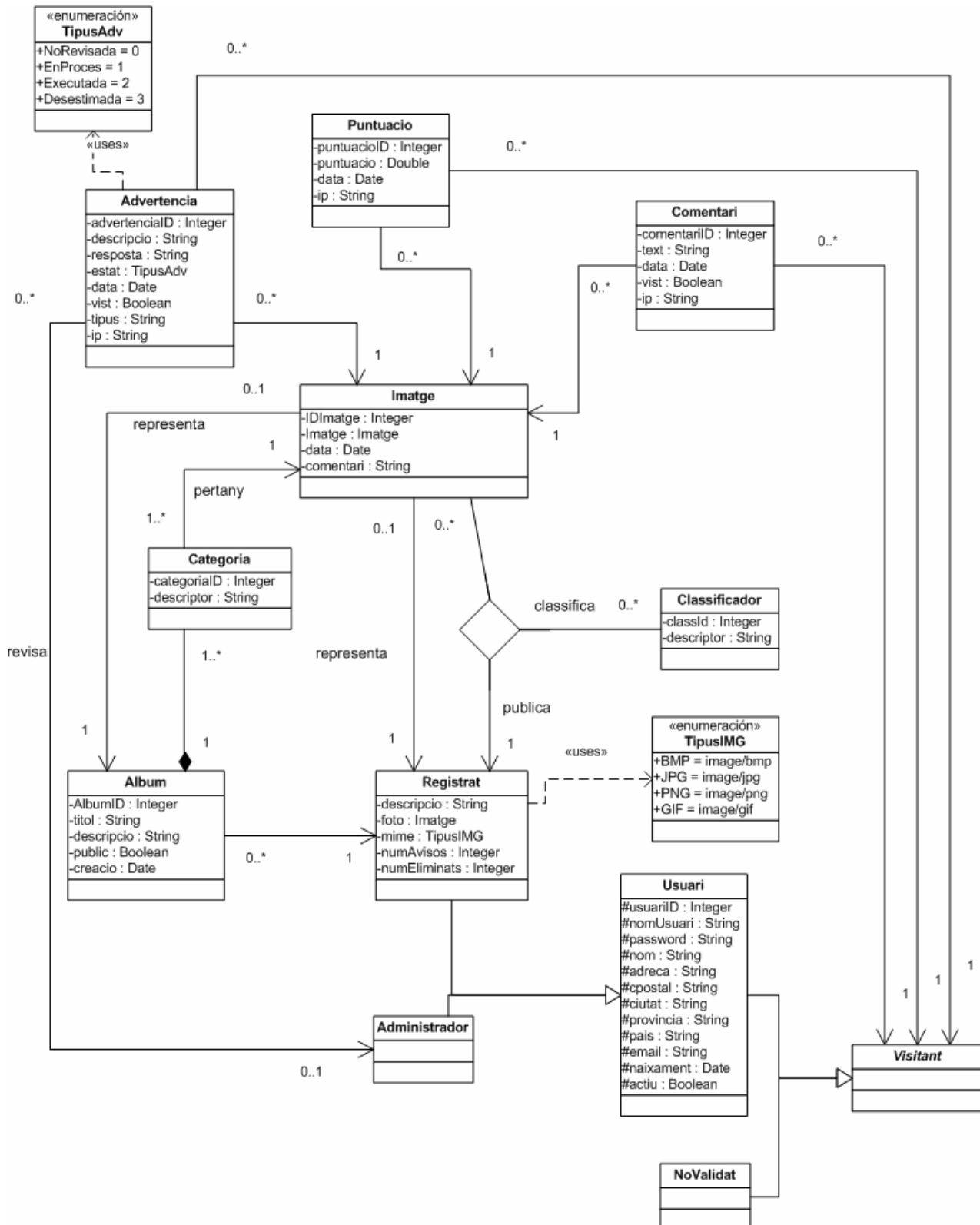
Capa de negoci: Conjunt de processos encarregats de retornar els resultats de les peticions que l'usuari sol·licita a l'aplicació. La capa de negoci rebrà les dades de la capa de dades i realitzarà les transformacions necessàries per carregar-les a la capa de presentació.

Capa de dades: És on resideixen les classes connectores. Encarregats de realitzar les diferents operacions amb la base de dades, (guardar, actualitzar, eliminar, recuperar). El conjunt d'instruccions per manipular la informació es generarà a partir d'un DataLayer, en format xsd, i dels adaptadors que aquest contingui, és el que coneixem com components d'accés a dades. En el gràfic també s'inclou els serveis web com a part de la capa de dades, aquests es limiten a recuperar les dades de la base de dades i a carregar-los en un control de la capa de presentació.

Finalment, tot i no ser una unitat funcional de l'aplicació tenim la base de dades. De cara a aconseguir un emmagatzematge de dades eficient, treballarem amb una base de dades relacional normalitzada, dissenyada mitjançant les tècniques d'entitat - relació. Aquesta serà implementada mitjançant Microsoft SQL Server.

Classes entitat

Diagrama



(figura 6) Diagrama de classes entitat. Els atributs de les classes és presenten com a privats. Seran llegits per mètodes Get i Set, no representats per evitar una redundància d'informació innecessària.

Les classes que s'enumeren i es descriuen en aquest apartat són els anomenats objectes de referència. Cadascun d'ells representa les entitats que formen part i intervenen en el sistema. Funcionalment s'encarreguen de contenir els valors dels diferents atributs d'una entitat per a que puguin ser escrits o llegits.

Descripció de les classes entitat

Visitant, Usuari, NoValidat, Registrat, Administrador

Són el conjunt de classes que defineixen la jerarquia d'usuaris que faran ús de l'aplicació. De la classe Usuari hereten les classes NoValidat i Registrat. Usuari i NoValidat són classes abstractes.

- Usuari és una referència global a tots aquells usuaris que utilitzin el sistema.
- NoValidat representa a tots els usuaris no registrats o registrats i no validats en el sistema que facin ús del mateix. Al no estar identificats no l'imputem cap mena d'atribut.

Registrat i Administrador hereten de la classe Usuari. Defineixen els dos tipus d'usuari que operen amb el sistema.

- Usuari és l'entitat que representa a l'usuari del sistema, indistintament del seu perfil. Presenta dos tipus d'atributs diferenciats, els relatius al compte d'usuari i a informació de caràcter personal. Aquest són comuns per les dues classes descendents.
- Registrat és la classe que representa al usuari client del sistema. Compta amb dos atributs específics, a més dels de caràcter general, descripció, foto i mime¹⁰.
- Administrador defineix a l'usuari administrador del sistema. És una classe que permet identificar el tipus d'usuari però que no compta en que no s'especialitza cap atribut.

Es defineixen relacions associatives entre la classe visitant i les classes Advertencia, Puntuacio i Comentari. La relació d'aquestes classes s'imputa a la classe Visitant, ja que tant els usuaris registrats i no validats poden crear objectes de les tres classes. La cardinalitat de les relacions serà Visitant [1,0..*] (Advertencia/Puntuacio/Comentari).

La classe Registrat manté dos associacions. Amb la classe Album amb cardinalitat Registrat [1,0..*] Album. L'altra relació és una associació n-aria entre tres classes (Registrat / Classificador / Imatge), amb la cardinalitat Registrat [1,0..*,0..*]

¹⁰ MIME definirà el tipus d'arxiu que s'emmagatzema en la base de dades. En el cas d'una imatge seria image/jpeg, on image indica el tipus d'arxiu i jpeg el format.

(Imatge/Classificador). Cada usuari pot publicar varies imatges i cada imatge pot estar etiquetada per varis classificadors i cada usuari pot crear varis classificadors¹¹.

Finalment la classe Administrador manté una relació d'associació amb la classe Advertencies. La cardinalitat és Administrador [0..1,0..*] Advertencies. Es vincula l'advertència amb l'administrador que la revisa.

Àlbum, Categoria

Les classes àlbum i categoria són les utilitzades per representar els classificadors de les imatges en el sistema. Les imatges poden ser publicades en àlbums, que a la vegada es divideixen en categories.

Les dos classes estan unides per una relació de composició. Un àlbum està compost per diverses categories. Una categoria ha de pertànyer obligatòriament a un àlbum, ja que no té sentit la seva existència per si sola.

En quant a associacions amb altres classes, per part del àlbum i la categoria trobem vincles amb la classe imatge amb cardinalitat (Album/Categoria) [1,0..1] Imatge. Un àlbum i una categoria pot estar vinculat a una imatge que serà utilitzada com a icona per representar l'àlbum o la categoria.

Imatge

La classe Imatge carrega les dades que pertanyen a la imatges emmagatzemades pel sistema. Destaca l'atribut Imatge, un camp binari encarregat que conte la informació que conte la imatge.

La classe Imatge és l'entitat central de l'aplicació. És la classe que manté un major nombre de relacions associatives. Està associada amb les classes Advertencia, Classificador, Puntuació, Comentari amb cardinalitat [1,0..*], on l'extrem amb cardinalitat 1 pertany a la classe imatge.

Finalment, és manté també una associació amb la classe Categoria, amb cardinalitat [1,1..*], on 1 és l'extrem que pertany a la classe imatge, que indica la pertinència de la imatge a una o varies categories.

Classificador

La classe Classificador carrega els descriptors/tags que s'assignen a una imatge. És una classe simple que compta amb un identificador numèric i el text del descriptor. Les associacions en les que intervé ja s'han descrit.

Comentari

La classe Comentari representa als comentaris fets a una imatge. Vinculada amb les classes visitant i imatge, les associacions en les que intervé ja s'han descrit

¹¹ Cada usuari crea els seus propis classificadors.

amb anterioritat. Compta amb un atribut identificador i amb l'atribut – de tipus booleà – vist que verifica si ja ha estat revisat per l'usuari.

Puntuació

Classe que representa els objectes puntuació. Vinculada per relacions associatives amb les classes Visitant i Imatge. Conté el atribut Puntuacio, de tipus doble, la data en que es realitza la puntuació i un identificador numèric.

Advertència

Classe que representa les advertències en el diagrama estàtic. Destaquem l'atribut booleà vist, que indicarà si l'atribut s'ha revisat per l'usuari, i l'atribut tipus, que marca indica el tipus d'advertència de la que es tracta (Veure numerador adjunt a la classe en el diagrama). Les relacions associatives que l'afecten ja s'han revisat amb anterioritat.

Diagrama de classes connectores

Les classes connectores són les classes encarregades d'enllaçar la base de dades amb l'aplicació. Es tracta d'una encapsulació de les funcionalitat de selecció, inserció, actualització i eliminació de les dades, per proporcionar un conjunt de funcionalitats al desenvolupador per a que realitzi aquestes tasques des de l'aplicació.

Descripció de les classes connectores

La idea es definir un Data Layer (arxiu .XSD) a partir del generador que ens ofereix Visual Studio, on es generaran els table adapters que contindran les instruccions SQL necessàries per gestionar la base de dades. A continuació es presenta la descripció del conjunt de classes que encapsulen aquests table adapters i que formés la capa de dades:

Descripció de elements comuns en les classes connectores

Conceptes previs

Cadascuna de les classes connectores te un atribut del tipus classe entitat agregat a la classe connectora. Per exemple, en la classe cnPuntuacio, l'atribut serà un objecte de la classe Puntuació. Entre els altres atributs imprescindibles tindrà els TableAdapters generats en el DataLayer. Tots aquets atributs seran privats.

Les classes connectores no tenen perquè estar relacionades entre elles. Tot i comptades excepcions (veure figura 7) es tracten d'unitats independents entre si, dedicades a gestionar les operacions dels objectes de dades sobre els que treballen.

MÈTODES GENERALS

Constructors: **New()**, **New({Entitat})**

Cada classe connectora comptarà amb dos constructors, el primer procés és el que compten totes les classes per defecte. El segon constructor rep un paràmetre del tipus corresponent a la classe entitat associada a la classe connectora

Mètode carregador

En totes les classes comptem amb dos mètodes carregadors, encarregats de carregar una objecte de la classe entitat vinculada a la classe connectora (com s'explica en l'apartat anterior). El nom dels mètodes carregadors s'obté a partir de concatenar Load + El nom de la classe *entitat*. Per exemple la classe *cnCategoria* tindrà el constructor *LoadCategoria*. Les diferències entre els dos carregadors són les següents:

- *LoadEntitat({Entitat})*: Es passa la classe entitat vinculada al connector com a paràmetre i que s'utilitza per dur a terme totes les operacions de la classe connectora.
- *LoadEntitat({clau})*: Es passa el valor de la clau amb que s'ha emmagatzemat la entitat en la base dades. En el cas d'una imatge serà el valor del camp *imatgeID*. Amb el paràmetre, el mètode carregador recupera els valors de la classe entitat de la base de dades i els guarda internament en l'atribut privat de la classe connectora.

En els dos casos es retorna la classe entitat.

Mètode Guardar()

Present en totes les classes amb la mateixa forma, sense paràmetres i retorna un valor integer que indica el nombre de registres que s'han guardat. Els valors que es guarden són els anteriorment emmagatzemats al connector pel mètode carregador. Internament és verificarà si el registre a guardar ja existeix. En el cas que així sigui s'actualitzarà amb un update. De no ser així s'inserirà.

Mètodes de llistats/selecció

Els mètodes de llistat realitzen una selecció múltiple de registres i la retornen perquè pugui ser presentada per pantalla¹². El nom dels mètodes carregadors s'obté a partir de concatenar Load + El nom de la classe *entitat*. Comptem amb dos tipus diferents de selector:

¹² El resultat serà assignat a la propietat de l'origen de dades d'un control *DataList* o *DataGrid* perquè mostri el resultat per pantalla.

- *LlistarDadesLlistar()*: On Dades a llistar, seran les dades de la classe secundaria (veure conceptes previs). Retornarà un llistat de dades llestes per mostrar per pantalla, quan siguin associades a la propietat origen de dades d'un control preparat per aquesta funció. Primer haurem d'utilitzar una funció carregadora (*LoadEntitat*) per carregar les dades a partir de les quals es realitzarà la consulta que retornarà el llistat.
- *LlistarDadesLlistar({clauEntitatPrincipal})*: Es passa el valor de la clau de la entitat principal ¹³ amb que s'ha emmagatzemat la entitat en la base dades. (Veure *LoadEntitat({clau})*). Retorna les dades en el mateix format que la funcionalitat anterior.

Pràcticament totes les classes tenen dos mètodes de llistat. El mètode principal de la classes serà de la segona classe. Li passarem la clau de l'entitat que emmagatzema i retornarà el llistat, per exemple en *cnCategoria* tenim *LlistarCategoria*(*IdAlbum:integer*), li passem la clau de l'àlbum del qual volem que se'ns retorni el llistat de categories. En aquesta mateixa classe tenim el mètode *LlistarImatges()*, que retornarà un llistat de les imatges de la categoria.

Mètodes d'eliminació de dades

Els mètodes d'eliminació de dades són presents en totes les classes amb les següents formés:

- *DeleteEntitat()*, *EliminarEntitat()*: on entitat és el un paràmetre de la classe entitat vinculada a la classe connectora. La funció eliminarà de la base de dades el registre que correspongui al atribut privat de la classe entitat i que s'haurà d'haver carregat prèviament cridant el mètode load.
- *DeleteEntitat({clauEntitatPrincipal})*,
EliminarEntitat({clauEntitatPrincipal}): Es passa com a paràmetre la clau de la classe principal de la qual és volen esborrar tots els elements secundaris. Per exemple el mètode *EliminarImatge*(*ImatgeID:integer*) de la classe *cnImatge*, se li passarà la clau primària de la imatge la qual volem eliminar de la base de dades.

Es retorna el nombre de registres esborrats, 1 si el esborrat a tingut èxit i 0 si ha fallat.

Existeix()

Aplicable a les classes cnRegistrat, cnComentari, cnPuntuacio

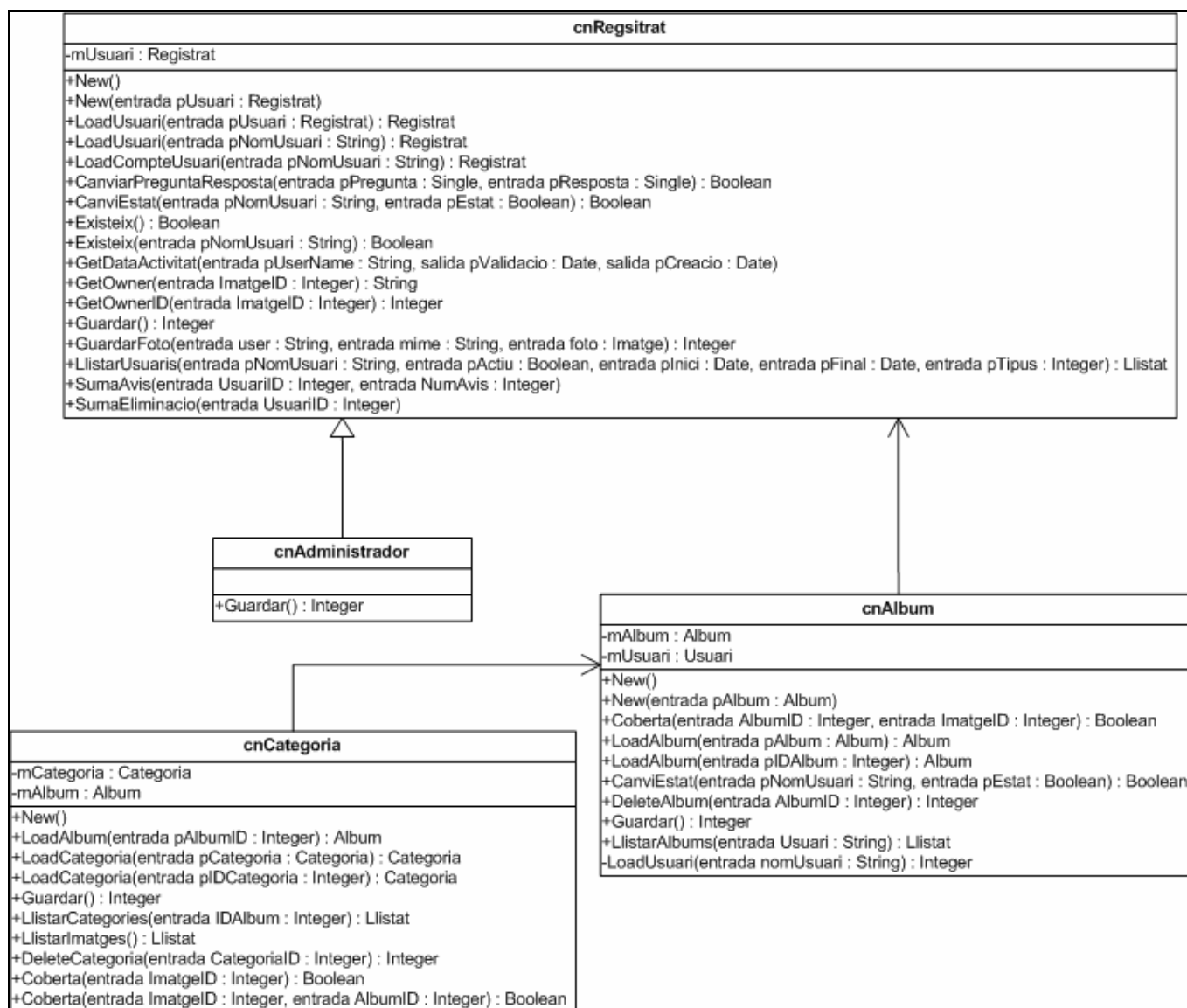
¹³ En la relació *cnRegistrat* - *cnAlbum*. El mètode llistar àlbum passem la clau *idUsuari* per a que retorni els àlbums del usuari.

El mètode existeix te per objectiu confirmar l'existència d'un objecte de les classes, Registrat, Comentari o Puntuacio. Aquestes tres entitats tenen la peculiaritat que compten amb atributs, a més del ID de la classe, que són únics. La funcionalitat Existeix verifica si ja hi ha un objecte amb el pertinent atribut atributs repetit. Si ja existeix retornarà True i de no ser així serà fals.

Prèviament s'ha d'haver carregat una classe entitat en la classe connectora a través del mètode load.

Descripció de elements particulars en les classes connectores

MÈTODES PARTICULARS DE CADA CLASSE



(figura 7) Grup de classes connectores vinculades a cnRegsitrat mitjançant relacions d'associació i d'herència.

cnRegistrat

- LoadCompteUsuari(pNomUsuari:String) Carrega només les dades d'usuari relatives al compte per validar al sistema o que hi tenen relació (nom usuari, password, e-mail, pregunta i resposta per recuperar les dades del compte) i les carregarà en la classe entitat que conte com atribut a la classe connectora.
- Existeix(pNomUsuari:String): Similar al mètode Existeix estàndard. La única diferència és que passem el nom d'usuari com a paràmetre i realitza la consulta en la base de dades directament sense necessitat de carregar l'entitat Usuari a través del mètode LoadUsuari.
- GuardarFoto(user:String, mime:string, foto:Image): Guarda la foto vinculada al perfil d'usuari. Com a paràmetre es passa el nom d'usuari el MIME de la imatge i la imatge en si, retorna un valor integer que indica si s'ha insertat el registre amb èxit.
- GetDataActivitat(NomUsuari:String, Validacio:Date, Creacio:Date): Mètode que carregarà la data de l'última validació i de creació en el sistema del usuari del qual li passem el seu nom d'usuari. Els paràmetres validacio i creacio estan passats per referència.
- GetOwner(ImatgeID:integer), GetOwnerID(ImatgeID:integer): Les dues funcions retornaran l'identificador únic del usuari propietari de la imatge que se li passa com a paràmetre. En el cas de la primera funció retorna el nom d'usuari i la segona el ID numèric.
- CanviEstat(NomUsuari:String, Estat:booleà): Desactiva o activa, segons el paràmetre estat, el compte d'usuari. Al desactivar el compte tots els àlbums del usuari passen a ser privats. Al reactivar el compte els àlbums continuaran com a privats i tindrà que ser l'usuari l'encarregat de convertir-los a públics. Retorna un paràmetre booleà per indicar si s'ha finalitzat correctament el procés.
- SumaAvis(UsuariID:Integer, NumAvis:Integer), SumaEliminacio(UsuariID:Integer): Mètodes encarregats de sumar 1, o en el cas de la suma d'avisos el nombre que se li passi a través del paràmetre NumAvis, als comptadors que s'imputen al usuari per indicar el nombre d'avisos rebuts i el nombre d'imatges publicades que l'administrador li han eliminat.
- CanviarPreguntaResposta(Pregunta:string,Resposta:string): funció encarregada de canviar la pregunta i resposta de seguretat del usuari. Retorna un valor booleà per indicar si les dades s'han canviat correctament.

cnAdministrador

La classe connectora cnAdministrador únicament hereta de la classe cnRegistrat i sobrecarrega el mètode guardar amb la intenció que al guardar l'usuari en la base de dades ho faci amb el perfil d'administrador.

cnAlbum

- Coberta(AlbumID:integer, ImatgeID:integer): Estableix la imatge, representada pel seu identificador únic ImatgeID, com a coberta del àlbum. Retorna un valor booleà per indicar si l'operació s'ha completat correctament.
- CanviEstat(NomUsuari:string, Estat:boolea) operació encarregada de canviar l'estat dels àlbums a l'estat definit per el paràmetre boolea que se li passa. Retorna un valor booleà per indicar si l'operació s'ha completat correctament.
- LoadUsuari(NomUsuari:string) mètode privat encarregat de carregar l'usuari amb el nom que se li passa com a paràmetre. Retorna integer per indicat si s'ha completat correctament l'operació, en cas d'haver un problema inesperat retorna un valor negatiu.

cnCategoria

- Coberta(ImatgeID:integer), Coberta(AlbumID:integer, ImatgeID:integer): Estableix la imatge, representada pel seu identificador únic ImatgeID, com a coberta de la categoria. La segona funció estableix la imatge com a coberta de l'àlbum. Retorna un valor booleà per indicar si l'operació s'ha completat correctament.
- LoadAlbum(AlbumID:integer): mètode privat encarregat de carregar l'objecte àlbum amb l'ID que se li passa com a paràmetre. Retornarà l'objecte que carregui.

cnImatge
-mImatge : Imatge
+New() +New(entrada pImatge : Imatge) +LoadImatge(entrada pImatge : Imatge) : Imatge +LoadImatge(entrada pIDimatge : Integer) : Imatge +LoadVincle(entrada pIDimatge : Integer) : Imatge +Guardar() : Integer +EliminarImatge(entrada ImatgeID : Integer) : Integer +LlistarImatgesCategoria(entrada IDCategoria : Integer) : Llistat +LlistarImatgesCategoria(entrada AlbumID : Integer, entrada CategoriaID : Integer, entrada pageIndex : Integer, entrada pageSize : Integer) : Llistat +GetImatgeRow(entrada ImatgeID : Integer) : Llistat

cnClassificador
-mClassificador : Classificador
+New() +New(entrada pClassificador : Classificador) +LoadClassificador(entrada pClassificador : Classificador) : Classificador +LoadClassificador(entrada pImatgeID : Integer, entrada pUsuariID : Integer) : Col·leccio +LoadClassificador(entrada pImatgeID : Integer, entrada pUsuari : String, entrada pSequencia : String) : Col·leccio +LoadClassificador(entrada pImatgeID : Integer, entrada pUsuari : String) : Col·leccio +Guardar() : Integer +BorrarAssignacioTags(entrada ImatgeID : Integer, entrada Usuari : String) : Integer +BorrarAssignacioTags(entrada ImatgeID : Integer, entrada UsuariID : Integer) : Integer +Existeix(entrada pUsuariID : Integer, entrada pClassificador : String) : Integer +Relacio(entrada ClassID : Integer, entrada ImatgeID : Integer) : Integer

cnComentari
-mComentari : Comentari
+New() +New(entrada pComentari : Comentari) +LoadComentari(entrada pComentari : Comentari) : Comentari +LoadComentari(entrada pIDcomentari : Integer) : Comentari +Eliminar(entrada ComentariID : Integer) : Integer +Existeix(entrada ImatgeID : Integer) : Integer +Guardar() : Integer +LlistarComentarisByUsuari(entrada NomUsuari : String) : Llistat +LlistarComentaris(entrada IDImatge : Integer) : Llistat +Vist(entrada ComentariID : Integer) : Integer

cnPuntuacio
-mPuntuacio : Puntuacio
+New() +LoadPuntuacio(entrada pPuntuacio : Puntuacio) : Puntuacio +Guardar() : Integer +ContarVots(entrada ImatgeID : Integer) : Double +Existeix() : Boolean +Ratio(entrada ImatgeID : Integer) : Double

cnAdvertencia
-mAdvertencia : Advertencia
+New() +Count(entrada ImatgeID : Integer) : Integer +Eliminar(entrada ImatgeID : Integer) : Integer +Guardar() : Integer +LlistarAvisos(entrada AdminID : Integer, entrada Inci : Date, entrada Final : Date, entrada Vist : Boolean, entrada AmbResposta : Boolean) : Llistat +LlistarAvisosUsuari(entrada NomUsuari : String) : Llistat +LoadAdvertencia(entrada pAdvertencia : Advertencia) : Classificador +LoadAdvertencia(entrada pAdvertenciaID : Integer) : Advertencia +Processar(entrada ImatgeID : Integer, entrada AdminID : Integer) : Integer +Validar(entrada ImatgeID : Integer, entrada AdminID : Integer) : Integer +ValidarTot(entrada ImatgeID : Integer, entrada AdminID : Integer) : Integer +Vist(entrada AvisID : Integer, entrada Resposta : String) : Integer

(figura 8) Grup de classes connectores vinculades a cnImatge, sense cap relació associativa entre elles.

cnImatge¹⁴

- LoadVincle(IDImatge:integer) A partir del identificador únic que se li passa com a paràmetre, retornarà el conjunt de dades que conformen el vincle entre la imatge, l'àlbum i la categoria.
- GetImatgeRow(ImatgeID:integer) Mètode de llistat que retorna el registre de la imatge de la qual se li passa l'identificador com a paràmetre.

¹⁴ Aquesta classe compta amb un mètode de llistat paginat. Tots els mètodes de llistat paginats és diferencien perquè compten amb els paràmetres pageIndex i PageSize.

cnClassificador

- Relacio(ClassID:integer, ImatgeID:integer): funció utilitzada per saber si s'ha assignat un classificador a una imatge. Si s'ha assignat retorna l'ID de l'assignació, si no retorna 0.

cnComentari

- LListarComentarisByUsuari(nomUsuari: string): retorna el llistat de tots els comentari pendents de revisar del usuari que li passem com a paràmetre.
- Vist(ComentariID): Dona com a vist el comentari amb l'ID que se li passa com a paràmetre. Retorna 1 si el comentari s'ha actualitzat correctament.

cnPuntuacio

- ContarVots(ImatgeID:integer): Retorna un valor numèric indicant el nombre de vots que ha rebut la imatge, l'ID de la qual se li passa com a paràmetre.
- Ratio(ImatgeID:integer): Retorna un valor numèric decimal que indica la mitja entre la suma total de les votacions dels usuaris dividit pel nombre de vots d'aquestos. La ràtio es correspon a la imatge de la que es passa com a paràmetre el seu ID.

cnAdvertencia

- Count(ImatgeID:integer): Retorna el nombre d'advertències que s'ha rebut d'una determinada imatge.
- LlistaAvisosUsuari(NomUsuari:string) Retorna un llistat dels avisos no revisats per l'usuari que se li passa com a paràmetre.
- Procéssar(ImatgeID:integer, AdminID:integer): col·loca en estat de "En procés" tots els avisos rebuts de la imatge que se li passi com a paràmetre. Retorna el nombre d'avisos que s'han canviat d'estat. AdminID indica l'administrador que ha procésat l'avís.
- Validar(AdvertenciaID: integer, AdminID:integer): Dona com a procésada l'advertència sense eliminar la imatge a la que fa referència. Retorna 1 si el procés finalitza correctament. AdminID indica l'administrador que ha procésat l'avís.
- ValidarTot(ImatgeID:integer, AdminID:integer): Procésa totes les advertències que te imputades la imatge sense eliminar-la. AdminID indica l'administrador que ha procésat l'avís. Retornarà el nombre d'avisos procésats.
- Vist(AdvertenciaID: integer, resposta:string): dona com a revisada l'advertència per part de l'usuari, afegint una resposta com a justificant de la publicació prèvia de la imatge. Retorna 1 si el procés es realitza correctament.

Classes de propòsit especial

Determinades funcions de l'aplicació requereixen la intervenció de varis TableAdapters emmarcats dins del DataLayer. Les funcionalitats que expressen no tant sols es limiten a un àmbit de la aplicació com pot ser la gestió de votacions o el manteniment d'àlbums i categories. En aquest apartat descriuen les classes que contenen les funcionalitats dedicades a generar les estadístiques de l'aplicació i les cerques que es realitzen des de aquesta.

Estadístiques

cnRanking és la classe que engloba les funcionalitats per generar les estadístiques dels usuaris registrats i els administradors, així com els rankings d'imatges que genera l'aplicació. S'ha habilitat com una classe estàtica, en forma de llibreria de funcions.

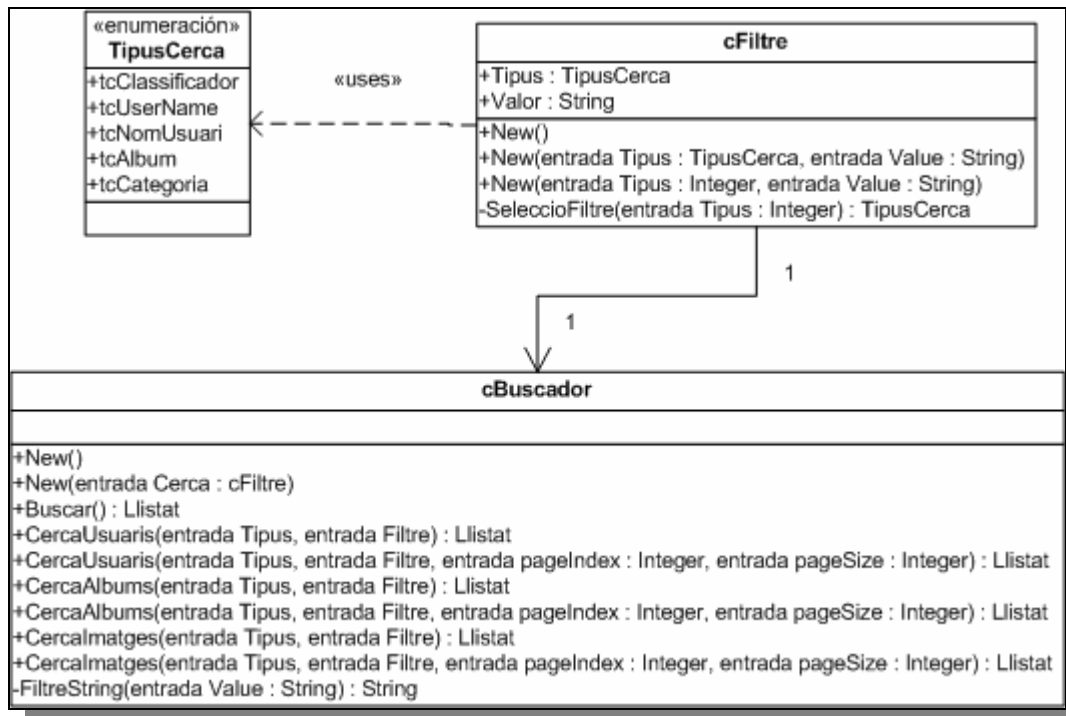
cnRanking
+LlistarImatges(entrada Inci : Date, entrada Final : Date, entrada Tipus : Integer) : Llistat
+LlistarImatges(entrada Inci : Date, entrada Final : Date, entrada Usuari : String) : Llistat
+AlbumStats(entrada Inci : Date, entrada Final : Date, entrada Usuari : String) : Llistat
+AlbumDetailStats(entrada Inci : Date, entrada Final : Date, entrada AlbumID : Integer) : Llistat
+AlbumPntStats(entrada Inci : Date, entrada Final : Date, entrada AlbumID : Integer) : Llistat
+AdvDetailStats(entrada Inci : Date, entrada Final : Date, entrada AlbumID : Integer) : Llistat
+ComDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+AdvDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+PntDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+AlbDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+IMGDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+USRDetailStats(entrada Inci : Date, entrada Final : Date) : Llistat
+TotalAdminStats(entrada Inci : Date, entrada Final : Date, entrada Tipus : Integer) : Llistat
+TotalUserStats(entrada AlbumID : Integer, entrada Inci : Date, entrada Final : Date, entrada Tipus : Integer) : Llistat

(figura 9) Classes cnRanking. No es mostren els table adapters declarats com atributs.

Tots els mètodes de la classe, exeptuant TotalAdminStats i TotalUserStats, estan vinculats a un ObjectDataSource per a ser enllaçats amb un control ASP.NET per a mostrar les dades. Les funcionalitats TotalAdminStats i TotalUserStats són utilitzats per el càlcul de totalitzacions en l'apartat d'estadístiques.

Buscador

El conjunt de classes que configuren el cercador de l'aplicació estan separades del sistema de classes connectores i de classes entitat. Al contrari que la classes cnRanking no es tracta d'una llibreria de funcions, les classes cBuscador, cFiltre i l'enumerador TipusCerca conformen un grup de classes orientades a objecte, similar a les classes entitat.



(figura 10) Classes cnRanking. No es mostren els table adapters declarats com atributs.

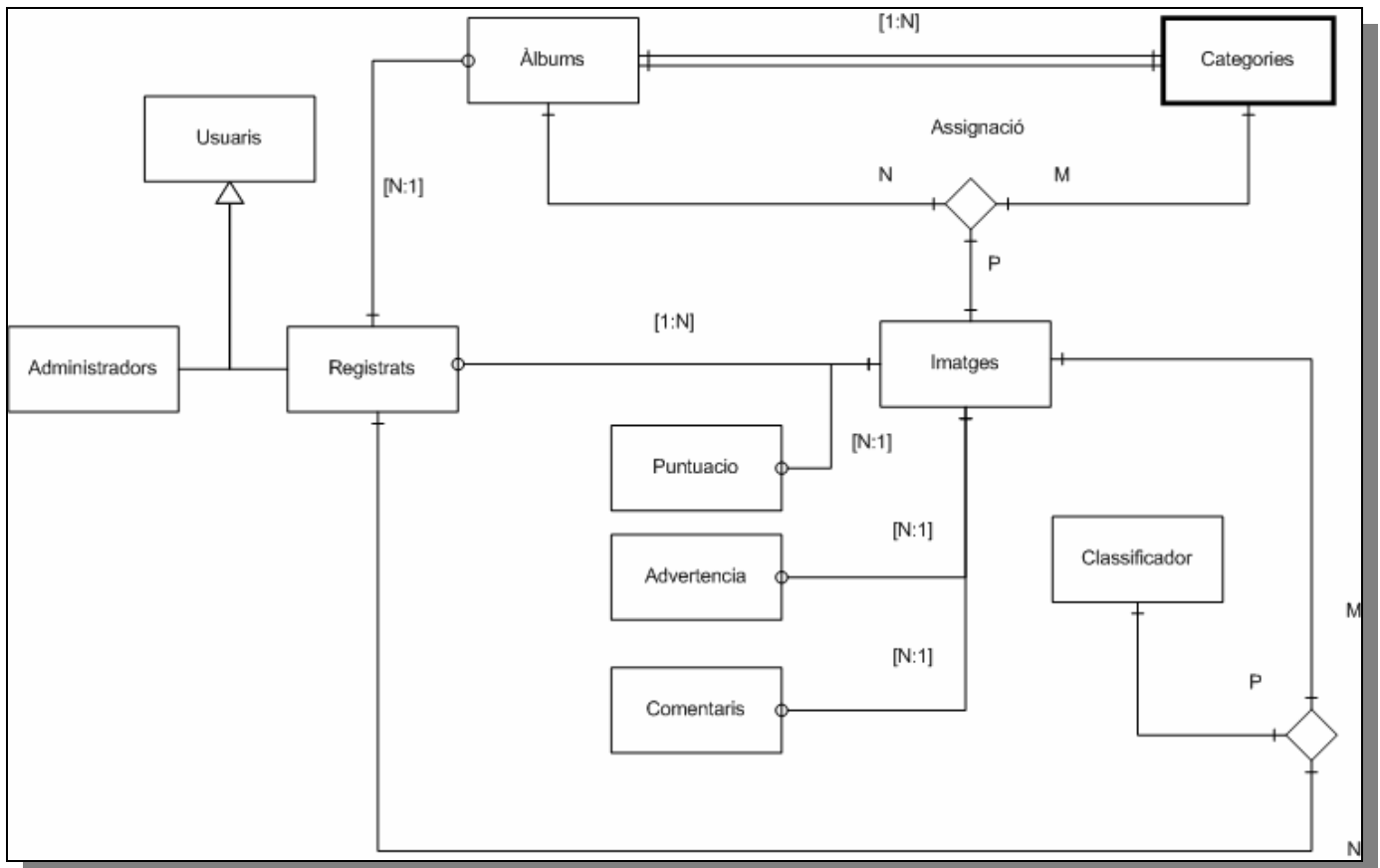
En la classe **cBuscador** és defineixen tots els mètodes per realitzar les cerques. Existeix una funció per a la definició de cadascuna dels diferents tipus de cerques¹⁵. Alguns d'aquests mètodes és dupliquen. S'implementa un segon mètode que retorni un llistat paginat. Els mètodes que retornin dades amb paginació són aquells que compten amb els paràmetres `pageIndex` i `pageSize`. La motivació que ens dur a utilitzar dades paginades es exposada en l'apartat d'implementació.

¹⁵ L'aplicació permet cerques per àlbum, categoria, nom d'usuari, nom personal del usuari i per classificadors

Disseny de la base de dades

El que es presenta a continuació és un disseny preliminar de la base de dades. Es projecta una base de dades normalitzada per ser implementada mitjançant el sistema de gestió de base de dades Microsoft SQL 2005 Express.

Diagrama Entitat/Relació



(figura 11) Diagrama d'entitat relació de la base de dades Photo Store

Descripció d'entitats i atributs

A continuació es presenta el detall de cadascuna de les entitats. Descriuint-ne els atributs (detallant-ne el tipus i la longitud, així com una), la clau principal i les foranies.

- Els camps en negreta representen la clau principal de l'entitat.
- Els camps en fons blau representen els atributs que permeten el valor nul.
- En la columna descripció es descriuen els camps amb un nom menys intuïtiu.
- En el nom de les entitats no s'utilitzen accents ni símbols no alfanumèrics, ja que podria generar problemes en la fase d'implementació del projecte.

Taula Usuaris¹⁶

Nom del camp	Tipus de dada	Longitud	Descripció
UsuariID	Autonumèric	8	Comptador numèric d'usuaris
NomUsuari	Caràcter	30	Nom del compte d'usuari
Nom	Caràcter	50	Nom de la persona
Adreca	Caràcter	255	
Cpostal	Caràcter	8	
Ciutat	Caràcter	125	
Província	Caràcter	75	
Pais	Caràcter	75	
Naixament	Data	-	Data de naixement
Descripcio	Caràcter	255	Breu descripció del usuari.
Tipus	Caràcter	20	Tipus d'usuari. Administrador o Usuari.
Foto	Imatge	-	Foto del usuari
Mime	Text	15	Defineix el tipus d'imatge a emmagatzemar.
Avisos	Numèric	8	Nombre d'aviso imputats a imatges del usuari
Eliminacio	Numèric	8	Nombre de imatges eliminades del usuari per part de l'administrador.

Taula Advertencia

Nom del camp	Tipus de dada	Longitud	Descripció
AdvertenciaID	Autonumèric	8	Comptador del nombre d'advertències d'una determinada foto
ImatgeID	Numèric	8	Codi identificador de la imatge.
UsuariID	Numèric	8	Usuari que realitza l'advertència.
AdminID	Numèric	8	Administrador que processa l'advertència.
Descripcio	Caràcter	255	Text que especifiqui per que s'ha enviat l'advertència.
Resposta	Caràcter	255	Text pot introduir l'usuari al revisar
UsuariIP	Caràcter	25	Adreça IP de l'usuari que puntua
Tipus	Caràcter	15	Classificador d'advertències
Data	Data	-	Data de creació
Vist	Booleà	1	Marca si l'usuari ha revisat l'advertència
Estat ¹⁷	Numèric	1	Identificador que marca com es resol l'advertència per l'administrador.

Claus Foranes

Origen		Destí	
Taula	Camp	Taula	Camp
Advertencia	ImatgeID	Imatges	ImatgeID

Taula Puntuacio

Nom del camp	Tipus de dada	Longitud	Descripció
PuntuacioID	Autonumèric	8	Comptador del nombre de puntuacions d'una determinada foto
ImatgeID	Numèric	8	Codi identificador de la imatge.
UsuariID	Numèric	8	Usuari que puntua
UsuariIP	Caràcter	25	Adreça IP de l'usuari que puntua
Puntuacio	Numèric	20	Puntuació de 1 al 5
Data	Data	-	Data de creació

¹⁶ Els usuaris de l'aplicació es gestionen amb el sistema Membership de Visual Studio. La informació dels comptes d'usuari és guardada en les taules aspnet_Users, aspnet_Membership i aspnet_Roles.

¹⁷ Valors possibles: 0 No revisada; 1: En procés; 2: Executada; 3: Desestimada

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Puntuacio	ImatgeID	Imatges	ImatgeID

Taula Comentaris

Nom del camp	Tipus de dada	Longitud	Descripció
ComentariID	Autonumèric	8	Comptador del nombre de comentaris d'una determinada foto
ImatgeID	Numèric	8	Codi identificador de la imatge.
UsuariID	Numèric	8	Usuari que puntua
UsuariIP	Caràcter	25	Adreça IP de l'usuari que puntua
Descriptor	Caràcter	255	Text del comentari
Data	Data	-	Data de creació
Vist	Booleà	1	Marca si un usuari ja ha llegit l'administrador

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Comentaris	ImatgeID	Imatges	ImatgeID

Taula Album

Nom del camp	Tipus de dada	Longitud	Descripció
AlbumID	Numèric	8	Identificador del àlbum
UsuariID	Numèric	8	Usuari que crea l'àlbum.
Títol	Caràcter	50	Nom de l'àlbum
Descripció	Caràcter	255	Text descriptiu del
Tipus	Booleà	1	Indica si l'àlbum esta oberta al públic o no.
Coberta	Numèric	8	Codi de la imatge escollida coma coberta del àlbum
Creacio	Data	-	Data de creació

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Album	UsuariID	Usuaris	UsuariID

Taula Categoria

Nom del camp	Tipus de dada	Longitud	Descripció
CategoriaID	Numèric	8	Identificador de la categoria.
AlbumID	Numèric	8	Identificador del àlbum al que es vincula la categoria.
Descriptor	Caràcter	50	Descriptor i identificador de la categoria dins del àlbum.
Coberta	Numèric	8	Codi de la imatge escollida coma coberta del àlbum

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Categoria	AlbumID	Album	AlbumID

Taula Assignacio

Nom del camp	Tipus de dada	Longitud	Descripció
AssignacioID	Autonumèric	8	Identificador numèric de l'assignació de la imatge a l'àlbum i a la categoria
AlbumID	Numèric	8	Identificador del àlbum
CategoriaID	Numèric	8	Identificador de la categoria
ImatgeID	Numèric	8	Identificador numèric de la imatge

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Assignacio	AlbumID, CategoriaID	Categoria	AlbumID, CategoriaID
Assignacio	ImatgeID	Imatges	ImatgeID

Taula Imatges

Nom del camp	Tipus de dada	Longitud	Descripció
ImatgeID	Numèric	8	Identificador numèric de la imatge
Imatge	Imatge	-	Fotografia a emmagatzemar.
DataCreacio	Data	-	Data de creació
Comentari	Text	255	Comentari sobre la imatge
UserID	Numèric	8	Usuari que publica la imatge.

Taula ClassImg

Nom del camp	Tipus de dada	Longitud	Descripció
ClassImg	Autonumèric	8	Identificador numèric de l'assignació del classificador a la imatge
ClassID	Numèric	8	Identificador del Classificador
ImatgeID	Numèric	8	Identificador numèric de la imatge

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
ClassImg	ImatgeID	Imatges	ImatgeID
ClassImg	ClassID	Classificador	ClassID

Taula Classificador

Nom del camp	Tipus de dada	Longitud	Descripció
ClassID	Autonumèric	8	Comptador del nombre de classificadors d'una determinada foto
UsuariID	Numèric	8	Usuari que crea el classificador
Descriptor	Text	30	Tag, classificador de la imatge.

Claus Foranes			
Origen		Destí	
Taula	Camp	Taula	Camp
Classificador	UsuariID	Usuaris	UsuariID

Implementació

Sistema de gestió d'usuaris

L'aplicació fa ús del sistema Membership de Microsoft com a gestor d'usuaris. El mencionat sistema proporciona als desenvolupadors un conjunt d'eines (llibreries i controls) per a la creació i gestió d'usuaris, així com el control de la validació en l'aplicació.

Rols i restriccions

Es crea la base de dades ASPNETDB on es guarda tota l'estructura de dades necessària pel funcionament de Membership. A través del lloc web es defineixen els rols que poden realitzar els usuaris que treballin en la nostra aplicació. Cadascun dels rols s'hauria de correspondre amb un dels actors que intervenen en els casos d'ús definits en el disseny de l'aplicació, però finalment es defineixen els rols administrador i usuari, aquest últim equival a l'actor usuari registrat. Es pretén crear un entorn amb restriccions que segons el rol que s'imputi a l'usuari se li permetrà l'accés a diverses funcionalitats reservades de l'aplicació.

Una vegada creats els rols procedim a crear les restriccions a l'aplicació. En la nostra aplicació hem creat una estructura de directoris. En dos d'aquests, admin i user, creem totes les funcionalitats reservades, d'administrador i usuaris registrats respectivament. Des del gestor de lloc web, establim les restriccions a cadascun dels directoris, donant pel rol del usuari.

Finalment crearem l'usuari administrador que gestionarà l'aplicació. Els administrador no tenen un assistent de creació d'usuaris, accessible des de la mateixa aplicació com el que es disposa per crear usuaris clients. A través del gestor de la pàgina web crearem l'usuari i li assignarem el rol administrador.

Procéssos i accions vinculats al sistema Membership

L'entorn de disseny de Visual Studio ens ofereix una col·lecció de controls predissenyats i amb funcionalitats específiques per a la gestió dels usuaris. A continuació descrivim els utilitzats en l'aplicació:

Creació d'usuari

Per a la creació d'usuaris dispossem del assistent CreateUserWizard amb una sèrie de passos predissenyats. En ell se'ns permet definir les dades del compte d'usuari. Es tracta d'un control molt intuïtiu on seguint les instruccions que se'ns demana, en el nostre cas només l'entrada de les dades de compte¹⁸. Una vegada completada el

¹⁸ Nom d'usuari, correu electrònic, password, pregunta i resposta de seguretat.

procés d'entrada de dades, validarà la informació i en cas de que sigui correcta ens donarà una confirmació permetem-nos l'entrada a l'entorn privat.

Validació

El control Login és la eina que se'ns proporciona per controlar la validació en el sistema. Aquest control incorpora un sistema de validació que ens permet validar i iniciar una sessió només introduint les dades d'usuari, sense necessitat d'inserir una sola línia de codi. En l'event **LoggedIn**, que s'executa després de la validació inicialitzem les variables per usuari.

Modificació de les dades de seguretat

Tant els usuaris registrats com els usuaris administradors compten dins del seu entorn privat d'una secció per modificar les dades de seguretat. La secció permet canviar el password d'usuari. Per dur a terme aquest utilitzem el control ChangePassword, el qual permet una validació del nou mot de pas d'una forma totalment automatitzada, ometent totes les accions relatives a encriptació / desencriptació del password en la base de dades.

La modificació de la pregunta i resposta de seguretat no esta implementada en forma de control pel sistema membership. En la nostra aplicació hem implementat un assistent, usant com a base el control Wizard, per realitzar aquest canvi. L'assistent compta els següents passos:

1. Visualitzem l'actual pregunta de seguretat.
2. Ens demana que entrem el nostre mot de pas.
3. Introduïm la nova pregunta i resposta.

Si és validen correctament les dades tornem al primer pas, on es mostra la pregunta que acabem d'entrar. En cas de que s'introdueixi alguna dada incorrecta, se'ns comunicarà en el moment de realitzar la validació final de les dades.

Recuperar les dades d'usuari

Implementem aquesta utilitat amb la intenció de que qualsevol usuari pugui recuperar l'accés al entorn privat de l'aplicació, en el cas que hagi oblidat o perdut el seu mot de pas. Per dur a terme aquesta tasca el sistema Membership compta amb el control PasswordRecovery, tot i així n'hem descartat l'ús ja que l'aplicació photostore utilitza una encriptació hash. L'operativa del control permetia recuperar el password via e-mail. Però el password rebut continuava estant encriptat, per tant hem decidit decantar-nos per la opció de crear un assistent que permeti canviar el password per un de nou.

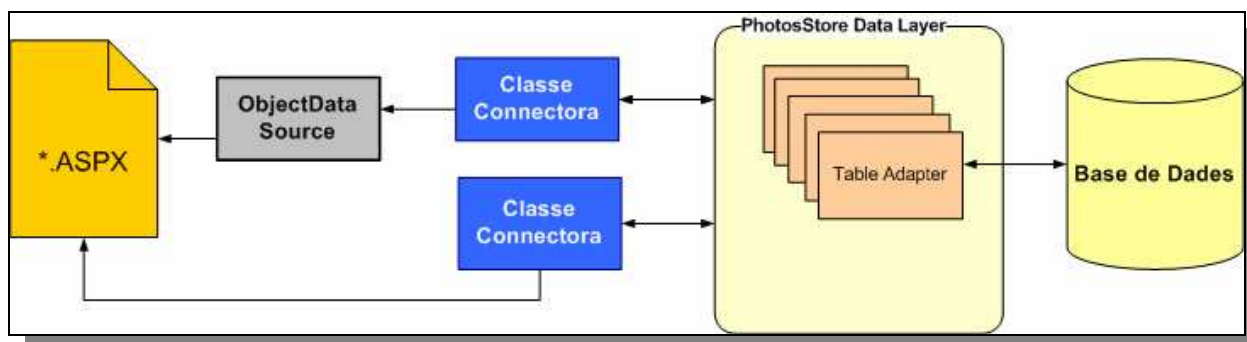
El procés s'estructura en forma d'assistent i compta amb els següents passos:

1. Introduïm el nom d'usuari del qual desitgem canviar-ne el mot de pas.
2. Se'ns requereix que contestem la pregunta de seguretat establerta per l'usuari.
3. Introduïm el nou password.

Finalment se'ns confirma si s'ha validat correctament el canvi. En cas de ser així, al finalitzar l'assistent se'ns retorna a la pàgina de validació per a que pugem accedir a l'entorn privat de l'aplicació. En cas d'introduir alguna dada incorrecta durant el procés se'ns notificarà en el moment d'accedir al següent pas.

Accés a dades

A continuació fem un repàs de les característiques del sistema d'accés a dades de la aplicació. El següent gràfic il·lustra les variants d'accés a dades de les que fem ús:



(figura 12) Diagrama d'entitat relació de la base de dades Photostore

Les dades s'emmagatzemen en una base de dades SQL SERVER 2005, sobre aquesta es realitzaran operacions de selecció, inserció, modificació i esborrat. A continuació fem una breu descripció de tots els components que intervenen en el procés.

PhotoStore Data Layer

Es tracta d'un arxiu amb extensió XSD i format XML, on es defineix tota la col·lecció d'objectes table adapters. Aquests objectes permeten la carrega de dades per mitjançant un conjunt de consultes definibles per el programador. Es dona la possibilitat d'implementar consultes per la modificació, inserció i eliminació de dades. Se'ns ofereix la possibilitat d'implementar processos emmagatzemats per tractar les dades, en el nostre cas només utilitzarem consultes SQL.

En aquest contenidor es vinculen les dues bases de dades utilitzades per l'aplicació Photostore.mdf i ASPNETDB.mdf.

La utilització d'aquest sistema ens permetrà un desenvolupament senzill de tot l'entramat de sentències utilitzades per accedir i modificar les dades. Es tracta d'un entorn amigable i senzill d'utilitzar.

Classes connectores

Com s'ha comentat en l'apartat de disseny, les classes connectores són el conjunt de classes que encapsulen el conjunt de sentències generades en els diferents table adapters. La encapsulació es dur a terme per tres motius:

1. Implementar funcionalitats que requereixen de l'execució de varies sentències, molt més complexes de les que es poden implementar directament en un table adapter.
2. Per implementar un objecte funcional que es pugui enllaçar amb un `ObjectDataSource`, que a la vegada estarà vinculat a un control que mostrarà les dades de la consulta.
3. Implementar objectes d'origen de dades paginats per enllaçar-los als `ObjectDataSource`

En la figura 12 es mostra les dues vies d'utilització d'aquestes classes. Declarada com una variable dins l'arxiu de codi del arxiu aspx (1) o a través d'un `ObjectDataSource` (2) i (3). Per habilitar una classe connectora per a ser enllaçada a un control origen de dades, hem d'encapçalat la classe amb la següent etiqueta:

```
<System.ComponentModel.DataObject()> _
```

I cada funció que realitzi una selecció parametrizada estarà encapçalada per l'etiqueta:

```
<System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, False)> _
```

Una vegada complimentats aquets requisits, ja podrem realitzar l'enllaç amb l'objecte pertinent.

ObjectDataSource

L' `ObjectDataSource` és un control del que disposa ASP.NET per tal de crear un vincle amb un origen de dades. L'objecte és neutre, permet enllaçar amb qualsevol tipus de base de dades.

En la nostra aplicació, serà enllaçat a mètodes de selecció de dades de les nostres classes connectores. En la propietat `typename` entrarem el nom de la classe connectora i en la propietat `selectmethod` el nom de la funció utilitzada per realitzar la consulta de selecció. Existeix la possibilitat de definir mètodes per la modificació, inserció i eliminació de la informació, però en la nostra aplicació aquestes funcions les realitzarem amb la crida de mètodes de les classes connectores.

Els mètodes utilitzats en els controls, estaran parametrizats per donar una major flexibilitat a la funcionalitat. Per definir els paràmetres recorrem a l'assistent que se'ns ofereix en la propietat *SelectParameters*. En aquest assistent enllacem el paràmetre de la funció amb la variable que emmagatzema el valor. Aquesta valor pot estar emmagatzemada en una variable de sessió, declarada quan l'usuari es valida en l'entorn. Les altres opcions són enllaçar el parametre directament a un control o a una variable *quereystring*¹⁹.

A través de les classes connectores es definiran els orígens de dades paginats. Es retornarà una variable del tipus *PagedDataSource*, que s'haurà configurat i on s'hauran carregat els valors prèviament.

Paginació

En certes funcionalitats de l'aplicació es requereix que es mostrin per pantalla grans quantitats de dades llistades en forma de graella o de taula. És poc òptim atendre aquestes peticions mostrant tots els registre de cop, el consum de recursos del sistema i el temps utilitzat en l'operació ho fa inacceptable. Per evitar-ho recorrem a la paginació de les dades. Aquest recurs ens permetrà mostrar els registres dividits en grups de n registres, on n es un nombre definible per l'equip de desenvolupament.

A l'hora de realitzar la paginació el comportament es diferent segons el control sobre el que es treballi. En l'aplicació és mostren les dades en controls com graelles i taules.

Les graelles²⁰ permeten paginar les dades des del mateix control. Editant les propietats *Permetre paginació* i *nombre de registres per pàgina* establim configurem el control per hostatjar dades paginades.

Les taules²¹ pel contrari, no donen les facilitats mencionades per paginar. Disposen d'un conjunt de propietats que s'encarreguen de filtrar les dades que es mostren per pantalla filtrant l'origen de dades amb el seus valors. Descartem aquesta possibilitat i vinculem el control a una classe connectora i a un mètode d'aquesta classe, implementat per retornar un origen de dades paginat. Les funcions amb aquesta característica es distingeixen de les demás per comptar amb els paràmetres *pageIndex* i *pageSize* per definir l'índex de pàgina que carreguem i el nombre de registres d'aquesta.

¹⁹ El valor es passa concatenant la variable i el valor a la URL de la pàgina.

²⁰ Les graelles es corresponen al control ASP.NET GridView

²¹ Les taules es corresponen al control ASP.NET DataList

Implementació de serveis

L'apartat d'implementació de serveis s'inclou dins de la secció capa de dades, degut a que es tracta a sistemes de càrrega de dades en controls de la llibreria AJAX Control Toolkit. Comptem amb dos serveis que realitzen les següents utilitats:

- Càrrega d'imatges d'una cerca o d'un àlbum en un slideshow per ser mostrades com a diapositives.
- Càrrega de la llista de per autocompletar el camp de classificadors/ tags en les dos modalitats que s'ofereixen.

Els dos serveis segueixen la mateixa estructura a l'hora de realitzar la carrega de les dades:

- Execució de la consulta per carregar la informació en un objecte de dades definit en el Data Layer Photostore. En el cas del slideshow es realitza la mateixa consulta que s'ha realitzat prèviament per realitzar la cerca.
- Mitjançant un bucle carreguem les dades en una matriu redimensionable d'un tipus d'objecte compatible amb el control on s'ha de carregar.
- Retornem la matriu

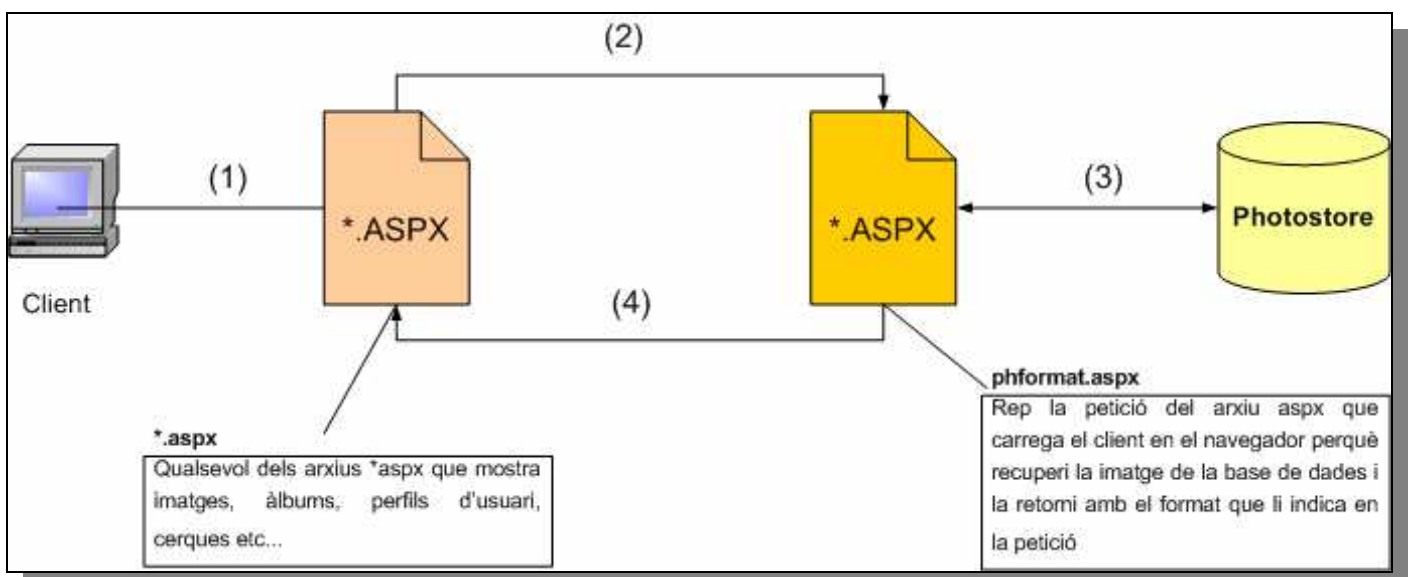
Formatat de les imatges

L'aplicació guarda les imatges emmagatzemades en una base de dades SQL Server, en el moment de recuperar-les de la taula on estan guardades i mostrar-les per pantalla se'ns presenta el primer dilema, crear la imatge físicament en el sistema d'arxius del servidor o carregar-la en memòria.

El primer sistema extrauria un arxiu de la base de dades i en crearia una física ocupant memòria física en l'equip on s'hostatgi l'aplicació, obligant a implementar un sistema de manteniment per alliberar aquestos recursos.

Es descarta aquesta opció per el possible consum de recursos que representa i s'escull l'alternativa de la copia de la imatge en memòria, que permetrà mostrar la imatge carregant el contingut en memòria de l'equip client, el qual fa la petició de visualització via navegador web, i alliberant els recursos que ocupi la imatge al descarregar la pàgina.

A continuació es descriu els passos que ha de realitzar el sistema per carregar i donar format a la imatge que es sol·licita:



(figura 11) Esquematzació de la carrega/formatat d'imatges de l'aplicació

1. L'usuari carrega una pàgina on és mostren imatges emmagatzemades en la base de dades. Visualització/manteniment d'àlbums, cerca d'imatges, compte d'usuaris...
2. És realitza una criada al arxiu phformat.aspx, enviant el paràmetre Id de la imatge i el paràmetre tipus que identifica el tipus de format que indica amb quines característiques ha de se carregada en memòria.
3. phformat.aspx recupera la imatge de la base de dades mitjançant una consulta.

4. Carreguem en memòria la imatge amb les característiques que corresponguin al paràmetre corresponent enviat en la petició (2).

Segons el context en que visualitzem una imatge, pot ser que requerim que les seves característiques a l'hora de ser mostrada per pantalla variïn. No es el mateix mostrar un resultat d'una cerca d'imatges que les miniatures que representen la coberta d'un àlbum /categoria o la imatge del compte d'usuari. En cada circumstància és requereix un tamany diferent per exemple. Per recuperar, formatar i carregar en memòria la imatge les funcions seguiran la següent estructura:

```
Private Sub GenerarImatge(ByVal ImatgeID As Integer, ByVal Opcio As Integer)
    Dim iH As Long, iW As Long
    Dim img As System.Drawing.Image
    Dim cnImg As New cnImatge
    Dim tbImg As PhotoStore.ImatgesDataTable=cnImg.GetImatgeRow(ImatgeID)
    Es carrega la imatge des de la base de dades

    If tbImg Is Nothing Then
        img =Image.FromFile(Server.MapPath("~/imatges/noalbum.png"))
        En cas de no trobar la imatge se'n carrega una per defecte
    Else
        Dim rwImg As PhotoStore.ImatgesRow = tblImatge(0)
        img = System.Drawing.Image.FromStream(New
            _System.IO.MemoryStream(rwImg.imatge))
        Convertim la imatge en un stream de memoria
    End If
    Call ImageResize(img.Size.Width, img.Size.Height, iW, iH, Opcio)
    Carreguem les caracteristiques amplada/longitud segons el paràmetre
    Opcio
    Call EscriuImatgeEnMemoria(img, iW, iH)
    Escrivim l'imatge en memoria
End Sub
```

Utilització de la tecnologia AJAX en l'aplicació

Augmentar la interactivitat de l'aplicació

Per tal d'evitar la recarrega de les pàgines web cada cop que es realitza una petició al servidor usarem el control update panel.

Incloem un gestor d'scripts a la pàgina i ha continuació hi inserim un update panel. Aquest control funciona com un contenidor, on afegim els controls o la part de la pàgina web que desitgem que treballi sota aquesta metodologia. En el moment d'executar el codi de qualsevol dels seus events no es recarregarà la pàgina però generarà un postback.

A l'hora de crear una pàgina amb les seves funcionalitats és realitzarà el següent procés:

1. Es crea una nova pàgina aspx, escollint una de les master pages²² predissenyades amb les que comptem.
2. S'afegeix un script manager al cos de la pàgina
3. Realitzem el disseny de la pàgina.
4. Abans de la implementació del codi decidirem quines parts del web han d'estar subjectes a la filosofia de treball AJAX i col·locarem els update panels que siguin necessaris.
5. Implementem el codi i realitzem les proves necessàries per assegurar els seu correcte funcionament
6. Finalment podem optar per potenciar l'usuabilitat de la pàgina mitjançant la col·lecció de controls i utilitats del AJAX Control Toolkit.

Hi ha determinats controls que no poden realitzar la seva tasca estan dins d'un UpdatePanel. Una de les utilitats centrals de la nostra aplicació, l'apartat de publicació d'imatges, compta amb un FileUpload. Aquest control es l'encarregat de seleccionar el fitxer per guardar a la base de dades. FileUpload requereix que el codi que interactua amb el control estigui fora del update panel, ja que del contrari pot provocar accions inesperades.

Finalment, mencionem l'ús del control Timer per complementar la funcionalitat encarregada de mostrar missatges d'informació. Al mostrar un missatge es posarà en marxa un Timer que al cap de tres segons ocultarà el missatge i deshabilitarà el temporitzador.

²² Plantilla base d'una plana aspx.

Millorar l'ergonomia de les interfícies

Per potenciar la usabilitat de la nostra aplicació i fer-la més atractiva a l'usuari, en quan a disseny, complementarem la maquetació de les pàgines aspx amb els components de l'AJAX Control Toolkit. Incorporarem a les nostres interfícies les següents utilitats:

- Missatges modals: Implementem un sistema de missatges/pop-ups modals que permeten mostrar panells emergents deixant l'aplicació en espera fins que aquest no s'oculti.
- Implementació d'interfícies aspx amb parts ocultes. Aquestes parts de la pantalla romandran ocultes fins que no es desencadeni l'acció que les mostra.
- Pantalla en acordió. Una segona variant seria la construcció de pàgines amb el contingut organitzat en forma d'acordió. La pantalla està dividida en varies seccions, de les quals només se'n mostra una. Al fer clic en la capçalera de les seccions és mostrarà, la secció adjunta a aquesta. Els apartats d'estadística de l'entorn privat en són un exemple.
- La visualització d'imatges en slideshow.
- La resta d'aplicacions d'aquesta llibreria són extensions de les funcionalitats de controls ASP.NET o controls propis d'AJAX Control Toolkit. Tabs, cercadors de llistes, sistema de puntuació o menús emergents en són alguns exemples.

Complements addicionals afegits a l'aplicació

Generador de gràfics

S'han incorporat controls externs a la plataforma ASP.NET de Visual Studio per a poder realitzar funcionalitats requerides en l'anàlisi. S'ha afegit el control WebChart.

<http://www.carlosag.net/Tools/WebChart/Default.aspx>

Descripció del procés

Aquest control genera un gràfic a de les dades que es vagin afegint a l'objecte Data. Per generar els gràfics realitzem el procés que es representa a continuació amb pseudocodi:

```
Protected Sub GenerarGraph()  
    ChartControll1.Charts.Clear() ` S'inicialitza el control  
    chart = New WebChart.ColumnChart() ` Crearem un nou gràfic  
    DataEnum = GetStatsEnumerator() `Obtenim les dades paginades23  
    While (DataEnum.MoveNext())  
        With chart  
            .Data.Add(New WebChart.ChartPoint(String.Format("{0:dd/MM}",  
dRow("Data")), dRow("contar")))  
            `afegim les dades al gràfic  
        End With  
    End While  
    `Finalment és formataria el gràfic  
End Sub
```

Inconvenients

El control webchart genera els gràfics creant una imatge amb la il·lustració de les dades que li passem. El problema es que la imatge creada no es guarda en memòria sinó que és genera un fitxer físic, en un directori temporal. No ha de ser un problema ja que és tracta de fitxers amb un tamany reduït. Una de les tasques de millora a realitzar en una possible segona fase seria la implementació d'un sistema que permetés generar el gràfic carregant-lo en memòria.

²³ L'origen de dades es un origen de dades paginat que es converteix en un objecte IEnumerator per tal de ser recorregut amb un bucle while

Línies de desenvolupament futur

A continuació s'enumeraran i es descriuran breument algunes de les possibles millores a realitzar en l'aplicació en una segona fase de desenvolupament.

1. Sistema d'impressió. Impressió d'àlbums, categories i fotografies de forma individual.
2. Implementar un sistema de cerca per text complet. Cerques de les descripcions introduïdes en els àlbums, categories i imatges.
3. Crear una aplicació d'escriptori que permeti publicar imatges sense validar-se l'entorn web.
4. Crear un sistema de favorits. On Per agregar els usuaris amics, els seus àlbums o imatges preferides.
5. Crear un sistema de subscripció RSS, per difondre els continguts que vagin publicant els usuaris.

Conclusions

Després de concloure la implementació del projecte, podem afirmar que s'ha assolit els objectius plantejats en els seu inici. Hem desenvolupat una aplicació senzilla i fàcil d'utilitzar, dirigida a un espectre ampli de públic que no requerirà de cap coneixement explícit ni avançat per a la seva utilització.

S'ha aconseguit dissenyar una aplicació el suficientment encapsulada i modularitzada per optimitzar-ne el manteniment i facilitar-ne futures modificacions. Podem considerar que l'aplicació de les tecnologies seleccionades per la realització del projecte a s'ha superat obtenint resultat notable.

Finalment, a nivell personal ha resultat un enriquiment com a professional del desenvolupament de software, la consolidació de coneixements sobre la tecnologia ASP.NET i l'adquisició de nous sobre tecnologia AJAX. En un altre aspecte a permès posar en pràctica tots els coneixements respecte a la gestió de projectes, així com anàlisi i disseny d'aplicacions informàtiques.

Bibliografia

Pàgines Web Oficials	
ASP.NET www.asp.net	Lloc web oficial de Microsoft per a la tecnologia asp.net
AJAX ajax.asp.net	Lloc web oficial de Microsoft per a la tecnologia AJAX
AJAX Control Toolkit www.asp.net/ajax/ajaxcontroltoolkit/	Lloc web de Microsoft per a la descarrega i documentació del AJAX control toolkit
Microsoft Developers Network http://msdn.microsoft.com	Web per a desenvolupadors de Microsoft
Microsoft TechNet http://technet.microsoft.com	Web de recursos per a professionals de les Technologies de la informació de Microsoft
Pàgines Web Oficials	
The Code Project www.thecodeproject.com	Pàgina especialitzada en el desenvolupament ASP.NET i la tecnologia AJAX
Windows Development Center www.windowsdevcenter.com/	Pàgina orientada al desenvolupament .NET. Utilitzada per a la documentació i la cerca d'exemples per la formatació d'imatges.
ASP Alliance http://aspalliance.com/	Pàgina on s'ofereixen tutorial, exemples i eines per al desenvolupament .NET