Projecte Argos

TFC – Xarxes de Computadors

Josep Romeu i Soberana

Enginyeria Tècnica en Informàtica de Gestió
Universitat Oberta de Catalunya

Tutora del treball
Maria Isabel March Hermo

13 de gener de 2007

UOC
www.uoc.edu

Copyright i llicència

Copyright © 2006, Josep Romeu i Soberana

D'acord amb l'article 31 de la Llei de Propietat Intel·lectual, aquest document només pot ser reproduït per a ús privat sense autorització de l'autor, sempre que la còpia es faci amb finalitats d'estudi, docència o recerca i no sigui utilitzada amb fins lucratius.

A la Magda, l'Ariadna i l'Aleida per la seva paciència

Resum

Aquest projecte és el Treball Fi de Carrera d'Enginyeria Tècnica en Informàtica de Gestió de l'autor, a la Universitat Oberta de Catalunya.

En aquesta memòria es pretén tractar tots els punts de disseny d'un *servidor proxy web*, des de la recollida i documentació de requisits fins a la seva implementació. Inicialment, i quan a l'abast d'aquest projecte, s'especifiquen i implementen dues de les tasques que pot dur a terme un *proxy*: el filtrat de continguts web i la creació d'un registre històric de tots els recursos web sol·licitats.

El projecte està basat en el model *client-servidor* i la *programació per capes*. L'anàlisi, disseny i implementació estan desenvolupats sota el paradigma de *programació orientat a objectes*, usant per a la seva especificació el llenguatge *UML (Unified Modeling Language)*.

Pensant en els possibles destinataris de l'aplicació (escoles, biblioteques, cibercafés, empreses, particulars, ...) i sent-ne l'autor partidari, s'utilitza *programari lliure* i *multiplataforma* per a desenvolupar el prototipus. En concret, es trien el llenguatge *PHP* (*PHP: Hypertext Preprocessor*) i el motor de base de dades *SQLite*.

L'autor posa èmfasi en el disseny del projecte, cimentat, com ja s'ha esmentat, en la *programació orientada a objectes* i en la *programació per capes*, les quals faciliten la seva reutilització i la seva implementació en diversos llenguatges (C++, Java, Visual Basic, ...).
L'elecció del llenguatge *PHP* per a implementar el projecte, representa per a l'autor un repte afegit i l'exploració de l'ús de les funcions de *sockets* de *PHP*, encara en fase experimental, però, segons les proves de test efectuades, força estables en aplicacions ben programades.

Índex

CAPÍTOL 1 - INTRODUCCIÓ 7

1.1. JUSTIFICACIÓ I CONTEXT. 7
1.2. OBJECTIUS. 8
1.3. ENFOCAMENT I MÈTODE. 8
1.4. PLANIFICACIÓ DEL PROJECTE. 8
1.5. DESVIACIÓ EN LA PLANIFICACIÓ DEL PROJECTE. 10
1.6. PRODUCTE OBTINGUT. 11
1.7. DESCRIPCIÓ DELS CAPÍTOLS. 11

CAPÍTOL 2. RECOLLIDA I DOCUMENTACIÓ DE REQUISITS 12

2.1. FONTS D'INFORMACIÓ. 12
2.2. INFORMACIÓ INICIAL. 12
2.3. EL PROGRAMA *PROXY*. 12
2.3.1. Filtrat de continguts. 14
2.3.2. Registre de peticions. 14
2.3.3. Control d'usuaris. 14
2.4. EL PROGRAMA *GESTORPROXY*. 14
2.4.1. Configuració del Proxy. 15
2.4.2. Seguiment de l'activitat del Proxy. 15
2.4.3. Gestió de l'eina de filtrat. 15
2.4.4. Gestió de l'historial. 16
2.4.5. Gestió d'usuaris. 16
2.5. EL CONTEXT DEL PROGRAMARI. 16
2.5.1. El model del domini. 16
2.5.2. El model de negoci. 17
2.5.3. El glossari del model de negoci. 19
2.6. ELS GUIONS. 19
2.7. ACTORS. 20
2.8. CASOS D'ÚS. 20
2.9. INTERFÍCIE D'USUARI 24
2.9.1. Perfils d'usuari. 24
2.9.2. Descripció de les tasques. 24
2.9.3. Requisits d'usabilitat. 25

CAPÍTOL 3. ANÀLISI DE REQUERIMENTS 26

3.1. PAQUETS. 26
3.2. REVISIÓ DE CASOS D'ÚS. 26
3.3. ESPECIFICACIÓ DE LES CLASSES D'ANÀLISI. 29
3.3.1. Identificació de les classes d'entitats. 29
3.3.2. Especificació dels atributs de les classes d'entitats. 29
3.3.3. Relacions de les classes d'entitats. 30
3.3.4. Identificació de les classes de frontera, de control i de les operacions. 32
3.4. ESPECIFICACIÓ FORMAL DELS CASOS D'ÚS. 36
3.5. INTERFÍCIE D'USUARI. 38

CAPÍTOL 4. DISSENY 42

4.1. PARADIGMES I TECNOLOGIES. 42
4.1.1. El model client-servidor. 42
4.1.2. La programació per capes. 42
4.1.3. PHP i SQLite. 43
4.2. DISSENY DEL MODEL DE BASE DE DADES. 43
4.2.1. Definició de l'estructura de base de dades. 44
4.3. DISSENY DE LA LÒGICA DE NEGOCI. 47
4.3.1. Control del flux de la informació. Sockets. 47
4.3.2 .Interpretació de les peticions de recursos web. Protocols HTTP i HTML. 47
4.3.3. El disseny del casos d'ús. 48

4.4. DISSENY DE LA INTERFÍCIE GRÀFICA. 51
4.4.1. La interfície gràfica: HTML....51
4.4.2. Descripció de l'estructura de la interfície....51
4.5. DISSENY DELS SUBSISTEMES. 52

CAPÍTOL 5. IMPLEMENTACIÓ I TESTING....54

5.1. FASES DE LA IMPLEMENTACIÓ. 54
5.1.1 Fase 1. Control del flux d'informació. Proxy bàsic....54
5.1.2 Fase 2. Subsistema de base de dades. 55
5.1.3 Fase 3. Proxy primari. Enllaç amb les base de dades. 56
5.1.4 Fase 4. Generació interfície gràfica....57
5.1.5 Fase 5. Prototipus Proxy. Servei Web Server....57
5.2. ESTRUCTURA DELS COMPONENTS DEL PROTOTIPUS. 57
5.3. TESTING. 58
5.4. ESTAT ACTUAL I FUTURES MILLORES....58

CAPÍTOL 6. CONCLUSIONS 59

GLOSSARI....60

BIBLIOGRAFIA....61

Índex de figures

1.1 CALENDARI TFC 10
2.1 ESQUEMA FUNCIONAMENT DEL PROXY 13
2.2 DIAGRAMA DE CLASSES DEL MODEL DEL DOMINI 17
2.3 RELACIÓ DELS USUARIS DEL PROGRAMARI I DEL QUÈ NECESSITEN 17
2.4 DIAGRAMA DE CASOS D'ÚS DEL MODEL DE NEGOCI 18
2.5 DIAGRAMA DE COL·LABORACIÓ DEL MODEL DE NEGOCI 18
2.6 DIAGRAMA DE CASOS D'ÚS 20
3.1 PAQUETS D'ANÀLISI I SERVEIS (ESQUEMA DE RELACIONS) 26
3.2 DIAGRAMA DE CASOS D'ÚS REVISATS 28
3.3 DIAGRAMA DE CLASSES D'ENTITATS 31
3.4 DIAGRAMA DE COL·LABORACIÓ CAS Nº 1 "INICIAR SESSIÓ" 32
3.5 DIAGRAMA DE COL·LABORACIÓ CAS Nº 2 "FER PETICIÓ" 32
3.6 DIAGRAMA DE COL·LABORACIÓ CAS Nº 3 "CONSULTAR HISTORIAL" 32
3.7 DIAGRAMA DE COL·LABORACIÓ CAS Nº 4 I 5 "GESTIONAR USUARIS (CONSULTA)" 33
3.8 DIAGRAMA DE COL·LABORACIÓ CAS Nº 4 I 5 "GESTIONAR USUARIS (ALTA)" 33
3.9 DIAGRAMA DE COL·LABORACIÓ CAS Nº 4 I 5 "GESTIONAR USUARIS (MODIFICACIÓ)" 33
3.10 DIAGRAMA DE COL·LABORACIÓ CAS Nº 4 I 5 "GESTIONAR USUARIS (BAIXA)" 33
3.11 DIAGRAMA DE COL·LABORACIÓ CAS Nº 6 "GESTIONAR FILTRE (CONSULTA)" 34
3.12 DIAGRAMA DE COL·LABORACIÓ CAS Nº 6 "GESTIONAR FILTRE (ALTA)" 34
3.13 DIAGRAMA DE COL·LABORACIÓ CAS Nº 6 "GESTIONAR FILTRE (MODIFICACIÓ)" 34
3.14 DIAGRAMA DE COL·LABORACIÓ CAS Nº 6 "GESTIONAR FILTRE (BAIXA)" 34
3.15 DIAGRAMA DE COL·LABORACIÓ CAS Nº 7 "GESTIONAR PROXY (CONSULTA)" 35
3.16 DIAGRAMA DE COL·LABORACIÓ CAS Nº 7 "GESTIONAR PROXY (CONFIGURAR)" 35
3.17 DIAGRAMA DE COL·LABORACIÓ CAS Nº 8 "TANCAR SESSIONS" 35
3.18 DIAGRAMA DE COL·LABORACIÓ CAS Nº 9 "CANVIAR CONTRASENYA" 35
3.19 DIAGRAMA DE COL·LABORACIÓ CAS Nº 10 "SORTIR DE SESSIÓ" 36
3.20 DIAGRAMA D'ESTATS CAS Nº 2 "FER PETICIÓ" 36
3.21 DIAGRAMA DE SEQÜÈNCIES CAS Nº 2 "FER PETICIÓ" 37
3.22 INTERFÍCIE DE LA CLASSE FRONTERA *P_IDENTIFICACIO* 38
3.23 INTERFÍCIE DE LA CLASSE FRONTERA *P_CANVICONTRASENYA* 38
3.24 INTERFÍCIE DE LA CLASSE FRONTERA *P_CONSULTARHISTORIAL* 38
3.25 INTERFÍCIE DE LA CLASSE FRONTERA *P_DADESHISTORIAL* 39
3.26 INTERFÍCIE DE LA CLASSE FRONTERA *P_ALTAUSUARI* 39
3.27 INTERFÍCIE DE LA CLASSE FRONTERA *P_CONSULTAUSUARI* 39
3.28 INTERFÍCIE DE LA CLASSE FRONTERA *P_MODIFICACIOUSUARI* 40
3.29 INTERFÍCIE DE LA CLASSE FRONTERA *P_ALTA/MODIFICACIOHOSTPROHIBIT* 40
3.30 INTERFÍCIE DE LA CLASSE FRONTERA *P_CONSULTAHOSTPROHIBIT* 40
3.31 INTERFÍCIE DE LA CLASSE FRONTERA *P_ALTA/MODIFICACIOETIQUETAPIC* 41
3.32 INTERFÍCIE DE LA CLASSE FRONTERA *P_ALTA/MODIFICACIOPARAULACLAU* 41
3.33 INTERFÍCIE DE LA CLASSE FRONTERA *P_CONFIGURARPROXY* 41
4.1 PROGRAMACIÓ PER CAPES. FUNCIONS DE LES CAPES 42
4.2 CLASSES GESTORES PRIMÀRIES DEL MODEL DE BASE DE DADES 46
4.3 CAPA DE DADES. CLASSES GESTORES 46
4.4 CLASSES PER AL CONTROL DEL FLUX D'INFORMACIÓ 47
4.5 CLASSES INTERPRETACIÓ PROTOCOLS HTTP I HTML 47
4.6 CAPA DE LA LÒGICA DE NEGOCI. CLASSES 49
4.7 COMPONENTS DE LA INTERFÍCIE GRÀFICA 51
4.8 DISSENY PANTALLA LOGIN.HTM 52
4.9 DISSENY PANTALLA MAIN.HTM 52
4.10 DISSENY INTERFÍCIE GENERADA PER DATAQUERY() 52
4.11 DISSENY INTERFÍCIE GENERADA PER DATAFORM() 52
4.12 DISSENY DELS SUBSISTEMES I LES SEVES DEPENDÈNCIES 53
5.1 ARBRE DE L'ESTRUCTURA DELS COMPONENTS DEL PROTOTIPUS 58

Capítol 1 - Introducció

1.1. Justificació i context.

Internet ha revolucionat el món de les comunicacions i de la informació, modificant la manera com l'ésser humà es comunica, aprèn o dirigeix els negocis, al proporcionar accés immediat a tot tipus d'informació mitjançant milions de recursos web (pàgines, documents, enciclopèdies, imatges, vídeos, ...). No obstant això, hi ha recursos que, pel seu contingut, no han de ser accessibles a tots els usuaris i, en particular, als nens.
Pares i professors han de poder decidir quins recursos permeten visualitzar o no als seus fills o alumnes, a casa o a l'escola, evitant l'accés a continguts violents, terroristes o pornogràfics. Les biblioteques, el cibercafés, les empreses, ... també volen limitar l'accés a Internet per a garantir la seguretat dels seus sistemes i, en el cas de les últimes, evitar la distracció dels seus empleats.

Cal, doncs, poder filtrar els recursos disponibles a Internet segons el seu tipus de contingut. Aquesta tasca, aparentment senzilla, comporta una complicació tècnica important degut a la dificultat de distingir si un determinat recurs conté informació considerada il·legal (obscenitat, pornografia infantil, apologia del terrorisme i de la violència, racisme, informacions falses, amenaces, ideologia i simbologia nazi, ...) o legal, però, no recomanada per a menors.
En aquest sentit, la Unió Europea, mitjançant la DECISIÓ Nº 276/1997CE DEL PARLAMENT EUROPEU I DEL CONSELL D'EUROPA [1], de data 25 de gener de 1999, va decidir animar al sector a oferir mètodes de filtrat i sistemes de classificació.
Un bon exemple n'és el sistema de classificació i etiquetat de continguts *PICS (Platform for Internet Content Selection)*, desenvolupat pel *Consorci World Wide Web (W3C)*, organisme encarregat d'establir els estàndards de la Web.

Les eines de filtrat de continguts permeten l'accés i visualització d'un recurs si es compleixen determinades polítiques, aplicant diverses tècniques de selecció establertes en base a llistes de dominis d'Internet [2] prohibits i permesos, classificació de pàgines mitjançant etiquetes *PICS* i cerca de paraules clau. Tot i així, les eines de filtrat actuals, tant gratuïtes com de pagament, no garanteixen del tot que els usuaris no tinguin accés a recursos no recomanats i, al mateix temps, que disposin de permís per a visualitzar-ne d'altres que no haurien d'estar bloquejats.
Com a complement i per a millorar la seva eficàcia, moltes de les eines de filtrat permeten fer el seguiment dels recursos sol·licitats pels usuaris, proporcionant la possibilitat de generar informes per a fer-ne un anàlisis i, en conseqüència, ajustar els mecanismes de filtratge.

Aquest projecte neix de la inquietud que aquesta temàtica provoca en l'autor, tant en l'àmbit professional com en el particular, i de la possibilitat d'utilitzar el llenguatge PHP per a implementar una solució multiplataforma, viable i eficient.
El projecte consisteix en dissenyar un *servidor proxy web* [3], anomenat normalment *proxy*, especificant i implementant dues de les moltes tasques que pot dur a terme i que són objectiu d'aquest treball: el filtrat de continguts web i el seguiment o monitorització, a través d'un registre històric, de tots els recursos sol·licitats, s'hagin denegat o no.
El treball s'ha enfocat aplicant un desenvolupament iteratiu i incremental basat en l'anomenat *Rational Unified Process* i amb la finalitat de facilitar la incorporació de més funcionalitats en futures versions.

[1] Disponible a **http://ec.europa.eu/consumers/cons_int/e-commerce/e_comm03_es.pdf**
[2] Un domini d'Internet és un nom d'equip que identifica el servidor on està situat un recurs.
[3] En resum, un servidor proxy web intercepta les peticions de recursos web efectuades pels usuaris, per diversos motius: seguretat, filtrat, rendiment, anonimat, ...

1.2. Objectius.

Com s'esmenta a l'apartat anterior, l'objectiu principal del projecte és el disseny d'un *proxy* que té com a funció **filtrar el recursos web** sol·licitats pels usuaris i **registrar-los en un historial**, indicant si s'han permès o no. Una característica important del disseny és que ha de permetre incorporar noves funcionalitats al proxy, per exemple, catxé de recursos, filtrat de correus electrònics, xats, FTP, P2P, ...).
A més, com a objectiu secundari, l'autor pretén **implementar el disseny en un prototipus**.

Assolir aquests objectius en comporta d'altres més concrets:

- En general, especificar la configuració de l'entorn per a garantir el correcte funcionament de l'aplicació i, així, evitar que els usuaris puguin accedir a recursos no autoritzats, i dotar a l'aplicació de mesures de seguretat, per tal d'evitar la seva manipulació i ús incorrecte, restringint-ne l'accés als usuaris autoritzats.
- Disposar dels coneixements i informació necessària sobre el funcionament del protocols de la xarxa Internet.
- En relació al filtrat, disposar de les dades i informació necessàries per a dur-lo a terme amb eficiència.
- Referent a l'historial guardar la informació adequada dels recursos sol·licitats.

1.3. Enfocament i mètode.

El desenvolupament del projecte es fonamenta en **tres àrees de treball**:

- El **control del flux d'informació** establert entre el navegador de l'usuari i Internet, és a dir, la intercepció de les peticions i respostes basades en el protocol *HTTP*.
- El **disseny del model de base de dades** que doni suport a les tasques de gestió i manteniment del filtratge i de la generació de l'historial.
- I, el **disseny de la interfície i la lògica de negoci** que permeti la interacció amb el programari i la consecució dels objectius.

Per a representar dins del programari els diversos elements que intervenen en el projecte i afavorir la seva reutilització, s'utilitza el paradigma de **programació orientat a objectes** i, per a la seva especificació, el llenguatge de modelat ***UML***.

El disseny es basa en el **model client-servidor** i en la **programació per capes**, amb la finalitat de separar la interfície d'usuari, la lògica de negoci i les dades i, així, facilitar la seva implementació en diferents llenguatges (PHP, C++, Java, Visual Basic, ...).

1.4. Planificació del projecte.

El projecte es divideix en 5 tasques: *la recollida i documentació de requisits*, *l'anàlisi i el disseny*, *la implementació i el testing*, *la revisió i lliurament de la documentació i del producte* i *la presentació resum del treball*.

En paral·lel al desenvolupament de les diverses tasques, s'elabora la documentació del projecte, és a dir aquesta memòria.

A continuació, es detallen les diverses tasques del projecte i quines fites s'han d'haver assolit al final de cadascuna d'elles.

Tasca 1	**Recollida i documentació de requisits.**
Durada / Dates	2 setmanes (de l'1 d'octubre al 14 d'octubre).
Descripció	Recollida i classificació d'informació rellevant per a la realització del projecte: • Informació sobre aplicacions de filtres de continguts web existents. • Documentació sobre tècniques de desenvolupament de programari, estructura de la informació i gestió de bases de dades. • Informació referent al protocols TCP/IP i HTTP. • Valoració de les eines de programació i de gestió de bases de dades adients per a la realització del projecte.
Objectius	• Disposar d'una visió general de les aplicacions de filtrat de continguts web existents avui en dia. • Revisar els conceptes d'anàlisi, disseny, implementació, testing i elaboració de documentació. • Aprofundir en els coneixements dels protocols TCP/IP i HTTP. • Seleccionar les eines de programació i de gestió de bases de dades per a la implementació del projecte. • Determinar l'arquitectura (client o client-servidor), el paradigma de programació i el llenguatge d'especificació o modelat utilitzats en el projecte. • Determinar la seva utilitat i usabilitat.
Fites	• Realitzar document descrivint les aplicacions de filtrat de continguts web actuals. • Confeccionar document descriptiu ♦ dels protocols que "intercepta" l'aplicació. ♦ de les eines seleccionades per a realitzar la programació i gestió de les dades de l'aplicació. ♦ de l'arquitectura, del paradigma de programació i del llenguatge de modelat escollit per a realitzar el projecte. ♦ de la utilitat i usabilitat de l'aplicació. • Confeccionar llistat de la bibliografia consultada en aquesta tasca.

Tasca 2	**Anàlisi i Disseny.**
Durada / Dates	5 setmanes (del 15 d'octubre al 19 de novembre).
Descripció	Anàlisi i disseny de l'aplicació.
Objectius	• Determinar els requisits i funcionalitats de l'aplicació. • Realitzar l'anàlisi i disseny del programari a implementar, en funció del requisits i funcionalitats determinades en el punt anterior.
Fites	• Obtenir l'anàlisi i disseny de l'aplicació, documentant les decisions preses.

Tasca 3	**Implementació i Testing.**
Durada / Dates	6 setmanes (del 20 de novembre al 31 de desembre).
Descripció	Implementació de l'aplicació i realització del joc de proves.
Objectius	• Implementar l'aplicació d'acord amb el disseny obtingut. • Crear el joc de proves per assegurar el correcte funcionament de l'aplicació i per a garantir la seva finalitat.
Fites	• Crear l'aplicació de filtrat de continguts web. • Confeccionar document descriptiu de la implementació (decisions, tècniques utilitzades, ...). • Confeccionar document descriptiu de les proves efectuades (*testing*).

Tasca 4	Revisió i lliurament de la documentació i del producte.
Durada / Dates	2 setmanes (de l'1 de gener al 13 de gener).
Descripció	Documentar la instal·lació de l'aplicació final i revisar la documentació del projecte (memòria).
Objectius	• Documentar la instal·lació i diverses opcions de configuració que pugui tenir l'aplicació. • Acabar i revisar la memòria del projecte.
Fites	• Confeccionar document descriptiu de la instal·lació i configuració de l'aplicació. • Redacció de conclusions. • Revisar i lliurar la memòria i el producte final.

Tasca 5	Presentació resum del treball.
Durada / Dates	1 setmana (del 14 de gener al 19 de gener).
Descripció	Realitzar presentació resum del treball, com a procés de síntesi del TFC.
Objectius	Sintetitzar de forma clara i concisa el treball realitzat i els objectius obtinguts.
Fites	Crear la presentació virtual (*PowerPoint*).

En resum, el calendari del treball és el següent.

		Dates	Activitat	Esdeveniment
SETMANA	1	20 - 24 set.	Tria TFC.	[24/09] Data límit per triar TFC.
	2	25 – 30 set.	Elaboració Pla de Treball.	[30/09] Lliurament Pla de Treball (PAC1).
	3 - 4	1 – 14 oct.	Recollida i documentació de requisits.	[07/10] Trobada Presencial Inici. [14/10] Informe quinzenal progrés.
	5 - 9	15 oct. – 19 nov.	Anàlisi i Disseny.	[28/10] Informe quinzenal progrés. [06/11] Lliurament PAC2.
	10 - 15	20 nov. – 31 des.	Implementació i Testing.	[25/11] Informe quinzenal progrés. [09/12] Informe quinzenal progrés. [18/12] Lliurament PAC3.
	16 - 17	1 – 13 gen.	Revisió i lliurament de la documentació i del producte.	[06/01] Informe quinzenal progrés. [13/01] Lliurament final (producte+memòria).
	18	14 – 19 gen.	Presentació resum del treball.	[19/01] Lliurament presentació.

Figura 1.1 Calendari TFC

1.5. Desviació en la planificació del projecte.

Durant la fase d'implementació l'autor ha dedicat més hores del previst al desenvolupament del programari, per tal que la planificació inicial no en quedés afectada.

1.6. Producte obtingut.

Dos són els productes resultants del projecte. El propi disseny descrit en aquesta memòria i el prototipus implementat seguint aquest mateix disseny.

Pensant en els possibles destinataris (escoles, biblioteques, empreses, particulars, ...), el prototipus s'ha desenvolupat amb **programari lliure i multiplataforma**. En concret, l'autor ha escollit el **llenguatge *PHP***, pel repte afegit d'implementar el *proxy* utilitzant les funcions de *sockets* d'aquest llenguatge, encara en fase experimental, i explorar les seves possibilitats.
Per a la gestió de les dades s'ha triat el motor de base de dades ***SQLite***, també programari lliure i multiplataforma.
En aquesta mateixa línia, s'ha decidit que la interfície d'usuari sigui web, per a facilitar la interacció amb l'aplicació des de qualsevol navegador.

Per al funcionament del prototipus s'aconsella disposar del producte *PHP*, versió 5 o superior, ja que inclou el motor de base de dades *SQLite*. En versions anteriors de PHP, el producte SQLite s'ha d'instal·lar per separat. En el document annex d'instal·lació del prototipus, s'indiquen els requeriments i configuració d'aquests paquets de programari.

1.7. Descripció dels capítols.

La memòria disposa de diversos capítols que reflecteixen les tasques efectuades durant el desenvolupament del projecte (la recollida i documentació de requisits, l'anàlisi de requisits, el disseny, la implementació i el testing).
S'inclouen, també, un apartat de conclusions finals, el glossari i la bibliografia.
Finalment, s'adjunten diferents annexes.

Tot seguit, es mostra una breu descripció dels capítols:

Capítol 2. Recollida i documentació de requisits. Es detallen els requeriments del projecte i es determina l'abast, justificant les decisions preses.

Capítol 3. Anàlisi de requeriments. Es descriu l'anàlisi dels requisits assenyalats en el capítol anterior amb tècniques orientades a objecte, utilitzant el llenguatge d'especificació *UML*. Es revisen els casos d'ús i s'especifiquen de manera formal, s'identifiquen les classes d'anàlisi i les seves relacions i s'analitza la interfície d'usuari.

Capítol 4. Disseny. Es descriuen els paradigmes en que es basarà el disseny (model client-servidor, programació orientada a l'objecte i per capes), el llenguatge de programació amb que s'implementarà i, per tant, els ajustaments o modificacions que cal fer sobre l'anàlisi previ i es defineix el model de base de dades, el disseny dels casos d'ús i de la interfície d'usuari.

Capítol 5. Implementació Testing. Es descriuen les fases seguides en el desenvolupament del programari, l'estructura dels components del prototipus i les proves de funcionament efectuades.

Capítol 6. Conclusions. L'autor descriu les seves impressions en relació al projecte i als objectius assolits.

Capítol 2. Recollida i documentació de requisits

2.1. Fonts d'informació.

La primera tasca en el desenvolupament del projecte és la recollida d'informació. Aquesta és necessària per a aprofundir en la temàtica objecte d'aquest treball i determinar-ne els requisits i l'abast. Bàsicament, les fonts d'informació utilitzades han estat Internet i diversos materials acadèmics de la UOC, relacionats a l'apartat de bibliografia.
Dins la xarxa hi ha disponible una gran quantitat d'informació referent al Web, protocols usats a Internet, *proxies*, filtrat de continguts i gestors de base de dades. El coneixement d'aquestes matèries és fonamental per a aconseguir els objectius enumerats al capítol anterior.

S'acompanya el document ***El Web i protocols d'Internet*** on l'autor fa una aproximació al *World Wide Web* i als protocols que el sustenten: *HTTP*, *TCP* i *IP*.
Completa aquesta informació, una introducció als *proxies* i al *filtrat de continguts web*, fent menció d'alguns del programes de filtrat més populars.

La informació cercada a Internet, juntament, amb els materials didàctics de la UOC han permès a l'autor aprofundir en el coneixement de les diverses matèries i avançar en la consecució del projecte.

2.2. Informació inicial.

El projecte *Argos* té com a finalitat el disseny d'un *servidor proxy web*, que disposi de la capacitat de filtrar continguts web i de registrar-ne les peticions efectuades pels usuaris, hagin estat permeses o no.

L'aplicació consta de dos programes: el programa *Proxy*, com a tal, i el programa *gestorProxy*, mitjançant el qual es gestionen l'activitat i configuració del proxy, les dades i configuració de la tasca de filtrat, les consultes i estadístiques de l'historial i la seguretat de l'aplicació.

El programari resultant ha de funcionar tant en un entorn de xarxa com en un únic ordinador.
En ambdós casos, s'ha de configurar l'accés a Internet de manera que totes les peticions HTTP passin per força a través del programa *Proxy*.

2.3. El programa *Proxy*.

El programa *Proxy*, en endavant *Proxy*, ha d'estar funcionant ininterrompudament per tal de donar servei a les peticions de recursos web dels usuaris.

El *Proxy* està format per una **part servidora**, que és la que rep les peticions dels usuaris, i una **part client**, que és la que sol·licita les peticions rebudes a Internet, directament o a través d'un altre proxy. A més, disposa d'una **cua de peticions** on es van afegint les peticions pendents d'atendre i del **gestor de tasques**, encarregat de gestionar les diverses tasques que pot dur a terme el *Proxy*. El gestor de tasques té tants mòduls com tasques o serveis pot efectuar el *Proxy*. En el nostre cas, el **mòdul de filtrat** i el **mòdul de registre històric**..

Per tal de millorar el seu rendiment, el *Proxy* s'executa en un procés propi i no disposa d'interfície gràfica. Només s'indica en un missatge que està funcionant i quan es dóna un error recuperable. Tot i així, des del programa *gestorProxy* se'n pot seguir l'activitat.

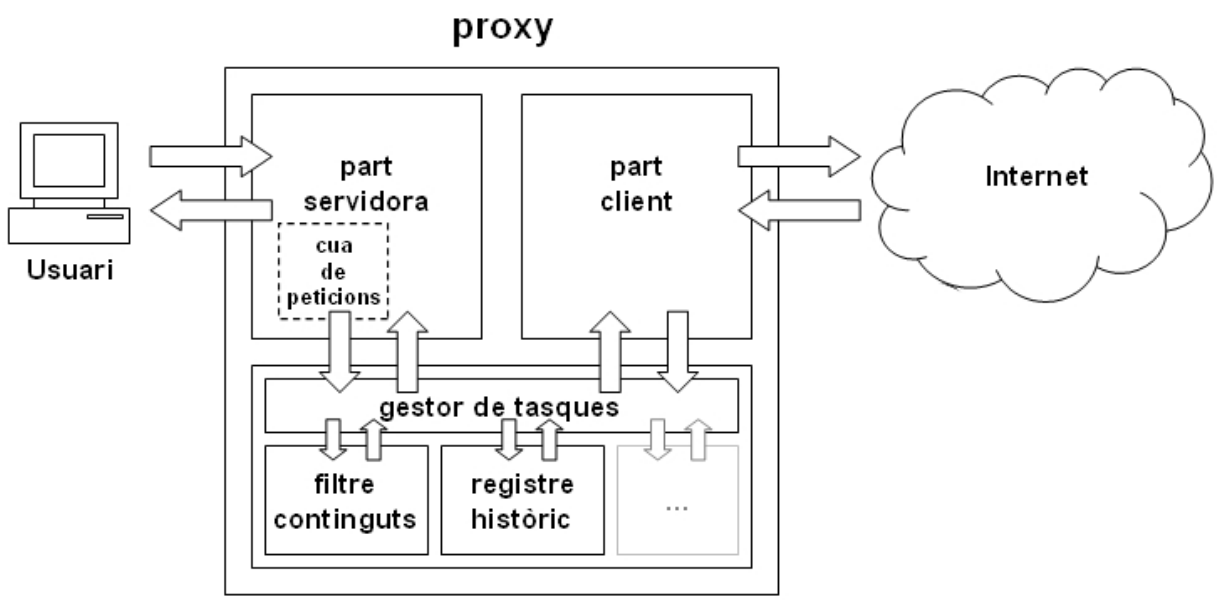


Figura 2.1 Esquema funcionament del *Proxy*

L'activitat del *Proxy* és la següent.

- El *Proxy* es posa en funcionament quedant la part servidora "escoltant", és a dir, esperant una petició d'un recurs web. Quan la part servidora rep una petició, la situa a la cua de peticions pendents.
- El gestor de tasques treu la petició pendent més antiga (la cua de peticions té una estructura de tipus *FIFO* [1]) per tal de processar-la i efectuar alguna acció abans de sol·licitar el recurs web a Internet.
  En el nostre cas, el gestor de tasques passa la petició al mòdul de filtrat. Aquest valida que el domini d'Internet del recurs no estigui prohibit. Si és així, genera una resposta informativa i la traspassa al gestor de tasques, el qual la fa arribar a la part servidora perquè la trameti a l'usuari. Si no està prohibida, es retorna la petició al gestor de tasques i, aquest, la fa seguir a la part client perquè la sol·liciti a Internet.
  Si el recurs sol·licitat és un fitxer (imatge, àudio, vídeo, ...) i està activada l'opció d'efectuar filtratge per tipus de fitxer, s'efectuarà l'acció corresponent i s'actuarà com en el cas de la validació del domini.
- Una vegada la part client ha demanat el recurs a Internet i en rep la resposta, la traspassa al gestor de tasques, per si cal fer alguna acció abans d'enviar-la a l'usuari.
  En el nostre cas, el gestor de tasques passa la petició al mòdul de filtrat. Aquest valida, si n'hi ha, l'etiqueta *PICS* del recurs web o comprova si el contingut és permissible, a través de la validació de paraules clau. Si el resultat de qualsevol de les dues validacions no permet fer arribar el recurs a l'usuari, es genera una resposta informativa, traslladant-la al gestor de tasques. En cas contrari, es trasllada la resposta d'Internet al gestor de tasques. Aquest fa seguir la resposta a la part servidora que, finalment, transmet la resposta (d'Internet o informativa) a l'usuari sol·licitant.
- En qualsevol cas, sigui permès o no el recurs sol·licitat, abans de traslladar la resposta a la part servidora, el gestor de tasques l'envia al mòdul de registre històric per a registrar la petició.

[1] Una cua es coneix també com una estructura de tipus FIFO (First-In, First-Out), és a dir, el primer element que entra a la cua es el primer element que surt de la cua.

2.3.1. Filtrat de continguts.

El filtrat de recursos s'efectua pel seu domini d'Internet, pel tipus de contingut del lloc web en base a etiquetes *PICS*, si s'indica, o pel propi contingut (paraules clau, tipus de fitxer, ...).

- L'aplicació gestiona dues llistes de dominis d'Internet. Una llista de dominis als que no s'hi permet l'accés pel seu tipus de contingut (**llista negra**) i una llista de dominis "de confiança" als que es pot accedir sense cap més comprovació (**llista blanca**).
  Exemples de domini d'Internet són *google.com*, *vilaweb.cat*, ...
- Si el recurs sol·licitat disposa d'etiqueta *PICS*, es filtra el recurs depenent del tipus de contingut especificat. En el cas que un recurs estigui etiquetat com a contingut no permès, el seu domini s'afegeix a la llista negra.
- Finalment, si el recurs demanat és una pàgina web, s'efectua el filtratge avaluant les paraules mitjançant un sistema de "pesos", del qual s'obté una puntuació (de 0 a 10, on 0 és permesa i 10 totalment denegada) que estableix si es permet o no. Cal tenir en compte que si el filtrat es basés en l'avaluació d'una paraula per si sola, es podria impedir l'accés a pàgines web amb informació que no tingués perquè ser prohibida. Si una pàgina web és denegada, s'afegeix a la llista negra, indicant-hi la puntuació i marcant-la per a ser revisada per l'administrador de l'aplicació.
  Si es tracta d'un fitxer, es pot activar o desactivar la seva descarrega depenent del tipus de fitxer (mp3, mpeg, avi, exe, zip, ...).

La gestió i manteniment de les llistes de dominis, d'etiquetes PICS i de paraules clau o tipus de fitxers s'explica a l'apartat *2.4.3 Gestió de l'eina de filtrat*.

2.3.2. Registre de peticions.

Com ja s'ha esmentat, es registren totes les peticions de recursos web efectuades pels usuaris, guardant-ne la següent informació: *URI* dels recurs sol·licitat, IP de la màquina sol·licitant, si s'ha permès o no el recurs i data/hora de la petició, mida en bytes del recurs, categoria del contingut si està identificat mitjançant etiqueta PICS o denegat).

2.3.3. Control d'usuaris.

El *Proxy* es pot configurar de manera que qualsevol usuari s'hagi d'identificar per a poder sol·licitar recursos a Internet. En aquest cas, quan l'usuari arranqui el navegador del seu ordinador, li apareixerà una pàgina web sol·licitant la seva identificació i paraula clau.
Si l'usuari està un cert temps (a configurar) sense fer cap petició, al realitzar-ne una de nova se li tornarà a demanar la identificació, per raons de seguretat.

Si el control d'usuaris està actiu, s'inclou l'identificador d'usuari al registrar la petició a l'historial.

2.4. El programa *GestorProxy*.

El programa *GestorProxy* permet gestionar l'activitat del *Proxy*, les dades i configuració del servei de filtrat de continguts (llistes blanca i negra, codis etiquetes PICS, paraules clau o tipus de fitxer), les consultes i estadístiques de l'historial i la seguretat de l'aplicació.

Només l'administrador de l'aplicació o personal autoritzat (gestors) tenen accés al programa. L'administrador disposa de tots els permisos i el gestor pot efectuar consultes i estadístiques de l'historial i gestionar els usuaris que puguin accedir a Internet.

La interfície d'usuari és web. D'aquesta manera s'hi pot accedir des de qualsevol ordinador de la xarxa on estigui instal·lat el programari, indicant la seva URI des d'un navegador.

2.4.1. Configuració del *Proxy*.

Aquesta opció permet configurar diversos paràmetres de funcionament del *Proxy*:

- L'adreça IP i el port on està "escoltant" el *Proxy*.
- En el cas de l'existència d'un *proxy* intermedi i, per tant, no disposant de connexió directa a Internet, l'adreça IP i el port del *proxy* intermedi.
- El nombre màxim de connexions simultànies que pot atendre el *Proxy*.
- La mida del buffer de lectura i escriptura del *Proxy*.
- Les tasques o serveis actius del *Proxy*, pe. catxé de recursos, filtrat de continguts, filtrat de correus electrònics, xats, FTP, P2P, registre històric, ...
- Si es requereix que tot usuari s'hagi d'identificar per accedir a Internet.

2.4.2. Seguiment de l'activitat del *Proxy*.

Aquesta opció permet seguir l'activitat del *Proxy*:

- Petició actual.
- Cua de peticions pendents.
- Errors recuperables que s'han produït.

2.4.3. Gestió de l'eina de filtrat.

El filtrat de continguts es basa en llistes de dominis, codificació d'etiquetes PICS, paraules clau i tipus de fitxer. Totes aquestes dades estan guardades en taules dins la base de dades.

La informació de filtrat està classificada per categories, pe., pornografia, nuesa, violència, terrorisme, racisme, xenofòbia, drogues, apostes, armes, ...
Es permet configurar les categories de manera que només siguin filtrades aquelles que es desitgi.

Les llistes de dominis.

La llista negra, és a dir, dels dominis no permesos, disposa de la següent informació: categoria de contingut, nom del domini i puntuació atorgada al domini. A més, s'indica si el domini en qüestió està pendent de revisió per part de l'administrador de l'aplicació, quan el domini és afegit automàticament a la llista pel *Proxy*, en base a les polítiques de filtrat.
La llista blanca o de dominis "de confiança" només disposa del nom de domini.
Ambdues llistes poden ser gestionades per l'administrador per afegir, modificar o esborrar dominis, manualment.

La llista d'etiquetes PICS.

La informació de la categorització i codificació de l'etiquetatge *PICS* està ja situat a la base de dades i es basa en els codis *ICRA* que es troben a l'adreça http://www.icra.org/decode/ i http://www.icra.org/vocabulary/mapping/.
La taula conté la categoria del contingut, el codi *ICRA* i la descripció explicativa del codi.
Es permet que l'administrador pugui afegir nous codis, degut a noves revisions d'aquesta codificació.

La llista de paraules clau.

La llista de paraules clau disposa de la següent informació: categoria de contingut i paraula o frase clau.
La llista pot ser gestionada per l'administrador per afegir, modificar o esborrar paraules o frases clau, manualment.

Tipus de fitxer.

Es permet especificar quins tipus de fitxer, en base a la seva extensió, ha de ser permès o no, pe., exe, zip, rar, avi, mpeg, mp3, jpg, jpeg, gif, png, ...
La llista de tipus de fitxer disposa de la següent informació: extensió i si està permès o no.
La llista també pot ser gestionada per l'administrador per afegir, modificar o esborrar tipus de fitxer i permetre la seva descarrega o no.

2.4.4. Gestió de l'historial.

Aquesta opció permet consultar i fer estadístiques de les peticions de recursos web que han efectuat els usuaris.
Així es pot obtenir les vegades que un recurs ha estat sol·licitat, quin tipus de recursos s'han denegat més, quina és l'hora o espai de temps en què s'efectuen més peticions, temps de connexió, ...

2.4.5. Gestió d'usuaris.

Aquesta opció permet donar d'alta, modificar o esborrar els usuaris que interactuen amb el programari. Es defineixen tres tipus d'usuaris: l'administrador, el gestor i l'usuari autoritzat.

L'administrador disposa de tots el permisos i, per tant, pot gestionar tots els tipus d'usuaris.

El gestor només pot gestionar els usuaris autoritzats a accedir a Internet, en el cas de que es requereixi que tot usuari que vulgui sol·licitar recursos web s'hagi d'identificar i, així, augmentar la seguretat.

Per cada usuari es guardarà el seu identificador, paraula clau, tipus d'usuari, nom, codi (que potser el NIF, número d'empleat, número de soci, número d'alumne, ... depenent de l'entitat on estigui instal·lat el programari), àrea (que pot ser el curs de l'alumne, el departament o secció on treballa l'empleat, ...) i si està suspès o no.

2.5. El context del programari.

2.5.1. El model del domini.

En aquest apartat es descriu l'entorn del programari, identificant en una primera aproximació els objectes o classes més importants.

De la lectura de la informació inicial es conclou que l'entorn físic on estarà situat el programari pot ser un sol ordinador o una xarxa situada en una escola, cibercafé, biblioteca o empresa.

En una primera exploració s'identifiquen els objectes o classes següents:

- Objectes del món exterior: usuari, usuari d'Internet, gestor, administrador, proxy, servei, filtre i historial.
- Objectes del domini: inici de sessió (identificació), petició d'un recurs.

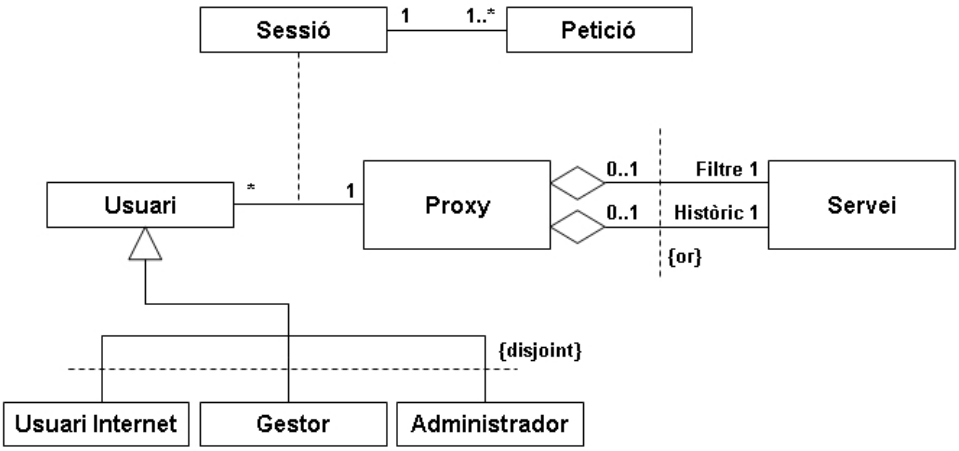


Figura 2.2 Diagrama de classes del model del domini

La figura 2.4 mostra les classes més importants amb les seves relacions.
L'entorn del programari queda acotat: Es tracta d'un proxy que pot efectuar diversos serveis. En aquest cas, disposa d'un filtre i d'un historial. S'hi poden connectar diversos tipus d'usuaris (usuari d'Internet, gestor o administrador) que poden iniciar sessions per tal de sol·licitar peticions. El fet d'establir sessió implica haver efectuat una petició, per exemple, la pàgina d'inici del navegador.

El gestor i l'administrador poden interactuar amb el programa *gestorProxy* a través del proxy realitzant peticions de les pàgines web del programa.

La descripció de les classes està situada a l'apartat *2.5.3 El glossari del model de negoci*.

2.5.2. El model de negoci.

En aquest apartat s'aprofundeix en el context del programari i se'n descriuen a grans trets els processos i entitats principals. Així, doncs, es descriuen en línies generals els casos d'ús, s'identifiquen les entitats que hi participen i, per últim, s'expliquen amb més detall els casos d'ús descrits, sempre des d'un punt de vista organitzatiu.

En primer lloc, s'obté una relació de les necessitats dels diferents usuaris del programari.

Usuaris	Què necessiten?
Usuari Internet	• Iniciar sessió. • Fer petició.
Gestor	• Iniciar sessió. • Fer petició. • Gestionar usuaris d'Internet. • Consultar historial.
Administrador	• Iniciar sessió. • Fer petició. • Gestionar tots els usuaris. • Consultar historial. • Gestionar filtre. • Gestionar proxy.

Figura 2.3 Relació dels usuaris del programari i del què necessiten

De la relació anterior se'n deriva el diagrama de casos d'ús del model de negoci.

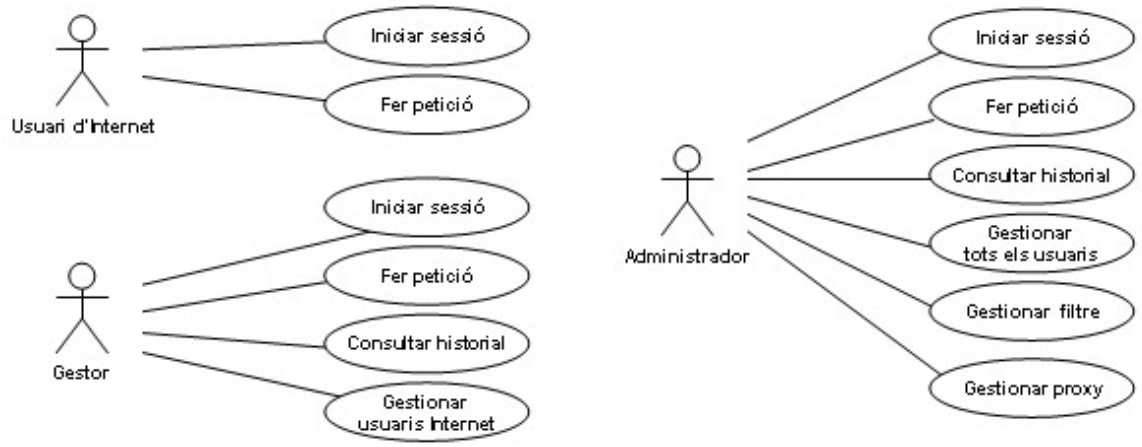


Figura 2.4 Diagrama de casos d'ús del model de negoci

Per mitjà dels diagrames de col·laboració s'identifiquen els objectes que fan servir els casos d'ús anteriors:

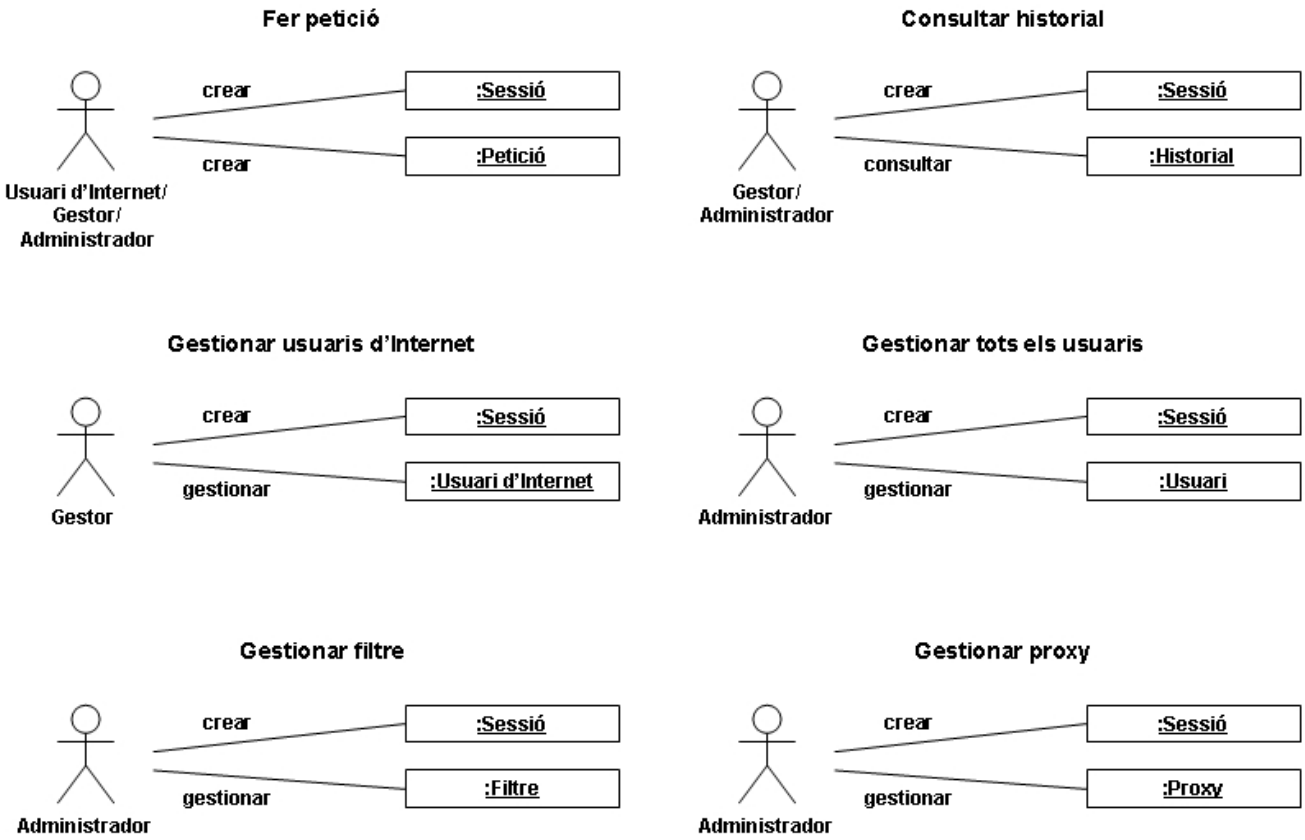


Figura 2.5 Diagrama de col·laboració del model de negoci

2.5.3. El glossari del model de negoci.

A continuació es relaciona el glossari del model de negoci:

Administrador: usuari que pot efectuar peticions i que disposa de tots els permisos per accedir al programari.

Filtre: servei del proxy que efectua el filtrat de continguts.

Gestor: usuari que pot efectuar peticions i que pot gestionar l'historial i els usuaris d'Internet.

Historial: servei del proxy que registra les peticions de recursos efectuades pels usuaris.

Petició: sol·licitud d'un recurs web.

Proxy: dispositiu amb el que els usuaris estableixen sessions i mitjançant el qual efectuen peticions de recursos web i en reben la resposta.

Servei: tasca que pot desenvolupar el proxy.

Sessió: cadascuna de les connexions d'un usuari amb el proxy.

Usuari: persona que pot establir sessions amb el proxy i fer peticions de recursos web.

Usuari d'Internet: usuari que pot efectuar peticions, però que no pot gestionar cap dada relacionada amb el programari.

2.6. Els guions.

Es detecten tres guions:

Guió de l'usuari d'Internet:

L'usuari vol accedir a Internet des del seu ordinador. Engega el navegador i se li presenta la pàgina inicial sol·licitant que s'identifiqui. Introdueix la seva identificació i contrasenya i el programari el valida, responent amb una pàgina de salutació. L'usuari ha iniciat una sessió i continua fent peticions de recursos web.

Guió del gestor:

El gestor ha de gestionar dades d'un usuari d'Internet. Engega el navegador i se li presenta la pàgina inicial sol·licitant que s'identifiqui. Introdueix la seva identificació i contrasenya i el programari el valida, responent amb una pàgina on disposa d'un menú amb les dues accions que pot efectuar (gestionar usuari d'Internet i consultar l'historial). El gestor ha iniciat una sessió i tria gestionar usuaris d'Internet.

Ara el gestor desitja consultar l'historial. Repeteix la mateixa seqüència. Si ja està en sessió, no se li demana la identificació. Del menú inicial, tria consultar historial.

Si el gestor vol efectuar peticions actua com l'usuari d'Internet.

Guió de l'administrador:

L'administrador ha de gestionar dades d'usuaris. Engega el navegador i se li presenta la pàgina inicial sol·licitant que s'identifiqui. Introdueix la seva identificació i contrasenya i el programari el valida, responent amb una pàgina on disposa d'un menú amb les quatre accions que pot efectuar (gestionar usuaris, gestionar el filtre, gestionar el proxy i consultar l'historial). El gestor ha iniciat una sessió i tria gestionar usuaris.

Ara l'administrador desitja gestionar el filtre. Repeteix la mateixa seqüència. Si ja està en sessió, no se li demana la identificació. Del menú inicial, tria gestionar el filtre.

Aquesta vegada l'administrador desitja gestionar el proxy. Repeteix la mateixa seqüència. Si ja està en sessió, no se li demana la identificació. Del menú inicial, tria gestionar el proxy.

Si l'administrador vol consultar l'historial o efectuar peticions actua com el gestor.

2.7. Actors.

Els actors són tres: **l'usuari d'Internet**, **el gestor** i **l'administrador**.

Existeix una relació d'especialització entre tots ells.

- L'**usuari d'Internet** és *usuari final* i té dos papers: iniciar sessió i fer peticions.
- El **gestor** és *usuari final* i té quatre papers: iniciar sessió, fer peticions, consultar l'historial i gestionar els usuaris d'Internet.
  El gestor és una especialització de l'usuari d'Internet perquè pot fer les mateixes tasques que ell i dues més (gestionar els usuaris d'Internet i consultar l'historial).
- L'**administrador** és *usuari privilegiat* o *gestor del sistema*, ja que pot configurar el *proxy* i té sis papers: iniciar sessió, fer peticions, consultar l'historial, gestionar tots els usuaris, gestionar el filtre i gestionar el proxy.
  L'administrador és una especialització del gestor perquè pot fer les mateixes tasques que ell (gestionar tots els usuaris inclou gestionar usuaris d'Internet) i vàries d'exclusives (gestionar el filtre, gestionar el proxy i gestionar tots els usuaris).

2.8. Casos d'ús.

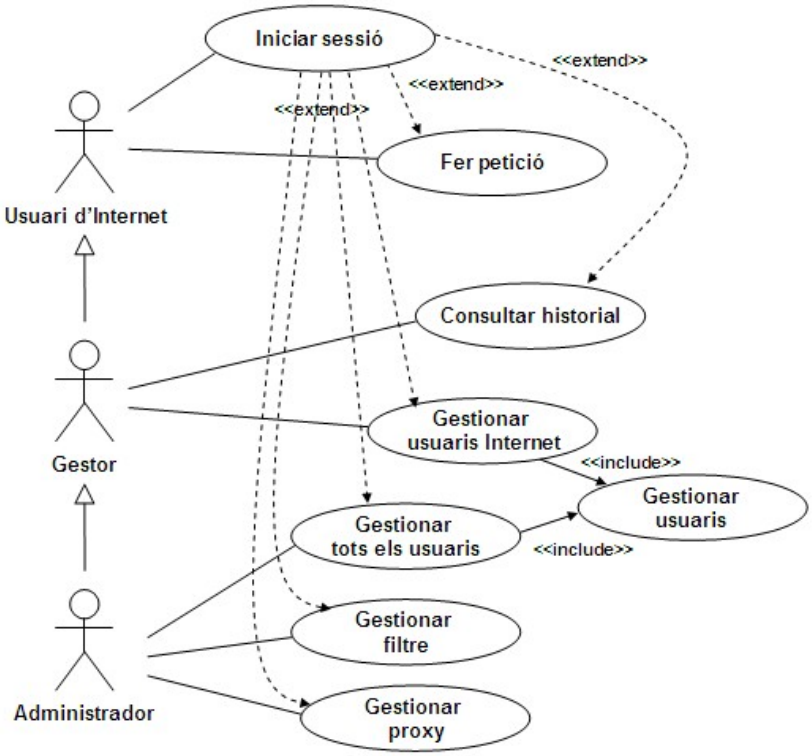


Figura 2.6 Diagrama de casos d'ús

Els casos d'ús s'obtenen dels guions. El diagrama mostra les relacions entre els actors i els casos d'ús.

A continuació es relacionen els casos d'ús, descrivint-los textualment.

Cas d'ús número 1: "Iniciar sessió"

Resum de la funcionalitat: Inicia una *sessió* de connexió amb un **usuari**.

Paper dins el treball de l'actor: Una de les tasques principals de tots els **usuaris**.

Actors: Usuari d'Internet, **gestor** i **administrador**.

Casos d'ús relacionats: Fer petició, Consultar historial, Gestionar usuaris Internet, Gestionar tots els usuaris, Gestionar filtre, Gestionar proxy.

Precondició: L'**usuari** ha d'estar a la base de dades i no ha d'estar suspès. Per a iniciar la *sessió*, cal que no estigui iniciada.

Postcondició: S'ha iniciat la *sessió*. L'**usuari** ja pot efectuar *peticions*. Si no es pot iniciar la *sessió*, s'indica amb un missatge informatiu.

Descripció: Tot **usuari** ha d'iniciar una *sessió* per a poder efectuar *peticions* de recursos web. Quan un **usuari** engega el navegador, se li mostra la pàgina inicial del navegador que sol·licita la identificació i la contrasenya de l'**usuari** per a poder validar-lo. Si la pàgina inicial és una altra o bé s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Observacions: No existeix una acció o cas d'ús específic per a realitzar el tancament de la *sessió*. És cancel·la per *timeout* quan fa un temps determinat que l'**usuari** no efectua *peticions*.

Cas d'ús número 2: "Fer petició"

Resum de la funcionalitat: L'**usuari** efectua una *petició* d'un recurs web.

Paper dins el treball de l'actor: Una de les tasques principals de tots els **usuaris**.

Actors: Usuari d'Internet, **gestor** i **administrador**.

Casos d'ús relacionats: Iniciar sessió.

Precondició: L'**usuari** ha iniciat una *sessió* i no està *suspès*.

Postcondició: L'**usuari** rep la resposta a la *petició* del recurs web sol·licitat.

Descripció: L'**usuari** sol·licita un recurs web, teclejant la seva URI a través del navegador o bé fent clic damunt un enllaç de la pàgina que té en pantalla. Una vegada feta la *petició*, rep la resposta corresponent.

Observacions: S'actualitza la *data i hora* de l'última *petició*, per tal d'evitar el tancament de la *sessió*. Si al fer una sol·licitud, s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 3: "Consultar historial"

Resum de la funcionalitat: Es fa una consulta a l'*historial*.

Paper dins el treball de l'actor: Una de les tasques principals del **gestor** i una de les tasques esporàdiques de l'**administrador**.

Actors: Gestor i **administrador**.

Casos d'ús relacionats: Iniciar sessió.

Precondició: El **gestor** o l'**administrador** han iniciat una *sessió*.

Postcondició: El **gestor** o l'**administrador** han consultat l'*historial*.

Descripció: El **gestor** o l'**administrador** trien l'opció de menú "Consultar *historial*". Una vegada tenen a disposició les consultes programades, en trien una. Tot seguit, reben la informació de la consulta.

Observacions: Si al fer una consulta, s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 4: "Gestionar usuaris d'Internet"

Resum de la funcionalitat: Es realitza l'alta, modificació, consulta o baixa d'un *usuari d'Internet*.

Paper dins el treball de l'actor: El fa servir ocasionalment el **gestor**. Durant el període recent després de la instal·lació del programari forma part de la tasca principal de treball del **gestor**.

Actors: Gestor.

Casos d'ús relacionats: Iniciar sessió, Gestionar usuaris.

Precondició: El **gestor** ha iniciat una *sessió*. Si s'efectua una alta, l'*usuari* no està a la base de dades (BD). Si s'efectua una modificació, consulta o baixa, l'*usuari* ha d'existir a la BD.

Postcondició: En el cas d'una alta, *l'usuari d'Internet* està a la BD. Si s'ha efectuat una modificació, han canviat les dades de l'*usuari d'Internet* a la BD. Si s'ha donat de baixa, l'*usuari d'Internet* ja no existeix a la BD. Si s'ha efectuat una consulta, s'han consultat les dades de l'*usuari d'Internet*.

Descripció: El **gestor** tria l'opció de menú "Gestionar *usuaris d'Internet*". Una vegada té a disposició les opcions d'alta, modificació, consulta o baixa, en tria una. En qualsevol cas, el **gestor** introdueix les dades requerides per a realitzar l'acció corresponent. En el cas d'efectuar una baixa, el **gestor** rep un missatge informatiu de confirmació de la baixa en el que té dues opcions: confirmar o cancel·lar. Si es confirma, es dóna de baixa l'*usuari d'Internet*. Si es cancel·la tot queda igual. Si es tracta d'una consulta, tot seguit, es rep la informació.

Observacions: Si al fer qualsevol gestió (alta, modificació, consulta o baixa), s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 5: "Gestionar tots els usuaris"

Resum de la funcionalitat: Es realitza l'alta, modificació, consulta o baixa d'un *usuari*.

Paper dins el treball de l'actor: El fa servir ocasionalment l'**administrador**. Durant el període recent després de la instal·lació del programari forma part de la tasca principal de treball de l'**administrador**.

Actors: Administrador.

Casos d'ús relacionats: Iniciar sessió, Gestionar usuaris.

Precondició: L'**administrador** ha iniciat una *sessió*. Si s'efectua una alta, l'*usuari* no està a la BD. Si s'efectua una modificació, consulta o baixa, l'*usuari* ha d'existir a la BD.

Postcondició: En el cas d'una alta, *l'usuari* està a la BD. Si s'ha efectuat una modificació, han canviat les dades de l'*usuari* a la BD. Si s'ha donat de baixa, l'*usuari* ja no existeix a la BD. Si s'ha efectuat una consulta, s'han consultat les dades de l'*usuari*.

Descripció: L'**administrador** tria l'opció de menú "Gestionar *usuaris*". Una vegada té a disposició les opcions d'alta, modificació, consulta o baixa, en tria una. En qualsevol cas, l'**administrador** introdueix les dades requerides per a realitzar l'acció corresponent. En el cas d'efectuar una baixa, l'**administrador** rep un missatge informatiu de confirmació de la baixa en el que té dues opcions: confirmar o cancel·lar. Si es confirma, es dóna de baixa l'*usuari*. Si es cancel·la tot queda igual. Si es tracta d'una consulta, tot seguit, es rep la informació.

Observacions: Si al fer qualsevol gestió (alta, modificació, consulta o baixa), s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 6: "Gestionar filtre"

Resum de la funcionalitat: Es realitza la gestió de les dades que condicionaran el filtrat dels continguts web.

Paper dins el treball de l'actor: Una de les tasques principals de l'**administrador**.

Actors: Administrador.

Casos d'ús relacionats: Iniciar sessió.

Precondició: L'**administrador** ha iniciat una *sessió*. Si s'efectua una alta, les dades no estan a la base de dades (BD). Si s'efectua una modificació, consulta o baixa, les dades han d'existir a la BD.

Postcondició: En el cas d'una alta, les dades estan a la BD. Si s'ha efectuat una modificació, han canviat les dades a la BD. Si s'han donat de baixa, ja no existeixen a la BD. Si s'ha efectuat una consulta, s'han consultat les dades.

Descripció: L'**administrador** tria l'opció de menú "Gestionar *filtre*". Una vegada té a disposició les opcions d'alta, modificació, consulta o baixa, en tria una. En qualsevol cas, l'**administrador** introdueix les dades requerides per a realitzar l'acció corresponent. En el cas d'efectuar una baixa, l'**administrador** rep un missatge informatiu de confirmació de la baixa en el que té dues opcions: confirmar o cancel·lar. Si es confirma, es donen de baixa les dades. Si es cancel·la tot queda igual. Si es tracta d'una consulta, tot seguit, es rep la informació.

Observacions: Si al fer qualsevol gestió (alta, modificació, consulta o baixa), s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 7: "Gestionar proxy"

Resum de la funcionalitat: S'efectua la configuració i seguiment del proxy.

Paper dins el treball de l'actor: Quan s'instal·la el programari, la configuració del *proxy* forma part de la part principal de treball de l'**administrador**. Més endavant, és un cas d'ús esporàdic. El seguiment del *proxy,* es usat ocasionalment per l'**administrador**.

Actors: Administrador.

Casos d'ús relacionats: Iniciar sessió.

Precondició: L'**administrador** ha iniciat una *sessió*.

Postcondició: Si s'han realitzat tasques de configuració, s'ha modificat la configuració del *proxy*. Si s'ha realitzat el seguiment, es coneix la situació actual del *proxy* (tasques que està realitzant, cua de peticions, ...).

Descripció: L'**administrador** tria l'opció de menú "Gestionar *proxy*". Una vegada té a disposició les opcions de configurar o fer el seguiment, en tria una. En qualsevol cas, l'**administrador** introdueix les dades requerides per a realitzar l'acció corresponent. Si es tracta del seguiment, es tria una opció i, tot seguit, es rep la informació.

Observacions: Si al fer qualsevol gestió, s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*. En el cas de modificar la configuració del *proxy*, depenent dels canvis efectuats pot ser necessari rearrancar el *proxy*.

2.9. Interfície d'usuari

A continuació, es descriuen els requisits que ha de complir la interfície d'usuari.

2.9.1. Perfils d'usuari.

L'*usuari d'Internet* només interactua amb el programari en el moment d'iniciar una sessió, quan es valida, i no té autoritzat l'accés a la resta de funcionalitats.

En general, els usuaris del programari (*gestor* i *administrador*) estan acostumats a utilitzar els ordinadors i alguns poden ser usuaris avançats. En el cas del particulars, el perfil pot ser molt divers, tot i així, el programari no presenta gran dificultat per a la seva instal·lació i gestió.
Es pot donar el cas que cap usuari hagi fet servir una aplicació semblant.

Tret de la tasca inicial de configuració, la resta de funcions del programari es faran servir de manera més continuada. Segurament el *gestor* les utilitzarà més sovint que l'*administrador*.

2.9.2. Descripció de les tasques.

Iniciar sessió.

L'usuari tecleja el seu identificador i la seva contrasenya. Una vegada comprovada la seva identitat, l'usuari rep una pantalla de benvinguda.

Fer petició.

L'usuari tecleja la URI al navegador. Si la URI és permesa rep la pàgina web corresponent. Si la URI no és permesa rep una pàgina web indicant que la petició no ha pogut ser atesa.

Consultar historial.

El gestor o l'administrador poden consultar tot l'historial o bé poden especificar què volen.

Es pot triar l'usuari, la IP de la màquina, la data de petició, el host i la categoria. També es pot especificar si es volen les peticions permeses, les denegades o ambdues.
Es pot donar un sol valor o diversos. Si es demana la consulta indicant l'identificador d'usuari, es mostren les peticiona que ha efectuat aquest usuari. Si s'indica la IP d'una màquina i es marca només peticions denegades, doncs, es mostren les peticions no permeses efectuades des de la màquina en concret, usuaris que han efectuat més peticions denegades, ...

Estadístiques historial.

El gestor o l'administrador poden disposar d'estadístiques triant la que volen: mitjana de temps de durada de les sessions, peticions per dia o per trams horaris, tipologia de les peticions denegades, tipus de categories prohibides més sol·licitades, ...

Afegir un usuari.

El gestor o l'administrador teclegen el nom de l'usuari, el seu codi (núm. empleat, alumne, ...), i, opcionalment, l'àrea a la qual pertany. En el cas de l'administrador, indica el tipus d'usuari.

Modificar un usuari.

El gestor o l'administrador teclegen l'identificador de l'usuari o bé el trien d'una llista on es mostra la resta de dades de l'usuari. Una vegada escollit l'usuari, modifiquen el nom, el codi (núm. empleat, alumne, ...), i/o l'àrea a la qual pertany. En el cas de l'administrador, pot modificar el tipus d'usuari.
Es pot marcar l'usuari com a suspès, evitant que pugui efectuar peticions.

Afegir un filtre.

- Host permès (llista blanca) L'administrador tecleja el nom del host (www.google.com).
- Host no permès (llista negra). L'administrador tecleja el nom del host i tria la categoria a la que pertany.
- Etiqueta PIC. L'administrador tecleja el tipus, el codi, la descripció (opcional) i tria la categoria.
- Paraula clau. L'administrador tecleja la paraula clau i en tria la categoria. A més, indica el "pes" que té.

Configurar proxy.

L'administrador indica si cal o no identificació d'usuari per accedir a Internet i, si és així, tria com s'assigna l'identificador (igual al codi afegint-hi una "U" davant o generat pel sistema de manera seqüencial), si el filtre està actiu, si es guarda historial o no de les peticions, quins tipus d'arxius són permesos o no (mp3, mpeg, ...), quin és el temps de *timeout* de les sessions i cada quans segons s'executarà el sistema automàtic de tancament de sessions.

2.9.3. Requisits d'usabilitat.

Els *gestors* han de ser capaços d'utilitzar el programari després d'una explicació i una pràctica d'una hora. L'*administrador* l'ha de saber utilitzar després d'una explicació i una pràctica de dues hores.

Capítol 3. Anàlisi de requeriments

L'anàlisi permet obtenir una visió general del programari i en descriu el seu funcionament. S'utilitza el model i les tècniques d'orientació a objectes, traduint els requisits especificats en el capítol anterior a un llenguatge més formal (models i diagrames UML) i identificant les classes fonamentals que seran la base de la implementació del programari.

3.1. Paquets.

Atenent al grau de dependència del casos d'ús, es distingeixen dos paquets d'anàlisi: ***Proxy*** que comprèn els casos d'ús relacionats amb les peticions (casos d'ús 1 i 2) i ***GestorProxy*** que inclou els casos d'ús referents a la gestió de les dades del *proxy* (resta de casos).

Dins del primer paquet es poden distingir tres paquets de serveis: ***Peticions*** i els paquets ***Filtre*** i ***Històrics***, ja que aquests dos últims poden ser opcionals.
Dins del segon paquet s'identifiquen quatre paquets de serveis: ***GestorHistòric*** (cas d'ús 3), ***GestorUsuaris*** (casos d'ús 4 i 5), ***GestorFiltre*** (cas d'ús 6) i ***ConfiguracioSeguiment*** (cas d'ús 7), tots ells opcionals.

L'esquema de les relacions entre els paquets és el següent:

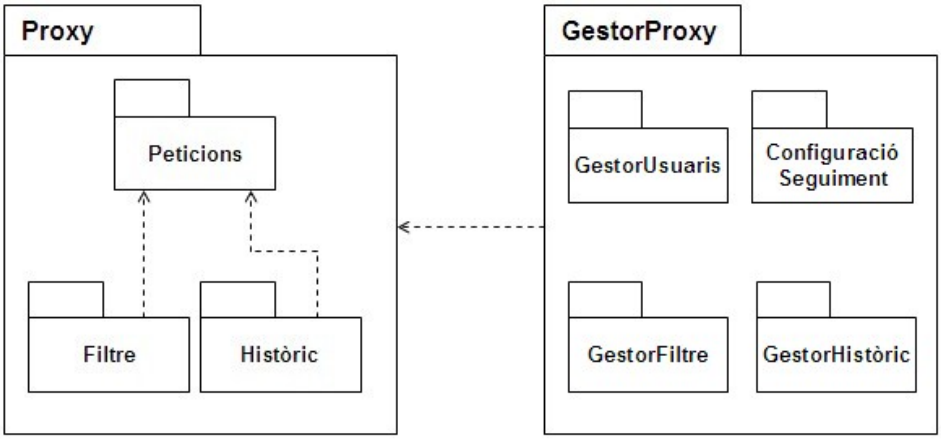


Figura 3.1 Paquets d'anàlisi i serveis (esquema de relacions)

3.2. Revisió de casos d'ús.

Es revisen els casos d'ús 1, 2 i 6 per a millorar-ne la precisió i el detall de la informació.

Cas d'ús número 1: "Iniciar sessió"

Descripció: S'afegeix que a l'iniciar-se la *sessió*, es guarden l'*identificador* de l'**usuari** que estableix la *sessió*, l'*adreça IP* de la màquina des d'on s'inicia la *sessió* i la *data i hora* en que s'ha iniciat la sessió.

Cas d'ús número 2: "Fer petició"

Descripció: (*S'amplia i es detalla la descripció*). L'**usuari** sol·licita un recurs web, teclejant la seva URI a través del navegador o bé fent clic damunt un enllaç de la pàgina que té en pantalla. De la petició efectuada es guarden l'*identificador* d'*usuari*, l'*adreça IP* de la màquina, la *data i hora*, la URI i el host de la petició dins la *cua de peticions* del *proxy*. S'actualitza la *data i hora* de l'última petició de la *sessió* corresponent a l'**usuari**. Una vegada processada la *petició*, es guarda a l'històric si la petició ha estat *permesa* o no, la

mida del recurs en bytes i la *categoria* i el motiu, en el cas de permetre's o no la seva consulta. Tot seguit, s'envia la resposta corresponent a l'**usuari**.

Cas d'ús número 6: "Gestionar filtre"

Descripció: (*S'amplia i es detalla la descripció*). El filtrat de continguts es realitza en base a diversa informació:

- Una llista de *host permesos* (*llista blanca*) on s'indica els *hosts* que no cal filtrar.
- Una llista de *hosts prohibits* (*llista negra*) on s'indica el *host* i la *categoria* del mateix.
- Una llista d'*etiquetes* (*PIC's*) on s'indica l'*organització* que ha efectuat la relació d'etiquetes i el *codi*, *categoria* i *descripció* de l'etiqueta.
- Una llista de paraules o petites frases on s'indica la *paraula* o frase, la *categoria* i el *pes* (indicat de 0 a 10, fins a 5 dubtós, més de 5 alta probabilitat) al que pertany.

Els diversos tipus de continguts que són filtrats pel *proxy* estan agrupats en *categories* (pe., pornografia, drogues, violència,...).

El *motiu* indica la causa de ser permès o no.

0 →Host permès, indicat a la llista blanca.
1 →Host permès, no indicat a la llista blanca.
2 →Host no permès per paraules clau (dubtós). Pendent de verificar per l'administrador.
3 →Host no permès per paraules clau (alta probabilitat). Pot ser verificat per l'administrador.
4 →Host no permès per etiquetes PIC.
5 →Host no permès, indicat a la llista negra.

Els casos d'ús 4, 5 i 6 corresponen al manteniment d'usuaris i de la informació del filtrat, per tant, es podrien detallar i separar en casos d'ús d'alta, modificació, baixa i consulta de les respectives dades, però, arribar a aquest detall no aporta cap concepte nou important i, en canvi, multiplica per quatre el nombre de casos d'ús.

S'afegeixen els següents casos d'ús:

Cas d'ús número 8: "Tancar sessions"

Resum de la funcionalitat: Tanca les *sessions* de connexió actives.

Paper dins el treball de l'actor: Una de les tasques principals del **temporitzador**.

Actors: Temporitzador.

Casos d'ús relacionats: Cap.

Precondició: Cap.

Postcondició: S'han tancat les *sessions* actives en que l'usuari no ha efectuat peticions des d'un temps determinat com a *timeout*.

Descripció: Cada un cert temps, indicat en la configuració del *proxy*, el **temporitzador** executa el procés de tancament de *sessions* d'*usuari*. En aquest procés, per cada *sessió*, s'avalua quan de temps ha transcorregut des de l'última petició que ha efectuat l'usuari. Si aquest supera el temps de *timeout*, es tanca la *sessió.*

Observacions: S'anota la *data hora* de tancament de la *sessió*.

Cas d'ús número 9: "Canvi de contrasenya"

Resum de la funcionalitat: L'**usuari** canvia la seva paraula clau.

Paper dins el treball de l'actor: Una de les tasques ocasionals de tots els **usuaris**.

Actors: Usuari d'Internet, **gestor** i **administrador**.

Casos d'ús relacionats: Iniciar sessió.

Precondició: L'**usuari** ha iniciat una *sessió* i no està *suspès*.

Postcondició: S'ha canviat la contrasenya de l'**usuari**.

Descripció: L'**usuari** sol·licita canviar la seva paraula clau, a través de l'opció corresponent. Se li presenta una pantalla on ha de teclejar la paraula clau actual i la nova contrasenya dues vegades. Una vegada l'usuari ha acceptat fer el canvi, si la paraula clau actual és correcta i no hi ha cap incidència amb la nova (no és igual que l'anterior, té la llargada requerida, ...), es modifica la contrasenya.

Observacions: Si al sol·licitar el canvi de contrasenya s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

Cas d'ús número 10: "Sortir de sessió"

Resum de la funcionalitat: L'**usuari** sol·licita acabar la sessió.

Paper dins el treball de l'actor: Una de les tasques principals de tots els **usuaris**.

Actors: Usuari d'Internet, **gestor** i **administrador**.

Casos d'ús relacionats: Iniciar sessió.

Precondició: L'**usuari** ha iniciat una *sessió* i no està *suspès*.

Postcondició: L'**usuari** ha sortit de sessió.

Descripció: L'**usuari** demana sortir de la *sessió*, a través de l'opció corresponent. S'anota la *data hora* de tancament de la *sessió*.

Observacions: Si al sol·licitar acabar la *sessió* s'ha superat el temps de *timeout*, des de la última *petició*, es presenta de nou la pàgina d'inici de *sessió*, sol·licitant una nova validació i s'inicia una nova *sessió*.

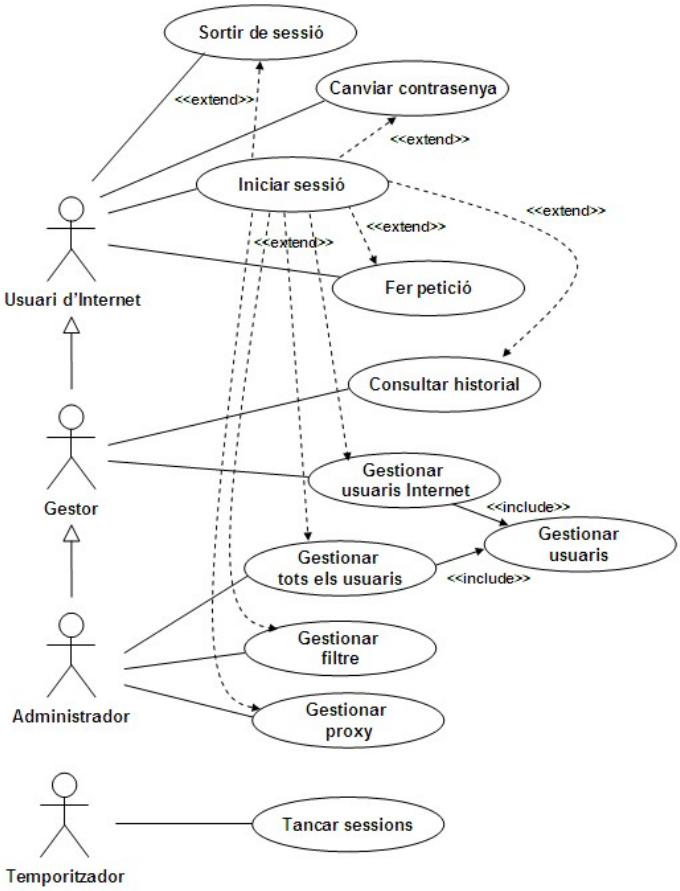


Figura 3.2 Diagrama de casos d'ús revisats

3.3. Especificació de les classes d'anàlisi.

En línia amb el disseny per capes i per a facilitar el manteniment del programari i futures modificacions, es distingeixen tres tipus de classes: de frontera, d'entitats i de control.
Les **classes de frontera** representen la interfície d'usuari, les **classes d'entitats** corresponen als objectes del domini (la majoria són persistents) i les **classes de control** identifiquen objectes interns del programari (no persistents).

Tot seguit, s'identifiquen les classes d'entitats. A continuació, i per cada cas d'ús, es presenta un diagrama de col·laboració on es mostren els tres tipus de classes i llurs operacions.

3.3.1. Identificació de les classes d'entitats.

En base als casos d'ús, s'identifiquen les classes d'entitats.

Si una classe s'ha identificat en un cas d'ús anterior, es marca amb un asterisc després del seu nom. Si la classe és dubtosa, s'indica amb un signe d'interrogant després del seu nom.

Cas d'ús número 1, "Iniciar sessió": *Usuari, UsuariInternet, Gestor, Administrador, Sessió, PeticioActual, Identificació?, Contrasenya?, Timeout?.*

Cas d'ús número 2, "Fer petició": *Usuari*, UsuariInternet*, Gestor*, Administrador*, Sessió*, PeticioActual*, URI?, PeticioEnCua, PeticioHistorial, Timeout?*.*

Cas d'ús número 3, "Consultar historial": *Usuari*, Gestor*, Administrador*, Sessió*, PeticioHistorial*, Timeout?*.*

Cas d'ús número 4, "Gestionar usuaris d'Internet": *Usuari*, UsuariInternet*, Gestor*, Sessió*, Timeout?*.*

Cas d'ús número 5, "Gestionar tots els usuaris": *Usuari*, Administrador*, Sessió*, Timeout?*.*

Cas d'ús número 6, "Gestionar filtre": *Usuari*, Administrador*, Sessió*, HostPermès, HostProhibit, EtiquetaPIC, ParaulaClau, Categoria, Timeout?*.*

Cas d'ús número 7, "Gestionar proxy": *Usuari*, Administrador*, Sessió*, ConfProxy, PeticioActual*, PeticioEnCua*, Timeout?*.*

Cas d'ús número 8, "Tancar sessions": *Usuari*, Sessió*, Timeout?*.*

Cas d'ús número 9, "Canvi de contrasenya": *Usuari*, UsuariInternet*, Gestor*, Administrador*, Sessió*, Timeout?*.*

Cas d'ús número 10, "Sortir de sessió": *Usuari*, UsuariInternet*, Gestor*, Administrador*, Sessió*, Timeout?*.*

Comentaris.

Les classes *Identificació* i *Contrasenya* es descarten ja que no disposen de cap atribut, alhora que es poden considerar atributs de la classe *Usuari*. El mateix succeeix amb la classe *URI*, que es pot considerar atribut de la classe *Petició* i de la classe *Historial*.
La classe Timeout és un atribut de la configuració del *proxy*.

En una primera aproximació, les classes d'entitats són: *Usuari, UsuariInternet, Gestor, Administrador, Sessió, PeticioActual, PeticioEnCua, PeticioHistorial, Categoria, HostPermès, HostProhibit, EtiquetaPIC, ParaulaClau, ConfProxy.*

3.3.2. Especificació dels atributs de les classes d'entitats.

Els atributs de les classes d'entitats especificades a l'apartat anterior són:

Classe *Usuari*: *identificador* (string), *contrasenya* (string), *codi* (string), *nom* (string), *area* (string), *suspes* (boolean).

Classe *UsuariInternet*: *identificador* (string), *contrasenya* (string), *codi* (string), *nom* (string), *area* (string), *suspes* (boolean).

Classe *Gestor*: *identificador* (string), *contrasenya* (string), *codi* (string), *nom* (string), *area* (string), *suspes* (boolean).

Classe *Administrador*: *identificador* (string), *contrasenya* (string), *codi* (string), *nom* (string), *area* (string), *suspes* (boolean).

NOTA:

- L'atribut *identificador* identifica l'usuari dins del sistema i, així, s'assegura que no hi haurà cap incidència al donar d'alta el usuaris.
- L'atribut *codi* indica el NIF, el número d'empleat, el codi d'alumne,... depenent de l'entitat que utilitzi el programari.
- L'atribut *area* (opcional), permet identificar, encara més, l'usuari. En el cas d'un centre educatiu pot ser el curs que fa l'alumne. En el cas d'una empresa, el departament o secció on treballa l'empleat. Etc ...
- L'atribut *suspes* indica si l'usuari ha estat suspès i no se li permet l'accés al sistema.

Classe *Sessió*: *idUsuari* (string), *IPmaquina* (string), *dataInici* (datetime), *dataUltimaPeticio* (datetime), *dataFi* (datetime).

Classe *PeticioActual*: *idUsuari* (string), *IPmaquina* (string), *dataPetició* (datetime), *URI* (string), *host* (string).

Classe *PeticioEnCua*: *idUsuari* (string), *IPmaquina* (string), *dataPetició* (datetime), *URI* (string), *host* (string).

Classe *PeticioHistorial*: *idUsuari* (string), *IPmaquina* (string), *dataPetició* (datetime), *URI* (string), *host* (string), *permís* (boolean), id*Categoria* (integer), *motiu* (integer), *midaRecurs* (integer).

Classe *Categoria*: *identificador* (integer), *nom* (string).

Classe *HostPermès*: *host* (string).

Classe *HostProhibit*: *host* (string), *idCategoria* (integer).

Classe *EtiquetaPIC*: *tipus* (string), *codi* (string), *idCategoria* (integer), *descripció* (string).

NOTA:

- L'atribut *tipus* identifica l'organització o nom de la norma d'etiquetatge.

Classe *ParaulaClau*: *paraula* (string), *idCategoria* (integer), *pes* (integer).

Classe *ConfProxy*: *usuaris* (integer), *filtre* (boolean), *historial* (boolean), *mp3* (boolean), *mpeg* (boolean), *avi* (boolean), *exe* (boolean), *zip* (boolean), *jpeg* (boolean), *gif* (boolean), *png* (boolean), *timeoutSessions* (integer), *tancarSessions* (integer).

NOTA:

- La classe *CpnfProxy* és única. Permet la configuració del *proxy*.
- L'atribut *usuaris* indica si es requereix identificació d'usuari per accedir a Internet (0) i si es així, si és igual al codi afegint-hi una "U" davant (1) o es generat pel sistema de manera seqüencial (2).
- Els atributs *filtre* i *historial* indiquen si està actiu el servei o no.
- Els atributs de tipus de fitxer (mp3, mpeg, avi, ...) indiquen si es permet o no la descarrega d'aquests tipus de fitxers.
- L'atribut *timeoutSessions* indica el temps de timeout.
- L'atribut *tancarSessions* indica cada quan s'executarà el procés de tancament de sessions.

3.3.3. Relacions de les classes d'entitats.

Relacions d'herència.

Com es pot observar, les classes *Usuari, UsuariInternet, Gestor* i *Administrador* tenen els mateixos atributs i, segurament, les mateixes operacions, ja que *UsuariInternet, Gestor* i *Administrador* són tipus d'usuari, per tant, generalitzant, les classes *UsuariInternet, Gestor* i *Administrador* es defineixen com a subclasses de la classe *Usuari* i, aquesta, queda definida com a classe abstracta passant a tenir tots els atributs.

Les classes *PeticioActual, PeticioEnCua* i *PeticioHistorial* es generalitzen quedant com a subclasses de la classe abstracta *Petició*. Les subclasses "perden" els atributs *idUsuari*, *IPmaquina*, *dataPetició*, *URI* i *host*.

Les classes *HostPermès* i *HostProhibit* es generalitzen quedant com a subclasses de la classe abstracta *HostFiltre*. Aquesta última passa a tenir l'atribut *host*.

Associacions.

Entre les classes d'entitats s'estableixen vàries associacions:

- Un usuari pot iniciar o no una sessió (classe associativa) en una data i pot iniciar sessions en vàries dates. En una data concreta, pot ser que cap usuari estableixi sessió o bé que ho facin varis usuaris.
- En una data concreta es poden efectuar vàries peticions o bé cap. Una petició es sol·licita en una data concreta.
- Una sessió pot tenir en cua cap o vàries peticions. Una sessió té una o cap petició que s'estigui processant. Una sessió té una o vàries peticions històriques (l'inici de sessió ja n'és una). En qualsevol cas, tota petició pertany a una sessió en concret.
- Una petició, una vegada ha estat resposta pot pertànyer a una categoria de les prohibides o no. Una categoria pot tenir cap o vàries peticions a l'històric.
- Un host prohibit, una etiqueta PIC o una paraula clau pertanyen a una categoria. Una categoria pot tenir o no hosts prohibits, etiquetes PIC o paraules clau registrades al sistema.

En el següent diagrama es pot observar que s'han pogut suprimir diversos atributs de les classes perquè són redundants amb les associacions.

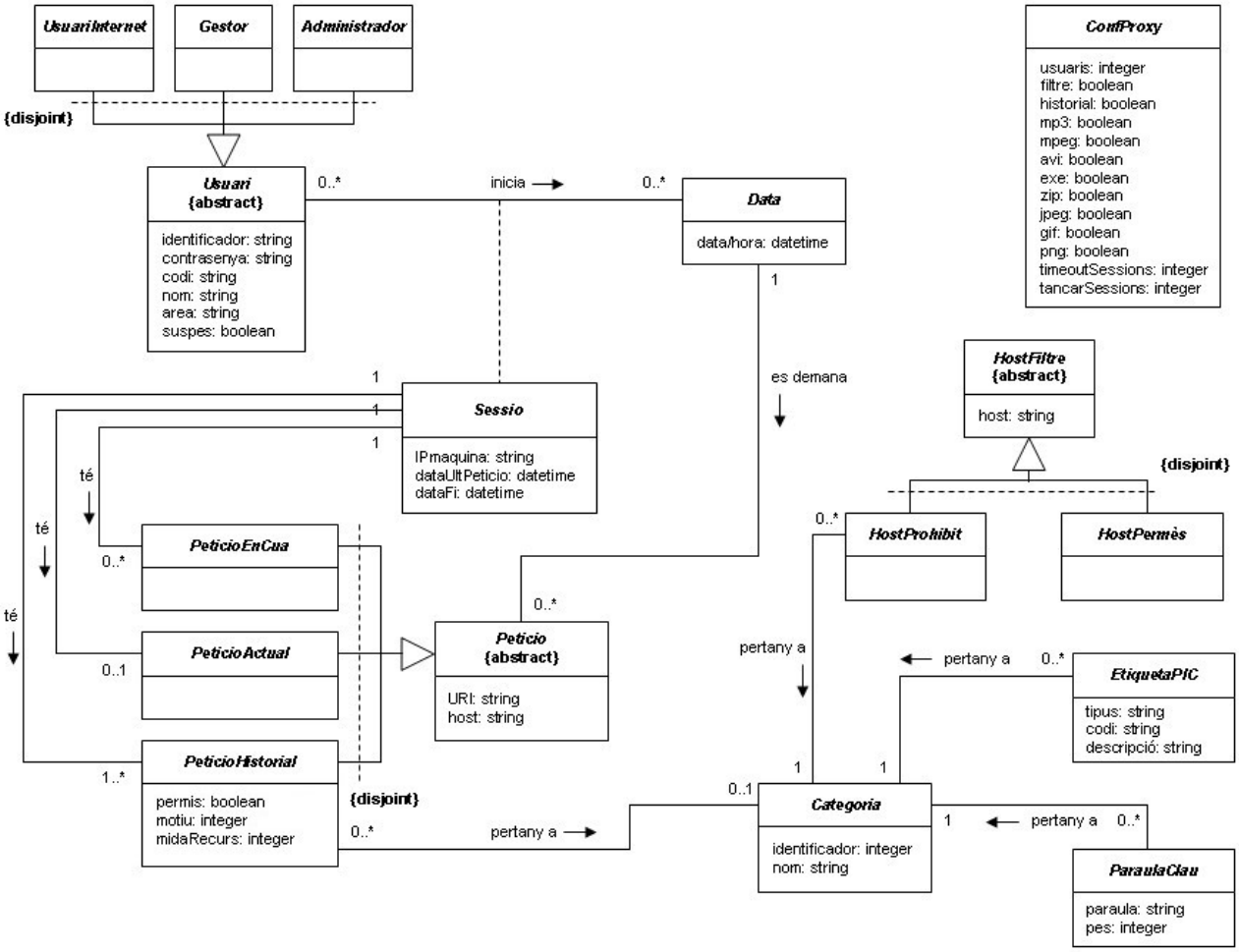


Figura 3.3 Diagrama de classes d'entitats

3.3.4. Identificació de les classes de frontera, de control i de les operacions.

Tot seguit es mostren els **diagrames de col·laboració** dels casos d'ús, els quals són de gran ajut per a identificar les classes de frontera, de control i les operacions de totes les classes.

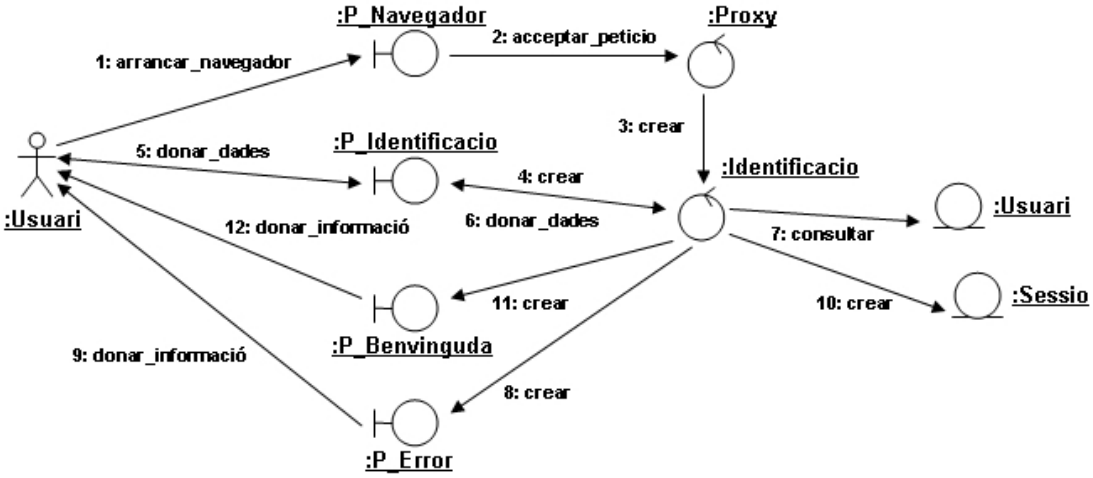


Figura 3.4 Diagrama de col·laboració cas nº 1 "Iniciar sessió"

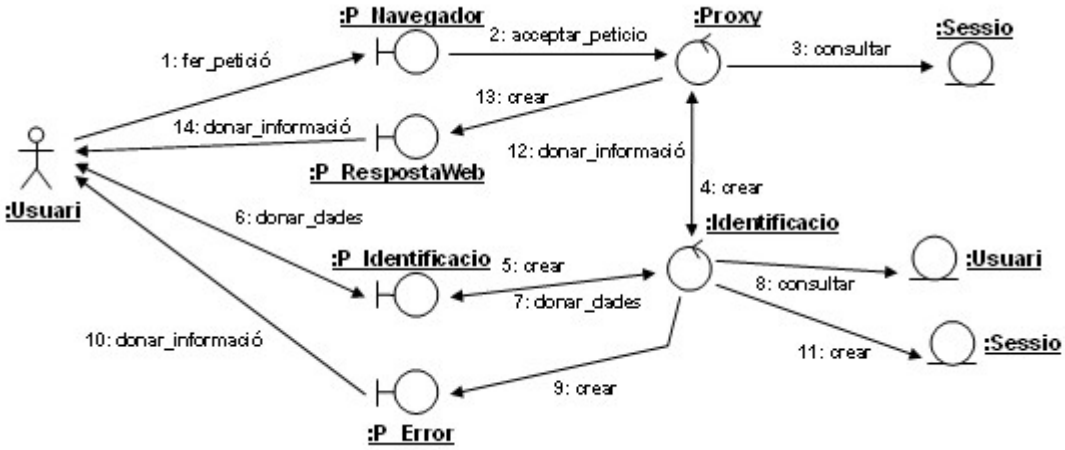


Figura 3.5 Diagrama de col·laboració cas nº 2 "Fer petició"

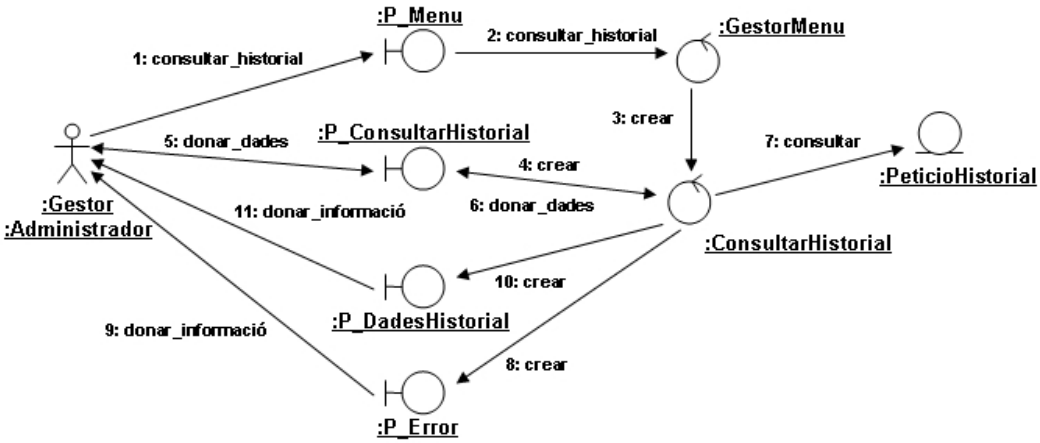


Figura 3.6 Diagrama de col·laboració cas nº 3 "Consultar historial"

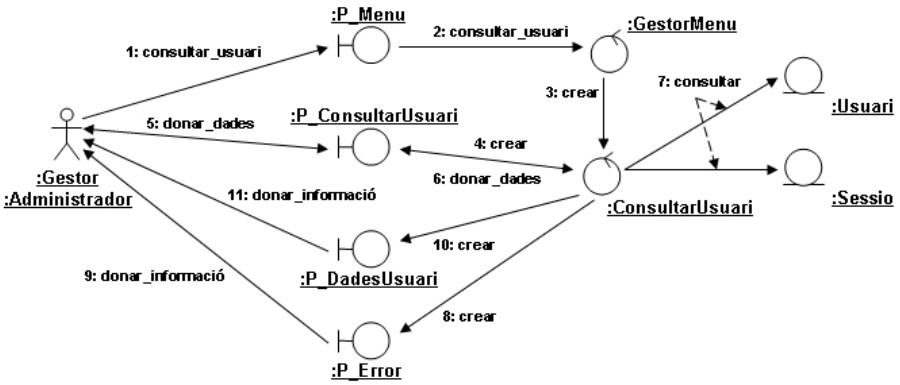


Figura 3.7 Diagrama de col·laboració casos nº 4 i 5 "Gestionar usuaris" (consulta)

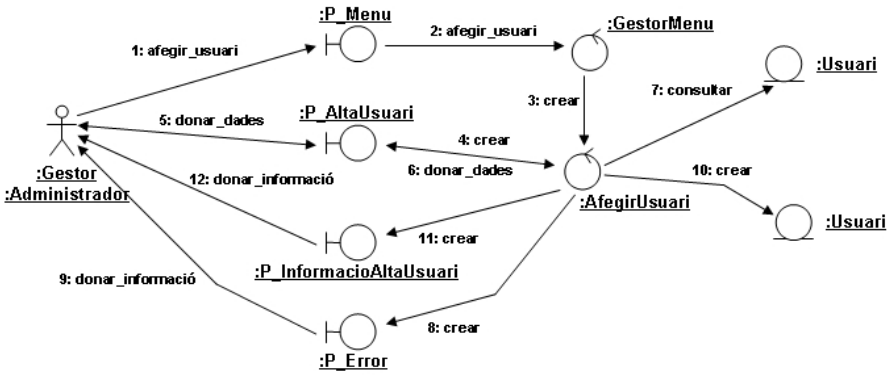


Figura 3.8 Diagrama de col·laboració casos nº 4 i 5 "Gestionar usuaris" (alta)

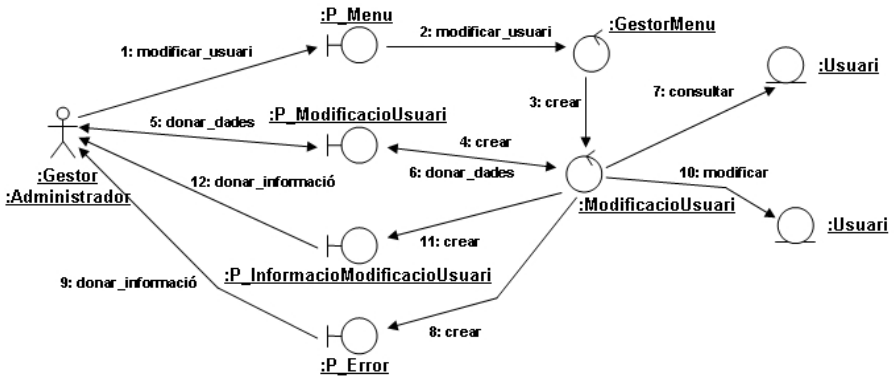


Figura 3.9 Diagrama de col·laboració casos nº 4 i 5 "Gestionar usuaris" (modificació)

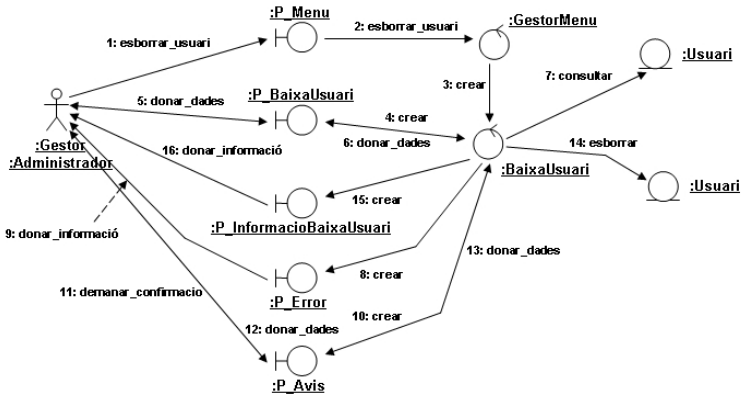


Figura 3.10 Diagrama de col·laboració casos nº 4 i 5 "Gestionar usuaris" (baixa)

Nota: [Taula] indica qualsevol de les següents classes d'entitats: *HostPermès, HostProhibit, EtiquetaPIC, ParaulaClau*.

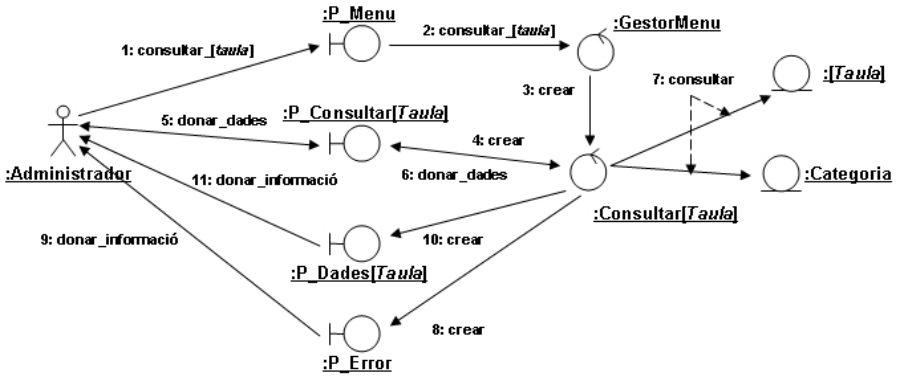


Figura 3.11 Diagrama de col·laboració cas nº 6 "Gestionar filtre" (consulta)

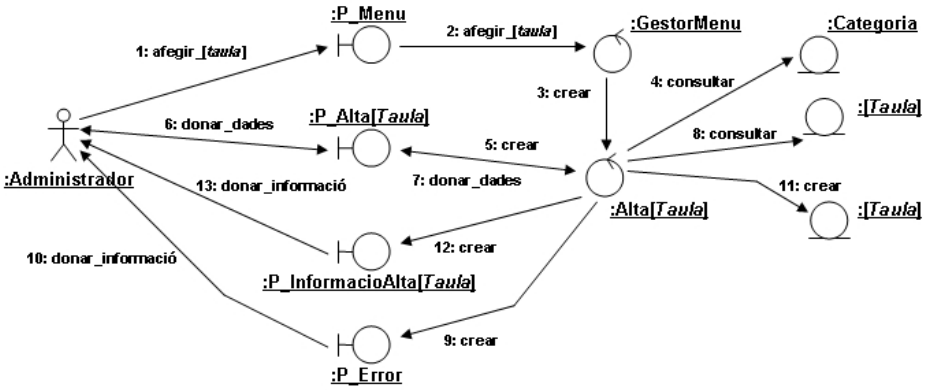


Figura 3.12 Diagrama de col·laboració cas nº 6 "Gestionar filtre" (alta)

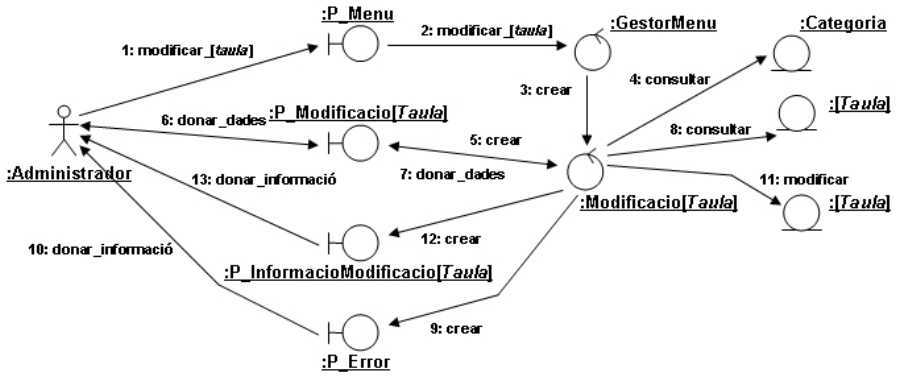


Figura 3.13 Diagrama de col·laboració cas nº 6 "Gestionar filtre" (modificació)

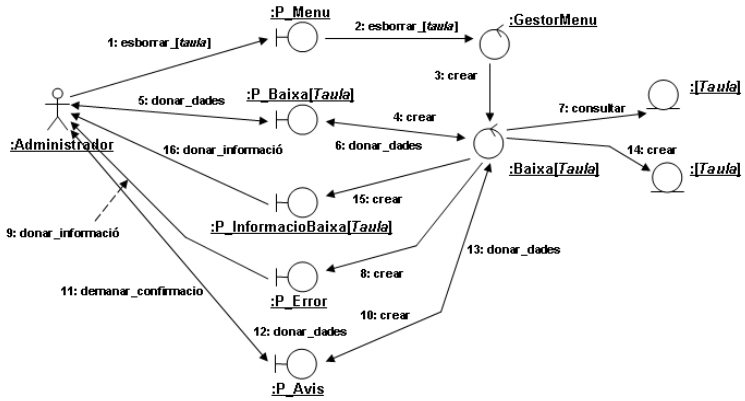


Figura 3.14 Diagrama de col·laboració cas nº 6 "Gestionar filtre" (baixa)

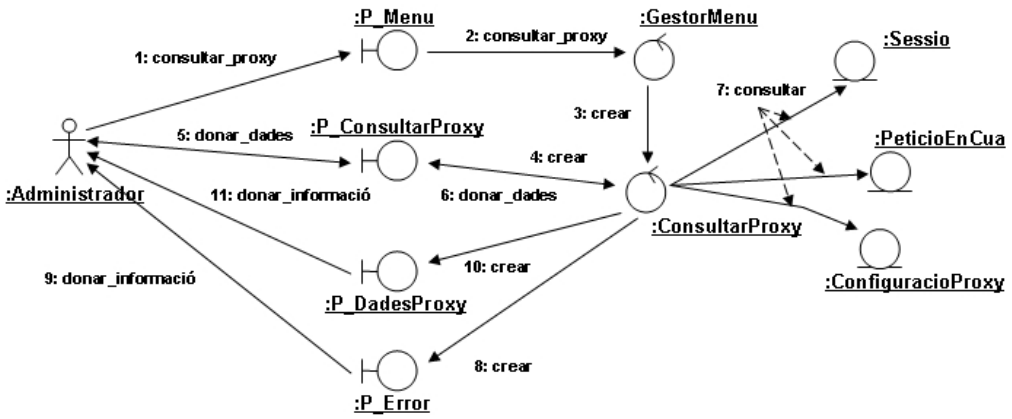


Figura 3.15 Diagrama de col·laboració cas nº 7 "Gestionar proxy" (consulta)

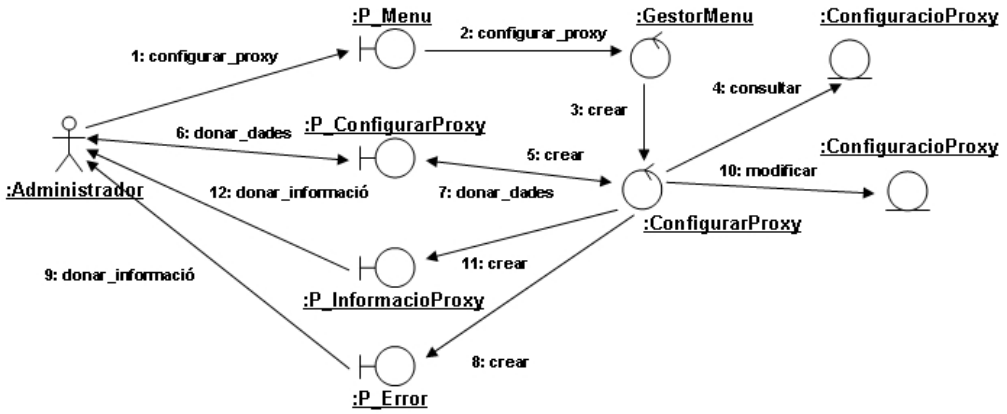


Figura 3.16 Diagrama de col·laboració cas nº 7 "Gestionar proxy" (configurar)

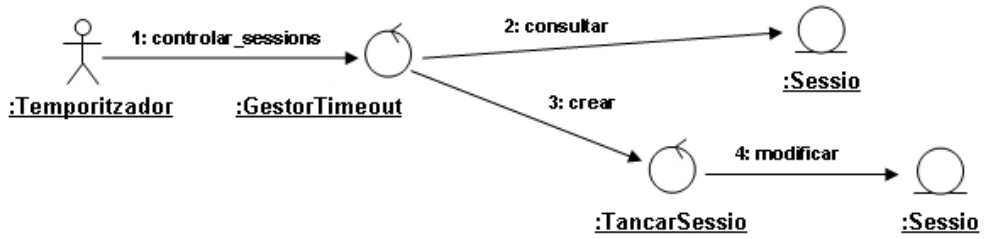


Figura 3.17 Diagrama de col·laboració cas nº 8 "Tancar sessions"

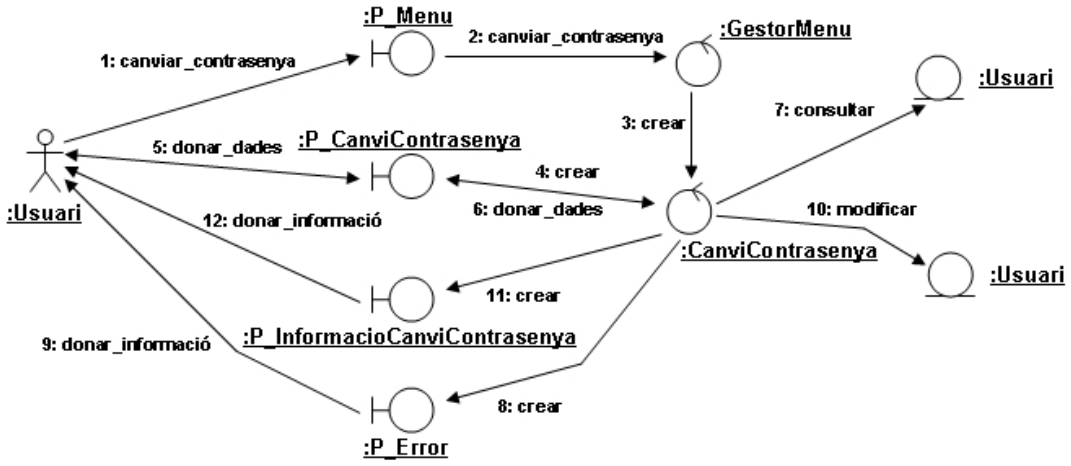


Figura 3.18 Diagrama de col·laboració cas nº 9 "Canviar contrasenya"

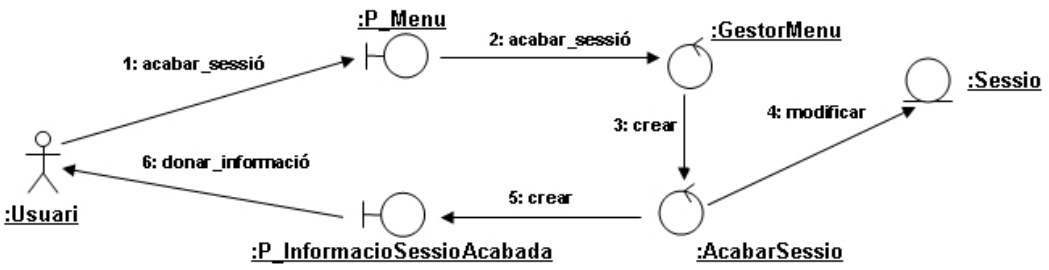


Figura 3.19 Diagrama de col·laboració cas nº 10 "Sortir de sessió"

3.4. Especificació formal dels casos d'ús.

En els diagrames de col·laboració simplificats de l'apartat anterior, hi ha missatges que s'envien si és compleix una condició, per exemple, en el cas d'ús número 2 "*Fer petició*" la classe de control *Proxy* envia el missatge *crear* a la classe de control *Identificacio* si l'usuari fa una petició i no està identificat.

En la majoria dels casos d'ús especificats, els diagrames de col·laboració anteriors són suficients per a l'anàlisi, però, tot just en l'exemple citat anteriorment no n'hi ha prou donada la complexitat del cas i, per tant, se'n fa una especificació complementària en el diagrama de seqüència de la *figura 3.21*, on es detallen els missatges i les operacions que s'efectuen quan un usuari demana la petició d'un recurs web, i en el diagrama d'estats següent, on s'especifiquen les diverses fases que travessa la petició des que és acceptada pel *proxy* i fins que aquest lliura la resposta.

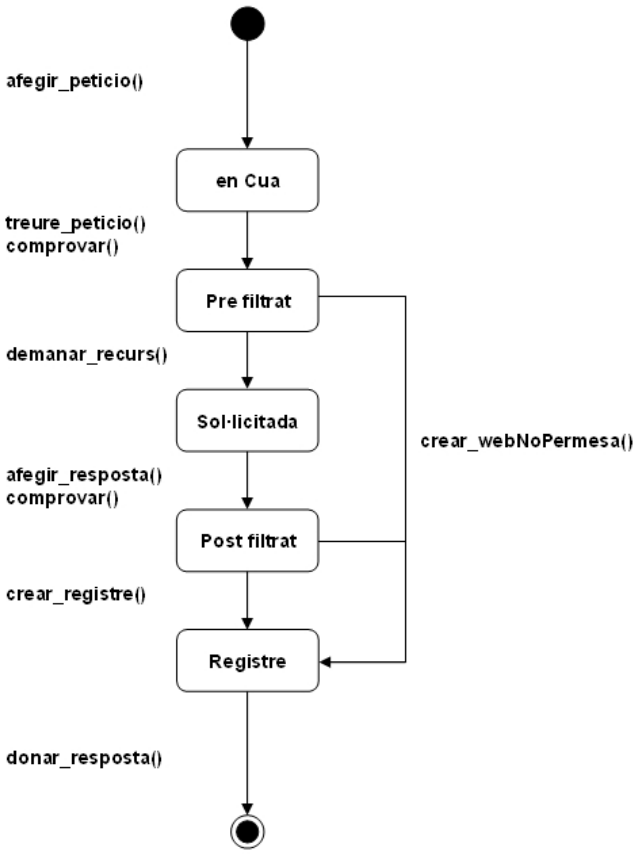


Figura 3.20 Diagrama d'estat cas nº 2 "Fer petició"

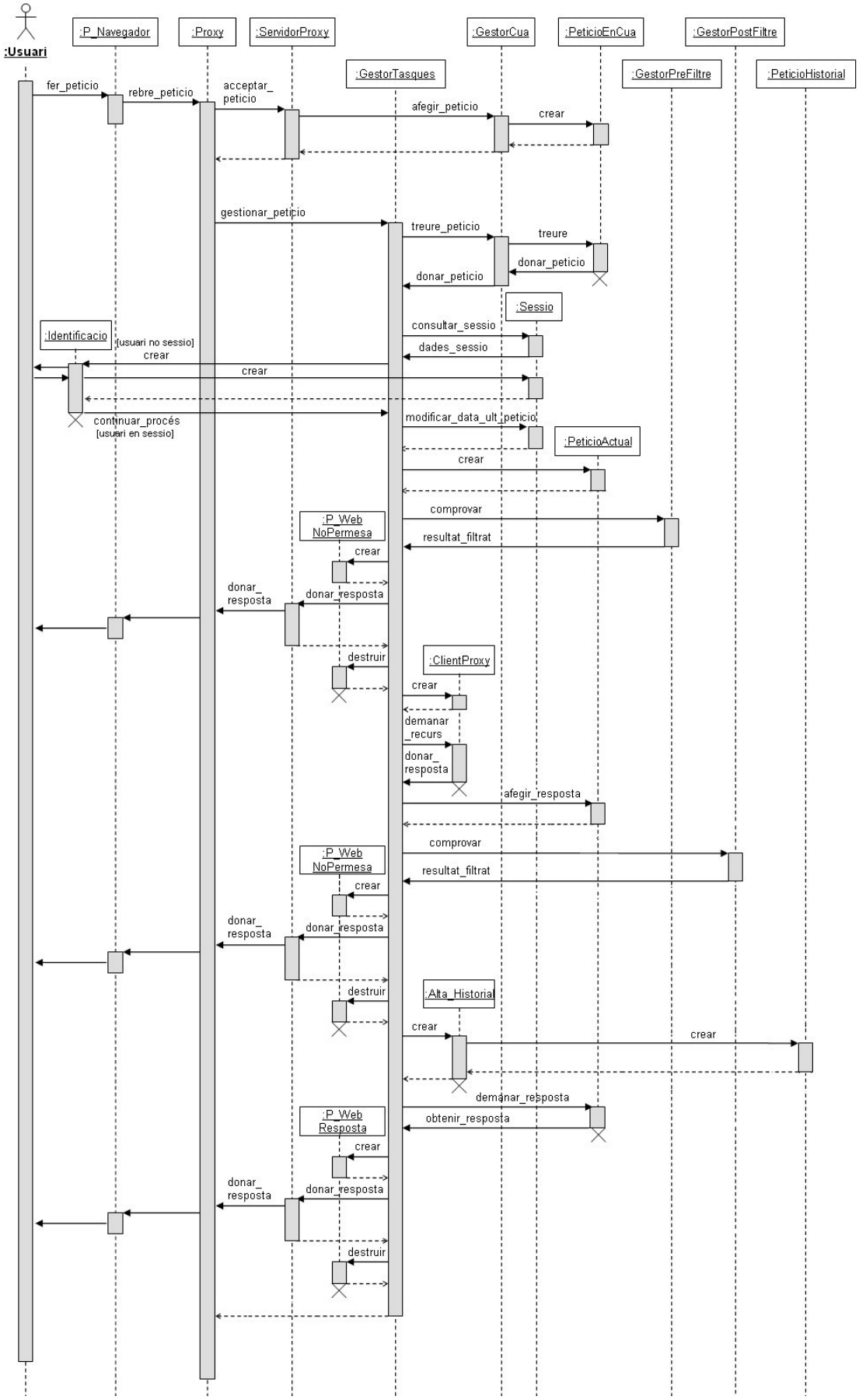


Figura 3.21 Diagrama de seqüències cas nº 2 "Fer petició"

3.5. Interfície d'usuari.

La interfície d'usuari s'obté de la descripció de les tasques (*punt 2.9.2*) i de la identificació de les classes frontera (*punt 3.3.4*). A continuació es mostren les interfícies associades a les classes frontera més rellevants.

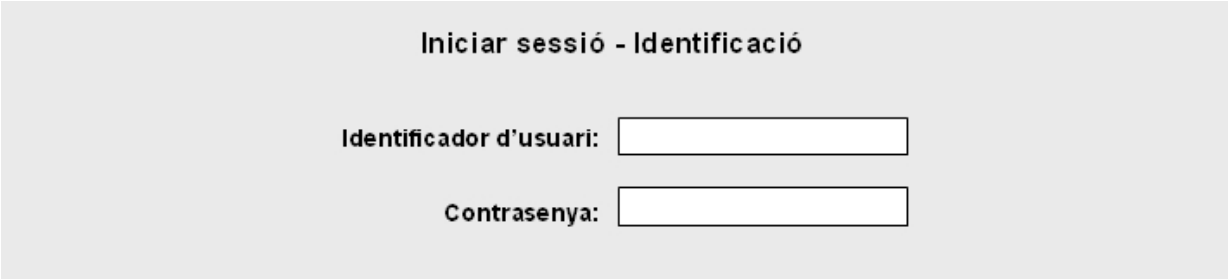


Figura 3.22 Interfície de la classe frontera *P_Identificacio*

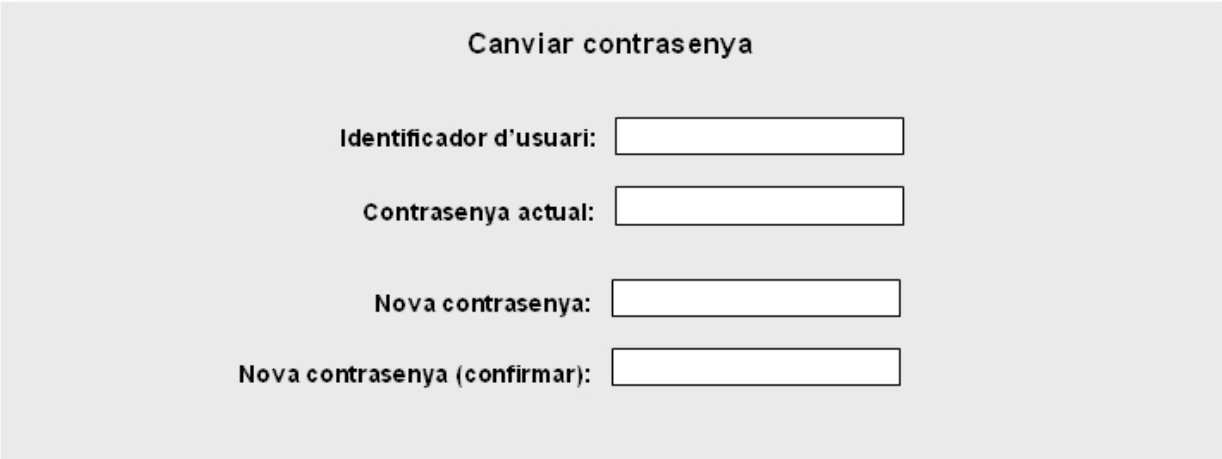


Figura 3.23 Interfície de la classe frontera *P_CanviContrasenya*

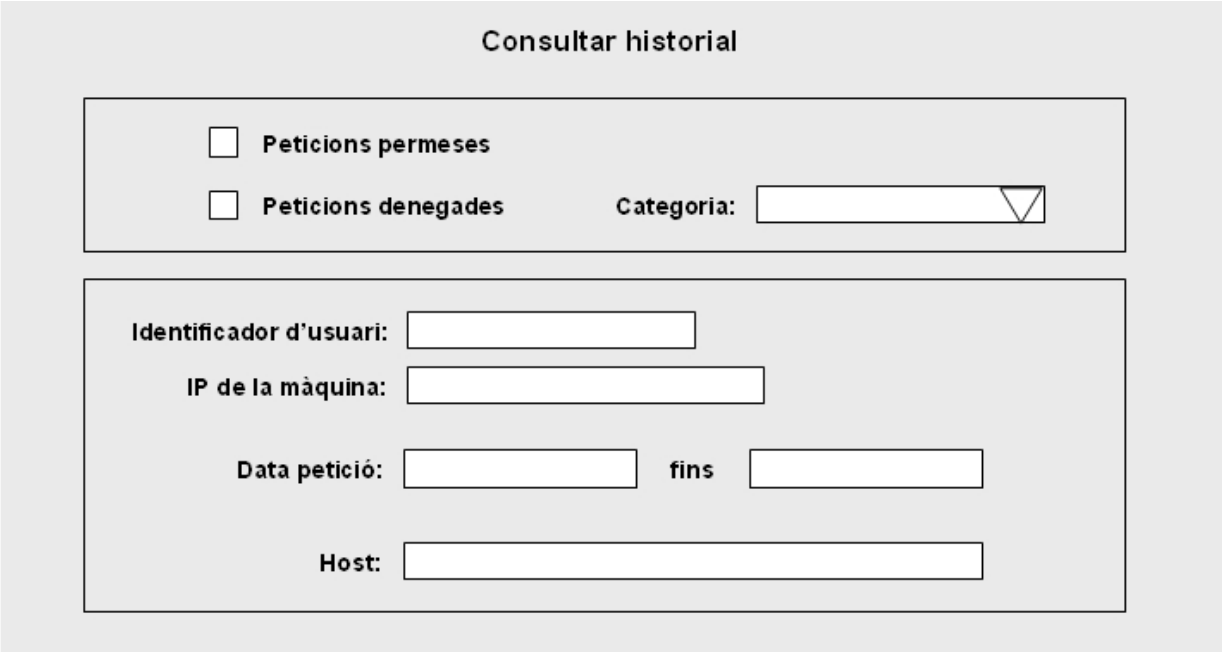


Figura 3.24 Interfície de la classe frontera *P_ConsultarHistorial*

Dades historial

*	URI	Usuari	IP	Data	Permís	Categoria	Motiu	Mida (Kb.)
☐	xxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxx.xxx.x.xxx	xx/xx/xx	Si	xxxxxxxxx	0	99.999.999
☐	xxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxx.xxx.x.xxx	xx/xx/xx	No	xxxxxxxxx	4	99.999.999
☐	xxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxx.xxx.x.xxx	xx/xx/xx	No	xxxxxxxxx	2	99.999.999
☐	xxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxx.xxx.x.xxx	xx/xx/xx	Si	xxxxxxxxx	0	99.999.999

[*] *Veure més informació*

Figura 3.25 Interfície de la classe frontera *P_DadesHistorial*

Alta usuari

Tipus d'usuari: ○ Usuari Internet ○ Gestor ○ Administrador

Nom de l'usuari: [ ]

Codi de l'usuari: [ ]

Àrea: [ ]

Figura 3.26 Interfície de la classe frontera *P_AltaUsuari*

Consultar usuari

Cercar Tipus d'usuari: ○ Usuari Internet ○ Gestor ○ Administrador

Nom de l'usuari: [ ] Codi de l'usuari: [ ]

Àrea: [ ] ☐ Suspès

M	E	Tp	Id. usuari	Nom	Codi	Àrea	Suspès
☐	☐	I	xxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxxxxxxxxxxxxxxxxxxxxx	No
☐	☐	I	xxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxxxxxxxxxxxxxxxxxxxxx	Si
☐	☐	G	xxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxxxxxxxxxxxxxxxxxxxxx	No
☐	☐	A	xxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	xxxxxxxxxxxxxxxxxxxxxx	No

[M] *Modificar* [E] *Esborrar*

Figura 3.27 Interfície de la classe frontera *P_ConsultaUsuari*

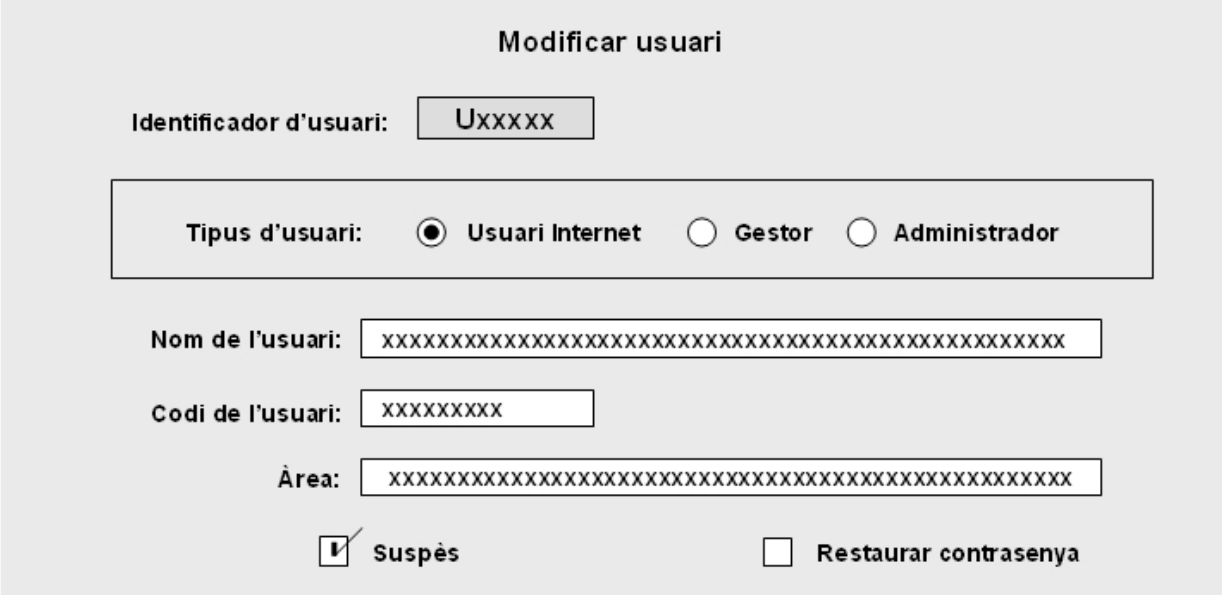


Figura 3.28 Interfície de la classe frontera *P_ModificacioUsuari*

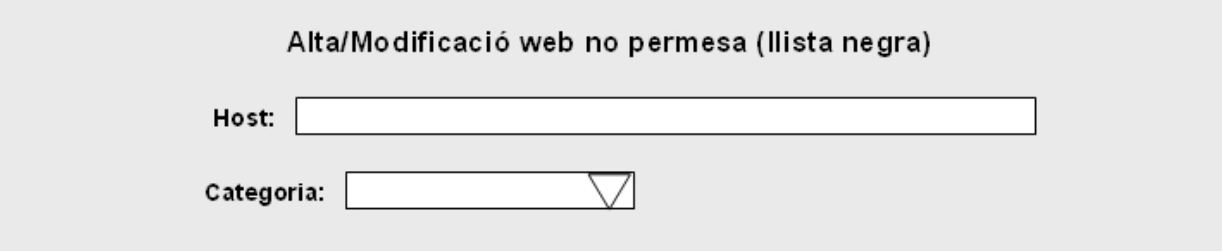


Figura 3.29 Interfície de les classes frontera *P_Alta/ModificacioHostProhibit*

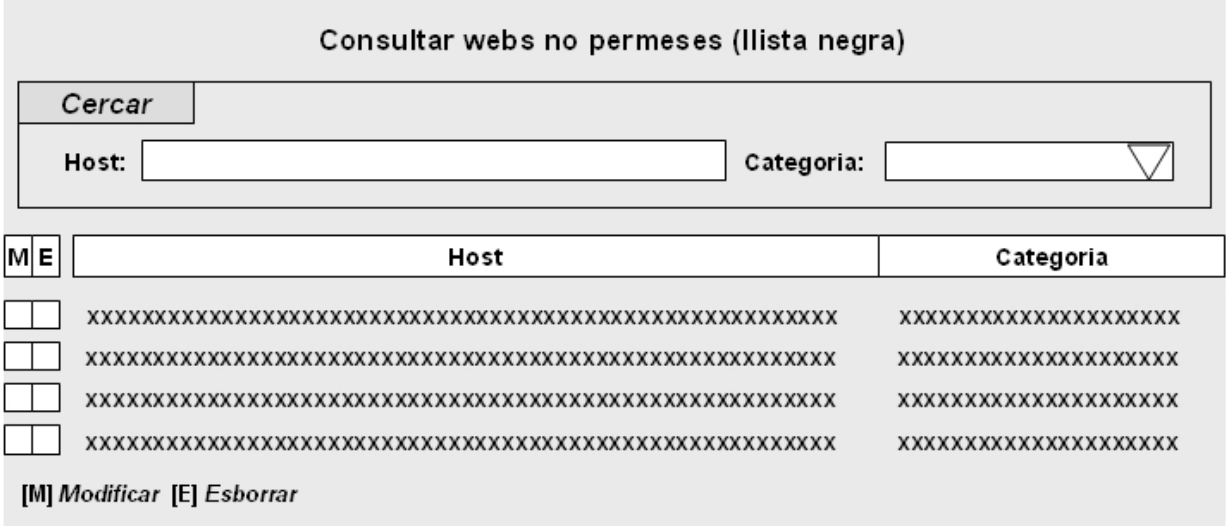


Figura 3.30 Interfície de la classe frontera *P_ConsultaHostProhibit*

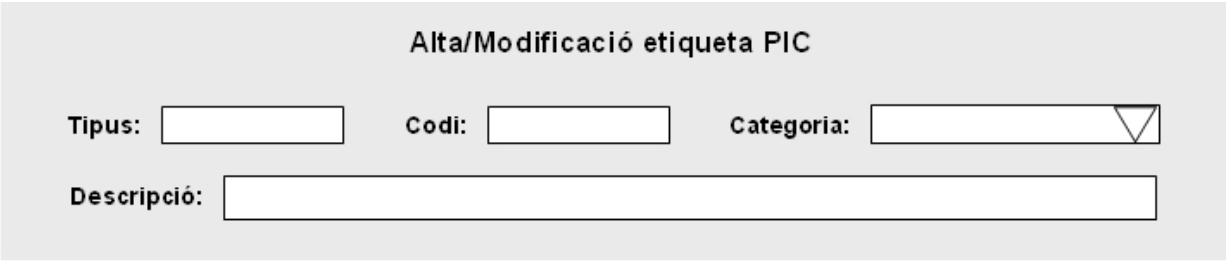


Figura 3.31 Interfície de les classes frontera *P_Alta/ModificacioEtiquetaPIC*

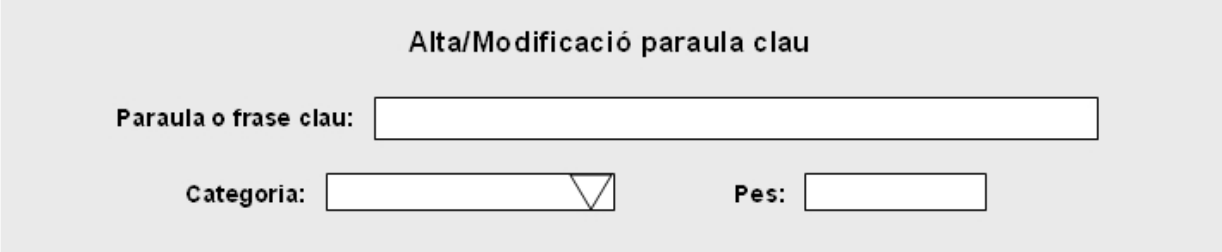


Figura 3.32 Interfície de les classes frontera *P_Alta/ModificacioParaulaClau*

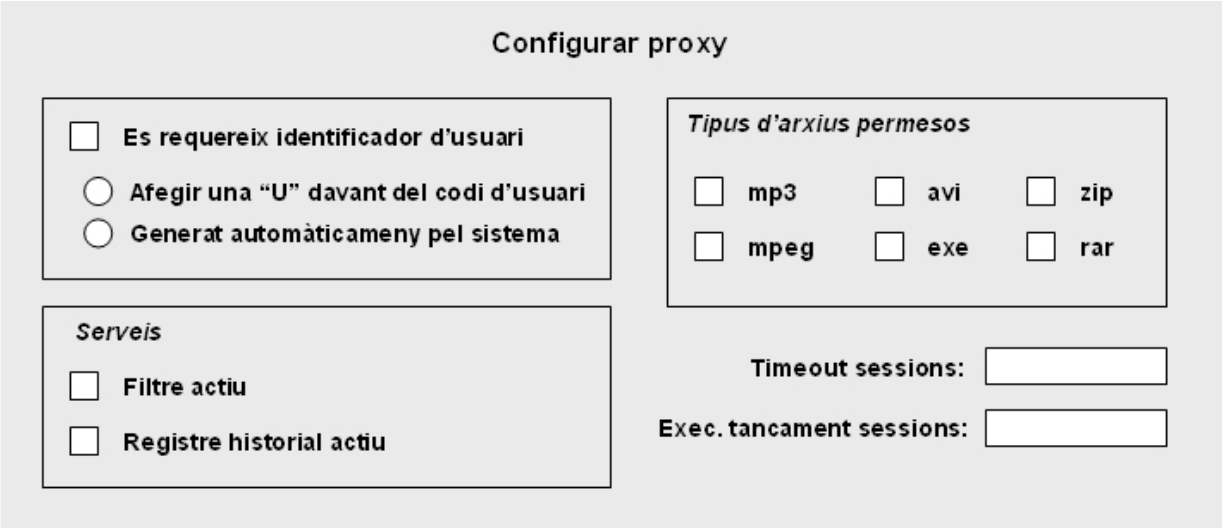


Figura 3.33 Interfície de la classe frontera *P_ConfigurarProxy*

Capítol 4. Disseny

4.1. Paradigmes i tecnologies.

El disseny està fonamentat en tres paradigmes: la **programació orientada a objectes**, el **model *client-servidor*** i la **programació per capes**.
A diferència de l'anàlisi, el disseny s'expressa en termes de la tecnologia emprada en el projecte, és a dir, el llenguatge de programació, el sistema gestor de les dades i l'eina de suport a la interfície gràfica d'usuari.

4.1.1. El model *client-servidor*.

Aquest model consisteix, bàsicament, en un programa, el *client*, que efectua peticions a un altre programa, el *servidor*, que li dóna les respostes. En general, el client es comunica amb el servidor a través d'un protocol d'alt nivell d'abstracció com és, per exemple, l'*SQL*.

Els avantatges del model *client-servidor* són, sobretot, de tipus organitzatiu ja que centralitzen la gestió de la informació, facilitant i clarificant el disseny del sistema.

Tot i que ambdós programes es poden executar en una sola màquina, la seva utilització és més avantatjosa en un sistema multiusuari distribuït a través d'una xarxa d'ordinadors. Els sistemes *multicapa*, en els que diversos ordinadors actuen com a servidor executant diferents programes, són una arquitectura molt comuna d'aquest model.

4.1.2. La programació per capes.

La programació per capes és un estil de programació que consisteix en dividir els components primaris de l'aplicació, programar-los per separat i, finalment, unir-los.

En aquest projecte, el disseny i desenvolupament s'efectua en base al model de programació en tres capes: **presentació**, **lògica de negoci** i **dades**.

Cada capa resolt una tasca concreta, el que permet un disseny escalable. A més, permet distribuir les tasques de disseny i desenvolupament en grups de treballs diferents i, en el cas de modificacions, només cal centrar-se en la capa requerida sense haver de revisar la resta.

Tot seguit, s'indiquen les funcions de cada capa.

Capa	Funcions
Presentació	• Recollir informació de l'usuari. • Enviar aquesta informació a la capa de la lògica de negoci. • Recollir els resultats de la capa de la lògica de negoci. • Presentar els resultats a l'usuari.
Lògica de negoci	• Rebre informació de la capa de presentació. • Interactuar amb la capa de dades per a realitzar la lògica de negoci de l'aplicació. • Enviar els resultats a la capa de presentació.
Dades	• Emmagatzemar les dades. • Rebre les dades. • Manteniment de les dades. • Integritat de les dades.

Figura 4.1 Programació per capes. Funcions de les capes

4.1.3. PHP i SQLite.

El *PHP* (acrònim recursiu de *PHP: Hipertext Preprocessor*), és un llenguatge de programació interpretat, de codi lliure i força popular, utilitzat per a generar contingut dinàmic de tipus web. La seva modularitat el fa molt útil per al seu ús en servidors web [1].

És molt semblant als llenguatges de programació C i C++ en el tipus de dades, sintaxi i funcions. Cal remarcar que des de la versió 5.0, inclou un major suport quant a la programació orientada a objectes. Es poden ampliar les seves prestacions mitjançant biblioteques de tercers. Existeixen versions de *PHP* disponibles per als sistemes operatius Windows, GNU/Linux i altres.

Des de la versió 5.0, la distribució de *PHP* inclou el motor de base de dades *SQLite* [2]. *SQLite* és una biblioteca desenvolupada en llenguatge C que implementa un motor de base de dades SQL. *SQLite* no és una biblioteca client per a accedir a un servidor de dades sinó que el propi *SQLite* és el servidor de base de dades, de manera que les aplicacions que usen la biblioteca *SQLite* poden utilitzar bases de dades sense instal·lar un motor de bases de dades extern.

4.2. Disseny del model de base de dades.

Les dades de l'aplicació es gestionen amb el motor de base de dades *SQLite*. La biblioteca *SQLite* disposa de la classe *SQLiteDatabase* per a gestionar les dades.

Amb la finalitat de poder utilitzar un altre motor de base de dades i fer reutilitzable la resta del disseny, es dissenyen dues classes per a gestionar les dades: *DB_Agent* i *Table_Agent*.

- la classe ***DB_Agent***, per a gestionar bases de dades.
- la classe ***Table_Agent***, per a gestionar les taules de la bases de dades.

Des d'aquestes classes es fa la connexió amb el motor *SQLite* i es d'on es crida la classe *SQLiteDatabase* i els seus mètodes.

Ja que l'*SQLite* permet disposar de base de dades en memòria, es dissenyen dues bases de dades, una en memòria i una a disc. Es creen, doncs, dues classes especialitzades de la classe *DB_Agent* amb aquesta finalitat:

- la classe ***DBMemory_Agent*** gestiona la base de dades temporal ubicada en memòria que existeix mentre el programari s'estigui executant. Al residir en memòria, aquesta base de dades permet un accés molt ràpid a les dades, que la fa idònia per a gestionar les peticions en cua, les peticions pendents d'identificació i els usuaris que estan en sessió (aquests dos últims casos es detallen a *l'apartat 4.2.1*).
- la classe ***DBProxy_Agent*** gestiona la BD *Proxy* que inclou la resta de dades i està a disc.

Les dades es guarden en taules. Per a gestionar les diverses taules es creen classes especialitzades de la classe *Table_Agent*, les quals s'especifiquen en el següent apartat.

Aquestes classes i la seves relacions es mostren a la *figura 4.2*.

Per a gestionar la informació continguda en fitxers plans, com és el cas de la configuració del Proxy, es crea la classe ***File_Agent*** que conté les crides a les funcions de PHP adequades per a tractar aquests tipus de dades.

La classe *File_Agent* té les mateixes operacions que la classe *Table_Agent* per tal de facilitar-ne la gestió.

[1] Per a més informació http://www.php.net
[2] Per a més informació http://sqlite.org

4.2.1. Definició de l'estructura de base de dades.

Les classes d'entitat persistents (*apartat 3.3.1*) són el punt de partida per a obtenir la definició de l'estructura de base de dades.

Les dades de l'aplicació es divideixen en 5 grups. Les que gestionen els usuaris, les sessions, les peticions, el filtre i la configuració del proxy.

Gestió dels usuaris.

Les classes d'entitat corresponents són *Usuari* (superclasse), *UsuariInternet*, *Gestor* i *Administrador*. Entre elles existeix una relació d'herència que cal suprimir. Per a fer-ho, s'afegeix l'atribut *tipus* que identifica el tipus d'usuari.

Així, doncs, es crea la taula ***Usuaris*** amb els atributs: *identificador, contrasenya, codi, nom, tipus, area* i *suspès*. La clau principal és l'atribut *identificador*.

La taula *Usuaris* forma part de la BD *Proxy* i conté les dades dels usuaris que interactuen amb l'aplicació. Per a la seva gestió, es crea la classe especialitzada ***Users_Agent***.

Gestió de les sessions.

La classe d'entitat corresponent és la classe *Sessio*.

Es crea la taula ***Sessions*** amb els atributs: *IPmaquina, idUsuari, dataInici* i *dataFi*. La clau principal està formada pels atributs *IPmaquina, idUsuari* i *dataInici*.

La taula *Sessions* forma part de la BD *Proxy* i conté les dades de les sessions o connexions que han efectuat els usuaris. Es crea la classe especialitzada ***Sessions_Agent***, per a la seva gestió.

Per a millorar el rendiment de l'aplicació, es crea la taula ***UsuarisEnSessio*** en memòria. Aquesta taula permet saber quins usuaris estan en sessió, en un moment determinat, i evita consultar aquesta informació a disc. Es situa en memòria ja que si el *proxy* està aturat no hi ha cap usuari en sessió i, per tant, no hi ha dades a guardar.

La taula ***UsuarisEnSessio*** té els atributs: *IPmaquina, idUsuari, dataInici* i *dataUltPeticio* (atribut que en l'anàlisi està situat a la classe *Sessio*). La clau principal és l'atribut *IPmaquina* ja que només hi pot haver un usuari en sessió per màquina, en un moment determinat.

La taula *UsuarisEnSessio* forma part de la BD en memòria. Per a la seva gestió, es crea la classe especialitzada ***UsersInSession_Agent***.

Gestió de les peticions.

Les classes d'entitat corresponents són *PeticioEnCua* i *PeticioHistorial*.

Per a les peticions que estan en cua, es crea la taula ***CuaPeticions*** amb els atributs: *handle*, *dataPeticio, uri, idUsuari, IPmaquina, host* i *peticio*. La clau principal està formada pels atributs *handle, dataPeticio* i *uri*. L'atribut *handle* identifica la connexió i l'atribut *peticio* conté el codi *HTTP* de la sol·licitud que ha efectuat l'usuari.

La taula *CuaPeticions* forma part de la BD en memòria. Per a la seva gestió, es crea la classe especialitzada ***QueueRequests_Agent***.

Quan un usuari efectua una petició i no està identificat, l'aplicació li sol·licita que s'identifiqui. Una vegada està identificat, se li serveix la petició que havia efectuat anteriorment. Per a guardar aquesta informació, es crea la taula ***PeticionsEnEspera*** amb els atributs: *IPmaquina* i *peticio*. La clau principal és l'atribut *IPmaquina*.

La taula *PeticionsEnEspera* forma part de la BD en memòria. Per a la seva gestió, es crea la classe especialitzada ***WaitingRequests_Agent***.

Per a disposar del registre històric de totes les peticions, es crea la taula ***Peticions*** amb els atributs: *idUsuari, IPmaquina, dataPeticio, uri, host, permis, categoria, motiu* i *kbytes*. La clau principal està formada pels atributs *idUsuari, IPmaquina, dataPeticio* i *uri*. L'atribut *permis* identifica si la petició ha estat permesa o denegada. En el cas de ser denegada, l'atribut *categoria* indica el tipus de contingut de la web i l'atribut *motiu* indica el perquè no s'ha permès (llista negra, etiqueta PIC o paraula clau). En el cas d'haver-se permès la petició, l'atribut *kbytes* indica la mida del recurs.

La taula *Peticions* forma part de la BD *Proxy. P*er a la seva gestió, es crea la classe especialitzada ***RequestsRegister_Agent***.

Gestió del filtre.

Les classes d'entitat corresponents són *Categoria, HostPermes, HostProhibit, EtiquetaPIC* i *ParaulaClau.*

Per a les categories, es crea la taula ***Categories*** amb els atributs: *identificador* i *nom*. La clau principal és l'atribut *identificador.* Es crea la classe especialitzada ***Categories_Agent***, per a la seva gestió.

Per a indicar els hosts que estan expressament permesos, es crea la taula ***LlistaBlanca*** amb l'atribut *host*. La clau principal és l'atribut *host*. Per a la seva gestió, es crea la classe especialitzada ***WhiteList_Agent***.

Per a indicar els hosts que estan expressament prohibits, es crea la taula ***LlistaNegra*** amb els atributs: *host, categoria* i *estat*. La clau principal és l'atribut *host.* L'atribut *estat* indica si el host està confirmat o pendent de confirmar. Per a la seva gestió, es crea la classe especialitzada ***BlackList_Agent***.

NOTA: quan un host és denegat per contenir paraules clau dins les seves pàgines web, s'afegeix a la llista negra amb estat "pendent de confirmar". A partir d'aquest moment, les peticions que s'efectuïn del host seran denegades. Si l'administrador confirma que el host no ha de ser permès, es passa a estat "confirmat"; en cas contrari, es treu el host de la llista negra i es pot posar a la llista blanca perquè no digui denegat en endavant.

Per a indicar les etiquetes PIC, es crea la taula ***Etiquetes*** amb els atributs: *tipus, codi, categoria* i *descripcio*. La clau principal està formada pels atributs *tipus* i *codi.* Per a la seva gestió, es crea la classe especialitzada ***PICS_Agent***.

Per a indicar les paraules o frases clau, es crea la taula ***ParaulesClau*** amb els atributs: *paraula, categoria* i *pes*. La clau principal és l'atribut *paraula.* Per a la seva gestió, es crea la classe especialitzada ***KeyWords_Agent***.

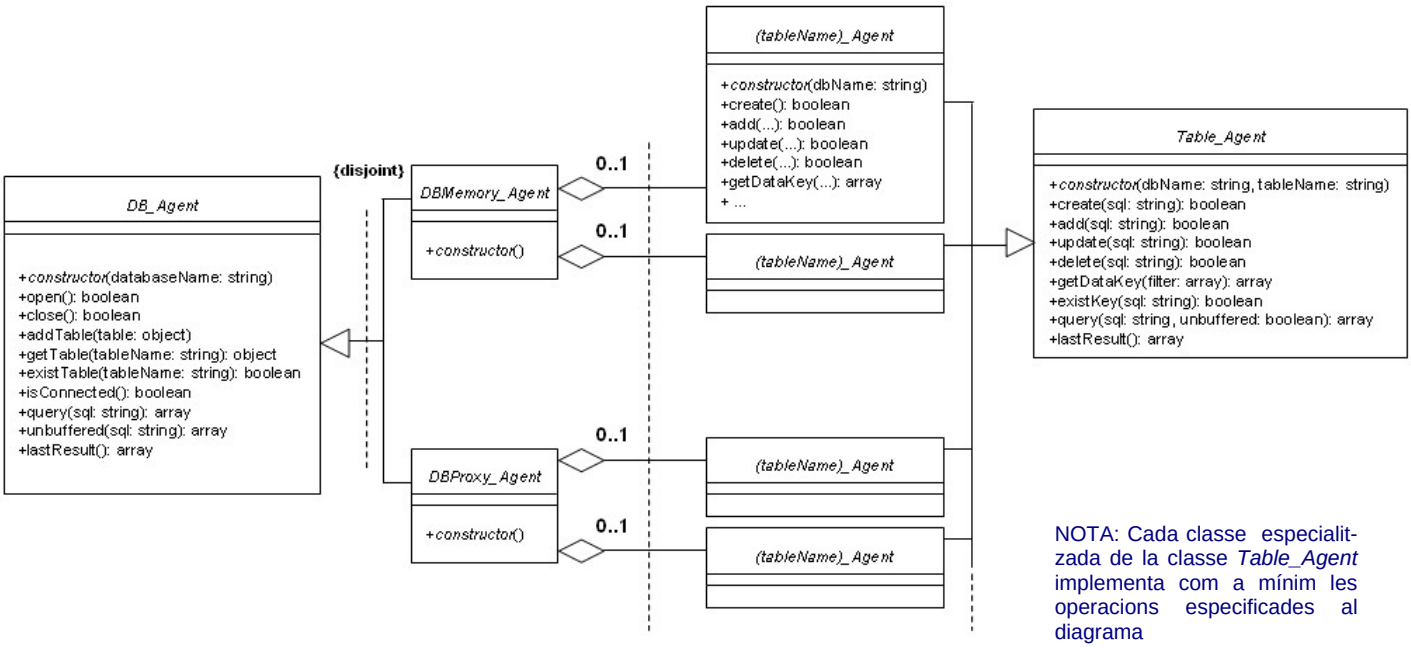
Al fer proves de *testing* (ja que s'ha utilitzat un procés iteratiu en el desenvolupament del projecte), s'ha observat que una consulta efectuada des d'un cercador, pe. *Google*, podia ser denegada per contenir paraules clau i, per tant, el host ser afegit a la llista negra. Per a evitar aquest fet, però denegar només la resposta, donat que es poden efectuar consultes de contingut permès des de qualsevol cercador, es crea la taula ***Cercadors*** que conté el nom del cercador. Per tant, la taula disposa de l'atribut *nom*, que alhora n'és la clau principal. Per a la seva gestió, es crea la classe especialitzada ***Searchers_Agent***.

Totes aquestes taules formen part de la BD *Proxy.*

Gestió de la configuració del *proxy*.

La classe d'entitat corresponent és la classe *ConfProxy.*

Al ser una informació única, la configuració del *proxy* es guarda en un fitxer pla. Per a la seva gestió, es crea la classe especialitzada ***ConfProxy_Agent***.



NOTA: Cada classe especialitzada de la classe *Table_Agent* implementa com a mínim les operacions especificades al diagrama

Figura 4.2 Classes gestores primàries del model de base de dades

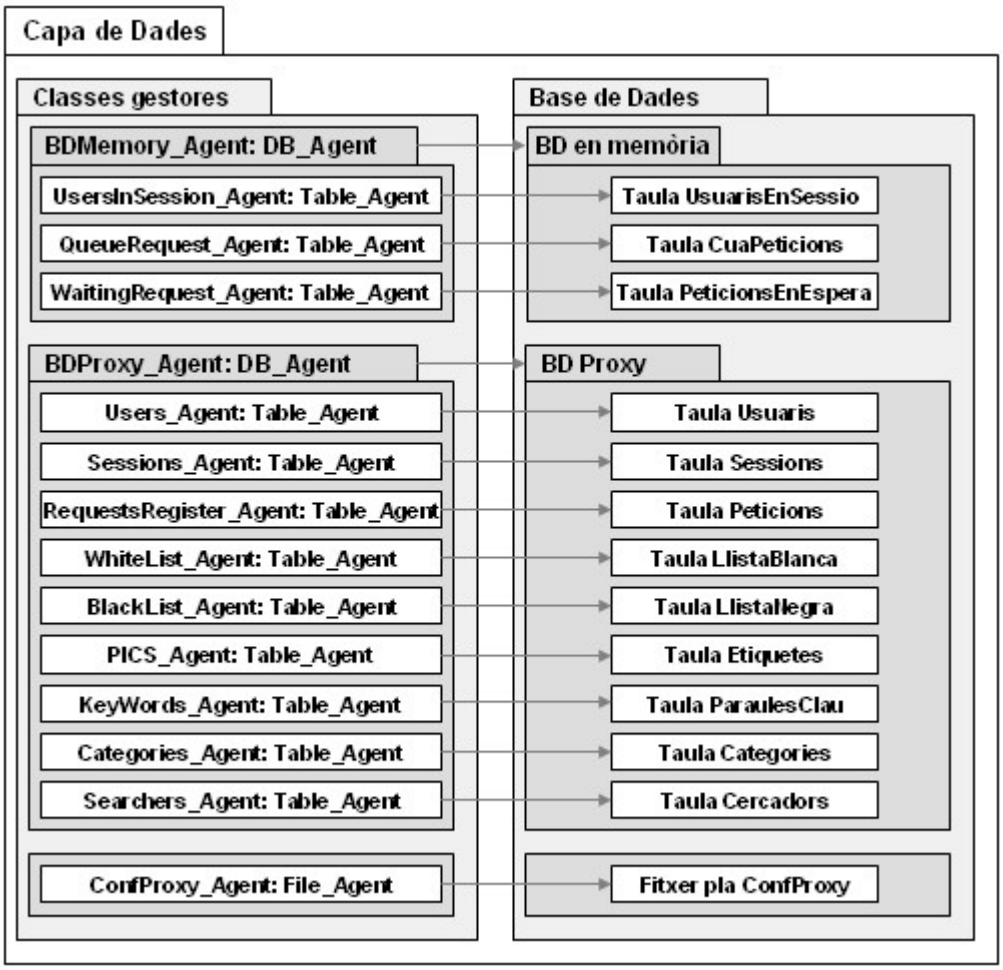


Figura 4.3 Capa de dades. Classes gestores

4.3. Disseny de la lògica de negoci.

4.3.1. Control del flux de la informació. *Sockets*.

Una de les tasques principals de l'aplicació és el control del flux de la informació entre el navegador de l'usuari i Internet. Aquest és el nucli de funcionament del *proxy*.

Amb aquest objectiu i per a facilitar la reutilització del disseny i desenvolupament del projecte, es crea la classe ***Socket*** que conté les crides a la biblioteca de funcions de que disposa el llenguatge PHP per a treballar amb *sockets*.

A més, es creen les classes especialitzades ***Server, Client, TCPServer, TCPClient*** i ***HTTPClient***. Tot seguit, es mostren aquestes classes i la seva relació d'especialització.

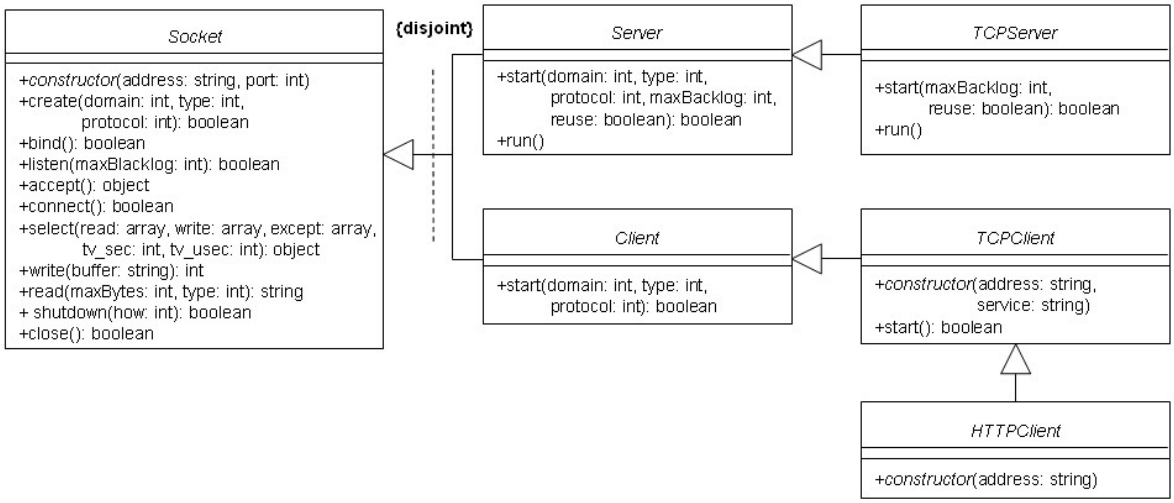


Figura 4.4 Classes per al control del flux de la informació

4.3.2. Interpretació de les peticions de recursos web. Protocols *HTTP* i *HTML*.

Una vegada es disposa del control del flux de la informació, cal interpretar-la. Així doncs, cal desxifrar el codi de les peticions de recursos web efectuades a través del protocol *HTTP* i el de les seves respostes. A més, per a efectuar el filtrat de paraules clau de la resposta cal entendre el protocol *HTML*.

Amb aquest objectiu, es creen les classes ***HTTP*** i ***HTML*** per a interpretar aquests protocols. Per a facilitar, encara més, l'anàlisi d'una petició o resposta d'un recurs web, es crea la classe ***HTTPdocument*** la qual dóna informació precisa del seu contingut.

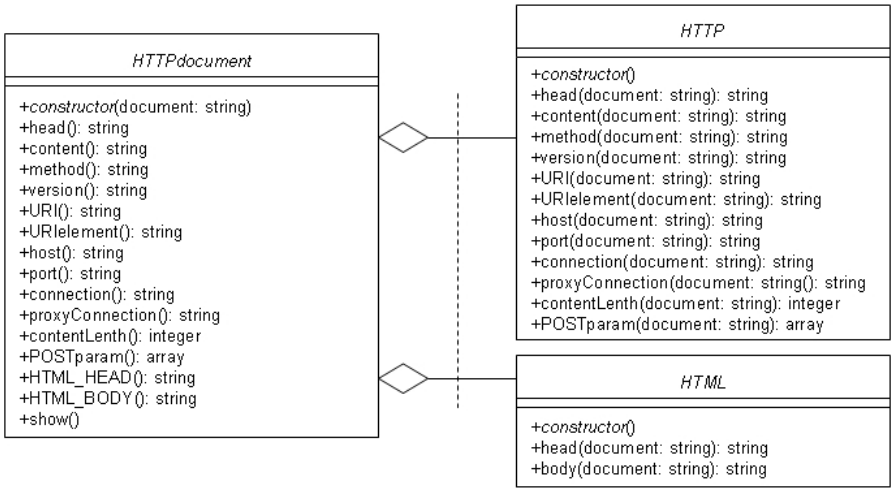


Figura 4.5 Classes interpretació protocols HTTP i HTML

4.3.3. El disseny del casos d'ús.

Les classes de control i d'entitat (*apartat 3.3.1*) són el punt de partida per a obtenir la definició de les classes que composaran la lògica de negoci.

Cas d'ús número 1: "Iniciar sessió"

Les classes de control corresponents són *Proxy* i *Identificacio.* Aquest cas d'ús s'inclou en el disseny del *cas d'ús número 2 "Fer petició".*

Cas d'ús número 2: "Fer petició"

És el cas d'ús més important i nucli de l'aplicació. La classe ***Proxy*** controla tot el cas d'ús.

Classe ***Proxy***	
Conté un objecte de les classes *ProxyServer, ProxyClient, ProxyQueueRequests* i *ProxyJobsAgent*.	
+constructor(address: string, port: integer) +start(maxBacklog: integer): boolean +run(): boolean	Adreça i port on escolta el proxy. Inicialitza el proxy. Nombre màxim de connexions. El proxy queda escoltant, accepta peticions i les processa.

Classe ***ProxyServer: TCPServer*** (gestiona la part servidora del proxy)	
Conté un objecte de les classes *ProxyServerClients*.	
+constructor(address: string, port: integer) +start(maxBacklog: integer): boolean +accept(): socket +receive(queue: ProxyQueueRequests): boolean +send(request: ProxyRequest): array +closeByTimeout(): array	Adreça i port on escolta el servidor. Inicialitza el servidor. Nombre màxim de connexions. Accepta una nova connexió. Retorna el nou socket. Rep una nova petició i la posa a la cua. Envia la resposta. Si cal tancar la connexió, retorna el client tancat. Es tanquen els clients que no han fet peticions. Es retornen els clients tancats.

Classe ***ProxyClient: HTTPClient*** (gestiona la part client del proxy)	
+constructor(proxyAddress: string, nextProxyAddress: string, nextProxyPort: integer) +send(request: ProxyRequest): boolean	Adreça on demanar la petició. Si hi ha un altre proxy intermedi s'indica la seva adreça i port. Envia la petició i rep la resposta.

Classe ***ProxyQueueRequests*** (permet la gestió de la cua) ***Enllaça amb la capa de dades**	
Conté un objecte de les classes *QueueRequest*.	
+constructor() +add(socket: Socket, dateRequest: time, request: ProxyRequest, idUser: int) +delete(handle: int, dateRequest: time, uri: string)	Crea la cua. Afegeix una petició a la cua. Esborra una petició de la cua.

Classe ***ProxyJobsAgent*** (gestiona els serveis del proxy –filtre, registre i servidor web[1]-, la llista d'usuaris en sessió, les sessions i les peticions que estan en espera d'identificació de l'usuari)	
Conté un objecte de les classes *ProxyRequest, ProxyUsers, ProxyUsersInSessions, ProxySessions, ProxyWaitingRequest, ProxyWhiteList, ProxyBlackList, ProxyPICS, ProxyKeyWords, ProxyRequestsRegister* i *ProxyWebServer*	
+constructor(address: string) +readRequest(queue: ProxyQueueRequests):	Crea l'ibjecte gestor de tasques. Llegeix la petició de la cua i la treu.

ProxyRequest +process(param: int): object +checkSession(): boolean +addSession(): boolean +deleteUsers(clientsClose: array)	 Executa un procés depenent de paràmetre. Verifica si l'usuari està en sessió. Afegeix un nou usuari en sessió. Treu els usuaris en sessió associats a les connexions tancades. Posa data/hora fi de sessió a Sessions.

La resta de classes s'enuncien i es poden consultar més àmpliament al codi de l'aplicació.

Classe	*Tasca*
ProxyServerClients	Gestiona les connexions de la part servidora del proxy.
ProxyRequest	Conté la informació de la petició que s'està processant
ProxyUsers *	Permet la consulta i gestió dels usuaris.
ProxyUsersInSession *	Permet la consulta i gestió dels usuaris en sessió.
ProxySessions *	Permet la consulta i gestió de les sessions.
ProxyWaitingRequest *	Permet la consulta i gestió de les peticions pendents.
ProxyWhiteList *	Permet la consulta i gestió de la llista blanca.
ProxyBlackList *	Permet la consulta i gestió de la llista negra.
ProxyPICS*	Permet la consulta i gestió de les etiquetes PIC.
ProxyKeyWords *	Permet la consulta i gestió de les paraules clau.
ProxyRequestsRegister *	Permet la gestió del registre històric.
ProxyWebServer	Gestiona la crida a les pàgines pròpies.

* Aquestes classes enllacen amb la capa de dades, mitjançant crides a la base de dades corresponent.

[1] L'aplicació serveix les pàgines d'identificació, manteniment de les dades, errors, ... des del propi proxy, de manera que s'incorpora un nou servei al proxy, el de **servidor web**. Les peticions identificades amb l'adreça on escolta el *proxy* són desviades, per aquest, al servidor web que les gestiona i dóna resposta.

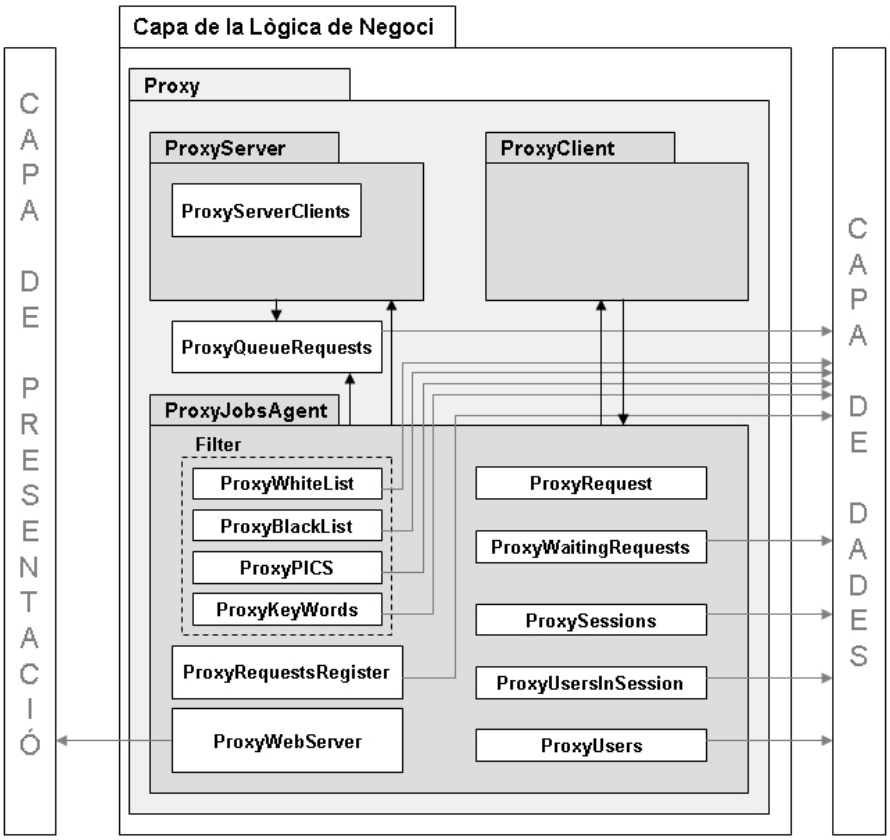


Figura 4.6 Capa de la lògica de negoci. Classes

Cas d'ús número 8 "Tancar sessions"

La classe ***Proxy*** gestiona el cas d'ús amb la col·laboració de la classe ***ProxyServer*** i de la classe ***ProxyJobsAgent***.

Per a fer-ho, la classe ***ProxyServer*** disposa de l'operació ***closeByTimeout()*** que cancel·la les connexions d'usuaris que han superat el temps de *Timeout* sense efectuar cap petició.

La classe ***ProxyJobsAgent*** disposa de l'operació ***deleteUsers(clientsClose)*** que esborra el client tancat de la taula *UsuarisEnSessio*, posa data de fi de connexió a la sessió oberta de la taula *Sessions* (que en guarda un històric) i n'esborra les peticions en cua o pendents.

En base a l'eficiència, la classe *Proxy* crida l'operació *ProxyServer:closeByTimeout* cada un cert temps, especificat en la configuració de l'aplicació. Per acabar la tasca, tot seguit, crida l'operació *ProxyJobsAgent:deleteUsers(clientsClose)*.

Cas d'ús número 10 "Sortir de sessió"

La classe ***ProxyJobsAgent*** administra el cas d'ús.

Quan l'usuari selecciona l'opció de menú *Sortir*, s'envia una petició, que es gestionada per la classe ProxyJobsAgent amb la seva operació ***deleteUsers(clientsClose)***, ja comentada al cas d'ús anterior.

La connexió és cancel·la per *Timeout* tal com s'explica en el cas d'ús anterior.

Casos d'ús números 3, 4, 5, 6, 7 i 9

La resta de casos d'ús són gestionats per la classe ***ProxyWebServer*** que és qui s'encarrega de servir les pàgines pròpies de l'aplicació, per a poder realitzar la identificació (cas d'ús núm. 1), el manteniment i consulta de les dades i la configuració de l'aplicació.

La classe *ProxyWebServer* realitza aquestes tasques amb l'ajut de les classes amb prefix ***wServer***. Per a cada cas d'ús, es crea una classe amb aquest prefix la qual gestiona les crides **GET** i **POST** referents a les dades que administra. Cada classe disposa de les operacions *get()* i *post()*, a tal efecte. Aquestes classes criden a les classes *Proxy[Taula]* respectives, mitjançant les quals s'obtenen les dades per a les consultes o es subministren per a realitzar el manteniment de les mateixes. També criden les classes de la interfície gràfica, subministrant-les a la classe *ProxyWebServer*.

A continuació es relacionen les classes *wServer*.

Classe	*Tasca*
wServerRegisterAgent	Permet la consulta del registre històric (cas d'ús 3).
wServerUsersAgent	Permet la consulta i gestió dels usuaris (casos d'ús 4 i5) i efectuar el canvi de contrasenya (cas d'ús 9).
wServerWhiteList	Permet la consulta i gestió de la llista blanca (cas d'ús 6).
wServerBlackList	Permet la consulta i gestió de la llista negra (cas d'ús 6).
wServerPICS	Permet la consulta i gestió de les etiquetes PIC (cas d'ús 6).
wServerKeyWords	Permet la consulta i gestió de les paraules clau (cas d'ús 6).
wServerProxy	Permet efectuar la configuració de l'aplicació, el seguiment de l'activitat (sessions actives i històriques) i obtenir-ne estadístiques (cas d'ús 7).

4.4. Disseny de la interfície gràfica.

4.4.1. La interfície gràfica: HTML.

Com s'ha esmentat, la interfície amb l'usuari serà gestionada pel propi *proxy*, qui a través del servidor web que incorpora, serveix la interfície en format *HTML*.

El disseny de tota la interfície segueix el mateix patró: cada pantalla a presentar a l'usuari està codificada en llenguatge HTML i en format plantilla. S'ha definit l'etiqueta {nom de camp o literal} que és substituïda pel valor corresponent abans de presentar la pantalla a l'usuari. A més, per a simplificar el disseny de la presentació, s'utilitzen fulls d'estil en cascada (CSS).

Per tant, una interfície gràfica determinada comporta una plantilla d'*HTML*, un full d'estil en cascada comú a l'aplicació i una classe gestora.

Per exemple, la interfície gràfica que es presenta a l'usuari per a la identificació consta de la plantilla ***Login.html***, el full d'estil en cascada ***Proxy.css*** i la classe gestora de la plantilla ***TemplateLogin_Agent***.

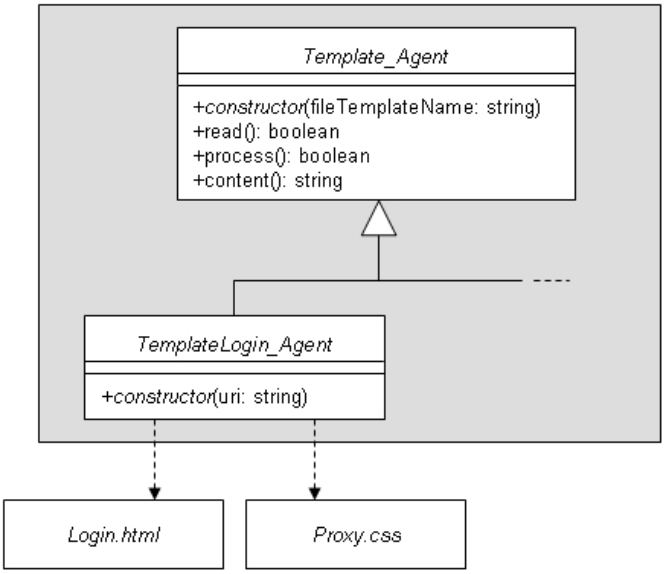


Figura 4.7 Components de la interfície gràfica

4.4.2. Descripció de l'estructura de la interfície.

L'aplicació disposa de dos tipus de plantilles bàsiques:

- la plantilla ***Login.htm***, que es la utilitzada en el cas d'ús *núm. 1 "Iniciar sessió"* i en la que es sol·licita la identificació d'usuari i la contrasenya (*figura 4.8)*.
  Com s'ha comentat, la classe ***TemplateLogin_Agent*** n'és la gestora.
- la plantilla ***Main.htm***, que inclou el menú principal del programa ***GestorProxy*** i dins de la qual s'hi presenta la interfície de la resta de casos d'ús que en disposen (*figura 4.9)*. És gestionada per la classe ***TemplateMain_Agent***.

Per a gestionar les interfícies dels diferents casos d'ús es creen diverses classes amb el prefix ***Template***, totes elles hereves de la classe *TemplateMain_Agent* i que disposen de dues operacions bàsiques, abstractes en la superclasse:

- l'operació ***dataQuery()*** que genera el codi HTML per a presentar un taula amb les dades sol·licitades i diversos controls per a que l'usuari hi pugui interactuar.
- l'operació ***dataForm()*** que genera el codi HTML per a presentar els diversos formularis necessaris per a efectuar el manteniment de les dades.

Tot seguit es relacionen les classes gestores de la interfície:

Classe	*Tasca*
TemplateRegister_Agent	Gestiona les interfícies del cas d'ús 3 "Consultar historial".
TemplateUsers_Agent	Gestiona les interfícies dels casos d'ús 4 i 5 (gestió usuaris). Inclou la interfícies del cas d'ús 9 "Canviar contrasenya".
TemplateWhiteList_Agent	Gestiona part de les interfícies del cas d'ús 6. Llista Blanca.
TemplateBlackList_Agent	Gestiona part de les interfícies del cas d'ús 6. Llista Negra.
TemplatePICS_Agent	Gestiona part de les interfícies del cas d'ús 6. Etiquetes PIC.
TemplateKeyWords_Agent	Gestiona part de les interfícies del cas d'ús 6. Paraules Clau.
TemplateProxy_Agent	Gestiona les interfícies del cas d'ús 7 "Gestionar Proxy".

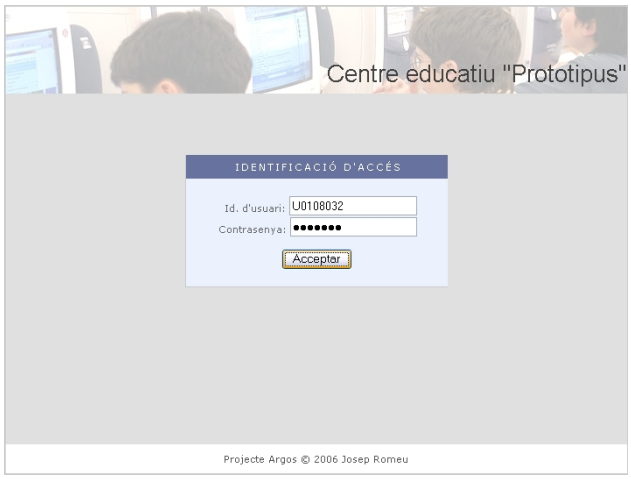


Figura 4.8 Disseny plantilla Login.htm

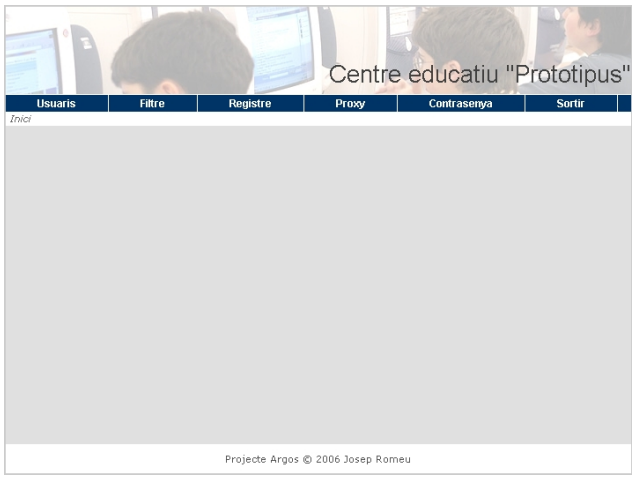


Figura 4.9 Disseny plantilla Main.htm

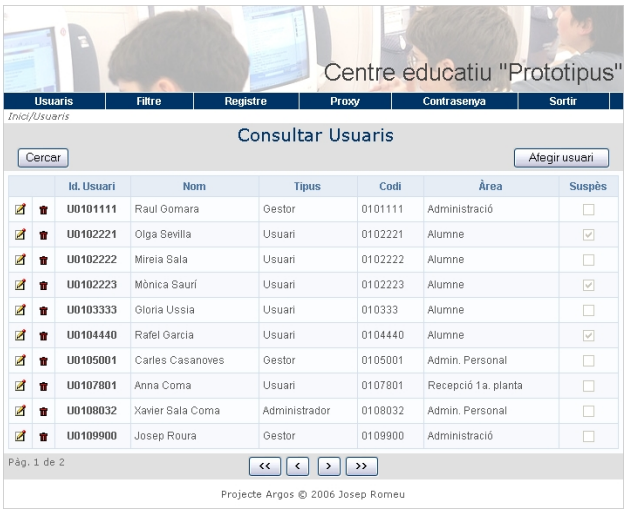


Figura 4.10
Disseny interfície generada per dataQuery()

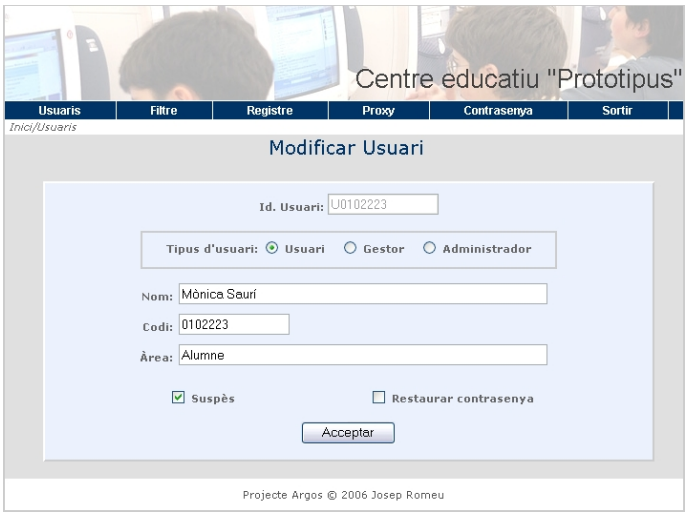


Figura 4.11
Disseny interfície generada per dataForm()

4.5. Disseny dels subsistemes.

Com s'ha especificat, el disseny es basa en la programació per capes. Cadascuna d'aquestes capes configura un subsistema compost per diversos components que es relacionen funcionalment i que, en alguns casos, tenen dependències uns dels altres.

Per tal de facilitar la substitució d'un subsistema o component per un altre que implementi la mateixa interfície, es mostren els diversos subsistemes i les seves dependències.

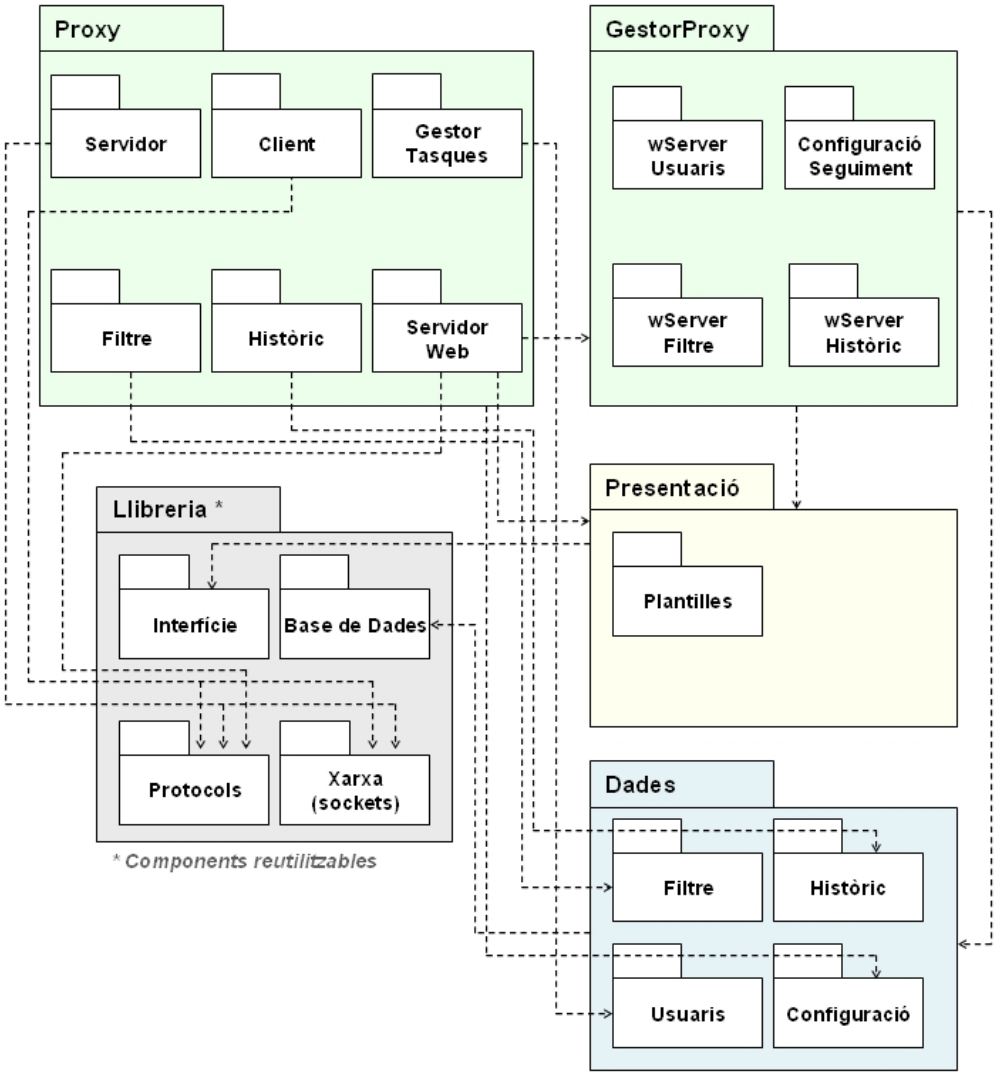


Figura 4.12 Disseny dels subsistemes i les seves dependències.

A més dels subsistemes de **Dades**, **Lògica de negoci** (***Proxy*** i ***Gestor Proxy***) i **Presentació**, s'ha afegit la **Llibreria**, on hi ha els components i classes reutilitzables.

- Els subsistemes *Dades, Lògica de negoci* i *Presentació* tenen una relació de dependència d'algun component de la *Llibreria*.
- El subsistema *Proxy* depèn del subsistema *Dades* ja que necessita consultar les seves dades per a poder efectuar les seves diverses tasques.
- El component *ServidorWeb* depèn del subsistema *GestorProxy* ja que aquest últim li subministra la interfície gràfica de les pàgines pròpies (programa de gestió i manteniment del *Proxy*). També depèn directament del subsistema *Presentació* per a obtenir la pantalla d'identificació.
- El subsistema *GestorProxy* depèn del subsistema *Dades* ja que necessita les dades per a poder generar les pàgines pròpies.

Capítol 5. Implementació i testing.

Per a facilitar la implementació, aquesta s'ha efectuat en fases, començant per resoldre les funcionalitats més bàsiques fins arribar a disposar d'un prototipus operatiu que disposa de les principals tasques esmentades en el projecte.

S'ha seguit un procés iteratiu, de manera que anàlisi, disseny i implementació s'han anat autoalimentant.

S'ha procurat aplicar amb cura el disseny especificat en el capítol anterior, a no ser que per raons de funcionament dels protocols de comunicacions o per limitacions del llenguatge del programació escollit, s'hagi hagut d'implementar de forma diferent. Si la desviació ha estat per manca de coneixements de l'autor, aquest ha procurat adquirir la informació necessària per a renovar el disseny de la part afectada i corregir-la. De tota manera, s'indica si s'ha donat el cas.

Per a la implementació només s'ha utilitzat el producte ***PHP v. 5.2.0*** que, com s'ha esmentat, inclou el gestor de base de dades ***SQLite***. Per tant, amb la instal·lació d'aquest producte n'hi ha prou per a que el prototipus funcioni.

5.1. Fases de la implementació.

La implementació s'ha realitzat des de zero, iniciant-la amb la programació dels subsistemes més bàsics per al funcionament de l'aplicació.

- **Fase 1. Control del flux d'informació de la xarxa. *Proxy* bàsic.**
- **Fase 2. Subsistema bases de dades.**
- **Fase 3. *Proxy* primari. Enllaç amb les bases de dades.**
- **Fase 4. Generació interfície gràfica.**
- **Fase 5. Prototipus *Proxy*. Servei WebServer.**

5.1.1 Fase 1. Control del flux d'informació. Proxy bàsic.

Aquesta fase és i ha estat fonamental per a la implementació del projecte, doncs, el seu objectiu principal és aconseguir controlar el flux d'informació que s'esdevé entre el navegador i la Xarxa, quan l'usuari efectua una petició d'un recurs.

És l'etapa més problemàtica ja que comporta saber com interactuen el navegador (client) i el servidor i quins missatges s'envien entre ells.

Com s'especifica en el disseny, s'implementa la ***classe Socket*** que encapsula les funcions *socket* de *PHP*, i la resta de classes especificades fins a disposar de:

- la ***classe TCPServer*** que implementa les funcionalitats d'un *servidor TCP* capaç d'establir connexions amb *clients TCP*, i que suporta els protocols de nivell d'aplicació *HTTP*, ...
- la ***classe HTTPClient*** que implementa un client que permet enviar i rebre peticions a la Xarxa.

Amb les classes bàsiques operatives, es programa una primera versió de la ***classe Proxy*** per a construir un *proxy* simple.

La funcionalitat principal del *proxy* s'implementa amb l'operació ***Proxy:run()***. Es tracta d'un bucle infinit que accepta i llegeix peticions de clients, les posa a la cua, en treu la primera que ha entrat (a la cua), envia la sol·licitud a destí, rep la resposta i la reenvia al client peticionari.

Aquesta era la funcionalitat a aconseguir: Poder controlar les peticions i les respostes, en format HTTP, i ésser transparent a l'usuari, és a dir, no fer cap transformació de la informació enviada o rebuda, simplement, actuar de passarel·la.

A continuació es mostra el codi de la primera versió de l'operació ***Proxy:run()***.

```
function run() {
   // S'inicialitza l'hora per a efectuar l'eliminació de clients
   // sense peticions.
   $this->lastDeleteClients = time();
   // Es gestionen les peticions dels clients.
   while (true) {
      // Si toca, s'eliminen els clients que no han efectuat
      // peticions en l'interval establert (_TIMEOUT_CLIENT);
      if ((time() - $this->lastDeleteClients) >= _INTERVAL_DELETE_CLIENTS) {
         $this->deleteClients();
      }
      // Hi ha peticions? Es posen a la cua.
      if (!$this->acceptAndReadTCPclients()) {
         return false;
      }
      // Es demana la 1a. petició.
      if (count($this->a_request) > 0 ) {
         $index = 0;
         $this->tcpClientHandle = $this->a_request[$index][_RQ_HANDLE];
         $this->request = $this->a_request[$index][_RQ_DATA];
         // Es demana la 1a. petició a destí.
         $retSend = $this->send();
         //
         $this->writeTCPclients($retSend);
         // Si tot ha anat, bé, es borra la petició processada.
         $tmp = array();
         for($index = 1; $index < count($this->a_request); $index++)
            $tmp[] = $this->a_request[$index];
         $this->a_request = array();
         for($index = 0; $index < count($tmp); $index++)
            $this->a_request[] = $tmp[$index];
      }
   }
   // Tancar totes les connexions, correctament.
   return true;
}
```

També s'implementa la ***classe HTTP*** que resol la parametrització dels missatges d'aquest protocol i que permet, de manera senzilla, obtenir-los i avaluar-los per, com exemple, fer canvis necessaris respecte la petició origen, que es guarda, de la que és enviada a través del *client proxy* a la Xarxa i, viceversa.

El *client proxy* sempre envia les peticions amb tipus de connexió *close* i tanca cada vegada la connexió amb els servidors de la Xarxa. Si la petició origen s'ha fet amb tipus de connexió *keep-alive*, és a dir, oberta per a altres peticions, el *client proxy* ha de modificar el tipus de connexió a *close*, el que implica poder substituir el paràmetre adient. Una vegada rebuda la resposta, ja que s'ha guardat l'original, es torna a posar el tipus de connexió sol·licitat pel client peticionari.

Com a última puntualització d'aquesta fase, cal dir que a nivell d'entorn, s'ha hagut de

- configurar en el programa, l'adreça:port on escolta el *proxy* i, si hi ha un altre *proxy* intermediari, configurar l'adreça:port on el *client proxy* enviarà les peticions.
- configurar els navegadors per a que enviïn les peticions a l'adreça:port on escolta el programa *proxy*.

5.1.2 Fase 2. Subsistema de base de dades.

La primera tasca d'aquesta fase ha estat encapsular les funcions *sqlite* de *PHP* per a gestionar de manera còmoda el motor de base de dades ***SQLite*** i per a fer independent la gestió de les dades de l'aplicació del tipus de motor de bases de dades. A tal efecte, s'han implementat la ***classe DB_Agent*** i la ***classe Table_Agent***.

No hi ha hagut complicacions en la implementació del subsistema de base de dades, ja que l'SQLite es gestiona de manera molt senzilla, amb sentencies SQL.

S'ha seguit el disseny proposat, implementant les classes gestores especificades, a excepció de:

- la gestió de la informació de les *peticions en espera*. Segons el disseny aquesta informació està ubicada en una taula de la BD en memòria, i hauria d'estar gestionada per la *classe WaitingRequests*. En canvi, s'ha implementat en una variable de tipus *array*, obtenint-ne la mateix funcionalitat i permetent a l'autor poder experimentar amb la gestió d'un *array* com si fos una taula.
  Així doncs, la ***classe ProxyWaitingRequests***, en compte de fer crides a una taula, fa crides a un *array*, essent transparent a la resta de l'aplicació, ja que les operacions tenen el mateix nom que les de la resta, la qual cosa exemplifica un dels molts beneficis de la programació orientada a objectes.
- la gestió de la informació referent als *cercadors*.

  Abans de res, explicar el perquè disposar d'aquesta informació.
  Al fer proves amb el prototipus i fer una cerca de paraules de tipus prohibit, l'eina de filtrar va afegir el cercador *Google* a la llista negra. Com a filtre el resultat va ser òptim, ja que l'usuari no ha de veure la informació de cerca, però no s'ha de posar el cercador a la llista negra, ja que impedeix fer qualsevol altra consulta amb el cercador. Així, doncs, en un cas com aquest, no es mostra la informació a l'usuari però es permet seguir utilitzant el cercador, si aquest està a la llista de cercadors.

  Per la poca quantitat d'*items* que es gestionen, en la versió del prototipus actual s'ha guardat aquesta informació en un arxiu pla, de fàcil edició.
  Per homogeneïtzació i seguretat es pot implementar com una taula, en futures versions.

5.1.3 Fase 3. Proxy primari. Enllaç amb les base de dades.

En base al *proxy* simple, construït a la 1a. fase, s'implementen els serveis de filtrat i registre, objectiu principal del projecte.

La programació d'aquests serveis presenta una problemàtica a resoldre en relació a atendre les diverses peticions que van arribant al *proxy*, mentre aquest està processant una petició, ja que, en aquesta versió de *PHP*, encara no es permès executar processos fills.
La solució trobada és intercalar el procés de comprovació de possibles peticions entre les diverses tasques que efectua el *proxy* al processar una petició.
En resum, l'operació ***Proxy:run()*** que controla el procés principal del *proxy* disposa d'un bucle infinit que realitza les següents tasques (entre parèntesis s'indica l'encarregat de dur-ho a terme):

- Eliminar clients per timeout (Server).
- *Comprovar peticions i posar-les a la cua (Server).*
- Treure petició de la cua, per a processar-la (Gestor de tasques).
- *Comprovar peticions i posar-les a la cua (Server).*
- Comprovar si l'usuari està en sessió (Gestor de tasques)
- *Comprovar peticions i posar-les a la cua (Server).*
- Efectuar pre-filtrat a la llista Blanca i Negra (Gestor de tasques – Mòdul Filtre)
- *Comprovar peticions i posar-les a la cua (Server).*
- Obtenir la petició de la Xarxa (Client).
- *Comprovar peticions i posar-les a la cua (Server).*
- Efectuar post-filtrat per etiquetes PIC i paraules clau (Gestor de tasques – Mòdul Filtre)
- *Comprovar peticions i posar-les a la cua (Server).*
- Enviar resposta al client peticionari (Server).
- *Comprovar peticions i posar-les a la cua (Server).*
- Registrar la petició al registre històric (Gestor de tasques – Mòdul Registre)

Mentre el *proxy* realitza aquestes tasques, es consulten i modifiquen dades. La implementació de l'accés al subsistema de base de dades s'ha efectuat tal com s'indica en el disseny i no ha presentat cap complicació.

5.1.4 Fase 4. Generació interfície gràfica.

La implementació de la interfície, basada en les classes ***Template*** i les plantilles *HTML*, no ha presentat cap problemàtica, més enllà del propi disseny gràfic i de tenir en compte la seva usabilitat i senzillesa d'ús.

Per a la seva realització s'ha utilitzat llenguatge ***HTML***, fulls d'estil en cascada (***CSS***) i scripts en *JavaScript*, per a controlar petits processos des de la màquina client.

Les crides entre la ***classe WebServer*** i les ***classes Template***, s'ha programat seguint el disseny especificat.

5.1.5 Fase 5. Prototipus Proxy. Servei Web Server.

En el disseny ja es va afegir el servei WebServer amb la finalitat de no utilitzar cap més producte en la realització del prototipus.

La seva implementació tampoc ha presentat una excessiva complexitat.

La tasca més delicada, però, ha estat com resoldre el procés d'identificació. El procés es fonamenta en presentar el recurs sol·licitat, per l'usuari, prèviament a la seva identificació una vegada s'hagi identificat. S'explica tot seguit amb un exemple:

- Un usuari sol·licita el recurs http://www.google.es.
- Es rep una petició HTTP amb la següent capçalera ***GET*** *http://www.google.es* ***HTTP/1.1***
- Es verifica si l'usuari està en sessió. Si no ho està, es guarda la petició a *WaitingRequest* i es redirecciona la petició a la pàgina local **login.php**, enviant una resposta ***HTTP 303 See Other*** al client.

```
HTTP/1.1 303 See Other
Date: Mon, 15 Jan 2007 20:29:07 GMT
Server: Proxy-Argos 1.0
Connection: close
Location: http://192.168.1.128/login.php
```

- El navegador envia, automàticament, la sol·licitud ***GET*** *http://192.168.1.128/login.php* ***HTTP/1.1***
- El *WebServer* serveix la pàgina d'identificació, *http://192.168.1.128/login.php*.
- Quan l'usuari s'identifica, es rep una petició **POST** amb les dades de la identificació.
- Si tot ha anat bé, el *WebServer* genera una altra resposta de redirecció, en aquest cas, a la petició prèvia.

```
HTTP/1.1 303 See Other
Date: Mon, 15 Jan 2007 20:29:16 GMT
Server: Proxy-Argos 1.0
Connection: close
Location: http://www.google.es
```

- El navegador envia, automàticament, la sol·licitud ***GET*** *http://www.google.es* ***HTTP/1.1***
- Es verifica si l'usuari està en sessió. En aquest cas, ja ho està i es demana el recurs corresponent a la Xarxa, seguint el procés normal.

5.2. Estructura dels components del prototipus.

Els diversos recursos (programes, classes, plantilles, imatges, arxius de configuració, ...) utilitzats pel prototipus s'han estructurat en forma d'arbre dins de carpetes, per a facilitar la seva identificació.

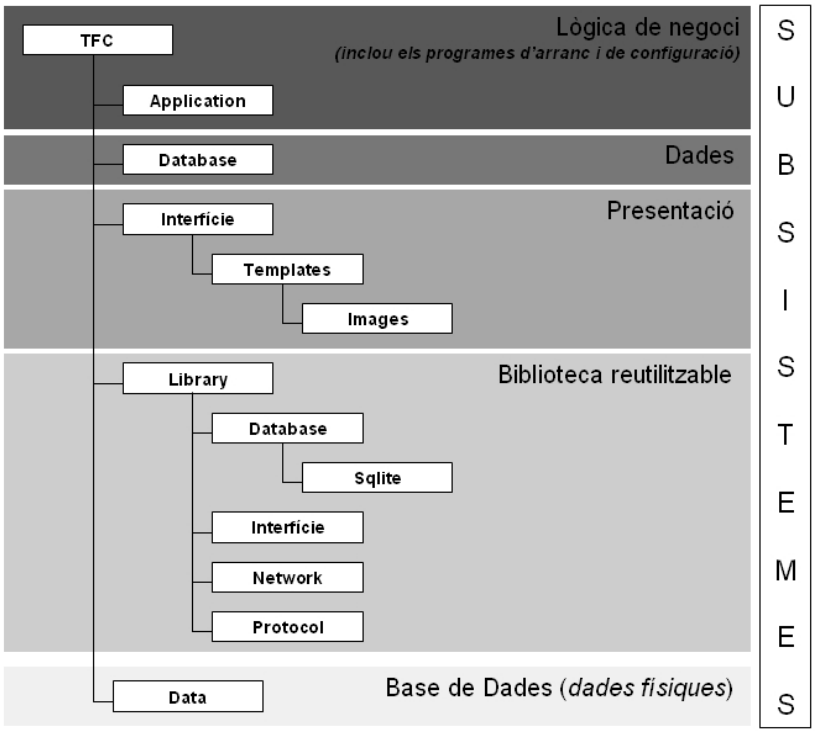


Figura 5.1 Arbre de l'estructura dels components del prototipus

5.3. Testing.

Les diverses proves efectuades, durant tot el desenvolupament del prototipus, han permès detectar els errors i depurar-los.

S'han realitzat proves de funcionament en cada fase de la implementació, per tal de minimitzar els possibles errors.

Per a la realització de les proves, s'han afegit instruccions al codi les quals mostren el valor de variables, estat dels objectes, ... (algunes d'elles encara consten, com a comentari, en el codi font). En casos concrets, com en el subsistema de dades, s'han creat petits programes per cadascuna de les taules (veure scripts *test.table[nom_de_la_taula.php* al codi font).

En la fase final, s'han realitzat proves, utilitzant usuaris i un entorn de xarxa real, per a verificar el correcte funcionament del prototipus (identificació, peticions, accés a la part gestora, manteniment d'usuaris, dades de filtrat, registre de sessions i històric de peticions), depurant els errors que s'han produït o posant missatge d'error per a no confondre l'usuari.

Les proves s'han efectuat exclusivament en entorn Windows XP. El prototipus ha funcionant prou bé, en una màquina amb un microprocessador Pentium a 500Mhz.

5.4. Estat actual i futures millores.

El prototipus actual es plenament operatiu quan a rebre i respondre peticions *HTTP*. Manca implementar el suport al protocol *HTTPS*.

En futures versions, l'autor té la intenció d'implementar aquest protocol, a més, de programar el canvi de contrasenya, la configuració del proxy des de l'entorn del programari i dissenyar i implementar un seguiment adequat que permeti disposar d'estadístiques de les peticions que realitzen els usuaris.

Capítol 6. Conclusions

L'autor ha pogut constatar la vertadera dificultat i complexitat per a realitzar un filtratge acurat dels continguts web, tot i que espera que futures versions del prototipus puguin ser més rigoroses, abastant més temàtiques i webs de diversos idiomes. Avui per avui no hi ha cap programa de filtrat que pugui garantir de manera infalible el filtratge de continguts web.
Sistemes d'etiquetatge PICS com els impulsats per les associacions *ICRA* i *RSAC* són un bon començament per a poder classificar els tipus de contingut de les pàgina web i, així, facilitar el seu filtrat.

Es constata que amb un bon anàlisi i disseny, la implementació és molt més senzilla i fàcil de dur a terme i que el model vàlid és el de la iteració en tot el procés. No cal dir que la utilització de les tècniques de programació orientades a objectes fan més entenedor tot el procés i faciliten la implementació i reutilització.

El rendiment del llenguatge PHP i la "facilitat" amb que s'ha construït el programari, ha sorprès gratament l'autor, ja que el fet que la biblioteca de funcions *socket* del llenguatge es trobi encara en fase d'experimentació (segons els seus autors), ha estat un repte afegit per a la consecució del projecte.

La simplicitat del protocol HTTP i la utilització de pocs paràmetres d'aquest protocol en el prototipus resultant ha impressionat l'autor.
Tot el contrari, amb la utilització dels full d'estil en cascada (CSS) molt més complexos ja que la interpretació dels mateixos varia d'uns navegadors a uns altres.

Per acabar, dir que la il·lusió amb que l'autor va escollir el projecte, s'ha anat incrementant en la mesura que aquest ha anat avançant i fent-se realitat. L'autor està satisfet d'haver-lo pogut realitzar, alhora que el seu aprenentatge s'ha enriquit notablement.

Agrair a la consultora els ànims en cadascun dels seus missatges, que m'han ajudat en la realització d'aquest projecte.

I finalment, agrair a la meva dona Magda la paciència que ha tingut amb mi al llarg de tota la carrera d'Enginyeria Tècnica de Gestió a la Universitat Oberta de Catalunya.

Glossari

buffer *m* Ubicació de la memòria en un ordinador reservada per a l'emmagatzemament temporal d'informació, en espera de ser processada.

client-servidor *m* Arquitectura informàtica on es divideixen les responsabilitats d'un sistema d'informació, separant la interfície d'usuari de la gestió de la informació. Bàsicament, consisteix en un programa (client) que efectua peticions a un altre programa (servidor) que li dóna resposta.

extensió de fitxer *m* Cadena de caràcters annexada al nom d'un fitxer i, usualment, separada d'aquest per mitjà d'un punt. La seva finalitat és diferenciar el contingut del fitxer perquè el sistema operatiu pugui executar-lo o interpretar-lo.

ICRA *f* Sigles d'*Internet Content Rating Association*. Organització internacional sense ànim de lucre que reuneix les empreses líders d'Internet amb la finalitat de desenvolupar una Xarxa més segura.

PICS *f* Sigles de *Platform for Internet Content Selection*. Plataforma endegada pel World Wide Web Consortium amb la finalitat de desenvolupar un sistema de filtrat basat en protocols. El sistema PICS permet la creació d'etiquetes que informen del tipus de contingut d'una pàgina web.

programació orientada a objectes *f* Paradigma de programació que defineix els programes en termes de "classes d'objectes".

programació per capes *f* Estil de programació que té com a objectiu primordial dividir els components primaris de l'aplicació, programar-los i després unir-los en temps d'execució. Destaca la programació en tres capes: dades, lògica de negoci i presentació.

programari lliure *m* Programari que pot ser usat, copiat, estudiat, modificat i redistribuït lliurement.

programari multiplataforma *m* Capacitat o característica que té un programari per a poder funcionar de forma similar en diferents sistemes operatius o plataformes.

protocol *m* Conjunt de regles que regulen un procés o activitat. En l'entorn informàtic i de telecomunicacions, estàndard o acord que regula la connexió, la comunicació i la transferència de dades entre dos sistemes.

proxy *m* Mot usat en el sentit d'intermediari. L'ús més comú és el de *servidor proxy*, i fa referència a un ordinador o programa que intercepta las connexions dins una xarxa.

RSAC *m* Sigles de *Recreational Software Advisory Council.* Organització internacional sense ànim de lucre nascuda amb el propòsit de protegir els menors dels continguts d'Internet no aptes per a ells.

socket *m* Concepte abstracte mitjançant el qual dos programes poden intercanviar qualsevol flux de dades, generalment, de manera fiable i ordenada.

SUSI *f* Sigles de *Safer Use of Services on the Internet.* Projecte que té com a objectiu informar a pares i professors de les magnífiques oportunitats que ofereix Internet i, alhora, advertir dels aspectes menys desitjables.

UML *m* Sigles d'*Unified Modeling Language.* Lenguatge gràfic per al modelat d'aplicacions o de processos orientats a objectes.

URI *m* Sigles *d'Uniform Resource Identifier*. Seqüència de caràcters usada a Internet, d'acord a un format estàndard, per a identificar inequívocament un recurs, sigui text o multimèdia

URL *m* Sigles *d'Uniform Resource Locator.* Seqüència de caràcters usada a Internet, d'acord a un format estàndard, per a localitzar inequívocament un recurs, sigui text o multimèdia.

Bibliografia

Barceló Ordinas, José Mª i **Iñigo Griera, Jordi** (2005). *TCP/IP: els protocols de la xarxa Internet.* UOC Fundació per a la Universitat Oberta de Catalunya

Campderrich Falgueras, Benet (2004). *Enginyeria del programari.* UOC Fundació per a la Universitat Oberta de Catalunya

ICRA (2006). *Working for a safer internet* [online] Disponible a http://www.icra.org

IQUA (2006). *Internet segura* [online] Disponible a http://www.iqua.net

Martí Escalé, Xavier i **Perramon Tornil, Xavier** (2005). *Aplicacions Internet.* UOC Fundació per a la Universitat Oberta de Catalunya

Peig Olivé, Enric i **Perramon Tornil, Xavier** (2005). *Programació de sockets.* UOC Fundació per a la Universitat Oberta de Catalunya

PHP (2006) PHP:Hypertext Preprocessor [online] Disponible a http://php.net/

SafeSurf (2006). *Making the Net Safe* [online] Disponible a http://www.safesurf.com

SQLite (2006). SQLite home page [online]. Disponible a http://sqlite.org/

SUSI (2006). *Be Safe Online* [online] Disponible a http://www.besafeonline.org

Villate, Javier (2001) *Libertad de expresión en Internet*. Disponible a l'arxiu de l'Observatori per a la CiberSocietat a http://www.cibersociedad.net/archivo/articulo.php?art=37

W3C (2006) W3C Search, Site Index, Keywords [online] Disponible a http://www.w3.org/Consortium/siteindex

Xhafa, Fatos (2002). *Tècniques de desenvolupament de programari.* UOC Fundació per a la Universitat Oberta de Catalunya