DESENVOLUPAMENT D'UNA APLICACIÓ PER AL FILTRAT DE PÀGINES WEB

MEMÒRIA

Estudiant: Daniel Nielles Sanchez
ETIS

Consultora: Maria Isabel March Hermo

19 de juny del 2006

Resum

El present Treball de Final de Carrera presenta un servidor proxy fàcilment configurable per a filtrar de pàgines web el contingut de les quals es vol prohibir a un usuari. L'aplicació ha estat desenvolupada en Java mitjançant les classes Socket i ServerSocket. L'aplicació incorpora la funcionalitat de confecció d'estadístiques del tràfic a través del proxy provinent de la internet i provinent del browser. L'usuari pot configurar el servidor proxy, és a dir, configurar les adreces a les quals es prohibeix l'accés, introduint comandes a través del browser. També pot introduir comandes per a visualitzar les estadístiques de la xarxa i la configuració del proxy. Aquestes comandes componen el protocol que en aquest treball anomenarem Protocol TFC.

Continguts

Índex de figures

1-Introducció

Fora de l'àmbit privat de cada persona adulta (és a dir, a casa de cadascú) és freqüent la necessitat de filtrar determinades adreces de pàgines web.

Un exemple clar el trobem en el món de l'empresa, on es prohibeix a determinats empleats consultar determinades pàgines web per tal d'evitar pèrdues de productivitat tant de l'empleat com del sistema d'informació. Un cas semblant el trobem en centres públics que permeten el lliure accés a internet, per exemple biblioteques o universitats, en què pot ser necessari vetllar per a que els recursos informàtics no s'ocupin en consultes innecessàries a internet i estiguin disponibles per a les activitats a què estan destinats: estudi i recerca.

És freqüent, per exemple, la prohibició de consultar pàgines amb contingut eròtic, pornogràfic, violent o xenòfob o la descàrrega d'arxius MP3 mitjançant aplicacions tals com Kazaa o eMule. Sense entrar fer un judici moral d'aquests continguts, són exemples clars d'informació que, en general, no aporta res a l'activitat de l'empresa pública o privada però que, en canvi, consumeix recursos i sobrecarrega els sistemes d'informació, provocant pèrdues de rendiment del sistema. Aquestes pèrdues de rendiment es tradueixen en pèrdua de productivitat dels empleats d'una empresa, insatisfacció dels usuaris del sistema i en majors costos d'administració del sistema.

En general, és possible plantejar-se la prohibició de l'accés a continguts que consumeixen molts recursos (perquè és visitat amb molta freqüència o bé perquè implica grans transaccions d'informació), sense que hi hagi justificació a permetre-hi accés.

Per una altra banda, també cal tenir en compte el cas en què els menors d'edat tenen accés a Internet, ja sigui en centres escolars o a la llar, i els pares o els educadors volen evitar que accedeixin a determinats tipus de continguts o facin servir determinades aplicacions (FTP, e-mail, etc.).

Per aquestes raons, té sentit disposar d'una aplicació de control fàcilment configurable que permeti filtrar les pàgines web a què es pot accedir i que permeti controlar en quines adreces es consumeixen els recursos del sistema, és a dir, que enregistri quines pàgines han estat visitades i quant de temps ha estat consumit en les visites.

La funcionalitat addicional d'enregistrar el temps consumit a cada pàgina visitada és una eina potent per a la feina d'administració d'una xarxa o per a decidir si cal prohibir l'accés a una pàgina web determinada. En quedar l'adreça enregistrada, permet accedir al recurs i jutjar si és necessari permetre-hi l'accés des del nostre sistema, i d'altra banda permet veure el grau d'utilització d'aquest recurs, tant en temps de transacció d'informació com en nombre d'accessos al recurs.

2-Objectius del treball

Es tracta de desenvolupar una aplicació de control fàcilment configurable que permeti filtrar les pàgines web a què es pot accedir i que permeti controlar en quines adreces es consumeixen els recursos del sistema. És a dir, es tracta de prohibir l'accés a determinades adreces http i d'emmagatzemar dades relatives a les pàgines visitades i al temps consumit en elles.

La implementació d'aquesta aplicació es farà mitjançant un servidor proxy per al filtratge de pàgines web que el servidor pot mostrar i aplicacions que es poden fer servir des del navegador d'acord amb l'esquema següent:

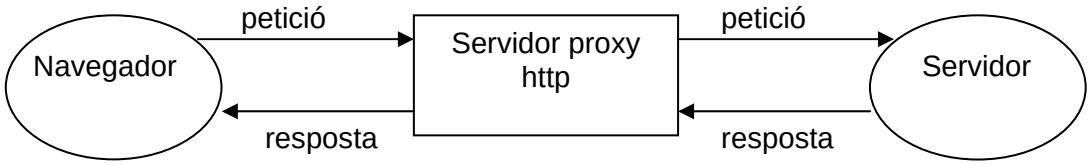

Figura 1: Esquema del servidor proxy

La configuració d'aquest servidor haurà de ser posible mitjançant comandes enviades al navegador. Per tant, caldrà crear un protocol que permeti la comunicació amb el servidor mitjançant comandes com ara "afegir www.adreça.com" o "eliminar www.adreça.com".

D'altra banda caldrà programar aquest servidor per tal que emmagatzemi estadístiques de les pàgines visitades o les pàgines que s'han intentat visitar per a permetre una anàlisi posterior. Les dades seran emmagatzemades en fitxers. Aquestes dades seran les següents: Adreça, Accés denegat S/N, Nombre de visites, Temps total de connexió. El fitxer podrà ser consultat mitjançant qualsevol processador de text, haurà de ser possible la seva visualització des del navegador enviant una comanda i haurà de ser possible esborrar-ne el contingut des del navegador mitjançant una comanda. Aquest conjunt de comandes estaran recollides en un protocol que el servidor reconeixerà i que anomenarem Protocol TFC.

L'aplicatiu haurà de ser fàcil d'utilitzar per qualsevol persona no familiaritzada amb la informàtica a qui li siguin explicades les instruccions d'instal·lació i les comandes del protocol. Les instruccions d'instal·lació i les comandes hauran de ser, per tant, simples, intuïtives i fàcils de recordar.

L'aplicació serà instal·lable en qualsevol PC i plataforma.

2.1-Aportació del TFC a l'estat de l'art del filtratge de pàgines web

Es tracta d'oferir una aplicació multi-plataforma per a un PC, gratuïta, amb un grau màxim de simplificació d'ús i que concentri les funcionalitats esmentades: filtre d'adreces URL d'accés no desitjat i confecció d'estadístiques de xarxa.

La confecció d'estadístiques de xarxa és un cas típic d'aplicació de servidors proxy i és d'ús generalitzat en l'administració de xarxes.

Symantec i altres fabricants d'antivirus i software de seguretat en xarxes de computadors ofereixen software per a servidors amb filtratge d'URL, en què es decideix denegar accessos a pàgines sospitoses de posar en perill la seguretat del sistema. Aquests programes també ofereixen a l'usuari la possibilitat de configurar el servidor per a denegar l'accés a una URL determinada.

El software SmartFilter (http://www.winproxy.com/products/sitefilter.shtml) del fabricant BlueCoat ofereix un programari de pagament integrable en el paquet WinProxy que incorpora la funció de filtratge d'adreces URL per a evitar l'accés a llocs web indesitjats amb un conjunt extens de comandes per a una detallada configuració de les polítiques d'accés a internet. No conté la funcionalitat de confecció d'estadístiques.

Un objectiu fonamental del TFC és, per tant, oferir un programari que permeti a l'usuari vincular fàcilment les adreces dels llocs web visitats amb la configuració del servidor proxy. Amb aquesta finalitat, l'aplicació ha d'oferir a l'usuari una presentació clara dels llocs web on es consumeixen els recursos del sistema per a decidir els llocs web als quals no es permet l'accés i una configuració senzilla del filtratge d'adreces d'internet.

3-Mètode seguit per al desenvolupament de l'aplicació i planificació

3.1-Punt de partida

L'estudiant compta amb l'experiència en programació en Java i xarxes de computadors adquirida en l'estudi de la diplomatura en ETIS. L'estudiant no compta amb experiència laboral en programació o en xarxes de computadors.

L'estudiant va triar la temàtica del TFC motivat per l'interès amb què havia seguit les assignatures relacionades amb les xarxes de computadors al llarg dels estudis i es va decidir per un dels temes proposats en el Pla docent de l'assignatura de TFC.

D'especial utilitat a la realització del TFC ha estat haver realitzat la pràctica 'Programació de sockets' de l'assignatura Xarxes de Computadors, que ha servit de punt de partida al TFC.

3.2-Fases del treball

El Treball de Final de Carrera ha estat estructurat en les següents fases:

3.2.1-Fase preliminar

Durada: del 28 de febrer al 8 de març del 2006.

Activitats realitzades:

- Aproximació a la temàtica del TFC.
- Reestudi de conceptes de xarxes de computadors.
- Reestudi de conceptes de Java

Documentació resultant:

- Document 'Pla de Treball' (equivalent a la PAC1).
- Especificació textual de l'aplicació a desenvolupar.

Es va decidir programar en Java perquè, a part que Java assegura la portabilitat desitjada a l'aplicació, l'avantatge de desenvolupar el programari en Java és l'existència de classes de Java per a la programació de sockets. La classe Socket de Java implementa tota la funcionalitat dels sockets en l'espai de noms d'internet. La classe ServerSocket està pensada per a ser usada a l'extrem de la comunicació del servidor, connectar el socket a una adreça i gestionar les peticions de connexió. A més, Java ofereix un ampli ventall de classes (PrintStreams, BufferedReader, etc.) per a processar els Streams de bytes en trànsit en una connexió a internet.

Els esmentats avantatges es tradueixen en un menor nombre de línies de codi i en una fàcil depuració del codi. A aquests avantatges hem d'unir l'existència d'eines IDE per a Java molt avançades i a innumerables fòrums de programadors en Java,

fets que garanteixen una implementació ràpida de l'aplicació i el compliments dels terminis de lliurament.

Com que l'estudiant havia llegit articles sobre tècniques com Extreme Programming, en què es recomana codificar al llarg de totes les fases del projecte, va semblar adient seguir aquestes recomanacions. D'aquesta manera, en la fase preliminar es va començar a codificar amb la intenció de refrescar Java i d'aprofundir en el coneixement de les classes Socket i ServerSocket de Java. Concretament, es va especificar un servidor proxy que atengués les trucades d'un client, i les enviés a un servidor extern si les adreces d'internet demanades no estaven contingudes en un fitxer d'adreces prohibides.

D'altra banda es van decidir les funcionalitats del programari i ja es va decidir que, a part de les funcions habituals de tot servidor intermediari, serien les següents:

- Prohibir l'accés a una pàgina determinada.
- Alliberar l'accés a una pàgina.
- Permetre consultar les pàgines a les quals es prohibeix l'accés.
- Emmagatzemar estadístiques.
- Permetre visualitzar les estadístiques emmagatzemades.
- Permetre reiniciar (posar a zero) les estadístiques emmagatzemades

Es deixava per a la fase d'especificació decidir si s'integraven totes les funcionalitats en el servidor proxy o bé s'implementaven programes a part, per exemple, per a afegir adreces prohibides.

Ja en aquesta fase es va decidir que, donat el poc volum i complexitat de les dades amb què caldria treballar, s'emmagatzemarien les dades en fitxers, concretament dos fitxers: un per a emmagatzematr les adreces prohibides i un altre per a emmagatzemar les estadístiques.

3.2.2-Especificació i anàlisi

Durada: del 9 de març al 17 d'abril.

Activitats realitzades:

- Estudi d'aplicacions d'internet: llenguatge HTML, protocol HTTP.
- Identificació dels guions de l'aplicació.
- Diagrama de casos d'ús.
- Procés d'el·lecció d'eines per a la realització del TFC i el desenvolupament de l'aplicació.
- Estudi de les llibreries de Java necessàries per al desenvolupament de l'aplicació.
- Definició de les comandes que formaran part del protocol de comunicació amb el servidor.
- Especificació textual de l'aplicació.
- Diagrama de classes.
- Revisió del diagrama de casos d'ús.
- Inici de la codificació del servidor proxy.

Documentació resultant:

- Document ‘Especificació i anàlisi’ (equivalent a la PAC2).

Es va decidir confecionar i fer servir casos d’ús per a l’anàlisi de l’aplicació. Aquesta metodologia és coneguda per l’estudiant perquè és la que s’imparteix en l’assignatura d’ETIS Enginyeria del Programari 1 de la UOC. D’altra banda, els casos d’ús permeten una fàcil traducció en un diagrama de classes, el qual és d’utilitat quan es programa en un llenguatge orientat a objectes com Java.

Es va implementar un proxy amb filtratge de pàgines prohibides. Aquest programa havia de servir de base al programari definitiu, la idea era crear un programa que implementés un servidor proxy amb unes funcionalitats mínimes al qual s’anirien afegint les funcionalitats del programa i que s’aniria refinant amb proves constants. El programa no incorporava ni gestió d’adreces prohibides ni d’estadístiques i consistia en tres classes:

- Proxy, on corria el programa principal i que era, com a la versió final, un servidor que escoltava peticions de connexió pel port 80.
- ProxyFiltre, s’analitzaven les capçaleres HTTP i es decidia si calia prohibir accés a les pàgines demanades. En cas de prohibició, enviava en HTTP un codi 403 (prohibició) i en cas de no prohibició, connectava amb el servidor a la internet.
- Constants, que contenia constants que s’usaven en l’execució del codi.

La implementació d’aquest programa va requerir l’estudi de l’especificació HTTP.

L’experiència recollida amb el programa va servir per a la confecció dels guions, els casos d’ús i el diagrama de classes.

Un procés important en aquesta etapa era plantejar com implementar la gestió del servidor proxy, és a dir, la interacció amb l’usuari per a que aquest passi les comandes necessàries al servidor per a la gestió dels fitxers d’adreces prohibides i d’estadístiques. Es van contemplar les següents possibilitats:

- Un programa a part, implementat en Java, en què l’usuari entrés dades a través d’una GUI.
- Fer el pas de paràmetres via web (CGI), per exemple mitjançant formularis.
- Introduir les comandes a través de la barra d’adreces del navegador.

Es va decidir seguir la darrera opció, ja que el nombre de comandes és reduït, requerint poc esforç memorístic per part de l’usuari per a recordar el nom de les comandes i afegeix un factor de simplicitat d’ús per a l’usuari, en no ser necessari executar un nou programa per a gestionar el servidor. De totes maneres, la decisió final no es prendria fins a la fase de disseny, on es comprovaria la viabilitat i el comfort per a l’usuari de passar paràmetres a través de la barra del navegador.

3.2.3-Disseny

Durada: del 18 d’abril al 29 de maig.

Activitats realitzades:
- Disseny de l’arquitectura de l’aplicació.
- Disseny de les classes de Java.

- Definició de l'estructura general de l'aplicació.
- Codificació de les classes.
- Primer prototipus de l'aplicació.

Documentació resultant:

- Document 'Disseny' (equivalent a la PAC3).

Com a pas previ a tancar el disseny de l'aplicació, es va decidir començar per tirar endavant amb la programació del servidor proxy iniciada a l'etapa d'anàlisi.

D'aquesta manera, es van anar afegint funcionalitats a l'aplicació i implementant les classes que es van detectar a la fase d'anàlisi a mida que era necessari.

Es va començar implementant la gestió del fitxer d'adreces prohibides mitjançant l'entrada de comandes per la barra d'adreces de l'Internet Explorer amb resultats satisfactoris. Només es va detectar el problema que, en entrar una comanda per la barra d'adreces, per exemple, "afegir www.uoc.edu", el que feia Internet Explorer en comptes de passar aquesta comanda al servidor, era suposar un error en l'entrada d'una adreça HTTP i enviar al servidor una petició de connexió a l'adreça http://auto.search.msn.com per a intentar que l'usuari pogués resoldre aquest error triant una adreça que s'aproximés a la sentència "afegir www.uoc.edu". Per a solucionar aquest error es va decidir canviar la sintaxi de les sentències que s'enviaven al servidor especificant que era una sentència que s'enviava al localhost, és a dir, és necessària la següent sintaxi per a una comanda com l'esmentada: http://127.0.0.1/afegir www.uoc.edu.

En implementar la creació i gestió d'estadístiques, es va veure que en fer una petició al servidor, es van creant i destruint sockets al llarg de la transmissió, i es creen sockets amb servidors que el client no ha demanat directament. Per exemple, en demanar una connexió a www.lavanguardia.es, es van creant les següents connexions:

web.lavanguardia.es 1,047 segons
web.lavanguardia.es 5,782 segons
secure-uk.imrworldwide.com 5,969 segons
web.lavanguardia.es 14,953 segons
web.lavanguardia.es 1,234 segons
ad.es.doubleclick.net 0,218 segons
ad.es.doubleclick.net 0,235 segons
www.lavanguardia.es 0,954 segons
www.lavanguardia.es 2,422 segons

Aquest fet provoca una desviació del plantejament inicial, ja que el que es volia era control·lar les adreces demanades per l'usuari.

Es va decidir canviar el plantejament inicial i emmagatzemar totes les connexions que fa el servidor proxy a qualsevol servidor d'internet. Es va decidir fer-ho així perquè d'aquesta manera emmagatzemem més informació rellevant i d'utilitat en la gestió del sistema. Al cap i a la fi, el que interessa és veure quin trànsit consumeix els nostres recursos, i la informació ha de ser el més complerta possible.

En aquest punt del desenvolupament, l'aplicació comptava ja amb les següents classes:

- Proxy, on s'executa el procés principal.
- ProcesProxy, on es fa la gestió del servidor (filtratge d'adreces, confecció d'estadístiques, ...), creant un Thread per cada connexió.
- GestorAdreces, contenint les operacions de gestió del fitxer d'adreces prohibides.
- Estadistica, contenint les operacions de gestió del fitxer d'adreces.
- ProtocolTfc, contenint les operacions a realitzar segons la comanda entrada per l'usuari.
- Constants, interfície que conté constants.

En implementar la comanda per a visualizar la configuració del proxy, la idea era enviar a través del browser una comanda que retornés en HTML al browser el contingut d'un fitxer. Es va intentar implementar d'una manera una mica farragosa, cridant des del programa principal una funció estàtica a la qual li era passat el socket per a que enviés el contingut del fitxer. Per això, es va plantejar fer un canvi en el disseny de l'aplicació i fer un programa separat del proxy que desplegués una GUI on se seleccionés l'operació a fer i que podria també servir per a visualitzar tant la configuració com les estadístiques. Aquesta opció es va descartar perquè no ofereix avantages evidents de comfort i d'economia i hagués suposat una gran inversió de feina quan la gestió passant les comandes per la barra d'adreces del navegador ja estava prou avançada.

La solució que es va triar per a evitar un farragós pas de paràmetres va ser fer que, cada cop que es cridés des de la classe ProcesProxy un mètode de la classe ProtocolTfc, aquest mètode, a més de fer l'operació indicada en cada cas, retornaria un String contenint un missatge en HTML contenint o bé la configuració del proxy, o bé el contingut del fitxer d'estadístiques, o bé un missatge de confirmació d'una operació realitzada.

Per a la confecció d'aquests missatges, es va crear la classe pantalla.

En aquest punt es va tancar la fase de disseny i es va documentar amb el document 'Disseny'.

Quedava per a la fase de proves solucionar problemes de traducció de bytes que provocaven que no es poguessin veure determinats objectes de les pàgines web rebudes. També quedava pendent capturar el port de les peticions HTTP que arribaven al proxy per a fer la connexió, ja que fins a aquest moment s'estaven creant tots els sockets de connexió al port 80 del servidor extern.

3.2.4-Implementació i proves

Durada**:** del 30 de maig al 18 de juny.

Activitats realitzades:
- Implementació i proves unitàries.
- Proves unitàries.
- Redacció de la memòria.
- Redacció del manual de l'aplicació en format HTML.

Lliuraments resultant:
- Memòria del TFC.
- Codi de l'aplicació.

- Fitxer executable de l'aplicació.
- Manual de l'aplicació en format HTML.

Partint del codi obtingut a la fase d'anàlisi, es van afegir noves característiques al codi.

Es va implementar la captura del port al qual es demana la connexió (com s'ha indicat, fins ara es feien totes les connexions al port 80 del servidor d'internet). Es va fer llegint adequadament la URL que arriba en HTTP, que té la següent forma:

http_URL = "http:" "//" host [":" número de port] [path]

Si no es detecta el caràcter ":" que correspon al port, es fa la connexió al port 80, si no, es captura el número de port i es fa la connexió al port sol·licitat.

Es va afegir al procés principal la creació d'una finestra que indica el port pel qual escolta el servidor proxy (80, encara que se'n podria haver triat un altre). Aquesta finestra serveix de recordatori a l'usuari per a que pugui configurar adequadament el navegador d'internet. La finestra desapareix automàticament quan es crea la primera connexió a internet.

Es va solucionar el problema de tractament dels bytes que arriben al servidor proxy de la internet i va millorar l'aparença de les pàgines rebudes, en què ja es processen correctament tots els objectes llevat d'algunes fotografies que no arriben complertes.

Es va intentar solucionar aquest problema amb les fotografies sense èxit. La lectura del Stream que arriba d'internet i la seva escriptura en el Stream cap al browser es fa mitjançant el següent bucle:

```
while(linia_rebuda2 != null){
   sortida1.write((linia_rebuda2 + CR_LF).getBytes(), 0,
Array.getLength((linia_rebuda2 + CR_LF).getBytes()));
   linia_rebuda2 = entrada2.readLine();
}
```

On linia_rebuda2 és un String que llegim del BufferedReader entrada2 que captura el InputStream provinent d'internet i sortida1 és un PrintStream cap al browser.

Com a alternativa, es va implementar el següent:

```
while(fi_llegit != -1){
     fi_llegit = entrada2.read(llegits, 0, 2048);
     sortida1.write(llegits, 0, Array.getLength(llegits));
}
```

On entrada2 és ara un BufferedInputStream i llegits és un array de bytes. El comportament del programa va ser idèntic al cas anterior, pel qual es va decidir no tocar aquesta part de codi.

Probablement el problema es troba en l'altre sentit de la comunicació, és a dir, possiblement caldria refer la comunicació des del browser a la internet per a resoldre el problema.

Es va intentar que es gestionessin correctament la comanda CONNECT d'HTTP, utilitzada quan hi ha intercanvis en SSL. Per falta de temps, no es va poder aconseguir que funcionessin correctament i resta com a una millora per al futur.

Un altre problema a solucionar és la gestió de la resposta HTTP/1.1 301 Moved Permanently. Quan aquesta resposta arriba al browser, aquest és incapaç de redireccionar a la URL indicada en la resposta, probablement per algun problema de format en passar pel proxy.

4-Productes obtinguts

Els productes obtinguts són un executable Proxy.exe per a Windows i plataformes Intel i un fitxer en format HTML Manual.html contenint el manual d'usuari de l'aplicació.

L'executable cal copiar-lo en el disc dur i la seva execució en Windows resulta en l'execució del servidor proxy amb filtratge de pàgines i web i confecció d'estadístiques objecte del TFC.

Per a permetre l'execució del programari en altres plataformes, també es lliura un fitxer Proxy.jar executable mitjançant la comanda "java -jar Proxy.jar" prèvia instal·lació a la màquina d'una JVM.

5-Breu descripció dels altres capítols de la memòria

A continuació es farà una presentació detallada de la documentació i productes obtinguts al final de cada etapa del desenvolupament de l'aplicació.

En l'apartat 'Especificació i anàlisi' s'expliquen les conclusions a què es va arribar a l'etapa d'anàlisi i es presenten les classes d'entitats que es van obtenir.

En l'apartat de 'Disseny' es justifiquen els canvis realitzats en el diagrama de classes obtingut a l'etapa d'anàlisi i es presenta el diagrama de classes definitiu de l'aplicació.

En l'apartat 'Implementació' es presenten les comandes que finalment formen el Protocol TFC de comunicació entre l'usuari i el servidor Proxy per a possibilitar la gestió d'aquest últim. Es mostra també l'aspecte dels fitxers que contenen les estadístiques i les adreces prohibides. Finalment s'incorpora una notació simplificada de la documentació generada per Java de les classes implementades.

En l'apartat 'Entorn de desenvolupament de l'aplicació' s'ennumeren les característiques del maquinari i el programari utilitzats en el desenvolupament de l'aplicació.

Per a completar la descripció de l'aplicació obtinguda, en l'apartat 'Manual de l'usuari' es descriu el funcionament de l'aplicació i s'explica el resultat d'introduir les comandes del Protocol TFC pel browser.

Finalment, en l'apartat 'Conclusions i possibles millores' es repassa la funcionalitat assolida pel programari i s'analitzen problemes i possibles millores i ampliacions de l'aplicació.

6-Especificació i anàlisi

6.1-Guions que existeixen en el domini

L'anàlisi del domini mostra l'existència dels següents guions a partir dels quals es basaran l'especificació i l'anàlisi de l'aplicació:

Guió de l'usuari estàndard

L'usuari estàndard, a través d'un navegador Internet Explorer realitza peticions HTTP a un servidor web, per exemple http://www.uoc.edu/ a través de la línia de comandes del navegador.

Guió de l'usuari privilegiat

L'usuari privilegiat, a més de les funcions de l'usuari estàndard, realitza funcions de configuració del servidor filtre i d'anàlisi de les dades. Per a la configuració d'aquest servidor filtre i l'anàlisi de les dades, l'usuari privilegiat haurà d'enviar comandes al navegador a través de la línia de comandes del navegador. L'usuari privilegiat, mitjançant les comandes del Protocol TFC haurà de ser capaç d'afegir adreces prohibides a la configuració del servidor i haurà de ser capaç d'eliminar adreces prohibides de la configuració per a permetre l'accés als continguts. D'altra banda, l'usuari privilegiat haurà de ser capaç de visualitzar la configuració del servidor. L'usuari privilegiat haurà de ser capaç de consultar les dades emmagatzemades i de reiniciar les dades, és a dir, "posar tots el comptadors a zero".

Guió del servidor proxy

El servidor escolta del port 80. Quan detecta una connexió ha de reconèixer si es tracta d'una petició http o d'una comanda en Protocol TFC. Per a cada petició de connexió http crea un socket i processa el socket. Processar el socket vol dir fer que el navegador mostri una pàgina indicant que la pàgina sol·licitada no es pot mostrar (perquè està prohibida) o bé crear un nou socket del port 80 a la pàgina sol·licitada en el cas que la pàgina no estigui prohibida. En el cas de detectar una comanda en Protocol TFC haurà d'executar la comanda que toqui tal com s'ha explicat en el guió de l'usuari privilegiat. D'altra banda el servidor haurà d'enviar a un fitxer la informació necessària per tal d'emmagatzemar estadístiques de les pàgines visitades o les pàgines que s'han intentat visitar per a permetre una anàlisi posterior per part de l'usuari privilegiat. Les dades que caldrà emmagatzemar són les següents: Adreça, Accés denegat Sí/No, Nombre de visites, Temps total de connexió.

6.2-Diagrama de casos d'ús en la fase d'anàlisi

A partir dels guions extreiem el següent diagrama de casos d'ús:

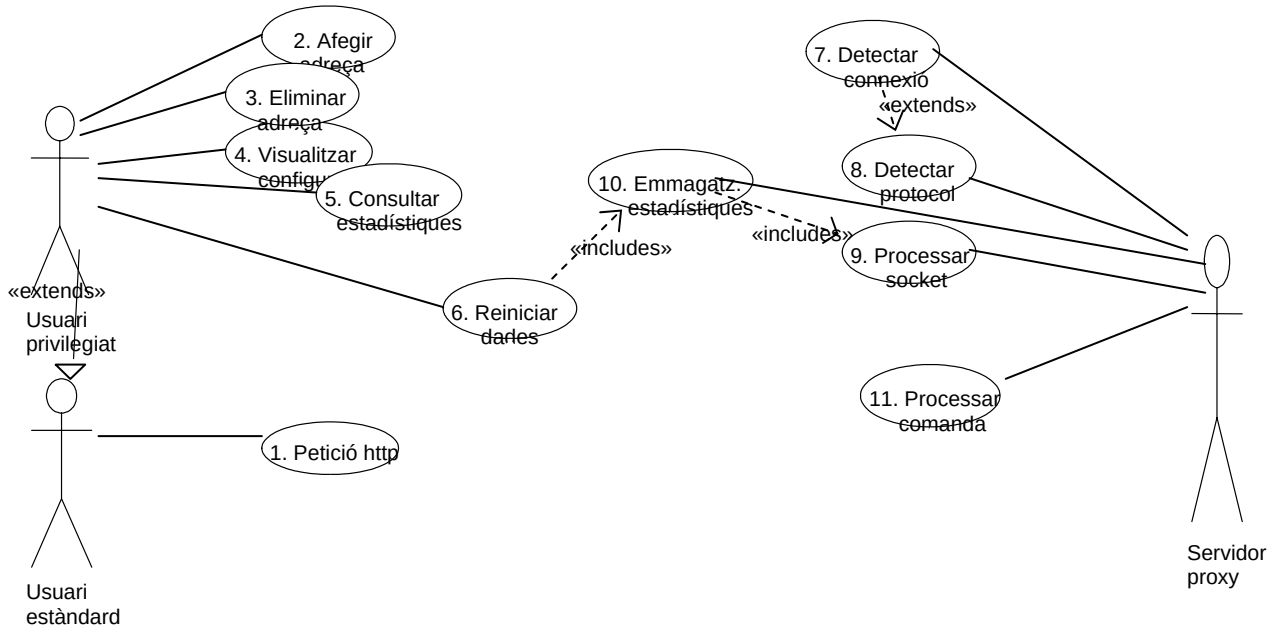

Figura 2: Diagrama de casos d'ús

6.3-Especificació textual dels casos d'ús

1. Petició http

Resum de la funcionalitat: realitza una petició http a través del navegador d'internet.
Actors: usuari estàndard/usuari privilegiat.
Casos d'ús relacionats: cap
Precondició: cap.
Postcondició: una petició de servei HTTP surt del navegador.
Procés normal principal:

1. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
2. L'usuari introdueix una l'adreça de la pàgina que vol visitar en el navegador.
3. L'usuari estàndard espera a rebre la pàgina.

2. Afegir adreça

Resum de la funcionalitat: enviem un missatge en Protocol TFC al servidor proxy per a que introdueixi una adreça HTTP a la llista d'adreces prohibides.
Actors: usuari privilegiat.
Casos d'ús relacionats: cap.
Precondició: cap.
Postcondició: l'adreça ha estat afegida a la llista d'adreces prohibides.
Procés normal principal:

1. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
2. L'usuari introdueix una comanda per a afegir l'adreça.

3. Eliminar adreça

Resum de la funcionalitat: enviem un missatge en Protocol TFC al servidor proxy per a que elimini una adreça http de la llista d'adreces prohibides.
Actors: usuari privilegiat.
Casos d'ús relacionats: cap.
Precondició: cap.
Postcondició: l'adreça ha estat eliminada de la llista d'adreces prohibides.
Procés normal principal:

1. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
2. L'usuari introdueix una comanda per a eliminar l'adreça.

Alternatives de procés i excepcions:

1. L'adreça no existeix a la llista d'adreces prohibides.
    a. El sistema envia, de totes maneres, un missatge a l'usuari indicant que l'adreça ha estat eliminada de la llista d'adreces prohibides.

4. Visualització configuració

Resum de la funcionalitat: enviem un missatge en Protocol TFC al servidor proxy per a visualitzar la llista d'adreces prohibides.

Actors: usuari privilegiat.
Casos d'ús relacionats: cap.
Precondició: cap.
Postcondició: a la pantalla del navegador apareix la llista d'adreces prohibides.
Procés normal principal:

1. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
2. L'usuari introdueix una comanda per a consultar la configuració.

5. Consultar estadístiques

Resum de la funcionalitat: enviem un missatge en Protocol TFC al servidor proxy per a consultar les estadístiques de les adreces visitades.
Actors: usuari privilegiat.
Casos d'ús relacionats: cap.
Precondició: cap.
Postcondició: a la pantalla del navegador apareixen les estadístiques de les adreces visitades.
Procés normal principal:

3. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
4. L'usuari introdueix una comanda per a visualitzar les estadístiques de les adreces visitades.

6. Reiniciar dades

Resum de la funcionalitat: enviem un missatge en Protocol TFC al servidor proxy per a reiniciar el fitxer d'estadístiques de les adreces visitades.
Actors: usuari privilegiat.
Casos d'ús relacionats: relació d'inclusió en el cas d'ús 10. Emmagatzemar estadístiques.
Precondició: cap.
Postcondició: el fitxer d'estadístiques de les adreces visitades ha estat reinicialitzat (posat a zero) o creat si no existia.
Procés normal principal:

1. El sistema executa el cas d'ús 7. Detectar connexió o el cas d'ús 8. Detectar protocol.
2. L'usuari introdueix una comanda per a reinicialitzar el fitxer.

7. Detectar connexió

Resum de la funcionalitat: el servidor proxy escolta del port 80 per a detectar connexions.
Actors: servidor proxy.
Casos d'ús relacionats: 8. Detectar protocol.
Precondició: cap.
Postcondició: el servidor crea una connexió si detecta alguna petició de connexió o segueix escoltant si no detecta cap petició.
Procés normal principal:

1. El sistema inicia l'execució del servidor proxy.
2. El servidor proxy es posa a escoltar.
3. Si detecta una petició de connexió crea un socket.

8. Detectar protocol

Resum de la funcionalitat: el servidor proxy detecta si ha rebut una petició HTTP o una comanda en Protocol TFC d'acord amb els casos d'ús 2. Afegir adreça, 3. Eliminar adreça, 4. Visualització configuració, 5. Consultar estadístiques o 6. Reiniciar dades.
Actors: servidor proxy.
Casos d'ús relacionats: relació d'extensió del cas d'ús 7. Detectar connexió.
Precondició: s'ha creat una connexió en el port 80.
Postcondició: el servidor executa el cas d'ús 9. Processar socket en el cas que rebi una petició http o el cas d'ús 11. Processar comanda en el cas d'haver rebut una petició en Protocol TFC.
Procés normal principal:

1. El servidor proxy executa aquest cas d'ús després d'haver executat el cas d'ús 7. Detectar connexió.
2. El servidor proxy processa el socket o la comanda.

9. Processar socket

Actors: servidor proxy.
Casos d'ús relacionats: 10. Emmagatzemar estadístiques .
Precondició: el sistema ha creat una connexió i ha rebut una petició de servei HTTP.
Postcondició: el sistema ha processat la petició.
Procés normal principal:

1. El proxy envia al navegador una pàgina indicant que la pàgina sol·licitada no es pot mostrar perquè està prohibida.
2. O bé el proxy crear un nou socket del port 80 a la pàgina sol·licitada en el cas que la pàgina no estigui prohibida.
3. En ambdós casos el proxy emmagatzema les estadístiques de les pàgines visitades.

Alternatives de procés i excepcions:

1. No es troba el servidor de la pàgina sol·licitada.
    a. El browser mostra un missatge comunicant que no es pot accedir al recurs.
2. La sintaxi de la petició HTTP és incorrecta.
    a. El browser mostra un missatge comunicant que no es pot accedir al recurs.

10. Emmagatzemar estadístiques

Actors: servidor proxy.
Casos d'ús relacionats: 9. Processar socket.
Precondició: el sistema ha creat ha processat una petició de servei HTTP.
Postcondició: la connexió ha estat tancada i el sistema ha emmagatzemat les estadístiques de la connexió HTTP.
Procés normal principal:

1. El proxy compta el temps de connexió.
2. Quan es tanca la connexió el proxy emmagatzema les estadístiques en un fitxer.

11. Processar comanda

Actors: servidor proxy.
Casos d'ús relacionats: 9. Processar socket.
Precondició: el sistema ha creat una connexió i ha rebut una petició en Protocol TFC.
Postcondició: el sistema ha processat la petició.
Procés normal principal:

1. El proxy crida un mètode que processi les comandes.

6.4-Obtenció de les classes d'entitats

Una primera anàlisi dels casos d'ús ens dóna les següents possibles classes (només incorporem les classes quan apareixen en primer lloc en el cas d'ús):

- Del cas d'ús 1. Petició http: petició http, adreça.
- Del cas d'ús 2. Afegir adreça: Protocol TFC, llista adreces.
- Del cas d'ús 4. Visualització configuració: pantalla llista adreces prohibides.
- Del cas d'ús 5. Consultar estadístiques: pantalla estadístiques, estadístiques.
- Dels casos d'ús 7 a 10: servidor proxy.

Pel que fa al servidor proxy, l'estudi de les classes Socket i ServerSocket de Java, que farem servir en el desenvolupament de l'aplicació, suggereix que sigui implementat de la següent manera: una classe implementa el mètode main on crea un ServerSocket que escolta del port 80 i quan detecta una connexió, crea els sockets que es passen a uns objectes que realitzen les connexions (creen els sockets entre navegador i servidor proxy i entre servidor proxy i navegador), i fan start() d'aquests objectes, que hereten de la classe Thread de Java. La classe que implementa el mètode main serà el servidor proxy pròpiament dit i, per tant, l'anomenarem Proxy. Els objectes que creen les connexions formaran part d'una classe que anomenarem ProcesProxy, ja que seran els que duguin a terme tot el processament del servidor proxy.

Eliminem la classe adreça perquè no sembla que hagi de dur a terme cap mètode que no poguem implementar amb classes de Java ja existents (per exemple, la classe String).

De la petició HTTP només haurem de processar la capçalera, per tant li canviem el nom: CapcaleraHttp.

Les classes pantalla llista adreces prohibides i pantalla estadistiques les podem generalitzar en una sola classe Pantalla que pot resoldre, amb sobrecàrrega els mateixos mètodes per a objectes de les classes llista adreces i estadistiques

Ens quedem, per tant, amb les següents classes d'entitat, ja amb els seus noms definitius: Proxy, ProcesProxy, CapcaleraHttp, ProtocolTfc, LlistaAdreces, Pantalla, Estadistica.

Com que haurem de fer un processament molt semblant dels objectes CapcaleraHttp i dels objectes ProtocolTFC, té sentit que agrupem els mètodes que comparteixin aquestes classes en una classe, segurament abstracta, de la qual heretaran tots dos i que anomenarem Capcalera.

6.5-Diagrama de classes d'entitats a la fase d'anàlisi

El diagrama de classes d'entitats amb les classes definides queda de la manera següent:

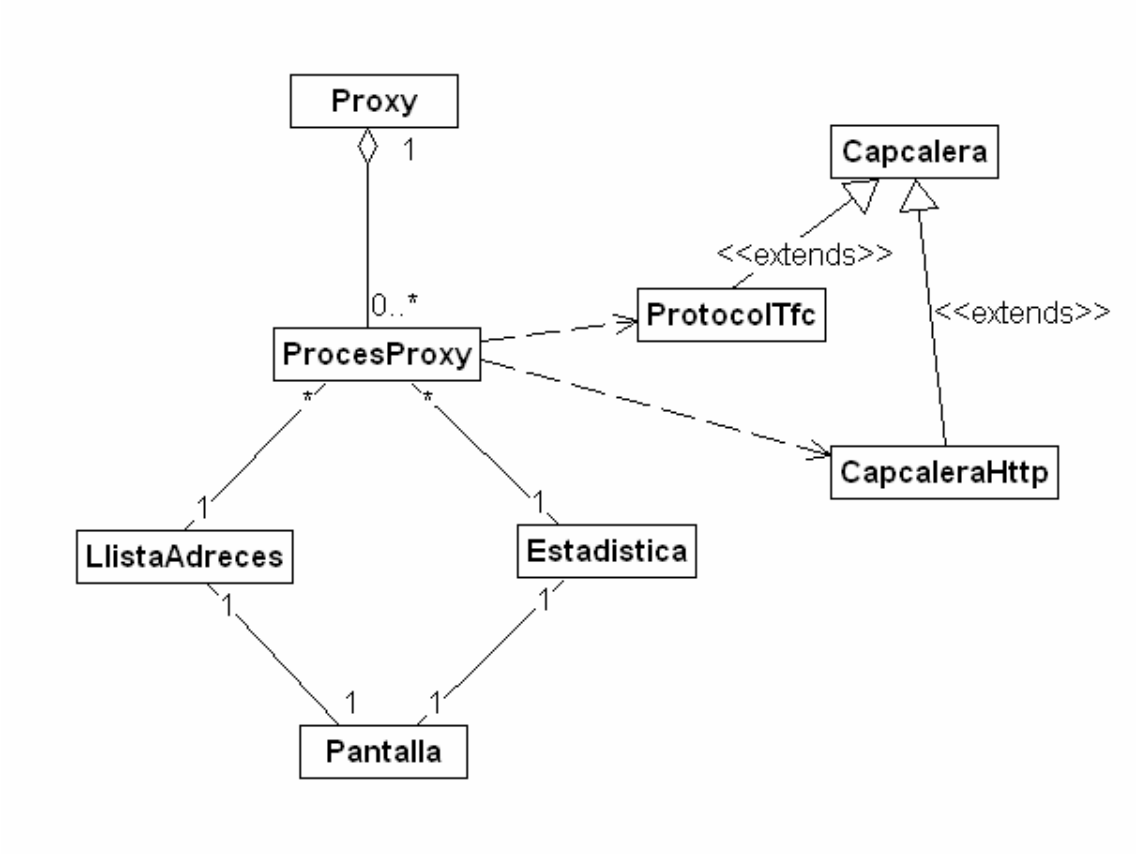

Figura 3: Diagrama d'entitats a la fase d'anàlisi

7-Disseny

7.1-Revisió de les classes d'entitats a la fase de disseny

En l'etapa d'anàlisi es van definir les següents classes d'entitats a partir del diagrama de casos d'ús: Proxy, ProcesProxy, CapcaleraHttp, ProtocolTfc, Capçalera, LlistaAdreces, Pantalla, Estadistica.

Al llarg de l'etapa de disseny i concretament en el procés d'escriure el codi de l'aplicació, s'ha detectat que era preferible processar les capçaleres HTTP directament en la classe ProcesProxy. La raó principal és que aquesta classe processa les connexions a través de sockets i es va veure que, en rebre una capçalera HTTP del browser només calia fer dues úniques operacions molt simples sobre capçaleres HTTP: una era extreure'n l'adreça d'internet per a determinar si era una adreça prohibida en la configuració del proxy i l'altra era crear una capçalera amb aquesta adreça per a enviar-la a la internet. Per tant, sembla clar que aquestes dues úniques operacions tan simples no justifiquen la creació de la classe CapcaleraHttp, que és eliminada. Automàticament és eliminada la classe Capcalera que estava pensada per a encapsular els mètodes comuns a les classes CapcaleraHttp i ProtocolTfc.

D'altra banda, en la implementació de l'aplicació es va veure que no era necessari crear instàncies de la classe LlistaAdreces ja que només contenia mètodes estàtics i li va ésser canviat el nom per GestorAdreces, que sembla més adient a les funcionalitats de la classe, que estan limitades a la gestió del fitxer d'adreces prohibides.

En implementar la classe Pantalla, pensada per a enviar missatges al servidor proxy mostrant estadístiques, configuració del proxy i missatges de confirmació, es va veure que calia un pas de paràmetres farragós passant sockets des de la classe ProcesProxy a la classe Pantalla o creant nous sockets. La següent solució simplifica el plantejament i el codi i s'ha optat per implementar-la.

Quan en un objecte de la classe ProcesProxy es detecta que s'ha rebut una capçalera de Protocol TFC, es creen objectes de la classe ProtocolTfc que contenen la comanda en Protocol TFC rebuda. En aquests objectes es delega el procés de la comanda en Protocol TFC i, en executar la comanda, retornen un missatge en HTML que pot ser escrit al socket del browser, mostrant el resultat del processament de la comanda en Protocol TFC per pantalla.

7.2-Revisió del diagrama de classes d'entitats

Els canvis esmentats donen com a resultat el següent diagrama de classes d'entitats:

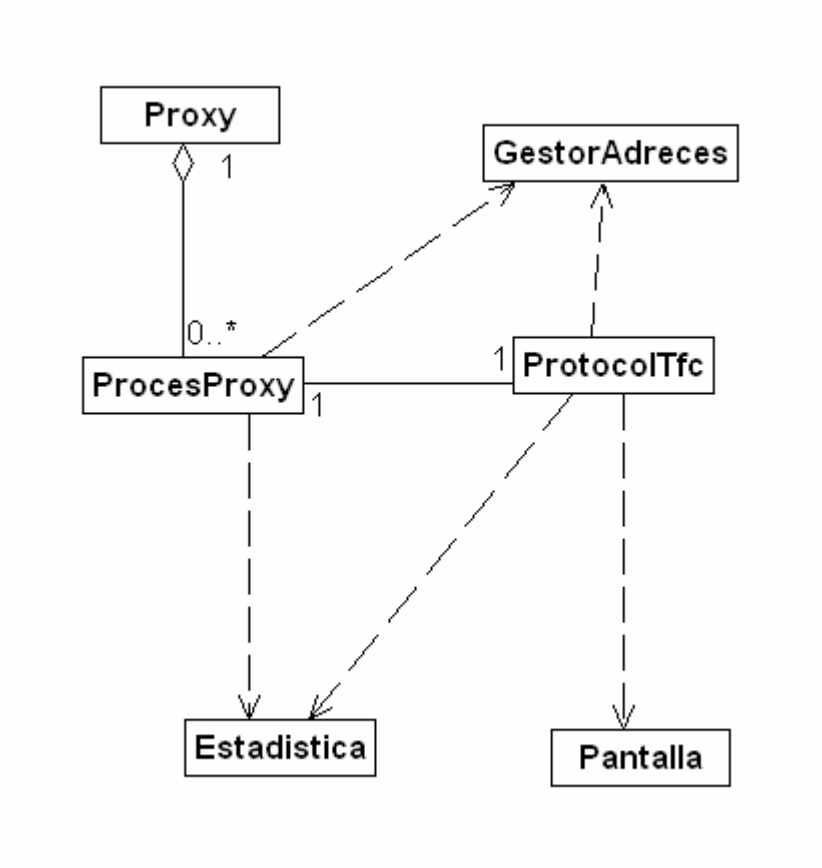

Figura 4: Diagrama de classes d'entitats a la fase de disseny

8- Entorn de desenvolupament de l'aplicació

Donada l'àmplia difusió del sistema operatiu Windows i la disponibilitat per part de l'estudiant d'aquest sistema operatiu, l'aplicació ha estat desenvolupada per a Windows.

El llenguatge de programació utilitzat ha estat Java i, concretament, les llibreries de Sockets de Java (classes Socket i ServerSocket). La versió de Java utilitzada ha estat JDK 1.4.2.

Com a entorn de programació s'ha fet servir l'IDE NetBeans 5.0.

El browser o navegador d'internet utilitzat ha estat el Microsoft Explorer 6.0 configurat per a funcionar amb un servidor proxy. Per a configurar el navegador en cas d'utilitzar accés a internet via una xarxa LAN, cal seguir les instruccions que trobem a l'adreça següent: http://www.univalle.edu.co/Conexion/proxy/iexplorer.html. Per a configurar el navegador en cas d'utilitzar un mòdem i connexió telefònica, cal configurar al navegador una nova connexió telefònica:

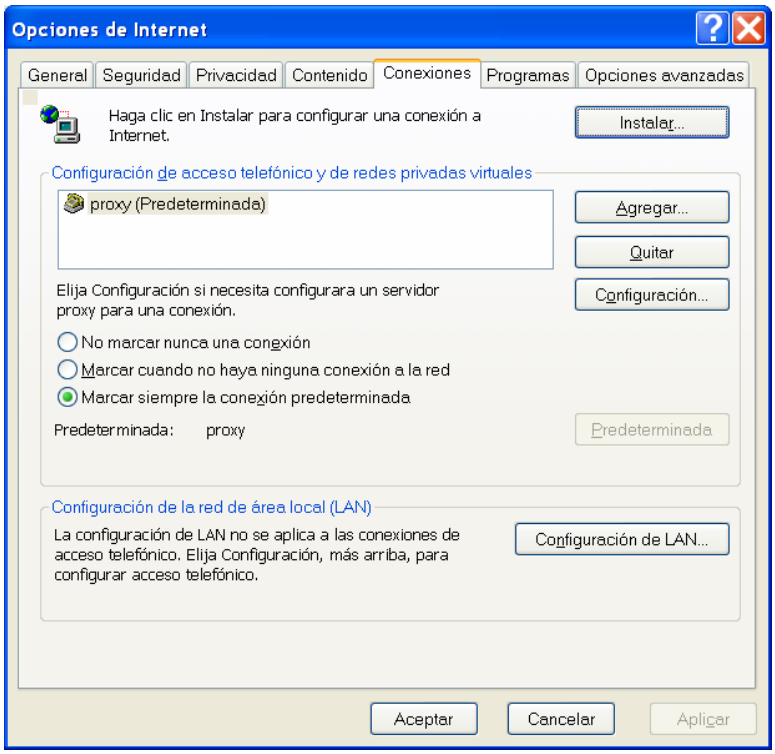

Figura 5: Configuració del browser (1)

I configurar aquesta connexió per a que utilitzi un proxy al localhost:

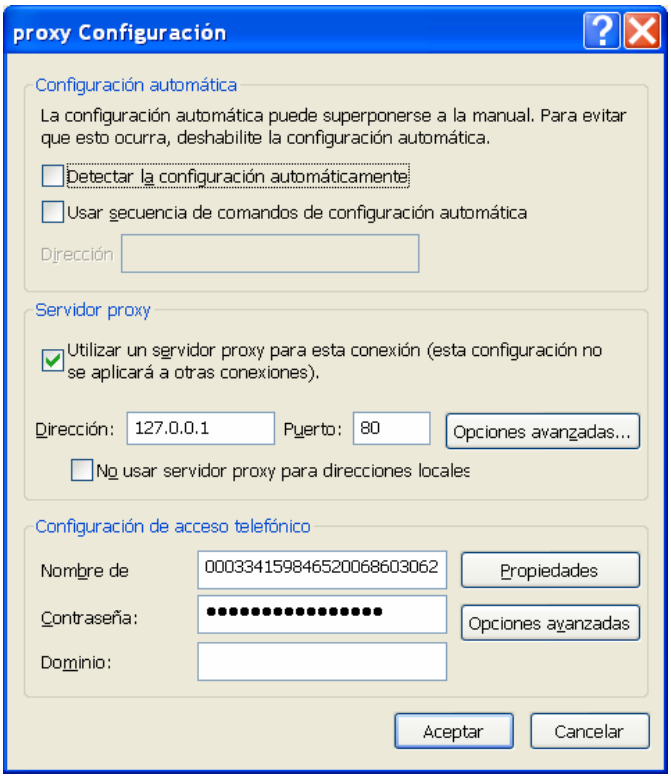

Figura 6: Configuració del browser (2)

Totes les eines estan instal·lades sobre un ordinador portàtil amb processador Intel Centrino i sistema Operatiu Windows XP.

9-Implementació

9.1-El Protocol TFC

Definim les següents comandes com a integrants del Protocol TFC que servirà per a la comunicació entre l'usuari privilegiat i el servidor proxy:

- afegir www.adreca.com: afegeix l'adreça www.adreca.com al fitxer d'adreces prohibides.
- eliminar www.adreca.com: elimina l'adreça www.adreca.com del fitxer d'adreces prohibides.
- visualitzar: treu per pantalla, a través del navegador d'internet, la configuració del servidor proxy, és a dir, el contingut del fitxer d'adreces prohibides.
- consultar: treu per pantalla, a través del navegador d'internet, les estadístiques d'ús d'internet (Adreça, Accés denegat Sí/No, Nombre de visites, Temps total de connexió).
- reiniciar: reiniciem les dades, posem a zero tots els comptadors de les estadístiques d'ús d'internet.

Les comandes en Protocol TFC s'envian des del browser com si fossin peticions web i han de tenir la següent sintaxi per tal de poder ser acceptades pel browser:

- http://127.0.0.1/afegir www.adreca.com
- http://127.0.0.1/eliminar www.adreca.com
- http://127.0.0.1/visualitzar
- http://127.0.0.1/consultar
- http://127.0.0.1/reiniciar

9.2-El fitxer d'adreces prohibides

El fitxer que conté les adreces prohibides és un simple fitxer de text (extensió .txt) i el seu contingut té el següent aspecte:

```
www.google.com
www.elpais.es
www.sport.es
www.marca.es
www.e-noticies.com
www.elmundo.es
www.uoc.edu
```

El seu contingut pot ser creat o modificat mitjançant les comandes del Protocol TFC o manipulant-lo mitjançant qualsevol processador de text. La seva ruta, continguda a la interfície Constants.java és C:\Proxy.

9.3-El fitxer d'estadístiques

El fitxer que conté les estadístiques és un simple fitxer de text (extensió .txt) i el seu contingut té el següent aspecte:

```
web.lavanguardia.es 0 43 81826
www.lavanguardia.es 0 9 40905
secure-uk.imrworldwide.com 0 2 4625
ad.es.doubleclick.net 0 1 219
www.avui.cat 0 1 296
www.google.de 0 8 9343
www.bellver.org 0 1 421
www.uoc.edu 9 55 82985
www.marca.es 1 0 0
```

El seu contingut és creat o modificat dins el procés `run()` de la classe ProcesProxy.

La primera columna correspon a l'adreça d'internet de la pàgina web visitada. La segona columna correspon al nombre de vegades que el proxy ha denegat la connexió. La tercera columna correspon al nombre de vegades que ha estat establerta una connexió. La quarta columna correspon al temps total de connexió en mil·lisegons.

La seva ruta, continguda a la interfície Constants.java és C:\Proxy.

9.4-Implementació de les classes

A continuació es presenta una descripció de les classes amb els seus atributs:

- **Classe Proxy:** classe on s'executa el procés principal, és a dir, el programa per a executar un servidor proxy. Conté el següent mètode:

```
public static void main(java.lang.String[] args)
                throws java.io.IOException
```

Mètode main que crea un ServerSocket que escolta del port 80 i, quan detecta una connexió, crea els sockets que es passen a objectes de la classe ProcesProxy que realitzen les connexions.

- **Classe ProcesProxy:** classe que processa els sockets quan es crea una connexió. Hereta de la classe Thread i conté un mètode run():

`public void run()`

Mètode run que llegeix la capçalera HTTP enviada pel browser i la processa. Si es tracta de Protocol TFC la processa com a tal i si es tracta de d'HTTP crea un socket amb un servidor a la internet i passa les dades rebudes al socket entre servidor proxy i browser.

- **Classe ProtocolTfc: c**lasse que defineix el protocol Protocol TFC de comunicació per a canviar la configuració del proxy, visualitzar la configuració del proxy i veure estadístiques de tràfic a través del proxy. Consta dels següents atributs:

`String`	**`adreca`** Atribut que conté el domini.
`String`	**`comanda`** Atribut que conté la comanda del Protocol TFC ("afegir", "eliminar", "visualitzar", "consultar" o "reiniciar").

Conté els següents mètodes:

`public ProtocolTfc()`

Mètode constructor per a crear una nova instància del ProtocolTfc.

`public ProtocolTfc(String comanda)`

Constructor de la classe ProtocolTfc amb un paràmetre.
Pre-condició: `comanda` identifica la comanda del protocol.
Post-condició: s'ha creat una instància de ProtocolTfc contenint la comanda entrada com a paràmetre.
Paràmetres:
`comanda` - String identificador de la comanda de la instància de ProtocolTfc. Es tracta d'un String que pot prendre els valors "visualizar", "consultar" o "reiniciar".

`public ProtocolTfc(String comanda, String adreca)`

Constructor de la classe ProtocolTfc amb dos paràmetres.
Pre-condició: `comanda` identifica la comanda del protocol i adreca identifica el domini sobre el que volem realitzar l'operació especificada per comanda.

Post-condició: s'ha creat una instància de ProtocolTfc contenint la comanda i l'adreça entrades com a paràmetres.
Paràmetres:
`comanda` - String identificador de la comanda de la instància de ProtocolTfc. Es tracta d'un String que pot prendre els valors "afegir" o "eliminar".
`adreca` - String identificador del domini sobre el que volem realitzar l'operació. Es tracta d'una adreca d'internet.

`public String` **`ProcessarProtocolTfc`**`()`
Pre-condició: mètode sense paràmetres per a processar la comanda del Protocol TFC.
Post-condició: existeix una instància de la classe ProtocolTfc. Retorna un missatge de confirmació en HTML a la classe ProcesProxy.
Ha estat processada l'operació definida per la instància de ProtocolTfc.

`public static boolean` **`esProtocolTfc`**`(String comanda)`
Mètode estàtic per a determinar si una comanda pertany al Protocol TFC.
Pre-condició: el servidor proxy ha rebut una comanda.
Post-condició: el mètode retorna cert si la comanda pertany al Protocol TFC ("afegir", "eliminar", "visualitzar", "consultar" o "reiniciar").
Paràmetres:
`comanda` - String conté la comanda la qual cal determinar si pertany al Protocol TFC.

- **Classe Pantalla:** classe per a gestionar els missatges que un servidor proxy HTTP que entén el Protocol TFC envia a un browser com a resposta a comandes en Protocol TFC. Conté els següents mètodes:

`public static String` **`PantallaAfegir`**`(String adreca)`
Mètode estàtic per a retornar el missatge de confirmació que l'accés al domini especificat pel paràmetre ha estat restringit.
Pre-condició: una adreça ha estat afegida al fitxer d'adreces prohibides.
Post-condició: el mètode retorna un missatge de confirmació en HTML al mètode **ProcessarProtocolTfc**() de la classe ProtocolTfc.
Paràmetres:
`adreca` - String conté el domini al qual ha estat restringit l'accés.

`public static String` **`PantallaEliminar`**`(String adreca)`
Mètode estàtic per a retornar el missatge de confirmació que l'accés al domini especificat pel paràmetre està permès.
Pre-condició: una adreça ha estat eliminada del fitxer d'adreces prohibides.
Post-condició: el mètode retorna un missatge de confirmació en HTML al mètode **ProcessarProtocolTfc**() de la classe ProtocolTfc..
Paràmetres:
`adreca` - String conté el domini al qual es permet l'accés.

`public static String` **`PantallaConfiguracio`**`()`
Mètode estàtic sense paràmetres per a retornar un missatge amb el contingut del fitxer d'adreces prohibides.
Pre-condició: l'usuari ha executat la comanda "visualitzar" del Protocol TFC.

Post-condició: el mètode retorna un missatge amb la informació en HTML al mètode **ProcessarProtocolTfc**() de la classe ProtocolTfc..

`public static String` **`PantallaEstadistiques`**`()`

Mètode estàtic sense paràmetres per a retornar un missatge amb les estadístiques del servidor proxy.

Pre-condició: l'usuari ha executat la comanda "consultar" del Protocol TFC.

Post-condició: el mètode retorna un missatge amb la informació en HTML al mètode **ProcessarProtocolTfc**() de la classe ProtocolTfc..

`public static String` **`PantallaReiniciar`**`()`

Mètode estàtic sense paràmetres per a retornar un missatge de confirmació en haver posat a zero les estadístiques.

Pre-condició: l'usuari ha executat la comanda "reiniciar" del Protocol TFC.

Post-condició: el mètode retorna un missatge de confirmació en HTML al mètode **ProcessarProtocolTfc**() de la classe ProtocolTfc..

- **Classe GestorAdreces:** classe per a gestionar la configuració (dominis prohibits/permesos) d'un servidor proxy http. Conté els següents mètodes:

`public static boolean` **`adrecaProhibida`**`(String adreca)`

Mètode estàtic amb un paràmetre per a determinar si el domini demanat està al fitxer d'adreces prohibides.

Pre-condició: el servidor proxy ha rebut una petició de connexió a una adreça d'internet.

Post-condició: la funció retorna cert si l'adreça està present al fitxer d'adreces prohibides.

Paràmetres:

`adreca` - String l'adreça d'internet per a la qual el servidor proxy ha rebut la petició de connexió.

`public static void` **`afegirAdreca`**`(String adreca)`

Mètode estàtic amb un paràmetre per a prohibir l'accés a una adreça d'internet, és a dir, per a afegir una adreça d'internet al fitxer d'adreces prohibides.

Pre-condició: el servidor proxy ha rebut la comanda "afegir" del Protocol TFC per a denegar l'accés a una adreça d'internet, és a dir, per a afegir una adreça d'internet al fitxer d'adreces prohibides.

Post-condició: es prohibeix l'accés, és a dir, l'adreça d'internet ha estat afegida al fitxer d'adreces prohibides si no hi era ja present.

Paràmetres:

`adreca` - String l'adreça d'internet a la qual es vol prohibir l'accés.

`public static void` **`eliminarAdreca`**`(String adreca)`

Mètode estàtic amb un paràmetre per a permetre l'accés a una adreça d'internet, és a dir, per a eliminar una adreça d'internet del fitxer d'adreces prohibides.

Pre-condició: el servidor proxy ha rebut la comanda "eliminar" del Protocol TFC per a permetre l'accés a una adreça d'internet, és a dir, per a eliminar una adreça d'internet del fitxer d'adreces prohibides.

Post-condició: s'allibera l'accés, és a dir, l'adreça d'internet ha estat eliminada del fitxer d'adreces prohibides si hi era ja present.

Paràmetres:

`adreca` - String l'adreça d'internet a la qual es vol permetre l'accés.

- **Classe Estadistica:** classe per a gestionar les estadístiques generades per un servidor proxy HTTP. Conté els següents mètodes:

`public static void` **`actualitzaEstadistica`**`(String a, long l)`

Mètode estàtic amb dos paràmetres per a actualitzar les estadístiques amb les dades corresponents a la darrera connexió a internet. Actualitzem el fitxer d'estadístiques amb una nova entrada o sumant el temps de connexió a una entrada existent. A més, sumem una connexió al comptador de connexions corresponent a l'adreça d'internet.

Pre-condició: el servidor acaba de tancar una connexió a internet a una adreça d'internet.

Post-condició: les estadístiques han estat actualitzades amb les dades corresponents a la darrera connexió.

Paràmetres:

`a` - String l'adreça d'internet a la qual el servidor proxy acaba de tancar la connexió.

`l` - long conté el temps en mil·lisegons que ha durat la connexió.

`public static void` **`actualitzaEstadistica`**`(String a)`

Mètode estàtic amb un paràmetre per a actualitzar les estadístiques amb les dades corresponents a la darrera connexió a internet denegada. Actualitzem el fitxer d'estadístiques amb una nova entrada o sumant un intent de connexió a una entrada existent.

Pre-condició: el servidor proxy acaba de denegar una connexió a internet a una adreça d'internet.

Post-condició: les estadístiques han estat actualitzades amb les dades corresponents a l'intent de connexió.

Paràmetres:

`a` - String l'adreça d'internet a la qual el servidor proxy acaba de denegar la connexió.

`public static void` **`reiniciarEstadistica`**`()`

Mètode públic sense paràmetres per a esborrar el contingut del fitxer on emmagatzemem les estadístiques.

Pre-condició: el servidor proxy ha rebut la comanda "reiniciar" pertanyent al Protocol TFC.

Post-condició: el contingut del fitxer on emmagatzemem les estadístques ha estat esborrat.

- **Interfície Constants:** conté les següents constants comunes a les classes que la implementen; les constants són implícitament public, static i final:

`String`	**`CR_LF`** Retorn de carro més salt de línia (CarriageReturn + LineFeed).
`String`	**`estadistiques`** Ruta/nom.extensió, per defecte, del fitxer de text que conté les

	estadistiques.
`static short`	**`PORT`** Número de port on corre el servidor proxy
`String`	**`prohibides`** Ruta/nom.extensió, per defecte, del fitxer de text que conté les adreces prohibides.

10-Manual de l’usuari

10.1-Instruccions d’instal·lació

Cal crear un directori de nom Proxy en el directori arrel del disc dur on es desitgi executar l’aplicació, per exemple, si es vol executar l’aplicació des d’una unitat de disc de nom C:, cal crear-hi el directori C:\Proxy.

Cal copiar l’executable Proxy.exe en aquest directori. Per a fer funcionar el programa cal executar el fitxer Proxy.exe.

El programa crearà els fitxers denegar.txt, que contindrà les adreces prohibides, i el fitxer estadistiques.txt, que contindrà les estadístiques de trànsit a través del servidor proxy en aquest directori el primer cop que s’executi.

10.2-Funcionament del servidor Proxy

Quan el programa es posa en funcionament, la següent finestra ens recorda de quin port escolta el servidor proxy:

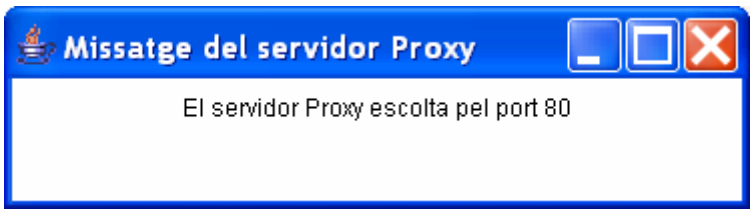

Figura 7: Missatge del proxy recordant el número de port

Aquesta informació ens és d'utilitat per a configurar el navegador d'internet.

Es produeixen les següents situacions al llarg d'una connexió amb internet quan el servidor Proxy està en funcionament:

1. L'usuari introdueix una adreça no prohibida pel Proxy: la connexió té lloc normalment, la presència del Proxy és transparent per a l'usuari que rep la pàgina web sol·licitada.

2. L'usuari introdueix una adreça prohibida pel Proxy. El Proxy no permet la connexió a l'adreça d'internet demanda i apareix el següent missatge pel browser:

Figura 8: Missatge de denegació d'accés

3. L'usuari privilegiat prohibeix l'accés a una adreça determinada, per exemple www.uoc.edu, mitjançant la comanda http://127.0.0.1/afegir www.uoc.edu. El proxy afegeix www.uoc.edu al fitxer d'adreces prohibides i apareix el següent missatge pel browser:

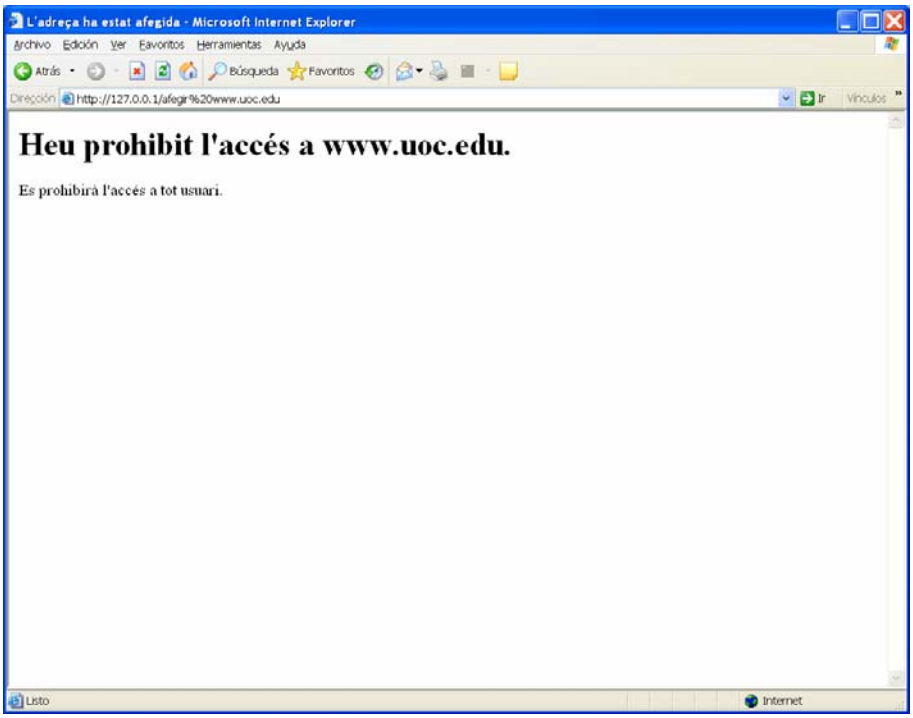

Figura 9: Missatge de confirmació en afegir una adreça prohibida

4. L'usuari privilegiat permet l'accés a una adreça determinada, per exemple www.uoc.edu, mitjançant la comanda http://127.0.0.1/eliminar www.uoc.edu. El proxy elimina www.uoc.edu del fitxer d'adreces prohibides i apareix el següent missatge pel browser:

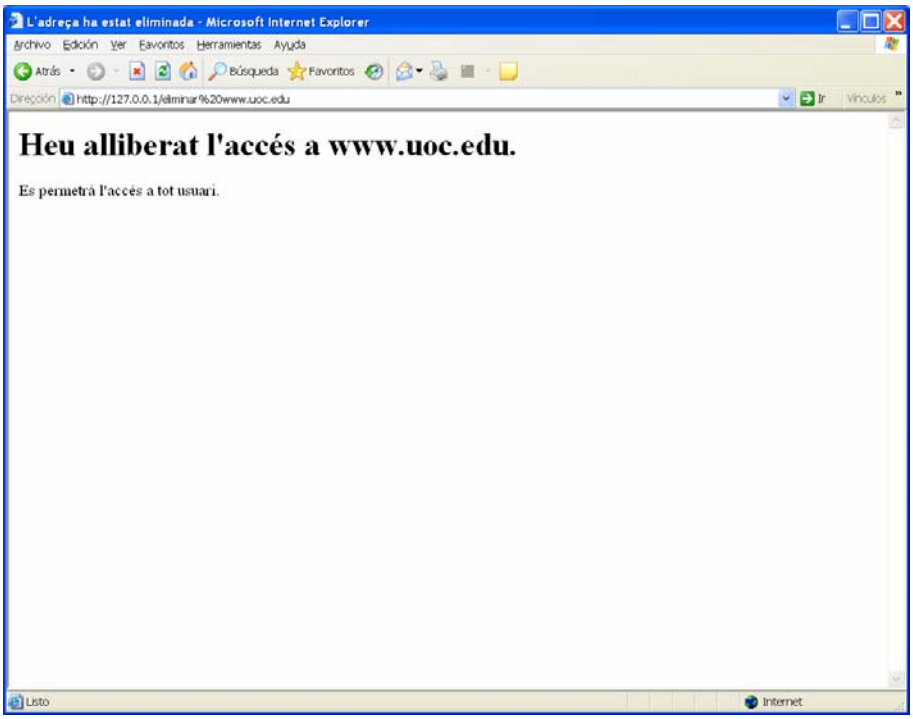

Figura 10: Missatge de confirmació en eliminat una adreça prohibida

5. L'usuari privilegiat consulta la configuració del Proxy mitjançant la comanda http://127.0.0.1/visualitzar. El següent missatge apareix al browser:

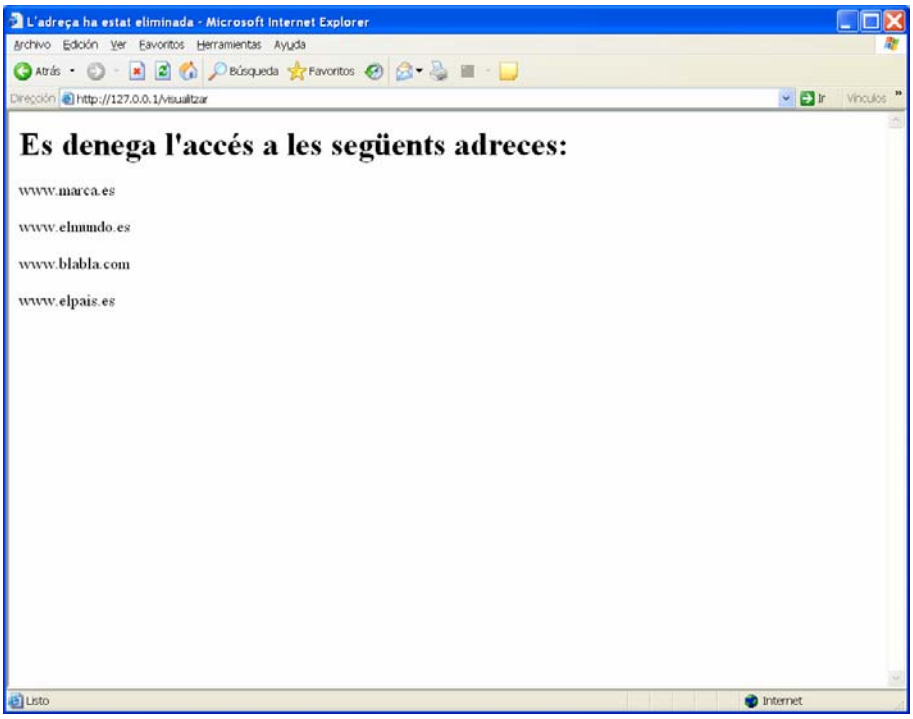

Figura 11: Visualització d'adreces prohibides

6. L'usuari privilegiat consulta les estadístiques del Proxy mitjançant la comanda http://127.0.0.1/consultar. El següent missatge apareix al browser:

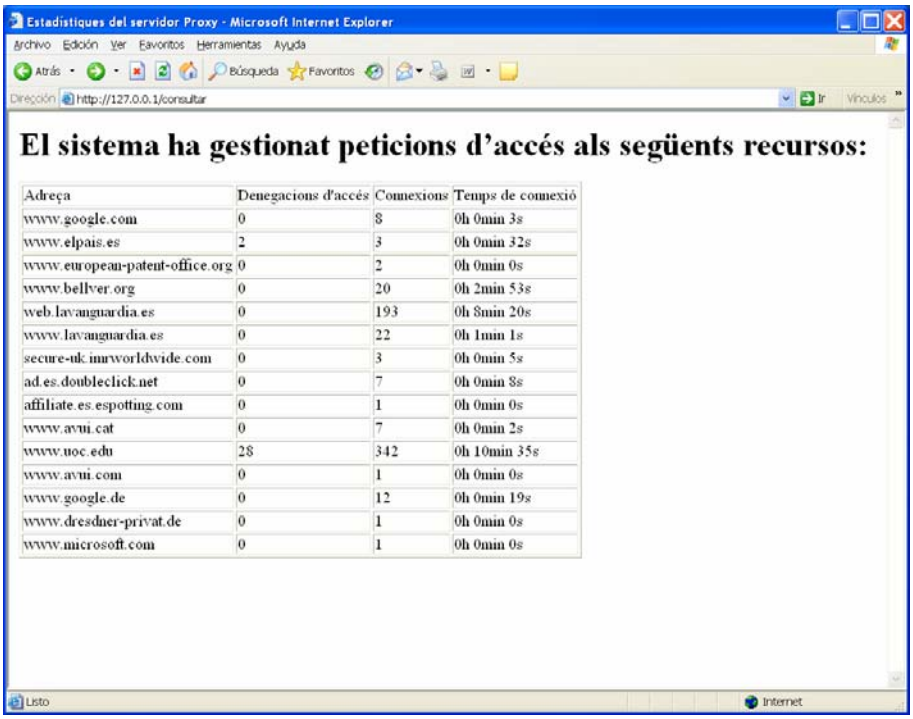

Figura 12: Visualització d'estadístiques

7. L'usuari privilegiat reinicia les estadístiques del Proxy mitjançant la comanda http://127.0.0.1/reiniciar. El Proxy esborra el contingut del fitxer d'estadístiques i el següent missatge apareix pel browser:

Figura 13: Missatge de confirmació de reinici d'estadístiques

11-Conclusions i possibles millores

La principal fita aconseguida pel programari és la senzilla gestió del servidor proxy. Sense necessitat d'una GUI, l'usuari pot gestionar de manera molt senzilla el servidor proxy mitjançant un protocol de cinc comandes. D'altra banda, la comunicació entre el proxy i l'usuari és també plenament satisfactòria, amb missatges molt clars de confirmació o contenint la informació de configuració o d'estadístiques de xarxa.

La visualització de pàgines transferides en HTTP és també satisfactòria, l'usuari no percep que hi ha un servidor proxy addicional entre ell i internet. Només un usuari avançat percebrà una menor velocitat de transmissió, conseqüència lògica i inevitable de la incorporació d'un nou element a la xarxa.

La restricció de l'accés a l'aplicació, és a dir, garantir que només l'usuari privilegiat pot tenir accés al servidor, es basa en una assumpció certament naïf: suposem que només l'usuari privilegiat coneix l'existència del proxy i el conjunt de comandes. Això és perfectament vàlid per a un gran nombre dels usuaris del sistema, però un usuari avançat pot fàcilment descobrir l'existència del proxy i canviar-ne la configuració. La identificació de l'usuari privilegiat o administrador és, per tant, desitjable, però cau fora de l'objecte del TFC i és, a més, fàcilment implementable i coneguda per un programador experimentat. Per aquests motius no s'ha implementat en el present TFC.

Per motius d'extensió del projecte (realitzat en el marc de l'assignatura de TFC de 7,5 crèdits de la UOC) el projecte s'ha limitat a la transmissió del protocol HTTP. Han aparegut problemes que no s'han aconseguit solucionar per a la transmissió d'intercanvis en SSL i per a tractar de manera adient la resposta 301 de l'HTTP. El codi ha avançat en la resolució d'aquests problemes però no s'ha trobat una solució satisfactòria. Amb una mica més de temps i un període addicional de proves i *trouble-shooting* del codi aquests problemes es podrien haver resolt.

Com a possibles ampliacions interessants del codi, podem citar les següents:

- Introduir en el servidor proxy la gestió del caché del browser.
- Estendre les funcionalitats oferides en HTTP a altres protocols i/o aplicacions com, per exemple, FTP.
- Identificació de l'usuari, amb introducció de contrassenya per a autoritzar l'us del servidor com a usuari privilegiat.

12-Glossari

Adreça d'internet: adreça d'un servidor d'internet formada per tres camps: host, entitat i top level domain, per exemple, www.uoc.edu.

Browser: aplicació que permet a l'usuari recuperar i reproduir documents d'hipertext, generalment escrits en HTML, des de servidors web. En són exemples coneguts Internet Explorer, Netscape o Mozilla.

FTP: acrònim de *File Transfer Protocol*, és un estàndard per a enviar fitxers entre ordinadors amb qualsevol sistema operatiu. Forma part de la capa d'aplicació del model TCP/IP.

HTTP: acrònim de *Hypertext Transfer Protocol*, és un estàndard per a l'intercanvi d'hipertext i multimèdia a la web.

HTML: acrònim d'*Hypertext Mark-up Language*, és un llenguatge de marcació, és a dir, combina text i informació addicional lligada al text. La informació addicional, per exemple sobre l'estructura o presentació del text, s'expressa fent servir la marcació, intercalada al text primari.

IDE: acrònim d'*Integrated Development Environment*, és a dir, un entorn integrat de desenvolupament, composat per diferents eines d'ajuda a la programació.

JVM: acrònim de *Java Virtual Machine*, màquina virtual que executa el codi resultant de la compilació d'un programa escrit mitjançant el llenguatge Java.

Navegador d'internet: sinònim de browser.

Protocol: conjunt de regles que control·len la seqüència de missatges entre dues entitats.

Protocol TFC: protocol per a la comunicació entre l'usuari i el servidor proxy desenvolupat en el present TFC.

Proxy: programa o dispositiu que realitza una acció en representació d'un altre.

Servidor proxy: programa o dispositiu proxy que intercepta la navegació dels clients per internet.

Socket: connector, punt d'accés als serveis de comunicació en l'àmbit del transport.

SSL: acrònim de *Secure Sockets Layer*, protocol criptogràfic per a garantir connexions segures a internet.

Thread: fil d'execució d'un programa.

URL: acrònim de *Uniform Resouce Locator*, identifica la localització d'un recurs amb la següent sintaxi: protocol":" "//" host [":" número de port] [path].

13-Bibliografia

Especificació HTTP: http://ftp.ics.uci.edu/pub/ietf/http/rfc1945.html

Pàgina sobre HTTP: http://www.jmarshall.com/easy/http/

Tutorial de la llibreria de sockets de Java:
http://java.sun.com/docs/books/tutorial/networking/sockets/

Ian F. Darwin, Java Cookbook, O'Reilly, 2a edició.

API del JDK de Java.

Apunts de l'assignatura de la UOC 'Xarxes de Computadors'.

Apunts de l'assignatura de la UOC 'Enginyeria del Programari 1'.

Pàgina sobre HTML: http://www.htmlcodetutorial.com/

Aplicatiu "Llenguatge HTML" de l'assignatura 'Multimèdia i Comunicació' de la UOC.

Gary Pollice, RUP and XP, Part I: Finding Common Ground, The Rational Edge, 2001.

Gary Pollice, RUP and XP, Part II: Valuing Differences, The Rational Edge, 2001.