TFCProxy: Filtrat de pàgines Web

Indalecio Villafranca González

Enginyeria Tècnica d'Informàtica de Gestió

Consultora: Maria Isabel March Hermo

Data: 18-06-2006

1 DEDICATÒRIA

Aquest Treball ha estat possible gràcies als ànims, el recolzament i l'estima de la meva dona, Lou, i en especial al naixement del meu fill Jacob. Han hagut d'aguantar les jornades d'estudi i cerca que he hagut de fer.

A la meva tutora, pels ànims que m'ha donat i que m'ha fet trobar la llum els dies que estaven foscos.

Al meu amic Peter pels ànims que m'ha donat des d'un començament perquè acabés la carrera i m'esforcés encara més per acabar els meus estudis.

A la meva família per comprendre'm en moments que no he pogut estar amb ella.

A tots, **MOLTES GRÀCIES!!!**

2 RESUM

Internet s'ha convertit avui en dia en una eina imprescindible de treball. No hi ha feina que no depengui en major o menor grau dels recursos d'Internet, ja sigui per buscar documentació, per presentar els productes de l'empresa, per vendre, per donar-se a conèixer, etc. Però, aquestos recursos no sempre estan ben aprofitats i la gent, molt sovint, utilitza Internet per esbarjo i no pas per temes de feina.

És, llavors, quan un mal ús d'aquestos recursos ha esdevingut un mal de cap per les empreses. Els directius volen limitar l'accés d'Internet per a usos de feina i és per això que es va plantejar com fer-ho. Doncs, filtrant la informació que la gent busca o vol accedir i això es pot aconseguir mitjançant un proxy.

I no solament a la feina, sinó que també a casa uns pares voldran restringir els accessos a Internet del seu fill. Segurament no voldran que el seu fill vegi pàgines de sexe, de violència, etc. Aquí també es pot delimitar les pàgines domèstiques mitjançant un filtre proxy.

Els centres públics d'ensenyament, acadèmies, biblioteques, locutoris públics amb accés a Internet, cybercafés, grans centres comercials, etc. també podrien gaudir de les característiques d'aquest programa per evitar situacions complicades.

Degut a aquest fet, en aquest TFC s'abordarà la creació d'un proxy què permetrà filtrar les pàgines web que continguin una sèrie d'adreces o paraules clau que haurem, prèviament, definit en un fitxer. Serà, doncs, una eina que permeti un major control als accessos a segons quines pàgines.

3 ÍNDEX

3.1 Índex de Continguts

3.2 Índex de Figures

3.3 Índex de Taules

4 INTRODUCCIÓ

L'objectiu d'aquest projecte és el desenvolupament d'una aplicació que faci la funció d'un filtre per fer més segura la navegació a través de les pàgines web d'Internet; és a dir, fer una restricció de l'accés de les pàgines que no es volen que es mostrin.

Hem pogut comprovar que al mercat hi ha un grapat d'aplicacions que desenvolupen aquesta tasca amb més o menys encert. N'hi han que filtren unes paraules que té definides al seu motor, n'hi han que es poden programar, n'hi han que enregistren l'accés indiscriminat a través d'un log, etc.

La nostra aplicació no vol ser una gran aplicació comercial amb gran quantitat de paràmetres i funcions com pot ser Filtrar[1], *Figura 1 Pantalla del programa Filtrar*, sinó una aplicació senzilla d'utilitzar i de la qual s'obtingui resultats eficients. És per això que s'ha dotat a l'aplicació d'una sèrie de funcionalitats com poden ser:

- creació d'un fitxer per introduir les paraules-clau que es volen filtrar;
- un fitxer per controlar els accessos indiscriminats a llocs que volem filtrar.
- A més a més, podria ser un programari de lliure distribució o de codi obert, perquè es pogués implementar encara més i dotar-lo d'altres funcionalitats. I és que la idea no és comercialitzar aquest aplicatiu, sinó més bé el contrari, donar-lo a conèixer perquè serveixi d'utilitat i poder-lo ampliar.

[1] Filtrar: www.filtrar.com

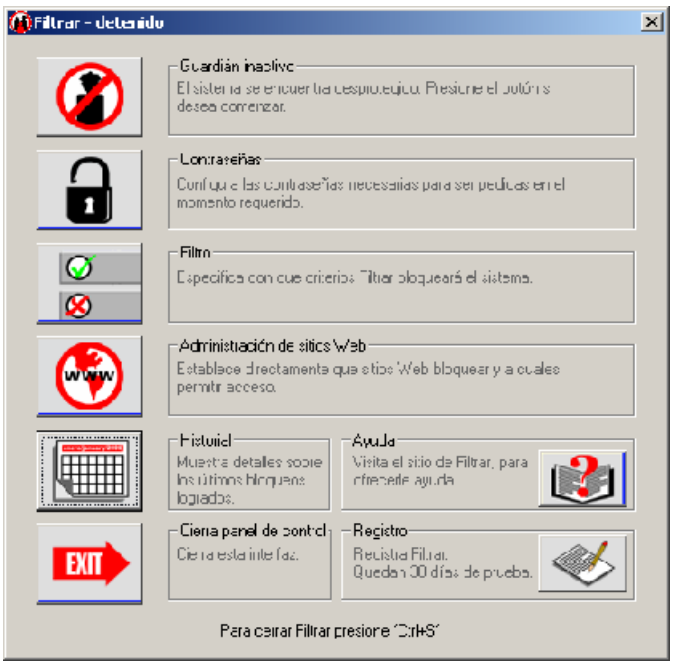

Figura 1 Pantalla del programa Filtrar

Aquest filtre de pàgines Web consisteix a interceptar unes paraules clau que no volem que es mostrin i denegar la visualització d'aquesta pàgina web al nostre navegador. Aquesta paraula clau pot estar tant a l'adreça de la pàgina web (a la URL) com dintre del cos del document a llegir.

Aquest conjunt de pàgines clau pot estar emmagatzemat dintre d'un fitxer pla, d'una base de dades, etc. i es pot anar modificant a mesura que es va utilitzant.

Al mercat ja existeix aquest tipus de programa de filtre de pàgines web, com poden ser Filtrar, NetNanny[2], Naomi[3], etc.

Totes aquestes aplicacions que hem esmentat són aplicacions de pagament, si bé existeixen d'altres de lliure distribució, com Kidkey [4], i que són igualment d'útils que ja esmentats anteiorment i que estan per altres plataformes, com poden ser, *iWayPatrol* [5]

[2] NeNanny: www.netnanny.com

[3] Naomi: http://www.naomifilter.org/index.html

[4] Kidkey: http://sp.kidkey.com/

[5] iWayPatrol: http://www.iwaypatrol.net/iway_index.html

N'hi han aplicacions que, per seguretat, els paràmetres del programa estan protegits per nom d'usuari i contrasenya, com el que es mostra a la *Figura 2 Entrada a l'aplicació NetNanny per canviar paràmetres.*

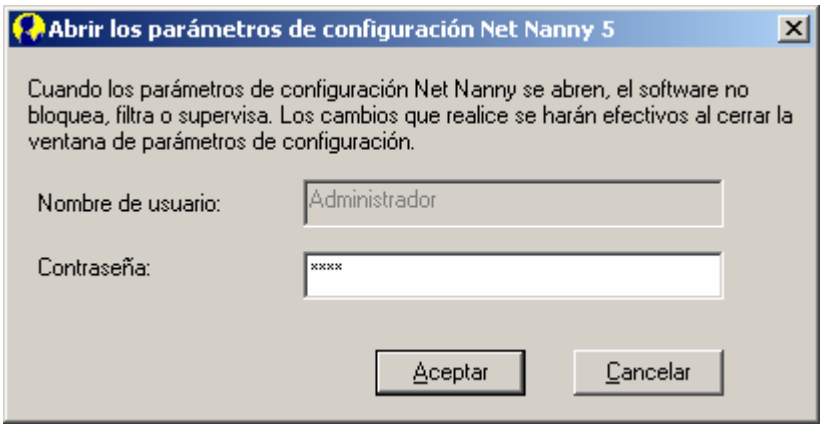

Figura 2 Entrada a l'aplicació NetNanny per canviar paràmetres

El tema de log o registre d'activitat està bastant aconseguit a l'aplicació Net Nanny, on té una finestra que mostra un historial del registre i dels usuaris, *Figura 3 Registre d'activitat en Net Nanny* i *Figura 4 Registre d'activitat en Net Nanny (2).*

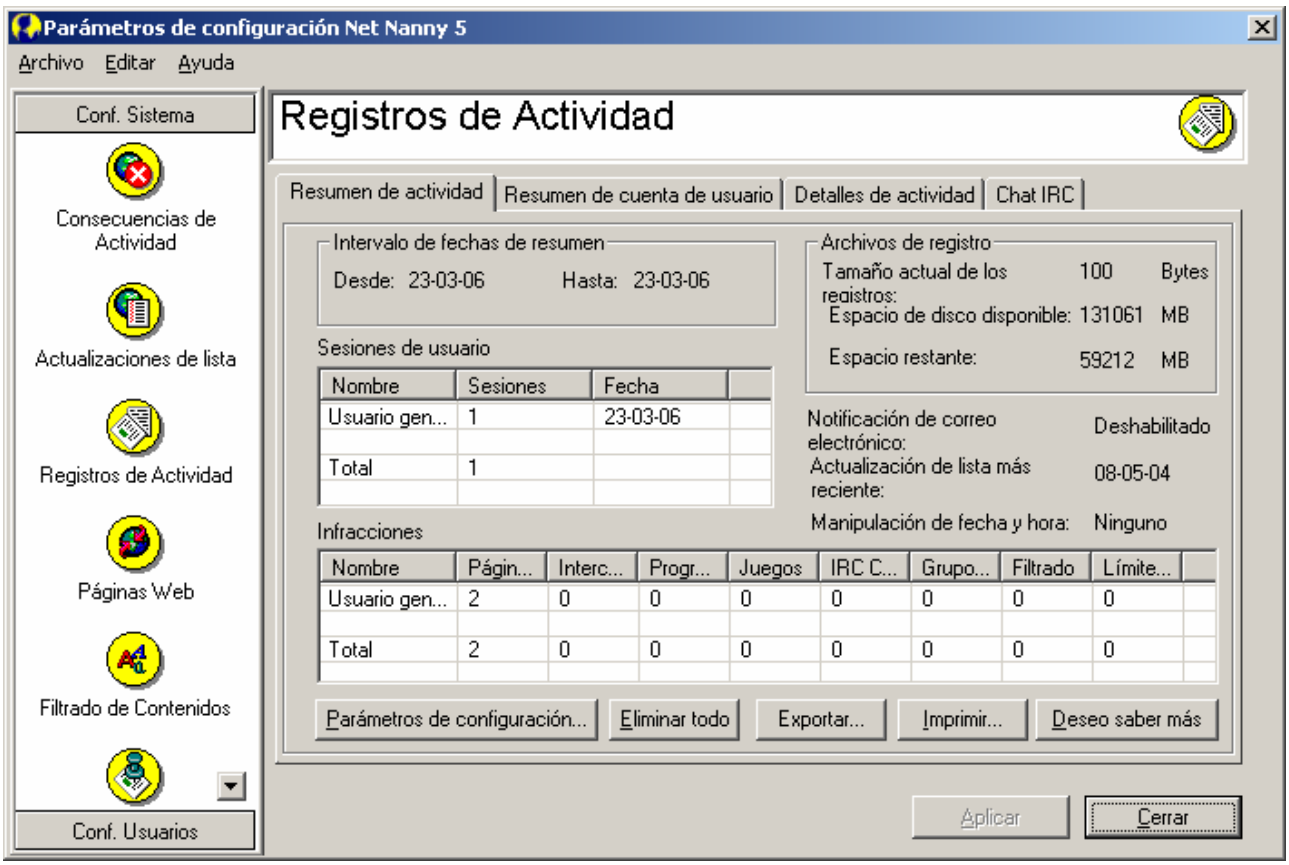

Figura 3 Registre d'activitat en Net Nanny (1)

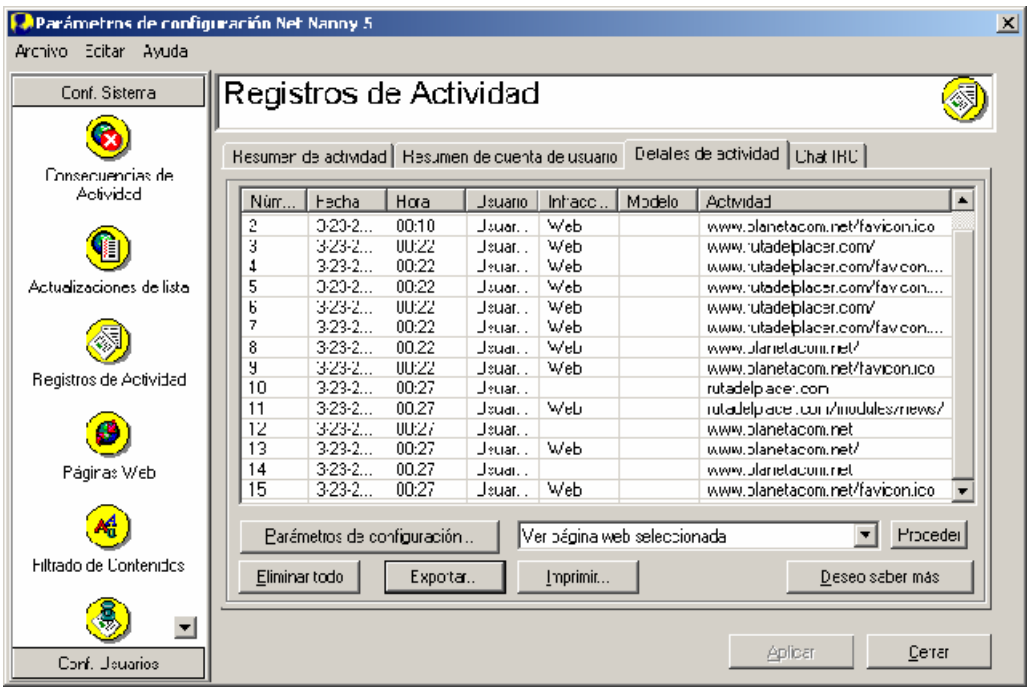

Figura 4 Registre d'activitat en Net Nanny (2)

Com hem dit anteriorment, la nostra aplicació no vol ser, en un principi, com una aplicació que ja està al mercat, sinó, més bé, senzilla d'utilitzar. I això queda reflectit en quant al tema d'introducció de les paraules o adreces que es volen filtrar: a la nostra aplicació les dades es fiquen a un fitxer pla, mentre que a d'altres aplicacions, i no totes, es poden fer al mateix programa, *Figura 5 Configuració de pàgines a bloquejar al programa Filtrar*.

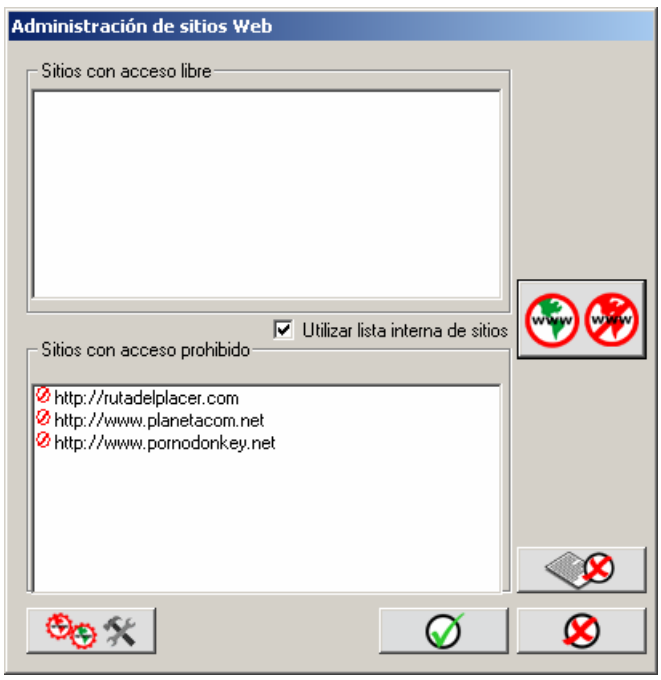

Figura 5 Configuració de pàgines a bloquejar al programa Filtrar

4.1 Justificació

El motiu de la meva selecció pel tema de filtre de pàgines web ha sigut un tema bastant personal i amb uns motius concrets:

- saber com es podia fer un proxy i que estigués en un segon pla, actuant d'una manera transparent per l'usuari
- sensibilització pel tema de seguretat d'un pare pels seus fills quan aquestos estan davant d'un ordinador navegant per Internet. Em complau molt saber que un pare estarà tranquil saben que el seu fill pot navegar per Internet perquè hi ha una aplicació que li pot filtrar tota mena d'informació que ell vulgui.
- una mica de curiositat, també, i empenta de nous coneixements vers un tema bastant de moda i poc conegut com són els proxys.

Així doncs, el tema de filtre de pàgines web l'he trobat molt interessant perquè obliga a conèixer una mica els diferents protocols que s'utilitzen normalment al navegar, com poden ser el TCP/IP, l'HTTP, l'FTP, etc.

També obliga a conèixer com utilitzar els fitxers per tal d'obrir-lo, llegir les paraules clau i ficar-les a un array i, finalment, tancar-lo.

Interessant ha sigut la utilització dels threads o fils d'execució i comprovar que es pot fer més d'un procés a la vegada a una velocitat ràpida degut a les característiques d'aquestos.

4.2 Objectius del TFC

Com s'ha dit anteriorment, aquest Treball té per objectiu el de poder filtrar pàgines web que continguin paraules clau que no volem que surtin per pantalla. Llavors, els objectius proposats han sigut els següents:

- Baix cost en el manteniment del programa
- Facilitat per afegir/esborrar paraules clau
- No mostrar per pantalla les pàgines web que continguin alguna paraula clau
- Control per mitjà d'un log dels accessos a pàgines que contenen alguna paraula clau

A més a més, una altra finalitat important alhora de poder filtrar les pàgines web podria ser:

- ajudar una empresa perquè els seus empleats no accedeixin a pàgines que han d'estar prohibides per a la seva tasca
- ajudar els col·legis al control indiscriminat per part dels alumnes a tota la xarxa
- ajudar els pares a tenir més confiança si deixen el nen davant d'un ordinador perquè el programa de filtre de pàgines web el mantindrà allunyat de llocs perversos o viciosos

4.3 Enfocament i mètode seguit

Primer de tot hem buscat força informació per Internet per veure quins programes realitzaven aquesta tasca de filtrar pàgines web. Després, els hem estudiat i fet un esquema de quines funcionalitats tenien cadascú i quines podrien ser les millores a aplicar. Hem observat que hi havien aplicacions que no tenien manera alguna de ficar les paraules o adreces a filtrar perquè ja les tenien predefinides. D'altres no gravaven cap registre dels accessos restringits, etc.

I el que hem intentat fer ha estat agafar coses de cada aplicació que hem vist i considerat interessant i crear-ne una aplicació nova i funcional de fàcil execució i enteniment.

Dels programes que hem trobat hem seleccionat uns quants i ens l'hem instal·lat i estudiat. Aquestos programes han estat:

- NetNanny
- Naomi
- Filtrar
- KidKey

Tots aquestos programes venen a fer el mateix, però uns permeten configurar més o menys paràmetres i tenir una auditoria sobre quins usuaris han volgut entrar a una pàgina prohibida o des de quina màquina s'ha intentat fer una connexió, quina és la paraula clau utilitzada, etc.

Una vegada observat com actuen aquestos programes hem buscat informació sobre els protocols HTTP i TCP/IP i l'estructura de les seves capçaleres i missatges que envien/reben.

També hem investigat una mica sobre proxys i firewalls, ja que aquestos últims també poden filtrar paquets i adreces que els hi podem predefinir.

Seguidament, hem fet un pseudocodi de l'estructura que tenia que fer el programa i quines funcions ens serien útils a l'hora de realitzar el programa i com fer-ho.

Una vegada que hem tingut clar què volíem hem començat a programar per fases, *veure apartat Planificació del Projecte*, on cada fase ha estat creada a partir de l'anterior.

4.4 Planificació del projecte

El TFC ha seguit un recorregut una mica sinuós amb uns quants entrebancs com podria ser el fet de com es pot executar un proxy en segon pla i com interceptar els missatges que circulen per un determinat port. Ha estat fruit de la consulta a un manual i la cerca per Internet per esbrinar com s'havia de fer. Una vegada fet això, un dels altres problemes, entre cometes, era que al carregar una i una altra vegada la mateixa pàgina havíem d'esborrar la memòria caché i les galetes, perquè si no carregava la mateixa pàgina i no mostrava els canvis que havíem fet.

4.4.1 Investigació

El primer que s'ha fet ha estat investigar a través de la xarxa els diferents programaris que hi ha sobre aquest tema. S'han estudiat i s'han fet les pertinents conclusions de com actuen i què fan. A partir d'aquí s'ha fet un petit esquema de com ho hauria de fer el programari que s'ha de desenvolupar i s'ha realitzat un primer pseudocodi de l'esquema.

No es pretén, ni molt menys, fer un programari del tot complet com n'hi han al mercat, però hem pogut constatar que en quant a funcionament pot ser millor perquè el nostre pot tenir un fitxer amb paraules a ometre, mentre que ens hem trobat algun que altre programari que ja porta incorporat això i no en permet fer cap canvi; és a dir, que no podríem afegir d'altres continguts per filtrar una nova pàgina.

S'ha afegit, també, un control del tipus log dels accessos restringits que s'han intentat fer. Aquest log tindrà el dia d'accés i l'hora.

4.4.2 Fase 1

En aquesta primera fase s'ha volgut tenir especial interès al tema del protocol que es preveu que s'ha d'utilitzar, com pot ser l'HTTP. S'ha buscat informació sobre les capçaleres i els tipus de missatges que es reben o s'envien quan es fa una connexió via HTTP.

4.4.3 Fase 2

Aquí hem començat a implementar l'aplicació mitjançant sockets per carregar una pàgina web en concret (una pàgina web de prova que s'ha seleccionat a l'atzar) i comprovar quins missatges rebíem, així com el temps d'accés i càrrega de la pàgina web. Hem tingut algun que altre error però ha estat ràpidament solucionat.

Els errors eren deguts a un mal tancament dels sockets; obríem un socket però no el tancàvem, amb el consegüent error. També hem tingut errors a l'hora de carregar un pàgina, ja que sempre intentàvem escriure-la i no havia de ser així en certs casos.

Quan això ha funcionat, hem introduït el tema dels threads, per poder executar més fils de l'aplicació; és a dir, que podem navegar per dos o més finestres i ho processarem tot des del mateix programa. Ha estat aquí quan hem tingut més problemes alhora de la implementació, per decidir-nos els paràmetres a utilitzar i com ho faríem.

Fer-lo o no amb threads suposa utilitzar més o menys memòria i ocupar més o menys recursos de la nostra màquina. A més a més, l'ús dels threads simplifica l'ús de recursos perquè totes les tasques comparteixen els mateixos recursos, al contrari que els processos.

4.4.4 Fase 3

Una vegada solucionat el tema dels threads, hem decidit com fer per trobar una paraula clau i on la tindríem emmagatzemada.

S'ha decidit crear un fitxer pla anomenat **FiltreWeb.txt** on a cada línia anirà una paraula clau, una paraula restringida. Es llegirà aquest fitxer i les dades (paraules) llegides s'emmagatzemaran a un array,.*Figura 6 Càrrega paraules clau a un Array*.

```
private File fitxerFiltre = new File("C:FiltreWeb.txt"); // Fitxer que conté les
paraules a filtrar
private String filtre[] = new String[20]; // Número màxim de paraules a filtrar

BufferedReader fisProxy = new BufferedReader(new FileReader(fitxerFiltre));
int i;
for (i=0; (filtre[i] = fisProxy.readLine()) != null; i++) ;
fisProxy.close();
```

Figura 6 Càrrega paraules clau a un Array

S'ha flexibilitzat l'aplicació; s'ha decidit que es llegirà el fitxer que conté les dades a filtrar cada vegada que s'executa un thread, perquè així es podrà modificar dinàmicament aquest fitxer i no farà falta reiniciar el navegador per poder llegir la llista de paraules a filtrar.

Per trobar la paraula clau dins una pàgina web s'ha decidit guardar les dades que es reben de la pàgina en un fitxer de text pla anomenat **DadesWeb.txt**, *Figura 7 Fitxer FiltreWeb.txt*. Seguidament, es llegeix i es comprova si dins les dades del fitxer existeix alguna paraula clau que hi hagi emmagatzemada a l'array creat anteriorment.

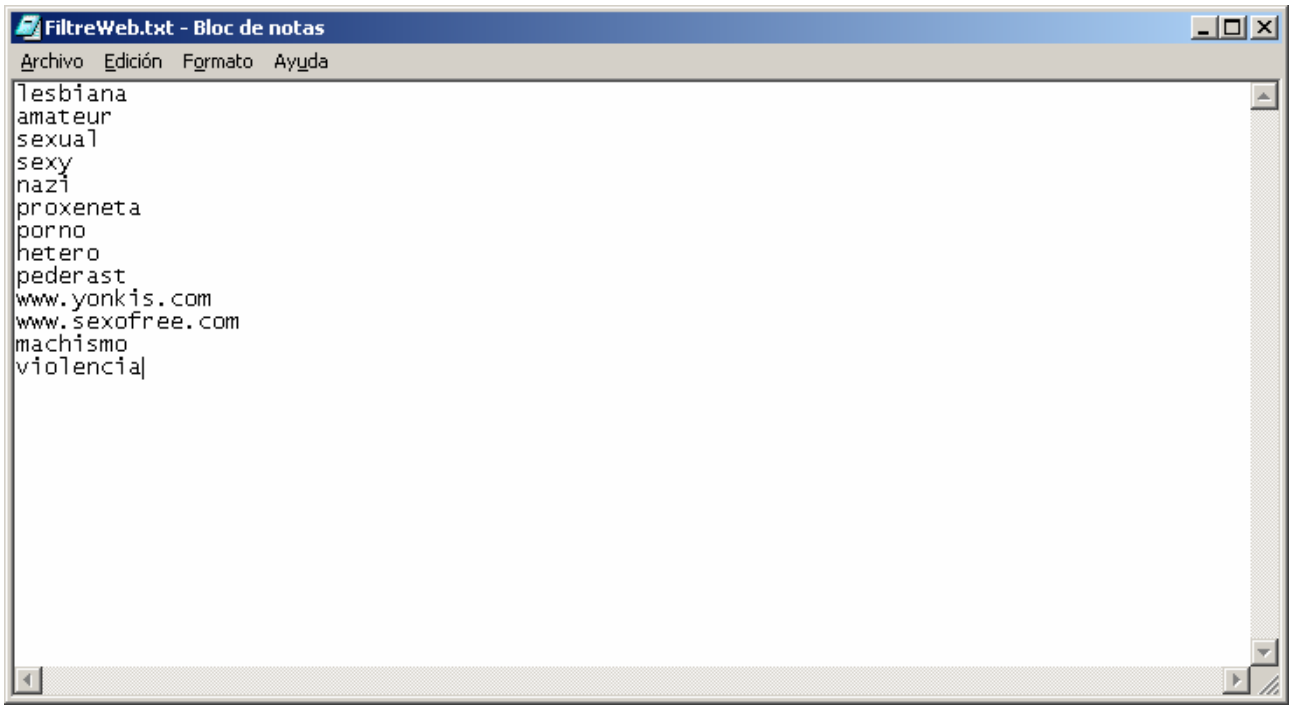

Figura 7 Fitxer FiltreWeb.txt

Si existeix una paraula clau, llavors s'afegirà a linia3, que contindrà uns marcadors en format HTML per numerar cada vegada que es troba una paraula, *Figura 8 Cerca de paraula clau a la pàgina web a carregar*.

```
    for (i = 0 ; filtre[i] != null; i++) {
        if (linia.toLowerCase().indexOf(filtre[i]) > 0 && linia != null) {
            //System.out.println ("Cadena trobada: " + filtre[i]);
            bError = true;
            byte b0[] = adreca.getBytes();
            byte b1[] = filtre[i].getBytes();
            gravaExcepcio(adreca, filtre[i].toString(), host);
            linia3 = linia3 + "<LI><b> Adreça: </b>" + adreca.toString() +
            "</LI> <b>Paraula: </b>" + filtre[i].toString() + "</P>\r\n";
        }
    }
    if (linia3 != "")
    {
        bProcesa = false;
        bParaula = false;
        fis.close();
        return (presentaWebError(sb, header,  "<OL>\r\n" + linia3, out, host));
    }
}
```

Figura 8 Cerca de paraula clau a la pàgina web a carregar

4.4.5 Fase 4

Ara que ja funciona el programa, per millorar-lo, s'ha decidit de gravar els accessos que tenen una paraula clau dins d'un fitxer log, *Figura 9 Contingut del fitxer LogTFCProxy.txt*. Així, gravarem la pàgina a la qual es volia accedir, el dia de l'incidència, l'hora de l'incidència i quina o quines paraula/paraules s'ha/n trobat. Amb això es pot tenir un fitxer de referència per saber on s'intenta accedir i quines paraules es busquen.

Si a una pàgina que s'intenta visitar conté més d'una paraula clau, llavors es gravarà el nom de la pàgina i totes les paraules clau que en contenia.

El fitxer log es pot guardar amb un nom diferent i es podria tenir un històric d'accessos restringits o prohibits.

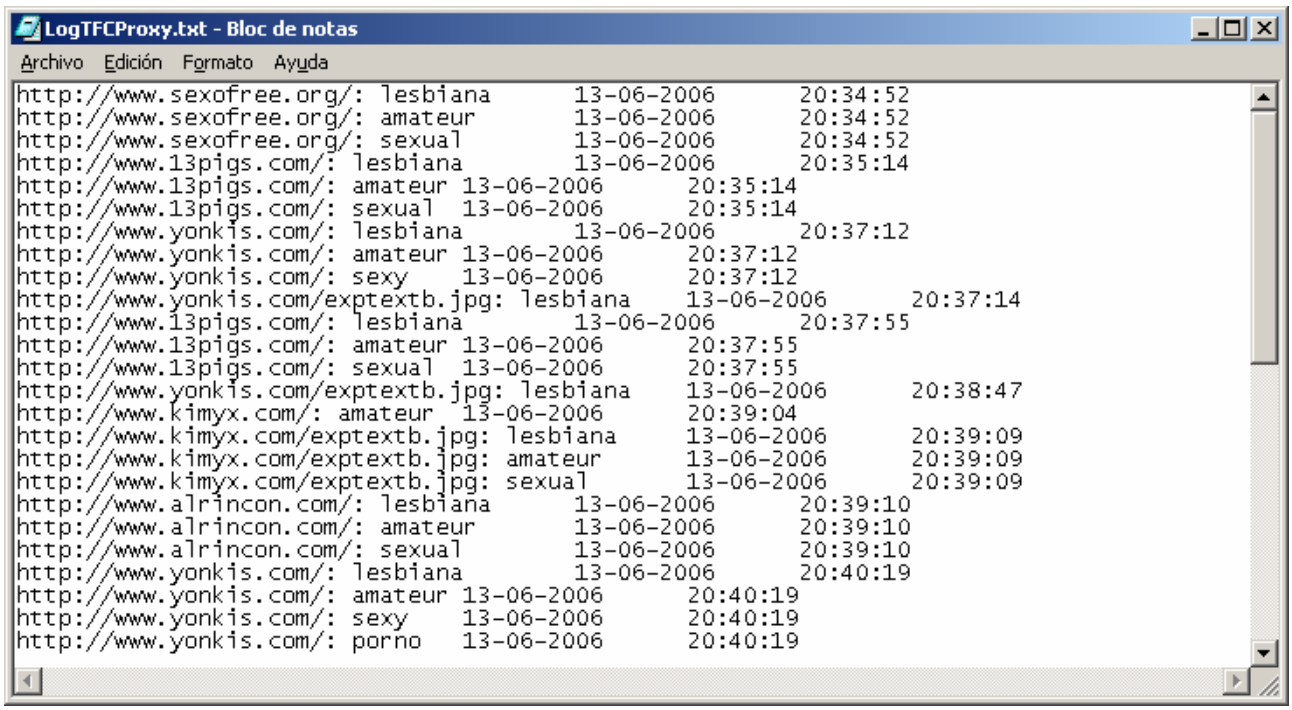

Figura 9 Contingut del fitxer LogTFCProxy.txt

En aquesta fase també s'ha volgut donar al programa d'una millora visual quan es bloqueja la pàgina prohibida. S'ha generat una senzilla pàgina en format HTML, *Figura 10 Pàgina en format HTML indicant l'intent d'accés a una pàgina i les paraules-clau a la pàgina*, per indicar que s'ha bloquejat la pàgina pel proxy TFCProxy, quina és la pàgina què es volia accedir i quina és la paraula clau causant del bloqueig.

Aquest fitxer en format HTML es diu TFCProxy.html i no és més que una simple pantalla mostrant un error o bloqueig d'accés. A més a més, les dades d'aquest fitxer es veuran modificades per presentar per pantalla quina és la pàgina web a visitar i quina o quines paraula o paraules clau s'han trobat en aquesta pàgina web.

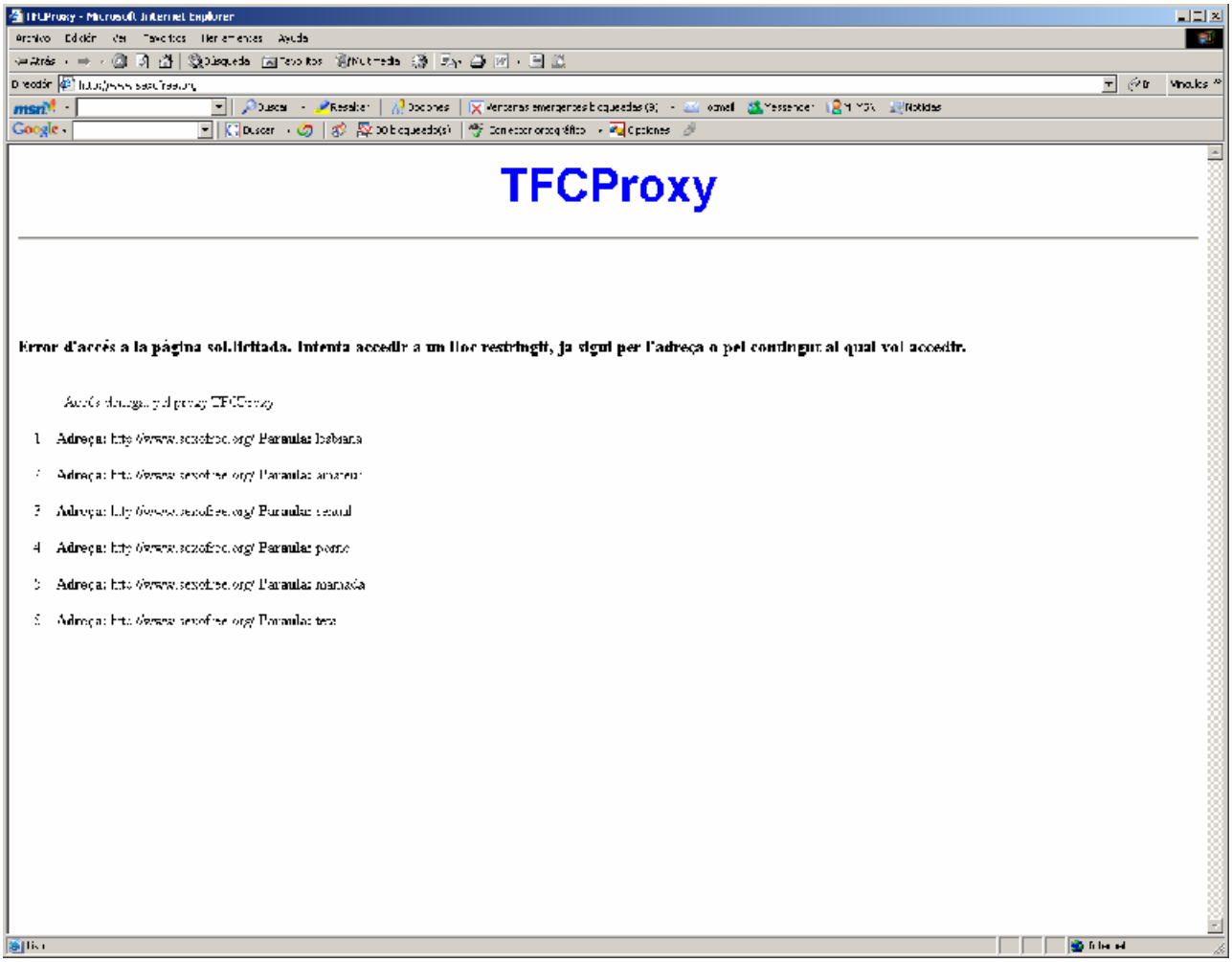

Figura 10 Pàgina en format HTML indicant l'intent d'accés a una pàgina i les paraules-clau a la pàgina

4.5 Productes obtinguts

Al llarg del desenvolupament del procés de l'aplicació ens hem trobat amb una sèrie de productes totalment distants entre sí, amb una sèrie de funcionalitats diferents. I és que els primers productes eren una sèrie de proves per comprovar les connexions amb sockets i temps de càrrega de les pàgines visitades.

En versions posteriors es va anant afegint funcionalitats com lectura del fitxer que conté les dades o paraules que es volen filtrar, escriptura/lectura de la pàgina web què es vol accedir, càrrega de la pàgina indicant que no denega el servei i se'n va treure'n d'altres com missatges per pantalla per indicar què feia el procés, el temps de càrrega d'una pàgina web, etc.

4.6 Breu descripció dels altres capítols de la memòria

Aquesta aplicació no es podria fer sense tenir una sèrie de coneixements previs sobre certs capítols importants com poden ser els diferents protocols que operen a Internet, així com els missatges que s'intercanvien i s'han de processar. Perquè, si no fos així, processaríem totes les pàgines, continguin o no errors, i podríem donar resultats erronis.

A l'apartat 5, doncs, expliquem el tema dels protocols que ens interessa per la nostra aplicació, així com un apartat diferenciant què és un firewall i què és un proxy. També expliquem certs temes importants del nostre aplicatiu com les connexions que es realitzen a través del sockets per un port.

A l'apartat 6, fem una descripció de quins requisits ha de tenir el servidor, les funcionalitats addicionals que s'han anat afegint i les diferents aplicacions que hem trobat.

En quant a la implementació del TFCProxy, apartat 7, hem fet una anàlisi del codi que s'ha utilitzat, els diferents productes que s'han anat obtenint i quines possibles millores es podrien afegir a l'aplicació, com la parametrització per finestres, visualització de logs de forma més fàcil, etc.

Al manual de funcionament, apartat 8, es descriu quins són els requisits previs per fer la connexió, com s'ha de realitzar la instal·lació del programari, quina configuració s'ha de realitzar i quin ha estat el joc de proves que s'ha utilitzat.

A valoracions econòmiques, apartat 9, es descriu quines avantatges té la utilització d'aquesta aplicació en termes econòmics; l'estalvi que pot suposar la introducció de l'aplicació a una empresa i, també, l'estalvi en temps de treball dels usuaris en tasques alienes a la seva feina.

A l'apartat 10 s'han descrit les conclusions que hem obtingut durant el període del TFC.

S'ha fet un petit glossari, apartat 11, de quines paraules són les més interessants emprades en aquesta memòria, així com la bibliografia, apartat 12, utilitzada pel desenvolupament del TFC, que ha estat, sobretot, per mitjà de cerques per Internet.

5 CONEIXEMENTS PREVIS

5.1 El protocol HTTP

El protocol HTTP (HyperText Transfer Protocol) és el protocol emprat en cada transacció de la web (WWW). L'hipertext és el contingut de les pàgines web i el protocol de transferència és el sistema pel qual s'envien les peticions d'accedir a una pàgina web, i la resposta d'aquesta pàgina web, remitint la informació que es veurà a pantalla. També serveix el protocol per enviar informació addicional en ambdós sentits, com els formularis amb missatges i altres similars.

L'HTTP és un protocol sense estats; és a dir, que no guarda cap informació sobre connexions anteriors. Al finalitzar la transacció totes les dades es perden. És per això que es van popularitzar les *cookies*, que són petits fitxers guardats al propi ordinador que pot ser un lloc web al establir connexió amb ell, i d'aquesta manera reconèixer a un visitant que ja va estar en aquest lloc anteriorment. Gràcies a aquesta identificació, el lloc web pot emmagatzemar un gran número d'informació sobre cada visitant, oferint-li, així, un millor servei.

5.1.1 Versions del protocol HTTP

5.1.1.1 Versió 1.0

Aquesta primera versió del protocol és molt senzilla i el client solament podia invocar tres operacions en el servidor

- GET: Per demanar una pàgina
- HEAD: Per demanar la capçalera d'una pàgina
- POST: Per enviar dades a una URL

Veiem el seu funcionament:

- El client envia una petició al servidor i aquesta petició està composada per un mètode a invocar al servidor (URI: Uniform Resource Identifier) i una versió del protocol, seguit d'un missatge compatible amb MIME (Multiporpose Internet Mail Extensions) amb els paràmetres de la petició, informació del client i un cos opcional amb més dades pel servidor. Exemple:

  - GET /index.html HTTP/1.0
  - Accept: text/plain
  - Accept: text/html
  - Accept: */*
  - User-Agent: Un Agent d'Usuari qualsevol.

- El servidor respon amb una línia d'estat, incloent la versió del protocol del missatge i si la petició ha tingut èxit o fracàs, amb un codi de resultat, seguit d'un missatge compatible amb MIME amb informació del servidor, metainformació (dades sobre la informació) de l'entitat sol·licitada i un cos opcional amb l'entitat sol·licitada. Exemple:
  - HTTP/1.0 200 OK
  - Server: MDMA/0.1
  - MIME-version: 1.0
  - Content-type: text/html
  - Last-Modified: Sat Mar 18 17:00:32 2006
  - Content-Length: 1547
  - <title>Pàgina de proves</title>
  - <hr>
  - ...
  - <hr>

Característiques de la versió 1.0:

- Les connexions del protocol TCP són lentes d'establir (connexió en tres pasos i ajust de finestres de recepció de dades) i com per cada pàgina i per a cada imatge que hi hagi a la pàgina ha d'establir-se una nova connexió, la transmissió de dades està ralentida per l'establiment de la connexió TCP.
- Una connexió per transmetre 1 KByte de dades triga al voltant de 500 ms.
- Quan es tanca la connexió TCP, el port del servidor utilitzat a la connexió es queda en estat TIME_WAIT un temps recomanat 240 segons, pel qual un servidor que rebi moltes peticions pot esgotar tots els ports TCP (65535) i deixar el servidor sense possibilitat d'enviar cap tipus de dades. Això suposa un problema d'escalabilitat molt important.

5.1.1.2 Versió 1.1

Aquesta versió és més potent que la anterior i té tretze mètodes diferents, a més d'un conjunt de característiques noves, com per exemple el temps pel qual el client ha de tornar a carregar la pàgina.

Característiques de la versió 1.1:

- Són connexions persistents (keep-alive): ja no es tanca la connexió quan s'ha realitzat l'enviament de cada part d'un document, evitant la sobrecàrrega de l'establiment de connexions TCP.
- Moltes peticions simultànies: un client pot realitzar vàries peticions utilitzant una única connexió, sense esperar la resposta del servidor per a cada una d'elles.
- Negociació del contingut: s'assignen diferents valors a les característiques de la comunicació, entre ells quant es pot degradar la qualitat de la connexió.
- Nous mètodes: juntament amb GET, POST i HEAD apareixen els mètodes :
  - DELETE per esborrar un recurs associat a l'URI d'esborrat;
  - TRACE per veure què està rebent el servidor del qual ell envia.;
  - PUT per enviar dades a un recurs associat a una URI:
  - PATCH per aplicar correccions en un recurs associat a una URI;
  - COPY per copiar uns recursos identificats per una URI en un altre lloc determinada URI a destí determinat;
  - MOVE per moure el recurs identificat per l'URI a un altre lloc;
  - DELETE per esborrar un recurs associat a una URI;
  - LINK per establir enllaços entre diferents recursos;
  - ULINK per treure enllaços establerts prèviament per LINK;
  - OPTIONS per a que el client pugui obtenir del servidor les seves característiques;
  - WRAPPED que permet unir vàries peticions i recobrir-les amb algun tipus de filtrat (per exemple encriptació).
- Nou mètode d'autenticació: a la RFC 2069 es descriu un nou mètode d'autenticació, en el qual les claus d'accés van encriptades per la xarxa, al contrari del que ocorre en HTTP 1.0.

5.1.1.3 HTTP-NG (HTTP Next Generation)

Aquest nou protocol pretén cobrir una gran quantitat de noves funcionalitats, entre les quals destaca el comerç electrònic. Els criteris del seu disseny han sigut:

- **Simplicitat** (seguint el criteri de l'HTTP/1.0) a l'hora de la implementació del protocol.
- **Rendiment**: ha de ser eficient transmetent objectes en xarxes de comunicacions.
- **Asincronia**: les peticions des dels clients han de poder-se fer en paral·lel a través d'una única connexió.
- **Seguretat**: els objectes que es transmeten han d'anar encriptats sense forçar cap política de seguretat en particular.
- **Autenticació**: s'ha de poder autenticar les dues parts de la connexió, així com a qualsevol intermediari, amb suport a la realització de pagaments en línia.
- **Servidors intermediaris**: s'ha de suportar la comunicació entre servidors per al manteniment de 'caches', 'mirrors' (miralls de dades) i 'proxys' (intermediaris de comunicació).
- **Visualització obligatòria** de certes dades: s'ha de poder obligar el client a mostrar certes dades al voltant de l'objecte que s'està transmetent, com l'autor, el copyrigth i la llicència.
- **Informació de registre**: la informació de registre (logs) ha de poder ser enviada entre diferents servidors.
- **Requeriments de xarxa**: el protocol ha de treballar de forma independent de la capa o trama de transport de què disposi, tot i que ha de funcionar especialment bé amb el protocol TCP per ser el més utilitzat a Internet.

Missatges d'estat

Existeixen cinc categories de missatges d'estat, organitzades pel primer dígit del codi numèric de la resposta:

Codi	***Descripció***
1xx	Missatges informatius. Per ara (en HTTP/1.0) no s'utilitzen i estan reservats per un futur ús
2xx	Missatges associats amb operacions realitzades correctament
3xx	Missatges de redirecció, que informen d'operacions complementàries que es tenen que realitzar per finalitzar l'operació
4xx	Errors del client; el requeriment conté algun error o no pot ser realitzat
5xx	Errors del servidor que no ha pogut portar a terme una sol·licitud

Taula 1 Categories dels missatges d'estat del protocol HTTP

Els missatges més comuns es recullen a la següent taula:

Codi	***Comentari***	***Descripció***
200	OK	Operació realitzada satisfactòriament
201	Created	La operació ha sigut realitzada correctament, i com a resultat s'ha creat un nou objecte, URL de la qual dóna accés es proporciona en el cos de la resposta. Aquest nou objecte ja està disponible. Pot ser utilitzat en sistemes d'edició de documents
202	Accepted	La operació ha sigut realitzada correctament, y com a resultat s'ha creat un nou objecte, URL de la qual l'accés es proporciona en el cos de la resposta. El nou objecte no està disponible pel moment. En el cos de la resposta es té d'informar sobre la disponibilitat de la informació
204	No Content	La operació ha sigut acceptada, però no ha produït cap resultat d'interès. El client no deurà modificar el document que està mostrant en aquest moment
301	Moved Permanently	L'objecte al qual s'accedeix ha sigut mogut a un altre lloc de forma permanent. El servidor proporciona, a més, la nova URL a la variable Location de la resposta. Alguns browsers accedeixen automàticament a la nova URL. En cas de tenir capacitat, el client pot actualitzar la URL incorrecta, per exemple, a l'agenda de *bookmarks*
302	Moved Temporarily	L'objecte al qual s'accedeix ha sigut mogut a un altre lloc de forma temporal. El servidor proporciona, a més, la nova URL a la variable Location de la resposta. Alguns browsers accedeixen automàticament a la nova URL. El client no ha de modificar cap de les referències a la URL errònia.
304	Not Modified	Quan es fa un GET condicional, i el document no ha sigut modificat, se retorna aquest codi d'estat
400	Bad Request	La petició té un error de sintaxis i no es entesa pel servidor
401	Unauthorized	La petició requereix una autorització especial, que normalment consisteix en un nombre y clau que el servidor verificarà. El campo WWW-Autenticate informa dels protocols d'autentificació acceptats per aquest recurs
403	Forbidden	Està prohibit l'accès a aquest recurs. No es possible utilitzar una

		clau para modificar la protecció
404	Not Found	La URL sol·licitada no existeix
500	Internal Server Error	El servidor ha tingut un error intern, y no pot continuar amb el processament
501	Not Imple-mented	El servidor no té capacitat, pel seu disseny intern, per portar a terme el requeriment del client
502	Bad Ga-teway	El servidor, que està actuant com a proxy o passarel·la, ha tractat un error a l'accedir al recurs que havia sol·licitat el client
503	Service Unavailable	El servidor està actualment deshabilitat, i no es capaç d'atendre el requeriment

Taula 2 Descripció Missatges d'estat del protocol HTTP

5.2 Què és un Firewall

Un firewall, *Figura 11*, és un dispositiu de seguretat que actua com un tallafocs entre xarxes, permetent o denegant les transmissions d'una xarxa a una altra. Un ús típic és situar-lo entre una xarxa local i la xarxa Internet, com a dispositiu de seguretat per evitar que els intrusos puguin accedir a informació confidencial.

Un firewall és simplement un filtre que controla totes les comunicacions que passen d'una xarxa a l'altra i en funció del que sigui permet o denega el seu pas. Per permetre o denegar una comunicació, el firewall examina el tipus de servei al que correspongui, com poden ser el web, el correu o l'IRC. Depenent del servei, el firewall decideix si el permet o no. A més a més, examina si la comunicació és entrant o sortint i depenent de la seva adreça la pot permetre o no.

D'aquesta manera, un firewall pot permetre des d'una xarxa local cap a Internet serveis de Web, correu i ftp, però no a IRC que pot ser innecessari pel nostre treball. També podem configurà els accessos que es facin des d'Internet cap a la xarxa local i podem denegar-los tots o permetre alguns serveis com el de la web. Depenent del firewall que tinguem, també podrem permetre alguns accessos a la xarxa local des d'Internet si l'usuari s'ha autenticat com usuari de la xarxa local.

Tipus de firewall

Un firewall pot ser un dispositiu software o hardware; és a dir, una aparell que es connecti entre la xarxa i el cable de la connexió a Internet o bé un programa que s'instal·la a la màquina que té el

mòdem que connecta amb Internet. Fins i tot, es pot trobar ordinadors molt potents amb software específic que l'únic que fan és monitoritzar les comunicacions entre les xarxes.

5.2.1 Software

És un programa que s'encarrega de filtrar els missatges que arriben entre el servidor i Internet. Examina tots els paquets que entren o surten del seu servidor. També control·la els ports de l'ordinador, que ningú s'intenti colar per ports que estan oberts.

Funciona per mitjà d'unes polítiques de seguretat, que consisteixen en una sèrie de regles que han de complir tots els paquets de dades i les sessions TCP/IP que vulguin establir amb el seu equip. Posteriorment, és el propi firewall el que s'encarrega de verificar que tots i cada un dels paquets amb destí al seu servidor les compleixin, descartant aquells que no ho facin.

5.2.2 Hardware

N'hi han dos accepcions:

- Una és la que alguns diuen a aquells firewalls que separen dues xarxes
- Una altra als que proveeixen de serveis de firewall a l'equip de treball

En general, són equips amb un sistema operatiu sobre el que corren programes de routing, firewalling, etc i això se li diu firmware.

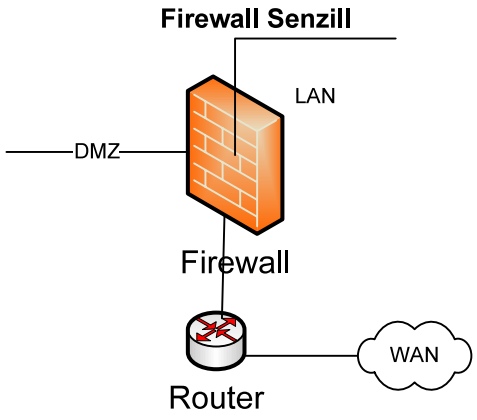

Figura 11 Firewall

5.3 Què és un Proxy

Un proxy és un software que obre un socket en un determinat port al nostre host, mitjançant el qual s'escolten les peticions d'Internet de la resta de màquines de la nostra xarxa local.

Per una altra part, el proxy reserva un espai de disc de longitud variable al qual denominarà caché. El software pren una d'aquestes peticions i el que fa tot seguit és buscar a la caché per veure si ja existeix una còpia de la pàgina o objecte que està sol·licitant el client. Si ja existeix solament agafarà aquesta còpia existent al nostre disc dur i l'enviarà al client sol·licitant. En canvi, si no existeix aquesta còpia, el proxy tindrà que baixar el contingut d'Internet per poder enviar-lo al client. En aquest cas, a més de fer aquest enviament, s'encarrega d'afegir aquest nou objecte o arxiu a la caché pel seu ús posterior.

La caché utilitza, normalment, un algoritme per determinar quan un document està obsolet i s'ha d'eliminar de la caché, depenent de la seva antiguitat, tamany i històric d'accés. Dos d'aquestos algoritmes bàsics són l'LRU (l'usat menys recentment, "Least Recently Used") i l'LFU (l'usat menys freqüentment, "Least Frequently Used").

Els proxis web també poden filtrar el contingut de les pàgines Web servides. Algunes aplicacions que intenten bloquejar contingut Web ofensiu estan implementades com proxies Web. Altres tipus de proxy canvien el format de les pàgines web per a un propòsit o una audiència específics, com per exemple mostrar una pàgina a un telèfon mòbil o una PDA. Alguns operadors de xarxa també tenen proxies per interceptar virus i altres continguts hostils servits per pàgines web remotes

Proxy NAT (Network Address Translation) / Enmascarament

Un altre mecanisme per fer d'intermediari a una xarxa és el NAT.

La traducció d'adreces de xarxa (Network Address Translation) també és coneguda com enmascarament d'IPs. És una tècnica mitjançant la qual les adreces font o destí dels paquets IP són reescrites, substituïdes per altres (d'aquí el nom "enmascarament").

Això és el que passa quan molts usuaris comparteixen una única connexió a Internet. Es disposa d'una única adreça IP pública que té compartida. Dintre de la xarxa d'àrea loca (LAN) els equips utilitzen adreces IP reservades per a ús privat i serà el proxy l'encarregat de traduir les adreces privades a aquesta única adreça per realitzar les peticions, així com de distribuir les pàgines rebudes a aquell usuari intern que la va sol·licitar.

Aquesta situació és molt comú a empreses i domicilis amb molts ordinadors en xarxa i un accés extern a Internet. L'accés mitjançant NAT proporciona una certa seguretat, ja que en realitat no hi ha connexió directa entre l'exterior i la xarxa privada i així els nostres equips no estan exposats a atacs directes des del exterior.

Mitjançant NAT també es pot permetre un accés limitat des del exterior i fer que les peticions que arriben al proxy siguin dirigides a una màquina concreta que hagi sigut determinada per a tal propòsit en el propi proxy.

La funció NAT resideix en els Tallafocs i resulta molt còmoda perquè no necessita de cap configuració especial en els equips de la xarxa privada que puguin accedir a través d'ell com si fos un encaminador.

Tipus de proxy:

- Servidor proxy web. És el que es coneix com a proxy. Intercepta la navegació dels clients per pàgines web i ho fa atenent a criteris de seguretat, rendiment, anonimat, ...
- Proxy FTP : Inclouen cada fitxer que un es descarregui. Inclús pot filtrar les paraules "inapropiades" dels llocs que visiti o escanejar aquests llocs en busca de virus.
- Proxy ARP. Fa d'enrutador en una xarxa perquè fa d'intermediari entre ordinadors.

5.3.1 Avantatges

- Control. Solament l'intermediari fa el treball real, per tant es poden limitar i restringir els drets dels usuaris i donar permisos solament al proxy
- Estalvi. Solament a un dels usuaris (el proxy) ha d'estar equipat per fer el treball real
- Velocitat. Si molts clients volen demanar el mateix recurs, el proxy pot fer caché: emmagatzema la resposta d'una petició per donar-la directament quan un altre usuari la demani. Així no té que tornar a contactar amb el destí i acaba més ràpidament.
- Fitrat. El proxy pot negar-se a respondre algunes peticions si detecta que han sigut prohibides

- Modificació. Com intermediari que és, un proxy pot falsificar informació o modificar-la seguint un criteri.
- Anonimat. Si tots els usuaris s'identifiquen com un de sol, és difícil que el recurs accedit pugui diferenciar-los. Però, això pot ser dolent, per exemple, quan s'ha de fer necessàriament la identificació

5.3.2 Desavantatges

- Abús. Al estar disposat a rebre peticions de molts usuaris i respondre.-les, és possible que faci alguna tasca que no li pertoca. Per tant, ha de controlar qui té accés i qui no als seus serveis, cosa que normalment és molt difícil.
- Càrrega. Un proxy ha de fer el treball de molts usuaris
- Intromissió. És un pas més entre origen i destí, i alguns usuaris poden no voler passar pel proxy, i molt menys si aquest fa còpia de les dades
- Incoherència. Si fa de caché, és possible que s'equivoqui i doni una resposta antiga quan hi ha una de més recent en el recurs de destí.
- Irreguratitat. El fet que el proxy representi a més d'un usuari dóna problemes en molts escenaris, en concret als que pressuposen una comunicació directa entre un emissor i un receptor (com TCP/IP).

5.3.3 Funcionament

Un proxy permet a altres equips connectar-se a una xarxa de forma indirecta a través d'ell. Quan un equip de la xarxa desitja accedir a una informació o un recurs, és realment el proxy qui realitza la comunicació i a continuació trasllada el resultat a l'equip inicial. En uns casos això es fa així perquè no és possible la comunicació directa i en altres perquè el proxy afegeix funcionalitat addicional, com pot ser la de mantenir els resultats obtinguis (com una pàgina web) en una caché que permeti accelerar successives consultes coincidents.

5.3.4 Diferència entre un proxy i un firewall

El proxy i el firewall són diferents, *Figura 12 Proxy i firewall*, però deurien estar sempre combinats. El proxy s'usa per a redirigir les peticions que rep de varis usuaris a Internet de forma transpa-

rent i s'encarrega de retorna'ls-hi les respostes (les pàgines web). També es pot utilitzar per FTP, POP3, SMTP, IMAP, TELNET, etc.

El firewall, si més no, és unicament un mètode de protecció de la xarxa local o d'un ordinador personal, amb el qual podem tancar o deixar oberts certs ports, IP's, aplicacions, etc.

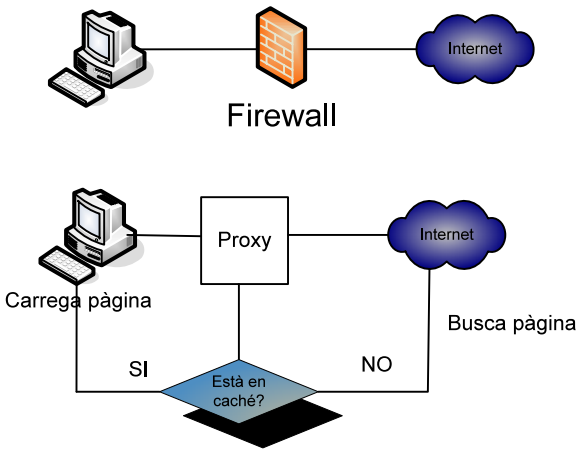

Figura 12 Proxy i firewall

5.4 Ports

En informàtica, un port és una forma genèrica de denominar a una interfície per la que diferents tipus de dades poden ser enviades i rebudes. Aquesta interfície pot ser física o a nivell de programari (per exemple, els ports que permeten la transmissió de dades entre diferents ordinadors).

Cada sistema operatiu ofereix una interfície i uns mecanismes per a que els processos puguin utilitzar els ports. Generalment l'operatiu assigna una cua finita per anar desant els missatges que arriben a un determinar port. L'accés sol ser síncron (el procés queda blocat si llegeix dades d'una cua buida).

El fet de permetre o denegar accés als ports és important perquè les aplicacions servidores (que accepten connexions) han "d'escoltar" en un port conegut d'avantmà per a què un client (que inicia la connexió) pugui connectar-se. Això vol dir que quan el sistema operatiu rep una petició per un port concret, li passa a l'aplicació que escolta per aquell port i a cap altre. Si no hi hagués cap aplicació escoltant pel port no es podria establir la connexió.

5.4.1 Tipus de ports

5.4.1.1 Port hardware

Un port hardware (o port de maquinari) permet acoblar a un sistema físic un connector o cable. Per exemple, la majoria dels ordinadors personals tenen un port pel teclat i un altre pel ratolí a on són connectats aquests perifèrics. Els ports hardware poden gairebé sempre dividir-se en dos grans grups: els que poden enviar i rebre només un bit a l'hora a través d'un cable (anomenats ports sèrie) i els que poden enviar diversos bits a la vegada utilitzant un conjunt de cables (anomenats ports paral·lels).

5.4.1.2 Port de xarxa

Un port de xarxa és una interfície utilitzada per comunicar-se amb un programa a través d'una xarxa. Els ports de xarxa acostumen a estar numerats i una certa implementació de protocol de transmissió de xarxa (com TCP o UDP) assigna algun d'aquests números de port a la informació que envia; la implementació del protocol en el destí farà servir aquest número per decidir a quin programa entregar les dades rebudes.

A TCP i UDP la combinació d'un número de port i una adreça de xarxa (adreça IP) acostuma a anomenar-se socket.

- Els números de ports de les aplicacions client són assignats dinàmicament i generalment són superiors al 1024. Quan una aplicació client vol comunicar-se amb un servidor busca un número de port lliure i el fa servir.
- Les aplicacions servidores utilitzen números de ports prefixats, inferiors a 1024, els anomenats ports *well-known* (ben coneguts), doncs són els mateixos en tots els hosts. Aquests ports estan definits a la RFC 1700, que són els documents on es va començar a publicar l'estàndar TCP/IP, i es poden consultar a http://www.ietf.org/rfc/rfc1700.txt

En una URL (Universal Resource Locator) els ports es denoten amb ':' a continuació del nom de la màquina. Per exemple *http://www.ccd.uab.es:80/material/24998-rmi-pract.pdf* vol dir que estem demanant el document "24998-rmi-pract.pdf" mitjançant HTTP connectant-nos al port 80 d'aquest servidor. Com que 80 és el port per defecte per a HTTP es pot ometre.

5.4.1.3 Port d'E/S (entrada/sortida) o port màquina – E/S mapejada a memòria

Gairebé totes les famílies de processadors fan servir les mateixes instruccions en ensamblador per accedir tant a la memòria com als registres de hardware. Tot i així, els microprocessadors Intel tenen instruccions (de IN i OUT) les quals són utilitzades específicament per E/S. Aquestes instruccions decideixen amb quin dispositiu hardware comunicar-se gràcies al concepte de port E/S (o port màquina), els quals estan numerats basant-se en el dispositiu hardware al que fan referència.

Els processadors Intel normalment permeten enviar o rebre un octet (byte) a cada instrucció. El dispositiu hardware decideix com interpretar aquesta informació que li va ser enviada i amb què respondre al processador.

5.4.2 Estat dels ports

Un port pot estar:

- **Obert**: Accepta connexions. Hi ha una aplicació escoltant pel port. Això no significa que es tingui accés a l'aplicació, només que hi ha la possibilitat de connectar-s'hi.
- **Tancat**: Es rebutja la connexió. Probablement no hi hagi cap aplicació escoltant o no es permet l'accés per algun motiu. Aquest és el comportament normal del sistema operatiu.
- **Bloquejat o silenciós**: No hi ha resposta. Aquest és l'estat ideal per a un client a Internet, d'aquesta manera ni tan sols se sap si l'ordinador està connectat. Normalment aquest comportament es deu a un tallafocs o a que l'ordinador estigui apagat.

5.5 Sockets

Els sockets són punts finals d'enllaços de comunicacions entre processos. Els processos els tracten com descriptors de fitxers, de forma que es poden intercanviar dades amb altres processos transmetent i rebent a través de sockets. El tipus de sockets descriu la forma en la que es transfereix informació a través del socket.

N'hi han diferents tipus de sockets.

5.5.1 Sockets Stream (TCP)

Són un servei orientat a connexió, on les dades es transfereixen sense enquadrar-los en registres o blocs.

El protocol de comunicacions amb streams és un protocol orientat a connexió, ja que per establir una comunicació utilitzant el protocol TCP s'ha d'establir en primer lloc una connexió entre un parell de sockets. Mentre un dels sockets atén peticions de connexió (servidor), l'altre sol·licita una connexió (client). Una vegada que els dos sockets estiguin connectats, es poden utilitzar per transmetre dades en ambdós direccions.

5.5.2 Sockets Datagrama (UDP)

Són un servei de transport sense connexió. Són més eficients que TCP, però en la seva utilització no està garantida la fiabilitat. Les dades s'envien i reben en paquets, entrega que està garantida. Els paquets poden ser duplicats, perduts o arribar en un ordre diferent al qual es va enviar.

El protocol de comunicacions amb datagrames és un protocol sense connexió, és a dir, cada vegada que s'enviïn datagrames és necessari enviar el descriptor del socket local i l'adreça del socket que ha de rebre el datagrama.

5.5.3 Diferència entre multitasca i multiprocés.

El multiprocés es refereix a dos programes que s'executen "aparentment", a la vegada, sota el control del Sistema Operatiu. Els programes no necessiten tenir relació entre sí, simplement el fet què l'usuari desitgi que s'executin a la vegada.

Multitasca es refereix a que dues o més tasques s'executen "aparentment" a la vegada, dintre d'un mateix programa.

Diem "aparentment" en ambdós casos, perquè normalment les plataformes tenen una única CPU, amb el qual, els processos no s'executen en realitat "concurrentment", sinó que comparteixen la CPU. En plataformes amb vàries CPU, sí és possible que els processos s'executin realment al mateix temps.

Tant en multiprocés com en multitasca (multifil), el Sistema Operatiu s'encarrega que es generi la il·lusió que tot s'executa a la vegada. Si més no, la multitasca pot produir programes que realitzin més treball a la mateixa quantitat de temps que el multiprocés, degut a que la CPU està compartida

entre tasques d'un mateix procés. A més a més, com el multiprocés està implementat a nivell de sistema operatiu, el programador no pot intervenir en el plantejament de la seva execució; mentre que al cas de la multitasca, com el programa té que ser dissenyat expressament per a que pugui suportar aquesta característica, és imprescindible que l'autor tingui que planificar adequadament l'execució de cada tasca.

Programes de flux únic

Un programa de flux únic, tasca única (single-thread) utilitza un únic flux de control (thread) per a controlar la seva execució. Molts programes no necessiten la potència o utilitat de múltiples tasques. Sense necessitat d'especificar explícitament que es requereix un únic flux de control, molts dels applets i aplicacions són de flux únic.

Quan es crida a main() és quan tot succeeix dintre d'una única tasca (thread).

Programes de flux múltiple

Els navegadors utilitzen diferents tasques executant-se en paral·lel per a realitzar vàries tasques, "aparentment" de forma concurrent.

Mentre que els programes de flux únic poden realitzar la seva tasca executant les subtasques seqüencialment, un programa multitasca permet que cada tasca comenci i acabi tant aviat sigui possible. Aquest comportament presenta una millor resposta a l'entrada en temps real.

5.5.4 Localhost

Un nom d'equip acostuma a ser una paraula senzilla, escollit per l'administrador. És lliure de donar tant noms descriptius (administració), com creatius (jhg, catalunya), com números (pc15), com qualsevol altre.

Molts servidors porten com a nom el servei que ofereixen, per exemple *www*, *proxy* o *ftp*. Això fa més fàcil l'accés extern (una vegada configurat el DNS), ja què quan s'escriu www.domini.com, el visitant està demanant contactar amb la màquina de nom *www* dintre del domini *domini.com*.

Quasi tots els sistemes operatius inclouen un nom predeterminat per a referir-se al propi ordinador. Aquest és el localhost i es pot usar sempre, malgrat se li hagi assignat un altre nom d'equip.

El terme localhost s'utilitza per descriure l'ordinador local, la màquina a la qual s'està executant l'aplicació. Quan es connecta a una xarxa IP, l'ordinador local ha de tenir una adreça IP, que pot aconseguir de maneres diferents.

Normalment, l'adreça donada al localhost és la 127.0.0.1.

5.6 Què és un thread?

Un thread (fil, tasca, flux de control del programa) representa un procés individual executant-se en un sistema, *Figura 13 Estats d'un thread*. A vegades se'ls hi diu processos lleugers o contextos en execució.

Tots els fils d'un procés comparteixen l'estat i els recursos del procés. Resideixen al mateix espai d'adreces i tenen accés a les mateixes dades. Quan un fil modifica una dada a memòria, els altres fils utilitzen el resultat quan accedeixen a la dada.

Un exemple de la utilització de fils és tenir un fil atent a l'interafase gràfica (icones, botons, finestres, ...), mentre un altre fil fa una llarga operació internament. D'aquesta manera el programa respon més àgilment a la interacció amb l'usuari.

Avantatges dels fils contra processos

Si bé els fils són creats a partir de la creació d'un procés, podem dir que un procés és un fil d'execució, conegut com a monofil. Però les avantatges dels fils es donen quan parlem de Multifils, un procés té múltiples fils d'execució els quals realitzen diferents activitats, que poden o no ser cooperatives entre sí. Els beneficis dels fils es deriven de les implicacions de rendiment.

Es tarda molt menys temps en crear un nou fil en un procés existent que a crear un procés.

Es tarda molt menys temps en finalitzar un fil que un procés, ja que quan s'elimina un procés es té que eliminar el PCB (Process Control Block) del mateix, mentre que un fil s'elimina el seu context i la pila

Es tarda molt menys temps en canviar entre dos fils d'un mateix procés.

Els fils augmenten l'eficiència de la comunicació entre programes en execució. A la majoria dels sistemes en la comunicació entre processos ha d'intervenir el nucli per oferir protecció dels recursos i realitzar la comunicació mateixa. En canvi, entre fils poden comunicar-se entre sí sense la invocació al nucli. Per tant, si hi ha una aplicació que s'ha d'implementar com un conjunt d'unitats d'execució relacionades, és més eficient fer-ho amb una col·lecció de fils que amb una col·lecció de processos separats.

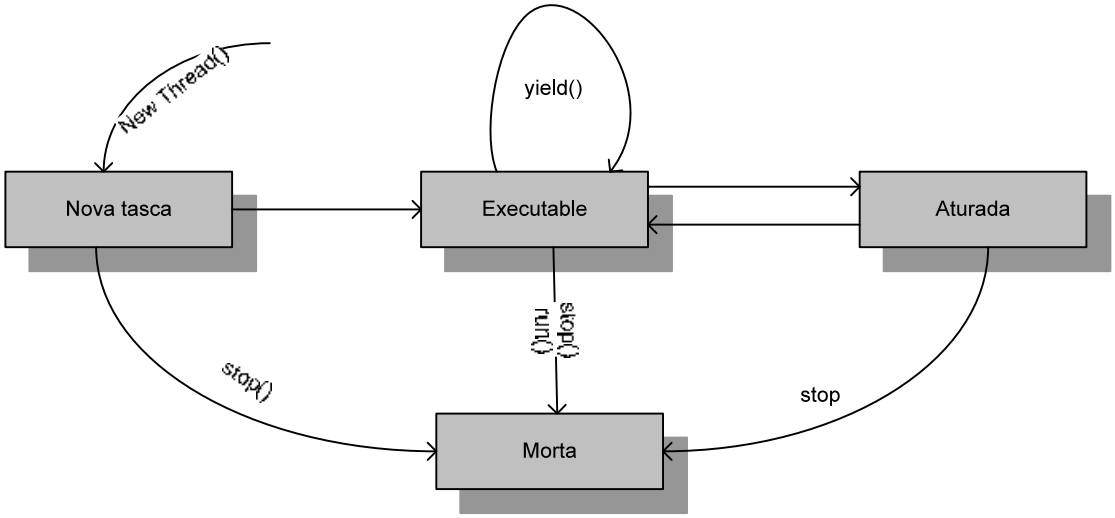

Figura 13 Estats d'un thread

6 ANÀLISI

6.1 Requisits del Servidor

A continuació es descriu quins requisits haurà de fer front el servidor per tal de poder executar el nostre proxy TFCProxy:

1. El programa està basat en l'arquitectura client/servidor i per aquest motiu es tindrà que estar a l'espera d'escoltar aquestes peticions.
2. S'haurà d'esperar les possibles connexions HTTP pel port del proxy 8080
3. Com pot haver més d'un client, es tindrà que suportar el multi-thread per poder atendre les peticions de cadascú dels clients
4. Solament es processaran pàgines que hagin donat com a resultat un codi d'acceptació OK, 200, En cas contrari, es deixarà al navegador que actuï en conseqüència, mostrant una pàgina d'error de càrrega si la pàgina no s'ha trobat o mostrant una altra pàgina d'error indicant perquè no s'ha pogut establir una connexió.
5. Per a cada petició es comprovarà si a l'adreça hi ha alguna paraula clau i si fos així no es mostrarà la pàgina que es volia veure
6. Si la capçalera no tenia cap paraula clau s'haurà de comprovar si el cos de la pàgina en conté alguna paraula clau. I farem com al cas anterior: si hi ha alguna paraula clau no es carregarà la pàgina web i es mostrarà un missatge d'error
7. Si la pàgina visitada no conté cap paraula clau es mostrarà per pantalla
8. Es resta a l'espera de possibles peticions

6.2 Funcionalitats Addicionals

En aquest apartat volem destacar quines funcionalitats hem afegit a la primera idea de l'aplicació que havíem desenvolupat tant en pseudocodi com en una primera implementació. I és que hem cregut molt oportú afegir-hi 3 punts destacables per la seva funcionalitat i millor enteniment de l'aplicació en curs.

Aquestes noves funcionalitats desenvolupades són les següents:

1. Creació d'un fitxer anomenat **FiltreWeb.txt** que contindrà totes les paraules clau que no es volen mostrar per pantalla. En un principi, es podria haver ficat al programa un array de totes les paraules clau, però s'ha cregut més convenient, més eficaç i més pràctic crear un fitxer, el qual es podrà modificar quan es vulgui.

2. Log. Es crea un fitxer per tenir enregistrat els accessos a pàgines web restringides. En aquest fitxer s'emmagatzemarà la pàgina a la qual es volia accedir, la paraula clau que ha fet l'incidència, el dia i l'hora.

3. Pàgina d'error. Quan s'intenta accedir a una pàgina restringida, per pantalla es carregarà una pàgina en format HTML, anomenada **LogTFCProxy.txt**, prèviament dissenyada, que indicarà que s'està accedint a una pàgina web prohibida o restringida.

   Aquesta pàgina en format HTML es podrà modificar dinàmicament mentre es vagin trobant paraules clau a la pàgina que s'intenta accedir, *Figura 9 Contingut del fitxer LogTFCProxy.txt*

6.3 Funcionament general de programes de filtre de pàgines web

Revisant diferents programes, tant comercials com gratuïts, sobre el tema de filtre de pàgines web trobem que n'hi han molts aspectes en comú però d'altres que difereixen notablement, com pot ser el tema de la parametrització del tipus de missatges o adreces a filtrar.

El funcionament sembla el mateix: un programa que resta en segon pla i que intercepta els missatges que envia el navegador a Internet o a l'inrevés. Llavors, depenent de l'adreça que es vol accedir o de la paraula que es vol buscar el programa tanca el navegador o bé mostra una pàgina amb un error.

Penso que el millor és tenir un apartat a on es pugui entrar les diferents adreces que es volen filtrar, així com el tipus de paraules que es volen evitar al realitzar una cerca via Internet.

Les adreces a filtrar poden ser amb el nom sencer de la pàgina web o bé ficant-hi caràcters comodins per indicar que es vol filtrar totes les pàgines d'una adreça en concret; exemple: www.guerra.com o *guerra*, llavors filtraríem solament l'adreça www.guerra.com o bé totes les

adreces que continguin la paraula guerra, com podria ser www.guerra.net, www.juanguerra.com, etc.

Això es podria fer generant un fitxer o una base de dades que contingui totes les adreces a filtrar, així com les paraules clau.

Una part optativa és el control log o registre d'activitats dels intents incorrectes o violacions de les regles introduïdes al programa.

- Definició de les adreces
- URL a les quals es pot restringir el tràfic o a quines es poden deixar lliure accés
- Sessions d'usuari
- Registres d'activitat
- Intervenir en xats IRC
- Resum de cada usuari

7 IMPLEMENTACIÓ

7.1 Desenvolupament del codi

La crida al projecte es fa mitjançant un paràmetre i aquest és el port què s'ha d'escoltar. Normalment, el port serà el 8080. I si no es fica quin port a escoltar com a paràmetre, per defecte s'agafarà el 8080.

Una vegada que comença a executar-se el programa, es genera un nou fil d'execució de TFCProxy passant-li com a paràmetres el port i el timeout. I com es pot navegar a través de més d'una finestra és per això que es crea un nou fil per cada execució.

Quan fiquem TFCProxy.start() és quan comença a executar-se el mètode run(). Llavors, aquí es crea un nou socket del tipus ServerSocket amb el port d'escolta que s'ha seleccionat. I a un bucle indefinit és on s'acceptaran les connexions del socket a un altre que anomenem client. Creem el nou fil amb aquest socket i processem les dades.

Seguidament es comença a llegir les dades del socket client amb el mètode getInputStream i ho fem amb un mètode amb búffer, concretament amb BufferedInputStream.

I per la sortida de dades (les dades que enviarà el nostre programa) utilitzarà el mètode getOutputStream del socket client i també serà amb un búffer, concretament BufferedOutputStream.

Com que tenim les paraules clau a un fitxer, el que hem de fer és llegir aquest fitxer anomenat **FiltreWeb.txt** i ficar les paraules clau a un array del tipus String.

Ara es llegirà de la pàgina web que s'intenta accedir quina és la seva petició. Per fer-ho es crida a getHTTPData passant-li com a paràmetres:

- clientIn: Les dades emmagatzemades al BufferedInputStream
- host: La passem per valor perquè ens doni a quin host s'intenta accedir
- false: Si s'espera per desconnexió

De la cadena que hem rebut comprovem si el protocol és l'http i si és així, obviarem el protocol i ens quedarem amb el host que es vol accedir. Per exemple: si la cadena de l'adreça que volem accedir és http://www.uoc.edu, ens quedarem amb la cadena www.uoc.edu

Si l'adreça què es vol accedir no té cap paraula clau es continuarà processant la pàgina web. Per fer-ho, comprovarem si hi ha dades i si el protocol és l'http. Si és així, crearem el fitxer temporal **Da-**

desWeb.txt a on es gravaran totes les dades de la pàgina web. I com ho fem? Doncs, creem un nou URL amb l'adreça de la pàgina (URL url = new URL(adreca)). Del nou url llegim les dades que entren a través del mètode openStrem() i el fiquem a paginaHtml que és del tipus BufferedReader. Llavors, llegim totes les línies fins que no sigui un nul i les gravem al fitxer temporal.

Una vegada tenim les dades de la pàgina web al fitxer temporal, llegim les dades d'aquest fitxer i comprovem que no tingui cap paraula clau.

Si la pàgina no conté cap paraula clau, llavors es processarà la càrrega de la pàgina web per pantalla i sempre forçarem aquesta escriptura amb el mètode out.flush().

Si hi ha dades, llavors fiquem esperaADesconnectart a false per evitar una desconnexió i escriure dades a la pàgina que es vol llegir..

7.2 Productes obtinguts

Primerament hem fet un programa que fos transparent al navegador i mostrés el nom de les pàgines per les quals navegàvem, així com els missatges obtinguts per cada una d'aquestes (missatges d'acceptació de la connexió, error d'adreça inexistent, etc).

Una vegada que va funcionar aquest producte, es va implementar per tal de trobar una paraula en concret dintre de la pàgina web. Aquí es va tenir bastants problemes perquè si carregàvem una pàgina que tingués la paraula clau, moltes vegades no la detectava el programa. Ni tampoc detectava si la paraula clau era una part del text a buscar en un cercador (ja sigui Google, Yahoo, etc.).

Llavors, es va optar per descarregar la pàgina a llegir a un fitxer per, seguidament, llegir-lo i comprovar si la paraula clau estava en aquest fitxer. I això va funcionar, ja què el creem i l'esborrem per cada petició que es fa a la xarxa. Així, si es fa un cerca d'una paraula clau es crea el fitxer amb la capçalera del missatge que envia el cercador, obrim el fitxer, el llegim i si la paraula clau hi és avortem el procés de càrrega de la pàgina web.

Pel contrari, si la paraula clau no està a la pàgina, llavors el programa actua de forma transparent i carrega la pàgina web com si res hagués passat.

Per últim, es va crear un fitxer, FiltreProxy.txt per ficar les paraules clau a filtrar. Aquest fitxer es pot modificar amb un processador de text qualsevol i afegir, esborrar o modificar paraules clau.

I, abans de llegir el fitxer creat amb la pàgina web descarregada, llegix el fitxer que té les paraules a filtrar i les fiquem a un array. Llavors, comprovem si al fitxer que conté la pàgina web descarregada

conté alguna paraula clau que hi ha a l'array de paraules clau. I si és que sí ja no deixarem continuar per descarregar la pàgina web i veure-la al navegador.

Hem optat per llegir el fitxer que conté les paraules clau cada vegada que processem perquè ens pot interessar modificar aquest fitxer sobre la marxa per afegir noves paraules, esborrar-les o modificar-les. Llavors, no farà falta reiniciar el navegador.

El funcionament del TFCProxy és el mostrat a la *Figura 14 Funcionament de TFCproxy*, on es mostra els passos que realitza, així com la creació

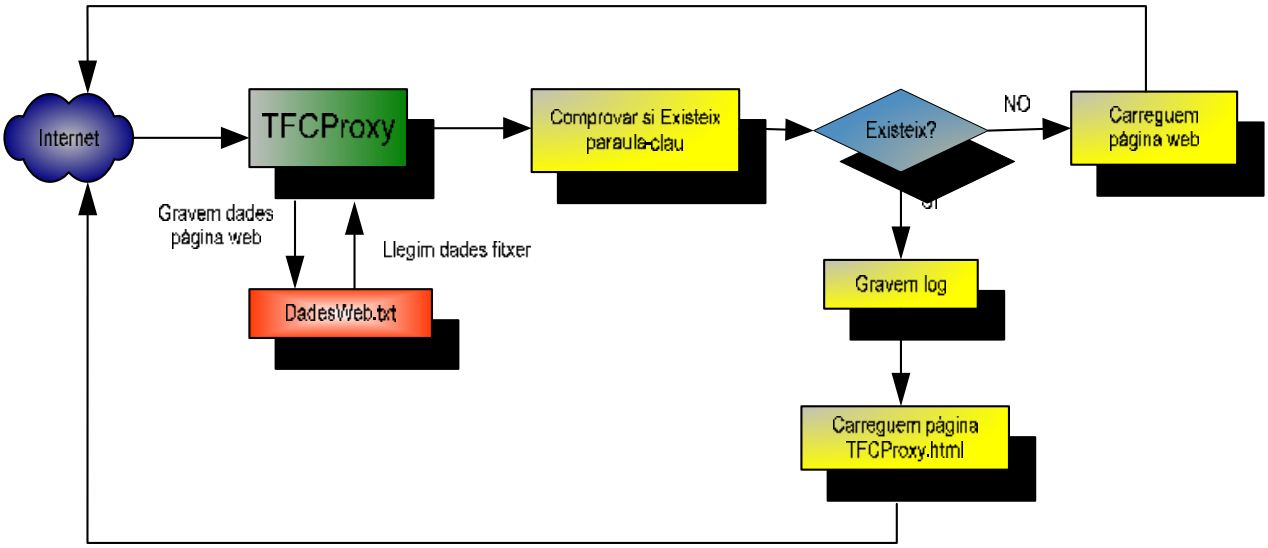

Figura 14 Funcionament de TFCproxy

7.3 Futures millores

Aquesta primera versió de l'aplicació TFCProxy té una funcionalitat específica i molt clara que és la de poder filtrar pàgines web. Si més no, s'han afegit millores per tal de fer-lo més funcional i fàcil d'utilitzar, com pot ser la creació d'un fitxer en format txt per afegir les paraules que no volem mostrar i evitar l'aparició de la pàgina web. Però, aquest fitxer s'ha de modificar a través d'un programa extern com pot ser un editor de text.

Possibles millores que creiem que es podrien fer:

- Creació d'una pantalla per mostrar les paraules clau que es volen utilitzar. Amb això aconseguirem que no farà falta cercar on està el fitxer que conté les dades.

- Creació d'una base de dades per gravar a una taula les paraules clau i a una altra les dades que conté el fitxer log; és a dir, un registre d'activitats
- Afegir seguretat per realitzar canvis a les taules creades anteriorment, mitjançant un nom d'usuari i contrasenya
- Afegir a la taula de log l'usuari que treballa a l'ordinador
- Presentació per pantalla de l'històric dels logs entre dates, usuaris, paraules, etc.

8 MANUAL DE FUNCIONAMENT

8.1 Requisits de connexió

Per a poder executar el programa, prèviament s'ha de complir un requisit molt important i és que s'ha de ficar al navegador que es vol utilitzar un proxy local i al port d'escolta 8080. Per a fer-ho s'haurà d'anar a connexions i marcar la casella de proxy i ficar els paràmetres, *Figura 16* Selecció del proxy a l'Explorer:

- adreça: localhost o 127.0.0.1
- port: 8080

Si no fem es pas anterior no tindrem el programa escoltant les entrades/sortides i es podrà navegar per allà on es vulgui sense restriccions.

8.2 Instal·lació

La instal·lació del nostre programa es pot realitzar de dos maneres diferents, amb requisits totalment diferents: una amb el programa compilat en Java i una altra seria el programa compilat en format EXE per alguna aplicació existent al mercat, però és de pagament.

El més fàcil és descomprimir el fitxer TFCProxy.zip a l'arrel del disc **C:** on es crearà la carpeta TFCProxy, i dintre d'aquesta es crearan cuatre més:

- Documentació: Carpeta on es trobarà la documentació del programa en format DOC i en format PDF
- Exe: Carpeta on es troba el programa compilar amb un versió **demo.**
- Font: Carpeta on es troba el programa font en java
- Java: Carpeta on es troba el programa per executar en java

8.2.1 Fitxer en format java

Per a realitzar la instal·lació de la nostra aplicació es fa d'una manera molt fàcil i és copiar els següents arxius a un directori qualsevol:

- TFCProxy.class
- FiltreWeb.txt

- TFCProxy.html

A més a més, tenim que tenir l'aplicació JAVA per poder llençar el programa, i es faria de la seguent manera:

java TFCProxy núm_port,

on núm_port seria el número del port que desitgem escoltar. Si no es fica el número de port, per defecte s'assumirà que és el 8080.

Un apunt a tenir en compta és que al fitxer FiltreWeb.txt s'hauria d'introduir les paraules que volem filtrar.

8.2.2 Fitxer en format EXE

Però, també es podria ficar el programa compilat en format **EXE** mitjançant alguna que altra aplicació que existeix al mercat, com poden ser JEXECreator [6], JToExe [7], J2Exe [8], i el que hem utilitzat nosaltres exe4j[9] (una versió d'avaluació), etc. En aquest cas, solament s'hauria de tenir els fitxers:

- TFCProxy.exe
- FiltreWeb.txt
- TFCProxy.html
- TFCProxy.class
- TFCProxyThread.class

En aquest altre cas, no faria falta tenir la màquina virtual de java, perquè ja tindríem un fitxer executable.

[6] JEXECreator: http://www.ucware.com

[7] JToExe: http://www.bravozulu.com/

[8] J2Exe: http://j2exe.tripod.com/cgi-bin/index.pl

[9] exe4j: http://www.ej-technologies.com

8.3 Configuració

I com podem fer que la nostra aplicació sigui un proxy i estigui en un segon pla? Doncs, el que farem serà modificar la configuració de la connexió i ficar que s'ha d'utilitzar un proxy que tindrà com a adreça el localhost o 127.0.0.1 i com a port d'escolta el 8080, *Figura 16 Selecció del proxy a l'Explorer*.

8.3.1 Explorer

La configuració seguirà els següents passos:

A la barra de menú de l'Explorer seleccionarem l'opció *Herramientas* i d'aquí *Opciones de Internet* on ens apareixerà una pantalla amb pestanyes, *Figura 15 Opcions d'Internet a l'Explorer*. Hem de seleccionar l'apartat *Conexiones*, , i d'aquí premem el botó que fica *Configuración de LAN* per poder entrar-hi els paràmetres del proxy.

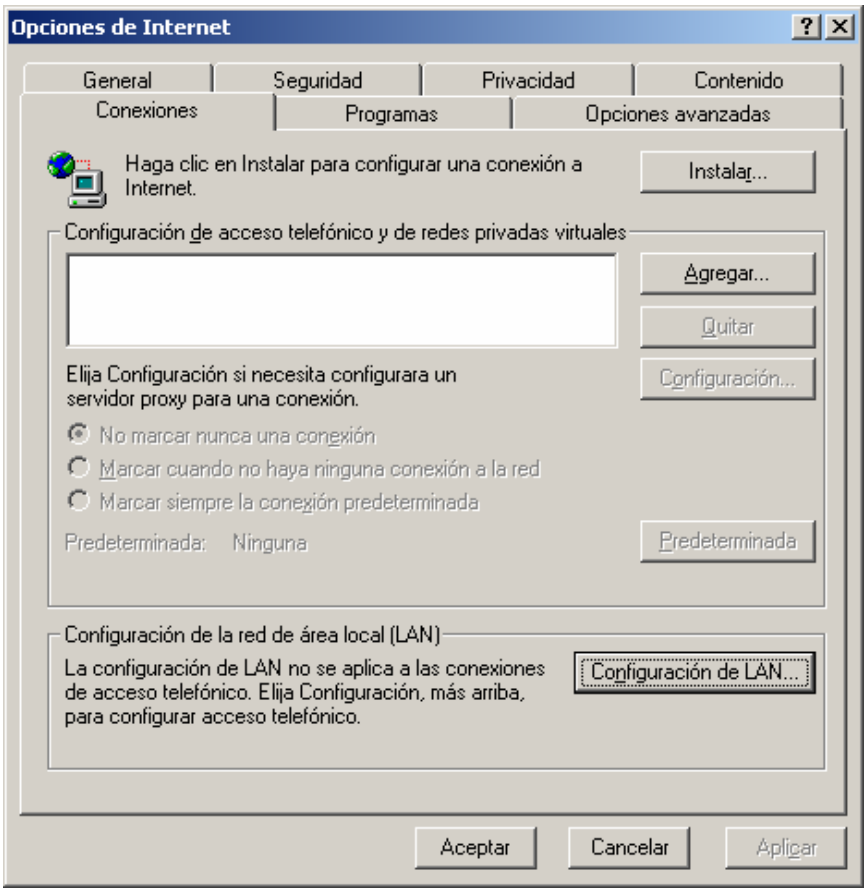

Figura 15 Opcions d'Internet a l'Explorer

Llavors, activem el botó *Servidor proxy* i omplim les dades de quina adreça ficarem com a proxy i per quin port, com a la *Figura 16 Selecció del proxy a l'Explorer.* Per confirmar els canvis premem el botó Aceptar i ja tenim configurat el navegador per poder utilitzar el proxy TFCProxy.

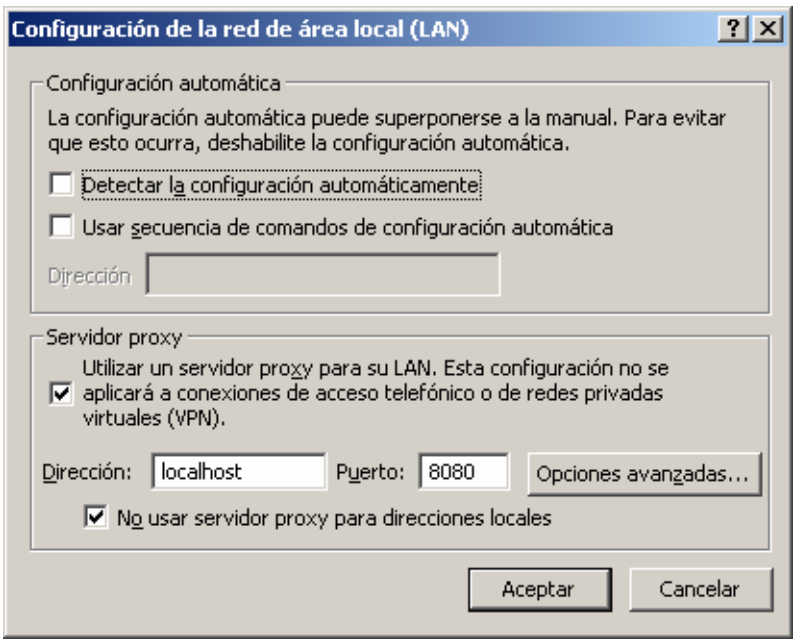

Figura 16 Selecció del proxy a l'Explorer

8.3.2 Firefox

Si, pel contrari, utilitzem el navegador Firefox, els passos a seguir seran els següents:

Seleccionarem de la barra de menús l'opció *Eines* i d'aquí seleccinem *Opcions*, on apareixerà una finestra amb pestanyes. Haurem de seleccionar la que fica *General* i premem el botó de *Paràmetres de connexió*, *Figura 17 Opcions d'Internet al Firefox.*

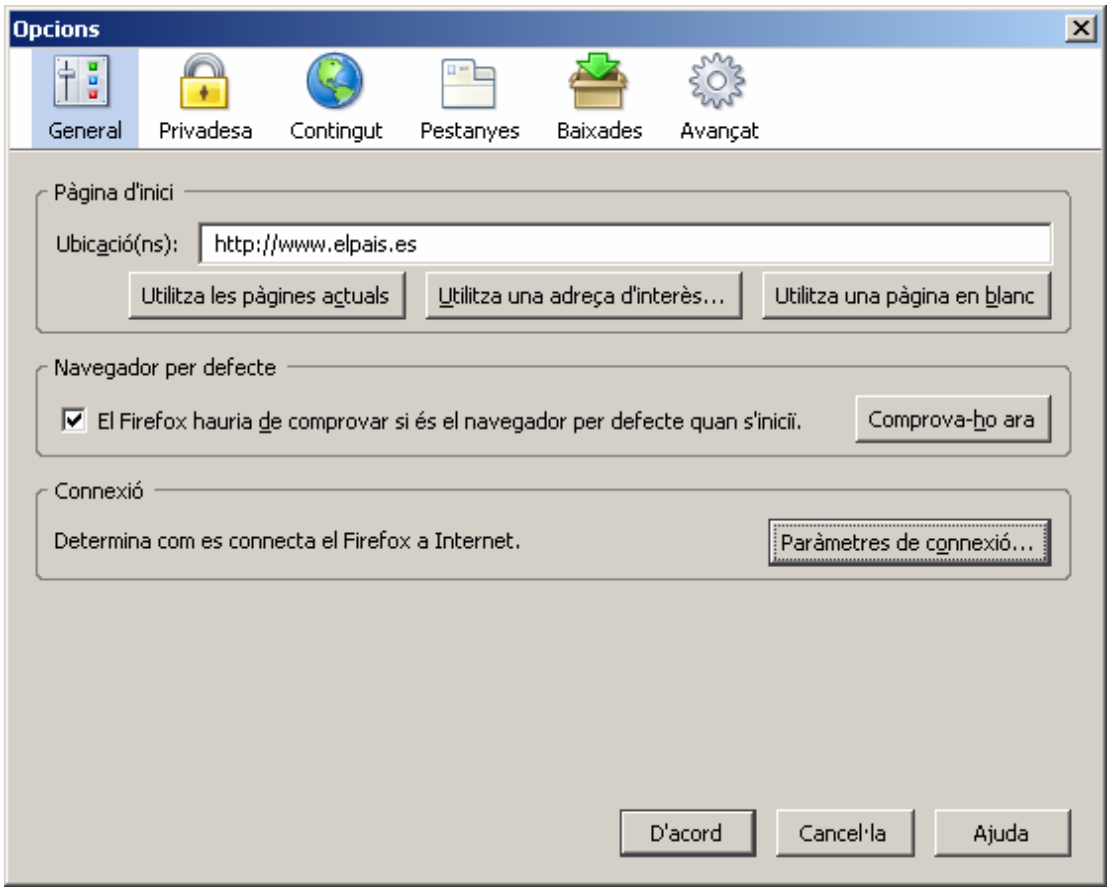

Figura 17 Opcions d'Internet al Firefox

Apareixerà una nova finestra a la qual seleccionarem l'opció *Configuració manual del servidor intermediari*, *Figura 18 Selecció del proxy al Firefox* a on omplirem les següents dades:

- Servidor intermediari d'HTTP: 127.0.0.1
- Port: 8080

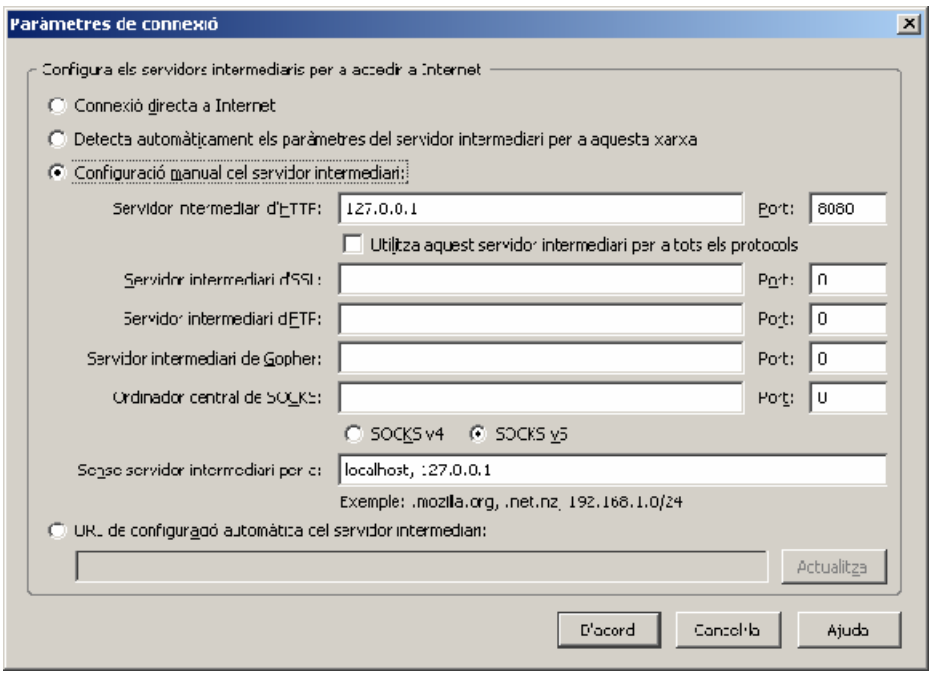

Figura 18 Selecció del proxy al Firefox

8.4 Joc de proves

Per a la realització de les proves s'ha utilitzat dues màquines amb les següents característiques:

- **PC1**
  - PC de sobretaula Pentium IV
  - 512 Mb de memòria RAM
  - S.O.: Windows 2000 Service Pack 4
- **PC2**
  - Portàtil Pentium IV
  - 1 Gb de memòria RAM
  - S.O.: Windows XP Service Pack 2
- **Xarxa**
  - Ethernet 10/100
  - Wireless

8.4.1 Proves realitzades

Les proves realitzades han estat, bàsicament, dos:

- La primera ha sigut carregar una pàgina web sense paraules clau
- La segona, carregar una pàgina web amb una o més paraules clau

Quan hem carregat una pàgina web que, prèviament, hem comprovat que no té cap paraula clau hem obtingut la pàgina carregada amb èxit.

Després, hem carregat una pàgina web que sabíem que contenia, com a mínim, una paraula clau, *Figura 21 Pàgina visualitzada amb el navegador i paraula clau marcada*. Llavors, ha ocorregut dos coses:

- una visible per l'usuari que ha estat que la pàgina s'ha bloquejat i no ha deixat veure la pàgina web a l'usuari, *Figura 19 Pàgina bloquejada pel TFCProxy*

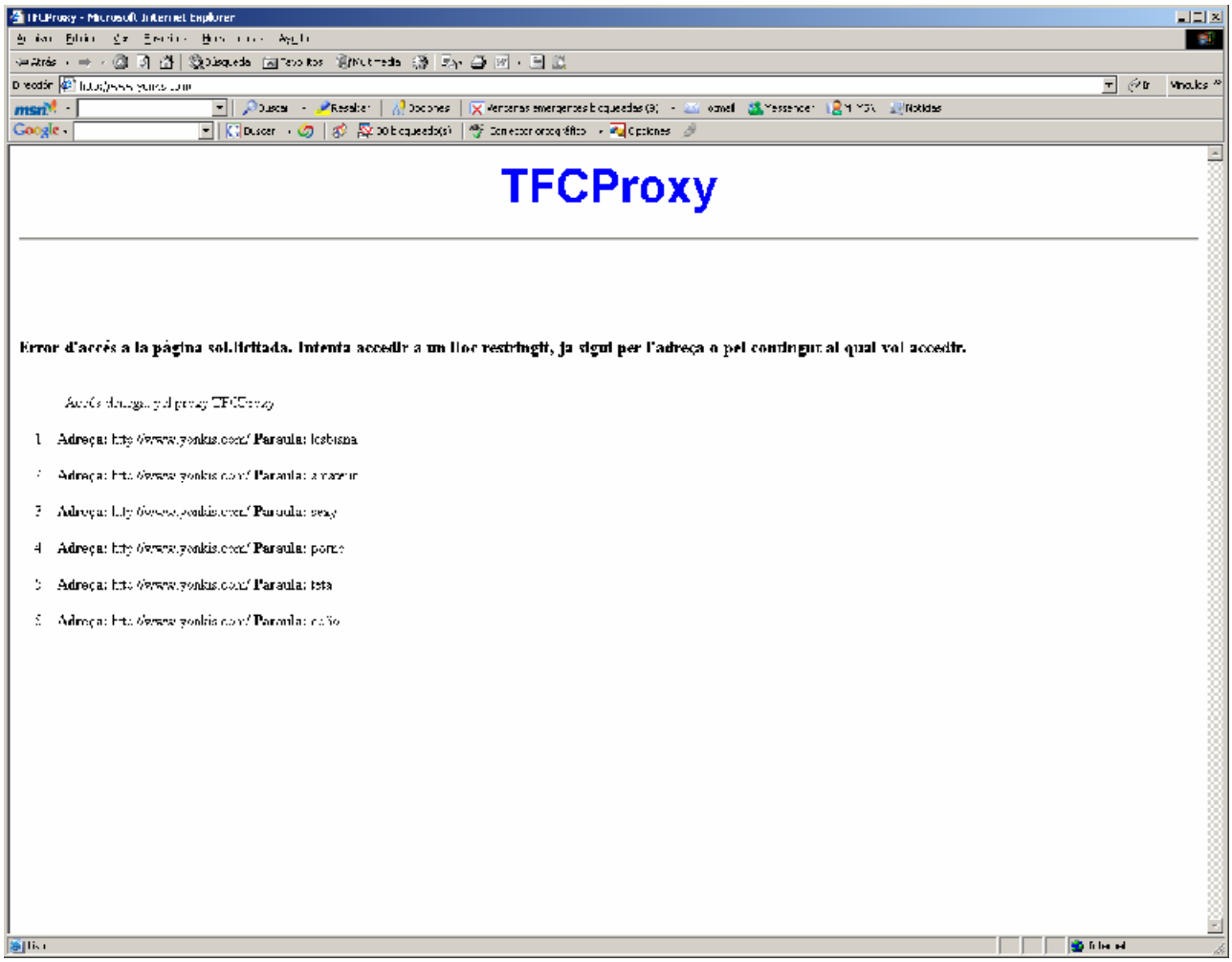

Figura 19 Pàgina bloquejada pel TFCProxy

- una altra que no és visible per l'usuari i que ha estat que aquest intent ha estat enregistrat al fitxer log, *Figura 20 Lectura del fitxer log*

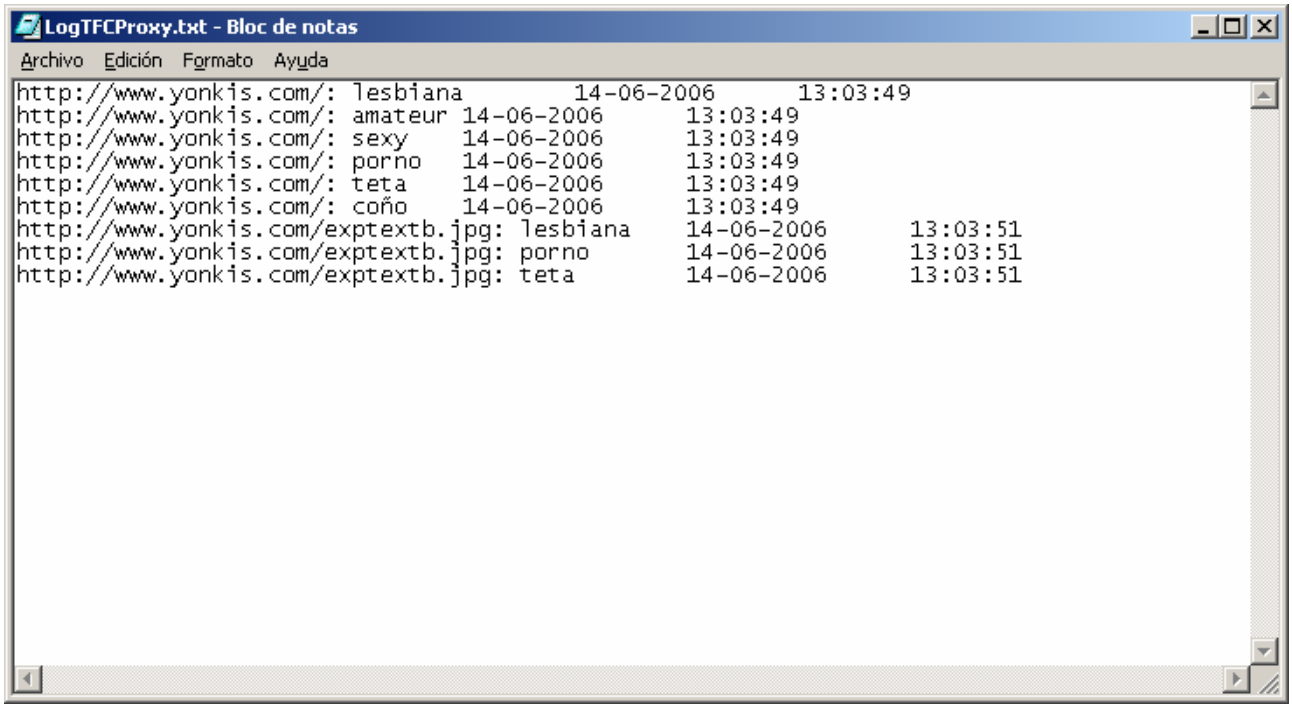

Figura 20 Lectura del fitxer log

Hem fet moltes proves i ens ha donat un bon percentatge d'èxit en la detecció d'aquestes pàgines que no volíem mostrar.

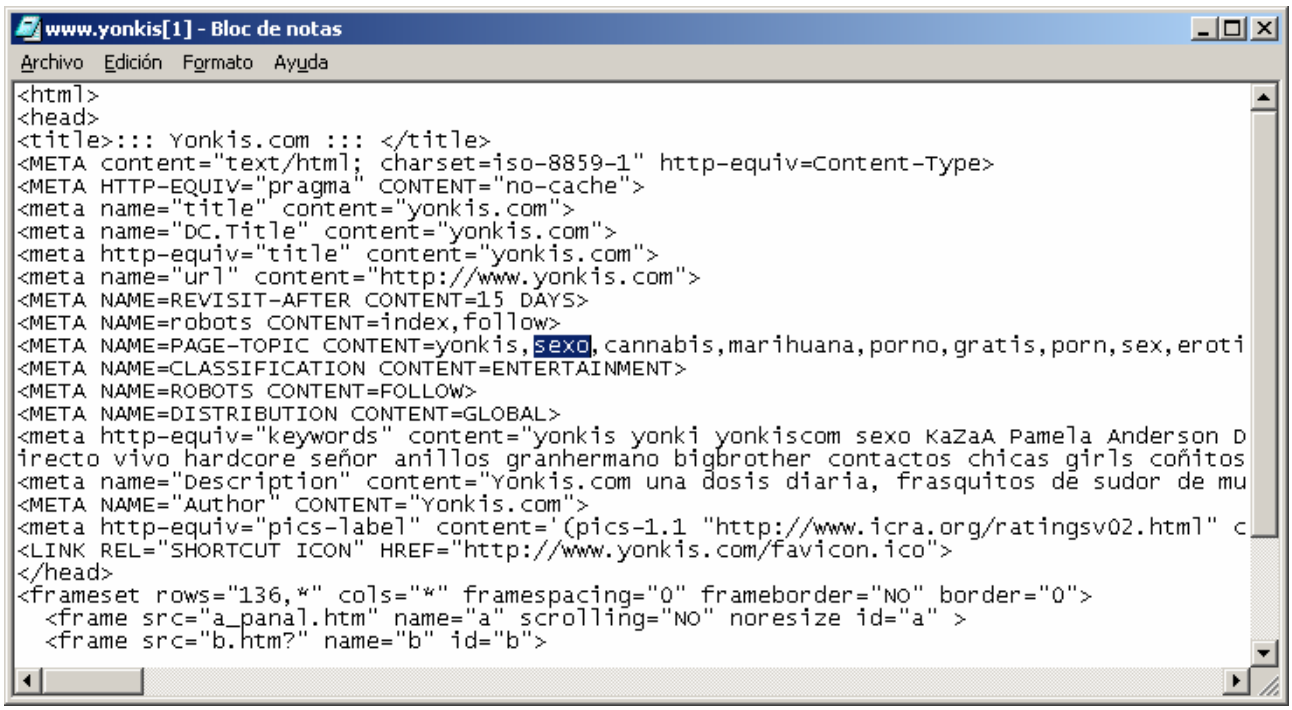

Figura 21 Pàgina visualitzada amb el navegador i paraula clau marcada

9 VALORACIONS ECONÒMIQUES

Un programa d'aquestes característiques pot repercutir positivament a l'economia familiar o d'una empresa perquè pot estalviar mal de caps a l'hora de navegar per llocs prohibits. Aquest estalvi serà en temps i econòmic.

A l'àmbit familiar perquè podem limitar a un fill que no pugui entrar per pàgines obscenes, malparlades, sexistes, bèl·liques, racistes, etc.

A l'àmbit empresarial perquè podem filtrar pàgines als empleats per tal que no puguin accedir a pàgines que no són de la seva tasca. Podem evitar que un empleat llegeixi pàgines de sexe, diaris, etc. D'aquesta manera, un empleat dedicarà la seva jornada a treballar i no pas a deambular pàgina rera pàgina en coses externes a la feina. Això pot representar un estalvi econòmic molt important per una empresa en concepte de feina inexistent; els treballadors no perdran el temps cercant per Internet coses que no ha de fer.

Suposant que un empleat pugui navegar una mitja d'una hora al dia, això representa unes cinc hores a la setmana, unes 20 hores al més o molt més. I si hi ha més empleats, la despesa per hores no treballades es dispararia considerablement. És aquí on un programa de filtre de pàgines web pot repercutir molt positivament a l'economia d'estalvi d'una empresa.

10 CONCLUSIONS

Una de les coses més interessants de l'aplicació ha sigut l'utilització dels fils d'execució, ja que fins aquest moment encara no havia fet res semblant. Sorprèn comprovar que els recursos emprats quan hi ha un client o n'hi ha més són mínims. Per aquest motiu s'ha seleccionat l'utilització dels fils.

Hem pogut fer proves amb Firefox i Explorer i hem obtingut molt bons resultats, filtrant les pàgines que hem decidit que no contindrien certes paraules. I no tenien perquè ser paraules obscenes sinó qualsevol paraula, ja sigui de cuina, de motor, etc.

Per comprovar-ho s'ha fet manualment: una vegada teníem carregada la pàgina hem editat la pàgina web carregada i hem buscat si existia alguna paraula clau.

Crec, sincerament, que el producte que hem realitzat ha estat superior al que en un primer moment havíem pensat. Teníem unes expectatives que s'han anant complint i, sorprenentment, s'han anant millorant i afegint. No sense haver fet un gran esforç amb el temps que hem pogut dedicar.

Realitzar aquest TFC m'ha aportat molt més coneixement del que tenia sobre temes com els protocols (portar-ho a la pràctica), com capturar les adreces d'una pàgina web, així com el text d'aquestes, com poder carregar una pàgina en format HTML que hem creat i que la podem modificar dinàmicament mentre afegim informació, etc.

Fent un repàs al començament d'aquest treball em vaig veure una mica fosc perquè no sabia per on començar. Ni que devia fer! Però, una vegada havia demanat ajut a la Consultora ja vaig començar a veure una mica més clar què es tenia que fer. Primer de tot, recopilar informació i després buscar la forma de com s'hauria d'aplicar a l'aplicació que es volia implementar.

I poc a poc hem anat obtenint resultats: unes vegades positius i d'altres negatius, però aquestos últims s'han anat solucionant. I no han estat pocs els problemes que hem hagut de resoldre perquè, de vegades, i degut a alguna errada "tonta" hem perdut molt de temps per esbrinar què passava.

En general, ha estat una molt bona experiència aquest TFC perquè hem après moltes coses noves en quant a conceptes i en quant a programació en Java.

11 GLOSSARI

- **Caché**: La memòria *caché* és una eina que permet estalviar temps a l'usuari. Té l'objectiu de memoritzar al disc dur de l'internauta els fitxers que ha consultat anteriorment, ja siguin de text, imatges o d'àudio. D'aquesta manera, quan l'usuari entra a una pàgina web que ha visitat anteriorment, no cal que s'estableixi una connexió amb la xarxa amb la conseqüent pèrdua de temps, ja que obtindrà la informació des del seu propi disc dur.

- **HTTP**: (HyperText Transfer Protocol) és el protocol emprat en cada transacció de la web (WWW).

- **LAN**: Local Area NetWork. Xarxa d'Àrea Local

- **Localhost**: Nom que té la màquina local, la qual està executant el programa, i que normalment té l'adreça 127.0.0.1

- **Multifil (Multithread)**: Vegi's Thread

- **Paraula-clau**: Definim aquest terme com la paraula o adreça que volem filtrar; llavors, si la pàgina web que volem accedir conté aquesta paraula es denegarà el servei i no es mostrarà per pantalla.

- **Port**: un port és una forma genèrica de denominar a una interfície per la que diferents tipus de dades poden ser enviades i rebudes. Aquesta interfície pot ser física o a nivell de programari (per exemple, els ports que permeten la transmissió de dades entre diferents ordinadors).

- **Socket**: Punt de comunicació bidireccional entre dos ordinadors

- **Tallafocs**: És simplement un filtre que controla totes les comunicacions que passen d'una xarxa a l'altra i en funció del que sigui permet o denega el seu pas

- **TCP/IP**: Protocol de comunicacions (Transfer Control Protocol/Internet Protocol)

- **URL**:(Uniform Resource Locator) Es pot dir que és l'extensió del concepte de nom complet d'un arxiu (path). Mitjançant un URL no solament pot apuntar-se a un arxiu en un directori

en un disc local, sinó que a més a més tal arxiu i tal directori poden estar localitzats en qualsevol ordinador de la xarxa, amb el mateix o diferent sistema operatiu. Les URLs possibiliten el direccionament de persones, fitxers i d'una gran varietat d'informació, disponible segons els diferents protocols o serveis d'Internet.

- **URI**: Qualsevol element present a la Web té un Identificador de Recurs Uniforme (URI). Els URI poden fer referència a documents, recursos, serveis, persones i indirectament a qualsevol cosa. Les URI són les tecnologies de direccionament que mantenen a la Web unida.

- **Thread**: (fil, tasca, flux de control del programa) representa un procés individual executant-se en un sistema

12 BIBLIOGRAFIA

- **http://www.tejedoresdelweb.com/307/article-1046.html**

Tejedores del Web. Tecnologías del Servidor

- **http://cdec.unican.es/libro/HTTP.htm**

El protocolo HTTP

- **http://www.cibernetia.com/**

Guía de cabeceras del protocolo HTTP

- **http://www.htmlweb.net/manual/tipos_mime.html**

Tipo MIME, por Luciano Moreno, del departamento de diseño web de BJS Software

- **http://es.wikipedia.org/wiki/Proxy**

Creación proxy y su funcionamiento

- **http://www.programacion.com/tutorial/proxy/1/**

Servidores Proxy

- **http://www.wikilearning.com/urls_localizadores_uniformes_de_recursos-wkccp-6102-4.htm**

URL's: Localizadors Uniformes de Recursos

- **http://www.xtec.es/~mcguri/d70/d70m4/desca.htm**

Descripció de l'aplicació "Apache". El protocol HTTP

- **http://www.hipertexto.info/documentos/localiz.htm**

Normalización, localización e identificación.

- **http://www.programacion.com/java/tutorial/joa_red/6/**

Programación en red. Comunicaciones mediante el protocolo UDP.

- **http://saloon.javaranch.com/cgi-bin/ubb/ultimatebb.cgi?ubb=get_topic&f=8&t=000177**

Post sobre ProxyServer

- **http://www.unav.es/cti/manuales/Java/indice.html**

Java. Centro de Tecnología Informática. Universidad de Navarra

- **Java 2. Manual de usuario y tutorial 3ª edición**

Agustín Froufe. Editorial Ra-Ma.