

Treball fi de carrera

**ENGINYERIA TÈCNICA EN
INFORMÀTICA DE SISTEMES**

**Facultat de Matemàtiques
Universitat de Barcelona**

EDICIÓ DE FITXERS VIA WEB

Antoni Cardell Crespí

Director: Mario Gorga i Sergio Escalera

Realitzat a: Departament de Matemàtica

Aplicada i

Anàlisi. UB

Barcelona, 12 de setembre de 2011

Resum

L'objectiu del projecte és crear una interfície web per poder navegar entre els fitxers de la web, poder editar-los online i a la vegada poder guardar l'històric de modificacions. Així, si en un moment donat hi ha un error en la web, poder seleccionar una versió anterior que sabem que és correcte. La idea va sorgir inicialment arrel d'un problema real de la web de la facultat de matemàtiques. La gent encarregada d'afegir continguts a la web, ho havien de fer accedint directament al codi. Això, per la gent que té pocs coneixements de programació pot ser una mica complicat. Per altra banda, se'ns produïa el problema de quan dues persones intentaven editar el mateix fitxer, només es quedava guardada la versió de l'última persona que guardava el fitxer, fent que es perdés la feina de la primera persona. Tot això provocava una pèrdua d'informació i era una font de problemes entre la gent que afegia continguts a la pàgina web de la facultat. En concret el que s'ha fet és adaptar a un navegador de fitxers web existent, un software de versions i un editor web, de tal manera que tinguem sempre guardat un històric dels fitxers que anem editant online.

Resumen

El objetivo del proyecto es crear una interfaz web para poder navegar entre ficheros de la web, poder editar-los online y a la vez poder guardar el histórico de modificaciones. De esta manera, si en un momento dado hay un error en la web, poder seleccionar una versión anterior que sabemos que es correcta. La idea surgió inicialmente a partir de un problema real de la web de la facultad de matemáticas. La gente encargada de añadir contenidos a la web, tenían que hacerlo accediendo directamente al código. Esta situación, provocaba que la gente que tenía pocos conocimientos de programación puede ser un poco complicado. Por otra parte, se producía el problema que cuando dos personas intentaban editar el mismo fichero, sólo quedaba guardada la versión de la última persona que guardaba el fichero, propiciando que se perdiese el trabajo de la primera persona. Todo esto provocaba una pérdida de información y era una fuente de conflictos entre la gente que añadía contenidos en la web de la facultad. En concreto lo que se ha echo es adaptar a un navegador de ficheros web existente, un software de versiones y un editor web, de tal manera que tengamos siempre guardado un histórico de los ficheros que vayamos editando online.

Abstract

The aim of the project is to create a web interface to navigate between files on the web, editing the files online and being able to save the historic changes. Thus, if at any time there is an error on the web, you can select a previous version you know is correct. The idea initially came from a real problem with the web of the faculty of mathematics. The people involved in adding content to the web, had to edit the files by modifying directly the code. This situation caused the people who had little knowledge of programming can be tricky. Moreover, the problem occurred when two or more people tried to edit the same file, only was saved the version of the last person who saved the file causing the loss of the first party work. All this caused a loss of information and a source of conflict between the people who added content on the web of the faculty. Specifically what has been done in the project is to adapt an existing web file browser, software versions and a web editor, so that we always kept a log of files we have been editing.

Índex de continguts

1	Introducció.....	6
1.1	Motivació.....	7
1.2	Estat de l'art.....	7
	Navegadors de fitxers via web.....	7
	Editors WYSIWYG per integrar amb navegador de fitxers.....	10
	Software de versions.....	11
	Tecnologies escollides.....	15
1.3	Objectius.....	15
2	Avaluació tecnològica.....	16
2.1	Llenguatges de programació.....	16
2.2	Llibreries.....	18
2.3	Anàlisi i especificació.....	18
	Catàleg de requeriments.....	19
	Casos d'us.....	20
3	Disseny i implementació.....	29
3.1	Diagrames de seqüència.....	29
3.2	Implementació de les funcionalitats afegides.....	31
	Adaptar software de versions SVN.....	31
	Crear arxius del tipus: txt, php, html i js.....	31
	Modificar arxius.....	32
	Guardar borradors.....	34
	Guardat automàtic del document en edició.....	35
	Login de distints usuaris.....	38
3.3	Eines utilitzades per al desenvolupament del projecte.....	40
	Navegadors Web	40
	Entorns de Desenvolupament i editors de codi	41
	Altres	41
4	Planificacions i costos.....	43
4.1	Estudi econòmic.....	44
5	Conclusions.....	46
6	Futures Millores.....	46
7	Bibliografia.....	47
7.1	Programari.....	47
7.2	Manual d'instal·lació.....	48
7.3	Manual d'usuari.....	52
7.4	Contingut del CD.....	58

1 Introducció

Internet ha sofert una evolució espectacular a partir de l'inici de la seva popularització ara fa uns quinze anys. Les primeres pàgines web d'Internet eren de contingut estàtic, és a dir, simples planes amb un text, una imatge i un enllaç que portava a una pàgina amb un altre contingut. L'edició d'aquestes planes s'havia de fer directament amb un editor de text, accedint al codi font. Això provocava que qui edités el fitxer hagués de tenir un mínim de coneixements tècnics de programació web. Els editors WYSIWYG¹ ens permeten editar pàgines web com si estiguéssim editant un fitxer de text.

El fet d'afegir o editar un contingut s'hagi de fer editant directament el fitxer sense cap tipus de control, provoca que quan dues o més persones intenten editar el mateix fitxer alhora, s'esborrin els continguts uns amb els altres. Actualment s'han aconseguit evitar aquests problemes, gràcies entre d'altres, a l'aparició dels CSM².

A la facultat de matemàtiques, hi ha una d'aquestes web estàtiques que va ser creada fa molt de temps i la seva adaptació a un CSM seria molt costosa en temps. El manteniment de la web és portada a terme per gent sense coneixements de programació i els hi resulta complicat la modificació dels fitxers html o php. Per això, s'ha pensat una solució més ràpida i que ens pugui servir de projecte final de carrera.

L'objectiu d'aquest projecte és adaptar algun navegador de fitxers web, fet en PHP i/o Ajax i adaptar-lo a les nostres necessitats. Incorporar si no el té ja, un editor WYSIWYG per a la fàcil edició de fitxers html i php i un sistema de control de versions per que en cas d'haver d'haver-hi conflicte alhora d'introduir continguts, no hi hagi pèrdues. Tot això ens hauria de permetre editar els fitxers html de la pàgina web de la facultat de manera fàcil, segura i intuïtiva sense haver de canviar la manera en que està estructurada la pàgina de la facultat.

El ventall de tecnologies, mètodes, llenguatges, models, etc, per escollir a l'hora de desenvolupar una aplicació web és molt ample. El que he fet, és triar el que més s'adapta a lo que ja tenim, sigui mes senzill i tenint en compte les llicències intentant que sigui de codi lliure. Per al navegador de fitxer m'he decantat pel **PGRFileManager**, per la seva senzillesa, la seva interfície intuïtiva, la facilitat alhora d'entendre el seu funcionament i sobretot la fàcil integració amb l'editor WYSIWYG **ckeditor**.

¹WYSIWYG: és l'acrònim de la frase anglesa "*What You See Is What You Get*" (el que veus és el que obtens). En l'àmbit de la programació i el disseny es refereix a la tecnologia informàtica que permet que el que es veu durant l'edició o programació es correspongui més o menys acuradament amb el resultat final (imprès o en pantalla).

²CMS: Content Manager System. És un programa informàtic que té per finalitat facilitar la creació de continguts y la presentació d'aquests continguts.

Pel software de versions m'he decantat pel subversion ja que al servidor de la universitat s'utilitza, és software lliure i té unes llibreries php5-svn molt útils per treballar en PHP.

1.1 Motivació

De cada dia, les aplicacions es decanten més cap a la web o el “cloud computing” com està de moda ara. La seva facilitat d'accedir des de qualsevol part del món amb un simple navegador, no necessitar la instal·lació d'una aplicació client a l'ordinador, la garantia que sempre és compatible amb qualsevol sistema operatiu, que les dades estan centralitzades i que per tant, és més difícil la seva pèrdua, fa molt atractiu i amb molta projecció tot aquest mercat del desenvolupament web.

Les empreses d'avui en dia s'estan especialitzant molt en tot el que és el “cloud computing”. Això fa que hi hagi un gran nombre d'ofertes de treball en aquest camp i els professionals estiguin molt cotitzats. Dins aquest món de la web, pel fet de ser lliure i molt senzill, està destacant el PHP, ja que és un llenguatge molt senzill d'aprendre i que ha tengut més acceptació en la majoria de projectes grans en la web en software lliure, com per exemple els CMS.

El fet d'intentar solucionar un problema que passa en el món real amb aquest tipus de tecnologia, és el que va fer decidir-me i motivar-me per a realitzar aquest projecte.

1.2 Estat de l'art

Per realitzar el nostre projecte, buscarem un navegador de fitxers via web que es pugui adaptar al màxim a les nostres necessitats. Aquest ha de ser en PHP ja que és la tecnologia de la que disposam al servidor de la facultat. Al navegador de fitxers que triem li farem les modificacions necessàries per tal que s'adapti a les nostres necessitats. Un software de versions que s'adapti fàcilment a la nostra aplicació i un editor de text i html en javascript per poder editar les pàgines on-line.

Navegadors de fitxers via web

Un navegador de fitxers és un programa informàtic que facilita la interacció amb el sistema de fitxers. Típicament permet realitzar operacions bàsiques com ara: accés, canvi de nom, còpia, eliminació, etc.

Explorer: és un file manager que utilitza Ajax–PHP. El podem trobar a <http://explorer.sourceforge.net>

Ens serveix per:

- navegar directoris i fitxers al servidor.
- editar, copiar, moure, eliminar fitxers.
- cercar, pujar i descarregar fitxers.
- crear i extreure arxius.
- crear fitxers i directoris.
- canviar permisos.

Possiblement és el navegador més sofisticat que he pogut trobar a la xarxa. Però, pel que necessitem, fins i tot té massa opcions. És un pel difícil de d'entendre bé el seu funcionament i volem orientar la nostra solució a persones amb coneixements bàsics d'informàtica.

Osfilemanager també és un filemanager fet amb Ajax-PHP i el podem trobar a: <http://www.osfilemanager.com>

Ens serveix per:

- Navegar directoris i fitxers al servidor
- Editar, copiar, moure, eliminar fitxers,
- Cercar, pujar i descarregar fitxers,
- Crear i extreure arxius,
- Crear fitxers i directoris,
- Canviar permisos.
- Crear usuaris.
- Manegar usuaris.

Requereix d'una de base de dades MySQL. Per aquest motiu l'hem descartat evitant la instal·lació de programari adicional al servidor.

CKFinder: és file manger molt intuïtiu i senzill d'utilitzar. Fet amb Ajax-PHP. El podem trobar <http://ckfinder.com>

Ens serveix per:

- Navegar directoris i fitxers al servidor.
- Editar, copiar, moure, eliminar fitxers.
- Cercar, pujar i descarregar fitxers.
- Crear i extreure arxius.
- Crear fitxers i directoris.

És un navegador que s'adapta molt fàcilment al Ckeditor, però la seva llicència no acaba de ser del tot lliure i per tant també el descartem.

Ajaxplorer: també és un filemanager fet amb Ajax-PHP molt intuïtiu i molt complet. El podem trobar a <http://www.ajaxplorer.info>

Ens serveix per:

- Navegar directoris i fitxers al servidor.
- Editar, copiar, moure, eliminar fitxers.
- Cercar, pujar i descarregar fitxers.
- Crear i extreure arxius.
- Crear fitxers i directoris.
- Canviar permisos.
- Crear usuaris.
- Manegar usuaris.

Com a navegador és un dels més complets, però el descartem perquè hi resulta complicat adaptar un editor.

KCFinder: també utilitza Ajax-PHP, és un dels més bonics i més fàcil funcionament. Està sota la llicència GPLv2³, que és un punt molt a tenir en compte. Utilitza les llibreries JQuery. El podem trobar a <http://kcfinder.sunhater.com/>

Ens serveix per:

- Navegar directoris i fitxers al servidor.
- Editar, copiar, moure, eliminar fitxers.
- Cercar, pujar i descarregar fitxers.
- Crear i extreure arxius.
- Crear fitxers i directoris.
- Descarregar distints fitxers comprimits en zip.

Com a navegador també és molt complet, però igual que abans no resulta gens fàcil adaptar un editor.

³La **GNU GPL**, és un tipus de llicència per a programari que permet la còpia, distribució (comercial o no) i modificació del codi, sempre que qualsevol modificació es continuï distribuint amb la mateixa llicència GPL. La llicència GPL no permet la distribució de programes executables sense el codi font corresponent o una oferta de com obtenir-lo gratuïtament.

PGRFileManager: un altre filemanager Ajax-PHP. És un dels més senzills i pot ser menys desenvolupats de tots ells. Està sota la llicència MIT⁴ i ja ve amb l'editor CKEditor integrat.

Ens serveix per:

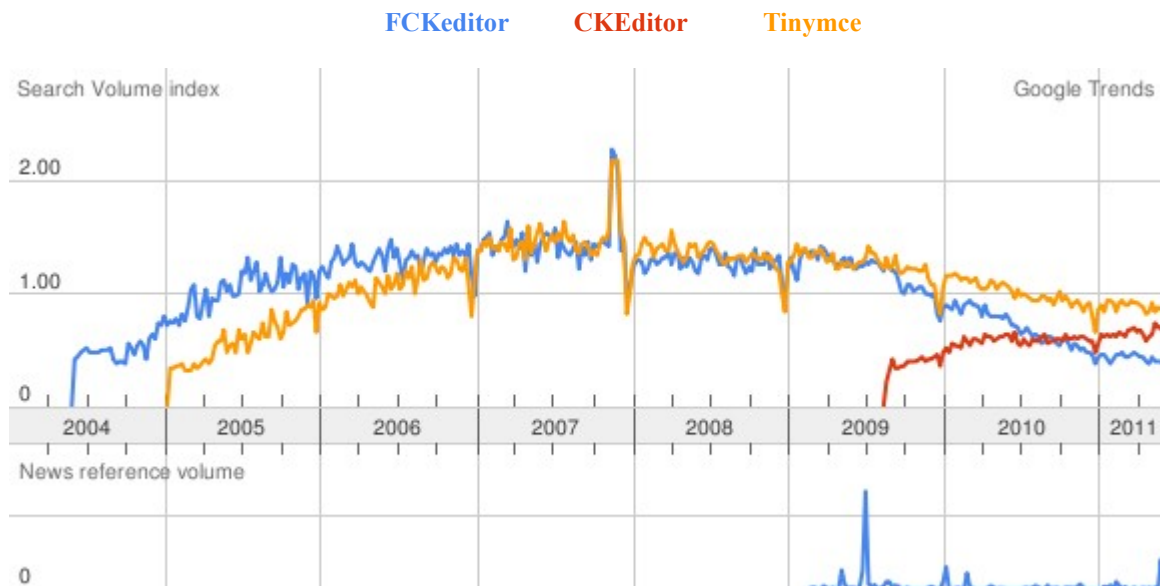
- Navegar directoris i fitxers al servidor.
- Editar, copiar, moure, eliminar fitxers.
- Cercar, pujar i descarregar fitxers..
- Crear directoris.

És el navegador que hem escollit, bàsicament per la seva integració amb el Ckeditor i la seva llicència.

Editors WYSIWYG per integrar amb navegador de fitxers

Un editor WYSIWYG, és un editor que li dóna a la web el poder de las aplicacions d'escriptori a l'estil dels processadors de text com el writer de l'openoffice, sense la necessitat d'instal·lar cap component a l'ordinador del client.

Tot i que n'hi ha moltíssims, n'hi ha dos que destaquen per sobre dels altres. Com podem veure amb el google trends, el seu ús és molt similar.



Il·lustració 1: Volum de cerques, segons google trends.

⁴MIT: s una llicència lliure de programari original del Massachusetts Institute of Technology (MIT). Específicament, és compatible amb la GNU General Public license permissiva, entenent que permet la re utilització en programari propietari amb la condició que la llicència sigui distribuïda amb el programa.

TinyMCE: És un editor web, fet en Javascript HTML WYSIWYG alliberat sota la llicència LGPL.

El podem trobar a <http://tinymce.moxiecode.com>

- Generació de codi XHTML 1.0.
- Suport CSS.
- Incorporar formularis.
- Formateig de Font.
- Tallar, copiar, aferrar.
- Inserció d'imatges
- Creació de taules.
- Menús contextuais amb botó dret.

El descartem perquè no s'adapta fàcil amb el navegador elegit.

CKEditor: És un editor de text HTML /WYSIWYG, successor del FCKEditor, molt similar al tinyMCE. Està suportat per la majoria de navegadors.

Les seves característiques són:

- Generació de codi XHTML 1.0.
- Suport CSS.
- Incorporar formularis.
- Formateig de Font.
- Tallar, copiar, aferrar.
- Inserció d'imatges
- Creació de taules.
- Menús contextuais amb botó dret.
- Successor del FCKeditor.

És l'editor elegit per la seva adaptació amb el navegador de fitxers que ens agrada, el PGRFileManager.

Software de versions

Un software de control de versions és un repositori remot on s'allotgen les còpies i revisions comuns. A la vegada permet fer una còpia local on es treballa de forma independent, quan es creu convenient, la còpia s'envia al repositori remot i el sistema verifica que no existeixen conflictes entre les diferents còpies.

Una altra característica és la de permetre la mostra de canvis entre diferents versions, encara que això té una major efectivitat entre els fitxers de text escrit, també és possible utilitzant-lo per a fitxers binaris.

En el món de software de versions lliures i gratuïts, en sobresurten dos: Git i subversion.

Un resum de les diferències:

- Git és molt més ràpid que Subversion.
- Subversion li permet veure només un subarbre d'un dipòsit, Git requereix clonar el repositori complet (inclosa la història) i crear una còpia de treball que reflecteix com a mínim un subconjunt dels elements sota control de versions.
- Repositoris Git són molt més petits que subversions (per al projecte Mozilla, més petit 30x)
- Git va ser dissenyat per ser totalment distribuït des de l'inici, permetent que cada desenvolupador té control local complet
- Les branques a Git són el recurs més senzill i menys pesat que el de Subversion.
- Branques git porten tota la seva història
- El "merge" en Git, no cal recordar que la revisió que es va fusionar a partir de (aquest benefici es va obviar amb el llançament de Subversion 1.5)
- Git proporciona una millor auditoria de la branca i també en combinar esdeveniments
- Formats de git repo arxiu són simples, de manera que la reparació és fàcil i la corrupció és estranya.
- Còpia de seguretat dels dipòsits de Subversion central és potencialment més simple - ja que es pot triar a les carpetes distribuïdes dins d'un dipòsit en git
- Git clons repositori de còpies de seguretat d'actuar com a repositori complet

- Interfície d'usuari de Subversion és més madur que Git
- Anar a través de les versions és més simple en Subversion, ja que utilitza números de seqüència de revisió (1,2,3,..); Git utilitza impredecible hash SHA-1. Anar cap enrere en Git és molt fàcil usant el "^" la sintaxi, però no hi ha manera fàcil de caminar cap endavant.

Característiques principals de Git sobre Subversion

La naturalesa distribuïda

Git va ser dissenyat des del principi com un sistema de control de versions distribuït.

En un VCS distribuïts com Git cada usuari té una còpia completa del repositori de dades emmagatzemades localment, de manera que l'accés a l'arxiu històric molt ràpid, així com permetre la funcionalitat completa quan es desconnecta de la xarxa. També significa que cada usuari té una còpia de seguretat completa del dipòsit.

En un VCS centralitzat com Subversion només el repositori central té la història completa. Això significa que els usuaris s'han de comunicar a través de la xarxa amb el repositori central per obtenir la història sobre un arxiu. Còpies de seguretat s'ha de mantenir independentment de la VCS. Si el dipòsit central es perd a causa d'un fallada del sistema que s'ha de restaurar la còpia de seguretat i els canvis que des última còpia de seguretat és probable que es perdi.

Les branques en Git és un concepte central utilitzat cada dia per tots els usuaris. En el Subversion són més complexes i sovint utilitzen amb moderació.

La raó de les branques són tan centrals en Git és el directori de treball tots els desenvolupadors és en si mateixa una branca. Fins i tot si dos desenvolupadors estan modificant dos arxius no relacionats diferents al mateix temps que és fàcil de veure aquests dos directoris diferents de treball com les diferents branques derivades de la revisió de la mateixa base comuna del projecte.

En Subversion, totes les branques i les etiquetes són còpies. De vegades això és un inconvenient, és fàcil de fer un “check out” de tot el repositori per error.

Git és extremadament ràpid. Com que totes les operacions (a excepció del “push” a buscar) són locals que no hi ha latència de xarxa involucrats:

- Realitzar un “diff”
- Veure arxiu històric.
- Confirmar els canvis.
- Combinar les branques.
- Obtenir qualsevol altra revisió d'un arxiu (no únicament la versió realitzada amb anterioritat).
- Intercanviar entre branques.

Repositori git i mida dels directoris de treball són molt petites en comparació amb SVN.

A Git els directoris de treball (inclosos els repositoris) són més petits que els directoris de treball de l'SVN.

Subversion té algunes característiques notables que Git en l'actualitat no té ni mai tindrà.
Interfícies d'usuari de maduresa

Subversion té actualment més àmplia gamma d'eines d'interfície d'usuari que Git. Per exemple, hi ha plugins SVN disponible pels entorns de desenvolupament més populars.

Interfície d'usuari principal de Git és a través de la línia d'ordres. Hi ha dues interfícies gràfiques: git-gui (distribuït amb Git) i qgit, que està donant grans passos cap a la provisió altra interfície gràfica de funcions completes. També gitk, el navegador de la història gràfica, pot ser més que un simple lector de registre de fantasia.

Repositori únic

Atès que Subversion només admet un únic repositori no hi ha dubte sobre on s'emmagatzema alguna cosa. Una vegada que l'usuari coneix la URL del repositori que raonablement es pot suposar que tots els materials i totes les branques relacionades amb aquest projecte estan sempre disponibles en aquesta ubicació. Còpia de seguretat en cinta / CD / DVD també és senzill, ja que és precisament un lloc que necessita tenir el suport regularment.

Com que Git és distribuït, no tot el relacionat amb un projecte pot ser emmagatzemats en el mateix lloc. Per tant, pot haver cert grau de confusió d'on descarregar una branca particular, llevat que la ubicació del repositori és sempre explícitament especificat.

Controla l'accés

Atès que Subversion té un repositori central únic, és possible especificar llegir i escriure en els controls d'accés en un sol lloc i fer que s'apliquin en tot el projecte.

Subversion es pot utilitzar amb els arxius binaris (que es detecta automàticament, si la detecció falla, vostè ha de marcar el mateix arxiu binari). Igual que Git.

Només que amb Git, el valor per defecte és el d'interpretar els arxius com binaris, per començar.
Seguiment de canvis

Amb Subversion, es pot veure només com un subdirectori del dipòsit. Això no és possible amb Git. Per a un projecte gran, això vol dir que sempre has de descarregar el repositori complet, encara que només es necessita la versió actual d'un sub-directori. En temps on les connexions ràpides a Internet només estan disponibles en la majoria de les ciutats i el trànsit a través de connexions d'Internet mòbil és car, git pot costar molt més temps i diners en les zones rurals o amb dispositius mòbils. Aquest és, sens dubte mitigat per la petita grandària dels dipòsits git.

En primer lloc, com SVN assigna números de revisió seqüencial (a partir d'1), fins i tot molt antics projectes com Mozilla tenen un curt nombre de revisió únic (Mozilla és només fins a 6 dígit). Molts usuaris consideren aquesta pràctica en entrar en les versions de fins d'investigació històrica.

També trobem aquest número fàcil d'integrar en el seu producte, suposadament el que és fàcil determinar quines fonts es van utilitzar per crear un executable en particular. No obstant això, atès el nombre de revisió és global per a tot el repositori, incloent totes les branques, encara hi ha una qüestió de quina branca el nombre de revisió correspon.

Llevat que la revisió confirmada última es registra. Atès que les revisions són globals per a un dipòsit, la revisió confirmats per últim cop fa possible determinar quina branca es va utilitzar

Tecnologies escollides

El ventall de tecnologies, mètodes, llenguatges, models, etc per escollir a l'hora de desenvolupar una aplicació web és molt ample. El que he fet, és el q mes s'adapta a lo que ja tenim i sigui més senzill i tenint en compte les llicències de codi lliure.

Per al navegador de fitxer m'he decantat pel **PGRFileManager**, per la seva senzillesa, la seva interfície intuïtiva, la facilitat alhora d'entendre el seu funcionament i sobretot la fàcil integració amb l'editor de text WYSIWYG **CKEditor**.

Pel software de versions, tot i que el git sembla a priori superior en prestacions al subversion, m'he decantat pel subversion ja que al servidor de la universitat s'utilitza, és software lliure i existeixen unes llibreries php5-svn molt útils per treballar en PHP.

1.3 Objectius

Les principals fites que volíem assolir en el projecte serien:

1. Crear una interfície web a partir d'alguna aplicació ja existent per adaptar-la a les nostres necessitats. Aquesta interfície ha de ser intuïtiva i ha de permetre als usuaris crear , modificar i esborrar fitxers .
2. Adaptar aquesta interfície web per que funcioni amb un *software de versions*⁵, així poder accedir en tot moment a versions anteriors de la web en cas de necessitat.
3. Garantir que no es perd informació.
4. Permetre l'edició simultània de més d'un fitxer de la web.
5. Detecció, control i solució de modificacions incorrectes del contingut.

Acadèmicament els objectius principals són posar en pràctica i aprofundir els diferents coneixements adquirits al llarg de la carrera. Familiaritzar-me i adquirir experiència amb els software de versions. Aprofundir en els coneixements de programació i disseny web amb PHP, javascript i Ajax.

⁵ software de versions: Els sistemes de control de versions faciliten l'administració de les distintes versions del software desenvolupat, així com les possibles especialitzacions realitzades.

2 Avaluació tecnològica

En aquest apartat procedim a definir i a explicar detalladament les tecnologies utilitzades per a desenvolupar el projecte.

2.1 Llenguatges de programació

HTML: Acrònim en anglès de HyperText Markup Language. Desenvolupada per la World Wide Web Consortium (W3C). És un llenguatge estàtic d'etiquetes que es pot incorporar en el codi d'altres llenguatges .

Avantatges:

- Senzill que permet descriure hipertext.
- Text presentat de forma estructurada i agradable.
- No necessita de grans coneixements quan es conta amb un editor de pàgines web .
- Arxius petits.
- Desplegament ràpid.
- Llenguatge de fàcil aprenentatge.
- Ho admeten tots els exploradors.

Desavantatges:

- Llenguatge estàtic.
- La interpretació de cada navegador pot ser diferent.
- Guarda moltes etiquetes que poden convertir-se en “escombraries” i dificulten la correcció.
- El disseny és més lent.
- Les etiquetes són molt limitades.

PHP: acrònim recursiu que significa “PHP Hypertext Pre-processor”. És un llenguatge de script interpretat en el costat del servidor utilitzat per a la generació de pàgines web dinàmiques, embegudes en pàgines HTML i executades en el servidor. La major part de la seva sintaxi ha estat presa de C, Java i Perl amb algunes característiques específiques.

Avantatges:

- Molt fàcil d'aprendre.
- Es caracteritza per ser un llenguatge molt ràpid.
- Suporta en certa mesura l'orientació a objecte. Classes i herència(versió 4, en la versió 5 es millora).

- És un llenguatge multi-plataforma: Linux, Windows, entre uns altres.
- Capacitat de connexió amb la majoria dels sistemes de gestió de base de dades: MySQL, PostgreSQL, Oracle, MS SQL Server, entre d'altres.
- Capacitat d'expandir el seu potencial utilitzant mòduls.
- Posseïx documentació en la seva pàgina oficial la qual inclou descripció i exemples de cadascuna de les seves funcions.
- És lliure, pel que es presenta com una alternativa de fàcil accés per a tots.
- Inclou gran quantitat de funcions.
- No requereix definició de tipus de variables ni maneig detallat del baix nivell.
- Hi ha molts llocs d'albergatge gratuïts que suporten PHP.

Desavantatges:

- Es necessita instal·lar un servidor web.
- Tot el treball ho realitza el servidor i no delega al client. Per tant pot ser més ineficient a mesura que les sol·licituds augmentin de nombre.
- La llegibilitat del codi pot veure's afectada al barrejar sentències HTML i PHP.
- La programació orientada a objectes és encara molt deficient per a aplicacions grans (versió 4, en la versió 5 es millora).
- Dificulta la programació modular (versió 4, en la versió 5 es millora).
- Dificulta l'organització per capes de l'aplicació.

Seguretat:

PHP és un poderós llenguatge i intèrpret, ja sigui inclòs com part d'un servidor web en forma de mòdul o executat com un binari CGI separat, és capaç d'accedir a arxius, executar comandes i obrir connexions de xarxa en el servidor. Aquestes propietats fan que qualsevol cosa que sigui executada en un servidor web sigui insegura per naturalesa. PHP està dissenyat específicament per a ser un llenguatge més segur per a escriure programes CGI que Perl o C.

Javascript: És un llenguatge interpretat, no requereix de compilació prèvia. Utilitzat principalment per a pàgines web. És similar a Java, encara que no és un llenguatge orientat a objectes, ell mateix no té herències. La majoria dels navegadors en les seves últimes versions interpreten codi Javascript.

Avantatges:

- Llenguatge de scripting segur i fiable.
- Els scripts tenen capacitats limitades, per raons de seguretat.
- El codi Javascript s'executa en el client.

Desavantatges:

- Codis visibles per qualsevol usuari.
- El codi ha de descarregar-se completament.
- Pot posar en risc la seguretat del lloc, amb l'actual problema cridat XSS 3.

2.2 Llibreries

JQuery: és una llibreria JavaScript ràpida, que permet simplificar la manera d'interactuar amb els documents HTML, manipular l'arbre DOM2, manejar esdeveniments, desenvolupar animacions i facilitar crides AJAX per aconseguir connexions asíncrones.

Característiques:

- No és intrusiu (no es mescla el codi HTML amb el de JavaScript).
- Simplifica les tasques habituals de JavaScript.
- És independent del navegador.
- Els seu pes és molt poc.
- Extensible amb una comunitat de desenvolupament molt gran.
- Molt fàcil d'aprendre.

AJAX: acrònim de Asynchronous JavaScript And XML, és una tècnica emprada en el desenvolupament d'aplicacions webs interactives. S'executen en la banda del client, i manté una comunicació asíncrona amb el servidor en segon pla, així es pot aconseguir modificar la pagina sense tenir que recarregar-la.

AJAX no és una tecnologia en si, és un terme que engloba JavaScript, l'objecte XMLHttpRequest, el DOM, XHTML i CSS (separant informació de presentació)

PHP-SVN: És una llibreria de php que ens permet comunicar amb els repositoris i còpies locals de l'SVN. És una llibreria en estat experimental i la documentació és molt millorable.

2.3 Anàlisi i especificació

Sistema d'usuaris

Usuari : És tot aquell que no esta registrat al nostre sistema, per tant, només pot visitar la pàgina i no pot utilitzar el nostre navegador de fitxers.

Col·laborador: És aquell usuari que està registrat en el nostre sistema i té permisos per utilitzar el nostre navegador de fitxers.

Administrador: Usuari especial que serà notificat i haurà de supervisar el que ha passat en cas de que hi hagi una incidència.

Catàleg de requeriments

Per començar a treballar en el nostre treball, primer hem de definir uns requeriments que volem que tingui l'aplicació per tal d'assolir els objectius i que permetin marcar les pautes a seguir en el desenvolupament de l'aplicació.

Després d'haver triat el navegador d'arxius i haver analitzat aspectes com les necessitats que creiem que poden tenir els nostres usuaris, definirem les modificacions que li farem. En el nostre cas realment només definirem el de l'usuari col·laborador, ja que l'usuari normal només pot navegar per la web i l'administrador, rebrà un correu amb la notificació de quins arxius han entrat en conflicte i ell ja s'encarregarà de solucionar-ho amb unes altres eines.

Requeriments del Col·laborador:

- Crear arxius de tipus: txt, php, html i js.
- Modificar arxius del tipus anomenats anteriorment.
- Eliminar arxius tant dels anomenats anteriorment com fotos i pdf's.
- Guardar arxius com a borradors.
- Pujar fotos.
- Pujar qualsevol tipus de fitxer.

Requeriments de l'aplicació (funcionalitats afegides al PGRFilemanager):

- Crear un /home/usuari on l'usuari pugui guardar els borradors que vagi creant.
- Guardar automàticament, cada cert temps i de manera transparent a l'usuari una còpia de seguretat de l'arxiu en edició.
- Creació de nous fitxers.
- Login de distints usuaris.
- Adaptació del navegador de PGRFilemanager amb el software de versions "subversion".
- Resolució de conflictes si s'està editant alhora un mateix fitxer.

Casos d'us

Un cas d'ús és una descripció de la seqüència d'interaccions que es produeixen entre un actor i el sistema, quan l'actor utilitza el sistema per portar a terme una tasca específica.

El sistema té tres usuaris diferenciats que són: usuari visitant, usuari col·laborador i l'administrador.

En el diagrama de casos d'ús representem el sistema com una caixa rectangular amb el nom del cas a dins. Els casos d'ús estan situats a l'interior de la caixa del sistema i els actors situats a fora. Cada actor està unit a cada cas d'ús que participa.

L'especificació dels casos d'ús està composta pels camps: nom, actors (aquells que interactuen amb el sistema), precondicions (fets que s'han de complir per a que el cas d'ús es pugui portar a terme), postcondicions (resultat d'haver conclòs satisfactòriament el cas d'ús), el curs d'aconteixement, que correspon a l'execució normal i exitosa del cas. I per últim, els camins alternatius que poden succeir durant el transcurs del cas.

2.1 Definició del actors.

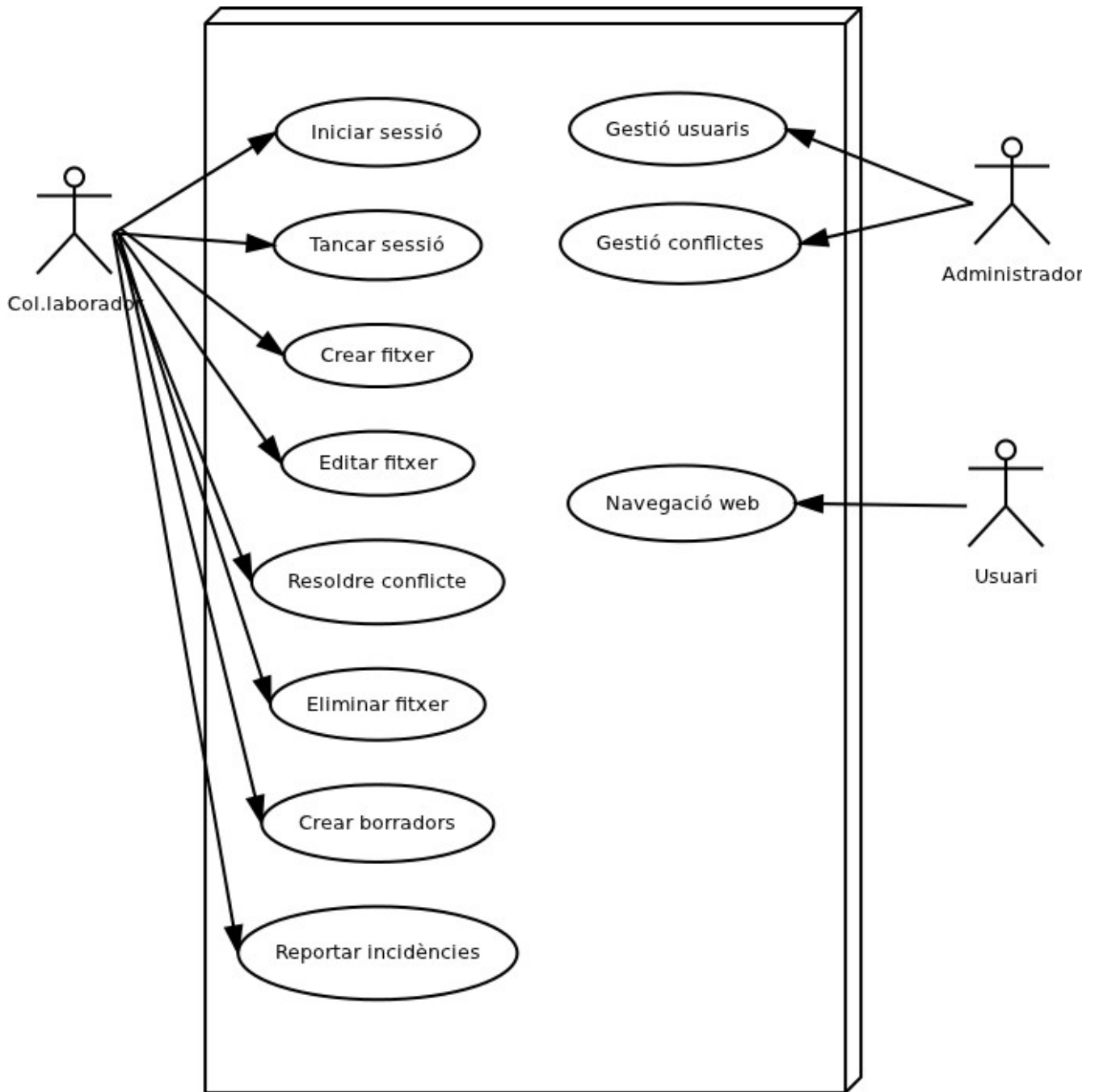
ACT-01	Usuari Col·laborador
Descripció	Persona que edita continguts.

ACT-02	Administrador
Descripció	Persona responsable de la supervisió i resolució dels possibles conflictes.

ACT-03	Usuari visitant
Descripció	Persona que navega pels continguts de la web.

Cas d'us:

Seguidament veurem els casos d'us complets de tots els actors que intervenen.



Especificació:

Cas d'ús: Iniciar sessió	
Descripció:	
Un usuari registrat en el sistema s'identifica per accedir al contingut.	
Actor principal:	
Usuari col·laborador.	
Precondició:	
L'usuari col·laborador està registrat en el sistema.	
Postcondició:	
S'inicia la sessió de l'usuari.	
Aconteixements:	
Actor	Sistema
1. L'usuari registrat intenta accedir al sistema.	
2. L'usuari registrat introdueix les seves credencials.	
	3. El sistema comprova que les credencials introduïdes són correctes.
	4. El sistema permet l'accés a l'usuari registrat.
Camins alternatius:	
3a. Les credencials introduïdes són incorrectes	
3a1. El sistema notifica a l'usuari que les credencials introduïdes no són vàlides.	

Cas d'ús: Tancar sessió	
Descripció:	
Un usuari registrat vol tancar la sessió oberta en el sistema.	
Actor principal:	
Usuari col·laborador	
Precondició:	
L'usuari col·laborador està registrat en el sistema i té una sessió oberta.	
Postcondició:	
Es tanca la sessió de l'usuari.	
Aconteixements:	
<i>Actor</i>	<i>Sistema</i>
1. L'usuari registrat intenta tancar la sessió.	2. El sistema tanca la sessió de l'usuari.
Camins alternatius:	
-	

Cas d'ús: Crear fitxer	
Descripció:	
Un usuari registrat en el sistema vol crear un fitxer.	
Actor principal:	
Usuari col·laborador	
Precondició:	
L'usuari col·laborador està registrat i té sessió oberta en el sistema.	
Postcondició:	
Es crea un nou fitxer.	
Aconteixements:	
<i>Actor</i>	<i>Sistema</i>
1. L'usuari registrat prem el botó "crear nou fitxer" per a crear un nou fitxer.	2. El sistema li treu un editor WYSIWYG.
3. L'usuari introdueix les dades.	3. El sistema li demana que posi un nom al fitxer.
4. L'usuari introdueix el nom.	5. El sistema guarda el nou fitxer.

Camins alternatius:

-

Cas d'ús: Editar fitxer

Descripció:

Un usuari registrat en el sistema vol editar un fitxer existent.

Actor principal:

Usuari col·laborador

Precondició:

L'usuari col·laborador està registrat i té sessió oberta en el sistema.

Postcondició:

Edita un fitxer existent.

Aconteixements:

Actor

Sistema

1. L'usuari registrat demana editar un fitxer.

3. L'usuari introdueix les dades.

2. El sistema li treu un editor WYSIWYG amb el contingut del fitxer.

4. L'usuari afegeix o modifica informació de l'editor i guarda.

3. El sistema li demana que posi un nom al fitxer.

5. El sistema guarda el fitxer modificat.

Camins alternatius:

5a. El sistema detecta que hi ha una versió del fitxer més actual al servidor i intenta ajuntar la nova informació amb la versió que es troba al servidor.

- 5aa. El software de versions és capaç de ajuntar els fitxers i guarda el fitxer modificat.

- 5ab. El software de versions no és capaç de ajuntar els fitxers i demana a l'usuari si vol resoldre el conflicte. Veure cas d'us "resoldre conflicte".

Cas d'ús: Resoldre conflicte

Descripció:

Un usuari registrat en el sistema vol guardar un fitxer que entra en conflicte amb un altre.

Actor principal:

Usuari col·laborador

Precondició:

L'usuari col·laborador està registrat, té sessió oberta en el sistema i està intentant guardar un fitxer que ha editat anteriorment que entra en conflicte.

Postcondició:

Resol un conflicte entre fitxers existent.

Aconteixements:

Actor

Sistema

- | | |
|---|--|
| <ul style="list-style-type: none"> 2. L'usuari accepta. 3. L'usuari junta o modifica els fitxer i decideix quin és la versió a guardar. | <ul style="list-style-type: none"> 1.El software de versions no és capaç de ajuntar els fitxers i demana a l'usuari si vol resoldre el conflicte. 2. El sistema li treu dos editors amb el contingut dels dos fitxers que han entrat en conflicte. 4. El sistema guarda la versió que l'usuari a decidit. |
|---|--|

Camins alternatius:

3a. L'usuari no està segur i decideix enviar la incidència a l'administrador. Veure cas d'ús ""reportar incidència"".

Cas d'ús: Eliminar fitxer	
Descripció:	
Un usuari registrat en el sistema vol eliminar un fitxer.	
Actor principal:	
Usuari col·laborador	
Precondició:	
L'usuari col·laborador està registrat, té sessió oberta en el sistema.	
Postcondició:	
Elimina un fitxer existent al sistema.	
Aconteixements:	
Actor	Sistema
1. L'usuari sol·licita eliminar el fitxer.	2. El sistema elimina el fitxer.
Camins alternatius:	
-	

Cas d'ús: Crear borrador	
Descripció:	
Un usuari registrat en el sistema vol guardar una borrador d'un fitxer existent o un de nou.	
Actor principal:	
Usuari col·laborador	
Precondició:	
L'usuari col·laborador està registrat, té sessió oberta en el sistema.	
Postcondició:	
Guarda un borrador.	
Aconteixements:	
Actor	Sistema
1. L'usuari sol·licita guardar un borrador	2. El sistema guarda el borrador del fitxer al directori de borradors de l'usuari.
Camins alternatius:	
-	

Cas d'ús: Reportar incidència	
Descripció:	
Un usuari registrat en el sistema, té un conflicte amb dos fitxers i vol informar l'administrador.	
Actor principal:	
Usuari col·laborador	
Precondició:	
L'usuari col·laborador està registrat, té sessió oberta en el sistema i hi ha un conflicte.	
Postcondició:	
S'envia la incidència a l'administrador.	
Aconteixements:	
Actor	Sistema
	1. El sistema detecta un conflicte i permet a l'usuari enviar la incidència a l'administrador.
2. L'usuari envia la incidència a l'administrador	
Camins alternatius:	
-	

Cas d'ús: Gestió usuari	
Descripció:	
L'usuari administrador introdueix en el sistema un nou usuari.	
Actor principal:	
Administrador.	
Precondició:	
L'usuari administrador.	
Postcondició:	
S'ha creat un nou usuari.	
Aconteixements:	
Actor	Sistema
1. L'usuari crea un nou usuari	
	2. El sistema demana password per l'usuari
Camins alternatius:	

Cas d'ús: Gestió incidències	
Descripció:	
L'administrador rep un correu amb les incidències	
Actor principal:	
Administrador	
Precondició:	
Un usuari col·laborador ha enviat un correu amb una incidència.	
Postcondició:	
L'administrador ha rebut un correu amb un incidència	
Aconteixements:	
<i>Actor</i>	<i>Sistema</i>
	1. El sistema envia un correu electrònic a l'administrador.
Camins alternatius:	

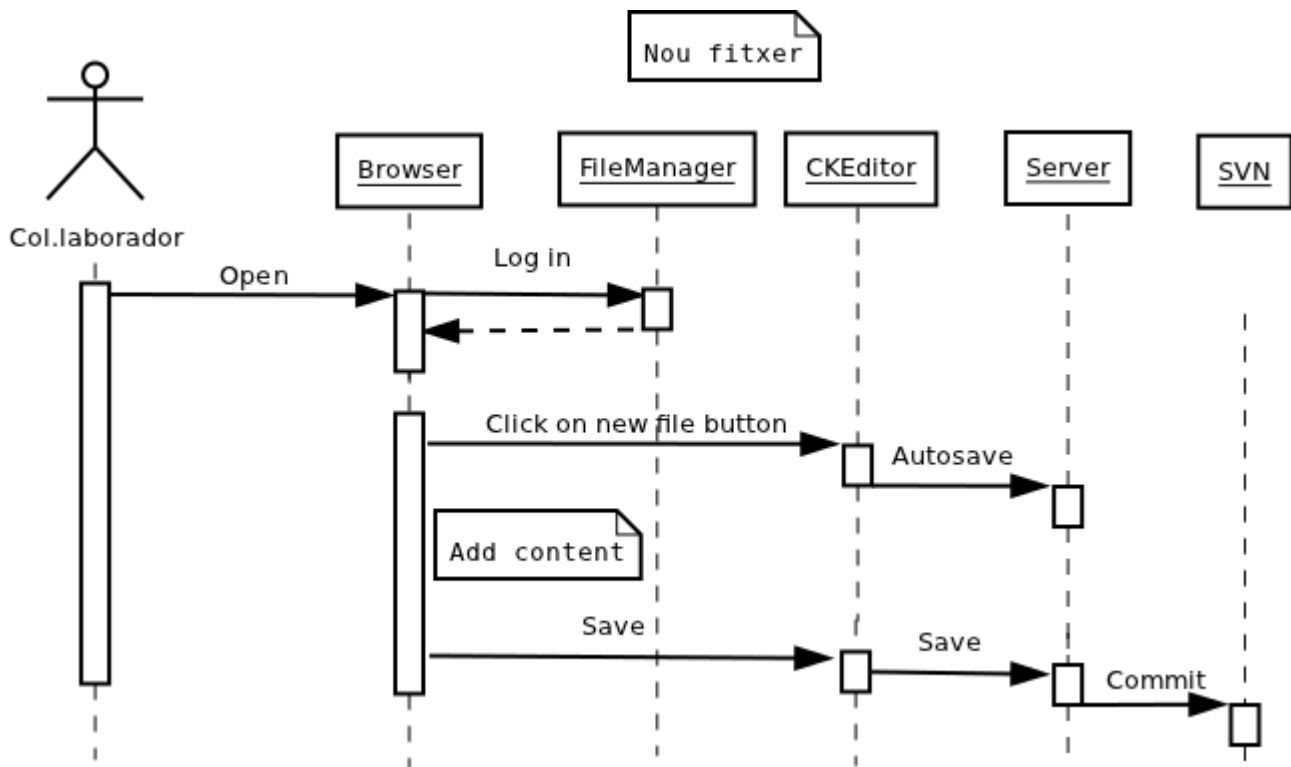
Cas d'ús: Navegació web	
Descripció:	
Un usuari visitant navega per les pàgines de la web.	
Actor principal:	
Usuari visitant.	
Precondició:	
-	
Postcondició:	
Aconteixements:	
<i>Actor</i>	<i>Sistema</i>
1. L'usuari sol·licita veure una pàgina web.	2. El sistema li retorna la pàgina.
Camins alternatius:	

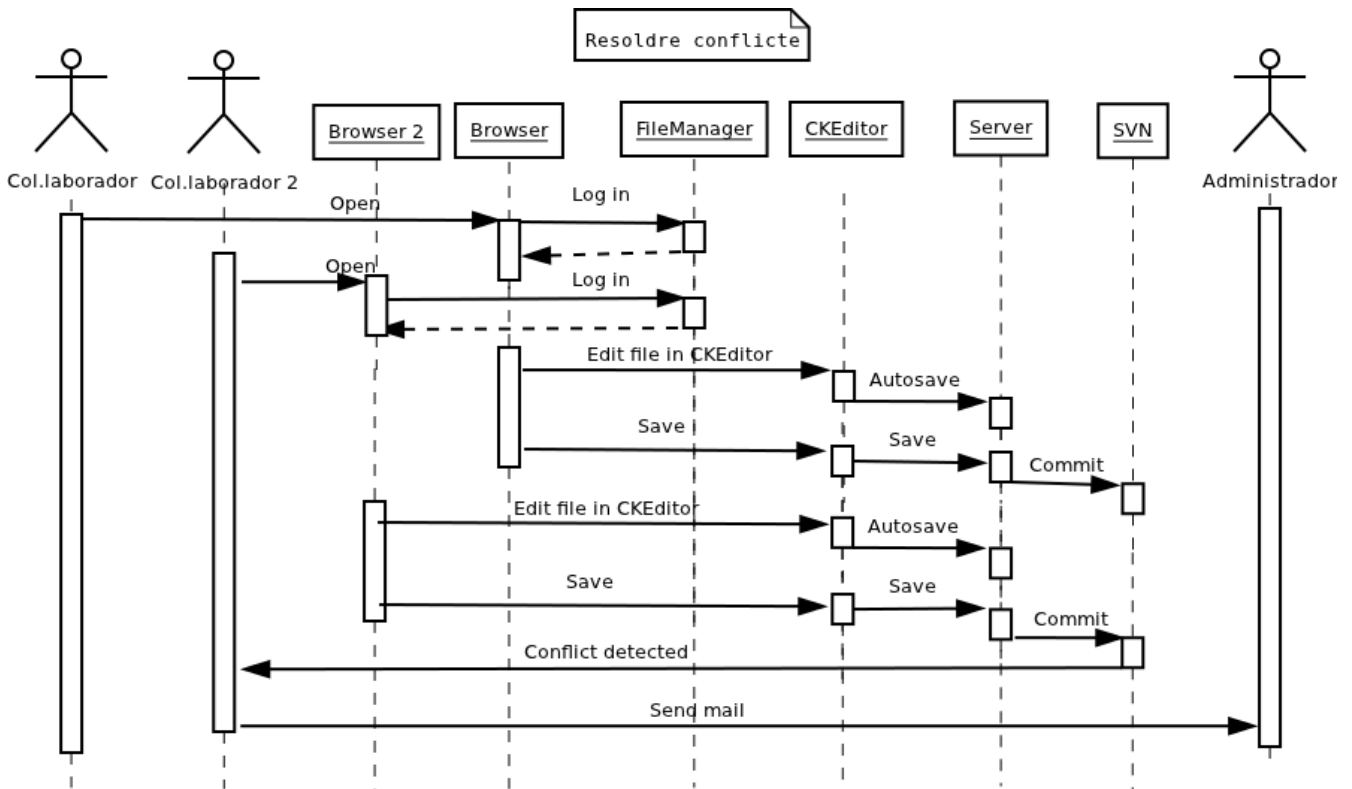
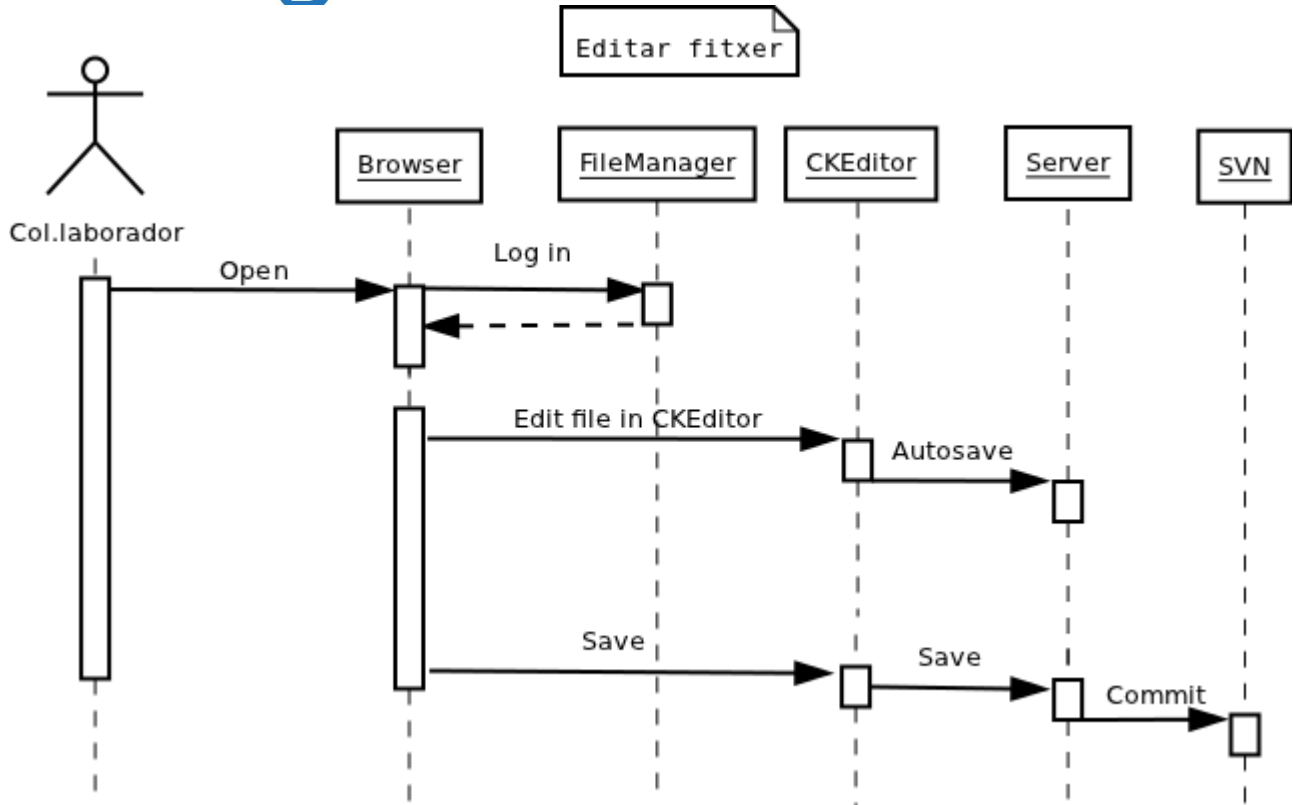
3 Disseny i implementació

Aquí es mostrarà la interacció entre els diferents mòduls de l'aplicació i la manera que interactuen entre ells, així com els detalls de la implementació de cada un d'ells.

3.1 Diagrames de seqüència

Diagrames de seqüència dels casos més rellevants.





3.2 Implementació de les funcionalitats afegides

Per a poder afegir totes aquestes funcionalitats, s'ha afegit un repositori svn al servidor. Com que no es pot treballar directament amb el repositori, és a dir, no es pot modificar el contingut d'aquest. S'ha de crear un “check out”, que és una còpia del repositori per poder-la modificar al nostre gust i després poder guardar els canvis.

Cada vegada que un usuari fa una modificació al directori de treball (“check out”), aquesta l'hem de guardar al repositori, per que ens quedi constància de la modificació i poder accedir a ella en qualsevol moment. Si aquesta modificació no la guardéssim en el repositori, no hi podríem accedir i per tant la perdríem.

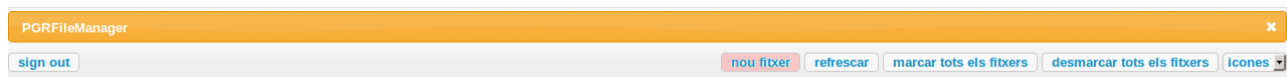
Això ens porta la dificultat d'haver de portar sempre en perfecta sincronia aquests elements: el repositori i el directori de treball principal.

Adaptar software de versions SVN

Una vegada instal·lat i configurat el software de versions, hem d'adaptar el navegador de fitxers que hem instal·lat (PGRFileManager). Per fer-ho, hem utilitzat les llibreries php5-svn, substituint en tot el navegador de fitxers les funcions de creació i eliminació de fitxers per les funcions creació i eliminació de la llibreria php5-svn.

Crear arxius del tipus: txt, php, html i js

Per afegir la funcionalitat per a crear un nou fitxer. S'ha afegit un nou botó al panell de control:



Il·lustració 2: Panell de control.

Quan estem creant un nou fitxer, el podem guardar com a borrador o com a fitxer definitiu en qualsevol part del directori de treball mentre hi tenguem permisos.

Si el guardem com a fitxer definitiu, el nou fitxer es guardarà sempre en el directori actual de treball. Si en canvi, el guardem com a borrador, es guardarà a “home/user/drafts/” + “ruta/directori/actual”

Modificar arxius

Per a modificar un arxiu, la seqüència normal seria: obrir document, editar document, guardar document i guardar al repositori. Però en el nostre cas aquesta seqüència no és vàlida ja que volem que el nostre sistema sigui multi-usuari.

Quan dos o més usuaris estan editant un mateix fitxer i aquest no es capaç de ser barrejat pel nostre software de versions svn, s'ha de generar un conflicte. El software de versions svn ho fa quan són usuaris distints o provenen de còpies (“check outs”) diferents generen el conflicte.

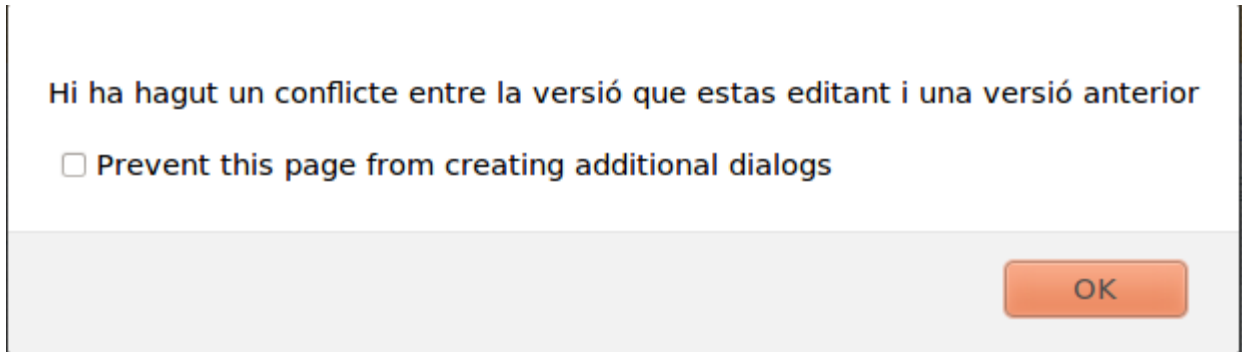
En el nostre cas, encara que dos o més usuaris estiguin editant un mateix fitxer en la nostra aplicació web, l'usuari que tracta amb el repositori és l'usuari del servidor web (apache) i sempre estem fent “commit” (guardar al repositori) de la mateixa còpia (“check out”) això provoca que mai es produeixin conflictes entre usuaris i per tant facin el que facin els usuaris, el software de versions dóna per bona l'última modificació realitzada, ja que per ell sempre és el mateix usuari i la mateixa còpia.

Per a solucionar aquest problema, el que s'ha de fer és o que siguin usuaris distints qui fan els canvis o que es faci “commit” des d'una còpia diferent. Per a solucionar aquest problema, el que he fet és crear un directori temporal “tmp” a dintre de cada directori home de cada usuari. I cada vegada que es vol fer una modificació d'algun fitxer es genera una còpia nova del directori sencer del fitxer a modificar dins “/home/user/tmp”. Quan s'ha acabat de modificar el document es guarda el nou document dins la nova còpia a tmp i es fa un commit al repositori. Això provoca que el commit sempre provengui de còpies diferents i si un fitxer està essent editat per més d'una persona i escau es generin conflictes.

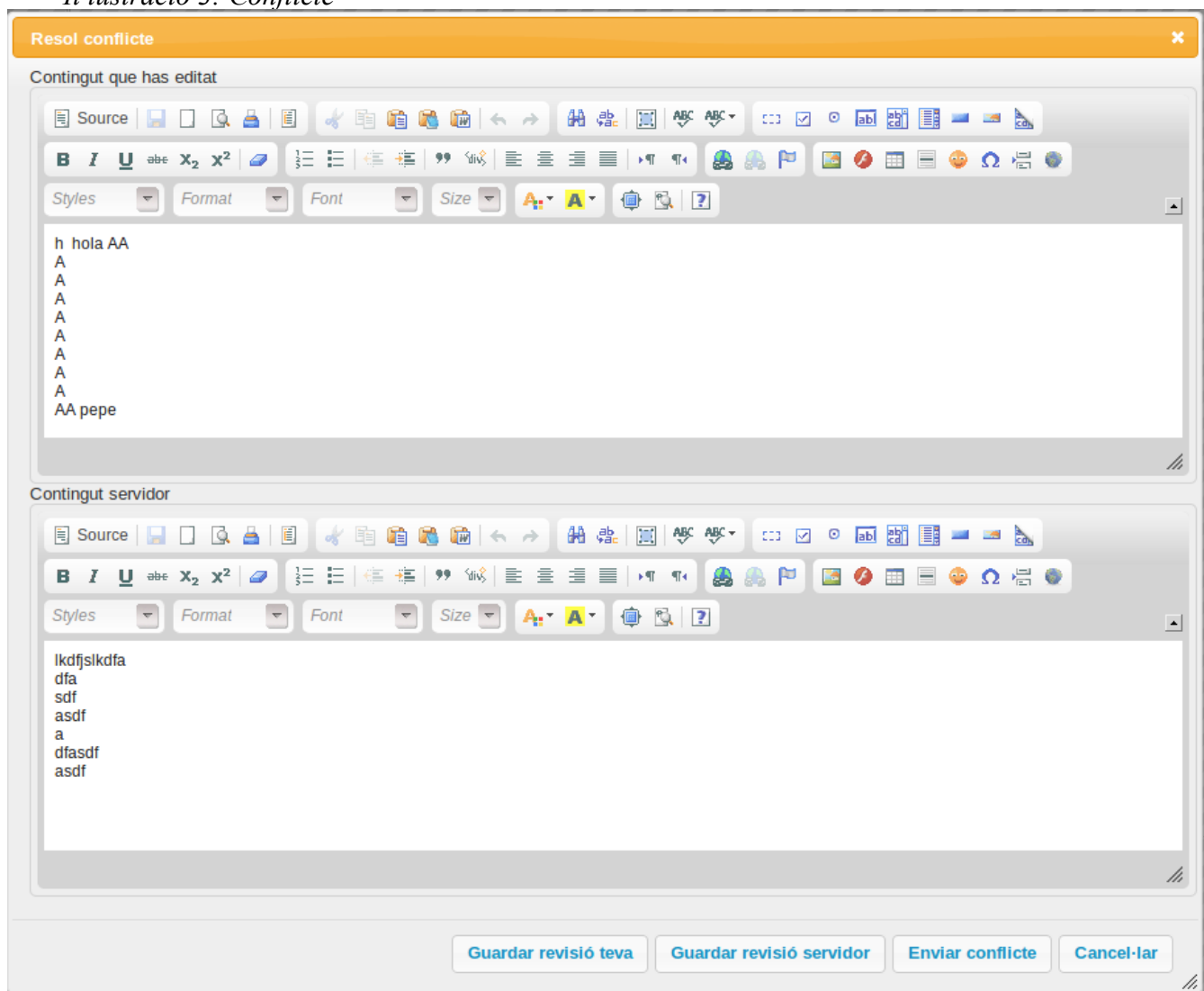
Quan l'SVN detecta un conflicte, genera en el directori del fitxer en conflicte tres fitxers:

- **fitxer_en_conflicte.mine**: fitxer que ha guardat l'usuari.
- **fitxer_en_conflicte.rXX**: últim fitxer guardat en el repositori, on XX representa el número de l'última versió.
- **fitxer_en_conflicte.rYY**: penúltim fitxer guardat en el repositori, on $YY = XX - 1$

Per tant, per poder saber quan un fitxer ha entrat en conflicte, es suficient comprovar si s'ha generat un fitxer .mine al directori. Si és així, es pregunta si es vol solucionar el conflicte.



Il·lustració 3: Conflicte



Il·lustració 4: Resoldre conflicte

Com es pot veure a la il·lustració 4, per a resoldre el conflicte surt una pantalla amb dos editors: en un d'ells hi ha l'última versió del repositori i a l'altre la versió que l'usuari ha guardat i ha generat conflicte.

Aquí l'usuari, té tres opcions, guardar la seva versió, la del servidor o si no està segur que fer, comunicar-ho mitjançant mail a l'administrador perquè es faci càrrec del conflicte.

L'administrador rebrà un correu informatiu, indicant-li on es troba el conflicte i quins fitxers estan involucrats

Guardar borradors

Per a poder guardar arxius com a borradors, s'ha creat un nou directori /home a l'aplicació, que és on es guarden tots els directoris de cada usuari registrats en el sistema. Cada vegada que un usuari es logeja al sistema es comprova si dins el directori home hi té un directori amb el seu nom i un subdirectori “drafts”, si no existeix els crea.

Dins el directori home de l'usuari hi haurà sempre un subdirectori “tmp” i un subdirectori “drafts”. Quan guardem un fitxer com a borrador, aquest es guardarà sempre dins “/drafts” + “directori/actual”, és a dir, si per exemple estem editant el fitxer “<http://localhost/php/index.php>” i el guardem com a borrador, quedaria guardat a “/home/user/php/index.php” creant els directoris que facin falta. Això ho fem per poder tenir guardada la ruta on estava el nostre fitxer.

Si a un moment donat volem recuperar el nostre document situat al directori de borradors (drafts). L'editem de manera normal i el guardem al servidor. Si estem editant un borrador, el comportament alhora de guardar canvia. Si estem editant un borrador i premem el botó de “guardar servidor” aquest sobreescrirà el fitxer original, és a dir, si estem editant “home/user/drafts/php/index.php” i guardem al servidor reescriurà “<http://localhost/php/index.php>”.

Si per altra banda estem editant un borrador i guardem com a borrador, reescriurà el fitxer en edició, és a dir el borrador mateix.

Guardat automàtic del document en edició

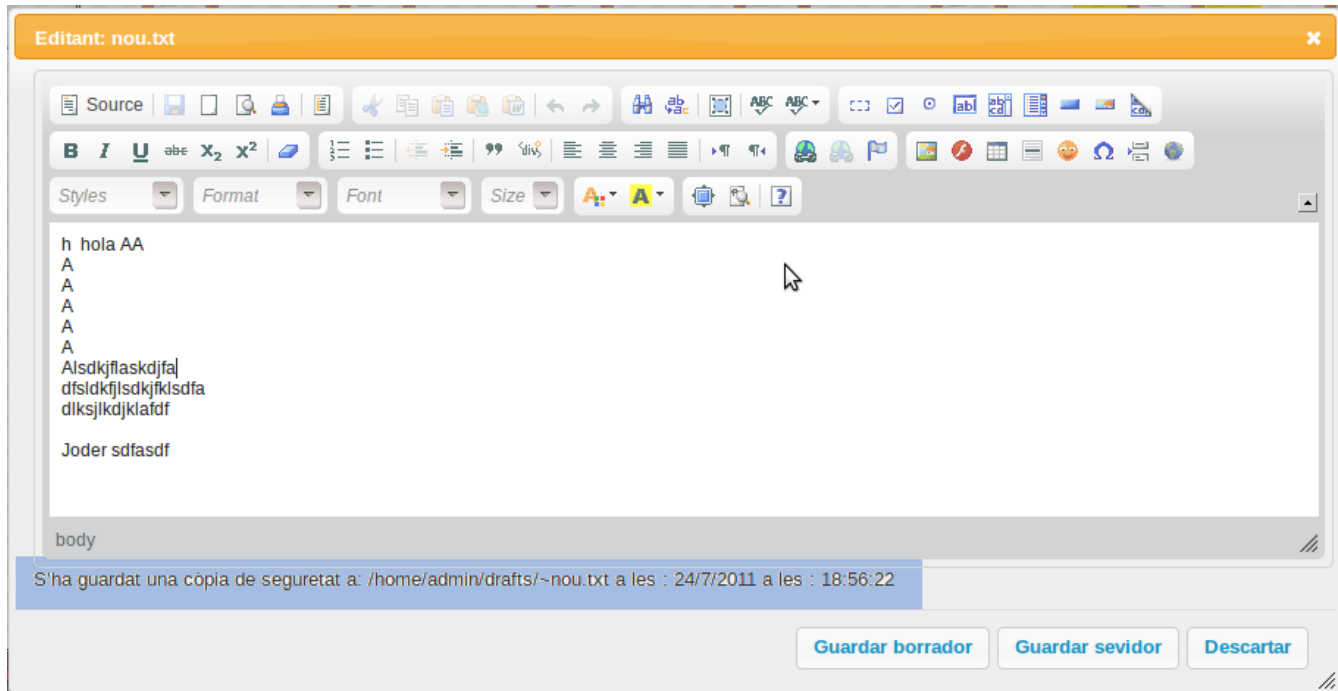
Per fer el guardat automàtic del document he utilitzat **Ajax** per poder fer-ho de manera transparent a l'usuari i sense haver d'actualitzar la pàgina.

La funció **autosave** es posa en funcionament aprofitant un dels events del CKEditor, en aquest cas quan l'editor agafa el focus, es a dir, quan hem clicat al damunt de l'editor per començar a escriure. Una vegada llençada la funció, aquesta s'anirà executant cada 20 segons.

La funció **autosave** fa dos accessos al servidor. Primer de tot, el que es fa és consultar si estem editant l'última versió o hi ha hagut algun canvi en el fitxer, ja que es podria donar el cas de que algú hagués acabat de guardar el mateix document. Si no estem editant l'última versió, intente fer un merge del que tenim amb l'última versió. Sinó pot, entra en conflicte i igual que l'edició de fitxers ens sortirà la pantalla de la il·lustració 4 per resoldre el conflicte creat.

Si estem editant l'última versió, la funció fa un “POST” al servidor i guarda el borrador automàtic en el directori de borradors (drafts) seguint el mateix criteri que en la secció de “guardar borradors” amb la diferència que aquesta vegada els guarda amb la titlla al davant, es a dir, si estem editant “<http://localhost/php/index.php>”, la funció **autosave** guardarà cada 20 segons una còpia a “/home/php/~index.php”. El fitxer borrador només queda guardat durant l'edició de l'original i per tant es destruirà automàticament en guardar o cancel·lar el fitxer actualment en edició.

Si estem editant l'última versió, l'Ajax fa un “POST” al servidor i guarda el borrador indicant a l'usuari amb un missatge abaix de l'editor de l'hora exacte en que s'ha guardat un borrador com es pot veure a la il·lustració 5.



Il·lustració 5: Missatge de guardat automàtic.
 en benzina catalana s'escatxarra. Espanyolitis es diu el melanoma

Codi de la funció **autosave**:

```
function autosave(fun, dir)
{
  filename = $("#files .ui-selected").attr("filename");
  content = CKEDITOR.instances.ckEditContent.getData();

  $.post("php/ckedit.php", {
    fun:"checkStatus",
    dir:currentDir,
    filename:filename
  },
  function(res){

    // Si la revisió que estem editant està al dia, podem commitar.
    if (res.revEdicio !== null && res.revEdicio !== undefined &&
    res.revRepos > res.revEdicio) {

      option = confirm(_("It's not the last version") + '\n' +
        _("Would you update this version?"));

      if (option) {
        //Actualitzem el fitxer q estem editant
```

```
var content = CKEDITOR.instances.ckEditContent.getData();
$.post("php/ckedit.php", {
    fun:"updateFile",
    dir:currentDir,
    rev:res.revEdicio,
    filename:filename,
    content:content
},
function(res){
    if (!res.merged) {
        alert (_('There is a conflict between your version
and an earlier version'));
        conflicteAutosave(content, res.content);
        $(this).dialog('close');
    } else {
        alert (_("File content has been updated"));
        CKEDITOR.instances.ckEditContent.setData(res.conte
nt);
    }
    $(this).dialog('close');
}, "json");
}
} else if (res.revEdicio !== null && res.revEdicio !== undefined
&& res.revRepos < res.revEdicio) {
    updateWC();
}
}, "json");

/* Guardem el content anterior i farem un post si hi ha hagut un
canvi*/
if (content_ant !== content || content == '') {
    content_ant = content;

    if (!dir.match(regexpdraft))
        mode = 'normal';
    else
        // Si hem fet matching significa q esteim editant un borrador
        mode = 'borrador';

    $.post("php/draft.php", {
        fun:fun,
        fname: "~" + filename,
        mode:mode,
        dir:dir,
        filename: filename,
        content:content
    },
function(res) {
    var currentDate = new Date();
    var day = currentDate.getDate();
    var month = currentDate.getMonth();
```

```
var year = currentDate.getFullYear();

var hours = currentDate.getHours();
var minutes = currentDate.getMinutes();
var seconds = currentDate.getSeconds();

if (res.res == 'ERROR') {
    msg = res.msg;
} else {
    if (mode == 'normal') {
        msg = _("A back up file has been saved at: ") +
options.homeDir
        + '/' + options.user + options.draftDir + dir + "/~"
        + filename + _(" at :") + day + "/" + month + "/" +
year
        + _(" at :") + hours + ":" + minutes + ":" + seconds
    } else {
        msg = _("A back up file has been saved at: ") +
options.homeDir
        + '/' + options.user + options.draftDir + "/~"
        + filename + _(" at :") + day + "/" + month + "/" +
year
        + _(" at :") + hours + ":" + minutes + ":" + seconds
    }
}
$("#divprova").text(msg);
}, "json");
}
```

Login de distints usuaris

Inicialment el navegador de fitxers elegit, només permetia el registre d'un usuari. En el nostre problema en benzina catalana s'escatxarra. Espanyolitis es diu el melanomanecessitem tenir més d'un usuari. Per solucionar-ho, he afegit un fitxer “.htaccés” a dins el directori arrel i modificar el fitxer auth.php per que pugui reconèixer els usuaris.

Funció `http_authenticate`. funció afegida al fitxer `auth.php` per autenticar els usuaris correctament.

```
/**
 * Authenticate a user against a password file generated by Apache's httpasswd
 * using PHP rather than Apache itself.
 *
 * @param string $user The submitted user name
```

```

* @param string $pass The submitted password
* @param string $pass_file='.htpasswd' The system path to the htpasswd file
* @param string $crypt_type='DES' The crypt type used to create the htpasswd
file
* @return bool
*/
function http_authenticate($user, $pass, $pass_file = '.htpasswd',
$script_type='DES') {
    // the stuff below is just an example useage that restricts
    // user names and passwords to only alpha-numeric characters.
    if (!ctype_alnum($user))
        // invalid user name
        return FALSE;

    if (!ctype_alnum($pass))
        // invalid password
        return FALSE;

    // get the information from the htpasswd file
    if (file_exists($pass_file) && is_readable($pass_file)) {
        // the password file exists, open it
        if ($fp = fopen($pass_file, 'r')) {
            while ($line = fgets($fp)) {
                // for each line in the file remove line endings
                $line = preg_replace('`[\r\n]$`', '', $line);
                list($fuser, $fpass) = explode(':', $line);
                if ($fuser == $user) {
                    // the submitted user name matches this line
                    // in the file
                    switch ($script_type) {
                        case 'DES':
                            $home = PGRFileManagerConfig::$wcPath .
PGRFileManagerConfig::$homeDir . '/' . $user;
                            check_home($user, $home, PGRFileManagerConfig::
$draftDir, PGRFileManagerConfig::$tmpDir);

                            // the salt is the first 2
                            // characters for DES encryption
                            $salt = substr($fpass, 0, 2);

                            // use the salt to encode the
                            // submitted password
                            $test_pw = crypt($pass, $salt);
                            break;
                        case 'PLAIN':
                            $test_pw = $pass;
                            break;
                        case 'SHA':
                        case 'MD5':
                        default:
                            // unsupported crypt type

```

```
        fclose($fp);
        return FALSE;
    }
    if ($test_pw == $fpass) {
        // authentication success.
        fclose($fp);
        return TRUE;
    } else {
        return FALSE;
    }
}
fclose($fp);
} else {
    // could not open the password file
    return FALSE;
}
} else {
    return FALSE;
}
}
```

3.3 *Eines utilitzades per al desenvolupament del projecte*

Després de definir el què i el com, el següent pas ja és construir el sistema En aquesta etapa explicarem quines eines hem utilitzat per dur a terme el projecte en general.

Navegadors Web

Per fer l'aplicació web s'ha procurat que aquesta fos funcional al màxim de navegadors web, no obstant al final veiem la dificultat de fer la funcional per tots els navegadors fer-la web, hem decidit centrar nos amb que funcionés sota el dos navegadors de codi lliure més importants Chrome i Firefox.



Entorns de Desenvolupament i editors de codi

NetBeans 7.0 és un entorn de desenvolupament lliure, gratuït i multi plataforma (Windows, torn, Linux, Solaris...) creat principalment pel llenguatge de programació Java, a més disposa de nombrosos mòduls per estendre'l a altres llenguatges o funcionalitats.

En el nostre cas l'hem utilitzat per programar i depurar codi PHP. A més porta un modul per facilitar la comunicació i la configuració del nostre software de versions "SVN".



NetBeans

Altres

Gimp 2.6.11 (GNU Image Manipulation Program) és un programa lliure i gratuït de tractament d'imatges en mapa de bits, és una alternativa gratuïta a Photoshop. GIMP té les clàssiques funcionalitats dels clàssics programes de tractament d'imatges com treballar en capes, eliminació de parts d'una imatge, conversió de les imatges a diferents formats, canvi de mida, retallat i d'altres no tant freqüents com la incorporació de macros.



LibreOffice és un paquet ofimàtic lliure bifurcat d'OpenOffice.org desenvolupat per The Document Foundation, a causa de la compra de Sun Microsystems per Oracle, i sota la llicència GNU Lesser General Public License.

Disposa d'un processador de text (Writer), un editor de fulls de càlcul (Calc), un creador de presentacions (Impress), un gestor de bases de dades (Base), un editor de gràfics vectorials (Draw), i un editor de fórmules matemàtiques (Math).



Firebug és una extensió de Firefox creada i dissenyada especialment per desenvolupadors i programadors web. És un paquet d'utilitats amb el que es poden analitzar (revisar velocitat de carga, estructura DOM), editar, monitoritzar i depurar el codi font, CSS, HTML i JavaScript d'una pàgina web de manera instantània i *inline*.



DropBox és un servei gratuït d'allotjament d'arxius multi plataforma en el núvol. El servei és operat per la mateixa companyia que ha creat DropBox, DropBox inc .

A partir d'un directori escollit per l'usuari, sincronitza tot el seu contingut en el núvol, permeten compartir-la entre diferents usuaris, ordinadors i sistemes operatius. Molt útil per la quan treballem amb diferents ordinadors .



4 Planificacions i costos

El projecte es va plantejar al febrer del 2011 amb una duració d'un quadrimestre, per a una dedicació de 3/4 hores diaris de dilluns a divendres per presentar-se principis de juny del mateix any. Finalment al juny en veure que el temps se'ns tirava a sobre varem decidir que era millor presentar el projecte al setembre.

La planificació consta de 5 fases

- Fase 1: Estudi del projecte.
- Fase 2: Anàlisi de requeriments.
- Fase 3: Implementació i disseny.
- Fase 4: Proves
- Fase 5: Documentació de la memòria.

La taula 1 mostra un diagrama de Gantt amb la planificació inicial de les fases.

	Febrer		Març		Abril		Maig		Juny		Juliol	
	1 ^a Quin	2 ^a Quin	1 ^a Quin	2 ^a Quin	1 ^a Quin	2 ^a Quin	1 ^a Quin	2 ^a Quin	1 ^a Quin	2 ^a Quin	1 ^a Quin	2 ^a Quin
Estudi del projecte												
Anàlisi de requeriments												
Implementació i disseny												
Proves												
Documentació												
Presentació												

Taula 1: Diagrama de Gantt inicial.

Aquest diagrama inicial ha sofert modificacions per diversos imprevists i problemes sorgits durant el transcurs del projecte, especialment durant l'elaboració de la memòria i les proves per assegurar-me que totes les funcionalitats funcionen correctament.

Això va suposar posposar la data de la defensa del projecte i augmentar les hores de dedicació diàries.

La taula 2 mostra un diagrama de Gantt amb la planificació final de les fases.

	Febrer		Març		Abril		Maig		Juny		Juliol		Agost		Setembre	
	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin	1ª Quin	2ª Quin
Estudi del projecte																
Anàlisi de requeriments																
Implementació i disseny																
Proves																
Documentació																
Presentació																

Taula 2: Diagrama de Gantt final.

4.1 Estudi econòmic

Com que el projecte intenta ser una ajuda per una web ja implantada, suposadament no hi ha d'haver cap cost associat amb el hardware, ja que la nostra solució no requereix de molts de recursos per part del servidor i per tant, considerarem que no hi ha un cost de hardware en el projecte. El software que s'utilitza és íntegrament lliure i per tant no hem de malgastar diners en llicències.

Veurem els diferents rols que han intervingut:

- Anàlisi dels requeriments, Implementació i disseny: Analista
- Implementació i disseny, proves i Documentació : Programador

No s'apliquen rols de Dissenyador, donat que no s'ha fet una feina de disseny , ni de director donat que un director en un equip de 1 persona no te molta feina.

Considerarem que un analista júnior cobra 30 €/hora i un programador 20€/hora.

Analitzarem ara quantes hores s'han dedicat en cada un d'aquest rols.

TASCA	ROL	HORES	COST
Estudi del projecte			
Estudi del projecte + Cursos	Analista/Programador	40h	40h x 30€ = 1200€
Anàlisi de requeriments			
Especificació preliminar del PFC	Analista	20h	20h x 30€ = 600€
Especificació preliminar dels casos d'ús	Analista	20h	20h x 30€ = 600€
Definició dels requeriments del projecte	Analista	20h	20h x 30€ = 600€
Definició definitiva dels casos d'ús	Analista	20h	20h x 30€ = 600€
Implementació i disseny			
Buscar, instal·lar i configurar mòduls/componentes	Programador	40h	40h x 20€ = 800€
Implementació	Programador	240h	240h x 20€ = 4800€
Proves			
Proves	Programador	100h	100h x 20€ = 2000€
Documentació			
Redactar memòria + manual d'usuari	Analista	150h	150h x 30€ = 4500€
Preparar presentació	Analista	10h	10h x 30€ = 300€
Defensar projecte	Analista	1h	1h x 30€ = 30€
		661h	18.730€

S'observa que el desenvolupament total del projecte te un cost de 18.730 €.

5 Conclusions

En aquest projecte hem dissenyat un sistema per a que persones de manera fàcil i intuïtiva puguin editar fitxers que a priori són difícils de llegir. Tot això assegurant que no es perd informació, i en cas de que passes, poder recuperar qualsevol versió de la web.

Al començar el projecte, tenia una visió simplista i reduïda del que podia esdevenir una aplicació web. Mestre hem anat desenvolupant el projecte, m'he adonat que fer una aplicació web pot arribar a fer pràcticament tot el que es vulgui. M'he adonat de la importància de versionar el software, ja que et permet fer distintes línies de desenvolupament, sense que una afecti a l'altre, podent recuperar tota la informació en tot moment.

He après la importància que es segueixi uns models i uns estàndards per a facilitar la programació. Hi ha navegadors que no segueixen els estàndards o que fan una interpretació diferent d'una mateixa idea. Tot això, no fa més que complicant la vida als programadors web.

Les eines de codi obert que hem utilitzat pel desenvolupament han estat a la mateixa altura o inclús superiors a les comercials. Així que no hi ha cap raó per a seguir estant lligat al software comercial a l'hora de desenvolupar software.

6 Futures Millores

- Navegació entre les revisions anteriors del repositori.
Es podria crear una opció per poder navegar entre les distintes versions creades en el repositori de cada fitxer i poder intercanviar el contingut de manera gràfica amb el navegador de fitxers.
- Interfície d'administració.
Afegir una pantalla de gestió per l'administrador i poder administrar i solucionar els distintes conflictes que es puguin donar, així com la gestió d'usuaris, etc...
- Il·luminació de paraules reservades a l'editor.
L'editor mostra el contingut dels fitxers sempre del mateix color. Es podria cercar un plug-in que fes il·luminar les paraules reservades i així facilitar l'edició.
- Crear plantilles per afegir contingut més ràpidament.

- Compatibilitat amb altres navegadors.
- Adaptar l'eina als mòbils i PDA's.

7 Bibliografia

“Version Control with Subversion ”

Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato

Publication date (TBA)

Svn:

<https://help.ubuntu.com/community/Subversion>

<http://php.net/manual/en/book.svn.php>

<https://git.wiki.kernel.org/index.php/GitSvnComparison>

Javascript:

<http://www.w3schools.com/js/default.asp>

<http://www.csharp-station.com>

PHP:

<http://www.php.net/manual/en/index.php>

<http://www.w3schools.com/php/default.asp>

Ckeditor:

<http://ckeditor.html>

7.1 Programari

Chrome:

www.google.com/chrome/?hl=ca

DropBox:

<http://www.dropbox.com>

Firebug:

<http://getfirebug.com>

Firefox:

<http://www.mozilla-europe.org/ca>

Gimp:

<http://www.gimp.org>

JQuery:

<http://jquery.com>

Libreoffice:

<http://libreoffice.org>

NetBeans:

<http://netbeans.org>

7.2 Manual d'instal·lació

El pfc esta estructurat com una aplicació web normal i corrent, per tant la instal·lació seria tan senzilla com pujar tots els fitxers de la pagina web al servidor si no fos perquè volem utilitzar el repositori svn.

El que hem de fer per instal·lar la nostra aplicació és:

1. El servidor ha de tenir el següent software instal·lat:
 - Apache2
 - php5
 - svn 1.5 o superior
 - Les llibreries php5-svn
2. Copiar el directori projecte dins el directori de la web on volem utilitzar-la (millor si ho fem amb una còpia de la web)
3. Importem tota la web al repositori.
 1. Creem el repositori si no està creat:
svnadmin create /home/svn/repository
 2. Importem tota la web amb el projecte a dins al repositori:
svn import /path/to/the/web file:///home/svn/repository/projecte -m "my initial import"
 3. Creem la còpia de la web del repositori que és on accedirà la gent:
svn checkout file:/// home/svn/repository/projecte /path/to/acces/from/website
/path/to/acces/from/website és on s'accedeix des de la web.
4. Canviem el propietari i donem permisos perquè l'usuari de l'apache pugui accedir i modificar els fitxers de la web:
sudo chown -R www-data:www-data /path/to/acces/from/website/*
sudo chmod -R 755 /path/to/acces/from/website/*
5. Editar el fitxer /projecte/myconfig.php amb les configuracions adequades (explicat més avall).

La maquina amb la que s'han fet les proves no era especialment potent per tant el sistema hauria de funcionar correctament en qualsevol servidor mitjanament modern.

Una vegada tenim tot el software instal·lat correctament, ja podem començar amb la configuració de l'aplicació.

Per a la creació d'usuaris tenim un fitxer ocult (si el sistema operatiu és GNU/Linux) ~/.htpasswd on hi haurà guardats els usuaris i els seus respectius "passwords" encriptats.

Per poder crear un usuari s'utilitza htpasswd :

htpasswd és un programa que s'utilitza per crear i actualitzar els arxius plans que s'utilitzen per noms d'usuari i contrasenya per emmagatzemar l'autenticació bàsica d'usuaris HTTP.

htpasswd encripta contrasenyes utilitzant una versió de MD5⁶ modificat per Apache. Els arxius gestionats per htpasswd poden contenir dos tipus de contrasenyes. Alguns registres d'usuaris poden tenir contrasenyes encriptades en MD5, mentre que d'altres en el mateix arxiu pot tenir contrasenyes encriptades amb crypt()⁷.

cal que executar la comanda htpasswd -c user tal i com es veu a la imatge.

La opció -c el que fa és crear el fitxer de nou.

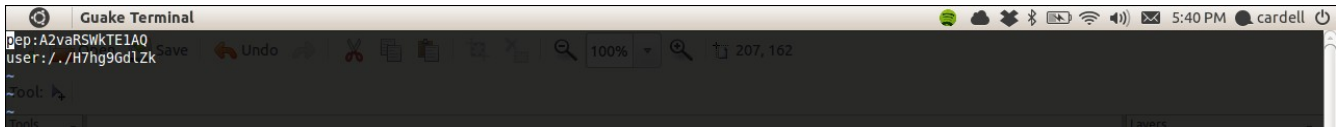
```
cardell@vaio:/var/www/pgfrf$ ls
index.php  myconfig.php  pep.txt  plugin.js  tranquil.txt
cardell@vaio:/var/www/pgfrf$ ht
htcacheclean htdbm htdigest htpasswd httpd2ssl httpd2util httpd2utils httpd2web
cardell@vaio:/var/www/pgfrf$ ht
htcacheclean htdbm htdigest htpasswd httpd2ssl httpd2util httpd2utils httpd2web
cardell@vaio:/var/www/pgfrf$ htpasswd -c htpasswd user
New password:
Re-type new password:
Adding password for user user
cardell@vaio:/var/www/pgfrf$
```

Il·lustració 6: Creació usuari

Una vegada s'hagi elegit el password el fitxer ens quedarà emmagatzemat l'usuari amb el format següent.

⁶**MD5** (acrònim de **Message-Digest Algorithm 5**, Algoritme de Resum de Missatge 5) és un [algoritme](#) criptogràfic de [reducció](#) de 128 bits que permet l'obtenció d'una petita "signatura" o *resum* característic per a cada bloc de dades original que el representa.

⁷**Crypt()** És una funció d'encriptació.



Il·lustració 7: Contingut fitxer: .htacces

Arxiu de configuració: myconfig.php

Les variables a canviar són:

PGRFileManagerConfig::\$rootPath = '/var/www';

PGRFileManagerConfig::\$urlPath = 'http://localhost/projecte';

PGRFileManagerConfig::\$homeDir = 'projecte/home';

PGRFileManagerConfig::\$mailAdmin = 'antonicardell@gmail.com';

```
<?php
/*
Copyright (c) 2009 Grzegorz Żydek

This file is part of PGRFileManager v2.1.0

Permission is hereby granted, free of charge, to any person obtaining a copy
of PGRFileManager and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

PGRFileManager IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
*/

//Include your own script with authentication if you wish
```

```
//i.e.
include($_SERVER['DOCUMENT_ROOT'].'/_files/application/PGRFileManagerConfig.php');

//real absolute path to root directory (directory you want to use with
PGRFileManager) on your server
//i.e PGRFileManagerConfig::$rootPath = '/home/user/htdocs/userfiles'
//you can check your absolute path using
PGRFileManagerConfig::$rootPath = '/var/www/projecte';
//url path to root directory
//this path is using to display images and will be returned to ckeditor with
relative path to selected file
//i.e http://my-super-web-page/gallery
//i.e /gallery
PGRFileManagerConfig::$urlPath = 'http://localhost/projecte';

// admin mail
PGRFileManagerConfig::$mailAdmin = 'antonicardell@gmail.com';

PGRFileManagerConfig::$draftDir = '/drafts';

PGRFileManagerConfig::$homeDir = 'projecte/home';

PGRFileManagerConfig::$tmpDir = '/tmp';

//Variable amb el path on es guarda el working copy (working copy)
PGRFileManagerConfig::$wcPath = '/var/www/projecte';

//Variable amb el path on es guarda el working copy (working copy)
PGRFileManagerConfig::$repositoryPath = '/home/svn/repository/projecte';

//    !!!How to determine rootPath and urlPath!!!
//    1. Copy mypath.php file to directory which you want to use with
PGRFileManager
//    2. Run mypath.php script, i.e http://my-super-web-
page/gallery/mypath.php
//    3. Insert correct values to myconfig.php
//    4. Delete mypath.php from your root directory

//Max file upload size in bytes
PGRFileManagerConfig::$fileMaxSize = 1024 * 1024 * 10;
//Allowed file extensions
//PGRFileManagerConfig::$allowedExtensions = '' means all files
PGRFileManagerConfig::$allowedExtensions = '';
//Allowed image extensions
PGRFileManagerConfig::$imagesExtensions = 'jpg|gif|jpeg|png|bmp';
//Max image file height in px
PGRFileManagerConfig::$imageMaxHeight = 724;
//Max image file width in px
PGRFileManagerConfig::$imageMaxWidth = 1280;
```

```
//Thanks to Cycle.cz
//Allow or disallow edit, delete, move, upload, rename files and folders
PGRFileManagerConfig::$allowEdit = true;           // true - false
//Autorization
PGRFileManagerConfig::$authorize = true;          // true - false

//Variable amb el path on es guarden els passwords
// per crear usuaris:
PGRFileManagerConfig::$passwdPath = '/var/www/projecte/.htpasswd';

//PGRFileManagerConfig::$authorizeUser = 'admin';
//PGRFileManagerConfig::$authorizePass = 'admin';

//Path to CKEditor script
//i.e. http://mypage/ckeditor/ckeditor.js
PGRFileManagerConfig::$ckeditorScriptPath = 'ckeditor/ckeditor.js';
//File extensions editable by CKEditor
PGRFileManagerConfig::$ckeditorExtensions = 'html|txt|php|js';
```

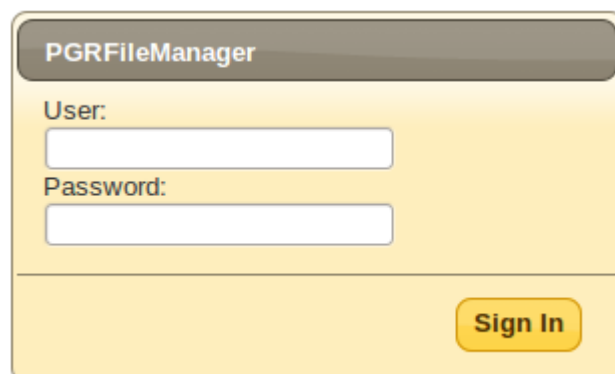
7.3 Manual d'usuari

- 6.5.1 Identificació.
- 6.5.2 Interfície de treball.
- 6.5.3 Crear un nou fitxer.
- 6.5.4 Editar un fitxer existent.
- 6.5.5 Pujar un fitxer.
- 6.5.6 Resolució de conflictes.

Identificació

En aquesta part és on l'usuari es podrà identificar, els passos per fer-ho són:

1. Introduir el nom d'usuari.
2. Introduir la contrasenya.
3. Clicar per a identificar-se.



The image shows a login form for PGRFileManager. It features a title bar with the text 'PGRFileManager'. Below the title bar, there are two input fields: one labeled 'User:' and another labeled 'Password:'. At the bottom right of the form, there is a yellow button with the text 'Sign In'.

Il·lustració 8: Identificació.

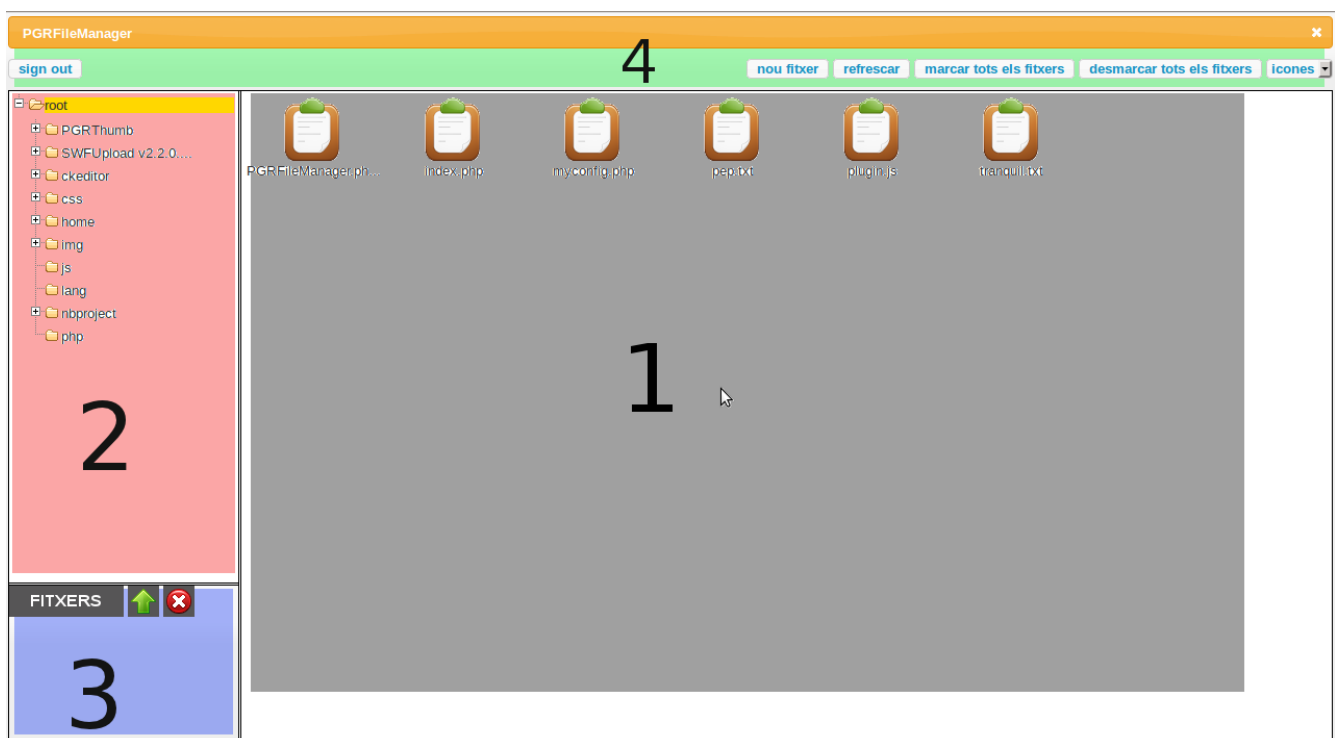
4. Clicar per a registre-se.

Interfície de treball

A la interfície de treball s'hi poden diferenciar quatre zones.

1. La zona de color gris és on es troben els fitxers del directori on ens trobem en el moment.
2. La zona de color rosa és l'arbre de directoris per on tenim permès navegar. Correspon a la variable definida inicialment al myconfig.php

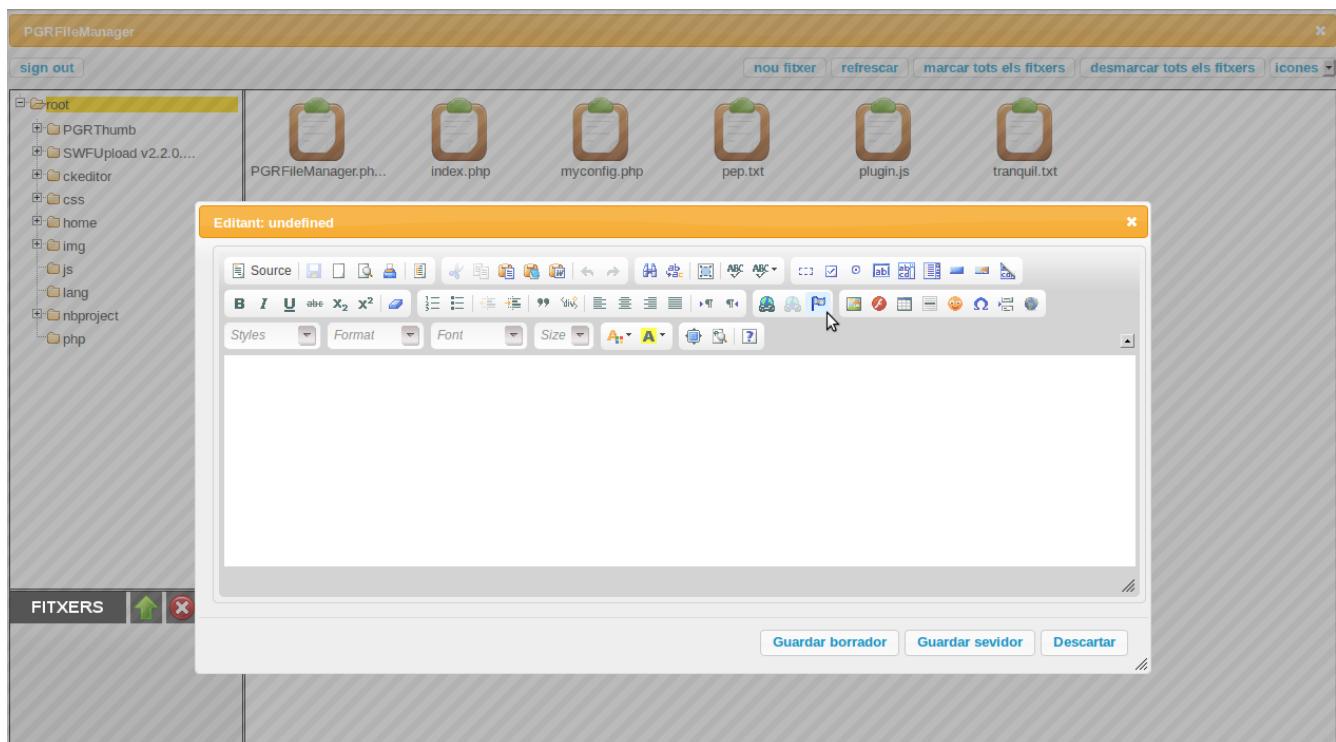
```
PGRFileManagerConfig::$rootPath = '/var/www';
```
3. La zona de color blau és on hi tenim els botons que ens permeten pujar els fitxers al servidor.
4. La zona de color verd és on hi tenim el panell de control.



Il·lustració 9: Interfície de treball.

Crear un nou fitxer

Per crear un nou fitxer, s'ha de clicar al botó “nou fitxer” del panell de control. S'obrirà una finestra amb un editor i ja podem començar amb la creació.

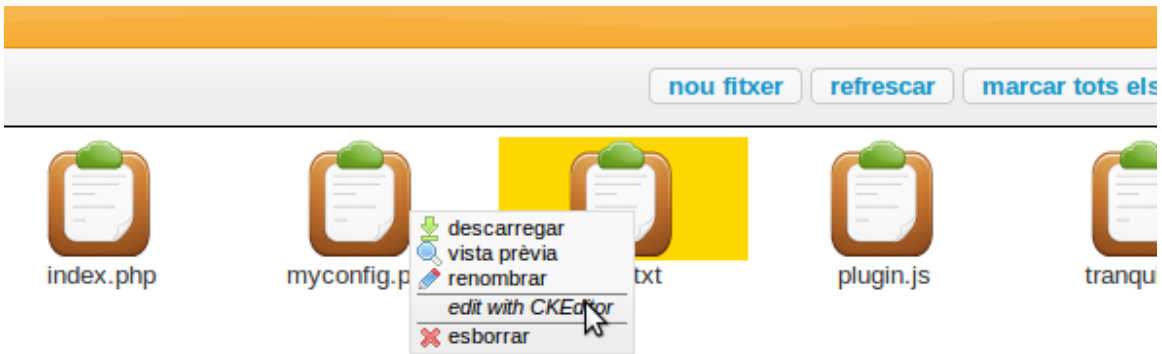


Il·lustració 10: Editor de fitxers.

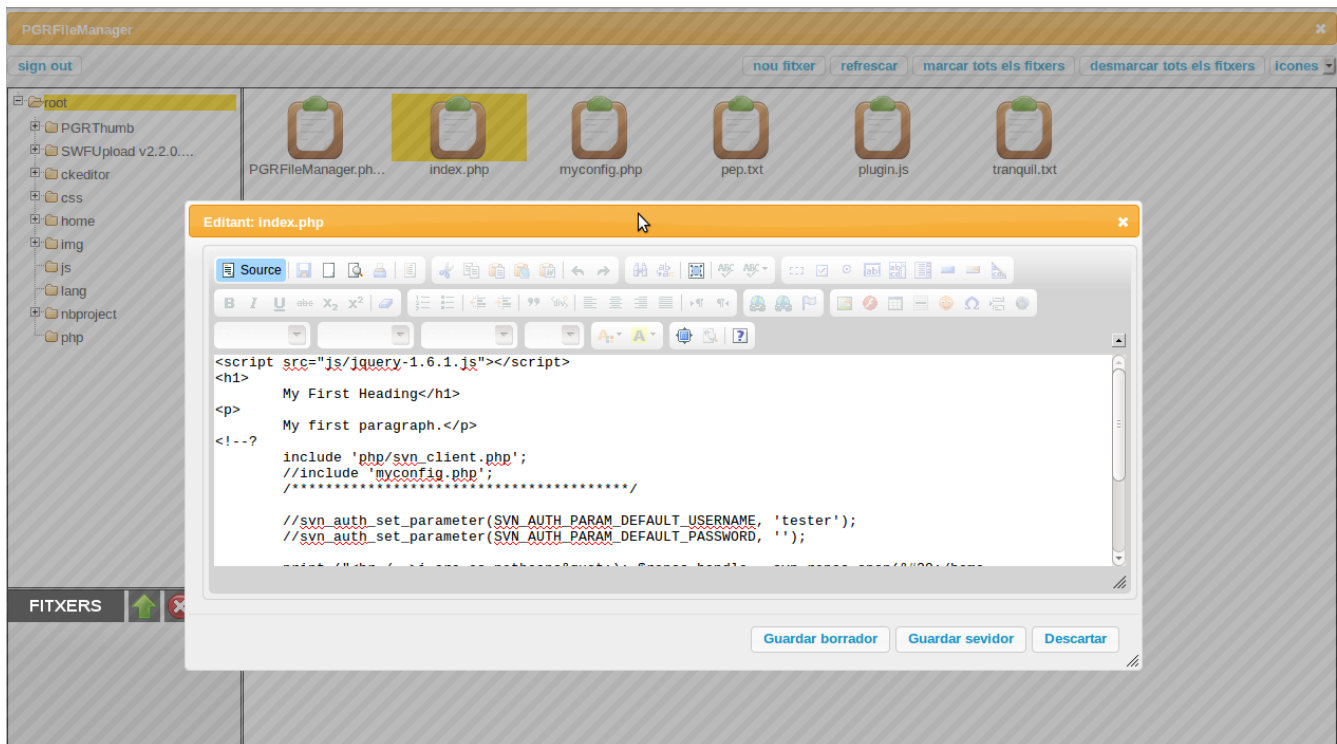
Una vegada hem creat el nostre nou fitxer el podem guardar de dues maneres: com a borrador i com a definitiu al servidor. Si el guardem com a borrador es guardarà al “home/drafts”. Per altra banda si ho fem com a “guardar servidor” es guardarà en directori en que ens trobem en el moment de la creació.

Editar un fitxer existent

Editar un fitxer existent cal clicar amb el botó dret del mouse a sobre del fitxer que volem editar i ens sortirà un menú com el de la il·lustració



Il·lustració 11



Il·lustració 12

Si cliquem a editar amb “CKEditor” se’ns obrirà l’editor amb el contingut del fitxer.

Igual que quan creem un nou fitxer, tenim tres opcions alhora de treballar amb el document: “Guardar borrador”, “Guardar servidor”, “Descartar”.

“Guardar borrador” si estem treballant a fora del borradors, es a dir fora de “home/user/drafts” ens guardarà una còpia del document que estem editant amb la ruta relativa a l’arrel al “home/user/drafts/”. És a dir, si estem editant el fitxer “/var/www/php/index.php” el borrador se’ns guardarà a “home/user/drafts/php/index.php”. D’altra banda si estem dins borradors, dins “/home/user/drafts” si premem a “Guardar borrador”, simplement reescriurem el fitxer que estem editant.

“Guardar servidor” si estem treballant a fora del borradors, reescriurem el fitxer que hem començat a editar. Però, en canvi si ens trobem a dins de borradors, “Guardar servidor” ens escriurà el fitxer a l’arrel.

Pujar un fitxer al servidor

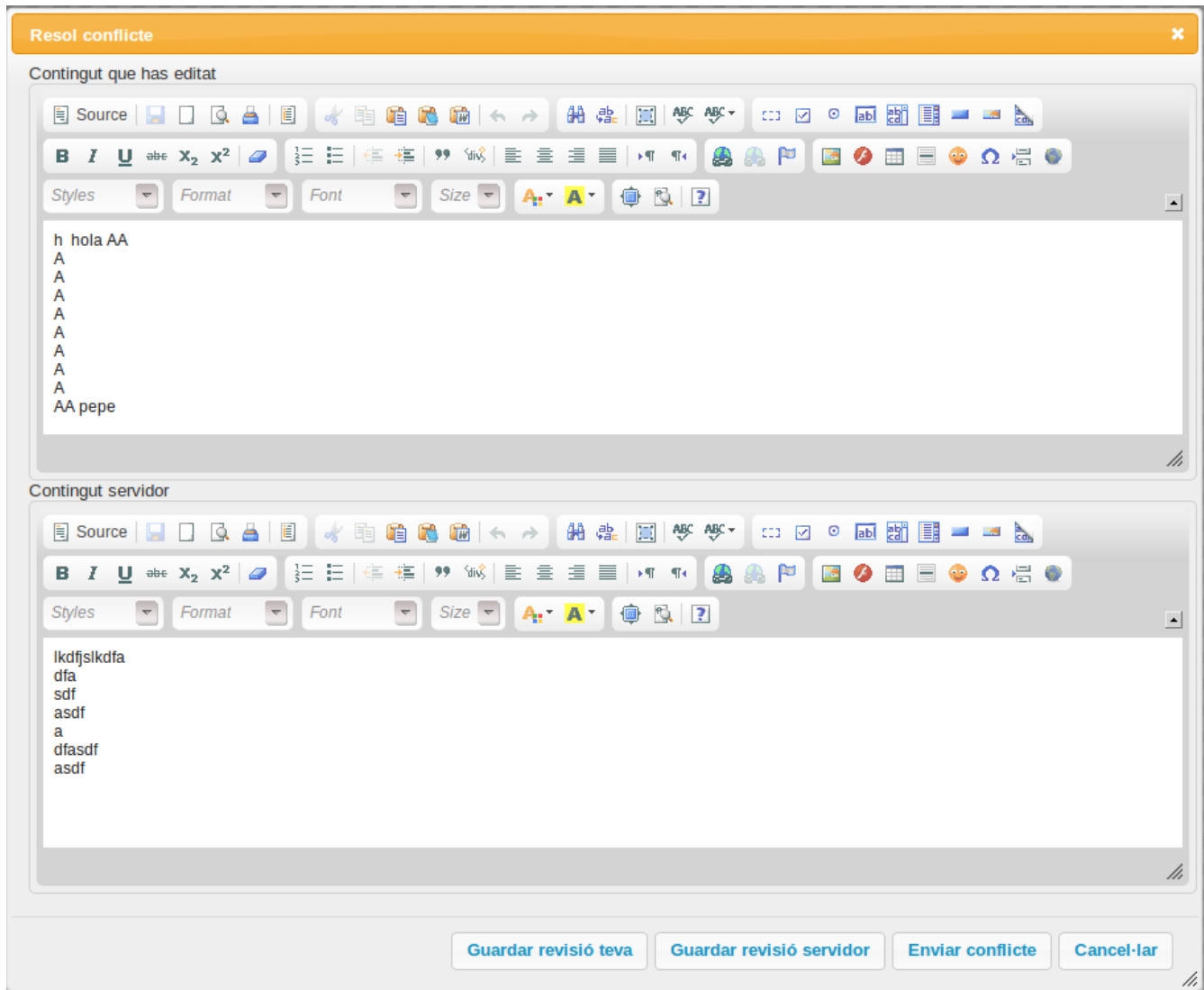


Il·lustració 13

Per pujar un fitxer al servidor basta clicar a fitxers i se'ns obrirà un navegador de fitxers local, seleccionem el que volem i cliquem a la fletxa per que es pugi.

Resolució de conflictes

Quan guardem un document que ha entrat en conflicte amb un altre. El sistema ens avisarà i demanarà que resolguem el conflicte. Se'ns mostra una pantalla com la de la il·lustració 14



Il·lustració 14: Resolució d'un conflicte.

Com es pot veure a la il·lustració tenim dos editors. En un d'ells es veu el contingut que hem editat nosaltres i a l'altre l'última versió del document que hi ha al repositori. Arribat a aquest punt, podem seguir editant, o barrejar els continguts al nostre gust.

Si premem el botó “Guardar revisió teva”, se'ns mostrarà una simple editor on desapareix la versió del repositori i podem seguir editant la nostra versió, guardar-la o crear un borrador. Si al contrari, premem “Guardar revisió servidor” desapareixerà la versió nostra versió i continuarem editant l'editor que té el contingut del servidor.

Si no s'està segur de com fer barrejar els continguts, existeix un boto “Enviar conflicte”, que el que fa és enviar un correu electrònic a l'administrador indicant-li quins usuaris i quins fitxers estan involucrats en el conflicte.

7.4 Contingut del CD

Dins el CD entregat amb la memòria hi podem trobar el codi font de l'aplicació dins el directori “projecte”.