

Programari lliure

Juan José Amor Iglesias
Israel Herraiz Tabernero
Gregorio Robles Martínez

90.804_a



Desenvolupament de projectes de programari lliure

Amb el suport de:



Generalitat de Catalunya
Departament d'Universitats, Recerca
i Societat de la Informació

David Megías Jiménez

Coordinador

Enginyer en Informàtica per la UAB.
Magíster en Tècniques Avançades
d'Automatització de Processos
per la UAB.

Doctor en Informàtica per la UAB.

Professor dels Estudis d'Informàtica i
Multimèdia de la UOC.

Jordi Mas Hernández

Coordinador

Enginyer de programari en l'empresa
de codi obert Ximian, on treballa en la
implementació del projecte lliure
Mono. Com a voluntari, col·labora en
el desenvolupament del processador
de textos Abiword i en l'enginyeria
de les versions en català del projecte
Mozilla i Gnome. També és
coordinador general de Softcatalà.
Com a consultor, ha treballat per a
empreses com Menta, Telépolis,
Vodafone, Lotus, eresMas, Amena i
Terra España.

Juan José Amor Iglesias

Autor

Llicenciat en Informàtica per la
Universitat Politècnica de Madrid.
Fundador de LuCAS (actualment
TLDP-ES). Fundador d'Hispalinux.
En l'actualitat cursa els estudis de
doctorat a la Universitat Rey Juan
Carlos.

Israel Herraiz Tabernero

Autor

Enginyer industrial per la Universitat
de Cadis. En l'actualitat, cursa els
estudis de doctorat d'Informàtica
i Modelització Matemàtica a la
Universitat Rey Juan Carlos.

Gregorio Robles Martínez

Autor

Enginyer de Telecomunicacions per la
Universitat Politècnica de Madrid.
Projecte final de carrera en la TU
Berlín. Professor ajudant a la
Universitat Rey Juan Carlos.
Actualment, està acabant el doctorat.
Consultor del màster internacional de
Programari Lliure a la UOC.

La Fundació per a la Universitat Oberta de Catalunya agraeix el suport del Departament d'Universitats,
Recerca i Societat de la Informació de la Generalitat de Catalunya per a la versió d'aquesta obra en català.

Primera edició: febrer 2006

© Fundació per a la Universitat Oberta de Catalunya

Av. Tibidabo, 39-43, 08035 Barcelona

Material realitzat per Eureka Media, SL

© Autors: Juan José Amor Iglesias, Israel Herraiz Tabernero, Gregorio Robles Martínez

Es garanteix permís per a copiar, distribuir i modificar aquest document segons els termes de la *GNU Free Documentation License*,
Versió 1.2 o qualsevol de posterior publicada per la Free Software Foundation, sense seccions invariants ni texts de coberta anterior
o posterior. Es disposa d'una còpia de la llicència en l'apartat "GNU Free Documentation License" d'aquest document.

Índex

Agraïments	7
Pròleg	9
Materials previs	9
Introducció	11
Objectius	12
Coneixements previs	13
Requisits tècnics	14
Continguts d'aquest material	14
1. Eines de gestió de projectes	17
1.1. Serveis útils per a projectes de programari lliure ..	18
1.2. Llocs de desenvolupament	19
1.2.1. Software-libre.org	20
1.2.2. Savannah	20
1.2.3. Alioth	21
1.2.4. BerliOS	21
1.2.5. SourceForge	21
1.3. Registre del projecte	22
1.4. El compte de línia d'ordres	27
1.4.1. Accés mitjançant SSH sense clau	28
1.4.2. Com es poden copiar els fitxers del web	29
1.5. Configuració del CVS	30
1.5.1. Accés anònim	30
1.5.2. Accés com a desenvolupador	31
1.6. Baixades del projecte	32
1.7. Llistes de correu	36
1.8. Tracker i el sistema de seguiment d'errors	39
2. Control de versions	45
2.1. Instal·lació i configuració inicial de CVS	47
2.1.1. Creació del dipòsit	47
2.1.2. Preparació del accés anònim	48
2.1.3. Obertura de comptes per als desenvolupadors	49

2.2. Operativa bàsica del CVS	49
2.2.1. Accés anònim a un dipòsit	49
2.2.2. Accés al CVS pel desenvolupador	50
2.2.3. Creació d'un projecte al dipòsit	50
2.2.4. Obtenció del projecte	51
2.2.5. Creació de fitxers i directoris	51
2.2.6. Modificació dels fitxers. Fusió de canvis	52
2.2.7. Eliminació de fitxers	54
2.3. Operativa avançada al CVS	54
2.3.1. Versions	55
2.3.2. Etiquetes	56
2.3.3. Branques	57
2.3.4. Informació sobre etiquetes i branques	59
2.4. Subversion: la pròxima generació	60
3. Sistemes de seguiment d'errors	63
3.1. Seguiment d'errors amb Bugzilla	64
3.1.1. Els errors en Bugzilla	64
3.2. Instal·lació i configuració de Bugzilla	65
3.2.1. Instal·lació de Bugzilla	65
3.2.2. Configuració de Bugzilla	66
3.3. Notificació d'errors	69
3.3.1. Creació de comptes	69
3.3.2. Notificació d'un error	70
3.4. Cerca i tractament d'errors	72
3.4.1. Cerca d'errors	72
3.4.2. Estats d'un error	73
3.4.3. Edició d'un error	75
4. Llistes de correu electrònic	77
4.1. Què és una llista de correu	77
4.2. Eines	78
4.3. Alternatives	79
4.4. Llistes de correu amb Mailman	80
4.4.1. Instal·lació	80
4.4.2. Configuració de la llista de gestió	83
4.4.3. Operativa d'usuari de llistas	84
4.4.4. Operativa d'administració de llistas	91
5. La gestió d'un projecte	97
5.1. Elecció de la llicència	97
5.2. El web del projecte	98
5.3. Estructura del web del projecte	100

5.4. Instal·lació senzilla	101
5.5. Consells per a la distribució i difusió del programari	102
5.6. A la recerca de l'efecte en xarxa	104
5.7. Codi i internacionalització	106
5.8. Esforç dedicat a tasques de gestió	107
5.9. Conclusions	110
Bibliografia	111
Appendix A. GNU Free Documentation License	113

Agraïments

Els autors agraeixen a la Fundació per a la Universitat Oberta de Catalunya (<http://www.uoc.edu>) el finançament de la primera edició d'aquesta obra, emmarcada en el màster internacional de Programari Lliure ofert per aquesta institució.

Pròleg

Materials previs

Alguns textos d'aquest curs estan basats en materials previs (utilitzats amb permís, quan no han estat completament reelaborats), entre els quals podem esmentar els següents (a risc d'oblidar-ne algun d'important):

- La part dedicada a l'alliberament de projectes de programari lliure ha estat elaborada a partir de les transparències i presentació que fa Jesús M. González Barahona, professor de la Universitat Rey Juan Carlos, en el curs de doctorat Programari Lliure que té lloc conjuntament a les universitats Politècnica de Madrid i Rey Juan Carlos.
- Les eines que es presenten també s'han pres de les transparències i les classes que imparteix Joaquín Seoane, professor de la Universitat Politècnica de Madrid, en el curs de doctorat esmentat.
- Alguns gràfics del capítol dedicat al sistema de control de versions (CVS) han estat presos del llibre *Open Source Development with CVS*, de Karl Fogel, que es publica sota una llicència lliure.
- La traducció de la llicència de documentació lliure de GNU és una actualització adaptada de la realitzada per Igor Támara i Pablo Reyes per a la versió 1.1, als quals agraïm la seva confecció i el seu permís per a modificar-la.

Introducció

El programari lliure s'ha convertit en els últims anys en un fenomen imparabile. Certament, el desafiament que representa va més enllà d'elements merament tècnics, ja que es diferencia del programari "tradicional" propietari en aspectes més fonamentals, que involucren des de raons filosòfiques fins a noves pautes econòmiques i de mercat. En aquest material abordarem exclusivament qüestions tècniques, que inclouen des de l'entorn en què es crea el programari lliure fins a la manera d'aconseguir que els nostres projectes de programari lliure aprofitin aquest fenomen per a tenir èxit.

Malgrat aquest èxit incipient del programari lliure encara són poc freqüents els documents i materials que expliquin de manera detallada què s'ha de fer perquè el nostre projecte es beneficiï de tots aquests aspectes que semblen estar-hi associats. En qualsevol cas, podem afirmar, amb tota certesa, que aquest coneixement encara no ha arribat ni als cursos universitaris o d'altres centres docents, ni als grans llibres d'enginyeria del programari.

Dos aspectes fonamentals caracteritzen el desenvolupament de programari lliure:

- a) Es tracta d'un desenvolupament en xarxa. És a dir, està orientat a unes pràctiques que possibilitin la creació de programari a persones que estan disperses, però que poden comunicar-se mitjançant Internet.
- b) Està molt orientat a les eines que el suporten. Això vol dir que hi sol haver una eina específica per a cada procés que s'ha convertit en una espècie d'estàndard *de facto*. Les eines utilitzades avui tenen el seu sentit històric i és possible que amb el pas del temps evolucionin amb el mateix programari lliure. Com el lector haurà pogut imaginar, es tracta d'eines que tenen un component molt arrelat a Internet, precisament perquè aquesta és la primera característica del desenvolupament de programari lliure.

Aprendre com és el desenvolupament de programari lliure consisteix en gran part a assimilar i saber utilitzar les eines que el sustenten. Per aquest motiu, gran part d'aquest material se centra en aquestes eines i mostra amb detall el seu ús.

Finalment, es pot assenyalar que l'èxit d'un projecte de programari –tant si és de programari lliure amb els mètodes que s'expliquen en aquest material com de programari propietari utilitzant qualsevol dels processos de desenvolupament actuals– depèn de nombrosos factors i no està assegurat en cap cas. Suposem que aquest aspecte no és nou per al lector, ja que, per desgràcia, l'enginyeria del programari encara està en una fase que permet indicar el camí per on cal anar, però no assegurar que s'arribi a la meta.

El lector trobarà aquí uns coneixements que han estat adquirits mitjançant l'observació i l'experiència de molts projectes de programari lliure i d'estudis elaborats fins ara. Aquests estudis i el dia a dia dels projectes de programari lliure i dels problemes a què el lector s'enfronta en el seu afany per tirar endavant un projecte de programari lliure seran, sens dubte, la millor font de saviesa que el programador podrà aprofitar. Si, finalment, el projecte de programari lliure té èxit, sempre podrà dir que ha estat gràcies als seus grans dots tècnics i de gestió de recursos (humans i tècnics). Si, al contrari, fracassa, podrà donar la culpa al coneixement escàs que es té d'aquest camp en l'actualitat i, en especial, a aquest text i als seus autors.

Objectius

Aquest material pretén ser una guia per als qui vulguin crear un projecte de programari lliure i es vulguin beneficiar dels aspectes que es destaquen en la literatura quan es parla del fenomen del programari lliure. Entre aquests aspectes, es podrien esmentar la incorporació d'altres programadors que ajudin al desenvolupament de l'aplicació, una alta realimentació per part dels usuaris o la creació d'una comunitat entorn de l'aplicació que no solament la utilitzi, sinó que dugui a terme tasques de promoció, documentació o traducció a altres idiomes.

Una vegada acabat el curs, el lector estarà completament familiaritzat amb les eines que s'utilitzen tant per al desenvolupament de col·laboració en el món del programari lliure com per a la intercomunicació dels diferents agents que hi participen. Amb aquests coneixements, estarà en disposició de crear i llançar un programari propi o entrar a formar part d'algun dels molts projectes de programari lliure que hi ha.

Coneixements previs

Per a l'aprofitament complet dels continguts que s'exposen en aquests materials, el lector haurà de tenir uns coneixements previs referents a la filosofia del programari lliure, alguns coneixements bàsics del seu procés de desenvolupament, les diferents llicències que hi ha i els projectes de programari lliure més importants avui dia. Gran part d'aquests coneixements, si no tots, es poden trobar a *Introducció al programari lliure*, el llibre de text que la Fundació Universitat Oberta de Catalunya (Fundació UOC) ha editat per al màster de Programari Lliure que ofereix i que està a disposició pública a Internet.

En l'àmbit tècnic, el lector ha d'estar familiaritzat amb un entorn GNU/Linux i amb algunes eines bàsiques com són el navegador web, el client de correu i la línia d'ordres (*shell*). En aquest sentit, es recomana el material *Introducció a GNU/Linux* del màster de Programari Lliure ofert per la Fundació UOC.

És important destacar que en aquest curs no s'ensenyarà a programar. Es dona per fet que el lector té les nocions de programació necessàries per a crear programari. Els aspectes en què s'incidirà tenen més a veure amb com es pot aconseguir que aquest programari entri en una dinàmica d'arribar a desenvolupadors i usuaris nous, i creï entorn seu una comunitat que finalment desenvolupi el programari de manera col·laborativa.

Segurament el lector es deu haver adonat que ser un bon programador és un requisit necessari, però no n'hi ha prou perquè el projecte de programari lliure tingui èxit. Són altres aspectes, més relacionats

amb la gestió o la comunicació, els que diferencien els projectes d'èxit dels que no aconsegueixen triomfar. I són aquests aspectes els que estudiarem aquí, els quals, fins i tot tenint poc a veure amb l'art de la programació, sens dubte es poden considerar coneixements tècnics, com podreu observar de seguida.

D'altra banda, també serà de gran ajuda un bon coneixement de la llengua anglesa, ja que molts dels entorns web i eines que s'utilitzaran estan en aquest idioma.

Requisits tècnics

Per al seguiment d'aquest material, el lector ha de tenir un sistema GNU/Linux instal·lat i configurat. Es recomana l'ús de la distribució Debian (en especial, l'última versió estable) perquè és el que s'ha utilitzat en la confecció d'aquests materials, però qualsevol distribució mitjanament actual que s'utilitzi no hauria de donar problemes. Si es vol, també es podran utilitzar altres sistemes UNIX (per exemple, de la família BSD o Solaris) i fins i tot Windows, encara que és probable que les instruccions i les eines canviïn amb l'entorn i no es corresponguin amb les que es presenten aquí (sobretot en el cas de Windows).

En qualsevol cas, haurà de tenir la possibilitat d'instal·lar paquets nous a la màquina i una (bona) connexió a Internet.

Continguts d'aquest material

A continuació es repassen breument els continguts que es tractaran en aquest material.

En el capítol 1 es presentaran les eines que s'utilitzen de manera gairebé universal en el desenvolupament de projectes de programari lliure. Com que hi ha llocs que les ofereixen gratuïtament per a projectes amb llicències lliures, es descriurà breument l'aspecte que tenen aquests llocs i com s'utilitzen. En aquest capítol es presentaran breument les eines més típiques, encara que es deixarà el seu estudi

detallat per a capítols següents. En particular, el lloc que hem triat per a aquesta tasca és el conegut lloc de desenvolupament SourceForge, encara que s'indicaran altres llocs de característiques semblants perquè el lector pugui decidir quin és el que més li convé.

El capítol 2 mostra en profunditat el sistema de control de versions més utilitzat en el món del programari lliure, el CVS (*concurrent versions system*). S'enfocarà des de dos punts de vista: com a administrador d'un dipòsit CVS o com a simple usuari. Des del punt de vista d'administrador, s'ensenyarà com s'instal·len, configuren i gestionen els comptes d'usuaris. Quant al punt de vista dels usuaris, s'incidirà en el seu mode de funcionament bàsic i ens endinsarem en alguns elements avançats.

El capítol 3 tracta els sistemes de gestió d'errors, en particular el sistema més difós i utilitzat en grans projectes de programari lliure: Bugzilla. Igual com amb el sistema de control de versions, també mostrarem com s'instal·la i es configura Bugzilla, per si el lector pretén fer d'administrador. Per als que tinguin com a objectiu simplement l'ús efectiu de Bugzilla, n'hi haurà prou que sàpiguen com es busquen, es notifiquen i es gestionen els errors en aquest sistema.

El capítol 4 està dedicat als gestors de llistes de correu, en especial a GNU Mailman. Aquest sistema també s'examinarà des de dos punts de vista: el de l'usuari de les llistes (per exemple, com es pot donar d'alta i de baixa o com configurar les opcions de lliurament de missatges) i el de l'administrador, que en aquest cas cobreix la instal·lació i configuració de Mailman i la gestió de llistes de correu.

Finalment, arribem a un apartat que és independent de les eines que utilitzem per al nostre projecte de programari lliure. El capítol 5 pretén ser una guia dels passos que cal seguir per a aconseguir crear una comunitat entorn del nostre projecte. Les pràctiques i les bones maneres que es proposen en aquest capítol són lliçons apreses dels projectes de programari lliure que han tingut èxit. És probable que aquest apartat sigui el més profitós per al lector a llarg termini, ja que l'aprenentatge de les eines sol ser molt més senzill. Però, sens dubte, en la temàtica que es tracta en aquest últim capítol hi ha la verdadera essència del desenvolupament de programari lliure.

1. Eines de gestió de projectes

El desenvolupament de programari lliure està fortament arrelat al desenvolupament d'Internet i a una sèrie d'eines de suport. De la mateixa manera que la tasca d'implementació se sol concebre com una activitat individual, llevat d'excepcions com la programació extrema, hi ha moltes tasques que requereixen la interacció de diverses persones i grups de treball.

Exemple

Així com la generació de documentació ha estat un dels cavalls de batalla en el cicle de desenvolupament de programari tradicional a la recerca d'una millora en la qualitat del programari, la pràctica de la documentació en el món del programari lliure no ha estat mai gaire estesa, en part per la naturalesa voluntària dels desenvolupadors, que eviten les tasques tedioses i menys divertides. Tanmateix, hi ha mecanismes d'intercanvi d'informació, necessari per al desenvolupament de programari amb altres desenvolupadors i amb els usuaris a diferents nivells que van des del mateix codi font fins a les llistes de correu electrònic.

Per a cada tipus de comunicació hi sol haver una eina específica que amb el temps s'ha convertit en l'estàndard *de facto*. Una vegada conegudes les tasques més comunes, van començar a aparèixer portals que les integren dins d'una mateixa interfície i que permetien que els desenvolupadors es recolzessin en les eines que se'ls oferien i s'oblidessin de la gestió.

Evidentment, a l'hora de triar la infraestructura necessària per al projecte, tenim dues alternatives clares:

1. Triar nosaltres mateixos totes les eines necessàries per a cada sistema del projecte (llistes de correu, CVS, sistema de seguiment d'errors, etc.), i dur a terme tota la configuració i administració.

Nota

En aquest material s'exposarà com es duu a terme el desenvolupament de programari amb la infraestructura necessària en els dos casos esmentats, començant pel més fàcil, que és quan ja tenim una plataforma llesta per a ser utilitzada.

2. Emprar algun servei en què totes aquestes eines ja s'hagin muntat, com els que es presentaran en l'apartat següent.

La decisió s'ha de prendre tenint en compte la mida del projecte. Llevat dels projectes realment grans, no està justificat dur a terme el muntatge, la configuració i l'administració de tota la infraestructura. Tanmateix, és difícil establir una frontera clara entre els projectes que requereixen una infraestructura pròpia i els que no la requereixen. Com a norma, podríem dir que si es tracta d'un projecte personal o d'un grup petit de participants, és més indicat utilitzar una plataforma externa per a no haver de completar les tasques d'instal·lació, configuració i manteniment relacionades. Si, al contrari, un projecte o projectes que duen a terme un grup nombrós de membres (una associació, una empresa mitjana o gran, etc.), llavors pot resultar convenient tenir una infraestructura pròpia.

1.1. Serveis útils per a projectes de programari lliure

En el desenvolupament de programari lliure se solen utilitzar les eines següents:

- Allotjament (*hosting*) de webs, amb possibilitat d'usar guions PHP i guions CGI per a fer les webs dinàmiques.
- Arxiu permanent de fitxers, on es pugui baixar el programari.
- Tauler i fòrum de missatges, per a la intercomunicació amb desenvolupadors i usuaris.
- Llistes de correu, per a la intercomunicació amb desenvolupadors i usuaris.
- Organitzador de tasques, per a gestionar els recursos humans del projecte.
- Sistema de seguiment d'errors i de sol·licitud de característiques noves, per a facilitar la notificació d'errors i la realimentació a la comunitat.

- Bases de dades per a l'administració del projecte (per exemple, per a tenir un lloc web basat en un gestor de continguts, on cada usuari té el seu compte).
- Sistema de control de versions, per a poder treballar de manera simultània en el codi; la majoria ofereixen CVS, encara que sembla que pròximament migraran a Subversion (SVN).
- Compte de línia d'ordres (*shell*) al servidor, per a manipular els fitxers relacionats amb el web.
- Administració total basada en una interfície web, que simplifica molt l'administració de tots els elements anteriors.

1.2. Llocs de desenvolupament

Molts llocs ofereixen allotjament gratuït per a projectes de programari lliure que inclouen els serveis que acabem de comentar. En general, aquests llocs estan construïts sobre la base d'una aplicació web anomenada *Alexandria*, que inicialment es va desenvolupar per al lloc SourceForge. Tanmateix, des de fa un parell d'anys, l'empresa que manté SourceForge ha decidit que les noves versions d'*Alexandria* no es publicarien sota una llicència de programari lliure, per la qual cosa va aparèixer el projecte Gforge, que no és més que una ramificació (*fork*) a partir de l'última versió lliure d'*Alexandria*.

Evidentment, en estar Gforge disponible com a programari lliure, qualsevol el pot instal·lar i configurar, i oferir un lloc nou per allotjar i desenvolupar projectes de programari lliure. Així ho fan molts, com podrà observar el lector si visita la pàgina principal del projecte, encara que no és una tasca fàcil si es vol oferir un servei impecable. Tanmateix, aquesta no és una tasca necessària per als nostres propòsits i no es tractarà en aquest material. Els llocs que s'encarreguen d'aquesta tasca disposen d'equips dedicats exclusivament a oferir aquest servei, i tenen la infraestructura (servidors, amplada de banda, espai en disc) i els coneixements necessaris per a aquesta finalitat.

Els llocs que s'esmenten a continuació estan basats en plataformes similars les unes amb les altres, encara que tant les funcionalitats com l'aparença poden variar lleugerament segons la versió d'Alexandria/GForge que utilitzin.

Nota

Software-libre.org:

<http://www.software-libre.org>

Nota

Savannah:

<http://savannah.gnu.org>

1.2.1. [Software-libre.org](#)

Software-libre.org va néixer com una iniciativa de l'associació Hispalinux per tal de tenir una forja de projectes pròpia. En principi va néixer com "un lloc on poder desenvolupar coneixement lliure en general i programari lliure en particular en idioma espanyol, encara que no es descarta l'ús d'altres idiomes". Tanmateix, avui dia la seva política restrictiva d'usuaris (no permet tenir compte a usuaris que no pertanyin a una organització que tingui un acord amb Hispalinux) n'ha dificultat la popularització.

A més, en aquest moment, només es troba en espanyol, per la qual cosa tampoc no és recomanable per a projectes de programari lliure que busquin una gran difusió.

1.2.2. [Savannah](#)

Savannah és una iniciativa de la Free Software Foundation. Qualsevol persona pot obrir un compte en aquest lloc, encara que en el cas de registre de projectes, s'imposa una sèrie de restriccions que es detallen a continuació:

- Els recursos posats a disposició dels usuaris no es poden utilitzar per a promocionar o desenvolupar programari no lliure, entenent com a programari lliure el que compleix la definició donada per la Free Software Foundation.
- El programari allotjat no pot dependre de programari no lliure.
- No es pot posar publicitat al lloc web, excepte anuncis comercials de suport al projecte.
- No es poden emprar fitxers d'imatge en format GIF perquè és un format patentat (encara que fa poc va expirar la patent a tot el món).

- La llicència del projecte ha de ser lliure; és a dir, qualsevol pot utilitzar, copiar, modificar i distribuir el programari sense restriccions, i s'han de respectar aquestes llibertats bàsiques en cas que es distribueixi.
- Si s'empra el terme *GNU* dins del nom del projecte, s'han de complir una sèrie de requisits addicionals.

Savannah té com a característica curiosa que distingeix entre els projectes que pertanyen al projecte GNU i els que no hi pertanyen.

1.2.3. Alioth

Alioth és el lloc de Debian per a oferir allotjament a projectes de programari lliure. La seva política d'usuaris no és restrictiva; és a dir, qualsevol usuari pot obtenir un compte en aquest lloc. Tampoc no té restriccions especials, excepte que el projecte ha de ser de programari lliure o estar-hi relacionat. La seva interfície està disponible en diversos idiomes, entre els quals hi ha l'espanyol i l'anglès.

1.2.4. BerliOS

BerliOS és similar als llocs anteriors. Ofereix tots els serveis ja esmentats i la seva política d'usuaris i projectes no és restrictiva. Està disponible en diversos idiomes, entre els quals hi ha l'espanyol i l'anglès.

Aquest lloc és un dels més populars entre els desenvolupadors de programari lliure i sempre s'ha vist com la competència de SourceForge que no ha acabat d'engegar mai.

1.2.5. SourceForge

Sens dubte, SourceForge és el lloc més famós entre els desenvolupadors de programari lliure. Té gairebé un milió d'usuaris registrats i gairebé cent mil projectes donats d'alta, encara que molts no han presentat mai cap activitat. Ofereix tots els serveis ja esmentats. La seva interfície només està disponible en anglès.

Nota

Alioth:
<http://alioth.debian.org>

Nota

BerliOS:
<http://developer.berlios.de>

Nota

SourceForge:
<http://sourceforge.net>

Té l'inconvenient de la massificació, que de vegades en perjudica el rendiment. Encara que, al seu torn, tanta activitat repercuteix sens dubte positivament en el foment del programari, tant des del punt de vista del coneixement de la seva existència com per la possibilitat d'atraure desenvolupadors.

1.3. Registre del projecte

De tots els llocs comentats, emprarem SourceForge per a mostrar com es comença un projecte de programari lliure.

El primer pas consisteix a obtenir un compte al lloc. En el cas de SourceForge no hi ha cap restricció perquè puguem crear-hi un compte. Per a això hem de pulsar sobre l'enllaç "New User Via SSL" a la pàgina principal de SourceForge. En el formulari d'inscripció hem d'escriure una contrasenya i una adreça electrònica. Després d'enviar aquestes dades, se'ns demanarà que revisem l'adreça electrònica i, si és correcta, que continuem amb el procés de registre. Si ho fem, rebrem un missatge a aquesta adreça electrònica per confirmar l'alta de l'usuari. El sistema ens respondrà que rebrem un missatge en el termini de vint-i-quatre hores. Normalment, el missatge sol ser instantani, encara que a causa dels problemes de massificació de SourceForge, pot ser que triguem una mica de temps a rebre'l.

El contingut del missatge és el següent:

From: SourceForge.net <noreply@sourceforge.net>
To: algu@algunlloc.com
Date: Wed, 12 Oct 2023 06:50:29 -0800
Subject: SourceForge.net Account Registration: Email Verification

This email has been generated automatically by SourceForge.net as part of your request to register a SourceForge.net user account.

The purpose of this email is to verify that the email address you provided to SourceForge.net exists and that you may read the mail sent to this address. It is important that you use a private email address which will remain in your control (not a disposable email address). SourceForge.net will use this email address in the future to notify you of account problems, or if you need to recover a lost account password.

To proceed with account registration, please access the following URL:

https://sourceforge.net/account/newuser_register.php?confirm_hash=xxxxxxx

This URL should be entered into your browser on a single line with no spaces.

By accessing this URL, you will confirm that we are able to send email to this email address (algu@algunlloc.com).

Nota

L'única eina que necessitem per ara per a seguir aquestes instruccions és un navegador web.

After email address validation is completed, you will be permitted to select a username and provide account details. Questions or concerns about this process may be directed to <http://SourceForge.net>.

Questions or concerns about this process may be directed to SourceForge.net staff at:
<https://sourceforge.net/docs/C01/>

We hope you enjoy SourceForge.net.

-- the SourceForge.net staff

Si seguim l'enllaç inclòs al missatge, haurem de tornar a confirmar l'adreça electrònica i la contrasenya escollida, i després podrem especificar finalment els detalls del nostre compte, com el nom de l'usuari o l'idioma escollit (entre els quals hi ha l'anglès i l'espanyol). En prémer el botó per a verificar les dades, el sistema ens mostrarà de nou el mateix formulari per a comprovar que les dades són correctes. Una vegada revisats, haurem obtingut per fi el nostre compte en SourceForge. Rebrem un missatge de benvinguda en l'adreça proporcionada.

Una vegada obtingut compte, hi podem entrar mitjançant l'enllaç "Login Via SSL" situat a la part superior esquerra de la pàgina. La primera pàgina que veiem després de la nostra identificació (*login* o connexió) és la pàgina personal de l'usuari. Des d'aquesta pàgina podrem accedir en un futur als diferents projectes que tinguem registrats.

El primer pas que haurem de fer és completar les nostres dades, tal com ens demana el missatge que apareix a la nostra pàgina personal:

You have not yet set the True Identity details for your account. Please do so at this time.

També és interessant completar el nostre perfil de desenvolupador. Per a això, utilitzem l'opció 'Account Options' de la nostra pàgina personal. Després escollim 'Skills Profile' i definim quines són les nostres habilitats com a desenvolupador.

Exemple

En la figura 1.1 hem definit que el nostre perfil sigui públic i hem afegit algunes habilitats fictícies. Com es pot observar, es pot triar l'antiguitat en cada habilitat i el nivell tècnic. Aquest perfil pot ser útil a l'hora de participar en altres projectes en SourceForge, com a currículum.

Figura 1-1. Dades de perfil de desenvolupador de SouceForge

The following option determines if others can see your resume online. If they can't, you can still enter your skills, and search for matching jobs.

Publicly Viewable:
 No
 Yes

Give us some information, either a resume, or an explanation of your experience.

Resume / Description of Experience:

[Update Profile](#)

Skill	Level	Experience	Action
Programming Language :: Python	Wrote The Book	< 6 Months	Update
Operating System :: Modern (Vendor-Supported) Desktop Operating Systems :: Linux	Competent	< 6 Months	Update
Operating System :: Modern (Vendor-Supported) Desktop Operating Systems :: FreeBSD	Want to Learn	< 6 Months	Update
Programming Language :: C++	Wizard	2 yr - 5 yr	Update

Add A New Skill

Database Environment :: Database API :: ADODB	Want to Learn	< 6 Months	Add Skill
---	---------------	------------	-----------

Una vegada que hàgim escrit el nostre perfil com a desenvolupador, podem triar entre unir-nos a un projecte ja existent, o crear-ne un de propi. El registre de projectes no és automàtic, sinó que requereix la validació del personal de SourceForge. Hem d'explicar en què consisteix el nostre projecte, quins són els seus objectius, i per què el volem registrar en SourceForge.

Vegem un exemple. A la pàgina personal de l'usuari hi ha un enllaç per a començar el procés de registre d'un projecte nou (a baix a la dreta).

La primera pàgina ens demana que aclarim si sabem en què consisteix el programari lliure i quina classe de projecte registrarem.

Una vegada fet això, una pàgina nova ens explica quins passos seguirem en el procés de registre. Els passos són els següents (entre claudàtors la traducció):

1. Hosting information [Informació de l'allotjament]

2. Registering a project (current step) [Registre de projecte (pas actual)]
3. Terms of use agreement [Acceptació dels termes d'ús]
4. Hosting requirements [Requisits d'allotjament]
5. Project license details [Detalls de la llicència del projecte]
6. Project description details [Detalls de la descripció del projecte]
7. Project name details [Detalls del nom del projecte]
8. Final review [Revisió final]
9. Submission completed [Acabament del procés]

Hem de llegir detingudament la informació mostrada a cada pas. Un requisit imprescindible per a registrar un projecte és que la seva llicència sigui de programari lliure. Podem escollir la llicència d'entre una llista o bé especificar una llicència pròpia, els termes de la qual estiguin d'acord amb el concepte de programari lliure.

En la descripció del projecte, hem d'escriure en què consisteix i quins són els seus objectius. Una de les descripcions serà la que aparegui a la pàgina del projecte i l'altra és la que es dirigeix als administradors de SourceForge perquè decideixin sobre el registre del projecte. Totes les descripcions han d'estar escrites en anglès.

Una vegada emplenats tots els formularis, haurem d'esperar que els administradors de SourceForge aprovin el nostre projecte. Aquesta aprovació pot trigar uns quants dies i, en general, no solen denegar projectes llevat que s'allunyin del tot de la temàtica del lloc.

Quan el nostre projecte s'aprovi, a la nostra pàgina personal apareixerà un enllaç a la pàgina del projecte (a baix a la dreta). Per exemple, en la figura 1.2 veiem que l'usuari té tres projectes registrats.

Figura 1-2. Projectes registrats a la pàgina personal d'un usuari de SourceForge

My Projects

- CAFLUR Chemical Reactors Analysis Tool
- GnuPinch
- CAGESOL

To remove yourself from a project, click the Trash Can icon. Admins must have their project admin flag dropped first.

[See a full list of your Active/Pending/Rejected/Deleted projects](#)

En polsar l'enllaç d'algun projecte, ens dirigirem a la pàgina del projecte seleccionat, on podem controlar totes les eines.

La figura 1.3 mostra les diferents opcions disponibles a la pàgina del projecte. Les anirem veient totes una per una, i posarem un èmfasi especial en les eines d'administració i desenvolupament.

Figura 1-3. Pàgina principal d'un projecte SourceForge

Project: [Summary](#) ← **NOMBRE DEL PROYECTO**

Summary | **Admin** | Home Page | Forums | Tracker | Bugs | Support | Patches | RFE | Lists | Tasks | Docs | Screenshots | News | CVS | Files | Donations |

ADMINISTRACION | **BARRA DE HERRAMIENTAS**

DESCRIPCION DEL PROYECTO

DEVELOPER INFO

Project Admins: [View](#)

Developers: 2
[View Members]

ADMINS Y DESARROLLADORES

CARACTERISTICAS

- Development Status: 4 - Beta
- Intended Audience: Developers, End Users/Desktop, Education, Science/Research
- License: GNU General Public License (GPL)
- Operating System: All POSIX (Linux/BSD/UNIX-like OSes), Linux
- Programming Language: C++, Python
- Topic: Education, Scientific/Engineering
- Translations: English, Spanish
- User Interface: Gnome

ESTADISTICAS

Project LINUX name: [gnome](#)

Registered: 2004-04-23 14:56

Activity Percentile (last week): 0%

[View project activity statistics](#)

[View list of RSS feeds available for this project](#)

[\[Set preferred support mechanism\]](#)

[\[Configure Tracve categorization\]](#) ← **EDITAR CATEGORIAS**

A la part superior dreta, a sota de 'Developer Info', podem veure la llista d'administradors del projecte i el nombre de desenvolupadors que formen part del projecte. Inicialment, només hi ha un administrador i un desenvolupador.

A la barra d'eines podem observar tota la infraestructura que SourceForge posa a disposició del nostre projecte. Pulsant qualsevol eina, obtenim una vista com a usuari d'aquesta eina. Si volem configurar alguna eina, ho hem de fer des de la interfície d'administració.

També podem veure la descripció que hem escrit durant el procés de registre del projecte. Quant a les donacions i les característiques del projecte, inicialment no veurem res. Per a editar la categorització del nostre projecte, hem d'utilitzar l'enllaç que assenyallem amb el text "EDITAR CATEGORIES".

A continuació, examinarem les eines més importants amb més deteniment.

1.4. El compte de línia d'ordres

SourceForge proporciona a cada usuari un compte de línia d'ordres per poder efectuar canvis al web i accedir com a desenvolupador al CVS. Podem accedir al nostre compte mitjançant SSH. Per a això utilitzem el client SSH, present en la majoria dels sistemes UNIX (entre els quals, el Linux). Dins d'un terminal de línia d'ordres, teclegem l'ordre següent:

```
ssh username@projname.sf.net
```

on *username* és el nom d'usuari i *projname* és el nom del projecte. Se'ns demanarà la contrasenya de l'usuari per poder entrar en el sistema.

Els fitxers del web del nostre projecte es troben al directori `/home/groups/p/pr/projname/htdocs/`. De la mateixa manera, el nostre directori d'usuari es troba a `/home/users/u/us/username/`, on *projname* és el nom del projecte (i *p* i *pr* són la primera i les dues primeres lletres del nom del projecte), i *username* és el nom de l'usuari (i *u* i *us* són la primera i les dues primeres lletres del nom de l'usuari).

La mateixa contrasenya que acabem d'escriure se'ns demanarà cada vegada que accedim al CVS com a desenvolupador. Per a evitar haver d'introduir cada vegada la clau, en l'apartat següent explicarem com instal·lar Keychain al nostre ordinador.

1.4.1. Accés mitjançant SSH sense clau

Keychain és una eina que podem emprar per a evitar teclejar la nostra contrasenya cada vegada que accedim al CVS o que copiem algun fitxer mitjançant l'ordre `scp`.

En sistemes basats en Debian, Keychain s'instal·la simplement amb l'ordre:

```
apt-get install keychain
```

Una vegada instal·lat, hem d'afegir les línies següents al fitxer `.bash_profile` present en el nostre directori d'usuari. La primera línia executa Keychain utilitzant el parell de claus generat i la segona línia llança el guió que crearem quan configurem Keychain.

```
keychain ~/.ssh/id_rsa ~/.ssh/id_dsa
. ~/.keychain/${HOSTNAME}-sh
```

El pas següent és generar un parell de claus, pública i privada, que s'empraran per a autenticar l'usuari. Per a això usem l'ordre:

```
ssh-keygen -t rsa
```

En executar aquesta ordre, se'ns demana un fitxer, on s'escriurà la clau privada, i una contrasenya per a utilitzar com a clau privada. Per al fitxer deixem l'opció per defecte. Quant a la contrasenya, molta gent tria no posar-n'hi cap, per no haver-la d'escriure quan accedeixen per SSH. Això és perillós, perquè si en algun moment el nostre compte d'usuari es veu compromès, també ho estarà el compte en SourceForge. A més, utilitzant Keychain, només caldrà introduir la nostra contrasenya una única vegada per sessió.

Nota

Per a altres distribucions, es poden trobar paquets a:

<http://dev.gentoo.org/~agriffis/keychain/>

La clau pública que hàgim generat serà al fitxer `.ssh/id_rsa.pub` al nostre directori d'usuari. En copiem el contingut, i ens dirigim a <https://sourceforge.net/account/editsshkeys.php> (aquest enllaç és a 'Account Options', dins de la nostra pàgina personal de SourceForge, al final, sota l'epígraf 'Number of SSH Shared Keys on file'). Enganxem el contingut al quadre de text i pitgem el botó 'Update'.

Quan hàgim acabat, haurem d'esperar unes quantes hores perquè els canvis tinguin efecte en el servidor, i després ja podrem accedir als comptes de la nostra línia d'ordres i al CVS teclejant la contrasenya que hem utilitzat en generar el parell de claus pública i privada.

La primera vegada que obrim un terminal virtual, o una consola virtual, Keychain ens demanarà la clau privada. A partir de llavors, Keychain la introduirà automàticament cada vegada que se'ns demani. En altres paraules, introduïm la contrasenya una vegada al principi i després es tornarà a utilitzar quan Keychain la necessiti.

1.4.2. Com es poden copiar els fitxers del web

Podem utilitzar l'ordre `scp` per a copiar els fitxers del web. Per a copiar un sol fitxer:

```
scp filename username@projname.sf.net:/home/groups/p/pr/projname/htdocs/
```

on *filename* és el nom del fitxer.

Si volem copiar tot un arbre de directoris i reproduir-lo exactament al servidor, podem utilitzar l'eina `rsync`:

```
rsync -v -rsh = "ssh -l username" source_path  
username@projname.sf.net:/home/groups/p/pr/projname/htdocs/
```

on *source_path* és el nom del directori local. Si hem utilitzat Keychain, no ens hauria de demanar la contrasenya aquesta segona vegada. Aquesta ordre copia el directori local *source_path* al servidor, respectant tota l'estructura del directori; és a dir, la còpia de fitxers al servidor és recursiva.

Nota

Com s'esmenta a la mateixa pàgina, qualsevol usuari anònim pot accedir al codi font d'un projecte i utilitzar el CVS. En canvi, per a poder modificar el codi font emmagatzemat al CVS, cal ser un desenvolupador del projecte.

1.5. Configuració del CVS

Si pitgem l'enllaç CVS, a la barra d'eines, obtindrem la interfície d'usuari del CVS.

El CVS (*concurrent versions system*) és una eina molt popular entre els projectes de programari lliure. S'empra per a gestionar els canvis al codi font del projecte, per tal que diversos desenvolupadors puguin treballar de manera coordinada sobre el mateix codi font.

En els apartats següents es mostrarà simplement com es pot accedir al codi allotjat en un dipòsit CVS, de manera anònima o com a desenvolupador d'un projecte. Més endavant, en un apartat íntegrament dedicat a la gestió i ús del CVS, veurem amb més detall com s'utilitza aquesta eina.

1.5.1. Accés anònim

Per a accedir de manera anònima al servidor CVS, utilitzarem el client CVS, present en la majoria de sistemes UNIX (inclòs Linux). Per a això, teclegem això en un terminal d'ordres:

```
cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/projname login
Login com a usuari anònim
cvs -z3 -d:pserver:anonymous@cvs.sf.net:/cvsroot/projname co modulename
Baixada del mòdul
```

on *projname* és el nom del projecte, i *modulename* és el nom del mòdul.

Podem esbrinar fàcilment quins mòduls hi ha al CVS polsant l'enllaç 'Browse CVS Repository' de la interfície web. En la figura 1.4 se'n mostra un exemple.

Figura 1-4. Interfície web d'un projecte

The screenshot shows a web interface for a CVS project. At the top, it says 'cvs: gpinch'. Below that, it indicates the current directory is '[cvs] / gpinch' and that 0 files are shown. A table lists the files in the directory:

File	Rev.	Age	Autho
CVSROOT/			
PinchPython/			
gnupinch/			

Below the table, there is a dropdown menu labeled 'Show files using tag:' with 'Select Branch' selected. At the bottom, there are links for 'Back to SourceForge.net' and 'Powered by ViewCVS 1.0-dev'.

En aquest cas hi ha dos mòduls: PinchPython i gnupinch.

1.5.2. Accés com a desenvolupador

L'accés com a desenvolupador ens permet realitzar canvis al dipòsit de codi font. Primer de tot, hem de dir al client CVS que emprï SSH per a comunicar-se amb el servidor. Això ho fem mitjançant l'ordre:

```
export CVS_RSH = ssh
```

Una altra variable d'entorn útil que podem definir és CVSROOT, per a no haver d'indicar el dipòsit mitjançant l'opció -d de l'ordre cvs.

```
export CVSROOT = :ext:username@cvs.sf.net:/cvsroot/projname
```

També podem afegir les dues línies anteriors al `.bash_profile` per evitar haver-ho d'escriure cada vegada que accedim al servidor CVS. Després d'això, executem l'ordre:

```
cvs -z3 co modulename
```

1.6. Baixades del projecte

Una altra opció de SourceForge que utilitzarem sovint és la baixada. Evidentment, sempre podrem optar per posar els fitxers en algun directori del nostre web i que es puguin baixar per HTTP. Tanmateix, si emprarem l'eina de baixades de SourceForge, podrem obtenir estadístiques de baixades, un historial de publicacions (*releases*) i, a més, tots els usuaris podran utilitzar totes les rèpliques (*mirrors*) de SourceForge per a baixar els fitxers (els webs i el CVS no estan replicats, només els fitxers de les publicacions), cosa que generalment n'agilita la baixada.

L'eina de baixades s'administra des de la interfície d'administració (vegeu la figura 1.3). Dins de la interfície d'administració, tenim accés a una barra amb les diferents eines d'administració (figura 1.5).

Figura 1-5. Barra d'eines d'administració d'un projecte SourceForge

Admin | Members | Recruitment | Public Info | Backups | Audit | Publicity | Registration | Removal | Donations | **File Releases** | Tracker | Tasks | DocManager | Screenshots | Lists/Forums | Shell/DB/Web | CVS | Stats

En aquest moment, ens centrarem en l'eina 'File Releases'. Si seguim aquest enllaç, arribem a una pàgina que ens informa sobre aquesta eina. Al final de la pàgina, tenim un formulari per a crear paquets nous (figura 1.6).

Figura 1-6. Interfície de creació de paquets nous

Releases	Package Name	Status	Update
[Add Release] [Edit Releases]	<input type="text"/>	Active ▾	<input type="button" value="Update"/>
[Add Release] [Edit Releases]	<input type="text"/>	Active ▾	<input type="button" value="Update"/>
[Add Release] [Edit Releases]	<input type="text"/>	Hidden ▾	<input type="button" value="Update"/>
[Add Release] [Edit Releases]	<input type="text"/>	Active ▾	<input type="button" value="Update"/>

New Package Name:

En un primer moment, no veurem cap paquet ja creat.

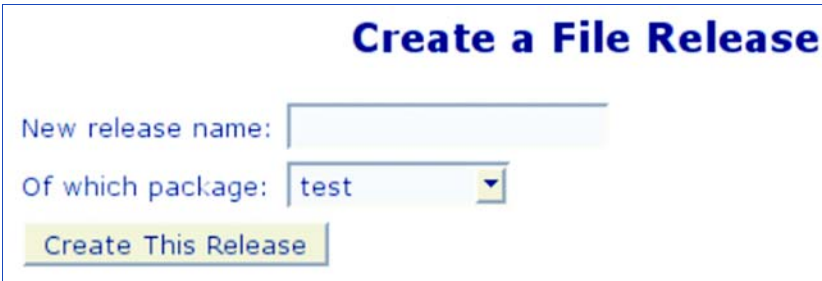
Les baixades s'organitzen en paquets i versions. Els **paquets** es poden entendre com a subprojectes dins del projecte i cada paquet pot contenir diverses versions.

El primer que cal fer per a afegir fitxers amb la finalitat de baixar és crear un paquet. Escrivim el nom del paquet i pitgem el botó 'Create This Package'.

Una vegada creat el paquet, el podem ocultar o activar. No es poden esborrar paquets una vegada creats, així que en lloc d'esborrar un paquet, l'ocultarem.

Per a afegir versions a un paquet, utilitzem l'enllaç 'Add Release'. N'escollim un nom (o un número de versió), el paquet del qual forma part, i creem aquesta versió.

Figura 1-7. Interfície de creació de versions



Create a File Release

New release name:

Of which package:

Després de crear la versió, arribarem a un formulari que consta de tres passos:

1. En el primer pas hem d'emplenar les notes de la versió i el registre de canvis. Les notes de la versió consisteixen en un resum de les funcionalitats noves i els errors corregits, mentre que el registre de canvis és la bitàcola detallada de canvis que s'han efectuat. D'altra banda, en els projectes de programari lliure hi ha el bon costum de tenir un fitxer anomenat `ChangeLog` on s'introdueix, després de cada canvi realitzat, la data del canvi, el nom del desenvolupador (i el seu correu electrònic), el fitxer modificat i una breu descripció del que s'ha

Exemple

En el nostre projecte podem tenir dos paquets: *programa* i *documentació*. I al paquet *programa*, la versió 1.0, la 2.0, etc.

modificat. En aquest cas, el contingut d'aquest fitxer es pot copiar al camp destinat a aquest propòsit.

Figura 1-8. Interfície de registre de canvis d'una versió

Step 1: Edit Existing Release

Release Date:

Release Name:

Status:

Of Package: test

Edit the Release Notes or Change Log for this release of this package. These changes will apply to all files attached to this release. You can either upload the release notes and change log individually, or paste them in together below. Release Notes and Change Log content must be between 20 and 256000 bytes in length.

Upload Release Notes:

Upload Change Log:

Paste The Notes In:

Paste The Change Log In:

Preserve my pre-formatted text.

2. El segon pas consisteix a escollir els fitxers que formen part de la versió. Hem de triar els fitxers de la nostra versió d'entre els que apareixen a la llista (figura 1.9). Els fitxers que apareixen en aquesta llista són els del directori `incoming` de la màquina `upload.sf.net`. Podem posar qualsevol fitxer en aquest directori accedint per FTP anònim. Els noms dels fitxers no poden contenir espais, parèntesis ni el caràcter `'~'`.

És important escollir només els nostres fitxers, ja que una vegada que se seleccionen per a formar part d'una versió, ja no tornen a aparèixer a la llista (i per tant, el propietari no els podrà utilitzar per a la seva versió i els haurà de tornar a pujar). Dins d'una versió podem incloure tants fitxers com vulguem.

Figura 1-9. Interfície de selecció de fitxers d'una versió

Step 2: Add Files To This Release

Next, choose your files from the list below. Choose **ONLY YOUR** files. If you choose someone else's files, they will not be able to access them and they will be rightfully upset.

You can upload new files using **anonymous FTP** to **upload.sourceforge.net** in the **incoming** directory. When you are done uploading, just hit the refresh button to see the new files. Further information regarding this process may be found in our Guide to the File Release System; please refer to this document if you encounter any difficulties in performing this file release.

PLEASE NOTE: filenames may not contain a space, parenthesis or the "~" character..

- FreeDoko_0.6.7b.hpux.tgz
- GroundTexture-128x128.tar.gz
- cc_v1_3.tgz
- cc_v1_4.tgz
- epc-4.4.0.tar.gz
- imc-1.2-11.rhel3.i386.rpm
- libcli-1.8.4.tar.gz
- motioncleanup.tar.gz
- motionwatchdog.tar.gz
- mygosuclan-patch-sql.zip
- scim-1.1.0.tar.gz
- scim-tables-0.5.0.tar.gz
- webinject-1.30.src.tar.gz
- webinject-1.30.win32.zip

3. El tercer pas consisteix a classificar cada fitxer de la versió (figura 1.10). Hem d'indicar per a quina arquitectura és el fitxer (o si és independent d'arquitectura; només els fitxers binaris haurien de ser dependents de l'arquitectura) i de quina mena de fitxer es tracta (codi font, documentació, etc.). També podem esborrar un fitxer d'una versió.

Figura 1-10. Interfície de classificació dels fitxers d'una versió

Step 3: Edit Files In This Release

Once you have added files to this release you **must** update each of these files with the correct information or they will not appear on your download summary page.



Filename Release	Processor Release Date	File Type Update
Chronus.jar <input type="text" value="test : 1.0"/>	<input type="text" value="Must Choose One"/> <input type="text" value="2005-01-05"/>	<input type="text" value="Must Choose One"/> <input type="button" value="Update/Refresh"/> <input type="button" value="Delete File"/> <input type="checkbox"/> I'm Sure

Amb això ja haurem definit una versió dins d'un paquet. Ara, el paquet hauria d'aparèixer a la llista de baixades (figura 1.11) i, en pitjar el paquet, la versió que acabem de crear.

Figura 1-11. Llista de baixades

Summary | Admin | Home Page | Forums | Tracker | Bugs | Support | Patches | RFE | Lists | Tasks | Docs | Screenshots | News | CVS | **Files** | Donations |

Below is a list of the files for the selected file package. Other views: [all files released by this project] Before downloading, you may want to read Release Notes and ChangeLog (accessible by clicking on release version).

Package	Release & Notes	Filename	Size	Date D/L	Arch.	Type
test						
 1.0 [show only this release]				2005-01-05 19:16		
Chronus.jar			1384804	0	None	None
Project Totals:	1	1	1384804	0		

Una vegada que hàgim publicat una versió nova del nostre programari, l'hauríem d'anunciar convenientment perquè tingui la màxima difusió. Amb aquesta finalitat és una bona idea utilitzar les llistes de correu electrònic que ofereix SourceForge, l'administració de les quals es presenta en l'apartat següent.

1.7. Llistes de correu

SourceForge posa a disposició dels projectes la possibilitat de crear i gestionar llistes de correu. Per a aquest propòsit empra el gestor de llistes de correu GNU Mailman, encara que la creació i la gestió de les llistes de correu s'efectua des de la interfície de SourceForge.

Les llistes de correu es poden administrar amb la barra d'eines que mostra la figura 1.12.

Figura 1-12. Eina d'administració de les llistes de correu

Admin | Members | Recruitment | Public Info | Backups | Audit | Publicity | Registration | Removal | Donations | File Releases | Tracker | Tasks | DocManager | Screenshots | **Lists/Forums** | Shell/DB/Web | CVS | Stats

Una vegada a la pàgina d'administració, podem escollir l'opció d'habilitar o deshabilitar les llistes de correu i els fòrums a la pàgina del projecte (figura 1.13).

Figura 1-13. Eina d'administració de les llistes de correu i els fòrums de discussió

Mailing Lists and Discussion Forums

When developing software, communication is vital. To aid in your efforts to communicate within your development team and for your developers to effectively communicate with your end-users, SourceForge.net provides projects the ability to establish mailing lists (email-based) and discussion forums (web-based). For management of support issues, bugs, patches and feature requests, we also provide the **Tracker** system.

Enable Mailing Lists? The following box should be checked to enable Mailing Lists:

Enable Discussion Forums? The following box should be checked to enable Discussion Forums:

[Update](#)

Mailing List Admin: Mailing Lists are managed in two places. The creation of mailing lists is handled from the Mailing List Admin page on the SourceForge.net site. Mailing lists may be created by Project Admins. Once created, [mailing lists cannot be removed](#) (flagging the list as deleted merely removes it from the listing of available mailing lists, but does not shut down the mailing list -- contact the SourceForge.net team by submitting a Support Request for further assistance).

Once a list has been created, it is configured using the Mailman mailing list management software (which SourceForge.net uses to provide its mailing list service). Access to manage a mailing list via Mailman is provided using a separate admin password for each list -- this password does not match your SourceForge.net user password; there is only one admin password for each list, regardless of how many people administrate that list.

At time of list creation, a default password is sent to the list creator via email. If this password is lost, it may be reset via the Administer/Update Lists page for your project.

- [Manage Mailing Lists](#)
- Links to the Mailman list administration interface for each list may be found

Per a l'administració de les llistes, anirem a l'enllaç 'Manage Mailing Lists'. Quan hi siguem, hi podrem afegir llistes noves o gestionar les que ja hi ha (figura 1.14).

Figura 1-14. Creació i gestió de llistes de correu

Admin

[Add Mailing List](#)

[Administer/Update Lists](#)

Si escollim afegir-hi una llista nova, haurem d'emplenar el formulari que es mostra en la figura 1.15.

Figura 1-15. Interfície de creació d'una nova llista de correu

Admin

Lists are named in this manner:
projectname-listname@lists.sourceforge.net

It will take **6-24 Hours** for your list to be created.

Existing Mailing Lists

list_name

projectname@lists.sourceforge.net

Mailing List Name:

projectname@lists.sourceforge.net

Is Public?

Yes

No

Description:

Once created, this list will ALWAYS be attached to your project and cannot be deleted!

[Add This List](#)

Regarding the "Is Public?" flag: Public mailing lists are shown when a user goes to the Mailing Lists page for your project, and have mailing list archives that are accessible to the public. The ability to subscribe and post to a list is controlled in the Mailman interface for your list, not via this flag. Lists flagged as deleted will not appear in the listing of mailing lists for your project, but will still be present within Mailman.

Nota

Poden passar unes quantes hores fins que la llista es creï definitivament.

Les llistes no es creen immediatament, sinó que la creació de llistes és una tasca de cron. Les tasques de cron s'executen periòdicament, segons com les hagi configurat l'administrador del lloc.

Hem de tenir en compte que les llistes no es poden esborrar. Tan sols podem marcar una llista com en desús i no se'n podran consultar els arxius (si són públics, com expliquem a continuació).



Les llistes poden ser públiques o privades. Les **l·listes privades** només permeten accedir a l'arxiu de la llista als usuaris subscrits a aquesta llista.

Posteriorment, des de la interfície de Mailman podrem configurar les opcions de subscripció perquè qualsevol usuari s'hi pugui registrar o perquè demani l'aprovació de l'administrador de la llista.

Quan la llista de correu ja s'ha creat, podem canviar algunes opcions de configuració (figura 1.16).

Figura 1-16. Opcions de configuració d'una llista de correu

Admin

You can administrate lists from here. Please note that private lists can still be viewed by members of your project, but are not listed on SourceForge.net.

List	Status	Update	List Admin
llogador@gnulinux.org	Is Public? <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Deleted	Update	[Administer this list in GNU Mailman] [Search/Display subscriber list] [Change list admin password]
Description: Llista de correu per als membres individuals			
llogador@gnulinux.org	Is Public? <input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Deleted	Update	[Administer this list in GNU Mailman] [Search/Display subscriber list] [Change list admin password]
Description: Llista principal de contactes, corresponents			

Regarding the "Is Public?" flag: Public mailing lists are shown when a user goes to the Mailing Lists page for your project, and have mailing list archives that are accessible to the public. The ability to subscribe and post to a list is controlled in the Mailman interface for your list, not via this flag. Lists flagged as deleted will not appear in the listing of mailing lists for your project, but will still be present within Mailman.

Des d'aquesta pàgina també podem accedir a la interfície de Mailman per a configurar al detall les opcions de la llista de correu.

1.8. Tracker i el sistema de seguiment d'errors

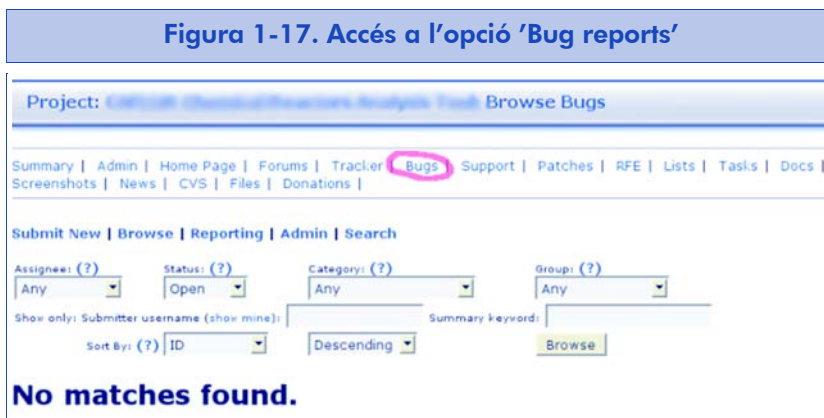
El sistema de seguiment d'errors de SourceForge s'anomena *Tracker*, i no està ideat només per a la gestió d'errors. Les possibilitats que inclou són (traducció entre claudàtors):

- Support requests [Peticions de suport]
- Bug reports [Informes d'errors]

- Feature requests [Peticions de funcionalitat]
- Patches [Pedaços]

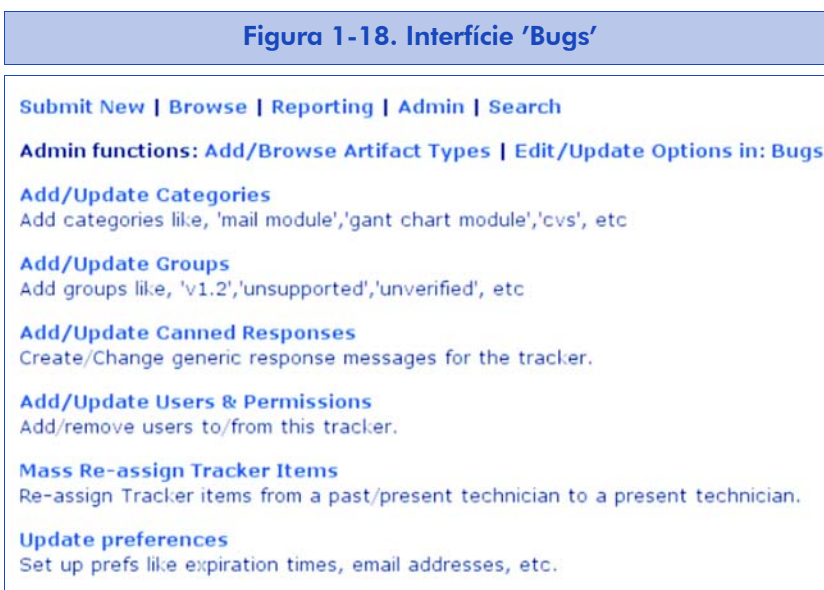
El funcionament per als tres tipus d'informes ('Support requests', 'Bug reports' i 'Feature requests') és similar. Nosaltres ens centrarem en el cas dels errors. A la interfície per a l'usuari s'accedeix des de l'enllaç 'Bugs' de la llista d'eines de la pàgina principal del projecte (figura 1.17).

Figura 1-17. Accés a l'opció 'Bug reports'



Si hem entrat amb el nostre compte d'usuari, veurem l'enllaç 'Admin' a la barra d'enllaços de l'eina. Si pitgem aquest enllaç, podem gestionar el Tracker per a totes les categories. Totes les interfícies són iguals, ja que només canvia el propòsit de cada Tracker. Per exemple, en el cas d'errors, podem gestionar les opcions que apareixen en la figura 1.18.

Figura 1-18. Interfície 'Bugs'



Les opcions són:

- Afegir i actualitzar categories. Per exemple, dins d'un sistema operatiu, els errors del nucli o *kernel* i els errors de la línia d'ordres.
- Afegir i actualitzar grups d'errors. Per exemple, grups d'errors per versions del programari.
- Afegir i actualitzar respostes automàtiques als informes d'errors.
- Afegir i actualitzar usuaris i permisos (per exemple, dotar alguns usuaris de permisos per a modificar els informes d'errors).
- Assignar un grup d'errors d'un desenvolupador a un altre.
- Actualitzar altres preferències (visibilitat del Tracker, períodes de temps per a canviar un error a obsolet, etc.).

Una vegada configurat el Tracker, els usuaris i desenvolupadors poden enviar errors al Tracker usant l'enllaç 'Submit New' de la interfície de l'eina (figura 1.19).

Figura 1-19. Interfície 'Submit New'

Submit New | Browse | Reporting | Admin | Search

For Project:
Can't find the project you're looking for?

Category: (?)
None (admin)

Group: (?)
None (admin)

Assigned To: (?)
None (admin)

Priority: (?)
5 - Medium

Summary: (?)

Detailed Description:

DO NOT enter passwords or other confidential information!

Check to Upload and Attach a File: (?)

File Description:

L'usuari pot assignar un error a un desenvolupador. Quan creem un error, apareix automàticament al Tracker (figura 1.20).

Figura 1-20. Assignació d'un error a un desenvolupador al Tracker

Submit New | Browse | Reporting | Admin | Search

Assignee: (?) Any | Status: (?) Open | Category: (?) Any | Group: (?) Any

Show only: Submitter username (show mine): | Summary keyword: |

Sort By: (?) ID | Descending | Browse

Request ID	Summary	Open Date	Priority	Assigned To	Submitted By
<input type="checkbox"/> 1096834	Test	2005-01-05 23:46	5	nobody	

Check All - Clear All

Posteriorment, un administrador del projecte o un desenvolupador que tingui prou permís pot canviar el desenvolupador assignat a un error (figura 1.21).

Figura 1-21. Canvi del desenvolupador assignat a un error

[1096834] Test

Monitor (?)

Submitted By:
Zoradi Herrera - Admin

Last Updated By:
Item Submitter - Tracker Item Submitted

Number of Comments:
0

Data Type: (?)
Bugs

Category: (?)
None (admin)

Assigned To: (?)

None (admin)

Status: (?)
Open

Summary: (?)
Test
test

Use Canned Response: (?)
None (admin)

Date Submitted:
2005-01-05 23:46

Date Last Updated:
No updates since submission

Number of Attachments:
0

Submit Changes

Group: (?)
None (admin)

Priority: (?)
5 - Medium

Resolution: (?)
None

Quan s'ha assignat un error a un desenvolupador, rebrà un correu electrònic de manera automàtica i l'error apareixerà a la seva pàgina principal (figura 1.22).

Figura 1-22. Anunci d'error a la pàgina principal del desenvolupador del projecte SourceForge

My Assigned Tracker Items (Open)	
1096834 - Bugs	
1096834	Test

L'usuari que ha notificat l'error també rep un correu electrònic amb cada novetat que es va produint i el pot veure a la seva pàgina personal de SourceForge (figura 1.23).

Figura 1-23. Anunci d'error a la pàgina principal de l'usuari

My Submitted Tracker Items (Open)	
1096834 - Bugs	
1096834	Test

Quan un desenvolupador resol un error, haurà de modificar l'estat de l'error a *Tancat*, i l'error quedarà emmagatzemat com a tal a SourceForge per a consultes posteriors. S'informa tant el desenvolupador com l'usuari que ha notificat l'error mitjançant correus electrònics del canvi d'estat en l'error.

2. Control de versions

Quan un o diversos programadors desenvolupen un mateix programari, normalment volen mantenir les diferents versions dels fitxers que desenvolupen. A més, sovint, en un moment determinat, se solen produir bifurcacions en el desenvolupament, i diversos programadors comencen a desenvolupar funcionalitats diferents de manera independent sobre un mateix fitxer.

En un moment determinat, el responsable d'aquest fitxer pot decidir barrejar els canvis fets en les diverses branques i crear fitxers nous que reuneixin la feina de tots els desenvolupadors.



Les eines de control de versions proporcionen, com a mínim, mecanismes per al següent:

- a) Emmagatzemar versions diferents d'un mateix fitxer perquè qualsevol les pugui recuperar en qualsevol moment.
- b) Definir diferents branques de desenvolupament, perquè els programadors puguin generar versions noves independents en cada branca.
- c) Automatitzar la fusió (en anglès, *merge*) de fitxers de versions o branques diferents. D'aquesta manera, la integració de la feina de cada programador sobre el mateix fitxer serà, en la mesura que es pugui, automàtica.

D'altra banda, uns sistemes de control de versions seran més interessants que d'altres segons les seves possibilitats.

Exemple

En CVS [Fogel00, Cederqvist04, Vesperman03], el sistema de control de versions més emprat, no és possible mantenir diferents versions dels directoris. En canvi, en Subversion (l'abreviatura és SVN, que es presenta com el sistema de control de versions de la pròxima generació), sí que es pot.

L'eina de control de versions també serà molt útil en la publicació de projectes lliures: sovint es crea l'usuari anònim, sense contrasenya, que permet accedir als dipòsits de programari, encara que no permet escriure-hi, tal com ja hem assenyalat en el capítol dedicat a les eines que proveeix SourceForge.

Així, els usuaris interessats poden obtenir les últimes versions del programari en el mateix moment en què els programadors les envien al control de versions, cosa que permet provar les últimes característiques implementades. Això farà que qui hi estigui interessat pugui provar l'última versió del nostre programari i fins i tot enviar-nos un pedaç que corregeixi algun error o hi afegixi alguna funcionalitat. En qualsevol cas, oferir la possibilitat de baixada anònima del CVS és una pràctica recomanable llevat que hi hagi continguts privats.

En els apartats següents veurem com s'instal·la i es configura un dipòsit CVS propi i a continuació ensenyarem els usos bàsics i avançats del CVS. Tenir un dipòsit CVS propi pot resultar interessant per a moltes coses: des d'efectuar còpies de suport dels nostres arxius fins a, per què no, allotjar els nostres projectes de programari lliure.

Per a fer un ús encertat del CVS, cal tenir en compte un detall: el versionat només funciona amb fitxers de text, ja que es basa en eines que comproven la diferència entre dos arxius de text com a `diff`. Els fitxers binaris, per tant, no gaudiran dels avantatges que ofereixen aquestes eines i es guardaran sempre íntegrament.

2.1. Instal·lació i configuració inicial de CVS

En moltes distribucions de GNU/Linux trobarem CVS empaquetat i amb instal·lació i configuració gairebé automàtica, cosa que ens permet des preocupar-nos en dels detalls. A continuació, mostrarem com configurar a mà un dipòsit CVS que compleixi les condicions d'ús recomanable.

1. Tots els desenvolupadors disposaran d'un compte d'accés al servidor CVS.
2. L'accés es durà a terme mitjançant connexions xifrades per a evitar la tramesa en clar de les contrasenyes.
3. S'habilitarà un mecanisme d'accés anònim perquè els usuaris puguin accedir als dipòsits i obtenir les versions més recents, sense que puguin modificar res.

Cada desenvolupador haurà de tenir un compte UNIX al servidor CVS. Definirem un grup UNIX a què pertanyeran tots els desenvolupadors i atorgarem a aquest grup el permís d'escriptura al dipòsit. Assumirem que aquest grup existeix i es diu `src`.

D'altra banda, per a habilitar l'accés anònim, hem d'habilitar el protocol `pserver` de CVS, sense contrasenya i amb accés de lectura únicament.

2.1.1. Creació del dipòsit

Vegem pas a pas les instruccions per a realitzar les tasques esmentades. Per a crear el dipòsit, s'han d'executar les ordres següents:

```
umask 002

mkdir /var/lib/cvs

chgrp src /var/lib/cvs

chmod 3775 /var/lib/cvs

cvs -d /var/lib/cvs init
```

Aquestes ordres han servit per a crear el dipòsit i obliguen mitjançant l'`sticky-bit` de fitxers de UNIX que tots els objectes nous creats dins del directori principal també pertanyin al grup `src`. D'altra banda, també es garanteixen els permisos necessaris perquè els usuaris d'aquest grup puguin escriure sempre. Finalment, l'ordre `init` de CVS crearà una sèrie de fitxers que conformaran l'estat inicial del dipòsit. Aquests fitxers es poden modificar per a un ús avançat de CVS que supera els objectius d'aquest document.

2.1.2. Preparació de l'accés anònim

L'accés anònim al dipòsit s'assigna pel protocol `pserver`, que permet que qualsevol usuari amb un client CVS obtingui el dipòsit des de qualsevol lloc.

En primer lloc, crearem el fitxer `/var/lib/cvs/passwd` amb el contingut següent:

```
anoncvs::cvsanonimo
```

També crearem el fitxer `/var/lib/cvs/readers` amb el contingut:

```
anoncvs
```

Cap dels dos fitxers no haurà de permetre l'escriptura a ningú, per raons de seguretat. Per a això, executarem l'ordre següent:

```
chmod 0444 /var/lib/cvs/passwd /var/lib/cvs/readers
```

El primer fitxer, `passwd`, estableix un compte per a accés `pserver`, anomenat `anoncvs`, sense contrasenya. A més, quan aquest usuari hi accedeixi, s'utilitzarà l'usuari UNIX `cvsanonim`, que hauré de crear, però deixant bloquejat el seu accés al sistema, ja que no serà necessari per a cap altra funció:

```
useradd -g src -d /tmp -s /bin/true cvsanonimo
```

```
passwd -l cvsanonimo
```


La funció del fitxer `readers` és indicar a CVS que aquest usuari només tindrà accés de lectura al dipòsit.

Finalment, és necessari activar el protocol `pserver`. Per a això, l'hem d'engegar com un servei més del superservidor `inetd`. Editarem el fitxer `/etc/inetd.conf` i hi afegirem la línia:

```
cvspserver stream tcp nowait root /usr/sbin/tcpd /usr/sbin/cvs-pserver
```

2.1.3. Obertura de comptes per als desenvolupadors

Tots els desenvolupadors que vulguin escriure al nostre CVS hauran de tenir un compte UNIX normal i pertànyer, almenys, al grup `src`. Per exemple, podríem crear els comptes amb una ordre similar a la següent:

```
useradd -g developer -G src -d /home/programador1 -m programador1
```

Atès que el mètode d'accés serà per SSH, és necessari que l'usuari tingui compte de línia d'ordres complet. Una vegada creat, li assignarem una contrasenya inicial, que el desenvolupador pot canviar més tard en entrar al sistema:

```
passwd programador1
```

2.2. Operativa bàsica del CVS

Després d'instal·lar i configurar el nostre dipòsit CVS, veurem com s'utilitza el CVS com a client. Veurem que el dipòsit permet un conjunt molt ampli d'interaccions, encara que el que s'utilitza més sovint és limitat, ja que es redueix a accedir al dipòsit, obtenir el codi d'un projecte, sincronitzar el dipòsit amb les nostres modificacions i sincronitzar la nostra còpia local amb el dipòsit.

2.2.1. Accés anònim a un dipòsit

Per a accedir a un dipòsit CVS de manera anònima, s'utilitzaran les ordres CVS corresponents, encara que prèviament caldrà entrar com

a anònim. Un usuari que, per exemple, vulgués obtenir del nostre sistema el dipòsit `projecte1`, escriuria l'ordre següent:

```
export CVSROOT=:pserver:anoncvs@nom-maquina.domini.com:/var/lib/cvs
cvs login
Password: (pitjarà INTRO deixant-la en blanc)
cvs co projecte1
```

Podrà fer qualsevol de les altres operacions CVS, sempre que no representin modificar el dipòsit.

2.2.2. Accés al CVS pel desenvolupador

Els desenvolupadors accedeixen al CVS mitjançant un túnel xifrat SSH, però no per això és complicat. Es fa de manera similar a l'accés que hem vist per a SourceForge; és a dir, primer es llança l'ordre:

```
export CVS_RSH=ssh
```

També el podem afegir al `.bash_profile` per no haver-lo d'escriure cada vegada que anem a accedir al servidor CVS. Després d'això, executem la sentència següent per accedir al `projecte1` del nostre dipòsit:

```
cvs -z3 -d :ext:usuari@nom-maquina.domini.com:/var/lib/cvs co projecte1
```

2.2.3. Creació d'un projecte al dipòsit

Cada projecte que es vulgui mantenir al dipòsit s'ha d'enviar per primera vegada. Per a això, un desenvolupador qualsevol amb drets d'escriptura se situarà al directori que contingui els fitxers que vulgui enviar inicialment (o un directori buit) i llançarà l'ordre:

```
cvs import projecte1 company inici
```

Això crearà el `projecte1` al nostre dipòsit. A més, s'anota el nom de la nostra empresa (`company`) i una etiqueta que s'afegirà als fitxers que hàgim enviat en aquesta operació (si el directori és buit, l'etiqueta no s'utilitzarà).

Totes les operacions d'escriptura en un dipòsit requereixen incloure un comentari que detalli qualsevol cosa rellevant. Per a això, se'ns obrirà l'editor estàndard (normalment, el `vi` de UNIX). En desar aquest fitxer de detall, es produirà l'escriptura efectiva al dipòsit.

2.2.4. Obtenció del projecte

Una vegada creat el projecte, cada desenvolupador –inclosos els usuaris anònims– pot obtenir la seva còpia local de treball mitjançant l'ordre:

```
cvs co projecte1
```

Aquesta operació d'obtenció de fitxers del dipòsit s'anomena *check-out*. Després del *check-out*, apareixerà un directori especial al costat dels fitxers de treball anomenat CVS, que l'aplicació utilitzarà i el contingut del qual podem ignorar per ara. Observeu que no n'hi ha prou de crear un projecte al dipòsit, sinó que a més, tal com hem fet en el punt anterior, caldrà obtenir-lo del mateix dipòsit. Allò que ens baixem del dipòsit es diu *còpia local*.

2.2.5. Creació de fitxers i directoris

Des de dins del directori de treball del projecte, podem afegir qualsevol fitxer nou al dipòsit del projecte mitjançant l'ordre:

```
cvs add fitxer(s)
```

Això també es pot fer amb els directoris que vulguem crear.

Quan es volen desar fitxers binaris al dipòsit, cal afegir a l'ordre l'opció següent:

```
cvs add -kb fitxer(s)
```

En aquest cas, no gaudirem d'algunes característiques de CVS (com ara la barreja automàtica de versions), perquè només són possibles amb fitxers de text.

L'operació `add` sol marcar els fitxers per ser enviats, però no els envia. La tramesa es duu a terme amb una operació de *check-in*. Sense paràmetres, s'efectuarà un *check-in* recursiu de tots els fitxers i directoris marcats perquè s'enviïn al dipòsit:

```
cvs ci
```

Com totes les operacions d'escriptura, ens sol·licitarà que expliquem els detalls dels fitxers que enviem.

2.2.6. Modificació dels fitxers. Fusió de canvis

En acabar de preparar les nostres modificacions d'un fitxer de la nostra àrea de treball, les podem enviar al dipòsit mitjançant l'operació de *check-in*:

```
cvs ci fitxer(s)
```

En altres sistemes de control de versions, l'obtenció de fitxers en l'àrea de treball (*check-out*) bloqueja les modificacions a altres usuaris. En canvi, en CVS no és així. Per tant, és probable que en efectuar el nostre *check-in*, el sistema detecti que un altre desenvolupador també ha modificat el fitxer. En aquest cas ens avisarà de la situació:

```
$ cvs ci main.c
cvs commit: Examining
cvs commit: Up-to-date check failed for 'main.c'
cvs [commit aborted]: correct above errors first!
```

Aquí s'haurà de fer una fusió del nostre treball amb les modificacions efectuades per altres desenvolupadors. Per a això utilitzarem l'ordre:

```
cvs update
```

En actualitzar (*update*), poden passar dues coses:

1. Que CVS detecti que les modificacions són en seccions diferents del fitxer. En aquest cas, CVS és capaç de fusionar automàticament els nostres canvis amb els dels altres desenvolupadors.
2. Que CVS no sigui capaç de fusionar els canvis automàticament. En aquest cas, es produeix un conflicte.

CVS ens avisarà de la situació de conflicte en efectuar l'actualització:

```
$ cvs update
cvs update: Updating
RCS file: /home/devell/cvsrepos/projecte1/main.c,v
retrieving revision 1.1
retrieving revision 1.2
Merging differences between 1.1 and 1.2 into main.c
rcsmmerge: warning: conflicts during merge
cvs update: conflicts found in main.c
C main.c
```

Els conflictes figuraran a la nostra còpia del fitxer, de manera que els puguem editar i resoldre a mà.

Exemple

El nostre fitxer podria tenir el conflicte següent:

```
#include <stdlib.h>

void main()
{
<<<<<< main.c
  puts("Hello world!"); puts("Good bye!");
=====
  puts("Hello world!");
>>>>>> 1.2
}
```

Les nostres modificacions són el tros entre '`<<<<<< main.c`' i '`=====`', que entren en conflicte amb les que vénen al CVS en el nostre cas, concretament de la revisió 1.2.

Corregirem el conflicte sobre el fitxer anterior. Una vegada conformes amb el resultat, mitjançant una operació *check-in* normal sincronitzarem la nostra versió local i la del dipòsit.

Encara que les operacions de fusió automàtiques no solen fallar, quan el sistema ens avisi d'un error, convé revisar el fitxer per si s'han introduït malament.



En general, és bona pràctica efectuar una operació d'actualització abans d'editar el fitxer per si un altre desenvolupador ha introduït els canvis. Però quan treballen diversos desenvolupadors en paral·lel, és pràcticament impossible evitar la necessitat d'un *merge*.

2.2.7. Eliminació de fitxers

És possible eliminar fitxers del dipòsit. Per a això utilitzarem l'ordre:

```
cvs rm fitxer(s)
```

Seguida del corresponent *check-in*.

Si els fitxers no s'han esborrat encara del nostre espai de treball, caldrà afegir el modificador `-f` a l'ordre.

Cal assenyalar que els fitxers realment no s'esborren del dipòsit, sinó que es mouen en un directori especial, anomenat `Attic`. Això ens permetrà recuperar qualssevol de les versions dels fitxers esborrats en el futur, si volem.

2.3. Operativa avançada al CVS

Amb les operacions indicades en la secció anterior, es pot treballar amb un control de versions bàsic, sense preocupar-nos de temes

avançats com etiquetes o branques. Encara que no és un objectiu d'aquest document aprofundir en aquests temes, plantejarem una introducció a la gestió d'etiquetes i branques en CVS.

2.3.1. Versions

Cada vegada que efectuem un *check-in* o un *commit* al CVS, s'assigna un número de versió al fitxer. Podem veure el número de versió al codi font del fitxer si hi incloem la macro `$(Id)`. En incloure aquesta macro com a comentari del llenguatge en què hi ha el fitxer de codi font i fer el *check-in*, la macro s'expandeix i ens dona diverses informacions, com ara el número de versió assignat.

Exemple

```
/* $(Id): main.c,v 1.4 2004/12/30 22:09:57 devel1 Exp $ */
```

Potser el més interessant d'aquesta línia de comentari sigui saber la versió que manegem (1.4), la data de la tramesa i i l'usuari que la va fer (devel1).

Es pot recuperar una versió concreta d'un fitxer mitjançant l'ordre següent:

```
cvs update -r versió fitxer
```

També podem escollir el número de versió que volem enviar, sempre que sigui més alt que els que ja s'han utilitzat (per tant, no podem sobre escriure una versió, sinó que sempre hi haurà una versió nova al dipòsit):

```
cvs ci -r versió fitxer
```

Nota

Per a més informació, remetem a la bibliografia següent, que es pot consultar íntegrament a la Xarxa:

- *Manual de GNU*:
http://www.gnu.org/software/cvs/manual/html_chapter/cvs_toc.html
- *OpenSource Development with CVS*:
<http://cvsbook.red-bean.com/>

En un altre cas, el *check-in* sempre buscarà un número de versió automàticament.

Exemple

La història d'un fitxer típic pot ser la següent:

1.1 → 1.2 → 1.3 → 3.0 → 3.1 → 3.2 ...

En aquest cas, la versió 3.0 l'hem especificat nosaltres, i les altres s'han generat automàticament.

2.3.2. Etiquetes

No n'hi ha prou amb els números de versió per a identificar els fitxers. Suposem que volem publicar una versió beta del projecte que desenvolupem perquè el públic la provi. En general, un projecte té desenes de fitxers, els uns amb més versions i els altres amb menys. La tasca d'empaquetar una versió de cada fitxer per a conformar la versió beta que volem publicar requerirà una feina tediosa de localització i extracció de cada versió dels fitxers.

En qualsevol sistema de control de versions és possible etiquetar versions de fitxers amb un identificador comú.

El més senzill és etiquetar les versions de cada fitxer que tenim a l'espai de treball en cada moment. Per a això, al directori arrel del projecte llançarem l'ordre següent:

```
cvs tag etiqueta
```

Aquesta ordre posa l'etiqueta triada en cada fitxer de l'espai de treball. En la figura 2.1 veiem les versions que es podrien haver etiquetat.

Figura 2-1. Etiquetatge de fitxers per versions

File A	File B	File C	File D	File E
			1.1	
			1.2	
			1.3	
			1.4	
			1.5	
			1.6	
			1.7	
	1.1		1.8	
	1.2		1.9	
	1.3		1.10	1.1
	1.4		1.11	1.2
	1.5		1.12	1.3
	1.6		1.13	1.4
	1.7	1.1	1.14	1.5
	1.8	1.2	1.15	1.6
1.1	1.9	1.3	1.16	1.7
1.2	1.10	1.4	1.17	1.8
1.3	1.11	1.5		1.9
		1.6		1.10

L'operació d'etiquetatge de fitxers és una de les poques d'escriptura al CVS que no ens demana explicar-ne la causa i l'edita en un fitxer de registre de sessió (*log*). Tanmateix, per aquesta mateixa raó convé que l'etiqueta sigui com més descriptiva millor. És una bona pràctica incloure-hi, també, la data.

Una bona etiqueta pot ser *Release-Beta1-20041110*, que ens indicaria que és l'alliberament de la versió Beta1 i inclou, a més, la data de la publicació.

Una vegada etiquetades, l'accés a les versions es fa com si se sol·licités una revisió concreta; és a dir, mitjançant el modificador `-r`. Així doncs, si un usuari anònim vol obtenir aquesta versió Beta1 del CVS, llançarà l'ordre següent:

```
cvs -r Release-Beta1-20041110 co projectel
```

2.3.3. Branques

Finalment, ens introduïrem en el maneig de les **branques** de CVS. Vegem en primer lloc un exemple pràctic.

Suposem que hem alliberat una versió del nostre projecte. A partir d'ara, només ens interessa corregir errors de seguretat i altres errors greus, i deixem aparcades les noves funcionalitats per a un altre alliberament. Això es pot gestionar fàcilment al CVS mitjançant les branques.



En un moment determinat es pot marcar un inici de branca als fitxers del projecte. A partir d'aquell moment, el desenvolupador pot triar entre modificar els fitxers de la branca principal o bé els de l'altra branca. És a dir, apareixen dos desenvolupaments independents.

Seguint amb el nostre exemple, en algun moment voldrem integrar les funcionalitats noves afegides al projecte amb les correccions de seguretat que s'han anat implementant en l'altra branca. És el moment de tornar a unir les dues branques per mitjà d'una operació de fusió.

La definició i l'accés a les branques amb el CVS es fa de manera molt similar a l'etiquetatge. El més senzill és crear una branca a partir de l'última versió de treball de cada fitxer, amb l'ordre següent:

```
cv$ tag -b nom-branca
```

Després d'aquesta operació, el nostre espai de treball continuarà a la branca principal. En canvi, un altre desenvolupador pot accedir a qualsevol de les dues branques per treballar-hi. Per exemple, per a accedir a la branca en què hem treballat, escriurà:

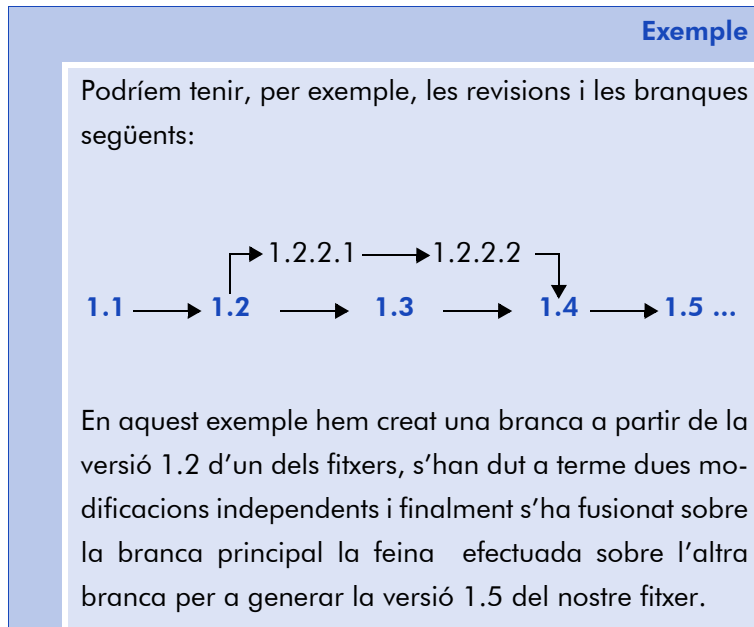
```
cv$ co projecte
```

Per accedir a la branca nova escriurà:

```
cv$ co -r nom-branca projecte
```

A partir d'aquí, quan faci *check-in*, els seus canvis s'enviaran a una branca o a una altra, segons la que s'hagi escollit.

Observeu que els números de versió a les branques noves tindran xifres noves, per a identificar en quina branca som.



La fusió de les dues branques és senzilla. Per exemple, per a fusionar el nostre espai de treball pertanyent a la branca principal amb la feina realitzada en la branca *pedaços-seguretat-1-0*, s'ordenarà l'operació següent:

```
cvs update -j pedaços-seguretat-1-
```

Aquesta fusió és freqüent en versions amb moltes diferències, de manera que és bastant probable que sorgeixin conflictes que hàgim de resoldre a mà, tal com ja hem vist.

Una vegada editat el fitxer per a resoldre el conflicte, enviarem els canvis (*check-in*) i així es generarà la nova revisió a la branca principal, que barreja els canvis de la branca de pedaços (la versió 1.5 de l'exemple anterior).

2.3.4. Informació sobre etiquetes i branques

L'ús de l'etiquetatge i les branques en un projecte pot complicar bastant l'elecció de les versions de feina. Amb CVS es pot obtenir infor-

mació sobre les branques i etiquetes de cada fitxer mitjançant l'ordre `status`.

Per exemple, si el nostre fitxer `main.c` ha tingut la història de revisions i branques de l'exemple anterior i acabem d'introduir la versió 1.2.2.2, aquesta ordre ens mostraria el següent:

```
$ cvs status -v main.c
=====
File: main.c Status: Up-to-date
Working revision: 1.2.2.2 Fri Dec 31 00:17:04 2004
Repository revision: 1.2.2.2/home/devell1/cvsrepos/proyecto1/main.c,v
Sticky Tag: pedaços-seguretat-1-0 (branch: 1.2.2)
Sticky Date: (none)
Sticky Options: (none)
Existing Tags:
    pedaços-seguretat-1-0 (branch: 1.2.2)
```

D'aquesta manera sabem que treballem amb una versió de la branca *pedaços-seguretat-1-0* (pel número de revisió 1.2.2.x). L'ordre `status` vista des de la branca principal ens mostrarà també aquesta etiqueta, de manera que sabrem amb quines etiquetes hem de sol·licitar a CVS la fusió del codi.

2.4. Subversion: la pròxima generació

Subversion és la generació següent de control de versions que previsiblement conquerirà pròximament el món del programari lliure [Collins04]. L'objectiu principal dels seus creadors era tenir un sistema de control de versions que solucionés les principals deficiències de CVS, però que al seu torn es basés en la filosofia i manera de treballar del CVS per facilitar la migració d'un sistema a un altre.

Subversion ofereix les característiques addicionals següents respecte a CVS:

- **Versionar directoris.** Amb CVS només els fitxers tenen versions.

- **Control real de versions.** Així, per exemple, es poden moure els fitxers de directori en directori, mentre que en CVS aquesta tasca només es pot fer al dipòsit.
- **Commits atòmics.** D'aquesta manera s'eliminen problemes de concurrència, entre d'altres.
- **Metadades.** Subversion admet dades addicionals tant sobre els fitxers com sobre la seva versió.
- **Diferents modes d'accés.**
- **Consistència de dades.** Subversion no diferencia entre fitxers de text i fitxers binaris.

3. Sistemes de seguiment d'errors

Els sistemes de seguiment d'errors (*bug tracking systems* o BTS, en anglès) són eines destinades a la gestió automàtica de notificació d'errors i la seva correcció.

Quan un usuari autoritzat detecta un error en un programari, acudirà al BTS per informar-ne i el BTS ho notificarà de manera automàtica al responsable del programari. Els programadors poden tenir accés al sistema, la qual cosa els permet conèixer els informes de treball que tenen oberts, i també notificar-ne la resolució.



El sistema de seguiment d'errors (BTS) notifica de manera automàtica (per correu electrònic) els canvis d'estat d'un informe a les persones que hi puguin estar interessades: l'usuari que ha notificat l'error, els programadors que l'han de corregir o un tercer (per exemple, l'equip responsable del control de qualitat de l'empresa de desenvolupament).

En resum, un sistema de seguiment d'errors ha de tenir les característiques següents:

- Capacitat d'emmagatzemar un històric d'errors, i dins de cadascun, els seus diferents estats.
- Capacitat d'assignar diversos atributs d'interès a cada error, com la plataforma sobre la qual corre, que admeti la classificació per severitat o prioritat.
- Capacitat de tramesa automàtica de notificacions de canvi a les parts afectades.

Nota

Hem vist les aplicacions de tipus SourceForge en el capítol 1.

- Sistema de permisos que autoritzi i desautoritzi determinades operacions a determinats tipus d'usuari.
- Possibilitat (desitjable) de generar algun tipus d'estadístiques sobre els errors emmagatzemats.

Hi ha en el mercat bastants variants de BTS, algunes com a programari lliure i d'altres com a programari propietari. Altres vegades formen part de paquets més grans, com el que s'inclou en les aplicacions de tipus SourceForge i que ja hem vist breument. De vegades, el BTS té una interfície de gestió basada en web o en una aplicació client, la qual cosa en facilita el maneig.

En aquest capítol veurem el sistema de seguiment d'errors més utilitzat entre els grans projectes de programari lliure: Bugzilla.

3.1. Seguiment d'errors amb Bugzilla

Bugzilla és un sistema de seguiment d'errors molt popular, probablement perquè és fàcil d'instal·lar i gestionar, i perquè té una certa maduresa i flexibilitat [Bugzilla04]. Encara que potser també ho és perquè és el que utilitzen projectes com Mozilla, GNOME o KDE.

Bugzilla s'administra gairebé per complet mitjançant la seva interfície web, i en cas que no l'hàgim d'instal·lar a mà, és estrany que hàgim d'executar tasques administratives des de la línia d'ordres (tasques que se centrarien en modificacions directes de la seva base de dades, o dels mateixos guions de Bugzilla).

Bugzilla està escrit en llenguatge Perl i ha estat dissenyat de manera que separa els guions i les plantilles de la interfície web, i així es pot adaptar fàcilment a un entorn concret.

3.1.1. Els errors en Bugzilla

Abans de continuar, ens detindrem breument a analitzar com organitza Bugzilla els errors. En aquest BTS, els errors formen part d'un component, i cada component forma part d'un producte.

Així doncs, a l'hora d'organitzar el nostre BTS, hem de crear un producte per cada sistema de programari que estigui sota el nostre control i, dins de cada producte, dividirem els diferents subsistemes per definir els components.

Cada component tindrà un propietari (a qui inicialment s'assignen tots els errors d'aquest component) i les versions són comunes a tots els components d'un producte.

Normalment, una aplicació s'assimilarà, en el sentit de la seva supervisió, a un producte, de manera que es crearà un component per cada subsistema, que estarà coordinat per un responsable.

Altres aspectes de la gestió d'errors, com els diferents nivells de severitat, els diferents estats d'error possibles o les plataformes, s'han d'establir per edició directa dels guions i la base de dades, la qual cosa se surt del propòsit d'aquest material.

3.2. Instal·lació i configuració de Bugzilla

En els apartats següents s'explica com s'efectua una instal·lació completa de Bugzilla, i com es creen els comptes necessaris per a poder començar a treballar amb aquest BTS.

3.2.1. Instal·lació de Bugzilla

El millor és deixar que el paquet s'instal·li automàticament, ja que la configuració íntegrament manual és molt laboriosa.

Assumim que tenim instal·lat Debian (Sorge o posterior). Abans d'instal·lar Bugzilla ens assegurarem que està instal·lat i configurat MySQL, i que coneixem la contrasenya de l'usuari *root* de MySQL (que és l'usuari amb capacitat administrativa en tot el sistema de base de dades). Per defecte, la contrasenya d'aquest usuari està en blanc, per la qual cosa és recomanable que s'hagi canviat abans.

Exemple

Possiblement, en gairebé qualsevol aplicació podrem tenir un subsistema d'interfície d'usuari, un altre de base de dades, etc.

Si es compleixen les condicions anteriors, instal·larem Bugzilla amb l'ordre següent:

```
apt-get install bugzilla
```

El programa d'instal·lació ens preguntarà si volem fer una instal·lació assumint usuaris i valors per defecte o si, al contrari, els volem proporcionar nosaltres. Aquesta última opció és millor, sobretot si hem canviat alguns valors predeterminats de la instal·lació de MySQL, o simplement si hem posat una contrasenya a l'usuari administratiu de la base de dades.

El sistema ens preguntarà per l'usuari administratiu (que serà *root*), la seva contrasenya, el servidor de base de dades (*localhost*), el port (acceptarem el valor predeterminat, 3306), l'usuari que volem en MySQL per a gestionar la base de dades del sistema Bugzilla (*bugs*), el nom de la base de dades (també *bugs*), el correu electrònic de l'administrador de Bugzilla (que rebrà notificacions i, per tant, ha d'existir; a més, també és l'identificatiu d'administrador del sistema Bugzilla), el nom de l'administrador i la contrasenya que vulguem donar a l'administrador de Bugzilla.

Bugzilla s'instal·larà, crearà la base de dades i quedarà instal·lat i llest per a la seva configuració.

3.2.2. Configuració de Bugzilla

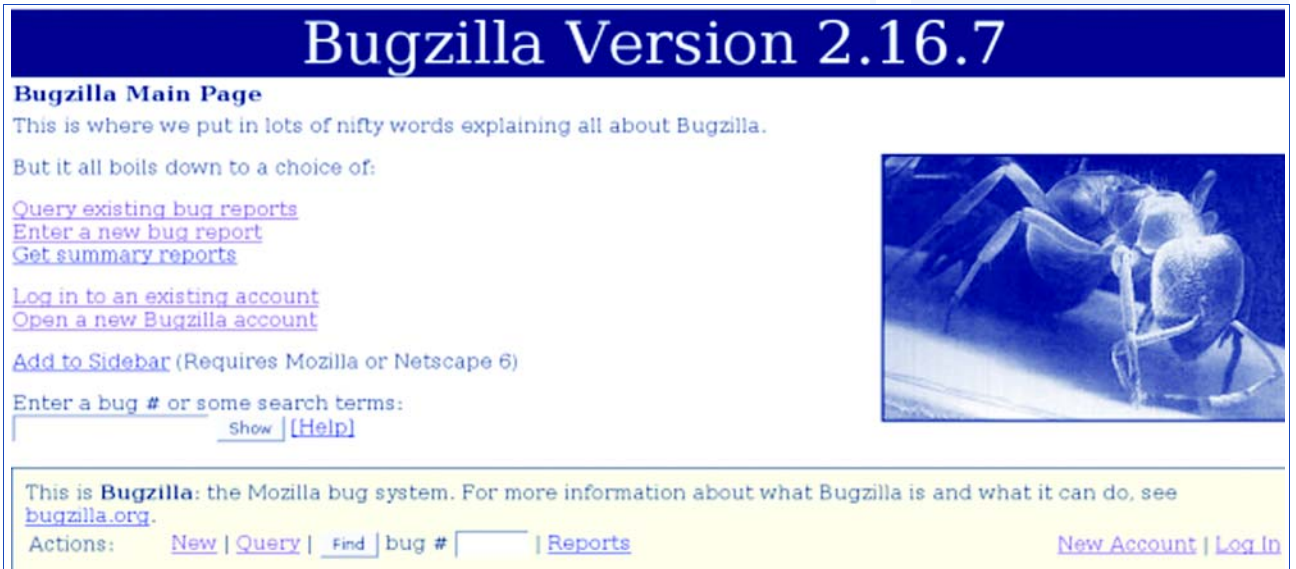
Si hem seguit els passos anteriors, tindrem un sistema Bugzilla instal·lat, funcional però amb alguns aspectes pendents de configuració.

La resta de la configuració es duu a terme des de la interfície principal de Bugzilla, a la qual s'accedeix per un navegador. Obrirem un navegador amb l'adreça:

```
http://nom-servidor/bugzilla
```

Ens apareixerà la pantalla principal de Bugzilla (figura 3.1).

Figura 3-1. Interfície principal de Bugzilla



Bugzilla Version 2.16.7

Bugzilla Main Page
This is where we put in lots of nifty words explaining all about Bugzilla.
But it all boils down to a choice of:

[Query existing bug reports](#)
[Enter a new bug report](#)
[Get summary reports](#)

[Log in to an existing account](#)
[Open a new Bugzilla account](#)

[Add to Sidebar](#) (Requires Mozilla or Netscape 6)

Enter a bug # or some search terms:
 [Show](#) | [\[Help\]](#)

This is **Bugzilla**: the Mozilla bug system. For more information about what Bugzilla is and what it can do, see bugzilla.org.

Actions: [New](#) | [Query](#) | [Find](#) | bug # | [Reports](#) [New Account](#) | [Log In](#)

Farem clic sobre 'Log In' a la part inferior i introduïrem com a usuari el correu electrònic d'administrador i, com a contrasenya, la que hem proporcionat durant la instal·lació.

A continuació, se'ns mostrarà una pantalla de consulta d'errors, com un formulari ple d'opcions. Per ara ens desplaçarem cap a la part inferior de la pàgina fins a un quadre com el de la figura 3.2.

Figura 3-2. Quadre d'accés a les accions de Bugzilla



This is **Bugzilla**: the Mozilla bug system. For more information about what Bugzilla is and what it can do, see bugzilla.org.

Actions: [New](#) | [Query](#) | [Find](#) | bug # | [Reports](#) | [My Votes](#) [Edit prefs](#), [parameters](#), [users](#), [products](#), [attachment statuses](#), [groups](#), [keywords](#) | [Sanity check](#) | [Log out](#)

Preset Queries: [My Bugs](#)

Farem clic sobre 'Parameters' i emplenarem el formulari que apareixerà a continuació. Hem d'introduir tots els valors, però específicament el del nom del responsable del BTS ('Bugzilla Maintainer').

Abans de crear els comptes d'usuari, s'han de configurar els grups. Els **grups** no són més que patrons que s'utilitzen per a assignar per-

misos als nous usuaris. No aprofundirem gaire en aquest tema i simplement configurarem el nostre BTS perquè els nous usuaris puguin editar només les notificacions que siguin pròpies. Per a això, seleccionarem l'enllaç 'groups' i en la pàgina que ens mostrarà, deixarem en blanc tots els patrons de 'User RegExp' que apareixen per als camps abans de pitjar el botó d'enviar (figura 3.3).

Figura 3-3. Interfície de configuració de grups

Bit Name Description User RegExp Active Action

Submit changes Add Group

Submit changes

Name is what is used with the `UserInGroup()` function in any customized cgi files you write that by email to limit a bug to a certain groupset.

Description is what will be shown in the bug reports to members of the group where they can group.

User RegExp is optional, and if filled in, will automatically grant membership to this group to matches this regular expression.

The **Active** flag determines whether or not the group is active. If you deactivate a group it will although bugs already in the group will remain in the group. Deactivating a group is a much longer group would be.

In addition, the following groups that determine user privileges exist. You can only edit the User RegExp duplicate the Names of any of them in your user groups.

Also please note that both of the Submit Changes buttons on this page will submit the changes for convenience.

Bit	Name	Description	User RegExp
1	tweakparams	Can tweak operating parameters	
2	editusers	Can edit or disable users	
4	creategroups	Can create and destroy groups.	
8	editcomponents	Can create, destroy, and edit components.	
16	editkeywords	Can create, destroy, and edit keywords.	
32	editbugs	Can edit all aspects of any bug.	
64	canconfirm	Can confirm a bug.	

Submit changes

This is Bugzilla, the Mozilla bug system. For more information about what Bugzilla is and how to use it, see the Bugzilla documentation.

A continuació, pot ser una bona idea crear comptes d'usuari, almenys amb els responsables del programari que supervisarem. Seleccionarem l'enllaç 'Users' de la pàgina anterior, i a dins, l'enllaç 'Add a new user'. Ens apareixerà la pantalla següent (figura 3.4).

Figura 3-4. Interfície d'alta de nous usuaris

Add user

Login name:

Real name:

Password: (enter new password to change)

Disable text:

If non-empty, then the account will be disabled, and this text should explain why.

Groups and Privileges:

The new user will be inserted into groups based on their userregexps. To change the group permissions for this user, you must edit the account after creating it.

Add

El nom d'usuari ('Login name') serà el correu electrònic del desenvolupador, pel qual rebrà les notificacions. Hem d'introduir aquest valor, el camp 'Real Name' i una contrasenya inicial, i deixar buit el quadre de text 'Disable text'.

Amb aquesta interfície haurem de crear, almenys, els comptes dels desenvolupadors que hagin de ser responsables inicials de cada component de programari que se supervisi. Més endavant, els mateixos interessats ja els podran crear directament des de la interfície web.

A continuació, seleccionarem 'Products' i crearem un producte per cada sistema de programari que hem de supervisar. A dins de cada producte, en podem editar els components (subsistemes) i les versions. Recordem que cal assignar un desenvolupador inicial per a cada component, a qui s'assignarà tots els errors que s'enviïn per a aquest component (i els rebrà).

Fet això, podem donar el nostre BTS per configurat. A partir d'aquest moment, cal que els desenvolupadors i usuaris el coneguin i ens comencin a enviar errors pel sistema.

3.3. Notificació d'errors

Un BTS per a programari lliure ha d'acceptar errors enviats per qualsevol usuari, encara que inicialment no en sapiguem res. En Bugzilla no crearem nosaltres els comptes d'usuari, sinó que els mateixos usuaris els podran crear la primera vegada que ens vulguin enviar un error. L'administrador de Bugzilla crearà directament només els comptes dels desenvolupadors necessaris per a adjudicar els errors.

3.3.1. Creació de comptes

Probablement, un usuari que accedeixi al nostre BTS des de l'adreça web de la seva interfície voldrà començar per notificar un error. Tanmateix, en intentar-ho, se li sol·licitarà donar-se d'alta en un compte. Així doncs, farà clic sobre l'enllaç corresponent i emplenarà el formulari que hi aparegui (com a dada imprescindible ha de donar una adreça de correu electrònic, que s'utilitzarà per a enviar-li una con-

trasenya d'accés, i també les notificacions posteriors que li corresponguin).

Figura 3-5. Interfície de creació d'un compte d'usuari

Create a new Bugzilla account

To create a Bugzilla account, all that you need to do is to enter a legitimate e-mail address. The account will be created, and its password will be mailed to you. Optionally you may enter your real name as well.

E-mail address:

Real name:

3.3.2. Notificació d'un error

L'usuari utilitzarà la seva contrasenya acabada d'adquirir per a accedir al sistema des de l'enllaç 'Login' que apareix al quadre inferior de la pàgina (figura 3.2). Una vegada a dins, accedirà de nou al quadre inferior i buscarà l'enllaç 'New'. Se li mostrarà el formulari per a crear l'informe d'error (figura 3.6).

Figura 3-6. Interfície de notificació d'errors

Enter Bug This page lets you enter a new bug into Bugzilla.

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

Reporter: Product: TrainControl
 Version: Component:
 Platform: OS:
 Priority: Severity:
 Assigned: (Leave blank to assign to default component owner)
 To:
 Cc:
 URL:
 Summary:
 Description:

A l'hora de crear l'informe d'error, l'usuari ha de seguir unes normes amb l'objectiu de descriure'l de la manera més concisa i útil possible. És per això que haurà d'assenyalar el producte, el component i la versió exactes que li han produït l'error, i també (si és aplicable i possible) la plataforma i el sistema operatiu. Emplenarà els camps de prioritat i severitat amb el seu criteri, però preferiblement consultarà altres errors previs per escollir els valors adequats per a aquest cas. Deixarà en blanc els camps d'Assignació a' i 'Còpia a' perquè el sistema els doni un valor per defecte.

El camp de la URL no sol ser aplicable. En cas de ser-ho (per exemple, URL de l'aplicació que origina l'error), s'ha de proporcionar. El resum haurà de ser com més concís millor (una frase descriptiva). La descripció és un text lliure, però és desitjable que contingui la informació següent:

- **Resum de l'error.** És un text que explica de la manera més detallada possible l'error detectat (quatre o cinc línies).
- **Passos per a reproduir l'error.** Si l'error és reproducible, hem d'indicar els passos que hem seguit per provocar-lo.
- **Resultats esperats i obtinguts.** Hem d'incloure els resultats que hem observat en contrast amb els que esperàvem.
- **Informació addicional.** Potser volem introduir informació addicional sobre com es reproduceix l'error (per exemple, versions d'algun programari que hagi d'estar instal·lat) o que faciliti la feina del desenvolupador que l'hagi d'analitzar i corregir.



Hem de notificar l'error de la manera més breu i precisa possible, la qual cosa contribuirà sens dubte que l'error es corregeixi en un termini de temps més curt.

Quan premem el botó de tramesa, es registrarà l'error i s'enviarà una notificació al responsable del component escollit.

Abans de decidir-nos a enviar un error, és recomanable revisar, en tant que sigui possible, que no s'hagi enviat ja o s'hagi corregit. Per a això podem fer una cerca (tal com es mostrarà en la secció següent) i potser procurar tenir instal·lada l'última versió del programari que ens ha donat l'error. Tanmateix, quan el nombre d'errors emmagatzemats al sistema és molt elevat, resulta molt difícil no repetir els mateixos errors que altres persones, si bé el desenvolupador els detectarà aviat i marcarà cada repetició com a 'Duplicat'. Els grans projectes de programari lliure que utilitzen Bugzilla solen tenir tanta quantitat d'informes d'errors que sovint tenen una persona dedicada exclusivament a gestionar-los [Villa03]. Es tracta d'una tasca molt àrdua, però molt agraïda pels desenvolupadors, ja que, si es fa bé, els permet classificar els errors i dedicar-se directament a corregir-los.

3.4. Cerca i tractament d'errors

Bugzilla no solament permet informar dels errors, sinó que és una excel·lent eina per a fer-ne el seguiment. Un dels objectius que van motivar-ne la creació va ser facilitar la tasca dels desenvolupadors perquè poguessin identificar d'una manera senzilla i ràpida els errors del seu programari i el seu grau de prioritat.

A més, cada error es pot assignar a un altre desenvolupador o es pot canviar el seu estat. A continuació veurem com es duen a terme aquestes accions amb Bugzilla.

3.4.1. Cerca d'errors

La pàgina de cerques de Bugzilla es pot consultar amb compte d'usuari o sense (encara que recordem que només en podrem modificar el contingut amb un compte). Si al quadre inferior de qualsevol pàgina de Bugzilla seleccionem l'enllaç 'Query', se'ns mostrarà un formulari amb molts camps, que emplenarem per afinar la cerca tant com sigui possible. En la figura 3.7 mostrem una part de la pàgina de cerques.

Figura 3-7. Interfície de cerques de Bugzilla

The screenshot shows the 'Search for bugs' interface in Bugzilla. It includes several sections for filtering search results:

- Summary:** A text input field with a dropdown menu set to 'contains all of the words/strings' and a 'Search' button.
- Product:** A dropdown menu showing 'TrainControl'.
- Component:** A dropdown menu showing 'CommunicationSubsystem'.
- Version:** A dropdown menu showing '1.0', '1.1pre', and 'unspecified'.
- A comment:** A text input field with a dropdown menu set to 'contains all of the words/strings'.
- The URL:** A text input field with a dropdown menu set to 'contains all of the words/strings'.
- Status:** A dropdown menu with options: UNCONFIRMED, NEW, ASSIGNED, REOPENED, RESOLVED, VERIFIED, CLOSED.
- Resolution:** A dropdown menu with options: FIXED, INVALID, WON'TFIX, LATER, REMIND, DUPLICATE, WORKSFORME.
- Severity:** A dropdown menu with options: blocker, critical, major, normal, minor, trivial, enhancement.
- Priority:** A dropdown menu with options: P1, P2, P3, P4, P5.
- Hardware:** A dropdown menu with options: All, DEC, HP, Macintosh, PC, SGI, Sun.
- OS:** A dropdown menu with options: All, Windows 3.1, Windows 95, Windows 98, Windows ME, Windows 2000, Windows NT.
- Email and Numbering:** Two columns of checkboxes for 'Any of:' including 'bug owner', 'reporter', 'CC list member', and 'commenter'. Below are dropdown menus for 'contains'.
- Bug Changes:** Two sections: 'Only bugs changed in the last days' and 'Only bugs where any of the fields' (with a dropdown menu for '[Bug creation]', 'assigned_to', 'bug_file_loc', 'bug_severity') 'were changed between and'.
- Only include:** A dropdown menu and a text input field for 'bugs numbered:'.

En fer la cerca, el sistema ens contestarà amb una llista d'errors, possiblement dividida en diverses pàgines. Quan hàgim trobat el que busquem, podem seleccionar-ne l'enllaç i arribem a la pàgina de visualització i edició de l'error (figura 3.8).

Figura 3-8. Interfície de visualització i edició d'errors

The screenshot displays a bug tracking page for bug 216181. The main form includes the following fields and information:

- Bug#:** 216181 alias: [input field]
- Product:** Mozilla Application Suite
- Component:** Composer
- Status:** VERIFIED
- Resolution:** DUPLICATE of bug 46845
- Assigned To:** This bug is awaiting triage to an appropriate owner or component -<general@browser.bugs>
- QA Contact:** general@browser.bugs
- URL:** [input field]
- Summary:** selected item in Combo / Drop-down list not refreshed correctly after reload.
- Keywords:** [input field]
- Hardware:** PC
- OS:** All
- Version:** 1.4 Branch
- Priority:** --
- Severity:** major
- Reporter:** Jerome Lacoste <offi...>
- Add CC:** [input field]
- CC:** [input field]
- Target Milestone:** --
- Flags (Help):** blocking-awary1.0-L10N, blocking1.7.5, blocking1.7.6, blocking1.8a8

Below the main form, there is a table for attachments:

Attachment	Type	Created	Size	Flags	Actions
Create a New Attachment (proposed patch, testcase, etc.) View All					

Additional information includes:

- Bug 216181 depends on:** [input field] [Show dependency tree](#)
- Bug 216181 blocks:** [input field] [Show dependency graph](#)
- Votes:** 0 [Show votes for this bug](#) [Vote for this bug](#)
- Additional Comments:** [input field]
- Actions:**
 - Leave as VERIFIED DUPLICATE
 - Reopen bug
 - [Cancel](#)
- [View Bug Activity](#) | [Format For Printing](#)

The description at the bottom reads: "I have a web page that is generated by a servlet. This page contains comments among other text fields and drop down lists. The items in the drop down list are selected. I modify the BB in which the values used for the page are stored using an external resource. I refresh the data using the CTRL or refresh button. Result:

3.4.2. Estats d'un error

Una de les característiques que més ens interessin en consultar un error és el seu estat. L'estat d'un error és el camp que ens resumeix com ha quedat l'error, si ha estat corregit o no, etc.

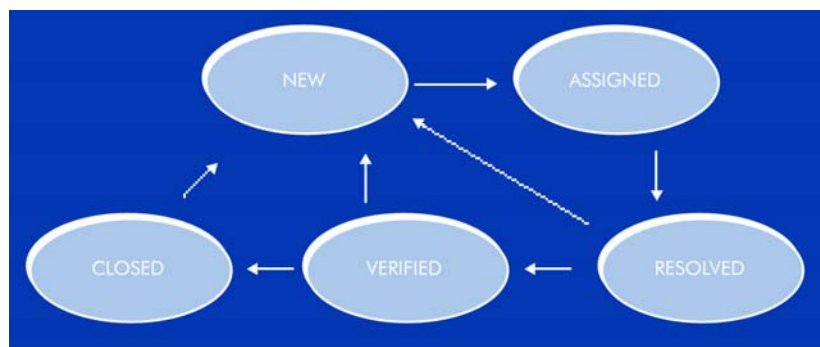
Encara que els estats es poden personalitzar, els més habituals són:

- NEW (nou). És l'estat que té un error quan l'acaba d'enviar un usuari.
- ASSIGNED (assignat). Aquest estat s'assigna quan un desenvolupador ha acceptat i es responsabilitza de la correcció.

- c) **RESOLVED** (resolt). Indica que l'error s'ha resolt. Associat a aquest estat hi ha les causes. Les més habituals són:
- **FIXED** (corregit). L'error s'ha corregit.
 - **DUPLICATE** (duplicat). S'ha detectat que hi ha un informe d'aquest error fet en un altre moment, i amb això s'anota que aquest error és idèntic a un altre (s'acompanya del número de l'altre error).
 - **INVALID** (invàlid). L'error no és tal. Generalment si se selecciona aquest estat, el desenvolupador ha d'explicar clarament per què no ho és.
 - **WONTFIX** (no es corregirà). És un error que per alguna causa no es corregirà, almenys en la versió actual.
- d) **VERIFIED, CLOSED** (verificat, tancat). Són estats que indiquen que l'error s'ha resolt i que posteriorment se'n va verificar la resolució o bé definitivament es va tancar.
- e) **REOPENED/NEW** (reobert). Un error es reobre quan es comprova que l'error no s'ha corregit realment, malgrat que s'ha marcat com a *corregit*.

En la figura 3.9 podem veure el cicle de vida d'un error.

Figura 3-9.



Com es pot observar, quan es resol un error, sigui quin sigui l'estat de resolució, es pot reobrir i passar de nou a l'estat NEW (sense assignació). Tanmateix, és millor no tornar a obrir els errors que s'han verificat o s'han tancat. Normalment, seran els membres de l'equip de control de qualitat els que tancaran els errors definitivament i dei-

xaran la verificació a l'usuari que va enviar l'error originalment. Tanmateix, sovint els usuaris no s'ocupen de marcar els errors verificats, la qual cosa en deixa centenars en estat de *resolt*, sense que es tanquin en molt de temps.

3.4.3. Edició d'un error

Quan som a la pàgina corresponent a un error, podem efectuar, segons el tipus de permisos que tinguem, diversos tipus de canvis:

- **Afegir-hi un comentari.** Qualsevol usuari, en general, pot fer comentaris sobre qualsevol error, per exemple, aclarir els passos per a reproduir-lo o donar possibles pistes sobre la seva resolució.
- **Canvi d'estat.** Això només ho ha de fer el desenvolupador que té assignat l'error (per a marcar l'error com a *resolt*, per exemple) o el mateix usuari (per a reobrir un error erròniament tancat, per exemple).
- **Reassignar.** És possible canviar el desenvolupador que té assignada la resolució d'un error. Normalment ho farà el responsable del projecte o bé algun dels desenvolupadors involucrats.
- **Canvi de dades bàsiques.** Canviar el resum o recol·locar l'error en un altre producte o component no es fa gaire sovint, només en cas d'error evident.

Tots els usuaris poden editar, de manera predeterminada, qualsevol atribut d'un error, fins i tot marcar-lo com a corregit. És una bona pràctica que l'administrador modifiqui els permisos dels usuaris per impedir-ho, tal com ja indiquem en editar els grups d'usuaris. Així, només els usuaris que notifiquen un error i els desenvolupadors que tenen l'error assignat el podran modificar, encara que tots podran introduir-hi comentaris en qualsevol moment.

4. Llistes de correu electrònic

Les llistes de correu electrònic són eines molt utilitzades en qualsevol comunitat a Internet, no solament en la de programari lliure. Qualsevol afició a la vida real pot portar a crear una comunitat virtual a Internet basada, gairebé sempre, en una llista de correu o en un fòrum en què estan subscriptes els membres d'aquesta comunitat.

La necessitat de les llistes de correu en el món del programari lliure, moltes vegades, resulta estratègica per aconseguir la divulgació d'un projecte: d'una banda, en començar el desenvolupament gairebé sempre necessitem una llista perquè els programadors puguin comunicar-se i coordinar-se. De l'altra, quan publiquem la nostra aplicació, segurament es crearà una llista per a afavorir la comunicació i l'intercanvi d'experiències entre els usuaris.

Per poc que creixi un projecte, sorgiran noves iniciatives, com ara traduir-lo a altres idiomes o crear documentació. Al voltant d'aquestes activitats sorgiran comunitats de persones que hi treballen.



Les comunitats que sorgeixen al voltant d'un projecte de desenvolupament de programari lliure es basaran, amb molta seguretat, en llistes de correu.

4.1. Què és una llista de correu

Podem definir una llista de correu com una aplicació basada en correu electrònic que té les característiques següents:

- a) Emmagatzema un conjunt d'adreces de correu electrònic anomenades *subscriptors*.

- b) Qualsevol missatge enviat a la llista es remet automàticament a tots els subscriptors.

A més, segons l'eina escollida, es podrà disposar d'algunes característiques més, entre les quals destaquem:

- Mecanismes automatitzats per a la gestió de la llista (és a dir, alta i baixa de subscriptors) basats en correu electrònic.
- Interfícies web per a la gestió de la llista.
- Arxiu automàtic dels missatges de la llista. Un sistema avançat és capaç d'arxivar tot el que s'envia a la llista i proporcionar algun mètode d'accés a aquests arxius (gairebé sempre, mitjançant una interfície web).
- Funcions avançades: permisos de tramesa, tramesa periòdica (*digest*), filtrada, etc.

4.2. Eines

En qualsevol servidor de correu electrònic normalment hi ha una eina per a posar en marxa immediatament una llista de correu: el fitxer d'àlies (que s'acostuma a trobar a `/etc/aliases`).

Un **fitxer d'àlies** maneja directament l'agent de correu (Sendmail, Postfix, etc.), i conté una llista de parells (àlies, adreces). Quan hi ha aquest fitxer, qualsevol missatge enviat a un dels seus àlies es remetrà automàticament a cadascuna de les adreces de l'àlies.

Exemple

Suposem, per exemple, que el nostre fitxer `/etc/aliases` fos el següent:

```
# Fitxer /etc/aliases

projecte1-devel:usuari1@domini1,
usuari2@domini2, usuari3@domini3
```

```
projectel-users:usuari4@domini4,  
usuari5@domini5, usuari6@domini6,  
usuari7@domini7, usuari8@domini8
```

Cada vegada que algú escrigui a `projectel-devel@elnostredomini.com` es remetrà una còpia del missatge a `usuari1@domini1`, `usuari2@domini2` i `usuari3@domini3`.

De la mateixa manera, quan algú escrigui a l'adreça `projectel-users@elnostredomini.com`, es remetrà una còpia del missatge a cadascun dels membres que apareixen en forma de llista al fitxer.

Les possibilitats de `/etc/aliases` acaben aquí, i per a afegir-hi o treure'n usuaris, l'administrador del sistema haurà d'editar el fitxer `i`, com que no hi ha gestió automàtica, gestionar-lo serà una tasca molt avorrida per a l'administrador per poc que creixi la comunitat. A més, els missatges no s'emmagatzemen a cap lloc (hi ha trucs per a fer-ho, però són poc flexibles), i no es poden establir permisos ni qualsevol altra funció mitjanament avançada. No obstant això, els fitxers d'aliases poden ser la solució per a comunitats petites.

Hi ha solucions més avançades que aquesta per a gestionar llistes de correu electrònic. Nosaltres ens centrarem en la més potent que hi ha dins del repertori del programari lliure, el gestor Mailman de llistes de correu GNU. Encara que n'hi ha uns altres de no menys populars com Sympa, Ecartis, FML, Majordomo o Listserv (aquests dos últims, per cert, amb llicències que no són lliures).

4.3. Alternatives

Abans de mostrar la configuració i gestió de Mailman, esmentarem altres alternatives utilitzades per a muntar comunitats d'usuaris o desenvolupadors:

- USENET. Els missatges als grups de USENET són similars als de correu electrònic, almenys pel que fa al format. Tanmateix, en

lloc de lliurar-los a les bústies dels usuaris, són els usuaris els que s'han de connectar al seu servidor local de USENET per obtenir-los. Fa temps, USENET va ser molt popular, malgrat la proliferació de sistemes similars basats en interfície web. Probablement, la invasió del correu brossa o *spam* que han experimentat aquestes solucions n'han fet disminuir enormement tant l'ús com la popularitat.

- Fòrums web. Són semblants a USENET, en el sentit que els missatges no s'envien a les bústies dels usuaris, sinó que són els usuaris que es connecten al fòrum per obtenir-los. Els fòrums web són bastant populars en algunes comunitats d'usuaris poc tècnics perquè eviten que l'usuari s'hi hagi de subscriure i fer altres gestions pròpies de la llista.
- Sistemes mixtos. De vegades, s'uneixen els avantatges del fòrum web amb els de la llista de correu electrònic. Són sistemes que permeten enviar els missatges del fòrum a les bústies de correu dels usuaris, però alhora permeten les consultes per web. I de la mateixa manera, els missatges a la llista es poden enviar tant per correu electrònic com des de la interfície web. Un dels serveis més populars d'aquesta mena és el que ofereix Yahoo Groups.

4.4. Llistes de correu amb Mailman

Tenir una llista de correu per al nostre projecte és, sens dubte, el més convenient, atesa la seva flexibilitat i facilitat de gestió. En els apartats següents instal·larem un sistema Mailman i presentarem les opcions de configuració més interessants.

4.4.1. Instal·lació

La instal·lació de Mailman pot ser completament manual; és a dir, és possible obtenir les fonts de la pàgina principal, compilar-les i configurar a mà tot el sistema [Warsaw04].

Tanmateix, atès que la configuració de Mailman amb tots els seus serveis requereix bastant feina, tornarem a recórrer a les facilitats

d'instal·lació gairebé automàtica que ens ofereix el paquet Debian corresponent.

Així, en una distribució Debian (Sarge o posterior), procedirem amb l'ordre següent:

```
apt-get install mailman
```

Això iniciarà la instal·lació de paquets. Durant aquest procés se'ns formularan algunes preguntes: en primer lloc, en quins idiomes volem la implementació (escollirem normalment la llengua anglesa i el nostre idioma local, com a mínim). Després, escollirem l'idioma predeterminat (segons l'ús que donarem al sistema o al caràcter més "internacional" del projecte, potser ens interessarà decantar-nos per l'anglès com l'idioma predeterminat, encara que després els usuaris podran escollir qualsevol dels idiomes instal·lats des de la interfície web).

Acabada la instal·lació, crearem la primera llista: la llista de gestió, que és la que s'utilitza com a remitent dels missatges de gestió del sistema [Kollar00]. Per a això, executarem:

```
newlist mailman
```

Haurem d'especificar alguns aspectes que detallem a continuació (marquem en negreta la part que haurem d'escriure com a administradors):

```
$ newlist mailman
```

```
Enter the email of the person running the list: correu-electrònic@domini
```

```
Initial mailman password: xxxxxx
```

To finish creating your mailing list, you must edit your `/etc/aliases` (or equivalent) file by adding the following lines, and possibly running the `'newaliases'` program:

```
## mailman mailing list
```

```
mailman: "|/var/lib/mailman/mail/mailman post mailman"
```

```
mailman-admin: "|/var/lib/mailman/mail/mailman admin mailman"
```

```
mailman-bounces: "|/var/lib/mailman/mail/mailman bounces mailman"
```

```
mailman-confirm: "|/var/lib/mailman/mail/mailman confirm mailman"
```

Nota

Prèviament, assumim que es té instal·lat i configurat correctament un agent de correu (com Postfix o Sendmail) i un servidor web (com Apache) en els quals es basarà Mailman.

```
mailman-join: "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave: "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner: "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request: "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe: "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe: "|/var/lib/mailman/mail/mailman unsubscribe mailm

Hit enter to notify mailman owner...
```

Veiem que ens deixa una tasca per a fer a mà: editar el nostre fitxer `/etc/aliases` (o similar, segons l'agent de correu que tinguem instal·lat).

Si el fitxer `/etc/aliases` canvia, és important incloure tota la informació necessària sobre la llista: propòsit, administrador o administradors i tot el que creguem necessari.

Per exemple, el nostre fitxer `/etc/aliases` podria quedar així:

```
# Llistes de Correu Mailman

## mailman mailing list

## Administrador: usuari@elnostredomini.com

## Propòsit: Gestió interna de Mailman

mailman: "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin: "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces: "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm: "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join: "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave: "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner: "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request: "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe: "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe: "|/var/lib/mailman/mail/mailman unsubscribe mailm
```

No oblidem que després d'editar aquest fitxer hem d'actualitzar la base de dades de l'agent de correu, amb l'ordre:

```
newaliases
```

I fins aquí arriba la fase de la configuració inicial. Si tot ha anat bé, podem iniciar el nostre sistema Mailman i no ens hauria de donar cap error. Executarem:

```
/etc/init.d/mailman start
```

Una vegada arrencat, podrem accedir a la interfície web de Mailman, a l'adreça següent:

```
http://servidor.elnostredomini.com/cgi-bin/mailman/listinfo
```

4.4.2. Configuració de la llista de gestió

Abans de passar el nostre sistema a producció, hem de configurar mínimament la llista de correu de gestió, anomenada *mailman*, que hem creat durant la instal·lació. El més important serà segurament ocultar-la i fer que l'accés al seu arxiu sigui de caràcter privat. Per a això, hem de llegir el correu electrònic de l'administrador inicial, on haurà arribat durant la instal·lació un missatge de benvinguda amb la contrasenya d'administrador. El missatge contindrà un text semblant a aquest:

```
From: mailman-admin@servidor.elnostredomini.com
```

```
To: usuari@elnostredomini.com
```

```
The mailing list 'mailman' has just been created for you. The following is some basic information about your mailing list.
```

```
Your mailing list password is:
```

```
mailman
```

```
You need this password to configure your mailing list. You also need it to handle administrative requests, such as approving mail if you choose to run a moderated list.
```

```
You can configure your mailing list at the following web page:
```

```
http://servidor.elnostredomini.com/cgi-bin/mailman/admin/mailman
```

```
[...]
```

Entrarem en la interfície d'administració de la llista amb la contrasenya indicada al missatge: introduïrem al navegador web l'adreça indicada al missatge i se'ns demanarà la contrasenya. La pàgina web per a administrar qualsevol llista conté un menú superior com el que es mostra en la figura 4.1.

Figura 4-1. Interfície d'administració de la llista de correu de Mailman



Començarem per seleccionar l'opció 'Passwords' per a establir una nova contrasenya d'administració (no cal establir la de moderador, ja que no s'utilitzarà en aquesta llista inicial).

Després seleccionarem en el menú superior, 'Privacy options...' per marcar afirmativament l'opció 'Require approval', i que els que es vulguin apuntar a la llista hagin de tenir el nostre vistiplau; i negativament l'opció 'Advertise this list', perquè no aparegui anunciada públicament a la relació de les llistes de correu electrònic del servidor. Finalment, premerem el botó 'Submit your changes'.

A continuació, seleccionarem 'Archiving options' del menú superior i marcarem "Is archive file source for public or private archival?" a la part inferior, l'opció 'Private'. D'aquesta manera, els missatges d'aquesta llista seran d'accés exclusiu per als seus subscriptors (no hi subscriurem ningú de moment; si és necessari, ho farem més endavant administrant aquesta llista com qualsevol altra).

4.4.3. Operativa d'usuari de llistes

Les llistes de correu s'han dissenyat perquè els interessats s'hi puguin apuntar i puguin consultar de manera senzilla els missatges enviats

fins al moment. En els apartats següents es descriu com es duen a terme aquests processos en Mailman.

Subscripció com a convidat

El primer que necessitem com a usuaris d'una llista de correu és subscriure'ns-hi [Oda03]. Hi ha diverses maneres de fer-ho, però la més senzilla és que ens convidin, és a dir, que l'administrador proposi la nostra entrada a la llista i nosaltres només l'hàgim de confirmar.

En cas de ser convidats, rebriem un missatge semblant al següent:

```
From: projectel-users-request@servidor.elmeudomini.com
```

```
To: usuari@elseudomini.com
```

```
Subject: confirm cbf28fadlee8290848ab
```

Com a usuari reconegut de Projectel, et convidem a pertànyer a la llista de suport a usuaris, esperant que et sigui útil i la puguis fer més útil als altres.

Your address "usuario@seudominio.com" has been invited to join the Projectel-users mailing list at *servidor.midominio.com* by the Projectel-users mailing list owner. You may accept the invitation by simply replying to this message, keeping the Subject: header intact. You can also visit this web page:

<http://servidor.elmeudomini.com/cgi-bin/mailman/confirm/projectel-users/cbf28fadlee8290848ab>

Or you should include the following line -- and only the following line -in a message to *projectel-users-request@servidor.midominio.com* confirm cbf28fadlee8290848ab. Note that simply sending a 'reply' to this message should work from most mail readers. If you want to decline this invitation, please simply disregard this message. If you have any questions, please send them to *projectel-users-owner@servidor.midominio.com*.

Aquest missatge ens notifica que ens conviden a entrar a la llista *projectel-users*. Si acceptem la invitació, només hem de contestar el missatge amb el nostre programa de correu preferit, procurant no modificar l'assumpte, que conté un *hash* que permet identificar la invitació.

Una altra opció és, segons les instruccions del missatge, pitjar l'enllaç web proporcionat per a confirmar-la. I finalment, sempre podem rebutjar la invitació, simplement fent cas omís del missatge.

També és possible que ens hi subscriguin directament, sense demanar-nos permís; en aquest cas normalment rebrem una notificació del fet, en la qual se'ns indicaran les instruccions per a deixar la llista, si és això el que volem fer.

Subscripció per web

La manera més senzilla d'entrar a la llista de correu com a usuari és utilitzar l'adreça web de la llista i interactuar-hi.

Exemple

En navegar per la pàgina d'un projecte del nostre interès, és probable que en algun lloc se'ns indiqui una adreça web on subscriure'ns, per exemple:

```
http://servidor.domini.com/cgi-bin/mailman/listinfo/  
projectel-users
```

Aquesta adreça introduïda al nostre navegador ens portarà a una pàgina web en la qual trobarem, entre altres coses, un formulari de subscripció com el de la figura 4.2.

Emplanarem el formulari amb la nostra adreça de correu, el nostre nom (encara que no és obligatori) i la contrasenya que volem utilitzar per gestionar la nostra subscripció (l'haurem d'introduir dues vegades). En prémer el botó 'Subscriure' rebrem un missatge de correu semblant al de la invitació, amb el qual podrem confirmar la subscripció, igual com hem fet abans, des de la pàgina web o simplement responent el missatge.

Aquest correu de confirmació s'envia per a evitar que algú ens hi pugui subscriure sense el nostre permís. Aquesta política és un dels avantatges de Mailman davant d'altres gestors de llistes de correu.

Figura 4-2. Interfície de subscripció a un projecte de programari lliure

Subscribing to Proyecto1-users

Subscribe to Proyecto1-users by filling out the following form.

Se le mandará un mensaje de correo electrónico pidiéndole una confirmación, para prevenir que otras personas le suscriban sin que usted lo sepa. Esta lista es privada, lo que significa que los suscriptores de la lista no están disponibles a los que no estén suscritos.

Your email address:

Your name (optional):

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:

Reenter password to confirm:

Which language do you prefer to display your messages?

Would you like to receive list mail batched in a daily digest? No Yes

Subscripció per correu electrònic

També és possible subscriure's voluntàriament al projecte sense utilitzar la interfície web en cap moment. Per a això, ens proporcionaran una adreça de correu, semblant a:

`nom-de-llista-request@servidor.elmeudomini.com`

Per a sol·licitar la subscripció, n'hi haurà prou d'enviar a aquest lloc un missatge amb la paraula *subscriu*.

Una vegada més rebrem un missatge de confirmació com a resposta semblant al que hem vist per a les invitacions i les subscripcions per web, per evitar subscripcions involuntàries.

La interfície de correu electrònic permet fer moltes més coses. Per a conèixer-les, podem remetre a aquesta mateixa adreça un missatge amb el text *help*.

Tramesa de missatges a la llista

Per a fer arribar un missatge a la llista, l'adreça de tramesa serà semblant a la següent:

`nom-de-llista@servidor.elmeudomini.com`

En principi, qualsevol missatge que enviem a la llista s'haurà de remetre a tots els subscriptors. Quan això no passa, es pot deure a alguna de les raons següents:

- Que la llista sigui moderada. En aquest cas, un missatge ens informará de la situació, i haurem d'esperar que un moderador autoritzi el missatge.
- Que no estiguem autoritzats. D'algunes llistes, se n'ha de ser subscriptor per a poder-hi escriure. Això evita la majoria de correu brossa o correu no sol·licitat.

A més, si estem subscrits a la llista, normalment rebrem una còpia del missatge que hem enviat. Encara que no sempre, ja que de vegades les llistes es configuren perquè no ens remetin còpia dels nostres missatges.

Consulta de l'arxiu

Es pot consultar el magatzem de missatges enviats a la llista, fins i tot encara que no hi estiguem subscrits (normalment, les llistes de programari lliure són de consulta pública). Així doncs, algunes pàgines web de desenvolupament de projectes de programari lliure ens poden informar de l'adreça on es troba l'arxiu, que sol ser semblant a aquesta:

<http://servidor.elmeudomini.com/pipermail/nom-de-llista/>

L'arxiu permet consultar la llista per dates (mesos o setmanes), nom o autors, segons estigui configurada i vulguem nosaltres.

Exemple

En la figura 4.3 mostrem un arxiu de llista de correu en un navegador.

Figura 4-3. Arxiu de llista de correu d'un projecte

Diciembre 2004 Archives by fecha

- Messages sorted by: [\[hilo\]](#) [\[asunto\]](#) [\[autor\]](#)
- [More info on this list...](#)

Starting: *Mie Dic 1 04:25:25 CET 2004*

Ending: *Jue Dic 30 21:23:31 CET 2004*

Messages: 33

- [\[Gnome-usuarios\] Re:loi de arena en Gnome?](#) Alberto García Fernández
- [\[Gnome-usuarios\] Re:loi de arena en Gnome?](#) La Mula Francis
- [\[Gnome-usuarios\] Re:loi de arena en Gnome?](#) Alberto García Fernández
- [\[Gnome-usuarios\] Re:loi de arena en Gnome?](#) La Mula Francis
- [\[Gnome-usuarios\] Gnome 2.8 en sarge...¿hay alguna fecha ya?](#) Ruben
- [Re: \[Gnome-usuarios\] Gnome 2.8 en sarge...¿hay alguna fecha ya?](#) Francisco Camenforte Torres
- [\[Gnome-usuarios\] \[sin asunto\]](#) Marco mg
- [\[Gnome-usuarios\] \[sin asunto\]](#) David Moreno Garza
- [\[Gnome-usuarios\] Gnome 2.8 en sarge...¿hay alguna fecha ya?](#) Alberto García Fernández
- [\[Gnome-usuarios\] Gnome 2.8 en sarge...¿hay alguna fecha ya?](#) David Moreno Garza
- [\[Gnome-usuarios\] \[sin asunto\]](#) La Mula Francis
- [\[Gnome-usuarios\] Gnome 2.8 en sarge...¿hay alguna fecha ya?](#) Alberto García

Baixa de la llista

Igual com per a subscriure'ns a una llista, Mailman ens proporciona diverses maneres per esborrar-nos-en:

- Per correu electrònic. Se sol·licita l'ordre `unsubscribe` a l'adreça d'administració automàtica de la llista, normalment amb la forma `llista-request@domini`. Com a resposta, rebrem un correu electrònic de sol·licitud de confirmació. En contestar el correu de la manera habitual (sense alterar el *hash* que apareixerà a l'assumpte), quedarem esborrats de la llista.
- Per web. A la pàgina d'informació de la llista es pot trobar una manera d'esborrar-se fàcilment. Accedirem a la pàgina, mitjançant una ordre com la següent:

`http://servidor.domini.com/cgi-bin/mailman/listinfo/projecte1-users`

I anirem al final fins a localitzar un formulari com el que es mostra en la figura 4.4.

Figura 4-4. Interfície de baixa de subscripció d'un projecte

Escriurem la nostra adreça de correu en aquest formulari i farem clic sobre 'Anular subscripció o editar preferències'. Arribarem a una altra pàgina en què trobarem un botó 'Baixa de subscripció' i, com altres vegades, en pitjar-lo ens enviarà un correu electrònic perquè confirmem que ens volem esborrar.

Altres opcions interessants de l'usuari

Com a usuaris de la llista rebrem un missatge de benvinguda amb una contrasenya. És la que utilitzarem, juntament amb la nostra

adreça electrònica, per a canviar les nostres opcions de subscripció des de la interfície web.

Per a això, ens dirigirem a la pàgina de gestió de comptes de la llista, la que té una adreça similar a la següent:

`http://servidor.elmeudomini.com/cgi-bin/mailman/options/nom-de-llista/`

Com en l'apartat anterior, buscarem a la part inferior una casella on introduïrem la nostra adreça de correu i premerem el botó 'Anul·lar subscripció o editar preferències'. Accedirem a la pàgina de preferències des de la qual podem efectuar algunes operacions interessants:

- Esborrar-nos de la llista.
- Canviar la nostra contrasenya.
- Canviar les nostres opcions de configuració de la llista. Per a veure les opcions, buscarem una taula al final de la pàgina, semblant a la que es mostra en la figura 4.5.

Figura 4-5. Interfície de configuració d'una llista de correu

Mail delivery Set this option to <i>Enabled</i> to receive messages posted to this mailing list. Set it to <i>Disabled</i> if you want to stay subscribed, but don't want mail delivered to you for a while (e.g. you're going on vacation). If you disable mail delivery, don't forget to re-enable it when you come back; it will not be automatically re-enabled.	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled <input type="checkbox"/> Set globally
Set Digest Mode If you turn digest mode on, you'll get posts bundled together (usually one per day but possibly more on busy lists), instead of singly when they're sent. If digest mode is changed from on to off, you may receive one last digest.	<input type="radio"/> Off <input checked="" type="radio"/> On
Get MIME or Plain Text Digests? Your mail reader may or may not support MIME digests. In general MIME digests are preferred, but if you have a problem reading them, select plain text digests.	<input type="checkbox"/> MIME <input checked="" type="checkbox"/> Plain Text <input type="checkbox"/> Set globally
Receive your own posts to the list? <small>(Predictably, you will not receive any of your own messages you post to the list. If you don't want to</small>	<input type="checkbox"/> No <input checked="" type="checkbox"/> Yes

Entre aquestes opcions, destacarem les d'ús més habitual:

- *Mail delivery*: permet desactivar o reactivar la recepció de missatges de la llista. D'aquesta manera hi podem estar subscrits sense que ens arribin missatges. Això és interessant si volem tenir dret de publicació en una llista privada sense que ens n'arribin els missatges.

- *Digest mode*: si activem aquesta opció, rebrem un missatge diari amb el resum (*digest*) de tots els missatges enviats aquell dia, en lloc d'un de nou cada vegada. Si se superés un determinat límit de missatges al dia, podríem rebre més d'un resum.

4.4.4. Operativa d'administració de llistes

Per acabar aquest capítol dedicat a les llistes de correu electrònic, i en particular a Mailman, ens endinsarem en algunes de les tasques a què un administrador s'enfronta més sovint.

Alta i baixa de llistes de correu

Durant la instal·lació ja avancem com es creen les llistes de correu amb la llista de gestió de Mailman. Per a crear qualsevol llista, executarem l'ordre:

```
newlist -l es nom-de-llista administrador@domini.com nom-de-llista
```

que ens mostrarà per pantalla els àlies que hem de posar al fitxer `/etc/aliases` del nostre agent de correu. Aquesta manera de crear la llista inclou a la línia d'ordres la contrasenya inicial, i també el nom de la llista. L'administrador d'aquesta l'haurà de canviar com més aviat millor.

És poc habitual donar de baixa una llista de correu, però si fos necessari, caldria decidir si ens quedem amb una còpia dels arxius de la llista o bé l'esborrem. En el primer cas, n'esborrarem els àlies de `/etc/aliases` i llançarem l'ordre:

```
rmlist nom-de-llista
```

Si a més volem eliminar els arxius, l'ordre és:

```
rmlist -a nom-de-llista
```

Configuració de la llista

Una vegada creada la llista, el seu administrador en prendrà el control i procedirà a entrar a la interfície web d'administració amb la contrasenya que se li subministrarà per correu electrònic.

Per a això, utilitzarà l'adreça web que se li proporcionarà al mateix missatge, normalment de la manera següent:

`http://servidor.elnostrodomini.com/cgi-bin/mailman/admin/nom-de-llista`

En la figura 4.6 es mostren algunes opcions de configuració interessants, encara que segurament l'usuari interessat les voldrà explorar totes.

Figura 4-6. Opcions de configuració d'una llista de correu

Categorías de configuración	Otras actividades administrativas
<ul style="list-style-type: none"> • [Opciones Generales] • Claves: • Opciones de idiomas • Administración de los suscriptores... • Opciones de entrega regular • Opciones de recopilaciones 	<ul style="list-style-type: none"> • Opciones de Privacidad... • Gestión de rebotes • Opciones de Almacenaje • Pasarela de Correo-e<->Noticias • Contestador automático • Filtrado de contenido • Temas
	<ul style="list-style-type: none"> • Ocuparse de las peticiones pendientes de moderar • Ir a la página de información general sobre la lista. • Editar el código HTML de las páginas de acceso público • Ir al archivo de la lista <p>• Desconexión</p>

a) **Opcions generals.** En aquesta categoria tenim les opcions generals de configuració d'una llista de correu (*anonymous_list*). Les d'ús més habitual són:

- El text descriptiu de la llista (*info*).
- Els textos annexos als missatges d'alta i baixa d'usuaris a la llista (*welcome_msg*, *goodbye_msg*).
- La possibilitat d'ocultar les adreces dels usuaris per a evitar el correu brossa (*anonymous_list*).
- El filtre administratiu, que intercepta els missatges de gestió de llistes enviats a la llista i no a l'administrador (*administrivia*).

- La longitud màxima acceptada dels missatges, per a evitar grans fitxers annexos que poden molestar altres usuaris (*max_message_size*).
 - Des de fa poc temps, Mailman separa el rol d'administrador del de moderador (*moderator*). Per tant, podem incloure en aquesta opció la llista de moderadors que rebran les peticions de moderació.
- b) **Claus.** Aquesta opció permet canviar la nostra contrasenya i també la clau usada pels moderadors.
- c) **Administració dels subscriptors.** Aquesta opció permet portar un control de les subscripcions existents i crear-ne d'altres de noves:
- En la llista de subscriptors: podem canviar qualsevol opció de qualsevol dels subscriptors. Pot ser útil per a tornar a activar usuaris que hagin quedat desactivats per errors del compte de correu (atenció!, perquè també es poden haver desactivat voluntàriament).
 - En subscripcions massives: podem enganxar moltes adreces de correu i ordenar-ne la subscripció, o només la invitació. Es recomanen utilitzar la invitació, perquè els usuaris decideixin per si mateixos si volen ser a la llista o no.
 - En baixes massives: es poden donar de baixa diverses adreces alhora.
- d) **Opcions de lliurament regular.** Aquí ajustarem opcions que afecten només els missatges que es lliuren directament als usuaris, no per lots (*digest*). L'habitual en aquest apartat és personalitzar el peu dels missatges (*msg_footer*).
- e) **Opcions de recopilacions.** Només afecten els subscriptors que reben els missatges per lots (*digest*). Podem decidir el període d'enviament mínim (diari, mensual, etc.) i la mida màxima (*digest_size_threshold*) del missatge de recopilació (un cop superat, es rebrà més d'un missatge).

f) Opcions de privadesa. Les més interessants són:

- La política de subscripció (*subscribe_policy*), en què establirem si es requereix l'aprovació de l'administrador per a acceptar subscriptors i si és necessària la confirmació del sol·licitant. Per a evitar subscripcions no desitjades, sempre s'ha d'activar, com a mínim, l'opció de confirmació.
- La visibilitat de la llista de subscriptors (*private_roster*), amb la qual podem ocultar les adreces dels usuaris de la resta de subscriptors.
- Podem gestionar els filtres de remitent, la qual cosa farà que la llista sigui moderada. Com a mínim, s'hauria d'activar l'opció de retenir en l'opció '*generic_nonmember_action*', per a evitar que arribin a la llista els missatges de tercers (que de vegades, i si es fa molta publicitat de la llista, seran correu brossa). Evidentment, no es descarten, sinó que un moderador haurà de revisar el missatge per autoritzar-lo o no. Un altre filtre interessant és activar l'opció *default_member_moderation* per a fer la llista totalment moderada.

Moderació

Quan activem l'opció de moderació en alguna de les opcions de privadesa mostrades d'una llista, o quan algun usuari no autoritzat envia un missatge a la llista, s'enviarà un avís als moderadors. Ja hem indicat que els rols d'administrador i moderador s'han separat en versions recents de Mailman. Però com que, en realitat, qualsevol administrador pot moderar, els administradors també rebran els avisos de moderació.

Un avís de moderació tindrà l'aspecte següent:

```
From: nom-de-llista-owner@servidor.elmeudomini.com
To: nom-de-llista-owner@servidor.elmeudomini.com
Subject: La tramesa a nom-de-llista d'usuari@unaltredomini.net necessita aprovacio
Date: Sun, 02 Jan 2005 21:12:03 +0100
```

As list administrator, your authorization is requested for the following mailing list posting:

List: `nom-de-llista@servidor.elmeudomini.com`

From: `usuari@altredomini.net`

Subject: Prova de correu

Reason: Missatge dirigit a una llista privada procedent d'una adreça que no pertany a la llista

At your convenience, visit:

`http://servidor.elmeudomini.com/cgi-bin/mailman/admindb/nom-de-llista`

to approve or deny the request.

En versions més recents de Mailman, s'hi annexa, a més, el missatge complet rebut i unes instruccions per a moderar directament, sense necessitat d'utilitzar la interfície web. En aquest cas, n'hi ha prou de respondre el missatge perquè Mailman entengui que es vol descartar.

Per a acceptar el missatge, utilitzarem la interfície web. Seleccionarem l'adreça indicada, se'ns sol·licitarà la contrasenya d'administrador o de moderador i se'ns mostrarà les sol·licituds pendents de moderar (figura 4.7).

Figura 4-7. Interfície amb la llista de sol·licituds pendents de moderació

Categorías de configuración		Otras actividades administrativas
<ul style="list-style-type: none"> • [Opciones Generales] • Claves: • Opciones de idiomas • Administración de los suscriptores... • Opciones de entrega regular • Opciones de recopilaciones 	<ul style="list-style-type: none"> • Opciones de Privacidad... • Gestión de rebotes • Opciones de Almacenaje • Pasarela de Correo-e<->Noticias • Contestador automático • Filtrado de contenido • Temas 	<ul style="list-style-type: none"> • Ocuparse de las peticiones pendientes de moderar • Ir a la página de información general sobre la lista. • Editar el código HTML de las páginas de acceso público • Ir al archivo de la lista <p>• Desconexión</p>

Una de les opcions més interessants és la que permet descartar tots els missatges pendents, sobretot quan veiem que tot el que hi ha pendent és correu brossa. Per a això, marcarem l'opció 'Discard all messages marked Defer' i premerem el botó 'Enviar todas las dades'.

En cas contrari, marcarem la casella 'Acceptar' dels missatges que s'hagin de publicar. Fins i tot encara que no hi estigui subscrit, si és un remitent habitual de missatges interessants, ens pot interessar marcar 'Acceptar' en l'opció 'Afegir *usuari@domini* a un d'aquests filtres de remitents'.

Ordres interessants des de la línia d'ordres

L'administrador de llistes de correu no necessita tenir accés de línia d'ordres a la màquina que acull les llistes. Tanmateix, de vegades té accés com a administrador de la màquina i, en aquests casos, pot llançar directament algunes instruccions interessants:

- `add_members -r fitxer nom-de-llista`
Aquesta ordre permet afegir a la llista seleccionada les adreces contingudes al fitxer.
- `remove_members -f fitxer nom-de-llista`
Aquesta ordre permet, al contrari, esborrar de la llista indicada tots els subscriptors inclosos al fitxer.
- `genaliases`
Aquesta ordre permet obtenir els àlies de totes les llistes definides que cal introduir al fitxer `/etc/aliases`.
- `newlist` i `rmlist`, ordres que ja havíem vist per crear i eliminar llistes de correu.

Per a conèixer la resta d'ordres a la seva disposició, l'administrador pot fer un cop d'ull als arxius que hi ha a:

```
/usr/lib/mailman/bin
```


5. La gestió d'un projecte

Fins ara, hem vist les eines que s'utilitzen generalment en el desenvolupament de programari lliure i ens hem centrat en les més populars, que s'han descrit més exhaustivament. Tanmateix, de la mateixa manera que no n'hi ha prou de tenir un bon programa per a seguir unes pràctiques que ens assegurin l'èxit (sigui quina sigui la definició d'èxit que més s'ajusti a la nostra situació), tampoc no ens ho garanteix l'ús d'aquestes eines, encara que, evidentment, tots dos elements hi ajuden en gran manera.

En aquest capítol veurem les bones maneres de procedir per a assolir l'èxit que es busca. Sens dubte, donem per fet que tenim un programa amb prestacions que poden ser interessants a la comunitat i amb una infraestructura (pròpia o en un portal que ofereixi aquests serveis) que permeti el desenvolupament col·laboratiu. Estem davant d'una de les etapes decisives del projecte: el seu **alliberament**.



El nostre objectiu és crear una comunitat entorn del projecte perquè en algun moment el nivell d'activitat arribi a ser tan gran que el desenvolupament del projecte sigui autocatalfíic; és a dir, que la mateixa comunitat resolgui les necessitats que es planteja per si mateixa.

Si passa això el nostre projecte té èxit, és probable que la nostra tasca canviï radicalment i passem a dedicar molt de temps i esforç a gestionar tots aquests recursos externs.

5.1. Elecció de la llicència

En aquest moment del desenvolupament cal prendre una sèrie de decisions addicionals, que són tan importants com les de caràcter tècnic

per a l'èxit futur del projecte. La primera decisió que hem de considerar és l'elecció de la llicència sota la qual es regiran la distribució, l'ús, les modificacions i, fins i tot, les redistribucions del nostre programari (inclosa la documentació). Segons els nostres objectius, ens podrà interessar més un tipus de llicència robusta (o *copyleft*), que assegurí que la naturalesa del programari que creem sempre serà lliure; o permissiva (aquestes llicències també són conegudes com a tipus BSD), si no ens importa que apareguin versions de la nostra feina sota altres condicions (fins i tot amb llicències propietàries) en el futur. Sempre es pot escollir una llicència dual que, com a autors del programari, ens ofereix flexibilitat. Finalment, també hi ha la possibilitat de crear una llicència pròpia, encara que això no és una tasca fàcil.

Nota

Les eines de creació de projectes de programari de GNU (autoconf/automake) ens poden generar aquests fitxers automàticament.

En qualsevol cas, els tipus de llicències i les seves característiques són un tema que s'escapa de l'àmbit d'aquest material, per la qual cosa remetem el lector a [Gbarahona04]. L'important per a nosaltres en aquest punt és que, una vegada escollida la llicència, el nostre programari n'hauria d'incloure el text perquè tothom pugui saber les condicions que regeixen el programari. Generalment aquest arxiu rep el nom de 'COPYING' o 'LICENSE'.

Així mateix, la llicència haurà de constar, juntament amb tota la informació de *copyright*, en tots els arxius de codi font de què es compon el nostre programari (incloent-hi la documentació, si triem la mateixa llicència per a la documentació).

Ara ja estem en condicions de posar el nostre programari a disposició del públic perquè comenci el procés de distribució. A partir d'ara, hauré de distribuir i modificar el programa com si no fos nostre. En altres paraules, encara que tenim el reconeixement com a autors del programari, gràcies a la llicència de programari lliure que li hem atribuït, ningú no en té, *de facto*, la propietat.

5.2. El web del projecte

Un aspecte de la infraestructura d'un projecte de programari lliure sobre el qual, de moment, hem passat de puntetes és el seu web.

Juntament amb la instal·lació i la interfície d'usuari, el web és el tercer dels elements que proporcionen la primera impressió sobre el nostre projecte.

El web d'un projecte de programari lliure és fonamental. Els objectius perseguits amb el lloc web en el desenvolupament d'un projecte de programari lliure són diversos. En primer lloc, ha de proporcionar visibilitat; és a dir, ha de donar a conèixer el projecte als usuaris i als desenvolupadors. A més, el web és fonamental per a assolir una base d'usuaris, col·laboradors ocasionals i desenvolupadors suficient per a la sostenibilitat del projecte.

De la mateixa manera que les empreses i les institucions inverteixen una quantitat considerable de diners en imatge corporativa, un projecte de programari lliure també ha de cuidar la seva imatge. En primer lloc, el web serà el primer contacte de qualsevol persona amb el projecte, per la qual cosa ha de tenir una estructura adequada i clara i contenir la informació necessària per a una primera presa de contacte amb el projecte. Evidentment, la infraestructura dependrà del públic objectiu del nostre web: no és el mateix un web destinat a augmentar la visibilitat del nostre projecte que un d'orientat a involucrar altra gent com a codesenvolupadors.

Per a mantenir prou nivell d'activitat en el projecte, cal procurar motivar la implicació dels usuaris. Per a això, es disposa de diversos mitjans, com fòrums o llistes de correu d'usuaris. Algun integrant del projecte s'hauria d'encarregar del seu manteniment, per solucionar els problemes que poguessin aparèixer. A més, la comunitat que forma el projecte hauria de resoldre les preguntes i els dubtes que es plantegin al fòrum i a les llistes de correu.

En qualsevol cas, és important mantenir la informació actualitzada. També es pot disposar d'informació al web del projecte sobre eines útils que es poden emprar juntament amb el nostre programari o de projectes relacionats.

És molt important que al web hi hagi documentació disponible sobre el projecte, destinada tant a usuaris com a desenvolupadors. Sense aquesta documentació, difícilment s'hi implicarà un usuari o un desenvolupador.

Exemple

Un lloc web l'última notícia del qual és de fa un parell d'anys donarà la sensació d'inactivitat, col·loquialment, de "projecte mort".

5.3. Estructura del web del projecte

En aquesta secció es proposa una estructura per al web que compleixi tots els requisits exposats fins ara.

En primer lloc, hi ha d'haver una secció **d'informació del projecte**. En aquesta secció s'ha d'incloure una descripció del projecte: en què consisteix, per a quins sistemes està disponible, de quines versions es disposa (per exemple, *estable* i en *desenvolupament*), etc. A més, s'hi ha d'incloure informació sobre les últimes versions que s'han publicat, amb enllaços que portin directament a les pàgines d'aquestes versions i, si és possible, una llista de les últimes notícies del projecte.

També cal que hi hagi una secció de **baixada del programari**, en què hi hagi les diferents versions enllaçades (per exemple, *estable* i en *desenvolupament*), tant en forma de paquets binaris com en paquets de codi font. També pot ser interessant tenir paquets preparats per a les distribucions més comunes com Debian, RedHat o SUSE. Molts projectes també ofereixen la possibilitat de baixar un paquet amb l'última versió del codi font que hi ha al CVS. Generalment, això es fa de manera automàtica mitjançant un guió que cada nit recopila el que hi ha al CVS i l'empaqueta degudament.

Una altra secció que no s'ha de descuidar és la **d'últimes notícies i comunicats**, en la qual s'han d'incloure totes les notícies relacionades amb el projecte. Aquesta secció s'ha d'actualitzar regularment per a evitar la sensació de falta d'activitat. Un projecte actiu sempre té un efecte de crida per a usuaris i desenvolupadors, ja que fa la impressió que en podran resoldre els dubtes o, si no hi ha la funcionalitat específica que necessiten, esperar que la hi contingui pròximament.

Una secció fonamental és la de **documentació**. La documentació pot estar dividida en dos grups: la dirigida a usuaris i l'orientada a desenvolupadors. A més, és convenient que estigui disponible tant en forma de paquets per a baixar com per a consultar en línia. En el cas de la documentació que es pot baixar, ha d'estar disponible en diversos formats (per exemple, en HTML i PDF).

Entre la documentació es poden incloure exemples d'ús, programes d'aprenentatge o *tutorials*, etc. que mostrin l'ús del programari. També és interessant incloure una secció de consells i trucs. Si la nostra aplicació té una interfície gràfica, hi podem afegir unes quantes captures de pantalla. Això fa que molts usuaris es decideixin a provar-la.

Finalment, també pot ser útil una secció que inclogui **eines relacionades**. S'hi pot afegir informació sobre la infraestructura de suport del nostre projecte, indicar on són els fòrums d'usuaris, quines llistes de correu hi ha, on és el sistema de seguiment d'errors o les instruccions sobre com ha d'emprar l'usuari aquests mitjans.

5.4. Instal·lació senzilla

Una de les regles bàsiques que aprendrem a apreciar amb el temps és que hem de cuidar la instal·lació. Si el programari és complex d'instal·lar perquè té moltes dependències, és necessari compilar un parell de biblioteques abans que el nostre programari, s'ha de modificar a mà el `makefile` en alguns casos, etc., la majoria dels usuaris no arribaran a instal·lar-se'l. Cal eliminar totes aquestes barreres d'entrada per als usuaris inexperts. D'altra banda, a aquells usuaris que es decideixin a instal·lar-lo els sorgiran preguntes, i possiblement errors, i ens haurem de dedicar a resoldre dubtes d'instal·lació en comptes de dedicar aquest preciós temps a altres tasques probablement més productives.

No s'han d'estalviar esforços en aquest sentit, ja que quan un usuari busca un programari, el seu procés d'avaluació se sol limitar a la instal·lació d'un producte i a un primer cop d'ull de la seva funcionalitat (si és en mode gràfic, a la seva interfície). Amb un error en la instal·lació n'hi sol haver prou perquè l'usuari rebutgi el nostre programa per molt bo que sigui. I si això passa sovint, limitem les nostres possibilitats, no solament des del punt de vista de nombre d'usuaris, sinó també des de la perspectiva de la realimentació quant a informes d'errors, documentació externa, o fins i tot, que hi hagi usuaris que puguin arribar a ser codesenvolupadors. És per aquest motiu que resulta gairebé imprescindible una documentació exhaustiva del projecte, almenys pel que fa a les instruccions de compilació, instal·lació i empaquetament.

Exemple

Les *demos* poden consistir, per exemple, en programes d'aprenentatge amb una sèrie de fitxers del nostre programa per a seguir els exemples que s'hi desenvolupen.

També és una bona idea proporcionar una sèrie de *demos*; és a dir, arxius de demostració d'allò que pot fer la nostra aplicació. Només si s'eviten aquestes barreres d'entrada podrem aconseguir una massa crítica d'usuaris que possibiliti que el desenvolupament del projecte sigui sostenible.

Atès que comencem a intuir que aquestes tasques de distribució resulten més aviat àrdues, probablement el millor és que ho facin d'altres per nosaltres. En el món del programari lliure hi ha canals que se n'encarreguen de manera efectiva: les distribucions. Una bona manera perquè incloguin el nostre programari és simplificar-ne el procés de compilació, instal·lació i, fins i tot, empaquetament perquè el prenguin tal com és. Hi ha eines com `autoconf` o `automake` –molt esteses en el món del programari lliure que facilitaran aquestes tasques.

5.5. Consells per a la distribució i difusió del programari

En el món del programari lliure, els programes solen seguir unes pautes de distribució i difusió. No és obligatori seguir-les, però si es fa, facilita molt la integració del nostre projecte dins de la comunitat de programari lliure. Per exemple, serà més probable que arribi a formar part d'una distribució de GNU/Linux o que algun dels codes-envelopadors es dediqui a crear els paquets per distribuir-lo.

En primer lloc, haurem de distribuir el nostre programari de les maneres més diverses possibles perquè l'usuari pugui escollir la que més li convingui. Com ara:

- a) Un paquet comprimit (generalment `tar.gz`, encara que si és molt gran, ens podem decantar per `tar.bz2`) amb el codi font, incloent-hi guions i *makefiles* per a compilar el programa fàcilment. Si la nostra aplicació es pot executar sobre Windows, també és una bona idea incloure el paquet comprimit en format `.zip`, ja que la seva difusió és més gran en aquesta plataforma.

- b) Un paquet comprimit (`tar.gz`, `tar.bz2` o `zip`) amb els binaris ja compilats.
- c) Un paquet Debian, per a les distribucions Debian i basades en Debian com KNOPPIX, Ubuntu, etc.
- d) Un paquet RPM (*red hat package manager*) per a les distribucions que empen aquest format de paquets; fins i tot es poden oferir paquets RPM especialment construïts per a les distribucions més populars (Fedora, SUSE, Mandrake).

La secció de baixades ha de ser clara perquè qualsevol usuari pugui trobar fàcilment el programari en el format que l'interessa, i baixar-lo només en aquest format.

Els paquets, tant si contenen el codi font com els binaris, han d'incloure uns fitxers molt comuns en tota la distribució de programari lliure:

- El fitxer 'README', amb informació general sobre el programari i informació de contacte: URL del projecte, on es troba ajuda i documentació, etc. Alguns fitxers d'aquesta mena també inclouen la seqüència d'instal·lació.
- Un fitxer amb la llicència (generalment anomenat 'COPYING'), tal com hem vist en aquest capítol.
- Un fitxer amb les instruccions d'instal·lació ('INSTALL'). Si el nostre programari en requereix d'altres per al seu funcionament correcte, aquesta informació se sol incloure en aquest fitxer o en un d'anomenat 'DEPENDS'.
- Un fitxer amb la bitàcola de canvis efectuats al programari ('CHANGELOG'). La informació continguda en aquest fitxer permetrà als desenvolupadors conèixer els últims canvis i on han estat efectuats.
- Hi ha el costum de tenir un fitxer anomenat 'AUTHORS' (o similar) on hi ha una llista dels autors del programari. Se sol distingir en aquests casos entre els que han portat el pes del projecte i els desenvolupadors que hi han participat de manera esporàdica o puntual.

- Documentació de l'aplicació, generalment en un subdirectori anomenat 'doc'.
- Alguns binaris per a diferents arquitectures; per a assegurar una compatibilitat més gran amb tots els sistemes possibles, és millor que estiguin enllaçats de manera estàtica amb les biblioteques que utilitzi el projecte.

En qualsevol cas, el nom del paquet i alguns dels fitxers han de donar informació sobre la versió del programa.

A més, el paquet de codi font ha d'incloure també un guió *configure* que generi el *makefile*, per poder compilar i instal·lar el programari fàcilment. Aquest mètode estàndard de compilació també facilitarà la creació de paquets per a les diferents distribucions i la seva possible integració dins de les distribucions.

5.6. A la recerca de l'efecte en xarxa

Podria semblar que s'ha acabat el procés d'alliberament del nostre projecte i, tanmateix, no hem fet més que començar. A partir d'ara cal emprar un esforç, en absolut menyspreable, per a dur a terme una sèrie d'accions fonamentals per al seu èxit.



En resum, en aquest punt es tracta de maximitzar la difusió del nostre programari entre la comunitat per a incrementar la probabilitat de captar l'atenció d'usuaris i possibles codesenvolupadors.

Aconseguir usuaris i desenvolupadors sol ser el primer pas per a aconseguir més usuaris i més desenvolupadors, un fenomen que s'ha arribat a anomenar *efecte en xarxa*. A continuació es presenten algunes de les pràctiques comunes per a aconseguir aquest objectiu.

Hem de deixar el programari en algun lloc públic on es pugui baixar, preferentment mitjançant HTTP i/o FTP. Evidentment, el codi font també hauria d'estar disponible als mateixos llocs i, com hem descrit

en el capítol dedicat al CVS, mai no està de més oferir la possibilitat que l'última versió es pugui obtenir de manera anònima.

Per a facilitar-ne la màxima difusió, haurem de llançar una espècie de campanya publicitària per donar a conèixer el nostre programa. Ens valdrem tant de la infraestructura pròpia del projecte com dels processos de difusió d'informació dels quals disposa la comunitat del programari lliure. Així, ho anunciarem a les nostres llistes de correu electrònic i als fòrums d'usuaris, dins del nostre propi projecte. En general, és una bona idea donar d'alta el projecte al portal Freshmeat, dedicat exclusivament a informar de les últimes versions de programes lliures. Així mateix, si s'aconsegueix arribar a llocs de notícies populars, com Barrapunto, Libertonía o Slashdot, el nostre projecte guanyarà en audiència considerablement.

També hi ha la possibilitat de promocionar el nostre programari en les jornades i els congressos de programari lliure que es realitzen regularment per tot el planeta. En aquests casos, és important aprofitar el fet d'aparèixer personalment per a convèncer que el nostre programari funciona, per la qual cosa no solament s'hauria de fer una presentació amb transparències, sinó que no s'hauria d'estalviar esforços a ensenyar, almenys durant alguns minuts, el que pot fer. D'altra banda, moltes d'aquestes jornades exigeixen que les ponències i els textos s'enviïn per escrit, per la qual cosa una vegada han acabat, la presentació i la documentació generades es podran penjar al web del projecte.

Últimament, també són molt populars entre els desenvolupadors de programari lliure els *blogs*, on es van detallant els detalls del procés de desenvolupament del projecte. Aquests *blogs* se solen unir en "planetes", que aglutinen en una sola pàgina web les entrades en diferents bitàcoles. Per tant, cal procurar que el nostre *blog* formi part d'algun planeta la temàtica del qual estigui relacionada amb el nostre projecte.

Exemple

Així, a tall d'exemple, si el nostre projecte està escrit en el llenguatge de programació C# i fa ús de la plataforma Mono, el nostre *blog* es pot unir al Planeta Mono-Hispano (<http://planeta.monohispano.org/>), que és un lloc on es concentra la comunitat de Mono-Hispano i que probablement ens representarà beneficis quant a realimentació, usuaris i publicitat.

Nota

Freshmeat

<http://www.freshmeat.net>

En qualsevol cas, quan es generen aquestes notícies, cal tenir en compte dos principis bàsics: no precipitar-se mai; és a dir, generar notícies només quan el projecte hagi assolit realment una fita; i evitar la publicitat excessiva, que pot arribar a identificar els nostres anuncis gairebé com a correu brossa.

5.7. Codi i internacionalització

Una vegada que hàgim aconseguit cridar l'atenció d'alguns desenvolupadors, és convenient facilitar-los la col·laboració. És, per tant, una bona idea mantenir el nostre codi net i estructurat, amb tots els comentaris i els noms de variables i funcions comprensibles. Moltes vegades, si no sempre, això significa utilitzar l'anglès.

Tot i tenir la infraestructura bàsica i el codi en anglès, el programari ha d'estar preparat per a ser fàcilment adaptable a altres idiomes, en particular si el públic objectiu del nostre projecte és l'usuari del carrer. És fonamental que les biblioteques que emprem en el desenvolupament es puguin integrar fàcilment amb les eines de traducció, com per exemple GNU gettext. La nostra tasca no consisteix només a traduir la nostra aplicació, que tècnicament s'anomena *localització* (o **I10n**), sinó que, a més, s'han de modificar altres paràmetres de caràcter cultural com la moneda, els formats de data, etc. **Hem de facilitar l'anomenada internacionalització** (i**18n**) del programari, que consisteix precisament a utilitzar els mitjans tècnics perquè la localització sigui possible.

En aquest punt també té importància tenir una infraestructura que afavoreixi la col·laboració i que ja s'ha descrit amb detall en els capítols anteriors. Recordem que és fonamental tenir llistes de correu per a la comunicació entre usuaris i desenvolupadors, sistema de control de versions per al treball coordinat de diversos desenvolupadors i sistema de seguiment d'errors per a la gestió dels nostres errors. També es poden muntar fòrums per als usuaris més inexperts i una pàgina de notícies relacionades amb el projecte. Tot això perquè la informació sobre el projecte flueixi el màxim possible.

Nota

Les abreviatures d'*internacionalització* (i18n) i *localització* (I10n) corresponen a la primera i última lletres de cada paraula i els caràcters del mig se substitueixen pel nombre de lletres rebutjades.

La infraestructura només és el primer pas, serveix per a mimar els nostres col·laboradors, encara que siguin esporàdics.



La part més important consistirà a gestionar les contribucions, els pedaços, les modificacions i els suggeriments rebuts.

5.8. Esforç dedicat a tasques de gestió

Durant tot aquest material, hem indicat repetidament que el model de desenvolupament que seguim no es limita només a desenvolupar el programari i a tasques merament tècniques. Després de descriure exhaustivament les diferents eines que se solen utilitzar i les pràctiques més recomanables perquè el nostre projecte guanyi en visibilitat dins de la comunitat, estem segurs que el lector s'haurà adonat que les tasques de gestió ocupen una gran part del temps dedicat a la generació de programari lliure.

En aquest apartat, ens agradaria mostrar quant de temps impliquen totes aquestes tasques. Per a això, ens referirem a unes estimacions proporcionades per Brian Behlendorf [Behlendorf99], un dels creadors d'Apache. En el seu document, Brian divideix l'esforç en dos: la part dedicada al llançament del projecte, d'una banda, i l'esforç continuat que necessitem dedicar a la seva gestió, de l'altra. El primer aspecte és el nombre d'hores que caldrà dedicar-hi al començament del projecte, mentre que les unitats del segon corresponen al nombre d'hores a la setmana.

El primer punt que s'analitza és el de la gestió de la infraestructura. Es necessitaran recursos humans perquè algú iniciï les llistes de correu electrònic, els fòrums, el servidor web, el gestor d'informes d'errors, el dipòsit CVS, etc. No queda gaire clar si en aquesta tasca s'inclou el disseny del web i la creació dels seus continguts. El temps estimat per a tenir-ho tot engegat és d'unes cent hores i l'esforç continuat se situa entorn de vint hores a la setmana, la qual cosa seria un treball a temps parcial. Quan es va escriure aquest article, no hi havia llocs especialment dissenyats per a hostatjar projectes de programari lliure

com els que hem vist en l'exemple de SourceForge, per la qual cosa pot ser que avui dia –almenys per a projectes de mida petita i mitjana– sigui un esforç que ens puguem estalviar. Això no passarà si tenim un projecte gran, ja que la flexibilitat que proporciona disposar d'una infraestructura pròpia fa rendible aquest esforç.

La tasca de coordinació del desenvolupament, inclòs el control de qualitat, és la que es requereix per a netejar el codi i fer que tot estigui disponible per a consum públic. Behlendorf estima que a aquesta tasca s'han de dedicar entre quaranta i dues-cents hores en el moment del llançament, mentre que l'aplicació i supervisió de pedaços i dels *commits* al CVS representa unes vint hores setmanals.

Una altra tasca molt lligada a l'anterior, però que no toca el codi font de l'aplicació pròpiament, és la gestió dels errors que arriben al sistema de control d'errors. Com s'ha mostrat en el capítol dedicat als sistemes de control d'errors, en particular a Bugzilla, la gestió dels errors és una tasca complexa que necessita intel·ligència humana per a dur-la a terme convenientment. L'encarregat de l'atenció dels informes d'errors ha de facilitar, en la mesura que sigui possible, la comunicació entre usuaris i desenvolupadors, de manera que els primers vegin la resolució dels errors dels quals informen, i els segons tinguin accés ràpid i ordenat als errors que han de corregir. Behlendorf no especifica el temps que cal invertir en aquesta tasca al començament del projecte –simplement indica que el desenvolupador responsable d'aquesta tasca s'hauria de familiaritzar amb el codi i les eines de control d'errors–, però afegeix: de deu a quinze hores setmanals durant la vida del projecte.

Tenir bona documentació, tant del codi per als desenvolupadors com del programa per als usuaris, s'estima en unes seixanta hores per a la primera publicació, tenint en compte, això sí, que no n'hi ha cap més. A partir d'aquí, cal dedicar deu hores setmanals a actualitzar aquesta informació.

Finalment, cal afegir un component de relacions públiques que publici el projecte al màxim, com s'ha vist en diversos apartats d'aquest mateix capítol. En aquest cas tampoc no es dona una estimació per al començament del projecte, però sí que es diu que serien necessàries unes vint hores a la setmana per a dur-lo a terme.

Evidentment, totes les estimacions s'han de prendre amb cautela, ja que no s'especifica la mida del projecte. A partir del currículum de l'autor, ens podem imaginar que es refereix a projectes de la mida d'Apache, de mida mitjana en línies de codi (entre cinquanta mil i cent mil línies), però amb una àmplia comunitat tant de desenvolupadors com d'usuaris.

Si sumem tots els temps estimats, obtenim un valor d'entre 200 i 360 hores per a llançar un projecte de programari lliure i una dedicació setmanal de vuitanta a vuitanta-cinc hores. Com s'ha indicat en el seu moment, en el cas del llançament i gràcies als serveis oferts per alguns portals, aquest temps es pot minimitzar fins al punt que una gran part de l'esforç es resumeix a netejar el codi i afegir documentació. Per la seva banda, segons les estimacions amb què treballem, el manteniment del projecte implicaria tenir dues persones a temps complet dedicades exclusivament a la gestió de la infraestructura i l'intercanvi d'informació en el projecte. Evidentment, aquest temps no és de desenvolupament pròpiament dit, sinó de suport a la nostra "comunitat", per la qual cosa es pressuposa que el que guanyem amb aquesta ajuda hauria de ser més gran que el que podrien oferir aquestes dues persones treballant a temps complet específicament a millorar el programa.

5.9. Conclusions

En aquest capítol hem donat diversos consells per a l'èxit del nostre projecte partint de la base que ja tenim una infraestructura de desenvolupament que s'ha anat presentant al llarg d'aquest text. Repetirem aquí els principis fonamentals que no podem oblidar:

- Cal atraure els usuaris al web del nostre projecte.
- Cal mantenir els usuaris en el projecte; per a això, cal mantenir un nivell mínim d'activitat i de difusió de les novetats del projecte.
- Si podem crear una comunitat amb usuaris i desenvolupadors, el manteniment del projecte serà autocatalític, i no caldrà que nosaltres ens responsabilitzem del seu nivell d'activitat, almenys de tot el projecte en el seu conjunt.

- Els usuaris han de tenir poques dificultats per a utilitzar i instal·lar el programari i participar en la comunitat del projecte.
- Cal aconseguir automatitzar tot el que puguem, perquè això facilitarà la delegació de tasques en el futur, i disminuirà la càrrega de treball dels administradors del projecte.

Bibliografia

Behlendorf, Brian [Behlendorf99] "Open Source as a Bussiness Strategy", (en "Open Sources, Voices from the Open Source Revolution").

<http://www.oreilly.com/catalog/opensources>

Bugzilla Team, The [Bugzilla04] "The Bugzilla Guide - 2.18 Release".

<http://www.bugzilla.org/docs/2.18/html/>

Cederqvist, Per y otros [Cederqvist04] "Version Management with CVS".

<https://www.cvshome.org/docs/manual/>

Collins-Sussman, Ben; Fitzpatrick Brian W.; Pilato, Michael [Collins04]

"Version Control with Subversion". <http://svnbook.red-bean.com/>

Fogel, Karl; Bar, Moshe [Fogel00] "Open Source Development with CVS, 3rd Ed." <http://cvsbook.red-bean.com/>

González Barahona, Jesús M. [Gbarahona04] "¿Y cómo hago para que mi código sea libre?" (en "Sobre Software Libre. Compilación de ensayos sobre software libre"). <http://gsync.escet.urjc.es/~grex/sobre-libre/>

Collar, Christopher [Kollar00] "GNU Mailman List Manager Documentation v.2.0". <http://staff.imsa.edu/~ckolar/mailman/>

Oda, T. [Oda03] "GNU Mailman-List Member Manual".

<http://www.gnu.org/software/mailman/mailman-member/index.html>

Vesperman, Jennifer [Vesperman03] "Essential CVS". (1.a edició, junio de 2003). ISBN: 0-596-00459-1. Editorial O'Reilly.

Villa, Luis [Villa03] "Large Free Software Projects and Bugzilla. Lessons from GNOME Project QA" (Proceedings of the Linux Symposium 2003)

<http://archive.linuxsymposium.org/ols2003/Proceedings/All-Reprints/Reprint-Villa-OLS2003.pdf>.

Warsaw, Barry [Warsaw04] "GNU Mailman - Installation Manual".

<http://www.gnu.org/software/mailman/mailman-install/index.html>

Appendix A. GNU Free Documentation License

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of *copyleft* which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The *Document* below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as *you*. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A *Modified Version* of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A *Secondary Section* is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a *Secondary Section* may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The *Invariant Sections* are certain *Secondary Sections* whose titles are designated, as being those of *Invariant Sections*, in the notice that says that the Document is released under this License. If a section does not fit the above definition of *Secondary* then it is not allowed to be designated as *Invariant*. The Document may contain zero *Invariant Sections*. If the Document does not identify any *Invariant Sections* then there are none.

The *Cover Texts* are certain short passages of text that are listed, as *Front-Cover Texts* or *Back-Cover Texts*, in the notice that says that the Document is released under this License. A *Front-Cover Text* may be at most 5 words, and a *Back-Cover Text* may be at most 25 words.

A *Transparent* copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise *Transparent* file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not *Transparent*. An image format is not *Transparent* if used for any substantial amount of text. A copy that is not *Transparent* is called *Opaque*.

Examples of suitable formats for *Transparent* copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of *transparent* image formats include PNG, XCF and JPG. *Opaque* formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The *Title Page* means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, *Title Page* means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section *Entitled XYZ* means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as *Acknowledgements*, *Dedications*, *Endorsements*, or *History*. To *Preserve the Title* of such a section when you modify the Document means that it remains a section *Entitled XYZ* according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified

Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled *History*. Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled *History* in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the *History* section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled *Acknowledgements* or *Dedications* Preserve the Title of the section, and preserve in the section all the

substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled *Endorsements* Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled *Endorsements* or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled *Endorsements*, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled *History* in the various original documents, forming one section Entitled *History*; likewise combine any sections Entitled *Acknowledgements*, and any sections Entitled *Dedications*. You must delete all sections Entitled *Endorsements*.

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an *aggregate* if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled *Acknowledgements*, *Dedications*, or *History*, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License or *any later version* applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the

GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled *GNU Free Documentation License*

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the *with...Texts.* line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

