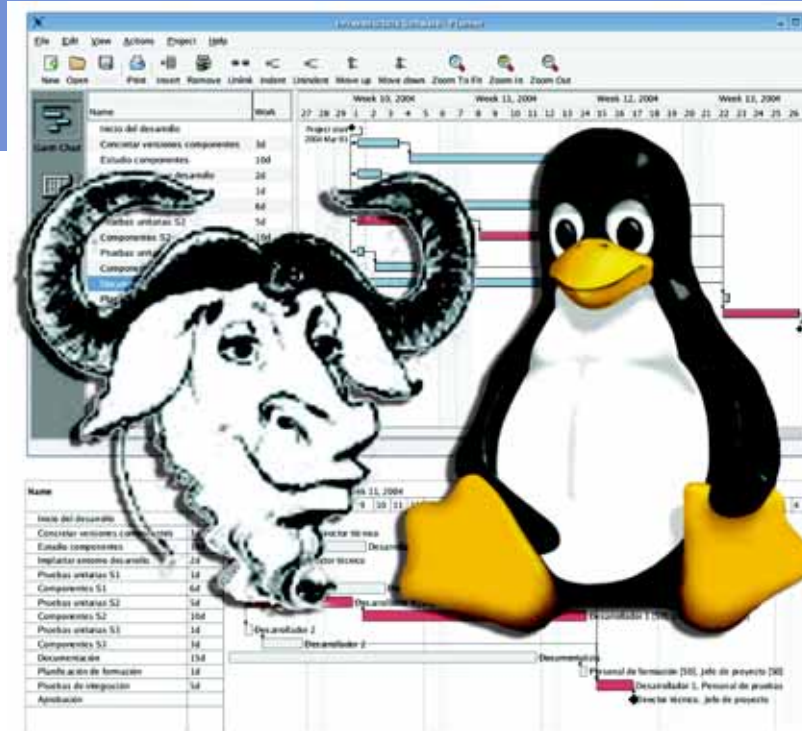


# Programari lliure

Alberto Otero García

XP06/M2021/02158



# Projecte de direcció de sistemes d'informació

## David Megías Jiménez

Coordinador

Enginyer d'Informàtica per la Universitat Autònoma de Barcelona. Màster de Tècniques Avançades d'Automatització de Processos per la Universitat Autònoma de Barcelona. Doctor en Informàtica per la Universitat Autònoma de Barcelona. Professor dels Estudis d'Informàtica i Multimèdia de la Universitat Oberta de Catalunya.

## Jordi Mas

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. És també coordinador general de Softcatalà. Com a consultor ha treballat per a empreses com a Menta, Telépolis, Vodafone, Lotus, eresMas, Amena i Terra España.

## Alberto Otero García

Autor

Enginyer d'Informàtica per la Universitat Ramon Llull. Llicenciat en Investigació i Tècniques de Mercat per la Universitat Oberta de Catalunya. Professor titular de l'assignatura Administració de Sistemes Operatius en Enginyeria i Arquitectura La Salle. Soci fundador i cap de projectes de Cometa Technologies, empresa dedicada a donar solucions en tecnologies de la informació, basades en l'ús d'estàndards i eines de codi obert. Consultor del Màster Internacional en Programari Lliure de la UOC.

Segona edició: febrer 2007

© Fundació per a la Universitat Oberta de Catalunya. Av. Tibidabo, 3  
9-43, 08035 Barcelona

Material realitzat per Eureka Media, SL

© Autors: Alberto Otero García

Es garanteix permís per a copiar, distribuir i modificar aquest document segons els termes de la *GNU Free Documentation License, Version 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

<b>Agraïments</b> .....	5
<b>Introducció</b> .....	7
<b>Objectius</b> .....	9
<b>1. Estudi de viabilitat</b> .....	11
1.1. Establiment de l'abast del sistema .....	12
1.2. Estudi de la situació actual.....	15
1.3. Definició de requisits del sistema .....	19
1.4. Estudi d'alternatives de solució .....	21
1.5. Valoració de les alternatives .....	22
1.6. Selecció de la solució .....	26
<b>2. Anàlisi del sistema</b> .....	29
2.1. Definició del sistema .....	29
2.2. Establiment de requisits .....	34
2.3. Definició d'interfícies d'usuari .....	37
2.4. Especificació del pla de proves .....	39
<b>3. Disseny del sistema</b> .....	43
3.1. Arquitectura .....	44
3.1.1. Definició de nivells d'arquitectura.....	44
3.1.2. Especificació d'estàndards, normes de disseny i construcció.....	47
3.1.3. Identificació de subsistemes .....	49
3.2. Casos d'ús reals .....	51
3.2.1. Revisió de casos d'ús per subsistema .....	52
3.2.2. Especificacions de desenvolupament i proves.	54
3.2.3. Requisits d'implantació .....	58
<b>4. Desenvolupament</b> .....	63
4.1. Planificació de les activitats d'integració de sistema .....	64
4.2. Triar la llicència més adequada .....	67
4.3. Entorn de desenvolupament .....	70
4.4. Documentació .....	71

<b>5. Implantació</b> .....	73
5.1. Formació .....	74
5.2. Implantació del sistema i proves.....	74
5.3. Nivell de servei.....	76
5.4. Acceptació del sistema.....	77
<b>6. Manteniment</b> .....	79
<b>Resum</b> .....	81
<b>Bibliografia</b> .....	83
<b>GNU Free Documentation License</b> .....	85

## Agraïments

L'autor agraeix 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.



## Introducció

Per a dur a terme un projecte de sistemes d'informació en entorns de programari lliure, com en qualsevol altre tipus de projecte, és necessari seguir un procés que ens porti des de la comprensió de l'abast del problema que volem resoldre fins a la implantació i manteniment de la solució que hàgim triat.

Encara que el director de sistemes d'informació en entorns de programari lliure, o d'un projecte concret basat en la utilització de programari lliure, no necessita conèixer totes i cada una de les tècniques i eines utilitzades al llarg dels projectes, sí que necessita saber quines fases s'hauran de seguir, quins productes s'hauran d'obtenir al final de cada una d'elles i, a trets generals, com obtenir-los, a fi de poder:

- Planificar. El director de projecte ha de planificar quins recursos cal assignar a cada una de les fases del projecte, estimar el temps que durarà completar-les, el seu cost econòmic, etc.
- Organitzar. El director de projecte ha de poder organitzar els recursos dels quals disposa de la manera més òptima, coordinar l'avenç d'aquest projecte amb la resta de projectes de sistemes de la informació i de l'empresa, alinear els objectius del projecte amb els de l'organització, conèixer quines eines són les més indicades per al seu ús, etc.
- Controlar. El director de projecte ha de poder testar la bona marxa del projecte, comprovar la qualitat dels resultats obtinguts, oferir ajuda als integrants de l'equip en qualsevol de les fases en cas de ser necessari, etc.

En aquest curs es pretenen repassar aquelles fases que és necessari seguir al llarg de tot projecte de sistemes de la informació, i que el director de projecte haurà de supervisar.

Aquestes fases són les següents:

- **Estudi de viabilitat:** s'estudiarà en línies generals quins problemes es volen resoldre, quines solucions possibles hi ha i quina d'elles és la més adequada.
- **Anàlisi:** es descriurà detalladament el sistema que es vulgui construir, quins requisits ha de complir i quins usuaris ha de satisfer.
- **Disseny:** es realitzarà el plantejament tecnològic de la solució.
- **Desenvolupament:** es durà a terme la programació, integració, instal·lació, etc., dels diferents subsistemes que componguin el projecte.
- **Implantació:** es passarà el sistema construït a producció a fi que els seus usuaris comencin a utilitzar-lo.
- **Manteniment:** es realitzaran tant les correccions dels possibles errors que puguin sorgir en el sistema implantat, com les millores evolutives que es considerin oportunes.

**Figura 1.** Fases d'un projecte de sistemes d'informació



Aquestes fases seran presents, d'una o una altra manera, amb aquests noms o amb d'altres, en qualsevol projecte de sistemes d'informació, des dels gestionats mitjançant el mètode "clàssic" (per exemple, en fases seguides seqüencialment, en cascada, etc.), fins als gestionats com suggereix el conjunt de metodologies conegudes com a àgils.

#### Nota

Al llarg de tot el material del curs, es desenvolupa un cas pràctic a fi d'exemplificar les explicacions donades. Aquest cas pràctic no constitueix de cap manera un estudi exhaustiu del projecte proposat, sinó que simplement serveix com a marc per a oferir diferents exemples de les parts de què es compon amb finalitats únicament didàctiques.



## Objectius

Els objectius que el lector haurà d'assolir en finalitzar el curs de Projecte de direcció de sistemes d'informació són els següents:

- Haver comprès de manera global el que representa la direcció d'un projecte de sistemes d'informació, en especial en un entorn tecnològic de programari lliure.
- Haver assimilat quines fases integren un projecte de sistemes d'informació, i quines tasques s'han de dur a terme en cada una d'elles, especialment des del punt de vista de la seva direcció.
- Haver reflexionat sobre quines eines de programari lliure poden ajudar en cada una de les fases d'un projecte de sistemes d'informació.



## 1. Estudi de viabilitat

L'objectiu de la realització de l'**estudi de viabilitat** és, donat un conjunt de necessitats plantejades, escollir aquella solució que millor les cobreixi d'entre totes les possibles (o descartar-les totes en cas que cap no les satisfaci).

En l'estudi de viabilitat es consideraran les diferents solucions possibles, tenint en compte:

- L'estat inicial del sistema.
- La situació actual.
- Els requisits plantejats.

Cada una de les solucions proposades en l'estudi de viabilitat haurà de recollir els aspectes següents:

- Econòmics: s'haurà d'incloure un estudi econòmic preliminar que prevegi els costos associats a cada una de les solucions.
- Tècnics: s'haurà d'incloure un estudi tècnic preliminar de cada una de les solucions.
- Legals: s'haurà d'incloure un estudi d'aquells aspectes legals que puguin influir en la viabilitat de la solució.
- Operatius: s'haurà d'incloure un estudi previ de l'operativa de cada una de les solucions proposades.

Una vegada plantejada cada una de les solucions, s'escollirà la millor tenint en compte:

- L'impacte en l'organització.
- La inversió que cal realitzar.
- Els riscos associats.

Els apartats següents descriuen amb més detall cada una de les tasques que s'han de dur a terme per a fer l'estudi de viabilitat.

## 1.1. Establiment de l'abast del sistema

En aquesta fase de l'estudi de viabilitat es pretén **estudiar l'abast de les necessitats plantejades pel client** (tant si són tercers, en cas de tractar-se d'un projecte dirigit a altres organitzacions, com si són usuaris interns, en cas de tractar-se d'un projecte per a la mateixa organització).

El primer que serà necessari fer és la **descripció general** de les necessitats plantejades pel client. En aquesta descripció general s'hauran d'incloure els aspectes bàsics descrits en l'apartat anterior (econòmics, tècnics, legals i operatius) que tinguin especial rellevància.

### Cas pràctic

#### Renovació de la infraestructura del programari de Solucions Obertes, SA.

L'empresa Solucions Obertes, SA, dedicada al desenvolupament de programari, ha decidit renovar tot el maquinari de què disposen:

- Quatre servidors dedicats a donar suport a tota l'empresa.
- Vint ordinadors personals, en els quals es distingeixen dos perfils d'ús: el del personal no tècnic (ofimàtica, navegació per Internet, etc.) i el del personal tècnic (programació, test de programari, navegació per Internet, etc.).
- Un sistema de còpies de seguretat en cinta.

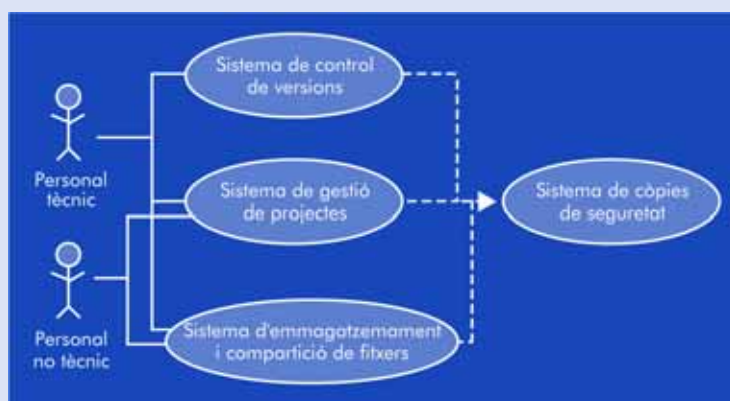
Juntament amb el canvi de maquinari, es considera fer també una renovació o ampliació del programari usat en la infraestructura bàsica de l'empresa, a fi d'actualitzar-lo i fer créixer els costos de manteniment en la mesura possible. Aquest programari és el següent:

- Sistema d'emmagatzemament i compartició de fitxers: és el dipòsit central d'arxius, on cada usuari (tècnic o no) desa els fitxers dels quals vol que hi hagi còpia de seguretat i que permet compartir documents en directoris ordenats per grups de treball.

- Sistema de control de versiones: actualment no existeix, i seria utilitzat pel personal tècnic a fi de mantenir les versions, canvis entre aquestes, etc., del codi font dels diferents productes programari de l'empresa.
- Sistema de còpies de seguretat: és l'encarregat de fer les còpies de seguretat totals i incrementals de la informació que es considera més valuosa en l'empresa (la base de dades del sistema de gestió de projectes, els fitxers d'usuaris tècnics i no tècnics, i el codi emmagatzemat en el sistema de control de versions).

Des del punt de vista econòmic, perquè la renovació sigui viable haurà d'implicar la menor despesa possible, ja que la partida pressupostària més important corresponent a sistemes d'informació es destinarà a renovar el maquinari. Tècnica-ment, les necessitats plantejades són molt poc restrictives, ja que Solucions Obertes és una empresa dedicada a les tecnologies de la informació i, per tant, disposa de personal qualificat que afronta i gaudeix de qualsevol repte tècnic sense més problemes. Des del punt de vista legal, s'exigeix que les solucions aportades siguin tan flexibles com sigui possible, ja que es valora molt negativament el fet de no disposar de la màxima llibertat per a copiar i/o modificar els sistemes programari que s'implantïn. Operativament, l'única necessitat plantejada consisteix en el fet que en cap cas no s'ha de perdre funcionalitat de la qual en aquests moments ja es disposa, sinó en tot cas, ampliar-la.

**Figura 2.** Descripció general del sistema



Així mateix, a més de descriure de manera general el projecte, s'ha de tenir en compte **com afectarà a**:

- Altres projectes de tecnologies de la informació ja en curs o que es pensen posar en marxa.
- Les diferents unitats de l'organització, tenint en compte qui en són els responsables i quina n'és l'estructura.

#### Cas pràctic

##### Abast del projecte de renovació de la infraestructura de programari de Solucions Obertes, SA.

En el cas de Solucions Obertes, SA, el projecte de renovació de la infraestructura del programari afectarà:

- El projecte de renovació del maquinari de l'empresa. Segons el programari de base escollit, serà més apropiat un maquinari o un altre, i per això serà necessari coordinar ambdós projectes.
- El projecte de renovació dels entorns integrats de programació (IDE) utilitzats pel personal tècnic, ja que de manera ideal aquest s'hauria d'integrar totalment amb el sistema de control de versions.

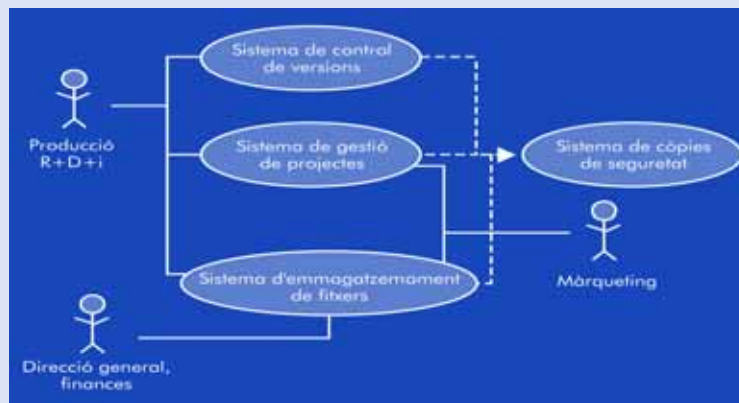
D'altra banda, el canvi de la infraestructura del programari de l'empresa afectarà els departaments següents:

- Direcció general, finances: aquests departaments en resultaran afectats únicament durant el procés de migració a les noves plataformes, ja que en algun moment la disponibilitat del sistema d'emmagatzematge de fitxers pot notar-ne les conseqüències. Fora d'això, la influència de la renovació hauria de ser mínima.
- Màrqueting: aquest departament en resultarà afectat, igual que el de direcció general i finances, per la renovació del sistema d'emmagatzemament de fitxers. A més, notaran les conseqüències pel canvi del sistema de gestió de projectes, a causa que les persones d'aquesta unitat de l'empresa són les que gestionen el contacte amb el client i per tant en són usuaris habituals (ja que els permet intro-

duir peticions de millora dels productes, descripcions d'errors, etc.).

- Producció, recerca i desenvolupament (R+D): aquests dos departaments seran els més afectats pel canvi, ja que per a ells implica, a més del que s'ha comentat per als departaments anteriors, la incorporació de la utilització del sistema de control de versions en el procés productiu i d'investigació.
- S'haurà d'informar puntualment els responsables de cada un dels departaments dels detalls i l'abast del projecte, en la mesura que els afecti.

**Figura 3.** Descripció general del sistema



## 1.2. Estudi de la situació actual

En aquesta fase de l'estudi de viabilitat es pretén **estudiar la situació en la qual es troba el sistema d'informació de l'empresa i realitzar-ne un diagnòstic**, sempre referent al projecte que ens ocupa.

La primera tasca que cal fer dins de l'estudi de la situació actual és la d'**identificar aquells sistemes que s'hagin de descriure**, és a dir, de quins sistemes farem l'estudi perquè resultin afectats d'alguna manera pel projecte inclòs en l'estudi de viabilitat. És interessant tam-

bé fixar quins usuaris participaran en l'estudi de la situació actual de cada un dels sistemes escollits.

#### Cas pràctic

##### Identificació dels sistemes actuals

Atès que el projecte consisteix a renovar i ampliar la infraestructura del programari de Solucions Obertes, SA, haurem d'estudiar com a mínim la situació actual dels sistemes que volem renovar. En aquest cas, com ja s'ha comentat, aquests sistemes seran el de gestió de projectes, control de versions, emmagatzemament de fitxers i còpia de seguretat. Per tal de fer un diagnòstic tan complet com sigui possible, hem decidit treballar juntament amb usuaris dels departaments de màrqueting i producció (ja que són els usuaris més intensius d'aquests sistemes). El sistema de control de versions s'obviarà en aquesta fase de l'estudi de viabilitat, ja que actualment l'empresa en manca.

El pas següent dins de l'estudi de la situació actual serà el de **descriure cada un dels sistemes identificats** en el pas anterior. La descripció es farà tenint en compte la informació recollida en les sessions de treball amb els usuaris seleccionats com a representatius, i haurà d'assolir el nivell de detall suficient per a poder fer un diagnòstic encertat de l'estat real de cada un dels sistemes estudiats. Així mateix, la dedicació de recursos en aquesta fase dependrà de la informació de partida de la qual es disposi (la descripció de la situació actual pot ser trivial en alguns casos o tremendament complicada en d'altres).

#### Cas pràctic

##### Descripció dels sistemes actuals

A continuació es descriu la situació actual dels sistemes de Solucions Obertes, SA seleccionats amb anterioritat:

- Sistema de gestió de projectes: actualment aquest sistema consisteix en una intranet desenvolupada per la mateixa empresa. Aquesta intranet està allotjada en un servidor amb sistema operatiu GNU/Linux, servidor web Apache i programada en PHP, basant-se en el sistema gestor de base de dades relacional (SGBDR) MySQL.



- Sistema d'emmagatzemament i compartició de fitxers: actualment aquest sistema consisteix en un servidor amb directors compartits, als quals pot accedir qualsevol persona de l'empresa. El servidor esmentat té com a sistema operatiu Microsoft Windows 2000, aprofitant la funcionalitat de compartició de carpetes que ofereix.
- Sistema de còpies de seguretat: actualment aquest sistema consisteix en el bolcat diari a cinta de tots els fitxers emmagatzemats en el sistema d'emmagatzemament. El servidor al qual està connectada la unitat lectora de cintes té com a sistema operatiu Microsoft Windows NT, i les còpies es fan mitjançant el programari subministrat amb el mateix sistema operatiu.

**Figura 4.** Descripció general del sistema actual



Per a completar l'estudi de la situació actual dels sistemes, se n'haurà de fer un **diagnòstic**; és a dir, analitzar la informació obtinguda detectant possibles problemes i punts de millora.

## Cas pràctic

## Diagnòstic dels sistemes actuals

Una vegada analitzada la informació obtinguda en la descripció de la situació actual dels sistemes estudiats en l'empresa Solucions Obertes, SA, s'ha arribat a les conclusions següents:

- Sistema de gestió de projectes: s'ha detectat que la funcionalitat disponible és molt limitada, ja que cada nova característica que es vol afegir implica fer una programació a mida (recordem que aquest sistema l'ha desenvolupat la mateixa empresa). Aquest fet farà que en el futur o bé l'eina quedi obsoleta o bé els seus costos de manteniment siguin massa elevats.
- Sistema d'emmagatzemament i compartició de fitxers: es fa patent la necessitat de crear un sistema d'usuaris i permisos que permeti consultar certs fitxers únicament a aquelles persones o grups de persones que estiguin autoritzats per a això. A més, hi ha certa preocupació per l'escalabilitat de tot el sistema, ja que últimament se n'ha notat cert alentiment a causa que cada vegada és més usat (es desen més fitxers, s'hi accedeix més, suporta més usuaris, etc.).
- Sistema de còpies de seguretat: s'ha arribat a la conclusió que és necessari renovar l'estratègia de còpies de seguretat, ja que en aquest moment es fan sempre còpies completes de tots els fitxers. També s'ha detectat la necessitat de començar a fer còpies de dades que fins ara no es feien i que resideixen en servidors que no comparteixen directoris per cap mètode dels habituals (p. ex., el servidor que allotja el sistema de gestió de projectes).

### 1.3. Definició de requisits del sistema

Una vegada descrita la situació actual del sistema, i tenint en compte les opinions dels diferents usuaris implicats, es passarà a **descriure de manera general els requisits que haurà de complir el projecte** del qual s'estudia la viabilitat.

La descripció del conjunt de requisits que haurà de complir el projecte servirà posteriorment per a avaluar cada una de les possibles solucions alternatives existents. És per això que a més de la descripció esmentada, és interessant incloure una qualificació de la prioritat de cada un dels requisits, a fi que tingui present la seva importància relativa respecte a la resta.

La descripció de cada requisit inclourà una explicació d'aquest, la prioritat que s'hi assigna i una catalogació dins d'un conjunt de categories definides.

#### Cas pràctic

##### Definició general de requisits del sistema de còpies de seguretat de Solucions Obertes, SA

Mitjançant l'estudi del sistema de còpies de seguretat actual, els punts de millora i problemes detectats, i les entrevistes amb els usuaris, s'han identificat i catalogat els requisits següents (la prioritat de cada un d'ells està indicada amb un número entre el 0 i el 100, 100 és el prioritari):

##### Requisits tècnics

- (100) Arquitectura: s'hauran de poder fer còpies de servidors remots connectats mitjançant la xarxa al servidor de còpies de seguretat.
- (100) Arquitectura: les còpies de seguretat s'hauran de poder fer de servidors amb sistemes operatius iguals o diferents del del servidor de còpies de seguretat.
- (80) Seguretat: les còpies de seguretat han de raure en un servidor que sigui segur, ja que contindran informació molt rellevant per a l'empresa.

#### Nota

En la resta d'aquest apartat ens centrarem en l'estudi del sistema de còpies de seguretat com a exemple particular del cas pràctic plantejat fins al moment. Per a la resta de sistemes (gestió de projectes, emmagatzematge i compartició de fitxers, control de versions), seria necessari seguir el mateix procés que es descriurà d'ara endavant amb el sistema de còpies de seguretat.

- (50) Normatives i/o estàndards: les còpies de seguretat s'hauran d'emmagatzemar en un format tan obert com sigui possible. En la mesura que es pugui, s'intentarà que aquest format sigui independent de la plataforma sobre la qual es facin les còpies.

#### Requisits operatius

- (100) Operativa: s'han de poder fer còpies de seguretat completes i incrementals.
- (80) Operativa: s'han de poder restaurar fitxers concrets que resideixin dins de certa còpia de seguretat, no aquesta sencera.
- (20) Configuració: la interfície de programació de còpies ha de ser tan intuïtiva i fàcil d'usar com sigui possible.
- (50) Configuració: la periodicitat de realització de les còpies de seguretat ha de ser totalment configurable.
- (100) Suports: el programari de còpies de seguretat ha de ser compatible amb el maquinari escollit com a suport d'emmagatzemament (la unitat de cintes escollida).
- (60) Suports: el programari de còpies de seguretat ha de ser compatible amb el màxim nombre de suports d'emmagatzematge possibles.

#### Requisits legals

- (60) La llicència d'ús del programari de còpies de seguretat ha de ser tan poc restrictiva com sigui possible.
- (60) La llicència d'ús del sistema operatiu del servidor de còpies de seguretat ha de ser tan poc restrictiva com sigui possible.

#### Requisits econòmics

- (80) En el cas que sigui necessària una despesa en concepte de llicència d'ús del programari de còpies de seguretat, haurà de ser tan petita com sigui possible.
- (80) La despesa corresponent al sistema operatiu del servidor de còpies de seguretat ha de ser tan petita com sigui possible.

## 1.4. Estudi d'alternatives de solució

Una vegada expressats els requisits que haurà de complir el projecte del qual es fa l'estudi de viabilitat, es passarà a **proposar diverses solucions alternatives** que compleixin aquests requisits. En aquesta fase es tindrà en consideració, així mateix, tota la informació recollida fins al moment: descripció general, abast, situació actual, etc.

Per a cada alternativa s'haurà d'especificar en què consisteix, tant funcionalment com tècnicament (estudiant en quina mesura es cobreixen els requisits descrits prèviament), si es basa o no en algun producte ja existent en el mercat (en aquest cas s'estudiarà aquest producte, descrivint possibles costos de llicències, evolució prevista, estàndards que compleix, etc.), i si representa o no la necessitat de fer algun desenvolupament a mida (en aquest cas se l'haurà de descriure de manera que quedi clar quin n'és l'abast).

### Cas pràctic

#### Solucions alternatives per al sistema de còpies de seguretat de Solucions Obertes, SA

Com a exemple d'alternatives al sistema de còpies de seguretat es proposen tres solucions possibles, les quals tenen les característiques següents:

- Microsoft Windows + aplicació propietària: en aquest cas el sistema operatiu del servidor de còpies de seguretat serà Microsoft Windows 2000, i es faran mitjançant l'adquisició d'un programari específic de realització de còpies de seguretat, per exemple Arkeia (<http://www.arkeia.com/>). S'ha comprovat que el programari de còpies de seguretat triat compleix els requisits funcionals i tècnics definits sobre això. Referent als requisits legals i econòmics de la solució proposada, ni el sistema operatiu ni el programari de còpies de seguretat no compleixen el que s'hi ha expressat.

- GNU/Linux + aplicació propietària: en aquest cas el sistema operatiu del servidor de còpies de seguretat seria GNU/Linux (en qualsevol de les seves distribucions); tanmateix, el programari de còpia de seguretat seria propietari, per exemple Arkeia. S'ha comprovat que el programari de còpies de seguretat triat compleix els requisits funcionals i tècnics definits sobre això. Referent als requisits legals i econòmics, el sistema operatiu els compleix (ja que no és necessari pagar cap llicència d'ús, i a més se'ns permet fins i tot estudiar-ne i modificar-ne el codi font), però no així el programari de còpies de seguretat (ja que aquest és propietari).
- GNU/Linux + aplicació lliure: en aquest cas, tant el sistema operatiu del servidor de còpies de seguretat com el programari de còpies de seguretat serien lliures (per exemple, Amanda, <http://www.amanda.org/>). S'ha comprovat que el programari de còpies de seguretat escollit compleix els requisits funcionals i tècnics definits sobre això, llevat del d'oferir una interfície fàcil d'utilitzar. Quant als requisits legals i econòmics, aquests es cobreixen perfectament, ja que les llicències són molt flexibles i els costos d'adquisició són nuls.

En els tres casos serà necessari fer un mòdul de programari que reculli la informació emmagatzemada en la base de dades del sistema de gestió de projectes i l'emmagatzemi en fitxers que es puguin copiar en el sistema de còpies de seguretat.

Sobre la base de la informació obtinguda per diferents canals (informes, fòrums de discussió, experiència del personal de l'empresa, etc.), s'ha considerat que el cost d'instal·lació i manteniment dels components és igual en els tres casos.

### 1.5. Valoració de les alternatives

Una vegada s'han estudiat les solucions alternatives dins del projecte del qual es fa l'estudi de viabilitat, s'ha de passar a **valorar-les con-**

siderant la seva viabilitat econòmica (anàlisi costos/beneficis) i riscos que comporten.

Per a cada una de les possibles solucions, s'haurà d'estudiar la seva viabilitat econòmica, és a dir, confeccionar una anàlisi costos/beneficis que deixi patent la despesa que serà necessària fer i el que s'espera obtenir a canvi (tant tangiblement com intangiblement).

### Cas pràctic

#### Anàlisi costos/beneficis del sistema de còpies de seguretat de Solucions Obertes, SA

Els costos d'adquisició imputats a cada una de les solucions són els següents:

- Microsoft Windows + aplicació propietària = 550 € + 500 € = 1.050 €
- GNU/Linux + aplicació propietària = 0 € + 500 € = 500 €
- GNU/Linux + aplicació lliure = 0 € + 0 € = 0 €

Atès que considerem que els costos d'instal·lació i manteniment són els mateixos per als tres casos, no tindran efecte en la comparació que exposem (en un cas real podrien tenir molta importància, ja que no únicament es compararien les diferents solucions, sinó que s'estudiaria la viabilitat econòmica de triar qualsevol d'elles).

En el cas de la tercera opció, GNU/Linux + aplicació lliure, hem de tenir en compte el cost afegit associat a l'aprenentatge en la utilització de l'aplicació lliure de còpies de seguretat, ja que suposem que la interfície d'usuari no és tan fàcil de manejar com la de l'aplicació propietària. Suposem que el fet d'utilitzar l'aplicació lliure davant l'aplicació propietària requerirà deu hores extres de dedicació (a un preu mitjà de 30 €/hora) sobre la utilització de la primera. És necessari sumar per tant aquest cost al d'adquisició: 0 € + 300 € = 300 €.

Els beneficis de cada una de les solucions són els descrits en apartats anteriors (requisits, descripció, etc.).

#### Nota

Els preus indicats són ficticis, utilitzats únicament a tall d'exemple.

A més d'estudiar la viabilitat econòmica de les diferents solucions, haurem de tenir en compte els riscos associats a cada una d'elles. Per a cada una de les alternatives existents descriurem quines incerteses, problemes potencials, etc., hi ha.

### Cas pràctic

#### Riscos en les alternatives del sistema de còpies de seguretat de Solucions Obertes, SA

Els riscos associats a cada una de les solucions alternatives són els següents:

##### Microsoft Windows + aplicació propietària:

- Sistema operatiu: canvi en l'estratègia de negoci del fabricant, desapareix el suport donat fins al moment i es fa necessària una actualització.
- Sistema operatiu: errors de seguretat detectats però no solucionats pel fabricant en un període de temps raonable.
- Aplicació propietària: desaparició del fabricant del producte, o canvi d'estratègia de negoci (atès que no es disposa del codi font de l'aplicació, aquest fet representaria que qualsevol error o problema no podria ser reparat).

##### GNU/Linux + aplicació propietària:

- Sistema operatiu: hi podria haver la falta de suport en determinats casos, ja que no hi ha un únic fabricant que centralitzi el desenvolupament del sistema operatiu.
- Aplicació propietària: desaparició del fabricant del producte, o canvi d'estratègia de negoci (atès que no es disposa del codi font de l'aplicació, aquest fet representaria que qualsevol error o problema no podria ser solucionat).



**GNU/Linux + aplicació lliure:**

- Sistema operatiu: es podria donar la falta de suport en determinats casos, ja que no hi ha un sol fabricant que centralitzi el desenvolupament del sistema operatiu.
- Aplicació lliure: desaparició de l'equip principal de desenvolupadors que mantenen l'aplicació.

Arribats a aquest punt, s'haurà de fer una **proposta d'enfocament a fi de pal·liar en la mesura possible els riscos** descrits abans. D'aquesta manera intentarem reflectir si aquests riscos són salvables, fent-se rellevant la seva importància relativa.

**Cas pràctic****Pal·liació de riscos en les alternatives del sistema de còpies de seguretat de Solucions Obertes, SA**

Els possibles enfocaments a fi de pal·liar els riscos associats a cada una de les solucions alternatives són els següents:

**Microsoft Windows + aplicació propietària:**

- Sistema operatiu: signatura de contracte de suport del sistema operatiu amb el seu fabricant per un període de temps igual a aquell que estimem que serà la vida del sistema de seguretat tal com l'estudiem. Aquesta solució ha de ser acceptada pel fabricant per a poder-se dur a terme.
- Sistema operatiu: signatura de contracte de suport amb indemnitzacions en cas que es produeixin errors en la seguretat del sistema a causa de problemes en el sistema operatiu. Aquesta solució ha de ser acceptada pel fabricant per a poder-se dur a terme.
- Aplicació propietària: signatura de contracte en el qual el fabricant es compromet a subministrar almenys el codi font de la seva aplicació en cas que cessi la seva activitat.

**GNU/Linux + aplicació propietària:**

- Sistema operatiu: es pot contractar el suport d'una empresa externa que es comprometi a centralitzar i resoldre els possibles problemes que puguin sorgir.
- Aplicació propietària: signatura de contracte en el qual el fabricant es comprometi a subministrar almenys el codi font de la seva aplicació en cas que cessi la seva activitat.

**GNU/Linux + aplicació lliure:**

- Sistema operatiu: es pot contractar el suport d'una empresa externa que es comprometi a centralitzar i resoldre els possibles problemes que puguin sorgir.
- Aplicació lliure: s'ha de valorar l'estabilitat i abast de la comunitat formada entorn de l'aplicació, ja que en cas que l'equip principal de desenvolupadors desaparegui, la seva continuïtat dependrà del nombre de persones que la utilitzin i desenvolupin esporàdicament a tot el món.

**1.6. Selecció de la solució**

Per a posar fi a l'estudi de viabilitat, es triarà una solució d'entre les diferents alternatives estudiades.

La decisió sobre quina és la millor solució (o si cap no ho és) es prendrà tenint en compte la informació acumulada fins al moment:

- Descripció general i abast del projecte.
- Situació actual del sistema.
- Requisits que haurà de complir la solució adoptada.
- Descripció de les solucions alternatives considerades.
- Anàlisi de costos/beneficis de les diferents solucions i riscos associats a cada una d'elles.

**Cas pràctic****Selecció de la solució adoptada en el sistema de còpies de seguretat de Solucions Obertes, SA**

Donada la descripció general del sistema i la seva situació actual, s'han considerat els factors següents a fi d'escollir la solució:

- Requisits plantejats i descripció de cada una de les solucions: totes les solucions cobreixen més o menys els requisits bàsics funcionalment i tècnicament. Quant als aspectes econòmics i legals, la solució GNU/Linux + aplicació lliure és la clara guanyadora.
- Anàlisi costos/beneficis: aquesta anàlisi ha donat com a resultat tres costos entre els quals la solució GNU/Linux + aplicació lliure és la més barata. Atès que els beneficis aportats per cada solució són semblants en termes generals (sens dubte es podrien discutir certs detalls en els quals sí que hi ha diferències significatives, però que no decanten definitivament la balança per una o altra solució), s'ha optat per valorar com a més positiva la solució GNU/Linux + aplicació lliure.
- Riscos: s'han detectat possibles problemes de diferents tipus en cada una de les solucions, els de solució més fàcil són els relacionats amb el sistema operatiu GNU/Linux i l'aplicació de còpies de seguretat lliure (precisament pel seu caràcter marcadament obert en comparació amb la resta).

Es decideix, per tant, que la solució en GNU/Linux + aplicació lliure de còpies de seguretat és la més adequada d'entre totes les considerades.



## 2. Anàlisi del sistema

L'objectiu de la realització de l'**anàlisi del sistema** és, donada la solució escollida d'entre les descrites en l'estudi de viabilitat, la d'especificar-la detalladament (orientat a facilitar el disseny del sistema, fase coberta en el capítol següent).

Els apartats següents descriuen amb més detall cada una de les tasques que es poden efectuar per a fer l'anàlisi del sistema.

### 2.1. Definició del sistema

En aquesta fase de l'anàlisi s'haurà de **descriure el sistema**, establir **com es comunicarà amb d'altres** en cas de ser necessari i **quins usuaris seran representatius** en el seu ús.

Com el lector recordarà, ja en la fase d'estudi de viabilitat es va procedir a descriure el sistema genèricament, i també a definir com afectava la resta de sistemes ja existents (o projectes que es pensava dur a terme). El treball fet en la fase esmentada servirà com a base de les tasques fetes en aquesta anàlisi.

Utilitzant com a punt de partida la descripció dels requisits feta en l'estudi de viabilitat, es **determinaran els requisits exactes del sistema**. Així mateix, s'estudiarà **com es comunica el sistema** amb la resta de sistemes existents (sia rebent o enviat informació a aquests).

## Cas pràctic

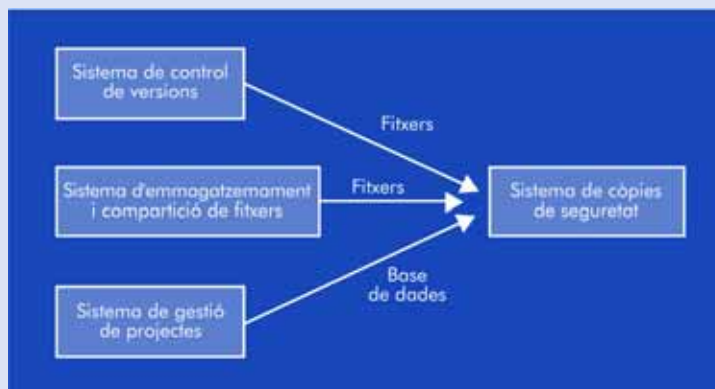
**Requisits exactes del sistema de còpies de seguretat de Solucions Obertes, SA**

El sistema de còpies de seguretat de Solucions Obertes, SA haurà de complir els requisits següents:

- S'hauran de poder fer còpies de servidors remots connectats mitjançant la xarxa local de l'empresa al servidor de còpies de seguretat.
- Les còpies de seguretat s'hauran de poder fer de servidors amb sistemes operatius iguals (GNU/Linux) o diferents del del servidor de còpies de seguretat (d'altres UNIX o Microsoft Windows).
- Les còpies de seguretat han de residir en un servidor que sigui segur, ja que contindran informació molt rellevant per a l'empresa. Aquest servidor haurà de complir les normes de seguretat fixades per a tots els servidors de l'empresa.
- Les còpies de seguretat s'hauran d'emmagatzemar en un format tan obert com sigui possible. En tant que sigui possible, s'intentarà que el format esmentat sigui independent de la plataforma sobre la qual es facin les còpies.
- S'han de poder fer còpies de seguretat completes (p. ex., una a la setmana) i incrementals (p. ex., una al dia).
- La interfície de programació de còpies ha de ser tan intuïtiva i fàcil d'usar com sigui possible. Aquesta interfície podria consistir en una sèrie de fitxers de text modificables pels administradors de les còpies de seguretat.
- S'han de poder restaurar fitxers concrets que resideixin dins de certa còpia de seguretat, no aquesta sencera. Aquest procés es durà a terme mitjançant la interfície del programari de còpies de seguretat, triant aquells fitxers que volem restaurar.

- La periodicitat de realització de les còpies de seguretat ha de ser totalment configurable mitjançant la interfície de programació de còpies de seguretat.
- El programari de còpies de seguretat ha de suportar el maquinari triat com a suport d'emmagatzemament (la unitat de cintes escollida serà la comprada en el projecte de renovació de maquinari).
- La llicència d'ús del programari de còpies de seguretat ha de ser el menys restrictiva possible, en concret haurà de ser de codi obert o lliure.
- La llicència d'ús del sistema operatiu del servidor de còpies de seguretat serà la corresponent a GNU/Linux; és a dir, *GNU General Public License*.
- La despesa en concepte de llicència d'ús del programari de còpies de seguretat ha de ser nul·la.
- La despesa corresponent al sistema operatiu del servidor de còpies de seguretat ha de ser nul·la.
- Les comunicacions existents amb altres sistemes (emmagatzemament i compartició de fitxers, gestió de projectes i control de versions) consistiran en la recollida de la informació de la qual s'han de fer còpies de seguretat per part del servidor de còpies de seguretat. En el primer i tercer cas (sistema d'emmagatzemament i compartició de fitxers, control de versions), serà necessari indicar quins arxius es volen copiar i amb quina periodicitat. En el segon cas (gestió de projectes), l'objectiu bàsic serà fer una còpia de seguretat de la base de dades, fet pel qual serà necessari desallotjar-la com un arxiu que es pugui copiar com la resta.

**Figura 5.** Descripció general del sistema de còpies de seguretat



En la definició del sistema també serà necessari definir l'**entorn tecnològic** del projecte, respecte del qual ja es va incloure informació en l'estudi de viabilitat.

### Cas pràctic

#### Entorn tecnològic del sistema de còpies de seguretat de Solucions Obertes, SA

L'entorn tecnològic del sistema de còpies de seguretat serà el següent:

- Sistema operatiu: GNU/Linux (distribució per determinar).
- Sistema de còpies de seguretat: s'haurà de poder executar en el sistema operatiu GNU/Linux i estar fet en un llenguatge que l'equip de persones de Solucions Obertes, SA conegui (p. ex., C, C++, Perl, Python).
- Desenvolupaments a mida: en el cas que sigui necessari fer algun tipus de desenvolupament, es durà a terme amb tecnologies àmpliament disponibles en qualsevol sistema operatiu UNIX (p. ex., *shell scripts*, *Perl scripts*, etc.).



Per a completar la descripció del sistema, serà necessari fer referència al conjunt d'estàndards i normes que cal considerar en la seva implementació.

#### Cas pràctic

##### Normes que cal seguir en el sistema de còpies de seguretat de Solucions Obertes, SA

Les normes i estàndards que cal seguir en la implementació del sistema de còpies de seguretat seran les següents:

- Quant al sistema operatiu, se seguirà el procés documentat com a "Instal·lació de servidors GNU/Linux " de Solucions Obertes, SA.
- El programari de còpies de seguretat haurà de cobrir els estàndards implementats en les unitats de cinta utilitzades (p. ex., DLT).
- Els possibles desenvolupaments a mida seguiran les normes internes de Solucions Obertes, SA; és a dir, les recollides en el document "Normes de desenvolupament de Solucions Obertes, SA". En aquest document queden recollides les normes que cal seguir en el desenvolupament de qualsevol projecte, com ara la utilització de diagrames UML, format de documentació del codi, etc.

Una vegada descrit el sistema, es procedirà a **identificar aquells usuaris que intervindran en la definició de requisits del sistema i en la seva acceptació definitiva**. És especialment important disposar de la col·laboració dels usuaris al llarg de tot el procés de desenvolupament del sistema.

#### Cas pràctic

##### Identificació d'usuaris del sistema de còpies de seguretat de Solucions Obertes, SA.

El personal involucrat en la definició de requisits i acceptació de la solució final del sistema de còpies de seguretat de Solucions Obertes, SA és:

- Els caps de cada un dels departaments de l'empresa ajudaran a establir paràmetres com la periodicitat necessària de realització de les còpies de seguretat segons el seu nivell d'importància, quins elements s'han de copiar, etc.
- Els administradors del sistema seran aquelles persones que utilitzin el sistema de còpies de seguretat, tant en la programació de les còpies de seguretat com en la seva recuperació. Seran ells els que hauran d'aprovar els requisits funcionals quant a interfície d'ús, versatilitat de programació, etc. i donaran el vistiplau definitiu al sistema.

## 2.2. Establiment de requisits

L'objectiu d'aquesta fase serà **completar els requisits definits anteriorment**, disposant de la informació subministrada pels usuaris. En la mesura del necessari, es **dividirà el sistema en subsistemes** que permetin el seu estudi separatament, a fi de facilitar-ne l'anàlisi.

La comparació de la descripció de cada un dels requisits expressats en aquesta fase del projecte amb el disseny creat amb posterioritat ens permetrà comprovar la correcció del disseny creat.

El primer pas en el procés d'establiment de requisits serà **obtenir els requisits a partir de la informació subministrada pels usuaris**. Els requisits recollits en les reunions mantingudes amb els usuaris triats en la fase anterior seran bàsicament els tipus següents:

- Funcionals (p. ex., mitjançant el sistema de còpies de seguretat s'haurà de poder recuperar un fitxer en concret sense haver de desfer totes les còpies de seguretat).
- Rendiment (p. ex., el procés de recuperació d'un fitxer de qualsevol còpia de seguretat de qualsevol dia de la setmana no pot durar més d'una hora de principi a fi).

- Seguretat (p. ex., només podran recuperar fitxers de les còpies de seguretat aquelles persones que estiguin autoritzades per a això).
- Implantació (p. ex., les còpies de seguretat es desaran en un lloc físicament segur).
- Disponibilitat (p. ex., el sistema de còpies de seguretat s'ha d'utilitzar almenys una vegada a la setmana a fi de comprovar el seu correcte funcionament).

#### Cas pràctic

##### Requisit periodicitat de còpies de seguretat del sistema de còpies de seguretat de Solucions Obertes, SA

Juntament amb els caps de departament de Solucions Obertes, SA, s'ha determinat que s'haurà de fer una còpia diària de tots i cada un dels arxius emmagatzemats en el sistema de compartició de fitxers i en el sistema de control de versions. Així mateix, es farà una còpia també diària de la informació emmagatzemada en la base de dades del sistema de gestió de projectes. Mitjançant entrevistes amb els administradors del sistema, s'ha determinat que aquestes còpies de seguretat diàries s'han de fer de manera automàtica en una hora en què la utilització dels recursos informàtics sigui mínima, per exemple, de matinada.

Una vegada descrit cada un dels requisits, es procedirà a especificar els casos d'ús de cada un d'ells. Els casos d'ús, a més de la descripció en si mateixa del problema, cobriran com els usuaris interactuaran amb el sistema, quines interfícies utilitzaran i com es tractaran les condicions d'error.

#### Cas pràctic

##### Cas d'ús de periodicitat de còpies de seguretat del sistema de còpies de seguretat de Solucions Obertes, SA

La periodicitat de les còpies de seguretat dels sistemes d'emmagatzemament i compartició de fitxers i de gestió de projectes serà totalment independent de la seva utilització (els usuaris no hauran de dur a terme cap acció especial).



**Cas pràctic****Associacions del cas d'ús de periodicitat en el sistema de còpies de seguretat de Solucions Obertes, SA**

El cas d'ús que descriu com es tractarà la periodicitat de les còpies de seguretat en el sistema està directament relacionat amb el cas d'ús que descriu com es faran les còpies de seguretat del sistema de gestió de projectes.

La informació del sistema de gestió de projectes estarà emmagatzemada en una base de dades de la qual s'haurà de fer un bolcat al fitxer per a poder-la copiar. Aquest bolcat s'haurà de fer d'acord amb la política de periodicitats establerta en el cas d'ús de "periodicitat de còpies de seguretat". A més, el fitxer que contingui el bolcat s'haurà de gravar periòdicament en un directori del qual es facin còpies de seguretat.

**2.3. Definició d'interfícies d'usuari**

En aquesta fase de l'anàlisi s'especifiquen com seran les diferents interfícies que hi haurà entre el sistema que descrivim i els seus usuaris tenint en compte els diferents perfils d'usuaris, la flexibilitat necessària, els tipus d'accions que cal dur a terme, etc.

El primer pas en la definició de les interfícies d'usuari serà **definir els perfils d'usuaris que utilitzarà el sistema**. D'aquesta manera, es podrà descriure posteriorment a quins tipus d'interfícies accediran cada un d'ells.

**Cas pràctic****Perfils d'usuaris de la utilitat de bolcat de base de dades del sistema de còpies de seguretat de Solucions Obertes, SA**

La utilitat que faci el bolcat al fitxer de la base de dades del sistema de gestió de projectes (a fi de fer la seva còpia de seguretat) serà utilitzada per usuaris que compleixin les funcions d'administradors de sistema, amb la qual cosa tindran les característiques següents:

- Usuaris amb un perfil tècnic.
- Usuaris acostumats a utilitzar programes sobre terminals, mitjançant intèrprets d'ordres amb interfícies textuais.
- Usuaris acostumats a utilitzar fitxers de configuració en format text.
- Usuaris acostumats a utilitzar eines en entorns UNIX.

A continuació, s'hauran d'**especificar els principis generals de la interfície d'usuari**, per exemple, si s'utilitzaran interfícies de text o gràfiques, com es mostraran els missatges d'error, com s'obtindrà ajuda, etc.

#### Cas pràctic

##### Principis generals de la interfície d'usuari de la utilitat de bolcat de base de dades del sistema de còpies de seguretat de Solucions Obertes, SA

La utilitat que faci el bolcat al fitxer de la base de dades del sistema de gestió de projectes (a fi de fer la seva còpia de seguretat) tindrà les característiques següents:

- La programació de la utilitat es farà mitjançant fitxers de configuració en format text.
- El fitxer de configuració en format text seguirà els estàndards més habituals utilitzats en aplicacions UNIX clàssiques (p. ex., separació de camps amb el caràcter ":", continuació de línies amb el caràcter "\", llista de dades amb el caràcter ",", etc.).
- Els missatges d'error es mostraran per pantalla, amb un codi numèric associat, llevat d'aquells que impliquin que el bolcat no s'ha pogut fer (en aquest cas, el missatge d'error s'enviarà per correu electrònic a una adreça indicada).

- Els fitxers de configuració permetran la inclusió de línies amb comentaris.
- L'ajuda del programa de bolcat s'obté per mitjà dels canals habituals en l'administració de sistemes (p. ex., utilitat *man*).

Una vegada identificades les característiques generals de la interfície d'usuari, es passarà a **especificar-la per a cada un dels casos d'ús** definits en l'apartat anterior.

#### Cas pràctic

##### Interfície d'usuari de la utilitat de bolcat de base de dades del sistema de còpies de seguretat de Solucions Obertes, SA

La utilitat de bolcat s'haurà de configurar perquè emmagatzemi el contingut d'una certa base de dades en un fitxer determinat. Per a això, caldrà configurar mitjançant un fitxer de text dades com l'usuari i la contrasenya que s'haurà d'utilitzar per a accedir a la base de dades, en quin servidor resideix, etc.

#### Exemple

Exemple de fitxer de configuració:

```
# Línia amb comentaris.  
user = usuari:contrasenya  
host = projects.example.com  
database = projects  
e-mail = alberto@cometatech.com,marcg@cometatech.com  
file = /var/local/dbDump  
compression = true
```

## 2.4. Especificació del pla de proves

Per a acabar la fase d'anàlisi es procedirà a fer l'especificació del pla de proves, que ens **servirà per a establir si el sistema compleix els requisits establerts pels usuaris**.

Es podran fer **proves** del sistema a **diversos nivells**:

- Proves unitàries (p. ex., prova de connexió a la base de dades del sistema de gestió de projectes).
- Proves d'integració (p. ex., prova de còpia de seguretat del fitxer producte del bolcat de la base de dades del sistema de gestió de projectes).
- Proves de sistema (p. ex., prova de la còpia incremental diària dels sistemes d'emmagatzematge de fitxers i del de gestió de projectes).
- Proves d'implantació (p. ex., prova de còpia de seguretat sobre la unitat de cintes de la qual es disposa).
- Proves d'acceptació (p. ex., còpia completa del sistema que validaran els usuaris corresponents). Aquest conjunt de proves és crític, ja que serà el que permetrà validar el sistema complet. A més del funcionament correcte del sistema, haurà de tenir en compte paràmetres com la seguretat, el rendiment, la disponibilitat, etc.

Per a cada una de les proves que cal fer, se n'haurà de definir l'abast (p. ex., usuaris implicats en les proves, productes de les proves, criteris d'acceptació de les proves, etc.), i els **requisits en l'entorn** de proves (maquinari necessari, llibreries disponibles, configuració d'accessos, etc.).

#### Cas pràctic

**Prova d'integració del programari de còpies de seguretat de Solucions Obertes, SA amb la utilitat de bolcat de la base de dades del sistema de gestió de projectes.**

La prova tindrà les característiques següents:

- Permetrà als administradors del sistema comprovar que la còpia del fitxer del bolcat de la base de dades del sistema de gestió de projectes s'ha fet correctament.



- Com a producte de la prova, obtindrem un fitxer amb les diferències existents entre l'arxiu emmagatzemat a la còpia de seguretat i l'arxiu producte del bolcat de la base de dades.
- La prova es donarà per correcta quan el fitxer de diferències entre arxius sigui buit.

A fi de poder fer la prova, serà necessari:

- Disposar dels servidors de còpies de seguretat i de sistema de gestió de projectes.
- Disposar d'una base de dades de gestió de projectes amb contingut real.
- Disposar d'un accés remot a la base de dades del sistema de gestió de projectes.
- Disposar d'una unitat de cinta sobre la qual es faran les còpies de seguretat.



### 3. Disseny del sistema

L'objectiu de la fase de **disseny** d'un sistema d'informació és obtenir els **models i les especificacions** que el defineixen a partir de l'anàlisi feta en la fase anterior. Les activitats que duguem a terme en aquesta fase ens permetran determinar les especificacions de desenvolupament i integració, i també definir l'entorn de proves i implantació necessaris per al seu funcionament correcte.

Concretament, els resultats que haurem d'obtenir en aquesta fase seran els següents:

- La definició del model arquitectònic del sistema. Mitjançant la identificació dels seus components, les seves interaccions i l'ajuda d'eines de modelatge, obtindrem un mapa dels subsistemes i recursos que intervenen en tots els processos.
- Les especificacions i estàndards que s'usaran tant en aquesta mateixa fase com durant el desenvolupament del sistema.
- La identificació de cada subsistema, els seus requisits d'integració, llicència i funcionalitats cobertes.
- Els casos d'ús aplicats dels subsistemes anteriorment identificats, degudament revisats per a reflectir el model i especificacions definits.
- Els components, classes o interfícies que haurem de construir en la fase de desenvolupament.
- Els requisits necessaris per a procedir amb èxit a la implantació del sistema.

Com veiem, molts conceptes i especificacions es determinaran en aquesta fase, i encara que algunes metodologies recents aconsellen

barrejar-la amb la fase de desenvolupament en un cicle combinat de disseny i construcció iteratiu, amb l'objectiu d'obtenir resultats aviat o d'identificar errors en el disseny a temps, és obvi que totes les decisions quant a especificacions, estàndards o subsistemes que prenguem aquí facilitaràn totes les tasques futures.

És especialment important la identificació dels components que cal usar i les seves llicències, ja que aquests poden determinar part de la funcionalitat, la necessitat de desenvolupaments interns de comunicació entre subsistemes o el tipus de llicència amb què haurem de distribuir (si escau) el resultat del nostre projecte.

### 3.1. Arquitectura

La definició de l'**arquitectura** del sistema és el primer pas per a **identificar-ne els components** i dóna lloc a les fases següents de disseny en què aprofundirem en cada un d'ells. L'objectiu és disposar d'un conjunt de documents i diagrames complets i concisos que siguin comprensibles per a la direcció i alhora serveixin de base per a aprofundir en el disseny del sistema.

Abans de fer el disseny més detallat del sistema, serà necessari definir les normes i estàndards de disseny i construcció.

Una vegada acordats els estàndards de disseny, podrem aprofundir ja i determinar els subsistemes, repetint el mateix procés que seguim amb l'arquitectura general del sistema, aquesta vegada amb més granularitat en cada un d'ells.

#### 3.1.1. Definició de nivells d'arquitectura

Hi ha diverses maneres de veure o entendre l'arquitectura d'un sistema:

- **Arquitectura conceptual:** el seu propòsit és dirigir l'atenció sobre els grans blocs que formen el sistema, sense entrar en detalls, i identificar les relacions entre aquests blocs. És molt útil per a comunicar a la direcció o a departaments no tècnics una visió global del sistema.

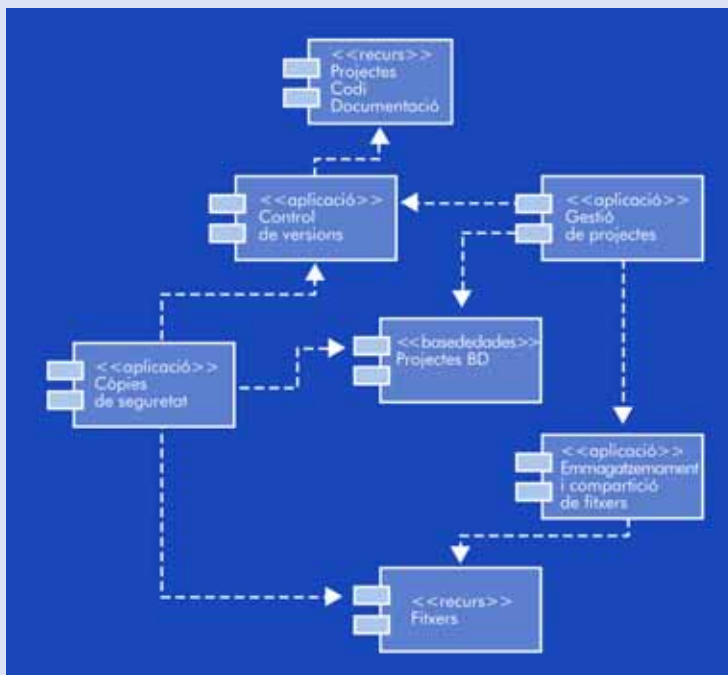
- Arquitectura lògica: afegeix detall a l'anterior i incorpora la definició de les interfícies de comunicacions entre els components, la qual cosa permetrà als desenvolupadors de cada component treballar sense dependències entre ells.

Cas pràctic

Definició de l'arquitectura de la nova infraestructura programari de Solucions Obertes, SA

Per a expressar l'arquitectura de la nova infraestructura, usarem la notació UML en els diagrames, i usarem targetes CRC (classe-responsabilitat-col·laborador) de suport.

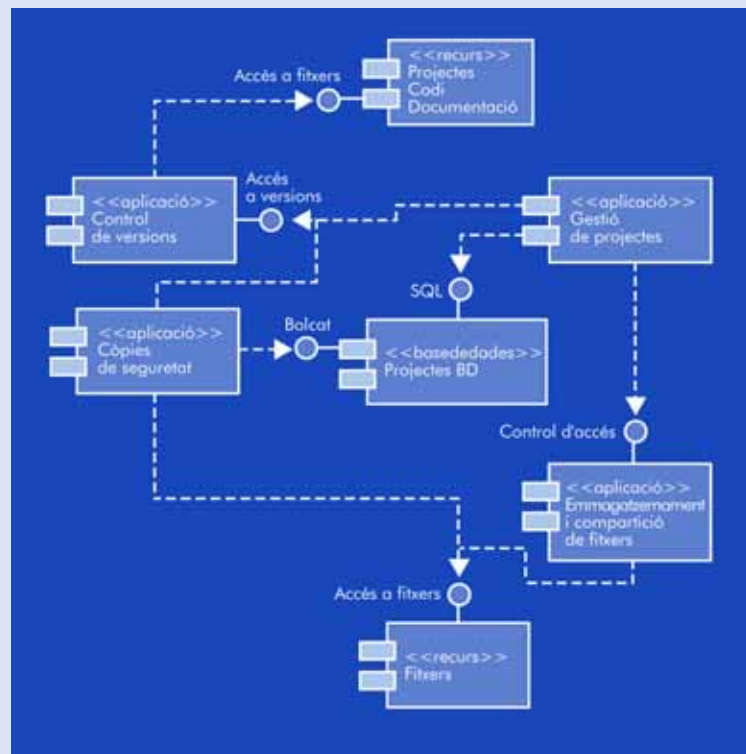
Figura 7. Diagrama UML de components



En el diagrama anterior veiem els components del sistema i els connectors que els uneixen. Aquests connectors indiquen que algun tipus de comunicació es produeix entre ells. És positiu ja barrejar components de negoci i components tècnics en aquest diagrama (que s'identificaran mitjançant els estereotipis `<<basedades>>` o `<<aplicació>>` o d'altres).

Una vegada consensuada aquesta visió general del sistema, passarem a aprofundir en les interfícies dels components per obtenir l'arquitectura lògica del sistema. Per a això, estendrem el diagrama de components anterior detallant els processos de comunicació.

**Figura 8.** Diagrama UML de components amb interfícies



Com a suport a la generació del diagrama anterior o per a consensuar les interfícies, podem utilitzar les targetes CRC.

**Taula 1.**

Control de versions	
Control de versions Proporciona versió de treball Extreu versions anteriors i diferències Emmagatzema versions i canvis Etiqueta versions Coneix projectes	Fitxers

- En la part superior figura el nom del component.

- En la columna esquerra haurem de reflectir tot el que el component sap o fa sobre ell mateix. S'hi inclourà tot allò que creiem que és de la seva responsabilitat i la informació que haurà de mantenir.
- En la part dreta figuraran els components amb qui es relaciona per a poder dur a terme les responsabilitats de la part esquerra.

Aquestes targetes són molt utilitzades en les metodologies eXtreme Programming i Agile, i permeten "crear" el diagrama de components del sistema dinàmicament a sobre d'una taula simplement situant les targetes pròximes o llunyanes les unes de les altres segons el seu grau de comunicació, per consensuar així una visió general de l'arquitectura lògica del sistema durant una reunió.

**Figura 9.** Exemple de situació de targetes CRC



### 3.1.2. Especificació d'estàndards, normes de disseny i construcció

És important acordar amb les persones encarregades del disseny del sistema i amb els equips que procediran a construir-lo unes normes a seguir en la notació de diagrames i documents. Aquestes normes poden estar determinades per estàndards o per recomanacions, o bé ser d'ús i creació internes. Òbviament, sempre és recomanable basar-se en estàndards, ja que facilitarà la comunicació, reusabilitat i comprensió per part d'entitats externes o acabades d'incorporar a l'equip.

Haurem de definir:

- Format i plantilla dels documents de disseny.
- Notació que s'usarà en els diagrames de disseny.
- Recomanacions quant a l'estil, idioma i format de la documentació tècnica.

#### Cas pràctic

##### Definició del conjunt de normes i notacions de la nova infraestructura de programari de Solucions Obertes, SA

És convenient que tots els documents creats d'ara endavant i que seran objecte de revisió per part d'equips diferents comparteixin unes característiques i mantinguin un format coherent. Per a això, després d'estudiar els estàndards i les recomanacions sobre el tema, s'arriba a les conclusions següents:

- Documents de disseny: aquests documents els han de poder consultar tant el personal tècnic implicat, com el personal no tècnic que els pugui revisar o consultar. S'acorda que es treballin en format RTF (*rich text format*) i que la versió més recent estigui simultàniament en PDF per a consultar-la. Es crearà una plantilla que contingui en la primera pàgina:
  - Títol del document.
  - Responsable del document.
  - Llista d'autors que hi han intervingut i la data de la seva primera intervenció.
  - Llista resumida de canvis introduïts en el document a mesura que es produeixin (canvi, data i autor).
- Diagrames de disseny: per als diagrames de disseny s'acorda usar la notació UML (<http://www.omg.org/uml/>) en la seva versió 1.5, definida per l'Object Management Group ([www.omg.org](http://www.omg.org)).



- Documentació tècnica: la documentació tècnica serà possiblement la que experimentarà més revisions i contindrà també enllaços a documentacions de les eines usades, especificacions de programació (API), etc., per la qual cosa es recomana usar un format tan flexible com sigui possible i integrable a les pròpies eines de desenvolupament que s'us: es decideix fer servir DocBook ([www.docbook.org](http://www.docbook.org), <http://www.oasis-open.org/docbook/>), que ens permetrà:
  - Partir un document en diversos fitxers estructurats, susceptibles de ser revisats independentment.
  - Incloure fàcilment referències a altres documents (enllaços http, figures, etc.).
  - Generar fàcilment diversos formats per a visualitzar-los (PDF, HTML) i amb la possibilitat de separar el contingut del document del seu format.
  - Usar independentment un editor, ja que és una implementació d'XML i, per tant, modificable en qualsevol editor de text.
  - Incorporar documentació continguda en el codi font generat en la fase de desenvolupament, de manera automàtica en molts casos.

Es pot destacar que en prendre les decisions s'ha donat importància a la implantació del format o notació en la indústria i a la seva accessibilitat; és a dir, a la disponibilitat d'exemples i documentació, i també d'un ampli conjunt d'eines que treballin amb ells.

### 3.1.3. Identificació de subsistemes

Per a reduir la complexitat que representaria dissenyar detalladament tot el sistema, s'haurà de dividir en subsistemes per a facilitar-ne la comprensió, revisió i reutilització.

Per a fer aquesta divisió, podem tenir en compte diversos aspectes dels components:

- Funcionalitat comuna o relacionada per característiques de l'execució.
- Gestió de dades, accés a dades comunes.
- Integració en una interfície d'usuari comuna.
- Optimització de línies de comunicacions o recursos.

#### Cas pràctic

#### Identificació i disseny dels subsistemes de la nova infraestructura programari de Solucions Obertes, SA

Fent una primera divisió per funcionalitat, identifiquem clarament els subsistemes següents:

- Subsistema de còpies de seguretat
- Subsistema de gestió de fitxers
- Subsistema de gestió de projectes
- Subsistema de control de desenvolupament i versions

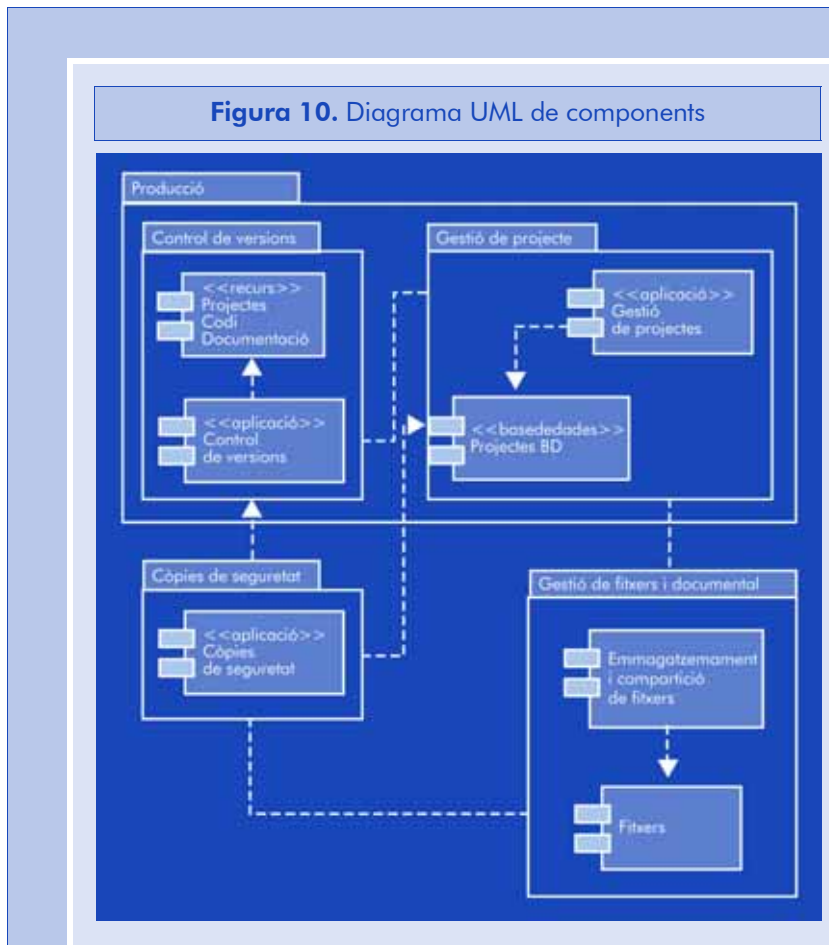
Si apliquem el criteri d'accés a dades en la identificació de subsistemes:

- Subsistema de producció. Inclou les funcionalitats de gestió de projectes, desenvolupament i versions.
- Subsistema d'accés documental. Inclou la funcionalitat d'accés a fitxers compartits i a la gestió de projectes amb els seus documents relacionats.

Segons aquest criteri, el subsistema de còpies de seguretat no seria rellevant, ja que accedeix a totes les dades sense afegir-hi informació ni interpretar-les.

Aplicant el criteri d'interfície comuna, en aquest cas seria possible incloure tots els subsistemes vistos en una interfície integrada i obtenir així un únic subsistema. Així doncs, aquest criteri no serveix per a identificar subsistemes en aquest cas.

El criteri d'optimització de recursos o línies de comunicació tampoc no és aplicable en aquest cas.



En aquest diagrama hem utilitzat el diagrama de components vist anteriorment i hem incorporat els diferents subsistemes identificats en forma de paquets que inclouen un o diversos components de l'arquitectura del sistema.

Disposem ara d'un mapa complet de l'arquitectura global del sistema. Pròximament, haurem d'aprofundir en cada subsistema i els seus components i la millor manera de fer-ho serà mitjançant la revisió dels casos d'ús vistos en l'anàlisi del sistema.

### 3.2. Casos d'ús reals

Una vegada identificats els subsistemes, és el torn de revisar els **casos d'ús** fets en la fase d'anàlisi i que **determinin les operacions** que hauran d'implementar les interfícies de cada un d'ells.

Així doncs, a partir dels escenaris recollits en la fase d'anàlisi, determinarem quins subsistemes hi estan implicats i dissenyarem el seu funcionament tenint en compte:

- L'entorn tecnològic en què s'apliquen.
- Les excepcions que es produeixen en cada cas d'ús.
- Detalls relacionats amb la implementació que ja puguem identificar en aquesta fase.
- Restriccions o característiques de la interfície d'usuari.
- Nous requisits que puguem identificar.

Si el sistema que dissenyem se centra en el desenvolupament, els casos d'ús dels subsistemes hauran d'incorporar la definició de les classes o objectes i, per tant, els diagrames que obtinguem hauran de representar també la seva interacció.

Així doncs, durant aquesta fase establirem les característiques, revisarem els requisits i dissenyarem les classes (amb els seus atributs, operacions i relacions) de tot el sistema. De manera natural durant el procés, obtindrem també el disseny de les proves que asseguraran el bon funcionament del sistema durant el desenvolupament i les seves condicions d'implantació.

### **3.2.1. Revisió de casos d'ús per subsistema**

Per a cada cas d'ús haurem de definir:

- Subsistemes i actors que hi intervenen.
- Missatges que intercanvien els objectes.

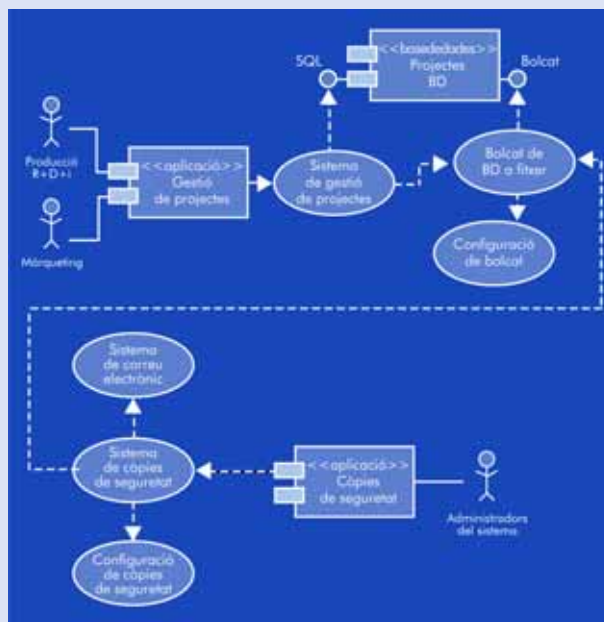
La definició dels missatges ens servirà per a comprovar i detallar les interfícies de cada subsistema tenint en compte tots els casos d'ús en què intervé i així completar la definició de subsistemes feta en fases anteriors.

Cas pràctic

Revisió de casos d'ús del subsistema de còpies de seguretat de Solucions Obertes, SA

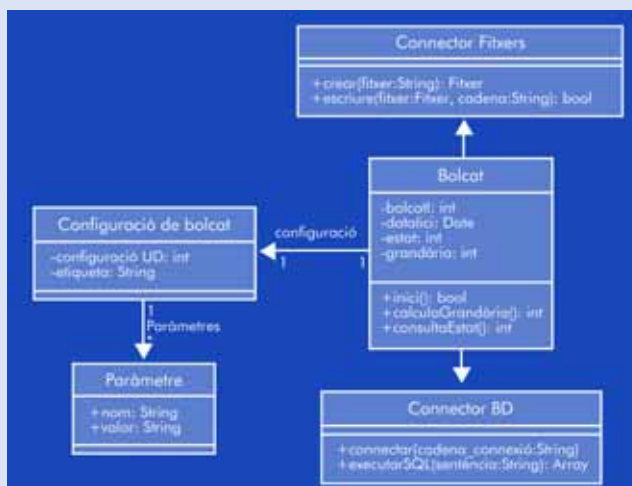
A continuació, completarem la definició de subsistemes feta amb la revisió dels casos d'ús implicats en la gestió de còpies de seguretat.

Figura 11. Diagrama UML de revisió de cas d'ús amb components implicats



A continuació, enumerarem els missatges que intervenen en el bolcat de la base de dades al fitxer i els sistemes o objectes amb què s'intercanvien aquests missatges, la qual cosa ens portarà cap al diagrama de classes del subsistema, on es representen els objectes que hi intervenen i els seus atributs i mètodes.

Figura 12. Diagrama UML de classes



### 3.2.2. Especificacions de desenvolupament i proves

Arribats a aquest punt, estarem en condicions d'establir les condicions i característiques de l'entorn de desenvolupament en els termes següents:

- Entorn tecnològic: maquinari, programari i comunicacions.
- Eines de desenvolupament: IDE, generadors de codi, compiladors, etc.
- Eines de documentació.
- Restriccions tècniques.
- Requisits de seguretat de l'entorn.

També haurem de ser capaços de definir les proves necessàries que s'hauran de fer per a assegurar el funcionament del sistema una vegada implantat. Aquestes s'haurien de definir com a proves unitàries, és a dir, proves amb el mínim nivell possible de dependència entre elles per a permetre un desenvolupament o una integració programari per components, on cada equip pugui treballar i provar independentment, deixant per a la fase final les proves d'integració.

L'especificació de les proves unitàries es divideix en proves de caixa blanca i proves de caixa negra:

- Caixa negra: es considera el component des del punt de vista funcional, analitzant les seves entrades i sortides, i comparant totes les seves possibilitats amb els resultats esperats.
- Caixa blanca: es considera el component com una estructura amb una seqüència lògica d'esdeveniments i se'n comprova la validesa, el codi no usat, comprovacions previstes, etc.

Normalment, s'usarà una combinació dels dos tipus de proves, segons el component o la seva funcionalitat. Tradicionalment, es tendeix a fer les proves dels components una vegada desenvolupats. Les metodologies més recents recomanen invertir aquest procés, i justifiquen la realització de les proves prèviament a les dels mateixos components programari per les raons següents:

- En disposar de la funcionalitat requerida dels components, estem en disposició de dissenyar-les.

- Des del primer moment podem provar que els nostres components compleixen o van complint les funcionalitats.
- És millor crear un component amb l'objectiu de passar les proves dissenyades, que crear-ne un que tingui la funcionalitat requerida. D'aquesta manera s'evita que els programadors introdueixin efectes col·laterals o creguin que determinada funcionalitat també serà necessària i la introdueixin.
- S'ha demostrat que si les proves estan ben dissenyades i són completes, els programadors trigaran igual o menys a crear un component que compleixi les funcionalitats que un que simplement passi els tests dissenyats.
- Encara que tradicionalment es deixen les proves per al final del desenvolupament, aquestes són les que més freqüentment provoquen retards en el projecte. Aquest enfocament de proves unitàries mitiga aquests retards i optimitza el desenvolupament que ara s'enfoca directament a obtenir resultats, entesos com a tests passats satisfactòriament.

#### Cas pràctic

##### **Definició de les especificacions de desenvolupament i proves del subsistema de còpies de seguretat de Solucions Obertes, SA**

El subsistema de còpies de seguretat ha de ser capaç de copiar i restaurar tota la informació de la resta de sistemes. Alguns dels recursos on s'emmagatzema la informació en permeten l'accés i la recuperació directa, en un format intel·ligible pel sistema de còpies. D'altres exigiran un desenvolupament per a proporcionar les seves dades en un format susceptible de ser copiat i ser restaurat sense afectar-ne la integritat. És el cas de la base de dades.

Així doncs, s'haurà de desenvolupar:

- Una interfície que permeti el bolcat d'una imatge de la base de dades en un instant determinat. Aquesta interfície haurà de poder funcionar a petició d'un usuari o bé de manera desatesa quan es facin les còpies de seguretat periòdicament.
- Una interfície que permeti la substitució total o parcial de les seves dades per d'altres de provinents de la còpia de seguretat.

En determinar el llenguatge triat per al desenvolupament, es consideren les alternatives següents:

- Perl (<http://www.perl.org>): llenguatge de *script* molt implantat en tasques administratives. Molt flexible i versàtil, disposa de multitud de llibreries de suport i una comunitat d'usuaris extensa. Permet orientació a objectes.
- Bash (<http://www.gnu.org/software/bash/>): llenguatge de *script* sobre el mateix intèrpret d'ordres del sistema operatiu. Encara que molt potent, està molt condicionat pel mateix entorn d'execució (variables d'entorn de l'usuari, etc.) i la seva funcionalitat és menor que la d'un llenguatge de programació clàssic.
- Python (<http://www.python.org>): llenguatge de *script* d'ús cada vegada més freqüent. Permet una fàcil creació d'interfícies gràfiques, i disposa de multitud de llibreries de suport. El seu intèrpret i el mateix llenguatge no estan tan difosos com la resta d'alternatives.

Per la seva àmplia acceptació i disponibilitat de llibreries de suport, i també la seva integració amb el sistema operatiu, Solucions Obertes, SA decideix utilitzar el llenguatge Perl.

En determinar l'entorn de desenvolupament, es consideren les alternatives següents:

- Jedit (<http://www.jedit.org>): es tracta d'un entorn molt potent desenvolupat en Java i per tant multiplataforma, amb un conjunt de connectors que proporciona funcionalitats addicionals com la detecció de sintaxi i navegació avançada pel codi integració amb sistemes de control de versions, etc.
- Vim (<http://www.vim.org>): es tracta d'una extensió de l'editor 'vi' inclòs en gairebé tots els sistemes Unix. Ha estat millorat per a facilitar-ne l'ús, i disposa d'interfície gràfica i interfície de text.



- Emacs (<http://www.gnu.org/software/emacs/>): es tracta d'una eina molt potent i complexa. En el seu origen és un intèrpret de Lisp amb funcionalitats d'edició de textos. La seva corba d'aprenentatge és molt pronunciada.

Per la seva facilitat d'ús i potència, amb possibilitat de desenvolupament d'extensions i d'incorporació de les moltes ja existents, es decideix utilitzar l'entorn de desenvolupament Jedit.

La resta d'especificacions de desenvolupament provenen de les preses anteriorment, ja que el format de documentació, i també el marc de treball de les proves unitàries, sorgeixen de manera natural a partir del llenguatge de programació escollit.

- Marc de treball de proves unitàries: mòdul Perl Test: SimpleUnit
- Documentació del mateix desenvolupament: format POD (*plain old documentation*)
- Documentació tècnica de les interfícies i el seu ús: DocBook.

A continuació, haurem d'enumerar les proves unitàries, extrems de les funcionalitats i interfícies del subsistema:

- Connexió i desconnexió al gestor de base de dades.
- Obtenció de les bases de dades que s'han de bolcar.
- Connexió i desconnexió a cada base de dades que s'ha de bolcar.
- Inici i final de transacció amb la base de dades per a evitar dades corruptes durant el procés de bolcat.
- Obtenció de les dades d'estructura de cada taula de cada base de dades.
- Obtenció de les dades de cada taula de cada base de dades.
- Compresió de les dades obtingudes per a optimitzar recursos.
- Emmagatzematge de les dades.
- Obtenció de les dades emmagatzemades.

- Descompressió de les dades emmagatzemades.
- Substitució de l'estructura actual per l'obtinguda de les dades emmagatzemades.
- Substitució de les dades actuals per les obtingudes de les dades emmagatzemades.
- Comprovació de la integritat de la base de dades.

Per a cada una d'aquestes proves, haurem de definir els seus possibles paràmetres o informació d'entrada, i els seus possibles resultats o informació de sortida. Això ens permetrà més endavant programar cada un dels tests sota el marc de treball de tests unitaris triat.

### **3.2.3. Requisits d'implantació**

Els requisits d'implantació seran els que hagi de complir cada component o subsistema quan treballi en l'entorn real conjuntament amb la resta de sistemes. Per entorn no entendrem únicament entorn tecnològic, sinó que tindrem en compte també els usuaris del subsistema.

Així doncs, la implantació del subsistema tindrà implicacions per als usuaris i caldrà determinar si els seus coneixements actuals són suficients per a usar el nou subsistema, o bé haurem de desenvolupar un pla de formació.

De la mateixa manera, des del punt de vista tecnològic, haurem de determinar les condicions de l'entorn on implantarem el subsistema, la capacitat actual dels seus recursos i les seves condicions de funcionament per a comprovar que el nostre nou subsistema no esgotarà els recursos existents i no provocarà canvis en els nivells de servei de la resta del sistema.

El document que reculli els requisits d'implantació haurà de preveure:

- Gestió de la documentació. Qui hi tindrà accés i en quin format i condicions.
- Formació dels usuaris.

- Necessitats de maquinari i programari.
- Necessitats de comunicacions.
- Restriccions de rendiment que tinguin lloc en l'entorn d'implantació.

Aquest document es tindrà en compte en la fase d'implantació, i pot preveure casos addicionals com la migració del sistema, la recuperació davant desastres, les alternatives o les possibles ampliacions d'alguns dels recursos més crítics.

Al seu torn, aquest conjunt de requeriments serà de gran utilitat per a les proves d'implantació i integració del subsistema en el seu entorn de funcionament.

#### Cas pràctic

##### Definició dels requisits d'implantació del subsistema de còpies de seguretat de Solucions Obertes, SA

Ja hem vist que la implantació l'haurèm de fer sota dos enfocaments, el de l'usuari del subsistema i el tecnològic i els seus recursos.

Des del punt de vista de l'usuari serà necessari definir un responsable de la política de còpies de seguretat de l'empresa. Aquesta figura tindrà les responsabilitats següents:

- Conèixer el funcionament del subsistema, a la documentació del qual tindrà accés.
- Dissenyar la política de còpies de seguretat i consensuar-la amb els responsables de la resta de sistemes que gestionen els recursos dels quals s'han de fer còpies de seguretat.
- Comprovar i monitorar el correcte funcionament del subsistema.
- Conèixer els possibles riscos i com es poden solucionar i incorporar aquestes actuacions dins de la política de còpies de seguretat de l'empresa.

A causa de la importància que té aquest subsistema per al funcionament de l'empresa, i de la urgència amb què se solen presentar alguns dels seus casos d'ús, seria convenient nomenar un segon responsable que conegui la política de còpies de seguretat i els procediments necessaris per a suplantar el primer responsable si aquest no pogués atendre la incidència.

Des del punt de vista tecnològic, la implantació del subsistema de còpies de seguretat tindrà un impacte sobre tots els recursos d'emmagatzemament de l'empresa.

- El sistema de fitxers que emmagatzema els documents de l'empresa haurà de ser accessible pel servei que fa les còpies de seguretat, sense que el procés de còpia el modifiqui.
- El subsistema de control de versions haurà de ser accessible pel servei que fa les còpies de seguretat, al qual permetrà l'accés històric (a totes les versions) i l'accés a la versió actual de tots els projectes emmagatzemats.
- La base de dades de projectes haurà de ser accessible pel servei que fa les còpies de seguretat, i els canvis en la seva forma d'accés, usuaris i contrasenyes hauran de ser comunicats al responsable del servei per a evitar interrompre la prestació del servei.
- Prèviament a cada procés de còpia, es comprovaran les condicions inicials del servei, com ara:
  - Prevenció per evitar que modificacions simultànies al procés de còpia provoquin un error en el servei.
  - Dimensionament de recursos per assegurar que el procés de còpia comença i acaba correctament.
- Establiment d'horaris i durada del procés de còpia per evitar que afecti el funcionament normal de l'empresa i de la prestació dels serveis de la resta de subsistemes.
- Protecció del subsistema de còpia per evitar que un error en un dels seus subprocessos afecti el funcionament de la resta.

De tot aquest conjunt de requisits, extraïem les proves unitàries següents per a assegurar la correcta implantació i integració del component de bolcat de la base de dades.

- Obtenció de la mida de la base de dades i del seu temps de còpia.

- Obtenció del temps total de restauració de la base de dades.
- Ús simultani de l'aplicació de gestió de projectes durant el procés de còpia.
- Sistema de registrament i comunicació d'incidències al responsable.
- Integració del procés de còpia de la base de dades amb la resta de subprocessos de còpia.



## 4. Desenvolupament

L'objectiu de la fase de desenvolupament és **construir ordenadament el sistema** del qual s'ha avaluat la viabilitat, que s'ha analitzat i s'ha dissenyat.

L'inici del desenvolupament es produeix, en metodologies tradicionals, quan les fases anteriors s'han completat satisfactòriament i en la seva totalitat. Metodologies més recents, i en les quals es dóna preferència a l'*agilitat* del cicle de vida del projecte, aconsellen passar a aquesta fase tan aviat com sigui possible, avançant simultàniament amb l'anàlisi del sistema.

No obstant això, sempre és convenient adaptar les metodologies a les nostres necessitats, i sempre és interessant, igual que en molts altres àmbits del programari, utilitzar els models, les notacions o els mòduls dins d'una metodologia molt extensa, que s'adaptin a la nostra realitat.

Si part dels desenvolupadors s'han d'implicar en el disseny, pot ser avantatjós que es comenci el desenvolupament d'un subsistema, alhora que es treballa en el disseny d'un altre, i potser se serà capaç de detectar necessitats d'implantació o excepcions que d'una altra manera provocarien una revisió de part del disseny amb les conseqüències que podria tenir en revisar altres subsistemes o components.

Hi ha, tanmateix, un conjunt d'especificacions que concerneixen el desenvolupament, que han de ser definides quan es comenci aquesta fase, independentment del ritme que vulguem imprimir al projecte.

Haurem de ser capaços de planificar l'inici i el final del desenvolupament sincronitzat de les diferents activitats que al final deixaran el projecte en condicions d'implantar-lo, com són l'abast del mateix desenvolupament, els components de tercers que s'han d'usar o adquirir, les seves proves, els manuals d'usuari, la documentació, la formació, etc.

## 4.1. Planificació de les activitats d'integració de sistema

En aquest punt ja hem acumulat informació sobre què necessita ser desenvolupat, quin tipus de components de programari integrarem en el nostre sistema, quines eines utilitzarem, en quin entorn, etc.

Les activitats de desenvolupament que ens permetran assolir l'objectiu plantejat són:

- Concretar versions o alternatives dels components de programari, serveis o llibreries que usarem.
- Estudiar aquests components, serveis o llibreries.
- Implantar l'entorn de desenvolupament.
- Desenvolupar les proves unitàries.
- Desenvolupar els components necessaris.
- Fer la documentació.
- Planificar la formació a usuaris del sistema.
- Desenvolupar les proves d'integració del sistema.
- Aprovar el sistema.

L'objectiu final d'aquesta fase és **aprovar el sistema perquè es pugui implantar**. Per tant, tota la resta d'activitats s'ha de planificar amb vista al compliment de les condicions de la seva aprovació.

L'ordre de les activitats no té per què ser seqüencial, i en el seu grau de paral·lisme estarà molt relacionat amb la diversitat de recursos i de perfils que intervinguin en el desenvolupament. Tot i així, hi ha activitats que milloren el seu desplegament desenvolupant-se en paral·lel (com ara el mateix desenvolupament de components, la seva documentació tècnica i les seves proves unitàries) i altres en les quals el seu inici depèn de resultats obtinguts en altres activitats.

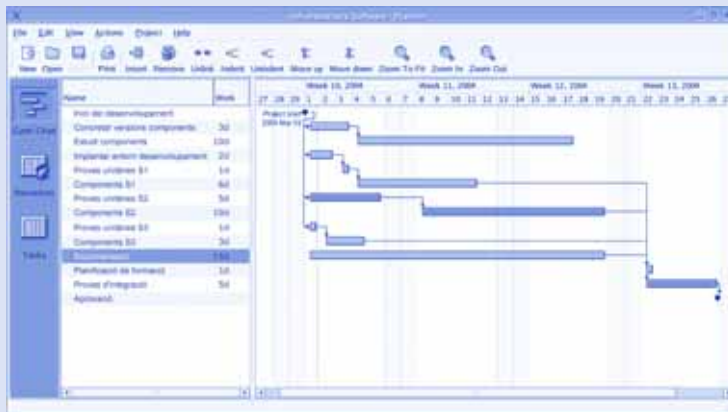


Cas pràctic

**Planificació del desenvolupament de la infraestructura del programari de Solucions Obertes, SA**

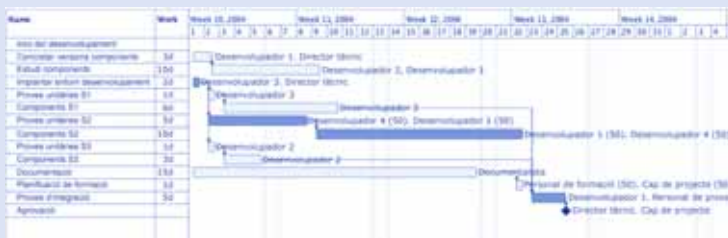
Un mètode molt usat per a expressar aquest tipus de planificacions és el diagrama de Gantt. En una primera versió, ens podem limitar a decidir la durada relativa i la sincronització de les diferents tasques i fites.

**Figura 13.** Planificació de projecte



A continuació podem afegir l'ús de recursos en cada tasca.

**Figura 14.** Informe de planificació completa de projecte



Aquests diagrames, acompanyats de sistemes de control de temps emprats en tasques (i registrats en fulls de seguiment de les tasques del projecte), permetran a Solucions Obertes, SA seguir l'evolució del desenvolupament i ajustar els seus recursos, fites i dates estimades segons la seva marxa.

La plantilla que s'usarà per a reflectir l'estat del projecte periòdicament tindrà aquest format:

**Taula 2.**

Projecte	Infraestructura programari de Solucions Obertes, SA
Data d'aquest informe	DD/MM/AAAA
Data d'acabament	DD/MM/AAAA Estimació original DD/MM/AAAA Estimació actual Canvi des de l'últim informe +/- DD dies
Ítems pendents de desenvolupament	18 defectes 9 funcionalitats
Ítems pendents d'aprovació	0 defectes 2 funcionalitats
Ítems tancats	10 funcionalitats
Recursos usats durant aquest període	Persona: 8 hores Persona: 23 hores Persona: 5 hores
Resum de l'estat	El projecte avança segons la planificació.
Documents relacionats	Plan de projecte. Disseny de mòduls.
<b>Estat detallat</b>	
Aquesta setmana ens hem concentrat en... Hem fet aproximadament el 30% del projecte i ens hem avançat dos dies respecte a la data prevista...	
<b>Gestió de riscos</b>	
Poden sorgir problemes amb la versió d'un dels components que s'han d'integrar. Potser hauríem d'avaluar les versions següents o anteriors.	
<b>Activitats planejades</b>	
Arreglar ítem #452 Arreglar ítem #19 Pròxima reunió de desenvolupament el DD/MM/AAAA Avançar en el desenvolupament del component Y	
<b>Actualització de la planificació</b>	
Aquí podem copiar el diagrama de Gantt vist en la planificació del projecte, degudament actualitzat segons l'ús de recursos, desviacions, etc.	

## Cas pràctic

### Concreció de versions o alternatives dels components de la infraestructura del programari de Solucions Obertes, SA

Després d'haver identificat els components necessaris, concreteu-los i incorporeu la seva documentació a la del projecte.

Taula 3.

Component	Paquet	Versió	Llicència
Sistema de còpies de seguretat	Amanda	2.4.4p2	Lliure distribució amb <i>copyright</i> de la Universitat de Maryland. BSD.
Sistema de gestió de projectes	Gforge	3.2	GPL
Sistema de compartició de fitxers	Samba	3.0.2	GPL
Base de dades	PostgreSQL	7.4.1	GPL
Sistema operatiu	GNU/Linux	2.4.24	GPL
Servidor web	Apache	2.0.48	Apache Software License 1.1. BSD.

## 4.2. Triar la llicència més adequada

Tant si es tracta d'un projecte intern, com d'un producte que tenim previst comercialitzar, **triar la seva llicència** o la de les parts que desenvoluparem abans d'iniciar-lo és altament aconsellable. Més encara quan integrem components programari o serveis, ja que la seva llicència pot condicionar els termes de la nostra, o obligar-nos a incloure-la.

La llicència escollida tindrà algun efecte sobre el següent:

- Els fitxers de codi font del nostre desenvolupament. Haurem d'incloure-la en tots i cada un d'ells, i en determinats casos fer menció

de parts de codi o llibreries que són propietat d'altres organitzacions.

- La documentació i els materials de formació. S'hi ha de reflectir la llicència escollida des de la primera versió del sistema. Això és especialment important en projectes de codi lliure que es posin a disposició públicament, per a evitar malentesos.
- Els components triats. Si el desenvolupament es comercialitza sota una llicència propietària, ens hem d'assegurar que els components que integrem ens ho permeten.
- El client que rep el producte. Hem d'informar el client respecte a quins drets té sobre el producte i quines garanties li proporcionen els seus drets.
- El seu manteniment, i el suport que serem capaços de proporcionar al producte. En la fase d'implantació veurem les estratègies de manteniment possibles segons el tipus de llicència escollit.

Bàsicament ens hem de concentrar en les incompatibilitats entre els diferents models de llicències existents, dins d'algun d'aquests escenaris:

- Combinarem el codi propietari amb el nostre i el distribuïrem sota llicència comercial? Hem adquirit aquests drets per part dels distribuïdors del codi propietari?
- Combinarem el codi lliure (en alguna de les seves variants, BSD, GPL, LPGL, etc.) amb el nostre i el distribuïrem sota llicència comercial? Ens ho permeten totes les llicències dels diferents components?
- Combinarem el codi lliure amb el nostre i el distribuïrem sota llicència lliure? Quina haurà de ser la llicència lliure resultant? Volem imposar alguna restricció a la nostra llicència lliure? Ens interessa mantenir el *copyright*? Donar garantia i suport?

En tot cas, la resposta a les preguntes que sorgeixen a cada escenari s'hauran de resoldre després d'una lectura detinguda de

les llicències dels components i d'una anàlisi del nostre model de negoci.

### Cas pràctic

#### Elecció de la llicència del desenvolupament de la infraestructura programari de Solucions Obertes, SA

El cas pràctic plantejat és un desenvolupament intern, per la qual cosa la llicència triada no tindrà efectes sobre el model de negoci ni sobre la seva distribució a clients. Això no evita que hàgim d'incloure una llicència en el codi que desenvolupem, i en aquest cas, tenim les alternatives següents:

- Llicència propietària: en el nostre sistema combinem llicències estil BSD i GPL. Si no redistribuïm el nostre sistema, el podem desenvolupar sota llicència propietària.
- Llicència estil BSD: aquesta llicència ens permet mantenir el *copyright* sobre el nostre desenvolupament, i és coherent amb les llicències de la resta de components. No ens obliga a distribuir el codi font resultant, però sí que permet al destinatari l'ús del programari, la còpia, la modificació, la redistribució o la venda. També ens permetrà la seva incorporació futura a un producte comercialitzable sota una llicència propietària.
- Llicència GPL: aquesta llicència ens permet mantenir el *copyright* sobre el nostre desenvolupament, i és coherent amb les llicències de la resta de components. Ens obliga a distribuir el codi font resultant i impedeix la seva futura comercialització sota una llicència propietària.

Una vegada analitzades les alternatives, decidim adoptar la llicència BSD, ja que deixa obert el sistema a noves incorporacions de components i a la seva futura comercialització sota els termes que considerem oportuns en aquell moment.

### 4.3. Entorn de desenvolupament

L'objectiu d'aquesta fase és **implantar l'entorn de desenvolupament seleccionat** en l'equip que el durà a terme i procedir-hi. Entrant en detall sobre les activitats vistes en la planificació del desenvolupament, identifiquem les tasques següents:

- Preparació de l'entorn de generació i desenvolupament.
- Generació del codi dels components o procediments.
- Execució de les proves unitàries.
- Execució de les proves d'integració.

Si s'han seguit les recomanacions quant a metodologia d'anàlisi i disseny en les fases anteriors, i es disposa d'eines d'assistència al desenvolupament o de generació de codi a partir dels diagrames de components o classes, l'equip de desenvolupadors disposa de tota la informació i eines necessàries per a acabar amb èxit la seva participació en el projecte.

Si hem implicat els desenvolupadors en activitats de disseny (com suggereixen algunes metodologies recents) el temps d'estudi d'aquest i dels requisits del sistema es reduirà, amb la qual cosa s'imprimirà més ritme al projecte en l'inici del desenvolupament. Per contra, si s'han deixat fases del disseny pendents d'aprovació perquè s'ha començat el desenvolupament d'altres subsistemes completament analitzats, hem d'incorporar aquestes activitats que interrompan el desenvolupament dins de la planificació.

Els resultats de les proves unitàries dissenyades i desenvolupades anteriorment són un perfecte testimoni de l'avenç del projecte, i ens permetran conèixer en tot moment si el ritme del desenvolupament és el que volem o ens desviem de la nostra planificació.

**Cas pràctic****Entorn de desenvolupament del programari de bolcat de la base de dades per al subsistema de còpies de seguretat de Solucions Obertes, SA**

Segons les decisions preses en les fases anteriors, estem en condicions de dur a terme el desenvolupament de l'eina de bolcat de la base de dades. A partir de la planificació del desenvolupament, hem de fer el següent:

- Instal·lar l'IDE Jedit als ordinadors dels desenvolupadors.
- Acordar un estil de codificació. Triem basar-nos en el definit per l'autor del mateix llenguatge:  
<http://www.perldoc.com/perl5.6/pod/perlstyle.html>.
- Acordar un conjunt de preferències del funcionament de l'editor, mida de tabulació, estil de codi, etc., d'acord amb l'estil de codificació acordat.
- Generar codi a partir dels diagrames de classes, els casos d'ús revisats i les proves unitàries.
- Executar proves unitàries de manera concurrent amb el desenvolupament.

**4.4. Documentació**

L'objectiu d'aquesta fase és **elaborar la documentació d'usuari**. Sobre la base de les decisions preses sobre això en fases anteriors, relatives al seu format i disponibilitat, se n'ha de desenvolupar l'estructura i el contingut.

Si es tracta d'una documentació tècnica, haurem de poder incorporar documentació d'altres fonts o components que integrem en el sistema. De la mateixa manera, haurem de poder incorporar la documentació de les operacions i procediments que es trobin en el codi font.

En el cas d'incorporar referències externes, és de vital importància esmentar explícitament la versió que s'ha integrat en el sistema, ja que els enllaços externs poden fer referència a l'última versió (que pot coincidir amb la integrada en el moment de crear la documentació), i més endavant es podrien produir discrepàncies.

L'estil de redacció de la documentació ha de concordar amb el seu destinatari final, i s'han d'indicar clarament els canvis produïts des de la versió anterior.

#### Cas pràctic

##### Documentació del programari de bolcat de la base de dades per al subsistema de còpies de seguretat de Solucions Obertes, SA

Les decisions preses en fases anteriors indiquen que la documentació ha d'estar en format DocBook.

La documentació de les llibreries usades per a connexió a la base de dades, tests unitaris, etc., es troba en format POD, i també la documentació tècnica continguda en el codi font.

Mitjançant la utilitat Pod-DocBook (<http://search.cpan.org/~nandu/Pod-DocBook-1.0/>) podrem bolcar la documentació de llibreries i del nostre propi codi en la documentació.

Altres components o llibreries que no disposin d'utilitats similars, es podran incorporar a la documentació com a enllaços hipertext.



## 5. Implantació

El **pas a producció del sistema** en l'entorn en el qual operarà requerirà una planificació acurada de les seves activitats. En els casos en què el sistema en substitueixi un altre de funcionalitat similar, ens trobarem amb un tipus d'escenaris determinat, i en els casos en què el sistema que es vulgui implantar afegeixi prestacions o funcionalitat a un sistema ja implantat, la incidència de determinades activitats serà diferent.

En tot cas, és en aquesta fase en què haurem d'implicar els usuaris participants dels casos d'ús analitzats del sistema i formar-los en les seves noves responsabilitats o comeses.

També haurem d'acordar el nivell de servei que ha de prestar el sistema, segons les especificacions, i comprovar que aquest es compleix. En cas que el sistema s'hagi d'integrar a un que ja existeixi, comprovar que la seva implantació no afecti el seu funcionament serà una de les activitats clau.

### Cas pràctic

#### Planificació de la implantació de la nova infraestructura programari de Solucions Obertes, SA

El calendari de la implantació s'haurà de fer atenent les diferents fases implicades. Per a la seva representació i seguiment, es poden usar el mateix tipus d'eines utilitzades en la planificació del desenvolupament.

En implantar tot el sistema, els equips de treball i usuaris de la resta de subsistemes s'hauran d'implicar en el seu testeig, i per tant és molt important disposar d'eines col·laboratives de comunicació i registre d'incidències. El mateix gestor de projectes ens pot servir per a registrar els problemes o incidències que es produeixin i la seva solució.

El diagrama de Gantt corresponent a la planificació de la implantació del sistema podria ser el següent:

**Figura 15.** Diagrama de Gantt per a la planificació de la implantació



## 5.1. Formació

La formació en si mateixa no sol formar part de les metodologies relacionades amb projectes de sistemes d'informació. El que és important en aquesta fase és determinar els diferents perfils d'usuari que es necessitaran formar sobre el sistema que implantem i adaptar-la a aquests perfils. El format i tipus de formació (presencial, no presencial, etc.) també pot estar determinat per aquests perfils d'usuari, com pot ser el cas de treballadors que exerceixin la seva activitat fora de les instal·lacions de l'empresa o altres organitzacions que hagin d'usar el sistema implantat.

En qualsevol cas, els estudis dels casos d'ús fets en les fases d'anàlisi i disseny del sistema seran la base per a definir el pla i continguts de la formació.

## 5.2. Implantació del sistema i proves

En aquesta etapa es duran a terme les **actuacions per a implantar el sistema en el seu entorn operatiu definitiu**. Mentre que les proves unitàries i d'integració tenen lloc en un entorn diferent de l'entorn d'operació real, les **proves d'implantació** s'han de fer en el sistema en producció.

Per tant, s'haurà de comprovar com a pas previ que els recursos necessaris atenent els requisits especificats estan disponibles, i també serveis auxiliars, bases de dades, etc.

Adicionalment, és possible que s'hagin de carregar dades inicials en el sistema. Aquesta tasca s'haurà de preveure en aquesta fase, i s'haurà d'incorporar a les proves d'implantació amb els casos reals.

### Cas pràctic

#### Implantació i proves de la nova infraestructura programari de Solucions Obertes, SA

Amb tots els procediments documentats i el sistema provat pel que fa a integració, la implantació es limita en aquest cas a fer la instal·lació dels serveis als servidors definitius i a implantar les polítiques d'accés a aquests serveis, juntament amb les dades que es van especificar en els casos d'ús.

Així doncs, en aquesta activitat hem de fer el següent:

- Instal·lar els serveis en els servidors designats.
- Instal·lar els components desenvolupats en els servidors designats.
- Instal·lar el gestor de base de dades, definint els seus usuaris i les polítiques d'accés.
- Crear les bases de dades, l'estructura de les seves taules i carregar les seves dades inicials.
- Configurar els serveis externs que hagin d'usar els serveis, com per exemple el servidor de correu sortint.

Una vegada comprovada la correcta instal·lació del sistema, activarem les tasques periòdiques que s'executin desatases (com les còpies de seguretat) monitorant-les durant un cicle complet de la seva periodicitat.

**Cas pràctic**

A continuació executarem les proves d'implantació, d'acord amb les especificacions establertes. Entre d'altres, podrien ser:

- Recuperació del sistema forçant un comportament erroni d'alguns dels subsistemes. Hem de registrar les operacions que s'han seguit i l'estat de les dades i els serveis després de la recuperació.
- Rendiment del sistema en l'entorn d'operació. Temps de resposta, comunicacions, etc.

Una vegada revisats els resultats i contrastats amb els requisits, caldrà decidir si s'han de resoldre incidències i quin equip o subsistema n'és el responsable. Posteriorment, i segons la gravetat de les desviacions o incidències registrades, decidirem si tornem a executar el pla de proves completament o parcialment.

**5.3. Nivell de servei**

Segons els resultats obtinguts en les proves d'implantació, i els requisits del sistema, estarem en condicions de **fixar un nivell de servei** de cada subsistema. L'acord de nivell de servei haurà de preveure:

- Identificació dels serveis en els quals hi haurà acord. De quin tipus de serveis parlem:
  - Suport en línia: temps de resposta, disponibilitat.
  - Comunicacions.
  - Seguretat.
  - Gestió de recursos: capacitat, hores de servei per a l'usuari.
- Propietats de cada servei. Unitats en les quals es mesuri la prestació del servei. Poden ser numèriques (hores de temps de resposta, capacitat de disc per als usuaris, amplada de banda, etc.) o bé expressades en termes de restricció de les capacitats del sistema quan parlem de seguretat, per exemple.
- Estimació dels recursos emprats en prestar l'acord de nivell de servei segons els compromisos acordats.

## 5.4. Acceptació del sistema

Aquesta activitat consisteix a **presentar als responsables o a la direcció tota la documentació relativa a la implantació del sistema**, incloent-hi els resultats de les proves i l'acord de nivell de servei, **per a la seva aprovació**.

Només quan el sistema hagi estat presentat i hagi estat aprovat formalment, es considerarà el seu pas a producció, i per tant començarà la prestació del nivell de servei acordat i el seu manteniment.



## 6. Manteniment

Encara que hem situat aquesta fase després de l'aprovació del sistema (ja que és quan s'inicien les seves activitats), la seva planificació s'ha de produir al llarg de totes les fases del projecte. Moltes de les activitats ja fetes, com la documentació o l'establiment de requisits de desenvolupament, s'orienten també a facilitar el **manteniment del sistema**.

A diferència dels projectes en els quals usem programari propietari, on estem obligats a contractar l'organització propietària del programari o algun dels seus socis, el manteniment en projectes de programari lliure admet molta més flexibilitat. Ens podem trobar amb tots o alguns dels escenaris següents:

- Possibilitat de contractar un suport tècnic del component de programari lliure a l'empresa desenvolupadora.
- Possibilitat de formar els nostres desenvolupadors en el component, ja que disposem del codi font, podem assegurar que si posseeixen els suficients coneixements tècnics podran resoldre la majoria d'incidències que es produeixin. Si això succeeix, podem proporcionar la solució de la incidència als desenvolupadors del producte perquè la incorporin. Durant aquest procés, normalment no tindrem cap dificultat a aconseguir ajuda per part dels desenvolupadors del component.
- Possibilitat de contractar una tercera empresa, experta en la matèria, que ens proporcioni el suport necessari si l'empresa original no ofereix aquest servei o el temps de resposta a la nostra petició no s'ajusta al nostre calendari.

Depenent de la nostra capacitat tècnica, dels nostres recursos i de la urgència amb què necessitem resoldre la incidència o amb què hàgim d'implementar la nova funcionalitat, optarem per una fórmula o una altra.

## Cas pràctic

**Manteniment del subsistema de còpies de seguretat de Solucions Obertes, SA**

Durant l'última realització de les còpies de seguretat, hem trobat un problema en el bolcat de la base de dades. Com que el programari Amanda escollit per a fer les còpies en cinta no té en compte la mida de la cinta abans de fer la còpia, el procés no va acabar correctament.

Els nostres desenvolupadors ens indiquen que l'única solució seria modificar el codi font d'Amanda perquè no comencés el bolcat si els fitxers que s'han copiar en cinta excedeixen d'una mida determinada. D'aquesta manera es podria avisar l'administrador i la còpia es faria correctament després de resoldre la incidència.

Consultant el lloc web del producte, veiem que hi ha una secció on és possible contactar amb empreses que hi donen suport. Contactem amb diverses d'elles, que ens envien diferents pressupostos, i després de decidir-nos per la que ens ofereix millors condicions quant a temps de resposta, rebem el pedaç per al producte que passem a incorporar en un entorn segur.

A continuació, executem les proves d'integració, i posteriorment implantem i provem el subsistema implicat.



## Resum

Al llarg del material del curs s'han pogut repassar les diferents **fases de les quals consta un projecte** de sistemes d'informació. Aquestes fases, depenent de la metodologia utilitzada (especialment si forma part del grup de les denominades *clàssiques*, o al contrari, forma part de les noves metodologies àgils), es poden fer presents abans o després dins de la vida del projecte, desenvolupar-se de manera seqüencial o en paral·lel, etc.

Les fases bàsiques en què se solen dividir els projectes de sistemes d'informació són les següents:

- **Estudi de viabilitat:** en aquesta fase es considerarà si el projecte es pot dur a terme, tenint en compte les diferents solucions existents i els recursos dels quals es disposa.
- **Anàlisi:** en aquesta fase s'estudiaran les necessitats que es vol satisfer amb el nou projecte, a fi de poder enfocar la solució tecnològica. Així mateix, s'especificaran les interfícies d'usuari que els permetran interactuar amb el sistema.
- **Disseny:** en aquesta fase es farà el disseny tecnològic de la solució, proposant-ne una arquitectura global i estudiant cada un dels casos d'ús existents.
- **Desenvolupament:** en aquesta fase es construirà la solució, tenint en compte temes com l'entorn de desenvolupament utilitzat, les llicències utilitzades, la documentació generada, etc.
- **Implantació:** en aquesta fase es passarà a un entorn de producció la solució desenvolupada, i s'acceptarà definitivament.
- **Manteniment:** durant aquesta fase, que es prolongarà al llarg de la resta de la vida del projecte, se'n farà el manteniment, tant correctivament com a evolutivament.

En cas que es tracti d'un projecte que s'hagi de desenvolupar dins d'un marc de tecnologies de programari lliure, en cada una de les fases anteriors s'haurà de tenir en compte certs aspectes que, si bé normalment ja es tenen en compte, en aquest tipus d'entorn adquireixen una rellevància especial (p. ex., viabilitat de la solució, arquitectura global del sistema, llicències utilitzades).

## Bibliografía

**AgileAlliance** <http://www.agilealliance.org/>

**Dia a drawing program** <http://www.lysator.liu.se/~alla/dia/>.

**Extreme Programming** <http://www.extremeprogramming.org/>

**Free License Quick Reference** [http://zooko.com/license\\_quick\\_ref.html](http://zooko.com/license_quick_ref.html)

**Métrica 3.** “Consejo Superior de Informática y para el impulso de la Administración Electrónica”. Ministerio de Administraciones Públicas, <http://www.csi.map.es/csi/metrica3>.

**Planner** <http://planner.imendio.org/>

**ReadySet** <http://readyset.tigris.org/>

**The Object Management Group - UML** <http://www.omg.org/uml/>



## GNU Free Documentation License

GNU Free Documentation License  
Version 1.2, November 2002

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.

### 0. 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.

## 1. 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.

## 2. 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.

## 3. 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.

#### 4. 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.

## 5. 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".

## 6. 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.

## 7. 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.

## 8. 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.

## 9. 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.

## 10. 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.

### 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.







