

Recollida i documentació de requisits

Benet Campderrich Falgueras


P00/05007/00302

Índex

Introducció	5
Objectius	6
1. Els requisits	7
1.1. Classes de requisits	7
2. Fonts d'informació	8
3. Passos de la recollida i documentació de requisits	9
3.1. El context del programari	9
3.1.1. El model del domini.....	9
3.1.2. El model del negoci.....	10
3.2. Els guions	10
3.3. Identificació dels actors	11
3.4. Identificació dels casos d'ús.....	12
3.5. Identificació de les relacions entre casos d'ús	12
3.6. Identificació de les relacions d'especialització entre actors	13
3.7. La documentació dels casos d'ús	13
3.7.1. La documentació textual.....	13
3.7.2. La documentació formal	14
4. La recollida i documentació de requisits de la interfície d'usuari	16
4.1. Concepte d'interfície d'usuari	16
4.2. Identificació de les restriccions tècniques	17
4.3. Elaboració dels perfils d'usuari	17
4.4. Documentació de les tasques actuals i futures	17
4.5. Comparació entre tasques i casos d'ús	18
4.6. Especificacions d'usabilitat	19
5. Exemple	20
5.1. Informació inicial	20
5.2. Model del domini	20
5.3. Model del negoci	21
5.4. El glossari del model del negoci	23
5.5. Els guions	23
5.6. Casos d'ús	24
5.6.1. Actors.....	24
5.6.2. Diagrama de casos d'ús	24
5.6.3. Documentació textual.....	25


5.7. Requisits de la interfície d'usuari.....	28
5.7.1. Perfils d'usuari.....	28
5.7.2. Documentació de les tasques futures.....	29
5.7.3. Requisits d'usabilitat	30
Resum	31
Activitats	33
Exercicis d'autoavaluació	33
Solucionari.....	34
Glossari.....	34
Bibliografia.....	34

Introducció

En el mòdul “Introducció a l’enginyeria del programari OO” es va dir que en aquesta assignatura se seguiria el cicle de vida iteratiu i incremental de l’anomenat *Rational Unified Process*; es va veure també que el primer component de procés de cada iteració és la recollida de requisits, que és la base per als altres components de procés. En aquest mòdul es veurà com es fa la recollida de requisits en forma documentada, mitjançant notacions i conceptes d’UML que ja hem vist. 

La recollida de requisits busca d’obtenir informació sobre dos aspectes essencials del programari que es desenvolupa:

- 1) Els processos que ha de fer damunt les dades.
- 2) La manera com ha de demanar als usuaris les dades d’entrada i quina funció del programari volen fer servir en cada moment i presentar-los-en els resultats, que és el que coneixem com a *interfície d’usuari* del futur programari.

Encara que es tracta de dos aspectes del programari força diferents, com que les fonts d’informació són les mateixes, és molt convenient de recollir aquestes dues menes de requisits alhora. 

L’ús que es fa de la recollida de requisits a les diferents fases del cicle de vida esmentat és el següent:

- A la **fase d’inici** s’identifiquen gairebé tots els casos d’ús, per a delimitar l’abast del projecte, però només es descriuen detalladament els més crítics.
- Durant la **fase d’elaboració** es descriuen la gran majoria dels casos d’ús, si bé només se n’implementa una petita part.
- La resta de requisits es descriuen i implementen a la **fase de construcció**; a la **fase de transició** només intervenen els eventuais requisits nous o modificats.

Objectius

L'objectiu general d'aquest mòdul és que els estudiants aprenguin a fer la recollida i documentació dels requisits d'un programari segons una tècnica orientada a objectes, i concretament que procedeixin d'acord amb el Rational Unified Process i facin servir el model UML estudiat en els dos mòduls anteriors. Aquest objectiu general es descompon en aquests altres:

!
Vegeu el Rational Unified Process en el mòdul "Introducció a l'enginyeria del programari OO" d'aquesta assignatura.

1. Entendre el paper dels requisits descrits en forma de casos d'ús dins el procés de desenvolupament de programari.
2. Aprendre a recollir la informació sobre els casos d'ús.
3. Aprendre a identificar els actors.
4. Aprendre a documentar els casos d'ús.
5. Aprendre a determinar i documentar les relacions entre casos d'ús.
6. Aprendre a recollir dins la documentació dels casos d'ús els requisits quant a la interfície d'usuari.

1. Els requisits

Els requisits són l'especificació del que ha de fer el programari; són descripcions del comportament, propietats i restriccions del programari que cal desenvolupar.

Sovint es diu que els requisits han d'indicar què ha de fer el programari sense dir com ho ha de fer; però això és una mica massa radical, per diverses raons:

- Els desenvolupadors són tècnics, i potser els seria difícil d'entendre uns requisits extremadament abstractes.
- És clar que hi ha d'haver unes referències mínimes a la tecnologia utilitzada*.
- Finalment, el programari haurà de ser compatible amb l'entorn tècnic i organitzatiu.

* Per exemple, si una sortida ha de ser per impressora o per pantalla.


Els anomenats *requisits no funcionals* són d'aquesta mena.

Els requisits tenen un doble paper:

a) Servir de base per a un acord entre els usuaris (més exactament, l'empresa client) i els desenvolupadors sobre el programari que s'ha de desenvolupar. Això vol dir que la documentació dels requisits s'ha de fer d'una manera intel·ligible als usuaris, que l'hauran de revisar.

b) Els requisits són la informació de partida per a desenvolupar el programari; són l'entrada de l'etapa següent, l'anàlisi.

1.1. Classes de requisits

Hi ha dues classes de requisits, els funcionals i els no funcionals: 

- Els **requisits funcionals** descriuen què ha de fer el programari per als seus usuaris: acceptar, verificar i enregistrar dades, transformar-les, presentar-les, etc. Aquests requisits queden recollits en els casos d'ús.
- Els **requisits no funcionals** no van associats a casos d'ús concrets* i consisteixen en restriccions imposades per l'entorn i la tecnologia, especificacions sobre temps de resposta o volum d'informació tractat per unitat de temps, requisits quant a interfícies, extensibilitat, facilitat de manteniment, etc.

* Els requisits no funcionals es poden descriure en forma de casos d'ús ficticis.

2. Fonts d'informació

Per a recollir informació dels requisits que ha de complir el programari, haurem de recórrer a les fonts d'informació següents:

- a) Les **entrevistes** –i eventualment les **enquestes**– als futurs usuaris; és important que vagin acompanyades de l'observació directa del treball dels futurs usuaris.
- b) La **documentació sobre el sistema actual** existent; si el sistema està informatitzat s'hauran d'estudiar els manuals de l'aplicació i també els procediments manuals que es fan servir.
- c) **Col·legues dels usuaris**: és possible que els usuaris estiguin acostumats a fer la seva feina de la mateixa manera durant molt de temps i, per tant, el fet de conèixer altres punts de vista els ajudarà a sortir d'esquemes prefixats; hi poden parlar els usuaris, els desenvolupadors o uns i altres.
- d) Finalment, és també molt útil fer una revisió de **sistemes semblants** que hi hagi en el mercat. Hi ha funcions que els usuaris consideren tan evident que formen part del domini que creuen innecessari d'esmentar-les, i que, probablement, estan implementades en tot el programari semblant que hi hagi. El mateix passa amb les anomenades *funcions del sistema*, que els usuaris no fan servir en el seu treball normal però són convenients, i sovint imprescindibles, per al funcionament regular del programari: reorganització de les bases de dades, alta i baixa d'usuaris, manteniment dels paràmetres i les taules bàsiques del programari.

3. Passos de la recollida i documentació de requisits

Els passos de la recollida i documentació de requisits són els següents:

- 1) Conèixer el context del futur programari
- 2) Recollida i classificació dels guions
- 3) Identificació dels actors
- 4) Identificació dels casos d'ús a partir dels guions
- 5) Identificació de relacions entre casos d'ús (extensió, inclusió, especialització)
- 6) Identificació de relacions d'especialització entre actors
- 7) Documentació dels casos d'ús

3.1. El context del programari

Els desenvolupadors de programari que s'encarregaran de recollir els requisits generalment tindran formació i experiència informàtiques però no coneixeran l'activitat professional dels usuaris. Si és així, serà convenient que n'adquireixin un cert coneixement des del punt de vista organitzatiu i, com a conseqüència, de la terminologia que es fa servir; això és el que anomenem **context del programari**.

Hi ha dues maneres de descriure el context d'un programari:

- el **model del domini**, que és la manera simplificada, i
- el **model del negoci**, que és una modalitat més detallada.

En qualsevol cas, s'ha d'elaborar un **glossari** dels termes més utilitzats. Fins i tot si no es fa cap dels dos models, convé fer el glossari.

3.1.1. El model del domini


El model del domini recull els tipus d'objectes –les classes– més importants.

Com a objectes importants, podem establir la classificació següent:

- **objectes del negoci**: factures, expedients, comptes, etc.
- **objectes del món real** i conceptes relacionats amb aquests: client, historial de vendes, etc.

- **esdeveniments**, com ara arribades de trens, expiració del termini per a pagar una factura, etc.

Per a fer el model del domini es fa servir el diagrama de classes d'UML.

En modelitzar el context, cal tenir present que es tracta de fer un model de l'entorn del programari, no pas del programari; això darrer es fa dins un altre component de procés, l'anàlisi i disseny. 

3.1.2. El model del negoci

El model del negoci descriu a grans trets els processos i entitats principals a l'entorn del programari.

Aquest model descriu cadascuna de les grans activitats del negoci en termes de casos d'ús i d'entitats i unitats de treball (que són agrupacions d'entitats que tenen un significat per a l'usuari) que intervenen. Es fan servir el diagrama de casos d'ús i el d'objectes, i per a explicar els casos d'ús es poden fer servir diagrames d'interacció i d'activitats.

Diferències del model del domini i el model del negoci

El model del domini i el model del negoci són força diferents; no es pot dir que el model del domini sigui aquella part del model de negoci que en considera les entitats. Les classes del model del domini s'han obtingut a partir d'un estudi superficial del negoci, mentre que en el model del negoci primer es descriuen en línies generals els casos d'ús, després s'identifiquen les entitats que hi participen i finalment aquests casos d'ús es descriuen amb més detall, però sempre tenint en compte els aspectes organitzatius més que els informàtics.

El mot *negoci*

Realment el mot *negoci* no s'ha de prendre en sentit literal; l'entorn del programari pot ser no solament una organització no lucrativa (o una part d'aquesta), sinó també un robot, per exemple, si el programari n'ha de gestionar el funcionament.

3.2. Els guions


En realitat, difícilment els usuaris identificaran els casos d'ús d'una manera explícita i sistemàtica; generalment ens explicaran algunes de les sèries d'operacions més freqüents que fan en llur treball; això és el que s'anomena *guions**.

* En anglès, *scenarios*.

Els guions tenen les característiques següents:


- Són sessions que un actor duu a terme en relació amb el programari.
- Per a un sistema de programari hi ha molts guions possibles; n'hi ha prou de descriure'n un subconjunt en el qual apareguin almenys una vegada totes les funcions que hagi de tenir el programari.
- Serveixen per a extreure'n els casos d'ús.

Els guions són la font d'informació principal, ja que expressen les necessitats dels usuaris tal com les veuen ells; convé que cada guió descrigui de manera pre-

cisa i completa la interacció corresponent entre l'usuari i el programari: convé que s'indiquin totes les excepcions, casos particulars i precondicions. 

3.3. Identificació dels actors

Com hem vist, un actor és un paper, o més, de qualsevol entitat externa que es preveu que interactuarà amb el programari i li donarà informació o en rebrà; les entitats externes en qüestió poden ser persones, màquines, altres sistemes de programari o instants en el temps en els quals s'ha d'engegar automàticament algun procés.

 Vegeu els actors en el subapartat 2.2 del mòdul "UML (I): el model dinàmic i d'implementació" d'aquesta assignatura.

Cada actor té un paper per a cada cas d'ús en el qual intervé; un paper és primari si l'actor engega el cas d'ús corresponent.

Algunes consideracions sobre els actors:

- 1) A una persona, per exemple, poden correspondre diversos actors, si és que pot tenir diversos conjunts de papers independents en relació amb el programari; de fet, no ens interessa si dos actors poden ser la mateixa persona o no.
- 2) Se suposa que els actors no duen a terme, en principi, cap seqüència de casos d'ús determinada, sinó que cada actor invoca diferents cassos d'ús independentment en moments determinats per la seva activitat exterior al programari.
- 3) Cada actor ha de correspondre almenys a un usuari concret; així s'evita de definir actors massa abstractes.
- 4) N'hi ha prou d'identificar els actors, no cal descriure'ls detalladament.
- 5) No té sentit que dos actors intervinguin exactament en els mateixos casos d'ús amb els mateixos papers.

De vegades pot ser útil aquesta classificació dels actors:

- L'**usuari final** o col·lectiu de persones que interactuaran directament amb el sistema i faran servir les funcions d'usuari.
- L'**usuari privilegiat** o **gestor del sistema** encarregat de definir la forma de funcionament del sistema, del qual utilitzarà principalment les funcions del sistema.
- L'**entorn informàtic**, concepte que agrupa tot el que fa referència a la interfície no humana del programari.

3.4. Identificació dels casos d'ús

Els casos d'ús tenen aquestes característiques:

1) Són processos autònoms iniciats per un actor o per un altre cas d'ús. Dos processos iniciats per actors diferents no poden ser part del mateix cas d'ús, com tampoc aquells processos que no siguin mai engegats directament per un actor, excepte els casos d'ús que formin part d'altres mitjançant relacions *include*.

! Vegeu les relacions *include* en el subapartat 6.3 del mòdul "UML (I): el model estàtic" d'aquesta assignatura.

2) Representen funcions ofertes pel programari i n'identifiquen les entrades i les sortides. Un cas d'ús ha de donar sempre un resultat definit a l'actor primari.

3) Descriuen el què d'aquestes funcions, no pas el com, tret de les especificacions no funcionals, a les quals de vegades es pot donar la forma de casos d'ús.

4) Poden servir de base per a proves de caixa negra.

5) Els casos d'ús que es descriu per primera vegada en una iteració determinada han d'encaixar amb els de les iteracions anteriors; si és indispensable es poden modificar aquests darrers.

És important de descriure tots els casos d'ús relatius al programari considerat; però dins un cicle de vida iteratiu i incremental, s'anirà fent per parts. !

Els casos d'ús s'obtenen dels guions, i s'identifiquen les parts autònomes i eventualment comunes a diversos guions en les quals participa un mateix actor; també es pot esdevenir que un cas d'ús agrupi diversos guions sencers. A cada cas d'ús se li dona un nom, que generalment consta d'un verb i un complement directe.

3.5. Identificació de les relacions entre casos d'ús

En el subapartat anterior hauran quedat clares les relacions d'extensió, inclusió i generalització entre casos d'ús. Amb vista a decidir si una relació entre casos d'ús és d'un tipus o d'un altre es poden tenir en compte aquestes regles:

! Vegeu les relacions entre casos d'ús en el subapartat 2.3 del mòdul "UML (II): el model dinàmic i d'implementació" d'aquesta assignatura.

1) Una **relació d'extensió** sempre va lligada a una condició; així, quan hi ha diversos fluxos de procés possibles o bé casos especials o errors que hagin de ser tractats de manera diferent, tindrem relacions d'extensió. Però, tot i això, el cas d'ús que s'executa condicionalment té autonomia en el sentit que dona un cert resultat concret a l'actor que l'ha iniciat, que és el mateix que el del cas que s'estén.

2) Una **relació d'inclusió** és únicament un recurs per a evitar haver de descriure una mateixa part de procés dins diversos casos d'ús.

Exemple de relació d'inclusió entre casos d'ús

Una factura que té un error s'ha de poder rebutjar i retornar tant si la factura és per una compra com per un servei; és a dir, el cas d'ús de rebuig d'una factura s'inclou tant dins el cas d'ús de tractament d'una factura de compra com dins del tractament d'una factura de servei.

Per tant, no cal que el cas d'ús inclòs tingui autonomia ni que sigui engegat directament per un actor (que en tot cas seria el mateix que engega els casos d'ús en els quals s'inclou). En cas contrari, probablement el millor serà considerar que la relació és d'extensió, ja que res no s'oposa al fet que un cas d'ús n'estengui diversos amb el mateix actor.

Exemple de casos d'ús amb el mateix actor

Pot ser que s'hagi de donar d'alta un client que no sigui ni en el cas d'ús de l'enregistrament d'una comanda ni en el de planificació d'una visita d'un venedor; l'actor hauria de ser en tots dos casos aquell que pot donar d'alta clients.

3) Una **relació d'especialització** indica que dels dos casos d'ús relacionats un és una versió especialitzada de l'altra, en el sentit, per exemple, que el primer s'aplica a una subclasse de la classe a què s'aplica el segon.

Exemple de relació d'especialització

El cas d'ús de matrícula d'un alumne de nou ingrés es pot considerar una especialització del cas d'ús de matrícula d'un alumne en general, si el procés addicional que s'ha de fer en el primer cas és un conjunt d'operacions independents i disperses; però si aquest procés addicional tingués entitat pròpia, la relació podria ser d'extensió, no pas entre els mateixos casos, sinó entre el cas general i la part que només es fa per als alumnes nous.

3.6. Identificació de les relacions d'especialització entre actors

L'actor *A* és una especialització del *B* si *A* té tots els papers de *B* (és a dir, fa els mateixos papers en relació amb els mateixos casos d'ús) i algun més. No sembla que la identificació d'aquesta mena de relacions tingui cap problema.

3.7. La documentació dels casos d'ús

Es poden distingir dos tipus de documentació dels casos d'ús:


- documentació textual i
- documentació formal.

3.7.1. La documentació textual

La documentació textual es compon de la descripció textual dels casos d'ús i el glossari dels termes de la descripció textual.

La descripció textual

Cal que hi hagi una descripció textual de cada cas d'ús. Convé fer-la mitjançant una plantilla i després aplicar sistemàticament unes normes de format.

A continuació es descriu una manera de fer la documentació textual d'un cas d'ús. 

Si establim les convencions següents:

- Noms d'actor en **negreta**.
- Altra terminologia de l'usuari en *cursiva*.
- Referències a d'altres casos d'ús subratllades.


Donarem a la documentació textual l'estructura següent:

- **Capçalera:** número i nom del cas d'ús, resum de la funcionalitat, paper del cas d'ús dins el treball de l'usuari, casos d'ús relacionats, actors.
- **Cos:** preconditioni i postcondició del sistema, etapes, contingut de les entrades i sortides, alternatives de procés i excepcions.
- **Final:** qüestions que cal aclarir (les respostes s'inclouran, amb autor i data, dins els comentaris), hipòtesis que es fan sobre aspectes no aclarits, comentaris, canvis.

El glossari

És molt convenient per a unificar la terminologia i la seva interpretació. Es pot extreure del model del domini o del negoci, però també es pot obtenir a partir d'entrevistes amb usuaris.

3.7.2. La documentació formal

És imprescindible un diagrama de casos d'ús que els mostri tots amb les relacions entre aquests i entre els actors. A més, pot ser que per a casos d'ús concrets convingui de fer servir algun altre diagrama com a complement de la descripció textual: diagrama d'activitats, d'estats o d'interacció. 

Potser podria crear alguna confusió el fet que a l'anàlisi es fan servir també aquests diagrames per a descriure de manera detallada i formalitzada els casos d'ús; però hi ha diferències importants:

a) En l'àmbit de requisits la descripció dels casos d'ús és principalment textual, i els diagrames hi fan només una funció complementària i no s'elabo-

ren sistemàticament per a tots els casos (no oblidem que els usuaris han d'entendre els requisits, cosa que obliga a no usar notacions molt formals); en canvi a nivell d'anàlisi els diagrames d'interacció o d'activitats es fan sistemàticament per a tots els casos d'ús com un pas de formalització en termes d'objectes previ a llur implementació.

b) Els diagrames de l'anàlisi fan servir les classes definides al diagrama estàtic que s'hi elabora, mentre que els objectes que surten als diagrames dels requisits hauran estat identificats en el model del domini o en el model del negoci, o simplement en el moment de fer el diagrama en què surten, i, per tant, són provisionals.

4. La recollida i documentació de requisits de la interfície d'usuari

En la introducció d'aquest mòdul vam veure que cal recollir tant els requisits de funcionalitat com els d'interfície d'usuari. Amb Weinschenk, Jamar i Yeo podem fer aquesta especificació dels objectius de la recollida de requisits de la interfície d'usuari.

Segons els autors esmentats, la recollida i documentació de requisits de la interfície d'usuari consisteix a documentar i verificar informació sobre els usuaris, el seu treball actual i la seva visió del seu treball amb el programari futur, amb vista al disseny de la interfície d'usuari del nou programari.

Lectura complementària

Podreu trobar l'especificació dels objectius de la recollida de requisits de la interfície d'usuari a l'obra següent:

D. Weinschenk; P. Jamar; S.C. Yeo (1997). *GUI Design Essentials*. John Wiley & Sons.

Concretament es tracta d'obtenir la documentació següent:

- Restriccions organitzatives i tècniques;
- Perfils dels usuaris;
- Descripció de les tasques actuals i futures;
- Especificacions d'usabilitat;
- Guions per als casos d'ús.

4.1. Concepte d'interfície d'usuari

La interfície d'usuari és allò que els usuaris veuen del funcionament del programari. També se l'anomena *interfície home-màquina*.

De la interfície d'usuari en depenen molt els factors següents:

- La **comoditat de l'usuari**: afecta l'ansietat, frustració, confusió, fatiga.
- La **productivitat de l'usuari**: és més bona en la mesura que calgui seleccionar menys tecles i botons, i que els recorreguts que hagi de fer amb el ratolí siguin més petits i escassos.
- La **imatge del programari**: els usuaris jutgen la qualitat del programari sobretot a causa del que veuen més directament, és a dir, a causa de les interfícies. No s'adonen de la qualitat de la programació, almenys mentre no afecti de manera perceptible el funcionament del programari; per tant, la interfície d'usuari pot determinar l'èxit o el fracàs d'un programari.

Exemples d'interfícies d'usuari

Els casos més habituals d'interfícies són entrades mitjançant teclat i pantalla i sortides per pantalla i impressora. També es poden considerar entrades i sortides sonores i entrades per escàner.

A més, una interfície d'usuari inadequada pot provocar errors de l'usuari, sobretot en cas de circumstàncies no habituals, en què l'usuari té com a única guia la interfície mateixa; i també pot provocar la infrautilització del programari (en termes d'usuaris potencials i de funcions disponibles), i fins i tot el seu rebuig total.

4.2. Identificació de les restriccions tècniques

Cal determinar si s'ha establert l'ús d'alguna tecnologia, eina, plataforma o normativa concretes, i també la disponibilitat de ratolí, pantalles de colors i tipus de lletra.

4.3. Elaboració dels perfils d'usuari

No es pot dissenyar bé una interfície d'usuari sense saber per a qui es dissenya, ja que un disseny apropiat per a un usuari (en sentit col·lectiu) pot no ser-ho per a un altre. Cal evitar especialment l'error que els desenvolupadors del programari dissenyin la interfície d'usuari com si els usuaris haguessin de ser ells mateixos, ja que tenen una cultura professional i uns coneixements d'informàtica molt diferents dels de la gran majoria d'usuaris.

Els perfils d'usuari tenen aquests aspectes:

- Experiència en l'entorn de maquinari i programari: ratolí, teclat, finestres, etc.
- Experiència en aplicacions del mateix domini.
- Experiència en la feina.
- Freqüència d'ús del programari i grau de rotació del personal.

4.4. Documentació de les tasques actuals i futures

Cal recollir de cada tasca els aspectes següents:

- La tasca en si, la seva freqüència i quins usuaris la fan.
- L'entorn en què es duu a terme, especialment si hi ha limitacions d'espai, la il·luminació, si hi ha brutícia que pugui afectar alguns dispositius com els ratolins.
- La seva situació dins el flux de tasques, és a dir, quines la precedeixen, la segueixen o la interrompen i les interdependències amb altres tasques que obliguin a seguir un ordre determinat.
- Quina informació entra i surt i quins són els resultats i cap a on van.


- Quins documents i eines calen.
- Problemes i errors més freqüents.
- Queixes i suggeriments sobre com es fa la tasca.

Aquesta informació no es documenta de manera formalitzada; una tasca o seqüència de tasques es pot representar mitjançant un diagrama d'activitat simplificat en el qual només es representin els fluxos i els estats d'activitat, que correspondrien a operacions, ja sigui manuals o informatitzades.


El conjunt de les tasques es pot representar en forma d'una taula que tingui una fila per a cada tasca i en la qual les columnes corresponen als aspectes indicats. Òbviament es documenten per separat les tasques actuals i les futures.

4.5. Comparació entre tasques i casos d'ús

Tot i que les fonts d'informació per als requeriments de funcionalitat i d'interfície d'usuari són les mateixes, la informació necessària no és la mateixa; així, en descriure els casos d'ús des del punt de vista de la funcionalitat, es posa èmfasi en allò que fa el programari, mentre que quan es descriuen des del punt de vista de la interfície d'usuari, el que interessa més són les accions que fa l'usuari.

Així, podem veure les comparacions entre tasques i casos d'ús següents: 

- a) Allà on des del punt de vista d'un cas d'ús el programari presenta determinada informació, des del punt de vista de les tasques l'usuari comprova aquesta informació i pren una decisió.
- b) En un cas d'ús que té diverses possibilitats, no cal indicar la freqüència de cadascuna; en una tasca sí.
- c) A les tasques purament manuals no els correspon cap cas d'ús, i dins un cas d'ús tampoc no s'indiquen les operacions manuals dins una tasca, com posar un disquet abans de començar un treball; en la descripció d'una tasca s'han de tenir en compte, si més no per a determinar la durada de la tasca.
- d) Quan hi ha una anomalia, el programari es limita a avisar l'usuari, segons el cas d'ús, mentre que segons la descripció de la tasca, l'usuari ha de prendre una decisió i potser començar una altra tasca independent.

Per tant, cal recollir informació sobre les mateixes funcions des de dos punts de vista. 

Una bona manera de recollir requisits per als dos usos seria que dos desenvolupadors fessin plegats les entrevistes als usuaris i que cadascun s'interessés per un dels dos aspectes; sembla clar que és molt millor que els dos entrevistadors facin una entrevista i no pas dues independents per dos motius: per a no prendre més temps de l'usuari i per a evitar de tenir dues versions no coordinades de les necessitats de l'usuari.

4.6. Especificacions d'usabilitat

Les especificacions d'usabilitat són els requisits no funcionals relatius a la interfície d'usuari.


Les especificacions d'usabilitat es poden referir a facilitat d'utilització o aprenentatge o a rapidesa i precisió en l'execució de les tasques; convé expressar-los de manera quantitativa.

Exemples d'especificacions d'usabilitat

Heus ací alguns exemples:

- Que el 90% dels usuaris que han treballat almenys un any amb el sistema antic siguin capaços d'aprendre en una setmana les funcions indicades en una llista i valent-se només de la documentació i l'ajuda en línia.
- Que el 80% dels usuaris després d'una setmana d'aprenentatge pugui enregistrar una mitjana de deu factures en trenta minuts o menys.
- Que després d'haver donat tres altes de clients el 75% dels usuaris pugui donar d'alta deu clients en vint minuts o menys i el 90% en vint-i-cinc minuts o menys.

5. Exemple

Farem servir un exemple de projecte de programari del qual veurem la documentació que es va generant en els passos successius de les etapes de recollida de requeriments, anàlisi i disseny; per raons didàctiques s'ha escollit un exemple molt més senzill que els casos reals. 

L'exemple tracta de la informatització de la gestió de lloguers de locals comercials que fa una agència immobiliària. La informació no estructurada de l'exemple està en lletra més petita, a fi de distingir-la de les explicacions eventuais.

5.1. Informació inicial

Aquest és el resum de la primera entrevista que va tenir l'equip de desenvolupadors del programari amb el representant de l'agència immobiliària:


L'agència immobiliària (que des d'ara s'anomenarà simplement *l'agència*) lloga locals comercials, que poden ser botigues magatzem, oficines o immobles. Els immobles són edificis que es poden llogar sencers o bé per oficines i botigues magatzem individuals. A més, hi ha uns locals, anomenats *polivalents*, que es poden considerar tant oficines com botigues magatzem.

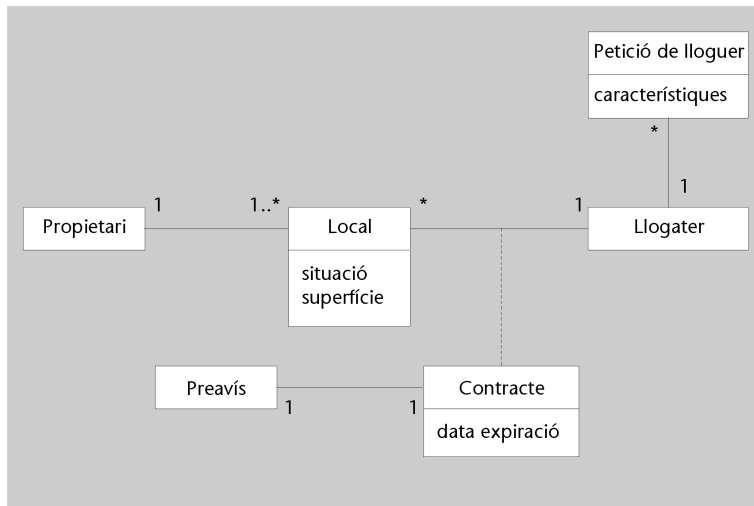
Els propietaris dels locals poden ser particulars o empreses. El propietari pot cancel·lar en qualsevol moment l'oferta del seu local, però mentre no ho faci no el pot llogar pel seu compte i està obligat a acceptar el llogater que proposi l'agència si compleix les condicions estipulades; aleshores l'agència prepara el contracte de lloguer, que sempre és per un temps determinat. Quan es compleix el termini de preavís de la renovació o no del contracte de lloguer, si hi ha renovació l'agència prepara el nou contracte i si no hi ha renovació o bé el llogater cancel·la el contracte abans del termini esmentat, l'agència proposa al propietari de buscar-li un altre llogater.

Un altre servei que fa l'agència és rebre peticions de lloguer de locals amb característiques determinades (situació i superfície, principalment) i llavors es busquen locals que les tinguin, entre els que lloga l'agència i si cal d'altres agències; quan se'n troba un de convenient, es proposa al propietari que esdevingui client i, si accepta, es prepara el contracte.

5.2. Model del domini

A primer cop d'ull s'identifiquen els objectes o classes 'local', 'propietari', 'llogater' i 'contracte de lloguer' (objectes del món exterior), 'petició de lloguer de local' (objecte del domini) i 'termini de preavís de fi del lloguer' (esdeveniment). S'han posat alguns atributs a les classes de manera orientativa (sense tipus, o reunint en un el que poden ser diversos, com és el cas de *situació*).

No cal que els noms dels atributs, i fins i tot de les classes, respectin les restriccions de cap llenguatge perquè aquest diagrama no es farà servir per a etapes posteriors. 

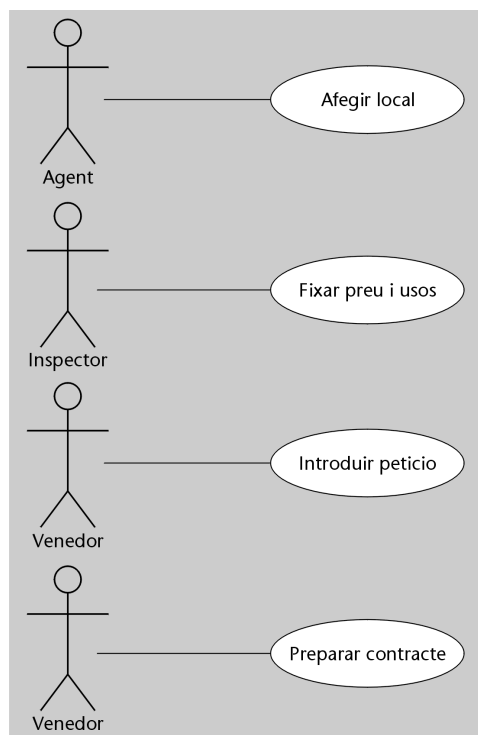


5.3. Model del negoci

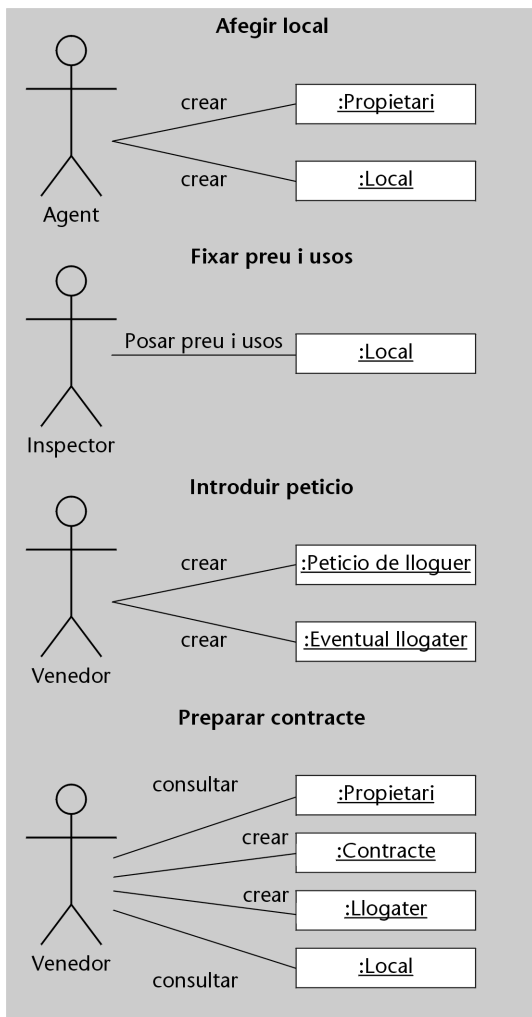
Per a elaborar el model del negoci caldria més informació amb vista a elaborar uns primers casos d'ús; la informació addicional és aquesta:

Quan un propietari ofereix un local és entrevistat per un agent, el qual li demana les dades identificatives i l'adreça, superfície, característiques i restriccions (que el llogater no hi pugui posar un bar de nit, per exemple) del local, i introdueix aquestes dades a l'ordinador; després un inspector visita el local i fa un informe, a partir del qual fixa el preu del lloguer i els usos per als quals és adequat. Quan es produeix una petició de lloguer, un venedor entrevista l'eventual llogater i introdueix a l'ordinador les dades i les característiques del local que demana; quan es fa un contracte de lloguer el venedor introdueix les dades del llogater i del contracte.

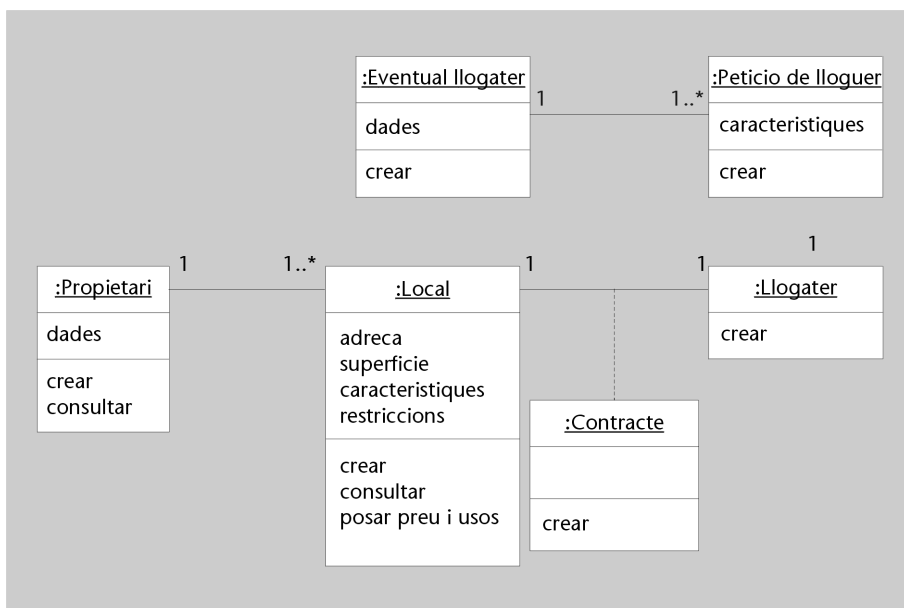
El diagrama de casos d'ús del model del negoci és aquest:



Per mitjà de diagrames de col·laboració identifiquem els objectes que fan servir els casos d'ús anteriors:



El diagrama d'objectes del model del negoci és aquest:



5.4. El glossari del model del negoci

A continuació es presenta el glossari del model de negoci:

Agència: agència immobiliària X, dedicada a la intermediació en el lloguer de locals comercials.

Agent: empleat de l'agència que entrevista els propietaris.

Codi del local: dada que identifica un local, formada pel codi de la zona, el tipus de local i un número correlatiu.

Contracte: contracte legal de lloguer d'un local, signat pel propietari i el llogater.

Eventual llogater: persona o empresa que demana a l'agència que li busqui un local amb unes determinades característiques per a llogar-lo amb vista a un ús concret.

Informe: report que elabora un inspector de l'agència sobre un local després de visitar-lo.

Inspector: empleat de l'agència que visita els locals abans d'oferir-los en lloguer.

Llogater: persona o empresa que lloga un local comercial.

Local: immoble o part d'aquest que es lloga per a usos comercials, és a dir, qualsevol ús tret d'habitatge o indústria. Pot ser immoble, oficina, botiga magatzem o polivalent.

Preu: import mensual del lloguer d'un local.

Propietari: persona o empresa que té el títol de propietat del local. Els clients de l'agència són els propietaris dels locals que lloga.

Venedor: empleat de l'agència que entrevista els llogaters i llogaters eventuais, busca locals adients per a aquests darrers i prepara els contractes.

5.5. Els guions

S'han trobat tres guions:

El guió de l'agent

Arriba un client i se li demanen les dades del local que vol llogar: adreça, tipus, superfície, característiques, restriccions, el volum si es tracta d'una botiga magatzem i les característiques polivalents si és un polivalent, i se li assigna un codi que conté la zona, el tipus i un número correlatiu. Si el propietari no estava introduït per a un local anterior, cal donar-ne la raó social o els cognoms i nom –segons que es tracti d'una empresa o d'un particular–, l'adreça, el telèfon i el NIF. Un cop afegit el local, no es pot llogar encara, sinó que queda pendent d'inspecció.

El guió de l'inspector

La inspecció consisteix en una visita de l'inspector al local. L'inspector examina diversos aspectes del local (forma, accessibilitat, instal·lacions, estat de conservació, visibilitat, etc.) amb vista a determinar-ne el preu de lloguer i els usos possibles. L'inspector mateix introdueix aquesta informació.

El guió del venedor

Un cop se li ha posat preu, el local està pendent de llogar; s'anuncia a la premsa i els venedors l'oferixen als llogaters eventuais. Quan arriba un eventual llogater l'atén un venedor, el qual l'entrevista i introdueix les seves dades (cognoms i nom, NIF, telèfon, adreça, tipus de local que desitja, zona o zones, superfície mínima i requisits diversos); tot seguit, el venedor fa una cerca entre els locals pendents de llogar; si troba locals adients, li esmenta les característiques (llevat de l'adreça del local i les dades del propietari); si al eventual llogater l'interessa algun local, es marca el local com a compromès i s'engega la preparació del contracte, que comença amb l'avis al propietari. Si entre els locals que l'agència ofereix no n'hi ha cap d'apropiat, l'agència busca més locals per altres mitjans, i quan en troba un de convenient ofereix al propietari de presentar-li un llogater a canvi que el propietari esdevingui client de l'agència.

5.6. Casos d'ús

Al llarg dels subapartats següents farem la identificació dels casos d'ús corresponent a la situació plantejada.

5.6.1. Actors

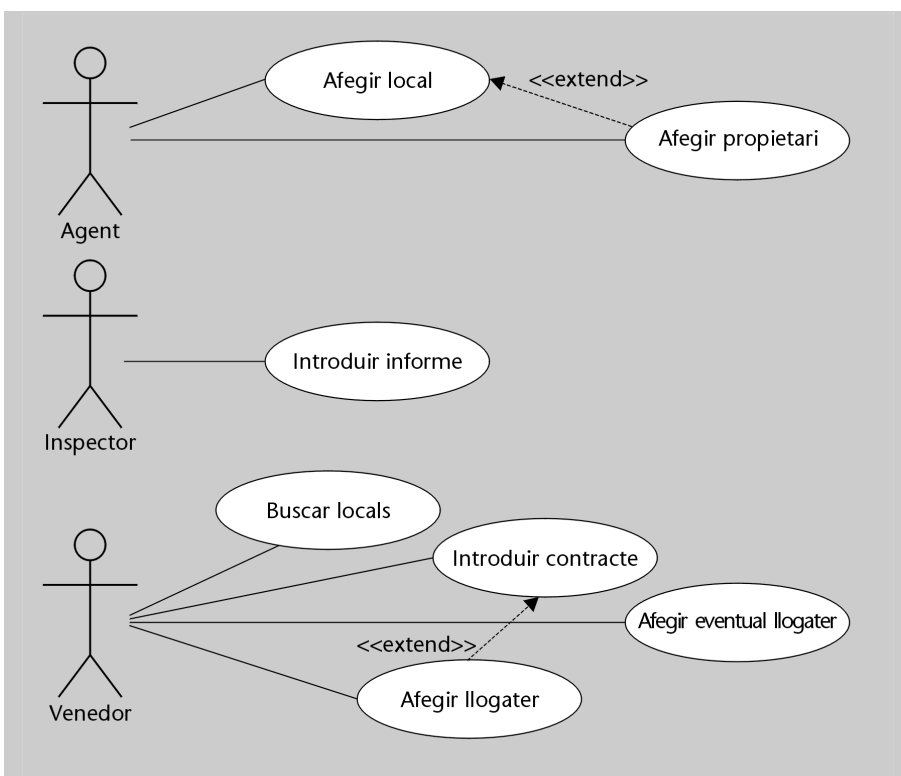
Els actors són tres: **agent**, **inspector** i **venedor**, tots els quals són usuaris finals directes del sistema; en canvi, no són actors ni els propietaris ni els llogaters o llogaters eventuals, ja que no tenen interacció amb el sistema, sinó que tota la relació que hi tenen és mitjançant els agents i els venedors (per tant, el programari no els "veu").

L'**agent** té dos papers, ja que afegeix locals, d'una banda, i propietaris, de l'altra; el **venedor** en té quatre, ja que introdueix llogaters, llogaters eventuals i contractes, i busca locals. Tots aquests papers són primaris.

Els tres actors són independents; entre ells no hi ha cap relació d'especialització, ja que, com veurem tot seguit, no hi ha cap cas d'ús que puguin fer dos dels actors.

5.6.2. Diagrama de casos d'ús

A continuació, veiem el diagrama de casos d'ús relacionat amb l'exemple:



5.6.3. Documentació textual

Es fan servir les convencions i l'estructura descrites abans, però, com passa també a la realitat, no tots els casos d'ús tenen totes les dades possibles (per exemple, descomposició en etapes).

! Vegeu les convencions i l'estructura sobre descripcions textuales en el subapartat 3.7.1 d'aquest mòdul didàctic.

No es dona el glossari dels casos d'ús perquè, per la senzillesa del cas, coincidiria amb el glossari del model del negoci; tanmateix, en els casos reals generalment s'hauran introduït termes nous a les descripcions dels casos d'ús.

! Vegeu el glossari del model de negoci en el subapartat 5.4 d'aquest mòdul didàctic.

Cas d'ús número 1: "Afegir local"

Resum de la funcionalitat: afegeix un *local* a la base de dades.

Paper dins el treball de l'usuari: és el cas d'ús principal del treball dels *agents*.

Actors: **agent**.

Casos d'ús relacionats: Afegir propietari, Introduir informe, Introduir contracte.

Precondició: el *local* no existeix a la base de dades.

Postcondició: el *local* està incorporat a la base de dades.

L'**agent** introdueix les dades del *local*: un *codi del local* –que conté la zona, el tipus (botiga magatzem, oficina, polivalent o immoble sencer), i un número correlatiu–, l'adreça, la superfície, un text de característiques, un altre de restriccions, el NIF del propietari, el volum si és botiga magatzem i el text de característiques polivalents si és un polivalent.

Alternatives de procés i excepcions: si el *propietari* no estava introduït, cal afegir-lo segons el cas d'ús Afegir propietari.

Qüestions que cal aclarir:

- S'ha d'enregistrar quin **agent** ha afegit el *local*? Resposta: sí; el nom de l'**agent** es pren d'una llista.

Cas d'ús número 2: "Afegir propietari"

Resum de la funcionalitat: afegeix un *propietari* a la base de dades.

Paper dins el treball de l'usuari: és un cas d'ús esporàdic dins el treball dels *agents*.

Actors: **agent**.

Casos d'ús relacionats: Afegir local, Introduir contracte.

Precondició: el *propietari* no existeix a la base de dades.

Postcondició: el *propietari* està incorporat a la base de dades.

L'**agent** introdueix les dades del *propietari*: el NIF, la raó social si és una empresa o els cognoms i nom si és un particular, l'adreça i el telèfon.

Aquest cas d'ús pot ser executat independentment o bé ser cridat des del cas d'ús Afegir local.

Cas d'ús número 3: "Introduir informe"

Resum de la funcionalitat: afegeix l'*informe* de l'*inspector* a un *local*.

Paper dins el treball de l'usuari: és el cas d'ús únic dins el treball dels *inspectors*.

Actors: **inspector**.

Casos d'ús relacionats: Afegir local, Introduir contracte.

Precondició: el *local* està introduït a la base de dades però encara no té l'*informe* i, per tant, no està disponible per a llogar.

Postcondició: el *local* té incorporat l'*informe* i està disponible per a llogar.

L'**inspector** introdueix el *codi del local* i després els textos sobre cadascun dels aspectes del *local*: la forma, l'accessibilitat, les instal·lacions, l'estat de conservació, la visibilitat i aspectes diversos; també introdueix els usos recomanats, els agafa d'una llista, i el preu. A més, pot modificar l'adreça, la superfície, el tipus i el text de característiques.

Qüestions que cal aclarir:

- S'ha d'enregistrar quin **inspector** ha fet l'*informe*? Resposta: sí; el nom de l'**inspector** es pren d'una llista.

Cas d'ús número 4: "Introduir contracte"

Resum de la funcionalitat: introdueix a la base de dades la informació necessària per a preparar un *contracte*.

Paper dins el treball de l'usuari: forma part de la part principal del treball dels *venedors*.

Actors: **venedor**.

Casos d'ús relacionats: Afegir llogater.

Precondició: el *local* està disponible per a llogar.

Postcondició: el *local* està llogat.

El **venedor** introdueix el *codi del local*: i després el NIF del **llogater**; després s'afegeixen les dates d'inici i de fi del *contracte*.

Alternatives de procés i excepcions: si el **llogater** no estava introduït, se l'afegeix segons el cas d'ús Afegir llogater.

Qüestions que cal aclarir:

- S'ha d'enregistrar quin **venedor** ha introduït el *contracte*? Resposta: sí; el nom del **venedor** es pren d'una llista.
- Cal imprimir directament el *contracte*? Resposta: no, però s'han d'imprimir les dades del *propietari*, *llogater*, *local* i data d'inici i fi del *contracte* amb vista a preparar el *contracte* fora de l'aplicació.

Cas d'ús número 5: "Afegir llogater"

Resum de la funcionalitat: afegeix les dades d'un *llogater* a la base de dades.

Paper dins el treball de l'usuari: el fan servir ocasionalment els *vedadors* com a pas previ a la introducció d'un *contracte*.

Actors: **venedor**.

Casos d'ús relacionats: Afegir eventual llogater, Afegir contracte.

Precondició: el *llogater* no existeix a la base de dades.

Postcondició: el *llogater* està incorporat a la base de dades i per tant pot figurar en un *contracte*.

El **venedor** introdueix les dades del *llogater*: el NIF, els cognoms i nom (o la raó social, si és una empresa), el telèfon i l'adreça.

Cas d'ús número 6: "Afegir eventual llogater"

Resum de la funcionalitat: afegeix les dades d'un *eventual llogater* a la base de dades.

Paper dins el treball de l'usuari: el fan servir ocasionalment els *vedadors* com a pas previ a la cerca de *locals* que puguin convenir al *eventual llogater*.

Actors: **venedor**.

Casos d'ús relacionats: Afegir llogater, Afegir contracte.

Precondició: el *eventual llogater* no existeix a la base de dades.

Postcondició: el *eventual llogater* està incorporat a la base de dades i per tant pot figurar en un *contracte*.

És com el cas d'ús Afegir llogater, llevat que a més el **venedor** introdueix les dades següents: el tipus de local, la zona o zones desitjades, la superfície mínima i requisits diversos.

Cas d'ús número 7: "Buscar locals"

Resum de la funcionalitat: busca els locals que compleixen determinades condicions.

Paper dins el treball de l'usuari: el fan servir ocasionalment els *venedors* per a la cerca de *locals* que puguin convenir al *eventual llogater*.

Actors: **venedor**.

Casos d'ús relacionats: cap.

Precondició: cap.

Postcondició: cap, en ser una consulta.

Busca tots aquells *locals* que compleixen unes condicions especificades abans de l'execució en termes de tipus, zona i superfície i obté una llista de la qual es poden seleccionar *locals* un a un per a veure'n tota la informació o bé tota menys l'adreça i el NIF del *propietari*.

Altres requisits

La informació sobre propietaris, llogaters, locals i contractes s'ha de poder consultar i modificar i, aquella que ja no es faci servir, esborrar-la.

5.7. Requisits de la interfície d'usuari

Aquí determinarem els requisits que ha de complir la interfície d'usuari.

5.7.1. Perfils d'usuaris

D'entrada, es fa una explicació dels perfils dels usuaris.

Tots els usuaris coneixen a fons la seva feina. Els agents i els venedors estan acostumats a fer servir ordinadors, però els inspectors no gaire. Cap usuari no ha fet servir cap aplicació semblant. La rotació del personal és molt baixa. Totes les funcions del programari es faran molt sovint.

5.7.2. Documentació de les tasques futures

Aquí només s'inclou la documentació de les tasques futures; la de les tasques antigues es faria de la mateixa manera.

Afegir un local

L'agent tecleja l'adreça i superfície del local (en m²) i els textos de característiques i restriccions. Pel que fa al codi del local, la zona i el tipus es trien de les llistes respectives. El sistema l'assigna automàticament el número correlatiu. L'agent tecleja també el NIF de l'usuari i, si no existeix, l'agent pot optar entre canviar-lo i afegir un nou propietari amb aquell NIF.

Quan s'afegeix un propietari el NIF és el que s'havia donat amb les dades del local; l'agent tecleja l'adreça, el telèfon i el nom i els cognoms o bé la raó social.

Introduir un informe

L'inspector porta escrita en un imprès la informació que ha recollit durant la visita. Per a introduir-la a l'ordinador, tecleja primer de tot el codi del local, i després veu tota la informació que es va donar quan es va afegir el local; la comprova i la pot modificar, llevat del codi del local i el text de restriccions. Tecleja els camps textuais de forma, accessibilitat, instal·lacions, visibilitat i preu, i selecciona d'una llista un valor per a l'estat de conservació, i d'una altra llista, el valor o valors dels usos possibles del local.

Introduir un contracte

El venedor introdueix el codi del local i li surt tota la informació sobre ell; la comprova i si hi havia algun error, cancel·la la tasca, i si no n'hi ha cap, introdueix el NIF del llogater. Si el llogater existeix, en surten totes les dades i el venedor pot cancel·lar la tasca o bé continuar, i modificar les dades o no fer-ho; si no existeix, surten les dades en blanc i pot o bé introduir-les o bé cancel·lar la tasca; si introdueix les dades i les confirma, està en el mateix punt que si hagués trobat el llogater inicialment. Si continua, introdueix la data d'inici i la data de fi de la vigència del contracte. Aleshores pot confirmar la tasca o cancel·lar-la; si la confirma s'imprimeixen les dades del contracte, que són aquestes:

- del propietari, els cognoms i nom o bé la raó social, el NIF i l'adreça;
- del local, l'adreça;
- del llogater, els cognoms i nom o bé la raó social, el NIF i l'adreça;
- les dates d'inici i de fi de la vigència.

Introduir un eventual llogater

El venedor n'introdueix el NIF i el pot confirmar o cancel·lar la tasca; si aquell NIF ja existeix a la base de dades, surt tota la informació del eventual llogater, que pot ser modificada, i si no, surten les mateixes dades però buides; aquestes dades són els cognoms i nom o bé la raó social, l'adreça, el telèfon, el tipus de local i les zones desitjades, la superfície mínima en m² i el camp de requisits diversos. El tipus de local i les zones s'escullen de les llistes respectives, mentre que la resta de dades es tecleja.

Buscar locals

Es poden buscar locals si es dóna alguna de les dades següents:

- el codi del local;
- la zona;
- el NIF del propietari;
- el NIF del llogater o eventual llogater;
- la superfície mínima (en m²).

Només es pot donar un d'aquests valors, llevat de la zona i la superfície mínima, que es poden donar tots dos alhora. Quan es dóna el NIF d'un llogater, surten els locals que té llogats, i si és un eventual llogater, surten aquells locals que compleixen els seus requisits quant a superfície i zona. Es pot escollir entre que surtin tots els locals o només els disponibles, i que surtin totes les dades dels locals o bé totes llevat del NIF del propietari i l'adreça. La llista surt per pantalla i opcionalment es pot imprimir.

En la taula següent resumim tots aquests aspectes:

Aspectes de les tasques					
Aspecte	Afegir local	Introduir informe	Introduir contracte	Afegir eventual llogater	Buscar locals
Freqüència	50 per dia	50 per dia	90 per dia	30 per dia	30 per dia
Usuaris	Agents	Inspector	Venedors	Venedors	Venedors
Entorn	Normal	Normal	Normal	Normal	Normal
Tasques anteriors	Cap	Afegir local	Cap	Cap	Afegir eventual llogater
Tasques següents	Intriduir informe	Cap	Cap	Buscar locals	Cap
Entrades	Dades del local i eventualment del propietari	Informe de l'inspector	NIF propietari, NIF llogater, codi local i dates	Dades de l'eventual llogater	Cap
Sortides	Cap	Cap	Dades per al contracte	Cap	Llista de locals
Destí de les sortides	–	–	Servei jurídic	–	Venedor o eventual llogater
Documents i eines utilitzats	Mapa de la ciutat	Cap	Cap	Mapa de la ciutat	Mapa de la ciutat
Problemes i errors més freqüents	Superfície mal mesurada	Cap	Cap	Requisits incomplets	Cap

5.7.3. Requisits d'usabilitat

Els agents i inspectors han de ser capaços de fer servir el sistema després d'una explicació i pràctica d'una hora, i els venedors després d'una explicació i pràctica de dues hores.

Resum

S'ha vist un procediment per a recollir i documentar els requisits del programari que es vol desenvolupar des dels seus dos vessants: els processos i la interfície d'usuari. Abans de procedir a la recollida de requisits del programari els desenvolupadors han d'adquirir i documentar un cert coneixement de l'entorn en què es farà servir el programari, cosa que s'aconsegueix fent o bé el model del domini o bé el model del negoci (més extens, com sabem).

Per bé que hi ha diverses fonts d'on es pot extreure informació útil amb vista a establir els requisits, la font més important són els futurs usuaris, per mitjà d'entrevistes amb ells; en cada entrevista convé d'establir tant els requisits quant a procés com els requisits quant a interfície d'usuari, que serveixen per a especificar els casos d'ús i les tasques respectivament.

S'ha vist un exemple de recollida i documentació de requisits. De la mateixa manera que en un cas real, no s'ha donat tota la informació de cop, sinó que se suposa que s'ha completat durant l'elaboració de la documentació i en resposta a dubtes que es presentaven aleshores; també es pot veure que la documentació de les tasques diu algunes coses que no diu la documentació dels casos d'ús i a l'inrevés, ja que la finalitat amb què es fan és diferent, però no hi hauria d'haver contradiccions entre les dues, ja que serien errors.

Activitats

1. S'ha de fer un programari de gestió de l'activitat d'assegurances d'automòbils en una companyia asseguradora.

Tot comença amb l'emissió d'una pòlissa d'assegurança a la qual consten el NIF, nom i cognoms, any de naixement, telèfon i adreça del propietari del vehicle i dels altres conductors autoritzats, la marca, model i matrícula del cotxe i el número, codi de cobertura i data d'emissió de la pòlissa; les pòlisses tenen vigència durant un any a partir de la data d'emissió.

Un altre aspecte és la gestió dels sinistres; un sinistre està identificat per un número correlatiu, la data en què es va produir i el número de la pòlissa que el cobreix, i, a més, hi ha el NIF o DNI del conductor contrari, el seu nom, adreça i telèfon i la matrícula, companyia i número i data de fi de vigència de la pòlissa del vehicle contrari, i també l'import dels danys del contrari i, si el codi de cobertura ho requereix, dels danys propis. Cada mes es fa un procés que prepara la renovació de les pòlisses que vencen al segon més després, en el qual per a cada pòlissa es calcula l'import de la renovació segons el codi de cobertura i dels sinistres dels dotze mesos anteriors.

Si portem a terme la documentació dels requisits segons el mètode d'aquest mòdul, podem fer suposicions sobre aquella informació addicional que eventualment ens faci falta.

Exercicis d'autoavaluació

1. Quines són les fonts d'informació sobre els requisits d'un programari, a part de les entrevistes i observació del treball dels usuaris?
2. Quins són els criteris per a determinar si una relació entre casos d'ús és d'extensió, inclusió o especialització?
3. Compareu les tasques amb els casos d'ús.

Solucionari

Exercicis d'autoavaluació

1. La documentació sobre el sistema actual, l'opinió dels col·legues dels futurs usuaris i la documentació dels sistemes semblants.
2. Trobareu la resposta en el subapartat 3.5.
3. Trobareu la resposta en el subapartat 4.5.

Vegeu l'apartat 2 d'aquest mòdul didàctic.



Glossari

actor

Conjunt de papers d'una entitat exterior en relació amb el sistema de programari considerat.

cas d'ús

Documentació d'una interacció entre el programari i un actor o més.

context (del programari)

Entorn dins el qual es farà servir el programari que cal desenvolupar; ens interessa especialment l'organització i la terminologia.

interfície d'usuari

Allò que veuen els usuaris del funcionament del programari.

requisits

Descripció del comportament, propietats i restriccions del programari.

Bibliografia

Jacobson, I; Booch, G.; Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Addison-Wesley.

Weinschenk, S.; Jamar, P.; Yeo, S.C. (1997). *GUI Design Essentials*. John Wiley & Sons.

Rosenberg, D.; Scott, K. (1999). *Use Case Driven Object Modeling with UML: A Practical Approach*. Addison-Wesley.

Jacobson, I. (1994). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.