

# **Servei de Generació de Informes Corporatiu**

**Carlos Bardi Zaragoza**

Enginyeria en Informàtica

**Xavier Ferró Garcia**

Data Lliurament: 25 de juny de 2008

## **Resum**

Avui en dia, les empreses disposen de diferents programaris per al tractament de la informació que gestionen en els seus processos quotidians.

Es vol crear un entorn de treball que permeti als diferents usuaris de l'empresa, obtenir informació de les seves dades, tant a nivell de direcció, per propòsits de divulgació, ajuda a la presa de decisions, anàlisis i control, com a nivell de consulta de les dades de l'operativa bàsica diària per diferents necessitats.

Així, l'objectiu general d'aquest projecte, serà el desenvolupament d'una aplicació que permeti la realització d'informes, fent transparent, a l'usuari final, l'origen, ubicació i tecnologies de les diferents bases de dades distribuïdes, ni programaris amb que necessita treballar.

Dins de l'àmbit del projecte, s'utilitzaran tecnologies de la plataforma J2EE.

Es desenvoluparà un aplicació Java d'escriptori per a la definició i administració del nou sistema i una aplicació web JSP per l'accés als informes per tal de poder-hi accedir des de qualssevol ubicació i dispositiu.

Per la realitzar una connectivitat amb els diferents orígens de dades s'utilitzarà JDBC. S'utilitzaran drivers de tipus 4, drivers Java purs. Aquests drivers són els més fàcils de desplegar ja que no hi ha llibreries addicionals ni software intermediari que instal·lar.

Per tal de que el nou sistema no depengui de cap origen de dades, la definició de la lògica del sistema és emmagatzemarà en arxius XML. La representació final de les dades dels informes també es representarà amb aquesta tecnologia per tal de que es puguin visualitzar en format web i formatar mitjançant XSLT.

## **Paraules Clau**

UOC, enginyeria en informàtica, J2EE, PFC, Java, JSP, Servlet, XML, JDOM, XSLT, WAI, W3C, JDBC, ODBC, orígens de dades, drivers, informes, enginyeria del programari, Xavier Ferró Garcia, Carlos Bardi Zaragoza

## **Nom de l'àrea del PFC**

J2EE

## Índex

1.	Introducció.....	5
1.1.	<b>Justificació del PFC.....</b>	<b>5</b>
1.2.	<b>Objectius del PFC.....</b>	<b>7</b>
1.3.	<b>Enfocament i mètode seguit.....</b>	<b>8</b>
1.4.	<b>Planificació del PFC.....</b>	<b>9</b>
1.5.	<b>Productes obtinguts.....</b>	<b>11</b>
1.6.	<b>Breu descripció dels altres capítols.....</b>	<b>11</b>
2.	Anàlisi.....	12
2.1.	<b>Definició del concepte operacional.....</b>	<b>12</b>
2.1.1.	Objectius.....	12
2.1.2.	Descomposició del programari.....	12
2.1.3.	Requisits funcionals.....	12
2.1.4.	Requisits no funcionals.....	13
2.2.	<b>Descripció dels casos d'ús.....</b>	<b>14</b>
2.2.1.	Guió de l'usuari final.....	14
2.2.2.	Guió de l'usuari expert.....	14
2.2.3.	Guió de l'administrador del sistema.....	14
2.2.4.	Diagrama de casos d'ús.....	15
2.2.5.	Descripció dels casos d'ús.....	16
2.3.	<b>Model d'interfície gràfica d'usuari.....</b>	<b>19</b>
2.4.	<b>Diagrama de classes de l'anàlisi.....</b>	<b>24</b>
3.	Disseny.....	25
3.1.	<b>Disseny arquitectònic.....</b>	<b>25</b>
3.2.	<b>Descomposició del programari en paquets.....</b>	<b>29</b>
3.3.	<b>Estructura de menús.....</b>	<b>29</b>
4.	Conclusions.....	30
5.	Biografia.....	32
6.	Deployment de les aplicacions.....	33

## **Índex figures**

Figura 1. Planificació del projecte. Diagrama de Gantt.....	pàgina 10
Figura 2. Diagrama de casos d'ús del SGIC.....	pàgina 15
Figura 3. Aplicació web. Validació d'usuari.....	pàgina 19
Figura 4. Aplicació web. Menú d'informes.....	pàgina 20
Figura 5. Aplicació web. Informe.....	pàgina 20
Figura 6. Aplicació d'escriptori. Aplicació base.....	pàgina 21
Figura 7. Aplicació d'escriptori. Finestra manteniment.....	pàgina 22
Figura 8. Aplicació d'escriptori. Diàleg confirmació.....	pàgina 22
Figura 9. Aplicació d'escriptori. Edició propietats entitat.....	pàgina 23
Figura 10. Aplicació d'escriptori. Edició propietats informe.....	pàgina 23
Figura 11. Aplicació d'escriptori. Afegir elements d'altres entitats del sistema.....	pàgina 23
Figura 12. Diagrama de classes estàtic del SGIC.....	pàgina 24

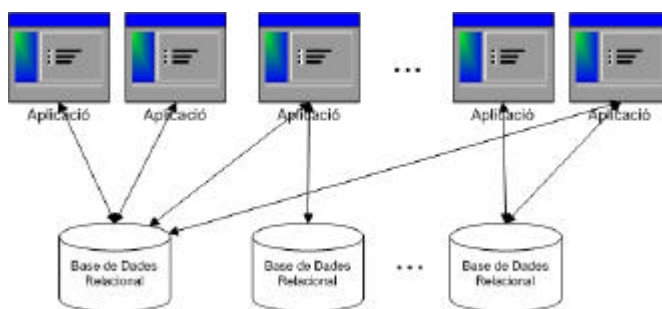
# 1. Introducció

## 1.1. Justificació del PFC

Avui en dia, les empreses disposen de diferents programaris per al tractament de la informació que gestionen en els seus processos quotidians.

Aquests programaris, han estat desenvolupats al llarg de la vida de la empresa mitjançant diferents tecnologies, magatzems de dades i equips de persones.

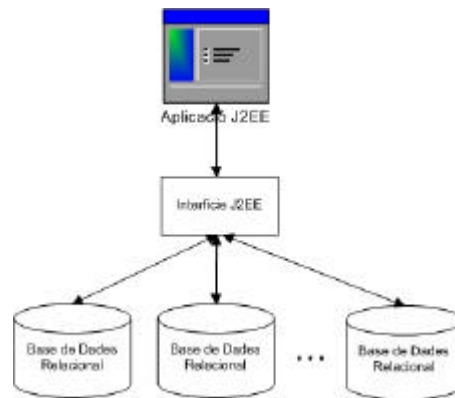
Així, normalment, ens trobem amb un conjunt de bases de dades distribuïdes, diferentment estructurades, i un conjunt de programaris per accedir-hi.



Les empreses necessiten treballar, a diferents nivells, amb les dades que els són necessàries o útils d'aquests diferents magatzems. Molts cops en la feina diària, es necessita contrastar informació d'aquestes dades i s'ha de recórrer a sol·licitar diferents informes a diferents departaments de l'organització.

Es vol crear un entorn de treball que permeti als diferents usuaris de l'empresa, obtenir informació de les seves dades, tant a nivell de direcció, per propòsits de divulgació, ajuda a la presa de decisions, anàlisis i control, com a nivell de consulta de les dades de la operativa bàsica diària per diferents necessitats.

Així, l'objectiu general d'aquest projecte, serà el desenvolupament d'una aplicació que permeti la realització d'informes, fent transparent, a l'usuari final, l'origen, ubicació i tecnologies de les diferents bases de dades distribuïdes, ni programaris amb que necessita treballar.



D'aquesta forma, també es pretén afegir un nivell més d'abstracció que ens permeti interactuar en base de dades relacionals, sense necessitat de gestionar-les directament. Per aconseguir-ho haurem de definir un protocol de definició d'orígens de dades i dissenyar un conjunt d'eines (interfaces) necessàries per implementar les funcionalitats requerides.

Dins de l'àmbit del projecte, s'utilitzaran tecnologies de la plataforma J2EE.

Per la realitzar una connectivitat amb els diferents orígens de dades s'utilitzarà JDBC. JDBC ens abstrau la connectivitat de bases de dades de les aplicacions Java. L'API JDBC ofereix una alternativa a la utilització dels APIs específics dels diferents venedors de bases de dades. Un driver JDBC és una capa de programari intermediari que tradueix les crides JDBC als APIs específics dels venedors. Amb independència d'aquest fet, per accedir a les dades necessitem un driver de base de dades, probablement subministrat pel mateix venedor o per un proveïdor del servidor J2EE.

S'utilitzaran drivers de tipus 4, drivers Java purs. Aquests drivers són els més fàcils de desplegar ja que no hi ha llibreries addicionals ni software intermediari que instal·lar. També ofereixen generalment un millor rendiment que els drivers de tipus 2. Tots els principals venedors de base de dades ofereixen diversos JDBC de tipus 4 per a les seves bases de dades.

Per tal de poder treballar també en les plataformes de Microsoft Windows, s'utilitzarà el pont JDBC-ODBC. Open Database Connectivity (ODBC) proporciona un API estàndard per a SQL en aquestes plataformes. Aquesta solució per al accés a dades no és eficient per a requeriments d'accés a dades d'alt rendiment degut a les múltiples capes de desreferència per a cada crida d'accés a base de dades, i limita la funcionalitat del API JDBC en favor de la funcionalitat del driver ODBC. Per aquests motius, l'ús de JDBC-ODBC només s'utilitzarà quan sigui necessari.

També es podrà utilitzar arxius XML com a orígens de dades. Aquesta tecnologia J2EE per a serveis web, com a llenguatge autodescriptiu, ens permet definir tant

dades com metadades. Aquesta tecnologia ens proporciona un mecanisme d'integració de dades existents en nous sistemes.

Igualment, els informes resultants de l'aplicació es generaran com a arxius XML. Aquesta tecnologia ens permetrà representar els informes resultant en format web i poder-hi així accedir des de diferents tipus de dispositius. La representació de la informació davant l'usuari final es podrà formatà depenent, de les necessitats finals, mitjançant l'estàndard XSLT de la organització W3C. La unió de XML i XSLT permet separar el contingut i la presentació, augmentant així la productivitat.

El servei de generació de informes corporatiu es desenvoluparà mitjançant JSP per tal que pugui ser utilitzat també, des de diferents ubicacions i dispositius.

Aquesta aplicació web, haurà de seguir els mínims recomanats per la Web Accessibility Initiative (WAI) de la organització W3C.

L'eina de definició d'orígens de dades, usuaris i permisos serà implementada com un Applet Java per facilitar el seu tractament i gestió.

## 1.2. Objectius del PFC

- Anàlisi de les tecnologies que permeten utilitzar els diferents orígens de dades.
- Estudi de les representació de les consultes sobre bases de dades relacionals sobre sortides basades en XML i XSLT.
- Facilitat d'ús de la aplicació web i utilització de les directrius de la WAI.

Es vol que aquest entorn disposi de les següents funcionalitats:

- Definició del diccionari de dades amb les diferents fonts de dades amb que es vol treballar.

El sistema ha de permetre definir les diferents fonts de dades de l'empresa. De cada font de dades, haurà de permetre especificar les diferents entitats amb les quals es vol treballar. I per cada entitat, haurà de definir quins camps hi ha disponibles tant per a realitzar la consulta, com per visualitzar com a resultat.

També s'haurà de poder associar a aquestes entitats, els rols de permisos necessaris per tal d'accedir a aquesta informació.

- Eina de definició de consultes i visualització de la informació resultant

Mitjançant aquest eina, els usuaris finals del sistema podran realitzar les consultes sobre les dades que estiguin autoritzats, i visualitzar i/o exportar el resultat obtingut a partir de la consulta.

- Gestió d'usuaris

Manteniment d'usuaris i assignació de rols d'usuaris per tal de poder gestionar els permisos sobre les diferents utilitats del programari i sobre les dades disponibles.

- Accés controlat al sistema

El sistema haurà de controlar quins són els usuaris autoritzats a gestionar l'aplicació i a quins conjunts de dades del diccionari definit hi poden accedir.

Tindrem dos tipus d'usuaris:

- Els administradors del sistema, que hauran de poder definir el diccionari de dades amb que pot treballar el sistema, així com gestionar els usuaris i els permisos necessaris sobre la informació definida.
- Els usuaris de consulta final, que hauran de poder accedir a la definició de consultes i visualització de la informació resultant.

- Utilització de la signatura digital per part dels usuaris o a nivell d'aplicació.

Estaria bé que el sistema tingués la possibilitat d'utilitzar algorismes de signatura digital per tal de firmar digitalment els llistats de dades resultants en suport electrònic en cas de necessitat.

### **1.3. Enfocament i mètode seguit**

El desenvolupament del projecte s'ha fet pensant en seguir, de manera simplificada, les fases del procés unificat de desenvolupament de programari, utilitzant les tecnologies de la plataforma J2EE en la fase d'implementació.

El fet de seguir les metodologies de la enginyeria del programari ens permetrà el desenvolupament de les diferents aplicacions amb independència d'una de l'altra, i de forma simultània.

Es pretén obtenir una aplicació funcional, mitjançant el desplegament de diferents components, desenvolupats amb diferents tecnologies, que corresponen a les



aplicacions clients de la part d'administració i la d'accés i visualització dels informes.

Així, es pretén obtenir un producte de qualitat on la facilitat d'ús sigui una característica fonamental al haver de poder ser utilitzat per qualssevol persona de l'organització i es tingui en compte a les persones amb discapacitats.

#### 1.4. Planificació del PFC

A la planificació del projecte es tenen en compte les dates dels diferents lliuraments parcials i les diferents tasques a desenvolupar.

Així les dates dels lliuraments són les següents:

Títol	Inici	Lliurament
PAC1	28/02/2008	14/03/2008
PAC2	15/03/2008	14/04/2008
PAC3	15/04/2008	19/05/2008
Entrega Final	20/05/2008	25/06/2008

Les fases del projecte seran les següents:

- Planificació
- Anàlisi
- Disseny
- Implementació
- Proves

El temps estimat de cada tasca i la seva temporalització i fites importants es representa en el diagrama de GANNT següent:

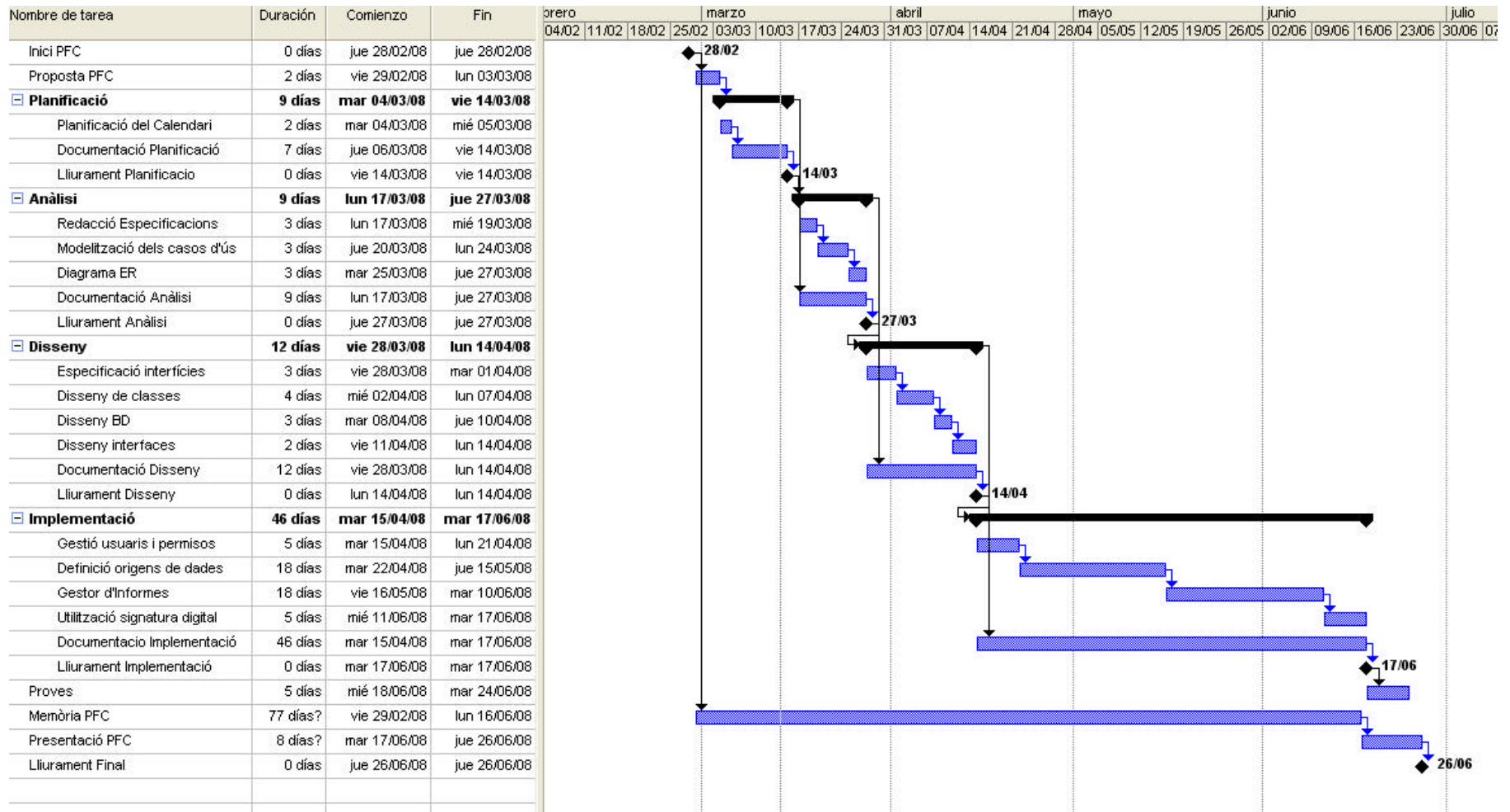


Figura 1. Planificació del projecte. Diagrama de gannt

## **1.5. Productes obtinguts**

En aquest projecte s'obtenen dos productes: una aplicació d'escriptori per l'administració del nou sistema i una aplicació web per l'accés als informes i la informació que contenen.

Tots dos productes funcionen sense la necessitat d'utilitzar cap base de dades que no siguin les orígens de dades als que es volen accedir.

L'aplicació d'escriptori l'utilitzarà l'administrador del sistema o usuaris experts. Aquests usuaris seran els responsables de la definició i manteniment dels diferents orígens de dades, diccionari de dades, informes, usuaris i permisos.

L'aplicació d'escriptori registrarà en fitxers XML, com a magatzem de dades, la lògica del sistema que es vulgui definir.

L'aplicació web l'haurà de poder utilitzar qualsevol usuari de l'organització des de diferents tipus de dispositius. Els usuaris hauran de poder accedir als informes que tinguin autorització.

L'aplicació web serà dinàmica a partir dels fitxers de configuració XML definits i els orígens de dades.

## **1.6. Breu descripció dels altres capítols**

### **Anàlisi**

Es documenta extensament la fase d'anàlisi del projecte. S'hi especifiquen els objectius, la descomposició del programari, els requisits, els casos d'ús, el model d'interfície gràfica, i les classe d'anàlisi.

### **Disseny**

Es detallen les solucions de disseny que s'utilitzen en els diferents components del programari. Es comenten les solucions arquitectòniques que s'han pres

### **Conclusions**

S'analitza el producte final. S'especifiquen millores a implementar en el projecte si s'hagués tingut més temps disponible.

## 2. Anàlisi

### 2.1. Definició del concepte operacional

#### 2.1.1. Objectius

El programari *Servei de Generació de Informes Corporatiu*, SGIC en endavant, és una aplicació que implementa una ferramenta per la generació de informes corporatius fent transparent a l'usuari final del sistema, els programaris de gestió i l'origen, ubicació i tecnologies de les bases de dades distribuïdes a les que accedeix, i que té com a objectiu l'estudi i utilització de les tecnologies de la plataforma J2EE.

El SGIC s'oferirà als usuaris que s'hagin donat d'alta al sistema, i els informes finals s'implementaran mitjançant tecnologies web per que puguin ser consultats a través d'Internet des de qualsevol ubicació.

#### 2.1.2. Descomposició del programari

El programari disposarà d'una part d'administració en la que es definiran els elements necessaris per la gestió d'usuaris, permisos i realització de les consultes, un altra part per al disseny de les consultes, aquestes dues parts s'implementaran mitjançant un applet Java, i un altra part web amb tecnologia J2EE, per a l'accés als informes resultants.

#### 2.1.3. Requisits funcionals

El sistema ha de permetre la utilització d'algorismes de signatura digital per tal de firmar digitalment els llistats de dades resultants en suport electrònic en cas de necessitat.

Dins el sistema podrem disposar de tres perfils d'usuari:

L'*administrador*, que gestionarà els usuaris, permisos, orígens de dades i el diccionari de dades de que disposarà el sistema.

Un *usuari expert* que podrà definir informes a partir del diccionari de dades del sistema.

Un *usuari final* que disposarà d'accés als informes publicats.

L'usuari expert heretarà les funcionalitats de l'usuari final i l'administrador les heretarà del usuari expert.

L'aplicació haurà de gestionar diferents nivells de permisos. Així, s'haurà de poder definir el rol de cada tipus d'usuari, donar permisos de visibilitat sobre les entitats, tant als usuaris experts com als usuaris finals i permisos de visibilitat sobre els informes, si és necessari.

Del conjunt de informes publicats a la part web amb accés per als usuaris finals, l'aplicació ha de permetre gestionar permisos sobre els informes per tal de poder discriminar quins usuaris del sistema els poden veure.

#### **2.1.4. Requisits no funcionals**

L'aplicació utilitzarà com a orígens de dades corporatius i per a la gestió del diccionari de dades, els usuaris i permisos, bases de dades relacionals, maximitzant l'ús de estàndards i la persistència de les dades. El sistema ha d'estar obert a la possibilitat de que és puguin utilitzar arxius XML com a orígens de dades.

Els informes resultants del sistema es representaran mitjançant arxius XML que es podran formatar mitjançant l'estàndard XSLT depenent de les necessitats finals.

Com a base de dades relacional necessària per al funcionament del sistema, s'utilitzarà Microsoft SQL Server en la seva versió gratuïta.

Com a patró arquitectònic, utilitzarem arquitectura de 3 capes on es diferencia clarament la capa de representació, la de la lògica del negoci i l'accés a les dades, així podrem treballar en el disseny i implementació de cada capa amb un nivell d'abstracció diferent.

## **2.2. Descripció dels casos d'ús**

El sistema té tres actors ben diferenciats: l'administrador del sistema, l'usuari expert i l'usuari final. L'administrador del sistema podrà utilitzar les mateixes funcionalitats que l'usuari expert i aquestes mateixes que l'usuari final, així trobem dues relacions de herència per especialització.

### **2.2.1. Guió de l'usuari final**

L'usuari final del sistema serà aquell que pot accedir als serveis web de visualització de informes.

Prèvia identificació, accedirà als informes dels que tingui permís de consulta, poden elegir aquell/s que vulgui visualitzar.

### **2.2.2. Guió de l'usuari expert**

L'usuari expert, a part de poder accedir al servei web de visualització de informes com a usuari final, podrà accedir a l'applet Java, al generador de informes corporatiu.

També prèvia identificació dins l'aplicació, gestionarà les consultes dels diferents informes i gestionarà les publicacions i permisos sobre els mateixos. Els permisos sobre els informes prevaldran als permisos sobre les entitats.

### **2.2.3. Guió de l'administrador del sistema**

L'administrador del sistema tindrà accés a totes les funcionalitats del sistema. Hereta totes les funcionalitats de l'usuari expert i a més serà ell qui defineixi els usuaris, els rols dels usuaris dins el sistema, els orígens i el diccionari de dades i els permisos sobre les entitats del diccionari.

## 2.2.4. Diagrama de casos d'ús

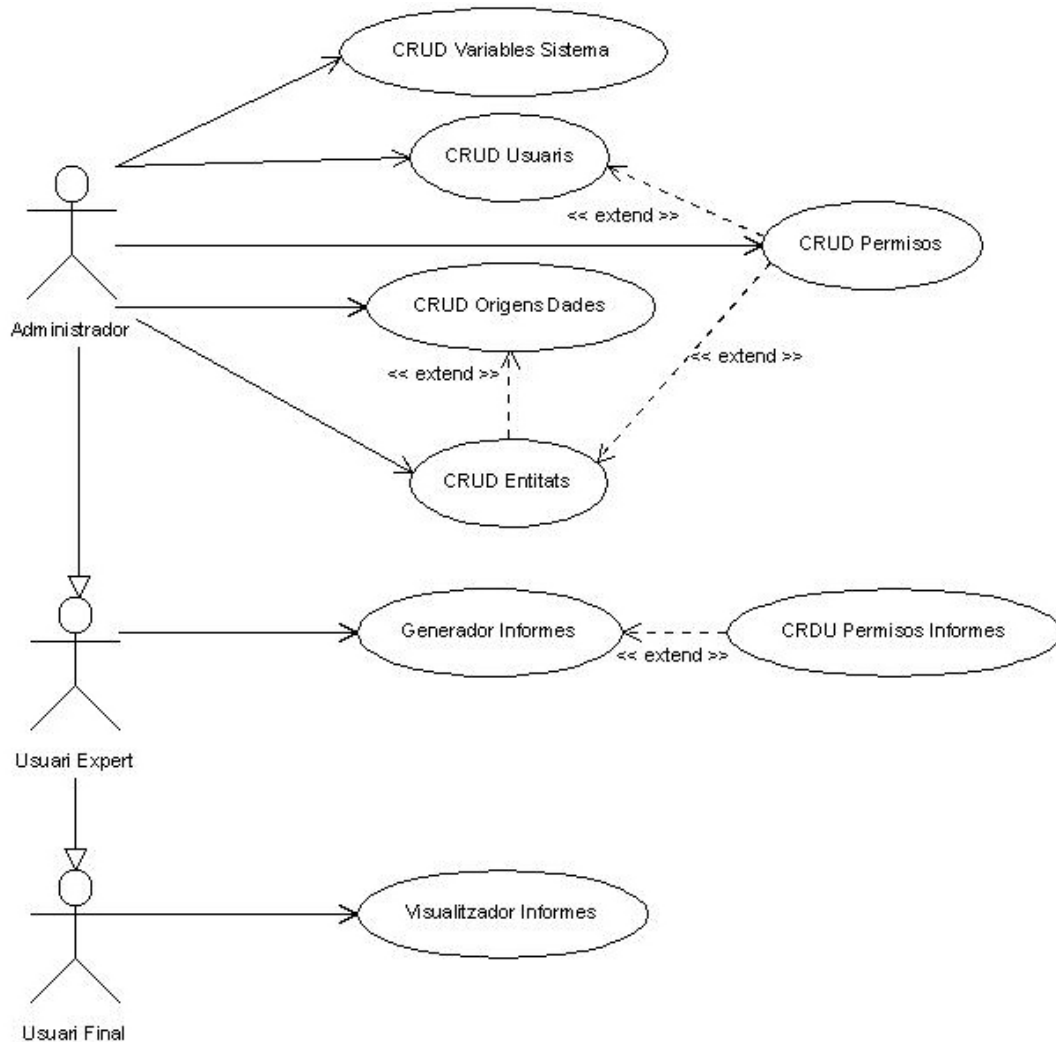


Figura 2. Diagrama de casos d'ús del SGIC

## 2.2.5. Descripció dels casos d'ús

No s'especifica el cas d'ús d'identificació de l'usuari o login que serà necessari per tal de poder accedir a qualsevol funcionalitat del sistema i saber així quin dels rols pot desenvolupar l'usuari del sistema.

S'identifica amb les sigles CRUD l'aglomeració dels casos d'ús de Alta/Consulta/Modificació/Baixa que es tracten amb un únic manteniment dins de l'aplicació.

- **CRUD Variables Sistema**

- **Descripció de la funcionalitat:** l'administrador del sistema podrà configurar aquelles variables del sistema que es puguin parametritzar. Per exemple, el directori per defecte de emmagatzemament dels informes. Igualment, podrà definir variables que puguin ser d'utilitat i utilitzar-se en l'elaboració dels informes.
- **Paper dins el treball de l'usuari:** és un cas d'ús secundari.
- **Actors:** Administrador.
- **Casos d'ús relacionats:** cap.
- **Precondició:** les variables del sistema poden estar o no assignades i poden no existir en la base de dades. Hi ha variables que no es poden eliminar.
- **Postcondició:** les variables del sistema han pogut estar creades, eliminades o modificades en la base de dades.

- **CRUD Usuaris**

- **Descripció de la funcionalitat:** l'administrador podrà gestionar els usuaris del sistema i assignar-los el rol que els pertorqui. Es podrà accedir a l'assignació als usuaris dels permisos de visibilitat sobre les entitats.
- **Paper dins el treball de l'usuari:** és un cas d'ús principal.
- **Actors:** Administrador.
- **Casos d'ús relacionats:** CRUD Permisos.
- **Precondició:** els usuaris poden existir o no a la base de dades i si existeixen tenen un rol assignat.
- **Postcondició:** els usuaris han pogut estar creats, eliminats o modificats a la base de dades. Sempre tindran un rol definit.

- **CRUD Orígens de dades**

- **Descripció de la funcionalitat:** l'administrador podrà configurar els diferents orígens de dades que li serviran per crear el diccionari de dades i accedir a la informació al realitzar les consultes. Es podrà accedir a la gestió de les entitats del diccionari de dades.



- **Paper dins el treball de l'usuari:** és un cas d'ús principal.
  - **Actors:** Administrador.
  - **Casos d'ús relacionats:** CRUD Entitats.
  - **Precondició:** els orígens de dades poden existir o no a la base de dades.
  - **Postcondició:** els orígens de dades han pogut estar creats, eliminats o modificats a la base de dades.
- **CRUD Entitats**
    - **Descripció de la funcionalitat:** L'administrador podrà definir el diccionari de dades seleccionant aquelles entitats dels diferents orígens de dades que vulgui afegir i indicant quins camps es faran visibles. Es podrà accedir a l'assignació als usuaris dels permisos de visibilitat sobre les entitats. Es pot accedir des de el menú principal de l'aplicació o des de el CRUD de orígens de dades.
    - **Paper dins el treball de l'usuari:** és un cas d'ús principal.
    - **Actors:** Administrador.
    - **Casos d'ús relacionats:** CRUD Orígens de dades, CRUD Permisos.
    - **Precondició:** les entitats poden existir o no a la base de dades.
    - **Postcondició:** les entitats han pogut estar creades, eliminades o modificades a la base de dades.
  - **CRUD Permisos**
    - **Descripció de la funcionalitat:** L'administrador podrà gestionar els permisos de visibilitat dels usuaris sobre les entitats del diccionari de dades. Es pot accedir des de el CRUD de usuaris i el CRUD de entitats.
    - **Paper dins el treball de l'usuari:** és un cas d'ús principal.
    - **Actors:** Administrador.
    - **Casos d'ús relacionats:** CRUD Usuaris, CRUD Entitats.
    - **Precondició:** els permisos poden existir o no a la base de dades.
    - **Postcondició:** els permisos poden ser afegits o eliminats a la base de dades.
  - **Generador Informes**
    - **Descripció de la funcionalitat:** els usuaris autoritzats podran definir informes sobre la base de dades. Per fer-ho seleccionaran la o les entitats que hi participen, les relacions que hi puguin haver entre elles i els camps que es podran visualitzar com a resultat del informe. Igualment, haurà de poder publicar o deixar de fer públics aquells informes que desitgi. Es pot accedir al manteniment dels permisos dels informes.
    - **Paper dins el treball de l'usuari:** és un cas d'ús principal.
    - **Actors:** Administrador i usuari expert.
    - **Casos d'ús relacionats:** CRUD Permisos Informes.
    - **Precondició:** la definició dels informes pot existir o no a la base de dades.

- **Postcondició:** la definició dels informes pot haver estat creada, eliminada o modificada a la base de dades.
  
- **CRUD Permisos Informes**
  - **Descripció de la funcionalitat:** l'usuari expert haurà de poder donar permisos als usuaris finals sobre els informes que ha publicat. Es pot accedir a aquesta funcionalitat des de el generador de informes.
  - **Paper dins el treball de l'usuari:** és un cas d'ús principal.
  - **Actors:** Administrador i usuari expert.
  - **Casos d'ús relacionats:** Generador Informes.
  - **Precondició:** els permisos sobre els informes poden existir o no a la base de dades.
  - **Postcondició:** els permisos sobre els informes hauran pogut ser afegits, eliminats o modificats a la base de dades.
  
- **Visualitzador Informes**
  - **Descripció de la funcionalitat:** l'usuari final haurà de poder accedir al menú d'informes disponibles amb els permisos relacionats al seu usuari del sistema. Podrà navegar depenent de la classificació existent i accedir al resultat final dels informes.
  - **Paper dins el treball de l'usuari:** és un cas d'ús principal.
  - **Actors:** Administrador, usuari expert i usuari final.
  - **Casos d'ús relacionats:** cap.
  - **Precondició:** l'usuari ha d'estar validat al sistema per tal de saber de quins permisos disposa sobre els informes existents.
  - **Postcondició:** cap.

### 2.3. Model d'interfície gràfica d'usuari

El programari SGIC està compost de dos parts ben diferenciades:

La part web a la que podran accedir els usuaris finals, i que disposarà d'un menú amb la classificació dels informes per tal de que sigui còmoda la cerca dels informes requerits.

Aquesta part és desenvolupa mitjançant tecnologia web per tal que sigui accessible des de qualssevol ubicació i des de qualssevol tipus de dispositiu. És dissenyarà una interfície d'usuari senzilla, consistent i de fàcil navegabilitat.

Aquesta aplicació s'iniciarà amb una web per la identificació i validació dels usuaris. A partir d'aquesta web accedirem a la següent web des de la que podrem visualitzar i seleccionar els diferents informes sobre els que tinguem permís. Un cop seleccionat un informe, se'ns obrirà una nova finestra del navegador amb el contingut de la informació resultant de l'informe.

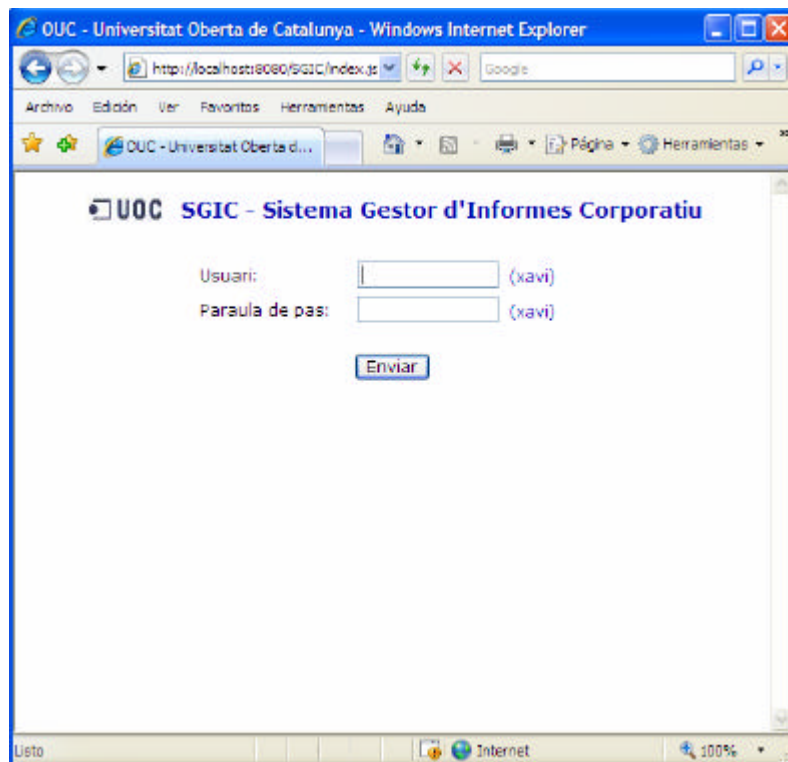


Figura 3. Aplicació web. Validació d'usuari

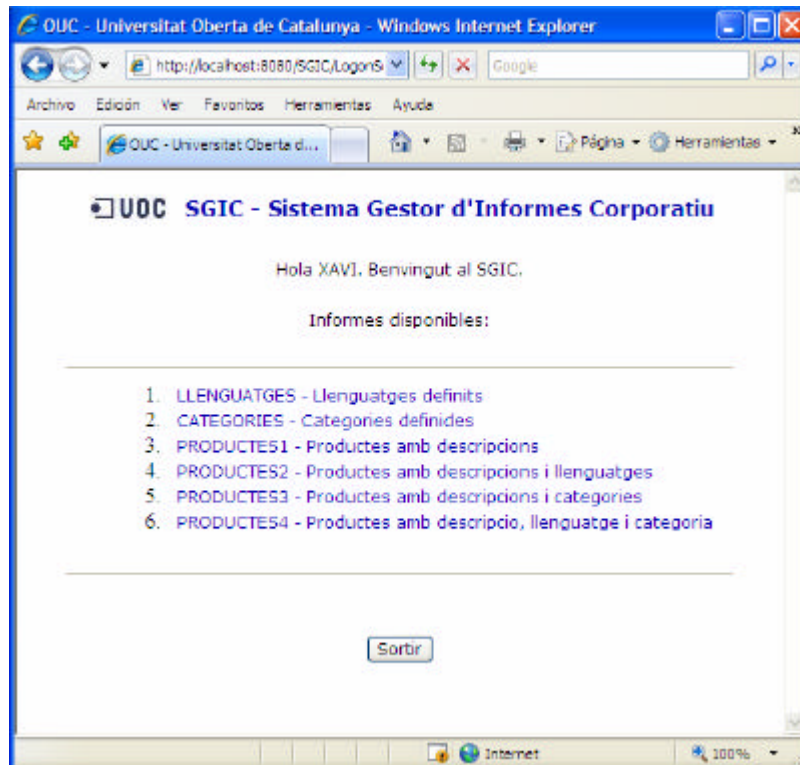


Figura 4. Aplicació web. Menú d'informes

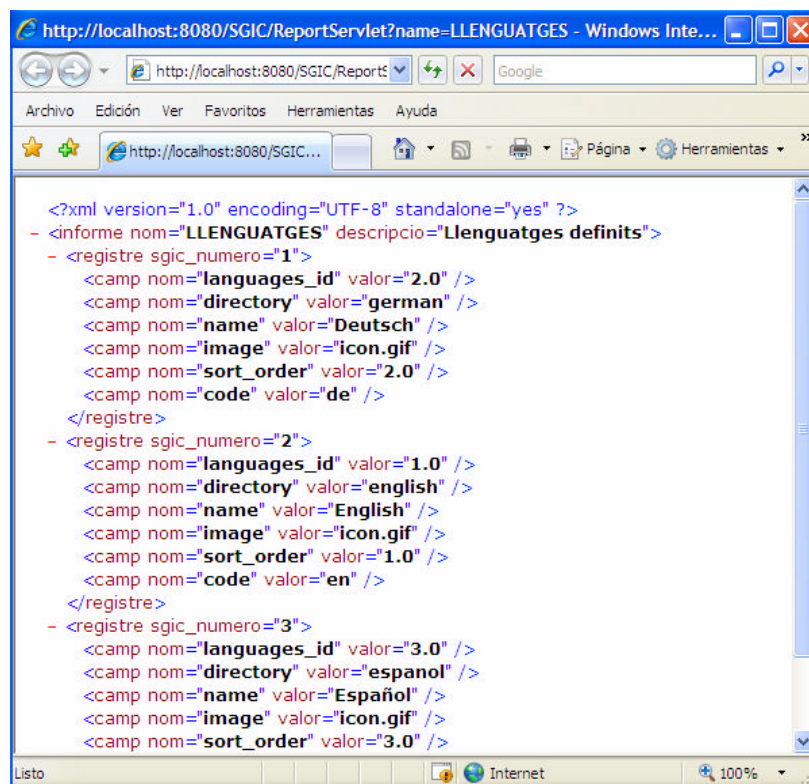


Figura 5. Aplicació web. Informe

La part d'administració i gestió d'informes, que permetrà al usuari expert definir els informes a partir del diccionari de dades existent, publicar-los i assignar-los permisos de quins usuaris finals els podran visualitzar en cas de ser convenient. I a l'administrador del sistema li permetrà definir els usuaris del sistema, el seu rol, els orígens de dades, les entitats del diccionari de dades i els permisos sobre les mateixes. Es dissenyarà un menú d'aplicació, on es mostraran les funcionalitats depenen del rol de l'usuari (administrador o usuari expert) i a partir del qual podrem accedir als diferents manteniments.

Aquesta part es desenvoluparà mitjançant una aplicació Java convencional per a que ens sigui més còmode el tractament i gestió de les dades que s'han de definir i treballar.

L'aplicació constarà en la part superior d'un menú on podrem trobar totes les funcionalitats del sistema segons els permisos de que disposem. En la part central disposarem d'un escriptori on podrem anar treballant i gestionant les diferents finestres que anem obrint.



Figura 6. Aplicació d'escriptori. Aplicació base

En aquest escriptori se'ns aniran obrint les finestres dels manteniments que anem seleccionant.

Al seleccionar un manteniment, ens apareixerà una finestra, dins l'escriptori de l'aplicació on disposarem d'una taula central amb els diferents elements que contingui l'entitat de la que estem realitzant el manteniment. En la part dreta de l'aplicació disposarem de botons funcionals per tal de poder realitzar l'alta, baixa i edició de l'element del manteniment que tinguem seleccionat en la taula.

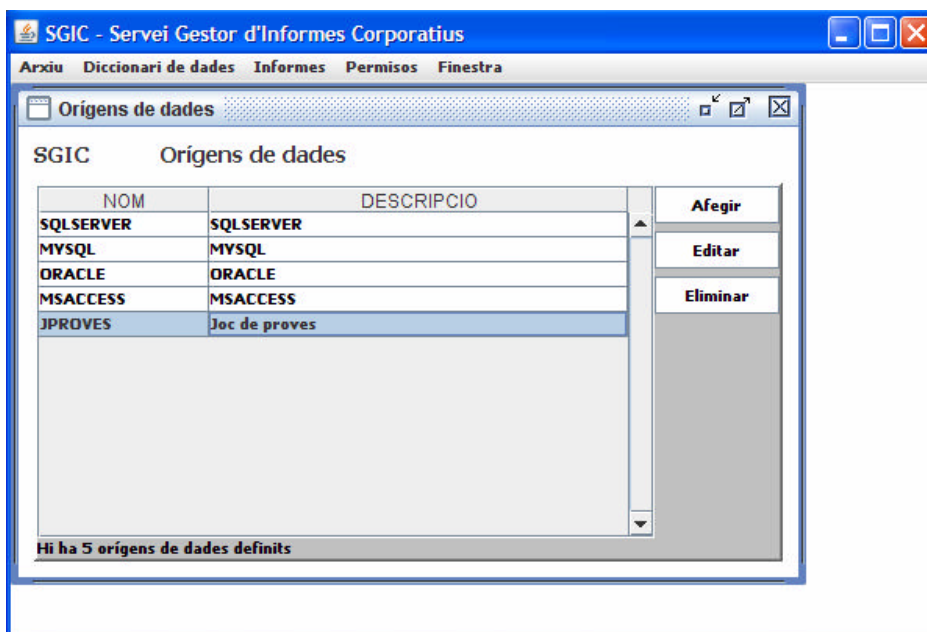


Figura 7. Aplicació d'escriptori. Finestra manteniment

Per tal de eliminar un element, l'hauréu de tenir seleccionat i ens haurà d'aparèixer un diàleg modal de confirmació de l'eliminació, per tal que sigui confirmada la baixa de l'element.

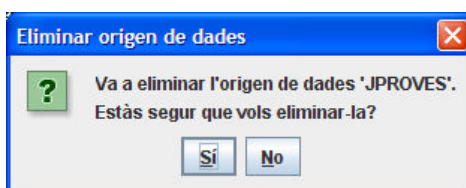


Figura 8. Aplicació d'escriptori. Diàleg confirmació

Al editar o donar d'alta un element d'una entitat, ens apareixerà un diàleg modal on, en la part central ens apareixeran els camps corresponents a les propietats de l'entitat que es puguin mantenir, i en la part inferior ens apareixeran els botons per tal d'acceptar o cancel·lar els canvis. En el cas en que la finestra d'edició de les propietats de la entitat, requereixi de més botons funcionals, es col·locaran en la part inferior, amb els botons d'acceptar o cancel·lar.

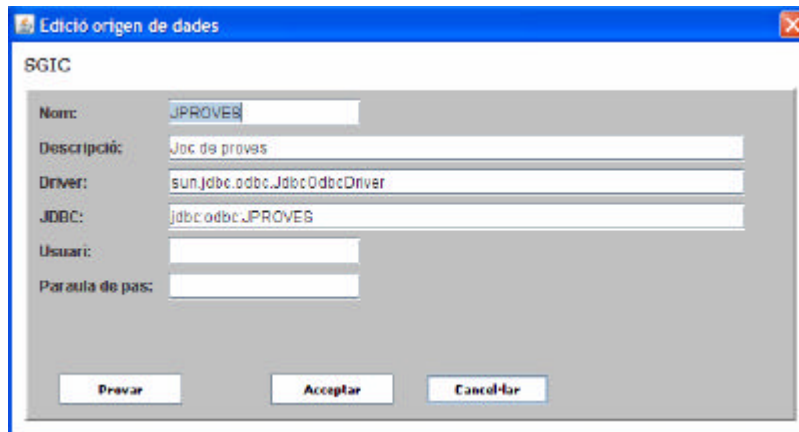


Figura 9. Aplicació d'escriptori. Edició propietats entitat

El diàleg d'edició dels informes, disposarà de dos taules per tal de poder mantenir i visualitzar d'una forma fàcil les entitats i joins que defineixen l'informe.

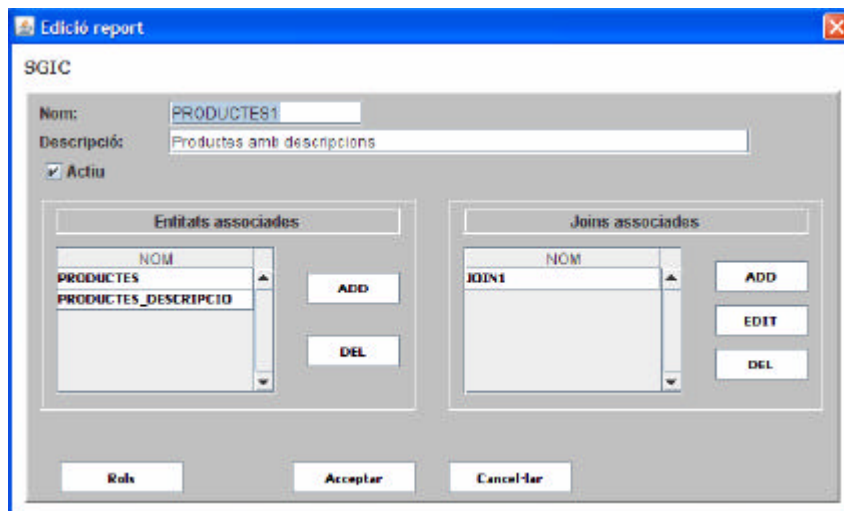


Figura 10. Aplicació d'escriptori. Edició propietats informe

Per tal d'afegir elements definits com a altres entitats, com les entitats del diccionari de dades o els rols, mostrarem un diàleg modal amb una taula amb la llista de elements i els botons d'acceptar i cancel·lar en la part inferior.

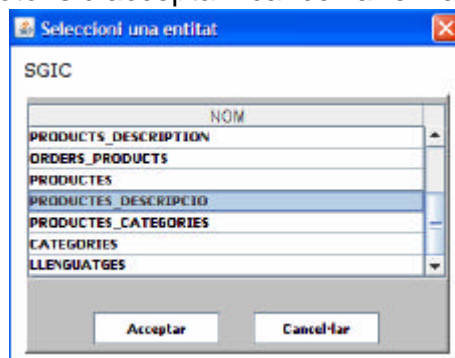


Figura 11. Aplicació d'escriptori. Afegir elements d'altres entitats del sistema

## 2.4. Diagrama de classes de l'anàlisi

En el diagrama de classes de l'anàlisi definim les entitats necessàries per a la resolució de les funcionalitats de l'aplicació segons els casos d'ús que s'han especificat.

En el diagrama i per facilitar la llegibilitat del mateix, no s'han especificat els mètodes accessoris de les classes.

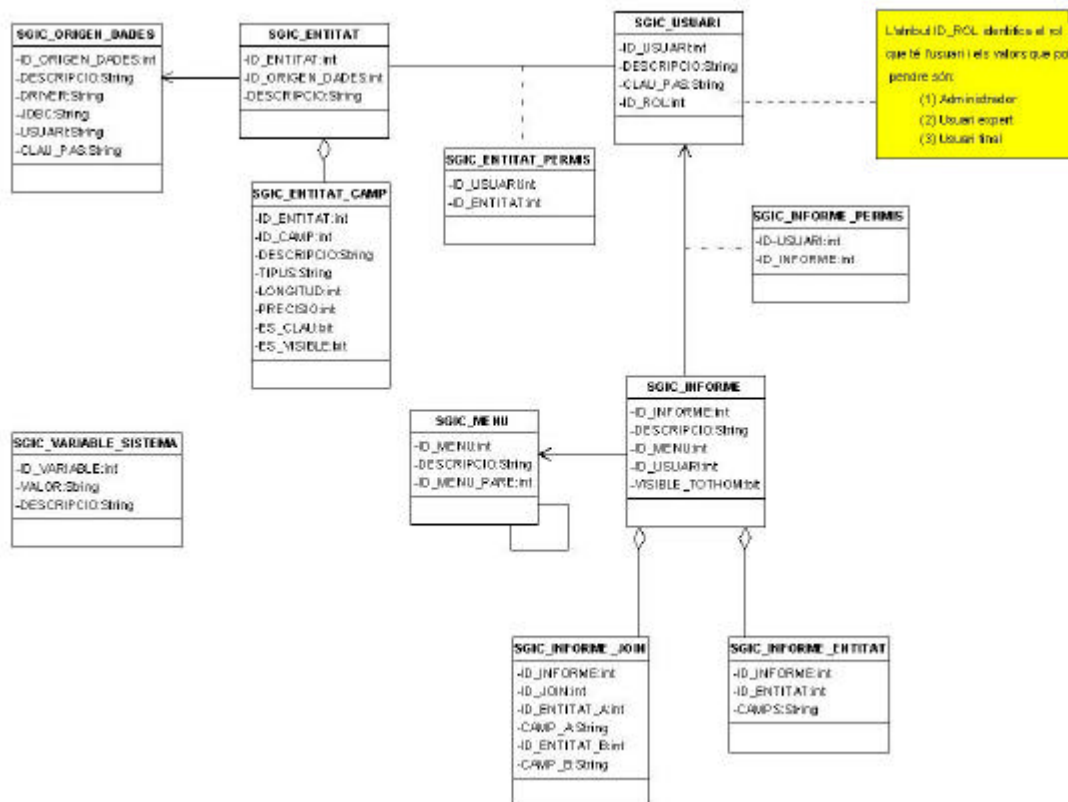


Figura 12. Diagrama de classes estàtic del SGIC



## 3. Disseny

### 3.1. Disseny arquitectònic

Com a disseny arquitectònic entendrem la definició de les relacions entre cadascun dels elements estructurals del programa. Així, haurem de identificar els subsistemes i mòduls que formaran la nova aplicació i la forma de controlar-los i comunicar-los.

El SGIC el descomposarem en 5 subsistemes:

- **Subsistema de usuaris i permisos:** aquest subsistema permetrà la gestió dels usuaris del sistema. Podran donar-se d'alta, de baixa, consultar-los i modificar-ne els atributs. També permetrà la identificació dels usuaris dins el sistema, i l'assignació de permisos als usuaris tant a les entitats com als informes.
- **Subsistema del diccionari de dades:** ens permetrà la gestió dels orígens de dades, així com de les entitats que compondran el diccionari de dades. Podrem realitzar el manteniment complet.
- **Subsistema d'elaboració de informes:** permetrà la gestió dels informes corporatius. Es podrà realitzar el manteniment complet, essent alta o edició dels informes el component principal.
- **Subsistema de generació de informes:** a partir del informe seleccionat, generarà segons la seva definició, la sortida amb la informació resultant obtinguda dels orígens de dades.

En la actualitat, Java és un dels llenguatges de programació més elaborats i utilitzats per a la creació de software d'empresa. J2EE és una plataforma que proporciona la infraestructura necessària per a la creació d'aplicacions de servidor en Java.

Així, J2EE ens ofereix un conjunt de tecnologies estàndards, independents de la plataforma que utilitzem, que ens permetran augmentar la productivitat, mitjançant la modularitat i la reusabilitat, disposar de fiabilitat i disponibilitat en les aplicacions, assegurar-nos la seguretat de la informació i la integració entre diferents aplicacions.

Entre d'altres, disposem de les següents tecnologies J2EE per a la construcció de d'aplicacions distribuïdes de empresa, les classifi quem segons el seu ús:

- Tecnologies de components: són utilitzades per a albergar la major part de la aplicació: la lògica de empresa. Existeixen 4 tipus de components: pàgines JSP, XML, servlets i Enterprise JavaBeans.
- Tecnologies de serveis: proporcionen els components de l'aplicació amb serveis reforçats per a que funcionen d'un mode eficient. Tenim JDBC, Java Transaction API (JTA), Java Transaction Service (JTS), Java Naming and Directory Interface (JNDI), Java Message Service (JMS), Java Authentication and Authorization Service (JASS), ...
- Tecnologies de comunicació: proporcionen mecanismes per a la comunicació entre les diferents parts de l'aplicació, ja siguin locals o remotes. Disposem de protocols de Internet com HTTP, TCP/IP, i SSL, i protocols de objecte remot com Remote Method Invocation (RMI), RMI-IIOP i JavaIDL per accés a CORBA.

Utilitzarem l'arquitectura de tres capes. Així dividirem l'aplicació en tres capes lògiques diferenciades, cadascuna d'elles amb un grup de interfaces perfectament definit.

- La capa de representació: consistirà en les interfícies de usuari.
- La capa intermèdia o capa d'empresa: consisteix en la lògica de la empresa.
- La capa de dades: consistent en les dades necessàries per a la aplicació.

Per tal de facilitar el manteniment, disminuir la complexitat, millorar el comportament i afavorir la reutilització del disseny, utilitzarem la experiència dels patrons arquitectònics i de disseny.

#### - **Patró arquitectònic: Model, Vista i Controlador (MVC)**

Aquest model organitza les responsabilitats d'interacció amb l'usuari. Es vol desacoblar la interfície gràfica de la resta del nostre sistema. Així divideix el sistema en tres tipus de components:

- Model: encapsula l'estat i domini del sistema implementat les funcionalitats i dades del sistema.
- Vista: components amb que representem de dades als usuaris i recullen les seves interaccions.
- Controlador: estableix la correspondència entre les accions de l'usuari i els esdeveniments del sistema. També decideix quines vistes es mostren als usuaris.

Ens permetrà també utilitzar diferents tecnologies de visualització.

També permet canviar la lògica del negoci o la de visualització sense haver de canviar l'altra al ser totalment independents.

En la part web del sistema, l'**Struts** ens permetrà, amb un únic controlador, avaluar les peticions de l'usuari mitjançant un arxiu configurable.

#### - **Patró arquitectònic: Injecció de dependències**

Aquest patró ens permetrà canviar la implementació del contingut dels informes, sense que hagi de canviar la capa de representació dels mateixos. Així, mitjançant l'editor de informes podrem canviar l'origen o informació del informe o servei, sense que hi depengui cap implementació concreta, sinó de la interfície del servei.

#### - **Patró de disseny: Data Access Object (DAO)**

Utilitzarem aquest patró en la capa de dades per abstraure i encapsular tots els accessos a les diferents fonts de dades al gestionar-nos la connexió per l'accés a les dades persistentment.

Aquest patró sorgeix històricament de la necessitat de gestionar diversitat de fonts de dades, encara que el seu ús s'estén al problema d'encapsular la font de dades i a més ocultar la forma d'accedir a les dades.

Així el programari es centrarà en les dades que necessita i s'oblida de com es realitza l'accés a les dades i quina és la font d'emmagatzemament.

Les aplicacions poden utilitzar l'API JDBC per accedir a les dades de una base de dades relacional. Aquest API permet una forma estàndard d'accedir i manipular les dades de una base de dades relacionals. Permet a les aplicacions J2EE utilitzar sentències SQL, que són el mètode estàndard per accedir a les taules i vistes. La idea d'aquest patró és ocultar la font de dades i la complexitat de la utilització de JDBC a la capa de representació o de negoci.

Així DAO ens defineix la relació entre la lògica de presentació i empresa, i la de dades.

#### - **Patró de disseny: Composite View**

En la capa de representació, en la part web, utilitzarem el patró de disseny Composite View per la navegació pels informes. Així, la vista del usuari estarà composta de forma modular per diferents subvistes implementades mitjançant fitxers JSP o sortides de servlets.

- **Patró de disseny: Instància única (Singleton)**

S'implementarà una classe d'instància única, que només podrà ser instanciada per ella mateixa, per tal totes les classes que accedeixin a la base de dades en la nostra aplicació ho facin per mitjà de la mateixa instància, de forma que puguem controlar fàcilment el nombre de connexions establertes contra les diferents base de dades, així com possibles catxes de dades emmagatzemades en memòria per l'aplicació.

- **Patró de disseny: Ajudant de Vista**

Per tal de no sobrecarregar ni duplicar codi Java les pàgines JSP, utilitzarem aquest patró per encapsular el codi Java en classes d'ajuda . Així la vista delega el processament de empresa o de presentació en l'objecte ajudant.

Així en JSP permet als dissenyadors web interactuar amb ell mitjançant l'ús de etiquetes JSP estàndards o personalitzades.

### 3.2. Descomposició del programari en paquets

En l'aplicació web, els Servlets estaran continguts en un únic paquet. Les pàgines JSP estaran contingudes en l'arrel de l'aplicació i les pàgines JSP comunes, capçalera, peu, ... i el full d'estils de l'aplicació estaran contingudes en la carpeta includes.

L'aplicació d'escriptori, la descomposarem en els següents paquets:

- app: contindrà la classe principal del programari així com els controladors necessaris de l'aplicació.
- db: contindrà l'ajudant i les classes necessàries per a l'accés a base de dades.
- gui: contindrà les classes corresponents a la vista de l'aplicació.
- model: contindrà l'ajudant i les classes del model de l'aplicació.

### 3.3. Estructura de menús

L'aplicació d'escriptori disposarà d'un menú d'aplicació. Aquest estarà estructurat de la següent manera:

- submenú Arxiu: contindrà el manteniment de variables i l'opció sortir de l'aplicació.
- submenú Diccionari de dades: contindrà el manteniment d'origens de dades i de les entitats del diccionari de dades.
- submenú Informes: contindrà el manteniment d'informes.
- submenú permisos: contindrà el manteniment d'usuaris i permisos.
- submenú finestra: contindrà funcions d'ajuda per a interactuar amb l'escriptori de l'aplicació.

## 4. Conclusions

El programari s'ha desenvolupat pensant en aconseguir un producte de qualitat i de fàcil utilització.

El resultat del projecte és una aplicació, a partir dels requeriments, anàlisi i disseny especificat, que permeti a l'usuari realitzar un treball productiu i alliberar-lo de la necessitat de saber on és i com accedir a la informació de la seva empresa.

Així, el que permet la aplicació és la definició de informes a partir dels orígens de dades distribuïts que disposen a les empreses, abstractant a l'usuari final la forma d'obtenir les dades i permeten creuar informació que es troba en diferents formats, estructures i ubicacions dins de la nostra empresa.

Aquest projecte és un primer pas cap a una solució tecnològica que ens permeti la creació d'un portal homogeni i centralitzat dels informes i informació que és gestionen en els diferents processos de les empreses.

Com a millores que es poden enumerar una sèrie de suggerències que permetin finalitzar el treball i potenciar-lo per poder gaudir de molta més funcionalitat i millorar la presentació final.

Així, enumero les següents millores:

- Manca la realització dels manuals de la aplicació.
- Manca un estudi de usabilitat, a partir de la utilització del programari per usuaris finals.
- Manca el control d'accés i permisos sobre les diferents funcionalitats de la aplicació d'escriptori per a que pugui ser utilitzada tant per l'administrador del sistema com per usuaris experts.
- Manca la implementació i manteniment d'un sistema de menús tipus arbre en els informes finals per tal de poder-los estructurar i agrupar.
- Parametrització de la ubicació dels arxius XML amb la lògica del sistema des de la aplicació web.
- Elecció dels camps que es vulguin utilitzar de les entitats als informes, així com la creació d'àlies als camps per possibles problemes de nomenclatura.
- Estudi i implementació de varios exemples de representació de la informació mitjançant XSLT.
- Estudi i utilització de certificats digitals per signar els informes resultants.

- Implementació d'una major potència en la definició de les joins dels informes per tal de poder gaudir de moltes més possibilitats.
- Creació del concepte “condició” en els informes per tal de poder especificar condicions amb valors fixes en les consultes que defineixen els informes.
- Utilització de paràmetres en els informes que ens permetin realitzar criteris de cerca més acurats sobre la informació resultant, així com el sistema per que siguin especificats per els usuaris finals.

## 5. Biografia

- Programación Java Server con J2EE Edición 1.3. Editorial Anaya
- Manual de XML. Editorial Prentice Hall
- Java y XSLT. Editorial Anaya / O'Reilly
- Usabilidad. Diseño de sitios web. Editorial Prentice Hall
- J2EE Desing patterns. Ed. O'Reilly
- <http://www.w3.org/WAI>
- <http://www.ibm.com> -> Simplify XML programing with JDOM
- <http://www.google.com> -> Sant Google en general



## 6. Deployment de les aplicacions

El projecte està implementat mitjançant dues aplicacions ben diferenciades. Una aplicació d'escriptori per la definició i manteniment dels informes corporatius, així com dels usuaris i permisos que ha de mantenir el sistema, i una aplicació web per tal d'accedir als informes des de qualsevol ubicació i dispositiu.

Cap de les dues aplicació no utilitza cap base de dades per al seu funcionament. S'emmagatzema el model conceptual que utilitza el sistema per a la definició dels informes corporatius amb arxius xml. Sí que utilitza bases de dades, aquelles que configuri l'usuari, per a obtenir la informació que utilitzarà l'aplicació per tal de generar els informes resultants.

Tot i que el sistema permet treballar amb qualsevol origen de dades que es pugui accedir via JDBC, s'adjunta un arxiu ACCESS, en la carpeta PFCCBZ, anomenat 'jproves.mdb', amb un joc de proves per tal de testejar el funcionament de les aplicacions. D'aquesta forma, el sistema ja porta definits entitats e informes amb aquest origen de dades per tal de testejar-lo.

Per tal que funcioni aquest joc de proves, només s'ha de definir un ODBC amb el nom 'jproves' que apunti a aquest arxiu, o bé posar-li el nom que es vulgui i canviar la definició del origen de dades en la aplicació d'escriptori.

Per altra banda, es pot crear el joc de proves que es consideri més adient per testejar el sistema. Per tal de fer-ho, s'hauria de definir, a partir de l'aplicació d'escriptori que s'entrega, un origen de dades vàlid, una o dos entitats a partir d'aquest origen i un informe que utilitzés les entitats. Ja hi ha usuaris i rols definits en el sistema que es poden utilitzar.

En aquesta entrega, en les llibreries del sistema, hi ha drivers jdbc per connectar amb bases de dades MySQL, Progress i SQL SERVER per fer proves. També és pot connectar via el pont JDBC-ODBC.

Si es requereixen més drivers jdbc, es poden posar en la carpeta de les llibreries (/PFCCBZ/dist/lib) i l'executable de la aplicació d'escriptori les carregarà al iniciar-se. En la aplicació web, aquestes llibreries s'hauran de posar en la carpeta desplegada (WEB-INF/lib) a partir del war que s'adjunta, i reiniciar el servidor web.

S'adjunta un arxiu comprimit amb el contingut de la pràctica.

Es tracta de la carpeta **PFCCBZ** que conté els següents directoris:

- **SGIC**: aplicació d'escriptori que implementa les funcionalitat d'administració i definició dels orígens de dades, diccionari de dades e informes corporatius, així com el manteniment de usuaris i permisos sobre els informes.
- **Web**: aplicació web que permet la visualització dels informes.

### **SGIC**

Està compost per els següents directoris:

- **src**: amb el codi font de l'aplicació
- **xml**: on es troben els fitxers xml que defineixen el model conceptual dels informes.
- **dist**: on trobarem el directori **lib** amb les llibreries que s'utilitzen i l'arxiu **sgic.jar** generat i el directori **bin** on hi ha l'executable de l'aplicació anomenat **SGIC.bat**.

Per tal que l'arxiu executable funcioni, la carpeta **PFCCBZ** s'ha de descomprimir a l'arrel de la unitat C. D'altra forma, s'haurà de canviar l'script d'execució de l'aplicació.

### **Web**

Està compost per els següents directoris:

- **src**: amb el codi font dels servlets.
- **web**: amb les pàgines JSP i els arxius de configuració de l'aplicació web.
- **dist**: on es troba l'arxiu **Web.war** per al desplegament de l'aplicació.

Per tal de desplegar aquesta aplicació, haurem de copiar la l'arxiu **Web.war** en la carpeta del servidor web corresponent i reiniciar-lo per que desplegui el seu contingut.

Per tal que funcioni l'aplicació web, espera trobar els arxius de configuració dels informes corporatius en la carpeta **c:/PFCCBZ/SGIC/xml**,



Amb les dues aplicacions s'adjunten els arxius de construcció 'build.xml' per tal d'empaquetar i generar els executables a partir de possibles canvis en el codi font o llibreries del sistema.

