

---

**CubeExplorer**

**Sistema Gestor de Cubs OLAP .NET**

Xavier Sànchez Perxachs  
**Enginyeria Informàtica**

**Consultor: Jordi Ceballos Villach**

Gener de 2012

*Dedico aquest projecte a la Mercè.*

*Per tot el temps que no hem pogut estar junts.*

*Gràcies pels teus ànims, la teva comprensió i paciència.*

## Índex

1. Introducció.....	7
1.1. Justificació i context del projecte .....	7
1.2. Descripció del projecte .....	8
1.3. Objectius.....	9
1.4. Planificació del projecte i calendari.....	10
1.4.1. Temporització del projecte .....	10
1.4.2. Calendari.....	11
1.4.3. Dates clau .....	11
1.4.5. Eines utilitzades .....	12
1.4.6. Riscos del projecte.....	12
2. Requeriments inicials .....	13
2.1. Perfils d'usuari.....	13
2.2. Requeriments funcionals .....	13
2.2.1. Requeriments de seguretat.....	13
2.2.2. Requeriments de explorador.....	14
2.3. Requeriments no funcionals .....	15
2.3.1. Requeriments d'interfície .....	15
2.3.2. Requeriments de seguretat.....	15
2.3.3. Requeriments de dades .....	15
3. Anàlisi del sistema.....	16
3.1. Diagrama de casos d'ús .....	16
3.2. Descripció textual de casos d'ús.....	18
3.2.1. CU01 – Inici de sessió .....	18
3.2.10. CC05 – Eliminació de cubs .....	22
3.2.11. GU01 – Gestió d'usuaris .....	23
3.2.12. GU02 – Creació d'usuaris .....	23
3.2.13. GU03 – Modificació d'usuaris .....	24
3.2.14. GU04 – Eliminació d'usuaris .....	24
3.2.15. GU05 – Configuració aplicació.....	25
3.2.16. GU06– Creació de seccions.....	25
3.2.17. GU07 – Modificar seccions.....	26
3.2.18. GU08 – Eliminar seccions.....	26
3.2.2. CU02 – Finalitzar sessió.....	19
3.2.3. CU03 – Cerca de cubs .....	19
3.2.4. CU04 – Visualitzar cub .....	19
3.2.5. CU05 – Modificar cub.....	20
3.2.6. CC01 – Creació de cubs .....	20
3.2.7. CC02 – Connexió a base de dades.....	21
3.2.8. CC02 – Selecció origen de dades .....	21
3.2.9. CC04 – Definició de cubs.....	22
3.3. Entitats del domini.....	27
4. Disseny .....	28
4.1. Arquitectura global .....	28
4.1.1. Vista física .....	28
4.1.2. Vista lògica .....	29
4.1.3. Vista de components.....	30
4.2. Decisions tecnològiques .....	30
4.3. Components externs .....	32

---

4.4. Diagrama estàtic de disseny .....	33
4.5. Diagrama de seqüència.....	34
4.6. Disseny persistència .....	35
4.6.1. Model relacional de la base de dades .....	35
4.6.2. Diagrama base de dades .....	36
4.6.3. Format de fitxer .....	36
4.7. Prototipus interfície d'usuari .....	36
5. Implementació .....	37
5.1. Arquitectura de la solució.....	37
5.2. Estructura.....	37
5.2.1. Capa de presentació .....	37
5.2.2 Lògica de negoci.....	38
5.2.3. Persistència .....	38
5.3. Patrons de disseny .....	39
5.4. Components externs utilitzats.....	40
5.4.1. Components DevExpress.....	40
5.4.2. Components d'accés a dades .....	41
5.5. Accés al catàleg del sistema .....	42
5.5.1. Microsoft SQL Server .....	42
5.5.2. Microsoft Access .....	42
5.6. Arxiu de configuració.....	42
5.7. Captures de pantalla.....	43
5.7.1. Pantalla inicial .....	43
5.7.10. Modificació del cub .....	49
5.7.11. Imprimir/Exportar.....	50
5.7.12. Guardar el cub .....	50
5.7.13. Explorador de cubs / seccions .....	51
5.7.14. Eliminar un cub .....	51
5.7.15. Crear una nova secció.....	52
5.7.16. Eliminar seccions.....	52
5.7.17. Gestió d'usuaris .....	53
5.7.18. Creació nou usuari .....	54
5.7.19. Eliminació d'un usuari.....	54
5.7.2. Validació usuari .....	44
5.7.20. Usuari amb permisos restringits .....	55
5.7.3. Pantalla principal .....	45
5.7.4. Cub Nou. Orígens de dades. ....	46
5.7.5. Sel.leccionar proveïdor base de dades.....	46
5.7.6. Paràmetres de connexió.....	47
5.7.7. Nom origen de dades .....	48
5.7.8. Selecció de dades.....	48
5.7.9. Seleccionar camps .....	49
6. Millores futures .....	56
6.1. Funcionalitat .....	56
6.2. Seguretat .....	56
6.3. Interfície d'usuari .....	56
7. Conclusions .....	57
8. Bibliografia.....	58

## Índex de figures

Figura 1 - Temporització del projecte .....	10
Figura 2 - Calendari .....	11
Figura 3 - Casos ús principal .....	16
Figura 4 - Casos ús creació cubs .....	16
Figura 5 - Casos ús configuració .....	17
Figura 6 - Entitats del domini .....	27
Figura 7 - Vista física .....	28
Figura 8 - Vista lògica .....	29
Figura 9 - Vista de components .....	30
Figura 10 - Diagrama de classes .....	33
Figura 11 - Diagrama de seqüència .....	34
Figura 12 - Correspondència components .....	35
Figura 13 - Diagrama base de dades .....	36
Figura 14 - Exemple codi "LlistarCubs" .....	38
Figura 15 - Exemple codi "ValidarUsuari" .....	39
Figura 16 - Exemple codi "UsuariObrir" .....	39
Figura 17 - Exemple codi "Connectar" .....	41
Figura 18 - Exemple codi "Catàleg Access" .....	42
Figura 19 - Arxiu de configuració .....	42
Figura 20 - Pantalla Inicial .....	43
Figura 21 - Validació usuari .....	44
Figura 22 - Pantalla principal .....	45
Figura 23 - Orígens de dades .....	46
Figura 24 - Proveïdor de dades .....	46
Figura 25 - Paràmetres de connexió .....	47
Figura 26 - Nom Orígen de dades .....	48
Figura 27 - Selecció de dades .....	48
Figura 28 - Selecció de camps .....	49
Figura 29 - Modificació del cub .....	49
Figura 30 - Exportació del cub .....	50
Figura 31 - Guardar el cub .....	50
Figura 32 - Explorador de cubs / seccions .....	51
Figura 33 - Missatge eliminar cub .....	51
Figura 34 - Crear nova secció .....	52
Figura 35 - Eliminar seccions .....	52
Figura 36 - Missatge eliminar seccions .....	52
Figura 37 - Gestió d'usuaris .....	53
Figura 38 - Creació / modificació usuaris .....	54
Figura 39 - Eliminació usuari .....	54
Figura 40 -Usuari sense permisos .....	55

## **Resum**

Els sistemes OLAP són solucions molt utilitzades actualment en el camp de la intel·ligència empresarial (Business Intelligence).

Un cub OLAP és una estructura multidimensional que conté dades resumides de bases de dades empresarials.

El cub OLAP ens ofereix una gran rapidesa en el temps de resposta i permet tenir un accés instantani a dades de gran rellevància per el funcionament de la empresa.

Sovint els sistemes que construeixen aquest tipus de solucions son massa complexos, difícils de instal·lar i configurar o massa costosos.

El present treball de Fi de Carrera ofereix una solució informàtica que facilita la implementació d'aquest tipus de sistemes, presentant una aplicació molt configurable que pot generar cubs de diferents sistemes gestors de bases de dades i que per tant és pot adaptar a diferents entorns empresarials.

L'aplicació s'ha desenvolupat en plataforma .NET que és un entorn de programació molt utilitzat amb una gran comunitat de desenvolupadors i de gran implantació a nivell mundial.

Dels diferents llenguatges de programació suportats per aquesta plataforma he escollit C# perquè crec que és el llenguatge estrella d'aquesta plataforma. És molt potent, totalment orientat a objectes i de sintaxi molt similar a Java.

La interfície gràfica de la aplicació s'ha desenvolupat per WPF (Windows Presentation Foundation), aquesta tecnologia permet realitzar aplicacions amb una interfície d'usuari enriquida més atractiva i molt més personalitzable que les clàssiques interfícies de windows.

Per implementar aquestes interfícies i per el desenvolupament del control del cub OLAP hem utilitzat els controls DevExpress que són molt potents i redueixen considerablement el temps de desenvolupament.

Aquestes llibreries de components són àmpliament utilitzades per la comunitat de desenvolupadors de .NET.

Espero que la solució presentada agradi i sigui de utilitat.

## 1. Introducció

### 1.1. Justificació i context del projecte

En el context de la assignatura PFC he escollit l'àrea de desenvolupament en .NET perquè m'interessava millorar els coneixements d'aquesta tecnologia i desenvolupar un cas de projecte real.

Els sistemes OLAP els coneixo tant de la assignatura de magatzems de dades i models multidimensionals com de la meva experiència laboral en diferents empreses.

En l'àmbit laboral sovint he observat que aquests sistemes van molt lligats a aplicacions ERP on el sistema OLAP sovint s'inclou com una part del propi ERP del fabricant i no es pot adaptar a d'altres entorns.

D'altra banda les solucions com Microsoft Analysis Services són difícils de desenvolupar i configurar i van molt lligades a la base de dades del fabricant (Microsoft SQL Server).

No sempre és fàcil trobar aplicacions que resolguin aquesta problemàtica, per aquest motiu he dissenyat i desenvolupat aquest sistema.

Existeixen en el mercat tres tipus de sistemes OLAP:

- Sistemes ROLAP: Implementació que emmagatzema les dades en un motor de dades relacional. Va més adreçat a bases de dades de petit tamany.
- Sistemes MOLAP: Implementació que emmagatzema les dades en un motor de dades multidimensional. Va dirigit a solucions per grans sistemes.
- Sistemes HOLAP: Implementació híbrida. Una part s'emmagatzema en el sistema relacional i un altre en un sistema multidimensional.

D'aquest tres tipus la solució construïda es basa en els sistemes relacionals (ROLAP) per ser els d'ús més habitual.

## *1.2. Descripció del projecte*

Aquesta aplicació s'engloba en l'àrea coneguda com BI (business intelligence) que correspon al conjunt d'estratègies i eines destinades a crear coneixement mitjançant l'anàlisi de dades existents a l'organització o empresa.

L'objectiu d'aquest projecte és dissenyar i implementar una aplicació que permeti facilitar la gestió de cubs OLAP.

Un cub és una estructura multidimensional que permet agrupar les dades que provenen d'una base de dades relacional de diferents formes de forma interactiva.

Aquesta estructura permet un anàlisi flexible i ràpid d'un volum de dades moltes vegades considerable.

Per tant dues característiques que tenen aquests sistemes és que son flexibles i dinàmics.

El programa ha de permetre mantenir un historial dels cubs que es van creant i sobre aquests cubs permetre realitzar una visualització del cub, modificació del cub i seriació a disc de les seves dades.

L'aplicació ha de permetre connectar a fonts de dades diverses principalment Microsoft Access i SQL Server, tot i que seria desitjable oferir la possibilitat de connexió a altres SGBD a través dels proveïdors d'accés a dades de la tecnologia .NET.

Per automatitzar la creació del cub s'inclou una gestió d'orígens de dades que permet la selecció d'una cadena de connexió prèviament utilitzada.

Els cubs del magatzem s'han de poder agrupar en diferents seccions, per exemple: vendes, compres, comptabilitat, producció. Aquestes seccions poden correspondre típicament a les àrees de qualsevol empresa.

És necessari crear una gestió d'usuaris ja que té molt sentit poder definir permisos que permetin accedir o restringir l'accés a les diferents seccions. A més no tots els usuaris volen ni han de poder crear cubs nous donat que és un procés relativament complex.

Com a funcionalitats interessants que es podrien incloure en un futur podem destacar: mostrar les dades de forma gràfica a través de gràfics estadístics, possibilitats d'incloure camps calculats, internacionalització, personalització interfície usuari, etc. Aquests aspectes només seran abordats com a possibles millores en cas de disposar de temps suficient.

Amb aquestes característiques s'obté una aplicació que permetrà definir un magatzem de dades corporatiu i heterogeni ja que pot treballar amb diverses bases de dades.



L'aplicació va dirigida fonamentalment a dos perfils d'usuaris:

- 1) Administratius i executius amb poc temps que desitgen tenir un accés personalitzat ràpid a les dades més rellevants del seu negoci. Aquestes persones bàsicament consulten cubs ja creats i les operacions que realitzen són les de filtratge de dades i explotació de dades a través d'exportació a altres formats i impressió.
- 2) Personal tècnic. És una persona coneixedora de l'aplicació, de les bases de dades de l'empresa, que és l'encarregada de crear nous cubs segons uns requeriments previs.

### ***1.3. Objectius***

A grans trets els objectius d'aquest projecte són els següents:

- Posar en pràctica coneixements adquirits al llarg de la carrera.
- Enfrontar-se a una aplicació d'una certa envergadura que resolgui una problemàtica real.
- Millorar els coneixements i la capacitat per gestionar un projecte.
- Aprofundir els coneixements de la tecnologia .NET.
- Aprendre a treballar amb diferents eines com Visual Studio, SQL Server, DevExpress.

#### 1.4. Planificació del projecte i calendari

Inicialment s'ha fixat un pla de treball i un calendari d'entregues.

Cada entrega consisteix en un lliurament on cal entregar un document, o el codi de l'aplicació.

##### 1.4.1. Temporització del projecte

L'estimació de tasques i temps de dedicació en cada tasca va ser la següent:

<b>Pla de Treball</b>	<b>13 dies</b>	<b>22/09/2011</b>	<b>03/10/2011</b>
Lectura Documentació TFC	1 dia	22/09/2011	22/09/2011
Elecció del projecte	4 dies	23/09/2011	26/09/2011
Instal·lació programari	1 dia	26/09/2011	26/09/2011
Proves programari	2 dies	27/09/2011	28/09/2011
Definició del pla de treball	6 dies	28/09/2011	03/10/2011
Entrega PAC1 (3/10/2011)	Fita	03/10/2011	
<b>Anàlisi, Disseny i Prototip IU</b>	<b>28 dies</b>	<b>04/10/2011</b>	<b>31/10/2011</b>
Anàlisi Funcional	5 dies	04/10/2011	08/10/2011
Estudi de tecnologies a utilitzar	3 dies	04/10/2011	06/10/2011
Selecció de components a utilitzar	2 dies	07/10/2011	08/10/2011
Disseny tècnic	8 dies	11/10/2011	18/10/2011
Disseny accés a les dades	3 dies	11/10/2011	13/10/2011
Disseny gestió del cub	5 dies	14/10/2011	18/10/2011
Prototip interfície d'usuari	12 dies	19/10/2011	31/10/2011
Interfície Usuari principal	3 dies	19/10/2011	21/10/2011
Interfície Usuari creació cub	3 dies	24/10/2011	26/10/2011
Interfície Usuari visualització cub	5 dies	27/10/2011	31/10/2011
Entrega PAC2 (31/10/2011)	Fita	31/10/2011	
<b>Implementació</b>	<b>50 dies</b>	<b>01/11/2011</b>	<b>19/12/2011</b>
Interfície d'usuari	14 dies	01/11/2011	14/11/2011
Component de Gestió de l'accés dades	11 dies	19/11/2011	29/11/2011
Component de Gestió del Cub	7 dies	30/11/2011	06/12/2011
Integració de components	8 dies	07/12/2011	14/12/2011
Exemples i proves	5 dies	15/12/2011	19/12/2011
Entrega PAC3 (19/12/2011)	Fita	19/12/2011	
<b>Final: Memòria i presentació</b>	<b>24 dies</b>	<b>20/12/2011</b>	<b>26/01/2012</b>
Elaboració memòria	13 dies	20/12/2011	02/01/2012
Elaboració presentació	7 dies	03/01/2012	09/01/2012
Entrega Final (09/01/2012)	0 dies	09/01/2012	09/01/2012
Debat virtual	4 dies	23/01/2012	26/01/2012
Final del projecte	Fita	26/01/2012	

Figura 1 - Temporització del projecte

### 1.4.2. Calendari

A continuació es mostra el calendari del projecte en Microsoft Project.

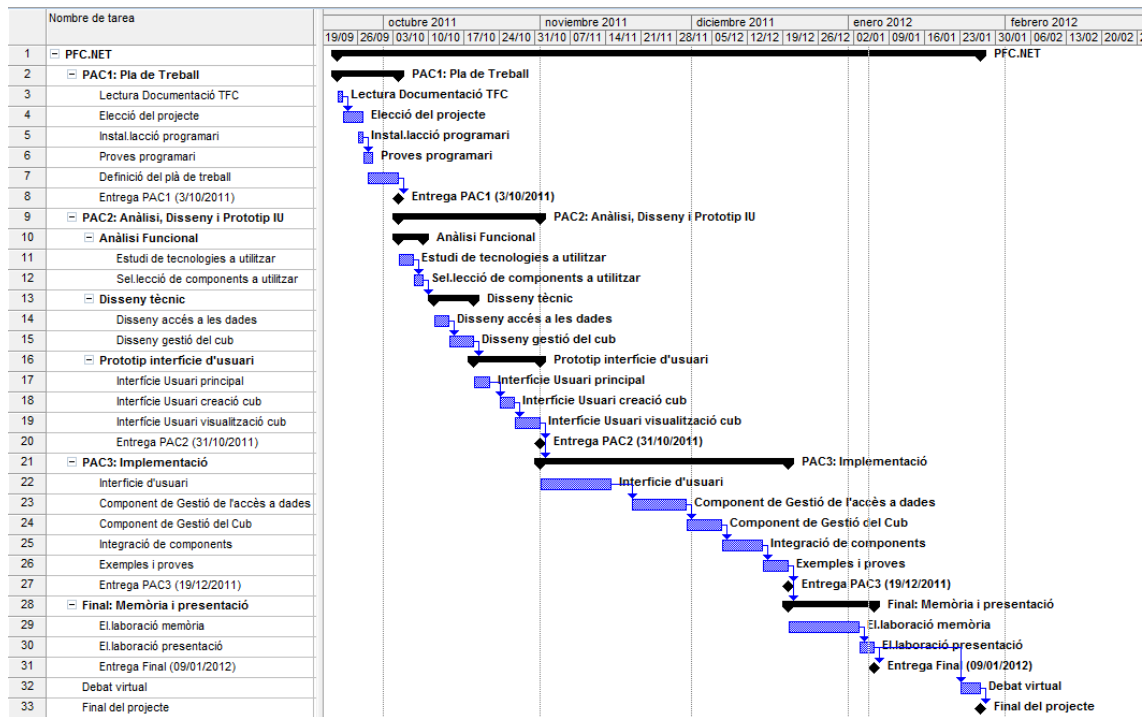


Figura 2 - Calendari

### 1.4.3. Dates clau

La planificació ve ajustada segons el calendari de la assignatura per les següents dates:

**PAC1.** Elaboració del pla de treball. Del 22/09/11 al 03/10/11.

**PAC2.** Anàlisi, Disseny i prototip interfície. Del 04/10/11 al 31/10/11.

**PAC3.** Implementació de l'aplicació. De 01/11/11 al 19/12/11.

**Lliurament final.** Memòria i presentació. Del 20/12/11 al 09/01/11.

**Debat.** Defensa del projecte. Del 23/01/11 al 26/01/11.

#### ***1.4.5. Eines utilitzades***

- PC amb Intel Core2 Duo 2.40 GHZ, 4 Gb RAM.
- Sistema operatiu Windows 7
- Microsoft Visual Studio 2010 Ultimate
- Llenguatge de programació C#
- Microsoft .NET Framework 4.0
- Microsoft SQL Server 2008 R2 Express
- Microsoft Access 2007
- Microsoft Word 2007
- Microsoft Project 2003
- Microsoft Visio 2003
- DevExpress versió evaluació per WPF.

#### ***1.4.6. Riscos del projecte***

Els associats a un projecte d'aquestes característiques. Provenen del nivell de complexitat del producte que es desitja construir, de la manca de temps i de un nivell de coneixements previ en les tecnologies a utilitzar insuficient.

- 1) Desconeixement de la tecnologia utilitzada. Pot provocar lentitud al començament per necessitat d'aprenentatge de les tecnologies utilitzades amb la conseqüent necessitat d'ajustament del calendari.
- 2) Previsions de calendari massa optimistes. Cal reajustar el calendari i suprimir funcionalitats.
- 3) Possible baix rendiment d'algun component: especialment el cub, cal tenir en compte que estem fent servir una versió d'avaluació d'aquest component.
- 4) Dificultats en la implementació. La majoria de tecnologies a utilitzar son suficientment madures i provades: C#, ADO.NET, WPF, etc. Es considera el component més delicat el cub per tractar-se d'un component extern no estàndard de Microsoft. Per aquest motiu resulta molt important escollir-ne un d'adient que ens ofereixi les garanties suficients.

## **2. Requeriments inicials**

L'aplicació a desenvolupar consisteix en una eina de desenvolupament i gestió de cubs OLAP.

En principi ens centrarem en l'ús de bases de dades relacionals perquè són les més habituals en l'àmbit empresarial, tot i que el control que hem utilitzat pel cub preveu altres orígens de dades com fitxers XML o bases de dades multidimensionals, aquest darrer cas està especialment indicat per implementar aplicacions OLAP donat que el seu disseny permet el processament de grans quantitats de dades a una gran velocitat.

L'aplicació ha de permetre en teoria connectar amb qualsevol base de dades del mercat que disposi d'un proveïdor .NET, o OLE DB. Inicialment només oferirem connectivitat a les bases de dades Access i SQL Server però implementant les funcions de lectura del catàleg del sistema ha de ser possible l'accés a d'altres bases de dades com Oracle, MySQL, etc.

La creació de nous cubs i exploració de cubs existents ha de ser molt simple pensat per un usuari no massa expert. El control del cub ha de permetre modificar les dades que es processen, els camps que es tracten com a files, columnes i dades i d'altres aspectes de processament del cub i presentació. El refresc dels canvis en el cub s'ha de fer en temps real.

### **2.1. Perfils d'usuari**

- Usuari administrador: pot accedir a tota la funcionalitat de l'aplicació. S'encarrega de gestionar la seguretat dels usuaris.
- Usuari desenvolupador: pot accedir a tota la funcionalitat de l'aplicació tret de la gestió d'usuaris. S'encarrega de crear cubs nous.
- Usuari de consulta: pot consultar cubs ja creats i modificar-los però no en pot crear de nous.

### **2.2. Requeriments funcionals**

#### **2.2.1. Requeriments de seguretat**

Inclouen la gestió dels usuaris que poden accedir a l'aplicació. Els usuaris només poden ser gestionats per un usuari amb permisos d'administrador.

##### Iniciar sessió

L'usuari introdueix el nom i la clau, si aquests són correctes es carreguen els permisos segons el nivell de seguretat que té assignat i es configura la interfície de l'aplicació.

##### Finalitzar sessió

Es desconnecta la sessió oberta i es tanca la interfície de la aplicació.

##### Llistar usuaris

Es llisten tots els usuaris actuals de l'aplicació.

##### Alta d'usuari

Es dona d'alta un nou usuari de l'aplicació.

### Baixa d'usuari

Es dona de baixa un usuari. No es pot donar de baixa l'usuari amb que ens hem validat al sistema.

### Modificació d'usuari

Es modifica les dades de connexió d'un usuari.

## **2.2.2. Requeriments de explorador**

### Gestió de seccions

Es permet crear noves seccions i eliminar-ne. Conceptualment una secció és un nivell d'agrupació de cubs que representa una àrea de la empresa: comptabilitat, vendes, etc. A cada secció li correspon una carpeta on s'emmagatzemen els cubs.

### Explorador de cubs

Els cubs estaran emmagatzemats en carpetes. L'aplicació podrà explorar aquestes carpetes i mostrar un llistat dels cubs que s'han creat i les seves propietats: data de creació, tipus de cub, notes, usuari, etc.

### Connexió amb bases de dades a través de proveïdors .NET

La connexió s'efectua a través de llibreries estàndard de Microsoft. S'utilitzen proveïdors d'accés a dades OLE DB i .NET que permeten accedir a els SGBD més habituals del mercat.

El diàleg de connexió és anàleg al que podem trobar a d'altres aplicacions d'explotació de dades com (Office, Crystal Reports, etc).

### Selecció de dades (taula, vista, consulta, procediment emmagatzemat)

Es crearà un formulari que permetrà visualitzar totes les taules, consultes i procediments emmagatzemats disponibles a la base de dades. Aquesta informació s'obindrà mitjançant consultes al catàleg del sistema. Donat les diferències entre els diferents SGBD inicialment només s'implementarà l'accés per utilitzar Microsoft Access i SQL Server. Cal poder seleccionar un element que ens proporcioni les dades necessàries per construir el cub.

### Construcció de cubs (selecció de files, columnes, camps de dades, tipus d'agrupació)

Es crearà un formulari que ens permetrà escollir els camps de la taula, aquests camps es podran definir com a files, columnes o dades en cas que el camp sigui numèric.

Hem de poder indicar-li el tipus de operació sobre dades numèriques: suma i recompte com a mes usuals.

### Explotació de cubs

Cal poder modificar la visualització del cub, filtrar dades, imprimir les dades del cub, emmagatzemar el cub a disc i carregar-lo de disc com a opcions més bàsiques.

Si el cub és connectat s'ha de poder actualitzar les dades del cub. El refresc de les dades s'efectuarà al carregar el cub i quan l'usuari ho demani prement un botó determinat. En cas de que el cub sigui desconnectat el refresc no serà possible.

### **2.3. *Requeriments no funcionals***

#### **2.3.1. *Requeriments d'interfície***

L'aplicació serà d'escriptori i s'executarà en un únic PC. La interfície serà desenvolupada en WPF (Windows Presentation Foundation).

#### **2.3.2. *Requeriments de seguretat***

La validació del usuari al entrar al programa introduint nom d'usuari i paraula d'accés l'identifica en un grup amb un nivell de permisos.

#### **2.3.3. *Requeriments de dades***

Les dades del explorador s'emmagatzemen en una base de dades en Microsoft SQL Server.

Per poder gestionar els cubs cal poder emmagatzemar les següents dades:

- Identificador: és un camp intern identificador del cub que es genera al donar-lo de alta.
- Nom del cub: és el nom amb que l'usuari identifica el cub.
- Secció: secció a la qual pertany el cub: per exemple: vendes.
- Data creació: data de creació del cub.
- Data modificació: data modificació del cub.
- Tipus: cub desconnectat o cub connectat.
- Ruta: Ruta completa del arxiu que emmagatzema dades del cub i la seva presentació en arxiu propietari del fabricant del cub.
- Connexió: Cadena de connexió d'accés a dades, ens permet reconstruir el cub i refrescar les seves dades.
- Sql: Consulta sql de selecció de dades. Aquesta consulta s'utilitza per permetre el refresc de les dades del cub.
- Notes: Permet emmagatzemar dades explicatives del cub.
- Data creació: Data i hora de creació del cub.
- Data modificació: Data i hora de modificació del cub.
- Usuari: Usuari que va crear el cub.

### 3. Anàlisi del sistema

Aquesta secció desenvolupa l'anàlisi del sistema.

#### 3.1. Diagrama de casos d'ús

El següent diagrama recull una vista dels actors i els casos que descriuen els requeriments funcionals de l'aplicació.

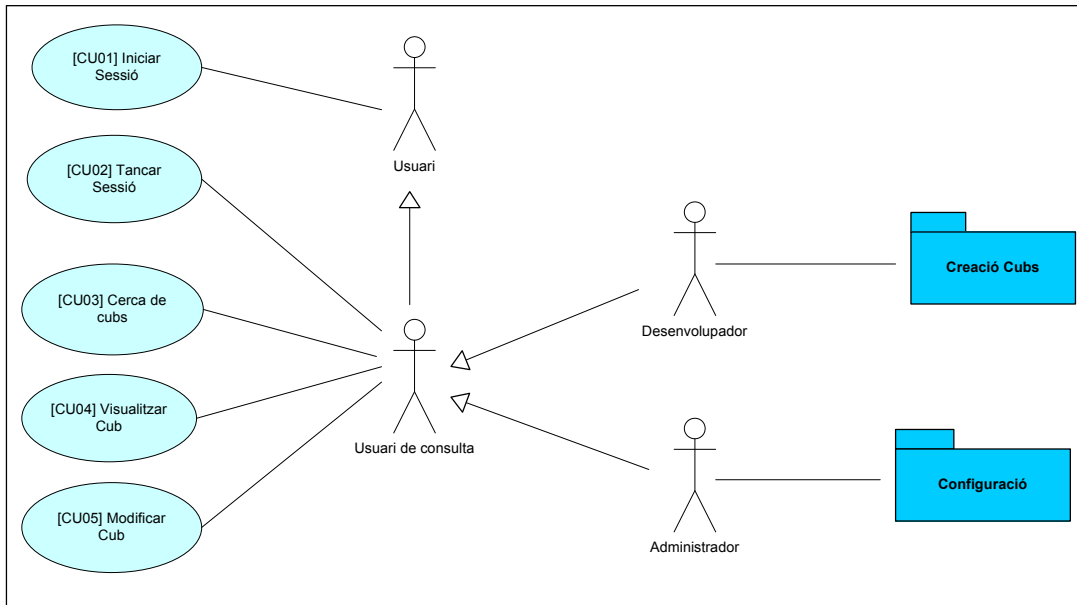


Figura 3 - Casos ús principal

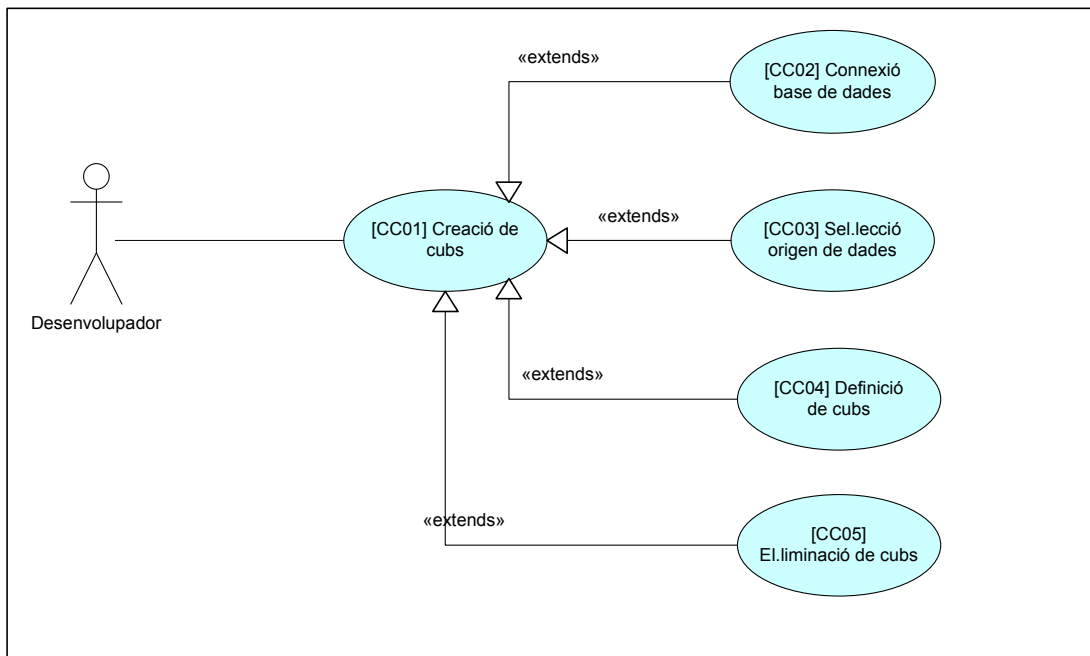


Figura 4 - Casos ús creació cubs



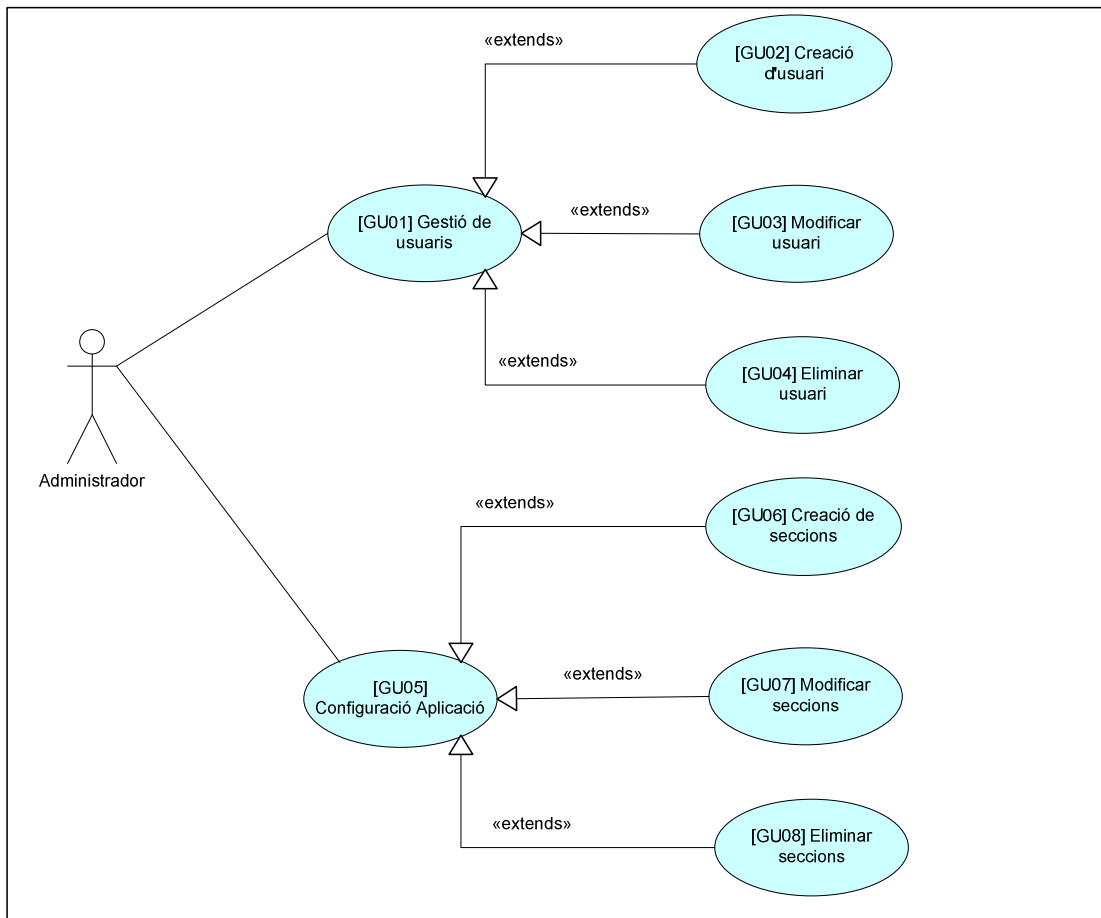


Figura 5 - Casos ús configuració

Com es pot comprovar hi ha tres tipus diferents d'actors:

- Usuari de consulta: És el nivell bàsic d'accés a l'aplicació, pot consultar els cubs emmagatzemats al sistema però no pot crear cubs nous.
- Usuari desenvolupador: Pot consultar cubs i crear-ne de nous.
- Usuari administrador: A més de consultar i crear cubs pot administrar els usuaris de l'aplicació i configurar altres aspectes de l'aplicació.

La següent taula resumeix els casos d'ús definits:

<b>Codi</b>	<b>Descripció</b>	<b>Actor</b>
CU01	Inici de sessió	Usuari
CU02	Tancar sessió	Usuari de consulta
CU03	Cerca de cubs	Usuari de consulta
CU04	Visualitzar cub	Usuari de consulta
CU05	Modificar cub	Usuari de consulta
CC01	Creació de cubs	Desenvolupador
CC02	Connexió bases de dades	Desenvolupador
CC03	Selecció origen de dades	Desenvolupador
CC04	Definició de cubs	Desenvolupador
CC05	Eliminació de cubs	Desenvolupador
GU01	Gestió d'usuaris	Administrador
GU02	Creació d'usuaris	Administrador
GU03	Modificar usuaris	Administrador
GU04	Eliminar usuaris	Administrador
GU05	Configuració aplicació	Administrador
GU06	Crear seccions	Administrador
GU07	Modificar seccions	Administrador
GU08	Eliminar seccions	Administrador

### 3.2. Descripció textual de casos d'ús

#### 3.2.1. CU01 – Inici de sessió

<b>Identificador</b>	CU01
<b>Nom</b>	Inici de sessió
<b>Resum</b>	Inicia la sessió al sistema
<b>Actor</b>	Usuari
<b>Precondicions</b>	No hi ha cap sessió activa
<b>Postcondicions</b>	El usuari ha iniciat la sessió
<b>Accions</b>	<p>El usuari accedeix a l'aplicació.</p> <p>El sistema sol·licita el nom d'usuari i la clau d'accés.</p> <p>El usuari introdueix el nom d'usuari i la clau d'accés. Prem el botó de connexió.</p> <p>El sistema valida les dades introduïdes si són correctes la sessió queda iniciada, si no son correctes es mostra un missatge.</p>
<b>Excepcions</b>	La validació no ha estat correcta

### 3.2.2. CU02 – Finalitzar sessió

<b>Identificador</b>	CU02
<b>Nom</b>	Finalitzar sessió
<b>Resum</b>	Finalitza la sessió del sistema
<b>Actor</b>	Usuari de consulta
<b>Precondicions</b>	El usuari té una sessió iniciada
<b>Postcondicions</b>	La sessió està tancada
<b>Accions</b>	El usuari prem el botó de desconnexió. El sistema finalitza la sessió i torna al cas CU01.
<b>Excepcions</b>	Cap

### 3.2.3. CU03 – Cerca de cubs

<b>Identificador</b>	CU03
<b>Nom</b>	Cerca de cubs
<b>Resum</b>	Cerca de cub sobre l'explorador de cubs
<b>Actor</b>	Usuari de consulta
<b>Precondicions</b>	El usuari té una sessió iniciada
<b>Postcondicions</b>	S'ha trobat el cub cercat
<b>Accions</b>	El usuari cerca manualment sobre el explorador el cub que vol seleccionar. El usuari cerca utilitzant la cerca automàtica i prem el botó de cerca, el sistema cerca el cub pel seu nom.
<b>Excepcions</b>	La cerca no troba cap coincidència

### 3.2.4. CU04 – Visualitzar cub

<b>Identificador</b>	CU04
<b>Nom</b>	Visualitzar cub
<b>Resum</b>	Visualització de les dades del cub a partir d'una determinada configuració
<b>Actor</b>	Usuari de consulta
<b>Precondicions</b>	El usuari ha cercat un cub
<b>Postcondicions</b>	Cap
<b>Accions</b>	El usuari prem el botó de visualitzar sobre un cub seleccionat. El sistema llegeix el cub de disc i carrega les seves dades i la seva presentació. El control carregat es mostra en pantalla.
<b>Excepcions</b>	Cap

### 3.2.5. CU05 – Modificar cub

<b>Identificador</b>	CU05
<b>Nom</b>	Modificar cub
<b>Resum</b>	Modificació del cub ja creat
<b>Actor</b>	Usuari de consulta
<b>Precondicions</b>	El usuari ha cercat un cub
<b>Postcondicions</b>	Cap
<b>Accions</b>	El usuari prem el botó de modificar sobre un cub seleccionat. El sistema llegeix el cub de disc i carrega les seves dades i la seva presentació. El control carregat es mostra en pantalla. L'usuari pot modificar el disseny i la presentació del cub i emmagatzemar els canvis.
<b>Excepcions</b>	Cap

### 3.2.6. CC01 – Creació de cubs

<b>Identificador</b>	CC01
<b>Nom</b>	Creació de cubs
<b>Resum</b>	Creació d'un cub nou
<b>Actor</b>	Usuari desenvolupador
<b>Precondicions</b>	L'usuari ha de saber la base de dades que vol connectar i com fer-ho. L'usuari ha de saber quina taula, consulta o procediment emmagatzemat s'ha de seleccionar. L'usuari ha de saber com dissenyar el cub i les dades que necessita.
<b>Postcondicions</b>	Cap
<b>Accions</b>	Aquest cas inclou tres subcasos: CC02 - Connexió a la base de dades CC03 - Selecció del origen de dades CC04 - Definició del cub
<b>Excepcions</b>	

### 3.2.7. CC02 – Connexió a base de dades

<b>Identificador</b>	CC02
<b>Nom</b>	Connexió a base de dades
<b>Resum</b>	Connecta a una base de dades
<b>Actor</b>	Usuari desenvolupador
<b>Precondicions</b>	És necessari saber la base de dades que volem connectar i disposar de les dades de connexió necessàries. Cal tenir un proveïdor d'accés a la BD instal·lat a la màquina. Aquest proveïdor pot ser en tecnologia .NET, OLE DB, o ODBC.
<b>Postcondicions</b>	Obtenim una cadena de connexió a la base de dades que permet obrir una connexió
<b>Accions</b>	Es mostra un diàleg de connexió bastant comú en moltes aplicacions. Aquest diàleg permet escollir l'origen de dades i el proveïdor d'accés. Un cop seleccionats podem indicar les dades de connexió (nom servidor, base de dades, tipus autenticació, usuari i clau accés).
<b>Excepcions</b>	Si la connexió no es pot dur a terme s'ha de capturar l'excepció i mostrar un missatge informatiu.

### 3.2.8. CC03 – Selecció origen de dades

<b>Identificador</b>	CC03
<b>Nom</b>	Selecció origen de dades
<b>Resum</b>	Selecciona un origen de dades
<b>Actor</b>	Usuari desenvolupador
<b>Precondicions</b>	Cal tenir una connexió oberta contra la base de dades. L'usuari ha de conèixer quines dades necessita i quin element de la base de dades ha de seleccionar.
<b>Postcondicions</b>	Cap
<b>Accions</b>	A partir de la connexió s'obté una llista de taules, vistes (consultes) i procediments emmagatzemats. L'usuari selecciona el element més adient per el cub que vol construir.
<b>Excepcions</b>	Problemes en la obtenció de les llistes. Caldrà capturar aquest error i mostrar missatge informatiu.

### 3.2.9. CC04 – Definició de cubs

<b>Identificador</b>	CC04
<b>Nom</b>	Definició de cubs
<b>Resum</b>	Definició del cub amb l'estadística que es vol obtenir.
<b>Actor</b>	Usuari desenvolupador
<b>Precondicions</b>	Cal tenir una connexió oberta contra la base de dades. Cal haver seleccionat una taula, consulta o procediment emmagatzemat vàlid.
<b>Postcondicions</b>	Cap
<b>Accions</b>	Cal executar una consulta contra la base de dades per tal de obtenir les dades de la consulta. En el cas de l'execució d'un procediment emmagatzemat caldrà demanar els paràmetres del procediment. Un cop obtingudes les dades l'usuari ha de poder seleccionar quins camps de la consulta intervenen en el cub. Es construeix el cub i s'emmagatzema a disc. El cas enllaça amb el cas CU05.
<b>Excepcions</b>	Problemes en l'obtenció de les dades.

### 3.2.10. CC05 – Eliminació de cubs

<b>Identificador</b>	CC05
<b>Nom</b>	Eliminació de cubs
<b>Resum</b>	El cas inclou la eliminació del cub de l'explorador i físicament de disc.
<b>Actor</b>	Usuari desenvolupador
<b>Precondicions</b>	Cal haver seleccionat un cub.
<b>Postcondicions</b>	Cap
<b>Accions</b>	Una vegada l'usuari selecciona el cub l'usuari prem el botó de eliminació. L'aplicació demana confirmació de l'eliminació del cub. L'aplicació borra físicament l'arxiu de disc. L'aplicació elimina el cub del explorador.
<b>Excepcions</b>	Manca de permisos de la carpeta on es vol eliminar el cub. Cal capturar l'error, mostrar missatge i no permetre la eliminació.

### 3.2.11. GU01 – Gestió d'usuaris

<b>Identificador</b>	GU01
<b>Nom</b>	Gestió d'usuaris
<b>Resum</b>	El cas mostra com gestionar els usuaris de l'aplicació.
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	L'aplicació ha de tenir creat un usuari administrador.
<b>Postcondicions</b>	Cap
<b>Accions</b>	El cas comença quan l'usuari prem el botó de gestionar usuaris. Es mostra una llista de tots els usuaris registrats al sistema. Es mostren tres possibles opcions de gestió d'usuaris: <ul style="list-style-type: none"><li>- Crear un usuari nou.</li><li>- Donar de baixa un usuari.</li><li>- Modificar les dades del usuari.</li></ul>
<b>Excepcions</b>	Cap

### 3.2.12. GU02 – Creació d'usuaris

<b>Identificador</b>	GU02
<b>Nom</b>	Creació d'usuaris
<b>Resum</b>	Creació i registre dels usuaris que poden accedir a l'aplicació
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	L'aplicació ha de tenir com a mínim un usuari administrador.
<b>Postcondicions</b>	Cap.
<b>Accions</b>	L'usuari prem el botó de nou. L'aplicació mostra el formulari de creació d'usuaris. Un cop introduïdes les dades el usuari prem el botó de confirmar. Si les dades són correctes és dona d'alta el usuari, es tanca el formulari i es mostren els canvis a la llista d'usuaris.
<b>Excepcions</b>	Validació de les dades incorrecta (usuari duplicat, clau d'accés no especificada, etc). Cal capturar els errors i mostrar un missatge.

### 3.2.13. GU03 – Modificació d'usuari

<b>Identificador</b>	GU03
<b>Nom</b>	Modificació usuari
<b>Resum</b>	Les dades del usuari poden ser modificades.
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	L'usuari ha d'estar creat.
<b>Postcondicions</b>	Cap.
<b>Accions</b>	L'usuari prem el botó de modificar. L'aplicació mostra el formulari de modificació d'usuari. Un cop introduïdes les dades el usuari prem el botó de confirmar. Si les dades són correctes es modifiquen les dades de l'usuari, es tanca el formulari i es mostren els canvis a la llista d'usuari.
<b>Excepcions</b>	Validació de les dades incorrecta (usuari duplicat, clau d'accés no especificada, etc). Cal capturar els errors i mostrar un missatge.

### 3.2.14. GU04 – Eliminació d'usuari

<b>Identificador</b>	GU04
<b>Nom</b>	Eliminació d'usuari
<b>Resum</b>	Els usuari es poden donar de baixa del sistema.
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	Selecció d'un usuari. No es pot eliminar l'usuari amb que ens hem logat.
<b>Postcondicions</b>	Cap
<b>Accions</b>	L'usuari prem el botó de eliminar el usuari. El sistema mostra un missatge de confirmació. Si l'usuari confirma l'operació l'usuari es borra.
<b>Excepcions</b>	Cap.



### 3.2.15. GU05 – Configuració aplicació

<b>Identificador</b>	GU05
<b>Nom</b>	Configurar l'aplicació
<b>Resum</b>	El cas inclou la configuració dels paràmetres de l'aplicació. Principalment les seccions que es mostren a l'explorador i que permeten un nivell d'agrupació de cubs.
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	Cap
<b>Postcondicions</b>	Cap
<b>Accions</b>	Mostrar una llista de les seccions disponibles. L'usuari selecciona una secció i es mostren les dades de la mateixa en un formulari específic. L'usuari disposa de tres opcions: Crear noves seccions (cas GU06). Modificar seccions (cas GU07). Eliminar seccions (cas GU08).
<b>Excepcions</b>	Cap.

### 3.2.16. GU06– Creació de seccions

<b>Identificador</b>	GU06
<b>Nom</b>	Creació de seccions
<b>Resum</b>	Creació d'una nova secció
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	Cap
<b>Postcondicions</b>	Cap
<b>Accions</b>	L'usuari prem el botó de nou. Es mostra el formulari d'introducció de dades. L'usuari prem el botó de confirmar. Si les dades són correctes es mostren els canvis a l'explorador creant una nova secció a la llista de configuració i un nou nivell d'agrupament.
<b>Excepcions</b>	Cal capturar els errors de validació de dades i mostrar el missatge d'error.

### 3.2.17. GU07 – *Modificar seccions*

<b>Identificador</b>	GU07
<b>Nom</b>	Modificar seccions
<b>Resum</b>	Modificar les dades de la secció
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	Cap
<b>Postcondicions</b>	Cap
<b>Accions</b>	L'usuari prem el botó de modificar. Es mostra el formulari de modificació de dades. L'usuari prem el botó de confirmar. Si les dades són correctes es mostren els canvis a l'explorador. Si es modifica el nom de la secció caldrà refrescar el canvi a l'explorador. En cas de modificar la carpeta , si existeixen cubs inclosos a la carpeta antiga cal permetre transferir els arxius a la nova carpeta.
<b>Excepcions</b>	Si hi ha problemes de permisos de carpetes cal cancel·lar la operació i mostrar missatge informatiu.

### 3.2.18. GU08 – *Eliminar seccions*

<b>Identificador</b>	GU08
<b>Nom</b>	Eliminar seccions
<b>Resum</b>	Eliminar una secció
<b>Actor</b>	Usuari administrador
<b>Precondicions</b>	Cal que la secció no tingui cubs inclosos. En cas contrari no es deixarà borrar.
<b>Postcondicions</b>	Cap
<b>Accions</b>	L'usuari prem el botó de borrar la secció. El sistema demana confirmació. Si l'usuari confirma la operació el sistema elimina la secció i refresca el canvi a l'explorador.
<b>Excepcions</b>	Cap.

### 3.3. Entitats del domini

El model del domini representa les principals classes identificades en el problema. Donada que la gestió d'aquesta aplicació és únicament dels cubs que els usuaris poden crear i les seccions on poden classificar aquests cubs el model és realment senzill i es pot representar amb dues classes.

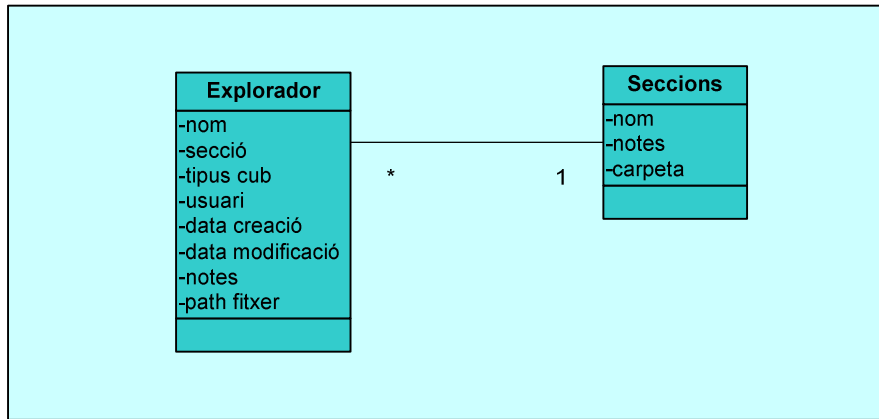


Figura 6 - Entitats del domini

- Explorador: aquesta entitat permet gestionar les dades del explorador de cubs. Cada element conté les dades que permeten gestionar el cub.
- Seccions: conté la definició de les diferents seccions de l'aplicació. Cada secció representa conceptualment una agrupació de cubs en una carpeta de la xarxa o d'un PC. Cada element de l'explorador conté obligatòriament un enllaç a una secció. L'aplicació necessita tenir una secció creada per poder funcionar.

## 4. Disseny

### 4.1. Arquitectura global

Definim tres tipus diferents de vistes:

- Vista Física: Ens permet veure els dispositius hardware i l'esquema de la xarxa sobre la qual s'executa l'aplicació.
- Vista lògica: Permet veure les diferents capes de l'aplicació.
- Vista de components: Permet visualitzar els diferents components de l'aplicació.

#### 4.1.1. Vista física

L'aplicació es pot executar en un únic PC o es pot instal·lar en una xarxa corporativa. El diagrama següent mostra una possible instal·lació en una xarxa.

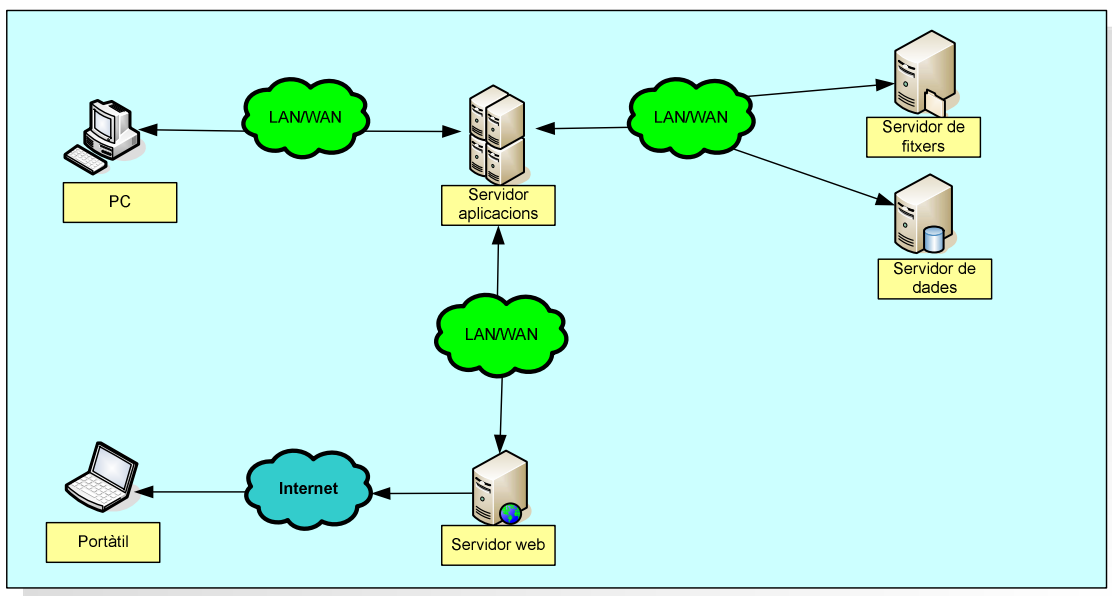


Figura 7 - Vista física

Consta dels següents elements:

- **Servidor de dades:** Conté les dades de l'aplicació en un servidor SQL Server, dependent de la configuració de la xarxa podrem accedir a diferents servidors per crear els orígens de dades que ens permetran crear els cubs.
- **Servidor de fitxers:** Conté les carpetes i els cubs que es van creant.
- **Servidor aplicacions :** Pot contenir l'aplicació instal·lada en una carpeta i tenir activats els serveis IIS (Internet Information Services) per permetre tenir sempre actualitzats els PC del clients a través de la tecnologia "ClickOnce".
- **Equip PC:** L'aplicació s'autoinstal·la i executa en un PC a través d'un accés directe al servidor d'aplicacions. L'equip client necessita tenir instal·lar el Framework 4 de Microsoft.

- **Servidor web:** Té l'aplicació instal·lada i els serveis IIS i permet un accés extern a través de internet. Utilitzant la tecnologia "ClickOnce" els equips remots poden descarregar i instal·lar l'aplicació.

- **Portàtil:** Són equips remots que poden executar l'aplicació un cop descarregada. Per definir cubs es pot accedir a servidors dins la pròpia LAN o a través d'una WAN si l'arquitectura de la xarxa ho permet, per exemple a través de connexions VPN.

#### 4.1.2. Vista lògica

L'aplicació s'estructura en capes perquè aquest disseny ofereix molts avantatges com més claredat, més solidesa degut a un millor nivell d'encapsulament i més flexibilitat. A continuació es mostra un diagrama de més a menys nivell d'abstracció i es descriu cada una de les capes.

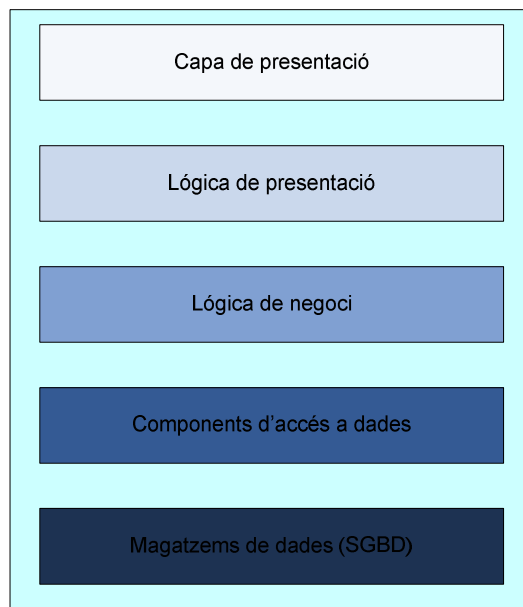


Figura 8 - Vista lògica

- **Capa de presentació:** Conté els elements que configuren la interfície d'usuari (conté formularis, arxius de configuració i arxius de codi). És la capa que veu l'usuari.
- **Lògica de presentació:** S'encarrega de gestionar la interfície d'usuari. Gestiona la configuració de la presentació i el seu funcionament. Controla la capa de presentació.
- **Lògica de negoci:** Conté la lògica de l'aplicació a nivell d'algorismes i estructures de dades.
- **Components d'accés a dades:** Conté les classes que encapsulen les funcions de crides als proveïdors d'accés a dades. Permeten ocultar les particularitats de cada SGBD.
- **Magatzems de dades (SGBD):** Conté les bases de dades utilitzades per l'aplicació. També inclou els proveïdors d'accés a dades que permeten l'accés al SGBD.

### 4.1.3. Vista de components

Donat que la nostra aplicació es pot executar en un únic PC i que inicialment no es preveu una aplicació molt extensa ni amb un nombre gaire elevat de classes creiem que resulta més senzill mantenir la aplicació en un únic programa executable.

Per tant no es preveu inicialment la creació de components tot i que en cas de voler reutilitzar parts de l'aplicació en d'altres projectes resultaria força interessant.

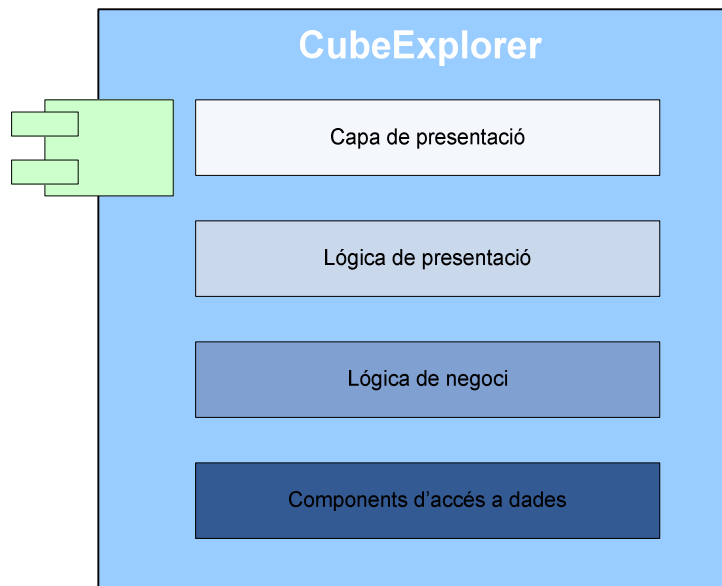


Figura 9 - Vista de components

## 4.2. Decisions tecnològiques

### Llenguatge de programació

El llenguatge de programació utilitzat és el C# versió 4. S'utilitza aquest llenguatge per la seva potència, elegància, estabilitat i implantació en la tecnologia .NET.

### Framework .NET

S'utilitza el framework de Microsoft versió 4.0. És la última versió d'aquest framework aparegut l'Abril del 2010.

### Entorn de desenvolupament

S'utilitza el IDE Microsoft Studio 2010 Ultimate. Aquest IDE permet desenvolupar aplicacions en .NET en qualsevol dels llenguatges suportats per aquesta plataforma.

### SGBD

S'utilitza Microsoft SQL Server 2008 R2 Express.

La versió gratuïta és suficient per desenvolupar un projecte d'aquestes característiques. Aquest SGBD s'utilitzarà per emmagatzemarà les dades del programa i per crear els exemples de cubs.

Per crear exemples també utilitzarem bases de dades Microsoft Access. Aquesta base de dades segueix utilitzant-se molt actualment malgrat les seves limitacions i ens permet demostrar la heterogeneïtat de la aplicació desenvolupada i la flexibilitat per adaptar-se a diferents SGBD.

### **Accés a dades**

L'accés a dades a les diferents bases de dades es realitza utilitzant els proveïdors d'accés a base de dades, en el cas de SQL Server es pot utilitzar el proveïdor d'accés a dades SQL Native Client de .NET o bé el proveïdor per OLE DB.

En el cas de Microsoft Access cal utilitzar el proveïdor OLE DB ja que no existeix un proveïdor .NET específic.

Per la execució de consultes i accés al catàleg del SGBD s'utilitzen les classes de la llibreria ADO.NET inclosa en el framework .NET.

S'utilitza aquesta tecnologia perquè la nostra aplicació requereix un accés a bases de dades heterogènies i accedir al catàleg de la base de dades per obtenir-ne els objectes.

La utilització de frameworks com ADO.NET Entity Framework o LINQ no ens ofereixen cap avantatge en el context d'aquesta aplicació i per tant no son adients.

L'ús de serveis web o de capes de serveis windows a través de WCF (Windows Communication Foundation) és molt interessant i ofereix amplies possibilitats d'obtenció de dades a través de protocols web. Malgrat tot donat l'abast i complexitat de la solució no es contempla en aquest projecte.

### **Interfície d'usuari**

Existeixen actualment diverses alternatives dins la tecnologia .NET (Windows Forms, WPF i ASP.NET).

S'ha optat per escollir la tecnologia WPF (Windows Presentation Foundation) per ser una tecnologia moderna que permet crear aplicacions molt atractives des de un punt de vista visual.

Aquesta tecnologia permet l'execució de l'aplicació tant de forma independent com dins de l'explorador web. En el nostre cas crearem una aplicació d'escriptori.

### 4.3. Components externs

El component essencial d'aquest projecte és el cub OLAP.

Només plantejar aquest projecte va resultar evident que calia buscar un control existent al mercat ja que aquest control no està inclòs com a control estàndard i resulta inviable plantejar desenvolupar-ne un de propi.

He optat per buscar un component comercial a Internet. Els criteris d'elecció han estat els següents:

- 1) 100% orientat i desenvolupat en .NET. Cal evitar tecnologies obsoletes com COM i ActiveX ja que no són suportades pels fabricants i moltes vegades tenen problemes de compatibilitat amb els sistemes actuals.
- 2) Que ofereixi màxim funcionalitat i que sigui realment un cub i no una grid.
- 3) Que el fabricant proporcioni el codi font de exemples, i que aquests funcionin.
- 4) Una documentació del component completa i entenedora.

De totes les versions avaluades una de adient m'ha semblat la de la companyia DevExpress versió 2011. Aquest framework disposa d'un control Pivot Grid per WPF) que permet implementar cubs OLAP.

Aquest component permet realitzar operacions com:

- Suport per múltiples fonts de dades.
- Càlcul automàtic de cada cel·la.
- Nombre indefinit de camps de dades, de columnes i de files.
- Totals calculats de forma automàtica.
- Filtratge de dades.
- Ordenació automàtica de dades de files i columnes.
- “Drag and drop” de files i columnes, possibilitat de modificació interactiva.

S'han utilitzat d'altres components d'aquest framework per construir la capa de presentació de l'aplicació.

Els components que s'han utilitzat han estat: DXNavBar, DXGrid, DXLayoutControl, i DXPivotGrid per la implementació del cub.

Aquests components són molt sòlids i complerts, permeten crear amb relativa facilitat interfícies d'usuari molt atractives i amigables. Tenen una documentació excel·lent amb abundants exemples.

La versió descarregada i instal·lada ha estat la d'avaluació. En principi al tractar-se d'una aplicació no comercial es pot fer ús d'aquests components durant un període concret de temps.

Cal tenir instal·lat aquests components tant per compilar el codi font de l'aplicació com per executar l'aplicació.



#### 4.4. Diagrama estàtic de disseny

El diagrama mostra les classes més rellevants de l'aplicació, de cada classe es mostren els seus atributs i els seus mètodes principals. També es poden observar les relacions existents entre les diferents classes.

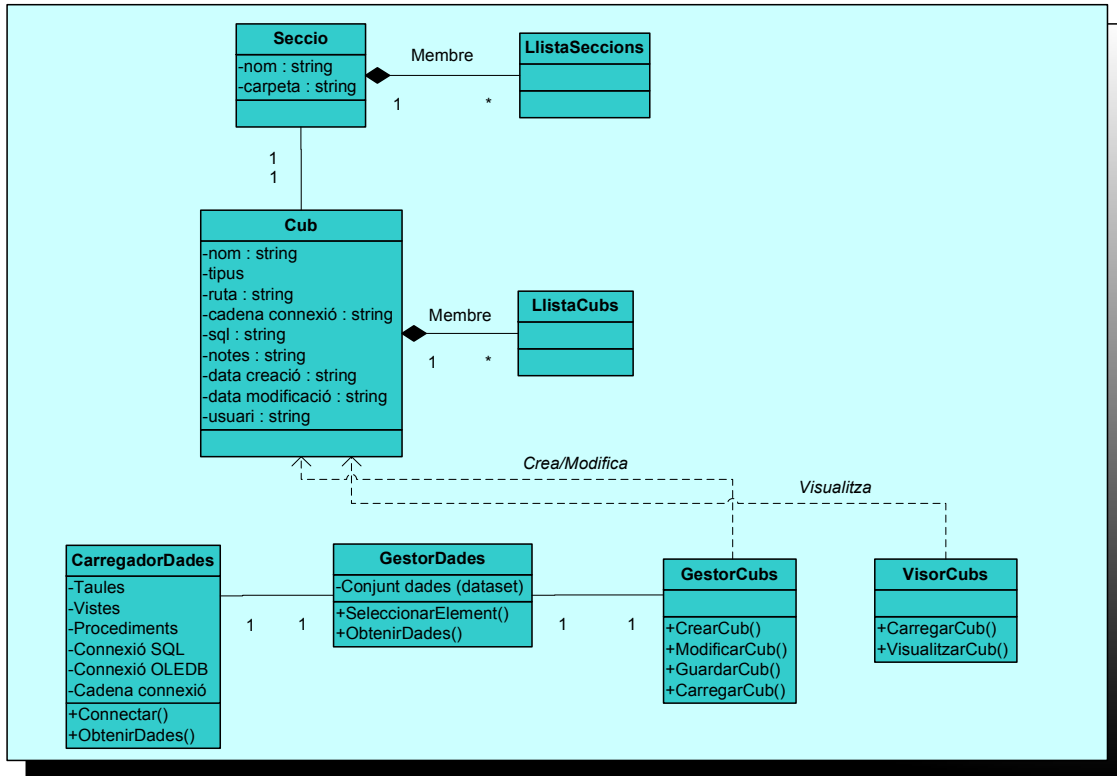


Figura 10 - Diagrama de classes

#### 4.5. Diagrama de seqüència

El següent diagrama mostra el diagrama de seqüència que correspon a l'acció de creació d'un cub nou com a cas més representatiu d'aquesta aplicació.

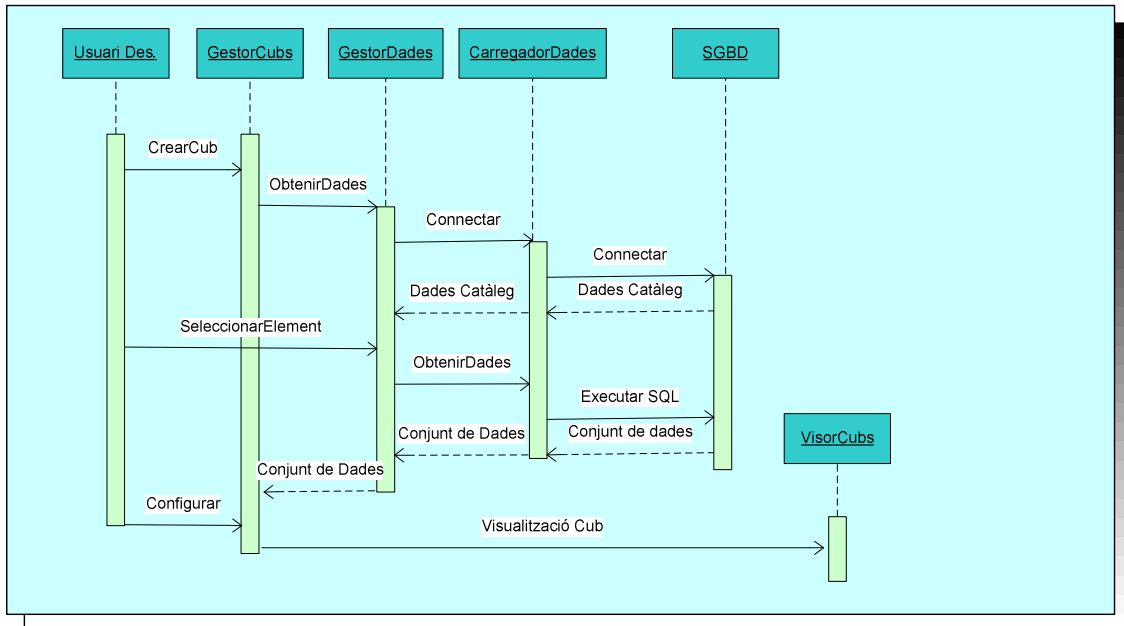


Figura 11 - Diagrama de seqüència

La seqüència de creació d'un cub és la següent:

- L'usuari envia la senyal de creació d'un cub a través de la seva interfície.
- El gestor de cubs carrega el gestor de dades i aquest a la seva vegada crida al carregador de dades.
- El carregador de dades mostra el diàleg de connexió a una base de dades a través d'un proveïdor .NET o bé OLEDB.
- Si la connexió s'ha efectuat correctament el carregador de dades crea una llista amb les taules un altre amb les consultes i una amb els procediments emmagatzemats.
- El Gestor de dades carrega una interfície de selecció amb les dades que li proporciona el carregador de dades.
- El usuari selecciona un dels elements de la llista (només en pot seleccionar un).
- El Gestor de dades sol·licita al carregador de dades les dades corresponents al element seleccionat. En el cas d'un procediment emmagatzemat caldrà sol·licitar la introducció dels valors dels paràmetres per poder obtenir un conjunt de dades (dataset).
- El carregador de dades executa aquesta consulta contra la connexió al SGBD que té oberta i retorna un conjunt de dades desconnectat (dataset).
- El Gestor de dades a la seva vegada retorna al GestorCubs aquest conjunt de dades.
- El GestorCubs mostra un diàleg on l'usuari configura la estructura i l'aparença del cub.
- El GestorCubs emmagatzema el cub a disc i crida al VisorCubs perquè visualitzi el cub emmagatzemat.

Correspondència entre els components de la capa lògica i alguns dels objectes descrits anteriorment.

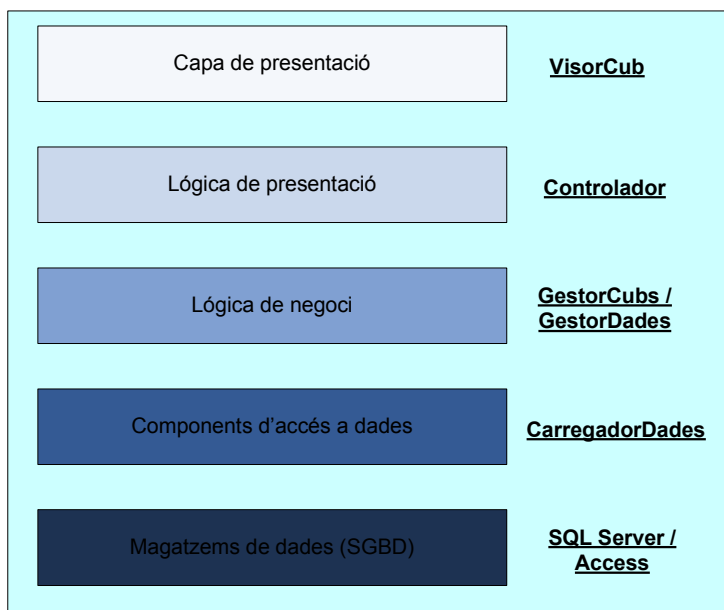


Figura 12 - Correspondència components

#### 4.6. Disseny persistència

##### 4.6.1. Model relacional de la base de dades

El camp Id és la clau primària de cada una de les taules.

**Cubs**(Id,Nom,Descripció,Tipus,Ruta,Connexio,Sql,Notes,DataCreacio,DataModificacio,IdUsuari és clau forana a Usuaris, IdSeccio és clau forana a Seccions)

**Seccions** (Id,Nom,Descripció,Carpeta)

**Usuaris** (Id,Nom,ParaulaClau,Perfil valors {1,2,3})

#### 4.6.2. Diagrama base de dades

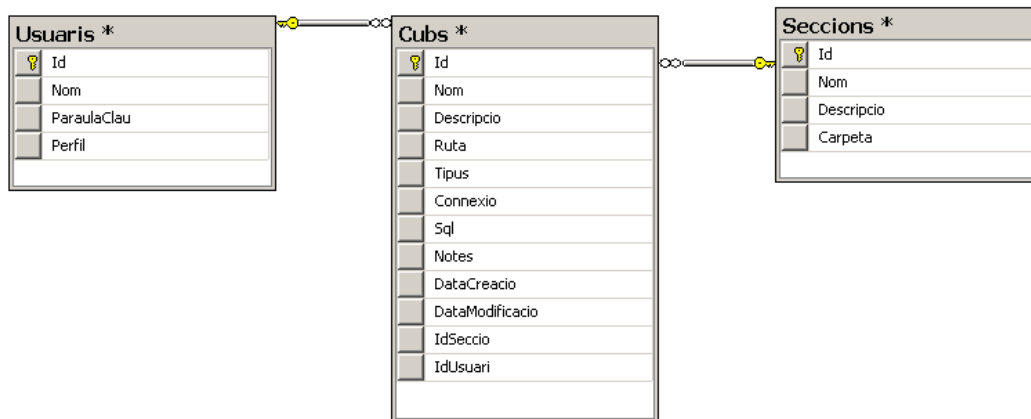


Figura 13 - Diagrama base de dades

#### 4.6.3. Format de fitxer

El format de l'arxiu cub és binari i és propietari del fabricant del component, s'emmagatzema comprimit. Permet guardar tant les dades del cub i la presentació del cub.

#### 4.7. Prototipus interfície d'usuari

En aquest apartat es van dissenyar les diferents interfícies que componen l'aplicació i el seu funcionament a nivell operatiu.

La interfície es va dissenyar amb el propi Visual Studio 2010 i va resultar essencialment molt similar a com ha resultat al final.

Ometo la inclusió de les captures de pantalla perquè es troben a la secció d'implementació d'aquesta memòria.

## 5. Implementació

La següent secció descriu els aspectes més rellevants de la implementació.

### 5.1. Arquitectura de la solució

#### Projecte

L'aplicació consta d'un únic projecte CubeExplorerUI.csproj que conté el component principal de l'aplicació.

#### Espai de noms

S'utilitza un únic espai de noms CubeExplorerUI.

#### Carpetes

El codi font s'organitza en carpetes per una major claredat i comoditat. Així s'han creat les següents carpetes:

- Cubs: Conté tot el tractament de orígens de dades, connexions i gestió de cubs.
- Seccions: Conté la gestió de seccions en que s'organitzen els cubs.
- Usuaris: Conté la gestió d'usuaris de l'aplicació.
- Carpeta principal del projecte: Conté la finestra principal i la gestió de l'explorador de cubs.

### 5.2. Estructura

L'aplicació segueix un model de tres capes en que tenim capa de presentació, lògica de negoci i capa de persistència.

#### 5.2.1. Capa de presentació

Conté els arxius en format .xaml.

XAML és un llenguatge de format per a interfícies d'usuari en WPF.

Es tracta d'un llenguatge declaratiu basat en XML optimitzat per escriure interfícies d'usuari riques des de un punt de vista gràfic.

Aquest arxiu XAML té una classe associada amb el mateix nom que té com a classe pare una classe Window. Aquesta classe s'encarrega d'inicialitzar la finestra i de controlar i personalitzar el seu comportament.

L'aplicació funciona amb una finestra principal que conté el menú principal de l'aplicació, el panell de connexió i desconnexió del usuari, la grid amb els cubs disponibles segons la secció activa. Una previsualització del cub es mostra en el panell inferior de la grid. Segons la opció que s'activa es mostra un diàleg sobre la finestra principal. Per exemple en la creació i modificació del cub.

El estil de presentació escollit és "Office 2007 blue", aquest estil es pot seleccionar pels components de presentació de DevExpress.

### 5.2.2 Lògica de negoci

Són les classes que implementen la lògica de l'aplicació.

En aquesta aplicació són classes de lògica de negoci aquelles que implementen funcionalitat per exemple la gestió del assistent que permet la creació d'un cub.

### 5.2.3. Persistència

La capa de persistència està implementada sobre una base de dades SQL Server 2008 Express R2. La capa de persistència s'ha creat utilitzant el model de accés de dades directe de ADO.NET.

Es podria haver utilitzat un framework de persistència, per exemple ADO.NET Entity Framework però he cregut que donat que el nombre de taules a mantenir és petit la utilització d'aquest framework no hagués suposat un estalvi significatiu en el temps de desenvolupament i hagués afegit complexitat a la solució.

L'accés a dades es realitza en classes específiques, generalment una per cada entitat de la base de dades: Seccions, Usuaris, Explorador (cubs i orígens de dades).

Aquestes classes tenen definides operacions específiques per actualitzar les entitats de la base de dades, típicament: Crear, Modificar, Eliminar i Llistar.

Per accedir a la base de dades s'executa un procediment emmagatzemat contra la base de dades de l'aplicació. Aquests procediments emmagatzemats s'han definit en llenguatge T-SQL i ja es troben inclosos a la base de dades.

S'ha creat un procediment emmagatzemat per cada una de les operacions d'actualització en cada una de les taules: Crear, Llistar, Modificar i Eliminar.

Aquest procediments reben els paràmetres necessaris que són subministrats des de l'aplicació. En alguns casos retornen valors que són consultats per l'aplicació.

Aquest model resulta avantatjós perquè no tenim codi T-SQL inclòs dins l'aplicació i per tant facilita la comprensió i el manteniment de l'aplicació.

A més és més eficient que la execució directa de codi SQL donat que el propi servidor pot executar més d'una sentència SQL dins d'un mateix procediment emmagatzemat i per tant es transfereix procés en el servidor de dades i es redueix la transferència de dades entre el servidor i l'aplicació.

Al exemple veiem la funció que ens retorna la llista de cubs creats per una determinada secció.

```
public DataTable LlistarCubs(int IdSeccio)
{
    try {
        conn.Open();

        // Llistar tots els cubs
        SqlDataAdapter da = new SqlDataAdapter("[LlistarCubs]", conn);
        da.SelectCommand.CommandType = CommandType.StoredProcedure;
        da.SelectCommand.Parameters.Add("@IdSeccio", SqlDbType.Int);
        da.SelectCommand.Parameters["@IdSeccio"].Value = IdSeccio;

        DataTable dt = new DataTable();
        da.Fill(dt);
        conn.Close();

        return dt;
    } catch (Exception e) {
        MessageBox.Show(e.Source + ": " + e.Message, "CubeExplorer");
        return null;
    }
}
```

Figura 14 - Exemple codi "LlistarCubs"

### 5.3. Patrons de disseny

Com aspectes més rellevants del codi font podríem destacar la utilització d'un patró de disseny orientat a objectes observador que permet realitzar la subscripció i notificació d'events entre classes.

Aquesta comunicació és molt útil en la comunicació d'events entre diferents classes de la interfície d'usuari i permet separar la funcionalitat entre cada classe.

Així per exemple la classe `UsuariObrir` que manipula la interfície de connexió d'un usuari a l'aplicació no controla directament la lògica de connexió sinó que delega aquesta tasca a la classe principal `MainWindow`.

Per implementar aquest patró definim a la classe principal el delegat `ValidarUsuariHandler` que s'associa al mètode `ValidarUsuari`.

La classe `UsuariObrir` es subscriu a aquest event.

```
public delegate void ValidarUsuariHandler (string usuari, string clau);

public void ValidarUsuari(string usuari, string clau)
{
    // codi de la validació del usuari
}

Public MainWindow()
{
    UsuariObrir usOpen = new UsuariObrir();
    // subscripció del event
    usOpen.OnValidarUsuari += new ValidarUsuariHandler(ValidarUsuari);
}
```

**Figura 15 - Exemple codi "ValidarUsuari"**

```
public partial class UsuariObrir : Window
{
    public string nom { get; set; }
    public string clau { get; set; }
    public event ValidarUsuariHandler OnValidarUsuari;

    public UsuariObrir()
    {
        InitializeComponent();
    }

    private void btnConnect_Click(object sender, RoutedEventArgs e)
    {
        string usuari = txtUsuari.Text;
        string clau = pbeClau.Text;

        // cridar el event de validació del usuari
        OnValidarUsuari(usuari, clau);
    }
}
```

**Figura 16 - Exemple codi "UsuariObrir"**

#### ***5.4. Components externs utilitzats***

S'utilitzen dues llibreries de components per la realització de tasques específiques de l'aplicació.

##### ***5.4.1. Components DevExpress***

És una llibreria de components de presentació per .NET.

Disposa de components per diferents tecnologies com ASP.NET, Winforms, WPF.

Hem utilitzat una versió de avaluació per WPF vàlida per aplicacions no comercials.

Aquesta llibreria ens ofereix un ric conjunt de components. Els que s'han utilitzat en el projecte són:

- BarManager: Per implementar el menú principal de l'aplicació.
- DockLayoutManager: Permet dividir la finestra en diferents zones i redimensionar el seu tamany.
- Layout Panel: Correspon a una zona de la finestra. En aquest panell es pot insertar un altre finestra.
- Stack Panel: Permet posicionar els controls d'una finestra. Per exemple posicionar els controls a l'esquerra de la finestra.
- GridControl: Ofereix un control grid, s'ha utilitzat per implementar el explorador i els assistents de construcció del cub.
- PivotGridControl: Ofereix un control cub OLAP. Aquest control s'ha utilitzat per implementar la construcció, modificació i visualització del cub.



### 5.4.2. Components d'accés a dades

Utilitzem el SDK estàndard de Microsoft Data Designer Extensibility (DDEX) SDK. Aquest SDK es troba inclòs en la carpeta de instal·lació de Microsoft Visual Studio 2010. Per utilitzar-lo cal referenciar al projecte les llibreríes Microsoft.Data.ConnectionUI.dll i Microsoft.Data.ConnectionUI.Dialog.dll.

Aquest SDK ofereix informació sobre els orígens de dades disponibles i permet cridar als diàlegs de connexió estàndard que permeten seleccionar el proveïdor de connexió tant sigui .NET com OLEDB o ODBC, la base de dades i introduir els paràmetres de connexió corresponents.

L'ús d'aquests components es troba encapsulat en la classe CarregadorDades. A continuació veiem un exemple del mètode de connexió.

```
public bool Connectar()
{
    try
    {
        // Diàleg de connexió
        DataConnectionDialog objDataConnectionDialog = new DataConnectionDialog();

        // Afegir fonts de dades estàndard
        DataSource.AddStandardDataSources(objDataConnectionDialog);

        // Mostrar diàleg de connexió
        if (DataConnectionDialog.Show(objDataConnectionDialog) ==
            System.Windows.Forms.DialogResult.OK)
        {
            // Si confirmat agafem les dades de connexió
            name = objDataConnectionDialog.SelectedDataSource.Name;
            provider = objDataConnectionDialog.SelectedDataProvider.Name;
            strconnexio = objDataConnectionDialog.ConnectionString;

            // Càrrega del catàleg del sistema
            connectat = CarregarCatalog();

            objDataConnectionDialog.Dispose();
        }
        return connectat;
    }
    catch (Exception e) {
        MessageBox.Show (e.Source + ": " + e.Message, "CubeExplorer");
    }
    connectat = false;
    return connectat;
}
```

Figura 17 - Exemple codi "Connectar"

## 5.5. Accés al catàleg del sistema

Per aquest projecte hem implementat suport per als SGBD Microsoft Access i Microsoft SQL Server.

Cal tenir en compte que les consultes de accés al catàleg de un sistema al altre difereixen completament.

### 5.5.1. Microsoft SQL Server

En aquest cas hem utilitzat SQL directament contra el catàleg del sistema.

- Taules: taula “sys.tables”
- Vistes: taula “sys.views”
- Procediments emmagatzemats: taula “sys.procedures”
- Paràmetres procediments emmagatzemats: taula “sys.parameters”

### 5.5.2. Microsoft Access

En aquest cas utilitzem mètodes que ens permeten accedir al catàleg del sistema sense utilitzar consultes SQL. S’ha escollit aquest sistema perquè en algunes bases de dades que s’han provat s’han detectat dificultats per executar consultes contra el catàleg del sistema.

```
DataTable table = cnnOleDb.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, restrictions);
```

**Figura 18 - Exemple codi "Catàleg Access"**

## 5.6. Arxiu de configuració

L’únic paràmetre que necessita l’aplicació per poder-se compilar i executar correctament és la cadena de connexió on s’emmagatzema la base de dades de l’aplicació.

L’arxiu Settings.Designer.cs inclou una propietat que emmagatzema el valor d’aquest paràmetre.

```
[global::System.Configuration.ApplicationScopedSettingAttribute()]  
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]  
[global::System.Configuration.DefaultSettingValueAttribute("Data Source=XAVIER-PC;initial catalog=CubeExplorer;user  
id=sa;password=1234;")]  
public string ConnectionString {  
    get {  
        return ((string)(this["ConnectionString"]));  
    }  
}
```

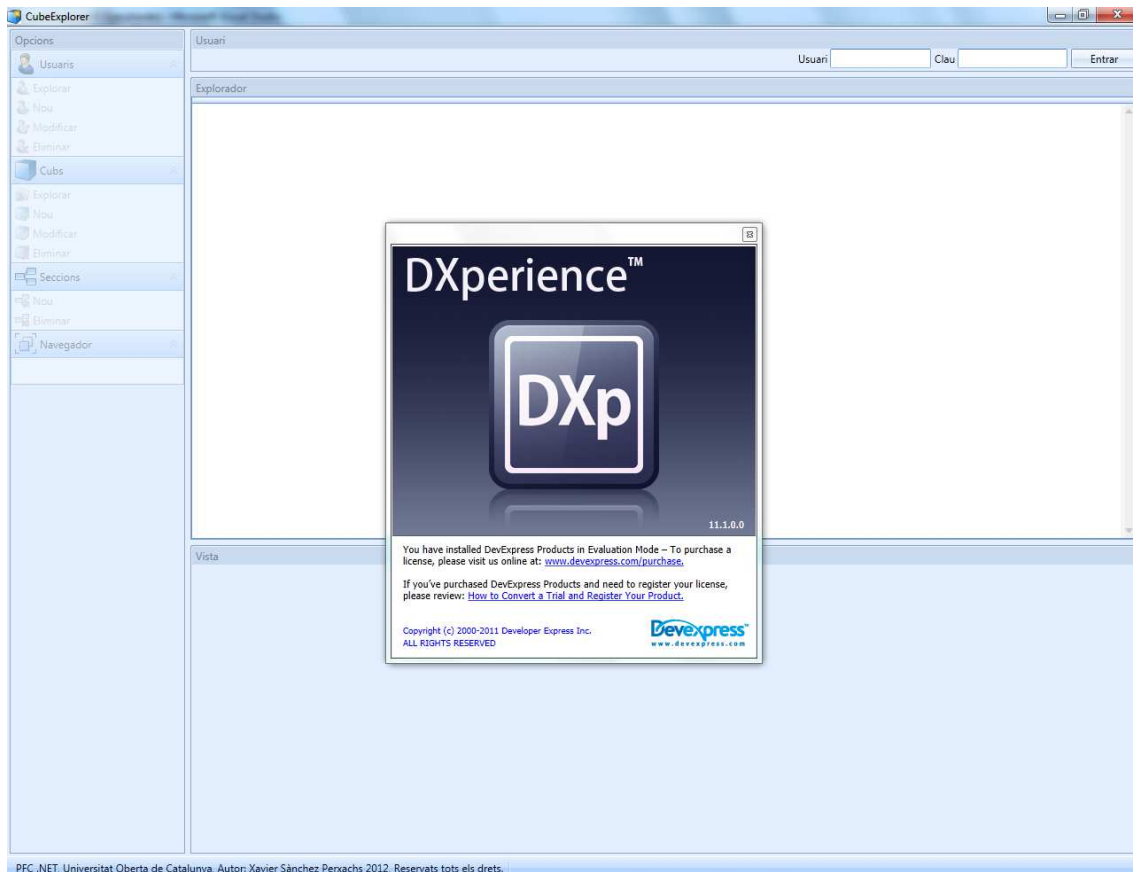
**Figura 19 - Arxiu de configuració**

## 5.7. Captures de pantalla

A continuació mostrem un recull de les pantalles més rellevants que demostren la funcionalitat de les possibilitats de l'aplicació.

### 5.7.1. Pantalla inicial

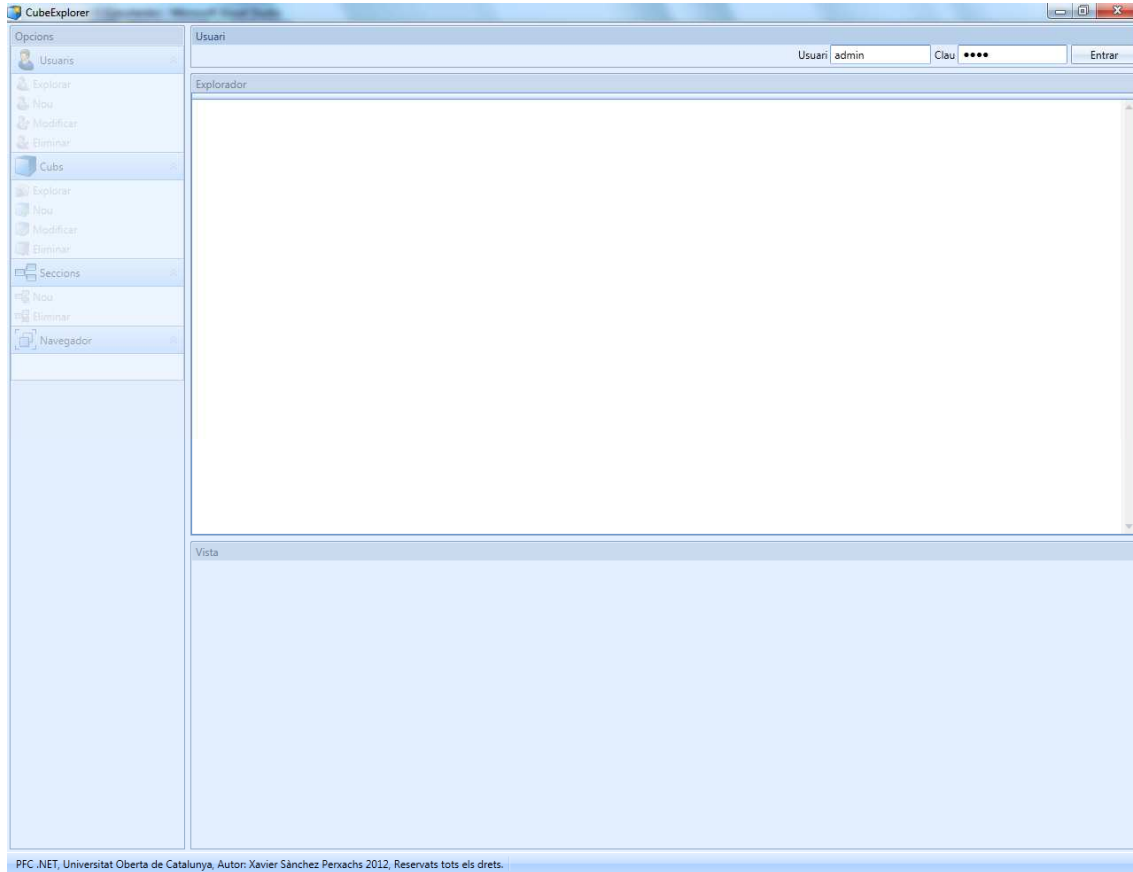
Quan entrem a l'aplicació apareix el diàleg que ens informa que estem utilitzant els components d'avaluació de DevExpress.



**Figura 20 - Pantalla Inicial**

### 5.7.2. Validació usuari

Per utilitzar l'aplicació hem de introduir un usuari vàlid del sistema i prémer el botó per entrar. Al exemple el usuari admin és administrador i té password "1234". En cas d'introduir un usuari no vàlid o un password incorrecte es mostrarà un missatge d'error.



**Figura 21 - Validació usuari**

### 5.7.3. Pantalla principal

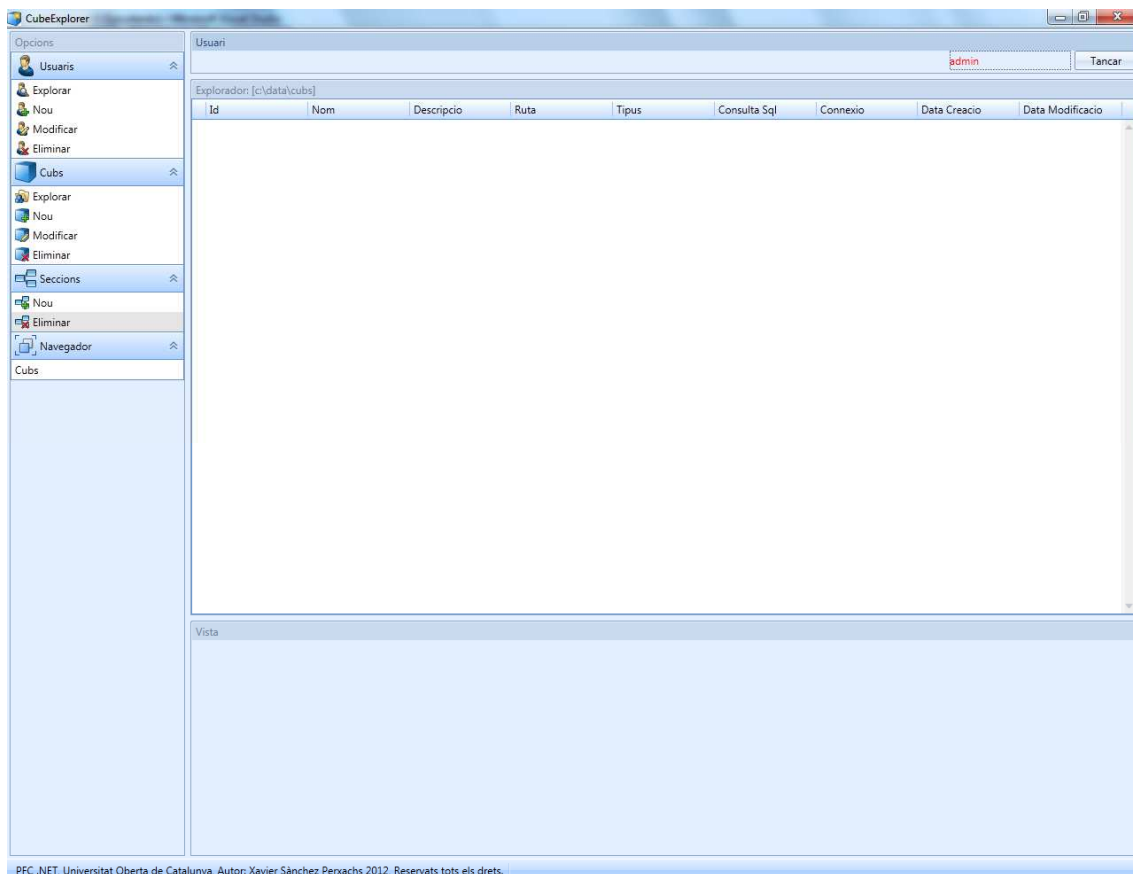
Un cop validats al sistema podem veure que la finestra principal de l'aplicació s'ha dividit en diferents seccions.

El panell usuari serveix per fer la connexió i la desconnexió de l'aplicació.

El panell opcions conté totes les opcions disponibles de l'aplicació.

La grid central serveix per explorar els cubs de la secció (carpeta) que tenim seleccionada). Al exemple podem veure que la secció cubs està buida.

El panell inferior permet fer una visualització del cub segons la fila de la grid de cubs que tenim seleccionada. Aquest panell també s'utilitza per mostrar el formulari de creació i modificació d'usuaris.



**Figura 22 - Pantalla principal**

#### 5.7.4. Cub Nou. Orígens de dades.

Per accedir a aquesta opció cal prémer el botó de Nou de la pestanya de Cubs. El diàleg que es mostra ens permet gestionar els orígens de dades. Un origen de dades conceptualment és equivalent a una cadena de connexió i és útil per estalviar-nos haver de crear la cadena de connexió cada vegada que volem crear un cub. En aquest cas com que no tenim orígens de dades disponibles cal crear-ne un de nou.

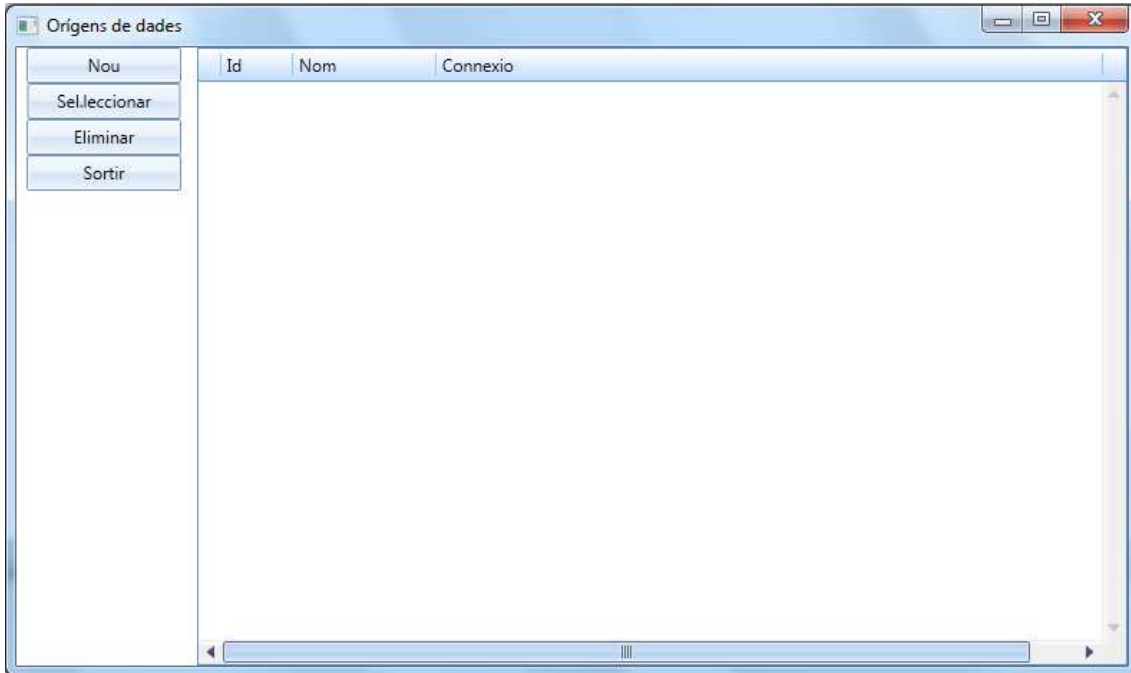


Figura 23 - Orígens de dades

#### 5.7.5. Seleccionar proveïdor base de dades

Cal especificar la base de dades i el proveïdor de connexió.

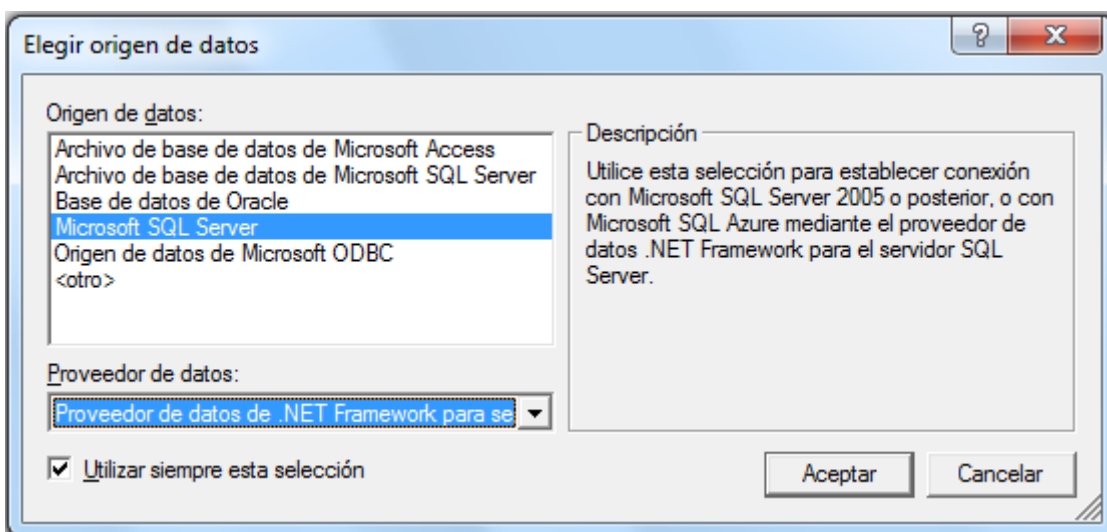


Figura 24 - Proveïdor de dades

### 5.7.6. Paràmetres de connexió

Indiquem el nom del servidor, nom de usuari, la clau d'accés i la base de dades si ens autèntiquem contra SQL Server o Windows si volem utilitzar la configuració del usuari de sistema operatiu.

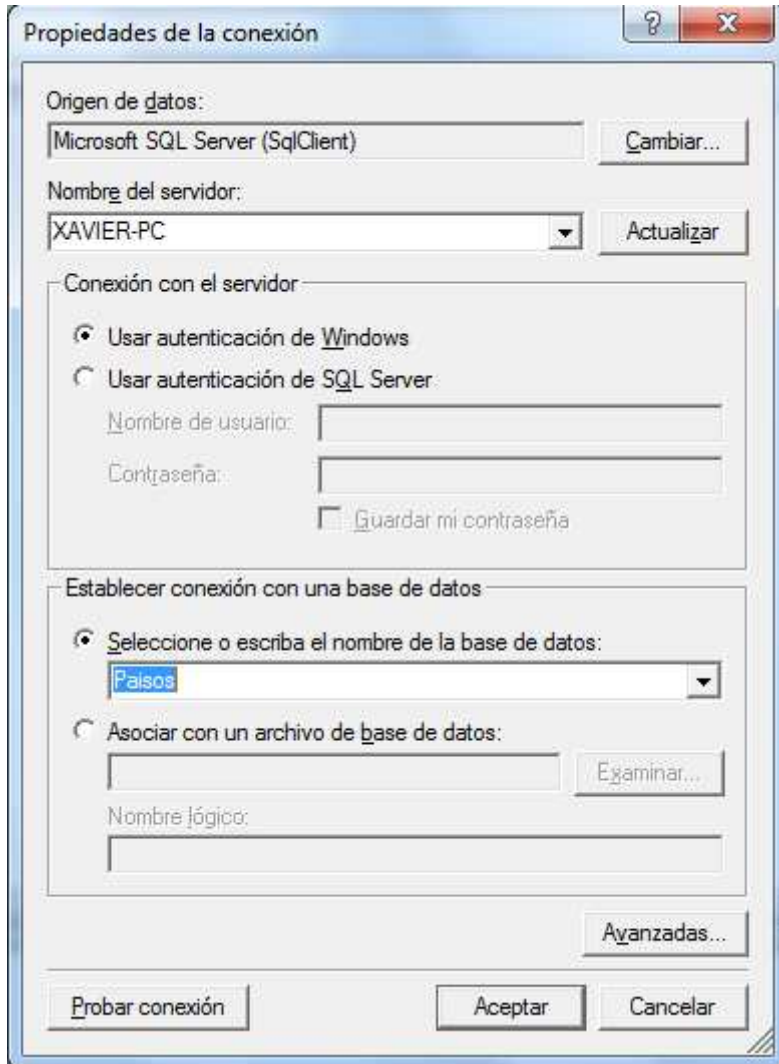


Figura 25 - Paràmetres de connexió

### 5.7.7. Nom origen de dades

El sistema ens demana el nom que li volem donar al origen de dades.  
Hem d'introduir un nom identificatiu que ens permeti identificar la cadena de connexió.



Figura 26 - Nom Origen de dades

### 5.7.8. Selecció de dades

Hem de seleccionar la taula, consulta o procediment emmagatzemat que ens proporciona les dades que ens permetran construir el cub.  
Al exemple seleccionem la taula de poblacions.

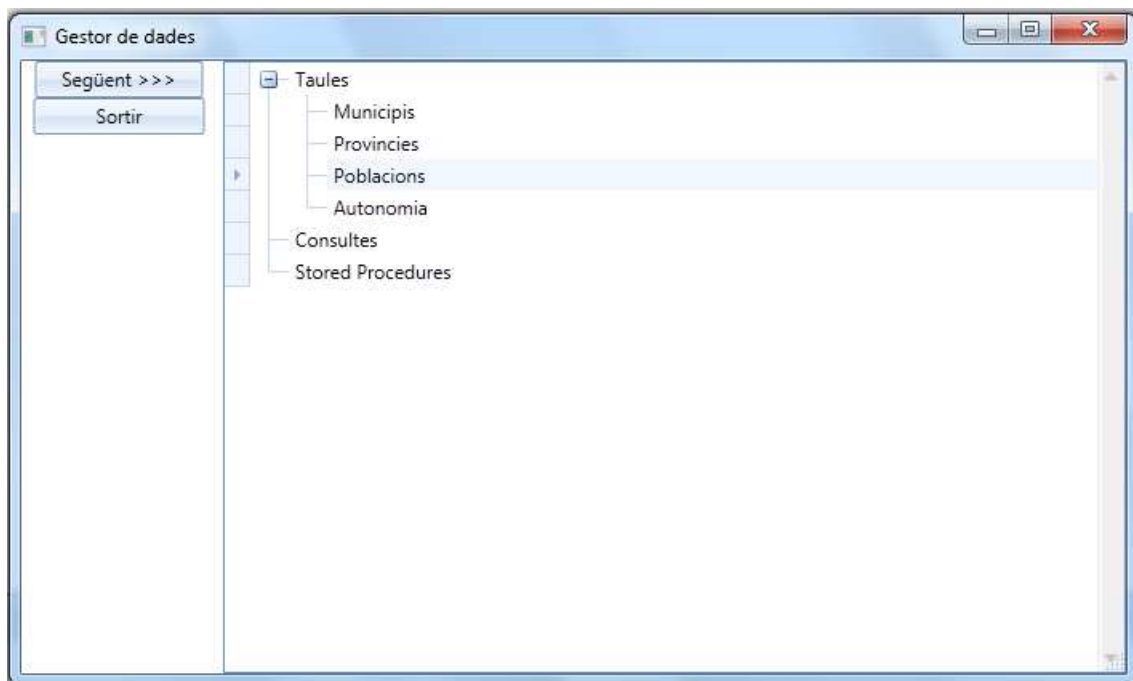


Figura 27 - Selecció de dades



### 5.7.9. Seleccionar camps

El sistema ens mostra una grid on hem de seleccionar els camps de la consulta que intervenen en la construcció del cub, la seva posició dins el cub (àrea), el títol que se li dona a cada camp i el format numèric del camp pels camps de dades.

Per afegir un nou camp dins el cub cal clicar el botó de nou. Quan premem el botó de següent validem la selecció dels camps.

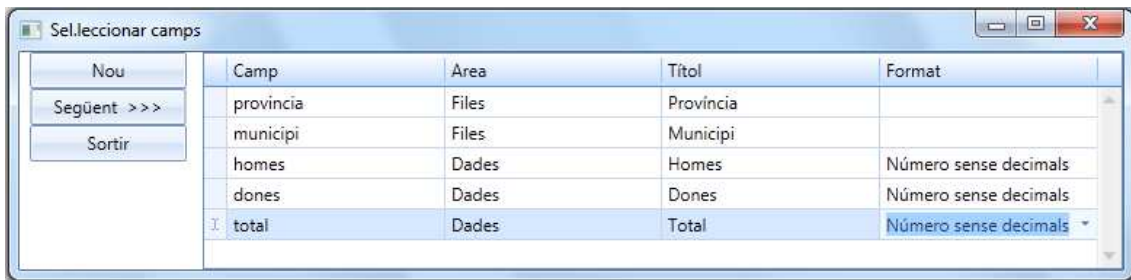


Figura 28 - Selecció de camps

### 5.7.10. Modificació del cub

Aquest diàleg ens permet modificar el cub. Al exemple següent hem indicat un filtre perquè ens visualitzi només 4 províncies. Per aplicar aquest filtre cal clicar a la clau que apareix sobre el camp agrupat.

Una vegada hem acabat premem el botó guardar.

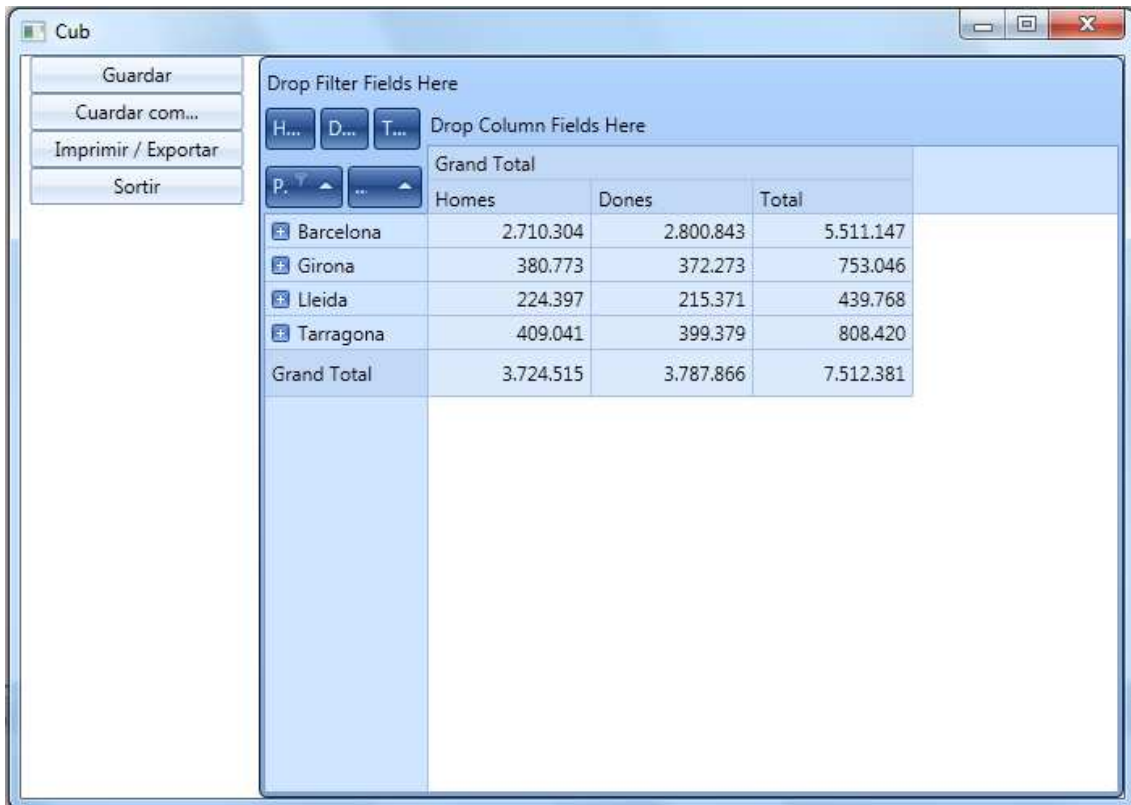
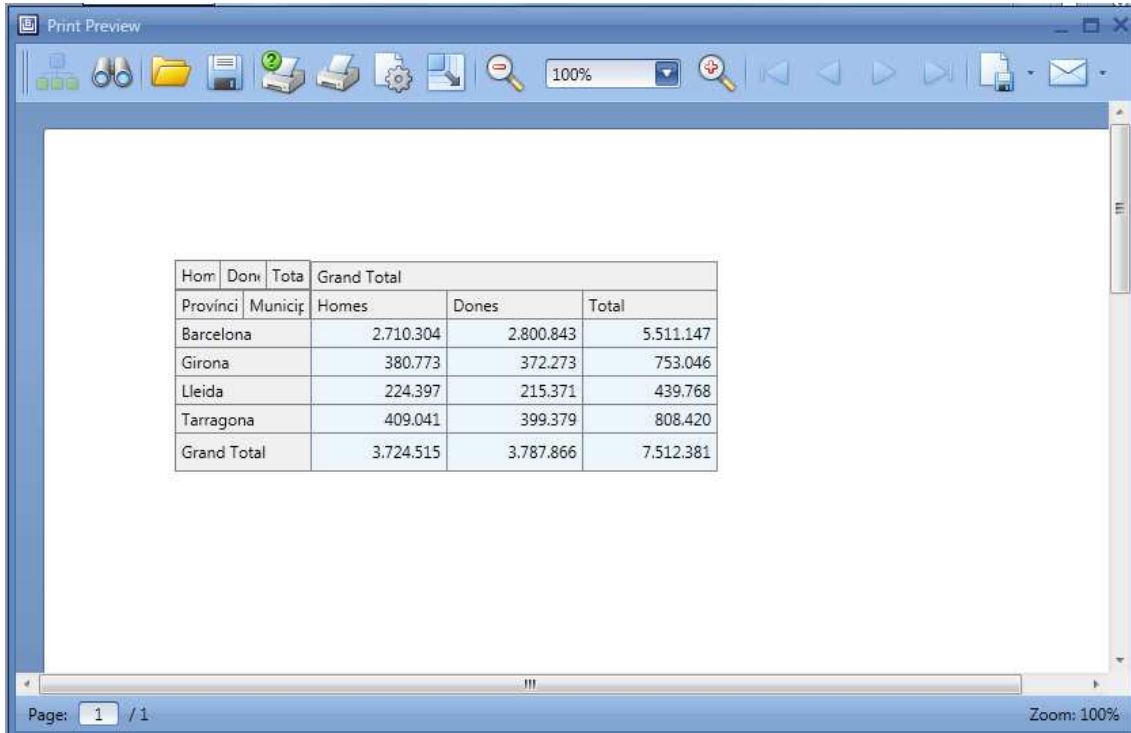


Figura 29 - Modificació del cub

### 5.7.11. Imprimir/Exportar

Si premem aquest botó s'obre un diàleg que permet imprimir les dades del cub i fer diferents operacions sobre les dades, com per exemple fer una exportació en diferents formats (.XLS, .PDF, .TXT, etc).



The screenshot shows a 'Print Preview' window with a toolbar at the top containing icons for print, save, and zoom. The main content area displays a data table with the following structure:

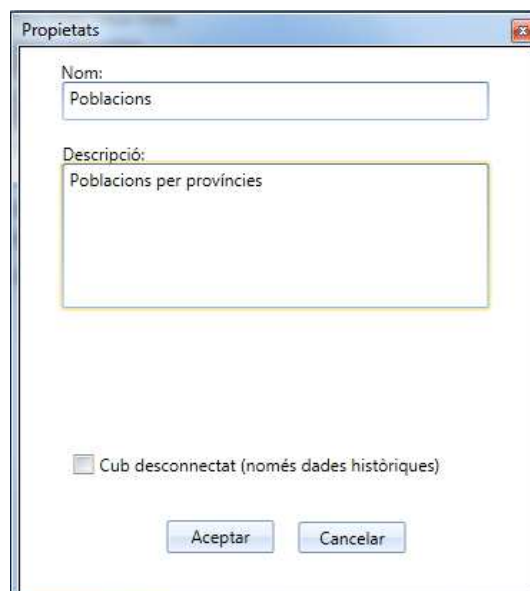
Hom	Don	Tota	Grand Total		
Província	Municipi		Homes	Dones	Total
Barcelona			2.710.304	2.800.843	5.511.147
Girona			380.773	372.273	753.046
Lleida			224.397	215.371	439.768
Tarragona			409.041	399.379	808.420
Grand Total			3.724.515	3.787.866	7.512.381

At the bottom of the window, it shows 'Page: 1 / 1' and 'Zoom: 100%'.

Figura 30 - Exportació del cub

### 5.7.12. Guardar el cub

En aquest diàleg podem indicar el nom del cub, una descripció del seu contingut i si el cub és desconnectat. L'arxiu del cub s'emmagatzema amb extensió .cub.



The 'Propietats' dialog box contains the following elements:

- Nom:** A text box containing the value 'Poblacions'.
- Descripció:** A text area containing the value 'Poblacions per províncies'.
- Cub desconnectat (només dades històriques)
- Aceptar** and **Cancelar** buttons at the bottom.

Figura 31 - Guardar el cub

### 5.7.13. Explorador de cubs / seccions

El explorador mostra els cubs de la secció activa. Al exemple veiem el cub creat als apartats anteriors a la secció “Cubs”. Carpeta: c:\data\cubs.

Al panell vista veiem una visualització del cub seleccionat al explorador, si seleccionem un altre cub la vista canvia.

La icona de reciclatge que apareix al panell de visualització del cub serveix per refrescar les dades del cub. Aquesta opció només serà possible si els cubs són connectats.

Si volem mostrar els cubs d'un altre secció hem de seleccionar un altre secció (pengen de les opcions del navegador).

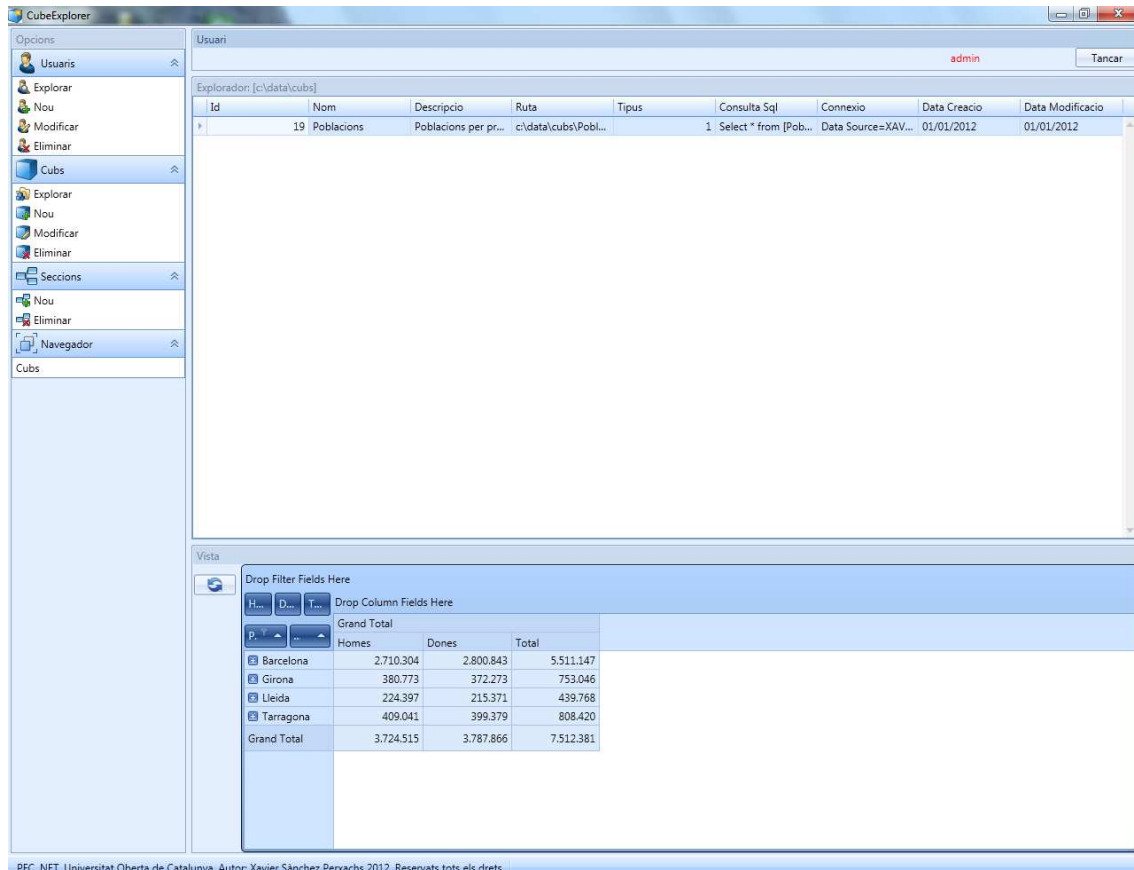


Figura 32 - Explorador de cubs / seccions

### 5.7.14. Eliminar un cub

Hem de prémer el botó eliminar. El sistema ens demana confirmació.

Si confirmem la eliminació el cub s'elimina del explorador però l'arxiu .cub no es borra físicament de la carpeta.

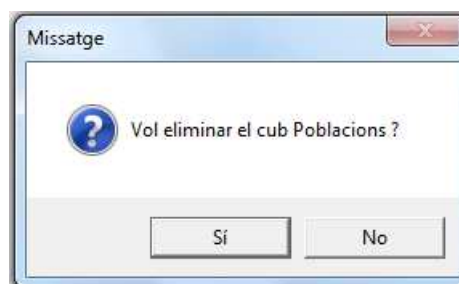


Figura 33 - Missatge eliminar cub

### 5.7.15. Crear una nova secció

Hem de prémer el botó nou de seccions.

Apareix un diàleg on indiquem el nom de la secció, la descripció i seleccionem una carpeta on s'emmagatzemen els cubs.

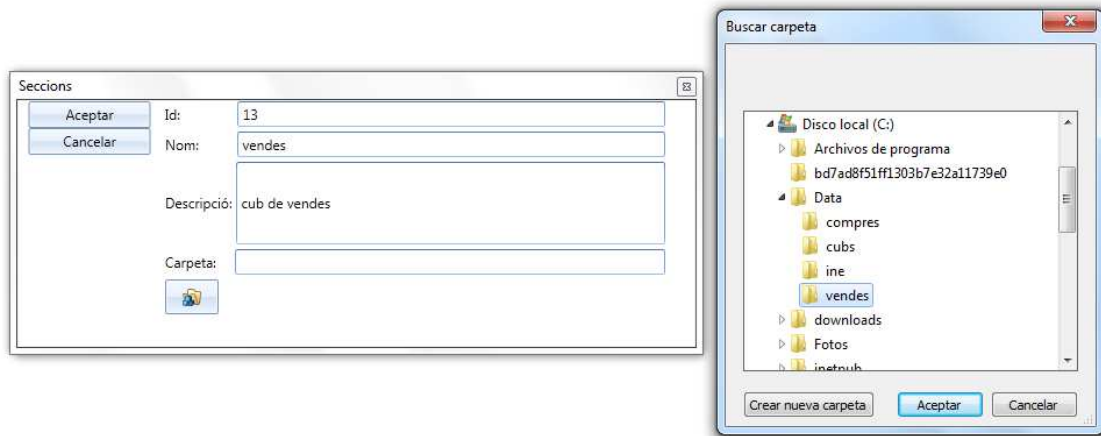


Figura 34 - Crear nova secció

### 5.7.16. Eliminar seccions

La secció s'elimina clicant al botó eliminar. Apareix un missatge de confirmació.



Figura 35 - Eliminar seccions

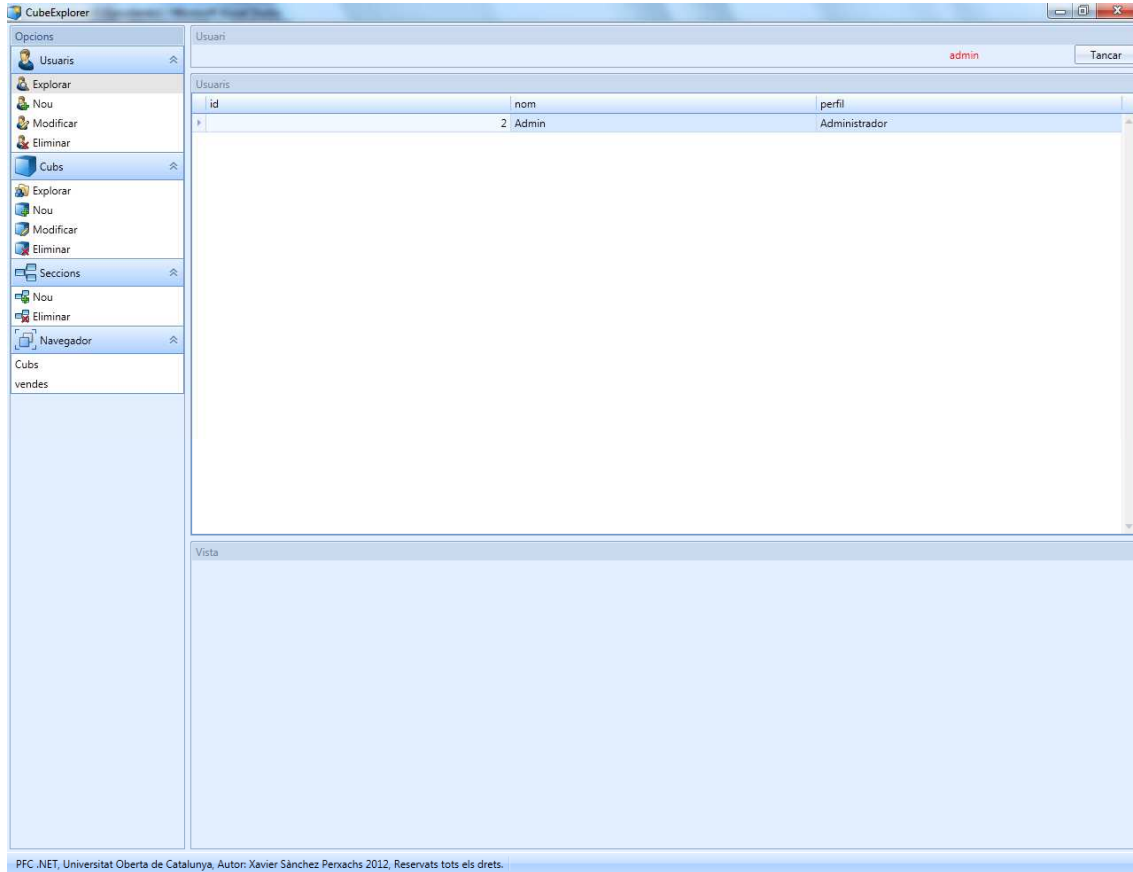
No es permet eliminar una secció amb cubs creats. Cal anar eliminant els cubs de forma individual i finalment eliminar la secció.



Figura 36 - Missatge eliminar seccions

### 5.7.17. Gestió d'usuaris

Els usuaris amb permisos d'administradors poden veure els usuaris que poden accedir al sistema. Al exemple veiem que hi ha un únic usuari creat.



**Figura 37 - Gestió d'usuaris**

### 5.7.18. Creació nou usuari

Hem de prémer el botó nou. Podem introduir les dades del usuari: nom, clau d'accés (no visible) i nivell d'accés. Per confirmar cal prémer Acceptar.  
La modificació de les dades de l'usuari funciona de forma anàloga.

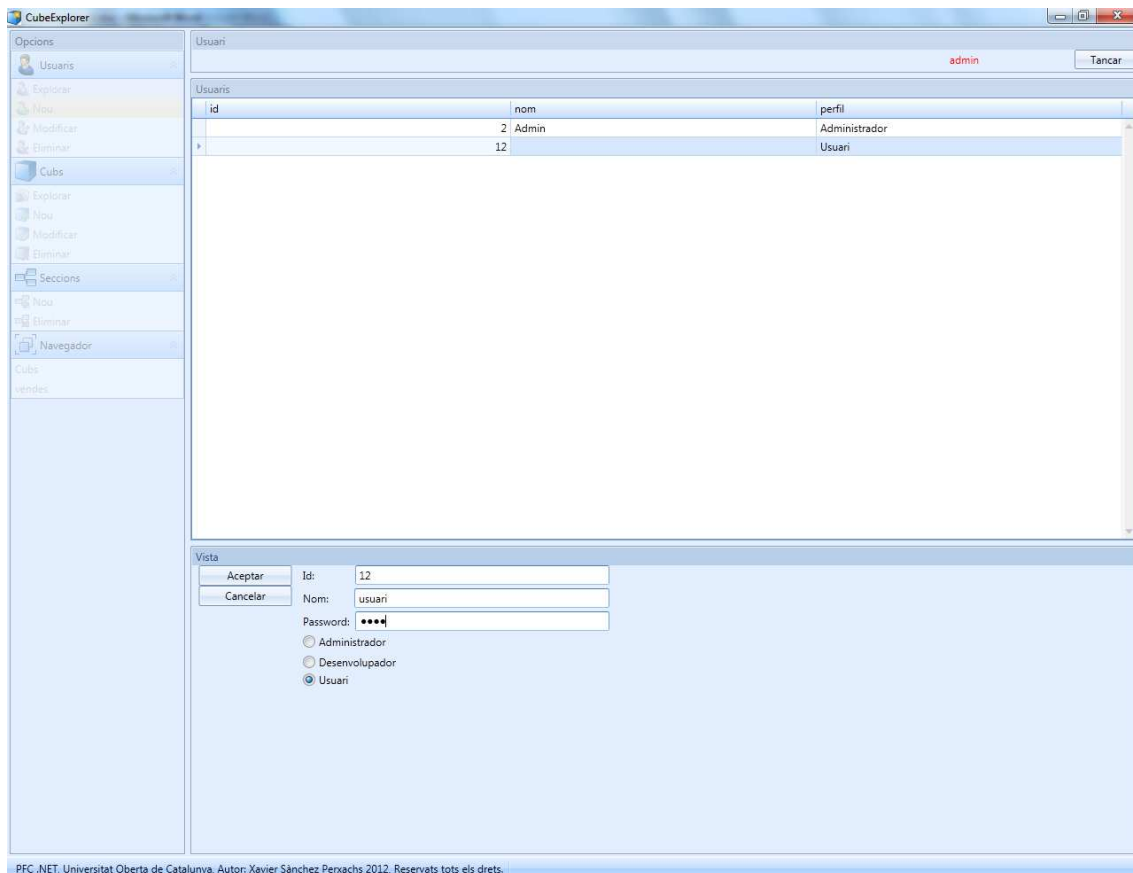


Figura 38 - Creació / modificació usuaris

### 5.7.19. Eliminació d'un usuari

Si seleccionem un usuari i premem el botó de eliminar el sistema ens demana confirmació de la eliminació.

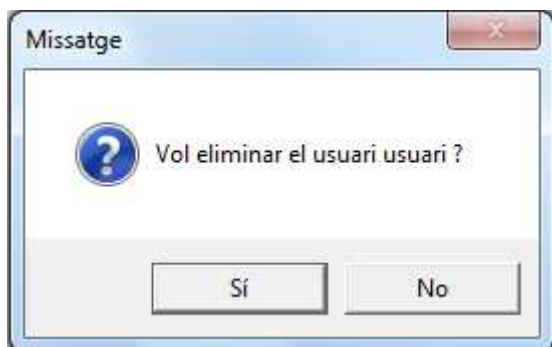


Figura 39 - Eliminació usuari

### 5.7.20. Usuari amb permisos restringits

Al exemple hem tancat la sessió (botó tancar) i hem tornat a entrar amb l'usuari que havíem creat prèviament.

Veiem com moltes opcions del sistema apareixen inactives ja que aquest tipus d'usuari només poden consultar cubs.

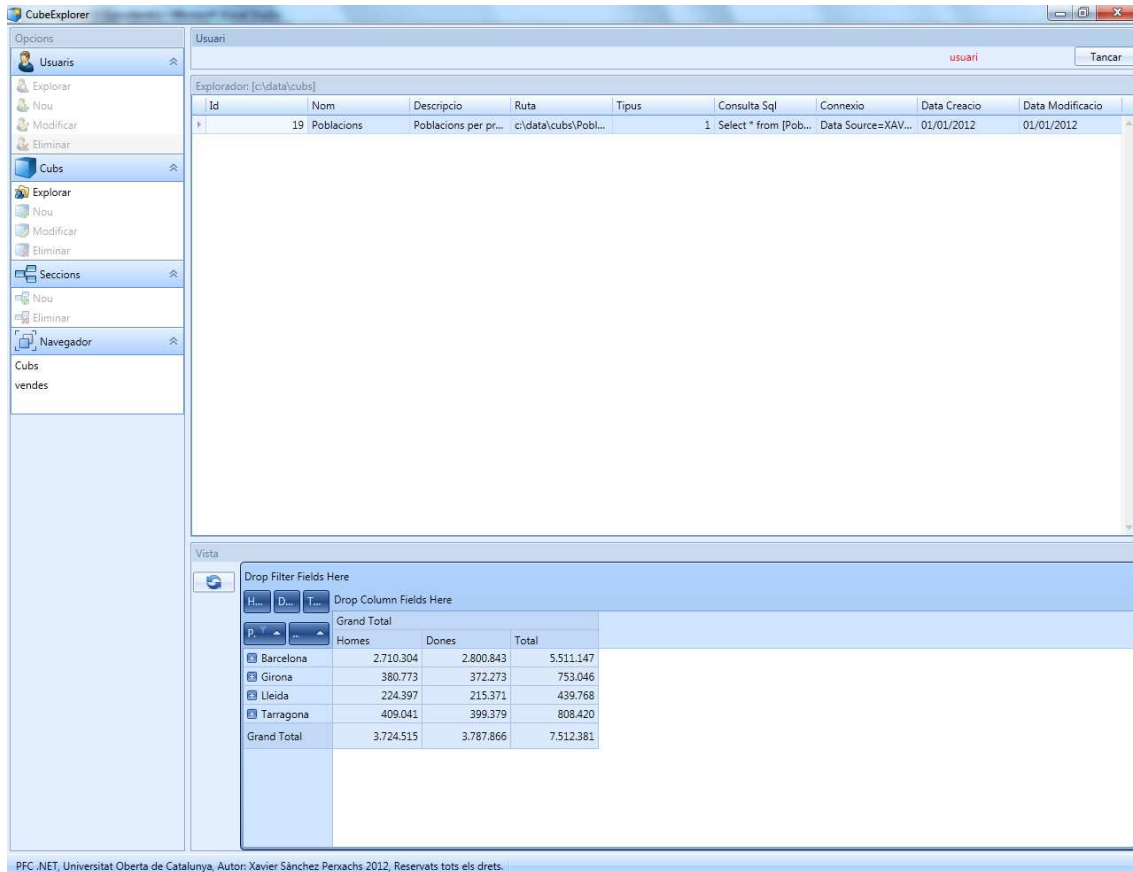


Figura 40 - Usuari sense permisos

## **6. Millores futures**

Al llarg del desenvolupament han anat apareixent idees de com millorar l'aplicació.

Per manca de temps no les he pogut dur a terme.

Crec que val la pena enumerar-les en una secció per veure el camí per on podria evolucionar l'aplicació en un futur.

### **6.1. Funcionalitat**

- Permetre connectar a diferents proveïdors de dades d'altres fabricants com Oracle, MySQL, DB2, etc.
- Permetre la connexió a dades a través de WCF. Seria molt útil ja que permetria connectar a orígens de dades remots a través de Web.
- Incloure la possibilitat d'edició de consultes SQL perquè el sistema actual resulta una mica limitat ja que obliga a tenir les consultes creades al SGBD prèviament.
- En el cas de cubs basats en procediments emmagatzemats es sol·liciten els valors dels paràmetres per cada actualització del cub. Seria interessant poder emmagatzemar els valors a la base de dades i que no es demanessin cada vegada.
- L'explorador de cubs podria incloure una opció de cerca de cubs.
- Permetre aplicar diferents tipus d'operacions aritmètiques sobre les dades totalitzades del cub.
- Incloure la possibilitat de crear camps totalitzats en funció dels camps de dades escollits.
- Gràfics estadístics sobre les dades dels cubs.

### **6.2. Seguretat**

- Xifratge de passwords de les bases de dades.
- Permetre restringir l'accés a les seccions en funció del nivell d'accés de l'usuari.
- Millorar la gestió d'usuaris.

### **6.3. Interfície d'usuari**

- Es podria fer més personalitzable, per exemple poden escollir el skin de presentació.
- Millora de la presentació en general: tant pel que fa a la interfície principal com als diferents assistents.
- Internacionalització de les interfícies.



## 7. Conclusions

La realització d'aquest projecte ha suposat un gran esforç degut a que requereix una dedicació constant de varies hores diàries durant tot els mesos que dura el projecte.

En alguns moments m'he sentit una mica fatigat ja que pel temps que disposava diàriament era necessària una dedicació molt constant intentant de aprofitar al màxim el temps de dedicació diària.

He pogut aprendre molt sobre programació en .NET i sobre els controls DevExpress. No m'ha costat excessivament adquirir els coneixements necessaris per desenvolupar en C#. Crec que per una persona que tingui experiència en llenguatges orientats a objecte com C++ o Java no li ha de resultar difícil fer la transició a aquest llenguatge.

He pogut posar en pràctica coneixements adquirits al llarg de la carrera sobretot de les àrees d'Enginyeria del Programari, Bases de dades, Models multidimensionals i Gestió de Projectes.

La planificació que es va estimar inicialment ha resultat vàlida i s'ha pogut seguir rigorosament. Crec que la metodologia basada en fites és molt encertada ja que permet distribuir molt bé la carrega de treball.

Respecte el resultat final estic força satisfet ja que s'han assolit els objectius proposats inicialment. L'aplicació compleix la funcionalitat desitjada i és perfectament usable.

En la meua opinió el nivell de qualitat de l'aplicació és correcte.

Sóc perfectament conscient de les seves limitacions i que és molt millorable en bastants aspectes sobretot si es volgués arribar a comercialitzar algun dia.

Comento algunes d'aquestes millores en l'apartat de millores futures. Desitjo que algun dia es puguin dur a terme.

Finalment agrair al consultor els ànims donats i les propostes de millora de l'aplicació.

## **8. Bibliografia**

- Ferguson, J.; Patterson B.; Beres, J. (2003). La Biblia C#. Anaya Multimedia.
- Hotek, M. Microsoft SQL Server 2008. Anaya Multimedia.

## **Materials de assignatures UOC**

### **PFC**

- Ceballos, J. Introducció a .NET. FUOC.
- Pérez, A.; Bataller, A; Beneito, R; Sáenz, N; Vidal, R. (2008). Treball Final de Carrera. FUOC.

### **Models multidimensionals i magatzems de dades**

- Abelló, A. (2003). Disseny multidimensional. FUOC.

### **Internet**

- MSDN Centro para desarrolladores .NET - <http://msdn.microsoft.com/es-es/netframework>
- MSDN Centro para desarrolladores WPF - <http://msdn.microsoft.com/es-es/windows/ff798266>
- DevExpress: WinForms, ASP.NET, WPF, Silverlight - <http://devexpress.com/>
- INE. Instituto Nacional de Estadística. Demografía y Población - [http://www.ine.es/inebmenu/mnu\\_cifraspob.htm](http://www.ine.es/inebmenu/mnu_cifraspob.htm)