

DISSENY DE SOFTWARE

TOTSERIES

ITERACIO 3

Angel Bergantiños – 16319144

Francisco Diaz – 16828405

Índex

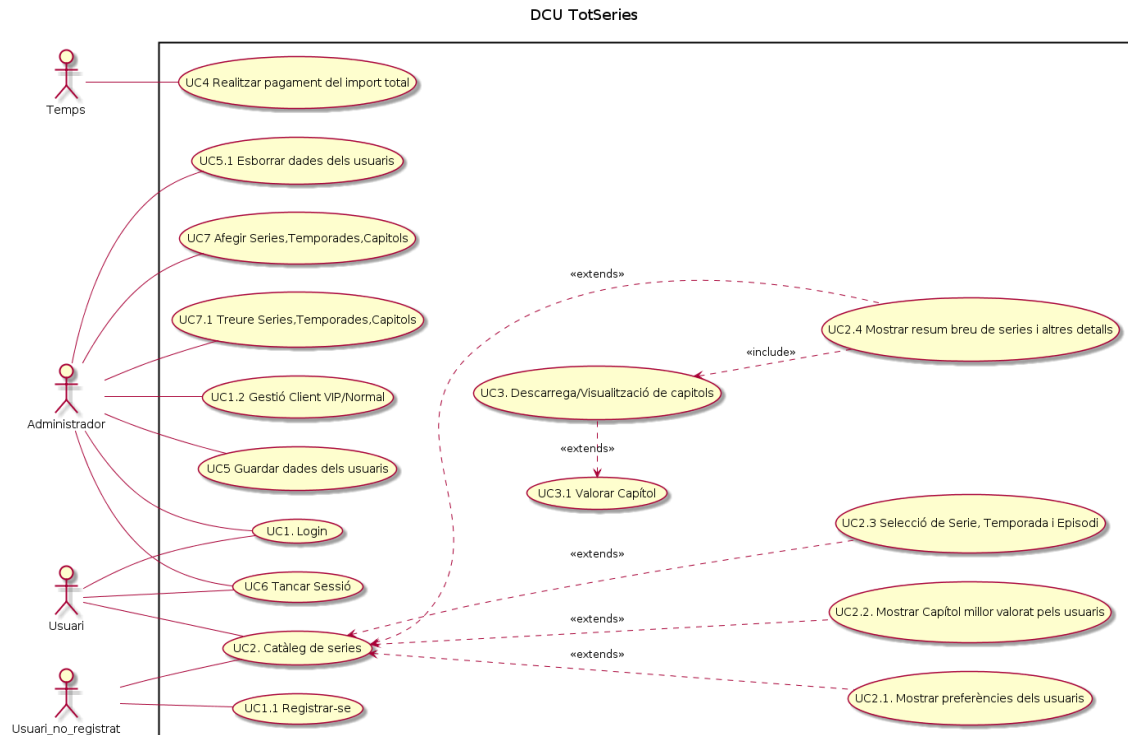
Models.....	4
DCU.....	4
Casos d'us textual.....	5
UC1: Registre d'Usuaris.....	5
UC 1.1: Login.	5
UC1.2: Selecció de client VIP o normal.	6
UC2: Catàleg de Sèries.	6
UC2.1: Preferències dels usuaris.....	7
UC2.2 Mostrar capítol més valorat	7
UC2.3: Selecció de Sèrie/Temporada/Capítol.....	8
UC2.4: Mostrar breu resum de les sèries i altres detalls.	8
UC3: Descarrega/Visualització via streaming de capítols.	9
UC3.1: Valorar Capítol.....	9
UC4: Realitzar pagament del import total.	10
UC5: Guardar/Esborrar dades dels usuaris.	10
UC5.1: Esborrar dades dels usuaris.....	11
UC6: Tancar Sessió	11
UC7: Afegir/Treure Sèries, Temporades, Capítols.....	12
UC7.1 : Treure Series, Temporades, Capítols.....	12
Model de domini	13
DSS.....	14
DSS 1: Registrar usuari	14
DSS 2: Visualitzar catàleg	14
DSS 3: Visualitzar i valorar capítol	15
DSS 4: Visualitzar capítols més valorats	15
DS	16
DS 1: Registrar usuari	16
DS 2: Visualitzar catàleg	17
DS 3: Visualitzar i valorar capítol.....	18
DS 4-1: Visualitzar capítols més valorats.....	20
DS 4-2: Visualitzar capítols més vists	21
DCD.....	22
Observacions.....	23

Conclusions	24
Codis	25
DCU.....	25
Model de domini	26
DSS.....	27
DSS1.....	27
DSS2.....	27
DSS3.....	28
DSS4.....	29
DS	30
DS1	30
DS2	31
DS3.1	32
DS3.2	33
DS3.3	34
DS4.1	35
DS4.2	36
DCD.....	37

Models

DCU

En el DCU trobarem tots els actors que tindrà la nostra aplicació, així com les accions que podran donar a terme.



Com podem veure, tindrem 4 actors: usuari (registrat), usuari no registrat, administrador i el temps.

L'usuari registrat tindrà accés a les funcionalitats de la pàgina relacionades amb les series.

L'usuari no registrat només podrà veure el catàleg.

L'administrador podrà donar el VIP a usuaris registrats.

El temps serà un actor que ens servirà per saber quan cobrar als usuaris registrats.

Casos d'us textual

En aquesta part, aprofundirem més en tots els casos d'us abans esmentats, donant una explicació més detallada.

UC1: Registre d'Usuaris

Descripció: UC1: Registre d'Usuaris
Actors: Usuari no registrat.
Precondicions: L'aplicació ha d'estar inicialitzada.
Flux Bàsic: 1) L'aplicació mostra per pantalla un pop-up/finestra on l'usuari pot registrar-se. 2) L'usuari ha de introduir una sèrie de dades corresponents: nom, contrasenya, data de naixement i nacionalitat. 3) Després pot interactuar amb totes les funcions de l'aplicació (UC2,UC2.1,UC2.2,UC2.3, UC2.4).
Flux Alternatiu: 1.a) En cas de que l'usuari entri les dades malament el programa mostrarà per pantalla un missatge d'error.
Postcondicions: L'usuari acaba registrat i es fa automàticament el UC1.1

UC 1.1: Login.

Descripció: UC 1.1: Login.
Actors: Usuari registrat, Administrador.
Precondicions: L'aplicació ha d'estar inicialitzada.
Flux Bàsic: 1) L'aplicació mostra un pop-up on l'Usuari o l'Administrador pot iniciar sessió. 2) L'Usuari o l'Administrador introdueix les dades necessàries per loguejar-se.
Flux Alternatiu: 2a) L'Usuari o Administrador introdueix les dades de manera errònia. Apareixerà un missatge per pantalla dient que s'han introduït les dades malament.
Postcondicions: L'Usuari o l'Administrador es logueja dins l'aplicació.

UC1.2: Selecció de client VIP o normal.

Descripció: UC1.2: Selecció de client VIP o normal.
Actors: Administrador.
Precondicions: L'usuari ha d'estar al menys registrat.
Flux Bàsic: 1) L'administrador seleccionarà si aquest usuari es tornarà VIP o només serà un usuari de tipus normal. 2) En cas de que l'administrador decideixi que un usuari sigui VIP s'aplicarà un descompte del 10% del import mensual. En cas contrari l'import serà del 100% (usuari normal).
Flux Alternatiu: 1a) En el cas de que no hi hagin usuaris registrats dins de la aplicació es mostrarà per pantalla un missatge dient que no hi ha cap usuari i per tant no es pot realitzar la operació desitjada.
Postcondicions: Usuari acaba sent VIP o no.

UC2: Catàleg de Sèries.

Descripció: UC2: Catàleg de Sèries.
Actors: Usuari No Registrat, Usuari Registrat, Administrador.
Precondicions: Aplicació inicialitzada.
Flux Bàsic: 1) Després de haver iniciat l'aplicació es mostra per pantalla el llistat de totes les sèries. 2) Els usuaris registrats i L'administrador poden interactuar amb els UC2.1/UC2.2/UC2.3/UC2.4 que poden realitzar dins del catàleg.
Flux Alternatiu: 2a) L'Usuari No Registrat no pot fer res més que veure el catàleg, si vol interactuar amb ell ha de realitzar el cas d'ús UC1.
Postcondicions: Els actors han vist el catàleg de series.

UC2.1: Preferències dels usuaris.

Descripció: UC2.1: Preferències dels usuaris.
Actors: Usuari Registrat, Usuari No Registrat, Administrador.
Precondicions: Aplicació Inicialitzada.
Flux Bàsic: 1) Al passar el UC2 també es mostra les preferències dels usuaris. 2) L'Usuari registrat i l'Administrador poden interactuar amb l'UC3.
Flux Alternatiu: 2a) L'Usuari no registrat no pot accedir-hi perquè per fer-ho ha d'estar registrat. (UC1)
Postcondicions: Accés a les preferències dels usuaris (Sèries/Temporades/Capítols).

UC2.2 Mostrar capítol més valorat

Descripció: UC2.2: Mostrar Capítol més valorat.
Actors: Usuari, Administrador.
Precondicions: Aplicació inicialitzada.
Flux Bàsic: 1) Al passar el UC2, també es mostra per pantalla el capítol més valorat positivament pels usuaris. 2) L'Usuari, que ha d'estar registrat, i l'administrador poden accedir-hi i interactuar amb aquesta.
Flux Alternatiu: 2a) En ell cas que l'usuari no estigui registrat no podrà accedir-hi perquè ha de complir el UC1.
Postcondicions: Accés al capítol més valorat pels usuaris.

UC2.3: Selecció de Sèrie/Temporada/Capítol.

Descripció: UC2.3: Selecció de Sèrie/Temporada/Capítol.
Actors: Usuari Registrat, Administrador.
Precondicions: Aplicació inicialitzada i Usuari/Administrador loguejats.
Flux Bàsic: 1) Des de el UC2 es selecciona una sèrie-temporada-capítol. 2) Després es pot interactuar amb les diferents opcions a poder realitzar.
Flux Alternatiu: 1a) Es busca una sèrie-temporada-capítol que no està disponible. Es mostrarà un error per pantalla dient que no existeix o no està disponible.
Postcondicions: Accés a la sèrie-temporada-capítol.

UC2.4: Mostrar breu resum de les sèries i altres detalls.

Descripció: UC2.4: Mostrar breu resum de les sèries i altres detalls.
Actors: Usuari (Registrat o no), Administrador.
Precondicions: Aplicació inicialitzada.
Flux Bàsic: 1) Juntament amb el UC2 es mostra per pantalla un petit resum de les sèries i altres detalls com per exemple l'elenc. 2) L'usuari i l'administrador poden interactuar amb les sèries.
Flux Alternatiu: 2a) En el cas de que l'usuari sigui un no registrat es mostrarà per pantalla un missatge d'error dient que s'ha d'estar registrat, és a dir, complir el UC1.
Postcondicions: Accés a la sèrie.

UC3: Descarrega/Visualització via streaming de capítols.

Descripció: UC3: Descarrega/Visualització via streaming de capítols.
Actors: Usuari registrat.
Precondicions: L'usuari ha d'estar loguejat. Única visualització alhora.
Flux Bàsic: 1) L'usuari, després d'haver seleccionat la sèrie, la temporada i el capítol que vol veure haurà de descarregar-lo via streaming per poder visualitzar-lo. 2) Al visualitzar aquest capítol, es carregarà al compte del usuari un import de 1€. 3) A final de mes s'haurà de realitzar el pagament corresponent del total del import generat per les visualitzacions dels capítols que l'usuari ha escollit.
Flux Alternatiu: 1.a) En cas de que l'usuari intenti veure més d'un capítol alhora apareixerà per pantalla un missatge d'error informant a l'usuari que aquesta acció que vol realitzar està prohibida.
Postcondicions: Import carregat al compte, capítol marcat com a vist.

UC3.1: Valorar Capítol.

Descripció: UC3.1: Valorar Capítol.
Actors: Usuari Registrat.
Precondicions: Usuari ha de estar loguejat i haver visualitzat un capítol.
Flux Bàsic: 1) Després de complir el UC3, l'usuari pot comentar el capítol i posar una valoració entre 0 i 5 (en forma d'enters per tal de no comportar cap problema).
Flux Alternatiu: 1a) En el cas de que l'usuari introdueixi la valoració malament es mostra per pantalla un missatge d'error.
Postcondicions: La valoració es guarda.

UC4: Realitzar pagament del import total.

Descripció: UC4: Realitzar pagament del import total.
Actors: Temps.
Precondicions: L'usuari que ha d'haver visualitzat un capítol com a mínim per cobrar.
Flux Bàsic: 1) L'usuari, després de fer el UC3, se li cobrarà un import total a número de capítols vists. 2) L'import es pagarà a final de mes.
Flux Alternatiu: 1a) L'usuari no ha visualitzat cap capítol aquest mes, per tant no ha de pagar res. 2a) L'usuari no paga l'import d'un mes, per tant el seu compte es eliminat seguint el UC5.
Postcondicions: Pagaments realitzats.

UC5: Guardar/Esborrar dades dels usuaris.

Descripció: UC5: Guardar dades dels usuaris.
Actors: Administrador.
Precondicions: Aplicació inicialitzada, Administrador loguejat.
Flux Bàsic: 1) L'administrador comprova si hi han usuaris nous registrats, canvis en els usuaris antics o si hi ha usuaris que no compleixen el UC4. 2) L'administrador procedeix a guardar les dades dels usuaris.
Flux Alternatiu: 1a) En el cas de que no hi hagi cap usuari llavors es mostrarà per pantalla un missatge d'error dient que no hi ha cap usuari registrat.
Postcondicions: Canvis en les informacions dels usuaris.

UC5.1: Esborrar dades dels usuaris

Descripció: UC5.1: Esborrar dades dels usuaris.
Actors: Administrador.
Precondicions: Aplicació inicialitzada, Administrador loguejat.
Flux Bàsic: 1) L'administrador comprova si hi han usuaris nous registrats, canvis en els usuaris antics o si hi ha usuaris que no compleixen el UC4. 2) En el cas de que no hi hagi cap usuari llavors es mostrarà per pantalla un missatge d'error dient que no hi ha cap usuari registrat.
Flux Alternatiu: 2 ^a) En cas de que no hi hagi cap usuari es mostrarà per pantalla un missatge dient que no hi han usuaris.
Postcondicions: Canvis en les informacions dels usuaris registrats.

UC6: Tancar Sessió

Descripció: UC6: Tancar Sessió
Actors: Usuari Registrat, Administrador
Precondicions: Usuari registrat i Administrador loguejats.
Flux Bàsic: 1) L'usuari/administrador ha acabat de realitzar totes les accions que volien. 2) Seleccionen l'opció de tancar sessió. 3) Automàticament l'aplicació tanca la sessió i retorna al UC2.
Flux Alternatiu: N/A
Postcondicions: Usuari/Administrador han tancat sessió.

UC7: Afegir/Treure Sèries, Temporades, Capítols.

Descripció: UC7: Afegir Sèries, Temporades, Capítols.
Actors: Administrador
Precondicions: L'administrador ha d'estar loguejat.
Flux Bàsic: 1) L'administrador decideix afegir una nova sèrie/temporada/capítol. 2) Introdueix totes les dades necessàries. 3) S'afegeix un/a nova sèrie/temporada/capítol.
Flux Alternatiu: 1a) En el cas de que no hi hagi cap sèrie dins de l'aplicació es mostrarà un missatge per pantalla dient que no hi ha cap sèrie.
Postcondicions: S'afegeix sèrie/temporada/capítol.

UC7.1 : Treure Series, Temporades, Capítols

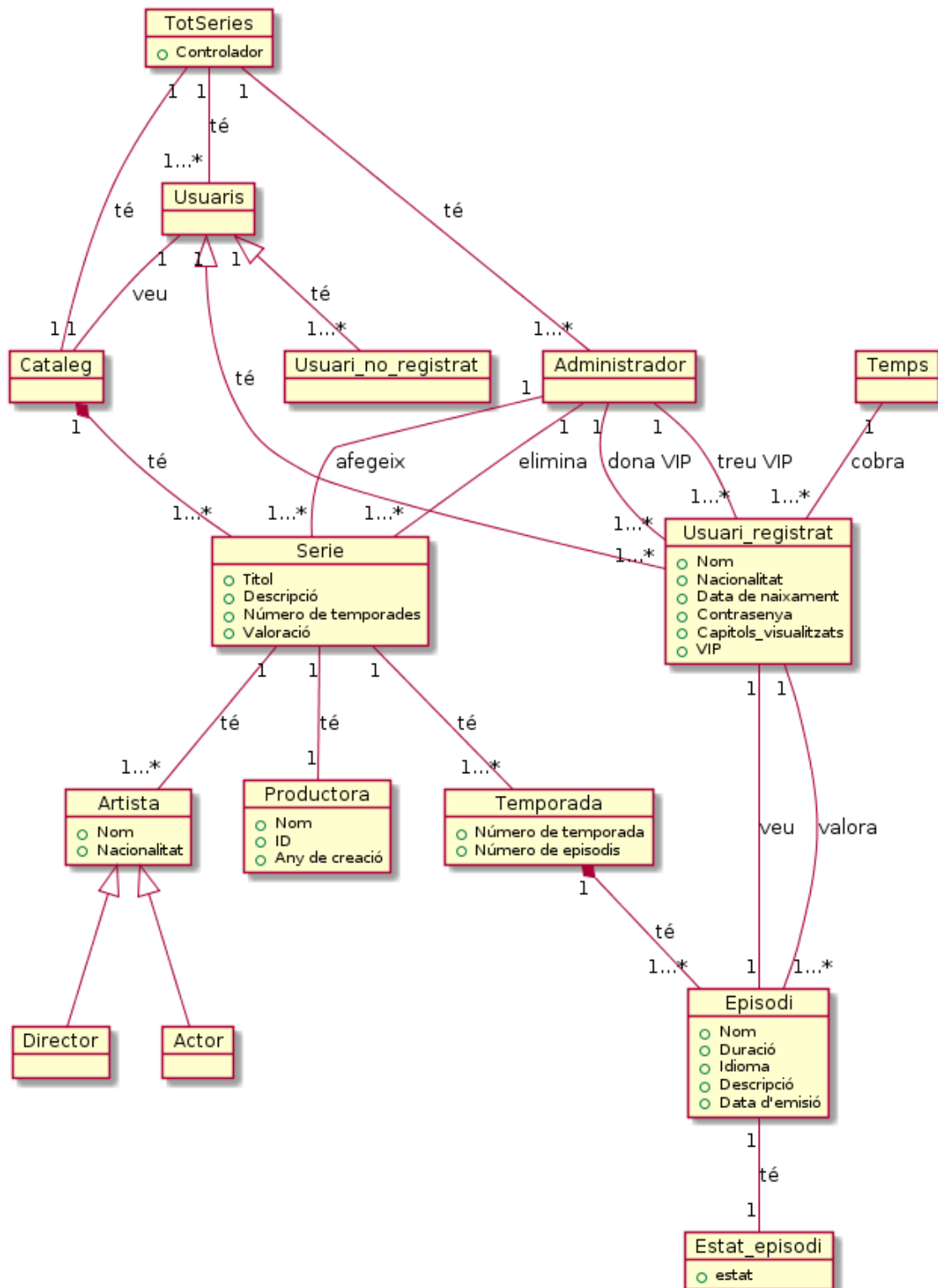
Descripció: UC7.1: Treure Sèries, Temporades, Capítols.
Actors: Administrador
Precondicions: L'administrador ha d'estar loguejat.
Flux Bàsic: 1) L'administrador decideix eliminar un/a sèrie/temporada/capítol. 2) Selecciona el que vol eliminar i l'esborra del catàleg.
Flux Alternatiu: 1a) En el cas de que no hi hagi cap sèrie dins de l'aplicació es mostrarà un missatge per pantalla dient que no hi ha cap sèrie.
Postcondicions: S'elimina sèrie/temporada/capítol.

Model de domini

En el model de domini trobem la relació entre tots els elements que volem relacionar dins la nostra aplicació.

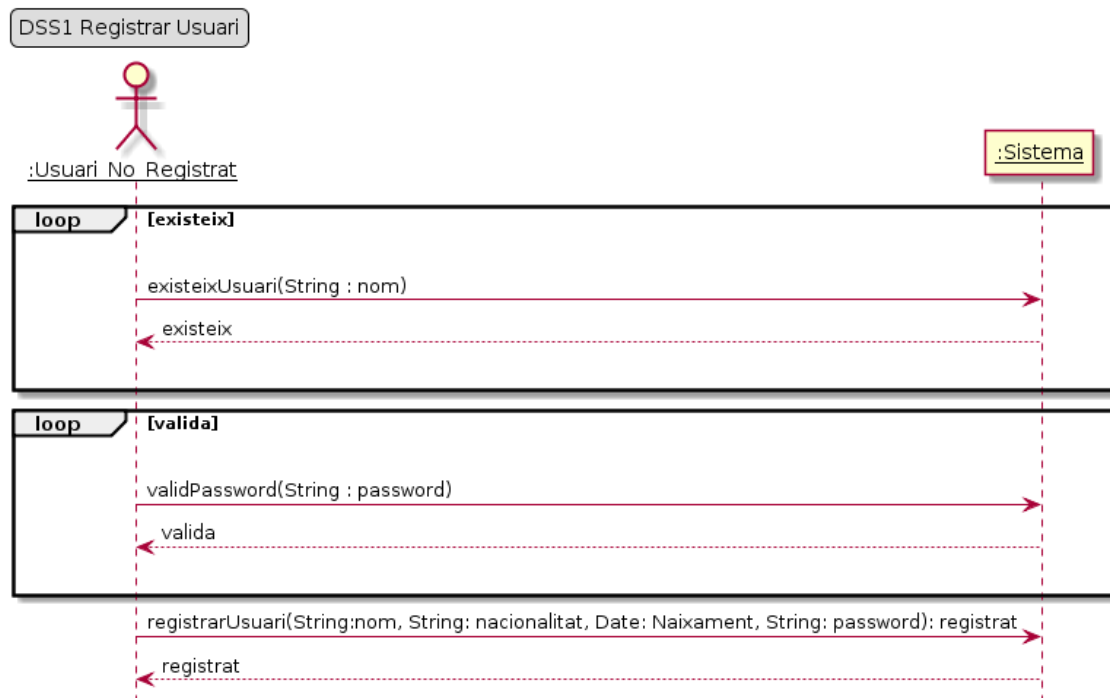
També afegim els atributs que tindran les nostres classes un cop comencem amb la programació.

TotSeries - Model de domini



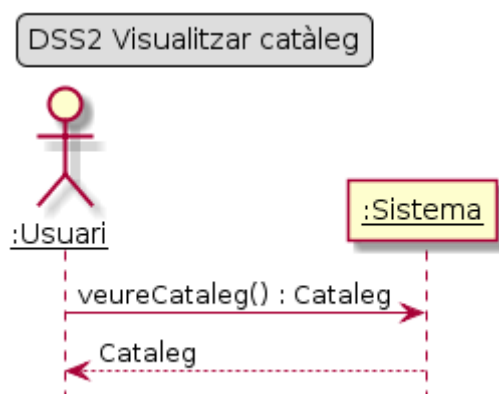
DSS

DSS 1: Registrar usuari



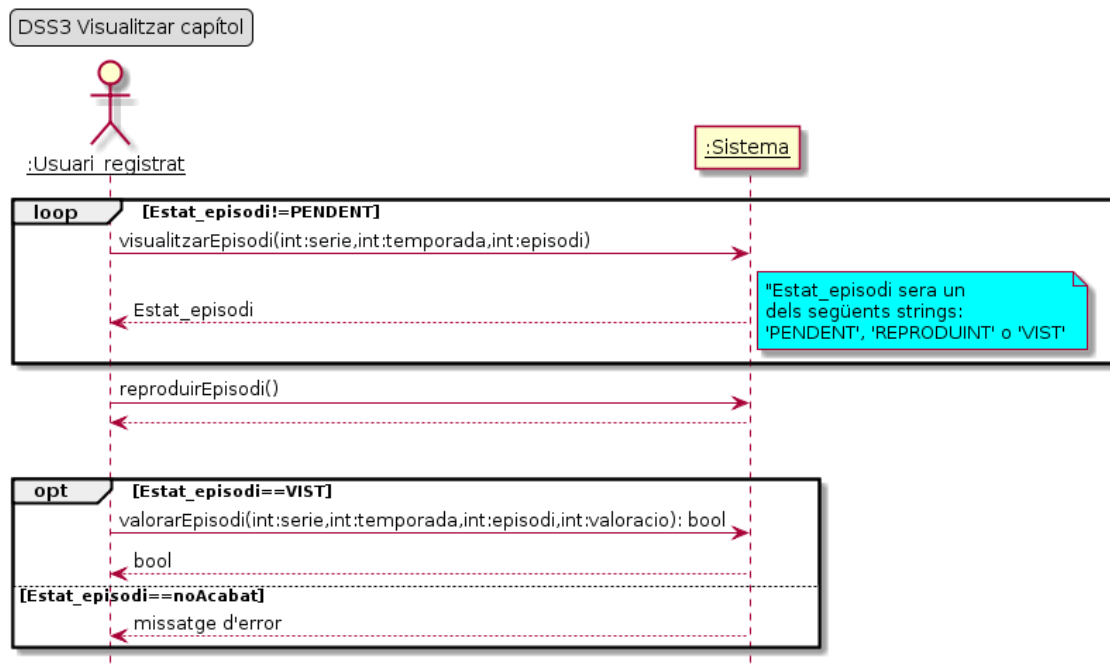
En el DSS1 es mostra que faria l'usuari no registrat per tal de poder registrar-se dins de l'aplicació.

DSS 2: Visualitzar catàleg



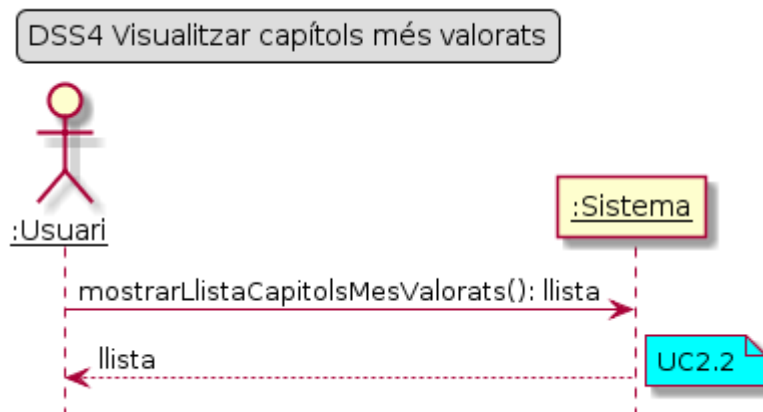
Aquest DSS mostra com es faria el procés de mostrar el catàleg a l'usuari, S'ha d'aclarir que l'usuari no té perquè estar registrat ja que el catàleg es mostra al qualsevol per igual però en el cas de no estar registrat no es podrà accedir a altres funcions com per exemple reproduir un episodi o valorar-lo.

DSS 3: Visualitzar i valorar capítol



Aquest DSS mostra com funciona l'opció de reproduir un episodi i posteriorment valorar-lo. Primer l'usuari escull l'episodi que vol visualitzar i el reproduceix. Una vegada l'ha acabat el pot valorar. En el cas de que vulgui valorar-lo abans de haver acabat l'episodi el sistema li retornarà un missatge d'error.

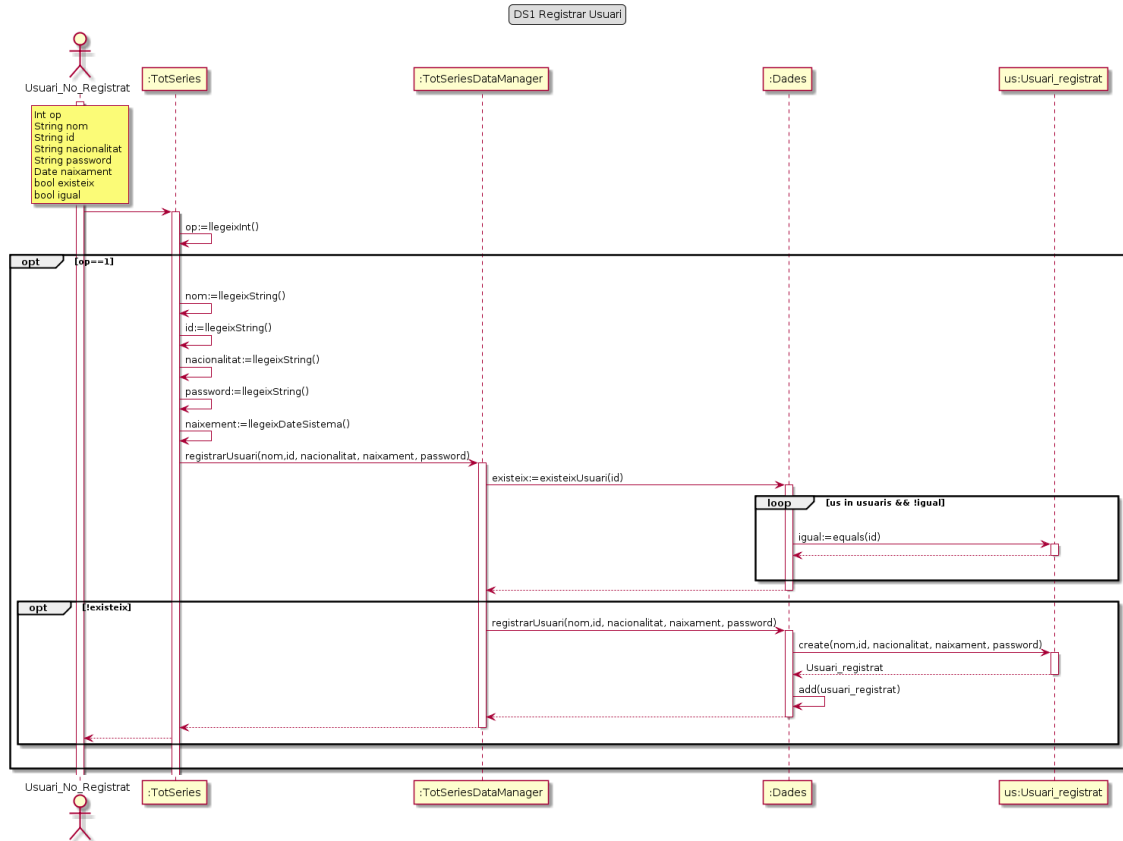
DSS 4: Visualitzar capítols més valorats



Aquest DSS mostra com és el procés de mostrar la llista de tots els capítols millor valorats pels usuaris que ja han visualitzat els episodis.

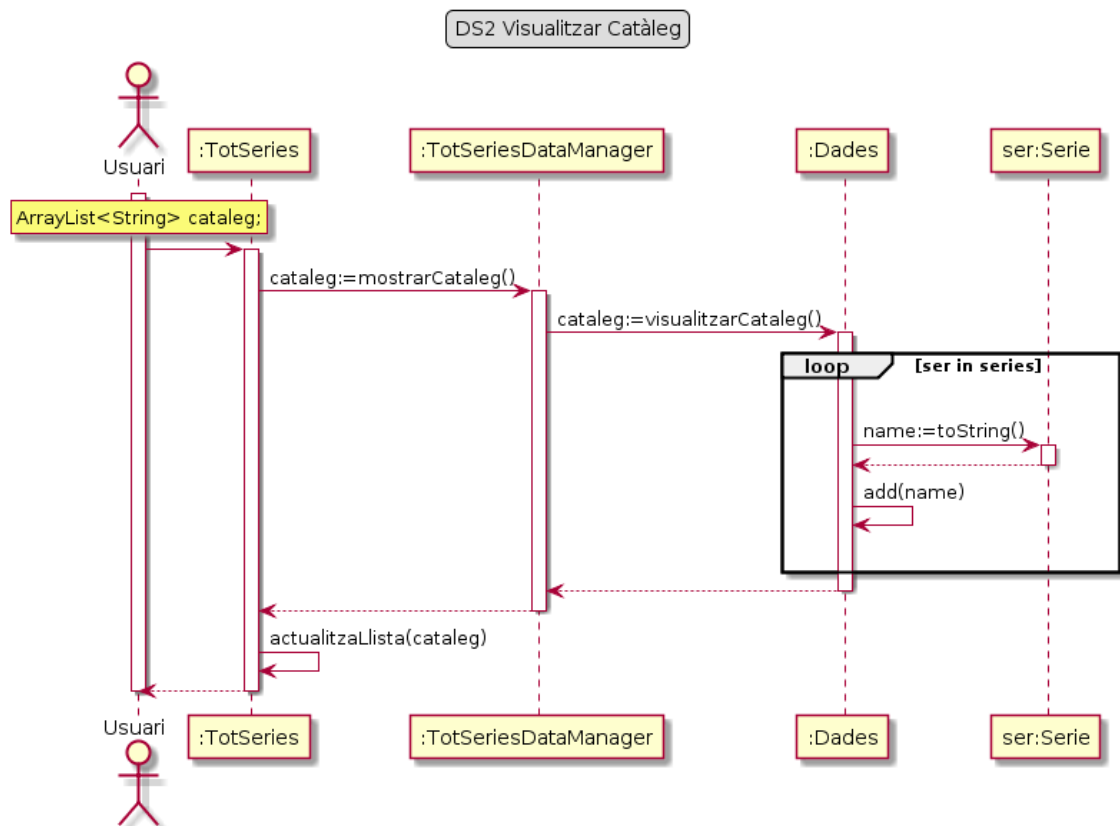
DS

DS 1: Registrar usuari



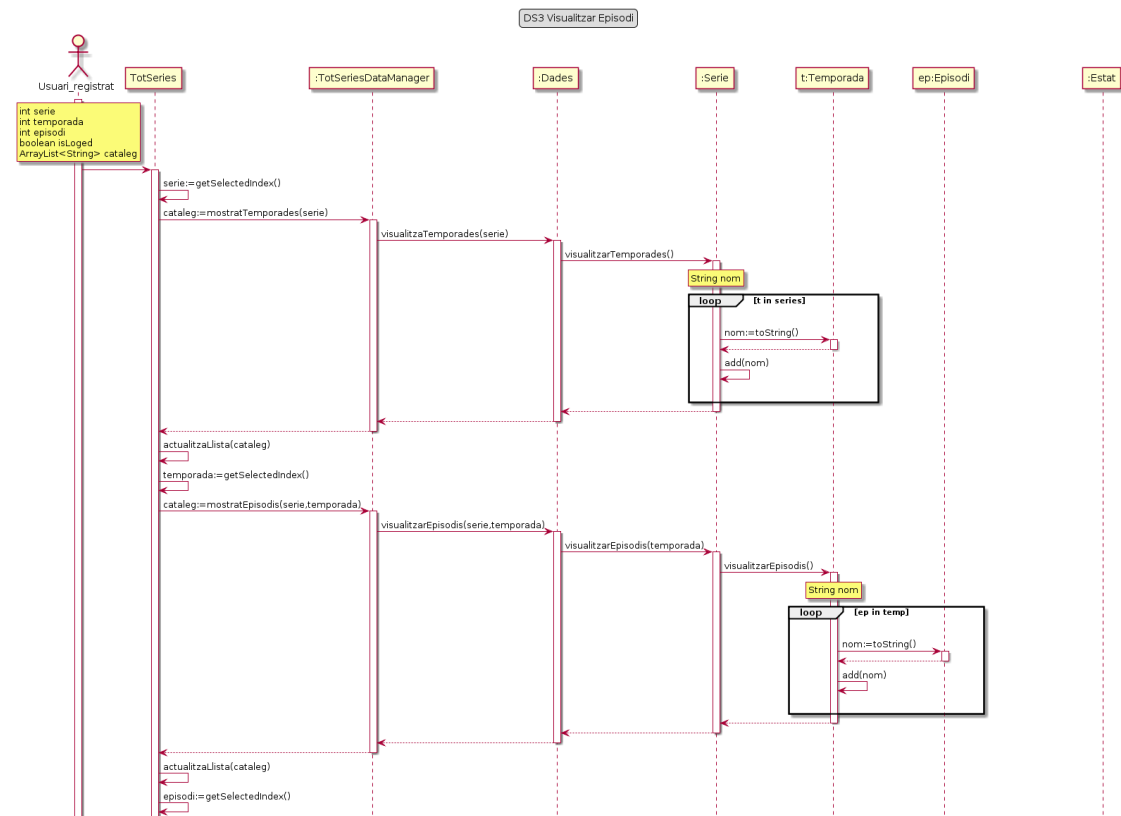
El DS1 mostra el procés que segueix el sistema sencer quan l'usuari, que en aquest cas encara no està registrat, vol registrar-se dins de l'aplicació per poder accedir a les diferents funcionalitats que aquesta té i que mentre sigui no registrat no podrà utilitzar (com per exemple reproduir un episodi).

DS 2: Visualitzar catàleg

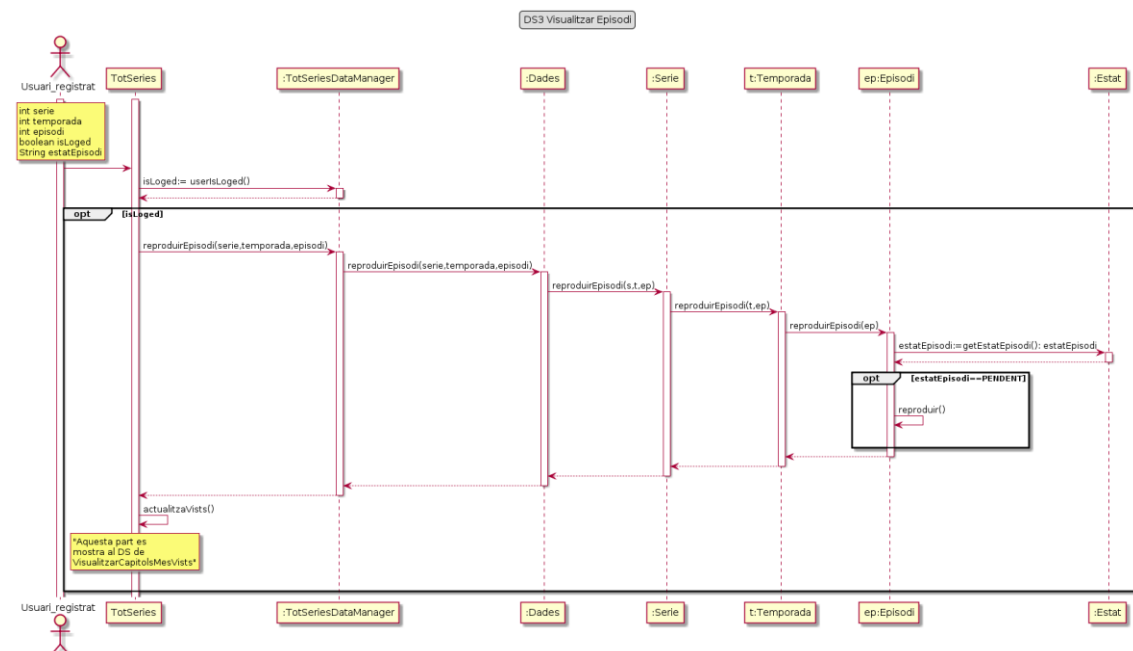


El DS2 és el procés que segueix l'usuari quan vol visualitzar el catàleg. En primer lloc l'usuari escull l'opció dins del menú que s'encarrega de mostrar el catàleg. En aquest cas l'usuari no té perquè estar registrat ja que no importa perquè igualment el catàleg es mostrarà.

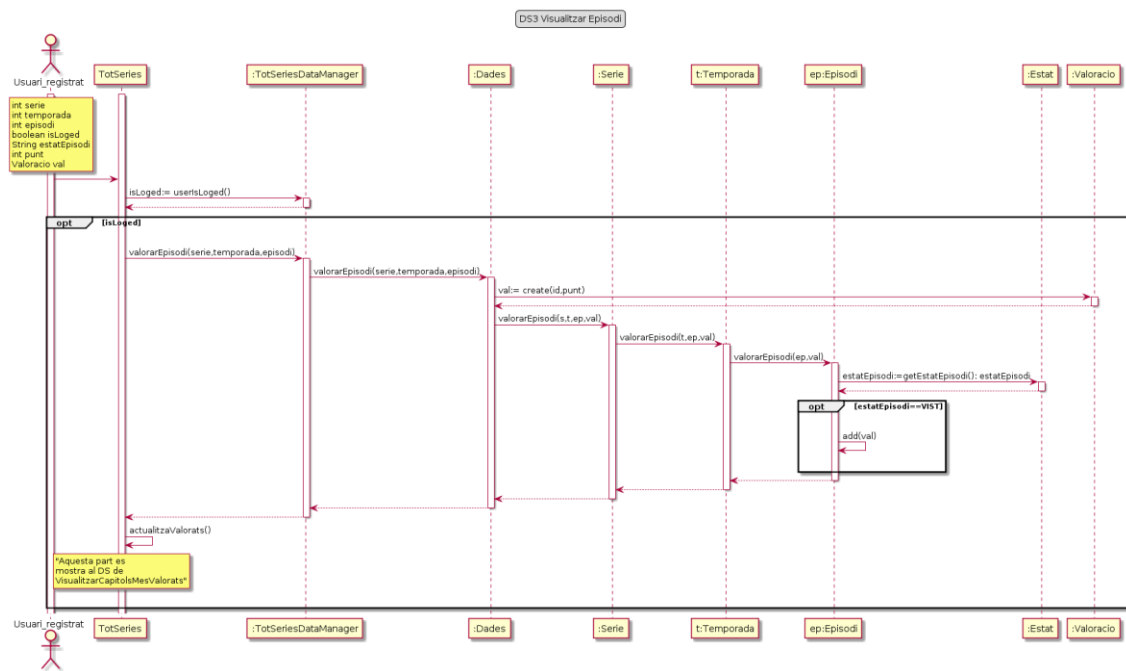
DS 3: Visualitzar i valorar capítol



Escollim la sèrie, temporada i episodi.

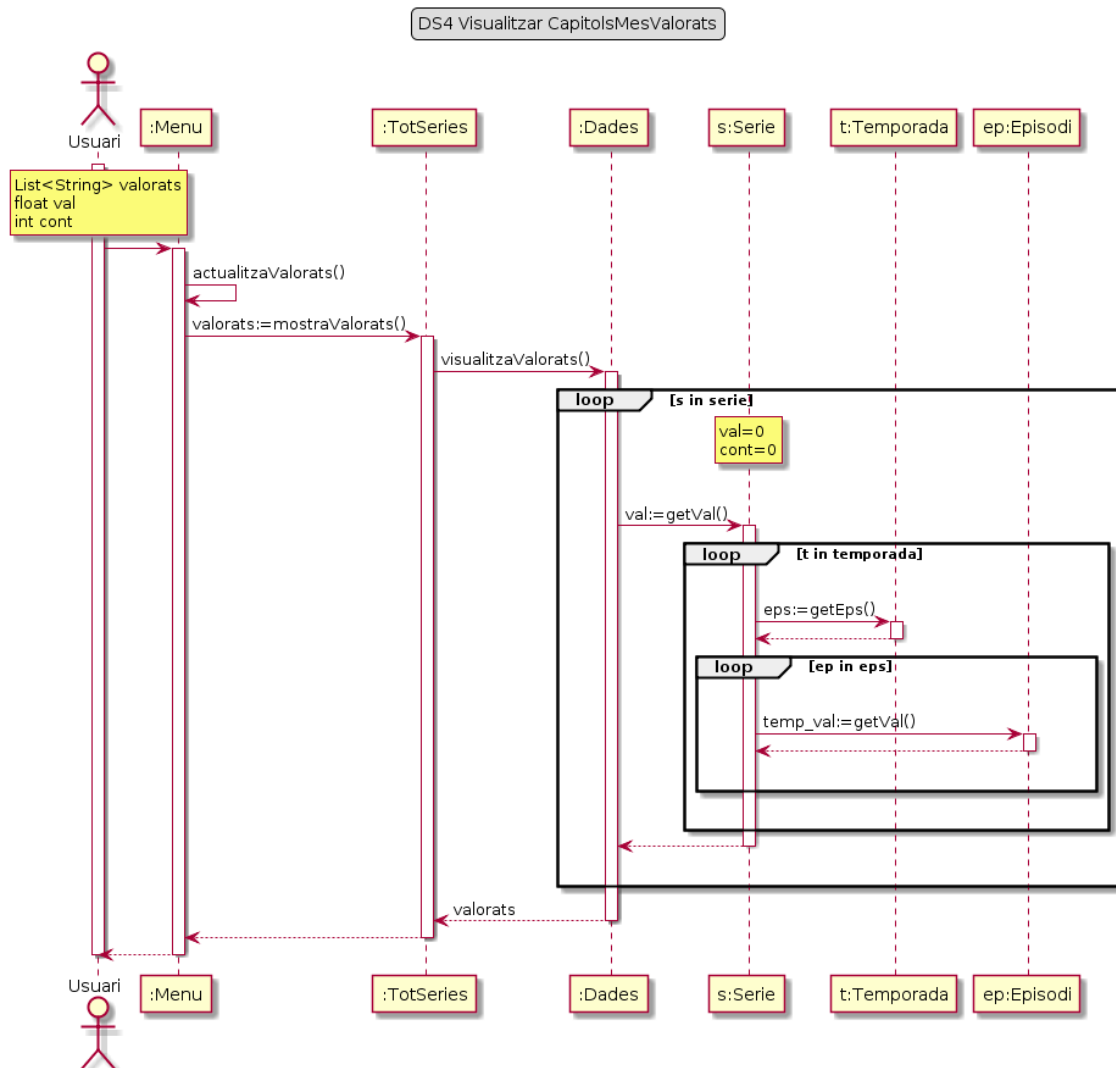


Reproduïm l'episodi i actualitzem la llista de més vists.



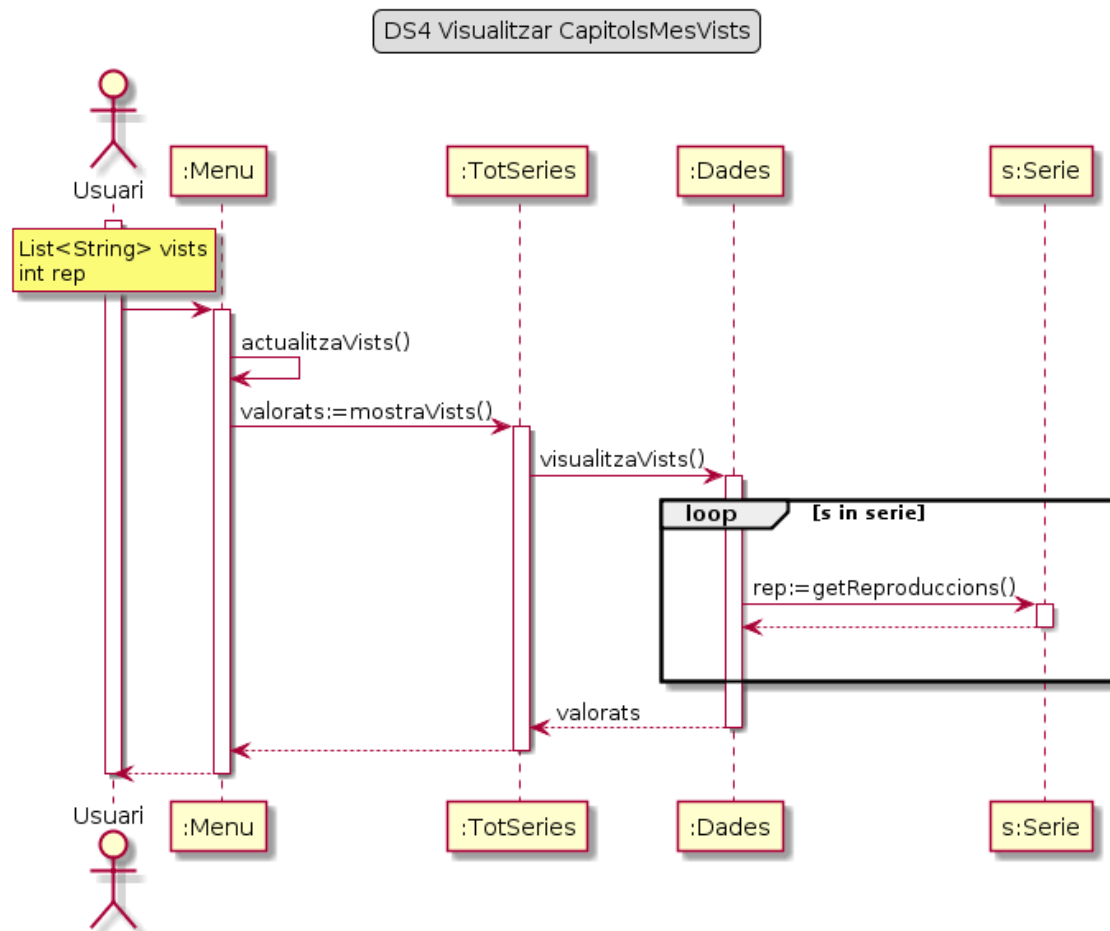
En aquest DS l'usuari el que vol fer es visualitzar un episodi i valorar-lo. Primer de tot escull l'episodi que vol veure i el reproduïx. Mentre l'usuari no ha acabat de veure el capítol no podrà valorar-lo però en el moment en que si 'hagi acabat llavors tindrà accés a poder valorar el capítol.

DS 4-1: Visualitzar capítols més valorats



Aquest DS mostra com és el procés de mostrar la llista de tots els capítols millor valorats pels usuaris que ja han visualitzat els episodis.

DS 4-2: Visualitzar capítols més vists



Amb el model de domini podem trobar la relació entre tots els elements que volem relacionar dins la nostra aplicació i per tant la podem utilitzar per fer el DCD on només afegirem els atributs que contenen les nostres classes un cop comencem amb la programació.



Observacions

Tot i l'aparent simplicitat de l'enunciat, hem acabat trobant-nos amb un gran nombre de factors amb els que no comptàvem.

Tant el model de domini com el DCU han acabat mostrant un tamany força considerable un cop hem afegit tots els elements que consideràvem rellevants per al problema.

Pel que fa als DSS no han sigut molt llargs ja que mostraven una simplicitat considerable en comparació amb els DS o el DCD. Pel que fa a aquests dos, han acabat sent massa grans i dins de l'informe les imatges d'aquests no es poden visualitzar adequadament perquè les lletres que contenen aquests són massa petites com per poder veure-les amb claredat.

Entrant amb el codi, encara que al principi semblava ja bastant complicat a l'hora de completar-ho ha demostrat ser més difícil del que es pensava i hem acabat necessitant més temps del que havíem pensat que necessitaria, sobretot a la part de dissenyar la interfície perquè havíem de fer un disseny que ha de permetre a l'usuari realitzar totes les accions que se'ns ha demanat com pot ser valorar un episodi després d'haver-lo reproduït. També una part difícil ha sigut implementar els patrons de tal manera que seguissin els DS.

A part també a l'hora de fer les llistes on apareixen les series més valorades i més reproduïdes perquè hem hagut de fer diferents mètodes que recorren la llista amb les series proporcionades per Parser i al mateix temps també les temporades i episodis per tal de poder agafar les valoracions i el nombre de reproduccions per tal de fer una mitja de cadascuna i posteriorment mostrar per pantalla en la JList les series juntament amb el seu total de reproduccions i la mitjana de totes les valoracions dels usuaris, valorada entre 0 i 5.

Respecte als patrons que hem utilitzat són: el Singleton, el patró expert i el patró factoria entre les diferents classes com pot ser la classe Dades.java i han sigut aplicats per tal de millorar la eficiència a l'hora de fer els diagrames, aconseguint que aquests puguin ser entesos d'una manera més senzilla i poder passar-los a format de codi amb més facilitat.

Conclusions

En aquesta pràctica hem començat amb l'anàlisi de les exigències de l'empresari a l'hora de realitzar l'aplicació TotSeries, realitzant els diagrames de casos d'us, juntament amb els casos d'us textual, fent servir l'eina gratuïta PlantText.

Pel que fa el model de domini, hem anat més enllà del DCU, donant major forma a la nostra aplicació, mostrant les classes involucrades, els atributs de les mateixes i els elements que estaran relacionats un cop comencem amb la programació.

En el cas dels DSS no ha estat massa difícils de fer perquè només s'havia de exposar com era el comportament entre l'usuari i el sistema, és a dir, quins mètodes cridaria l'usuari per tal de que el sistema compleixi amb la sol·licitud que aquest li envia i que el sistema li retorni el que ha demanat.

Pels DS, com ja teníem els DSS fets només hem hagut de aprofundir una mica més en detalls afegint totes les interaccions que fa el sistema amb les classes que té aquest per tal de retornar a l'usuari allò que ha sol·licitat.

Amb el DCD hem aprofundit encara més dins de la nostra aplicació perquè hem mostrat com seria la relació de codi entre les diferents classes i paquets del projecte i quins atributs i mètodes contenen aquestes classes.

Hem implementat el DCD a codi utilitzant el Parser que se'ns ha proporcionat i hem fet una interfície adequada amb la que l'usuari podrà interactuar amb les diferents funcions que ha de complir l'aplicació com per exemple registrar-se o visualitzar el catàleg. En aquesta interfície hem utilitzat 3 JLists: una per visualitzar totes les series del catàleg juntament amb les temporades i episodis corresponents, un altre per mostrar les series en les que els seus episodis són més reproduïts i l'última per mostrar les series millor valorades pels usuaris. Juntament amb aquestes JList hem creat dos botons: un que dóna l'opció al usuari de registrar-se i l'altre permet loguejar-se a un usuari que prèviament s'ha registrat. Cadascun d'aquests botons crida un JDialog on l'usuari haurà d'introduir les dades corresponents.

En el cas del JDialog que correspon al registre l'usuari té uns JTextFields on podrà introduir les seves dades que són el seu nom, username (d'aquí en endavant ID), DNI, nacionalitat, data de naixement i també un JPasswordField on introduirà la contrasenya. Al final d'aquest diàleg es troba un botó que finalitza el registre creant el nou usuari que es guardarà en una llista d'usuaris de tipus Usuari_Registrat i es desactivarà el botó que permet el registre del JFrame principal.

L'altre JDialog només conté un JTextField i un JPasswordField on l'usuari introduirà el seu ID i la seva contrasenya respectivament. A sota d'aquests hi ha dos botons que permetran a l'usuari loguejar-se o cancel·lar el log in. En el cas de que si faci el log in llavors es desactivarà el botó que permet a l'usuari loguejar-se i on abans posava "Login" ara hi apareixerà la ID del usuari.

Codis

DCU

@startuml

title DCU TotSeries

left to right direction

actor Usuari as Usuari

actor Temps as Temps

actor Administrador as Admin

actor Usuari_no_registrat as UsuariNR
@startuml

title DCU TotSeries

left to right direction

actor Usuari as Usuari

actor Temps as Temps

actor Administrador as Admin

actor Usuari_no_registrat as UsuariNR

Rectangle {
usecase (UC1. Login) as UC1

usecase (UC1.1 Registrar-se) as UC1.1

usecase (UC1.2 Gestió Client VIP/Normal) as UC1.2

usecase (UC2. Catàleg de series) as UC2

usecase (UC2.1. Mostrar preferències dels usuaris) as UC21

usecase (UC2.2. Mostrar Capítol millor valorat pels usuaris) as UC22

usecase (UC2.3 Selecció de Serie, Temporada i Episodi) as UC23

usecase (UC2.4 Mostrar resum breu de series i altres detalls) as UC24

usecase (UC3. Descarrega/Visualització de capítols) as UC3

usecase (UC3.1 Valorar Capítol) as UC31

usecase (UC4 Realitzar pagament del import total) as UC4

usecase (UC5 Guardar dades dels usuaris) as UC5

usecase (UC5.1 Esborrar dades dels usuaris) as UC5.1

usecase (UC6 Tancar Sessió) as UC6

usecase (UC7 Afegir Series,Temporades,Capítols) as UC7

usecase (UC7.1 Treure Series,Temporades,Capítols) as UC7.1

}

Usuari -- UC1

Usuari -- UC2

Usuari -- UC6

UsuariNR -- UC1.1

UsuariNR -- UC2

Temps - UC4

Admin -- UC1

Admin -- UC1.2

Admin -- UC5

Admin -- UC5.1

Admin - UC6

Admin -- UC7

Admin -- UC7.1

UC3 <.. UC24 : <<include>>

UC31 <. UC3: <<extends>>

UC2 <... UC21: <<extends>>

UC2 <... UC22: <<extends>>

UC2 <... UC23: <<extends>>

UC2 <... UC24: <<extends>>

@enduml

Model de domini

@startuml

title TotSeries - Model de domini

```
object TotSeries{
    +Controlador
}
```

```
object Cataleg{
}
```

```
object Usuaris{
}
```

```
object Temps{
}
```

```
object Serie {
    +Titol
    +Descripció
    +Número de temporades
    +Valoració
}
```

```
object Usuari_registrat {
    +Nom
    +Nacionalitat
    +Data de naixement
    +Contrasenya
    +Capitols_visualitzats
    +VIP
}
```

```
object Usuari_no_registrat {
}
```

```
object Artista{
    +Nom
    +Nacionalitat
}
```

```
object Director{
}
```

```
object Actor{
}
```

```
object Productora{
    +Nom
    +ID
    +Any de creació
}
```

```
object Temporada{
    +Número de temporada
    +Número de episodis
}
```

```
object Episodi{
    +Nom
    +Duració
    +Idioma
    +Descripció
    +Data d'emissió
}
```

```
object Administrador{
}
```

TotSeries "1" -down- "1" Cataleg: té
TotSeries "1" -down- "1..." Usuaris: té
TotSeries "1" -down- "1..." Administrador: té

Cataleg "1" *-down- "1..." Serie: té

Usuaris "1" <|-down- "1..." Usuari_registrat: té
Usuaris "1" <|-down- "1..." Usuari_no_registrat: té

Serie "1" -down- "1..." Artista: té
Serie "1" -down- "1" Productora: té
Serie "1" -down- "1..." Temporada: té

Artista <|-down- Actor
Artista <|-down- Director

Temporada "1" *-down- "1..." Episodi: té

Usuari_registrat "1" -down- "1" Episodi: veu
Usuari_registrat "1" -down- "1" Cataleg: veu
Usuari_registrat "1" -down- "1..." Episodi: valora

Usuari_no_registrat "1" -down- "1" Cataleg: veu

Administrador "1" -down- "1..." Serie: afegeix
Administrador "1" -down- "1..." Usuari_registrat:
dona VIP
Administrador "1" -down- "1..." Serie: elimina
Administrador "1" -down- "1..." Usuari_registrat:
treu VIP

Temps "1" -down- "1..." Usuari_registrat: cobra

@enduml

DSS

DSS1

```
@startuml
hide footbox
skinparam sequenceParticipant underline
skinparam noteBackgroundColor aqua
legend top left
    DSS1 Registrar Usuari
end legend
actor ":Usuari_No_Registrat" as usuari
participant ":Sistema" as sistema
loop existeix
    |||
    usuari -> sistema : existeixUsuari(String :
nom)
    sistema --> usuari : existeix
```

```
    |||
end
loop valida
    |||
    usuari -> sistema : validPassword(String :
password)
    sistema --> usuari : valida
    |||
End
usuari -> sistema :
registrarUsuari(String:nom, String:
nacionalitat, Date: Naixament, String:
password): registrat
sistema --> usuari : registrat
@enduml
```

DSS2

```
@startuml
hide footbox
skinparam sequenceParticipant underline
skinparam noteBackgroundColor aqua
legend top left
    DSS2 Visualitzar catàleg
end legend
actor ":Usuari" as usuari
participant ":Sistema" as sistema

usuari -> sistema : veureCataleg() : Cataleg
usuari <--- sistema : Cataleg
@enduml
```

DSS3

@startuml

hide footbox

skinparam sequenceParticipant underline

skinparam noteBackgroundColor aqua

legend top left

DSS3 Visualitzar capítol

end legend

actor ":Usuari_registrat" as usuari

participant ":Sistema" as sistema

loop Estat_episodi!=PENDENT

usuari -> sistema : visualitzarEpisodi(int:serie,int:temporada,int:episodi)

usuari <--- sistema : Estat_episodi

note right : "Estat_episodi sera un\ndels següents strings:\n'PENDENT', 'REPRODUINT' o 'VIST'

end

usuari -> sistema : reproduirEpisodi()

sistema --> usuari

|||

opt Estat_episodi==VIST

usuari -> sistema : valorarEpisodi(int:serie,int:temporada,int:episodi,int:valoracio): bool

usuari <--- sistema : bool

else Estat_episodi==noAcabat

usuari <--- sistema : missatge d'error

end

@enduml

DSS4

@startuml

hide footbox

skinparam sequenceParticipant underline

skinparam noteBackgroundColor aqua

legend top left

DSS4 Visualitzar capítols més valorats

end legend

actor ":Usuari" as usuari

participant ":Sistema" as sistema

usuari -> sistema : mostrarLlistaCapitolsMesValorats(): llista

usuari <--- sistema : llista

note right : UC2.2

@enduml

DS

DS1

@startuml

legend top

DS1 Registrar Usuari

end legend

actor Usuari_No_Registrat as usuari

activate usuari

rnote over usuari

String nom

String id

String nacionalitat

String password

Date naixament

bool existeix

end rnote

participant ":Menu" as menu

participant ":TotSeries" as TS

participant ":Dades" as dades

participant ":Usuari_registrat" as us

usuari -> menu

activate menu

menu -> menu: nom:=llegeixString()

menu -> menu: id:=llegeixString()

menu -> menu: nacionalitat:=llegeixString()

menu -> menu: password:=llegeixString()

menu -> menu:

naixement:=llegeixDateSistema()

menu -> TS: registrarUsuari(nom,id,
nacionalitat, naixament, password)

activate TS

TS -> dades : existeix:=existeixUsuari(id)

activate dades

dades ---> TS

deactivate dades

opt !existeix

TS -> dades : registrarUsuari(nom,id,
nacionalitat, naixament, password)

activate dades

dades -> us : create(nom,id, nacionalitat,
naixament, password)

activate us

us ---> dades : Usuari_registrat

deactivate us

dades ---> TS

deactivate dades

TS ---> menu

deactivate TS

menu ---> usuari

end

@enduml

DS2

@startuml

legend top

DS2 Visualitzar Catàleg

end legend

actor Usuari as usuari

activate usuari

rnote over usuari

Int op

end rnote

participant ":Menu" as menu

participant ":TotSeries" as TS

participant ":Dades" as dades

participant "ser:Serie" as serie

usuari -> menu

activate menu

menu -> menu : op:=llegeixInt()

opt op==2

|||

menu -> TS: visualitzarCataleg()

activate TS

TS -> dades: visualitzarCataleg()

activate dades

loop ser in series

|||

dades -> serie : name:=toString()

activate serie

serie --> dades

deactivate serie

dades -> menu : escriu(name)

menu ---> dades

|||

end

dades --> TS

deactivate dades

TS ---> menu

deactivate TS

menu --> usuari

deactivate menu

deactivate usuari

|||

end

@enduml

DS3.1

@startuml

legend top

DS3 Visualitzar Episodi

end legend

actor Usuari_registrat as usuari

activate usuari

rnote over usuari

int serie

int temporada

int episodi

boolean isLogged

ArrayList<String> cataleg

end rnote

participant "TotSeries" as menu

participant "TotSeriesDataManager" as TS

participant "Dades" as dades

participant "Serie" as serie

participant "t:Temporada" as temp

participant "ep:Episodi" as episodi

participant "Estat" as estat

usuari -> menu

activate menu

menu -> menu: serie:=getSelectedIndex()

menu -> TS :

cataleg:=mostratTemporades(serie)

activate TS

TS -> dades: visualitzaTemporades(serie)

activate dades

dades -> serie : visualitzarTemporades()

activate serie

rnote over serie

String nom

end rnote

loop t in series

|||

serie->temp : nom:=toString()

activate temp

temp --> serie

deactivate temp

serie -> serie: add(nom)

|||

end

serie --> dades

deactivate serie

dades --> TS

deactivate dades

TS-->menu

deactivate TS

menu -> menu:actualitzaLlista(cataleg)

menu -> menu: temporada:=getSelectedIndex()

menu -> TS :

cataleg:=mostratEpisodis(serie,temporada)

activate TS

TS -> dades:

visualitzarEpisodis(serie,temporada)

activate dades

dades -> serie : visualitzarEpisodis(temporada)

activate serie

serie->temp:visualitzarEpisodis()

activate temp

rnote over temp

String nom

end rnote

loop ep in temp

|||

temp->episodi : nom:=toString()

activate episodi

episodi --> temp

deactivate episodi

temp -> temp: add(nom)

|||

end

temp --> serie

deactivate temp

serie --> dades

deactivate serie

dades --> TS

deactivate dades

TS-->menu

deactivate TS

menu -> menu:actualitzaLlista(cataleg)

menu -> menu: episodi:=getSelectedIndex()

@enduml

DS3.2

@startuml

legend top

DS3 Visualitzar Episodi

end legend

actor Usuari_registrat as usuari

activate usuari

rnote over usuari

int serie

int temporada

int episodi

boolean isLogged

ArrayList<String> catalog

String estatEpisodi

end rnote

participant "TotSeries" as menu

participant ":TotSeriesDataManager" as TS

participant ":Dades" as dades

participant ":Serie" as serie

participant "t:Temporada" as temp

participant "ep:Episodi" as episodi

participant ":Estat" as estat

activate menu

usuari -> menu

menu->TS :isLogged:= userIsLogged()

activate TS

TS-->menu

deactivate TS

opt isLogged

|||

menu ->TS

:reproduirEpisodi(serie,temporada,episodi)

activate TS

TS-

>dades:reproduirEpisodi(serie,temporada, episodi)

activate dades

dades -> serie: reproduirEpisodi(s,t,ep)

activate serie

serie->temp: reproduirEpisodi(t,ep)

activate temp

temp->episodi: reproduirEpisodi(ep)

activate episodi

episodi -> estat:

estatEpisodi:=getEstatEpisodi():

estatEpisodi

activate estat

estat --> episodi

deactivate estat

opt estatEpisodi==PENDENT

|||

episodi ->episodi: reproduir()

|||

end

episodi --->temp

deactivate episodi

temp ---> serie

deactivate temp

serie---> dades

deactivate serie

dades---> TS

deactivate dades

TS ---> menu

deactivate TS

menu -> menu : actualitzaVists()

rnote over menu

"Aquesta part es

mostra al DS de

VisualitzarCapitolsMesVists"

end rnote

|||

end

@enduml

DS3.3

@startuml

legend top

DS3 Visualitzar Episodi

end legend

actor Usuari_registrat as usuari

activate usuari

rnote over usuari

int serie

int temporada

int episodi

boolean isLogged

String estatEpisodi

int punt

Valoracio val

end rnote

participant "TotSeries" as menu

participant ":TotSeriesDataManager" as TS

participant ":Dades" as dades

participant ":Serie" as serie

participant "t:Temporada" as temp

participant "ep:Episodi" as episodi

participant ":Estat" as estat

participant ":Valoracio" as val

activate menu

usuari -> menu

menu->TS :isLogged:= userIsLogged()

activate TS

TS-->menu

deactivate TS

opt isLogged

|||

menu ->TS

:valorarEpisodi(serie,temporada,episodi)

activate TS

TS-

>dades:valorarEpisodi(serie,temporada,epi
sodi)

activate dades

dades -> val: val:= create(id,punt)

activate val

val --> dades

deactivate val

dades -> serie: valorarEpisodi(s,t,ep,val)

activate serie

serie->temp: valorarEpisodi(t,ep,val)

activate temp

temp->episodi: valorarEpisodi(ep,val)

activate episodi

episodi -> estat:

estatEpisodi:=getEstatEpisodi():

estatEpisodi

activate estat

estat --> episodi

deactivate estat

opt estatEpisodi==VIST

|||

episodi ->episodi: add(val)

|||

end

episodi --->temp

deactivate episodi

temp ---> serie

deactivate temp

serie---> dades

deactivate serie

dades---> TS

deactivate dades

TS ---> menu

deactivate TS

menu -> menu : actualitzaValorats()

rnote over menu

"Aquesta part es

mostra al DS de

VisualitzarCapitolsMesValorats"

end rnote

|||

end

@enduml

DS4.1

@startuml

legend top

DS4 Visualitzar CapitolsMesValorats

end legend

actor Usuari as usuari

activate usuari

rnote over usuari

List<String> valorats

float val

int cont

end rnote

participant ":Menu" as menu

participant ":TotSeries" as TS

participant ":Dades" as dades

participant "s:Serie" as serie

participant "t:Temporada" as temporada

participant "ep:Episodi" as episodi

usuari -> menu

activate menu

menu -> menu : actualitzaValorats()

menu->TS : valorats:=mostraValorats()

activate TS

TS -> dades : visualitzaValorats()

activate dades

loop s in serie

rnote over serie

val=0

cont=0

end rnote

|||

dades -> serie : val:=getVal()

activate serie

loop t in temporada

|||

serie->temporada: eps:=getEps()

activate temporada

temporada --> serie

deactivate temporada

loop ep in eps

|||

serie -> episodi : temp_val:=getVal()

activate episodi

episodi --> serie

deactivate episodi

|||

end

|||

end

serie ---> dades

deactivate serie

|||

end

dades -->TS : valorats

deactivate dades

TS --> menu

deactivate TS

menu --> usuari

deactivate menu

deactivate usuari

@enduml

DS4.2

@startuml

legend top

DS4 Visualitzar CapitolsMesVists

end legend

actor Usuari as usuari

activate usuari

rnote over usuari

List<String> vists

int rep

end rnote

participant ":Menu" as menu

participant ":TotSeries" as TS

participant ":Dades" as dades

participant "s:Serie" as serie

usuari -> menu

activate menu

menu -> menu : actualitzaVists()

menu->TS : valorats:=mostraVists()

activate TS

TS -> dades : visualitzaVists()

activate dades

loop s in serie

|||

dades -> serie : rep:=getReproduccions()

activate serie

serie ---> dades

deactivate serie

|||

end

dades -->TS : valorats

deactivate dades

TS --> menu

deactivate TS

menu --> usuari

deactivate menu

deactivate usuari

@enduml

DCD

```
@startuml
title TotSeries
skinparam classAttributeIconSize 0
```

```
namespace java.util {
    class Timer
}
```

```
namespace Vista {
    class TotSeries {
        -instance : TotSeries
        -dades :Dades
        -cont:TotSeriesDataManager
        -opt : ArrayList<String>
        -place : int
        -serie:int
        -temporada:int
        -episodi:int
        -last:int

        +RegisterActionPerformed(ActionEvent)
        +LoginActionPerformed(ActionEvent)
        +turnOffLogin()
        +catalogValueChanged(ActionEvent)

        +actualitzaLlista(lIista:ArrayList<String>)
        +actualitzaValorats()
        +actualitzaVists()
        +changeLogin()
    }
}
```

```
namespace Controlador {
    class TotSeriesDataManager {
        -instance:TotSeriesDataManager
        -data:Dades
        +obtenirDades(nomFitxer:String)
        +crearSerie(String:id, String: title,
String: desc)
        +crearTemporada(String:
numTemporada, String: numEpisodis, int: i)
        +crearEpisodi(String :title, String:
duration, String :idioma, String
:description, String :data, int :i, int: j)
```

```
        +crearArtista(String id, String nom,
String tipus, String idSerie, String
nacionalitat)
        +crearProductora (String id, String
nom, String idSerie)
        +crearValoracio(String id, String client,
String episodi, String puntuacio, String
data)
        +crearAdmin(String id, String nom,
String usuari, String password)
        +crearClient(String nom, String dni,
String adreca, String usuari, String
password, String vip)
        +crearClient(String nom, String dni,
String adreca, String usuari, String
password)
        +reproduirEpisodi(int s, int t, int ep)
        +valorarEpisodi(int serie, int
temporada, int episodi, int valoracio)
        +mostratCataleg():ArrayList<String>
        +mostratTemporades(int
s):ArrayList<String>
        +mostratEpisodis(int s, int
t):ArrayList<String>
        +mostrarValorats():ArrayList<String>
        +mostrarVists():ArrayList<String>
        +userIsLogged():boolean
    }
}
```

```
class TotSeriesXMLParser{
    -dataManager: TotSeriesDataManager
    +parse(nomFitxer:String)
    +obtenirSeries(doc:Document)
    +obtenirProductora(eSerie:Element ,
idSerie:String)
    +obtenirArtistes(eSerie:Element,
idSerie:String)
    +obtenirTemporades(eSerie:Element,
idSerie:String, i:int)
    +obtenirEpisodis(
eTemporada:Element, idSerie:String, i:int,
j:int)
    +obtenirValoracions(doc:Document)
```

```

+obtenirAdministradors(doc:Document)
+obtenirClients(doc:Document)
}

```

```

TotSeriesDataManager <->

```

```

TotSeriesXMLParser

```

```

}

```

```

namespace Model {

```

```

class Dades{

```

```

-instance:Dades

```

```

-loggedUser:String

```

```

+existeixUsuari(id : String):boolean

```

```

+posUsuari(id : String):int

```

```

+comprovarPassword(pos:int,

```

```

pass:String):boolean

```

```

+visualitzarCatalog():ArrayList<String>

```

```

+visualitzarValorats() : list<String>

```

```

+visualitzarVists() : list<String>

```

```

+visualitzarTemporades(s:Int) :

```

```

list<String>

```

```

+visualitzarEpisodis(s:Int, t:Int) :

```

```

list<String>

```

```

+reproduirEpisodi(s:Int, t:Int, ep: Int)

```

```

+valorarEpisodi(s:int,t:int,ep:int,val:int)

```

```

+crearSerie(id:String,title:String,desc:String
)

```

```

+crearTemporada(String

```

```

numTemporada, String numEpisodis, int i)

```

```

+crearEpisodi(String title, String

```

```

duration, String idioma, String description,
String data, int i, int j)

```

```

+afegirAdmin(String id, String nom,
String usuari, String password)

```

```

+afegirArtista(String tipus, String id,
String nom, String idSerie, String
nacionalitat)

```

```

+afegirProductora(String id, String
nom, String idSerie)

```

```

+registrarUsuari(nom : id : String,
nacionalitat : String, naixament : Date,
password : String)

```

```

+registrarUsuari(String nom, String
dni, String adreca, String usuari, String
password)
}

```

```

abstract class Usuari{
}

```

```

class Usuari_registrat {

```

```

- nom : String

```

```

- usuari : String

```

```

- nacionalitat : String

```

```

- naixement : Date

```

```

- contrasenya : String

```

```

- visualitzat

```

```

- VIP : bool

```

```

}

```

```

class Usuari_no_registrat{
}

```

```

class Administrador{

```

```

-nom : String

```

```

-usuari : String

```

```

-contrasenya :String

```

```

+ferVip(usuari : Usuari_registrat) : bool

```

```

}

```

```

enum Estat_episodi{

```

```

-estat: int

```

```

+getEstatEpisodi() : String

```

```

-setEstatEpisodi(estat:String)

```

```

}

```

```

class Episodi{

```

```

-nom : String

```

```

-duracio : float

```

```

-idioma : String

```

```

-descripcio : String

```

```

-emisio : Date

```

```

-valoracio : float

```

```

+reproduirEpisodi()

```

```

+toString() : String

```

```

+reproduir()

```

```

+updateValoracio()

```

```

+valorarEpisodi(val:Valoracio)
}

class Temporada{
    -numero :int
    -episodis : int
    +toString() : String
    +visualitzarEpisodis() : list<String>
    +reproduirEpisodi(ep: Int)
    +valorarEpisodi(ep:int,val:int)
    +afegirEpisodi(temp:episodi
}

class Productora{
    -nom : String
    +ID : String
    +creacio : Date
}

class Serie{
    -titol : String
    -descripcio : String
    -temporades : Inintt
    +toString() : String
    +afegirTemporada(temp:Temporada)
    +afegirEpisodi(temp:Episodi, pos:int)
    +visualitzarTemporades() : list<String>
    +visualitzarEpisodis( t:Int) : list<String>
    +reproduirEpisodi(t:Int, ep: Int)
    +valorarEpisodi(t:int, ep:int,
val:Valoracio)
}

abstract class Artista{
    +nom : String
    +nacionalitat : String
}

```

```

class Director{
}

class Actor{
}

Dades "1"*--"1..." Serie
Serie "1" *-- "1..." Productora
Serie "1" *-- "1..." Temporada
Serie "1" *-- "1..." Artista
Temporada "1" *-- "1..." Episodi
Episodi "1" <--> "1" Estat_episodi

Dades "1" *--"1..." Usuari
Dades "1" *--"1..." Administrador
Usuari <|-- Usuari_registrat
Usuari <|-- Usuari_no_registrat
Administrador --> Usuari_registrat : dona
VIP

Usuari_registrat --> Episodi : valora
Usuari_registrat --> Episodi : visualitza
Usuari_registrat <--> Estat_episodi :
modifica

Artista <|-- Actor
Artista <|-- Director
}

Controlador.TotSeriesDataManager "1" <-->
Model.Dades : Consta
Controlador.TotSeriesDataManager "-ctrl
1" <--> "-menu 1" Vista.TotSeries : Veu
Model.Dades -> Vista.TotSeries : escriu
java.util.Timer "1" <- "-timer 1"
Controlador.TotSeriesDataManager : Usa
java.util.Timer "1" -> "-timer 1"
ModelUsuari_registrat : Cobra
@enduml

```