**DISSENY DE SOFTWARE**

**TOTSERIES**

**ITERACIO 2**

Angel Bergantiños – 16319144

Francisco Diaz – 16828405

Índex

[Models 3](#_Toc467965127)

[DSS 3](#_Toc467965128)

[DSS 1: Registrar usuari 3](#_Toc467965129)

[DSS 2: Visualitzar catàleg 3](#_Toc467965130)

[DSS 3: Visualitzar i valorar capítol 4](#_Toc467965131)

[DSS 4: Visualitzar capítols més valorats 4](#_Toc467965132)

[DS 5](#_Toc467965133)

[DS 1: Registrar usuari 5](#_Toc467965134)

[DS 2: Visualitzar catàleg 5](#_Toc467965135)

[DS 3: Visualitzar i valorar capítol 5](#_Toc467965136)

[DS 4: Visualitzar capítols més valorats 5](#_Toc467965137)

[DCD 5](#_Toc467965138)

[Observacions 7](#_Toc467965139)

[Conclusions 8](#_Toc467965140)

[Codis 9](#_Toc467965141)

[DSS 9](#_Toc467965142)

[DSS1 9](#_Toc467965143)

[DSS2 9](#_Toc467965144)

[DSS3 9](#_Toc467965145)

[DSS4 9](#_Toc467965146)

[DS 9](#_Toc467965147)

[DS1 9](#_Toc467965148)

[DS2 9](#_Toc467965149)

[DS3 9](#_Toc467965150)

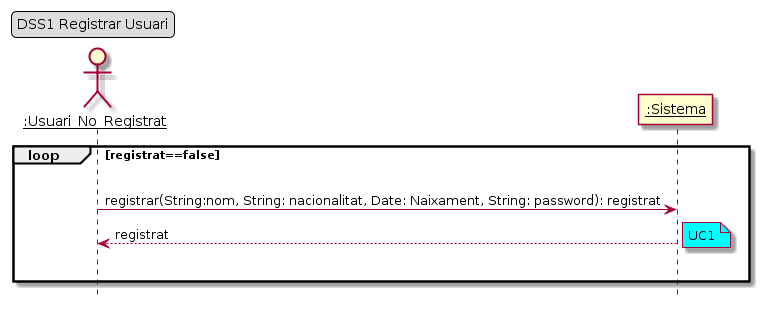
[DS4 9](#_Toc467965151)

[DCD 10](#_Toc467965152)

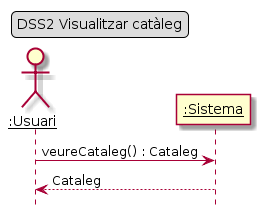
# Models

## DSS

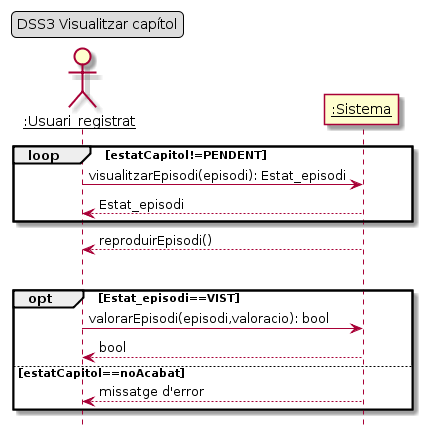
### DSS 1: Registrar usuari



### DSS 2: Visualitzar catàleg



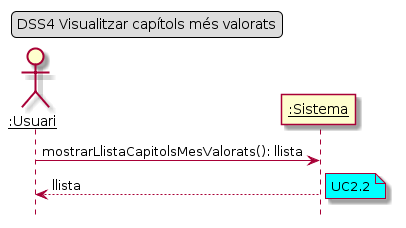
### DSS 3: Visualitzar i valorar capítol



a

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 reprodueix. 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.

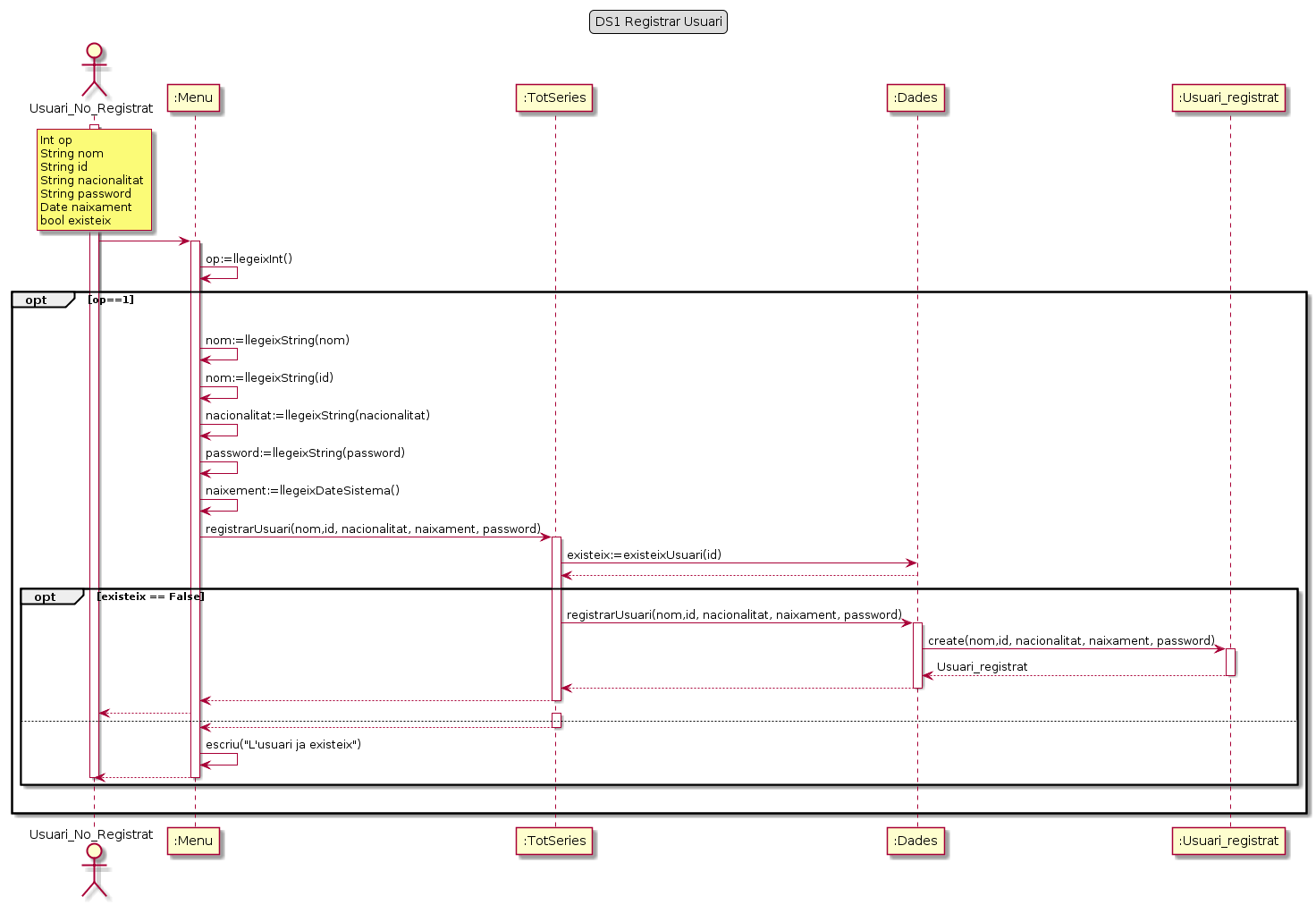
### DSS 4: Visualitzar capítols més valorats



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.

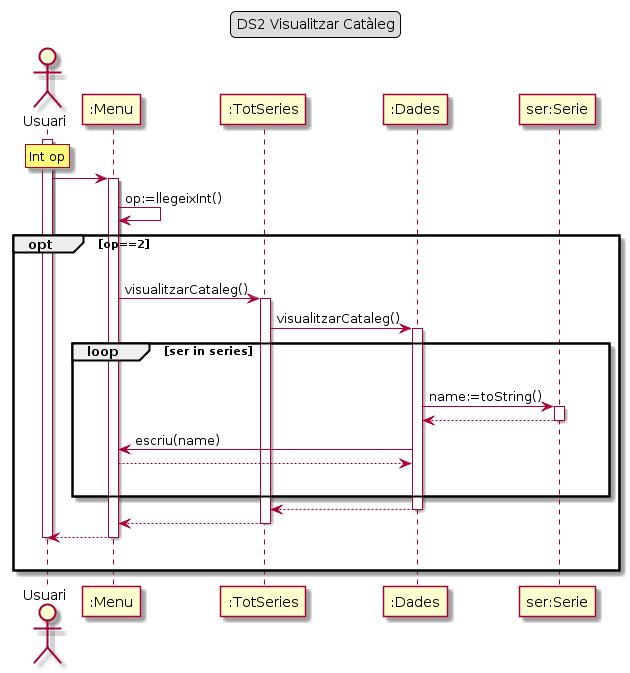
## DS

### 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



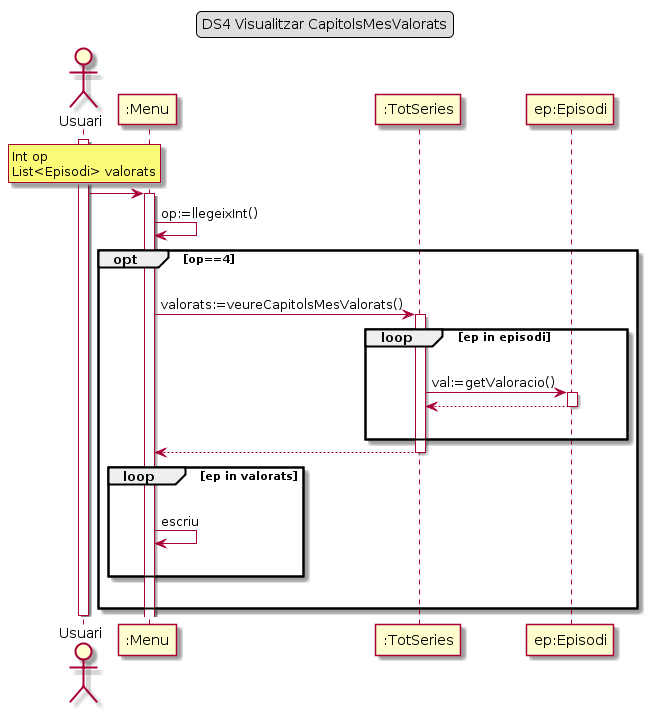
El DS2 és el procés que segueix l’usuari quan vol visualitzar el catàleg. En primer lloc l’usuari escull l’opció dins del menú que s’encarrega de mostrar el catàleg. En aquest cas l’usuari no té perquè estar registrat ja que no importa perquè igualment el catàleg es mostrarà.

### DS 3: Visualitzar i valorar capítol



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 reprodueix. 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.

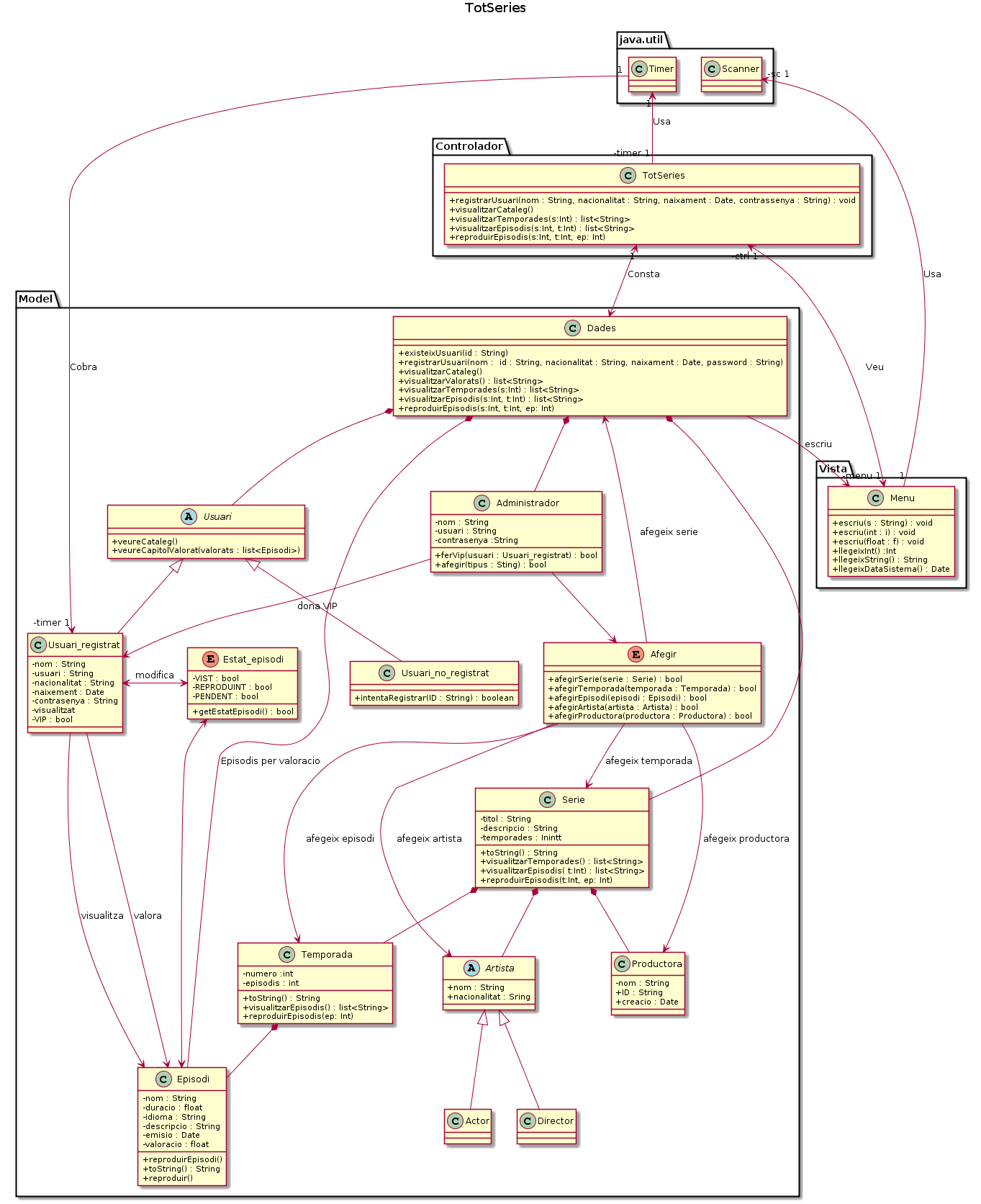
### DS 4: Visualitzar capítols més valorats



El DS4 mostra el procés que seguiria l’usuari quan aquest vol veure la llista de tots els episodis millor valorats pels altres usuaris. L’usuari escull l’opció en el menú i aquest li retorna la llista.

## DCD

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.

# 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.

# Codis

## 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 registrat==false

|||

usuari -> sistema : registrar(String:nom, String: nacionalitat, Date: Naixament, String: password): registrat

sistema --> usuari : registrat

note right: UC1

|||

end

@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 estatCapitol!=PENDENT

usuari -> sistema : visualitzarEpisodi(episodi): Estat\_episodi

usuari <--- sistema : Estat\_episodi

end

usuari <--- sistema : reproduirEpisodi()

|||

opt Estat\_episodi==VIST

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

usuari <--- sistema : bool

else estatCapitol==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

Int op

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: op:=llegeixInt()

opt op==1

|||

menu -> menu: nom:=llegeixString(nom)

menu -> menu: nom:=llegeixString(id)

menu -> menu: nacionalitat:=llegeixString(nacionalitat)

menu -> menu: password:=llegeixString(password)

menu -> menu: naixement:=llegeixDateSistema()

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

activate TS

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

dades ---> TS

opt existeix == False

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

else

activate TS

TS ---> menu

deactivate TS

menu -> menu : escriu("L'usuari ja existeix")

menu ---> usuari

deactivate menu

deactivate usuari

end

|||

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

### @startuml

### legend top

### DS3 Visualitzar Episodi

### end legend

### actor Usuari\_registrat as usuari

### activate usuari

### participant "Menu" as menu

### participant ":TotSeries" as TS

### participant ":Dades" as dades

### participant ":Serie" as serie

### participant ":Temporada" as temp

### participant ":Episodi" as episodi

### usuari -> menu

### activate menu

### menu -> menu : op=llegeixInt()

### opt op==3

### |||

### menu->menu : val=llegeixInt()

### opt 0<val && val<5

### |||

### menu->TS : valorarEpisodi(s,t,ep,val)

### activate TS

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

### activate dades

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

### activate serie

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

### activate temp

### temp->episodi: valorarEpisodi(val)

### activate episodi

### 

### episodi--->temp

### deactivate episodi

### temp--->serie

### deactivate temp

### serie--->dades

### deactivate serie

### dades--->TS

### deactivate dades

### TS--->menu

### deactivate TS

### menu--->usuari

### |||

### else

### |||

### menu->menu: escriu("Valoracio no valida")

### deactivate menu

### |||

### end

### |||

### deactivate usuari

### end

### @enduml

### DS4

@startuml

legend top

DS4 Visualitzar CapitolsMesValorats

end legend

actor Usuari as usuari

activate usuari

rnote over usuari

Int op

List<Episodi> valorats

end rnote

participant ":Menu" as menu

participant ":TotSeries" as TS

participant "ep:Episodi" as episodi

usuari -> menu

activate menu

menu -> menu : op:=llegeixInt()

opt op==4

|||

menu -> TS: valorats:=veureCapitolsMesValorats()

activate TS

loop ep in episodi

|||

TS -> episodi : val:=getValoracio()

activate episodi

episodi ---> TS

deactivate episodi

|||

end

TS --> menu

deactivate TS

loop ep in valorats

|||

menu -> menu : escriu

|||

end

|||

end

deactivate usuari

@enduml

## DCD

@startuml

title TotSeries

skinparam classAttributeIconSize 0

namespace java.util {

class Timer

class Scanner

}

namespace Vista {

class Menu {

+ escriu(s : String) : void

+ escriu(int : i) : void

+ escriu(float : f) : void

+llegeixInt() :Int

+llegeixString() : String

+llegeixDataSistema() : Date

}

}

namespace Controlador {

class TotSeries {

+ registrarUsuari(nom : String, nacionalitat : String, naixament : Date, contrassenya : String) : void

+visualitzarCataleg()

+visualitzarTemporades(s:Int) : list<String>

+visualitzarEpisodis(s:Int, t:Int) : list<String>

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

}

}

namespace Model {

class Dades{

+existeixUsuari(id : String)

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

+visualitzarCataleg()

+visualitzarValorats() : list<String>

+visualitzarTemporades(s:Int) : list<String>

+visualitzarEpisodis(s:Int, t:Int) : list<String>

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

}

abstract class Usuari{

+veureCataleg()

+veureCapitolValorat(valorats : list<Episodi>)

}

class Usuari\_registrat {

- nom : String

- usuari : String

- nacionalitat : String

- naixement : Date

- contrasenya : String

- visualitzat

- VIP : bool

}

class Usuari\_no\_registrat{

+intentaRegistrar(ID : String) : boolean

}

class Administrador{

-nom : String

-usuari : String

-contrasenya :String

+ferVip(usuari : Usuari\_registrat) : bool

+afegir(tipus : Sting) : bool

}

enum Afegir{

+afegirSerie(serie : Serie) : bool

+afegirTemporada(temporada : Temporada) : bool

+afegirEpisodi(episodi : Episodi) : bool

+afegirArtista(artista : Artista) : bool

+afegirProductora(productora : Productora) : bool

}

enum Estat\_episodi{

-VIST : bool

-REPRODUINT : bool

-PENDENT : bool

+getEstatEpisodi() : bool

}

class Episodi{

-nom : String

-duracio : float

-idioma : String

-descripcio : String

-emisio : Date

-valoracio : float

+reproduirEpisodi()

+toString() : String

+reproduir()

}

class Temporada{

-numero :int

-episodis : int

+toString() : String

+visualitzarEpisodis() : list<String>

+reproduirEpisodis(ep: Int)

}

class Productora{

-nom : String

+ID : String

+creacio : Date

}

class Serie{

-titol : String

-descripcio : String

-temporades : Inintt

+toString() : String

+visualitzarTemporades() : list<String>

+visualitzarEpisodis( t:Int) : list<String>

+reproduirEpisodis(t:Int, ep: Int)

}

abstract class Artista{

+nom : String

+nacionalitat : Sring

}

class Director{

}

class Actor{

}

Dades \*-- Serie

Serie \*-- Productora

Serie \*-- Temporada

Serie \*-- Artista

Temporada \*-- Episodi

Episodi <--> Estat\_episodi

Dades \*-- Usuari

Dades \*-- Administrador

Dades \*-- Episodi : Episodis per valoracio

Usuari <|-- Usuari\_registrat

Usuari <|-- Usuari\_no\_registrat

Administrador --> Usuari\_registrat : dona VIP

Administrador --> Afegir

Usuari\_registrat --> Episodi : valora

Usuari\_registrat --> Episodi : visualitza

Usuari\_registrat <-> Estat\_episodi : modifica

Afegir --> Dades : afegeix serie

Afegir --> Serie : afegeix temporada

Afegir --> Temporada : afegeix episodi

Afegir --> Artista : afegeix artista

Afegir --> Productora : afegeix productora

Artista <|-- Actor

Artista <|-- Director

}

Controlador.TotSeries "1" <-> Model.Dades : Consta

Controlador.TotSeries "-ctrl 1" <-> "-menu 1" Vista.Menu : Veu

Model.Dades -> Vista.Menu : escriu

java.util.Timer "1" <- "-timer 1" Controlador.TotSeries : Usa

java.util.Timer "1" -> "-timer 1" Model.Usuari\_registrat : Cobra

java.util.Scanner "-sc 1" <- "1" Vista.Menu : Usa

@enduml