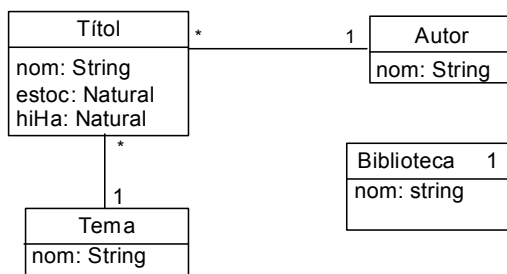


Xarxa de Biblioteques.

Un ajuntament de Catalunya ha establert una xarxa de biblioteques amb fons comú, és a dir, a totes les biblioteques hi ha els mateixos títols. A cada biblioteca s'assignen un nombre d'exemplars de cada llibre en funció de la població del barri on està situada cada una de les biblioteques.

L'ajuntament vol establir un servei de préstec de llibres mitjançant el qual un ciutadà pot acudir a qualsevol biblioteca de la xarxa i consultar el contingut del fons. El sistema ha de donar-li informació, per a un títol determinat, de quines biblioteques de la xarxa tenen exemplars disponibles.



RI textuais:

1. Claus (Títol,nom), (Autor,nom), (Tema,Nom)
2. La resta de RI no tenen interès aquí.

Els usuaris de la biblioteca poden tornar el llibre a la biblioteca on vulguin per la qual cosa s'admet que una biblioteca pugui tenir disponibles més llibres que l'estoc que té assignat. El diagrama de classes de partida és el de la figura on:

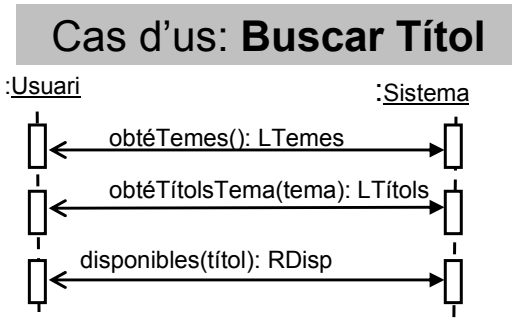
- *estoc* té el nombre d'unitats de cada títol assignades a la biblioteca.
- *hiHa* té el nombre d'unitats de cada títol disponibles a la biblioteca.

Per controlar els llibres vol crear tres casos d'ús a cada biblioteca. A continuació teniu la descripció de cada un dels casos després de fer el procés d'assignació de responsabilitats a capes. Els serveis de les capes de presentació no són d'interès aquí.

Biblioteca		Hi ha

OK Cancel

<<Zona de missatges d'error>>



Operacions de la capa de Domini

context CapaDomini::obtéTemes(): Set(String)

post: 2. result=noms dels temes enregistrats els sistema

context CapaDomini::obtéTítolsTema(tema: String): Set(String)

pre: 1. *tema* correspon a un tema enregistrat al sistema.

post: 2. result=noms dels llibres enregistrats el sistema de *tema*.

context CapaDomini::disponibles(títol: String): Set(String+natural)

pre: 1. *títol* correspon a un títol enregistrat al sistema.

post: 2. result=conjunt de tuples amb el nom de les biblioteques de la xarxa i els exemplars disponibles, en cada una d'elles, de *títol*.

Descripció del cas d'us

1. El sistema mostra la llista de temes.
2. L'usuari escull un tema.
3. El sistema mostra els títols associats al tema
4. L'usuari escull un títol.
5. El sistema mostra el nombre d'exemplars disponibles en cada biblioteca de la xarxa del títol escollit per l'usuari, inclosa aquella des de la qual es fa la petició.

Cas d'us: Nous Títols

Descripció del cas d'us

1. A les 00:00 de cada dia, el sistema inicia el cas d'us.
2. El sistema recupera els títols nous.
3. El sistema incorpora els nous títols al fons de la biblioteca

:Sistema

obtéNovetats(): LNous

altaTítols(LNous)

Operacions de la capa de Domini

Nous={títol: String, autor: String, tema: string, estoc: natural}

context CapaDomini::obtéNovetats(): Set(Nous)

post: 2.1. result=conjunt de novetats enregistrades al sistema i pendents d'enregistrar a la biblioteca.

context CapaDomini::altaTítols(LNous: Set(Nous))

pre: 1: Cap dels títols inclosos en *LNous* existeix a la Biblioteca.

post: 2.1. Si *LNous.autor* no existeix, es dona d'alta una instància d'Autor.

post: 2.2. Si *LNous.tema* no existeix, es dona d'alta una instància de Tema.

post: 2.3. És dona d'alta un Títol amb nom=*LNous.títol*, estoc=*LNous.estoc*, disponible=0.

post: 2.4. El nou títol s'associa a l'autor.

post: 2.5. El nou títol s'associa al tema.

Títol

☒ Cessió

☐ Devolució

OK

Cancel

<<Zona de missatges d'error>>

Cas d'us: Cessió de Títol

:Usuari

:Sistema

obtéTítols(): LTítols

canvia(títol,cessió): trobat

Operacions de la capa de Domini

context CapaDomini::obtéTítols(): Set(String)

post: 2. result=noms dels llibres enregistrats els sistema.

context CapaDomini::canvia(títol: String, cessió: booleà): Booleà

pre 1. Existeix una instància de la classe Títol de nom *títol*.

post: 2.1. Si *cessió=cert* no hi ha disponibles de títol, no fa res i torna fals.

post 2.2. Altrament torna cert i

2.2.1 Si *cessió*, es decrementa 1 a l'atribut *hiHa* de la instància de la classe Títol corresponent a *títol*.

2.2.2 Si no *cessió*, s'incrementa 1 a l'atribut *hiHa* de la instància de la classe Títol corresponent a *títol*.

Descripció del cas d'us

1. El sistema mostra la llista de títols.
2. L'usuari escull un títol.
3. L'usuari tria cessió o devolució.
4. El sistema registra el canvi

El dissenyador decideix establir una arquitectura orientada a serveis per a desenvolupar l'aplicació. Per aconseguir-lo, s'ofereix a les biblioteques un controlador de serveis de nom **Serveis Comuns** on estan disponibles els següents serveis:

Atributs del DTO d'intermediació *dtoAltes*:

dtoAltes=*altes*: Set(Nous)

Atributs del DTO d'intermediació *dtoLDisp*:

dtoLDisp= *Ldisp*: Set({*bib*: String, disponible: Natural})

context ServeisComuns:: *altes*(*bib*: String): *dtoAltes*

exc: 1.1 biblioteca-no-existeix: *bib* no correspon a cap biblioteca del sistema

post: 2.1. *alt* conté els títols enregistrats al servidor i pendents encara d'enregistrar a *bib*.

2.2. S'esborra la llista de títols nous no enregistrats a la biblioteca *bib* del servidor de Serveis Comuns

context ServeisComuns::*disponibles*(*bib*: String, *títol*: String): *dtoLDisp*

exc 1.1 biblioteca-no-existeix: *bib* no correspon a una biblioteca del sistema

1.2 *títol*-no-existeix: *títol* no correspon a un títol enregistrat al sistema.

post: 2.1. *LDisp* = conjunt de tuples amb el nom de les biblioteques de la

xarxa, tret de la biblioteca *bib*, i els exemplars disponibles en ella de *títol*.

Cada biblioteca ha d'oferir al sistema un controlador de serveis, de nom **Serveis Biblioteca**, on està el servei:

context Serveis Biblioteca::tinc(bib: String, títol: String): Natural

exc 1.1 biblioteca-no-existeix: bib no és la seva referència de biblioteca.

1.2 títol-no-existeix: títol no correspon a un títol enregistrat a la biblioteca.

post: 2.1. dispo-so-de = valor de l'atribut hiHa de la instància de la classe Títol corresponen a títol.

El dissenyador decideix també que tots els controladors de la capa de domini siguin transaccionals i que el model de disseny sigui Domini Model + Data Mapper amb consultes. Finalment, i per a deixar el disseny preparat per a futures ampliacions, decideix dotar de navegabilitat doble, en la biblioteca, a les dues associacions de l'esquema de classes.

Es demana que dissenyeu:

1. Els controladors transaccionals.
2. Totes les operacions de les capes de domini dels casos d'ús.
3. El controlador de serveis **Serveis Biblioteca**.
4. El **servei disponibles** del controlador **Serveis Comuns**. Aclariu si es necessari fer alguna adaptació al diagrama de classes de la biblioteca quan s'apliqui al servidor on residirà el controlador de **Serveis Comuns**.