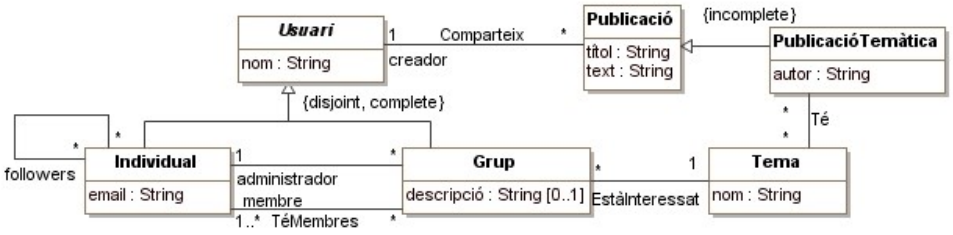
Segon Control AS – Grup 10 Exercici 2 22-4-20

Hora inici: 18:50
Hora límit entrega Atenea: 20:30

Volem redissenyar una variant de la xarxa social per compartir publicacions del 1er control. A continuació disposeu de l'esquema conceptual del sistema que volem redissenyat:



R.I. Textuals:
- Claus: (Usuari, nom); (Publicació, títol); (Tema, nom);
- Un usuari de grup només pot compartir publicacions temàtiques que tinguin almenys un tema que coincideixi amb el tema en el que l'usuari de grup està interessat.
- Un usuari individual no es pot seguir a si mateix.
- L'usuari administrador d'un usuari de grup és un dels usuaris individuals que és membre.
- Altres restriccions no rellevants pel problema

Disposem de dos serveis que enregistren part de la informació que emmagatzemàvem en el nostre sistema. El servei Usuaris (SvUsus) permet registrar la informació dels usuaris. El servei Publicacions (SvPubs) és un repositori gran de publicacions que diverses xarxes socials (també la nostra) mantenen actualitzat. A continuació disposeu de les operacions que proporcionen els serveis.

context SvPubs :: getPublicacions (): Set (TupleType(títol: String, text: String, temes: Set(nom: String))
post es retorna el títol, text i temes de totes les publicacions enregistrades al servei.

context SvPubs :: getPublicacionsxTema (tema: String): Set (TupleType(títol: String, text: String, temes: Set(nom: String))
exc temaNoExisteix: el tema amb nom *nom* no existeix al servei.
post es retorna el títol, text i temes de totes les publicacions que tenen el tema *tema*.

context SvPubs :: getPublicacionsTemàtiques ():Set (TupleType(títol: String, text: String, temes: Set(nom: String))
post es retorna el títol, text i temes de totes les publicacions que com a mínim un tema assignat.

context SvPubs :: getPublicacionsNoTemàtiques ():Set (TupleType(títol: String, text: String)
post es retorna el títol, text i temes de totes les publicacions que no tenen cap tema assignat.

context SvPubs :: getTemes ():Set (nom: String)
post es retorna el nom dels temes que hi ha enregistrats al servei.

context SvPubs :: creaPublicació (títol: String, text: String, temes: Set(nom: String))
exc temaNoExisteix: algun tema del conjunt *temes* no existeix al servei.
exc publicacióExisteix: la publicació amb el títol *títol* existeix al servei.
post es crea una publicació i se li assigna el títol, el text i, si n'hi ha, els temes del conjunt temes.

context SvUsus :: getUsuaris (): Set (TupleType(nom: String, email: String, followers: Set(nom: String))
post es retorna el nom, email i followers de tots els usuaris enregistrats al servei.

context SvUsus :: getUsuari (nom: String): TupleType(nom: String, email: String, followers: Set(nom: String)
exc usuariNoExisteix: l'usuari amb nom *nom* no existeix al servei.
post es retorna el nom, email i followers l'usuari amb el nom indicat al paràmetre.

Volem dissenyar l'operació de la capa de domini *compartirPublicacióTemàtica*. Aquesta operació crea una publicació temàtica i obté les publicacions interessants per a l'usuari creador. A continuació disposeu del contracte de l'operació:

context CapaDomini:: compartirPublicacióTemàtica (nomUsuariCreador: String, títol: String, text: String, temes: Set(nom:String), autor: String): Set(TupleType(títol: String, text: String))
pre *hiHaTemes:* el conjunt *temes* té algun tema i tots els temes existeixen al servei.
pre *coincidènciaDades:* si la publicació existeix al repositori de publicacions, les dades dels paràmetres coincideixen amb les dades que hi ha al repositori.
exc *usuariNoExisteix:* l'usuari amb nom *nomUsuariCreador* no existeix.
exc *publicacióExisteix:* la publicació temàtica amb títol *títol* existeix al nostre sistema.
exc *usuariNoPotCompartir:* l'usuari creador és individual i no té followers o l'usuari és de grup i té només un membre.
exc *publicacióDeGrupSenseTema:* l'usuari creador és de grup i el conjunt de temes no té el tema en el que està interessat el grup.
post *creacióPublicacióTemàtica:* es dóna d'alta la publicació temàtica i se li assignen les dades.
post *publicacionsInteressants* = es retorna el títol i text de les publicacions que són al repositori tals que:

- Si l'usuari creador és de grup, les publicacions temàtiques que tenen algun tema que coincideix amb el tema en el que està interessat el grup.
- Si l'usuari creador és individual, totes les publicacions no temàtiques (no tenen cap tema).

a) **[5,5 punts]** Suposant que les navegabilitat existents són dobles a excepció de *EstàInteressat* que va de Grup a Tema i que l'operació que retorna les publicacions interessants d'un usuari (última postcondició del contracte) ha de ser reusable, es demana:
   1. Diagrama de classes (incloent els atributs però no operacions) del paquet Domain Model de la capa de domini del sistema de la central de reserves. **Com a criteri principal de disseny volem minimitzar el nombre d'invocacions remotes. Com a criteri secundari també volem mantenir els criteris habituals de reusabilitat, canviabilitat i minimitzar la redundància de les dades**.
   2. Diagrama de seqüència de l'operació *compartirPublicacióTemàtica* de la capa de domini. Indiqueu clarament en el diagrama de seqüència els singleton i les interfaces. Podeu suposar l'existència de l'operació PublicacióTemàtca :: validaTemes(temes:Set(Temes), u:Usuari): Booleà que retorna cert si: 1) l'usuari és individual o 2) l'usuari és de grup i el tema en el que està interessat està dins dels temes del paràmetre *temes*. En cas contrari activa l'excepció *publicacióDeGrupSenseTema*.

b) **[0,5 punts]** Digueu si l'excepció de l'operació *usuariNoExisteix* podria convertir-se en precondició i ser garantida per la Capa de Presentació o la interfície (pantalles). Explica com es garantiria aquesta precondició i si seria necessari invocar a algun servei per garantir-la.