DOCUMENTO DE CAMBIOS EN LOS UML

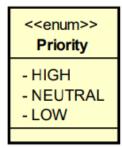
Grupo 24

Antes:

Después:



- HIGH : String = "HIGH"
- NEUTRAL : String = "NEUTRAL"
- LOW : String = "LOW"



<<datatype>> **Status**

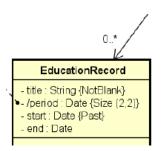
- PENDING : String = "PENDING"
- REJECTED : String = "REJECTED"
- DUE : String = "DUE"
- ACCEPTED : String = "ACCEPTED"
- CANCELLED : String = "CANCELLED"

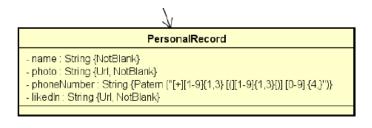
<<enum>> Status

- PENDING
- REJECTED
- DUE
- ACCEPTED
- CANCELLED

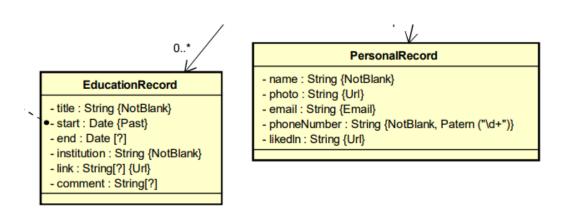
• En los modelos enviados en la entrega D02 teníamos definidos Status y Priority como datatypes y con el avance del proyecto hemos repercutido en que una mejor implementación sería usando enums.

Antes:





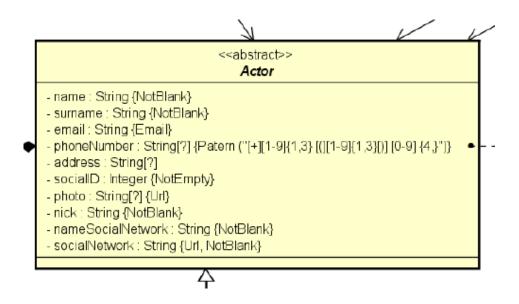
Después:



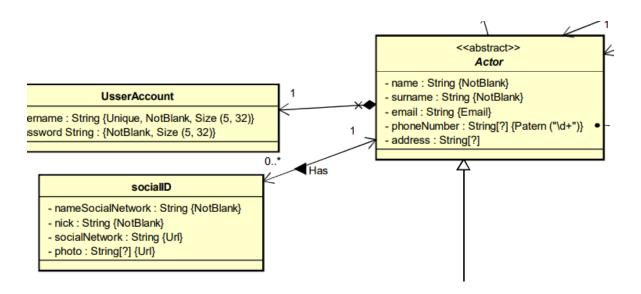
 Añadimos en PersonalRecord un nuevo atributo que nos faltaba "email" de tipo String con la restricción del Email.

Añadimos en EducationalRecord los atributos "institution", "link" y "comment" todos de tipo String y eliminamos el atributo derivado period.

Antes:



Después:



 En Actor eliminamos los atributos "socialID", "photo", "nick", "nameSocialNetwork", "socialNetwork".

Creamos una nueva clase llamada socialID en la que incluimos los atributos que eliminamos anteriormente.

Configuration

- banner : String {Url}

- message : String {NotBlank}

- spamWords : String [+]

- tax : Double {Min(0)}

- countryCode : String {NotBlank}

- catalogueTag : String [+]- treeCategory : String [+]

- catalogueText : String [+]

- other : String [*]

 Añadimos una clase Configuration con sus correspondientes atributos que nos permite añadir valores por defecto.

Application

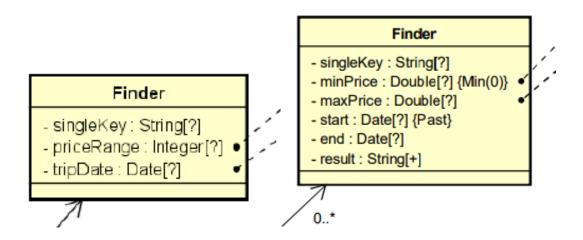
Application

- moment: Date {Past}
- status: Status {NotEmpty}
- comment: String[*]

Application

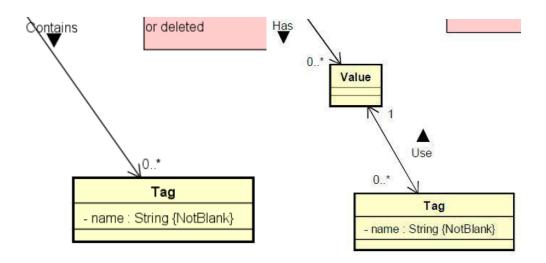
- moment: Date {Past}
- status: Status {NotBlank}
- comment: String[*]

 En el modelo de dominio quitamos la clase asociación y la colocamos como una clase normal con su multiplicidad para que a la hora de pasar a java podamos implementarlo bien. Antes: Después:



• En la clase Finder eliminamos los atributos "priceRange" y "tripDate" y le añadimos "minPrice", "maxPrice", "start", "end" y "result".

Antes: Después:



 Para que se cree un id arbitrario para la clase tag creamos una clase intermedia con Trip llamada "value" para que realice dicha tarea.

•	Hemos añadido varios cambios en las multiplicidades de todo el modelo
	conceptual y de dominio que con el avance del proyecto hemos repercutido en
	que necesitaban dichas modificaciones.