**ENTREGA 4**

**CAMBIOS UML Y JAVA**

**Diseño y Pruebas**



**Grupo 20:**

**José Ángel Domínguez Espinaco**

**Daniel Lozano Portillo**

**José Joaquín Rodríguez Pérez**

**María Ruiz Gutiérrez**

**Miguel Ternero Algarín**

**Laura Vera Recacha**

Índice

[Introducción 3](#_Toc498352930)

[Cambios realizados UML 3](#_Toc498352931)

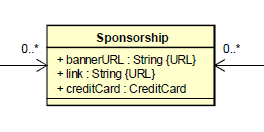
[Cambios Realizados Java 8](#_Toc498352932)

# Introducción

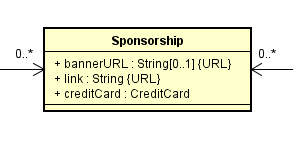
En este documento serán descritos todos los cambios de nuestro Modelo de Dominio y Modelo Java, los cuales irán acompañados de capturas.

# Cambios realizados UML

1. En la clase Sponsorship el atributo bannerURL puede ser opcional por lo tanto hemos cambiado la multiplicidad a [0..1].

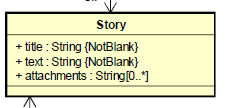
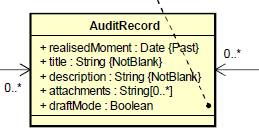
 **Antes:**

**Después:**

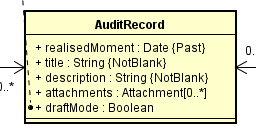
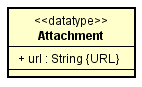
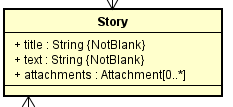


1. En las clases Story y Auditrecord se ha cambiado el tipo String del atributo attachments, creándose una clase datatype Attachment ya que al estar puesto como colección de String no podíamos poner una restricción de tipo @URL y al crear el datatatype ahora es ahí donde establecemos la restricción @URL.

**Antes:**

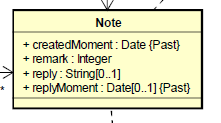


**Después:**

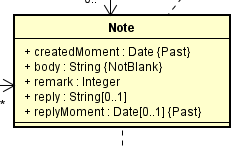


1. En la clase Note hemos añadido el atributo body de tipo String ya que es necesario a la hora de escribir el contenido de una nota.

**Antes:**



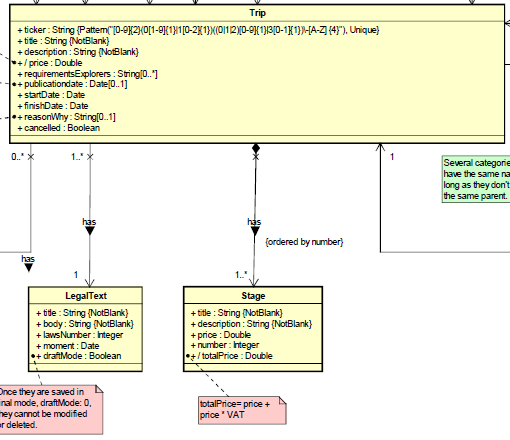
**Después:**



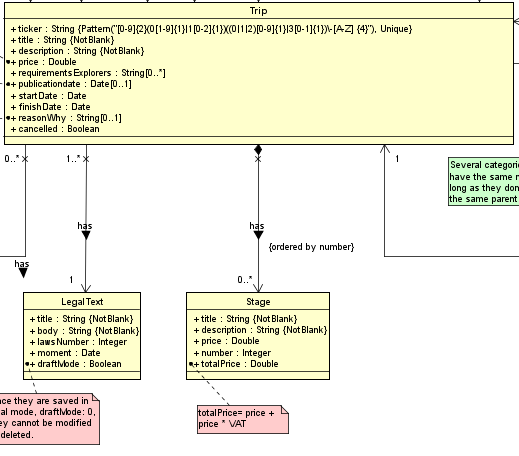
1. En la asociación de Trip con Stage hemos cambiado la multiplicidad a [0..\*] ya que puede existir un Trip que aún no tenga ninguna stage asiganada.

En las clases Stage y Trip sus atributos derivados totalPrice y price pasan a ser básicas ya que se calcularán en un método en los servicios StageService y TripService respectivamente.

**Antes:**

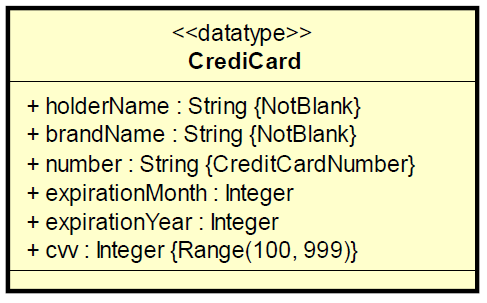


**Después:**

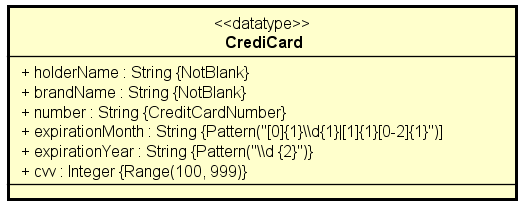


1. En CreditCard hemos cambiado los tipos de los atributos expirationMonth y expirationYear por String para comprobar sus restricciones mediante patrones que también se han añadido.

**Antes:**

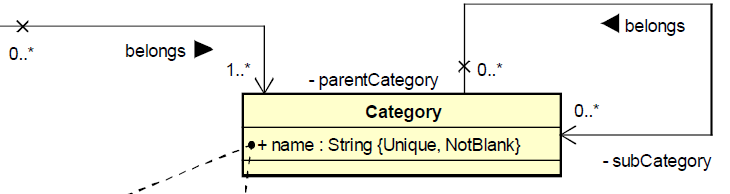


**Después:**

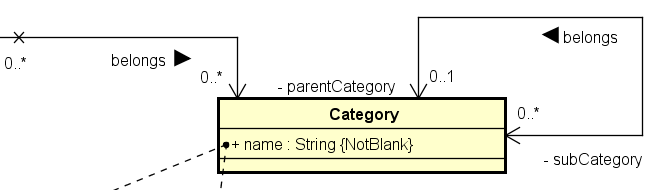


1. En la clase Category hemos cambiado la multiplicacidad de la asociación con trip que pasa a ser [0..\*] porque cuando creamos un Trip no tenemos por qué asignarle una categoría y la asociación y navegabilidad de una parentCategory que pasa a ser bidireccional y con multiplicidad [0..1].

**Antes:**

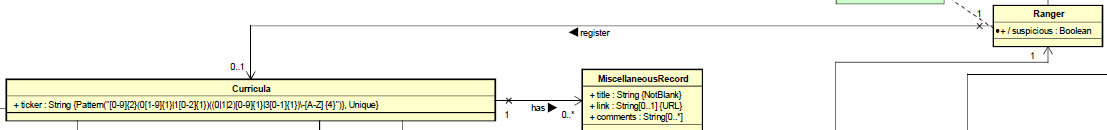


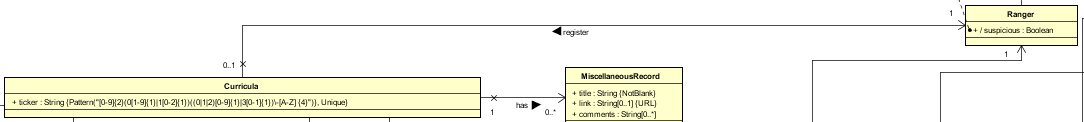
**Después:**



1. En la asociación de la clase Ranger con la clase Curricula hemos cambiado la navegabilidad al contrario, siendo ahora de Ranger hacia Curricula porque no puede existir una Curricula sin asignarle ningún Ranger pero sí puede haber un Ranger sin Curricula.

**Antes:**

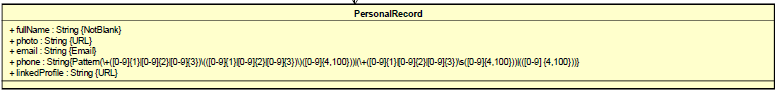


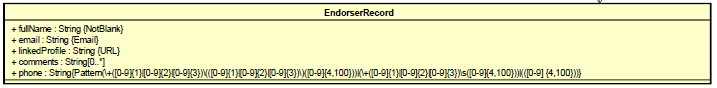
**Después:** 

1. En las clases PersonalRecord, EndorserRecord y ContactEmergency hemos eliminado el patrón del atributo phone ya que aún no cumpliendo el patrón debe de guardarse en el sistema.

También se ha hecho un poco más pequeño dicho patrón para utilizarlo en un método en los servicios.

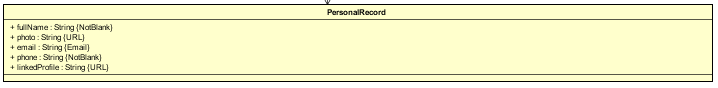
**Antes:**

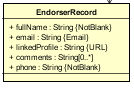


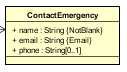




**Después:**

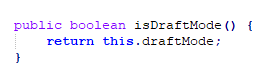


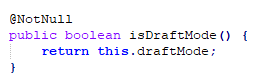




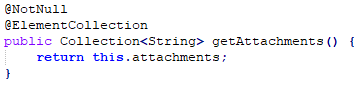
# Cambios Realizados Java

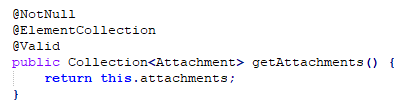
1. En la clase AuditRecord hemos puesto @NotNull en el atributo isDraftMode debido a que no puede ser nulo.

 **Antes:**

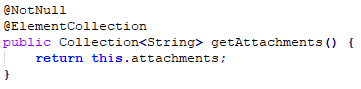
 **Después:**

1. En la clase AuditRecord en el método getAttachments al devolvernos una colección hemos añadido @Valid por lo mismo que hemos explicado anteriormente en el punto 2.

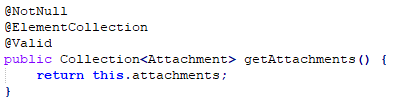
 **Antes:**

 **Después:**

1. En la clase Story en el método getAttachments al devolvernos una colección hemos añadido @Vallid por lo mismo que hemos explicado anteriormente en el punto 2.

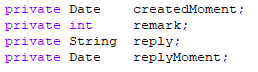
 **Antes:**

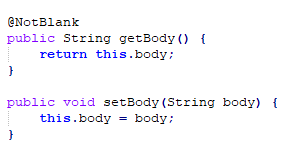
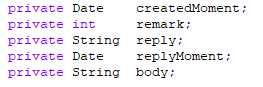
**Después:**



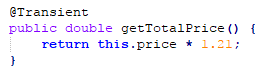
1. En la clase Note hemos añadido el atributo body de tipo String para así poder ver el contenido de una nota.

También hemos añadido sus correspondientes métodos getBody, el cual lleva la restricción @NotBlank, y setBody.

 **Antes:**

 **Después:**

1. En la clase Stage como el atributo totalPrice pasa a ser básico añadimos un setTotalPrice y el getTotalPrice lo modificamos.

 **Antes:**

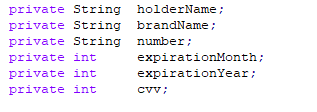
 **Después:**

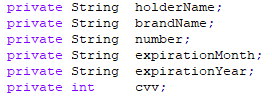


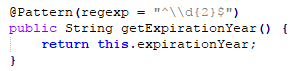
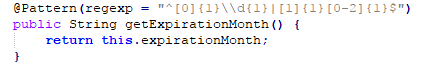
1. En la clase CreditCard hemos cambiado el tipo de atributo de expirationMonth y expirationYear ya que vamos a restringirlo mediante un patrón.

También añadimos el patrón en los métodos getExpirationMonth y getExpirationYear así como el tipo que devuelven.

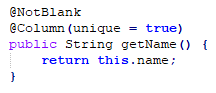
Modificacmos el tipo de atributo que se le pasa a los métodos setExpirationMonth y setExpirationYear por el tipo String.

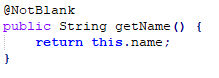
 **Antes:**

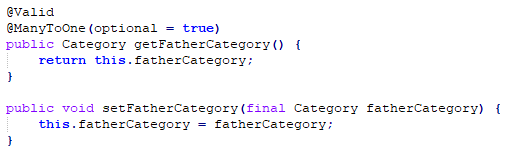
 **Después:**



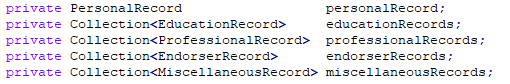
1. En la clase Category al ponerla bidireccional añadimos el atributo fatherCategory así como sus correspondientes métodos getFatherCategory, setFatherCategory y sus restricciones. También eliminamos @Column.

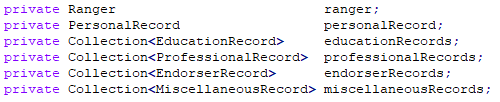
 **Antes:**

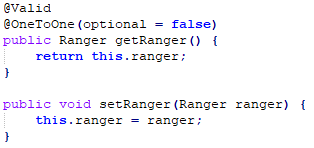
 **Después:**



1. En la clase Curricula al cambiar la navegabilidad hemos añadido el atributo ranger con sus respectivos métodos getRanger y setRanger.

**Antes:** 

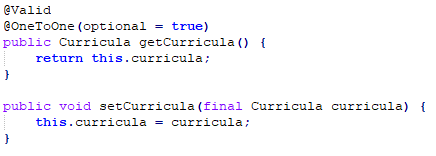
**Después:** 



1. En la clase Ranger al cambiar la navegabilidad hemos eliminado el atributo curricula y también sus respectivos métodos getCurricula y setCurricula.

**Antes:**

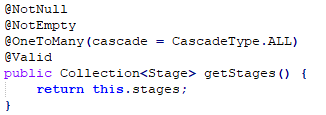




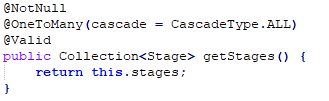
**Después:**



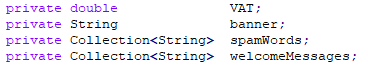
1. En la clase Trip hemos eliminado la restricción de @NotEmpty de getStages ya que un trip puede no tener ninguna stage referenciada.

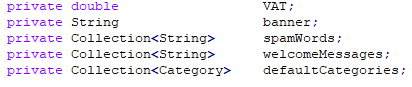
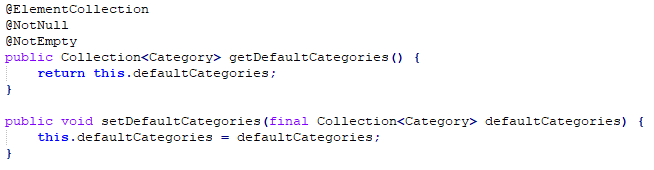
 **Antes:**

**Después:**



1. En la clase ConfigurationSystem hemos añadido una colección con las categorios por defecto que tiene el sistema con sus correspondientes métodos getDefaultCategories y setDefaultCategories;

 **Antes:**

 **Después:**