DOCUMENTO DE CAMBIOS EN LAS CLASES JAVA

Grupo 24

Versión 1.1

Antes:

Priority.java

```java
package domain;

public class Priority {


    private static final String HIGH = "HIGH";
    private static final String NEUTRAL = "NEUTRAL"
    private static final String LOW = "LOW";

    public static String getHigh() {
        return HIGH;
    }

    public static String getNeutral() {
        return NEUTRAL;
    }

    public static String getLow() {
        return LOW;
    }
}
```

Status.java

```java
package domain;

public class Status {


    private static final String PENDING = "PENDING";
    private static final String REJECTED = "REJECTED";
    private static final String DUE = "DUE";
    private static final String ACCEPTED = "ACCEPTED";
    private static final String CANCELLED = "CANCELLED";


    public static String getPending() {
        return PENDING;
    }


    public static String getRejected() {
        return REJECTED;
    }


    public static String getDue() {
        return DUE;
    }


    public static String getAccepted() {
```

Después:

Priority.java

```java
package domain;

import javax.persistence.Access;

@Embeddable
@Access(AccessType.PROPERTY)
public enum Priority {

    HIGH, NEUTRAL, LOW;

}
```

Status.java

```java
package domain;

import javax.persistence.Access;

@Embeddable
@Access(AccessType.PROPERTY)
public enum Status {

    PENDING, REJECTED, DUE, ACCEPTED, CANCELLED;

}
```

- Anteriormente, “Status” y “Priority” eran clases java configuradas como Datatypes, para esta entrega hemos cambiado estas clases a Enum conservando los valores que ya tenían.

Antes:

EducationRecord.java | PersonalRecord.java

```java
package domain;

import java.util.ArrayList;

public class EducationRecord {

    private String title;
    private Date start;
    private Date end;


    @NotBlank
    public String getTitle() {
```

EducationRecord.java | PersonalRecord.java

```java
package domain;

import javax.validation.constraints.Pattern;

public class PersonalRecord extends DomainEntity {

    private String name;
    private String photo;
    private String phoneNumber;
    private String likedIn;


    @NotBlank
    public String getName() {
```

Después:

EducationRecord.java | PersonalRecord.java

```java
package domain;

import java.util.Date;

@Entity
@Access(AccessType.PROPERTY)
public class EducationRecord extends DomainEntity {

    // Constructors

    public EducationRecord() {
        super();
    }

    // Attributes

    private String title;
    private Date start;
    private Date end;
    private String institution;
    private String link;
    private String comment;

    @NotBlank
    public String getTitle() {
```

EducationRecord.java | PersonalRecord.java

```java
package domain;

import javax.persistence.Access;

@Entity
@Access(AccessType.PROPERTY)
public class PersonalRecord extends DomainEntity {

    // Constructors

    public PersonalRecord() {
        super();
    }

    // Attributes

    private String name;
    private String photo;
    private String email;
    private String phoneNumber;
    private String likedIn;

    @NotBlank
    public String getName() {
```

- En EducationRecord, modificamos la clase añadiendo los atributos “institution”, “link” y “comment” y eliminado el atributo “period”.
- En PersonalRecord, añadimos un nuevo atributo, “email”.

Antes:

```java
Actor.java
import java.util.Collection;

public abstract class Actor extends DomainEntity {


    private String name;
    private String surname;
    private String email;
    private String phoneNumber;
    private String address;
    private Integer socialID;
    private String photo;
    private String nick;
    private String nameSocialNetwork;
    private String socialNetwork;


    private UserAccount userAccount;
    private Folder inBox;
    private Folder outBox;
    private Folder notificationBox;
    private Folder trashBox;
    private Folder spamBox;
    private Collection<Folder> customFolders;


    private Message received;
    private Collection<Message> sent;
```

Después:

```java
Actor.java   SocialId.java
@Entity
@Access(AccessType.PROPERTY)
@Inheritance(strategy = InheritanceType.TABLE_PER_(
public abstract class Actor extends DomainEntity {

    // Constructors

    public Actor() {
        super();
    }

    // Attributes

    private String name;
    private String surname;
    private String email;
    private String phoneNumber;
    private String address;

    @NotBlank
    public String getName() {
```

```java
Actor.java   SocialId.java
@Entity
@Access(AccessType.PROPERTY)
public class SocialId extends DomainEntity {

    // Constructors

    public SocialId() {
        super();
    }

    // Attributes

    private String photo;
    private String nick;
    private String nameSocialNetwork;
    private String socialNetwork;

    @URL
    public String getPhoto() {
```

- Creamos una nueva clase, SocialID, en la que añadimos los atributos “photo”, “nick”, “nameSocialNetwork” y “socialNetwork”, que anteriormente estaban en la clase Actor.

```java
Configuration.java

@Entity
@Access(AccessType.PROPERTY)
public class Configuration extends DomainEntity {

    // Constructors

    public Configuration() {
        super();
    }

    // Attributes

    private String banner;
    private String message;
    private Collection<String> spamWords;
    private Double tax;
    private String countryCode;
    private Collection<String> catalogueTag;
    private Collection<String> treeCategory;
    private Collection<String> catalogueText;
    private Collection<String> other;

    @URL
    @NotBlank
    public String getBanner() {
        return banner;
```

- Creamos una clase, Configuration, con sus correspondientes atributos que nos permite añadir valores por defecto.

Antes:

```java
Finder.java

import java.util.Date;

public class Finder extends DomainEntity{

    private String singleKey;
    private Integer priceRange;
    private Date tripDate;

    private Explorer explorer;


    public String getSingleKey(){
```

Después:

```java
Finder.java

@Entity
@Access(AccessType.PROPERTY)
public class Finder extends DomainEntity

    // Constructors

    public Finder() {
        super();
    }

    // Attributes

    private String singleKey;
    private Double minPrice;
    private Double maxPrice;
    private Date start;
    private Date end;
    private Collection<String> result;

    public String getSingleKey() {
```

- En la clase Finder eliminamos los atributos “priceRange” y “tripDate” y le añadimos “minPrice”, “maxPrice”, “start”, “end” y “result”.

Antes:

```java
Tag.java
package domain;

import javax.validation.Valid;

public class Tag extends DomainEntity {

    private String name;

    private Trip trip;

    @NotBlank
    public String getName() {
```

Después:

```java
Tag.java   Value.java

@Entity
@Access(AccessType.PROPERTY)
public class Tag extends DomainEntity {

    // Constructors

    public Tag() {
        super();
    }

    // Attributes

    private String name;

    @NotBlank
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // Relationships

    private Value value;
```

```java
Tag.java   Value.java

@Entity
@Access(AccessType.PROPERTY)
public class Value extends DomainEntity{

    //Constructors

    public Value(){
        super();
    }

    //Relationships

    private Trip trip;
    private Collection<Tag> tag;

    @Valid
    @NotNull
    @ManyToOne(optional = false)
    public Trip getTrip() {
```

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

- A todas las clases java les añadimos los constructores, restricciones y multiplicidades necesarias para su implementación.