

Analysis Report D02



Mario Sánchez Naranjo

marsannar2@alum.us.es
Student #4

Diseño y Pruebas II
Grupo C1.02.01

15/02/2023

Tabla de contenido

<i>Sumario</i>	<i>1</i>
<i>Historial de versiones</i>	<i>2</i>
<i>Contenido</i>	<i>4</i>
<i>Requisito 1:Entidad Company</i>	<i>4</i>
Requisito 2:Entidad Practicum	5
Requisito 3:Entidad Session	6
Conclusiones	7
<i>Bibliografía</i>	<i>7</i>

Sumario

Durante este sprint trataremos la construcción de las entidades que compondrán nuestra base de datos y con los cuales trabajaremos a fondo , como pueden ser las entidades **Practicum**(las prácticas que ofrece una empresa) , **Company**(un rol que simula el papel de una empresa real) y **Session**(las sesiones que se imparten sobre una práctica).

Historial de versiones

Fecha	Versión	Descripción	Sprint
07/03/2023	1.0	creación del documento	1

Contenido

Requisito 1:Entidad Company

- 1) There is a new project-specific role called company, which has the following profile data: name (not blank, shorter than 76 characters), VAT number (not blank, shorter than 26 characters), summary (not blank, shorter than 101 characters), and an optional link with further information.

El cliente nos pide que implementemos un nuevo rol para la página, que es la entidad Empresa, la cual debe tener los siguientes atributos:

- **Name:** Contiene el nombre de la empresa. No puede estar en blanco, pues es necesario para que el cliente pueda reconocer a la empresa, y no puede contener más de 76 caracteres. Indudablemente, he concluido de que este atributo se trata de una cadena de caracteres.
- **VAT number:** el número VAT es un atributo que contiene el número de identificación fiscal de la empresa. Tras el análisis e investigación de este requisito, he hallado dos formas de implementar este atributo:

1) Que sea un tipo String, ya que tras investigar un poco, he concluido en que puede contener letras (aunque depende del país en el que te encuentres) y la descripción del atributo me da a entender que se trata de un string, ya que otros atributos que son string están enunciados de la misma forma.

2) Que sea un tipo Integer, ya que el nombre da a entender que se trata de un número con una cantidad de dígitos no superior a veintiséis.

En definitiva, he concluido en que se implemente como un string, y en caso de que al 'lecturer' no le convenza esta solución, atribuirle un patrón, de forma que solo pueda contener dígitos enteros. La segunda alternativa tendría más peso en caso de que se tenga que hacer cálculos con este atributo, lo cual dudo mucho, ya que se trata de un simple identificador.

- **Summary:** Contiene una idea general del objetivo y funciones de la empresa. No puede estar en blanco, y no puede exceder los 101 caracteres. En definitiva, he concluido que este atributo se trata de una cadena de caracteres, con una anotación `@Length(min = 0, max = 101)` y un `@NotBlank`.
- **Optional Link:** Contiene un enlace de la página de la empresa. Tal y como se ve en el enunciado, este atributo es opcional y se trata de un enlace, por lo que no es necesario aportarlo. Lo único que se exige de este atributo es que tenga un patrón propio de una URL. Esto se puede conseguir con la restricción `@URL` implementado por javax.

Requisito 2:Entidad Practicum

2)A practicum helps put hands-on courses into practice in the context of a real company. The system must store the following data about them: a code (pattern "[A-Z]{1,3}[0-9]{0-9}{3}", not blank, unique), a title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), some goals (not blank, shorter than 101 characters), and an estimated total time (in hours, computed from the corresponding sessions plus/minus 10%)

El cliente nos pide que guardemos información sobre las prácticas que ofrece una empresa, las cuales tienen los siguientes atributos:

- **Code:** un código único que identifique a cada práctica. Este atributo requiere que siga un patrón específico; [A-Z]{1,3}[0-9]{0-9}{3}, es decir, primero puede contener una, dos o tres letras de la A a la Z, seguido de un número de uno al nueve, y por último, otros tres últimos números comprendidos en el mismo rango que el anterior. Este atributo tampoco puede ser vacío.

Para abordar este problema, he decidido aplicar la restricción @Pattern(regex="") y dentro de regex vendría el patrón especificado arriba pero adaptado a la gramática que establece java: [A-Z]{1,3}\\d\\d{3}, siendo \\d un dígito latino del 0 al 9.

Para que el atributo sea único, he usado la anotación @Column(unique=true), que permite definir aspectos importantes sobre las columnas, entre las cuales se incluye la unicidad del mismo.

Por último, para que el atributo no esté vacío, he usado la anotación @NotBlank mencionado anteriormente

- **Title:** contiene el título de la práctica. Este atributo no puede ser vacío y debe contener una longitud máxima de 76 caracteres. Este atributo ha sido implementado como un string con las anotaciones @NotBlank y @Length(min = 0, max = 76)
- **Abstract:** contiene el abstracto de la práctica. Este atributo no puede ser nulo y debe contener menos de 101 caracteres. Este atributo ha sido definido como un string con las anotaciones @Length(min = 0, max = 101) y @NotBlank
- **Goals:** contiene los objetivos que se pretenden alcanzar con esta práctica. Este atributo no puede ser vacío y debe contener menos de 101 caracteres. Este atributo ha sido definido como un string con las anotaciones @Length(min = 0, max = 101) y @NotBlank
- **Estimated Time:** contiene el tiempo total estimado de cada práctica, calculada a partir del tiempo de cada sesión asociada a la misma +- un 10%. Como se puede ver, este es un atributo derivado, ya que se parte de otros atributos para calcularlo. Como este atributo se trata de un tiempo estimado en horas y minutos, he llegado a la conclusión de que se trata de un atributo tipo

Date,añadiendo una notacion Temporal(value = TemporalType.TIME),para que de esta forma el framework detecte que se trata de un atributo tipo TIME con hora y minutos y segundos.

En cuanto al cálculo de este atributo,he pensado que se podría hacer a partir de la información de un dashboard que contenga el tiempo que toma cada sesión,para luego seleccionar el tiempo de cada uno de las sesiones relacionadas con una práctica,aplicando la diferencia o la suma de ese porcentaje que se exige en el enunciado(10% en este caso).

Requisito 3:Entidad Session

- 1) The system must store the following data about the sessions in a practicum: a title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), a time period (at least one week ahead, at least one week long), and an optional link with further information.

El cliente nos pide que guardemos la siguiente información acerca de las sesiones que se llevan a cabo de una práctica.Tiene que contener los siguientes atributos:

- **Title:** Contiene el título de la sesión.Este atributo no puede ser vacío y no puede contener más de 76 caracteres. Se entiende que este atributo se trata de una cadena de caracteres ,con las anotaciones @Length(min = 0,max = 76) y @NotBlank mencionados anteriormente
- **Abstract:** Contiene el abstracto de la sesión.Este atributo no puede ser vacío y no puede pasar de los 101 caracteres. He implementado este atributo como un string y he incluido las anotaciones @Length(min = 0,max = 101) y @NotBlank
- **time period:** fecha y hora en la que se va a impartir la sesión.Esta fecha debe durar una semana y debe ser la semana siguiente la la creación de la misma.Este atributo se trata de un atributo Date que contiene una estampa de tiempo,para ello he usado la anotación @Temporal(value = TemporalType.TIMESTAMP),para establecer el formato en yyyy-MM-dd HH:mm:ss.

Para representar que se trata de una período de tiempo,he dividido este atributo en dos subatributos:

- timePeriodStart:Fecha inicial del período de tiempo.Obviamente,esta fecha debe ser anterior al fin del período
- timePeriodEnd:Fecha inicial del período de tiempo.Esta fecha debe distanciarse del inicio al menos una semana,puesto que el período debe tener una duración de al menos una semana.
- **Optional link:** Contiene un enlace de la sesión. Tal y como se ve en el enunciado, este atributo es opcional y se trata de un enlace,por lo que no es necesario aportarlo. Lo único que se exige de este atributo es que tenga un patrón propio

de una URL. Esto se puede conseguir con la restricción @URL implementado por javax.

Requisito 4:Company Dashboard

- 1) The system must handle company dashboards with the following data: total number of practica regarding theory or hands-on courses grouped by month during the last year; average, deviation, minimum, and maximum period length of the sessions in their practica; average, deviation, minimum, and maximum period length of their practica.

En este requisito nos piden que implementemos un dashboard que contenga la información:

- Número total de practicas teóricas, agrupado por mes
- Número de cursos impartidos, agrupado por mes
- Media de duración de las sesiones
- Desviación de duración de las sesiones
- Mínimo de duración de las sesiones
- Máximo de duración de las sesiones
- Media de duración de las prácticas
- Desviación de duración de las prácticas
- Mínimo de duración de las practicas
- Máximo de duración de las practices

Para satisfacer este requisito, he decidido usar una clase que herede de AbstractForm, un objeto que guarda información que no es persistida en base de datos. Dicha información que queremos manejar van a convertirse en atributos de dicha clase:

- TotalNumberPracticumTheory: de tipo Integer
- TotalNumberCoursesHandsOn: de tipo Integer
- AverageSessionPeriodLength: de tipo Date
- DeviationSessionPeriodLength: de tipo date
- MinimunSessionPeriodLength: de tipo date
- MaximumSessionPeriodLength: de tipo date
- AveragePracticumPeriodLength: de tipo date
- DeviationPracticumPeriodLength: de tipo date
- MinimunPracticumPeriodLength: de tipo date
- MaximumPracticumPeriodLength: de tipo date

De esta forma, podemos guardar esta información para luego utilizarla para crear la funcionalidad de algunos atributos derivados y validaciones que sean necesarias.

Conclusiones

En este sprint hemos aprendido a crear las entidades de nuestra aplicación, para luego persistirlas en la base de datos, con sus respectivos datos de prueba con rangos diferentes de valores para poder probar que se cumplen nuestros requisitos. Es cierto que han habido algunas ambigüedades a las que me he tenido que enfrentar aportando alternativas y una solución en base a estas.

Bibliografía

<https://getquipu.com/blog/que-es-el-vat-y-como-tramitar-en-espana/>