

MEMORIA MÓDULO OLTP

Ejemplo de proyecto Spring-MVC y JPA

“Aplicación OLTP para la gestión
de reddit”

Miguel Ferreiro Díaz

Índice

1.	Breve descripción general del proyecto	1
2.	Descripción de las entidades y relaciones finalmente incluidas.....	1
3.	Descripción de las historias de usuario	2
4.	Descripción de los elementos que conforman la implementación realizada	3
5.	Instrucciones de compilación y uso	6
6.	Conclusiones	7

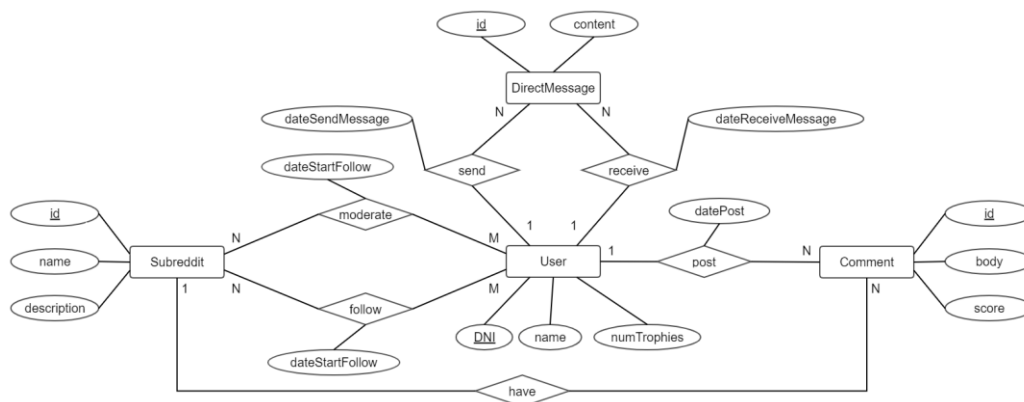
1. Breve descripción general del proyecto

El proyecto planteado trata de ofrecer diferentes funcionalidades para conseguir cubrir las necesidades iniciales de la gestión de una plataforma social como Reddit. Se desarrollará una aplicación Web Spring MVC donde se aportarán soluciones a las historias de usuario planteadas en la Sección 3.

En las historias de usuario se priorizan las funcionalidades de crear la base de gestión de los usuarios y de los subreddits. Además, se ha incluido en la aplicación la posibilidad de que un usuario siga a un subreddit para que se desarrollen las funcionalidades de publicar comentarios en un subreddit, los seguidores sean notificados.

2. Descripción de las entidades y relaciones finalmente incluidas

A continuación, se muestra el diagrama entidad relación diseñado para este proyecto.



En el diagrama se muestran las relaciones asociadas a la interacción de un usuario dentro de la plataforma reddit. Entre estas se encuentran, las acciones de seguir y moderar un subreddit por parte de un usuario, guardando el instante de tiempo en la que estas se producen. Por otro lado, el usuario puede enviar y recibir mensajes de forma directa por parte de otro usuario. Por último, el usuario publica comentarios en un subreddit determinado en un instante de tiempo.

Para la realización de este proyecto, las entidades y relaciones en las cuales se ha centrado el desarrollo de la aplicación Spring son aquellas referentes a usuarios y subreddits, en concreto centrándose en la entidad de que un usuario puede seguir a varios subreddits.

La elección de esta relación se ha basado en que se trata de la más compleja y completa en comparación con el resto. La realización, tanto de la gestión relativa a usuarios, subreddits y seguidores, permite profundizar en el manejo de relaciones muchos a muchos con atributos de por medio y da la opción de descubrir la necesidad de tener que utilizar una nueva clase que representa esta relación y sea la forma en la que se identifica que un usuario sigue a un subreddit a la hora de realizar los mapeos correspondientes para obtener o modificar referente a esta entidad.

3.Descripción de las historias de usuario

En esta sección se enumeran las historias de usuario que forman el Backlog de producto, tal y como la metodología Scrum indica, siguiendo una misma estructura:

- Nombre breve y descriptivo.
- Descripción de la funcionalidad.
- Criterio de validación y verificación que determina si considerar terminado y aceptable el desarrollo de la funcionalidad descrita.

HU01 Crear la gestión de usuarios

Como usuario de la plataforma Reddit
quiero tener los mecanismos
para poder manejar su información

Criterios de aceptación:

- Se debe poder añadir un nuevo usuario.
- Se debe poder editar un usuario existente en la plataforma.
- Se debe poder listar usuarios.
- Se debe poder eliminar la información de un usuario existente.

HU02 Crear la gestión de subreddits

Como usuario de la plataforma Reddit
quiero poder crear un subreddit
para poder tener un espacio para que los usuarios puedan seguir su actividad.

Criterios de aceptación:

- Se debe poder añadir un nuevo subreddit.
- Se debe poder editar un subreddit existente en la plataforma.
- Se debe poder listar subreddits.
- Se debe poder eliminar la información de un subreddit existente.

HU03 Crear la gestión de seguidores

Como usuario de la plataforma Reddit
quiero poder añadir un seguidor a un subreddit existente
para poder que los seguidores estén al tanto de las últimas novedades

Criterios de aceptación:

- Un usuario puede ser seguidor de un subreddit.
- Un usuario puede dejar de ser seguidor de un subreddit.

4. Descripción de los elementos que conforman la implementación realizada

Se incluirán los archivos más relevantes en el desarrollo de la aplicación. En primer lugar de describirán los archivos asociados tanto a Spring Core como a Spring Data que se han utilizado en el desarrollo de la aplicación, ignorando aquellas entidades que no se han incluido finalmente.

java/com/si/entities/

- Subreddit.java: Entidad que representa una comunidad de Subreddit que tiene de atributos un nombre y una descripción.
- User.java: Entidad que representa un usuario de Subreddit que tiene de atributos su DNI, su nombre y el número de trofeos que tiene.
- UserFollowSubreddit.java: Entidad que representa el seguimiento de un usuario a un subreddit determinado en un momento dado.
- UserFollowSubredditID.java: Entidad que representa el identificador de la relación entre un seguidor y el subreddit al que sigue.

java/com/si/services/

- SubredditService.java: Clase que permite definir los métodos de gestión de vida de los objetos de la entidad Subreddit.
- SubredditServiceImpl.java: Clase que implementa los métodos definidos en SubredditService.java dónde se utilizan los métodos implementados que aporta Spring en los DAO y los que se construyen propiamente para realizar el mapeo de entidades a partir de las definiciones de los métodos en la clase SubredditDAO.java.
- UserFollowSubredditService.java: Clase que permite definir los métodos de gestión de vida de los objetos de la entidad UserFollowSubreddit.
- UserFollowSubredditServiceImpl.java: Clase que implementa los métodos definidos en UserFollowSubredditService.java dónde se utilizan los métodos implementados que aporta Spring en los DAO y los que se construyen propiamente para realizar el mapeo de entidades a partir de las definiciones de los métodos en la clase UserFollowSubredditDAO.java.
- UserService.java: Clase que permite definir los métodos de gestión de vida de los objetos de la entidad User.
- ServiceImpl.java: Clase que implementa los métodos definidos en UserService.java dónde se utilizan los métodos implementados que aporta Spring en los DAO y los que se construyen propiamente para realizar el mapeo de entidades a partir de las definiciones de los métodos en la clase UserDAO.java.

java/com/si/daos

- SubredditDAO.java: Clase que se utiliza para definir el mapeo sobre las entidades Subreddits filtrando por su nombre.
- UserDAO.java: Clase que se utiliza para definir el mapeo sobre las entidades Users filtrando por su nombre.

- UserFollowSubredditDAO.java: Clase que se utiliza para definir los diferentes mapeos que no vienen definidos por defecto por Spring y se realizarán en las entidades UserFollowSubreddit utilizando la clase UserFollowSubredditID como identificador de estas.

java/com/si/controllers/

- DefaultController.java: se trata del controlador que se invoca al acceder a la ruta base de la página web y muestra el contenido del index.html.
- SubredditController.java: se trata del controlador que se invoca al acceder a la ruta “/subreddits” y cuenta con las siguientes funciones:
 - *listSubreddits*: se encarga de listar los subreddits almacenados.
 - *refreshListSubreddits*: se encarga de listar los subreddits que contengan el nombre que se pasa por parámetro. En el caso de que no se le pase ningún parámetro, se listarán todos los subreddits.
 - *removeSubreddit*: se encarga de eliminar el subreddit asociado al id que se recibe. Para realizar esta acción, se comprueba previamente que este subreddit no tenga ningún seguidor.
 - *addSubredditView*: se encarga de mostrar la vista de añadir un nuevo subreddit.
 - *createSubreddit*: se encarga de añadir un nuevo subreddit, siempre y cuando los atributos recibidos cumplan sus validaciones asociadas y no ya exista un usuario con el dni indicado.
 - *editSubredditView*: se encarga de mostrar la vista de editar un subreddit existente.
 - *refreshSubreddit*: se encarga de editar la información de un subreddit, siempre y cuando los atributos recibidos cumplan sus validaciones asociadas.
- UserController.java: se trata del controlador que se invoca al acceder a la ruta “/users” y cuenta con las siguientes funciones:
 - *listUsers*: se encarga de listar los usuarios almacenados.
 - *refreshListUsers*: se encarga de listar los usuarios que contengan el nombre que se pasa por parámetro. En el caso de que no se le pase ningún parámetro, se listarán todos los subreddits.
 - *removeUser*: se encarga de eliminar el usuario asociado al dni que se recibe. Para realizar esta acción, se comprueba previamente que este usuario no siga a ningún subreddit.
 - *addUserView*: se encarga de mostrar la vista de añadir un nuevo usuario.
 - *createUser*: se encarga de añadir un nuevo usuario, siempre y cuando los atributos recibidos cumplan sus validaciones asociadas.
 - *editUserView*: se encarga de mostrar la vista de editar un usuario existente.
 - *refreshUser*: se encarga de editar la información de un usuario, siempre y cuando los atributos recibidos cumplan sus validaciones asociadas.
- UserFollowSubredditController.java: se trata del controlador que se invoca al acceder a la ruta “/follows” y cuenta con las siguientes funciones:
 - *listUsersFollowSubreddit*: se encarga de listar los seguidores de los subreddits almacenados.
 - *refreshListUsersFollowSubreddit*: se encarga de listar los seguidores de los subreddits que contengan el nombre del usuario o el nombre del subreddit que

se pasa por parámetro. En el caso de que no se le pase ningún parámetro, se listarán todos los seguidores.

- `removeUserFollowSubreddit`: se encarga de eliminar un seguidor de un subreddit.
- `addUserFollowSubredditView`: se encarga de mostrar la vista de añadir un seguidor a un subreddit indicando los usuarios y subreddits existentes.
- `createUserFollowSubredditView`: se encarga de añadir un nuevo seguidor a un subreddit.

java/com/si/config/

- `MvcConfigurer.java`: se trata de una clase anotada con `@Configuration` y que me permite controlar el cambio de idioma en la página y poder establecer los mensajes establecidos en las validaciones de las entidades.

En segundo lugar, se describirán los elementos asociados a las vistas de Spring MVC utilizando las plantillas thymeleaf.

resources/

- `application.yml`: fichero cuya función es definir la ruta de la carpeta dónde se encuentran los ficheros con los textos utilizados para la internacionalización.

resources/i18n/

- `messages.properties`: fichero que contiene la traducción de los textos al español.
- `messages_en.properties`: fichero que contiene la traducción de los textos al inglés..

resources/static/resources/

- `css/`: se almacenan los archivos .css asociados a bootstrap, jquery y a algunos estilos personalizados de la página guardados en el archivo `styles.css`
- `img/`: se almacenan las imágenes que se muestran en la página web.
- `js/`: se almacenan los archivos .js asociados a bootstrap y jquery.

resources/templates/

- `follows/`: carpeta que almacena tanto el fragmento de la vista de listar los seguidores de los subreddits como de añadir/editar un nuevo seguidor a un subreddit.
- `fragments/`: carpeta que almacena los fragmentos de vista comunes a todas las vistas.
- `subreddit/`: carpeta que almacena tanto el fragmento de la vista de listar subreddits como de añadir/editar un subreddit.
- `user/`: carpeta que almacena tanto el fragmento de la vista de listar usuarios como de añadir/editar un usuario.
- `error.html`: vista que se muestra en el caso de que se produzca algún tipo de error en la página.
- `index.html`: se trata de la primera vista que se muestra al entrar en la página web.

5. Instrucciones de compilación y uso

Para la construcción de este proyecto se necesita un servidor de base de datos MySQL y tener una versión de Maven superior a 3.5.x. A continuación, será necesario crear la base de datos que utilizará la aplicación a través de los comandos:

```
> mysql -u root -p [pedirá la contraseña de MySQL]
> mysql> create database reddit_si;
> mysql> grant all privileges on reddit_si.* to si@localhost identified by "si";
```

Adicionalmente, puede ser necesario establecer un formato de fecha compatible:

```
> mysql> set @@global.time_zone = '+00:00';
> mysql> set @@session.time_zone = '+00:00';
```

Una vez configurada la base de datos será situarse en la carpeta del proyecto y lanzar la aplicación a través del comando:

```
> mvn spring-boot:run
```

o mediante:

```
> mvn install
> java -jar target/pedidos-spring-1.0.jar
```

Por último, para acceder a la aplicación será necesario introducir en el navegador la siguiente dirección: <http://localhost:8080/>.

6. Conclusiones

Sobre el proyecto realizado para esta asignatura es una buena vía para tener contacto con el popular framework Spring y conocer de cerca los principales componentes que ofrece. La utilización de los módulos como Spring Boot o Spring Data, permite facilitar la vida al programador permitiendo ahorrarse gran cantidad de líneas de código y de tiempo en programarlas. También uno de los módulos que más útiles que he conocido es el Spring security, el cual aporta muchas herramientas muy útiles para conformar una estructura de grupos de usuario a través de los roles y controlar los recursos que se le muestran a cada uno de ellos.

Algunas de los inconvenientes que he visto al usar un framework de estas características, es principalmente la cantidad de anotaciones y ficheros de configuración. Ciertamente es que resultan muy útiles a la hora de programar, pero para realizar un uso correcto de estos elementos es preciso de un buen conocimiento de que se necesita para cada situación, cómo se usa y qué efectos tiene utilizar una cosa u otra.

Como conclusión, la práctica creo que ha servido para realizar un primer contacto con esta tecnología y ha permitido ver cómo se hace un manejo básico de las funcionalidades que aporta. También he conocido el motor de plantillas thymeleaf, el cual he podido experimentar la facilidad de reutilizar código a través del uso de fragmentos. Por otro lado, en la realización de la práctica he tenido alguna dificultad a la hora de realizar el mapeo de atributos en relaciones que tenía como clave primaria dos entidades ya que desconocía la necesidad de crear una nueva entidad que recogiese las claves de estas dos entidades. Por último, en esta práctica he podido investigar funcionalidades útiles en muchas aplicaciones como es la internacionalización de los textos que se muestran en la página y la validación de los datos de entrada, a través de las anotaciones indicadas en los atributos de cada entidad.

El desarrollo de este proyecto se puede encontrar en el repositorio de GitHub: <https://github.com/miferreiro/reddit-oltp>.