

MEMORIA DE LA PRÁCTICA

Programación Avanzada



UNIVERSIDADE DA CORUÑA



Carla Salgado López
Diego Sánchez Vázquez

Contenido

Introducción.....	2
Arquitectura global	2
Capa acceso a dato y capa de lógica de negocio	2
Capa de servicios	2
Capa de interfaz de usuario.....	2
Modelo	3
Clases persistentes	3
Interfaces de los servicios ofrecidos por el modelo	3
Otros aspectos	4
Interfaz gráfica	5
Problemas conocidos	5

Introducción

En esta práctica se implementa una aplicación web que gestiona apuestas deportivas. Se utilizan metodologías de desarrollo ágil para su implementación, como los son *Hibernate*, *Spring* o *Tapestry*. La aplicación sigue una arquitectura en capas, con una capa de acceso a datos, una capa de lógica de negocio, una capa de servicios y una capa de interfaz de usuario.

No se han implementado los trabajos tutelados.

Arquitectura global

La aplicación sigue una arquitectura en capas que se compone de la capa de acceso a datos, la capa de lógica de negocio, la capa de servicios y la capa de interfaz de usuario.

Aunque no se le dedica un apartado específico en la memoria, se debe tener consciencia del *módulo util*, en el que se implementan funcionalidades de apoyo a las otras capas como excepciones, directivas de autenticación, gestión de *cookies*, etc.

Capa acceso a dato y capa de lógica de negocio

Está compuesta por los paquetes del *módulo model* de *src/main/java* que no terminan en *service*, en el que se implementan las operaciones que permiten gestionar todo lo referente a las clases persistentes creadas para gestionar la web de apuestas deportivas y la forma de recuperación y guardado de datos.

Capa de servicios

Está compuesta por los paquetes del *módulo model* de *src/main/java* que terminan en *service*, en estos paquetes se implementan las funciones necesarias para dar servicio a la capa de interfaz de usuario para implementar los casos de uso extraídos del enunciado de la práctica.

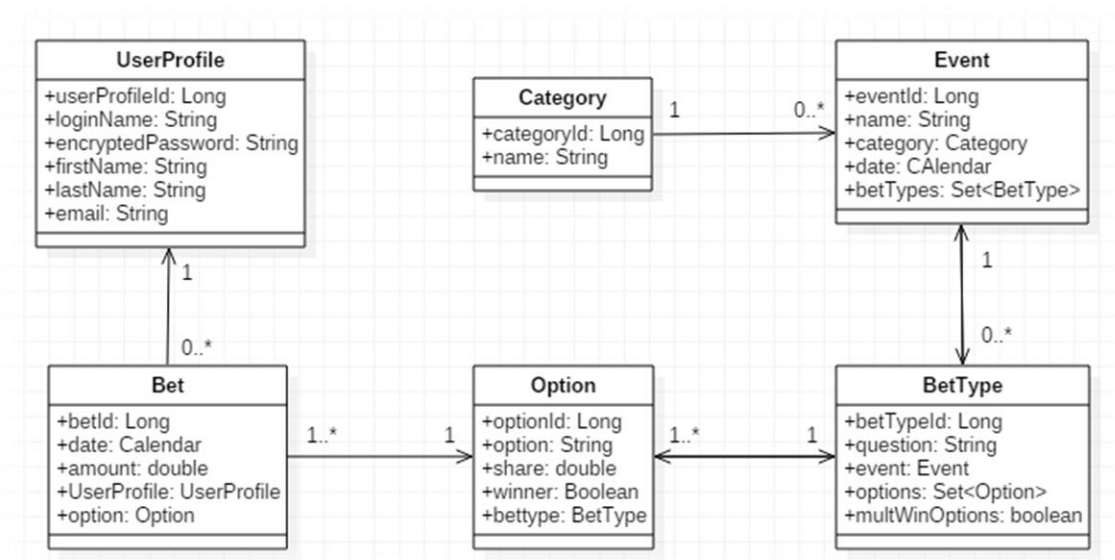
Capa de interfaz de usuario

Está compuesta por los paquetes del *módulo web* de *src/main/java*, en el que se encuentran las clases *.java* que necesita *Tapestry* para la implementación de las páginas de la aplicación, y del *módulo web* de *src/main/resources* en el que se encuentran los archivos *.tml* para la implementación de las páginas de la

implementación, los archivos *.properties* para implementar la internacionalización de la aplicación y el archivo de *layout* que se utiliza como plantilla para organizar la localización de los componentes en las páginas de la aplicación.

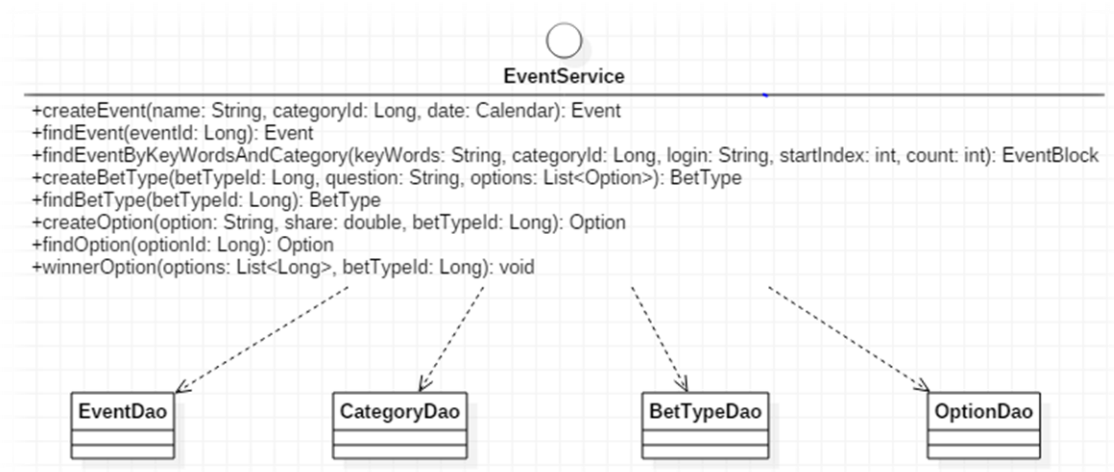
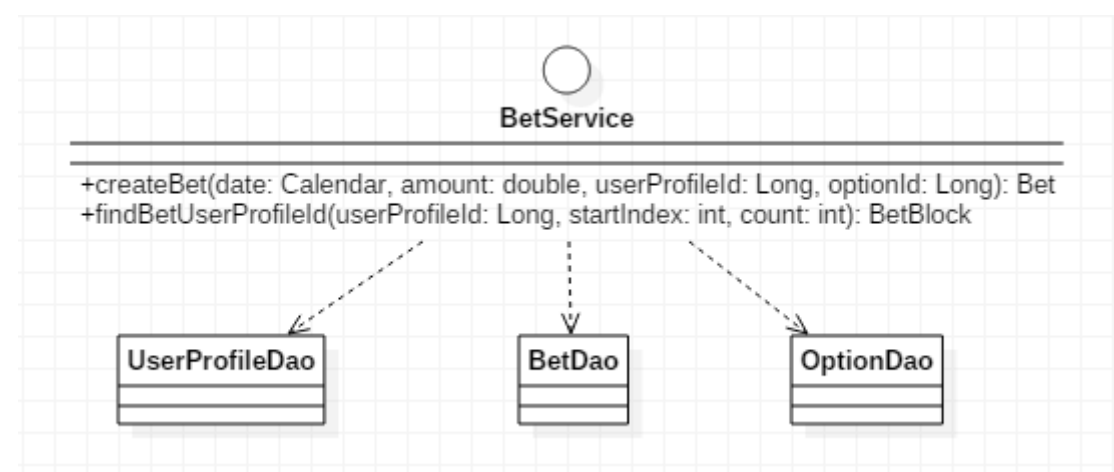
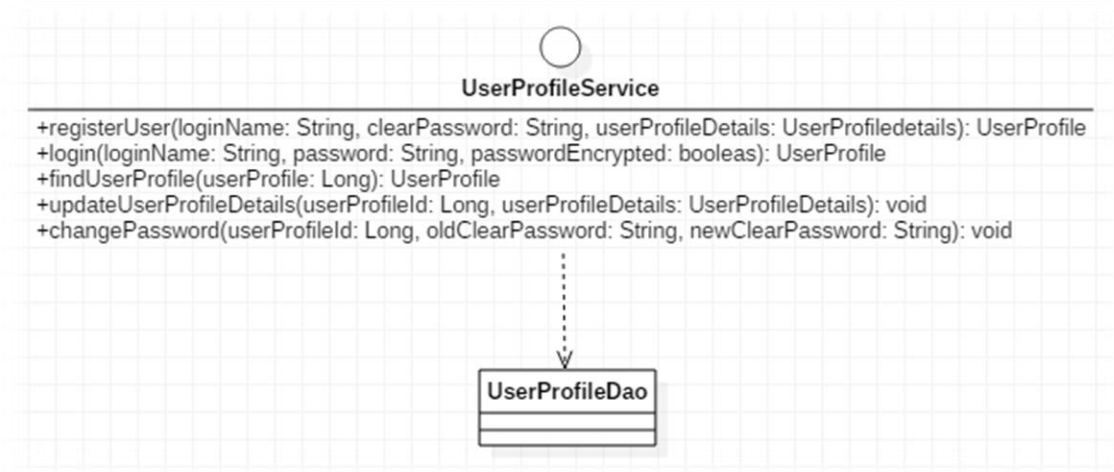
Modelo

Clases persistentes



Interfaces de los servicios ofrecidos por el modelo

Los casos de uso de la capa modelo se han dividido en tres fachadas o grupo diferentes. Se ha decidido dividir los casos de uso en tres interfaces de servicios: la que gestiona los casos de uso relacionados con los usuarios, la que gestiona los casos de uso relacionados con los eventos y la que gestiona los casos de uso relacionados con las apuestas. Se ha decidido utilizar esta organización porque son las tres clases permanentes sobre las que se centra la aplicación web.



Otros aspectos

También se hace uso de otras clases como las que se utilizan para poder utilizar el patrón page by page iteration, excepciones, etc. No se han incluido en los diagramas para facilitar la comprensión y sencillez de los mismos.

Interfaz gráfica

La interfaz gráfica se ha desarrollado de manera que sea intuitiva para el usuario.

En la parte superior aparecen las operaciones básicas a las que el usuario tiene acceso:

- Usuario no autenticado: buscar eventos, preferencias de idioma y autenticación.
- Usuario autenticado: buscar eventos, preferencias de idioma y acceso al perfil, donde además podrá consultar el estado de sus apuestas.
- Administrador: creación y búsqueda de eventos, preferencias de idioma y acceso al perfil.

En el cuerpo de la página se mostrarán los campos para mostrar y la información requerida por el usuario, así como enlaces a otras páginas para obtener información adicional y acceso a otras operaciones en función de la fecha actual y el estado del evento elegido.

Problemas conocidos

Al crear un evento, la fecha se debe introducir con el formato indicado por la aplicación.