

Ejercicios del contenedor de beans de Spring

Índice

1 Refactorización de la capa de negocio en beans de Spring.....	2
---	---

En las plantillas de la sesión se encuentra una aplicación denominada AmigosSpring. El objetivo de esta y del resto de sesiones del módulo consistirá en ir refactorizando la aplicación para introducir características de Spring.

Se trata de una aplicación en la que los usuarios pueden entrar con login y password (o registrarse si no tienen uno) , ver los otros usuarios en el sistema y mandarles mensajes.

En cuanto a la implementación, la capa de negocio consta de tres objetos (singleton):

- **Fachada:** la clase que actúa como punto de entrada de la capa de negocio, contiene a "grosso modo" un método por cada caso de uso del sistema. Esta clase no implementa lógica de negocio, para ello se apoya en dos interfaces auxiliares: `GestorUsuarios` y `GestorMensajes`
- `GestorUsuarios`: clase que implementa todos los casos de uso relativos a usuarios (registrar, buscar, etc).
- `GestorMensajes`: implementa todos los casos de uso relativos a mensajes (enviar, leer, etc).

1. Refactorización de la capa de negocio en beans de Spring

Los objetos de la capa de negocio no aprovechan las posibilidades que ofrece el contenedor de Spring. Vamos a transformarlos en beans de Spring y a configurarlos adecuadamente. Para ello habrá que hacer dos cosas:

1. En el código Java, cuando un objeto necesita de otro y obtiene una instancia con `getInstante`, eso debe cambiar. El objeto principal debe contener una variable del tipo del objeto auxiliar y confiar en que el contenedor la instanciará en tiempo de ejecución via el *setter* correspondiente.
2. Crear un fichero de configuración de beans de Spring y definir en ellos los beans de la capa de negocio. Tener en cuenta que además de las dependencias entre beans, hay que inicializar parámetros de algunos de ellos.
3. El `DataSource` al que ahora se accede con la clase `FuenteDatos` debe transformarse también en un bean de Spring.

