

ANEXO: FUTURIBLES

FORMATTERS Y PARSERS

Mejorar la implementación de los formatters y parsers, usando la clase *Formatter* de Java para evitar usar varias cadenas al especificar el formato. Puede encontrar más información en el anexo de problemas encontrados.

ACTUALIZAR LIBRERÍAS

Sería interesante actualizar a Spring 4. Esta última versión permitía mayor flexibilidad en cuanto a formatters y parsers. Concretamente, permitía utilizar las clases *Formatter*, *Printer* y *Parser* con un objeto *WebDataBinder*, lo que podría facilitar en gran medida la implementación de formatters y parsers, puesto que nos podríamos olvidar de los editores.

También sería interesante actualizar las librerías de Hibernate para poder usar la última versión de Hibernate-search.

Investigar la posibilidad de actualizar la versión del componente Datatables. La versión actual que se está utilizando en la plantilla es una versión ya considerada como legacy. Alguno de los problemas que esta versión presenta, podrían solucionarse con la actualización a la nueva versión. Implica cambios en diversos lugares de la aplicación, por lo que requiere cierta tarea.

HACER QUE LAS DATATABLES SEAN RESPONSIVE

Investigar la posibilidad de hacer que las datatables sean responsive. Se intentó realizar y no logramos la manera de hacerlo cuando la tabla cuenta con bastantes columnas. Una posible solución sería hacer uso de una fila que permita desplegar mayor cantidad de detalles cuando se selecciona. Aquí hay una referencia al respecto: [API Datatables Row Details](#). Otra posibilidad es cambiar la librería que se utiliza para mostrar las tablas, ya que como se ha podido comprobar ha sido una fuente de diversos problemas.

MEJORAR LA CACHE

Existe un problema con la caché, y es que se cachean de forma individual. El problema viene dado al cachear una colección: si un método modifica un objeto que estaba en un colección en en caché, ese objeto no se modificará en la colección, ya que Spring cachea la colección entera, no los objetos de forma individual.

REFACTORIZACIÓN DEL FRAMEWORK

Investigar la manera en que la persistencia, búsquedas full-text, seguridad, etc, se puedan especificar mediante ficheros de configuración externa, evitando las etiquetas en el modelo de dominio.

SOFTWARE FACTORY

A partir de un XML que represente un UML, generar el modelo de dominio y las clases necesarias para casos de uso básicos.

SEGUIR INVESTIGANDO DROOLS Y QUARTZ

Investigar el uso de reglas de negocio sensibles al paso del tiempo. Ahora mismo se ejecutan las reglas bajo demanda. Conseguir es interesante estudiar cómo se podría configurar Drools para que estas se ejecutasen automáticamente, en tiempo real. Estudiar estos aspectos de Drools más profundamente. Lanzar las reglas periódicamente mediante Quartz, recoger los objetos de la base de datos, cargarlos en memoria y ejecutar las reglas, por ejemplo, una vez al día.

Además, se podría investigar cómo hacer que se puedan actualizar reglas de negocio en tiempo real. Hemos intentado configurar el proyecto usando un objeto llamado “*KieScanner*”, que se encarga de comprobar periódicamente si ha habido alguna nueva versión de un *KieModule*. No hemos podido hacer uso de este objeto puesto que necesitaba que el *KieContainer* recibiese el *ReleaseId*, especificando el *group-id*, *artifact-id* y la versión del proyecto, y no ha habido forma de hacerlo funcionar. Se pensó una posible solución, y es añadiendo el proyecto al repositorio de Maven. Probar esto último.

INVESTIGAR ESCALABILIDAD EN SERVIDORES DE APLICACIONES Y BASE DE DATOS

Lograr la forma de que el servidor de base de datos o el de aplicaciones esté replicado, para balancear la carga de trabajo. Intentar desplegar el sistema sobre varios servidores de base de datos o aplicaciones.

INVESTIGAR E INTEGRAR SPRING WORKFLOW EN LA PLANTILLA

Spring Workflow permite definir flujos de trabajo. Ahora mismo se hace con casos de uso totalmente independientes. Mediante jBpm es posible definir dichos flujos, establecer la ligadura que puede existir entre distintos casos de uso.