Control check: Nivel C

Entidades	Actores	Quolet
Mensaje FixUp Warranty Category Application Phase Dashboard	Administrador HandyWorker Customer	Ticker Moment Body Picture IsFinal

Requisitos que cumplir:

- Ticker: Seguirá un patrón XXXX
- Body: Deberá ser más pequeño que XXXX y not blank
- Picture: Deben ser URL y es opcional
- Quolet: Se podrá editar, listar, mostrar y modificar o borrar si está en modo borrador
- Moment: Código de colores según:
  moment < un mes --> color X
  moment < dos meses --> color Y
  otro caso --> color Z
- Moment: El formato de la fecha varía según el idioma:
  español --> dd-mm-yy hh:mm
  inglés--> yy/mm/dd hh:mm
- Query: La media y la desviación típica de Quolet / FixUp
- Query: Ratio de publicadas / total
- Query: Ratio de no publicadas / total
- Readme.txt: Documento para comentar decisiones que se tomen respecto al enunciado
- Populate: Debe incluir: ¿?

- Modelo conceptual: PDF
- Modelo de dominio: PDF
- SQL y WAR: Pre-production
- CleverCloud: Desplegar y documento con la información necesaria

NOTA: Enviar todo en un archivo zip, cuyo nombre debe ser "Apellidos-Nombre-xxx.zip", donde xxx es su clave se control. Se debe mandar el proyecto Eclipse / Maven no el espacio de trabajo.

Suponemos que la entidad del nivel C será FixUp, el nombre de la entidad nueva es Quolet y el actor que estará relacionado con la entidad es Customer. Por otro lado, un Customer podrá ver todas sus Quolets y un HandyWorker verá todas las Quolets que existan y están en modo final.

Las relaciones entre FixUp y Quolet y entre Quolet y Customer serán bidireccionales por lo que tenemos que añadir en FixUp y en Customer una colección de Quolets con sus getters y setters. En el edit de fixUp tenemos que añadir en la vista "form:hidden path="quolet" y crear los dos convertidores.

Paso 1: Crear la entidad nueva en el domain (Quolet)

- Añadimos los atributos (ticker, moment, body, picture, isFinal)
- Añadimos las relaciones con customer y fixUp
- Añadimos los getters y setters
- Añadimos en el get de picture @URL
- Añadimos en el get de body @size (Modificar el tamaño según el enunciado)

Paso 2: Crear el repositorio y el servicio de la nueva entidad (Quolet)

- Método para crear una quolet (Aquí utilizamos un método auxiliar "randomTicker" es un generador de ticker, se deberá modificar según el enunciado)
- Método para guardad una quolet
- Método listQuoletsByCustomer, devuelve todas las Quolets de un customer dado.
-Método listQuolets, devuelve todas las Quolets que estén en estado final
- Método para borrar una Quolet
- Método para encontrar una Quolet

Paso 3: Controlador y vista de un HandyWorker (list)

- Controlador (list): Si el actor es un HandyWorker este podrá ver todas las Quolets que estén en modo final, si no, se redirigirá a la página de bienvenida.
- Vista (list): list_handyWorker
    - Creamos la vista
    - Añadimos en "18n-|10n.xml" la entidad Quolet
    - Añadimos en "title.xml" la entidad Quolet
    - Añadimos en el "title.xml" la nueva vista en ambos idiomas
    - Añadimos la referencia en el "security.xml"
    - Añadimos en el "Messages" lo necesario en ambos idiomas
    - Añadimos en el "header.jsp" la referencia

Paso 4: Controlador y vista de un HandyWorker(show)

- Controlador (show):
    - Añadimos la restricción de los colores de la fecha
    - Añadimos la restricción del formato de la fecha según el idioma
- Vista (show): show_handyWorker
    - Creamos la vista
    - Añadimos en el "title.xml" la nueva vista en ambos idiomas
    - Añadimos la referencia en el "security.xml"
    - Añadimos en el "Messages" lo necesario en ambos idiomas

Paso 5: Controlador y vista de un Customer (list)

- Controlador (list): Si el actor es un customer este podrá ver todas sus Quolets, si no, se redirigirá a la página de bienvenida.
- Vista (list): list_customer
    - Creamos la vista
    - Añadimos en el "title.xml" la nueva vista en ambos idiomas
    - Añadimos la referencia en el "security.xml"
    - Añadimos en el "Messages" lo necesario en ambos idiomas
    - Añadimos en el "header.jsp" la referencia

Paso 6: Controlador y vista de un Customer (create - edit)

- Controlador (create y createEditModelAndView): Si el actor es un customer este podrá crear quolets, si no, se redirigirá a la página de bienvenida.
- Controlador (edit y createEditModelAndView): Si el actor es un customer este podrá edit sus quolets siempre que no estén en modo final, si no, se redirigirá a la página de bienvenida. A la hora de crear llamamos a un método auxiliar "hasAnyEntityRequired" que compruebe si existe alguna entidad (fixUp) para crear la quolet, si no existe dicha entidad el botón de editar no aparecerá (echo en la vista).
- Controlador (save y createEditModelAndView):
- Vista (create - edit): edit
    - Creamos la vista
    - Añadimos los botones de show, edit y create
    - Añadimos en el "title.xml" la nueva vista en ambos idiomas
    - Añadimos la referencia en el "security.xml"
    - Añadimos en el "Messages" lo necesario en ambos idiomas

Paso 7: Controlador y vista de un Customer (Delete)

- Controlador (Delete): Si el actor es un customer este podrá borrar una quolet siempre que esté en modo borrador, si no, se redirigirá a la página de bienvenida. Hay que añadir la referencia en el "security.xml"

Paso 8: Controlador y vista de un Customer (show)

- Controlador (show):
    - Añadimos la restricción de los colores de la fecha
    - Añadimos la restricción del formato de la fecha según el idioma
- Vista (show): show_customer
    - Creamos la vista
    - Añadimos en el "title.xml" la nueva vista en ambos idiomas
    - Añadimos la referencia en el "security.xml"
    - Añadimos en el "Messages" lo necesario en ambos idiomas

Paso 9: Querys

-Repositorio: QuoletRepository
- Añadimos las cuatro Querys en el repositorio

-Servicio: QuoletService
- Añadimos en el servicio dichas Querys (método por query)

-AdministratorController: statistics
- Llamamos a los métodos de las Querys de QuoletService

-Vista: administrator > statisticsDashboard.jsp
- Añadimos los cambios de la vista
- Añadimos en el "Messages" lo necesario en ambos idiomas