EXAMEN 1: BECA JAVA

NOMBRE Y APELLIDOS: _____

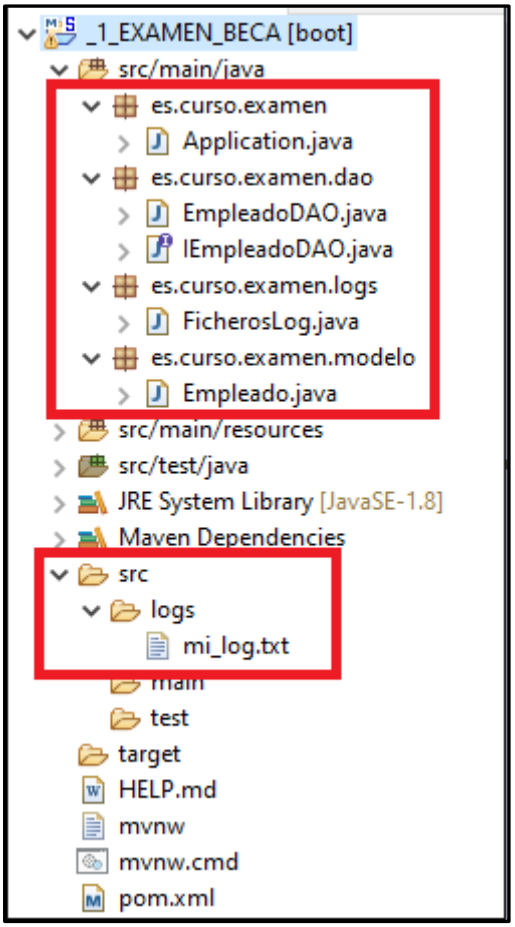
FECHA: 9/07/2021

1. **CREA UNA CARPETA CON EL SIGUIENTE FORMATO: NOMBRE_APELLIDO1_APELLIDO2.**

2. ESTA CARPETA ES LA QUE SE ENTREGA AL FINAL DEL EXAMEN. ***Antes de entregar borrar la carpeta .metadata***

3. Renombrar el proyecto con el siguiente formato:
   EXAMEN1_NOMBRE_APELLIDO1

4. Las preguntas se pueden contestar en un **txt** con el siguiente formato: **EXAMEN1_NOMBRE_APELLIDO1.txt o si tienes Word contesta en este mismo fichero.**



Partimos del siguiente proyecto. El objetivo es hacer un seguimiento de las operaciones que se realizan contra la base de datos (es una simulación, la clase solo lanza mensajes por consola con la operación que realiza) en el paquete es.curso.examen.dao disponemos de una interface y una clase que implementa una serie de operaciones create, delete y update sobre el bean "Empleado" que cumple las convenciones de los JavaBeans (constructor, getters / setters, etc).

Estas clases ya están implementadas y no hay que modificar nada.

Para saber cuándo se graban ,modifican o borran empleados en la base de datos queremos generar un fichero de log que registre la fecha, hora del sistema y una cadena detallando la operación que realiza con los datos del empleado que se vayan a grabar a borrar o actualizar. Este fichero se grabará en la carpeta **src/logs/mi_log.txt**. El código necesario para grabar en el log ya está implementado.

Se encuentra en la clase **FicherosLog** con un método estático que se pasa la cadena a grabar y la ruta al fichero (consulta el fichero: src/logs/mi_logs.txt). Puedes ver un ejemplo de uso en la clase principal del proyecto: **Application.java**. Consulta el formato del fichero de log que viene

de ejemplo. Consideramos que la generación del log es una operación transversal de la aplicación y queremos desacoplarla de las operaciones contra la base de datos, para ello tenemos que hacer una implementación basada en aspectos.

Se pide:

1) Implementar las clases y configuración necesaria para los aspectos. Puedes elegir entre configuración con XML o anotaciones. Se valorará positivamente hacer las dos implementaciones.

2) ¿Qué clases representan los aspectos? ¿Cuáles son los métodos que hay que interceptar?

3) ¿Dado que estamos interactuando con la BBDD que tipo de interceptores nos interesa utilizar?

4) ¿Qué clases tienes que añadir en el contexto de Spring? ¿Es necesario añadir todas las clases que se presentan? Justifica la respuesta.

5) Para comprobar que el programa funciona en la clase Application verás que hay dos métodos: pruebaConXML y pruebaConAnotaciones para que rellenes con tu código. Hay que cargar el contexto que corresponda según la implementación y al menos grabar un empleado, actualizar y borrar para ver que el fichero de log se genera con el formato correcto.

    a. El fichero de log le puedes borrar y se volverá a crear. Recuerda actualizar tu proyecto (refresh o pulsa F5 sobre el proyecto) después de ejecutar el código para que aparezca de nuevo en la carpeta.

    b. Un posible formato del log puede ser:

        09/07/2021 8:12:34.561 CREAR EMPLEADO: 1, Nombre_emp, CAP, Calle …
        09/07/2021 8:17:03.566 BORRAR EMPLEADO: 1
        09/07/2021 8:17:03.566 ÀCTUALIZAR EMPLEADO: 21, Nombre_emp, CAP, Calle

        A la vez irán saliendo los mensajes por consola según se lanzan estas operaciones.

    En caso de que implementes las dos formas puedes copiar este proyecto y renómbralo, pero entrégalo dentro de la carpeta principal con tu nombre y apellidos.

6) ¿Tenemos que hacer alguna modificación en el fichero POM?

7) Adjunta una URL con el proyecto en un repositorio público en GitHub.