

AccesData SL v2.0



Nuestra empresa quiere portar nuestro enunciado anterior a un almacenamiento NoSQL convencidos de que en algunos aspectos puede mejorar. Es por ello que nos piden como conocedores del problema dar con la solución adecuada para ello. Dicho enunciado lo tienes en:

<https://github.com/joseluisgs/AccesoDatos-03-2021-2022/blob/master/Practica/Practica03-2021.pdf>

1. Realizar el Diagrama de Clases asociado, mostrando la semántica, navegabilidad y cardinalidad de las relaciones, justificando la respuesta. A partir de este Diagrama y de las clases generadas, debemos analizar el diseño NoSQL que nos interesa, por ejemplo usando documentos embebidos, listas identificadores, árboles.... Es por ello que un aspecto crucial de esta práctica es asentar las bases de qué queremos y cómo queremos trabajar con ello y se valorará enormemente (es decir, que en 3 hojas no puedes hacerlo), y sobre todo las distintas propuestas de soluciones y por cual os decantáis. Si esta parte no está presente con toda la profundidad necesaria puede invalidar el resto de la práctica.

2 puntos

2. Implementar las clases con la estructura NoSQL con las notaciones JPA necesarias que justifique las restricciones semánticas, de navegabilidad y de estructuras NoSQL propuestas en el punto anterior.

2 puntos

3. Además aparece una nueva entidad **Login**, para que los Programadores puedan identificarse en el sistema. Para ello el **Programador** usará su email y su contraseña en SHA256. **Esta funcionalidad no se podrá usar mediante JPA debiéndose hacer**

usando las librerías y métodos adecuados. En el Login almacenaremos el id del usuario, el instante de indentificación y el token generado (UUID) y si está activo o no. El programador puede identificarse y salir del sistema. Debemos tener en cuenta que queremos mantener un histórico de accesos.

1 punto

4. Implementación de una arquitectura de 3 niveles basada en controlador, servicios y repositorios que permitan resolver la lógica del enunciado propuesto, así como las operaciones CRUD con las entidades relevantes.

3 puntos

5. Los Repositorios deben testearse completamente usando JUnit5. Los Servicios y Controladores usarán un mock de los repositorios usando Mockito y Junit5

2 puntos

6. Además, de las operaciones CRUD de obligatorio cumplimiento con las restricciones semántica indicadas y **solo haciendo uso de ellas lo máximo posible, y minimizando el número de llamadas al servidor** nos interesa saber además las siguientes:

- 1) Obtener de un departamento, los proyectos (información completa) y trabajadores asociados con sus datos completos.
- 2) Listado de issues abiertas por proyecto que no se hayan consolidado en commits.
- 3) Programadores de un proyecto ordenados por número de commits.
- 4) Programadores con su productividad completa: proyectos en los que participa, commits (información completa) e issues asignadas (información completa).
- 5) Obtener los tres proyectos más caros en base a su presupuesto asignado y el salario de cada trabajador que participa.
- 6) Listado de Proyectos con su información completa. Programadores que trabajan en él. Repositorio asociado con su listado de issues y de commits.
- 7) Número de accesos de un Programador e información de los mismos, es decir, fecha en la que se produjo cada uno de ellos.

1,5 punto

6.- Debemos tener en cuenta que necesitamos devolver cualquier salida pedidas en los puntos anteriores en JSON.

1 punto

Nuestro programa debe llamarse con un JAR de la siguiente manera

java -jar accesdata.jar. Se recomienda que el proyecto esté gestionado por Maven.

Se debe entregar:

- **Repositorio GitHub Personal y el de entrega** con la solución en el que incluyas:
 - o Readme explicando el proyecto y el nombre de los integrantes. Usa Markdown y mejora su estilo. Si no perderás puntos por la presentación.
 - o **Código fuente comentado y perfectamente estructurado de los dos proyectos** con JDoc. Además de los gitignore adecuados y que siga el flujo de trabajo GitFlow.
 - o No se deben incluir los ejecutables si no se deben poder crear los **jar**.
 - o **Docker con la Base de Datos que incluya datos de ejemplo para probar la aplicación desde su inicio o fichero de SQLite.**

- o **Documentación en PDF** donde expliques el diseño y propuesta de solución, así como clases y elementos usados haciendo especial énfasis en:

Diagrama de clases y justificación de este: navegabilidad, cardinalidad, expresividad y dependencias.

Diseño NoSQL con alternativas y justificación de diseño elegido.

Arquitectura del sistema y patrones usados.

Explicación de cómo se acceden a los datos.

La no entrega de este fichero invalidará la práctica.

Esta documentación es vital y debes trabajarla con profundidad.

La aplicación no debe fallar y debe reaccionar antes posibles fallos asegurando la consistencia y calidad de esta.

- o **Enlace en el readme al vídeo en YouTube** donde se explique las partes más relevantes de la práctica y **se muestre su ejecución y como mínimo algunos crud funcionando con sus salidas y las consultas pedidas**. La duración del vídeo debe ser de unos 20 minutos. **La no entrega de este vídeo y donde se vea su ejecución anulará el resultado de la práctica.**
- o Repositorio oficial de la entrega Enlace de entrega: https://classroom.github.com/a/hBV_eHQw. La subirán los dos miembros del equipo, si no está en este repositorio se invalidará la práctica no pudiéndose entregar por otros medios.
- o Se valorará la riqueza de técnicas y recursos utilizados.

NOTA:

La práctica no es obligatoria, pero no realizarla implica que los Resultados de Aprendizaje no se podrán evaluar a través de ella, lo que implica que el porcentaje de calificación asociado a este instrumento quedará calificado con un NO APTO y con ello el Resultado de Aprendizaje estará calificado como NO APTO y por lo tanto deberán ser evaluados por otros instrumentos como un examen práctico en el periodo de recuperación, de acuerdo con lo establecido en la Programación Didáctica.

Aprobar la práctica no implica que no se haga el examen teórico-práctico asociado al Resultado de Aprendizaje, pues para considerarse superado hay que aprobar los dos instrumentos (práctica y examen) y que la media de ambos sea mayor o igual a 5.

La copia de la práctica o fragmentos de ella implica la evaluación de todos Resultados de Aprendizaje con un 0, no pudiéndose recuperar hasta la evaluación ordinaria.

Fecha de entrega 4 de febrero de 2022 a las 09:00.