Segundo laboratorio Bases de Datos NoSql 2020

El segundo laboratorio consiste en la implementación de un sistema de software sin interfaz gráfica, cuyos servicios son ofrecidos y consultados únicamente a través de servicios rest.

Se debe implementar un sistema del estilo “mini twitter” del cual se esperan un conjunto de funcionalidades que se detallan más adelante, dichas funcionalidades serán ofrecidas a través de servicios Rest. El sistema debe utilizar una base de datos No Relacional de las vistas en el curso.

Requerimientos Funcionales

Operaciones:

1) Crear Usuario - Se pasará el correo electrónico del usuario a crear,
este correo debe ser único en el sistema,

2) Crear Comentario - Se pasará el correo electrónico de un usuario del sistema, y el texto de los comentarios (menor a 256 caracteres).El sistema le asignará un identificador de forma automática al comentario creado.

3) Listar Comentarios de Usuario - Se pasará el correo electrónico de un usuario del sistema, y se retorna una lista con el identificador y contenido de todos los comentarios realizados por el usuario, junto con su identificador.

4) Agregar emoción al comentario - Se pasará el identificador de un comentario existente en el sistema, y un calificador de emoción hacia el comentario que puede ser: Me gusta, No me gusta. Se debe verificar que el usuario exista en el sistema.

5) Leer Comentario - Se pasará el identificador de un comentario existente en el sistema, y se retorna el identificador del comentario, el texto que lo contiene, identificador de usuario de quién lo creó, y cantidad de me gustas y no me gusta del comentario.

Requerimientos No Funcionales

- Todo el intercambio de información será realizado mediante JSON a través de servicios rest.
- Las bases de datos NoSql que se pueden utilizar son todas las vistas en el curso.
- El código del trabajo y todo lo referente a documentación y configuración debe ser almacenado en git (git público).
- Se espera la entrega de los casos de pruebas en postman
- El lenguaje utilizado para implementar el sistema debe ser: Java, Python, Node, o .Net Core.
- Se debe entregar un documento donde se especifique la Arquitectura del sistema, diagrama de despliegue, formato de intercambio de datos, instalación, configuración, plataformas, lenguajes y bases de datos utilizadas.

Opcional

- Se valorará que el requerimiento para la consulta para la obtención de los mensajes acceda primeramente a un caché que debe ser implementado con una base de datos NoSql clave valor. La estrategia para almacenar y eliminar los datos en la caché será definida y justificada por el grupo.

Recomendaciones

- Se recomienda utilizar como tecnología Spring Boot con el IDE Eclipse STS.

Entrega

- Al momento de la entrega, los grupos deberán enviar un mail a apastorini@gmail.com con un enlace al git con el código final de la tarea antes del 12/11.

- Los grupos presentarán sus trabajos entre los días 12/11 y 19/11 en el horario de clase a través de Zoom, las presentaciones deben tener una duración no mayor a 10 minutos.

Para interiorizarse en el uso de spring-boot, maven, manejo de errores en mensajes, uso de postman para testear los servicios y la publicación de servicios rest utilizando spring boot, se recomienda consultar:

- https://spring.io/quickstart
- https://spring.io/guides/gs/rest-service/ https://spring.io/guides/gs/spring-boot/
- https://spring.io/guides/gs/maven/
- https://www.baeldung.com/postman-testing-collections
- https://www.baeldung.com/rest-with-spring-series
- https://www.baeldung.com/rest-api-error-handling-best-practices
- https://docs.spring.io/spring-data/data-redis/docs/current/reference/html/#reference