**PedidosYA**

Prueba técnica - Backend

línea horizontal

# Introducción

En el contexto de una arquitectura orientada a micro servicios, se desea implementar un servicio que permita gestionar las **puntuaciones** que los usuarios otorgan a **compras** que realizan en distintas **tiendas**.

Los requerimientos funcionales para el servicio son:

* Al registrar una **puntuación**, se debe guardar la fecha, los identificadores de **usuario** y **tienda** y **compra**, un texto conteniendo la opinión del usuario y una puntuación entre 1 y 5.
* Las compras, los usuarios y las tiendas se gestionan en **servicios externos independientes**, y solo se conoce su identificador.
* Solo se puede registrar una puntuación por compra.
* Se deben ofrecer operaciones para:
  + **Registrar una opinión**
  + **Modificar una opinión.**
  + **Eliminar una opinión**: los moderadores pueden eliminar una opinión si el lenguaje no es apropiado. La eliminación debe ser lógica.
  + Obtener una opinión determinada.
  + Obtener la **opinión para una compra** específica
  + Obtener todas las **opiniones de un usuario** en un rango de fechas. Solo opiniones no eliminadas.
  + Obtener todas las **opiniones para una tienda** en un rango de fechas. Solo opiniones no eliminadas.
* En base a las opiniones registradas, se debe calcular en forma periódica la **puntuación de una tienda**, siendo esta el promedio de las valoraciones registradas que no hayan sido eliminadas.
* Cada vez que se registre, modifique o elimine una opinión, otros servicios deben ser notificados del evento para que realicen acciones asociadas. En principio no se conoce la lista de servicios a notificar, y se sabe que la misma puede cambiar con el tiempo.

# Arquitectura

Dado el problema presentado en la sección anterior, y sabiendo que se registran varios millones de puntuaciones en forma mensual se necesita responder las siguientes preguntas:

1. Sabiendo que la **escalabilidad** y la **performance** son los dos atributos de calidad más importantes, proponga una arquitectura para resolver el problema, explicando cómo la misma tiene en cuenta los atributos de calidad mencionados.
2. Explique cómo almacenaría los datos para resolver el problema. Tenga en cuenta que puede seleccionar más de un tipo de almacenamiento para resolver lo mejor posible cada uno de los requerimientos funcionales.
3. Mencione cuál le parece la mejor opción para comunicar los eventos mencionados a otros servicios en las condiciones mencionadas.
4. En base a la arquitectura propuesta en la pregunta anterior, seleccione qué tecnologías utilizaría para implementar la solución, tanto a nivel de programación, almacenamiento y comunicación entre servicios. Como dato adicional se sabe que la empresa favorece las tecnologías libres y de código abierto frente a las propietarias y con costo de licenciamiento.

**Nota:** En caso de asumir aspectos del problema no mencionados en los requerimientos, aclararlos como parte de la respuesta.

# Diseño

Dado que se decide implementar el servicio como un único api rest, describa que endpoints tendría dicha api, mencionando el verbo http, la url, query string y body de los requests, así como el formato las responses.

# Implementación

Implemente en go, grails, java o python una versión simplificada que permita:

* registrar una opinión
* eliminar una opinión
* obtener la opinión para una compra
* obtener las opiniones para una tienda en un rango de fechas

**Importante**: es solo un prototipo, por lo que no es necesario que utilice las tecnologías de almacenamiento mencionadas en la sección anterior. Si es importante la prolijidad a la hora de codificar, el apego a estándares de codificación y la correcta organización del código del prototipo.

Puede utilizar el framework de su preferencia.

# Se Pide

Entregar un zip que contenga:

* Un documento con las respuestas a las preguntas en las secciones de Arquitectura y Diseño
* El código fuente junto con instrucciones para ejecutar la solución de lo pedido en la sección de Implementación.