

¿Cómo garantizaron la integridad de los datos?

*Garantizamos la integridad de los datos por medio de la implementación de checks como unique y not null en la creación de la base de datos. Adicionalmente, implementamos los triggers requeridos para la entrega que realizan checks (por ejemplo, verificar que un carrito no haya sido comprado a la hora de modificar algún detalle) e ingresan información automáticamente (por ejemplo, actualizar los totales del carrito al modificar o insertar detalles).*

Si el número de productos en la tienda aumentara significativamente, ¿qué modificaciones haría para garantizar el rendimiento?

*En cuanto al rendimiento, evitamos colocar lógica compleja dentro de los triggers para evitar que estos afecten. Al aumentar la cantidad de productos, podemos crear indexes en algunas columnas clave como tipo, marca y precio. Esto ayuda a mejorar el rendimiento al buscar columnas específicas.*

¿Qué pruebas pueden realizar para verificar el rendimiento óptimo de la base de datos?

*Se pueden realizar pruebas de velocidad de queries y de inserción, adicionalmente se pueden realizar pruebas para verificar el rendimiento con una cantidad más grande de usuarios concurrentes.*

¿Es su diseño escalable? ¿Por qué? Si la respuesta es no también respondan ¿qué están haciendo con su vida?

*Si, tomamos en cuenta las reglas de normalización y buscamos que el diseño evitara el almacenamiento de datos duplicados. Además, los triggers que implementamos permiten evitar errores de inserción o actualización de datos. Esto, en conjunto con los logs que implementamos permiten escalar el sistema sin tener que realizar cambios adicionales.*

¿Qué mejora pudieran hacer al sistema para mejorar su rendimiento?

Para mejorar el rendimiento del sistema podríamos implementar indexes, como se había mencionado anteriormente. Luego, tendríamos que realizar algunas pruebas de rendimiento para poder estimar el efecto de los triggers sobre el rendimiento. Con estas estimaciones, podríamos identificar algunos triggers potencialmente complejos y buscar alternativas más rápidas.