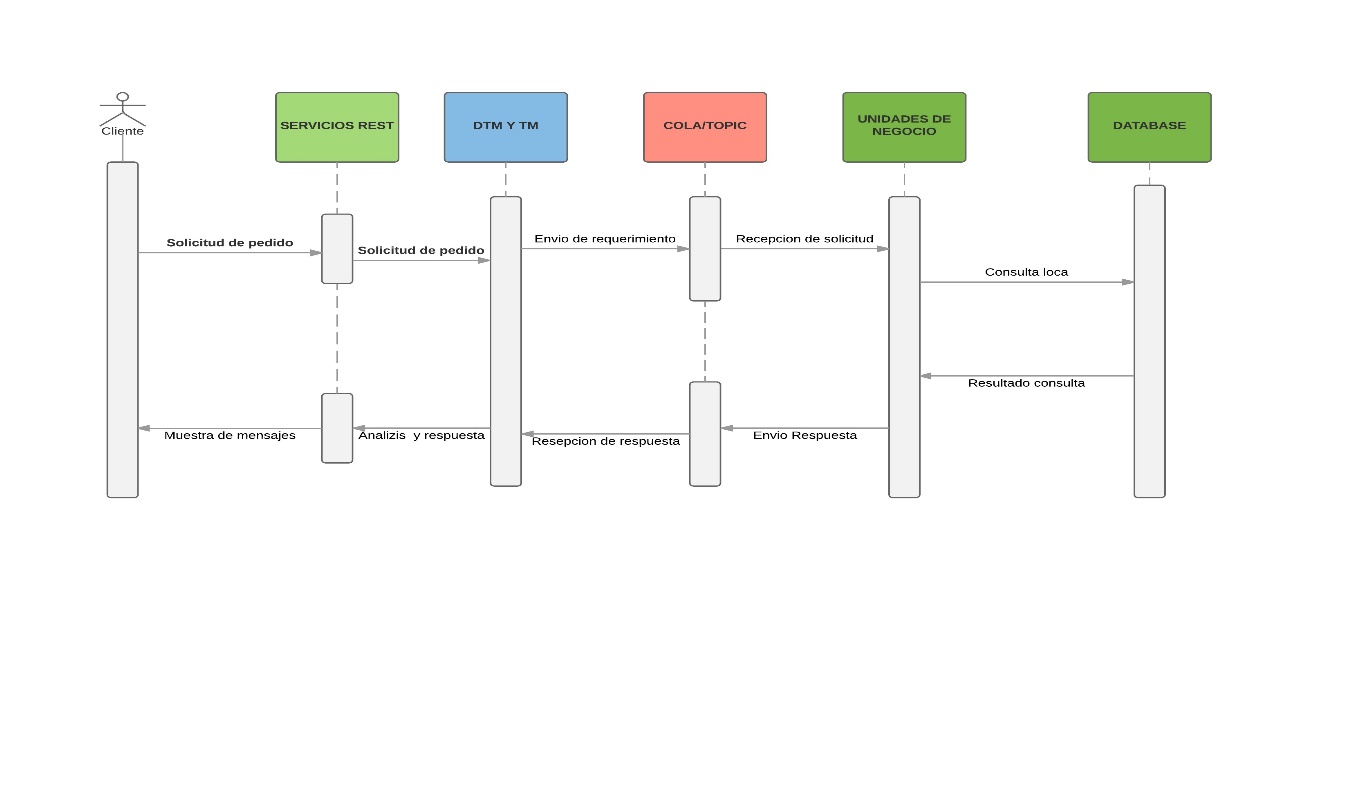
Para esta esta última iteración se revisó la arquitectura de la aplicación y se considero que la arquitectura actual no requiere ninguna revisión ni cambio, en la estructura básica de la aplicación, sin embargo las restricciones para eliminar FK se revaluaron para poder cumplir con el requerimiento de borrado, llevando a aplicar un cascate cuando se borra un restaurante extendiéndose a sus ítems ofrecidos, pero no afectando las ordenes ya echas en historial de órdenes.

A continuación, se muestra la lógica detrás de la aplicación del requerimiento 18



Se muestra la lógica donde un cliente solicita un pedido a una mesa por medio de los servicios rest de la aplicación de la rotonda, posterio del rest pasa al Tm y al Dtm donde se envía la solicitud a la cola o topic de la aplicación pidiendo los ítems solicitados a las rotondas participantes, posteriormente cualquier rotonda o unidad de negocio inscrita en la super rotonda puede atender el requerimiento enviado por alguna de ellas o sus mismos pedidos. Al momento que una rotonda decide atender el mensaje este hace una búsqueda local en su base de datos y envía a la cola de resultados los que se obtuvo en este caso si posee los productos requeridos. Luego la rotonda que solicito el servicio consulta los resultados de la cola de resultados y los analiza dentro de su lógica para determinar si el pedido se puede realizar o si no se disponible de algún producto para realizar el pedido, si la operación fue un éxito este debe registrar el pedido e enviar el mensaje de éxito, si no se cumple se debe realizar rollback de la operación e informar al usuario que no se puede completar el pedido, finalmente se debe informar al usuario del resultado de la operación.

Con el fin de cumplir con este requerimiento en los casos planteados, en ambos casos se debe garantizar desde la lógica que todos los productos que se piden de un menú estén disponibles por alguna de las rotondas participantes, en el primer caso de two phase commit se debe pedir una solicitud de consulta a todos los restaurantes participantes con los procutos requeridos, analizarlos y por medio de la lógica de la aplicación verificar que el pedido se encuentra completo, en caso que este se encuentre completo se debe registrar el gasto en cada rotonda en caso contrario se debe informar del rollback a todas las unidades, en el caso exitoso en la primera fase se debe enviar el mensaje para la preparación del commit, si todas las rotondas registraron exitosamente sus cuentas se procede a informar al usuario y registrar al orden, en caso contrario se debe realizar un rollback e informar al usuario las razones de este.

En el caso de colas se debe primero realizar un pedido de información de los productos solicitados, cada restaurante debe verificar si cuenta con los productos dentro de su rontonda y enviar el resultado a la cola de resultados, posteriormente la rotonda que solicito el servicio debe recuperar estos resultados y realizar un análisis de si el pedido esta completo y puede proceder, en este caso se puede hacer una excepción de la regla e informar al cliente si desea un pedido incompleto o cambiar algo gracias al protocolo de comunicación, en el caso que el pedido este completo o el usuario dese completar al orden se debe enviar el registro de la orden a las rotondas, y espera la respuesta de cada una, si la operación no se completa para alguna de las rotondas que proporciono un producto o el pedido no fue aprobado se debe enviar un mensaje de rollback a todos las unidades de rotoandes y descartar el pedido a la mesa.