

Lógica de los nuevos requerimientos:

Para nosotros es importante que la empresa tenga un producto lo más agrupado posible en una bodega. Por eso nuestro algoritmo funciona de la siguiente manera.

Inserción de una bodega:

Cuando se quiere añadir una bodega, se buscan las bodegas que almacenen el mismo tipo de alimentos. Si alguna de ellas tiene una capacidad disponible menor que la que llega, entonces se empiezan a pasar los alimentos (si hay n de este tipo de alimento se pasan los n) de esta bodega a la otra hasta que quede vacía o hasta que la capacidad disponible de la nueva bodega sea menor que la que menos capacidad tenía. Se siguen mirando las otras bodegas siguiendo el mismo algoritmo.

Eliminar una bodega:

La idea es quitar todos los ítems de esta bodega y enviarlos a las demás bodegas. Los ítems se tratan de enviar completos, es decir, toda la cantidad. Sin embargo, si no caben todos entonces se pasan lo que más se pueda y se continúa tratando de llenar otra bodega. Siempre se aumenta la cantidad de ítems en la bodega que más capacidad tiene al utilizar una lista de prioridad. En dado caso que la bodega no pueda ser eliminada porque los ítems no caben en las otras bodegas de su mismo tipo, entonces se cancela la operación y se hace *rollback*.

Las dos operaciones se hacen de manera transaccional al empezar la transacción con un *conexión.setAutocommit(false)* y haciendo *commit* al final de la transacción. Esto se comprobó al ejecutar los métodos en *debug* y verificando que los cambios sólo se vieran al final de la transacción en una sesión abierta en *sqlDeveloper*.