

## Logica Requerimientos

RF11. En este requerimiento se usa el nivel de aislamiento read committed para garantizar transaccionalidad y ACID, ya que mediante esto es posible asegurarse de que cuando se inserte la equivalencia entre ingredientes, ambos ingredientes existen en la base de datos. En este requerimiento se usa un JSON para especificar los ingredientes que se desean relacionar en la tabla EQUIVALENTES. Además, se usa rollback y commit para garantizar persistencia correcta.

RF12. Para este requerimiento se debe usar un json para enviar los platos que van a ser equivalentes, así se podrá verificar si estos productos pertenecen al mismo restaurante y si son de la misma categoría, en caso de que cumplan con eso, se insertaran en la tabla EQUIVALENCIAS\_PRODUCTOS los id de cada uno de los productos y el restaurante al que pertenecen. Para garantizar ACID, se efectúa commit solo al final de la transacción y en caso de error, se realiza un rollback. De este modo, se garantiza que ocurra todo o no ocurra nada, pero que en ningún caso se efectúe sólo parte de la transacción. Finalmente, también se usa el nivel de aislamiento reado committed para que al insertar los productos equivalentes ambos efectivamente existan en la base de datos.

RF13. En este requerimiento se usa el nivel de aislamiento read committed para garantizar transaccionalidad y ACID, ya que mediante esto es posible asegurarse de que no se venden platos mientras se está abasteciendo cada producto del restaurante, para evitar inconsistencias. En este requerimiento se usa un PathParam para especificar el nombre del restaurante al cual se le desea surtir productos. Además, se usa rollback y commit para garantizar persistencia correcta.

RF14. Para efectuar este requerimiento se debe indicar el menú por agregar y la equivalencia que se desea, todo desde la URL. Así, una vez verificado que el producto pertenezca al menú y que los dos productos sean equivalentes, se inserta el pedido en la tabla PEDIDO con el atributo cambio indicando el cambio de productos a realizar. Para garantizar la transaccionalidad y ACID, se usa nuevamente el nivel de aislamiento read committed, de forma que sea posible saber que la equivalencia no va ser eliminada ni modificada mientras se está insertando el pedido, además también se garantiza que no se agregue otro pedido simultáneamente con un id igual al de este pedido. Por otro lado, se usa commit (al final de todo el proceso) y rollback en caso de que ocurra alguna excepción para garantizar que no ocurran solo partes de la transacción en ningún caso.

RF15. En este requerimiento se usa el nivel de aislamiento read committed para garantizar transaccionalidad y ACID, ya que mediante esto es posible asegurarse que no se agregue un pedido simultáneamente el cual posea el mismo id del pedido que se desea agregar. En este caso se usa un JSON con toda la información del pedido además de un PathParam que especifica los productos que se desean registrar a nombre del usuario que se especifica en el JSON de pedido. Además, se usa rollback y commit para garantizar persistencia correcta.

RF16. Para poder realizar este requerimiento se necesita enviar el JSON del pedido que se entregó, de modo que lo único que se verifica es que el número del pedido exista para posteriormente, modificar la cantidad de vendidos y disponibles en los menús y platos que poseía. Para garantizar transaccionalidad y ACID, se usa commit al final de toda la transacción y rollback en caso de excepciones para obtener la persistencia correcta, y también se usa el nivel de aislamiento read

committed para poder asegurar que el pedido no sea modificado ni cancelado, ni que los productos o menús por modificar sean eliminados.

RFC17. En este requerimiento se usa el nivel de aislamiento serializable para garantizar transaccionalidad y ACID, ya que mediante esto no es posible que se agreguen nuevos productos al pedido que se desea cancelar, posteriormente a haber descartado los producto\_pedido que existían al inicio de la transacción, para poder borrar el pedido sin excepciones de tipo SQL. Además, se usa rollback y commit para garantizar persistencia correcta.

RFC7. Se usa un nivel de aislamiento READ\_ONLY, ya que aca se requiere que todos los datos sean consistentes pero no es necesario editar nada. ACID, en esta situación, está totalmente garantizado ya que no se edita la base de datos.

RFC8. Como este requerimiento es solamente de consulta y por tanto no modifica la base de datos, basta con usar el nivel de aislamiento read only ya que con este se garantiza la consistencia de los datos. Así, como no se edita ninguna tabla, la transaccionalidad y ACID están asegurados.