

1. (2%) Análisis

Entre los cambios realizados tuvimos que añadir el atributo de fechaPublicacion tipo timestamp a la clase oferta con el objetivo de conocer cuáles eran las ofertas de alojamiento que no tienen mucha demanda.

Añadimos la clase reserva colectiva para cumplir con el objetivo del RF7 y le pusimos los atributos definidos en el enunciado. También añadimos los métodos que hacen alusión a los RF de la segunda iteración en la clase principal ALOHANDES

Igualmente, en la clase reserva añadimos el atributo idReserva colectiva el cual es una llave foránea a la clase reserva colectiva y puede ser null, esto lo hacemos con el propósito de identificar aquellas reservas individuales que hagan parte de una reserva colectiva.

2. (18 %) Diseño de la aplicación

a. (8 %) A partir del diseño existente, analice el impacto que representa la introducción de los nuevos requerimientos y restricciones a nivel del modelo conceptual.

El requerimiento registrar reserva colectiva genera un impacto en la transaccionalidad, debido a que en el momento que en una máquina se realiza el procesamiento del registro de una reserva colectiva, es posible que en otra máquina se registre una reserva que en principio se suponía iba a hacer parte de la reserva colectiva. Esto puede generar lecturas sucias, lecturas inconsistentes, phantoms, pérdida de datos. También el hecho de cancelar una reserva colectiva puede generar este tipo de errores, esto puede ser en el caso de que en una máquina se solicite obtener las reservas y justo en ese instante se realiza una cancelación de la reserva colectiva. Esto hace que el que solicite obtener las reservas realice una lectura fantasma.

El cambio más relevante que realizamos es que en esta ocasión en lugar de solicitar todas las ofertas existentes, verificamos antes si su estado es disponible ya que si se encuentra en estado de deshabilitado no se debe mostrar debido a los requisitos del RF9.

b. Valide que su modelo se encuentra en BCNF y que no presenta anomalías de inserción, borrado o actualización con respecto a las reglas de negocio.

Done

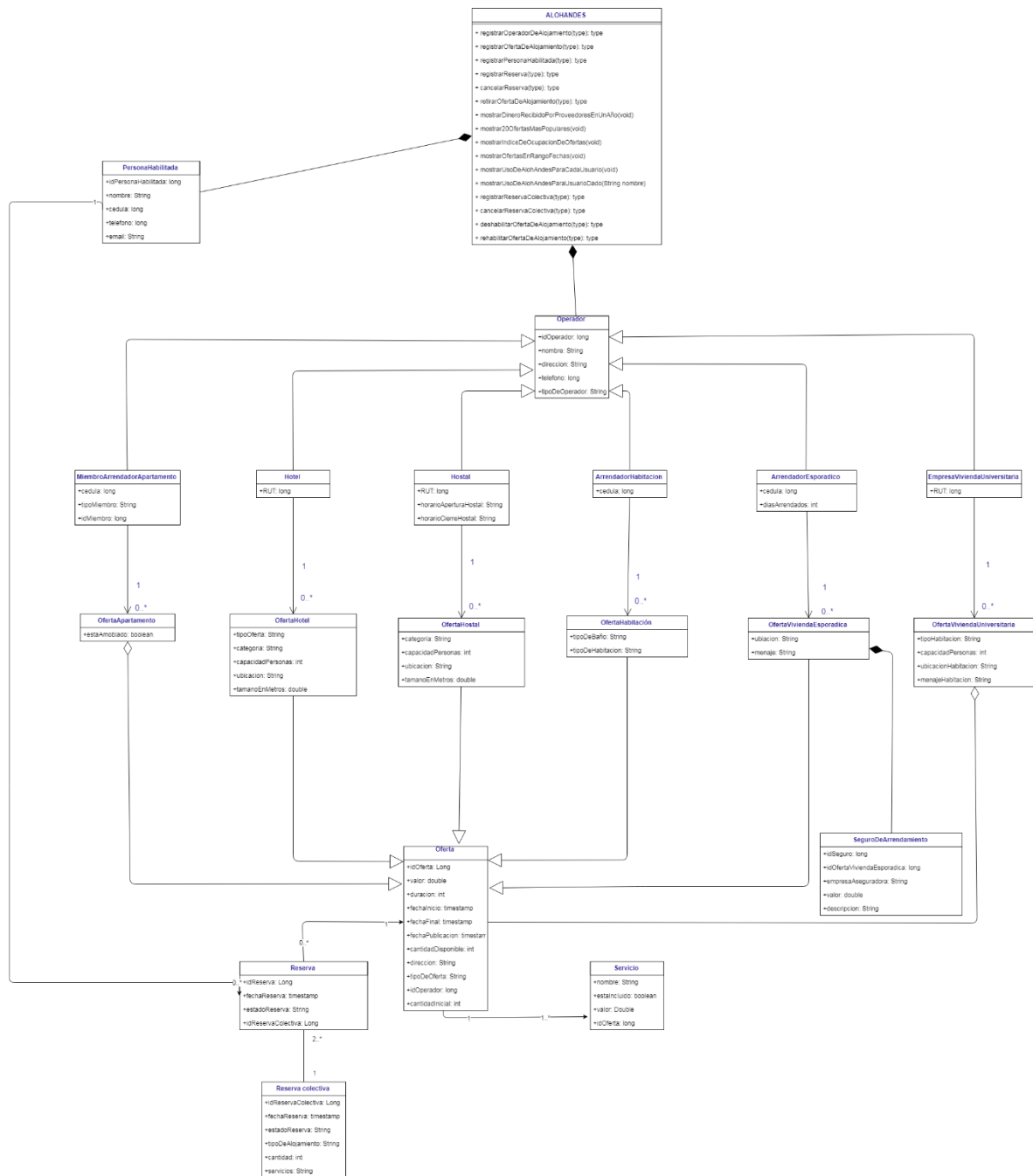
c. (10 %) Documente la lógica de los nuevos requerimientos a desarrollar, descritos en la sección de caso de estudio de este documento. En este punto se requiere definir los mecanismos que utiliza para garantizar las propiedades ACID del requerimiento

Para registrar la reserva colectiva primero que todo a través de la interface se ingresan los parámetros especificados en el enunciado: tipo de alojamiento deseado, cantidad deseada... Después, el transaction manager recibe esta información y en un método después de establecer la conexión delega a un DAO la tarea de buscar en la base de datos si existen ofertas que sean del tipo de alojamiento especificado por parámetro e igualmente verifica si la cantidad de ofertas existentes satisface la necesidad del cliente que realiza la reserva colectiva. Si no se hallan ofertas con los parámetros requeridos se informa que no existen reservas que satisfagan la condición. Si se hallan ofertas que satisfagan la condición, el sistema procederá a realizar el registro de la reserva una por una haciendo uso del RF4 de la iteración anterior.

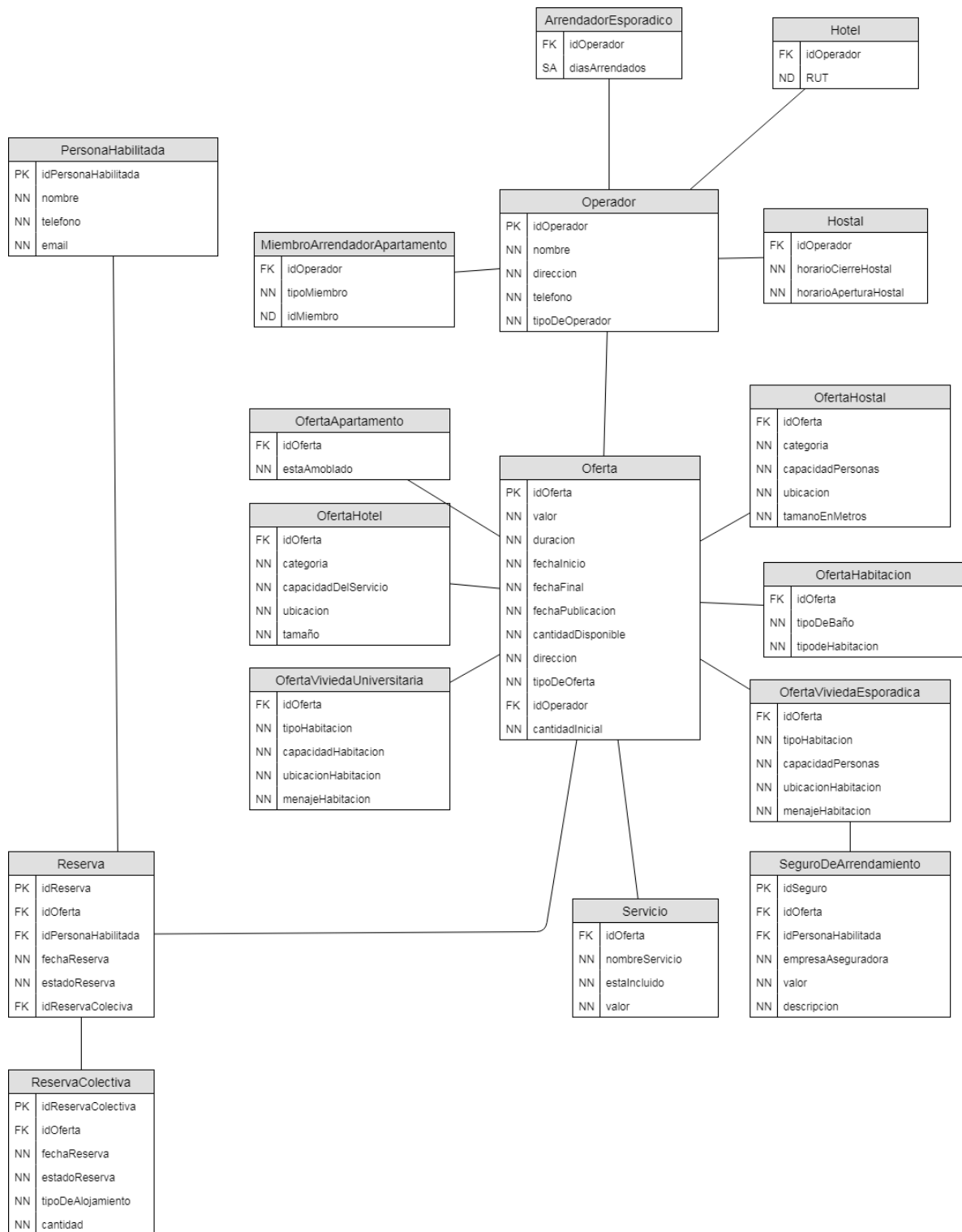
En el caso de cancelar la reserva colectiva, lo que se realiza es una cancelación una por una a cada reserva individual anteriormente generada haciendo uso del RF5 de la iteración anterior, inmediatamente después de esto se generan las penalizaciones que corresponden a cada una de las reservas y se informa al cancelador.

En el caso de deshabilitar y re-habilitar la oferta de alojamiento lo que se hace es uso de un atributo llamado estado en el que se describe si está habilitada o no la oferta. Al momento de deshabilitar una oferta, si esta tiene una reserva asociada se cancela usando el RF5 y no se genera penalización, de lo contrario si se va a rehabilitar se hace uso del RF4.

Estas problemáticas se pueden resolver de diferentes maneras, la primera opción es serializar, esto consiste en que la base de datos garantice que las operaciones se ejecuten en una secuencia, sin embargo, este método no resulta a nuestro parecer el mejor a implementar. La segunda opción tiene dos subdivisiones, estas consisten en control pesimista y control optimista. En el control optimista se implementan estampillas con la finalidad de saber quién llegó primero y esperar que no ocurra ninguna problemática y si esta ocurre se hará uso de las estampillas. No obstante, creemos que la mejor solución a implementar es el uso de candados, más específicamente candados exclusivos, dado a que la concurrencia no es parte fundamental de esta iteración ya que en el peor de los casos se hará uso de dos máquinas a la vez un candado exclusivo impide la ocurrencia de este tipo de problemáticas. Evidentemente, si en algún momento se llega a hacer uso de más de dos máquinas, será necesario pensar en implementar otro tipo de candado que favorezca un poco más la concurrencia.



Modelo conceptual (Figura 1)



Modelo relacional (Figura 2)