

## Caso de estudio: AlohAndes

### Listado de Requerimientos

<b>Nombre</b>	RF1. REGISTRAR LOS OPERADORES DE ALOJAMIENTO PARA ALOHANDES
<b>Resumen</b>	Este requerimiento consiste crear una tupla en la relación de operadores con los atributos definidos por las reglas del negocio. Es básicamente guardar en la base de datos toda la información relativa a los operadores como su identificación, nombre, dirección, tipo y demás. Aplica para el usuario administrador.
<b>Entradas</b>	
idOperador. Que podrá ser asignado por el sistema.	
nombreOperador.	
tipoOperador.	
direccionOperador.	
numeroOperador.	
<b>Resultados</b>	
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.	
<b>RNF asociados</b>	
<b>Transaccionalidad:</b>  La necesidad de un proceso transaccional depende de cómo se implemente el registro de los operadores de alojamiento en la base de datos de AlohAndes.  Si el registro de los operadores de alojamiento se realiza como una operación única que se ejecuta completamente o se deshace en caso de errores, y se cumple con todas las propiedades de una transacción (atomicidad, consistencia, aislamiento y durabilidad), entonces se puede considerar que es una operación transaccional.  Por ejemplo, supongamos que el registro de los operadores de alojamiento en la base de datos de AlohAndes se realiza mediante una transacción que crea una nueva entrada en la tabla de operadores de alojamiento y actualiza la tabla de hoteles para asociar el nuevo operador con el hotel correspondiente. Si esta transacción se ejecuta en su totalidad o se deshace en caso de error, y se cumple con las cuatro propiedades de una transacción, entonces se puede considerar que el registro de los operadores de alojamiento es transaccional.  En resumen, si el registro de los operadores de alojamiento se realiza como una operación transaccional, entonces se puede garantizar que la base de datos estará en un estado consistente y que se evitarán errores y problemas de integridad de datos.	
<b>Persistencia:</b>  Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.	

<b>Nombre</b>	RF2. REGISTRAR OFERTAS DE ALOJAMIENTOS PARA ALOHANDES.
<b>Resumen</b>	Este requerimiento consiste crear una tupla en la relación de propuestas con los atributos definidos por las reglas del negocio. Es básicamente guardar en la base de datos toda la información relativa a las ofertas de los operadores como su identificación, nombre, dirección, tipo y demás. Aplica para el usuario administrador.

<b>Entradas</b>	
idOperador.	Que podrá ser asignado por el sistema.
fechaOferta.	
fechaMaxidaDeRetiro.	Que será asignada por el sistema.
AlojamientoOfertado.	
<b>Resultados</b>	
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.	
<b>RNF asociados</b>	
<p><b>Transaccionalidad:</b></p> <p>El registro de una oferta de alojamiento puede ser transaccional si se cumple con las propiedades de una transacción en una base de datos.</p> <p>Para que el registro de una oferta de alojamiento sea transaccional, se debe garantizar lo siguiente:</p> <p><b>Atomicidad:</b> la operación de registro de la oferta de alojamiento debe ser atómica, lo que significa que debe ejecutarse en su totalidad o no ejecutarse en absoluto. Si la operación no se puede completar, se deben deshacer todas las acciones realizadas hasta ese momento.</p> <p><b>Consistencia:</b> la operación de registro de la oferta de alojamiento debe dejar la base de datos en un estado consistente. Esto significa que la operación debe cumplir con todas las restricciones de integridad referencial y otras reglas de negocio definidas para la base de datos.</p> <p><b>Aislamiento:</b> la operación de registro de la oferta de alojamiento debe ser aislada de otras operaciones que se están ejecutando simultáneamente en la base de datos. Esto significa que los cambios realizados por una transacción no deben ser visibles para otras transacciones hasta que se completen.</p> <p><b>Durabilidad:</b> la operación de registro de la oferta de alojamiento debe ser duradera, lo que significa que una vez que se ha completado, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p> <p>Si se implementa la operación de registro de la oferta de alojamiento como una transacción que cumple con estas cuatro propiedades, entonces se puede considerar que es transaccional.</p> <p>Por lo tanto, el registro de una oferta de alojamiento puede ser transaccional si se cumple con estas propiedades. De esta manera, se garantiza que la base de datos esté en un estado consistente y que se eviten errores y problemas de integridad de datos.</p>	
<p><b>Persistencia:</b></p> <p>Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p>	
<p><b>Concurrencia:</b></p> <p>Muchas ofertas de alojamiento se pueden registrar de manera simultanea por varios operadores, por ende, el sistema debe asegurar la calidad y eficacia del sistema cuando varios operadores usen la plataforma.</p>	

<b>Nombre</b>	RF3. REGISTRAR LAS PERSONAS HABILITADAS PARA UTILIZAR LOS SERVICIOS.
<b>Resumen</b>	Este requerimiento consiste crear una tupla en la relación de propuestas con los atributos definidos por las reglas del negocio. Es básicamente guardar en la base de datos toda la información relativa a las personas habilitadas para utilizar AlohaAndes, es decir, residentes. No es dable el registro de todo el estudiantado, profesorado y planta de personal de Uniandes debido que se estaría replicando una base de datos ya existente,

	por eso, es únicamente necesario registrar a quien reserven el servicio y cuenten con alguna clase de vinculo. Aplica para el usuario administrador.
<b>Entradas</b>	
idResidente.	Que podrá ser asignado por el sistema.
nombreResidente.	
vinculoUniandes.	
numAlojamientos.	Es básicamente el numero de veces que ha usado el servicio.
ultimaReserva.	
numReservas.	
<b>Resultados</b>	
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.	
<b>RNF asociados</b>	
<p><b>Transaccionalidad:</b></p> <p>El registro de personas habilitadas para utilizar un servicio puede ser transaccional si se cumple con las propiedades de una transacción en una base de datos.</p> <p>Para que el registro de personas habilitadas sea transaccional, se debe garantizar lo siguiente:</p> <p><b>Atomicidad:</b> la operación de registro de las personas habilitadas debe ser atómica, lo que significa que debe ejecutarse en su totalidad o no ejecutarse en absoluto. Si la operación no se puede completar, se deben deshacer todas las acciones realizadas hasta ese momento.</p> <p><b>Consistencia:</b> la operación de registro de las personas habilitadas debe dejar la base de datos en un estado consistente. Esto significa que la operación debe cumplir con todas las restricciones de integridad referencial y otras reglas de negocio definidas para la base de datos.</p> <p><b>Aislamiento:</b> la operación de registro de las personas habilitadas debe ser aislada de otras operaciones que se están ejecutando simultáneamente en la base de datos. Esto significa que los cambios realizados por una transacción no deben ser visibles para otras transacciones hasta que se completen.</p> <p><b>Durabilidad:</b> la operación de registro de las personas habilitadas debe ser duradera, lo que significa que una vez que se ha completado, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p> <p>Si se implementa el registro de personas habilitadas como una transacción que cumple con estas cuatro propiedades, entonces se puede considerar que es transaccional.</p> <p>Por lo tanto, el registro de personas habilitadas para utilizar un servicio puede ser transaccional si se cumple con estas propiedades. De esta manera, se garantiza que la base de datos esté en un estado consistente y que se eviten errores y problemas de integridad de datos.</p>	
<p><b>Persistencia:</b></p> <p>Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p>	

<b>Nombre</b>	RF4. REGISTRAR UNA RESERVA
<b>Resumen</b>	Este requerimiento consiste crear una tupla en la relación de operadores con los atributos definidos por las reglas del negocio. Es básicamente guardar en la base de datos toda

	la información relativa a una reserva hecha por algún usuario. Aplica para el usuario Residente.
<b>Entradas</b>	
idReserva. Asignada por el sistema.	
idResidente.	
idOperador.	
IdAlojamiento.	
fechaReserva.	
tiempoRerserva.	
<b>Resultados</b>	
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.	
<b>RNF asociados</b>	
<p>El registro de una reserva puede ser una operación transaccional si se cumple con las propiedades de una transacción en una base de datos.</p> <p>Para que el registro de una reserva sea transaccional, se debe garantizar lo siguiente:</p> <p>Atomicidad: la operación de registro de la reserva debe ser atómica, lo que significa que debe ejecutarse en su totalidad o no ejecutarse en absoluto. Si la operación no se puede completar, se deben deshacer todas las acciones realizadas hasta ese momento.</p> <p>Consistencia: la operación de registro de la reserva debe dejar la base de datos en un estado consistente. Esto significa que la operación debe cumplir con todas las restricciones de integridad referencial y otras reglas de negocio definidas para la base de datos.</p> <p>Aislamiento: la operación de registro de la reserva debe ser aislada de otras operaciones que se están ejecutando simultáneamente en la base de datos. Esto significa que los cambios realizados por una transacción no deben ser visibles para otras transacciones hasta que se completen.</p> <p>Durabilidad: la operación de registro de la reserva debe ser duradera, lo que significa que una vez que se ha completado, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p> <p>Si se implementa el registro de la reserva como una transacción que cumple con estas cuatro propiedades, entonces se puede considerar que es transaccional.</p> <p>Por lo tanto, el registro de una reserva puede ser una operación transaccional si se cumple con estas propiedades. De esta manera, se garantiza que la base de datos esté en un estado consistente y que se eviten errores y problemas de integridad de datos.</p>	
<p><b>Persistencia:</b></p> <p>Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p>	
<p><b>Concurrencia:</b></p> <p>Muchas reservas de alojamiento se pueden registrar de manera simultanea por varios residentes, por ende, el sistema debe asegurar la calidad y eficacia del sistema cuando varios operadores usen la plataforma.</p>	
<b>Nombre</b>	RF5. CANCELAR UNA RESERVA

<b>Resumen</b>	Este requerimiento consiste en retirar una tupla asociada a la relación de reservas que según la llave primaria que contenga el id de quien cancela. Aplica para el usuario Residente.
<b>Entradas</b>	
idResidente	
idReserva.	
fechaReserva	
<b>Resultados</b>	
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.	
<b>RNF asociados</b>	
<p><b>Transaccionalidad:</b></p> <p>La cancelación de una reserva en un hotel puede ser considerada como una operación transaccional siempre y cuando se cumplan las cuatro propiedades de las transacciones en una base de datos: atomicidad, consistencia, aislamiento y durabilidad.</p> <p><b>Atomicidad:</b> La cancelación de la reserva debe ser una operación atómica, lo que significa que debe ser realizada en su totalidad o no ser realizada en absoluto. Si la operación de cancelación no puede ser completada por algún motivo, se deben deshacer todas las acciones realizadas hasta ese punto.</p> <p><b>Consistencia:</b> La cancelación de la reserva debe dejar la base de datos en un estado consistente, lo que significa que todas las restricciones de integridad referencial y otras reglas de negocio deben ser cumplidas.</p> <p><b>Aislamiento:</b> La cancelación de la reserva debe ser aislada de otras operaciones que se estén ejecutando simultáneamente en la base de datos. Esto significa que los cambios realizados por una transacción no deben ser visibles para otras transacciones hasta que se completen.</p> <p><b>Durabilidad:</b> La cancelación de la reserva debe ser duradera, lo que significa que una vez que se haya completado, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p> <p>Si la cancelación de la reserva cumple con estas cuatro propiedades, entonces se puede considerar que es una operación transaccional.</p> <p>En general, es importante que las operaciones de una aplicación de reservas de hoteles, como la cancelación de reservas, sean transaccionales para garantizar la integridad y consistencia de los datos en la base de datos. Esto ayuda a evitar problemas de duplicación de reservas o inconsistencias en los datos, y a asegurar que los usuarios siempre tengan acceso a información precisa y actualizada sobre sus reservas.</p>	
<p><b>Persistencia:</b></p> <p>Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p>	
<p><b>Concurrencia:</b></p> <p>Muchas reservas de alojamiento se pueden cancelar de manera simultánea por varios residentes, por ende, el sistema debe asegurar la calidad y eficacia del sistema cuando varios operadores usen la plataforma.</p>	

<b>Nombre</b>	RF6. RETIRAR UNA OFERTA DE ALOJAMIENTO.
<b>Resumen</b>	Este requerimiento consiste en retirar una tupla asociada a la relación de ofertas que según la llave primaria que contenga el id de quien cancela. Aplica para el usuario Operador.
<b>Entradas</b>	
idOferta	

idOperador
AlojamientoOferta.
<b>Resultados</b>
Per se, no es un procedimiento que retorne algún tipo de dato mas allá del registro de la información en cuestión.
<b>RNF asociados</b>
<p><b>Transaccionalidad:</b></p> <p>La retirada de una oferta de alojamiento puede considerarse como una operación transaccional siempre y cuando se cumplan las cuatro propiedades de las transacciones en una base de datos: atomicidad, consistencia, aislamiento y durabilidad.</p> <p><b>Atomicidad:</b> La retirada de la oferta de alojamiento debe ser una operación atómica, lo que significa que debe ser realizada en su totalidad o no ser realizada en absoluto. Si la operación de retirada no puede ser completada por algún motivo, se deben deshacer todas las acciones realizadas hasta ese punto.</p> <p><b>Consistencia:</b> La retirada de la oferta de alojamiento debe dejar la base de datos en un estado consistente, lo que significa que todas las restricciones de integridad referencial y otras reglas de negocio deben ser cumplidas.</p> <p><b>Aislamiento:</b> La retirada de la oferta de alojamiento debe ser aislada de otras operaciones que se estén ejecutando simultáneamente en la base de datos. Esto significa que los cambios realizados por una transacción no deben ser visibles para otras transacciones hasta que se completen.</p> <p><b>Durabilidad:</b> La retirada de la oferta de alojamiento debe ser duradera, lo que significa que una vez que se haya completado, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p> <p>Si la retirada de la oferta de alojamiento cumple con estas cuatro propiedades, entonces se puede considerar que es una operación transaccional.</p> <p>En general, es importante que las operaciones en una aplicación de reservas de alojamiento, como la retirada de una oferta, sean transaccionales para garantizar la integridad y consistencia de los datos en la base de datos. Esto ayuda a evitar problemas de duplicación de ofertas o inconsistencias en los datos, y a asegurar que los usuarios siempre tengan acceso a información precisa y actualizada sobre las ofertas de alojamiento disponibles.</p>
<p><b>Persistencia:</b></p> <p>Si el requerimiento implementa la transaccionalidad implica que debe ser persistente, lo que significa que una vez que se ha completado la transacción, los cambios realizados en la base de datos deben ser permanentes y sobrevivir a cualquier falla del sistema.</p>
<p><b>Concurrencia:</b></p> <p>Muchas ofertas de alojamiento se pueden retirar de manera simultánea por varios operadores, por ende, el sistema debe asegurar la calidad y eficacia del sistema cuando varios operadores usen la plataforma.</p>

<b>Nombre</b>	RCF1. MOSTRAR EL DINERO RECIBIDO POR CADA PROVEEDOR DE ALOJAMIENTO DURANTE EL AÑO ACTUAL Y EL AÑO CORRIDO.
---------------	--

<b>Resumen</b>	Este requerimiento consiste en calcular el dinero total recibido por un proveedor por concepto de canones de arrendamiento, alquileres etc. Es una operación que deberá ser calculada por el sistema pues no hace parte de las operaciones básicas de registro. Aplica para administradores u operadores, según las reglas de negocio.
<b>Entradas</b>	
idOperador.	
<b>Resultados</b>	
Un entero o long que representa la cantidad recaudada por un proveedor en específico durante el año actual y el año corrido.	
<b>RNF asociados</b>	
Privacidad:	
La información retornada deberá ser tratada únicamente por quien tenga permiso sobre ella como administradores y los operadores en si mismo, mas no para otros roles como usuarios residentes.	
Concurrencia:	
Una consulta debe respetar el requisito no funcional de concurrencia en una base de datos porque, si no se respeta, puede causar problemas como la lectura de datos inconsistentes, la sobrecarga del servidor o incluso la pérdida de datos.	
La concurrencia se refiere a la capacidad de una base de datos para permitir que varios usuarios o procesos accedan a la misma información al mismo tiempo. Cuando varios usuarios o procesos intentan acceder a los mismos datos simultáneamente, puede haber conflictos que afecten la integridad y consistencia de los datos.	
Por lo tanto, es importante que cualquier consulta en una base de datos respete el requisito no funcional de concurrencia para garantizar que se acceda a los datos de manera segura y sin conflictos. Esto se logra mediante el uso de bloqueos y transacciones que aseguran que solo un usuario o proceso tenga acceso a los datos en un momento determinado, y que cualquier cambio realizado en los datos se realice de manera consistente y segura.	

<b>Nombre</b>	RCF2. MOSTRAR LAS 20 OFERTAS MÁS POPULARES.
<b>Resumen</b>	Este requerimiento consiste en ordenar todas las ofertas disponibles a partir de su popularidad, para luego retornar las veinte primeras, si las hay. Es una operación calculada por el sistema. Aplica para todos los usuarios.
<b>Entradas</b>	
Explícitamente ninguno, pues se quiere es acceder a toda una relación ordenada a partir de un criterio en específico.	
<b>Resultados</b>	
Un arreglo con los nombres de las veinte, si las hay, ofertas mas populares.	
<b>RNF asociados</b>	
Concurrencia:	
Una consulta debe respetar el requisito no funcional de concurrencia en una base de datos porque, si no se respeta, puede causar problemas como la lectura de datos inconsistentes, la sobrecarga del servidor o incluso la pérdida de datos.	
La concurrencia se refiere a la capacidad de una base de datos para permitir que varios usuarios o procesos accedan a la misma información al mismo tiempo. Cuando varios usuarios o procesos intentan acceder a los mismos datos simultáneamente, puede haber conflictos que afecten la integridad y consistencia de los datos.	
Por lo tanto, es importante que cualquier consulta en una base de datos respete el requisito no funcional de concurrencia para garantizar que se acceda a los datos de manera segura y sin conflictos. Esto se logra mediante	

el uso de bloqueos y transacciones que aseguran que solo un usuario o proceso tenga acceso a los datos en un momento determinado, y que cualquier cambio realizado en los datos se realice de manera consistente y segura.

<b>Nombre</b>	RFC3 - MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS OFERTAS DE ALOJAMIENTO REGISTRADAS.
<b>Resumen</b>	Este requerimiento consiste en mostrar a quien interese el número de personas que se encuentran habitando los alojamientos ofertados en promedio. Recordemos que muchas de las opciones alojamiento pueden ser compartidas por ende pueden estar habitadas antes de que un usuario cualquiera quiera reservar. Aplica para administradores y operadores, puesto que por tratarse de un índice no debería de ser importante para las operaciones que ejecute un residente.
<b>Entradas</b>	
Explícitamente ninguno, pues se quiere es acceder al resultado de un calculo a partir de la relación de ofertas.	
<b>Resultados</b>	
Un flotante que representa el índice de ocupación de cada una de las ofertas de alojamiento registradas.	
<b>RNF asociados</b>	
Privacidad: La información retornada deberá ser tratada únicamente por quien tenga permiso sobre ella como administradores y los operadores en si mismo, mas no para otros roles como usuarios residentes.	
Concurrencia: Una consulta debe respetar el requisito no funcional de concurrencia en una base de datos porque, si no se respeta, puede causar problemas como la lectura de datos inconsistentes, la sobrecarga del servidor o incluso la pérdida de datos. La concurrencia se refiere a la capacidad de una base de datos para permitir que varios usuarios o procesos accedan a la misma información al mismo tiempo. Cuando varios usuarios o procesos intentan acceder a los mismos datos simultáneamente, puede haber conflictos que afecten la integridad y consistencia de los datos. Por lo tanto, es importante que cualquier consulta en una base de datos respete el requisito no funcional de concurrencia para garantizar que se acceda a los datos de manera segura y sin conflictos. Esto se logra mediante el uso de bloqueos y transacciones que aseguran que solo un usuario o proceso tenga acceso a los datos en un momento determinado, y que cualquier cambio realizado en los datos se realice de manera consistente y segura.	
<b>Nombre</b>	RFC4 - MOSTRAR LOS ALOJAMIENTOS DISPONIBLES EN UN RANGO DE FECHAS, QUE CUMPLEN CON UN CONJUNTO DE REQUERIMIENTOS DE DOTACIÓN O SERVICIOS. POR EJEMPLO, COCINETA, TV CABLE, INTERNET, SALA.
<b>Resumen</b>	Este requerimiento consiste en filtrar la relación de alojamiento a partir de su numero de reservas, pues queremos saber si esta disponible, y de los servicios que incluye. Aplica para todo tipo de usuario pero especialmente para los residentes.
<b>Entradas</b>	
Fecha de inicio.	
Fecha de fin.	
Conjunto de servicios de interes.	
<b>Resultados</b>	
La relación de alojamientos filtrada por los atributos de interés.	
<b>RNF asociados</b>	



#### Concurrencia:

Una consulta debe respetar el requisito no funcional de concurrencia en una base de datos porque, si no se respeta, puede causar problemas como la lectura de datos inconsistentes, la sobrecarga del servidor o incluso la pérdida de datos.

La concurrencia se refiere a la capacidad de una base de datos para permitir que varios usuarios o procesos accedan a la misma información al mismo tiempo. Cuando varios usuarios o procesos intentan acceder a los mismos datos simultáneamente, puede haber conflictos que afecten la integridad y consistencia de los datos.

Por lo tanto, es importante que cualquier consulta en una base de datos respete el requisito no funcional de concurrencia para garantizar que se acceda a los datos de manera segura y sin conflictos. Esto se logra mediante el uso de bloqueos y transacciones que aseguran que solo un usuario o proceso tenga acceso a los datos en un momento determinado, y que cualquier cambio realizado en los datos se realice de manera consistente y segura.