

# Iteración 1 del proyecto del curso Sistemas Transaccionales - Caso ALOHANDES

Alonso Hernández Tavera, Cristian Bernardo Acuña Silva

Documento de la Iteración 1 – Caso ALOHANDES

Universidad de los Andes, Bogotá, Colombia

{f.hernandezt, cb.acuna}@uniandes.edu.co

Fecha de presentación: Febrero 26 de 2023

## Tabla de contenido

1	Introducción y Contextualización .....	1
2	Identificación de las reglas de negocio .....	2
3	Análisis y modelo conceptual .....	3
3.1	Requerimientos funcionales .....	<b>Error! Bookmark not defined.</b>
3.2	Modelo conceptual construido .....	3
4	Diseño de la base de datos.....	12
4.1	Modelo relacional construido .....	12
4.2	Consideraciones del modelo construido .....	<b>Error! Bookmark not defined.</b>
5	Control de calidad del modelo .....	16
5.1	Diagramas de secuencia .....	16
5.2	Requerimientos No Funcionales .....	18
6	Conclusiones .....	18
7	Bibliografía.....	19

## 1 Introducción y Contextualización

ALOHANDES corresponde a un servicio integrado de alojamiento que, por medio de un sistema digital, busca implementar la idea del proyecto ALOHA con el objetivo de facilitar opciones de alojamiento a la comunidad de la Universidad de los Andes. Para lograr generar una oferta amplia de opciones de alojamiento, la universidad ha establecido convenios con 10 hoteles del sector, 12 hostales cercanos, un operador de residencias universitarias vecino, ha convocado a la comunidad de Fenicia (que es vecina a la universidad) y también a la comunidad universitaria (empleados, profesores y estudiantes) para que puedan ofrecer los inmuebles que tienen bajo su propiedad en cualquiera de las modalidades que ha definido ALOHA (ALOHANDES – Descripción general del caso, 2023).

La estructura base que sigue el sistema ALOHANDES es el siguiente: Un operador, que es toda persona natural o empresa de alojamientos con la que la universidad ha establecido convenios, puede ofertar los inmuebles que tenga bajo su propiedad (y se encuentren en una zona vecina a las instalaciones centrales de la universidad) a partir de su descripción y caracterización. Algunas de las características que debe incluir son: La ubicación de ese inmueble, el área del inmueble, su capacidad y a qué tipo corresponde (vivienda o habitación). Además, también debe establecer la modalidad temporal en la que quiere que su inmueble sea reservado y ocupado (las opciones son: reserva por días, semanas, meses o semestres).

Al definir esos detalles, el operador puede inscribir su inmueble bajo una oferta de alojamiento, la cual se registra en el sistema de ALOHANDES junto a los servicios que ese operador defina pueden ser incluidos con un precio adicional o no. Un cliente registrado en el sistema (y que pertenece a la comunidad Uniandes) puede visualizar las diferentes ofertas realizadas por los operadores por medio del sistema y elegir la que mejor se ajuste a sus requerimientos y posibilidades. Una vez un cliente toma la decisión de aceptar una oferta de alojamiento, se crea un registro de la reserva. Esta reserva tiene en cuenta cualquier promoción a la que el cliente haya accedido, la fecha en la que se realizó la reserva, las fechas inicial y final de la reserva y, en caso de que se haya cancelado, también incluye la fecha de cancelación de esta.

El objetivo de esta primera parte del proyecto es analizar y diseñar el sistema ALOHANDES junto a una base de datos de tipo relacional que permita su funcionamiento correcto, teniendo en cuenta un conjunto de requerimientos funcionales y no funcionales definidos.

## **2 Identificación de las reglas de negocio**

Con el objetivo de identificar las limitaciones y restricciones del sistema que se busca construir, se han definido las siguientes reglas de negocio:

1. Cada oferta de alojamiento que realiza un operador puede estar relacionada con al menos y como máximo un inmueble (habitación, apartamento o casa).
2. Una oferta de alojamiento debe especificar su modalidad temporal, es decir, si su operador desea que el inmueble sea reservado por días, semanas, meses o semestres.
3. Las empresas de alojamiento que corresponden a hoteles y hostales, deben contar con un registro de empresa de servicio de hotelería en la cámara de comercio y en la superintendencia de turismo.
4. En el caso de las personas naturales (vecinos de la comunidad Fenicia o integrantes de la comunidad Uniandes) deben ofrecer un contrato por mínimo un mes, con la excepción de aquellos que ofrecen sus inmuebles de forma esporádica cuando hay eventos de gran magnitud.
5. Los inmuebles que son ofrecidos de forma esporádica (por días), no necesitan de un registro ante cámara de comercio o la superintendencia de turismo. Estos no pueden ser ofrecidos por más de 30 días en total en un mismo año.
6. Las empresas que administran viviendas universitarias deben ofrecer habitaciones por una duración mínima de 1 noche y ofrecer opciones mensuales y semestrales.
7. Es necesario que todos los operadores registrados en el sistema ALOHANDES, tengan una relación de convenio con la Universidad de los Andes, esas posibles relaciones son las descritas en el apartado de introducción y contexto.
8. Todo cliente registrado en el sistema debe tener uno de los siguientes vínculos con la institución: ser estudiantes, egresados, empleados, profesores, padres de estudiantes, profesores invitados o personas registradas en eventos Uniandes.
9. Cada reserva puede ser realizada por solamente un cliente registrado en el sistema ALOHANDES.
10. Un cliente puede realizar máximo una reserva de un inmueble en un mismo día.
11. Por cuestiones de seguridad, en caso que un inmueble tenga capacidad para más de una persona, cada persona que quiera acceder a la oferta de alojamiento debe realizar una reserva individual.
12. En relación a la regla de negocio 7, no es posible reservar un inmueble si su capacidad máxima es excedida luego de esa reserva.

13. El precio o costo de un inmueble, definido por medio de su oferta de alojamiento, corresponde a un dato derivado, pues depende de diversos factores como la inclusión de servicios o promociones. Se asume que todos los precios se van a manejar como datos de tipo entero positivo.

### 3 Análisis y modelo conceptual

A continuación, se puede visualizar el modelo conceptual construido con el objetivo entender el mundo del caso que se está tratando. Adjunto en el lugar de entrega de este documento, se encuentra también el mismo modelo en formato imagen para ser visualizado de mejor forma.

#### 3.1 Requerimientos funcionales

Nombre	RF1. REGISTRAR LOS OPERADORES DE ALOJAMIENTO PARA ALOHANDES
Resumen	Para este requerimiento funcional será necesario poder registrar la información de los operarios de AloHandes, para de este modo poder verificar requisitos o restricciones asociados a los operadores y a su información específica.
<b>Entradas</b>	
Para todos los operadores debemos conocer su relación uniandes(relaciónconUniandes:String)	
Si es una persona natural, debemos conocer si ofrece alojamiento por días (OfreceAlojamientoPorDias:Boolean)	
Si es una empresa, debemos conocer que tipo de empresa es, su hora de apertura y su hora de cierre.(tipoEmpresa:String, horaApertura: Date, horaCierre:Date)	
<b>Resultados</b>	
Como resultado obtenemos el registro de los operadores y toda su información necesaria.	
<b>RNF asociados</b>	
Privacidad, la información de los operadores solamente debe ser visible para sí mismo y para los administradores del sistema	
Persistencia, La información de los operadores debe quedar guardada para consultarla posteriormente	
<b>Comportamiento transaccional</b>	
Este requerimiento es transaccional debido a que cumple con las características ACID. Atomicidad, el registro del operador se debe realizar completamente. Coherencia, la información de los operadores que se está registrando debe ser coherente, por ejemplo no puede haber dos personas con un id igual. Aislamiento, cada registro se debe realizar por separado así se estén haciendo otros registros simultáneamente. Durabilidad, la información de los operarios debe persistir para tener la información disponible en cualquier momento que se necesite.	

Tabla 1. RF1

Nombre	RF2. REGISTRAR PROPUESTAS DE ALOJAMIENTOS PARA ALOHANDES
Resumen	Para este requerimiento funcional será necesario poder registrar la información de las ofertas de los operadores para que los clientes puedan ver las ofertas y decidir si quieren reservar el alojamiento.
<b>Entradas</b>	
Para registrar la oferta debemos contar con la siguiente información del inmueble ofertado: su identificador único, su categoría, su capacidad, su ubicación, su menaje y su tamaño. (categoria: String, capacidad: Integer, - ubicacion: String, menaje: String, tamanoEnSqMt: float)	
También es necesario tener información sobre los servicios ofrecidos por el operador y debemos tener la información del nombre del servicio, descripción del servicio, hora de apertura, hora de cierre, costo adicional. (nombre: String, descripcion: String, horaApertura: Date, horaCierre: Date, costoAdicional: Integer )	
<b>Resultados</b>	
Como resultado obtenemos una oferta de alojamiento asociada a un operador y a un grupo de servicios y adicionalmente cuenta con un id de oferta, un atributo que nos indica si la oferta está activa, un atributo que nos dice si la oferta está disponible, y a la temporalidad de la oferta. (idOferta: Integer, ofertaActiva: Boolean, ofertaDisponible: Boolean, modalidadTemporal: String)	
<b>RNF asociados</b>	
Privacidad, la información de las ofertas solamente debe ser visible para los ofertores, los usuarios con permisos para verlas y para los administradores del sistema	
Persistencia, La información de las ofertas debe quedar guardada para consultarla posteriormente	
<b>Comportamiento transaccional</b>	
Este requerimiento es transaccional debido a que cumple con las características ACID. Atomicidad, el registro de una oferta se debe realizar completamente, no debe quedar una oferta con información incompleta. Coherencia, la información de las ofertas debe ser coherente con la realidad, por ejemplo si un alojamiento está ocupado en determinadas fechas no se puede hacer una oferta de alojamiento en esas fechas. Aislamiento, cada registro de oferta se debe realizar por separado así se estén haciendo otros registros de oferta simultáneamente. Durabilidad, la información de las ofertas debe quedar guardada para posteriormente consultarla, incluso cuando ya se terminó el periodo de alojamiento, tener información de las ofertas históricas puede darnos información muy útil.	

Tabla 2. RF2

Nombre	RF3. REGISTRAR LAS PERSONAS HABILITADAS PARA UTILIZAR LOS SERVICIOS
Resumen	Para este requerimiento funcional será necesario poder registrar la información de las personas relacionadas con uniandes que podrán acceder a los servicios de AloHandes y podrán hacer reservas de alojamiento o podrán ofertar alojamientos.
<b>Entradas</b>	
Para registrar las personas habilitadas debemos conocer el id del cliente, y su rol en la universidad.(idCliente: Integer, vinculoConUniandes: String)	
<b>Resultados</b>	
Como resultado obtenemos una lista de clientes de los cuales sabemos su id y su rol en la universidad, y de este modo podemos ver si están habilitadas para acceder a determinados servicios.	
<b>RNF asociados</b>	
Privacidad, la información de los clientes solo deber estar disponible para sí mismos y para los administradores del sistema	
Persistencia, La información de los clientes debe quedar guardada para consultarla posteriormente	
<b>Comportamiento transaccional</b>	
Este requerimiento es transaccional debido a que cumple con las características ACID. Atomicidad, el registro de un cliente se debe realizar completamente. Coherencia, la información de los clientes debe ser coherente con la realidad, por ejemplo no debe haber dos clientes con el mismo identificador de cliente. Aislamiento, cada registro de cliente se debe realizar por separado así se estén haciendo otros registros de cliente simultáneamente. Durabilidad, la información de los clientes debe quedar guardada para posteriormente consultarla.	

Tabla 3. Rf 3

Nombre	RF4. REGISTRAR UNA RESERVA
Resumen	Para este requerimiento funcional será necesario poder registrar una reserva. Las reservas estarán asociadas a un cliente y a una oferta de alojamiento
<b>Entradas</b>	
Para registrar las reservas debemos tener una oferta previamente establecida y un cliente que quiera tomar esta oferta, cosa que hará a través de una reserva.	
<b>Resultados</b>	
Como resultado obtenemos una reserva que está relacionada con un cliente y con una oferta y que cuenta con la siguiente información: si tienes promociones activas, fecha de inicio de la reserva, fecha de finalización de la reserva, fecha de la reserva, fecha de cancelación. (promocionesActivas: String, - fechaInicialReserva: String, fechaFinalReserva: Date, fechaReserva: Date, fechaCancelacion: Date )	
<b>RNF asociados</b>	
Privacidad, la información de las reservas solo deber estar disponible para los clientes y los	
Concurrencia, ya que es posible que dos clientes estén interesados en la misma oferta e intenten hacer la reserva simultaneamente, se debe manejar estas situaciones para poder garantizar la coherencia de los datos.	
Persistencia, La información de los clientes debe quedar guardada para consultarla	
<b>Comportamiento transaccional</b>	
Este requerimiento es transaccional debido a que cumple con las características ACID. Atomicidad, el registro de una reserva debe realizarse por completo, no pueden quedar reservas sin información incompleta. Coherencia, la información de las reservas debe ser coherente con la realidad, no pueden quedar dos reservas para clientes distintos en la misma fecha. Aislamiento, cada reserva se puede hacer por separado ya que se pueden realizar diferentes reservas simultaneamente. Durabilidad, la información de las reservas debe quedar guardada para posteriormente consultarla.	

Tabla 4. RF4

Nombre	RF5. CANCELAR UNA RESERVA
Resumen	En este requerimiento debemos poder cancelar una reserva existente.
<b>Entradas</b>	
Para poder cancelar una reserva debemos primero tener la reserva que queremos cancelar.	
<b>Resultados</b>	
Como resultado obtenemos la cancelación de la reserva, esto se verá reflejado en que el cliente ya no tendrá una reserva asociada y la oferta también quedará sin reserva asociada	
<b>RNF asociados</b>	
Persistencia, La información de los clientes debe quedar guardada para consultarla	
<b>Comportamiento transaccional</b>	
<p>Este requerimiento es transaccional debido a que cumple con las características ACID.</p> <p>Atomicidad, la cancelación de una reserva debe realizarse por completo, no se puede realizar una cancelación a medias. Coherencia, la información de la cancelación debe verse reflejada en los demás actores relacionados y debe ser coherente con la realidad, por ejemplo no puede haber un cliente asociado a una reserva que ha sido cancelada. Aislamiento, cada cancelación de reserva se puede hacer por separado ya que se pueden realizar diferentes cancelaciones de reservas simultáneamente. Durabilidad, la información de las cancelaciones de reservas debe quedar guardada para posteriormente consultarla.</p>	

Tabla 5. RF 5



Nombre	RF6. RETIRAR UNA OFERTA DE ALOJAMIENTO
Resumen	En este requerimiento debemos poder retirar una oferta de alojamiento existente.
<b>Entradas</b>	
Para poder retirar una oferta de alojamiento debemos primero tener oferta de alojamiento que queremos retirar	
<b>Resultados</b>	
Como resultado obtenemos el retiro de una oferta de alojamiento, esto se verá reflejado en que el operador ya no tendrá una oferta de alojamiento asociada.	
<b>RNF asociados</b>	
Persistencia, La información de los clientes debe quedar guardada para consultarla	
<b>Comportamiento transaccional</b>	
Este requerimiento es transaccional debido a que cumple con las características ACID. Atomicidad, el retiro de una oferta de alojamiento debe realizarse por completo. Coherencia, la información del retiro de una oferta de alojamiento debe verse reflejada en los demás actores relacionados y debe ser coherente con la realidad, por ejemplo no puede haber un operador asociado a una oferta de alojamiento que ha sido retirada. Aislamiento, cada retiro de una oferta de alojamiento se puede hacer por separado ya que se pueden realizar diferentes retiros de oferta de alojamiento simultáneamente. Durabilidad, la información de las retiro de una oferta de alojamiento debe quedar guardada para posteriormente consultarla.	

Tabla 6. RF6

Nombre	RFC1. MOSTRAR EL DINERO RECIBIDO POR CADA PROVEEDOR DE ALOJAMIENTO DURANTE EL AÑO ACTUAL Y EL AÑO CORRIDO
Resumen	Este requerimiento funcional de consulta nos muestra la información de cada proveedor con respecto a cuanto dinero a recibido a lo largo del año actual y el año corrido.
<b>Entradas</b>	
Para este requisito no necesitamos información específica, solo la solicitud de consulta de la información.	
<b>Resultados</b>	
obtenemos la información de cada proveedor con respecto a cuanto dinero a recibido a lo largo del año actual y el año corrido.	
<b>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta consulta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 7. RFC1



Nombre	RFC2. MOSTRAR LAS 20 OFERTAS MÁS POPULARES
Resumen	Este requerimiento funcional de consulta nos muestra las 20 ofertas más populares
<b>Entradas</b>	
Para este requisito no necesitamos información específica, solo la solicitud de consulta de la información.	
<b>Resultados</b>	
obtenemos la información de cuales son las 20 ofertas más populares	
<b>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 8. RFC2

Nombre	RFC3. MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS OFERTAS DE ALOJAMIENTO REGISTRADAS
Resumen	Este requerimiento funcional de consulta nos muestra el índice de ocupación de cada una de las ofertas de alojamiento registradas
<b>Entradas</b>	
Para este requisito no necesitamos información específica, solo la solicitud de consulta de la información.	
<b>Resultados</b>	
obtenemos la información de el índice de ocupación de cada una de las ofertas de alojamiento registradas	
<b>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 9. RFC3

Nombre	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.
Resumen	Este requerimiento funcional de consulta nos muestra 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>Entradas</b>	
Para este requisito necesitamos consultar la información además de especificar el rango de fechas y los servicios que debe tener los alojamientos.	
<b>Resultados</b>	
obtenemos la información de 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>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 10. RFC4

Nombre	RFC5. MOSTRAR EL USO DE ALOHANDES PARA CADA TIPO DE USUARIO DE LA COMUNIDAD
Resumen	Este requerimiento funcional de consulta nos muestra el uso de alohandes para cada tipo de usuario de la comunidad
<b>Entradas</b>	
Para este requisito no necesitamos información específica, solo la solicitud de consulta de la información.	
<b>Resultados</b>	
obtenemos la información de el uso de alohandes para cada tipo de usuario de la comunidad	
<b>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 11. RFC5

Nombre	RFC5. MOSTRAR EL USO DE ALOHANDES PARA UN USUARIO DADO (NÚMERO DE NOCHES O MESES CONTRATADOS, CARACTERÍSTICAS DEL ALOJAMIENTO UTILIZADO, DINERO PAGADO).
Resumen	Este requerimiento funcional de consulta nos muestra el uso de alohandes para un usuario dado (número de noches o meses contratados, características del alojamiento utilizado, dinero pagado).
<b>Entradas</b>	
Para este requisito necesitamos consultar la información además de especificar el usuario específico de uniandes del cual queremos obtener la información	
<b>Resultados</b>	
obtenemos la información de el uso de alohandes para un usuario dado (número de noches o meses contratados, características del alojamiento utilizado, dinero pagado).	
<b>RNF asociados</b>	
Privacidad, solo los administradores del sistema deben tener acceso a la información de esta	
<b>Comportamiento transaccional</b>	
Este requerimiento no es transaccional ya que es un requerimiento de consulta.	

Tabla 12. RFC6

### 3.2 Modelo conceptual construido

A continuación, se puede visualizar el modelo conceptual construido con el objetivo entender el mundo del caso que se está tratando. Adjunto en el lugar de entrega de este documento, se encuentra también el mismo modelo en formato imagen para ser visualizado de mejor forma.

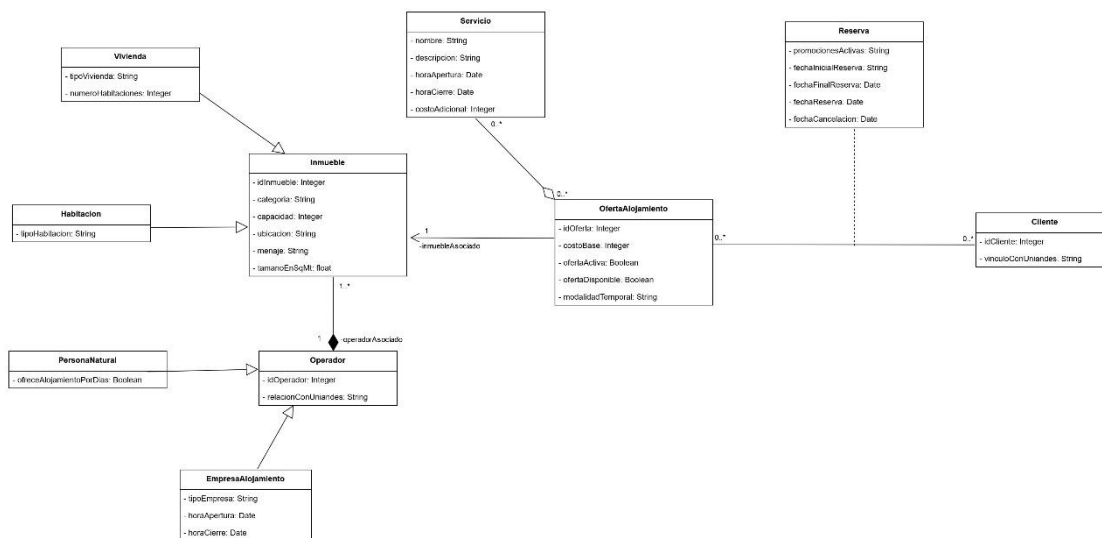


Figura 1. Diagrama conceptual construido para el caso ALOHANDES

### 3.3 Consideraciones y aclaraciones del modelo conceptual

Las consideraciones que se deben tener en cuenta para entender el diagrama conceptual son las siguientes:

1. Ofrecer alojamiento con días para una persona natural se relaciona con ofrecer alojamiento esporádico.
2. Se asume que todos los precios que se van a manejar van a ser de tipo entero positivo.
3. El hecho de que un inmueble se encuentre o no amoblado debe ser representado como un servicio incluido.
4. En ninguna clase se modela el atributo precio o costo total, pues este corresponde a un dato de tipo derivado (depende de otros atributos y variables para ser calculado). Los datos derivados impiden obtener la tercera forma normal (3FN) en el modelo (Huaraca, 2020) y, por tanto, no es deseable modelarlos en las relaciones de forma inicial. En su lugar, se incluye el atributo “costoBase” en la clase “OfertaAlojamiento”, el cual no es un dato derivado.
5. El atributo de “modalidadTemporal” en la clase OfertaAlojamiento se refiere a si el inmueble es ofrecido por días, semanas, meses o semestres.
6. Se supone que un cliente solo puede realizar una reserva por día, pero puede tener más de una reserva activa al mismo tiempo. Cada reserva la puede hacer máximo un cliente.
7. Las clases “Vivienda” y “Habitacion” se toman como clases que heredan de la superclase “Inmueble” debido a que esta última es hasta cierto punto una abstracción de ambas y contiene atributos que comparten. De la misma forma, “PersonaNatural” y “EmpresaAlojamiento” heredan de la superclase “Operador”.
8. La clase “Reserva” corresponde a una clase de asociación, la cual modela los diferentes atributos que caracterizan la relación de un cliente con una oferta de alojamiento.
9. La diferencia entre “ofertaActiva” y “ofertaDisponible” en la clase “OfertaAlojamiento” es la siguiente: Una oferta permanece activa mientras el operador que la publicó no haya decidido retirarla, en cambio, permanece disponible mientras un cliente no la haya reservado.
10. En la clase “Reserva” se modela un atributo llamado “fechaCancelación” con el objetivo de saber si una reserva fue o no cancelada y aplicar la respectiva penalización en caso de que aplique.

## 4 Diseño de la base de datos

### 4.1 Modelo relacional construido

Con el objetivo de estructurar la información del caso en una base de datos, se construyó un modelo relacional que se compone de las siguientes relaciones:

**Inmueble**(idInmueble, operadorAsociado, Categoria, Capacidad, Ubicacion, Menaje, TamanoEnSqMt)

**Operador**(idOperador, relacionConUniandes)

**PersonaNatural**(idOperador, ofreceAlojamientoPorDias)

**EmpresaAlojamiento**(idOperador, tipoEmpresa, horaApertura, horaCierre)

**Servicio**(ofertaAsociada, nombre, descripcion, horaApertura, horaCierre, costoAdicional)

**OfertaAlojamiento**(idOferta, inmuebleAsociado, costoBase, ofertaActiva, ofertaDisponible, modalidadTemporal)

**Cliente**(idCliente, vinculoConUniandes)

**Reserva**(idClienteAsociado, idOfertaAsociada, fechaReserva, fechaInicialReserva, fechaFinalReserva, promocionesActivas, fechaCancelacion)

A continuación, se pueden visualizar cada una de las relaciones en conjunto con sus restricciones de dominio.

Inmueble			
idInmueble	operadorAsociado	Categoria	Capacidad
PK	FKOperador.idOperador, PK	SA	NN, UA
192837465	987605431	"Business"	4

Ubicacion	Menaje	TamanoEnSqMt
NN, UA	NN, UA	NN, UA
"Cra 28b #45A-80, Teusaquillo, bogotá"	"Se incluye un set de cubiertos, un vaso y un plato h	60

Relación 1. Inmueble

Vivienda		
idInmueble	tipoVivienda	numeroHabitaciones
FKInmueble.idInmueble, PK	NN, UA	NN
192837465	"Individual"	2

Relación 2. Vivienda

Habitacion	
idInmueble	tipoHabitacion
FKInmueble.idInmueble, PK	NN, UA
192837462	"Compartida"

Relación 3. Habitacion

Operador	
idOperador	relacionConUniandes
PK	NN
987605431	"Miembro de la comunidad Fenicia"

Relación 4. Operador

PersonaNatural	
idOperador	ofreceAlojamientoPorDias
FKOperador.idOperador, PK	NN, UA
987605431	TRUE

Relación 5. PersonaNatural

EmpresaAlojamiento			
idOperador	tipoEmpresa	horaApertura	horaCierre
FKOperador.IdOperador, PK	NN		
987605432	"Vivienda Universitaria"	8:00	22:00

Relación 6. EmpresaAlojamiento

Servicio		
ofertaAsociada	nombre	descripcion
FKOfertaAlojamiento.idOferta, PK	PK	
132465879	"Acceso a Jacuzzi"	"El cliente puede acceder al jacuzzi principal del hotel"

horaApertura	horaCierre	costoAdicional
		NN
10:00	22:00	0

Relación 7. Servicio

OfertaAlojamiento		
idOferta	precioBase	inmuebleAsociado
PK	NN	FKInmueble.idInmueble, PK
132465879	12000000	192837465

ofertaActiva	ofertaDisponible	modalidadTemporal
NN, SA	NN, SA	NN, UA
TRUE	FALSE	"Plan Semestral"

Relación 8. OfertaAlojamiento

Cliente	
idCliente	vinculoConUniandes
PK	NN
334567898	"Estudiante de Posgrado"

Relación 9. Cliente

Reserva		
idClienteAsociado	idOfertaAsociada	fechaReserva
FKCliente.idCliente, PK	FKOfertaAlojamiento.idOferta, PK	PK
334567898	132465879	26/02/2023

fechaInicialReserva	fechaFinalReserva	promocionesActivas	fechaCancelacion
NN	NN	NN	
28/02/2023	15/03/2023	"No hay promociones activas"	NULL

Relación 10. Reserva

## 4.2 Análisis del nivel de normalización del modelo

**Primera Forma Normal (1FN):** Debido a que todos los atributos de las relaciones son atómicos, es decir, no contienen datos multivalor o colecciones de datos (como una lista o array de elementos), es posible afirmar que se cumple con los requisitos de la primera forma normal.

En forma de aclaración, se tiene en cuenta que el tipo de dato “Date” usado para las fechas y horas del modelo construido cuenta como un solo valor a pesar de contener información de días, meses, año y hora.

**Segunda Forma Normal (2FN):** Para que una relación cumpla esta forma, es necesario que se encuentre en primera forma normal y que sus atributos no primos dependan de **forma completa** de las llaves candidatas.

**Para la relación Inmueble:** La llave {idInmueble, operadorAsociado} y la llave {idInmueble, Categoria} logran determinar de forma completa todos los atributos no primos de la relación, por tanto, cumple la 2FN.

**Para la relación Habitación y Vivienda:** La llave foránea {idInmueble} logra determinar de forma completa sus demás atributos, por tanto, cumple la 2FN.

**Para la relación Operador:** La llave idOperador puede determinar de forma completa el atributo relacionConUniandes (no primo), por tanto, cumple la 2FN.

**Para la relación PersonaNatural:** La llave idOperador puede determinar de forma completa el atributo ofreceAlojamientoPorDias (no primo), por tanto, cumple la 2FN.

**Para la relación EmpresaAlojamiento:** La llave {idOperador, tipoEmpresa} logra determinar de forma completa el resto de los atributos, por tanto, cumple la 2FN.

**Para la relación Servicio:** La llave candidata {ofertaAsociada, nombre} puede determinar el resto de atributos que son no primos, por tanto, cumple la 2FN.

**Para la relación OfertaAlojamiento:** la llave {idOferta, inmuebleAsociado} pueden determinar de forma completa todos los demás atributos que son no primos, por tanto, cumple la 2FN.

**Para la relación Cliente:** La llave idOperador puede determinar de forma completa el atributo relacionConUniandes (no primo), por tanto, cumple la 2FN.

**Para la relación Reserva:** La llave candidata {idClienteAsociado, idOfertaAsociada, fechaReserva} logra determinar de forma completa los demás atributos de la relación, por tanto, cumple la 2FN.

**Conclusión:** Todas las relaciones modeladas cumplen la Segunda Forma Normal.

**Tercera Forma Normal (3FN):** De forma general, debido a que ninguno de los datos modelados en los atributos es un dato derivado, no hay problemas relacionados con la transitividad de atributos no primos. En lugar de modelar un atributo llamado precioFinal en la relación “ofertaAlojamiento” se modeló uno llamado precioBase para que no fuese un dato derivado de ciertos cálculos.

Por otra parte, al analizar de forma detenida cada relación propuesta en el modelo, las únicas relaciones de transitividad que se identifican se dan entre atributos primos y no primos, como en el caso de la relación Inmueble, donde el idInmueble determina la Categoria del inmueble y la Categoria puede determinar el tamaño en metros cuadrados, la capacidad y el menaje. En los demás casos, no es posible que un atributo no primo llegue a determinar otro no primo por medio de inferencia transitiva.



**Conclusión:** Todo el modelo relacional propuesto se encuentra en Tercera Forma Normal (3FN).

## 5 Control de calidad del modelo

### 5.1 Diagramas de secuencia



Diagrama 1. Diagrama de secuencia de RF4

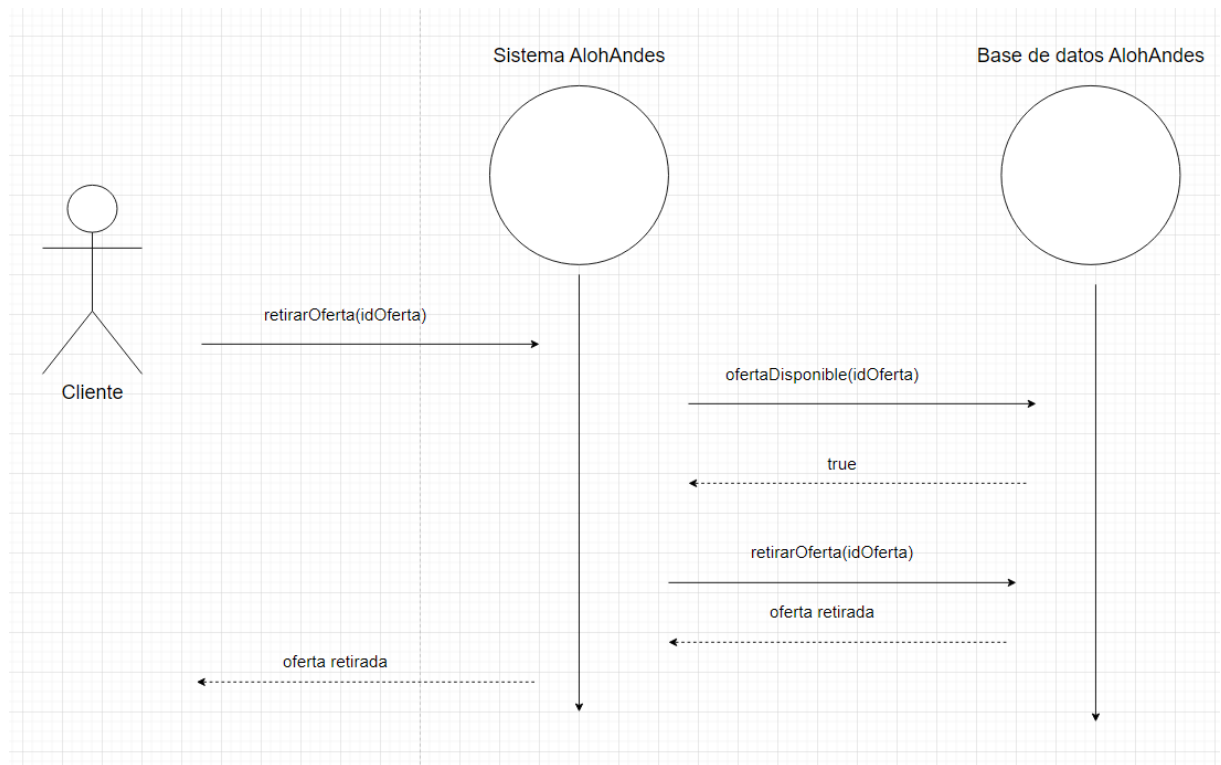


Diagrama 2. Diagrama de secuencia de RF6

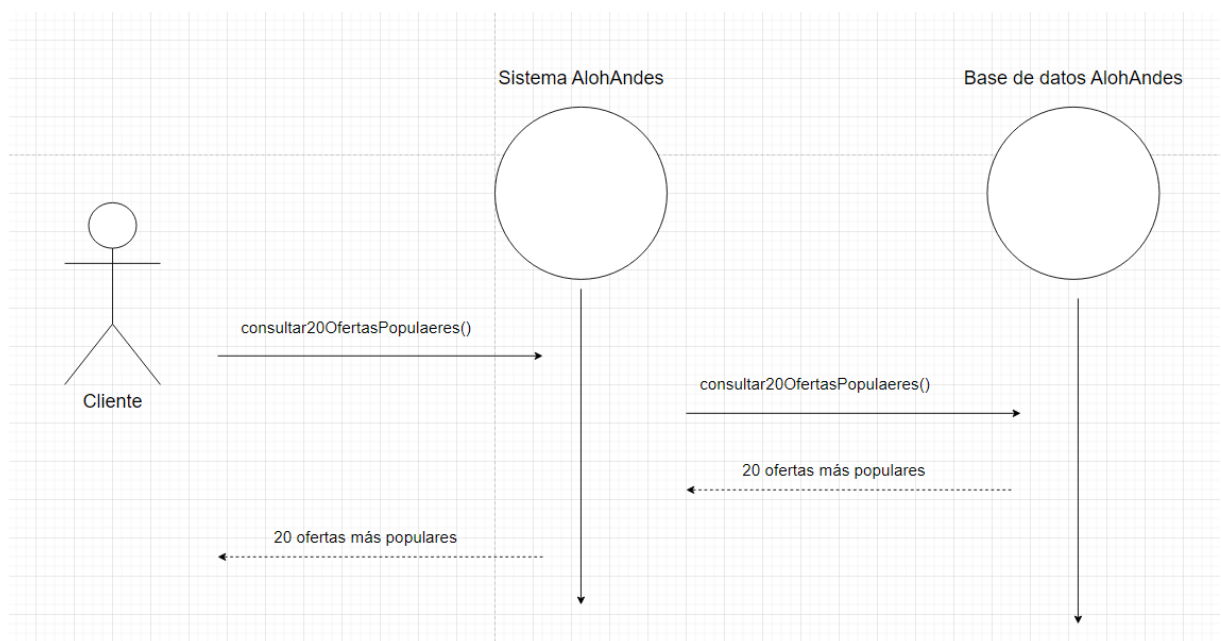


Diagrama 3. Diagrama de secuencia RFC2

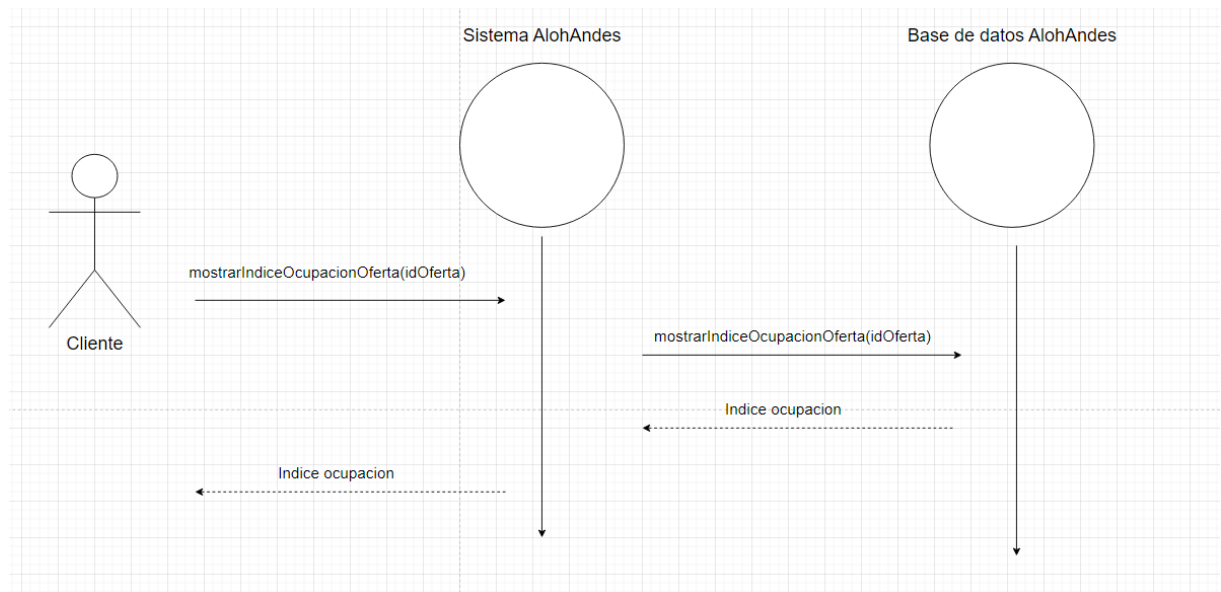


Diagrama 4. Diagrama de secuencia RFC3

## 5.2 Requerimientos No Funcionales

**Privacidad:** Para garantizar la privacidad de los datos hemos tenido en cuenta que todos los requerimientos funcionales están asociados a un tipo de usuario y que además todos los usuarios que deseen consultar las bases de datos y los resultados de los requerimientos funcionales deben tener determinados permisos para ver la información, y de este modo se garantiza la privacidad de la información de los usuarios.

**Persistencia:** La mayoría de los requerimientos funcionales del negocio deben guardar la información para utilizar esta información después y es por esto que se tiene en cuenta la persistencia en los requisitos funcionales, y así garantizamos la persistencia en el sistema.

**Concurrencia:** La concurrencia está presente en muchos de los requerimientos funcionales, sobre todo en los que son transaccionales y justamente identificamos los requerimientos transaccionales para poder manejar la concurrencia cuando pueda ocurrir, y de este modo garantizar la concurrencia y asimismo la transaccionalidad del sistema.

**Distribución:** El sistema se modeló como un sistema de negocio que maneja la lógica del negocio y también se modeló una base de datos que contiene toda la información persistente de manera que las consultas se hacen a la base de datos, que es una unidad donde los datos están centralizados y es de ahí de donde los demás recursos obtienen la información.

## 6 Conclusiones

A partir del entendimiento del problema, el análisis de los conceptos que lo componen y la ideación de un diseño para solucionar los requerimientos funcionales y no funcionales que se

solicitan para el sistema ALOHANDES, fue posible generar los 10 conceptos observados en el diagrama UML y luego su representación como tablas (relaciones) en el diseño relacional de la base de datos que este servicio requiere para funcionar de forma deseada. Las relaciones que terminan siendo el núcleo de la solución son: **Inmueble, Habitación, Vivienda, Operador, PersonaNatural, EmpresaAlojamiento, Servicio, OfertaAlojamiento, Cliente y Reserva.**

Después de analizar el nivel de normalización de todo el modelo a partir del análisis de cada una de sus relaciones, fue posible identificar que todas cumplen tanto la primera como la segunda y tercera formas normales. Es importante recalcar que desde el diseño del diagrama conceptual, los requisitos de estas formas normales fueron tenidos en cuenta para lograr este nivel de calidad de la base de datos propuesta.

Ahora que se tiene claridad del problema y la solución propuesta, junto a los detalles del modelo relacional, es posible pensar en su implementación, la cual no sería muy compleja al contar con un nivel alto de especificidad en

## 7 Bibliografía

1. *Enunciado del caso Iteración 1. Curso Sistemas Transaccionales 2023-10. Tomado de los contenidos del curso en Bloque Neón.*

2. *NORMALIZACIÓN DE BASES DE DATOS RELACIONALES.* (n.d.-b). CODIDEEP.  
Tomado de <https://codideep.com/blogpost/normalizacion-de-bases-de-datos-relacionales>

*NORMALIZACIÓN DE BASES DE DATOS RELACIONALES.* (n.d.-b). CODIDEEP.  
<https://codideep.com/blogpost/normalizacion-de-bases-de-datos-relacionales>