

Iteración 3 del Proyecto del Curso Sistemas Transaccionales - Caso ALOHANDES

Alonso Hernández Tavera, Cristian Bernardo Acuña Silva

Documento de la Iteración 3 – Caso ALOHANDES

Universidad de los Andes, Bogotá, Colombia

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

Fecha de presentación: Mayo 9 de 2023

Tabla de contenido

1	Introducción y Contextualización	1
2	Análisis y modelo conceptual	2
2.1	Modelo conceptual construido	2
2.2	Consideraciones y aclaraciones del modelo conceptual	3
3	Diseño de la base de datos.....	4
3.1	Modelo relacional construido	4
3.1.1	Cambios al modelo relacional	6
3.2	Análisis del nivel de normalización del modelo relacional	6
4	Resultados logrados Iteración 3	7
4.1	Creación de tablas en la base de datos	7
4.2	Carga de datos en la base de datos	7
4.3	Desarrollo de sentencias SQL para solucionar los requerimientos.....	8
4.4	Creación de clases en Java de los componentes del negocio.....	8
4.5	Implementar los nuevos requerimientos sobre la base de datos	8
5	Resultados no logrados Iteración 3	8
6	Documentación de los Requerimientos Iteración 3.....	8
7	Conclusiones	9
8	Bibliografía.....	10

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 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.

2.1 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.

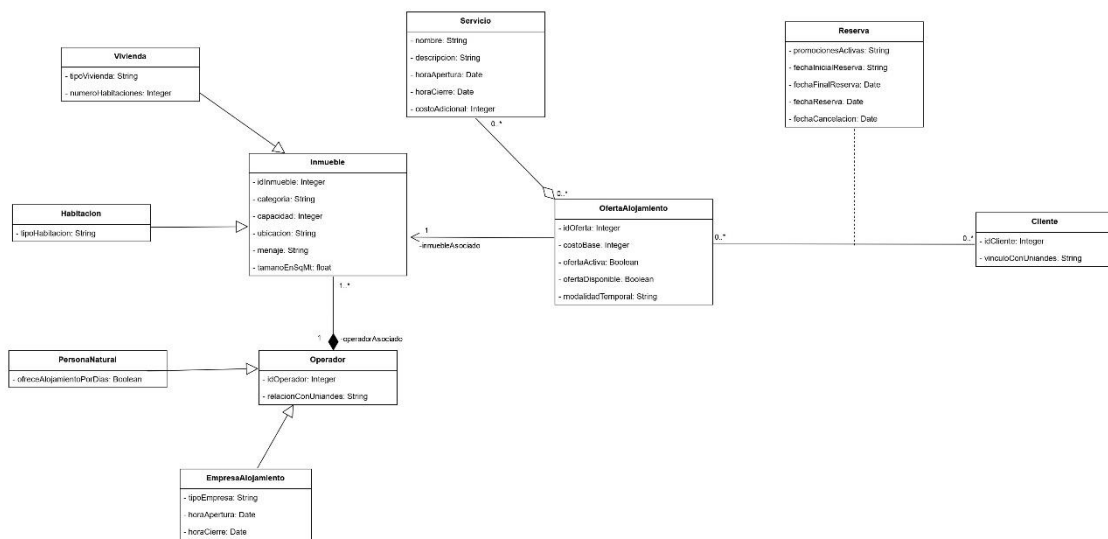


Figura 1. Diagrama conceptual construido para el caso ALOHANDES

2.2 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.

3 Diseño de la base de datos

3.1 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 h60	

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

3.1.1 Cambios al modelo relacional

Debido a que el RF7 nos pide poder identificar de manera única las reservas colectivas que se realicen se decidió crear una nueva relación. Esta relación tiene dos columnas una con el id de la reserva colectiva y otra columna con el id de la oferta asociada a la reserva. A continuación se muestra la tabla creada

ReservaColectiva	
idReservaColectiva	idOfertaAsociada
PK	FK(ofertaAlojamiento.idOferta),PK
5	2

3.2 Análisis del nivel de normalización del modelo relacional

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 Categoría del inmueble y la Categoría 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).

4 Resultados logrados Iteración 3

4.1 Creación de tablas en la base de datos

Se logró crear tablas que representan las relaciones presentes en el modelo relacional.

4.2 Carga de datos en la base de datos

Se logró cargar dichas tablas con datos artificiales con el objetivo de probar el funcionamiento de la base de datos.

4.3 Desarrollo de sentencias SQL para solucionar los requerimientos

Se generaron sentencias SQL para los diferentes requerimientos funcionales planteados. Estas sentencias SQL se generaron con el fin de posteriormente incorporarlas en la aplicación.

4.4 Creación de clases en Java de los componentes del negocio

Se crearon las clases en Java que reflejan la lógica del negocio y que tienen como fin manejar los datos en la aplicación.

4.5 Implementar los nuevos requerimientos sobre la base de datos

Fue posible sin problema esta vez, implementar los requerimientos y mostrarlos por medio de la Interfaz.

5 Resultados no logrados Iteración 3

No fue posible anexar los escenarios de prueba en este documento, por lo tanto, se va a anexar al repositorio en la carpeta documentos.

6 Documentación de los Requerimientos Iteración 3

Requerimientos Funcionales de Modificación:

RF 7:

Para este requerimiento lo primero que se hace es preguntarle al usuario cuales son las reservas que desea realizar. Después de tener la información procedemos a verificar que las oferta relacionadas a la reserva que quiere hacer el usuario estén disponibles. En caso de estar disponibles se registran las reservas, de lo contrario se envía un mensaje de error.

Posteriormente se añade a la relación de ofertas colectivas el id de la oferta colectiva junto con las ofertas asociadas a las reservas realizadas. Esto es posible por que una oferta solo puede ser asociada a una reserva.

RF 8:

Para este requerimiento se pregunta al usuario el id de reserva colectiva que desea eliminar. Teniendo el id de la reserva colectiva a eliminar se procede a buscar todas las ofertas asociadas al id de reserva colectiva. Una vez de se tienen los id de las ofertas asociadas se procede a eliminar de la tabla reservas estos ids. Por ultimo se elimina de la tabla de reserva colectiva todos los valores que coincidan con el id de reserva colectiva a eliminar

RF 9:

Para este requerimiento se le pide al usuario el id de la oferta que se va a deshabilitar, en caso de que la oferta no tenga reservas asociadas, simplemente se cambia la disponibilidad de la reserva al estado que muestra que no está disponible. Si sí tiene reservas asociadas se procede a eliminar las reservas asociadas y a la oferta deshabilitada y posteriormente se asignan a otra oferta que sí esté disponible

RF 10:

Para este requerimiento se pregunta al usuario por el id de la oferta que desea rehabilitar, teniendo esto se procede a cambiar el estado de la oferta de no disponible, a disponible.

Requerimientos Funcionales de Consulta:

RFC 7:

Para solucionar este requerimiento, fue necesario implementar 3 métodos diferentes que permitieran resolver las preguntas de análisis sobre la base de datos, estos métodos son:

1. CalcularFechasMayorDemanda
2. CalcularFechasMayoresIngresos
3. CalcularFechasMenorOcupación

En conjunto, estos métodos permiten resolver el requerimiento planteado.

RFC8 - Encontrar los clientes frecuentes

El método `encontrarClientesFrecuentes(long idInmueble)` en la clase `PersistenciaAlohandes` se usa para encontrar la información de los clientes frecuentes de un alojamiento específico. Un cliente se considera frecuente si ha utilizado (o tiene reservado) ese alojamiento al menos en tres ocasiones o durante al menos 15 noches.

Este método realiza una consulta SQL para seleccionar aquellos clientes que cumplen con estos criterios. La consulta se realiza en la tabla `RESERVA`, uniendo las tablas `OFERTAALOJAMIENTO` e `INMUEBLE` en el `IDINMUEBLE`. La información obtenida incluye el `IDCLIENTE`, el número de reservas realizadas y la duración total de las reservas.

RFC9 - Encontrar las ofertas de alojamiento que no tienen mucha demanda

El método `encontrarOfertasPocaDemanda()` en la clase `PersistenciaAlohandes` se utiliza para encontrar las ofertas de alojamiento que no han recibido clientes en periodos superiores a 1 mes.

Este método realiza una consulta SQL en la tabla `OFERTAALOJAMIENTO`, uniendo las tablas `RESERVA` e `INMUEBLE` en el `IDINMUEBLE`, y seleccionando aquellas ofertas que no han tenido reservas en más de un mes. La consulta devuelve una lista de ofertas de alojamiento, incluyendo el `IDOFERTA`, la última fecha de reserva y la cantidad de días desde esa fecha hasta la fecha actual.

7 Conclusiones

Se tuvo en cuenta el modelo conceptual y el modelo relacional que se planteó previamente con el objetivo de implementar el negocio Alohandes. Para esto se usaron conocimientos como SQL con el objetivo de realizar diferentes requerimientos funcionales planteados relacionados con el negocio Alohandes. Se creó una base de datos con datos artificiales con el objetivo de verificar el funcionamiento de la base de datos. Se creó un proyecto en java con el objetivo de

plasmar el negocio a través de una aplicación. A pesar de que no se logró el funcionamiento de la aplicación, se cumplieron la mayoría de los objetivos.

8 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>