“Iteración 2: Alohandes”

José D. Flórez Ruiz, Carlos M. Muñoz Almeida
Contexto de Presentación del documento
Universidad de los Andes, Bogotá, Colombia
{jd.florezr1, c.munoza}@uniandes.edu.co
Fecha de presentación: Abril 10 de 2023

Tabla de contenido

1 Introducción 1
2 Análisis Requerimientos Funcionales 1
3 Modelos 7
3.1 Modelo UML 7
3.2 Modelo relacional 7
4 Instrucciones de uso 9
4.1 Administradores 10
4.2 Clientes 10
4.3 Operadores usuarios 10
4.4 Operadores empresas 10
5 Resultados 10
5.1 Logros 10
5.2 ¿Qué se puede mejorar? 11
6 Balance de plan de pruebas 11
7 Reglas de negocio: supuestos y consideraciones 11

1 Introducción

En el presente documento, se ofrece un resumen conciso de los resultados obtenidos en nuestro proyecto para la asignatura de Sistemas Transaccionales, en el que abordamos el caso de Alohandes, un sistema dedicado a la publicación y reserva de alojamientos para miembros de una comunidad universitaria. Se nos presentaron seis Requerimientos Funcionales (RF) y seis Requerimientos de Consulta (RFC) como parte del proyecto. Logramos implementar todos los requerimientos funcionales y los primeros cuatro requerimientos de consulta. A continuación, expondremos los modelos solicitados, los resultados alcanzados y aquellos que no pudimos lograr, así como un balance del plan de pruebas y los supuestos adicionales sobre las reglas de negocio encontradas en el caso de estudio. Este documento tiene como objetivo proporcionar una visión clara y comprensible del trabajo realizado y sus resultados.

2 Análisis Requerimientos Funcionales

Nombre	RF1. **REGISTRAR LOS OPERADORES DE ALOJAMIENTO PARA ALOHANDES**
Resumen	Los operadores necesitan ser registrados en la base de datos de AlohAndes.
Entradas	

Los operadores se registrarán en la base de datos proveyendo su identificación, nombre y tipo de vinculación con la universidad que están definidas por los siguientes tipos:

- Estudiante
- Profesor
- Padre de Estudiante
- Trabajador
- Externo (Personas naturales)

Resultados

El resultado de esta transacción será un retorno por parte del sistema confirmando en el caso de que la transacción sea satisfactoria una respuesta de éxito, y en el caso contrario en el que la transacción no se realice de manera satisfactoria, se le debe enviar una respuesta de error/problemas.

RNF asociados

Este requerimiento exige un comportamiento transaccional puesto que necesitamos mantener una alta calidad en el manejo de los datos. Esto lo podemos lograr mediante un manejo transaccional que cumpla con ACID.

Este requerimiento necesita hacer uso de persistencia ya que como se puede intuir es de gran importancia mantener de manera permanente los datos después de que se realice la transacción.

Nombre	RF2. **REGISTRAR PROPUESTAS DE ALOJAMIENTOS PARA ALOHANDES.**
Resumen	Los posibles operadores pueden brindar propuestas a AlohAndes para que, en caso de ser seleccionados, se conviertan en proveedores de AlohAndes.

Entradas

Dependiendo del tipo de operador que esté tratando de registrar una propuesta de alojamiento para AlohAndes tendrán distintas opciones habilitadas como es el caso cuándo se es miembro de la comunidad institucional el individuo podrá registrar una habitación dentro de su hogar, un apartamento en alquiler o una vivienda temporal. Sin embargo, hay casos en los que no tienen todas estas habilitadas, como es el caso de las personas naturales externas a la universidad las que solo tendrán la posibilidad de registrar una habitación dentro de su hogar o una vivienda temporal.

El caso es distinto cuando una empresa trata de registrar una habitación. El tipo de habitación variara dependiendo del tipo de empresa que sean. Por ejemplo: hoteles solo podrán registrar habitación de hotel, hostales solo podrán registrar habitación de hostal y vivienda universitaria solo podrá registrar una habitación de vivienda universitaria. Las entidades cambian debido a que cada una tiene reglas de funcionamiento diferentes las cuales serían imposible de plasmar todas en conjunto.

Las entradas necesarias dependiendo cada caso son:

- **Habitación huésped / Habitación dentro de un hogar:**
    1. Comidas: Integer
    2. Acceso a Cocina: Boolean
    3. Tipo de baño: String
    4. Tipo de habitacion: String
    5. Cuota de luz: Integer
    6. Cuota de telefono: Integer
    7. Cuota de agua: Integer
    8. Cuota de tv cable: Integer
    9. Descuento de cada mes extra: Integer
- **Apartamento Alquiler:**
    1. Amoblado: Boolean
    2. Servicio público incluido: Boolean
    3. Tv incluido: Boolean
    4. Internet incluido: Boolean
    5. Administración incluida: Boolean
- **Vivienda Temporal:**

1. Número de habitaciones: Integer
2. Mensaje: String
3. Precio seguro de arrendamiento: Integer
4. Características de seguro: String
5. Días que ha sido alquilado en el año: Integer

- Habitación hotel:
    1. Tipo de habitación: String
    2. Tamaño habitación: String
    3. Tiene Bañera: Boolean
    4. Tiene Yacuzzi: Boolean
    5. Tiene sala: Boolean
    6. Tiene Cocineta: Boolean
- Habitación vivienda universitaria:
    1. Tipo de habitación: String
    2. Capacidad de habitación: Integer
    3. Descripción de menaje: String
- Habitación hostal:
    1. Aforo: Integer

Además, algunos otros datos que todos tienen que dar en general:

1. Ubicación: String
2. Costo base: Integer
3. Duración mínima: Integer
4. Costo Total: Integer

Resultados

Tras él envió de los datos la base de datos notificara si la transacción fue realizada exitosamente. Si efectivamente salió bien, se envía mensaje de éxito en el retorno, y en el caso contrario se envía un mensaje de error.

RNF asociados

Este requerimiento necesita transaccionalidad ya que como se evidencia en la descripción es necesario manipular los datos por medio de un conjunto de queries en forma de una transacción con el fin de poder registrar todos los datos necesarios dentro de la tabla necesarias.

Por otro lado, este requerimiento hace uso de persistencia esto con el fin de que tras haberse realizado la transacción los cambios realizados por este permanezcan en la base de datos.

Nombre	RF3. **REGISTRAR LAS PERSONAS HABILITADAS PARA UTILIZAR LOS SERVICIOS**

Resumen	Los clientes necesitan que su información sea registrada en la base de datos de AlohAndes.
Entradas	
Las personas habilitadas para utilizar los servicios proveerán los siguientes datos: • Nombre • Tipo de vínculo con la universidad • Identificación • Teléfono • Correo Electrónico	
Resultados	
Cuando se intente realizar la transacción de registro de un usuario habrá dos respuestas. Una de estas es la de éxito la cual se mostrará en dado caso todas las quieres de la transacción se hayan podido realizar exitosamente, en el caso contrario se retornará un error.	
RNF asociados	
Se necesita transaccionalidad ya que por medio de esta se manipula la base de datos de la manera que deseamos por medio de las queries pertenecientes a la transacción. Haciendo imperativo para poder manipular la base de datos de manera segura, manteniendo la integridad de los datos.	
Se necesita persistencia puesto que los datos del cliente deben conservarse tras realizarse la transaccion.	

Nombre	RF4. **REGISTRAR UNA RESERVA**
Resumen	Los clientes deben tener la capacidad de realizar una reserva en el sistema. El sistema debe considerar las preferencias del cliente. Por ejemplo, vivienda compartida, internet incluido, cocineta, etc.
Entradas	
Se reciben datos del cliente, tales como: su tipo de vinculo con la universidad, su identificación y su nombre. El tipo de vinculo puede ser: estudiante, egresado, empleado, profesor, padre de estudiante, profesor invitado, persona registrada en evento de Uniandes.	
Resultados	
Registro de una nueva reserva.	
Notificación al cliente y al operador de que se realizó una nueva reserva exitosa.	
RNF asociados	

Este requerimiento necesita transaccionalidad puesto que se debe garantizar que la operación de realizar una reserva se haga de manera completa y segura. También, es necesario contar una alta integridad que los datos que el cliente provea.
Este requerimiento necesita persistencia ya que la reserva debe ser almacenada.

Nombre	RF5. **CANCELAR UNA RESERVA**
Resumen	El cliente debe ser capaz de cancelar una reserva dentro del sistema.

Entradas
Se requieren datos de la reserva, tales como su id, fecha de inicio y fecha de fin.
Se requieren datos del cliente, tales como su identificación y nombre.
Resultados
Actualización en la base datos.
Notificación al operador y al cliente de que la reserva fue cancelada exitosamente.
RNF asociados
Se requiere transaccionalidad puesto que es necesario garantizar la completitud de la cancelación de reserva. Es decir, tanto el cliente como el operador deben ser conscientes de que la reserva fue cancelada. Esto por medio de la actualización de la base de datos.
Este requerimiento debe tener persistencia ya que se debe guardar en los registros que la reserva fue cancelada.

Nombre	RF6. **RETIRAR UNA OFERTA DE ALOJAMIENTO**
Resumen	El operador puede ser capaz retirar una oferta de alojamiento dentro del sistema. Es importante resaltar que la oferta de alojamiento se puede retirar solo después de la finalización de la última reserva vigente.

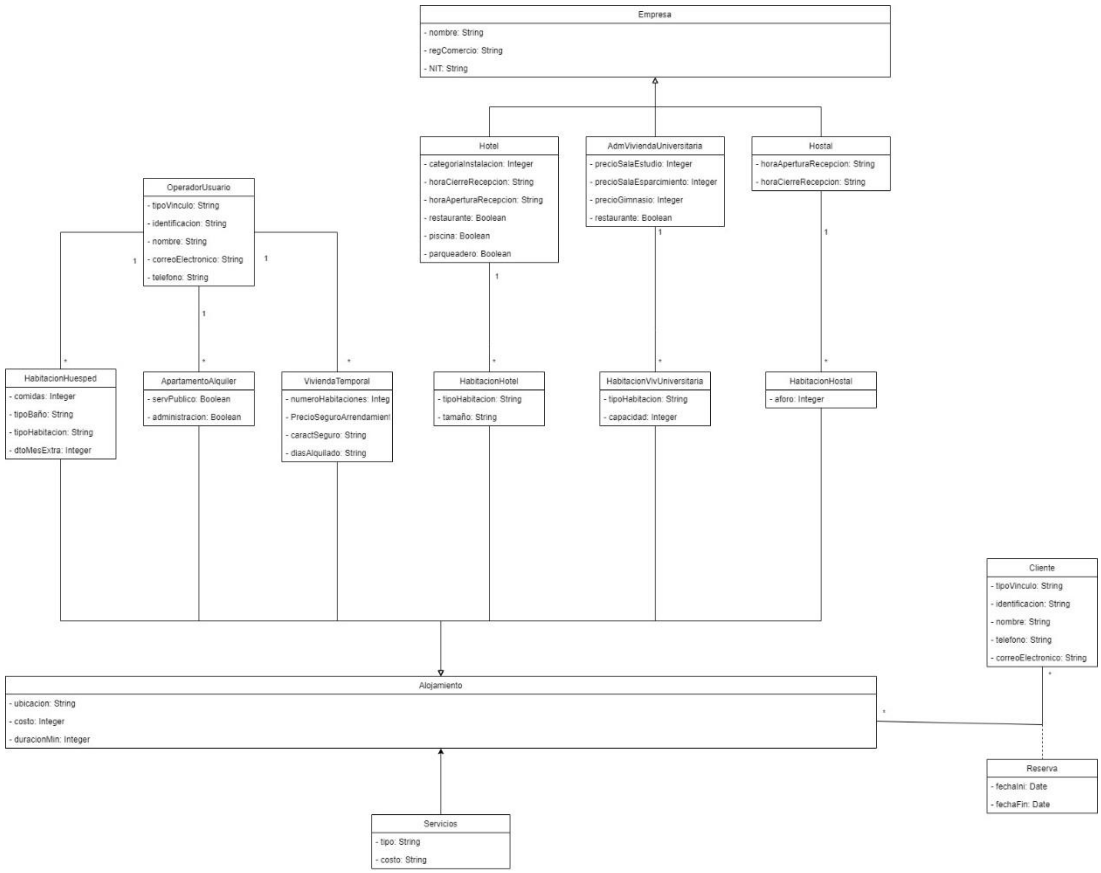
Entradas
Se requiere conocer que operador va a retirar que oferta. De modo que, se recibe el id del alojamiento que el operador va retirar.
En caso de que el operador sea una persona natural o un miembro de unidades, se recibe su identificación. En caso de que el operador sea una empresa se recibe su numero de registro en la cámara de comercio.
Resultados
Actualización en la base de datos.
Notificación al operador de que su oferta fue retirada con éxito.
RNF asociados

Este requerimiento tiene transaccionalidad ya que se necesita modificar la base datos de una manera íntegra y completa de manera que se retire por completo la oferta del operador. Y así, ya no sea visible para los clientes.
Este requerimiento hay persistencia ya que se debe guardar el registro de que la oferta fue retirada.

3 Modelos

En anteriores iteraciones se presentó un modelo el cual fue actualizado en virtud de mejorar la calidad de la implementación de los requerimientos solicitados. Se agrego una clase *Servicio*, y una clase asociación *AlojamientoServicio.* Además, se eliminaron atributos que no resultaban convenientes para la implementación del proyecto.

3.1 Modelo UML



* Para revisar a mayor detalle revise en la carpeta “Modelo Conceptual Alohandes”

3.2 Modelo relacional

CKs

CK1	('Y', 'N')
CK2	('privado', 'compartido')
CK3	('individual', 'compatida')
CK5	('profesor', 'empleado', 'egresado', 'estudiante', 'padre de estudiante', 'externo')
CK6	('estándar', 'semisuite', 'suites')
CK7	('salaEsparcimiento', 'tvCable', 'gimnasio', 'salaEstudio', 'wifi', 'banjera', 'sala', 'jacuzzi', 'menaje'

OperadorUsuario

identificacion	nombre	tipoVinculo	correoElectronico	telefono
PK, UA	NN	NN, CK5	NN	NN

HabitacionHuesped

idAlojamiento	comidas	tipoBaño	tipoHabitacion	dtoMesExtra	IDENTIFICACIONOPERADORUSUARIO
PK, FKAlojamiento.id	NN, CK >= 0	NN, CK2	NN, CK3	NN, 100 >= CK > 0	NN, FKOperadorUsuario.identificacion

ApartamentoAlquiler

idAlojamiento	servPublico	administracion	IDENTIFICACIONOPERADORUSUARIO
PK, FKAlojamiento.id	NN, CK1	NN, CK1	NN, FKOperadorUsuario.identificacion

ViviendaTemporal

idAlojamiento	numeroHabitaciones	PRECIOSEGUROARRENDAMIENTO	caracSeguro	diasAlquilado	propietario
PK, FKAlojamiento.id	NN, CK > 0	NN, CK > 0	NN	NN, CK <= 30	NN, FKOperadorUsuario.identificacion

Cliente

identificacion	nombre	tipoVinculo	correoElectronico	telefono	ultimaFechaReserva
PK, UA	NN	NN, CK5	NN	NN	NN

Reserva

id	fechaIni	fechaFin	identificacionCliente	idAlojamiento
PK, SA	NN	NN	NN, FKCliente.Identificacion	NN, FKAlojamiento.id

Alojamiento

id	ubicacion	duracionMin	Costo
PK, SA	NN,	NN, CK>0	NN, CK > 0

Hotel

regComercio	NIT	nombre	restaurante	parqueadero	piscina
PK, UA	NN	NN	NN, CK1	NN, CK1	NN, CK1

HabitaciónHotel

id	tipoHabitación	tamaño	idHotel
PK, FKAlojamiento.id	NN, CK6	NN	**NN, FKHotel.regComercio**

ViviendaUniversitaria

regComercio	NIT	nombre	precioSalaEstudio	precioSalaEsparcimiento	precioGimnasio	Restaurante
PK, UA	NN	NN	NN, CK(>=0)	NN, CK(>=0)	NN, CK(>=0)	NN

HabitaciónVivUniversitaria

id	tipoHabitación	capacidad	idViviendaUniversitaria
PK, FKAlojamiento.id	NN	CK(>0)	**NN, FKAdmViviendaUniversitaria.regComercio**

Hostal

regComercio	NIT	nombre	horaAperturaRecepción	horaCierreRecepción
PK, UA	NN	NN	NN	NN

HabitaciónHostal

id	aforo	idhostal
PK, FKAlojamiento.id	CK(>=0)	**NN, FKHostal.regComercio**

Servicio

id	tipo
PK	NN,CK 7

AlojamientoServicio

idAlojamiento	idServicio	Costo
PK, FKAlojamiento.id	FKServicio.id	CK (>=0)

4 Instrucciones de uso

A continuación, se muestra el manejo de la aplicación para los distintos usuarios que tienen acceso. Cabe mencionar que, dentro del proyecto, en la carpeta llamada *docs*, se encuentra información adicional y archivos de prueba.

4.1 Administradores

Los administradores de Alohandes tendrán acceso a todas las funcionalidades de la aplicación. Dentro de la interfaz se encuentra una sección exclusiva para administradores. Se solicitará una contraseña (Admin) y una clave (Admin), para proteger el acceso a esta sección.

4.2 Clientes

Los clientes, todo tipo de persona que tenga un vinculo con la universidad de los Andes, podrán crear su cuenta, e iniciar sesión ingresando los datos solicitados, una vez realizado el inicio de sesión, cada cliente podrá hacer gestión de sus reservas. Para realizar una reserva, el cliente, en primera instancia, deberá buscar el alojamiento que desea. Luego, debe tener presente el id del alojamiento que desea para ingresarlo al momento de realizar una reserva.

4.3 Operadores usuarios

Un operador usuario, aquella persona natural que puede ofertar distintos tipos de alojamiento como lo son: habitaciones, apartamentos y viviendas. Posee su propia sección, en la que deberá, en primer lugar, ingresar su identificación (el operador usuario debió haber sido registrado previamente por un administrador para poder ingresar a la app). Así, podrá gestionar los alojamientos a su disposición.

4.4 Operadores empresas

Un operador empresa es aquel que esta registrado en la cámara de comercio, puede ser un hotel, un hostal o una vivienda universitaria. Este, cuenta con su propia sección en donde debe ingresar su registro en la cámara de comercio para poder ingresar (al igual que el operador usuario debió haber sido registrado previamente por un administrador). Una vez haya iniciado sesión, la empresa podrá gestionar los alojamientos a su cargo.

5 Resultados

5.1 Logros

En el proyecto, consideramos que se lograron satisfactoriamente todos los requerimientos funcionales y no funcionales que eran solicitados. Dado que, se pueden todos los operadores de alojamiento para Alohandes al igual que sus propuestas. Los clientes que son habilitados para usar los servicios también pueden registrase. Además, cada cliente puede realizar la reserva que desee y, asimismo, cancelarla. También, en la aplicación, es posible retirar las ofertas de alojamiento.

En lo que respecta a los requerimientos no funcionales, cumplimos con la privacidad puesto que cada usuario de Alohandes solo puede manipular y consultar la información que les es propia. Esto lo logramos por medio de la implementación de distintas interfaces para cada usuario, además del inicio de sesión que cada uno debe realizar. También, podemos decir que nuestra aplicación es persistente, puesto que cada vez que se ingresa un nuevo dato, este es almacenado en la base de datos, claramente siempre se verifica que los datos que son ingresados tengan sentido, para que no afecten la consistencia de la base de datos. Por otra parte, cumplimos también con la concurrencia y distribución, en la aplicación varios usuarios pueden solicitar varios requerimientos al mismo tiempo sin generar problemas. Y, para cumplir con la distribución, todos los datos están centralizados en una sola base de datos.

En relación con los requerimientos funcionales de consulta logramos todos los solicitados (RFC1, RFC2, RFC3 Y RFC4) exitosamente. En la aplicación es posible ver el dinero recibido por cada proveedor durante el año actual y el año corrido. También, es posible consultar las 20 ofertas más populares, la popularidad es medida sobre la cantidad de reservas que tiene un

alojamiento, entre más reservas, más popular. Además, es posible ver el índice de ocupación de cada alojamiento, este índice se mide por medio de la cantidad de días que es reservado u ocupado un alojamiento sobre todos los 365 días de un año. Por último, también es posible consultar los requerimientos que cumplan estén dentro de un rango de fechas y cumplan con un conjunto de servicios, este requerimiento es de vital importancia a la hora de que un cliente pueda realizar una reserva. Puesto que, en primer lugar, para hacer una reserva, se deben consultar los alojamientos para poder saber que id de alojamiento se va a ingresar en la reserva.

Para finalizar, también incluimos una cantidad considerable de datos para que la aplicación pueda ser probada. Dentro de la carpeta *docs*, se encuentran archivos .csv y un script el cual con simplemente se ejecutado adicionado todos los datos en la base de datos. También, dentro de la carpeta *docs*, se encuentran los scripts con las sentencias SQL usadas para cumplir con los requerimientos, además de otros archivos complementarios.

5.2 ¿Qué se puede mejorar?

Consideramos que nuestra aplicación puede llegar a ser un poco brusca con el cliente en algunos aspectos. Por lo cual, el diseño de la interfaz debe ser mejorado para futuras entregas. Por otro lado, nos gustaría mejorar la eficiencia y optimizar tanto como sea posible todas las operaciones realizadas en nuestra aplicación.

6 Balance de plan de pruebas

A lo largo del desarrollo de la aplicación, se diseñaron diversos escenarios de prueba con el objetivo de garantizar su correcto funcionamiento y la integridad y calidad de los datos en la base de datos. Para evaluar la precisión y calidad del modelo en la base de datos, se llevaron a cabo pruebas exhaustivas en cada tabla, lo que evidencia el adecuado manejo de los datos persistentes tanto en la base de datos como en el sistema en general. Estas pruebas se ejecutaron mediante consultas SQL específicas, que permitieron verificar de manera eficiente todos los aspectos mencionados anteriormente.

En nuestro plan de pruebas seguimos los escenarios recomendados, es decir, para cada tabla en nuestro proyecto, realizamos pruebas de unicidad, pruebas de integridad con FK y pruebas de integridad respecto a restricciones de chequeo. Todas las pruebas dieron resultados satisfactorios. Estas pruebas se encuentran en la carpeta *docs* del proyecto.

7 Reglas de negocio: supuestos y consideraciones

Dentro de Alohandes, consideramos las siguientes reglas de negocio:

- No se puede reservar un alojamiento en donde las fechas de reserva interfieran con otra reserva de este.
- Todos los alojamientos se encuentran relacionados con Uniandes.
- Un cliente no puede reservar más de un alojamiento en un mismo día.
- Para retirar un alojamiento, el alojamiento no debe tener reservas por hacer.
- El alojamiento en vivienda universitaria sólo está habilitado a estudiantes, profesores, empleados y profesores visitantes.
- Para eliminar un alojamiento, el que intenta eliminarlo tiene que ser el dueño o el administrador.
- Externos solo pueden realizar ofertas de alojamiento temporal.
- Alquiler apartamento y habitación vivienda Universitaria tienen que tener una reserva mínima de 30 días.