

# Iteración 1: Proyecto del curso

Sebastian Garcia 201630047, Mateo Saravia 201629885

Grupo B-09

Sistemas Transaccionales

Ingeniería de Sistemas y Computación

Universidad de Los Andes, Bogotá, Co.

{js.garcia1, m.saravia}@uniandes.edu.co

Marzo 18 de 2018

## Tabla de contenido

1	Introducción .....	1
2	Análisis del caso de estudio Alohandes .....	1
3	Diseño de la base de datos .....	9
3.1	Análisis del modelo de datos relacional.....	9
4	Balance de pruebas.....	10
5	Resultados obtenidos.....	11

## 1 Introducción

El objetivo de la siguiente iteración es desarrollar habilidades en el proceso de diseño de una aplicación transaccional, a partir de la descripción de un caso de negocio. En este proyecto, el caso de negocio gira en torno al alojamiento de la comunidad universitaria: Alohandes. Alohandes es un facilitador de opciones de alojamiento para la comunidad universitaria.

## 2 Análisis del caso de estudio Alohandes

Luego de estudiar la descripción, los objetivos y la misión de Alohandes, se elaboró un modelo conceptual, basado en clases, (Figura 1) que representa los roles y entidades de negocio fundamentales para poder desarrollar la aplicación transaccional. En este modelo se puede observar cómo un usuario puede ser un cliente o un operador que ofrece su inmueble a la comunidad. Alohandes impone ciertas restricciones sobre los usuarios para que sean únicamente aquellos asociados con la Universidad de Los Andes los que puedan gozar de este servicio de alojamiento. Se entiende como usuario asociado a la universidad los estudiantes, profesores, empleados, padres de estudiantes, egresados, invitados y empresas hoteleras reconocidas que cuenten con todos los papeles legales vigentes para poder ofrecer el servicio de hotelería.

En el diagrama de clases (Figura 1) se representa cada tipo de inmueble como una entidad independiente. Sin embargo, en esta aplicación, sólo pueden existir inmuebles que hayan sido propuestos por un operador. A lo anterior se suman una serie de reglas particulares para cada tipo de inmueble. Por ejemplo, un apartamento debe ser reservado con una cantidad mínima de 30 días mientras que una vivienda no puede superar los 30 días por año.

Otro aspecto a tener en cuenta es el contrato. En el siguiente diagrama de clases, el contrato es representado como una `Reserva` que un `Cliente` hace sobre una `Propuesta` en la aplicación. La `Reserva` contiene información útil como lo es la fecha de registro de la reserva, la fecha en la que empezaría el cliente a utilizar el inmueble, y en caso de ser necesario, se cuenta con una fecha de cancelación que determinaría una multa que el `Operador` le impone al `Cliente`. Adicionalmente, este contrato es el que se hace responsable de conocer el costo total a pagar por un inmueble, y la cantidad de noches que se piensa hacer uso de éste.

Finalmente, un Operador puede ofrecer varios servicios básicos en su propuesta. Se entiende como servicio básico la luz, televisión por cable, internet, un baño, telefono, agua, cocina, diferentes comidas como el desayuno, almuerzo o comida; y un apoyo social o academico. Los anteriores puede contar un con costo adicional, que afectaría el costo total del inmueble.

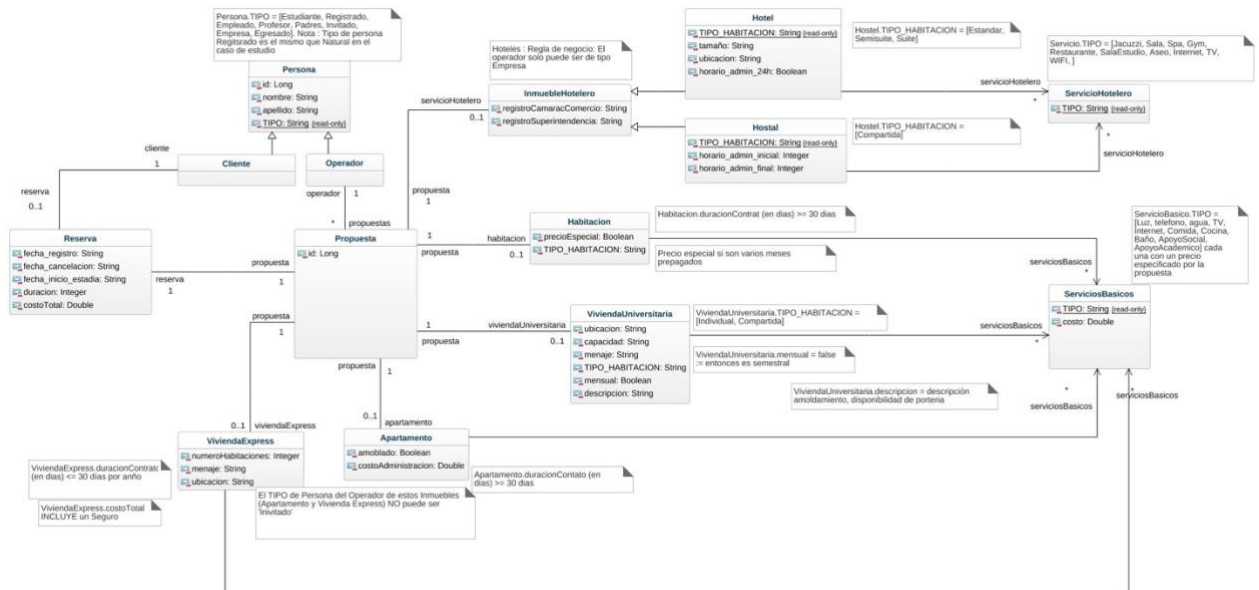


Figura 1. Modelo conceptual del caso de estudio.

A continuación se muestran los casos de uso.

## Casos de uso

Descripción General del Caso de Uso		
<b>CU01: registro de usuarios.</b>		
Registrar en la base de datos un usuario que puede ser un operador o un cliente. En este caso de uso se implementan las funcionales RF1 y RF3.		
Entidades Involucradas		
PERSONAS.		
Precondiciones		
El operador debe tener una vinculación con la institución: estudiante, egresado, acudiente de un estudiante, empleado o invitado.		
Flujo normal de Eventos		
	Usuario	Sistema
1	Selecciona la opción de “registro de usuario” en la interfaz.	Muestra un panel donde solicita los datos: tipo de identificación (cédula o nit), el número de la identificación, y el rol (cliente u operador).
2	El usuario selecciona dentro de una lista desplegable las variables: tipo de documento y rol y digita el número de identificación. Después de esto hace clic en el botón “aceptar”.	Con los datos capturados verifica en la base de datos si existe un usuario con esa información. A partir de esa información, si encuentra usuario con esos datos despliega el caso de excepción #1. Si no lo encuentra,

		despliega los campos de registro de: nombre, apellido, vínculo y e-mail.
3	El usuario selecciona de una lista el tipo de vínculo, y diligencia los datos: nombre, apellido y email.	<p>Valida que los datos no sean nulos, el nombre y el apellido solo contengan caracteres alfabéticos y que el e-mail tenga un formato válido de correo (ejemplos: <a href="mailto:alguien@ejemplo.com">alguien@ejemplo.com</a>).</p> <p>Si no se cumplen estas validaciones se envía el caso de excepción #2.</p>
4		Se inserta el registro en la base de datos en la entidad personas. Envía un mensaje diciendo usuario registrado. Si ocurre un error en la base de datos se envía al caso de excepción #3.
5	Se abre un panel con el mensaje del sistema y la persona hace clic en “aceptar”.	
Post-condiciones principales del caso de uso		
El usuario se registra en la base de datos.		
Caminos de Excepción		
<p>Caso 1: el sistema envía un mensaje diciendo “ya existe un usuario con esos datos” y finaliza el caso de uso.</p> <p>Caso 2: el mensaje envía un mensaje diciendo “los datos diligenciados son inválidos” y regresa al punto #3 del flujo.</p> <p>Caso 3: El sistema envía un mensaje diciendo “no se pudieron registrar los datos” y finaliza el caso de uso.</p>		

Descripción General del Caso de Uso
<p><b>CU02: Registrar propuesta de alojamiento.</b></p> <p>Registrar en la base de datos una propuesta que puede ser una vivienda express, un apartamento, una vivienda universitaria, una habitación o un inmueble hotelero. En este caso de uso se modela la funcionalidad RF2.</p>
Entidades Involucradas
<p>PROPUESTAS</p> <p>HOTELES</p> <p>HOSTELES</p> <p>SERVICIOS_BASICOS_HOTELEROS</p> <p>VIVENDAS_EXPRESS</p> <p>APARTAMENTOS</p> <p>VIVIENDAS_UNIVERSITARIAS</p> <p>HABITACIONES</p>

SERVICIOS_BASICOS		
TIPOS		
Precondiciones		
El usuario que va a realizar la propuesta debe ser un usuario de tipo operador.		
Flujo normal de Eventos		
	Usuario	Sistema
1	Selecciona la opción de “ingresar” y selecciona una opción que diga cédula o nit e ingresa el número correspondiente.	Verifica que exista en la base de datos y asimismo valida que este registrado como operador. Si no lo encuentra envía el caso de excepción #1.
2	Se abre un panel donde el usuario escoge si va a ofrecer una vivienda de tipo “hogar” o de tipo “inmueble hotelero”.	Si la opción elegida es “hogar” se muestra una lista con las opciones: habitación, vivienda express, apartamento y vivienda universitaria. Si no, se ofrece una lista con las opciones: hotel y hostel.
3	El usuario selecciona dentro de las listas desplegables la opción que desea ofrecer de alojamiento: habitación, hotel, hostel, vivienda express, apartamento, vivienda universitaria.	Si el usuario desea ofrecer una habitación ofrece los campos: si es precio, el tipo de habitación. Si el usuario desea ofrecer una vivienda express ofrece los campos: número de habitaciones, el menaje y la ubicación. Si el usuario desea ofrecer un apartamento se diligencian los campos: si es amoblado y el costo de la administración. Si el usuario desea ofrecer una vivienda universitaria entonces se diligencian los datos de: ubicación, capacidad, menaje, tipo de habitación, si es mensual, y una descripción. Además también debe seleccionar los servicios básicos que ofrece el usuario con su respectivo costo.
3		Si el usuario desea ofrecer un hotel debe envía una ventana donde deberá diligenciar: el registro a la superintendencia, el registro a la cámara de comercio, el tipo de habitación, si tiene un horario de administración de 24 horas, el tamaño, la ubicación, y la cantidad de noches. Si el usuario desea ofrecer un hostel, se abre una ventana donde se debe diligenciar: el registro a la cámara de comercio, el registro a la superintendencia, el tipo de habitación, el horario de administración inicial, el horario de administración final, el tamaño, la ubicación y la cantidad de

		noches. Además de estos se mostrará una lista de servicios hoteleros que el usuario debe seleccionar con su respectivo costo.
4	En la ventana que se muestra se llenan los campos requeridos y hace clic en “aceptar”.	Verifica que no hayan campos nulos (a menos que sea un costo de un servicio que no se vaya a ofrecer), de ser así se envía al caso de excepción #2. Verifica los campos que son numéricos, si alguno lo incumple se manda al caso de excepción #3.
5		Inserta el registro en la tabla de propuesta, y el hospedaje en la tabla que corresponde según el tipo de alojamiento que desea hacer el usuario y el servicio que desea ofrecer. Si ocurre un error en la base de datos se envía al caso de excepción #4.

#### Post-condiciones principales del caso de uso

Se registra en la base de datos la propuesta que el usuario diligencia

#### Caminos de Excepción

Caso 1: el sistema envía un mensaje diciendo “no existe un operador registrado con ese identificador” y finaliza el caso de uso.

Caso 2: el sistema envía un mensaje diciendo “información incompleta, por favor revise la información que está diligenciando” y regresa al punto #4 del flujo.

Caso 3: el sistema envía un mensaje diciendo “información inválida, por favor revise los datos que está insertando” y regresa al punto #4 del flujo.

Caso 4: El sistema envía un mensaje diciendo “no se pudieron registrar los datos” y finaliza el caso de uso.

#### Descripción General del Caso de Uso

##### **CU03: Registrar una reserva.**

Registrar en la base de datos una reserva de algún alojamiento. En este caso de uso se modelará la funcionalidad RF4.

#### Entidades Involucradas

RESERVAS  
 PROPUESTAS  
 HOTELES  
 HOSTELES  
 SERVICIOS\_BASICOS\_HOTELEROS  
 VIVENDAS\_EXPRESS  
 APARTAMENTOS

VIVIENDAS_UNIVERSITARIAS HABITACIONES SERVICIOS_BASICOS TIPOS		
Precondiciones		
El usuario que va a realizar la propuesta debe ser un usuario de tipo cliente.		
Flujo normal de Eventos		
	<b>Usuario</b>	<b>Sistema</b>
1	Selecciona la opción “tipo de identificador” y digitar el número.	Verifica que el usuario exista y que esté registrado como cliente. De lo contrario se enviará al caso de excepción #1.
2	Selecciona el tipo de hospedaje que desea: hogar o inmueble hotelero. Después hace clic en “aceptar”.	Si se escoge la opción “hogar” se despliega una lista con los tipos de vivienda que se puede tomar: habitación, vivienda express, apartamento y vivienda universitaria. Si no, se ofrece una lista con las opciones: hotel y hostel.
3	El usuario selecciona dentro de las listas desplegables la opción que desea ofrecer de alojamiento: habitación, hotel, hostel, vivienda express, apartamento, vivienda universitaria.	Dependiendo la opción escogida, se accede a la base de datos y selecciona las propuestas que cumplan con el parámetro. Después se muestra una lista con todos estos campos para que el cliente elija según su preferencia.
4	Elige entre las opciones disponibles que se muestran.	Verifica que la propuesta no esté en estado de “en retiro”, de lo contrario se enviará el caso de excepción #5. Muestra un panel pidiendo la diligencia de cuando se quiere realizar la estadía y la duración del contrato.
5	Diligencia los datos de estadía y duración del contrato	Valida que los datos insertados sean válidos, de lo contrario, se manda el caso de excepción #2. Valida que no haya un registro en esa fecha y ese hospedaje, de lo contrario se enviara al caso #4 de excepción. Se calcula el costo total de la reserva y se informa al cliente.
6	Revisa el costo y decide si quiere continuar con el servicio presionando el botón “aceptar”.	Se registra la reserva en la base de datos. Si ocurre un error en la base de datos se envía al caso de excepción #3
Post-condiciones principales del caso de uso		
El registro de la reserva en la base de datos.		
Caminos de Excepción		

Caso 1: el sistema envía un mensaje diciendo “no existe un cliente registrado con estos datos” y finaliza el caso de uso.
Caso 2: el sistema envía un mensaje diciendo “información inválida, por favor revise los datos diligenciados” y regresa al punto #5 del flujo.
Caso 3: el sistema envía un mensaje diciendo “no se pudieron registrar los datos” y finaliza el caso de uso.
Caso 4: el sistema envía un mensaje diciendo “ya existe una reserva en esa fecha” y regresa al punto #5 del flujo.
Caso 5: el sistema envía un mensaje diciendo “la propuesta no está disponible para más reservas, por favor elija otra” y se regresa al paso #4 del flujo.

Descripción General del Caso de Uso		
<b>CU04: Cancelar reserva.</b>		
Actualiza la base de datos informando que una reserva se cancela y la penalización que esta trae. Este caso implementa la funcionalidad RF5.		
Entidades Involucradas		
RESERVAS		
Precondiciones		
El usuario debe ser un cliente registrado. Debe haber una reserva registrada.		
Flujo normal de Eventos		
	Usuario	Sistema
1	Selecciona la opción “tipo de identificador” y digitar el número.	Verifica que el usuario exista y que esté registrado como cliente. De lo contrario se enviará al caso de excepción #1.
2	Selecciona la opción “cancelar reserva”.	Muestra un panel con las reservas que haya hecho el usuario hasta el momento. Si no existe una reserva se enviará al caso de excepción #2.
3	Elige la reserva que desea cancelar.	<p>Verifica la fecha del sistema y la compara con las fechas de reserva del alojamiento y la fecha de plazo máximo de cancelación.</p> <p>Si la fecha del sistema es antes que la fecha máxima de cancelación se reporta la multa del 10% en la base de datos y se informa al cliente. Si la fecha está entre la fecha máxima de cancelación y el primer día de estadía, se reporta la multa del 30% y se informa al cliente. Y si la fecha es mayor a la fecha de estadía, se reporta la multa de 50% y se informa al cliente.</p>

6	El cliente procede con la cancelación oprimiendo el botón “aceptar”	Se actualiza la base de datos. Si ocurre un error en la base de datos se envía al caso de excepción #3
Post-condiciones principales del caso de uso		
Se actualiza la tabla RESERVAS de la base de datos con la información de la cancelación.		
Caminos de Excepción		
Caso 1: el sistema envía un mensaje diciendo “no existe ningún cliente registrado con ese identificador” y finaliza el caso de uso.		
Caso 2: El sistema envía un mensaje diciendo “no hay reservas disponibles” y finaliza el caso de uso.		
Caso 3: El sistema envía un mensaje diciendo “Ha ocurrido un error en la base de datos, por favor inténtelo de nuevo más tarde” y finaliza el caso de uso.		

Descripción General del Caso de Uso		
<b>CU05: Retirar una oferta de alojamiento</b>		
Edita la base de datos eliminando la propuesta registrada correspondiente. Este caso modela la funcionalidad RF6.		
Entidades Involucradas		
PROPUESTAS RESERVAS HOTELES HOSTELES SERVICIOS_BASICOS_HOTELEROS VIVENDAS_EXPRESS APARTAMENTOS VIVIENDAS_UNIVERSITARIAS HABITACIONES SERVICIOS_BASICOS TIPOS		
Precondiciones		
El usuario debe estar registrado como operador. Debe existir una propuesta del usuario registrada en la base de datos.		
Flujo normal de Eventos		
	Usuario	Sistema
1	Selecciona la opción de “ingresar” y selecciona una opción que diga cédula o nit e ingresa el número correspondiente.	Verifica que exista en la base de datos y asimismo valida que este registrado como operador. Si no lo encuentra envía el caso de excepción #1.



2	Selecciona la opción “retirar propuesta”.	Abre un panel donde se despliegue la lista de las propuestas que haya registrado el usuario. Si no existe ninguna propuesta se envía al caso de excepción #2.
3	Escoge en la lista la propuesta que desea retirar.	Valida que no hayan reservas vigentes. De lo contrario se actualiza la tabla de propuestas al estado “en retiro”. Si no hay reservas vigentes se elimina el registro en la base de datos.
Post-condiciones principales del caso de uso		
Se actualiza la tabla de PROPUESTAS, o en su defecto se elimina toda la propuesta.		
Caminos de Excepción		
Caso 1: el sistema envía un mensaje diciendo “no existe un operador registrado con ese identificador” y finaliza el caso de uso.		

Tabla 1. Casos de Uso de los requerimientos funcionales.

### 3 Diseño de la base de datos

#### 3.1 Análisis del modelo de datos relacional

A continuación se muestra el modelo de datos relacional que se propuso para la elaboración de la presente iteración.

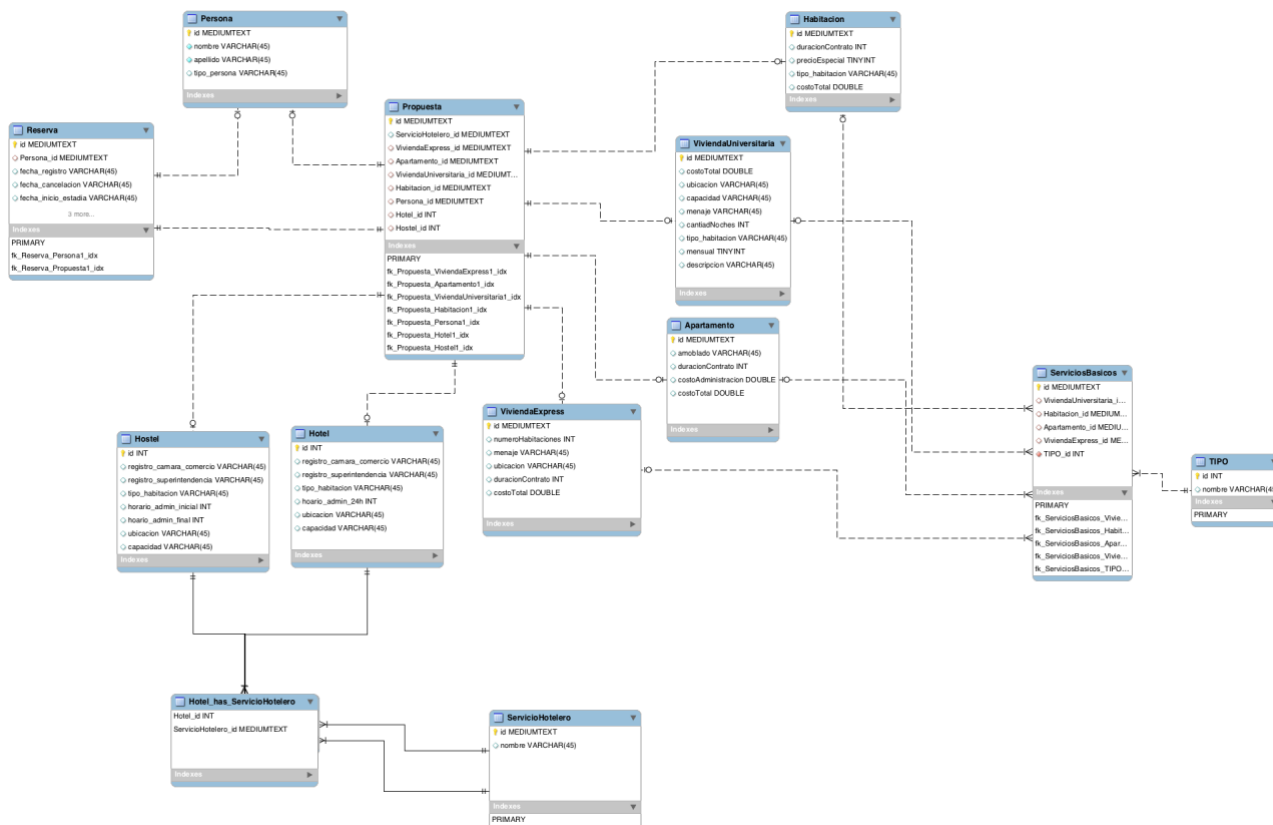


Figura 2. Modelo de datos relacional del caso de estudio.



la integridad de acuerdo con las restricciones de chequeo. Las anteriores sentencias se pueden encontrar en la carpeta `docs` de la iteración adjunta. Adicionalmente se realizaron pruebas de los servicios `GET` `POST` `PUT` y `DELETE` utilizando la herramienta de *Postman*. Las anteriores pruebas realizadas en *Postman* dan como resultado: todos los usuarios (clientes u operadores) registrados en la base de datos, todas las reservas realizadas y todas las propuestas registradas.

Para mantener la privacidad de cada usuario, se implementaron métodos que le muestran a un usuario dado sus reservas o propuestas.

Adicionalmente, se cumplieron con los `REQUERIMIENTOS FUNCIONALES` que se encargaban de registrar a los usuarios, a las reservas y a las propuestas.

Para poder realizar los `REQUERIMIENTOS FUNCIONALES DE CONSULTA` se implementaron las clases en java `POPULARES` y `DINERO_OPERADOR` para que el resultado en *Postman* de los anteriores requerimientos fuera un objeto entendible.

## 5 Resultados obtenidos

En el proyecto, después de haber realizado el programa y correr el servidor, se obtuvo colecciones con el contenido que persiste cada clase en la base de datos. Los métodos de *get* funcionan bien, y algunos de *post*. Pero los métodos de *delete* y *post* de una reserva se obtienen excepciones.

Por el lado de los requerimientos, se lograron los resultados esperados por requerimiento haciendo uso de sentencias *SQL* y de los servicios *restful* para mostrar los resultados en formas de objetos.

Finalmente, al intentar persistir una *Reserva* se obtiene la siguiente excepción de *SQL*: `ERROR: "Falta el parámetro IN o OUT en el índice:: 1"`. Luego se hizo búsqueda sobre el significado del anterior error en distintas páginas como *StackOverflow* no se logró encontrar el error a tiempo. Lo anterior deja como enseñanza que hay que escribir las sentencias en *SQL* y el nombre de las tablas o de las columnas de la misma forma, teniendo en cuenta las mayúsculas y minúsculas, tanto en la declaración de las tablas en *SQL Developer*, como en el código implementado en *Java*.