“Iteración 3: Alohandes”

José D. Flórez Ruiz, Carlos M. Muñoz Almeida
Iteración 3 - Documento
Universidad de los Andes, Bogotá, Colombia
{jd.florezr1, c.munoza}@uniandes.edu.co
Fecha de presentación: Mayo 07 de 2023

Tabla de contenido

1 Introducción

En el presente documento, se ofrece un análisis sobre el impacto de la inclusión de nuevos requerimientos y cambios en la aplicación de Alohandes, un sistema dedicado a la publicación y reserva de alojamientos para miembros de una comunidad universitaria. También, se incluyen las decisiones de negocio y diseño tomadas para la implementación de nuevas funcionalidades, además de la documentación de pruebas necesarias para dar muestra de que el sistema transaccional cumple las condiciones ACID.

2 Diseño de la aplicación

2.1 Impacto de nuevos requerimientos

Para los nuevos requerimientos funcionales y de consulta se tuvieron que realizar los

siguientes cambios:

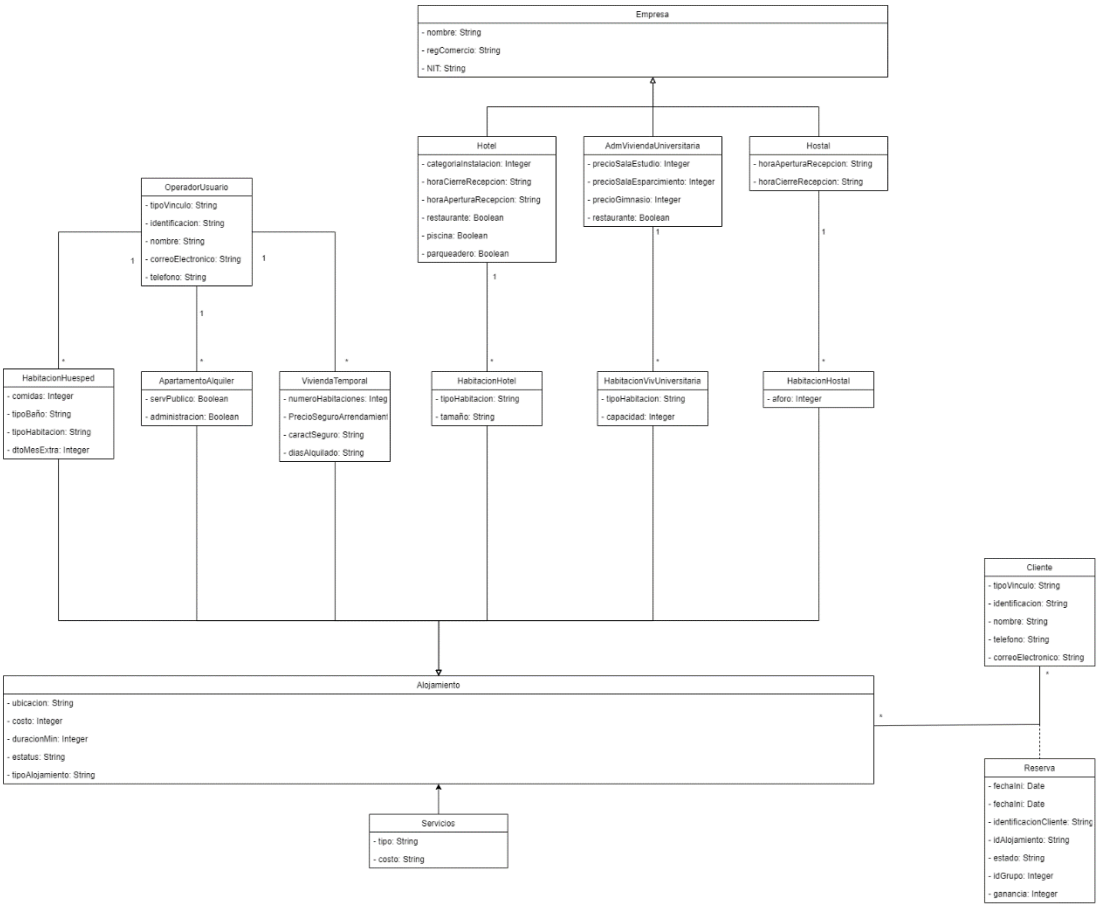
- Nuevos atributos para `Reserva`, como lo son:
  - `Estado`: El cual puede ser Y o N dependiendo de si la `Reserva` esta activa o inactiva respectivamente.
  - `IdGrupo`: El cual sirve para asociar reservas cuando estas se hagan de forma grupal.
  - `Ganancia`: Que representa lo que pago el cliente por esa `Reserva`.

- Nuevos atributos para `Alojamiento`:
  - `Estatus`: Que puede tomar valores como Y o N, habilitado o deshabilitado, respectivamente.
  - `TipoAlojamiento`: Que representa el tipo de alojamiento, este cambio resulta muy útil a la hora de realizar consultas.

Claramente, cada uno de estos cambios implico modificaciones en el modelo conceptual, relacional y de la aplicación.

2.2 Modelo conceptual

Se agregaron los nuevos atributos mencionados con anterioridad a su respectiva clase, como se muestra a continuación:



2.3 Modelo relacional

Se modificaron las tablas Alojamiento y Reserva, como resultado final se tiene:

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', 'baniera', 'sala', 'yacuzzi', 'menaje', 'luz', 'telefono', 'cocina', 'agua')
CK8	(HabitacionHuesped', 'ApartamentoAlquiler', 'ViviendaTemporal', 'HabitacionHotel', 'HabitacionViviendaUniversitaria', 'HabitacionHostal')

OperadorUsuario

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

HabitacionHuesped

idAlojamiento	comidas	tipoBaño	tipoHabitacion	dtoMesExtra	IDENTIFICACIONOPERADORUSUARIO
PK, FK Alojamiento.id	NN, CK >= 0	NN, CK2	NN, CK3	NN, 100 >= CK > 0	NN, FK OperadorUsuario.identificacion

ApartamentoAlquiler

idAlojamiento	servPublico	administracion	IDENTIFICACIONOPERADORUSUARIO
PK, FK Alojamiento.id	NN, CK1	NN, CK1	NN, FK OperadorUsuario.identificacion

Restricciones : El apartamento solo podra ser alquilado por un propietario que tenga algun tipo de vinculo con la comunidad uniandina.

ViviendaTemporal

idAlojamiento	numeroHabitaciones	PRECIOSEGUROARRENDAMIENTO	caracSeguro	diasAlquilado	propietario
PK, FK Alojamiento.id	NN, CK > 0	NN, CK > 0	NN	NN, CK <= 30	NN, FK OperadorUsuario.identificacion

Cliente

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

Reserva

id	fechaIni	fechaFin	identificacionCliente	idAlojamiento	idAlojamiento	Estado	IdGrupo	Ganancia
PK, SA	NN	NN	NN, FKCliente.Identificacion	NN, FKAlojamiento.id	NN, FKAlojamiento.id	NN, CK1	NN	NN

Alojamiento

id	ubicacion	duracionMin	Costo	Estatus	TipoAlojamiento
PK, SA	NN,	NN, CK>0	NN, CK> 0	NN, CK1	NN, CK8

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)

2.4 Análisis Calidad Modelo

Nivel de normalización

Para lograr determinar la forma normal en la que se encuentra nuestro modelo, realizaremos un recorrido por las diferentes normas y evaluaremos si nuestro modelo la cumple.

1FN: Según la bibliografía del curso, más específicamente, según Jan L. Harrington en "Relational Database Design and Implementation" especifica que un modelo que se encuentra en 1FN cumple con que no hay grupos repetidos, o dicho de otra forma sus dominios son

atómicos. Así pues, en nuestro modelo, cada uno de los atributos de cada entidad cumple con tal requisito.

2FN: Contando con la bibliografía ya mencionada, para cumplir con 2FN, se debe estar en 1FN, que ya lo verificamos anteriormente, y todos los atributos que no son llaves (o sea atributos no primos) deben ser funcionalmente dependientes de todos los atributos que conformen la llave primaria y no solo de una parte de ella.

Así pues, validando en nuestro modelo, encontramos que cada relación cuenta con una llave primaria compuesta por solamente un atributo. Por tal motivo, podemos afirmar que el resto de los atributos, que son no llaves, dependen funcionalmente la llave primaria completa.

3FN: En lo que respecta a esta forma, se debe cumplir con la 2FN y con que ningún atributo es determinado por otro atributo no primo. Así pues, revisando cada relación de nuestro modelo, solamente las llaves primarias son capaces de determinar a los demás atributos, ningún atributo que no es llave primaria es capaz de determinar a otro.

Boyce-Codd FN: Teniendo en cuenta la bibliografía sugerida llegamos a la definición de la forma normal de Boyce-Codd la cual dice: "para cada función de dependencia ($x \rightarrow y$), x debe ser una super llave", además tiene que ya ser de la forma normal 3. Teniendo en cuenta la definición anteriormente dada, podemos llegar a la conclusión de que nuestro modelo relacional cumple con ella en todas sus tablas y por ende este llega a la forma normal de Boyce-Codd.

De esta forma, y teniendo en cuenta la bibliografía ya mencionada, cuando un modelo se encuentra en FNBC es porque ha superado la gran mayoría de anomalías, y cuenta con buenas bases de diseño. Por tanto, ya no continuaremos con la revisión de las siguientes FN, y

concluiremos en que nuestro modelo se encuentra en FNBC.

2.5 Listado de tablas

Se adjunta la consulta SQL que permite consultar el listado de tablas con el nombre de la tabla, el nombre y el tipo de dato de sus campos, así como los nombres de restricciones de llaves primarias, llaves foráneas y de chequeo.

```
SELECT COLS.TABLE_NAME AS NOMBRE_TABLA, COLS.COLUMN_NAME AS NOMBRE_COLUMN, COLS.DATA_TYPE AS TIPO_DATO, CONS.CONSTRAINT_NAME AS NOM_RESTRICC
FROM ALL_TAB_COLUMNS COLS LEFT JOIN ALL_CONS_COLUMNS CONS_COLS
    ON COLS.TABLE_NAME = CONS_COLS.TABLE_NAME AND COLS.OWNER = CONS_COLS.OWNER AND COLS.COLUMN_NAME = CONS_COLS.COLUMN_NAME
    LEFT JOIN ALL_CONSTRAINTS CONS ON CONS_COLS.CONSTRAINT_NAME = CONS.CONSTRAINT_NAME AND CONS_COLS.OWNER = CONS.OWNER
WHERE COLS.OWNER = 'ISIS2304C19202310'
ORDER BY COLS.TABLE_NAME, COLS.COLUMN_NAME;
```

A continuación, un ejemplo de su resultado:

	NOMBRE_TABLA	NOMBRE_COLUMN	TIPO_DATO	NOM_RESTRICC
13	A_ALOJAMIENTOSERVICIO	IDALOJAMIENTO	NUMBER	PK_EMPRESASERVICIO
14	A_ALOJAMIENTOSERVICIO	IDALOJAMIENTO	NUMBER	FK_AL_ALOJAMIENTOSERVICIO
15	A_ALOJAMIENTOSERVICIO	IDSERVICIO	NUMBER	PK_EMPRESASERVICIO
16	A_ALOJAMIENTOSERVICIO	IDSERVICIO	NUMBER	FK_SERV_ALOJAMIENTOSERVICIO
17	A_APARTAMENTOALQUILER	ADMINISTRACION	VARCHAR2	CK_APARTAMENTOALQUILER_ADMINISTRACION

3. Reglas de negocio y de diseño

Para la implementación de los nuevos requerimientos de diseño se tomaron las siguientes decisiones de negocio y deseño:

- Al momento de agregar un nuevo Alojamiento, este ingresa con el valor Y, de habilitado a la base de datos.
- Cuando una reserva no se pueda relocalizar, su estado parara a N, lo que quiere decir que fue cancelada.
- El saldo de un cliente inicia en 0, al momento de que este realice una reserva, se le descontara del salo y este quedara en negativo, cuanto el cliente tenga su saldo en 0 quiere decir que esta al día con el pago de las reservas en aplicación.
- El uso de Alohandes para cada tipo de usuario se mide según el número de reservas que en total han realizado todos.
- Para analizar la operación de Alohandes, se recibe la unidad de tiempo, y según ello se muestra en esa unidad de tiempo cuáles fueron las fechas de mayor demanda (mayor cantidad de alojamientos ocupados), las de mayores ingresos (mayor cantidad de dinero recibido) y las de menor ocupación.
- Si se requiere relocalizar una reserva, se relocalizará a un alojamiento que cumpla con estar disponible en el rango de fechas de la reserva y que sea del mismo tipo del que fue

deshabilitado.
- Si un alojamiento se encuentra deshabilitado, no será posible reservarlo.

Además, se deben tener en cuenta las siguientes reglas mencionadas en anteriores documentos:

- 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 deben tener una reserva mínima de 30 días.

4. Implementación y lógica de los requerimientos

4.1 Requerimientos funcionales:

- **RF7 - REGISTRAR RESERVA COLECTIVA:**

Para este requerimiento, era necesario realizar una variación del requerimiento funcional 4 de la iteración pasada, en el cual debíamos realizar una reserva. El reto en esta iteración era lograr realizar múltiples reservas con la misma consulta. Para ello, decidimos hacer una interfaz en el cual se le pida al usuario ciertas características que busca en cada uno de los apartamentos que se van a reservar, después realizamos una consulta, verificamos que la cantidad de alojamientos solicitados estén disponibles con las características solicitadas y por último si esto se cumple se realiza la reserva de cada uno de ellos apalancándonos del RF4 que como ya mencionado ya habíamos realizado en la iteración pasada. Sin embargo, para que este requerimiento funcional funcionara de manera correcta, decidimos agregar una nueva columna en la tabla reserva, el cual se encargara de conservar el id de los grupos de reservas; con esto solo tenemos que buscar el id del grupo que queramos y el acceso al será muy simple. Cabe resaltar que todo este requerimiento funciona de manera transaccional, por lo cual, si en algún punto alguna querie llega a fallar, ningún cambio se realizara. También es importante saber que, si no hay la cantidad de alojamientos solicitada por el usuario, ningún alojamiento será reservado y el usuario será notificado de ello.

- **RF8 - CANCELAR RESERVA COLECTIVA:**

En este requerimiento se quiere realizar la operación opuesta a la anterior, cancelando un grupo de reservas. Para esto utilizamos el id de grupo, realizado una consulta para todas las reservas que

tengan el id de grupo solicitado. Ya teniendo las reservas que tenemos que cancelar hacemos uso del requerimiento funcional 5 realizado en la iteración 2, cancelando cada una de las reservas seleccionadas.

- **RF9 - DESHABILITAR OFERTA DE ALOJAMIENTO:**

  Para este requerimiento, se recibe por entrada el id del `alojamiento` a deshabilitar, además, el usuario debe seleccionar el tipo de `alojamiento` que desea deshabilitar, esto se hace para cumplir con el requisito no funcional de seguridad.

  Una vez se recibe el id del `alojamiento`, se procede a deshabilitarlo, es decir, se actualiza el atributo `Estatus` a `N`, luego se realiza una búsqueda de las `reservas` que existentes sobre ese alojamiento para que puedan ser relocalizadas.

  Para cada una de las `reservas` encontradas se consulta un `alojamiento` que cumpla con ser del mismo tipo que del deshabilitado y que esté disponible en el mismo rango de fechas de la reserva.

  Si se encuentra un `alojamiento` disponible, se actualiza la `reserva` con el nuevo `alojamiento`, es decir, se relocaliza, y se le notifica al usuario. Si no se encontró ningún alojamiento disponible, se procede a cancelar la reserva y a notificarle al usuario.

- **RF10 - REHABILITAR OFERTA DE ALOJAMIENTO:**

  En este requerimiento se recibe el id del `alojamiento` y se actualiza su atributo `Estatus` a `Y`, lo cual quiere decir que fue rehabilitado, una vez es rehabilitado, se permite que sea mostrado disponible para ser reservado.

4.2 Requerimientos de consulta:

- **RFC5 - MOSTRAR EL USO DE ALOHANDES PARA CADA TIPO DE USUARIO DE LA COMUNIDAD:**

  En esta consulta para cada tipo de usuario en Alohandes se muestra su uso. El uso se mide en la cantidad de reservas que en total ha realizado cada tipo de usuario. Dentro de los tipos de usuarios contemplados se encuentra: profesor, empleado, egresado, estudiante, padre de estudiante, profesor invitado, persona evento uniandes. Cabe resaltar que para este requerimiento fue de gran utilidad el cambio en `Alojamiento`, específicamente el atributo agregado `tipoAlojamiento`.

- **RFC6 - MOSTRAR EL USO DE ALOHANDES PARA UN USUARIO DADO:**

En este requerimiento se recibe la identificación del cliente. Para su implementación, fue de gran ayuda el cambio realizado en `Reservas`, específicamente el atributo de `ganancia`, dado que, por medio de ellos, se puede saber para un cliente dado el dinero que pagó por reservar ese alojamiento.

- **RFC7 - ANALIZAR LA OPERACIÓN DE ALOHANDES**

  Para este requerimiento se recibe como entrada la unidad del tiempo (mes o semana, entre otros) y la cantidad de esa unidad (1,2,3…), también, el cliente tiene que escoger el tipo de alojamiento.

  Luego, dependiendo de la unidad del tiempo elegida se muestra el año y la fecha de mayor demanda (mayor cantidad de alojamientos ocupados), la de mayores ingresos (mayor cantidad de dinero recibido) y la de menor ocupación.

  Para realizar este requerimiento se realizó una consulta SQL la cual involucra un ranking de la cantidad de reservas, el atributo ganancia en reserva y a partir de esto, se obtiene el mínimo y máximo para poder obtener las fechas en las que hubo mayor demanda, mayores ingresos y menor ocupación.

- **RFC8 - ENCONTRAR LOS CLIENTES FRECUENTES**

  Para este requerimiento no se recibe como entrada el id del alojamiento del cual se quiere consultar los clientes más frecuentes. Como descrito en el contexto dado, para que un cliente se considere frecuente tuvo que haber reservado o tener reservas futuras más de tres o también se puede considerar cliente frecuente si la suma de las reservas da más o igual cantidad de 15 días. Para realizar este querie de SQL lo que realizamos fue hacer dos subqueries cada una de ellas atacando cada uno de los casos y al final filtramos para el id de alojamiento que queremos y agrupamos a los clientes; dándonos así, los clientes que cumplen alguno de los dos requerimientos ya mencionados.

- **RFC9 - ENCONTRAR LAS OFERTAS DE ALOJAMIENTO QUE NO TIENEN MUCHA DEMANDA**

  En este requerimiento no es necesario solicitar ningún elemento como entrada, en este buscábamos encontrar los alojamientos con poca demanda – Definiendo poca demanda como todos alojamientos que no han sido reservados desde el día actual hasta treinta días hacia atrás - incluyendo las reglas de negocio de nuestro diseño que serían: primero, que las reservas canceladas no cuentan y que el alojamiento tiene que estar en estado activo al momento de la consulta. Para lograr esto, realizamos un filtro en el cual filtrábamos a todas las reservas que tuvieran una reserva en el mes anterior y después filtramos que el estado de esas reservas debía estar activas y que el alojamiento también estuviese activo.

4.3 Mecanismos para garantizar las propiedades ACID

Con el objetivo de garantizar las propiedades ACID en nuestro sistema transaccional, para cada requerimiento funcional que necesite de inserción, borrado o actualización, se valida que los datos que ingresen cumplan con las normas de negocio y tengan sentido, así garantizamos la

integridad. La durabilidad, se cumple al momento de que cada vez que el cliente realice una operación, aun así, se de forma concurrente, se realiza commit de tales cambios. Si no se pudo realizar la transacción por cualquier motivo, entramos a cubrir el aspecto de la consistencia, dado que se maneja la excepción y se realiza rollback hasta el último estado consistente de la base de datos. De esa forma, nuestra base de datos se mantiene consistente antes y después de cada transacción. Ahora bien, en lo que respecta a la atomicidad, se valida que la transacción se haga o no se haga, si la transacción inicia, pero no finaliza correctamente, como se mencionó anteriormente, se hace rollback, de manera que es como si no se hubiese hecho. Si esta inicia y acaba correctamente, se realiza commit de los cambios y entonces se hizo toda la transacción.

5. Pruebas y resultados

Pruebas RF7 - Pruebas de transacción exitosa			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('2','oeste','1','269','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('2','suite','grande','9521102');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Servicios: null | Fecha Inicio: 01/12/2023 | Fecha Fin: 20/12/2023 | Cantidad de alojamientos: 2 | Tipo de alojamiento: HabitacionHotel	Ok
c	El estado final de la base de datos	N/A	Se realizo la reserva colectiva de 2 alojamientos con id de grupo : 61; los alojamientos reservados fueron: #1 - id: 42 - precio: 2 #2 - id: 46 - precio: 6

Pruebas RF7 - Pruebas de transacción NO exitosa | Alojamientos insuficientes			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('2','oeste','1','269','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('2','suite','grande','9521102');	row inserted.
b	Los datos involucrados en la operación transaccional solicitada	Servicios: null | Fecha Inicio: 01/12/2023 | Fecha Fin: 20/12/2023 | Cantidad de alojamientos: 3 | Tipo de alojamiento: HabitacionHotel	Ok
c	El estado final de la base de datos	N/A	No fue posible realizar la reserva colectiva con id: 68 dado que no se encontraron alojamientos disponibles

Pruebas RF8 - Pruebas de transacción exitosa			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('2','oeste','1','269','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('2','suite','grande','9521102'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('1',to_date('12/01/23','DD/MM/RR'),to_date('12/06/23','DD/MM/RR'),'0402842626','1', 'Y', '444', '100'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('2',to_date('20/02/23','DD/MM/RR'),to_date('05/08/23','DD/MM/RR'),'1453890033','2', 'Y', '444', '100');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Grupal: 444	Ok
c	El estado final de la base de datos	N/A	Se cancelo la reserva colectiva con id: 444; las reservas canceladas fueron: #1 - id reserva individual: 1 - id de alojamiento: 1 #2 - id reserva individual: 2 - id de alojamiento: 2

Pruebas RF8 - Pruebas de transacción NO exitosa | Id Grupal No existente			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('2','oeste','1','269','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('2','suite','grande','9521102'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('1',to_date('12/01/23','DD/MM/RR'),to_date('12/06/23','DD/MM/RR'),'0402842626','1', 'Y', '444', '100'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('2',to_date('20/02/23','DD/MM/RR'),to_date('05/08/23','DD/MM/RR'),'1453890033','2', 'Y', '444', '100');	row inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Grupal: 443	Ok
c	El estado final de la base de datos	N/A	Se cancelo la reserva colectiva con id: 324; las reservas canceladas fueron:

Pruebas RF9 - Pruebas de transacción exitosa			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('1',to_date('12/01/23','DD/MM/RR'),to_date('12/06/23','DD/MM/RR'),'0402842626','1', 'Y', '444', '100'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Alojamiento: 1	Ok
c	El estado final de la base de datos	N/A	Se ha deshabilitado 1 alojamiento Se han encontrado 3 reservas asociadas al alojamiento Se relocalizo 1 Reserva [id=53, fechaIni=2023-01-04 00:00:00.0, fechaFin=2023-05-05 00:00:00.0, identificacionCliente=1, idAlojamiento=45] con id 53 al alojamiento con id 42 Se relocalizo 1 Reserva [id=8768, fechaIni=2023-05-04 00:00:00.0, fechaFin=2023-05-07 00:00:00.0, identificacionCliente=1, idAlojamiento=45] con id 8768 al alojamiento con id 44 Se relocalizo 1 Reserva [id=67, fechaIni=2023-07-01 00:00:00.0, fechaFin=2023-07-20 00:00:00.0, identificacionCliente=1, idAlojamiento=45] con id 67 al alojamiento con id 44

Pruebas RF9 - Pruebas de transacción NO exitosa | Id alojamiento no existente			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_RESERVA (ID,FECHAINI,FECHAFIN,IDENTIFICACIONCLIENTE,IDALOJAMIENTO, ESTADO, IDGRUPO, GANANCIA) values ('1',to_date('12/01/23','DD/MM/RR'),to_date('12/06/23','DD/MM/RR'),'0402842626','1', 'Y', '444', '100'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','Y','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Alojamiento: 5	Ok
c	El estado final de la base de datos	N/A	Se ha deshabilitado 0 alojamiento Se han encontrado 0 reservas asociadas al alojamiento

Pruebas RF10 - Pruebas de transacción exitosa			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','N','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Alojamiento: 1	Ok
c	El estado final de la base de datos	N/A	Se ha habilitado 1 alojamiento

Pruebas RF10 - Pruebas de transacción NO exitosa | Id alojamiento no existente			
#	Nombre Paso	Input	Output
a	El estado inicial de la base de datos	Insert into A_HOTEL (REGCOMERCIO,NIT,NOMBRE,RESTAURANTE,PARQUEADERO,PISCINA) values ('9422181','139270435','Kunde Inc','N','N','N'); Insert into A_ALOJAMIENTO (ID,UBICACION,DURACIONMIN,COSTO,ESTATUS,TIPOALOJAMIENTO) values ('1','oeste','1','99','N','HabitacionHotel'); Insert into A_HABITACIONHOTEL (ID,TIPOHABITACION,TAMANIO,IDHOTEL) values ('1','suite','grande','9422181');	rows inserted.
b	Los datos involucrados en la operación transaccional solicitada	Id Alojamiento: 5	Ok
c	El estado final de la base de datos	N/A	Se ha habilitado 0 alojamiento

*Para ver a más a detalle y con mejor calidad acceda al Excel que se encuentra en la carpeta ”Pruebas”