

Iteración No. 3 Sistemas Transaccionales

Juan David Serrano 201632249
Maria Alejandra Escalante 201631008

Febrero 2019

Índice

1. Modelos	2
1.1. Modelo Conceptual	2
1.2. Modelo de Datos	3
2. Diseño de la aplicación	5
2.1. Tablas en la BD	5
2.2. Nivel de Normalización	9
2.3. Cumplimiento ACID	9
3. Resultados Logrados	10
4. Resultados no logrados	10
5. Balance del plan de pruebas	10

1. Modelos

1.1. Modelo Conceptual

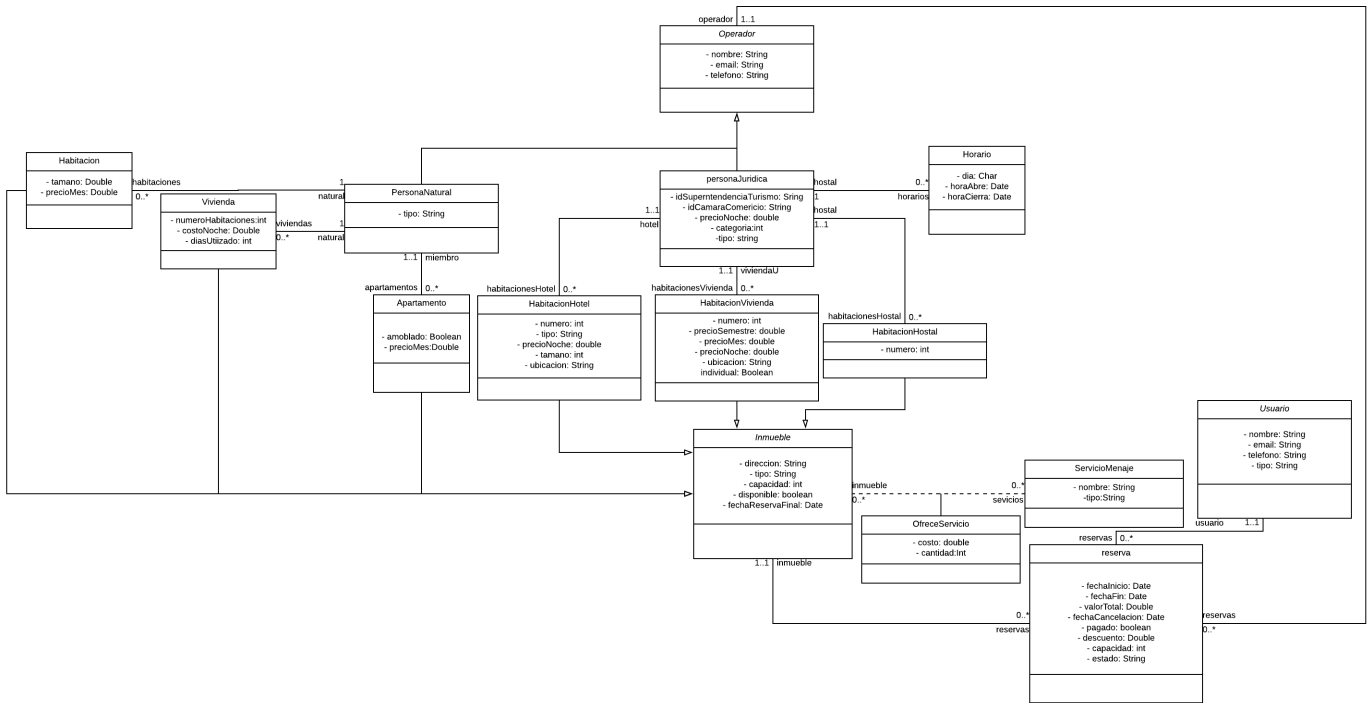


Figura 1: Modelo Conceptual

Supuestos adicionales sobre las reglas de negocio encontradas en el caso de estudio Por aplicación se revisa que se cumplan las siguientes cosas

1. Un alojamiento no acepta reservas que superen su capacidad
2. Solo se pueden arrendar inmuebles disponibles
3. Que un invitado solo pueda arrendar vivienda
4. Habitación reserva mínima es de un mes
5. En apto el contrato mínimo es de un mes
6. En vivienda verificar que los días utilizados no superen un mes
7. una persona no puede reservar mas de un alojamiento en un mismo día
8. el alojamiento de vivienda universitaria solo esta habilitado a estudiantes, profesores, empleados y profesor visitante
9. Para usos cobrados por semanas o meses, el tiempo limite está fijado en una semana.
10. Horario únicamente de hostel
11. calcular valor de reserva

12. revisar que todo menaje tiene cantidad y todo servicio tiene un costo asociado
13. precio noche es de hostel únicamente
14. categoría es de hotel únicamente

1.2. Modelo de Datos

La única modificación al esquema de la base de datos fue incorporar un atributo adicional a la tabla reserva para que fuera posible indicar si esta pertenecía o no a una reserva colectiva. Adicionalmente en el esquema en SQL se modificaron las secuencia para que existieran 4: para inmueble, para usuario, para operador, para reserva y para reserva colectiva.

operador			
id	nombre	email	telefono
SA PK	NN	NN ND	NN
NUMBER	VARCHAR2(40)	VARCHAR2(40)	VARCHAR2(15)

usuario				
id	nombre	email	telefono	tipo
SA PK	NN	NN ND	NN	NN $CK_{tipo_usuario}$
NUMBER	VARCHAR2(40)	VARCHAR2(40)	VARCHAR2(15)	VARCHAR2(40)

$tipo_usuario = [profesor, profesor_invitado, estudiante, egresado, empleado, padre_estudiante, invitado]$

persona_juridica					
id	id_superintendencia_turismo	id_camara_comercio	categoria	precio_noche	tipo
$FK_{operador.id}$ PK	NN ND	NN ND	NULL	NULL $CK_{\mathbb{R} \geq 0}$	NN $CK_{[hotel, hostel, vivienda]}$
NUMBER	NUMBER	NUMBER	NUMBER(1)	NUMBER	VARCHAR2(9)

horario			
id_hostal	dia	hora_abre	hora_cierre
PK $FK_{persona_juridica.id}$	NN PK	NN	NN
NUMBER	VARCHAR2(1)	DATE	DATE

habitacion_hotel						
id	id_hotel	numero	tipo	precio_noche	tamano	ubicacion
$FK_{inmueble.id}$ PK	$FK_{persona_juridica.id}$ NN	NN	NN CK_{tipo}	NN $CK_{\mathbb{R} \geq 0}$	NN $CK_{\mathbb{R} \geq 0}$	NN
NUMBER	NUMBER	NUMBER(5)	VARCHAR2(9)	NUMBER(10)	number(5)	VARCHAR2(40)

$tipo = [estandar, semisuite, suite]$

habitacion_vivienda							
id	id_vivienda	numero	precio_semestre	precio_mes	precio_noche	ubicacion	individual
$FK_{inmueble.id}$ PK	$FK_{persona_juridica.id}$ NN	NN $CK_{\mathbb{R} \geq 0}$	NN $CK_{\mathbb{R} \geq 0}$	NN $CK_{\mathbb{R} \geq 0}$	NN	NN $CK_{\mathbb{R} \geq 0}$	NN $CK_{[Y,N]}$
NUMBER	NUMBER	NUMBER (5)	NUMBER (60)	NUMBER (30)	NUMBER (10)	VARCHAR2 (40)	VARCHAR (1)

habitacion_hostal		
id	id_hostal	numero
$FK_{inmueble.id}$ PK	$FK_{persona_juridica.id}$ NN	NN
NUMBER	NUMBER	NUMBER(5)

inmueble					
id	direccion	tipo	capacidad	disponible	fecha_reserva_final
PK SA	NN	NN $CK_{tipo_i nmueble}$	NN	NN	NULL
NUMBER	VARCHAR2(40)	VARCHAR(12)	NUMBER(2)	VARCHAR (1)	DATE

$tipo_i nmueble = [vivienda, habitacion, apartamento, habitacion_hotel, habitacion_hostal, habitacion_vivienda]$

servicio_menaje	
nombre	tipo
PK UA	NN $CK_{[menaje, servicio]}$
VARCHAR2(40)	VARCHAR2(8)

ofrece_servicio			
id_servicio_menaje	id_inmueble	costo	cantidad
PK $FK_{servicio_menaje.nombre}$	PK $FK_{inmueble.id}$	NULL $CK_{\mathbb{R} \geq 0}$	NULL $CK_{\mathbb{N} \geq 0}$
VARCHAR2(40)	NUMBER	NUMBER(9,2)	NUMBER(2)

reserva					
id	fecha_inicio	fecha_fin	valor_total	fecha_cancelacion	pagado
SA PK	NN	NN	NN	NN SA	NN
NUMBER	DATE	DATE	NUMBER	DATE	NUMBER

Continuando la anterior tabla

reserva						
descuento	capacidad	estado	id_operador	id_usuario	id_inmueble	idColectiva
NN	NN	NN $CK_{0,1}$	$FK_{operador.id}$ NN	$FK_{usuario.id}$ NN	$FK_{inmueble.id}$ NN	NULL
NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER

persona_natural	
id	tipo
PK FK _{entidad.id}	NN
NUMBER	VARCHAR2(40)

habitacion			
id	tamano	precioMes	id_persona
PK FK _{inmueble.id}	NN	NN	FK _{persona_natural.id}
NUMBER	NUMBER	NUMBER	NUMBER

vivienda				
id	numero_habitaciones	costo_noche	dias_utilizado	id_persona
PK FK _{inmueble.id}	NN	NN	NN	FK _{persona_natural.id}
NUMBER	NUMBER	NUMBER	NUMBER	NUMBER

apartamento			
id	Amoblado	precioMes	id_persona
PK FK _{inmueble.id}	NN CK _{0,1}	NN	FK _{persona_natural.id}
NUMBER	NUMBER	NUMBER	NUMBER

2. Diseño de la aplicación

2.1. Tablas en la BD

A continuación se presenta el resultado de las consultas realizadas para conocer las tablas y su información y poder asegurar la calidad del mismo.

En la primera imagen se pueden ver las tablas y la cantidad de columna que tiene cada una en total, la cantidad de columnas que pueden ser nulas y las cantidad de columnas que son llaves foráneas a otras tablas

	NOMBRETABLA	NUMCOLUMNAS	NUMCOLSNUL	NUMCOLFKS
1	APARTAMENTO	4	0	2
2	HABITACION	4	0	2
3	HABITACIONHOSTAL	3	0	2
4	HABITACIONHOTEL	7	0	2
5	HABITACIONVIVIENDA	8	0	2
6	HORARIO	4	0	1
7	INMUEBLE	6	1	0
8	OFRECESERVICIO	4	2	2
9	OPERADOR	4	0	0
10	PERSONAJURIDICA	6	2	1
11	PERSONANATURAL	2	0	1
12	RESERVA	13	1	3
13	SERVICIOMENAJE	2	0	0
14	USUARIO	5	0	0
15	VIVIENDA	5	0	2

Figura 2: Numero de Columnas, Nulas y Foráneas

En las siguientes tablas se muestran la cantidad de columnas por tipo de dato en cada tabla y la longitud promedio de las mismas

	NOMBRETABLA	TIPODEDATOS	NUMCOLSTIPODATO	PROMEDIOLONGITUDCOL
1	APARTAMENTO	NUMBER	4	3.25
2	HABITACION	NUMBER	4	3.5
3	HABITACIONHOSTAL	NUMBER	3	3
4	HABITACIONHOTEL	NUMBER	5	3.2
5	HABITACIONHOTEL	VARCHAR2	2	14.5
6	HABITACIONVIVIENDA	NUMBER	7	3.43
7	HABITACIONVIVIENDA	VARCHAR2	1	10
8	HORARIO	NUMBER	3	3
9	HORARIO	VARCHAR2	1	2
10	INMUEBLE	DATE	1	5
11	INMUEBLE	NUMBER	3	3
12	INMUEBLE	VARCHAR2	2	17
13	OFRECESERVICIO	NUMBER	3	2.67
14	OFRECESERVICIO	VARCHAR2	1	9
15	OPERADOR	NUMBER	1	3
16	OPERADOR	VARCHAR2	3	14
17	PERSONAJURIDICA	NUMBER	5	3.4
18	PERSONAJURIDICA	VARCHAR2	1	8

Figura 3: Tipo de datos de las columnas

19	PERSONANATURAL	NUMBER	1	3
20	PERSONANATURAL	VARCHAR2	1	13
21	RESERVA	DATE	3	8
22	RESERVA	NUMBER	10	2.7
23	SERVICIOMENAJE	VARCHAR2	2	13
24	USUARIO	NUMBER	1	3
25	USUARIO	VARCHAR2	4	13.25
26	VIVIENDA	NUMBER	5	3.2

Figura 4: Tipo de datos de las columnas

En la siguiente tabla se ven las columnas que son llave primaria por tabla y el tipo de dato de las mismas

	NOMBRETABLA	NOMBRECOLSPK	TIPODEDATO
1	APARTAMENTO	ID	NUMBER
2	HABITACION	ID	NUMBER
3	HABITACIONHOSTAL	ID	NUMBER
4	HABITACIONHOTEL	ID	NUMBER
5	HABITACIONVIVIENDA	ID	NUMBER
6	HORARIO	DIA	VARCHAR2
7	HORARIO	IDHOSTAL	NUMBER
8	INMUEBLE	ID	NUMBER
9	OFRECESERVICIO	IDINMUEBLE	NUMBER
10	OFRECESERVICIO	IDSERVICIOMENAJE	VARCHAR2
11	OPERADOR	ID	NUMBER
12	PERSONAJURIDICA	ID	NUMBER
13	PERSONANATURAL	ID	NUMBER
14	RESERVA	ID	NUMBER
15	SERVICIOMENAJE	NOMBRE	VARCHAR2
16	USUARIO	ID	NUMBER
17	VIVIENDA	ID	NUMBER

Figura 5: Llaves primarias

En las siguientes imágenes se ven las columnas que son llaves foráneas por tabla y el tipo de dato de las misma.

	NOMBRETABLA	NOMBRECOLSPK	TIPODEDATO
1	APARTAMENTO	ID	NUMBER
2	APARTAMENTO	IDPERSONA	NUMBER
3	HABITACION	ID	NUMBER
4	HABITACION	IDPERSONA	NUMBER
5	HABITACIONHOSTAL	ID	NUMBER
6	HABITACIONHOSTAL	IDHOSTAL	NUMBER
7	HABITACIONHOTEL	ID	NUMBER
8	HABITACIONHOTEL	IDHOTEL	NUMBER
9	HABITACIONVIVIENDA	ID	NUMBER
10	HABITACIONVIVIENDA	IDVIVIENDA	NUMBER
11	HORARIO	IDHOSTAL	NUMBER
12	OFRECESERVICIO	IDINMUEBLE	NUMBER
13	OFRECESERVICIO	IDSERVICIOMENAJE	VARCHAR2
14	PERSONAJURIDICA	ID	NUMBER
15	PERSONANATURAL	ID	NUMBER
16	RESERVA	IDINMUEBLE	NUMBER
17	RESERVA	IDOPERADOR	NUMBER
18	RESERVA	IDUSUARIO	NUMBER

Figura 6: Llaves Foraneas

19	VIVIENDA	ID	NUMBER
20	VIVIENDA	IDPERSONA	NUMBER

Figura 7: Llaves Foraneas

2.2. Nivel de Normalización

El modelo se encuentra en primera forma normal ya que no existen nombres de atributos repetidos en la misma tabla y todos los atributos son mono valor.

El modelo se encuentra en segunda forma normal ya que en todas las tablas la llave primaria es un único atributo por lo cual no hay dependencias parciales. Las únicas tablas donde hay una llave primaria de 2 atributos es "ofrece_servicio" y "horario" pero en dichas tablas no hay dependencias parciales ya que los otros atributos los determinan la llave primaria completa.

Para estar en tercera forma normal no deben existir dependencias transitivas. Es decir que no debe haber atributos no primos que determinen otros no primos. Lo cual se cumple para nuestro modelo.

También, se cumple que para toda dependencia de A a B en las tablas, A es súper llave y no hay dependencias triviales. Por esta razón, Nuestro modelo se encuentra en BCNF.

2.3. Cumplimiento ACID

En primer lugar, la aplicación desarrollada cumple la propiedad de atomicidad ya que se siguió el siguiente esquema para todos los métodos de los requerimientos. Todos aquellos requerimientos que consistieran en solo una consulta sql se ubicaron en persistencia como una consulta simple. Por otro lado, todos los métodos que involucraran borrado, inserción, actualización o necesitaran de varias consultas SQL para desarrollarse se usó el

manejador de transacciones de las librerías de java para marcar el inicio de las transacciones con `tx.begin()` y su finalización con `tx.commit()` o `tx.rollback()` en caso de que se necesitara abortar la transacción, de esta forma se podía asegurar que las transacciones fueran atómicas. Esta fue una de las principales modificaciones que se hicieron con respecto a la iteración pasada. En segundo lugar, la propiedad de consistencia se cumple ya que las reglas de negocio se verificaban en las transacciones y la calidad de los datos con las restricciones impuestas sobre el esquema. En esta iteración se añadieron restricciones adicionales para conservar la consistencia con la realidad. Entre las nuevas restricciones estaban que la fecha de cancelación fuera antes de la fecha de inicio, que la fecha de finalización de una reserva fuera posterior a la fecha de inicio de la reserva, etc. Algunas reglas de negocio que se añadieron en esta iteración fueron que se verificaba que al crear reserva el usuario que lo hacia no tuviera reservas en esa misma época, también que un inmueble no tuviera reservas que se cruzasen, etc. La propiedad de aislamiento se cumple debido al esquema de aislamiento que tienen las transacciones. Debido a que el tamaño de la aplicación es relativamente pequeño y que el numero de consultas que se realizan en simultaneo es en su mayoría solo 1 se decidió que las transacciones manejaran un esquema de aislamiento serializable, de esta forma podemos asegurarnos que no se tendrán anomalías a la hora de desarrollar las transacciones de la aplicación. Para esto se añadió la linea código `tx.setIsolationLevel("serializable")`.^{al} inicio de cada transacción en la capa de persistencia. Por ultimo, la aplicación cumple la propiedad de durabilidad ya que esta se esta guardando en una base de datos Oracle.

3. Resultados Logrados

En el desarrollo de la interacción 3 se logro:

1. Cumplimiento de todas las reglas de negocio
2. Verificación de reglas de negocio en la aplicación
3. Todos los requisitos funcionales requeridos
4. todos los requerimientos de consulta requeridos
5. Creación de casos de prueba para todos los requisitos

4. Resultados no logrados

En el desarrollo de la iteración 3 se logro todo lo propuesto

5. Balance del plan de pruebas

Se realizaron las pruebas de todos los requerimientos en java