

PROYECTO DE CURSO – ITERACION 3

Juan Sebastián Sánchez, David Santiago Vargas
Universidad de los Andes, Bogotá, Colombia
{js.sanchezd1, ds.vargaspl}@uniandes.edu.co
Fecha de presentación: mayo 7 de 2023

Tabla de contenido

1. Introducción
2. Análisis y modelos
 - 2.1 Modelo conceptual actualizado
 - 2.2 Modelo relacional actualizado y análisis de impacto de los nuevos requerimientos
3. Diseño de la aplicación
 - 3.1 Listado con las tablas generadas en la base de datos
 - 3.2 Especificación de los nuevos requerimientos
4. Construcción de la aplicación
 - 4.1 Desarrollo y/o ajustes para cumplir con los nuevos requerimientos
 - 4.2 Cambios y desarrollo de las transacciones
 - 4.3 Cambios y desarrollo en el módulo de persistencia de la aplicación
5. Resultados de la iteración del proyecto
 - 5.1 Resultados logrados
 - 5.2 Resultados no logrados
6. Balance del plan de pruebas
 - 6.1 Pruebas de transacciones exitosas
 - 6.2 Pruebas de transacciones no exitosas
7. Bibliografías

1. Introducción

El presente documento tiene como objetivo analizar el trabajo llevado a cabo para la elaboración de la iteración 3 del proyecto del caso AlohaAndes. En primer lugar, se presenta nuevas versiones para los modelos para satisfacer los nuevos requerimientos y restricciones. Posteriormente, se describen detalladamente los requerimientos implementados y las modificaciones realizadas en el proyecto para cumplir dichos requerimientos de manera

transaccional y persistente. Tras esto, se presentan tanto los resultados logrados como los no conseguidos para hacer un balance general de la iteración 3. Finalmente, se presentan los nuevos escenarios de prueba para comprobar el correcto funcionamiento de las diferentes transacciones.

2. Análisis y modelos

2.1 Modelo conceptual actualizado

En la figura 1 se muestra el modelo de mundo actualizado para el desarrollo de los nuevos requerimientos propuestos para la iteración 3. En primer lugar, se optó por no incluir ninguna clase nueva ni tampoco eliminar alguna ya existente, debido a que no se creyó necesario para plasmar el modelo de negocio junto con sus nuevas necesidades. Los únicos cambios introducidos consistieron en agregar algunos atributos nuevos denominados ‘Estado’ y ‘Tipo’ en la clase de alojamiento. Por último, las relaciones siguen siendo exactamente las mismas.

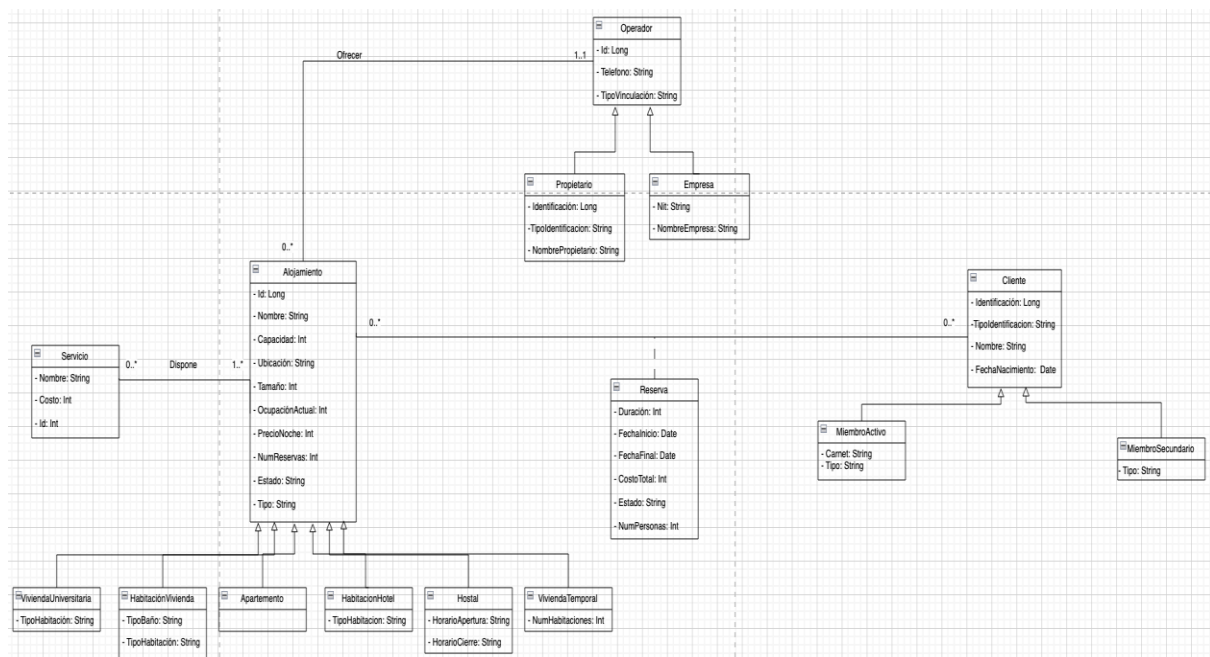


Figura 1: Diagrama UML con el modelo conceptual

2.2 Modelo relacional actualizado y análisis de impacto de los nuevos requerimientos

A continuación, se presenta el modelo relacional actualizado para ajustarse a los nuevos requerimientos de modificación propuestos para la iteración 3. En principio, los únicos cambios realizados fueron los ya descritos para el modelo conceptual, es decir, que se agregaron los atributos de ‘Estado’ y ‘Tipo’ con sus respectivas restricciones en la tabla de ‘Alojamiento’.

Adicionalmente, se introdujo un nuevo atributo denominado ‘IdReservaColectiva’ en la tabla de reservas, esto con la finalidad de facilitar la implementación tanto del registro de reservas colectivas como la posible cancelación de estas (RF7 y RF8). El hecho de tener identificadores

para las reservas colectivas permite definir un parámetro universal para identificar con mayor facilidad aquellas reservas individuales que hacen parte de una reserva colectiva.

Con respecto a los nuevos atributos de ‘Alojamiento’, el ‘estado’ se utiliza para indicar si el alojamiento dado se encuentra habilitado y deshabilitado, esto con el objetivo de implementar los requerimientos de habilitar y rehabilitar una oferta de alojamiento (RF9 y RF10). Por último, el atributo de ‘Tipo’ posibilita que se pueda resolver la consulta RF7 por medio de un ‘GROUP BY’, en lugar de tratar directamente con las subclases que heredan de ‘Alojamiento’.

Operador

IdOperador	Teléfono	TipoVinculacion
PK, SA	NN, ND	NN
5018392	3172751312	Egresado
5018393	3192451234	Estudiante

Propietario

IdOperador	Identificación	TipoIdentificación	NombrePropietario
PK, FK (Operador.IdOperador)	NN, ND	NN, CK (‘CC’, ‘TI’, ‘PA’)	NN
5018393	1123489123	CC	Juan Felipe Gómez

Empresa

IdOperador	NIT	NombreEmpresa
PK, FK (Operador.IdOperador)	NN, ND	NN
5018392	739.830.012 - 3	Hilton

Alojamiento

IdAlojamiento	Nombre	Capacidad	Ubicación	Tamaño	Precio Noche	OcupaciónActual	NumReservas	IdOperador	Estado	Tipo
PK, SA	NN	NN, CK (>0)	NN	NN, CK (>0)	NN, CK (>0)	NN, CK (>0)	NN, CK (>0)	NN, FK (Operador.IdOperador)	NN, CK (‘Habilitado’, ‘Deshabilitado’)	NN, CK1
1	HabitacionCityU	2	Centro	20	100000	0	50	5018393	Habilitado	ViviendaUni

										versita ria
2	Rosales	3	Sur	50	120000	2	70	50183 93	Desha bilitad o'	Habita cionVi vienda
3	Las fuentes	4	Sur	100	80000	2	80	50183 93	Habili tado	Aparta mento
4	Habitació n Hilton	2	Norte	20	200000	1	120	50183 93	Habili tado	Habita cionH otel
5	El faro	15	Centr o	70	45000	8	35	50183 93	Desha bilitad o'	Hostal
6	Apt 13 – Bella vista	1	Centr o	70	65000	1	22	50183 93	Habili tado	Vivien daTe mpora l

CK1: ('ViviendaUniversitaria', 'HabitacionVivienda', 'Apartamento', 'HabitacionHotel', 'Hostal', 'ViviendaTemporal')

ViviendaUniversitaria

IdAlojamiento	TipoHabitacion
PK, FK (Alojamiento.IdAlojamiento)	NN, CK ('Compartida', 'Individual')
1	Compartida

HabitacionVivienda

IdAlojamiento	TipoBaño	TipoHabitacion
PK, FK (Alojamiento.IdAlojamiento)	NN, CK ('Privado', 'Publico')	NN, CK ('Compartida', 'Individual')
2	Privado	Compartida

Apartamento

IdAlojamiento
PK, FK (Alojamiento.IdAlojamiento)
3

HabitacionHotel

IdAlojamiento	TipoHabitacion
PK, FK (Alojamiento.IdAlojamiento)	NN, CK ('Estandar', 'Semisuites', 'Suites')
4	

Hostal

IdAlojamiento	HorarioApertura	HorarioCierre
PK, FK (Alojamiento.IdAlojamiento)	NN	NN
5	18:00	10:00

ViviendaTemporal

IdAlojamiento	NumHabitaciones
PK, FK (Alojamiento.IdAlojamiento)	NN, CK (>0)
6	4

Cliente

IdCliente	TipoIdentificacion	NombreCliente	FechaNacimiento
PK, UA	NN, CK ('CC', 'TI', 'PA')	NN	NN
1000613198	CC	Santiago Vargas	09/12/2002
43724547	CC	Angela Prada	02/11/1969
1000658712	CC	Esteban Cuellar	04/01/2002
1447856981	CC	Juan Garavito	05/12/1989

MiembroActivo

IdMiembroActivo	Carnet	Tipo
PK, FK (Cliente.IdCliente)	NN, ND	NN, CK ('Estudiante', 'Profesor visitante', 'Empleado', 'Profesor titular')
1000613198	202013826	Estudiante
43724547	201914587	Empleado
1000658712	202014258	Estudiante

MiembroSecundario

IdMiembroSecundario	Tipo
PK, FK (Cliente.IdCliente)	NN, CK ('Padre', 'Egresado', 'Invitado')
1447856981	Padre

Servicio

IdServicio	Nombre	Costo
PK, SA	NN	NN, CK (>0)
101	Internet	50000
102	Cocineta	0

Dispone

IdServicio	IdAlojamiento
PK, FK (Servicio.IdServicio)	PK, FK (Alojamiento.IdAlojamiento)
101	1

Reserva

IdReserva	IdAlojamiento	IdCliente	Duracion	FechaInicio	FechaFinal	CostoTotal	Estado	NumPersonas	IdReservaColectiva
PK, SA	NN, FK (Alojamiento.IdAlojamiento)	NN, FK (Cliente.IdCliente)	NN, CK (>0)	NN	NN	NN, CK (>0)	NN, CK ('Activa', 'Cancelada', 'Finalizada')	NN, CK (>0)	
1	3	1000613198	15	15/01/2013	30/01/2013	1200000	Finalizada	2	NUL

3. Diseño de la aplicación

3.1 Listado con las tablas generadas en la base de datos

A continuación, se presenta un listado detallado de cada una de las tablas generadas en la base de datos desde SQL developer. Para cada una de ellas, se incluye el nombre, los nombres y tipos de los atributos, las restricciones existentes, etc.

A_ALOJAMIENTO						
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDALOJAMIENTO	NUMBER	No	(null)	1 (null)		
2 NOMBRE	VARCHAR2(255 BYTE)	No	(null)	2 (null)		
3 CAPACIDAD	NUMBER	No	(null)	3 (null)		
4 UBICACION	VARCHAR2(255 BYTE)	No	(null)	4 (null)		
5 TAMANO	NUMBER	No	(null)	5 (null)		
6 PRECIONOCHE	NUMBER	No	(null)	6 (null)		
7 OCUPACIONACTUAL	NUMBER	No	(null)	7 (null)		
8 NUMRESERVAS	NUMBER	No	(null)	8 (null)		
9 IDOPERADOR	NUMBER	No	(null)	9 (null)		
10 ESTADO	VARCHAR2(255 BYTE)	No	(null)	10 (null)		
11 TIPO	VARCHAR2(255 BYTE)	No	(null)	11 (null)		

Figura 2: información de la tabla ‘Alojamiento’

A_ALOJAMIENTO						
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION				
1 A_ALOJAMIENTO_PK	Primary_Key	(null)				
2 CK_A_ESTADO	Check	ESTADO IN ('Habilitado', 'Deshabilitado')				
3 CK_A_TIPO	Check	TIPO IN ('ViviendaUniversitaria', 'HabitacionVivienda', 'Apartamento', 'HabitacionHotel', 'Hostal', 'ViviendaTemporal')				
4 CK_CAPACIDAD	Check	CAPACIDAD > 0				
5 CK_NUMRESERVAS	Check	NUMRESERVAS >= 0				
6 CK_OCUPACIONACTUAL	Check	OCUPACIONACTUAL >= 0 AND OCUPACIONACTUAL <= CAPACIDAD				
7 CK_PRECIONOCHE	Check	PRECIONOCHE > 0				
8 CK_TAMANO	Check	TAMANO > 0				
9 FK_A_OPERADOR	Foreign_Key	(null)				
10 SYS_C00997707	Check	"NOMBRE" IS NOT NULL				
11 SYS_C00997708	Check	"CAPACIDAD" IS NOT NULL				
12 SYS_C00997709	Check	"UBICACION" IS NOT NULL				
13 SYS_C00997710	Check	"TAMANO" IS NOT NULL				
14 SYS_C00997711	Check	"PRECIONOCHE" IS NOT NULL				
15 SYS_C00997712	Check	"OCUPACIONACTUAL" IS NOT NULL				
16 SYS_C00997713	Check	"NUMRESERVAS" IS NOT NULL				
17 SYS_C00997714	Check	"IDOPERADOR" IS NOT NULL				
18 SYS_C00997715	Check	"ESTADO" IS NOT NULL				
19 SYS_C00997716	Check	"TIPO" IS NOT NULL				

Figura 3: restricciones de la tabla ‘Alojamiento’

A_APARTAMENTO						
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDALOJAMIENTO	NUMBER	No	(null)	1 (null)		

Figura 4: información de la tabla ‘Apartamento’

A_APARTAMENTO

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | S

</

Figura 5: restricciones de la tabla ‘Apartamento’

A_CLIENTE					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDCLIENTE	NUMBER	No	(null)	1 (null)	
2 TIPOIDENTIFICACION	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3 NOMBRECLIENTE	VARCHAR2(255 BYTE)	No	(null)	3 (null)	
4 FECHANACIMIENTO	DATE	No	(null)	4 (null)	

Figura 6: información de la tabla ‘Cliente’

A_CLIENTE					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION			
1 A_CLIENTE_PK	Primary_Key	(null)			
2 CK_C_TIPOIDENTIFICACION	Check	TIPOIDENTIFICACION IN ('CC', 'TI', 'PA')			
3 SYS_C00997750	Check	"TIPOIDENTIFICACION" IS NOT NULL			
4 SYS_C00997751	Check	"NOMBRECLIENTE" IS NOT NULL			
5 SYS_C00997752	Check	"FECHANACIMIENTO" IS NOT NULL			

Figura 7: restricciones de la tabla ‘Cliente’

A_DISPONE					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDSERVICIO	NUMBER	No	(null)	1 (null)	
2 IDALOJAMIENTO	NUMBER	No	(null)	2 (null)	

Figura 8: información de la tabla ‘Dispone’

A_DISPONE					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_DISPONE_PK	Primary_Key	(null)	(null)	(null)	(null)
2 FK_D_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK
3 FK_D_SERVICIO	Foreign_Key	(null)	ISIS2304A29202310	A_SERVICIO	A_SERVICIO_PK

Figura 9: restricciones de la tabla ‘Dispone’

A_EMPRESA					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDOPERADOR	NUMBER	No	(null)	1 (null)	
2 NIT	VARCHAR2(255 BYTE)	No	(null)	2 (null)	
3 NOMBREENPRESA	VARCHAR2(255 BYTE)	No	(null)	3 (null)	

Figura 10: información de la tabla ‘Empresa’

A_EMPRESA					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_EMPRESA_PK	Primary_Key	(null)	(null)	(null)	(null)
2 FK_E_OPERADOR	Foreign_Key	(null)	ISIS2304A29202310	A_OPERADOR	A_OPERADOR_PK
3 SYS_C00997702	Check	"NIT" IS NOT NULL	(null)	(null)	(null)
4 SYS_C00997703	Check	"NOMBREEMPRESA" IS NOT NULL	(null)	(null)	(null)
5 UN_EMPRESA_NIT	Unique	(null)	(null)	(null)	(null)

Figura 11: restricciones de la tabla ‘Empresa’

A_HABITACIONHOTEL					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencia					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDALOJAMIENTO	NUMBER	No	(null)	1	(null)
2 TIPOHABITACION	VARCHAR2(20 BYTE)	No	(null)	2	(null)

Figura 12: información de la tabla ‘HabitacionHotel’

A_HABITACIONHOTEL					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_HABITACIONHOTEL_PK	Primary_Key	(null)	(null)	(null)	(null)
2 CK_HH_TIPOHABITACION	Check	TIPOHABITACION IN ('Estandar', 'Semisuites', 'Suites')	(null)	(null)	(null)
3 FK_HH_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK
4 SYS_C00997738	Check	"TIPOHABITACION" IS NOT NULL	(null)	(null)	(null)

Figura 13: restricciones de la tabla ‘HabitacionHotel’

A_HABITACIONVIVIENDA					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencia					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDALOJAMIENTO	NUMBER	No	(null)	1	(null)
2 TIPOBANO	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3 TIPOHABITACION	VARCHAR2(20 BYTE)	No	(null)	3	(null)

Figura 14: información de la tabla ‘HabitacionVivienda’

A_HABITACIONVIVIENDA					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_HABITACIONVIVIENDA_PK	Primary_Key	(null)	(null)	(null)	(null)
2 CK_HV_TIPOHABITACION	Check	TIPOHABITACION IN ('Compartida', 'Individual')	(null)	(null)	(null)
3 CK_TIPOBANO	Check	TIPOBANO IN ('Privado', 'Publico')	(null)	(null)	(null)
4 FK_HV_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK
5 SYS_C00997730	Check	"TIPOBANO" IS NOT NULL	(null)	(null)	(null)
6 SYS_C00997731	Check	"TIPOHABITACION" IS NOT NULL	(null)	(null)	(null)

Figura 15: restricciones de la tabla ‘HabitacionVivienda’

A_HOSTAL					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencia					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDALOJAMIENTO	NUMBER	No	(null)	1	(null)
2 HORARIOAPERTURA	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3 HORARIOCIERRE	VARCHAR2(20 BYTE)	No	(null)	3	(null)

Figura 16: información de la tabla ‘Hostal’

A_HOSTAL					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_HOSTAL_PK	Primary_Key	(null)	(null)	(null)	(null)
2 FK_H_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK
3 SYS_C00997742	Check	"HORARIOAPERTURA" IS NOT NULL	(null)	(null)	(null)
4 SYS_C00997743	Check	"HORARIOCIERRE" IS NOT NULL	(null)	(null)	(null)

Figura 17: restricciones de la tabla ‘Hostal’

A_MIEMBROACTIVO					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDMIEMBROACTIVO	NUMBER	No	(null)	1 (null)	
2 CARNET	VARCHAR2(255 BYTE)	No	(null)	2 (null)	
3 TIPO	VARCHAR2(255 BYTE)	No	(null)	3 (null)	

Figura 18: información de la tabla ‘MiembroActivo’

A_MIEMBROACTIVO					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_MIEMBROACTIVO_PK	Primary_Key	(null)	(null)	(null)	(null)
2 CK_MA_MIEMBROACTIVO	Check	TIPO IN ('Estudiante', 'Profesor visitante', 'Empleado', 'Profesor titular')	(null)	(null)	(null)
3 FK_MA_CLIENTE	Foreign_Key	(null)	ISIS2304A29202310	A_CLIENTE	A_CLIENTE_PK
4 SYS_C00997755	Check	"CARNET" IS NOT NULL	(null)	(null)	(null)
5 SYS_C00997756	Check	"TIPO" IS NOT NULL	(null)	(null)	(null)
6 UN_MIEMBROACTIVO_CARNET	Unique	(null)	(null)	(null)	(null)

Figura 19: restricciones de la tabla ‘MiembroActivo’

A_MIEMBROSECUNDARIO					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDMIEMBROSECUNDARIO	NUMBER	No	(null)	1 (null)	
2 TIPO	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

Figura 20: información de la tabla ‘MiembroSecundario’

A_MIEMBROSECUNDARIO					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL					
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_MIEMBROSECUNDARIO_PK	Primary_Key	(null)	(null)	(null)	(null)
2 CK_MS_TIPO	Check	TIPO IN ('Padre', 'Egresado', 'Invitado')	(null)	(null)	(null)
3 FK_MS_CLIENTE	Foreign_Key	(null)	ISIS2304A29202310	A_CLIENTE	A_CLIENTE_PK
4 SYS_C00997761	Check	"TIPO" IS NOT NULL	(null)	(null)	(null)

Figura 21: restricciones de la tabla ‘MiembroSecundario’

A_OPERADOR					
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias					
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDOPERADOR	NUMBER	No	(null)	1 (null)	
2 TELEFONO	VARCHAR2(255 BYTE)	No	(null)	2 (null)	
3 TIPOVINCULACION	VARCHAR2(255 BYTE)	No	(null)	3 (null)	

Figura 22: información de la tabla ‘Operador’

A_OPERADOR						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	
1 A_OPERADOR_PK	Primary_Key	(null)	(null)	(null)	(null)	
2 SYS_C00997691	Check	"TELEFONO" IS NOT NULL	(null)	(null)	(null)	
3 SYS_C00997692	Check	"TIPOVINCULACION" IS NOT NULL	(null)	(null)	(null)	
4 UN_OPERADOR_TELEFONO	Unique	(null)	(null)	(null)	(null)	

Figura 23: restricciones de la tabla ‘Operador’

A_PROPIETARIO						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias						
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDOPERADOR	NUMBER	No	(null)	1	(null)	
2 IDENTIFICACION	VARCHAR2(255 BYTE)	No	(null)	2	(null)	
3 TIPOIDENTIFICACION	VARCHAR2(20 BYTE)	No	(null)	3	(null)	
4 NOMBREPROPIETARIO	VARCHAR2(255 BYTE)	No	(null)	4	(null)	

Figura 24: información de la tabla ‘Propietario’

A_PROPIETARIO						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	
1 A_PROPIETARIO_PK	Primary_Key	(null)	(null)	(null)	(null)	
2 CK_P_TIPOIDENTIFICACION	Check	TIPOIDENTIFICACION IN ('CC', 'TI', 'PA')	(null)	(null)	(null)	
3 FK_P_OPERADOR	Foreign_Key	(null)	ISIS2384A29202310 A_OPERADOR	A_OPERADOR	A_OPERADOR_PK	
4 SYS_C00997695	Check	"IDENTIFICACION" IS NOT NULL	(null)	(null)	(null)	
5 SYS_C00997696	Check	"TIPOIDENTIFICACION" IS NOT NULL	(null)	(null)	(null)	
6 SYS_C00997697	Check	"NOMBREPROPIETARIO" IS NOT NULL	(null)	(null)	(null)	
7 UN_PROPIETARIO_IDENTIFICACION	Unique	(null)	(null)	(null)	(null)	

Figura 25: restricciones de la tabla ‘Propietario’

A_RESERVA						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias						
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDRESERVA	NUMBER	No	(null)	1	(null)	
2 IDALOJAMIENTO	NUMBER	No	(null)	2	(null)	
3 IDCLIENTE	NUMBER	No	(null)	3	(null)	
4 DURACION	NUMBER	No	(null)	4	(null)	
5 FECHAINICIO	DATE	No	(null)	5	(null)	
6 FECHAFINAL	DATE	No	(null)	6	(null)	
7 COSTOTOTAL	NUMBER	No	(null)	7	(null)	
8 ESTADO	VARCHAR2(20 BYTE)	No	(null)	8	(null)	
9 NUMPERSONAS	NUMBER	No	(null)	9	(null)	
10 IDRESERVACOLETIVA	NUMBER	Yes	(null)	10	(null)	

Figura 26: información de la tabla ‘Reserva’

A_RESERVA						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	
1 A_RESERVA_PK	Primary_Key	(null)	(null)	(null)	(null)	
2 CK_R_COSTOTOTAL	Check	COSTOTOTAL > 0	(null)	(null)	(null)	
3 CK_R_DURACION	Check	DURACION > 0	(null)	(null)	(null)	
4 CK_R_ESTADO	Check	ESTADO IN ('Cancelada', 'Finalizada', 'Activa')	(null)	(null)	(null)	
5 CK_R_NUMPERSONAS	Check	NUMPERSONAS > 0	(null)	(null)	(null)	
6 FK_R_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK	
7 FK_R_CLIENTE	Foreign_Key	(null)	ISIS2304A29202310	A_CLIENTE	A_CLIENTE_PK	
8 SYS_C00997772	Check	"IDALOJAMIENTO" IS NOT NULL	(null)	(null)	(null)	
9 SYS_C00997773	Check	"IDCLIENTE" IS NOT NULL	(null)	(null)	(null)	
10 SYS_C00997774	Check	"DURACION" IS NOT NULL	(null)	(null)	(null)	
11 SYS_C00997775	Check	"FECHAINICIO" IS NOT NULL	(null)	(null)	(null)	
12 SYS_C00997776	Check	"FECHAFINAL" IS NOT NULL	(null)	(null)	(null)	
13 SYS_C00997777	Check	"COSTOTOTAL" IS NOT NULL	(null)	(null)	(null)	
14 SYS_C00997778	Check	"ESTADO" IS NOT NULL	(null)	(null)	(null)	
15 SYS_C00997779	Check	"NUMPERSONAS" IS NOT NULL	(null)	(null)	(null)	

Figura 27: restricciones de la tabla ‘Reserva’

A_SERVICIO						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias						
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDSERVICIO	NUMBER	No	(null)	1	(null)	
2 NOMBRE	VARCHAR2(255 BYTE)	No	(null)	2	(null)	
3 COSTO	NUMBER	No	(null)	3	(null)	

Figura 28: información de la tabla ‘Servicio’

A_SERVICIO						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	
1 A_SERVICIO_PK	Primary_Key	(null)	(null)	(null)	(null)	
2 CK_S_COSTO	Check	COSTO >= 0	(null)	(null)	(null)	
3 SYS_C00997765	Check	"NOMBRE" IS NOT NULL	(null)	(null)	(null)	
4 SYS_C00997766	Check	"COSTO" IS NOT NULL	(null)	(null)	(null)	

Figura 29: restricciones de la tabla ‘Servicio’

A_VIVIENDATEMPORAL						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 IDALOJAMIENTO	NUMBER	No	(null)	1	(null)	
2 NUMHABITACIONES	NUMBER	No	(null)	2	(null)	

Figura 30: información de la tabla ‘ViviendaTemporal’

A_VIVIENDATEMPORAL						
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL						
Acciones...						
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	
1 A_VIVIENDATEMPORAL_PK	Primary_Key	(null)	(null)	(null)	(null)	
2 CK_VT_NUMHABITACIONES	Check	NUMHABITACIONES > 0	(null)	(null)	(null)	
3 FK_VT_ALOJAMIENTO	Foreign_Key	(null)	ISIS2304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK	
4 SYS_C00997746	Check	"NUMHABITACIONES" IS NOT NULL	(null)	(null)	(null)	

Figura 31: restricciones de la tabla ‘ViviendaTemporal’

A_VIVIENDAUNIVERSITARIA					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDALOJAMIENTO	NUMBER	No	(null)	1 (null)	
2 TIPOHABITACION	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

Figura 32: información de la tabla ‘ViviendaUniversitaria’

A_VIVIENDAUNIVERSITARIA					
Columnas	Datos	Model	Restricciones	Permisos	Estadísticas
Acciones...					
CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1 A_VIVIENDAUNIVERSITARIA_PK	Primary_Key	(null)	(null)	(null)	(null)
2 CK_VU_TIPOHABITACION	Check	TIPOHABITACION IN ('Compartida', 'Individual')	(null)	(null)	(null)
3 FK_VU_ALOJAMIENTO	Foreign_Key	(null)	IS152304A29202310	A_ALOJAMIENTO	A_ALOJAMIENTO_PK
4 SYS_C00997726	Check	"TIPOHABITACION" IS NOT NULL	(null)	(null)	(null)

Figura 33: restricciones de la tabla ‘ViviendaUniversitaria’

3.2 Especificación de los nuevos requerimientos

Nombre	RF7 - REGISTRAR RESERVA COLECTIVA
Resumen	El usuario indica el tipo de alojamiento deseado y la cantidad deseada, por ejemplo, 100 habitaciones sencillas con ciertos servicios deseados. ALOHANDES debe revisar si está en capacidad de satisfacer esa solicitud, eventualmente con varios proveedores, y en caso afirmativo realizar las reservas individuales correspondientes. La reserva colectiva es identificable de manera individual.
Entradas	
Tipo de alojamiento deseado (ViviendaUniversitaria’, ‘HabitacionVivienda’, ‘Apartamento’, ‘HabitacionHotel’, ‘Hostal’, ‘ViviendaTemporal’)	
Cantidad de habitaciones a reservar	
Informacion general (identificador del cliente, duracion, fecha inicio, fecha final, etc)	
Resultados	
Si hay suficiente disponibilidad, se registran las nuevas reservas y se retorna el identificador correspondiente a la reserva colectiva realizada.	
Si no hay disponibilidad suficiente, entonces se notifica al usuario que no fue posible hacer la reserva colectiva dada.	
RNF asociados	
Con el propósito de que se cumplan las propiedades ACID para una transacción que requiere de los procesos de lectura y escritura, es necesario definir los mecanismos a utilizar. En primer lugar, sin importar el nivel de aislamiento de la transacción, siempre se colocarán candados exclusivos de escritura de larga duración sobre aquellas tuplas que estén siendo afectadas. Lo anterior hace que, cuando otra transacción solicite recursos que estén siendo utilizados por la transacción dada, se restrinja el acceso a estos datos hasta que no se libere el candado.	

En caso de que por alguna razón se deba abortar la transacción, herramientas como la función de ROLLBACK y los registros existentes en el LOG permiten deshacer los cambios realizados por la transacción. Los mecanismos descritos anteriormente, principalmente los candados de escritura de larga duración evitan la anomalía de datos por actualización, así como otros problemas que pueden potencialmente provocar que la base de datos termine en un estado inconsistente.

Nombre	RF8 - CANCELAR RESERVA COLECTIVA
Resumen	El usuario indica la reserva colectiva que quiere cancelar y ALOHANDES cancela las reservas individuales correspondientes y calcula también las penalizaciones correspondientes.
Entradas	
Id de la reserva colectiva a cancelar	
Resultados	
Se cambia el estado de reserva a ‘Cancelada’ en todas aquellas reservas que hagan parte de la reserva colectiva de interes	
RNF asociados	
<p>Con el propósito de que se cumplan las propiedades ACID para una transacción que requiere de los procesos de lectura y escritura, es necesario definir los mecanismos a utilizar. En primer lugar, sin importar el nivel de aislamiento de la transacción, siempre se colocarán candados exclusivos de escritura de larga duración sobre aquellas tuplas que estén siendo afectadas. Lo anterior hace que, cuando otra transacción solicite recursos que estén siendo utilizados por la transacción dada, se restrinja el acceso a estos datos hasta que no se libere el candado.</p> <p>En caso de que por alguna razón se deba abortar la transacción, herramientas como la función de ROLLBACK y los registros existentes en el LOG permiten deshacer los cambios realizados por la transacción. Los mecanismos descritos anteriormente, principalmente los candados de escritura de larga duración evitan la anomalía de datos por actualización, así como otros problemas que pueden potencialmente provocar que la base de datos termine en un estado inconsistente.</p>	

Nombre	RF9 - DESHABILITAR OFERTA DE ALOJAMIENTO
Resumen	Consiste en deshabilitar una oferta de alojamiento (impedir que se pueda reservar temporalmente). Esto implica que se tienen que las reservas vigentes deben relocarse en otras ofertas de alojamiento disponibles.
Entradas	
Id de la oferta de alojamiento a deshabilitar	

Resultados
Cambia las reservas vigentes a otras ofertas de alojamiento
Cambiar el atributo de 'estado' del alojamiento dado
Informar acerca de las reservas que se pudieron trasladar a otro alojamiento y aquellas que se tuvieron que cancelar
RNF asociados
<p>Con el propósito de que se cumplan las propiedades ACID para una transacción que requiere de los procesos de lectura y escritura, es necesario definir los mecanismos a utilizar. En primer lugar, sin importar el nivel de aislamiento de la transacción, siempre se colocarán candados exclusivos de escritura de larga duración sobre aquellas tuplas que estén siendo afectadas. Lo anterior hace que, cuando otra transacción solicite recursos que estén siendo utilizados por la transacción dada, se restrinja el acceso a estos datos hasta que no se libere el candado.</p> <p>En caso de que por alguna razón se deba abortar la transacción, herramientas como la función de ROLLBACK y los registros existentes en el LOG permiten deshacer los cambios realizados por la transacción. Los mecanismos descritos anteriormente, principalmente los candados de escritura de larga duración evitan la anomalía de datos por actualización, así como otros problemas que pueden potencialmente provocar que la base de datos termine en un estado inconsistente.</p>

Nombre	RF10 - REHABILITAR OFERTA DE ALOJAMIENTO
Resumen	Consiste en volver a habilitar una oferta de alojamiento para aceptar nuevas reservas
Entradas	
	Id de la oferta de alojamiento a rehabilitar
Resultados	
	Cambiar el atributo de 'estado' del alojamiento dado para permitir nuevas reservas
RNF asociados	
	<p>Con el propósito de que se cumplan las propiedades ACID para una transacción que requiere de los procesos de lectura y escritura, es necesario definir los mecanismos a utilizar. En primer lugar, sin importar el nivel de aislamiento de la transacción, siempre se colocarán candados exclusivos de escritura de larga duración sobre aquellas tuplas que estén siendo afectadas. Lo anterior hace que, cuando otra transacción solicite recursos que estén siendo utilizados por la transacción dada, se restrinja el acceso a estos datos hasta que no se libere el candado.</p>

En caso de que por alguna razón se deba abortar la transacción, herramientas como la función de ROLLBACK y los registros existentes en el LOG permiten deshacer los cambios realizados por la transacción. Los mecanismos descritos anteriormente, principalmente los candados de escritura de larga duración evitan la anomalía de datos por actualización, así como otros problemas que pueden potencialmente provocar que la base de datos termine en un estado inconsistente.

Nombre	RFC5 - MOSTRAR EL USO DE ALOHANDES PARA CADA TIPO DE USUARIO DE LA COMUNIDAD
Resumen	Muestra algunas características generales como el número de noches contratadas, el total de dinero pagado, entre otras. Esto se hace para cada uno de los tipos de usuarios dentro de ALOHAANDES (Estudiante, profesor, empleado, etc)
Entradas	
Ninguna	
Resultados	
Se retornan las diferentes características encontradas para cada uno de los tipos de alojamiento existentes en el sistema	
RNF asociados	
<p>Teniendo en cuenta de que el requerimiento en cuestión es de tipo consulta, para garantizar el cumplimiento de las propiedades ACID del requerimiento, por defecto se implementan candados de lectura para disminuir las posibilidades de que ocurra alguna anomalía y ofreciendo características parciales de aislamiento y atomicidad. Además, en caso de que la transacción se abortada por el usuario o por algún conflicto con otras operaciones, el sistema hace automáticamente ROLLBACK para deshacer los cambios hechos y posteriormente informa al usuario que no se pudo completar la transacción.</p> <p>Para retornar a un estado consistente o recuperar la información de la base de datos almacenada en memoria secundaria. Existen diferentes archivos de tipo bitácora que contienen la información de los cambios y las 'before images' para restablecer la base de datos. Lo peor que puede ocurrir al momento de ejecutar la transacción para RFC5, es la de que se padezca alguna anomalía de lectura (fantasmas, lecturas sucias, lecturas no repetibles, etc), dependiendo del nivel de aislamiento con el que se ejecute la transacción</p>	

Nombre	RFC6 - MOSTRAR EL USO DE ALOHANDES PARA UN USUARIO DADO (NÚMERO DE NOCHES O MESES CONTRATADOS, CARACTERÍSTICAS DEL ALOJAMIENTO UTILIZADO, DINERO PAGADO
---------------	---

Resumen	Muestra algunas características generales (el número de noches contratadas, el total de dinero pagado, etc) para un usuario específico de ALOHAANDES.
Entradas	
Id de un cliente dentro del sistema	
Resultados	
Se retornan las diferentes características encontradas para el cliente cuyo identificador concuerde con aquel que fue provisto como entrada por el usuario	
RNF asociados	
<p>Teniendo en cuenta de que el requerimiento en cuestión es de tipo consulta, para garantizar el cumplimiento de las propiedades ACID del requerimiento, por defecto se implementan candados de lectura para disminuir las posibilidades de que ocurra alguna anomalía y ofreciendo características parciales de aislamiento y atomicidad. Además, en caso de que la transacción se abortada por el usuario o por algún conflicto con otras operaciones, el sistema hace automáticamente ROLLBACK para deshacer los cambios hechos y posteriormente informa al usuario que no se pudo completar la transacción.</p> <p>Para retornar a un estado consistente o recuperar la información de la base de datos almacenada en memoria secundaria. Existen diferentes archivos de tipo bitácora que contienen la información de los cambios y las 'before images' para restablecer la base de datos. Lo peor que puede ocurrir al momento de ejecutar la transacción para RFC6, es la de que se padezca alguna anomalía de lectura (fantasmas, lecturas sucias, lecturas no repetibles, etc), dependiendo del nivel de aislamiento con el que se ejecute la transacción.</p>	

Nombre	RFC7 - ANALIZAR LA OPERACIÓN DE ALOHANDES
Resumen	Para una rango de fechas dado y un tipo de alojamiento, se encuentran los días dentro de ese rango con la mayor demanda, aquellos con el menor índice de ocupación y aquellos con el mayor número de ingresos.
Entradas	
Fecha de inicio del intervalo	
Fecha final del intervalo	
Tipo de alojamiento	
Resultados	
Fechas con la mayor demanda dentro del rango (Mayor número de alojamientos ocupados)	
Fechas con el mayor índice de ocupación dentro del rango	

Fechas con la mayor cantidad de ingresos resgistrados dentro del rango

RNF asociados

Teniendo en cuenta de que el requerimiento en cuestión es de tipo consulta, para garantizar el cumplimiento de las propiedades ACID del requerimiento, por defecto se implementan candados de lectura para disminuir las posibilidades de que ocurra alguna anomalía y ofreciendo características parciales de aislamiento y atomicidad. Además, en caso de que la transacción se abortada por el usuario o por algún conflicto con otras operaciones, el sistema hace automáticamente ROLLBACK para deshacer los cambios hechos y posteriormente informa al usuario que no se pudo completar la transacción.

Para retornar a un estado consistente o recuperar la información de la base de datos almacenada en memoria secundaria. Existen diferentes archivos de tipo bitácora que contienen la información de los cambios y las 'before images' para restablecer la base de datos. Lo peor que puede ocurrir al momento de ejecutar la transacción para RFC7, es la de que se padezca alguna anomalía de lectura (fantasmas, lecturas sucias, lecturas no repetibles, etc), dependiendo del nivel de aislamiento con el que se ejecute la transacción.

Nombre	RFC8 - ENCONTRAR LOS CLIENTES FRECUENTES
Resumen	Encontrar la información de sus clientes frecuentes de un alojamiento, los cuales son todos aquellos que tiene 3 o mas reservas a su nombre o se hospedado por lo menos 15 noches.
Entradas	
Id del alojamiento de interes	
Resultados	
Se retorna un listado con la informacion de los clientes frecuentes del alojamiento dado	
RNF asociados	
<p>Teniendo en cuenta de que el requerimiento en cuestión es de tipo consulta, para garantizar el cumplimiento de las propiedades ACID del requerimiento, por defecto se implementan candados de lectura para disminuir las posibilidades de que ocurra alguna anomalía y ofreciendo características parciales de aislamiento y atomicidad. Además, en caso de que la transacción se abortada por el usuario o por algún conflicto con otras operaciones, el sistema hace automáticamente ROLLBACK para deshacer los cambios hechos y posteriormente informa al usuario que no se pudo completar la transacción.</p> <p>Para retornar a un estado consistente o recuperar la información de la base de datos almacenada en memoria secundaria. Existen diferentes archivos de tipo bitácora que contienen la información de los cambios y las 'before images' para restablecer la base de datos. Lo peor que puede ocurrir al momento de ejecutar la transacción para RFC8, es la de que se padezca alguna anomalía de lectura</p>	

(fantasmas, lecturas sucias, lecturas no repetibles, etc), dependiendo del nivel de aislamiento con el que se ejecute la transacción.

Nombre	RFC9 - ENCONTRAR LAS OFERTAS DE ALOJAMIENTO QUE NO TIENEN MUCHA DEMANDA
Resumen	Encontrar las ofertas de alojamiento que no han recibido clientes en periodos superiores a 1 mes, durante todo el periodo de operación de AloHAndes.
Entradas	
Fecha actual en el formato indicado	
Resultados	
Se retorna la informacion de todos aquellos alojamientos que no han recibido clientes desde hace mas de 1 mes.	
RNF asociados	
<p>Teniendo en cuenta de que el requerimiento en cuestión es de tipo consulta, para garantizar el cumplimiento de las propiedades ACID del requerimiento, por defecto se implementan candados de lectura para disminuir las posibilidades de que ocurra alguna anomalía y ofreciendo características parciales de aislamiento y atomicidad. Además, en caso de que la transacción se abortada por el usuario o por algún conflicto con otras operaciones, el sistema hace automáticamente ROLLBACK para deshacer los cambios hechos y posteriormente informa al usuario que no se pudo completar la transacción.</p> <p>Para retornar a un estado consistente o recuperar la información de la base de datos almacenada en memoria secundaria. Existen diferentes archivos de tipo bitácora que contienen la información de los cambios y las 'before images' para restablecer la base de datos. Lo peor que puede ocurrir al momento de ejecutar la transacción para RFC6, es la de que se padezca alguna anomalía de lectura (fantasmas, lecturas sucias, lecturas no repetibles, etc), dependiendo del nivel de aislamiento con el que se ejecute la transacción.</p>	

4. Construcción de la aplicación

4.1 Desarrollo y/o ajustes para cumplir con los nuevos requerimientos

Para cumplir a cabalidad con los nuevos requerimientos funcionales solicitados, antes que nada, se tuvo que actualizar el esquema de creación de tablas en concordancia con los cambios

introducidos al modelo relacional. Como se incluyeron exclusivamente nuevos atributos y nuevas restricciones de integridad, no fue necesario crear nuevas clases ni tampoco modificar los archivos de tipo 'TablasBD' dentro del proyecto.

Adicionalmente, se crearon los métodos correspondientes para cada uno de los requerimientos dentro de la interfaz y se agregaron las opciones correspondientes en archivo de formato JSON. Para todos aquellos requerimientos de consulta o aquellos funcionales que requirieran comprobar información existente en la base de datos, se creó un método de listar para recorrer el arreglo de tuplas resultantes tras ejecutar la sentencia SQL respectiva y transformar estos datos en cadenas de textos para que se pudieran mostrar en la interfaz.

Por otro lado, se tuvo que cambiar los archivos 'Alojamiento.java', 'VOAlojamiento.java', 'Reserva.java', 'VOReserva' del módulo de negocio para abarcar los nuevos atributos de ambas tablas. Por último, se tuvo que rediseñar el archivo SQL con las tuplas de ejemplo para las pruebas para incluir los nuevos atributos.

Para hacer efectivos y visibles los cambios realizados por la aplicación de java, se tuvo que borrar los esquemas de las tablas anteriores, crear las tablas con el nuevo esquema para la iteración 3 e insertar las nuevas tuplas de prueba. Para que los datos fueran persistidos, como el 'autocommit' por defecto esta desactivado, se debió hacer el 'commit' de manera manual para que los cambios fueran visibles desde la aplicación.

4.2 Cambios y desarrollo de las transacciones

Para garantizar que la base de datos no quede en un estado inconsistente, existen diversos mecanismos en la aplicación de Java que ayudan sobre las condiciones ACID de las operaciones de negocio. En primer lugar, siempre que se esté ejecutando una transacción relacionada con algún requerimiento, se guarda un 'begin record' para que al momento de hacer 'Rollback' haya un límite que indique hasta que punto deshacer las operaciones.

Por otra parte, si la transacción se completa de manera exitosa, entonces se hace 'commit' de esta por medio del método 'commit()', de surgir un error durante su ejecución, se hace 'rollback()' de esta para mantener un estado consistente de la información. La instancia para la transacción en cuestión se crea a partir del 'persistence manager', el cual se utiliza principalmente para efectuar las sentencias SQL y hacer que los cambios efectivamente sean persistentes.

Otra forma en la que se garantiza un comportamiento transaccional adecuado es el uso de tanto 'Parranderos.log' como de 'Datanucleus.log'. El log de parranderos se utiliza principalmente para asegurarse de que todas las funciones que componen la transacción se hayan ejecutado sin contratiempos. En cuanto al log de datanucleus, su principal función es la de registrar todas las operaciones relacionadas con cada transacción llevada a cabo. Si por alguna razón se vuelve apremiante hacer 'Rollback' de una transacción, se deshacen las operaciones asociadas a esta a partir de las 'before images' de los registros de actualización hasta llegar al 'begin record' de dicha transacción.

En cuanto al manejo de la concurrencia, por defecto, las sentencias de inserción y modificación emplean candados exclusivos de larga duración, evitando así que puedan ocurrir anomalías de escritura dentro de la base de datos. En lo que concierne a la lectura, si bien es posible que aparezcan algunos tipos de anomalías al no ser todas las transacciones serializables, la

implementación de candados lectura mitigan los posibles efectos adversos que pueden tener lecturas concurrentes sobre los mismos datos

4.3 Cambios y desarrollo en el módulo de persistencia de la aplicación

Debido a las modificaciones que fueron aplicadas al esquema de los datos, los requerimientos funcionales y de consulta de registrar una oferta de alojamiento (RF2), registrar una nueva reserva (RF4) y mostrar los alojamientos disponibles en un rango de fechas (RFC4), se actualizaron para considerar los nuevos atributos.

En principio, RFC4 al ser un requerimiento de consulta parece no verse afectado por los recientes cambios, como se están solicitando los alojamientos disponibles, quiere decir que se tiene que revisar que estén habilitados para ser considerados con lo cual. Por lo cual, resulta indispensable revisar el atributo de estado de los alojamientos dentro de la sentencia SQL. En la figura 34 se presenta la respectiva consulta dentro del módulo de persistencia de la aplicación.

```
// RFC4: MOSTRAR LOS ALOJAMIENTOS DISPONIBLES EN UN RANGO DE FECHAS, QUE CUMPLEN CON UN CONJUNTO DE SERVICIOS
/**
 * Crea y ejecuta la sentencia SQL para encontrar la información de los Alojamientos disponibles en un rango de fechas que cuenten con
 * ciertos servicios de la base de datos de AlohaAndes.
 * @param pm - El manejador de persistencia
 * @return Una lista de arreglos de objetos. Los elementos del arreglo corresponden a los datos del alojamientos disponibles.
 */
public List<Alojamiento> darAlojamientosDisponibles (PersistenceManager pm, Date fecha1, Date fecha2, String nombreServicio)
{
    String sql = "SELECT A_Alojamiento.*";
    sql += " FROM " + pp.darTablaAlojamiento ();
    sql += " LEFT OUTER JOIN " + pp.darTablaDispone () + " ON A_Alojamiento.idalojamiento = A_Dispose.idalojamiento";
    sql += " LEFT OUTER JOIN " + pp.darTablaServicio () + " ON A_Servicio.idservicio = A_Dispose.idservicio";
    sql += " LEFT OUTER JOIN " + pp.darTablaReserva () + " ON A_Reserva.idalojamiento = A_Alojamiento.idalojamiento";
    sql += " WHERE (((A_Reserva.fechainicio NOT BETWEEN ? AND ?) AND (A_Reserva.fechafinal NOT BETWEEN ? AND ?)))";
    sql += " OR ((A_Reserva.fechainicio BETWEEN ? AND ?) OR (A_Reserva.fechafinal BETWEEN ? AND ?))";
    sql += " AND A_Reserva.estado = 'Cancelada'";
    sql += " AND A_Servicio.nombre = ?";
    sql += " AND A_Alojamiento.estado = 'Habilitado'";
    sql += " GROUP BY A_Alojamiento.idalojamiento, A_Alojamiento.nombre, A_Alojamiento.capacidad, A_Alojamiento.ubicacion, A_Alojamiento.tamano";

    Query q = pm.newQuery(SQL, sql);
    q.setParameters(fecha1, fecha2, fecha1, fecha2, fecha1, fecha2, fecha1, fecha2, nombreServicio);
    q.setResultClass(Alojamiento.class);
    return (List<Alojamiento>) q.executeList();
}
```

Figura 34: Sentencia SQL para RFC4

En cuanto a los requerimientos funcionales mencionados, para ambos se incluyeron los nuevos atributos dentro de las sentencias de tipo INSERT. Para el caso específico de RF4, debido a que para reservar un alojamiento este debe estar habilitado, antes de registrar una nueva reserva es obligatorio verificar el atributo de 'estado' del alojamiento a reservar junto con el rango de fechas y las reservas vigentes al nombre del cliente asociado.

Como ninguno de los nuevos atributos se retorna en un formato distinto a un objeto de tipo alojamiento, esto implica que los métodos previamente implementados para la iteración 2 se mantuvieron exactamente igual a como estaban. En las figuras 35 y 36 se muestran las sentencias correspondientes a la creación de un alojamiento y de una reserva, respectivamente. Mientras que en la figura 37 se puede apreciar la query empleada para verificar si es posible registrar una nueva reserva dada.

Figura 35: Sentencia SQL para RF2

Figura 36: Sentencia SQL para RF4

Figura 37: consulta de comprobación para RF4

Con respecto a los nuevos requerimientos solicitados, se agregaron nuevas sentencias en los archivos ‘SQLAlojamiento’, ‘SQLReserva’ y ‘SQLCliente’ del módulo de persistencia para implementar estos dentro de la aplicación. A continuación, se presenta de manera grafica cada uno de ellos, así como también una breve descripción para cada uno de ellos:

```
// RF7 - REGISTRAR RESERVA COLECTIVA

public List<Object> RevisarReservaColectiva (PersistenceManager pm, String tipoHabitacion, Date fechaInicio, Date
fechaFinal){

String sql = "SELECT a_alojamiento.idalojamiento, a_alojamiento.precio noche";

sql+= " FROM " + pp.darTablaAlojamiento();

sql+= " LEFT OUTER JOIN " + pp.darTablaReserva() + " ON A_alojamiento.idAlojamiento = A_reserva.idAlojamiento";

sql+= " WHERE A_alojamiento.estado = ? AND A_alojamiento.tipo = ?";

sql+= " AND A_alojamiento.idAlojamiento NOT IN ( SELECT A_reserva.idAlojamiento";

sql+= " FROM A_reserva";

sql+= " WHERE NOT((A_Reserva.fechaInicio IS NULL) OR ((? > A_Reserva.fechaInicio AND ? > A_Reserva.fechaFinal
OR (A_Reserva.estado = 'Cancelada')) OR (? < A_Reserva.fechaInicio AND ? < A_Reserva.fechaFinal OR
(A_Reserva.estado = 'Cancelada'))))";

sql+= " GROUP BY a_alojamiento.idalojamiento, a_alojamiento.precio noche";

Query q = pm.newQuery(SQL, sql);

q.setParameters("Habilitado", tipoHabitacion, fechaInicio, fechaInicio, fechaFinal, fechaFinal);

return q.executeList();}

public long RegistrarReservaIndividual(PersistenceManager pm, long idReserva, long idAlojamiento, long idCliente,
int duracion, Date fechaInicio , Date fechaFinal, long costoTotal, String estado, int numPersonas, long
idReservaColectiva){

SQLAlojamiento sqlAlojamiento = new SQLAlojamiento(pp);

Query q = pm.newQuery(SQL, "INSERT INTO " + pp.darTablaReserva() + "(idreserva, idalojamiento, idcliente,
duracion , fechainicio, fechafinal, costototal, estado, numpersonas, idreservacolectiva) values (?, ?, ?, ?, ?, ?, ?, ?, ?)");

q.setParameters(idReserva, idAlojamiento, idCliente, duracion , fechaInicio, fechaFinal, costoTotal, estado,
numPersonas, idReservaColectiva);

sqlAlojamiento.aumentarNumeroReservasAlojamiento(pm, idAlojamiento);

return (long) q.executeUnique(); }
```

Para registrar una reserva colectiva, primero se tiene que comprobar si existen suficientes alojamientos disponibles en el rango de fechas dado y del tipo de alojamiento especificado. Para tal cometido se creó la función de ‘revisarReservaColectiva()’. Si el número de

alojamientos disponibles es igual o mayor al número de alojamientos requeridos, entonces se registran las reservas respectivas en los alojamientos disponibles solicitados por medio del método ‘RegistrarReservaIndividual()’. Para calcular el costo total de la reserva se multiplica el precio por noche del alojamiento por la duración en noches de la reserva.

```
// RF8 - CANCELAR RESERVA COLECTIVA
```

```
public long cancelarReservaColectivaPorId (PersistenceManager pm, long idReservaColectiva){  
    Query q = pm.newQuery(SQL, "UPDATE " + pp.darTablaReserva() + " SET A_reserva.estado = 'Cancelada' WHERE  
    A_reserva.idreservacolectiva = ?");  
    q.setParameters(idReservaColectiva);  
    return (long) q.executeUnique(); }
```

Para detectar cuales reservas hacen parte de una reserva colectiva basta con detectar todas aquellas reservas con el mismo valor para el atributo ‘idReservaColectiva’. La función ‘cancelarReservaColectivaId()’ se encarga de cambiar el estado de la reserva de ‘Activa’ a ‘Cancelada’ de todas las reservas que tengan el mismo identificador para la reserva colectiva.

```
// RF10 - REHABILITAR OFERTA DE ALOJAMIENTO
```

```
public long rehabilitarAlojamiento( PersistenceManager pm, long idAlojamiento){  
    Query q = pm.newQuery(SQL, "UPDATE " + pp.darTablaAlojamiento() + " SET A_alojamiento.estado = 'Habilitado'  
    WHERE A_alojamiento.idAlojamiento = ? AND A_alojamiento.estado = 'Deshabilitado'");  
    q.setParameters(idAlojamiento);  
    return (long) q.executeUnique();}
```

La función de ‘rehabilitarAlojamiento()’ actualiza el estado del alojamiento de ‘Deshabilitado’ a ‘Habilitado’ para que los usuarios puedan nuevamente reservar dicho alojamiento. Si el estado de la reserva ya es ‘Habilitado’, entonces no se realiza ninguna modificación y se informa al usuario que ya el alojamiento dado se encontraba disponible.

```
// RFC5 - MOSTRAR EL USO DE ALOHANDES PARA CADA TIPO DE USUARIO DE LA COMUNIDAD
```

```
public List<Object> darInformacionMiembrosActivos (PersistenceManager pm){  
    String sql = "SELECT A_MiembroActivo.tipo, COUNT(A_Reserva.idCliente) AS Numero_reservas,  
    SUM(A_Reserva.duracion) AS Num_nochesReservadas, SUM(A_Reserva.costoSolo) AS Dinero_pagado";  
    sql += " FROM " + pp.darTablaCliente();  
    sql += " INNER JOIN " + pp.darTablaMiembroActivo() + " ON A_MiembroActivo.idMiembroActivo = A_cliente.idCliente";  
    sql += " LEFT OUTER JOIN " + pp.darTablaReserva() + " ON A_Reserva.idCliente = A_cliente.idCliente";  
    sql += " GROUP BY A_MiembroActivo.tipo";  
    Query q = pm.newQuery(SQL, sql);  
    return q.executeList();}  
  
public List<Object> darInformacionMiembrosSecundarios (PersistenceManager pm){
```

```
String sql = "SELECT A_MiembroSecundario.tipo, COUNT(A_Reserva.idCliente) AS Numero_reservas,
SUM(A_Reserva.duracion) AS Num_nochesReservadas, SUM(A_Reserva.costosTotal) AS Dinero_pagado";

sql += " FROM " + pp.darTablaCliente();

sql += " INNER JOIN " + pp.darTablaMiembroSecundario() + " ON A_MiembroSecundario.idMiembroSecundario =
A_cliente.idCliente";

sql += " LEFT OUTER JOIN " + pp.darTablaReserva() + " ON A_Reserva.idCliente = A_cliente.idCliente";

sql += " GROUP BY A_MiembroSecundario.tipo";

Query q = pm.newQuery(SQL, sql);

return q.executeList();}
```

Debido a que primero existe una relación de herencia con 2 subclases (miembroSecundario y MiembroActivo) y luego en cada una ellas se especifica el tipo de usuario, es imprescindible hacer dos consultas, una por cada subclase. Para ello, en los dos métodos se hace join entre las tablas de Cliente, MiembroActivo/MiembroSecundario y Reserva. Posteriormente, se cuentan las reservas, las noches reservadas y el dinero total pagado para cada tipo de usuario empleando agrupaciones y las funciones de agregación correspondientes.

// RFC6 - MOSTRAR EL USO DE ALOHANDES PARA UN USUARIO DADO

```
public List<Object> mostrarUsoPorUsuario (PersistenceManager pm, long idcliente){

String sql = "SELECT A_Reserva.idCliente, COUNT(A_Reserva.idCliente) AS Numero_reservas,
SUM(A_Reserva.duracion) AS Num_nochesReservadas, SUM(A_Reserva.costosTotal) AS Dinero_pagado ";

sql+= " FROM " + pp.darTablaReserva();

sql+=" WHERE A_Reserva.idCliente = ?";

sql+=" GROUP BY A_Reserva.idCliente";

Query q = pm.newQuery(SQL, sql);

q.setParameters(idcliente);

return q.executeList();}
```

Para mostrar el uso de AlohaAndes para cierto usuario se aplica una sentencia muy similar a las anteriormente vistas, con la diferencia de que se agrega un predicado para confirmar que los reservas provengan del mismo cliente y de que el agrupamiento se hace por el identificador del cliente.

// RFC8 - ENCONTRAR LOS CLIENTES FRECUENTES

```
public List<Cliente> encontrarClientesFrecuentes (PersistenceManager pm, long idAlojamiento){

String sql = "SELECT A_Cliente.*";

sql += " FROM " + pp.darTablaCliente();

sql += " LEFT OUTER JOIN " + pp.darTablaReserva() + " ON A_Cliente.idCliente = A_reserva.idCliente";

sql += " WHERE A_Reserva.idAlojamiento = ?";
```



```

sql += " GROUP BY A_Cliente.idCliente, A_Cliente.tipoIdentificacion, A_Cliente.nombreCliente, A_Cliente.fechaNacimiento";

sql += " HAVING COUNT(A_Reserva.idReserva) >= 3 OR SUM(A_Reserva.duracion) >= 15";

Query q = pm.newQuery(SQL, sql);

q.setResultClass(Cliente.class);

q.setParameters(idAlojamiento);

return (List<Cliente>) q.executeList();}

```

El método denominado ‘encontrarClientesFrecuentes()’ consiste en una unión entre las tablas de Cliente y Reserva, en donde se hace un agrupamiento de las reservas a nombre de un cliente. Finalmente, si el número de reservas del cliente es al menos 3 o la duración de las reservas es de al menos 15 días, entonces este es considerado como un cliente frecuente.

```

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

public List<Alojamiento> encontrarOfertasConBajaDemanda (PersistenceManager pm){

String sql = "SELECT A_Alojamiento.*";

sql += " FROM " + pp.darTablaAlojamiento ();

sql += " LEFT OUTER JOIN " + pp.darTablaReserva () + " ON A_Reserva.idalojamiento = A_Alojamiento.idalojamiento";

sql += " WHERE A_alojamiento.idAlojamiento NOT IN ( SELECT A_reserva.idAlojamiento";

sql += " FROM A_reserva";

sql += " WHERE NOT((A_Reserva.fechaInicio IS NULL) OR ((CURRENT_DATE - 30) > A_Reserva.fechaInicio AND (CURRENT_DATE - 30) > A_Reserva.fechaFinal OR (A_Reserva.estado = 'Cancelada')) OR (CURRENT_DATE < A_Reserva.fechaInicio AND CURRENT_DATE < A_Reserva.fechaFinal OR (A_Reserva.estado = 'Cancelada'))))";

sql += " GROUP BY A_Alojamiento.idalojamiento, A_Alojamiento.nombre, A_Alojamiento.capacidad , A_Alojamiento.ubicacion, A_Alojamiento.tamano, A_Alojamiento.precionoches, A_Alojamiento.ocupacionactual, A_Alojamiento.numreservas, A_Alojamiento.idoperador, A_Alojamiento.estado, A_Alojamiento.tipo";

sql += " ORDER BY A_Alojamiento.idalojamiento";

Query q = pm.newQuery(SQL, sql);

q.setResultClass(Alojamiento.class);

return (List<Alojamiento>) q.executeList();}

```

En la consulta en cuestión, se hace una unión entre las tablas de Alojamiento y la de Reserva. Tras esto, se realiza una consulta para obtener todas aquellas reservas que ocurrieron en los últimos 30 días. Si un alojamiento cuenta con una reserva asociada que cumpla dichas condiciones se descarta, en caso contrario se incluye dentro de los resultados finales. Por último, para que no aparezcan alojamiento repetidos, se agrupan las tuplas de la consulta por alojamiento.

5. Resultados de la iteración del proyecto

5.1 Resultados logrados

- **Actualización del esquema:** A partir de los cambios del modelo relacional, se ajustó el script SQL para la creación de las tablas para ajustarse a las nuevas modificaciones.
- **Actualización del archivo de prueba:** Se modificó el archivo con las tuplas de prueba para que concordaran con el nuevo esquema.
- **Actualización de la persistencia de la aplicación:** Se cambiaron los modelos de persistencia de 'Reserva' y 'Alojamiento' de la aplicación para coincidir con el esquema.
- **Actualización de las transacciones:** Se actualizaron las implementaciones de requisitos funcionales pasados para que sean compatibles con el esquema de la iteración 3.
- **Implementación de requerimientos funcionales de modificación:** se implementaron los requerimientos RF7, RF8, RF9, RF10 tanto en archivos sql como en la aplicación de java.
- **Implementación de requerimientos funcionales de consulta:** se implementaron los requerimientos RFC5, RFC6, RFC7, RFC8, RF9 tanto en archivos sql como en la aplicación de java.
- **Documentación pertinente:** Se actualizaron todos los modelos, el documento para la presente iteración y el resto de documentación requerida

- **Elaboración de plan de pruebas:** Se plantaron los escenarios de pruebas debido y se trabajaron con ellos para comprobar el correcto funcionamiento de los requerimientos solicitados.

5.2 Resultados no logrados

- **Carencia de facilidades de uso:** No en todos los casos se presenta se minimiza la posibilidad de datos erróneos por parte del usuario.
- **Documentación del código:** En ocasiones la documentación dentro del código resulta confusa o directamente no existe para ciertas secciones del código
- **Pruebas de verificación de las transacciones modificadas:** si bien se realizaron pruebas para los requerimientos de iteración 3, no se realizaron pruebas formales para aquellas transacciones pasadas que fueron modificadas.

6. Balance del plan de pruebas

En el presente apartado se describen los resultados obtenidos para cada los diferentes escenarios de pruebas desarrollados. Para cada requerimiento funcional se realizaron 2 pruebas, una transacción exitosa y otra que caso en que fuera esta infructuosa. Con respecto a los requerimientos de consulta, como solo se hacen lecturas de las bases de datos y muchas de ellas siempre arrojan el mismo resultado, se optó por únicamente incluir escenarios exitosos de prueba.

Los datos empleados para cada caso de prueba pueden ser consultados detalladamente en el archivo denominado ‘Datos de entrada y resultados esperados - Iteracion 3’ que se encuentra dentro del directorio ‘Docs’ del proyecto.

6.1 Pruebas de transacción exitosa

IDRESERVA	IDALOJAMIENTO	IDCLIENTE	DURACION	FECHAINICIO	FECHAFINAL	COSTOTOTAL	ESTADO	NUMPERSONAS	IDRESERVACOLECTIVA
1	1	1	1111111111	5 02/12/23	07/12/23	1000000	Activa	2	1
2	2	1	1212121212	10 02/01/23	12/01/23	2000000	Finalizada	2	1
3	3	2	1313131313	10 02/01/23	12/01/23	2000000	Cancelada	4	1
4	4	2	1414141414	10 02/11/23	12/11/23	2000000	Activa	6	(null)
5	5	3	1515151515	10 02/01/22	12/01/22	5000000	Finalizada	4	(null)
6	6	3	1616161616	10 02/11/22	12/11/22	6000000	Finalizada	6	(null)
7	7	4	1717171717	7 02/05/23	09/05/23	1000000	Activa	2	(null)
8	8	5	1515151515	7 10/02/23	17/02/23	2500000	Finalizada	2	(null)
9	9	6	1919191919	5 15/06/22	20/06/22	3000000	Finalizada	3	(null)
10	10	7	2020202020	10 15/07/23	25/07/23	4500000	Cancelada	5	(null)
11	11	8	2121212121	14 01/08/23	15/08/23	7000000	Activa	4	(null)
12	12	9	2222222222	7 10/02/23	17/02/23	2500000	Finalizada	2	(null)

Figura 38: El estado inicial de la base de datos antes de ejecutar RF7

Panel de información

En RegistrarReservaColectiva

Reserva colectiva registrada exitosamente. Numero de reservas individuales registradas: 5
Operación terminada

Figura 39: Resultado de la interfaz en la ejecución de RF7

45	45	39	3232323232	14 01/03/23	15/03/23	9000000 Finalizada	8	(null)
46	46	40	3333333333	4 15/09/22	19/09/22	2000000 Finalizada	3	(null)
47	166	4	1212121212	28 01/02/23	28/02/23	2520000 Activa	1	165
48	167	6	1212121212	28 01/02/23	28/02/23	1600000 Activa	1	165
49	168	8	1212121212	28 01/02/23	28/02/23	1960000 Activa	1	165
50	169	7	1212121212	28 01/02/23	28/02/23	1120000 Activa	1	165
51	170	10	1212121212	28 01/02/23	28/02/23	5600000 Activa	1	165

Figura 40: El estado final de la base de datos después de ejecutar RF7

Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índices	SQL	
	Ordenar...		Filtrar:										Acciones...
IDRESERVA	IDALOJAMIENTO	IDCLIENTE	DURACION	FECHAINICIO	FECHAFINAL	COSTOTOTAL	ESTADO	NUMPERSONAS	IDRESERVACOLECTIVA				
1	1	1	1111111111	5 02/12/23	07/12/23	1000000	Activa	2		1			
2	2	1	1212121212	10 02/01/23	12/01/23	2000000	Finalizada	2		1			
3	3	2	1313131313	10 02/01/23	12/01/23	2000000	Cancelada	4		1			
4	4	2	1414141414	10 02/11/23	12/11/23	2000000	Activa	6		(null)			
5	5	3	1515151515	10 02/01/22	12/01/22	5000000	Finalizada	4		(null)			
6	6	3	1616161616	10 02/11/22	12/11/22	6000000	Finalizada	6		(null)			
7	7	4	1717171717	7 02/05/23	09/05/23	1000000	Activa	2		(null)			
8	8	5	1515151515	7 10/02/23	17/02/23	2500000	Finalizada	2		(null)			
9	9	6	1919191919	5 15/06/22	20/06/22	3000000	Finalizada	3		(null)			
10	10	7	2020202020	10 15/07/23	25/07/23	4500000	Cancelada	5		(null)			

Figura 41: El estado inicial de la base de datos antes de ejecutar RF8

Panel de información

En cancelarReservaColectiva

Reserva colectiva cancelada exitosamente. Numero de reservas individuales canceladas: 3
Operación terminada

Figura 42: Resultado de la interfaz en la ejecución de RF8

Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índices	SQL	
	Ordenar...		Filtrar:										Acciones...
IDRESERVA	IDALOJAMIENTO	IDCLIENTE	DURACION	FECHAINICIO	FECHAFINAL	COSTOTOTAL	ESTADO	NUMPERSONAS	IDRESERVACOLECTIVA				
1	1	1	1111111111	5 02/12/23	07/12/23	1000000	Cancelada	2		1			
2	2	1	1212121212	10 02/01/23	12/01/23	2000000	Cancelada	2		1			
3	3	2	1313131313	10 02/01/23	12/01/23	2000000	Cancelada	4		1			
4	4	2	1414141414	10 02/11/23	12/11/23	2000000	Activa	6		(null)			
5	5	3	1515151515	10 02/01/22	12/01/22	5000000	Finalizada	4		(null)			
6	6	3	1616161616	10 02/11/22	12/11/22	6000000	Finalizada	6		(null)			
7	7	4	1717171717	7 02/05/23	09/05/23	1000000	Activa	2		(null)			
8	8	5	1515151515	7 10/02/23	17/02/23	2500000	Finalizada	2		(null)			
9	9	6	1919191919	5 15/06/22	20/06/22	3000000	Finalizada	3		(null)			
10	10	7	2020202020	10 15/07/23	25/07/23	4500000	Cancelada	5		(null)			

Figura 43: El estado final de la base de datos después de ejecutar RF8

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL										
Ordenar... Filtrar:										
IDALOJAMIENTO	NOMBRE	CAPACIDAD	UBICACION	TAMANO	PRECIONOCHE	OCUPACIONACTUAL	NUMRESERVAS	IDOPERADOR	ESTADO	TIPO
1	1 Cielo Azul	20	Sur	100	80000	3	1	1	Desabilitado	ViviendaUniversitaria
2	2 Oasis Tropical	1	Este	1	100000	1	2	2	Habilitado	ViviendaUniversitaria
3	3 Casa del Mar	4	Norte	30	50000	2	3	3	Habilitado	ViviendaUniversitaria
4	4 El Faro	10	Sur	70	90000	6	4	4	Habilitado	HabitacionVivienda
5	5 Villa Serena	30	Este	150	120000	15	5	5	Habilitado	HabitacionVivienda
6	6 La Perla del Caribe	5	Oeste	40	60000	4	6	6	Habilitado	HabitacionVivienda
7	7 El Paraíso	2	Norte	20	40000	2	7	7	Habilitado	HabitacionVivienda
8	8 Los Tres Hermanos	8	Este	60	70000	6	8	8	Habilitado	HabitacionVivienda
9	9 La Casa Blanca	12	Sur	90	250000	5	9	9	Habilitado	HabitacionVivienda
10	10 Finca El Encanto	8	Oeste	70	200000	2	10	10	Habilitado	HabitacionVivienda

Figura 47: El estado inicial de la base de datos antes de ejecutar RF10

Panel de información

En rehabilitarAlojamiento

El alojamiento con el id 1 fue rehabilitado de manera exitosa
Operación terminada

Figura 48: Resultado de la interfaz en la ejecución de RF10

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL										
Ordenar... Filtrar:										
IDALOJAMIENTO	NOMBRE	CAPACIDAD	UBICACION	TAMANO	PRECIONOCHE	OCUPACIONACTUAL	NUMRESERVAS	IDOPERADOR	ESTADO	TIPO
1	1 Cielo Azul	20	Sur	100	80000	3	1	1	Habilitado	ViviendaUniversitaria
2	2 Oasis Tropical	1	Este	1	100000	1	2	2	Habilitado	ViviendaUniversitaria
3	3 Casa del Mar	4	Norte	30	50000	2	3	3	Habilitado	ViviendaUniversitaria
4	4 El Faro	10	Sur	70	90000	6	4	4	Habilitado	HabitacionVivienda
5	5 Villa Serena	30	Este	150	120000	15	5	5	Habilitado	HabitacionVivienda
6	6 La Perla del Caribe	5	Oeste	40	60000	4	6	6	Habilitado	HabitacionVivienda
7	7 El Paraíso	2	Norte	20	40000	2	7	7	Habilitado	HabitacionVivienda
8	8 Los Tres Hermanos	8	Este	60	70000	6	8	8	Habilitado	HabitacionVivienda
9	9 La Casa Blanca	12	Sur	90	250000	5	9	9	Habilitado	HabitacionVivienda
10	10 Finca El Encanto	8	Oeste	70	200000	2	10	10	Habilitado	HabitacionVivienda

Figura 49: El estado final de la base de datos después de ejecutar RF10

Listando las principales características por tipo de usuario

MIEMBROS ACTIVOS

- [Tipo de usuario: Estudiante, Numero de reservas: 10, Total de noches reservadas: 37000000, Total de dinero pagado: 37000000 COP]
- [Tipo de usuario: Profesor visitante, Numero de reservas: 6, Total de noches reservadas: 22000000, Total de dinero pagado: 22000000 COP]
- [Tipo de usuario: Empleado, Numero de reservas: 8, Total de noches reservadas: 17000000, Total de dinero pagado: 17000000 COP]
- [Tipo de usuario: Profesor titular, Numero de reservas: 2, Total de noches reservadas: 5000000, Total de dinero pagado: 5000000 COP]

MIEMBROS SECUNDARIOS

- [Tipo de usuario: Padre, Numero de reservas: 4, Total de noches reservadas: 17000000, Total de dinero pagado: 17000000 COP]
- [Tipo de usuario: Egresado, Numero de reservas: 10, Total de noches reservadas: 58000000, Total de dinero pagado: 58000000 COP]
- [Tipo de usuario: Invitado, Numero de reservas: 6, Total de noches reservadas: 41000000, Total de dinero pagado: 41000000 COP]

Operación terminada

Figura 50: El resultado obtenido tras ejecutar RFC5

Panel de información

Listando las principales características de las reservas a nombre de ese cliente

La lista de las principales características de las reservas asociadas al cliente son:

- [ID del Cliente: 1111111111, Numero de reservas a su nombre: 2, Numero de noches reservadas: 10, Dinero pagado: 2000000 COP]

Operación terminada

Figura 51: El resultado obtenido tras ejecutar RFC6

Panel de información

Listando los clientes frecuentes que cumplen las condiciones necesarias para que sean considerados como tal
Los clientes frecuentes del alojamiento dado son:
1. Cliente [idCliente=3030303030, tipoidentificacion=TI, nombreCliente=Miguel Ramos, fechaNacimiento=null]

Operación terminada

Figura 53: El resultado obtenido tras ejecutar RFC8

Los alojamientos con poca demanda que no han sido reservados hace mas de un mes son:

- Alojamiento [idAlojamiento=1, nombre=Cielo Azul, capacidad=20, ubicacion=Sur, tamaño=100, precioNoche=80000, ocupacionActual=3, numReservas=1, idOperador=1, estado=Deshabilitado, tipo=ViviendaUniversitaria]
- Alojamiento [idAlojamiento=2, nombre=Oasis Tropical, capacidad=1, ubicacion=Este, tamaño=1, precioNoche=100000, ocupacionActual=1, numReservas=2, idOperador=2, estado=Habilitado, tipo=ViviendaUniversitaria]
- Alojamiento [idAlojamiento=3, nombre=Casa del Mar, capacidad=4, ubicacion=Norte, tamaño=30, precioNoche=50000, ocupacionActual=2, numReservas=3, idOperador=3, estado=Habilitado, tipo=ViviendaUniversitaria]
- Alojamiento [idAlojamiento=5, nombre=Villa Serena, capacidad=30, ubicacion=Este, tamaño=150, precioNoche=120000, ocupacionActual=15, numReservas=5, idOperador=5, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=6, nombre=La Perla del Caribe, capacidad=5, ubicacion=Oeste, tamaño=40, precioNoche=60000, ocupacionActual=4, numReservas=6, idOperador=6, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=7, nombre=El Paraíso, capacidad=2, ubicacion=Norte, tamaño=20, precioNoche=40000, ocupacionActual=2, numReservas=7, idOperador=7, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=8, nombre=Los Tres Hermanos, capacidad=8, ubicacion=Este, tamaño=60, precioNoche=70000, ocupacionActual=6, numReservas=8, idOperador=8, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=9, nombre=La Casa Blanca, capacidad=12, ubicacion=Sur, tamaño=90, precioNoche=250000, ocupacionActual=5, numReservas=9, idOperador=9, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=10, nombre=Finca El Encanto, capacidad=8, ubicacion=Oeste, tamaño=70, precioNoche=200000, ocupacionActual=2, numReservas=10, idOperador=10, estado=Habilitado, tipo=HabitacionVivienda]
- Alojamiento [idAlojamiento=11, nombre=Hotel La Estancia, capacidad=14, ubicacion=Norte, tamaño=100, precioNoche=300000, ocupacionActual=8, numReservas=11, idOperador=11, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=12, nombre=Hacienda San Miguel, capacidad=4, ubicacion=Este, tamaño=60, precioNoche=150000, ocupacionActual=1, numReservas=12, idOperador=12, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=13, nombre=Rancho Las Palmas, capacidad=40, ubicacion=Norte, tamaño=30, precioNoche=80000, ocupacionActual=4, numReservas=13, idOperador=13, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=14, nombre=Posada Los Amigos, capacidad=6, ubicacion=Sur, tamaño=80, precioNoche=120000, ocupacionActual=3, numReservas=14, idOperador=14, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=15, nombre=Posada La Hacienda, capacidad=18, ubicacion=Este, tamaño=120, precioNoche=350000, ocupacionActual=9, numReservas=15, idOperador=15, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=16, nombre=Hostal San Pedro, capacidad=10, ubicacion=Oeste, tamaño=90, precioNoche=180000, ocupacionActual=5, numReservas=16, idOperador=16, estado=Habilitado, tipo=Apartamento]
- Alojamiento [idAlojamiento=17, nombre=El Refugio, capacidad=14, ubicacion=Norte, tamaño=100, precioNoche=280000, ocupacionActual=6, numReservas=17, idOperador=17, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=18, nombre=La Cabaña del Bosque, capacidad=22, ubicacion=Sur, tamaño=130, precioNoche=450000, ocupacionActual=12, numReservas=18, idOperador=18, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=19, nombre=Casa de Campo, capacidad=6, ubicacion=Norte, tamaño=80, precioNoche=60000, ocupacionActual=2, numReservas=19, idOperador=19, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=20, nombre=La Quinta del Sol, capacidad=8, ubicacion=Este, tamaño=70, precioNoche=75000, ocupacionActual=1, numReservas=20, idOperador=20, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=21, nombre=Betelgeuse, capacidad=10, ubicacion=Oeste, tamaño=85, precioNoche=90000, ocupacionActual=4, numReservas=21, idOperador=21, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=22, nombre=Alpha Centauri, capacidad=2, ubicacion=Norte, tamaño=30, precioNoche=50000, ocupacionActual=2, numReservas=22, idOperador=22, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=23, nombre=Vega, capacidad=25, ubicacion=Sur, tamaño=40, precioNoche=35000, ocupacionActual=10, numReservas=23, idOperador=23, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=25, nombre=Proxima Centauri, capacidad=1, ubicacion=Oeste, tamaño=20, precioNoche=40000, ocupacionActual=1, numReservas=25, idOperador=25, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=26, nombre=Cassiopeia, capacidad=12, ubicacion=Norte, tamaño=95, precioNoche=85000, ocupacionActual=2, numReservas=26, idOperador=26, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=27, nombre=Aldebaran, capacidad=8, ubicacion=Este, tamaño=70, precioNoche=70000, ocupacionActual=1, numReservas=27, idOperador=27, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=28, nombre=Plutón, capacidad=2, ubicacion=Norte, tamaño=30, precioNoche=90000, ocupacionActual=0, numReservas=28, idOperador=28, estado=Habilitado, tipo=HabitacionHotel]
- Alojamiento [idAlojamiento=29, nombre=Venus, capacidad=8, ubicacion=Sur, tamaño=60, precioNoche=110000, ocupacionActual=2, numReservas=29, idOperador=29, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=30, nombre=Mercurio, capacidad=3, ubicacion=Este, tamaño=35, precioNoche=95000, ocupacionActual=1, numReservas=30, idOperador=30, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=31, nombre=Júpiter, capacidad=10, ubicacion=Norte, tamaño=70, precioNoche=150000, ocupacionActual=5, numReservas=3, idOperador=1, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=32, nombre=Tierra, capacidad=4, ubicacion=Este, tamaño=40, precioNoche=100000, ocupacionActual=1, numReservas=1, idOperador=2, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=33, nombre=Mars, capacidad=6, ubicacion=Oeste, tamaño=50, precioNoche=130000, ocupacionActual=5, numReservas=0, idOperador=3, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=34, nombre=Urano, capacidad=7, ubicacion=Sur, tamaño=55, precioNoche=140000, ocupacionActual=2, numReservas=1, idOperador=4, estado=Habilitado, tipo=Hostal]
- Alojamiento [idAlojamiento=35, nombre=Hotel Los Pinos, capacidad=11, ubicacion=Norte, tamaño=75, precioNoche=155000, ocupacionActual=3, numReservas=2, idOperador=5, estado=Habilitado, tipo=ViviendaTemporal]
- Alojamiento [idAlojamiento=36, nombre=Finca La Esperanza, capacidad=8, ubicacion=Este, tamaño=75, precioNoche=85000, ocupacionActual=2, numReservas=1, idOperador=6, estado=Habilitado, tipo=ViviendaTemporal]
- Alojamiento [idAlojamiento=37, nombre=Mercurio, capacidad=4, ubicacion=Centro, tamaño=40, precioNoche=60000, ocupacionActual=1, numReservas=0, idOperador=7, estado=Habilitado, tipo=ViviendaTemporal]
- Alojamiento [idAlojamiento=38, nombre=Urano, capacidad=6, ubicacion=Norte, tamaño=60, precioNoche=70000, ocupacionActual=3, numReservas=1, idOperador=8, estado=Habilitado, tipo=ViviendaTemporal]
- Alojamiento [idAlojamiento=39, nombre=El Castillo, capacidad=10, ubicacion=Sur, tamaño=80, precioNoche=90000, ocupacionActual=2, numReservas=0, idOperador=9, estado=Habilitado, tipo=ViviendaTemporal]
- Alojamiento [idAlojamiento=40, nombre=Villa La Playa, capacidad=90, ubicacion=Oeste, tamaño=110, precioNoche=100000, ocupacionActual=50, numReservas=1, idOperador=3, estado=Habilitado, tipo=ViviendaTemporal]

Figura 54: El resultado obtenido tras ejecutar RFC9

6.2 Pruebas de transacción NO exitosa

40	40	34	2727272727	10 01/05/22	11/05/22	5000000 Finalizada	6	(null)
41	41	35	2828282828	5 01/09/23	06/09/23	2500000 Activa	2	(null)
42	42	36	2929292929	3 10/03/23	13/03/23	1500000 Finalizada	2	(null)
43	43	37	3030303030	32 01/02/23	05/03/23	5000000 Finalizada	4	(null)
44	44	38	3131313131	5 20/06/22	25/06/22	2500000 Finalizada	2	(null)
45	45	39	3232323232	14 01/03/23	15/03/23	9000000 Finalizada	8	(null)
46	46	40	3333333333	4 15/09/22	19/09/22	2000000 Finalizada	3	(null)
47	166	4	1212121212	28 01/02/23	28/02/23	2520000 Activa	1	165
48	167	6	1212121212	28 01/02/23	28/02/23	1680000 Activa	1	165
49	168	8	1212121212	28 01/02/23	28/02/23	1960000 Activa	1	165
50	169	7	1212121212	28 01/02/23	28/02/23	1120000 Activa	1	165
51	170	10	1212121212	28 01/02/23	28/02/23	5600000 Activa	1	165

Figura 55: El estado inicial de la base de datos antes de ejecutar RF7

```
***** Error en la ejecución
No se pudo registrar la reserva colectiva porque no hay suficiente disponibilidad del tipo de habitacion deseada,
Revise datanucleus.log y parranderos.log para más detalles
```

Figura 56: Resultado de la interfaz en la ejecución de RF7

40	40	34	2727272727	10/01/05/22	11/05/22	5000000	Finalizada	6	(null)
41	41	35	2828282828	5/01/09/23	06/09/23	2500000	Activa	2	(null)
42	42	36	2929292929	3/10/03/23	13/03/23	1500000	Finalizada	2	(null)
43	43	37	3030303030	32/01/02/23	05/03/23	5000000	Finalizada	4	(null)
44	44	38	3131313131	5/20/06/22	25/06/22	2500000	Finalizada	2	(null)
45	45	39	3232323232	14/01/03/23	15/03/23	9000000	Finalizada	8	(null)
46	46	40	3333333333	4/15/09/22	19/09/22	2000000	Finalizada	3	(null)
47	166	4	1212121212	28/01/02/23	28/02/23	2520000	Activa	1	165
48	167	6	1212121212	28/01/02/23	28/02/23	1680000	Activa	1	165
49	168	8	1212121212	28/01/02/23	28/02/23	1960000	Activa	1	165
50	169	7	1212121212	28/01/02/23	28/02/23	1120000	Activa	1	165
51	170	10	1212121212	28/01/02/23	28/02/23	5600000	Activa	1	165

Figura 57: El estado final de la base de datos después de ejecutar RF7

Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índices	SQL
Ordenar... Filtrar: Acciones...												
IDRESERVA	IDALOJAMIENTO	IDCLIENTE	DURACION	FECHAINICIO	FECHAFINAL	COSTOTOTAL	ESTADO	NUMPERSONAS	IDRESERVACOLECTIVA			
1	1	1	1111111111	5/02/12/23	07/12/23	1000000	Cancelada	2		1		
2	2	1	1212121212	10/02/01/23	12/01/23	2000000	Cancelada	2		1		
3	3	2	1313131313	10/02/01/23	12/01/23	2000000	Cancelada	4		1		
4	4	2	1414141414	10/02/11/23	12/11/23	2000000	Activa	6		(null)		
5	5	3	1515151515	10/02/01/22	12/01/22	5000000	Finalizada	4		(null)		
6	6	3	1616161616	10/02/11/22	12/11/22	6000000	Finalizada	6		(null)		
7	7	4	1717171717	7/02/05/23	09/05/23	1000000	Activa	2		(null)		
8	8	5	1515151515	7/10/02/23	17/02/23	2500000	Finalizada	2		(null)		
9	9	6	1919191919	5/15/06/22	20/06/22	3000000	Finalizada	3		(null)		
10	10	7	2020202020	10/15/07/23	25/07/23	4500000	Cancelada	5		(null)		

Figura 58: El estado inicial de la base de datos antes de ejecutar RF8

Panel de información

***** Error en la ejecución

No se pudo cancelar la reserva colectiva, debido a que no existe ninguna con el id suministrado,

Revise datanucleus.log y parranderos.log para más detalles

Figura 59: Resultado de la interfaz en la ejecución de RF8

Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índices	SQL
Ordenar... Filtrar: Acciones...												
IDRESERVA	IDALOJAMIENTO	IDCLIENTE	DURACION	FECHAINICIO	FECHAFINAL	COSTOTOTAL	ESTADO	NUMPERSONAS	IDRESERVACOLECTIVA			
1	1	1	1111111111	5/02/12/23	07/12/23	1000000	Cancelada	2		1		
2	2	1	1212121212	10/02/01/23	12/01/23	2000000	Cancelada	2		1		
3	3	2	1313131313	10/02/01/23	12/01/23	2000000	Cancelada	4		1		
4	4	2	1414141414	10/02/11/23	12/11/23	2000000	Activa	6		(null)		
5	5	3	1515151515	10/02/01/22	12/01/22	5000000	Finalizada	4		(null)		
6	6	3	1616161616	10/02/11/22	12/11/22	6000000	Finalizada	6		(null)		
7	7	4	1717171717	7/02/05/23	09/05/23	1000000	Activa	2		(null)		
8	8	5	1515151515	7/10/02/23	17/02/23	2500000	Finalizada	2		(null)		
9	9	6	1919191919	5/15/06/22	20/06/22	3000000	Finalizada	3		(null)		
10	10	7	2020202020	10/15/07/23	25/07/23	4500000	Cancelada	5		(null)		

Figura 60: El estado final de la base de datos después de ejecutar RF8

Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índices	SQL
Ordenar... Filtrar: Accione												
IDALOJAMIENTO	NOMBRE	CAPACIDAD	UBICACION	TAMANO	PRECIONOCHE	OCCUPACIONACTUAL	NUMRESERVAS	IDOPERADOR	ESTADO	TIPO		
1	1Cielo Azul	20 Sur	100	80000	3	1	1	Habilitado	Vivienda	Universitaria		
2	2Oasis Tropical	1 Este	1	100000	1	2	2	Habilitado	Vivienda	Universitaria		
3	3Casa del Mar	4 Norte	30	50000	2	3	3	Habilitado	Vivienda	Universitaria		
4	4El Faro	10 Sur	70	90000	6	4	4	Habilitado	Habitacion	Vivienda		
5	5Villa Serena	30 Este	150	120000	15	5	5	Habilitado	Habitacion	Vivienda		
6	6La Perla del Caribe	5 Oeste	40	60000	4	6	6	Habilitado	Habitacion	Vivienda		
7	7El Paraíso	2 Norte	20	40000	2	7	7	Habilitado	Habitacion	Vivienda		
8	8Los Tres Hermanos	8 Este	60	70000	6	8	8	Habilitado	Habitacion	Vivienda		
9	9La Casa Blanca	12 Sur	90	250000	5	9	9	Habilitado	Habitacion	Vivienda		
10	10Finca El Encanto	8 Oeste	70	200000	2	10	10	Habilitado	Habitacion	Vivienda		

Figura 64: El estado inicial de la base de datos antes de ejecutar RF10

Panel de información

En rehabilitarAlojamiento

El alojamiento con el id 2 ya se encontraba habilitado dentro del sistema

Figura 65: Resultado de la interfaz en la ejecución de RF10

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL									
Ordenar... Filtrar									
IDALOJAMIENTO	NOMBRE	CAPACIDAD	UBICACION	TAMANO	PRECIONOCHE	OCUPACIONACTUAL	NUMRESERVAS	IDOPERADOR	ESTADO TIPO
1	1Cielo Azul	20 Sur	100	80000	3	1	1	Habilitado Vivienda	Universitaria
2	2Oasis Tropical	1 Este	1	100000	1	2	2	Habilitado Vivienda	Universitaria
3	3Casa del Mar	4 Norte	30	50000	2	3	3	Habilitado Vivienda	Universitaria
4	4El Faro	10 Sur	70	90000	6	4	4	Habilitado Habitacion	Vivienda
5	5Villa Serena	30 Este	150	120000	15	5	5	Habilitado Habitacion	Vivienda
6	6La Perla del Caribe	5 Oeste	40	60000	4	6	6	Habilitado Habitacion	Vivienda
7	7El Paraíso	2 Norte	20	40000	2	7	7	Habilitado Habitacion	Vivienda
8	8Los Tres Hermanos	8 Este	60	70000	6	8	8	Habilitado Habitacion	Vivienda
9	9La Casa Blanca	12 Sur	90	250000	5	9	9	Habilitado Habitacion	Vivienda
10	10Finca El Encanto	8 Oeste	70	200000	2	10	10	Habilitado Habitacion	Vivienda

Figura 66: El estado final de la base de datos después de ejecutar RF10

7. Bibliografías

- [1] SILBERSCHATZ, Abraham, KORTH, Henry F., SUDARSHAN, S. “*Database system concepts*”. Seventh edition. New York, NY: McGraw-Hill, [2020].
- [2] LEWIS, Philip, BERNSTEIN, Arthur, KIFER, Michael. "*Database Systems- An ApplicationOriented Approach*". Second Edition. Addison-Wesley, 2006.
- [3] GARCIA-MOLINA, Hector, ULLMAN, Jeffrey, WIDOM, Jennifer. "*Database Systems: The complete Book*". 2nd. Edition. Prentice Hall, 2009.
- [4] BAGUI, S., & EARP, R. (2003). Database Design Using Entity-Relationship Diagrams. Auerbach Publications, Boca Raton, Florida: CRC Press.