Manuela Lizcano
Juan Diego Lozano
Mariana Pineda

PROYECTO SISTEMAS TRANSACCIONALES
ENTREGA 3

1. **Análisis de Carga**
    a. Identificación de entidades y atributos

Entidad	Atributos
Servicio	Id Nombre Tipo Precio
Consumo	Id Habitación Fecha Consumo Servicio Entrada
Salida	Id Fecha Consumo
Tipo de Habitación	Id Nombre Dotación
Habitación	Numero Habitación Tipo Habitación
Reserva	Id Reserva Número de Personas Fecha de Entrada Fecha de Salida Habitación Usuario
Entrada	Id Entrada Reserva

    b. Cuantificación de entidades

Entidad	Numero de datos
Servicio	35
Consumo	250.000
Salida	50.000
tipoHabitación	3
Habitación	200
Reserva	50.000
Entrada	50.000

El que dice en el enunciado

    c. Análisis de operaciones de lectura y escritura y cuantificación

Entidad	Operación	Información Necesitada	Tipo	Rate
Servicio	Obtener los servicios	Servicios junto con su tipo y precio.	Read	2/semana
Servicio	Crear/Actualizar un servicio	Nuevo servicio con sus atributos.	Write	2/semana
Consumo	Obtener los consumos	Consumo, junto con la habitación que lo pidió, y su servicio correspondiente.	Read	250/día
Consumo	Crear/Actualizar un consumo	Nuevo consumo con sus atributos y su respectivo servicio.	Write	500/día
Salidas	Obtener las salidas	Información de la salida	Read	50/dia
Salidas	Crear/Actualizar salida	Información de la nueva salida, y su habitación y consumo correspondientes.	Write	100/día
tipoHabitacion	Obtener Tipos	Nombre del tipo de habitación	Read	1/semana
tipoHabitacion	Crear/actualizar tipos	Nombre y dotación para el tipo	Write	1/Semana
Habitación	Obtener las habitaciones	Habitación, junto con el tipo de información	Read	2/Semana
Habitación	Crear/Actualizar/Modificar una habitación	Nueva habitación y sus atributos.	Write	2/Semana
Reserva	Obtener las reservas	Numero de reserva	Read	50/día
Reserva	Crear/Modificar reservas	Nueva reserva	Write	100/día
Entrada	Obtener la entrada	Número del id de entrada	Read	50/día
Entrada	Crear/modificar	Nueva información de entrada	Write	100/día

Entities	Operation	Information Needed	Type	Rate
Books*	Fetch book details	Book details + rating	Read	1000/sec
Authors, Books*	Fetch an author and their books	Book titles + author details	Read	50/sec
Print Books	Fetch printed book titles where the stock level has fallen below 50	Book details + stock level	Read	2/day
Books*	Add/update book	Book details + stock level	Write	10/hour
Print Books	Sell copy of printed book	Stock level	Write	5/sec
Reviews	Fetch 10 reviews for a book	Reviews + reviewer rating	Read	200/sec
Reviews	Add review	Review + book rating	Write	50/sec
Users	Fetch user details	User details	Read	5/min
Users	Add/update user	User details	Write	1/sec

*eBooks, Audiobooks, and Printed Books

2. Descripción de las entidades de datos

a. Descripción y Cardinalidad

Entidad	Descripción	Relaciones
Servicio	En esta clase se tienen los servicios que ofrece el hotel, junto con su tipo y el costo que este tiene para sus clientes.	Esta clase se relaciona con la clase consumo, ya que cada consumo cuenta con un servicio, de este servicio es importante saber su precio y su costo cuando se consulta un consumo. Esta relación es de uno a muchos ya que un consumo tiene solo un servicio, pero un servicio tiene muchos consumos.
Consumo	En esta clase se tiene el consumo de un cliente en el hotel, este va asociado a una habitación y a una entrada especifica de un cliente. De igual manera, la información más importante es el servicio de este respectivo consumo, esto ya que de acá sale el costo y demás información del servicio.	Esta clase se relaciona con la clase servicio ya que un consumo cuenta con un servicio, esta relación es de uno a muchos. También se relaciona con la clase habitación ya que cada consumo se asocia con una habitación, lo que quiere decir que es una relación uno a muchos. Y con la clase entrada ya que a cada consumo se le asocia la llegada

		de un cliente, así que también es una relación uno a muchos.
Salida	En esta colección se registra la salida de los huéspedes del hotel junto con los servicios que ha consumido.	Se relaciona con la clase consumos ya que acá es donde se ve cuanto hay que cobrar al cliente al final de su estadía, esta relación es de uno a muchos.
tipoHabitacion	Esta colección contiene la información de todos los tipos de habitación. Dentro de esto se tiene el nombre de la habitación y la dotación de cada tipo.	El tipo de habitación se relaciona con la habitación ya que se encuentra embebida en esta. Esto se debe a su poca lectura y modificación
Habitación	Esta colección contine información acerca de las habitaciones, incluyendo los tipos de habitación y sus características como lo son el tipo, dotación y capacidad.	Esta clase se relación con la clase de tipo de habitación, considerando que en habitación esta embebida el tipo de habitación. Esta relación se debe a que, en el momento de realizar actualizaciones, se deben actualizar las dos clases en conjunto considerando la relación que existe entre ellas.
Reserva	Esta colección contiene información de los usuarios considerando a que cada reserva tiene usuarios asociados. Dentro de esta se tiene información del usuario como su id, nombre, email, usuario y contraseña. Esta información es indispensable para el correcto manejo de los usuarios en el sistema del Hotel de los Andes. Adicionalmente esta clase contiene información como el id de la reserva, el número de personas que se van a alojar con ese respectivo número de reserva, las fechas de entrada y salida del hotel y la habitación en la que se van a hospedar.	Esta clase tiene una relación con la clase de usuarios. Esta relación se construye tomando a la clase de usuario como embebida a la clase de reserva. Esta decisión se debe a que esta se relaciona existe una dependencia entre estas dos clases y esto hace que ambas tengan que ser creadas y actualizadas en el mismo momento. Adicionalmente, en el caso de acceder a la información de los usuarios se debe realizar por medio del id de la reserva con la que están asociados.
Entrada	Esta colección contiene la información relacionada con la entrada de un huésped al hotel. Esta clase tiene una relación con la clase de reservas considerando a que por cada entrada se debe asignar un id de reserva.	Esta clase tiene una relación de referencia con la clase de reservas considerando que ambas tienen una alta cardinalidad lo cual haría que la duplicación de los datos comprometa el manejo del sistema operativo del hotel.

b. Selección del esquema de asociación

Entidad	**Explicación**

Servicio	En esta clase no se embebe ni se hacen referencias, más bien, esta clase es embebida.
Consumo	En esta colección se embebe la clase servicios, esta decisión se toma con base a: 1. La información se necesita junta ya que cuando se quiere consultar un consumo normalmente se va a querer consultar cual es el servicio que se consumió. 2. Los consumos se leen seguido en un día mientras que los servicios se leen cada semana, por lo que es mejor que la clase padre sea consumos. 3. Los consumos pueden creer exponencial en los servicios ya que se tiene un total de 250.000 consumos aproximadamente. 4. Se tendría una mayor duplicación de datos si se embeben los consumos en los servicios. 5. Se tiene una relación uno a muchos, por lo que en cada consumo los servicios no crecerán exponencialmente, incluso solo se tiene un servicio por consumo. Por otra parte, se prefirió una referencia a la habitación esto ya que: 1. No se necesita la demás información de una habitación cuando se quiere consultar un consumo. 2. Se estaría guardando información innecesaria. 3. Solo se necesita el numero de la habitación que en este caso es el id.
Salida	En esta colección se decidido referenciar los consumos, esto más por el espacio que se consumiría y por la alta cantidad de datos que se tienen en ambas colecciones. Se sabe que si se embeben los consumos, estos pueden crecer sin control ya que un cliente puede consumir cuantos consumos desee mientras esta en el hotel.
tipoHabitacion	Esta clase está embebida en la habitación, esto se debe a que cada vez que se actualiza un tipo de habitación se debe a que existe un cambio en la clase de habitación, por esta razón ambas deben ser actualizadas en el mismo momento. La duplicación de los datos no comprometerá el sistema considerando que la relación de tipo de habitación no se accede con bastante frecuencia.
Habitación	Esta clase embebe a la clase de tipo de habitación, esto se debe principalmente que esta información es usualmente utilizada junta con el objetivo de poder asignar correctamente las habitaciones a los huéspedes según sus solicitudes. De igual forma, se toma la decisión de embeber a tipos de habitaciones considerando su pequeño tamaño y las pocas veces a las que es necesario acceder a esta información hace que este esquema sea el indicado.
Reserva	La clase de reserva embebe la clase de usuario, esto se debe principalmente a que ambas clases se deben actualizar en el mismo momento. Esto hace que el acceso de ambas clases es pertinente tenerlo en un esquema embebido.
Entrada	La clase de entrada referencia a las reservas, esto se debe a que considerando los esquemas del Hotel de los Andes se debe disponer

	de esta información de forma conjunta, sin embargo, considerando el tamaño de ambas clases y como crecen, este esquema es de referenciado.

Si se embebió o referencio y porque

c. Descripción grafica

Entidad	Representación

Servicio	No se tienen relaciones
	`_id: ObjectId('6569077142c56b31372e0287')`
	`nombre: "Chocolaterapia"`
	`tipo: "Spa"`
	`precio: 100`
	`_class: "com.example.demo.modelo.Servicio"`

Consumo	Servicio embebido
	`_id: ObjectId('65692a03dd2ee778049d20b6')`
	`habitacion: 300`
	`fechaConsumo: "10/03/2023"`
	`servicio: Array (1)`
	`0: Object`
	`nombre: "Chocolaterapia"`
	`tipo: "Spa"`
	`precio: 200`
	`_class: "com.example.demo.modelo.Consumo"`

Salida	Consumo referenciado
	`_id: ObjectId('656a85737b6c0407a08d704d')`
	`fecha: "10-09-2023"`
	`consumos: Array (1)`
	`0: DBRef('consumos', '656a85737b6c0407a08d704c')`
	`_class: "com.example.demo.modelo.Salida"`

tipoHabitacion	`_id: ObjectId('656b3484f1d8b675289e4df2')`
	`nombre: "hh"`
	`dotacion: Array (1)`
	`_class: "com.example.demo.modelo.TipoHabitacion"`

Habitación	Embebe a tipoHabitacion
	`_id: 300`
	`tipoHabitacion: Array (1)`
	`0: Object`
	`nombre: "Suite"`
	`dotacion: Array (1)`
	`capacidad: 3`
	`_class: "com.example.demo.modelo.Habitacion"`

Reserva	Embebe a usuario

<table>
<tr><td></td><td>

<pre>
_id: 12
usuario: Array (1)
  0: Object
    _id: 2345
    nombre: "Manuela Lizcano"
    email: "manulizc23@gmail.com"
    usuario: "manu23"
    contrasena: "sistrans"
  _class: "com.example.demo.modelo.Reserva"
</pre>

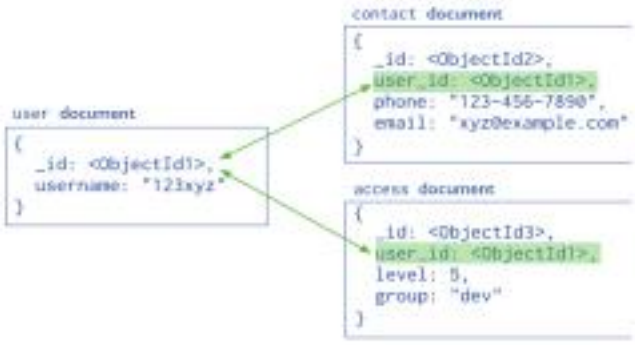
</td></tr>
<tr><td>Entrada</td><td>Referencia a reservas

<pre>
_id: ObjectId('656d157d05317140c6978b5b')
reserva: Array (1)
  0: DBRef('reservas', '12')
_class: "com.example.demo.modelo.Entrada"
</pre>

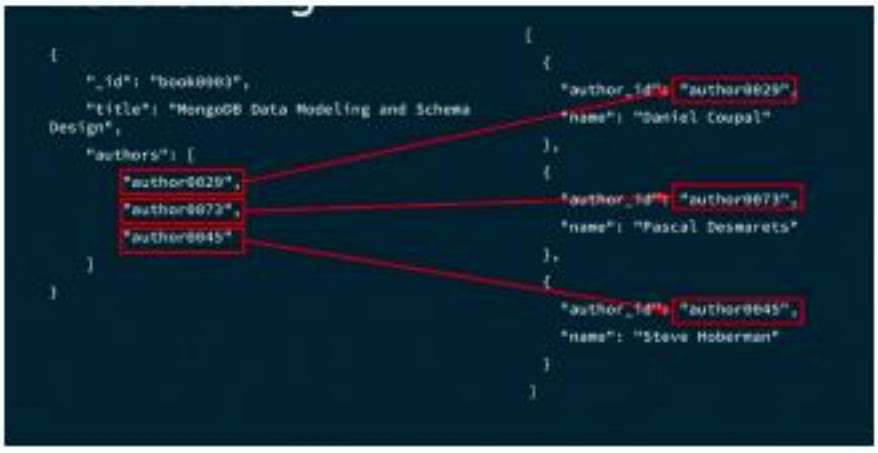
</td></tr>
</table>

ANEXO D – EJEMPLOS GRÁFICOS DE ESQUEMAS EMBEBIDOS Y REFERENCIADOS

Ejemplo de esquema embebido:

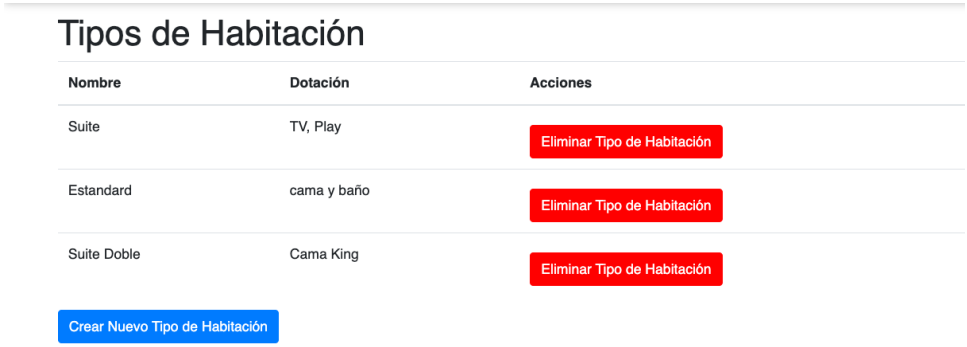


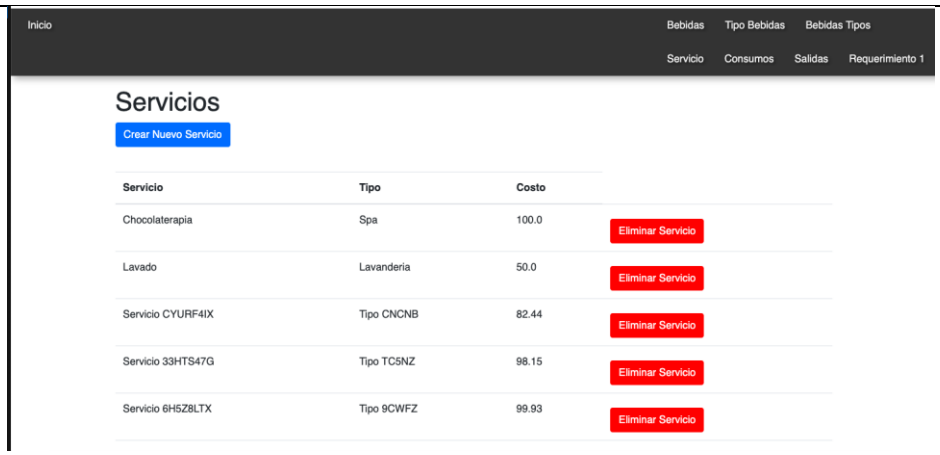
Ejemplo de esquema referenciado:

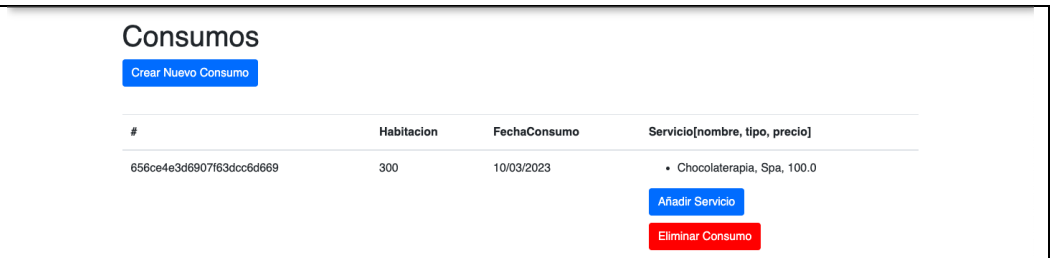
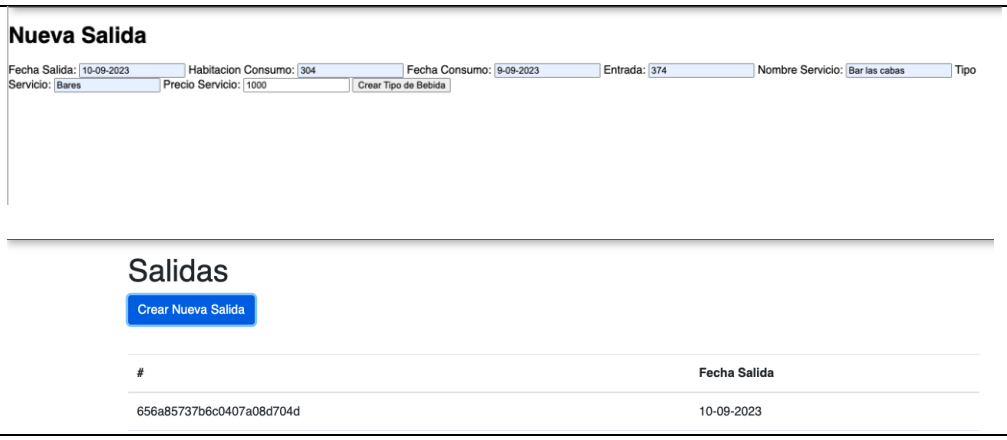


3. Escenarios de prueba

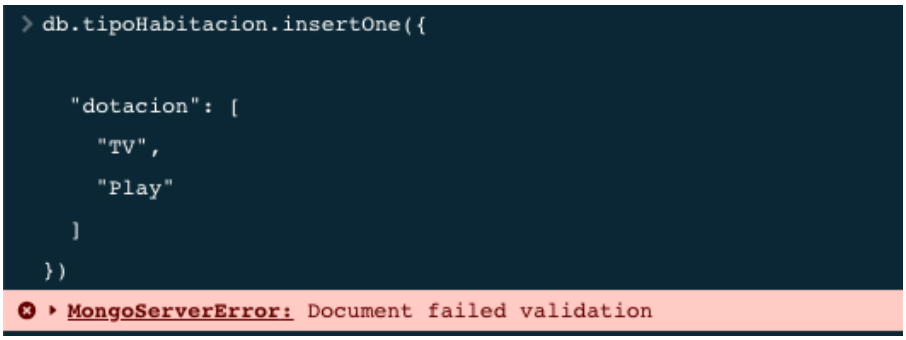
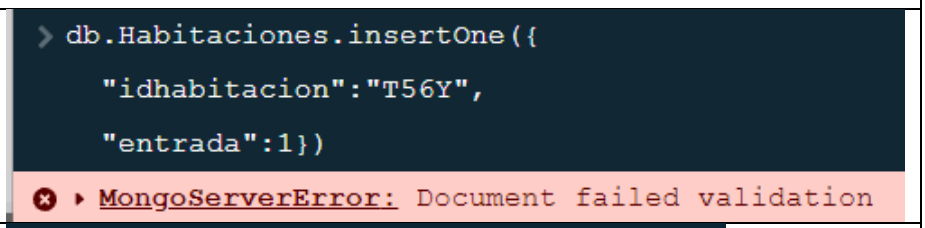
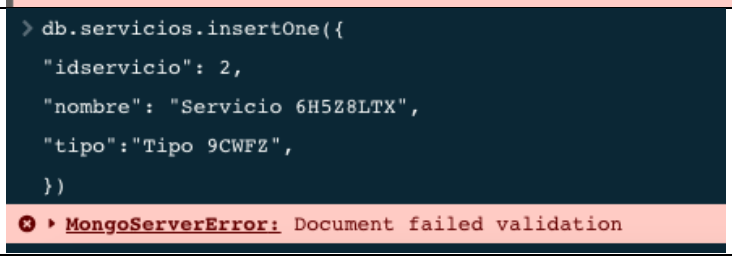
a. Interfaz y CRUD

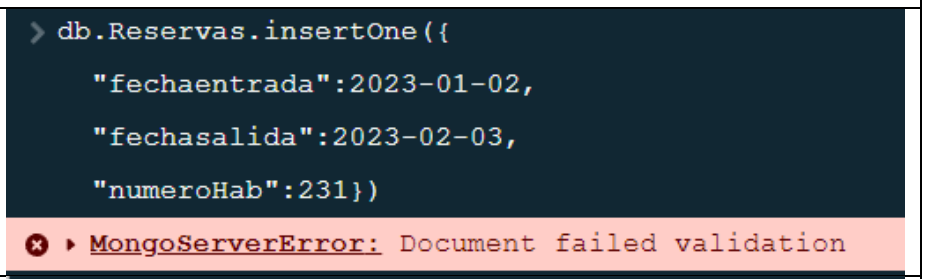
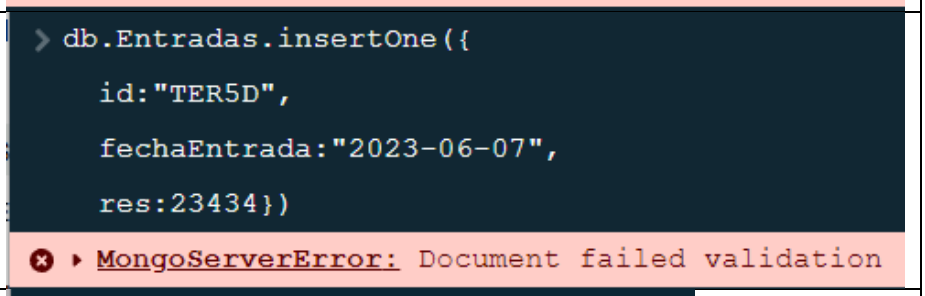
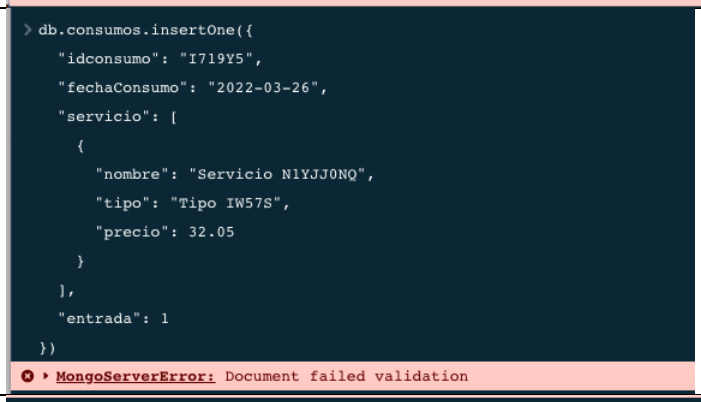
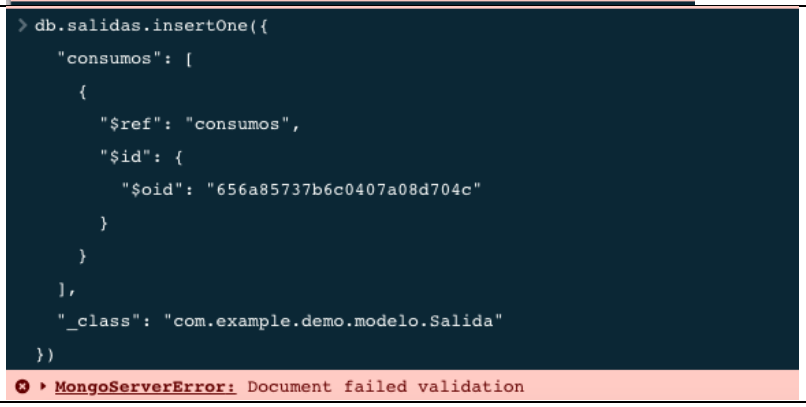
RF1	
RF2	
RF3	

	 Inicio Bebidas Tipo Bebidas Bebidas Tipos Servicio Consumos Salidas Requerimiento 1 Servicios Crear Nuevo Servicio Servicio Tipo Costo Chocolaterapia Spa 100.0 Eliminar Servicio Lavado Lavanderia 50.0 Eliminar Servicio Servicio CYURF4IX Tipo CNCNB 82.44 Eliminar Servicio Servicio 33HTS47G Tipo TC5NZ 98.15 Eliminar Servicio Servicio 6H5Z8LTX Tipo 9CWFZ 99.93 Eliminar Servicio
RF4	 Reservas IdReserva Personas FechaEntrada FechaSalida Habitacion Usuario[IdUsuario, Nombre, Email, Usuario, Contraseña] 12 • 2345, Manuela Lizcano, manulizc23@gmail.com, manu23, sistrans Eliminar Reserva Crear Nueva Reserva Nueva Reserva Id Reserva: ID Usuario: Nombre: Email: Usuario: Contraseña: Crear Reserva
RF5	 Entradas @Id private Integer identrada; @DBRef private List reserva; Id Entrada Reservas [Object Id] Crear Entrada Nuevo Entrada IdReserva: Personas: fechaEntrada: fechaSalida: habitacion: Crear Entrada
RF6	 Nuevo Consumo Habitacion: 300 Fecha Consumo: 10/03/2023 Nombre del servicio: Chocolaterapia Tipo: Spa Precio: 100 Crear Consumo

	
RF7	

b. Validación de esquemas

RF1	
RF2	
RF3	

RF4	
RF5	
RF6	
RF7	

c. Requerimientos funcionales de consulta

RF1	

<table>
<tr><td></td><td>

Habitaciones

Lista de Habitaciones

Numero Habitacion	Costo por Habitacion
89	478394
13	471522
86	470574
50	470365
51	468701
41	468046
38	466778
81	466117
40	466109
8	465441

</td></tr>
<tr><td>RF2</td><td>

</td></tr>
<tr><td>RF3</td><td>

Buscar Consumos por Rango de Fechas

Fecha Inicio (formato YYYY-MM-DD): YYYY-MM-DD Fecha Fin (formato YYYY-MM-DD): YYYY-MM-DD Buscar

Resultados

Fecha de Consumo	Nombre del Cliente	Email del Cliente
2022-05-18	Usuario 24209	usuario24209@example.com
2022-03-29	Usuario 22086	usuario22086@example.com
2022-09-26	Usuario 24995	usuario24995@example.com
2022-10-19	Usuario 44884	usuario44884@example.com
2022-06-05	Usuario 21524	usuario21524@example.com
2023-06-08	Usuario 23346	usuario23346@example.com
2023-05-15	Usuario 5453	usuario5453@example.com
2023-01-15	Usuario 36393	usuario36393@example.com
2023-06-19	Usuario 2346	usuario2346@example.com

</td></tr>
</table>

RF4 Avanzado