



# Sistemas Transaccionales

Documento de informe- iteración 3

Grupo C1

Santiago Jaimes

Juan Jose Cediel

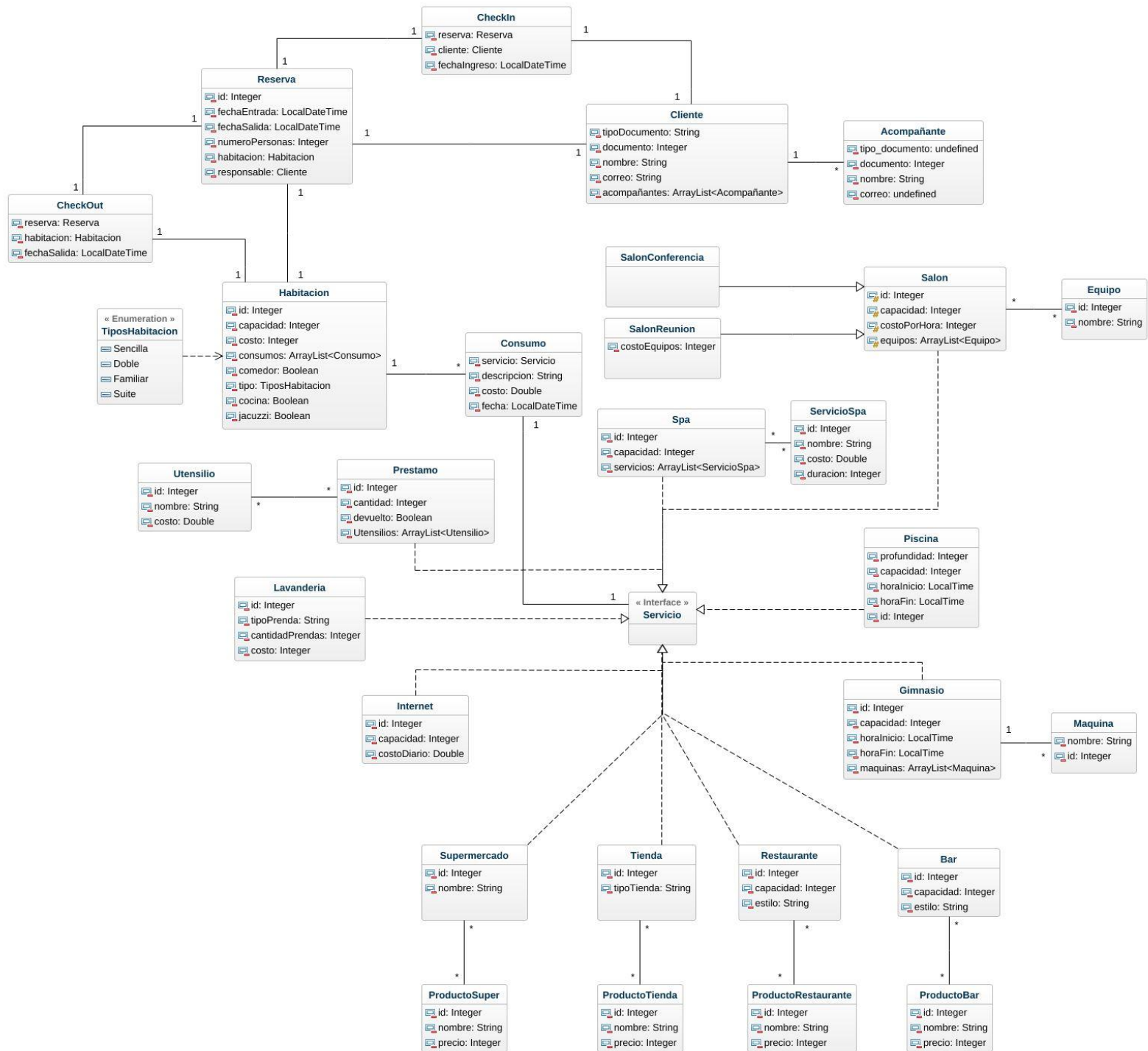
Camilo Mercado

Universidad de Los Andes  
Bogotá – Colombia

3 de diciembre de 2023

# 1. Análisis y modelado conceptual

## a. Modelo UML para la aplicación



## Clases y sus Atributos:

**Reserva:** Tiene atributos para el manejo de las fechas de reserva, check-in y check-out, así como una referencia a la clase Cliente y a la clase Habitación.

**Cliente:** Guarda información del cliente, como documento, tipo de documento, nombre, correo y una lista de acompañantes.

**Acompañante:** Contiene información similar a la clase Cliente, probablemente para representar a las personas que se quedan junto con el cliente principal.

**Checkin:** Asociado con Cliente y Reserva, guarda información de la fecha de ingreso y los cambios de reservas.  
**Checkout:** Similar a Checkin, pero para la salida del cliente del hotel.

**Habitación:** Una clase la cual tiene varios atributos significativos y además un tipo enumeration: Doble, Sencilla, Familiar, y Suite, indicando diferentes tipos de habitaciones disponibles.

**SalonConferencia y SalonReunion:** Parecen representar espacios que pueden ser reservados, como salones para conferencias o reuniones, respectivamente.

**Spa, Piscina, Gimnasio:** Representan diferentes servicios del hotel con atributos para manejar la capacidad y horarios.

**ServicioSpa:** Especifica los servicios del spa, probablemente con una lista de los diferentes servicios ofrecidos y sus duraciones.

**Lavanderia, Internet, Supermercado, Tienda, Restaurante, Bar:** Clases que ofrecen servicios concretos y están relacionadas con la interfaz Servicio.

**Consumo:** Registra los servicios consumidos, referenciando a la interfaz Servicio.

**Prestamo:** Maneja los préstamos de objetos o servicios a los clientes.

**Equipo:** Relacionada con Salon, podría representar el equipo disponible para los eventos.

**ProductoSuper, ProductoTienda, ProductoRestaurante, ProductoBar:** Clases que representan los productos disponibles en las distintas áreas del hotel.

## Relaciones entre Clases:

**Asociación:** La mayoría de las relaciones representadas son asociaciones simples, lo que indica que una instancia de una clase puede conocer a otra y/o interactuar con ella. Por ejemplo, Cliente está asociado con Reserva, lo que indica que un cliente realiza una reserva.

**Implementación:** Las clases como Lavanderia, Spa, Piscina, etc., implementan la interfaz Servicio. Esto se indica por una línea punteada con un triángulo en el extremo apuntando hacia Servicio.

**Relaciones con Multiplicidad:** Las líneas que conectan las clases sin multiplicidad sugieren relaciones sin restricciones específicas de cantidad, en el UML se evidencian que la mayoría de relaciones son bidireccionales y además cuentan la mayoría con su respectiva multiplicidad o cardinalidad.

## 2. Diseño de la base de datos

### 2.1. Análisis de carga de trabajo

#### a. Entidades y atributos

Entidad	Atributos
Cientes	- tipoDocumento: String - documento: Integer - nombre: String - correo: String
checkIn	- reserva: Reserva - cliente: Cliente - fechaIngreso: Date
Reservas	- id: Integer - fechaEntrega: Date - fechaSalida: Date - numeroPersonas: Integer - habitación: Habitaciones - responsable: Clientes
checkOut	- reserva: Reservas - habitación: Habitaciones - fechaSalida: Date
Habitaciones	- numero: Integer - capacidad: Integer - costo: Integer - consumos: ArrayList<Consumos> - tipo: TiposHabitacion
Tipos_habitacion	- tipo: String
Consumos	- servicio: Servicios - fecha: Date
Servicios	- tipo: String - costo: Integer - descripción: String

#### b. Cuantificación de las entidades

Entidad	Cantidad de registros esperados
Tipos_habitacion	20 (para uno de los hoteles de la cadena)
Habitaciones	200 (para uno de los hoteles de la cadena)
Servicios	35 (para uno de los hoteles de la cadena)
Reservas	50.000 (en una ventana de 3 años para uno de los hoteles de la cadena)
CheckIn	50.000 (en una ventana de 3 años para uno de los hoteles de la cadena)
Consumos	250.000 (en una ventana de 3 años para uno de los hoteles de la cadena)
CheckOut	50000 (en una ventana de 3 años para uno de los hoteles de la cadena)

#### c. Análisis de operación de lectura y escritura para cada entidad

Entidad	Operaciones	Información necesaria	Tipo
<b>Tipos_habitacion</b>	CRUD de los tipos de habitaciones de los cuales dispone el hotel con las dotaciones correspondientes	El tipo de la habitación	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>Habitaciones</b>	CRUD de las instancias de las habitaciones de las cuales dispone el hotel	El numero de la habitación + capacidad + costo	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>Servicios</b>	CRUD de la descripción de los servicios de los cuales dispone el hotel.	El tipo de Servicio + costo del servicio	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>Reservas</b>	CRUD de la llegada de un cliente al hotel	Habitación + numero de personas + fecha de entra + fecha de salida	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>CheckIn</b>	CRUD de la llegada de un cliente al hotel.	Reserva + cliente + fecha ingreso	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>Consumos</b>	CRUD del registro de un consumo de un servicio por parte de un cliente o sus acompañantes.	Servicio + fecha	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read
<b>CheckOut</b>	CRUD del registro de la salida de un cliente del hotel.	Reserva + habitación + fecha Salida	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read

d. Cuantificación de las operaciones de lectura y escritura para cada entidad

Entidad	Operaciones	Información necesaria	Tipo	Rango
<b>Tipos_habitacion</b>	CRUD de los tipos de habitaciones de los cuales dispone el hotel con las dotaciones correspondientes	El tipo de la habitación	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	1/semana
<b>Habitaciones</b>	CRUD de las instancias de las habitaciones de las cuales dispone el hotel	El numero de la habitación + capacidad + costo	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	2/Semana
<b>Servicios</b>	CRUD de la descripción de los servicios de los cuales dispone el hotel.	El tipo de Servicio + costo del servicio	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	2/Semana

<b>Reservas</b>	CRUD de la llegada de un cliente al hotel	Habitación + número de personas + fecha de entra + fecha de salida	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	Create, Update: 100/Semana  Read: 50/Semana
<b>CheckIn</b>	CRUD de la llegada de un cliente al hotel.	Reserva + cliente + fecha ingreso	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	Create, Update: 100/Semana  Read: 50/Semana
<b>Consumos</b>	CRUD del registro de un consumo de un servicio por parte de un cliente o sus acompañantes.	Servicio + fecha	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	Create, Update: 500/Semana  Read: 250/Semana
<b>CheckOut</b>	CRUD del registro de la salida de un cliente del hotel.	Reserva + habitación + fecha Salida	Registrar (Create): Write Actualizar (Update): Write Borrar (Delete): Write Consultar (Read): Read	Create, Update: 100/Semana  Read: 50/Semana

## 2.2. Descripción de las entidades de datos y relaciones

### a. Lista de entidades

Entidad	Descripción
<b>Tipo_habitacion</b>	La clase representa los tipos de habitaciones con los cuales cuenta el hotel y que el cliente antes de realizar el ingreso puede seleccionar para su estadía. En este caso puede ser: sencilla, doble, suite.
<b>Habitaciones</b>	La clase representa la habitación del hotel. la cual fue asignada al cliente, para poder ser distinguida del resto la habitación cuenta con un identificador único. También cuenta con una capacidad, el costo asociado por tomar la habitación, una lista que guarda los consumos que se han registrado para la cuenta de la habitación y el tipo de habitación que describe la clase de habitación que reservo el cliente.
<b>Servicios</b>	Representa la acción de asegurar la disponibilidad de un servicio adicional, como el spa o el restaurante, para un cliente en una fecha especificada. Similar como a la reserva, esto puede requerir el pago de tarifas adicionales.
<b>Reservas</b>	La clase representa la acción de asegurar una habitación para una fecha específica. El cliente debe especificar la fecha de entrada, la fecha de salida y el número de personas. La reserva garantiza la disponibilidad de la habitación y puede implicar el pago de una tarifa por adelantado.
<b>checkIn</b>	Representa el proceso de ingreso de entrada de un cliente al hotel. Realizado por el recepcionista, implica la verificación de la reserva, registro de información del cliente y asignación de la habitación.
<b>Consumos</b>	Representa la utilización de productos o servicios por parte del cliente durante su estadía en el hotel. Los consumos, como el minibar u otros servicios adicionales, se cargan a la cuenta de la habitación y se revisan al momento del check-out.
<b>checkOut</b>	Representa el proceso de salida de un cliente del hotel. Nuevamente realizado por el recepcionista, implica la revisión de todos los consumos realizados por el cliente, la generación de la cuenta correspondiente y el pago por parte del cliente.

b. Relaciones entre entidades

Entidad	Relación con (Entidad)	Cardinalidad
Reservas	Cientes	1 to 1
Reservas	Habitaciones	1 to 1
Habitaciones	Tipos_habitacion	1 to 1
Habitaciones	Consumos	1 to *
Consumos	Servicios	1 to 1
CheckOut	Habitaciones	* to *
CheckOut	Reservas	1 To 1
CheckIn	Cliente	1 to 1
CheckIn	Reservas	1 to 1

c. Análisis de selección de esquema de asociación

Entidad	Relación con (Entidad)	Esquema de asociación	Razón
Reservas	Cientes	Embebido	Atomicidad de las consultas, pues estas siempre se consultan juntas, de lo contrario no tiene sentido una reserva sin cliente
Reservas	Habitaciones	Referenciado	Evita la duplicidad de datos al tener varias reservas asociadas a una habitación
Habitaciones	Tipos_habitacion	Embebido	Tiene relación de pertenencia, pues una habitación tiene un tipo
Habitaciones	Consumos	Embebido	Atomicidad de las consultas, pues una habitación está asociada a consumos al consultarlos
Consumos	Servicios	Embebido	Tienen relación de tener uno, van juntos
CheckOut	Habitaciones	Referenciado	Son individuales, pues una habitación puede cambiar sin cambiar el checkOut
CheckOut	Reservas	Referenciado	Por individualidad, pues estos se registran en momentos diferentes y sin certeza
CheckIn	Cliente	Referenciado	Son individuales, pues un cliente puede cambiar sin cambiar el checkIn
CheckIn	Reservas	Referenciado	Por individualidad, pues estos se registran en momentos diferentes y sin certeza

d. Descripción gráfica JSON

Asociación

Imagen

Reservas-  
Clientes

### Reservas document

```
_id: ObjectId('65555661b49a8b3ab2611c4a')
numeroPersonas: 2
habitacion: ObjectId('6555565fb49a8b3ab2611c2a')
fechaEntrada: 2023-01-04T12:00:00.000+00:00
fechaSalida: 2023-01-14T12:00:00.000+00:00
▼ responsable: Object
  tipoDocumento: "CC"
  numDocumento: 123456
  nombre: "Eduardo benitez"
  correo: "e.benitez@gmail.com"
```

### Clientes document

```
_id: ObjectId('65555660b49a8b3ab2611c46')
tipoDocumento: "CC"
numDocumento: 123456
nombre: "Eduardo benitez"
correo: "e.benitez@gmail.com"
```

Reservas-  
Habitaciones

### Reservas document

```
_id: ObjectId('65555661b49a8b3ab2611c4a')
numeroPersonas: 2
habitacion: ObjectId('6555565fb49a8b3ab2611c2a')
fechaEntrada: 2023-01-04T12:00:00.000+00:00
fechaSalida: 2023-01-14T12:00:00.000+00:00
▼ responsable: Object
  tipoDocumento: "CC"
  numDocumento: 123456
  nombre: "Eduardo benitez"
  correo: "e.benitez@gmail.com"
```

### Habitaciones document

```
_id: ObjectId('6555565fb49a8b3ab2611c2a')
numero: 1
capacidad: 4
costo: 700000
► tipo: Object
► consumos: Array (3)
```

Habitaciones-  
Tipos\_habitacion

### Habitaciones document

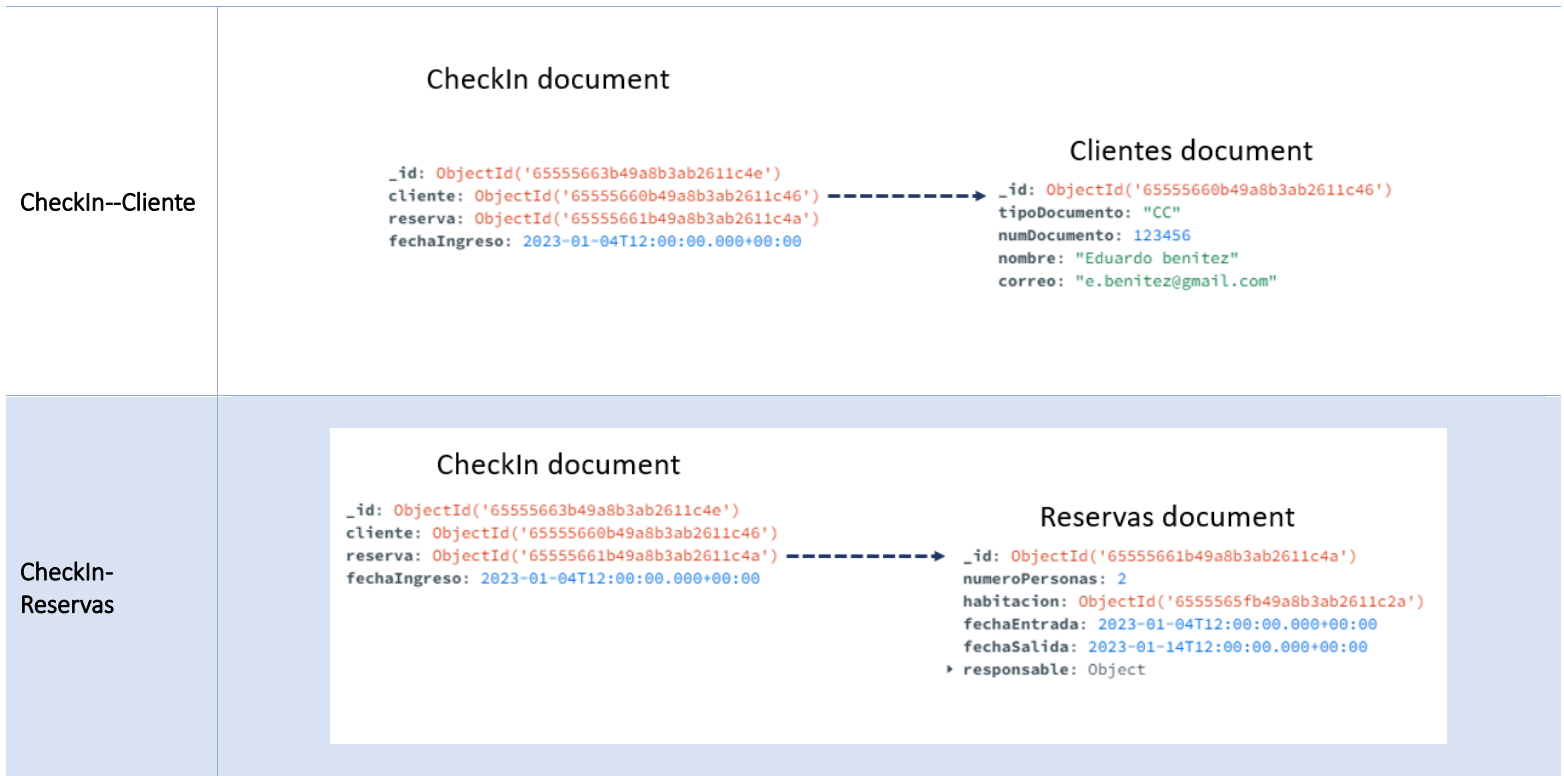
```
_id: ObjectId('6555565fb49a8b3ab2611c2a')
numero: 1
capacidad: 4
costo: 700000
▼ tipo: Object
  tipo: "suite"
  jacuzzi: 1
  comedor: 0
  cocina: 0
▼ consumos: Array (3)
  ▼ 0: Object
    ▼ servicio: Object
      tipo: "spa"
      costo: 10000
      descripcion: "esto es un spa"
      fecha: 2023-11-15T12:00:00.000+00:00
    ► 1: Object
    ► 2: Object
```

### Tipos\_habitacion document

```
_id: ObjectId('6555565fb49a8b3ab2611c27')
tipo: "suite"
jacuzzi: 1
comedor: 0
cocina: 0
```



Habitaciones-Consumos	<div><div>Habitaciones document</div><div><pre><code>_id: ObjectId('6555565fb49a8b3ab2611c2a') numero: 1 capacidad: 4 costo: 700000 tipo: Object   tipo: "suite"   jacuzzi: 1   comedor: 0   cocina: 0 consumos: Array (3)   0: Object     servicio: Object       tipo: "spa"       costo: 10000       descripcion: "esto es un spa"       fecha: 2023-11-15T12:00:00.000+00:00     1: Object     2: Object</code></pre></div></div> <div><div>Consumos document</div><div><pre><code>_id: ObjectId('6555565fb49a8b3ab2611c3e') servicio: Object   tipo: "spa"   costo: 10000   descripcion: "esto es un spa"   fecha: 2023-11-15T12:00:00.000+00:00</code></pre></div></div> <div></div>
Consumos-Servicios	<div><div>Consumos document</div><div><pre><code>_id: ObjectId('6555565fb49a8b3ab2611c3e') servicio: Object   tipo: "spa"   costo: 10000   descripcion: "esto es un spa"   fecha: 2023-11-15T12:00:00.000+00:00</code></pre></div></div> <div><div>Servicios document</div><div><pre><code>_id: ObjectId('6555565fb49a8b3ab2611c33') tipo: "spa" costo: 10000 descripcion: "esto es un spa"</code></pre></div></div> <div></div>
CheckOut-Habitaciones	<div><div>CheckOut document</div><div><pre><code>_id: ObjectId('65555665b49a8b3ab2611c52') habitacion: ObjectId('6555565fb49a8b3ab2611c2a') reserva: ObjectId('65555661b49a8b3ab2611c4a') fechaSalida: 2023-01-14T12:00:00.000+00:00</code></pre></div></div> <div><div>Habitaciones document</div><div><pre><code>_id: ObjectId('6555565fb49a8b3ab2611c2a') numero: 1 capacidad: 4 costo: 700000 tipo: Object consumos: Array (3)</code></pre></div></div> <div></div>
CheckOut-Reservas	<div><div>CheckOut document</div><div><pre><code>_id: ObjectId('65555665b49a8b3ab2611c52') habitacion: ObjectId('6555565fb49a8b3ab2611c2a') reserva: ObjectId('65555661b49a8b3ab2611c4a') fechaSalida: 2023-01-14T12:00:00.000+00:00</code></pre></div></div> <div><div>Reservas document</div><div><pre><code>_id: ObjectId('65555661b49a8b3ab2611c4a') numeroPersonas: 2 habitacion: ObjectId('6555565fb49a8b3ab2611c2a') fechaEntrada: 2023-01-04T12:00:00.000+00:00 fechaSalida: 2023-01-14T12:00:00.000+00:00 responsable: Object</code></pre></div></div> <div></div>



### 3. Resultados logrados y no logrados

Se lograron hacer los CRUDs de los primeros 7 requerimientos y las 4 consultas de los requerimientos de consulta, por lo tanto, se consideran logrados todos los objetivos propuestos para la iteración finalizando el proyecto. Todos los puntos del documento requeridos se obtuvieron de manera satisfactoria, generando un análisis apropiado para tomar decisiones de diseño.