

Santiago Najar 202021647

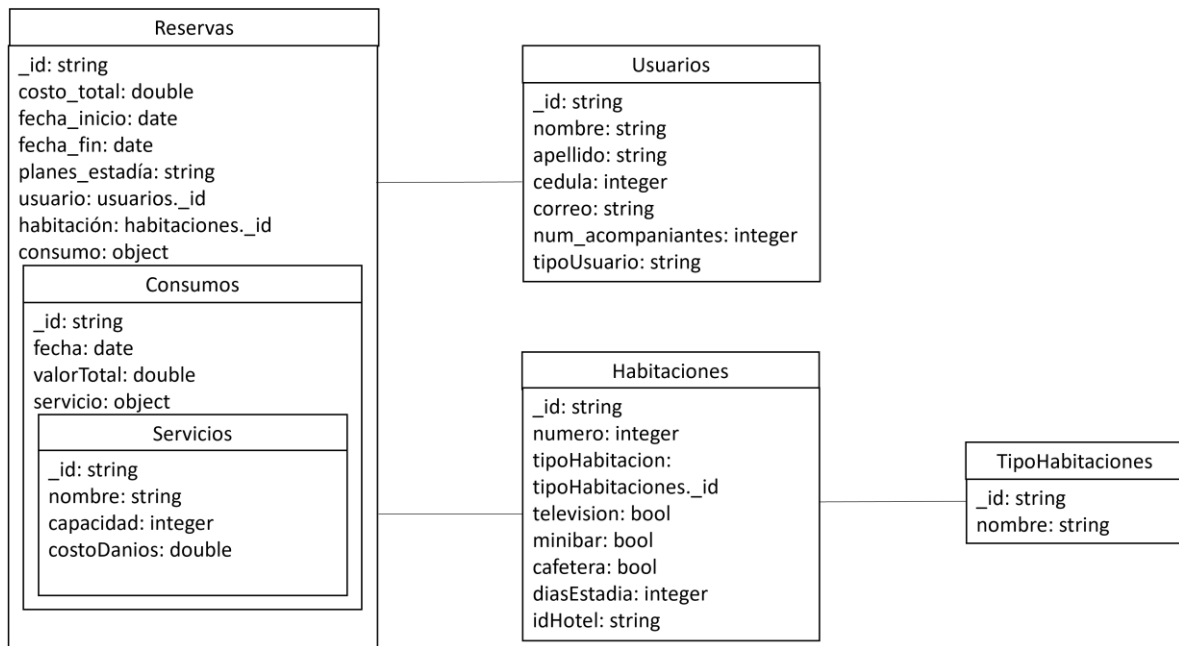
Juana Morales 202021591

Nicolas Rincon 202021963

### Iteración 3 – Proyecto Sistemas Transaccionales

2.

a)



3.

a)

- **Reservas:**
  - `_id`: Identificador único de la reserva (tipo string).
  - `costo_total`: Costo total de la reserva (tipo double).
  - `fecha_inicio`: Fecha de inicio de la reserva (tipo date).
  - `fecha_fin`: Fecha de fin de la reserva (tipo date).
  - `planes_estadia`: Descripción del plan de estancia (tipo string).
  - `usuario`: Referencia al ID del usuario que hace la reserva (tipo string, referencia a `Usuarios._id`).
  - `habitación`: Referencia al ID de la habitación reservada (tipo string, referencia a `Habitaciones._id`).

- consumo: Objeto que detalla los consumos realizados (tipo object).
- Consumos (embebidos en Reservas):
  - \_id: Identificador único del consumo (tipo string).
  - fecha: Fecha en que se realizó el consumo (tipo date).
  - valorTotal: Valor total del consumo (tipo double).
  - servicio: Objeto que detalla el servicio consumido (tipo object).
- Servicios (embebidos en Consumos):
  - \_id: Identificador único del servicio (tipo string).
  - nombre: Nombre del servicio (tipo string).
  - capacidad: Capacidad máxima de personas para el servicio (tipo integer).
  - costoDanios: Costo por daños asociados al servicio (tipo double).
- Usuarios:
  - \_id: Identificador único del usuario (tipo string).
  - nombre: Nombre del usuario (tipo string).
  - apellido: Apellido del usuario (tipo string).
  - cedula: Número de documento o cédula del usuario (tipo integer).
  - correo: Correo electrónico del usuario (tipo string).
  - num\_acompanantes: Número de acompañantes del usuario (tipo integer).
  - tipoUsuario: Categorización del usuario (tipo string).
- Habitaciones:
  - \_id: Identificador único de la habitación (tipo string).
  - numero: Número identificador de la habitación (tipo integer).
  - tipoHabitacion: Referencia al tipo de habitación (tipo string, referencia a TipoHabitaciones.\_id).
  - television: Indica si la habitación tiene televisión (tipo boolean).
  - minibar: Indica si la habitación tiene minibar (tipo boolean).
  - cafetera: Indica si la habitación tiene cafetera (tipo boolean).
  - diasEstadia: Número de días de la estadía (tipo integer).
  - idHotel: Identificador del hotel al que pertenece la habitación (tipo string).
- TipoHabitaciones:
  - \_id: Identificador único del tipo de habitación (tipo string).
  - nombre: Nombre del tipo de habitación (tipo string).

b)

La base de datos del sistema de gestión hotelera contendrá aproximadamente 80,000 documentos en la colección de reservas, cada una vinculada a un usuario, lo que podría significar hasta 80,000 documentos en la colección de usuarios, asumiendo que no hay usuarios repetidos. La colección de habitaciones contará con 100 documentos únicos, representando cada una de las habitaciones del hotel. Los tipos de habitaciones y servicios ofrecidos por el hotel no tienen una cantidad definida, pero serán menores en comparación con las reservas y usuarios. Además, los registros de consumos podrían ascender a cientos de miles, dependiendo del número de transacciones por reserva.

c)

Entidades	Operaciones	Información Necesaria	Tipo
Reservas	Consultar detalles de la reserva	Detalles de la reserva + detalles del usuario + detalles de la habitación	Lectura
Reservas	Añadir/actualizar reserva	Detalles de la reserva + ID del usuario + ID de la habitación	Escritura
Reservas	Consultar reservas de un usuario	ID del usuario + Detalles de la reserva	Lectura
Reservas	Eliminar reserva	ID de la reserva	Escritura
Usuarios	Consultar detalles del usuario	Detalles del usuario	Lectura
Usuarios	Añadir/actualizar usuario	Detalles del usuario	Escritura
Usuarios	Eliminar usuario	ID del usuario	Escritura
Habitaciones	Consultar detalles de la habitación	Detalles de la habitación	Lectura
Habitaciones	Actualizar disponibilidad de la habitación	ID de la habitación + estado actualizado de disponibilidad	Escritura
TipoHabitaciones	Consultar detalles del tipo de habitación	Detalles del tipo de habitación	Lectura
TipoHabitaciones	Añadir/actualizar tipo de habitación	Detalles del tipo de habitación	Escritura
Consumos	Añadir consumo a una reserva	ID de la reserva + Detalles del consumo	Escritura
Consumos	Consultar consumos de una reserva	ID de la reserva + Detalles del consumo	Lectura
Servicios	Consultar detalles del servicio	Detalles del servicio	Lectura
Servicios	Añadir/actualizar servicio	Detalles del servicio	Escritura

d)

Entidades	Operación	Información Necesaria	Tipo	Frecuencia
TipoHabitaciones	Consultar tipo de habitación	Detalles del tipo de habitación	Lectura	1 vez/semana
TipoHabitaciones	Crear/actualizar tipo de habitación	Detalles del tipo de habitación	Escritura	1 vez/semana
Habitaciones	Consultar habitaciones	Detalles de las habitaciones	Lectura	2 veces/semana
Habitaciones	Crear/actualizar habitaciones	Detalles de las habitaciones	Escritura	2 veces/semana
Servicios	Consultar servicios	Detalles de los servicios	Lectura	2 veces/semana
Servicios	Crear/actualizar servicios	Detalles de los servicios	Escritura	2 veces/semana
Reservas	Consultar reservas	Detalles de la reserva	Lectura	50 veces/día
Reservas	Crear/actualizar reserva	Detalles de la reserva	Escritura	100 veces/día
Consumos	Consultar consumos	Detalles de consumos	Lectura	250 veces/día
Consumos	Crear/actualizar consumos	Detalles de consumos	Escritura	500 veces/día
Check-in/Check-out	Consultar llegadas y partidas	Detalles de llegada y salida	Lectura	50 veces/día

Check-in/Check-out	Registrar/modificar llegada o salida	Detalles de llegada y salida	Escritura	100 veces/día
--------------------	--------------------------------------	------------------------------	-----------	---------------

#### 4.

##### a)

El diagrama ilustra el modelo de datos de un sistema de gestión de hotel utilizando MongoDB, una base de datos NoSQL. Se compone de cinco colecciones interrelacionadas:

**Reservas:** Esta colección almacena los detalles de las reservas hechas por los clientes. Cada documento incluye información como el coste total de la estancia, las fechas de inicio y fin, y un campo para planes de estancia. Además, establece relaciones con los documentos de usuarios y habitaciones mediante referencias de identificación. Los consumos realizados durante la estancia se almacenan como un objeto embebido dentro de cada reserva.

**Usuarios:** Aquí se encuentran los perfiles de los clientes del hotel, con datos personales como nombre, apellido y cédula. También se registra su información de contacto y la cantidad de acompañantes, junto con el tipo de usuario, lo que puede indicar diferentes niveles o categorías de clientes dentro del hotel.

**Habitaciones:** Esta colección detalla las habitaciones disponibles, numeradas y categorizadas por tipo. Además, se especifican los servicios incluidos en cada habitación, como televisión, minibar y cafetera, y se cuenta con un campo para la cantidad de días de la estadía.

**TipoHabitaciones:** Define las categorías de habitaciones que ofrece el hotel, por ejemplo, simples, dobles o suites, facilitando la clasificación y selección de habitaciones según las necesidades de los clientes.

**Consumos:** Aunque aparece como una colección separada, está integrada dentro de la colección de Reservas y registra los gastos adicionales de los clientes en servicios del hotel durante su estancia, como el uso del spa o del minibar, incluyendo la fecha y el coste total de cada consumo.

##### b)

- **Reservas a Usuarios:**
  - Uno a muchos: Un usuario (Usuarios) puede tener muchas reservas (Reservas), pero cada reserva está asociada a un único usuario. La referencia usuario: usuarios.\_id en la colección de Reservas indica esta relación.
- **Reservas a Habitaciones:**
  - Uno a muchos: Una habitación (Habitaciones) puede estar asociada a muchas reservas (Reservas) a lo largo del tiempo, pero cada reserva se refiere a una única habitación en un momento dado. Esto se representa con la referencia habitación: habitaciones.\_id en la colección de Reservas.

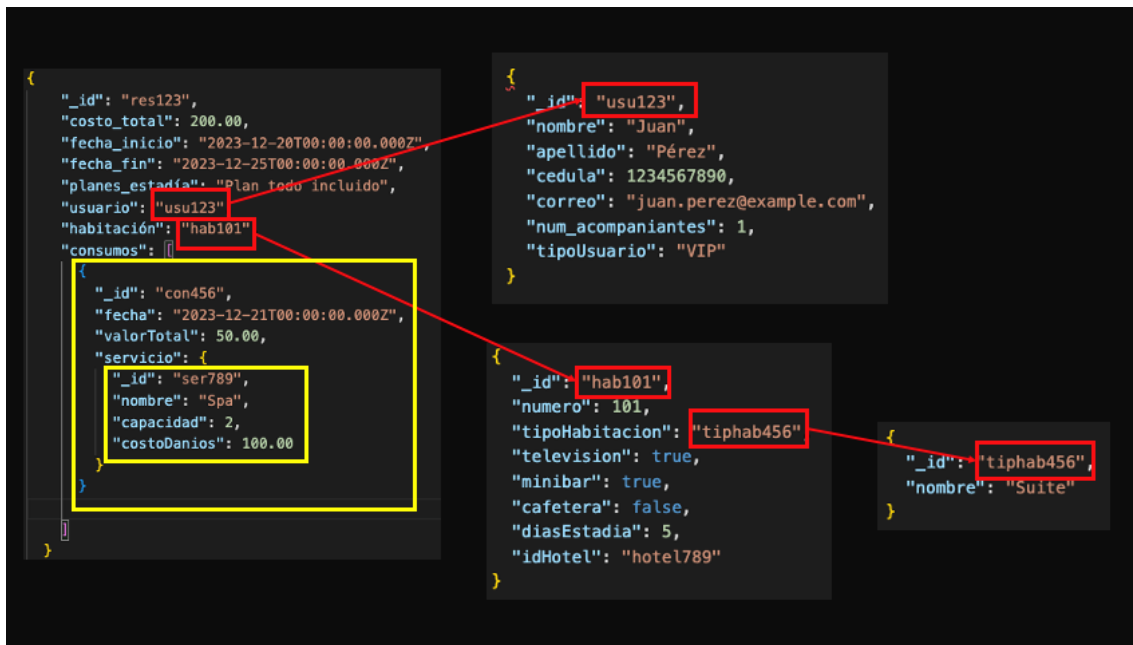
- Reservas a Consumos:
  - Uno a muchos embebidos: Una reserva (Reservas) puede incluir muchos consumos (Consumos), que se representan como un objeto embebido dentro de la reserva. Esto implica que cada reserva tiene una lista de consumos asociados.
- Consumos a Servicios:
  - Muchos a uno embebidos: Cada consumo (Consumos) está relacionado con un único servicio (Servicios), dado que el objeto servicio está embebido dentro de cada consumo. No obstante, un servicio puede estar asociado a muchos consumos diferentes.
- Habitaciones a TipoHabitaciones:
  - Muchos a uno: Muchas habitaciones (Habitaciones) pueden compartir el mismo tipo de habitación (TipoHabitaciones), pero cada habitación se asocia a un único tipo de habitación. La referencia tipoHabitacion: tipoHabitaciones.\_id en la colección de Habitaciones establece esta relación.

C)

Justificación de embeber o referenciar

- Reservas y Usuarios:
  - Dado que la carga de trabajo implica que las reservas se consultan y actualizan con frecuencia y que cada reserva pertenece a un solo usuario, es más práctico embeber la información del usuario en la reserva para operaciones de lectura atómica y simplicidad.
- Reservas y Habitaciones:
  - Las habitaciones pueden tener una alta cardinalidad en relación con las reservas y también se espera que existan independientemente de las reservas. Por lo tanto, una referencia es más adecuada para esta relación.
- Reservas y Consumos:
  - Los consumos están directamente relacionados con una reserva específica y probablemente se consulten y actualicen junto con ella. Por esto, embeber los consumos dentro del documento de reserva es adecuado
- Consumos y Servicios:
  - Básicamente la misma explicación de reservas y consumo. Cuando hay un consumo es porque se consumió un servicio y por ello es mejor embeberlo.

d)



Este es un ejemplo de una reserva como se vería en el json. Los colores rojos son referencias de una colección a otra por medio de sus identificadores, mientras que el color amarillo es una colección embebida dentro de otra.

C) Validación en el archivo .txt