Entrega 3

Hotel Andes

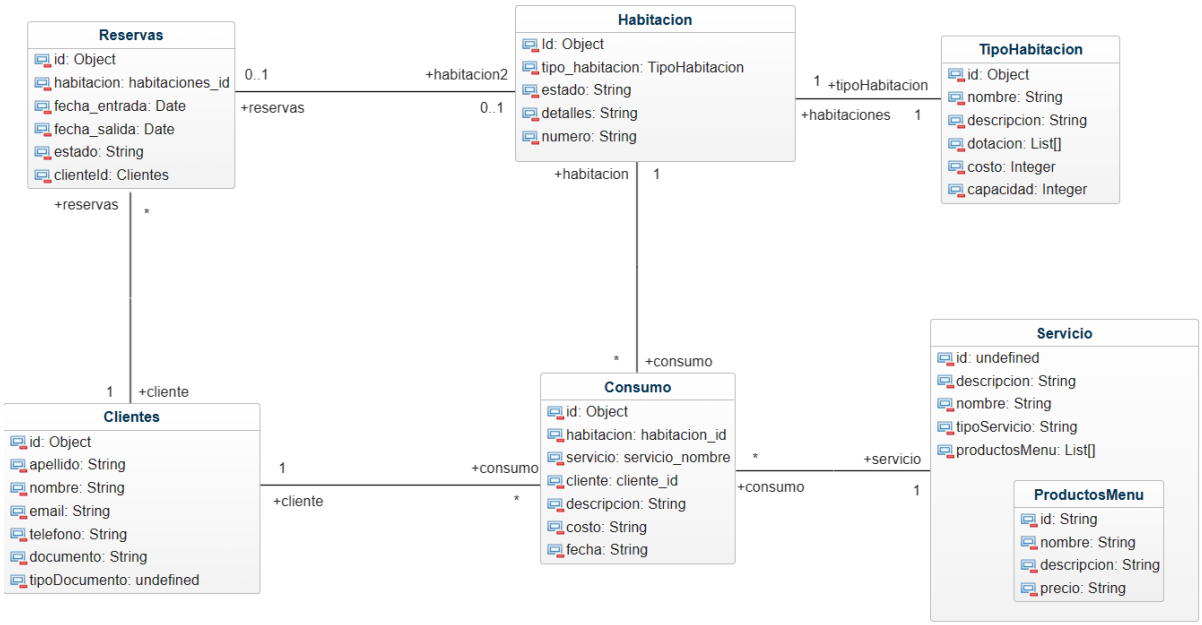
B9

Pablo R Santiago Peñaranda - 201922871

Adriana Sofia Rozo Cepeda – 202211498

Nicolas David Camargo Prieto - 202020782

1. Análisis y modelo conceptual



2. Diseño de la base de datos

a. Presentado nuestro modelo conceptual, se observa que las entidades planteadas para cumplir con los requerimientos del enunciado son: tipoHabitacion, Habitación, Cliente, Consumo, Servicio, Reservas y ProductosMenu. Las cuales tienen los mismos atributos que en entregas pasadas pero esta vez enfocándonos en la base de datos para MongoDB.

b. Para la cantidad de registros por entidad y por lo que se menciona en el enunciado, se distribuye de la siguiente manera:

Entidad	Numero de registros
TipoHabitacion	3
Habitación	3

Cliente	3
Reservas	3
Consumo	3
Servicio	12
ProductosMenu	3

Ahora bien, la siguiente tabla muestra las principales operaciones que realiza la aplicación de Hotel Andes según la entidad involucrada, la operación que realiza, la información requerida por el usuario, el tipo de operación y la tasa de tiempo.

c. Anexo a – tabla de análisis de operaciones de lectura y escritura

Entities	**Consultar todas las habitaciones**	Información necesitada	**Type**
Habitación	Anadir/ Borrar	Detalles habitación + Tipo habitacion	Write
TipoHabitacion	Consultar	Detalles tipo habitación (puede ser su tipo, nombre)	Read
TipoHabitacion	Anadir/Actualizar	Detalles+Id tipo habitacion	Write
Cliente	Registrar llegada/Salida		Read
Cliente	Añadir/Actualizar	Detalles del cliente	Write
Reservas	Consultar	Detalles de la reserva	Read
Reservas	Anadir	Detalles de la reserva+Id reserva+Id habitación+Id cliente	Write
Consumo	Consultar	Detalles del consumo, por ejemplo, número de habitación o cliente.	Read
Consumo	Anadir/Actualizar	Id del consumo	Write
Servicio	Consultar	Detalles del servicio, por ejemplo, nombre	Read
Servicio	Anadir/Actualizar	Id del consumo	Write
ProductosMenu	Consultar	Servicio Id al que este asociado	Read
ProductosMenu	Anadir/Actualizar	Servicio Id al que está asociado.	Write

Consumo	Consultar dinero recolectado por servicios por habitación	Información de los servicios-Id de servicio-Id de habitación	Read
HabitacionReservas	Consultar índice ocupación habitaciones	idHabitacion + Indice	Read
Consumo	Consultar consumo cliente por rango fechas	idCliente + Consumo	Read
Consumo	Consultar consumo general en rango de fecha	Info Clientes + Info Consumo	Read

d. Anexo b – tabla de cuantificación de operaciones de lectura y escritura

Entities	**Consultar todas las habitaciones**	Información necesitada	**Type**	**Rate**
Habitación	Anadir/ Borrar	Detalles habitación + Tipo habitacion	Write	60/sec
TipoHabitacion	Consultar	Detalles tipo habitación (puede ser su tipo, nombre)	Read	10/sec
TipoHabitacion	Anadir/Actualizar	Detalles+Id tipo habitacion	Write	60/sec
Cliente	Registrar llegada/Salida		Read	
Cliente	Añadir/Actualizar	Detalles del cliente	Write	60/sec
Reservas	Consultar	Detalles de la reserva	Read	10/sec
Reservas	Anadir	Detalles de la reserva+Id reserva+Id habitación+Id cliente	Write	60/sec
Consumo	Consultar	Detalles del consumo, por ejemplo, número de habitación o cliente.	Read	10/sec
Consumo	Anadir/Actualizar	Id del consumo	Write	60/sec
Servicio	Consultar	Detalles del servicio, por ejemplo, nombre	Read	10/sec
Servicio	Anadir/Actualizar	Id del consumo	Write	60/sec

ProductosMenu	Consultar	Servicio Id al que este asociado	Read	10/sec
ProductosMenu	Anadir/Actualizar	Servicio Id al que está asociado.	Write	60/sec
Consumo	Consultar dinero recolectado por servicios por habitación	Información de los servicios-Id de servicio-Id de habitación	Read	10/sec
HabitacionReservas	Consultar índice ocupación habitaciones	idHabitacion + Indice	Read	10/sec
Consumo	Consultar consumo cliente por rango fechas	idCliente + Consumo	Read	10/sec
Consumo	Consultar consumo general en rango de fecha	Info Clientes + Info Consumo	Read	10/sec

b. Ahora bien, según nuestro modelo conceptual propuesto, podemos identificar las **siguientes entidades y sus relaciones entre sí:**

- Habitación-Tipo Habitación:
  - Cada habitación tiene solo 1 tipo de habitación y lo mismo en el sentido contrario.
  - (Relación uno a uno)

- Habitación-Cliente:
  - Cada habitación puede tener 0 o muchos clientes asignados en diferentes momentos, mientras que un cliente solo puede tener una habitación asignada activa.
  - (Relación uno a muchos)

- Habitación-Consumo:
  - Una habitación puede tener 0 o muchos consumos de sus clientes, pero dichos consumos solo pueden estar asignados a 1 habitación.
  - (Relación uno a muchos)

- Consumo-Cliente:
  - Un cliente puede tener múltiples consumos, pero un consumo solo puede ser asignado a un cliente.
  - (Relación uno a muchos)

- Consumo-Servicio:
    - Un consumo de un cliente se relaciona con 1 solo servicio, pero un servicio puede tener múltiples consumos de varios clientes.
    - (Relación uno a muchos).

Ahora bien, a continuación, vemos cada entidad con su respectiva descripción.

Entidad	Descripción
Habitación	Modela una habitación del hotel, esta tiene atributos importantes y relaciones con otras entidades como tipo Habitación para determinar algunas de sus características.
TipoHabitacion	Modela los tipos de habitación presentes en el hotel los cuales son asociados con cada habitación para darle esa propiedad.
Cliente	Modela un cliente del hotel, guarda información importante de este para su instancia en el hotel.
Reservas	Modela una reserva, en esta esta asociada una habitación, un cliente y permite llevar el histórico de las reservas.
Consumo	Modela los consumos hechos por los clientes en el hotel, los cuales están asociados a este y un servicio.
Servicio	Modela todos los servicios que el hotel brinda. Estos servicios ofrecen o no productos.
ProductosMenu	Modela los productos que pueden ser vendidos por los servicios.

Tabla de análisis de esquema de relación entre entidades

Teniendo en cuenta la tabla vista en clases y nuestro análisis del diseño de la base de datos presentaremos nuestras justificaciones.

Guideline Name	Question	Embed	Referenc
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information have a "has-a," "contains," or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	Yes	No
Archival	Should the pieces of information be archived at the same time?	Yes	No
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	No	Yes
Data Duplication	Would data duplication be too complicated to manage and undesired?	No	Yes
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	No	Yes
Document Growth	Would the embedded piece grow without bound?	No	Yes
Workload	Are the pieces of information written at different times in a write-heavy workload?	No	Yes
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

- Habitación-Tipo Habitación:
    - Referenced relationship: decidimos que fuera referenciada ya que no queríamos que las entidades se tuvieran que actualizar al mismo tiempo. Es más común actualizar un Tipo de Habitación, cuando queremos añadir más elementos a este, por ejemplo, un televisor, una mesa de plancha etc. Una habitación por su ubicación que tiene un número no se actualiza tan seguido. Por ende, queríamos que la complejidad de actualización no fuese grande.

      Por otro lado, en nuestro modelo existen dos tipos de consulta. Tipos de habitación, creación, eliminación y consulta y lo mismo para Habitaciones. Mantenerlas separadas permite que estas labores sean mucho más sencillas.

      Se maneja perfecta una referencia ya que con una consulta simple podemos ver el tipo de habitación de una habitación.

- Habitación-Consumo:
    - Manejamos una referencia para cada consumo a una habitación específica. Las decisiones tomadas fueron principalmente porque no queríamos que

los consumos se asociaran por siempre a una habitación haciendo mucho más complicado hacer un filtro.

Nos enfocamos en hacer las consultas mucho mas fáciles, evitando entrar a objetos embebidos de los cuales no podríamos tener atomicidad de las consultas.

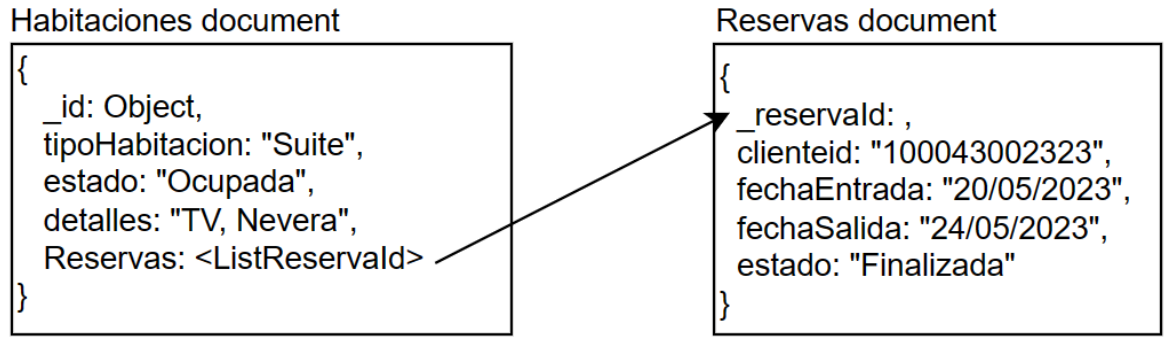
- Consumo-Cliente:

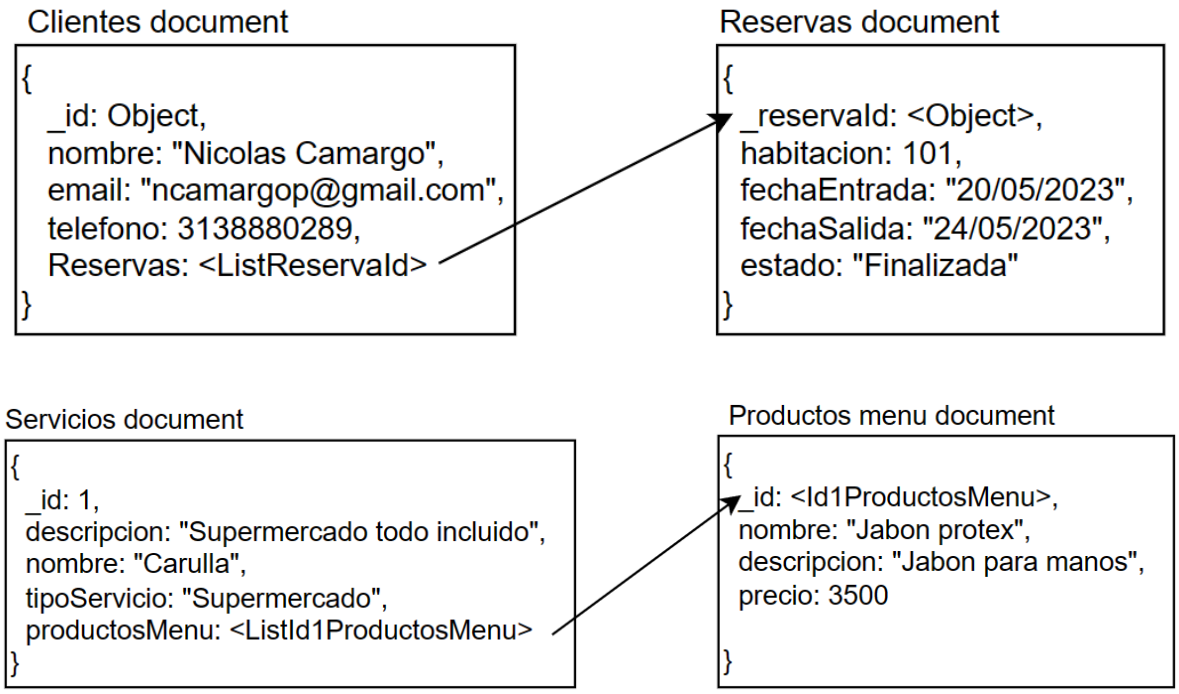
    - Realizamos esto de forma referenciada ya que queremos que, aunque no estén totalmente normalizadas las colecciones, podamos dividir diferentes aspectos del hotel. En este caso, esta relación garantiza individualidad para identificar los consumos de los clientes y que estos puedan existir sin un "padre". Lo anterior, es que puede que los clientes no tengan consumos en el Hotel de los servicios ofrecidos, por ejemplo, que almuercen afuera.

- Consumo-Servicio:
    - Quisimos manejar esto como una referencia principalmente porque ambos deben poder existir sin el otro individualmente y solo ser asociados al presentar un consumo por parte de un cliente que tiene una habitación. Por otro lado, queríamos evitar que el documento fuese demasiado grande.

- Servicio - ProductosMenu
    - Fue la única que mantuvimos embebida. Esto es porque un Servicio contiene si o si productos. Por otro lado, generalmente la atomicidad de las consultas en nuestro Hotel identifica que generalmente se buscan los servicios, pero siempre con sus productos.

Esquemas embebidos y referenciados




Clientes document
{
_id: Object,
nombre: "Nicolas Camargo",
email: "ncamargop@gmail.com",
telefono: 3138880289,
Reservas: <ListReservaId>
}
Reservas document
{
_reservaId: <Object>,
habitacion: 101,
fechaEntrada: "20/05/2023",
fechaSalida: "24/05/2023",
estado: "Finalizada"
}
Servicios document
{
_id: 1,
descripcion: "Supermercado todo incluido",
nombre: "Carulla",
tipoServicio: "Supermercado",
productosMenu: <ListId1ProductosMenu>
}
Productos menu document
{
_id: <Id1ProductosMenu>,
nombre: "Jabon protex",
descripcion: "Jabon para manos",
precio: 3500
}