

Caso 3 – Sistemas Transaccionales

Hotel Andes



3 de diciembre de 2023

Michael Blanquicett Carvajal – 202014156

Santiago Castro – 202122158

Juan Diego Acuña – 202121453   
Universidad de los Andes

Contenido

[Modelo Conceptual 3](#_Toc152429702)

[Análisis de la carga de trabajo 4](#_Toc152429703)

[Análisis de las consultas 5](#_Toc152429704)

[Análisis de las consultas 6](#_Toc152429705)

[Cuantificación de las consultas 6](#_Toc152429706)

[Análisis de cardinalidad 7](#_Toc152429707)

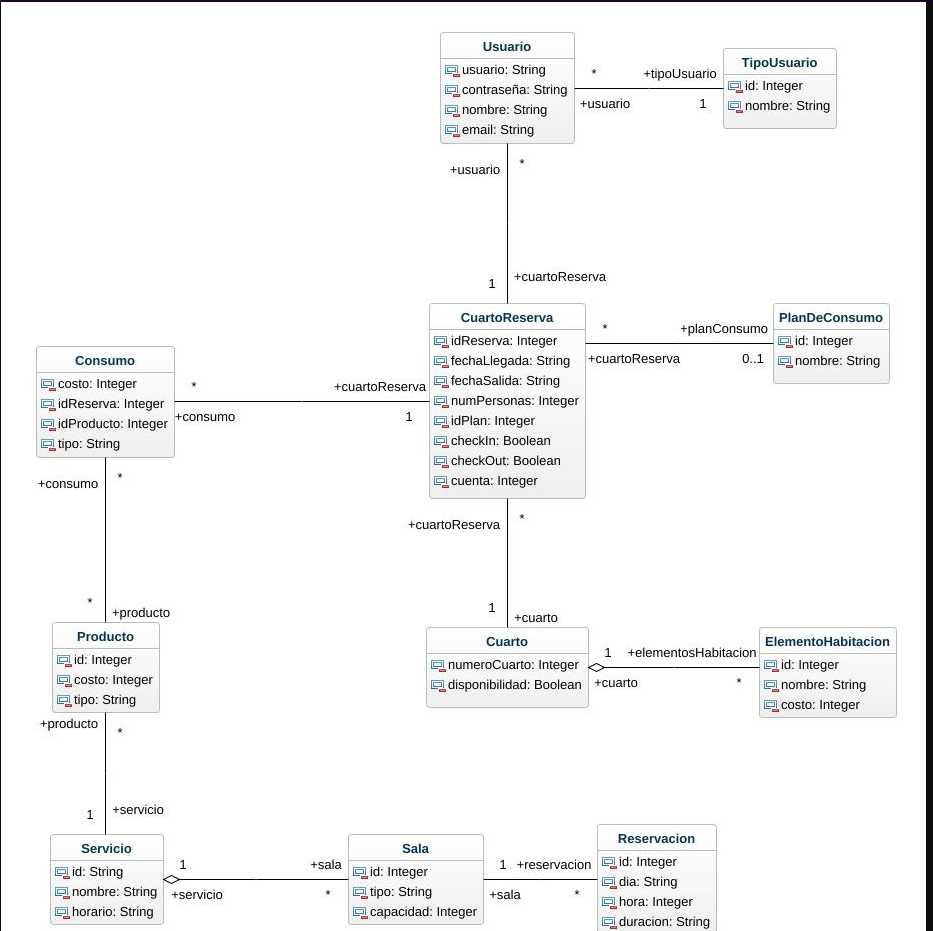
[Modelo Documental 8](#_Toc152429708)

[Modelo en JSON 8](#_Toc152429709)

[Ejemplo del Modelo 10](#_Toc152429710)

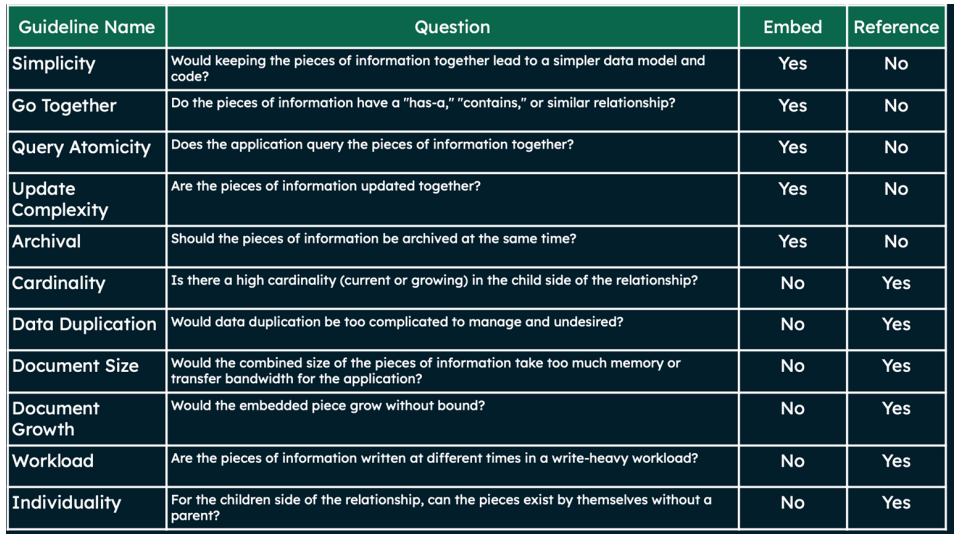
[Esquema de Validación 11](#_Toc152429711)

# Modelo Conceptual



# Análisis de la carga de trabajo

Para el desarrollo de esta base de datos documental, se tuvo en cuenta lo siguiente:

* Los tipos de habitación no se consultan, crean o modifican habitualmente.
* Las habitaciones (Cuarto) no se consultan, crean o modifican habitualmente.
* Los servicios del hotel (Servicio) no se consultan, crean o modifican habitualmente.
* La reservación de una habitación (CuartoReservacion) se consulta, crea y modifica habitualmente.
* Los consumos hechos por un cliente se consultan, crean y modifican muy habitualmente.
* La llegada y partida de un cliente se consulta, crea y modifica habitualmente.
* se estima que para uno de los hoteles de una gran cadena hotelera como Accord la cantidad de registros para los tipos de habitación podrían llegar a los 20, las habitaciones a las 200 y los servicios a unos 35. Con respecto a la reservación de una habitación, se estima que, por hotel por año, se puede llegar a tener hasta 17.000, lo cual quiere decir que en una ventana de tiempo de 3 años se podría llegar a 50.000 reservaciones. Este sería el número también de llegadas y partidas. En cuanto a los consumos de servicios, se estaría en los 250.000 en una ventana de 3 años.
* 

En cuanto al número de registros relacionado a cada entidad, tenemos:

* Tipos de habitación para el hotel de la cadena: 20 registros
* Habitaciones en el hotel: 200 registros
* Servicios ofrecidos: 35 registros
* Reservaciones por hotel por año: hasta 17.000 \* 3 años = 51.000
* Reservaciones en una ventana de 3 años: hasta 50.000
* Número de llegadas y partidas en una ventana de 3 años: hasta 50.000
* Consumos de servicios en una ventana de 3 años: 250.000

## Análisis de las consultas

En cuanto a las consultas, el análisis se realizó sobre si beneficia (La representación actual es buena para el manejo de las consultas) o perjudica (La representación actual es mala para el manejo de las consultas) frente a la forma en que se encuentra cada entidad representada (Si es una colección o esta embebida):

* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR TIPO DE HABITACIÓN
  + TipoHabitacion resulta beneficiada al ser una colección aparte, es fácil acceder a cada documento.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR HABITACIÓN
  + Habitación resulta beneficiada al ser una colección aparte, es fácil acceder a cada documento.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UN SERVICIO DEL HOTEL
  + Al ser una colección aparte, Servicio, se beneficia de las operaciones CRUD.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UNA RESERVA DE ALOJAMIENTO
  + Al ser una colección aparte, CuartoReserva, se beneficia de las operaciones CRUD.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR LA LLEGADA DE UN CLIENTE AL HOTEL
  + Al ser la llegada de cliente un atributo embebido en CuartoReserva, y ser esta solo coherente cuando el cliente tiene activa una reserva, se beneficia.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UN CONSUMO DE UN SERVICIO DEL HOTEL POR PARTE DE UN CLIENTE O SUS ACOMPAÑANTES
  + Al ser una colección aparte y tener los servicios embebidos, Consumo, se beneficia de las operaciones CRUD.
* REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR LA SALIDA DE UN CLIENTE
  + Al ser la salida de cliente un atributo embebido en CuartoReserva, y ser esta solo coherente cuando el cliente tiene activa una reserva, se beneficia
* MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO CORRIDO.
  + Para esta consulta es necesario encontrar todos los consumos realizados en un año, al tener servicios como colección, es una consulta que requiere una consulta anidada y una union, esto debido a que se debe acceder a el consumo de cada reserva las cuales si tienen la habitación asociada. Por lo que se concluye que esta beneficiada la representación de servicio
* MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL EN EL ÚLTIMO AÑO CORRIDO
  + Para esta consulta es necesario acceder a la ReservaCuarto, no es necesario hacer JOIN o consulta anidada a embebido, por lo que se beneficia mucho de su modelado.
* MOSTRAR EL CONSUMO EN HOTELANDES POR UN CLIENTE, EN UN RANGO DE FECHAS INDICADO.
  + Para esta consulta, los servicios cuentan con una fecha y un usuario el cual realizo la compra, una única consulta es suficiente, la representación actual es muy benéfica.
* CONSULTAR CONSUMO EN HOTELANDES Se quiere conocer la información de los clientes que consumieron al menos una vez un determinado servicio del hotel, en un rango de fechas. Los resultados deben ser clasificados según un criterio deseado por quien realiza la consulta. En la clasificación debe ofrecerse la posibilidad de agrupamiento y ordenamiento de las respuestas según los intereses del usuario que consulta como, por ejemplo, por los datos del cliente, por fecha y número de veces que se utilizó el servicio
  + Para esta consulta solo es necesario acceder a los servicios pues estos ya cuentan con el usuario y la fecha de realización.

## Análisis de las consultas

Para el análisis de operaciones de escritura y lectura:

|  |  |  |  |
| --- | --- | --- | --- |
| Entities | Operations | Information Needed | Type |
| TipoHabitacion | CRUD over TipoHabitacion | ALL TipoHabitacion | Write/Read |
| Cuarto | CRUD over Cuarto | ALL Cuarto | Write/Read |
| Servicio | CRUD over Servicio | ALL Servicio | Write/Read |
| CuartoReserva | CRUD over Reservas | ALL CuartoReserva | Write/Read |
| CuartoReserva | CRUD over entrada/salida del hotel | ALL CuartoReserva | Write/Read |
| Consumo, Servicio, CuartoReserva | CRUD over servicios | Consumo: Servicios ALL Servicio CuartoReserva: \_id | Write/Read |
| Consumo, Cuarto, CuartoReserva | Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido. | Consumo: Costo; Cuarto: \_id; CuartoReserva: Consumo, Cuarto. | Read |
| CuartoReserva | Mostrar el índice de ocupación de cada una de las habitaciones del hotel en el último año corrido. | FechaLlegada, FechaSalida, Cuarto. | Read |
| Consumo, Usuario | Mostrar el consumo en HotelAndes por un cliente, en un rango de fechas indicado. | Usuario: Nombre;  Consumo: Fecha, Costo | Read |
| Consumo, Servicio, Usuario | Consultar consumo en HotelAndes | ALL Consumo ALL Usuario | Read |

## Cuantificación de las consultas

Para la cuantificación de operaciones de lectura y escritura, tiendo en cuenta los siguientes datos:

* Creación o modificación de tipos de habitación: aproximadamente 1 vez por semana.
* Consulta de tipos de habitación: aproximadamente 1 vez por semana.
* Creación o modificación de habitaciones: aproximadamente 2 veces por semana.
* Consulta de habitaciones: aproximadamente 2 veces por semana.
* Creación o modificación de servicios: aproximadamente 2 veces por semana.
* Consulta de servicios: aproximadamente 2 veces por semana.
* Creación o modificación de una reserva: aproximadamente 100 veces al día.
* Consulta de una reserva: aproximadamente 50 veces al día.
* Creación o modificación de un consumo: aproximadamente 500 veces al día.
* Consulta de un consumo: aproximadamente 250 veces al día.
* Creación o modificación de una llegada o salida: aproximadamente 100 veces al día.
* Consulta de una llegada o salida: aproximadamente 50 veces al día.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entities | Operations | Information Needed | Type | Rate |
| TipoHabitacion | CRUD over TipoHabitacion | ALL TipoHabitacion | Write/Read | 1/week |
| Cuarto | CRUD over Cuarto | ALL Cuarto | Write/Read | 2/week |
| Servicio | CRUD over Servicio | ALL Servicio | Write/Read | 2/week |
| CuartoReserva | CRUD over Reservas | ALL CuartoReserva | Write/Read | 100/day  50/day |
| CuartoReserva | CRUD over entrada/salida del hotel | ALL CuartoReserva | Write/Read | 500/day 250/day |
| Consumo, Servicio, CuartoReserva | CRUD over servicios | Consumo: Servicios ALL Servicio CuartoReserva: \_id | Write/Read | 100/day 50/day |
| Consumo, Cuarto, CuartoReserva | Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido. | Consumo: Costo; Cuarto: \_id; CuartoReserva: Consumo, Cuarto. | Read | 1/year |
| CuartoReserva | Mostrar el índice de ocupación de cada una de las habitaciones del hotel en el último año corrido. | FechaLlegada, FechaSalida, Cuarto. | Read | 1/year |
| Consumo, Usuario | Mostrar el consumo en HotelAndes por un cliente, en un rango de fechas indicado. | Usuario: Nombre;  Consumo: Fecha, Costo | Read | 1/month |
| Consumo, Servicio, Usuario | Consultar consumo en HotelAndes | ALL Consumo ALL Usuario | Read | 1/week |

## Análisis de cardinalidad

Para la identificación de atributos y entidades se realizó el siguiente análisis sobre su cardinalidad:

* Por la cantidad de atributos y frecuencia de actualización, modificación y creación: **CuartoReserva**, **Usuario**, **Consumo**.
* **Cuarto** y **TipoHabitacion** se decidieron usar como colecciones debido a las operaciones CRUD que son necesarias sobre estas.
* **Tipo de habitación** esta referenciada en **habitación**.
* **CuartoReserva** debería tener referenciado **cuarto**, el cual a su vez debe tener embebido **los elementos de la habitación.**
* **CuartoReserva** debe tener referenciados los **consumos**, los consumos deben tener embebidos los **productos.** Los consumos tienen sentido solo si están asociados a una reserva.
* Por parte de los **servicios**, serán referenciados dentro de **consumo**, del cual **servicios** tendrá embebido a **producto**, tambiéntendrá embebido a **sala** y **sala** tendrá embebida a **reservacion**.
* El **plan de consumo** también esta embebido **en CuartoReserva,** estos planes están relacionados a las reservas realizadas, cuando estas son creadas.
* Los **usuarios** deben tener embebido el **TipoUsuario.**

# Modelo Documental

Resultamos con el siguiente modelo documental:

Imagen que contiene Diagrama

Descripción generada automáticamente

# Modelo en JSON

Con la siguiente estructura en JSON:

{

"Servicio": {

"Nombre": "String",

"Horario": "Date",

"Usuario": {"type": "ObjectId", "ref": "Usuario"},

"Producto": [{

"\_id": "ObjectId",

"Costo": "Integer",

"Tipo": "String"

}],

"Sala": [{

"\_id": "ObjectId",

"Tipo": "String",

"Capacidad": "Integer",

"Reservacion": [{

"\_id": "ObjectId",

"Fecha": "Date",

"Duracion": "String"

}]

}]

},

"Consumo": {

"\_id": "ObjectId",

"CuartoReserva": {"type": "ObjectId", "ref": "CuartoReserva"},

"Servicio": [{"type": "ObjectId", "ref": "Servicio"}],

"Costo": "Integer",

"Tipo": "String",

"Fecha": "Date"

},

"Usuario": {

"\_id": "ObjectId",

"Contraseña": "String",

"Nombre": "String",

"Email": "String",

"TipoUsuario": {

"\_id": "ObjectId",

"Nombre": "String"

}

},

"CuartoReserva": {

"\_id": "ObjectId",

"FechaLlegada": "Date",

"FechaSalida": "Date",

"NumPersonas": "Integer",

"CheckIn": "Boolean",

"CheckOut": "Boolean",

"Cuenta": "Integer",

"Consumo": {"type": "ObjectId", "ref": "Consumo"},

"Usuario": [{"type": "ObjectId", "ref": "Usuario"}],

"Cuarto": [{"type": "ObjectId", "ref": "Cuarto"}],

"PlanConsumo": {

"\_id": "ObjectId",

"Nombre": "String"

}

},

"Cuarto": {

"\_id": "ObjectId",

"Disponibilidad": "Boolean",

"TipoHabitacion": {"type": "ObjectId", "ref": "TipoHabitacion"},

"ElementoHabitacion": {

"\_id": "ObjectId",

"Nombre": "String",

"Costo": "Integer"

}

},

"TipoHabitacion": {

"\_id": "ObjectId",

"Nombre": "String"

}

}

# Ejemplo del Modelo

Con estos Datos podemos visualizar ejemplos como:

{

"Usuario": {

"\_id": "5fbd5b3d7e131f24ad4b84e5",

"Contraseña": "miContraseña123",

"Nombre": "Juan Perez",

"Email": "juan@example.com",

"TipoUsuario": {

"\_id": "5fbd5b3d7e131f24ad4b84eA",

"Nombre": "Cliente"

}

},

"TipoHabitacion": {

"\_id": "5fbd5b3d7e131f24ad4b84e8",

"Nombre": "Suite"

},

"Cuarto": {

"\_id": "5fbd5b3d7e131f24ad4b84e6",

"Disponibilidad": true,

"TipoHabitacion": "5fbd5b3d7e131f24ad4b84e8",

"ElementoHabitacion": [

{

"Nombre": "TV",

"Costo": 50

},

{

"Nombre": "Minibar",

"Costo": 30

}

]

},

"CuartoReserva": {

"\_id": "5fbd5b3d7e131f24ad4b84e9",

"FechaLlegada": "2023-12-01T12:00:00Z",

"FechaSalida": "2023-12-05T10:00:00Z",

"NumPersonas": 2,

"CheckIn": false,

"CheckOut": false,

"Cuenta": 0,

"Consumo": "5fbd5b3d7e131f24ad4b84eC",

"Usuario": ["5fbd5b3d7e131f24ad4b84e5"],

"Cuarto": ["5fbd5b3d7e131f24ad4b84e6"],

"PlanConsumo": {

"\_id": "5fbd5b3d7e131f24ad4b84eB",

"Nombre": "Todo Incluido"

}

},

"Consumo": {

"\_id": "5fbd5b3d7e131f24ad4b84eC",

"CuartoReserva": "5fbd5b3d7e131f24ad4b84e9",

"Servicio": ["5fbd5b3d7e131f24ad4b84eD"],

"Costo": 100,

"Tipo": "Alimentos",

"Fecha": "2023-12-03T19:30:00Z"

},

"Servicio": {

"\_id": "5fbd5b3d7e131f24ad4b84eD",

"Nombre": "Spa",

"Horario": "2023-12-03T14:00:00Z",

"Usuario": "5fbd5b3d7e131f24ad4b84e5",

"Producto": [

{

"Costo": 50,

"Tipo": "Masaje"

}

],

"Sala": [

{

"Tipo": "Privada",

"Capacidad": 2,

"Costo": 30,

"Reservacion": [

{

"Fecha": "2023-12-03T14:00:00Z",

"Duracion": "1 hora"

}

]

}

]

}

}

# Esquema de Validación

Con el siguiente esquema de validación:

{

$jsonSchema: {

bsonType: 'object',

properties: {

Servicio: {

bsonType: 'object',

required: [

'Nombre',

'Horario',

'Usuario'

],

properties: {

Nombre: {

bsonType: 'string'

},

Horario: {

bsonType: 'date'

},

Usuario: {

bsonType: 'objectId'

},

Producto: {

bsonType: 'array',

items: {

bsonType: 'object',

properties: {

Costo: {

bsonType: 'int'

},

Tipo: {

bsonType: 'string'

}

}

}

},

Sala: {

bsonType: 'array',

items: {

bsonType: 'object',

properties: {

Tipo: {

bsonType: 'string'

},

Capacidad: {

bsonType: 'int'

},

Costo: {

bsonType: 'int'

},

Reservacion: {

bsonType: 'array',

items: {

bsonType: 'object',

properties: {

Fecha: {

bsonType: 'date'

},

Duracion: {

bsonType: 'string'

}

}

}

}

}

}

}

}

},

Consumo: {

bsonType: 'object',

required: [

'CuartoReserva',

'Servicio',

'Costo',

'Tipo',

'Fecha'

],

properties: {

CuartoReserva: {

bsonType: 'objectId'

},

Servicio: {

bsonType: 'array',

items: {

bsonType: 'objectId'

}

},

Costo: {

bsonType: 'int'

},

Tipo: {

bsonType: 'string'

},

Fecha: {

bsonType: 'date'

}

}

},

Usuario: {

bsonType: 'object',

required: [

'Contraseña',

'Nombre',

'Email',

'TipoUsuario'

],

properties: {

'Contraseña': {

bsonType: 'string'

},

Nombre: {

bsonType: 'string'

},

Email: {

bsonType: 'string'

},

TipoUsuario: {

bsonType: 'object',

properties: {

Nombre: {

bsonType: 'string'

}

}

}

}

},

CuartoReserva: {

bsonType: 'object',

required: [

'FechaLlegada',

'FechaSalida',

'NumPersonas',

'CheckIn',

'CheckOut',

'Cuenta',

'Consumo',

'Usuario',

'Cuarto',

'PlanConsumo'

],

properties: {

FechaLlegada: {

bsonType: 'date'

},

FechaSalida: {

bsonType: 'date'

},

NumPersonas: {

bsonType: 'int'

},

CheckIn: {

bsonType: 'bool'

},

CheckOut: {

bsonType: 'bool'

},

Cuenta: {

bsonType: 'int'

},

Consumo: {

bsonType: 'objectId'

},

Usuario: {

bsonType: 'array',

items: {

bsonType: 'objectId'

}

},

Cuarto: {

bsonType: 'array',

items: {

bsonType: 'objectId'

}

},

PlanConsumo: {

bsonType: 'object',

properties: {

Nombre: {

bsonType: 'string'

}

}

}

}

},

Cuarto: {

bsonType: 'object',

required: [

'Disponibilidad',

'TipoHabitacion',

'ElementoHabitacion'

],

properties: {

Disponibilidad: {

bsonType: 'bool'

},

TipoHabitacion: {

bsonType: 'objectId'

},

ElementoHabitacion: {

bsonType: 'array',

items: {

bsonType: 'object',

properties: {

Nombre: {

bsonType: 'string'

},

Costo: {

bsonType: 'int'

}

}

}

}

}

},

TipoHabitacion: {

bsonType: 'object',

required: [

'Nombre'

],

properties: {

Nombre: {

bsonType: 'string'

}

}

}

}

}

}