

Iteración 3

Sistemas Transaccionales

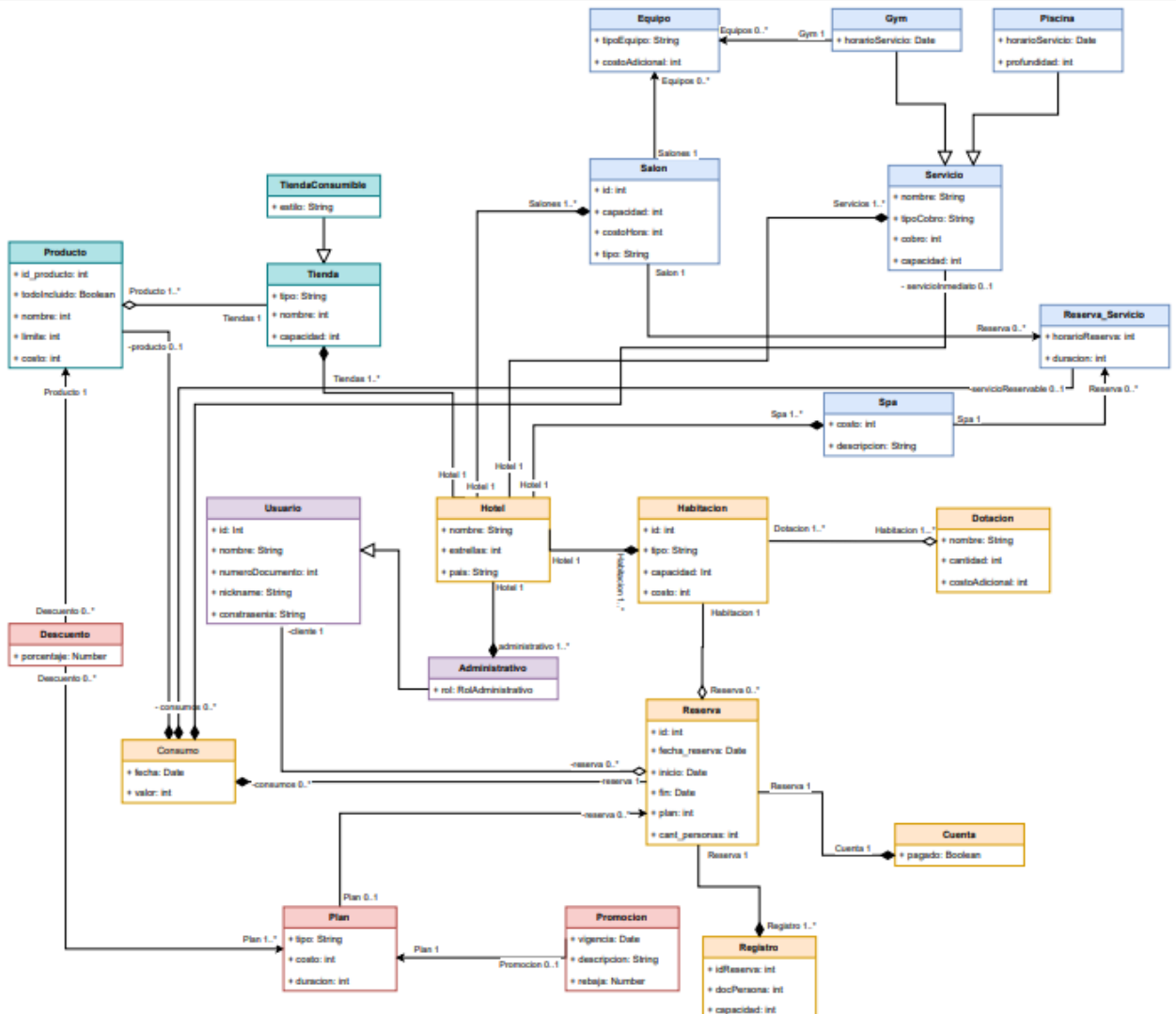
Daniela Parra Martinez - 202013036

Kevin David Álvarez Romero - 202022834

1. Análisis

Teniendo en cuenta los nuevos requerimientos en esta iteración 3 nuestros diagramas de UML y Relacional son acordes con estos requerimientos. (Ver diagrama al final del documento).

Diagrama UML:



2. Entidades y atributos:

Entidad	Atributos
Hotel	Id, nombre, estrellas, país.
Tienda	tipo, nombre, capacidad.
Productos	id_producto, incluido, nombre, limite, costo.
Habitación	id, tipo, capacidad, costo.

Reserva	idreserva, fecha_reserva, inicio, fin, cant_personas.
Consumo	Id, fecha, costo.
Cuenta	Id, Pagado.
Registro	doc_persona, capacidad.
Spa	Id, costo, descripción.
Reserva servicio	Id, horario, duración.
Salón	id, tipo, costo, capacidad.
Gym	Id
Piscina	Id, profundidad
Equipo	Id, tipoEquipo, costoAdicional.
Dotaciones	Id, nombre, cantidad, costo adicional
Servicios	Id, nombre, capacidad, tipoCobro, cobro.

3. Análisis de operaciones de lectura y escritura

Entidades	Operaciones	Información necesaria	Tipo
HOTEL	Consultar el consumo de un hotel	nombre	Leer
HABITACIÓN	CRUD de habitación y tipo de habitación	tipo + id	leer / escribir
RESERVA	CRUD de una reserva	id + inicio + fin	leer / escribir
CONSUMO	CRUD del consumo de un	fecha, valor, id reserva, producto	leer / escribir
	Consultar la cantidad de dinero (consumo de un cliente del hotel)		
CUENTA		pagado	Leer
REGISTRO	Consultar el consumo por un cliente	idreserva + docpersona	Leer
SPA	CRUD del servicio de Spa	costo	leer / escribir
SALON	CRUD del servicio de Salon	id + costohora	leer / escribir
GYM	CRUD del servicio de Gym	nombre + cobro	leer / escribir
PISCINA	CRUD del servicio de Piscina	nombre + cobro	leer / escribir
EQUIPO	CRUD del servicio de equipos	costo adicional	leer / escribir
TIENDA	CRUD del servicio de tienda	tipo + producto {idproducto, costo}	leer / escribir

4. Cuantificación de operaciones de lectura y escritura

Entidades	Operaciones	Información necesaria	Tipo	Tasa
HOTEL	Consultar el consumo de un hotel	nombre	Leer	-
HABITACIÓN	CRUD de habitación y tipo de habitación	tipo + id	leer / escribir	3/por semana(tipohabitación) + 3/2porsemana (habitación)
RESERVA	CRUD de una reserva	id + inicio + fin	leer / escribir	3/ 2 por semana
CONSUMO	CRUD del consumo de un	fecha, valor, id reserva, producto	leer / escribir	2/500 cada día + 1/250 cada día
	Consultar la cantidad de dinero (consumo de un cliente del hotel)			
CUENTA		pagado	Leer	2/500 cada día + 1/250 cada día
REGISTRO	Consultar el consumo por un cliente	idreserva + docpersona	Leer	2/100 cada día + 1/50 cada día
SPA	CRUD del servicio de Spa	costo	leer / escribir	2/2cada semana + 1/2 cada semana
SALON	CRUD del servicio de Salon	id + costohora	leer / escribir	2/2cada semana + 1/2 cada semana
GYM	CRUD del servicio de Gym	nombre + cobro	leer / escribir	2/2cada semana + 1/2 cada semana
PISCINA	CRUD del servicio de Piscina	nombre + cobro	leer / escribir	2/2cada semana + 1/2 cada semana
EQUIPO	CRUD del servicio de equipos	costo adicional	leer / escribir	2/2cada semana + 1/2 cada semana
TIENDA	CRUD del servicio de tienda	tipo + producto {idproducto, costo}	leer / escribir	2/2cada semana + 1/2 cada semana
			tasa:	#total de operaciones / tiempo total

5. Descripción de las entidades:

Hotel: Representa un establecimiento de alojamiento con atributos como identificador único (Id), nombre, número de estrellas que lo clasifican y el país en el que se encuentra.

Tienda: Modela una tienda con características como tipo (por ejemplo, ropa, electrónicos), nombre y capacidad para almacenar productos.

Productos: Describe productos con atributos como identificador único (Id_Producto), indicador de inclusión, nombre del producto, límite de unidades disponibles y su costo.

Habitación: Representa una habitación de un hotel con detalles como identificador único (Id), tipo de habitación, capacidad máxima de personas y costo por noche.

Reserva: Registra la reserva de una habitación en un hotel con detalles como identificador único de reserva (IdReserva), fecha de reserva, fecha de inicio y fin de la estancia, y la cantidad de personas incluidas.

Consumo: Registra el consumo de servicios o productos con atributos como identificador único (Id), fecha del consumo y costo asociado.

Cuenta: Representa la cuenta asociada a un cliente que puede contener consumos y otros cargos, con detalles como identificador único (Id) y un indicador de si la cuenta ha sido pagada.

Registro: Contiene información de registro para personas, con detalles como documento de identificación de la persona (Doc_Persona) y capacidad relacionada.

Spa: Representa un servicio de spa con atributos como identificador único (Id), costo del servicio y una descripción del spa.

Reserva Servicio: Registra la reserva de un servicio adicional, como en un spa o salon, con atributos como identificador único (Id), horario de reserva y duración.

Salón: Describe un salón de eventos con características como identificador único (Id), tipo de salón, costo y capacidad máxima.

Gym: Representa un gimnasio con un identificador único (Id).

Piscina: Representa una piscina con atributos como identificador único (Id) y profundidad.

Equipo: Describe un equipo con atributos como identificador único (Id), tipo de equipo y costo adicional asociado.

Dotaciones: Representa dotaciones o extras de una habitación con detalles como identificador único (Id), nombre, cantidad disponible y costo adicional.

Servicios: Describe servicios adicionales con atributos como identificador único (Id), nombre del servicio, capacidad, tipo de cobro y monto del cobro.

6. Relaciones entre entidades, su cardinalidad y esquema de asociación

Relaciones	Cardinalidad	Tipo
consumos - reserva	* --- 1	referenciada
consumos - reservaServicios	* --- 1	referenciada
consumos - producto	* --- 1	referenciada
consumos - servicios	* --- 1	referenciada
equipos - gym	* --- *	embebida
equipos - salones	* --- *	embebida
gym - servicio	* --- 1	referenciada
habitaciones - hotel	* --- 1	referenciada
habitaciones - dotaciones	* --- *	embebida
piscina - servicio	* --- 1	referenciada
productos - tienda	* --- 1	referenciada
registros - reserva	* --- 1	referenciada
reservaServicio - spa	* --- 1	referenciada
reservaServicio - salon	* --- 1	referenciada
reserva - habitacion	* --- 1	referenciada
reserva - cuenta	* --- 1	referenciada

7. Esquemas embebidos y referenciados

Consumo → Reserva | Reserva servicio | Producto | Servicio

```
{
  "reservaServicio": {
    "id": 11,
    "horario": "21/03/2023",
    "duracion": "21/08/2023",
  },
  "reserva": {
    "id_reserva": 1,
    "fechaReserva": "21/03/2023",
    "inicio": "21/08/2023",
    "fin": "30/03/2023",
    "cantidadPersonas": 4
  },
  "producto": {
    "id_producto": 1,
    "nombre": "shampoo",
    "limite": "5",
    "costo": "100000"
  },
  "servicio": {
    "id": 1,
    "nombre": "consumo",
    "capacidad": 10,
    "tipoCobro": "adicional",
    "cobro": 40000
  },
  "consumos": {
    "id": 1,
    "fecha": "24/08/2023",
    "costo": 58000,
    "id_reserva": 1, // Referencia a la reserva
    "id_reservaServicio": 11, // Referencia a la reservaServicio
  },
  "id_producto": 1, // Referencia al producto
  "id_servicio": 1 // Referencia al servicio
},
}
```

Equipos → Salones

```
{
  "Salones": {
    "idSalon": 101,
    "tipo": "Reuniones",
    "costo": 100,
    "capacidad": 50,
    "equipos": [
      {
        "idEquipo": 301,
        "tipoEquipo": "Proyector",
        "costoAdicional": 30
      },
      {
        "idEquipo": 302,
        "tipoEquipo": "Micrófono",
        "costoAdicional": 20
      }
    ]
  }
}
```

Equipos --> Gym

```
{
  "Gym": {
    "idGym": 201,
    "Equipos": [
      {
        "idEquipo": 302,
        "tipoEquipo": "Máquina de pesas",
        "costoAdicional": 50,
      }
    ]
  }
}
```

Gym --> Servicio

```
{
  "servicio": {
    "id": 1,
    "nombre": "consumo",
    "capacidad": 10,
    "tipoCobro": "adicional",
    "cobro": 40000
  },
  "Gym": {
    "idGym": 201,
    "id_servicio": 1 // Referencia al servicio
  }
}
```

Habitaciones --> Hotel

```
{
  "Hotel": {
    "idHotel": 1,
    "nombre": "Hotel Ejemplo"
  },
  "Habitacion": {
    "idHabitacion": 401,
    "tipo": "sencilla",
    "capacidad": 2,
    "costo": 50000,
    "idHotel": 1 // Referencia al Hotel
  }
}
```

Habitaciones --> Dotaciones

```
{
  "Habitacion": {
    "idHabitacion": 401,
    "tipo": "sencilla",
    "capacidad": 2,
    "costo": 50000,
    "Dotaciones": [
      {
        "idDotacion": 302,
        "nombre": "Cama",
        "cantidad": 2,
        "costoAdicional": 0,
      }
    ]
  }
}
```

Piscina --> Servicio

```
{
  "servicio": {
    "id": 1,
    "nombre": "consumo",
    "capacidad": 10,
    "tipoCobro": "adicional",
    "cobro": 40000
  },
  "Piscina": {
    "idPiscina": 201,
    "profundidad": 170,
    "id_servicio": 1 // Referencia al servicio
  }
}
```

Producto --> Tienda

```
{
  "Tienda": {
    "id": 1,
    "nombre": "deluno",
    "capacidad": 10,
    "tipo": "consumo"
  },
  "producto": {
    "id_producto": 1,
    "nombre": "shampoo",
    "limite": "5",
    "costo": "100000",
    "idTienda": 1 // Referencia a la tienda
  }
}
```

Registros --> Reserva

```
{
  "reserva": {
    "id_reserva": 1,
    "fechaReserva": "21/03/2023",
    "inicio": "21/08/2023",
    "fin": "30/03/2023",
    "cantidadPersonas": 4
  },
  "registro": {
    "docpersona": 1112223334,
    "capacidad": 4,
    "idReserva": 1 // Referencia a la reserva
  }
}
```

ReservaServicios--> Spa | Salones

```
{
  "Spa": {
    "id": 118,
    "costo": 50000,
    "descripcion": "Relajante",
  },
  "Salones": {
    "idSalon": 124,
    "tipo": "Reuniones",
    "costo": 100,
    "capacidad": 50,
  },
  "reservaServicio": {
    "id": 11,
    "horario": "21/03/2023",
    "duracion": "21/08/2023",
    "id_Spa": "118",
    "id_salon": null,
  }
}
```

Reserva--> Habitación | Cuenta

```

{
  "cuenta": {
    "idPagado": 1,
    "pagado": "Si"
  },
  "Habitacion": {
    "idHabitacion": 401,
    "tipo": "sencilla",
    "capacidad": 2,
    "costo": 50000,
  },
  "reserva": {
    "id_reserva": 1,
    "fechaReserva": "21/03/2023",
    "inicio": "21/08/2023",
    "fin": "30/03/2023",
    "cantidadPersonas": 4,
    "idPagado": 1, // Referencia a la cuenta
    "idHabitacion": 401 // Referencia a la habitacion
  }
}

```

8. Requerimientos

RF1 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR TIPO DE HABITACIÓN

```

//Crear/Actualizar el tipo de una Habitación:
db.habitacion.updateOne(
  { idHabitacion: 1 },
  { $set: { tipo: "NuevoTipo" } },
  { upsert: true }
);

//Consultar Habitaciones por Tipo:
db.habitacion.find({ tipo: "NuevoTipo" });

//Borrar el tipo de una Habitación:
db.habitacion.updateOne(
  { idHabitacion: 1 },
  { $unset: { tipo: "" } }
);

```

RF2 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR HABITACIÓN


```

//////////////////////////////////RF2//////////////////////////////////
//Crear (Insertar) una Habitación:
db.habitacion.insertOne({
  idHabitacion: 1,
  tipo: "Ejemplo",
  capacidad: 2,
  costo: 100,
  idHotel: 1,
  dotaciones: [
    { idDotacion: 1, nombre: "Wifi", cantidad: 1, costoAdicional: 0 },
  ]
});

//Consultar Habitación
db.habitacion.findOne({ idHabitacion: 1 });

//Actualizar Habitación
db.habitacion.updateOne(
  { idHabitacion: 1 },
  {
    $set: {
      tipo: "NuevoTipo",
      capacidad: 3,
      costo: 120,
      "dotaciones.0.cantidad": 2,
    }
  }
);

//Borrar Habitación
db.habitacion.deleteOne({ idHabitacion: 1 });

```

RF3 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UN SERVICIO DEL HOTEL

```

//////////////////////////////////RF3//////////////////////////////////
//Registrar (Insertar) un Servicio:
db.spa.insertOne({
  id: 1,
  costo: 50,
  descripcion: "Paquete de relajación con masaje y sauna"
});

db.piscina.insertOne({
  idPiscina: 1,
  profundidad: 2,
  id_servicio: 1 // Asociado al servicio de piscina
});

db.tienda.insertOne({
  id: 1,
  nombre: "Tienda de regalos",
  capacidad: 50,
  tipo: "Souvenirs"
});

db.gym.insertOne({
  idGym: 1,
  id_servicio: 5, // Asociado al servicio de gimnasio
  equipos: [
    { idEquipo: 1, tipoEquipo: "Caminadora", costoAdicional: 10 },
    // Otros equipos...
  ]
});

```

```

//Consultar Servicio
db.spa.find({ id: 1 });

db.piscina.find({ idPiscina: 1 });

db.tienda.find({ id: 1 });

db.gym.find({ idGym: 1 });

//Actualizar Servicio
db.spa.updateOne(
  { id: 1 },
  { $set: { costo: 60, descripcion: "Nuevo paquete de spa" } }
);

db.piscina.updateOne(
  { idPiscina: 1 },
  { $set: { profundidad: 3 } }
);

db.tienda.updateOne(
  { id: 1 },
  { $set: { capacidad: 60 } }
);

db.gym.updateOne(
  { idGym: 1 },
  { $set: { "equipos.0.costoAdicional": 15 } }
);

```

```

//Borrar Servicio
db.spa.deleteOne({ id: 1 });

db.piscina.deleteOne({ idPiscina: 1 });

db.tienda.deleteOne({ id: 1 });

db.gym.deleteOne({ idGym: 1 });

```

RF4 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UNA RESERVA DE ALOJAMIENTO

```

////////////////////////////////////RF4////////////////////////////////////
//Registrar (Insertar) una Reserva
db.reserva.insertOne({
  id_reserva: 1,
  fechareserva: new Date(),
  inicio: new Date(),
  fin: new Date(),
  cantidadPersonas: 2,
  idPagado: 1,
  idHabitacion: 101
});

//Consultar Reserva
db.reserva.find({ id_reserva: 1 });

//Actualizar Reserva
db.reserva.updateOne(
  { id_reserva: 1 },
  { $set: { cantidadPersonas: 3 } }
);

//Borrar Reserva
db.reserva.deleteOne({ id_reserva: 1 });

```

RF5 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR LA LLEGADA DE UN CLIENTE AL HOTEL

```
//////////////////////////////////RF5//////////////////////////////////
// Crear o actualizar el inicio de la reserva con id_reserva 1
db.reserva.updateOne(
  { id_reserva: 1 },
  { $set: { inicio: new Date() } },
  { upsert: true }
);

// Consultar reservas con una fecha de inicio específica (por ejemplo, hoy)
const fechaActual = new Date();
db.reserva.find({ inicio: { $gte: fechaActual.setHours(0, 0, 0, 0),
  $lt: fechaActual.setHours(23, 59, 59, 999) } });

// Actualizar la fecha de inicio de la reserva con id_reserva 1
db.reserva.updateOne(
  { id_reserva: 1 },
  { $set: { inicio: new Date() } }
);

// Borrar el inicio de la reserva con id_reserva 1
db.reserva.updateOne(
  { id_reserva: 1 },
  { $unset: { inicio: "" } }
);
```

RF6 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR UN CONSUMO DE UN SERVICIO DEL HOTEL POR PARTE DE UN CLIENTE O SUS ACOMPAÑANTES

```
//////////////////////////////////RF6//////////////////////////////////
//Registrar (Insertar) un Consumo
db.consumos.insertOne({
  id: 1,
  fecha: new Date(),
  costo: 20,
  id_reserva: 1,
  id_reservaServicio: 1,
  id_producto: 1,
  id_servicio: 1
});

//Consultar Consumo
db.consumos.find({ id: 1 });

//Actualizar Consumo
db.consumos.updateOne(
  { id: 1 },
  { $set: { costo: 25 } }
);

//Borrar Consumo
db.consumos.deleteOne({ id: 1 });
```

RF7 - REGISTRAR / ACTUALIZAR / BORRAR / CONSULTAR LA SALIDA DE UN CLIENTE

```
//////////RF7//////////
//Crear/Actualizar fin de una Reserva
db.reserva.updateOne(
  { id_reserva: 1 },
  { $set: { fin: new Date() } },
  { upsert: true } // upsert permite insertar si no se encuentra
);
// Consultar una fecha de fin específica (por ejemplo, hoy)
const fechaActual = new Date();
db.reserva.find({ fin: { $gte: fechaActual.setHours(0, 0, 0, 0),
  $lt: fechaActual.setHours(23, 59, 59, 999) } });

// Actualizar la fecha de fin de la reserva con id_reserva 1
db.reserva.updateOne(
  { id_reserva: 1 },
  { $set: { fin: new Date() } }
);

// Borrar fin de la reserva con id_reserva 1
db.reserva.updateOne(
  { id_reserva: 1 },
  { $unset: { fin: "" } }
);
```

RFC1 – MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACION EN EL ULTIMO AÑO CORRIDO

```
//////////RFC1//////////
db.consumos.aggregate([
  {
    $lookup: {
      from: "reservaServicio",
      localField: "id_reservaServicio",
      foreignField: "id",
      as: "reservaservicio"
    }
  },
  {
    $lookup: {
      from: "habitacion",
      localField: "reservaservicio.id_habitacion",
      foreignField: "idhabitacion",
      as: "habitacion"
    }
  },
  {
    $match: {
      "fecha": {
        $gte: ISODate("2022-12-03"),
        $lt: ISODate("2023-12-03")
      }
    }
  },
  {
    $group: {
      _id: "$habitacion.idHabitacion",
      total: {
        $sum: "$costo"
      }
    }
  }
])
```

RFC1 – MOSTRAR EL INDICE DE OCUPACION DE CADA UNA DE LAS HABITACIONES DEL HOTEL EN EL ULTIMO AÑO CORRIDO

```
//////////////////////////////////RFC2//////////////////////////////////  
// Fecha actual  
var fechaActual = new Date();  
  
// Fecha de inicio del último año corrido  
var fechaInicioUltimoAnio = new Date();  
fechaInicioUltimoAnio.setFullYear(fechaActual.getFullYear() - 1);
```

```
// Agregación para calcular el índice de ocupación  
var resultado = db.reserva.aggregate([  
  {  
    $match: {  
      "inicio": { $gte: fechaInicioUltimoAnio, $lt: fechaActual }  
    }  
  },  
  {  
    $group: {  
      _id: "$idHabitacion",  
      totalNoches: { $sum: { $divide: [{ $subtract: ["$fin", "$inicio"] }, 1000 * 60 * 60 * 24] } }  
    }  
  },  
  {  
    $lookup: {  
      from: "habitacion",  
      localField: "_id",  
      foreignField: "idHabitacion",  
      as: "habitacion"  
    }  
  },  
  {  
    $unwind: "$habitacion"  
  },  
  {  
    $project: {  
      _id: 0,  
      "Habitacion": "$_id",  
      "Tipo": "$habitacion.tipo",  
      "Capacidad": "$habitacion.capacidad",  
      "Costo": "$habitacion.costo",  
      "IndiceOcupacion": { $multiply: [{ $divide: ["$totalNoches", { $multiply: ["$habitacion.capacidad", 365] } ] }, 100] }  
    }  
  }  
]);  
  
// Mostrar resultados  
resultado.forEach(function (habitacion) {  
  printjson(habitacion);  
});
```

RFC3 - MOSTRAR EL CONSUMO EN HOTELANDES POR UN CLIENTE, EN UN RANGO DE FECHAS INDICADO.

```

//////////RFC3//////////
// Variables para el rango de fechas y el nombre del hotel
var fechaInicio = ISODate("2023-01-01");
var fechaFin = ISODate("2023-12-31");
var nombreHotel = "HOTELANDES";

// ID del cliente que deseas consultar
var idCliente = 123; // Reemplaza con el ID del cliente deseado

// Consulta de consumos en el hotel específico, para el cliente y en el rango de fechas
db.consumos.aggregate([
  {
    $match: {
      "fecha": { $gte: fechaInicio, $lte: fechaFin },
      "id_reservaServicio": {
        $in: db.reservaServicio.find([
          "id_spa": { $in: db.spa.find({ "idHotel": { $in: db.hotel.find({ "nombre": nombreHotel })}.map(h => h.idHotel) } )}.map(s => s.id) },
          "id_salon": { $in: db.salones.find({ "equipos.tipoEquipo": "Silla de Masaje" }).map(s => s.idSalon) }
        ]).map(rs => rs.id)
      }
    }
  },
  {
    $lookup: {
      from: "reservaServicio",
      localField: "id_reservaServicio",
      foreignField: "id",
      as: "reservaServicio"
    }
  },
  {
    $lookup: {
      from: "reserva",
      localField: "id_reserva",
      foreignField: "id_reserva",
      as: "reserva"
    }
  },
  {
    $lookup: {
      from: "producto",
      localField: "id_producto",
      foreignField: "id_producto",
      as: "producto"
    }
  },
  {
    $lookup: {
      from: "servicio",
      localField: "id_servicio",
      foreignField: "id",
      as: "servicio"
    }
  },
  {
    $lookup: {
      from: "habitacion",
      localField: "reserva.idHabitacion",
      foreignField: "idHabitacion",
      as: "habitacion"
    }
  },
  {
    $lookup: {
      from: "hotel",
      localField: "habitacion.idHotel",
      foreignField: "idHotel",
      as: "hotel"
    }
  },
  {
    $match: {
      "reserva.idUsuario": idCliente,
      "hotel.nombre": nombreHotel
    }
  },
  {
    $project: {
      "id_cliente": idCliente,
      "id_producto": "$producto._id",
      "id_servicio": "$servicio._id",
      "id_habitacion": "$habitacion._id",
      "id_hotel": "$hotel._id",
      "id_reserva": "$reserva._id",
      "id_reserva_servicio": "$reservaServicio._id",
      "fecha": "$fecha",
      "consumo": "$consumo"
    }
  }
])

```

```

{
  $lookup: {
    from: "producto",
    localField: "id_producto",
    foreignField: "id_producto",
    as: "producto"
  }
},
{
  $lookup: {
    from: "servicio",
    localField: "id_servicio",
    foreignField: "id",
    as: "servicio"
  }
},
{
  $lookup: {
    from: "habitacion",
    localField: "reserva.idHabitacion",
    foreignField: "idHabitacion",
    as: "habitacion"
  }
},
{
  $lookup: {
    from: "hotel",
    localField: "habitacion.idHotel",
    foreignField: "idHotel",
    as: "hotel"
  }
},
{
  $match: {
    "reserva.idUsuario": idCliente,
    "hotel.nombre": nombreHotel
  }
},
{
  $project: {
    "id_cliente": idCliente,
    "id_producto": "$producto._id",
    "id_servicio": "$servicio._id",
    "id_habitacion": "$habitacion._id",
    "id_hotel": "$hotel._id",
    "id_reserva": "$reserva._id",
    "id_reserva_servicio": "$reservaServicio._id",
    "fecha": "$fecha",
    "consumo": "$consumo"
  }
}

```

```

    {
      $project: {
        _id: 0,
        "Fecha": "$fecha",
        "Servicio": "$servicio.nombre",
        "Producto": "$producto.nombre",
        "Costo": "$costo"
      }
    },
    {
      $sort: {
        "Fecha": 1
      }
    }
  ]
});

```

RFC4 - CONSULTAR CONSUMO EN HOTELANDES

```

//////////RFC4//////////
// Variables para el rango de fechas y el nombre del hotel
var fechaInicio = ISODate("2022-01-01");
var fechaFin = ISODate("2022-12-31");
var nombreHotel = "HOTELANDES 4";

// Consulta de consumos en el hotel específico y en el rango de fechas
db.consumos.aggregate([
  {
    $match: {
      "fecha": { $gte: fechaInicio, $lte: fechaFin }
    }
  },
  {
    $lookup: {
      from: "reservaServicio",
      localField: "id_reservaServicio",
      foreignField: "id",
      as: "reservaServicio"
    }
  },
  {
    $lookup: {
      from: "servicio",
      localField: "reservaServicio.id_servicio",
      foreignField: "id",
      as: "servicio"
    }
  },
  {
    $lookup: {
      from: "reserva",
      localField: "id_reserva",
      foreignField: "id_reserva",
      as: "reserva"
    }
  }
]);

```



```

{
  $lookup: {
    from: "habitacion",
    localField: "reserva.idHabitacion",
    foreignField: "idHabitacion",
    as: "habitacion"
  }
},
{
  $lookup: {
    from: "hotel",
    localField: "habitacion.idHotel",
    foreignField: "idHotel",
    as: "hotel"
  }
},
{
  $match: {
    "hotel.nombre": nombreHotel
  }
},
{
  $project: {
    _id: 0,
    "Cliente": "$reserva.idUsuario",
    "Fecha": "$fecha",
    "Servicio": "$servicio.nombre",
    "Veces Utilizado": 1
  }
},
{
  $group: {
    _id: {
      Cliente: "$Cliente",
      Servicio: "$Servicio"
    },
    "Fecha": { $first: "$Fecha" },
    "Veces Utilizado": { $sum: 1 }
  }
},

```

```

{
  $sort: {
    // Aquí puedes especificar el criterio de clasificación deseado
    "_id.Cliente": 1,
    "Fecha": 1,
    "Veces Utilizado": -1
  }
}
]);

```