

## Universidad de Los Andes

### Sistemas Transaccionales

#### Documento de Análisis: Iteración 03

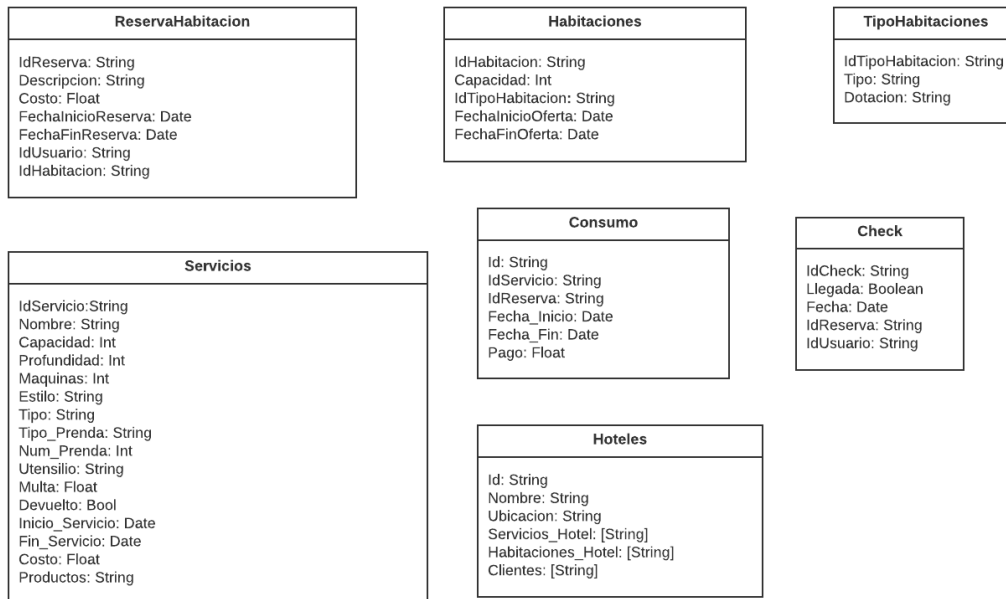
Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

#### Documento de Análisis: Iteración 03

### Análisis y Modelo Conceptual



### Diseño de la Base de Datos

#### Análisis de la carga de trabajo (workload)

#### Entidades y Atributos

Entidad	Atributos
ReservaHabitacion	<ul style="list-style-type: none"><li>• IdReserva: String</li><li>• Descripcion: String</li><li>• Costo: Float</li><li>• FechaInicioReserva: Date</li><li>• FechaFinReserva: Date</li><li>• IdUsuario: String</li><li>• IdHabitacion: String</li></ul>
Servicios	<ul style="list-style-type: none"><li>• IdServicio:String</li><li>• Nombre: String</li><li>• Capacidad: Int</li><li>• Profundidad: Int</li><li>• Maquinas: Int</li><li>• Estilo: String</li><li>• Tipo: String</li></ul>

## Universidad de Los Andes

### Sistemas Transaccionales

Documento de Análisis: Iteración 03

Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

	<ul style="list-style-type: none"><li>• Tipo_Prenda: String</li><li>• Num_Prenda: Int</li><li>• Utensilio: String</li><li>• Multa: Float</li><li>• Devuelto: Bool</li><li>• Inicio_Servicio: Date</li><li>• Fin_Servicio: Date</li><li>• Costo: Float</li><li>• Productos: String</li></ul>
Habitaciones	<ul style="list-style-type: none"><li>• IdHabitacion: String</li><li>• Capacidad: Int</li><li>• IdTipoHabitacion: String</li><li>• FechaInicioOferta: Date</li><li>• FechaFinOferta: Date</li></ul>
Hoteles	<ul style="list-style-type: none"><li>• Id: String</li><li>• Nombre: String</li><li>• Ubicacion: String</li><li>• Servicios_Hotel: [String]</li><li>• Habitaciones_Hotel: [String]</li><li>• Clientes: [String]</li></ul>
TipoHabitaciones	<ul style="list-style-type: none"><li>• IdTipoHabitacion: String</li><li>• Tipo: String</li><li>• Dotacion: String</li></ul>
Consumo	<ul style="list-style-type: none"><li>• Id: String</li><li>• IdServicio: String</li><li>• IdReserva: String</li><li>• Fecha_Inicio: Date</li><li>• Fecha_Fin: Date</li><li>• Pago: Float</li></ul>
Check	<ul style="list-style-type: none"><li>• IdCheck: String</li><li>• Llegada: Bool</li><li>• IdReserva: String</li><li>• IdUsuario: String</li></ul>

*Cantidad de registros que tendría la BD para cada una de las entidades*

Entidad	Cantidad de Registros
ReservaHabitacion	50.000

**Universidad de Los Andes****Sistemas Transaccionales**

Documento de Análisis: Iteración 03

Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

Servicios	35
Habitaciones	200
Hoteles	1
TipoHabitaciones	20
Consumo	250.000
Check	50.000

*Análisis de operaciones de lectura y escritura para cada entidad*

Entidades	Operaciones	Información necesaria	Tipo
TipoHabitaciones	Consultar tipo de habitación	Detalles tipo de habitación	Lectura
Habitaciones	Consultar habitación	Detalles habitación	Lectura
Servicios	Consultar servicio de hotel	Detalles servicio	Lectura
ReservaHabitacion	Consultar reserva de alojamiento	Detalles reserva de habitación	Lectura
Check	Consultar llegada y salida de un cliente	Detalles check	Lectura
Consumo	Consultar consumo de un servicio de hotel	Detalles consumo	Lectura
Consumo, ReservaHabitacion	Buscar el dinero recolectado por servicios en cada habitación	Fecha y habitación de reserva + pago por consumo	Lectura
ReservaHabitacion, Habitacion	Buscar el índice de ocupación de cada habitación	Detalles reserva + Capacidad de cada habitacion	Lectura
Consumo	Buscar el consumo por un cliente	Fecha de consumo	Lectura
TipoHabitaciones	Añadir/Editar tipo de habitación	Detalles tipo de habitación	Escritura
Habitaciones	Añadir/Editar habitación	Detalles habitación + idTipoHabitacion	Escritura
Servicios	Añadir/Editar servicio de hotel	Detalles servicio	Escritura
ReservaHabitacion	Añadir/Editar reserva de alojamiento	Detalles reserva de habitación + idHabitacion + idUsuario	Escritura
Check	Añadir/Editar llegada y salida de un cliente	Detalles check + idReserva + idUsuario	Escritura

## Universidad de Los Andes

### Sistemas Transaccionales

Documento de Análisis: Iteración 03

Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

Consumo	Añadir/Editar consumo de un servicio de hotel	Detalles consumo + idServicio + idReserva	Escritura
---------	---	---	-----------

#### Descripción de las entidades de datos y las relaciones entre ellas

##### *Lista de entidades*

- ReservaHabitacion: Representa las reservas realizadas para una habitación. La habitación y el usuario que hizo la reserva se referencia desde esta entidad.
- Servicios: Representa todos los servicios existentes de todos los hoteles de la aplicación.
- Habitaciones: Representa todas las habitaciones existentes en la aplicación. La habitación es referenciada en la entidad “ReservaHabitación” y las características especiales (Tipo de habitación y dotación) se referencian desde esta entidad.
- Hoteles: Representa todos los hoteles existentes en la aplicación. Contiene la información relevante de cada hotel, los servicios que presta el hotel, las habitaciones del hotel y los usuarios registrados.
- TipoHabitaciones: Representa el tipo de habitación y la dotación de la misma. Esta entidad es referenciada en la entidad: “Habitaciones”.
- Consumo: Representa el consumo de cada habitación, que está asociada a una determinada reserva.
- Check: Representa la llegada o la salida de un determinado usuario al hotel.
- Usuarios: Representa los usuarios relevantes para la aplicación.

##### *Relaciones entre entidades y cardinalidad*

Entidades	Cardinalidad	Descripción
Hoteles – Servicios	Uno a Muchos	Un hotel tiene muchos servicios
Consumo – Servicios	Uno a Uno	Un consumo tiene un servicio asociado
Hoteles – Habitaciones	Uno a Muchos	Un hotel tiene muchas habitaciones
ReservaHabitacion – Consumo	Uno a Muchos	Una reserva puede tener muchos consumos asociados
Habitaciones – ReservaHabitacion	Uno a Muchos	Una habitación tiene solamente una reserva asociada, pero una reserva puede estar asociada a muchas habitaciones (Se puede reservar la habitación múltiples veces)
TipoHabitacion – Habitacion	Uno a Muchos	Muchas habitaciones pueden tener el mismo tipo de habitación
Reserva – Check	Uno a Muchos	Una reserva puede tener muchos Checks – in y Checks – out asociados

## Universidad de Los Andes

### Sistemas Transaccionales

#### Documento de Análisis: Iteración 03

Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

#### Creación de las colecciones principales en MongoDB

##### *Creación de colecciones*

- Checks:

```
> db.createCollection("checks",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["llegada", "fecha", "idReserva"],
      properties:{
        llegada: {bsonType: "bool"},
        fecha: {bsonType: "date"},
        idReserva: {bsonType: "string"}}}}});
< { ok: 1 }
```

- Hoteles:

```
> db.createCollection("hoteles",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["nombre", "ubicacion"],
      properties:{
        nombre: {bsonType: "string"},
        ubicacion: {bsonType: "string"},
        servicios_hotel: {bsonType: "array",
                          items:{
                            bsonType:"string"}
                          }
      }
    }
  }
});
< { ok: 1 }
```

- Consumos:

```
> db.createCollection("consumos",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["idServicio", "idReserva", "fecha_inicio", "fecha_fin", "pago"],
      properties:{
        idServicio: {bsonType: "string"},
        idReserva: {bsonType: "string"},
        fecha_inicio: {bsonType: "date"},
        fecha_fin: {bsonType: "date"},
        pago: {bsonType: "double"}}}}});
< { ok: 1 }
```

- TiposHabitación:

```
> db.createCollection("tipos_habitacion",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["tipo", "dotacion"],
      properties:{
        tipo: {bsonType: "string"},
        dotacion: {bsonType: "string"}}}}});
< { ok: 1 }
```

- Habitaciones:

```
> db.createCollection("habitaciones",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["capacidad", "idTipoHabitacion", "fecha_inicio_oferta", "fecha_fin_oferta"],
      properties:{
        capacidad: {bsonType: "int"},
        idTipoHabitacion: {bsonType: "string"},
        fecha_inicio_oferta: {bsonType: "date"},
        fecha_fin_oferta: {bsonType: "date"}}}}});
< { ok: 1 }
```

- ReservaHabitaciones:

```
> db.createCollection("reserva_habitaciones",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["descripcion", "costo", "fecha_inicio_reserva", "fecha_fin_reserva", "idHabitacion"],
      properties:{
        descripcion: {bsonType: "string"},
        costo: {bsonType: "double"},
        fecha_inicio_reserva: {bsonType: "date"},
        fecha_fin_reserva: {bsonType: "date"},
        idHabitacion: {bsonType: "string"}}}}});
< { ok: 1 }
```

- Servicios:

```
> db.createCollection("servicios",{
  validator:{
    $jsonSchema:{
      bsonType: "object",
      required: ["nombre"],
      properties:{
        nombre: {bsonType: "string"},
        capacidad: {bsonType: "int"},
        profundidad: {bsonType: "int"},
        maquinas: {bsonType: "int"},
        estilo: {bsonType: "string"},
        tipo: {bsonType: "string"},
        tipo_prenda: {bsonType: "string"},
        num_prenda: {bsonType: "int"},
        utensilio: {bsonType: "string"},
        multa: {bsonType: "double"},
        devuelto: {bsonType: "bool"},
        inicio_servicio: {bsonType: "date"},
        fin_Servicio: {bsonType: "date"},
        costo: {bsonType: "double"},
        productos: {bsonType: "string"}}}}});
```

### **Escenarios de Prueba**

Para el crud del RF1 al RF7 se pueden cargar datos al azar pensados por el usuario y se ve de la siguiente manera:

Juan Felipe Caraballo Umbarila – 201923741

## RF5 y RF7



# Universidad de Los Andes

## Sistemas Transaccionales

### Documento de Análisis: Iteración 03

Juliana Sofía Ahumada Arcos – 201921471

Sebastián Umaña Peinado – 202013778

Juan Felipe Caraballo Umbarila – 201923741

#### Llegadas y Salidas

ID	Llegada	Reserva	Usuario
<a href="#">Agregar Datos</a> <a href="#">Ver Datos</a>			

RF6

#### Consumos

ID	ID de Servicio	ID de Reserva	ID de Habitación	Inicio	Fin	Pago
<a href="#">Agregar Consumo</a> <a href="#">Ver Datos</a>						

RF8

#### Dinero Recolectado por Habitación en 2023

ID Habitación	Total Recolectado
<a href="#">Ver Datos</a>	

RF9

#### Totales por Habitación

ID Habitación	Total
<a href="#">Ver Datos</a>	

Por dificultades con la carga de datos, que Java no recibía las fechas que mongoDB tenía en sus datos, pudimos cargar la base de datos, pero no se muestran los datos por un error en las fechas:

```
2023-12-03T23:44:02.472-05:00 ERROR 22828 --- [nio-8080-exec-3] o.a.c.c.C.[./].[dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: org.springframework.core.convert.ConversionFailedException: Failed to convert from type [java.lang.String] to type [java.util.Date] for value [2023-06-09T00:00:00.000+00:00]] with root cause
java.lang.IllegalStateException: null
```

Sin embargo, el crud funciona perfectamente y los requerimientos 8 y 9 también, pero dependen de los datos. Por lo que nuestro escenario de pruebas se limita a que el usuario agregue datos, borre, actualice y visualice.

El r10 está implementado, pero no funcional y el rfc11 no está implementado.