

## **Documento de informe Iteración 3**

### **Sistemas transaccionales**

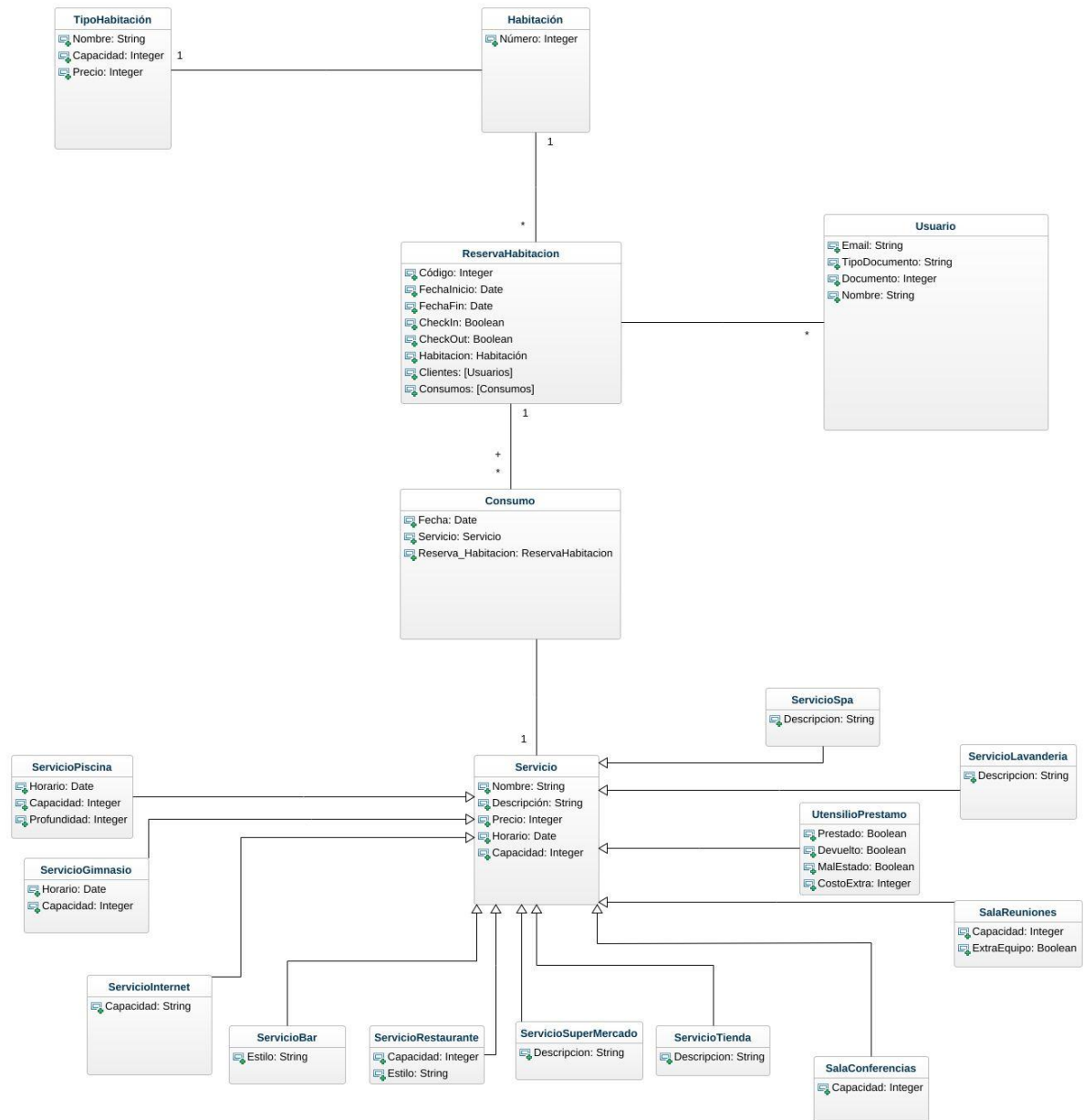
**Luis Felipe Plazas – 202013155 - l.plazasp**

**David Almanza -202011293 – d.almanza**

**Nicolás Lara Gómez – 202012455 – n.lara**

### **Diseño de la base de datos**

# 1. Análisis y modelo conceptual



## 2. Diseño de la base de datos

### a. Análisis de la carga de trabajo

#### a) Identifique identidades y sus atributos

Se identificaron las siguientes entidades, listadas con cada uno de sus atributos:

*Los atributos escritos en itálica pueden ir o no, dependiendo del documento.* Se obvia el PK\_id.

- **Habitación:**
  - (String) Número: El número de la Habitación.
- **Tipo\_Habitación:**
  - (String) Nombre: Nombre del tipo de habitación.
  - (Integer) Capacidad: Capacidad de personas del tipo de habitación
  - (Integer) Precio: Precio de la reserva por el tipo de habitación
- **Usuario:**
  - (Integer) Documento: Número de documento del usuario
  - (String) TipoDocumento: Tipo de documento del usuario
  - (String) Email: Email del usuario
  - (String) Nombre: Nombre del usuario
- **Servicio:**
  - (String) Nombre: Nombre del servicio
  - (String) Descripción: Una pequeña descripción del servicio
  - (Integer) Precio: Precio del servicio. Puede ser 0.
  - (Date) *Horario*: Horario de uso del servicio
  - (Integer) *Capacidad*: Capacidad de personas usando el servicio
- **Reserva\_Habitación:**
  - (Date) FechaInicio: Fecha de inicio del tiempo de reserva
  - (Date) FechaFin: Fecha de fin del tiempo de reserva
  - (Boolean) CheckIn: Para saber si el cliente ya hizo CheckIn
  - (Boolean) CheckOut: Para saber si el cliente ya hizo CheckOut
  - (Habitación) Habitación: Habitación reservada
  - (array[Usuario]) Clientes: Usuarios de la reserva de habitación
  - (array[Consumo]) Consumos: Consumos cargados a la habitación
- **Consumo:**
  - (Date) Fecha: Fecha en la que se hizo el consumo del servicio
  - (Servicio) Servicio: Servicio consumido
  - (Reserva\_Habitación) Reserva\_Habitación: Habitación a la que se hará el cobro

**b) Cuantifique las entidades**

- **Habitación:** Se estima que por hotel hay alrededor de 500 habitaciones, esto para soportar la elevada cantidad de reservaciones que hay en el hotel por año
- **Tipo\_Habitación:** Se estima que por hotel hay alrededor de 20 tipos de habitación
- **Usuario:** Teniendo en cuenta que se pueden tener hasta 17.000 reservas por año, se estima que la cantidad de usuarios puede ser de unos 30.000, suponiendo que el hotel lleva tan solo unos 3 años de funcionamiento.
- **Servicio:** Tan solo hay alrededor de 35 servicios por hotel
- **Reserva\_Habitación:** Se estiman alrededor de 17.000 reservas anuales, y suponiendo, una vez más, que el hotel tiene alrededor de 3 años de funcionamiento, se estima que en total hay 50.000 reservas.
- **Consumo:** Debido a la alta cantidad de reservas de habitación, por año se estiman unos 100.000 consumos.

**c) Analice y cuantifique las operaciones de lectura y escritura para cada entidad**

*Se resuelven los numerales c) y d) al mismo tiempo aquí.*

- **Habitación**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Habitación, Tipo_Habitación	Creación	Toda la info pertinente para crear una habitación	Escritura	1/semana
Habitación, Tipo_Habitación	Actualización	La info que se quiera actualizar para la habitación	Escritura	1/semana
Habitación	Borrado	_id de la habitación	Escritura	1/semana
Habitación	Consulta	Parámetro por el que se quiera buscar la habitación	Lectura	1/semana
Reserva_Habitación, Habitación, Consumo	RFC1: Dinero recolectado por una habitación	_id de la habitación a consultar	Lectura	1/año
Reserva_Habitación, Habitación	RFC2: Índice de ocupación de la habitación	_id de la habitación a consultar	Lectura	1/año
Reserva_Habitación, Habitación	RFC6: Habitaciones más y menos solicitadas	-	Lectura	1/año

- **Tipo\_Habitación**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Tipo_Habitación	Creación	Toda la info pertinente para crear un tipo de habitación	Escritura	1/semana
Tipo_Habitación	Actualización	La info que se quiera actualizar para el tipo de habitación	Escritura	1/semana
Tipo_Habitación	Borrado	_id del tipo de habitación	Escritura	1/semana
Tipo_Habitación	Consulta	Parámetro por el que se quiera buscar el tipo de habitación	Lectura	1/semana

- **Usuario**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Usuario	Creación	Toda la info pertinente para crear un Usuario	Escritura	7/semana
Usuario	Actualización	La info que se quiera actualizar para el Usuario	Escritura	1/semana
Usuario	Borrado	_id del Usuario	Escritura	1/semana
Usuario	Consulta	Parámetro por el que se quiera buscar al Usuario	Lectura	10/semana
Usuario, Reserva_Habitación, Consumo	RFC3: Consumos por cliente en rango de fechas	Fecha de inicio, fecha de fin y _id del usuario	Lectura	1/mes

- **Servicio**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Servicio	Creación	Toda la info pertinente para crear un Servicio	Escritura	2/semana
Servicio	Actualización	La info que se quiera actualizar para el Servicio	Escritura	1/semana

Servicio	Borrado	id del Servicio	Escritura	1/año
Servicio	Consulta	Parámetro por el que se quiera buscar al Servicio	Lectura	2/semana

- **Reserva\_Habitación**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Reserva_Habitación, Habitación, Usuario	Creación	Toda la info pertinente para crear una Reserva Habitación	Escritura	50/día
Reserva_Habitación	Actualización	La info que se quiera actualizar para la Reserva Habitación	Escritura	10/semana
Reserva_Habitación	Borrado	_id de la reserva de Habitación	Escritura	10/semana
Reserva_Habitación	Consulta	Parámetro por el que se quiera buscar la Reserva Habitación	Lectura	50/día
Reserva_Habitación, Usuarios	RFC7: Consulta de clientes excelentes	No se necesita información para esta consulta. Todo está en la BD	Lectura	1/trimestre

- **Consumo**

Entidades	Operaciones	Información Necesitada	Tipo	Tasa
Reserva_Habitación, Consumo	Creación	Toda la info pertinente para crear un Consumo	Escritura	125/día
Consumo, Reserva_Habitación	Actualización	La info que se quiera actualizar para al Consumo	Escritura	125/día
Consumo, Reserva_Habitación	Borrado	_id del Consumo	Escritura	10/día
Reserva_Habitación, Consumo	Consulta	Parámetro por el que se quiera buscar al Consumo e _id de la reserva de habitación	Lectura	250/día

Consumo, Servicio	RFC4: Consultar consumo en hotel	Rango de fechas, parámetros para la clasificación de los resultados	Lectura	1/semestre
Consumo, Servicio	RFC5: Consultar consumo en hotel	Rango de fechas, parámetros para la clasificación de los resultados	Lectura	1/semestre
Consumo, Servicio	RFC6: Consultar servicios más y menos consumidos	-	Lectura	1/año

## b. Entidades de datos y relaciones

### a) Lista de entidades con la descripción de cada una de ellas

Se identificaron las siguientes entidades:

- **Habitación:**

Una habitación en el sistema se identifica por un número único que la distingue de las demás. Este número actúa como un identificador exclusivo para cada habitación en el establecimiento. No se especifica un atributo de tipo, capacidad o precio en esta entidad, ya que estos detalles están asociados al tipo de habitación específico reservado por los usuarios.

- **Tipo\_Habitación:**

El tipo de habitación define las características particulares de las distintas opciones de alojamiento disponibles. Cada tipo de habitación tiene un nombre descriptivo, indicando la categoría o especificación, junto con la capacidad de personas que puede albergar y el precio asociado a la reserva. Estos detalles son esenciales para que los usuarios elijan la opción más adecuada a sus necesidades y presupuesto.

- **Usuario:**

La entidad de usuario almacena información clave sobre los clientes del sistema. Cada usuario se identifica por un número de documento único, y se registran detalles como el tipo de documento, el correo electrónico y el nombre del usuario. Estos datos son fundamentales para gestionar las reservas y proporcionar un servicio personalizado, facilitando la comunicación y la identificación de cada cliente.

- **Servicio:**

Los servicios representan las ofertas adicionales que los usuarios pueden solicitar durante su estancia. Cada servicio tiene un nombre descriptivo, una breve explicación de lo que implica y su precio asociado. Además, se registra el horario de disponibilidad y la capacidad máxima de personas que pueden utilizar el servicio simultáneamente. Estos detalles son cruciales para ofrecer opciones personalizadas y garantizar una experiencia cómoda para los clientes.

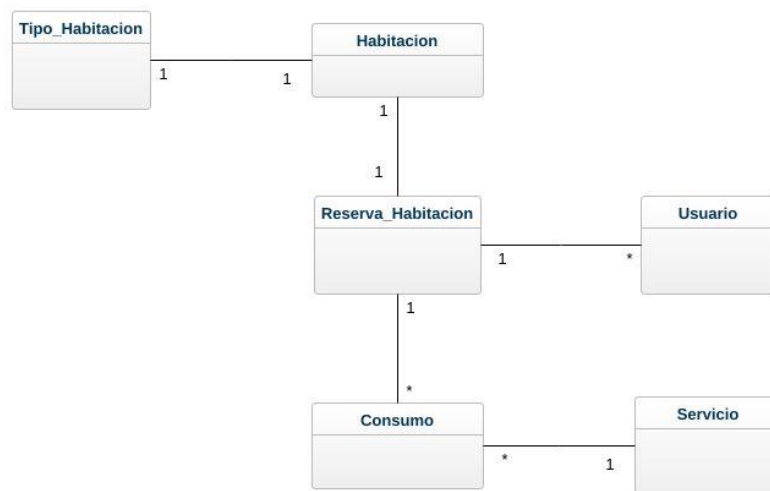
- **Reserva\_Habitación:**

La reserva de habitación es una entidad central que gestiona el tiempo de ocupación de una habitación específica. Se registra la fecha de inicio y fin de la reserva, junto con indicadores de CheckIn y CheckOut para conocer el estado de la ocupación. La entidad incluye la habitación reservada, así como la lista de usuarios asociados a la reserva y los consumos realizados durante el tiempo especificado.

- **Consumo:**

Los consumos registran las transacciones relacionadas con los servicios utilizados durante una reserva de habitación. Se registra la fecha en que se realizó el consumo, el servicio específico utilizado y la habitación a la que se aplicará el cobro. Esta entidad facilita un seguimiento detallado de los gastos adicionales realizados por los usuarios durante su estancia, contribuyendo a una gestión financiera eficiente.

**b) Relaciones entre entidades y su cardinalidad**



**c) Análisis de selección de esquema de asociación.**



### **Embebido:**

#### **Habitación:**

- Embebido en: Reserva\_Habitación.
- Justificación: La información de la habitación está directamente relacionada con una reserva específica, y embeberla en la reserva facilita el acceso a todos los detalles de la reserva sin necesidad de consultas adicionales.

#### **Tipo\_Habitación:**

- Embebido en: Habitación.
- Justificación: Al embeber los detalles del tipo de habitación dentro de la entidad "Habitacion", se mejora la eficiencia del acceso, ya que permite recuperar toda la información necesaria sobre una habitación, incluyendo su tipo, en una única consulta. Esto simplifica el modelado y la recuperación de datos, ya que los detalles del tipo de habitación son parte integral de la entidad habitación y suelen ser accedidos en conjunto, evitando la necesidad de realizar consultas adicionales para obtener información relacionada.

#### **Servicio:**

- Embebido en: Consumo.
- Justificación: Al embeber los detalles del servicio en la entidad "Consumo", se mejora la eficiencia del acceso, ya que permite recuperar toda la información necesaria sobre un consumo, incluyendo los detalles del servicio, en una única consulta. Esto simplifica el modelado y la recuperación de datos, ya que los detalles del servicio son parte integral de la entidad consumo y suelen ser accedidos en conjunto, evitando la necesidad de realizar consultas adicionales para obtener información relacionada. Además, al embeber el servicio, se asegura la coherencia y consistencia de los datos, ya que la información del servicio consumido está directamente incluida en el documento de consumo, evitando la necesidad de buscar referencias externas.

#### **Usuario:**

- Embebido en: Reserva\_Habitación (Colección "reservas\_habitacion").
- Justificación: Al embeber los detalles del usuario en la entidad "Reserva\_Habitación", se mejora la eficiencia del acceso, ya que permite recuperar toda la información necesaria sobre una reserva, incluyendo los detalles del usuario, en una única consulta. Esto simplifica el modelado y la recuperación de datos, ya que los detalles del usuario son parte integral de la entidad reserva y suelen ser accedidos en conjunto, evitando la necesidad de realizar consultas adicionales para obtener información relacionada. Además, al embeber el usuario, se asegura la coherencia y consistencia de los datos, ya que la información del usuario está directamente incluida en el documento de reserva, evitando la necesidad de buscar referencias externas.

### **Referenciado:**

### Reserva\_Habitación:

- Referencia a: Consumo
- Justificación: Se utilizan referencias para evitar la duplicación de datos y mantener un tamaño de documento razonable, ya que la información de los usuarios puede ser extensa y cambiar con el tiempo.

### Consumo:

- Referencia a: reserva\_habitación
- Justificación: La referencia a la colección "reservas\_habitacion" se emplea para evitar la duplicación de datos y mantener un tamaño de documento razonable. Dado que un consumo está asociado a una reserva de habitación específica, referenciar la reserva permite una gestión más eficiente de la información. Además, al separar la información de la reserva de la información del consumo, se facilita la actualización independiente de cada entidad, garantizando la integridad de los datos y una mayor flexibilidad para realizar consultas específicas sobre consumos o reservas por separado.

### d. Descripción gráfica JSON

```
"reserva_habitacion": { "_id": 1,
  "FechaInicio": "01/01/2023",
  "FechaFin": "02/02/2023",
  "CheckIn": true,
  "CheckOut": false,
  "habitacion": {
    "_id": 2,
    "numero": 69,
    "tipo_habitacion": {
      "_id": 3,
      "nombre": "Suite doble",
      "capacidad": 2,
      "precio": 1000
    },
    "clientes": [
      {
        "_id": 4,
        "documento": 999,
        "TipoDocumento": "Cédula",
        "Email": "hola@gmail.com",
        "nombre": "Juan Pérez"
      },
      {
        "_id": 5,
        "documento": 888,
        "TipoDocumento": "Cédula",
        "Email": "chao@gmail.com",
        "nombre": "Nicolás Mayorga"
      }
    ],
    "Consumos": [6]
  }
}
```

```
"consumo": { "_id": 6,
  "Fecha": "01/02/2023",
  "Servicio": {
    "_id": 7,
    "Nombre": "Piscina",
    "Descripcion": "servicio de piscina",
    "Precio": 100,
    "Horario": "08:00-18:00",
    "Capacidad": 50
  },
  "Reserva_habitacion": 1
}
```

### **c. Base de datos en MongoDB**

Para ello se crearon dos colecciones, la colección de ReservaHabitacion, y la colección de consumo.

Los esquemas de validación para ello fueron los siguientes:

#### **Colección ReservaHabitación:**

```
{
  $jsonSchema: {
    bsonType: 'object',
    required: [
      'codigo',
      'fecha_inicio',
      'fecha_fin',
      'check_in',
      'check_out',
      'habitacion'
    ],
    properties: {
      codigo: {
        bsonType: 'int'
      },
      fecha_inicio: {
        bsonType: 'date'
      },
      fecha_fin: {
        bsonType: 'date'
      },
      check_in: {
```

```
    bsonType: 'bool'
  },
  check_out: {
    bsonType: 'bool'
  },
  habitacion: {
    bsonType: 'object',
    required: [
      'numero',
      'tipo_habitacion'
    ]
  },
  clientes: {
    bsonType: 'array',
    items: {
      bsonType: 'object',
      required: [
        'nombre',
        'email',
        'tipo_documento',
        'documento'
      ]
    }
  },
  consumos: {
    bsonType: 'array',
    items: {
      bsonType: 'objectId',
      description: 'Debe ser el id de un consumo'
    }
  }
```

```
    }  
  }  
}
```

### **Colección Consumo:**

```
{  
  $jsonSchema: {  
    bsonType: 'object',  
    required: [  
      'fecha',  
      'servicio',  
      'reserva_habitacion'  
    ],  
    properties: {  
      fecha: {  
        bsonType: 'date'  
      },  
      servicio: {  
        bsonType: 'object',  
        required: [  
          'nombre',  
          'descripcion',  
          'precio',  
          'horario',  
          'capacidad'  
        ],  
      },  
      reserva_habitacion: {
```

```

    bsonType: 'objectId',
    description: 'Debe ser el id de una reserva'
  }
}
}
}

```

Se intentó crear una reserva de una habitación, pero con datos que no cumplían el esquema de validación, el resultado fue el siguiente:

```

> db.ReservaHabitacion.insertOne({"codigo": 32,
  "fecha_inicio": "10/02/2023",
  "fecha_fin": "15/03/2023", "check_in":true, "check_out":true,
  "habitacion":
  {
    "numero": 25
  }
});

```

✖ ▶ MongoServerError: Document failed validation

Después, Se intentó crear un consumo con atributos que no estaban en el esquema de validación, y el resultado fue el siguiente:

```

> db.Consumo.insertOne({"codigo": 32,
  "fecha_inicio": "10/02/2023",
  "fecha_fin": "15/03/2023", "check_in":true, "check_out":true,
  "habitacion":
  {
    "numero": 25
  }
});

```

✖ ▶ MongoServerError: Document failed validation

También se deja un ejemplo de documentos que si cumplen con los esquemas de validación para cada colección:

```
db.ReservaHabitacion.insertOne({"codigo": 32,  
  "fecha_inicio": (2023, 0, 1),  
  "fecha_fin": (2023, 0, 1), "check_in":true, "check_out":true,  
  "habitacion":  
  {  
    "numero": 25, "tipo_habitacion":"suite"  
  }});
```

```
db.Consumo.insertOne({"fecha": (2023,02,01),  
  "servicio":  
  {  
    "nombre": "spa", "descripcion":"servicio de spa", "precio":500,  
    "horario": "am", "capacidad":20  
  }, "reserva_habitacion": 32});
```