

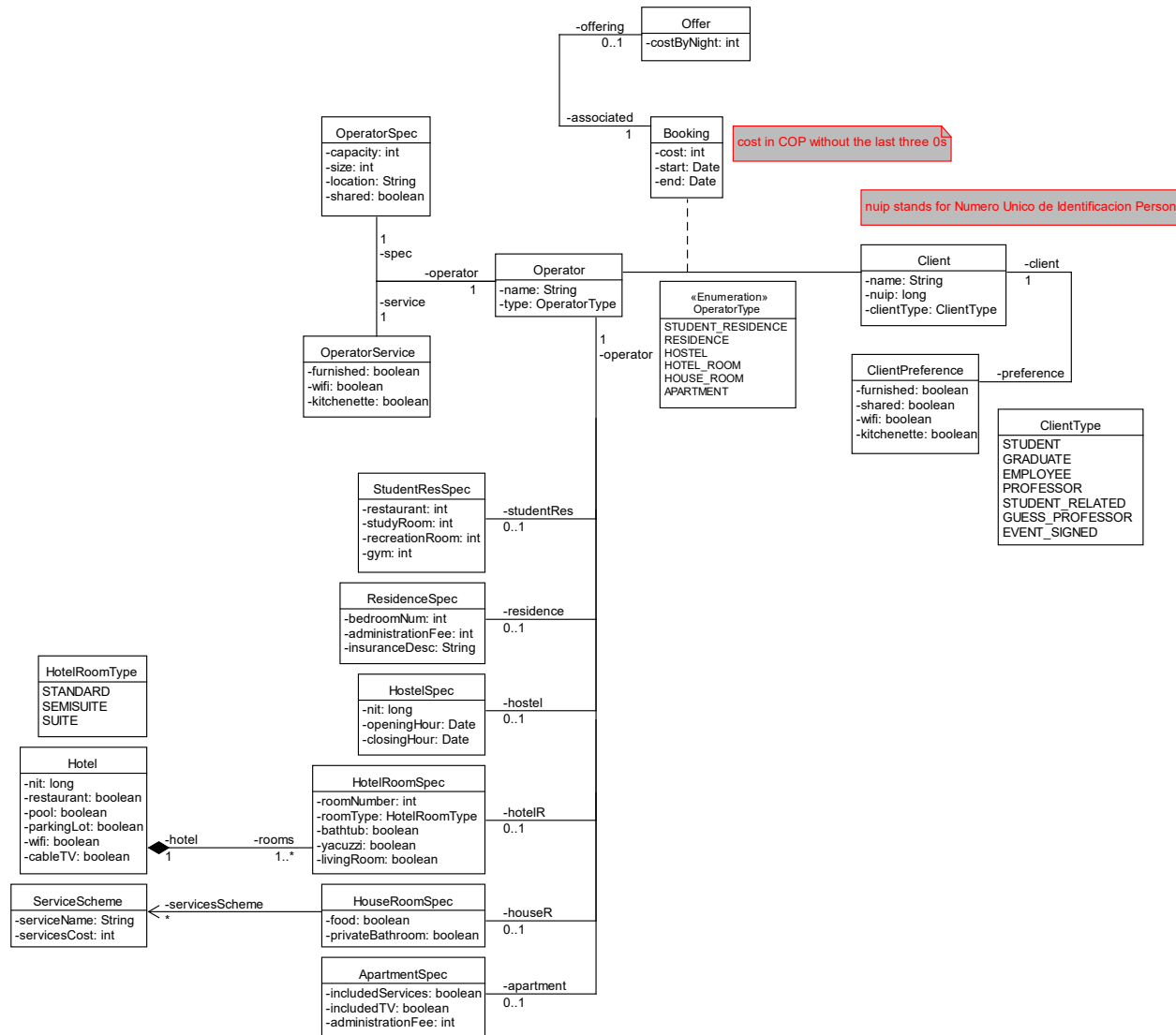
# Iteración 1

Pedro L. Lobato Barros, Nicolas D. Camargo Prieto  
Universidad de los Andes, Bogotá, Colombia  
{p.lobato, n.camargop}@uniandes.edu.co  
Fecha de presentación: Febrero 26 de 2023

## Tabla de contenido

|   |   |    |
|---|---|----|
| 1 | Modelo Conceptual .....                                     | 2  |
| 2 | Modelo relacional.....                                      | 3  |
| 3 | Requerimientos funcionales .....                            | 6  |
| 4 | Diagramas de secuencia y Control de calidad del modelo..... | 12 |

# 1 Modelo Conceptual



La totalidad de las *clases* de este diagrama deben ser persistentes.

Operator, OperatorSpec, OperatorService y todos los tipos de Operator deben ser persistentes para permitir las modificaciones y la recuperación total de la información que contienen.

Client, ClientPreference, Booking y Offer deben ser persistentes para conservar la información para recuperar y modificar en cualquier momento.

Hotel y ServiceScheme deben ser persistentes para mantener la constancia en sus relaciones con los respectivos tipos a los que están asociados.

## 2 Modelo relacional

| Operator  |                     |
|---|---------------------|
| varchar   | varchar             |
| PK,UA   | NN,CK(OperatorType) |
| name  | type                |
| CK(OperatorType): type IN<br>(STUDENT_RESIDENCE,RESIDENCE,HOSTEL,HOTEL_ROOM,HOUSE_ROOM,APARTMENT) |                     |

| OperatorSpec         |                    |               |          |        |
|----------------------|--------------------|---------------|----------|--------|
| varchar              | number             | number        | varchar  | bit    |
| PK,FK(Operator.name) | NN,CK(capacity>=0) | NN,CK(size>0) | NN       | NN     |
| name                 | capacity           | size          | location | shared |

| OperatorService      |           |      |             |
|----------------------|-----------|------|-------------|
| varchar              | bit       | bit  | bit         |
| PK,FK(Operator.name) | NN        | NN   | NN          |
| name                 | furnished | wifi | kitchenette |

| Client  |          |                   |
|---|----------|-------------------|
| number  | varchar  | varchar           |
| PK,UA   | NN,ND,UA | NN,CK(ClientType) |
| nuip  | name     | clientType        |
| CK(ClientType): clientType IN<br>(STUDENT,GRADUATE,EMPLOYEE,PROFESSOR,STUDENT_RELATED,GUESS_PROFESSOR,EVENT_SIGNED) |          |                   |

| ClientPreference   |           |        |      |             |
|--------------------|-----------|--------|------|-------------|
| number             | bit       | bit    | bit  | bit         |
| PK,FK(Client.nuip) | NN        | NN     | NN   | NN          |
| nuip               | furnished | shared | wifi | kitchenette |

| Booking                      |                               |                            |               |       |            |
|------------------------------|-------------------------------|----------------------------|---------------|-------|------------|
| number                       | number                        | varchar                    | number        | date  | date       |
| PK,SA                        | CK(NDBooking),FK(Client.nuip) | CK(NDBooking),FK(Operator) | NN,CK(cost>0) | NN    | NN,CK(end) |
| id                           | nuip                          | name                       | cost          | start | end        |
| CK(NDBooking): ND(nuip,name) |                               |                            |               |       |            |
| CK(end): end>start           |                               |                            |               |       |            |

| Offer  |                      |
|--------|----------------------|
| number | number               |
| PK,SA  | NN,CK(costByNight>0) |
| id     | costByNight          |

| StudentResSpec       |                      |                     |                          |               |
|----------------------|----------------------|---------------------|--------------------------|---------------|
| varchar              | number               | number              | number                   | number        |
| PK,FK(Operator.name) | NN,CK(restaurant>=0) | NN,CK(studyRoom>=0) | NN,CK(recreationRoom>=0) | NN,CK(gym>=0) |
| name                 | restaurant           | studyRoom           | recreationRoom           | gym           |

| ResidenceSpec        |                     |                            |               |
|----------------------|---------------------|----------------------------|---------------|
| varchar              | number              | number                     | varchar       |
| PK,FK(Operator.name) | NN,CK(bedroomNum>0) | NN,CK(administrationFee>0) | NN            |
| name                 | bedroomNum          | administrationFee          | insuranceDesc |

| HostelSpec                       |        |             |             |
|----------------------------------|--------|-------------|-------------|
| varchar                          | number | timestamp   | timestamp   |
| PK,FK(Operator.name)             | NN,ND  | NN          | NN,CK(end)  |
| name                             | nit    | openingHour | closingHour |
| CK(end): closingHour>openingHour |        |             |             |

| HotelRoomSpec  |                        |                 |         |         |            |
|--|------------------------|-----------------|---------|---------|------------|
| varchar  | number                 | String          | bit     | bit     | bit        |
| PK,FK(Operator.name)                                 | NN,ND,CK(roomNumber>0) | NN,CK(roomType) | NN      | NN      | NN         |
| name   | roomNumber             | roomType        | bathtub | yacuzzi | livingRoom |
| CK(roomType): roomType IN (STANDARD,SEMISUITE,SUITE) |                        |                 |         |         |            |

| Hotel  |            |      |            |      |         |
|--------|------------|------|------------|------|---------|
| number | bit        | bit  | bit        | bit  | bit     |
| PK     | NN         | NN   | NN         | NN   | NN      |
| nit    | restaurant | pool | parkingLot | wifi | cableTV |

| RoomsHotel       |                                 |
|------------------|---------------------------------|
| number           | number                          |
| PK,FK(Hotel.nit) | PK,FK(HotelRoomSpec.roomNumber) |
| nit              | roomNumber                      |

| HouseRoomSpec        |      |                 |
|----------------------|------|-----------------|
| varchar              | bit  | bit             |
| PK,FK(Operator.name) | NN   | NN              |
| name                 | food | privateBathroom |

| ServiceScheme             |             |                    |
|---------------------------|-------------|--------------------|
| varchar                   | varchar     | number             |
| PK,FK(HouseRoomSpec.name) | PK          | CK(serviceCost>=0) |
| name                      | serviceName | serviceCost        |

| ApartmentSpec        |                  |            |                             |
|----------------------|------------------|------------|-----------------------------|
| varchar              | bit              | bit        | number                      |
| PK,FK(Operator.name) | NN               | NN         | NN,CK(administrationFee>=0) |
| name                 | includedServices | includedTV | administrationFee           |

1NF ✓: No hay valores compuestos ni relaciones en los valores

2NF ✓: Los atributos dependen de la totalidad de la PK

3NF ✓: Los atributos dependen únicamente de la PK

BCNF ✓: No hay múltiples llaves candidatas que se crucen

### 3 Requerimientos funcionales

|   |  |
|---|--|
| <b>Nombre</b>   | RF1. Registrar los operadores de alojamiento para AlohaAndes   |
| <b>Resumen</b>  | Se quiere poder registrar los operadores para alojamiento, teniendo en cuenta que debe pertenecer a uno de los convenios establecidos con empresas o personas naturales. |
| <b>Entradas</b>   |  |
| Nombre de la empresa o persona natural (operador): String   |  |
| Tipo de operador (Hotel, Hostal, Administrador de residencia universitaria, Vecino Fenicia, Empleado, Profesor, o Estudiante): String |  |
| Cedula o NIT: String  |  |
| <b>Resultados</b>   |  |
| Lista con la información ingresada y confirmación de registro exitoso.  |  |
| <b>RNF asociados</b>  |  |
| Persistencia: La información debe quedar registrada correctamente en la base de datos, para futuras consultas de usuarios.            |  |

|  |   |
|--|---|
| <b>Nombre</b>  | RF2. Registrar propuesta de alojamiento para AlohAndes  |
| <b>Resumen</b>   | Se quiere poder registrar una propuesta de alojamiento a un operador. Y se debe indicar el nombre, tipo de alojamiento, el precio, la capacidad, la ubicación, el tamaño del alojamiento, duración mínima de estadía y servicios adicionales. |
| <b>Entradas</b>  |   |
| NIT o Cedula de la empresa o persona natural (operador): String  |   |
| Tipo de alojamiento: String  |   |
| Precio: Int  |   |
| Capacidad de personas: int   |   |
| Ubicación: String  |   |
| Tamaño del alojamiento: String   |   |
| Duración mínima de estadía o contrato: String  |   |
| Servicios adicionales: String  |   |
| <b>Resultados</b>  |   |
| Lista con la información ingresada asociada al operador y confirmación de registro exitoso. Retorna un id de alojamiento.                  |   |
| <b>RNF asociados</b>   |   |
| Persistencia: La información debe quedar registrada correctamente en la base de datos, para futuras consultas de usuarios.                 |   |
| <b>Nombre</b>  | RF3. Registrar las personas habilitadas para utilizar los servicios   |
| <b>Resumen</b>   | Se quiere poder registrar un cliente para que pueda usar los servicios. El cliente debe tener algún vínculo con los Andes.  |
| <b>Entradas</b>  |   |
| Nombre de la persona: String   |   |
| Cedula: String   |   |
| Vinculo con la institución (estudiante, egresado, empleado, profesor, padre de estudiante, profesor invitado, persona registrada en evento |   |
| <b>Resultados</b>  |   |
| Se retorna una lista con la información de la persona y un mensaje de registro exitoso.  |   |
| <b>RNF asociados</b>   |   |
| Privacidad: Solo los usuarios autorizados pueden ver información de la persona.  |   |
| Persistencia: Los datos deben mantenerse en la base de datos.  |   |

|   |   |
|---|---|
| <b>Nombre</b>   | RF4. Registrar una reserva  |
| <b>Resumen</b>  | Se quiere poder registrar en la base de datos una reserva de un alojamiento particular para un cliente específico.                |
| <b>Entradas</b>   |   |
| Cedula del cliente: String  |   |
| Tipo de hospedaje deseado: String   |   |
| Tiempo de estadía deseada: String   |   |
| Servicios que desea: String   |   |
| <b>Resultados</b>   |   |
| Se muestran las opciones de estadía según los parámetros de entrada con el costo.   |   |
| El usuario debe seleccionar una y presionar “Aceptar”. Un mensaje de registro exitoso se despliega con el id de la reserva.                   |   |
| <b>RNF asociados</b>  |   |
| Transaccionalidad: La reserva debe completarse o anularse en el intento si algún dato no es correcto o si el usuario lo requiere.             |   |
| Persistencia: Los datos deben quedar completamente guardados en la base de datos.   |   |
| Distribución: La información esta centralizada en la base de datos (siendo que las reservas son un aspecto fundamental).                      |   |
| <b>Nombre</b>   | RF5. Cancelar una reserva   |
| <b>Resumen</b>  | Se requiere poder cancelar una reserva según su id para un usuario, e informar según el tiempo mínimo o usado el cobro realizado. |
| <b>Entradas</b>   |   |
| Cedula del cliente: String  |   |
| Id de la reserva: String  |   |
| <b>Resultados</b>   |   |
| Mensaje de reserva cancelada exitosamente.  |   |
| Mensaje informativo del cobro realizado (10% si cancelo antes del tiempo mínimo, 30% si cancelo entre el tiempo límite y el día anterior a la |   |
| <b>RNF asociados</b>  |   |
| Privacidad: Solo el cliente asociado a la id de reserva puede cancelar el alojamiento.  |   |



|  |   |
|--|---|
| <b>Nombre</b>  | RF6. Retirar una oferta de alojamiento  |
| <b>Resumen</b>   | Como operador se quiere poder retirar una oferta de alojamiento del sistema.                        |
| <b>Entradas</b>  |   |
| NIT o Cedula del operador: String  |   |
| Id del alojamiento: String   |   |
| <b>Resultados</b>  |   |
| Se elimina de la base de datos la oferta de alojamiento, se muestra un mensaje de retiro exitoso.  |   |
| <b>RNF asociados</b>   |   |
| Privacidad: Solo el operador del alojamiento puede retirar el mismo.   |   |
| Distribucion: Los datos del operador y el id de alojamiento deben ser correctos y estar centralizados.                                     |   |
| Persistencia: El retiro debe reflejarse y guardarse correctamente en la base de datos.   |   |
| <b>Nombre</b>  | RFC1. Mostrar el dinero recibido por cada proveedor de alojamiento el año actual y el año corrido   |
| <b>Resumen</b>   | Se quiere mostrar al dinero recibido por cada operador en 2023 y durante el último año (2022-2023). |
| <b>Entradas</b>  |   |
| Ninguna.   |   |
| <b>Resultados</b>  |   |
| Lista con la CC o NIT y nombre de cada proveedor (operador) de alojamiento y el dinero recibido en 2023 y en el último año (por cada uno). |   |
| <b>RNF asociados</b>   |   |
| Integridad de datos: debe mostrar la información correcta.   |   |
| Concurrencia: La información puede ser consultada constantemente.  |   |

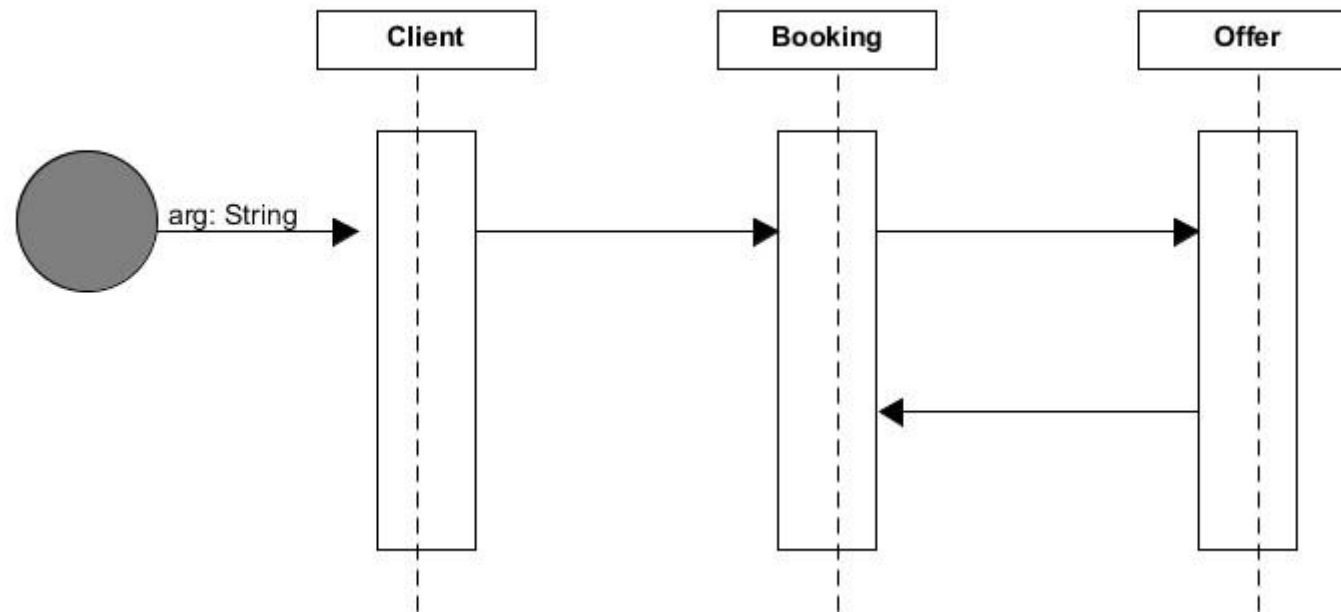
|  |   |
|--|---|
| <b>Nombre</b>  | RFC2. Mostrar las 20 ofertas más populares  |
| <b>Resumen</b>   | Se requiere poder consultar las 20 ofertas de alojamiento más reservadas.   |
| <b>Entradas</b>  |   |
| Ninguna.   |   |
| <b>Resultados</b>  |   |
| Una lista con las 20 ofertas mas reservadas. Por oferta de alojamiento debe mostrar el id del alojamiento y toda la información de la propuesta, |   |
| <b>RNF asociados</b>   |   |
| Concurrencia: Se accede a consultar la información constantemente.   |   |
| <b>Nombre</b>  | RFC3. Mostrar el índice de ocupación de cada una de las ofertas de alojamiento registradas                                  |
| <b>Resumen</b>   | Se quiere poder consultar el índice de ocupación de cada oferta de alojamiento en el sistema (Ocupación / Capacidad total). |
| <b>Entradas</b>  |   |
| Ninguna.   |   |
| <b>Resultados</b>  |   |
| Una lista con cada oferta de alojamiento (toda su información) y el índice de ocupación actual.  |   |
| <b>RNF asociados</b>   |   |
| Concurrencia: La consulta es muy requerida para los usuarios y para poder reservar.  |   |

|  |  |
|--|--|
| <b>Nombre</b>  | RFC4. Mostrar los alojamientos disponibles en un rango de fechas, que cumplen con un conjunto de requerimientos de dotación o servicios.           |
| <b>Resumen</b>   | Se quiere poder consultar los alojamientos que se encuentran disponibles en un rango de fechas dado que además cuenten con un servicio específico. |
| <b>Entradas</b>  |  |
| Rango de fechas: Date.   |  |
| Servicios deseados: String.  |  |
| <b>Resultados</b>  |  |
| Una lista con los alojamientos disponibles en el rango de fecha dado y los servicios requeridos, con toda la información de la oferta.   |  |
| <b>RNF asociados</b>   |  |
| Concurrencia: Los datos se requieren acceder de manera constante   |  |
| Distribución: La base de datos contiene la información centralizada.   |  |
| <b>Nombre</b>  | RFC5. Mostrar el uso de AlohaAndes para cada tipo de usuario de la comunidad   |
| <b>Resumen</b>   | Se quiere consultar el uso total de AlohaAndes de cada usuario y la información del usuario.   |
| <b>Entradas</b>  |  |
| Ninguna.   |  |
| <b>Resultados</b>  |  |
| Una lista con el uso de AlohaAndes y toda la información de cada usuario, con el numero de reservas realizadas, alojamientos reservados, |  |
| <b>RNF asociados</b>   |  |
| Concurrencia: Se debe realizar constantemente la consulta.   |  |

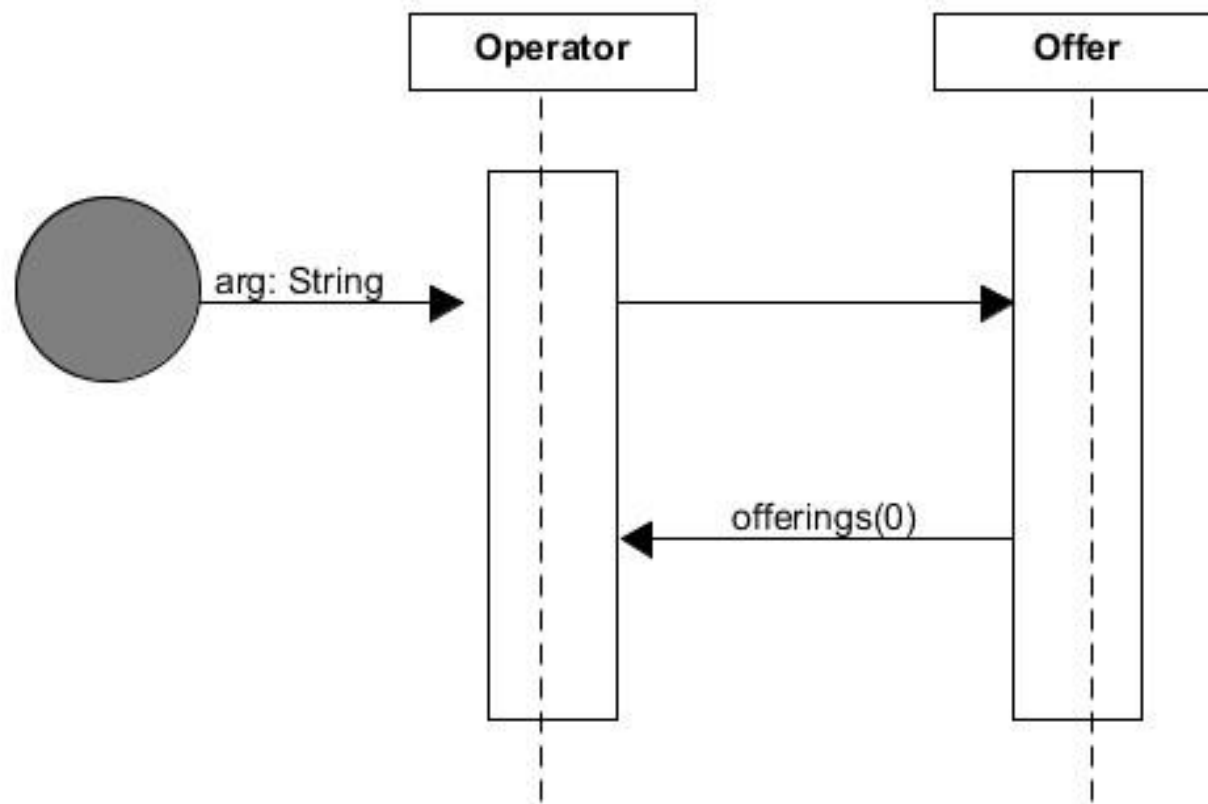
|   |  |
|---|--|
| <b>Nombre</b>   | RFC6. Mostrar el uso de AlohAndes para un usuario dado.        |
| <b>Resumen</b>  | Se quiere consultar el uso total de AlohAndes dado un usuario. |
| <b>Entradas</b>   |  |
| Cedula del usuario: String  |  |
| <b>Resultados</b>   |  |
| Una lista con la información del usuario, el numero de reservas, alojamientos, cobros y tiempo total usado de alojamientos. |  |
| <b>RNF asociados</b>  |  |
| Concurrencia: Se requiere consultar la información constantemente.  |  |

#### 4 Diagramas de secuencia y Control de calidad del modelo

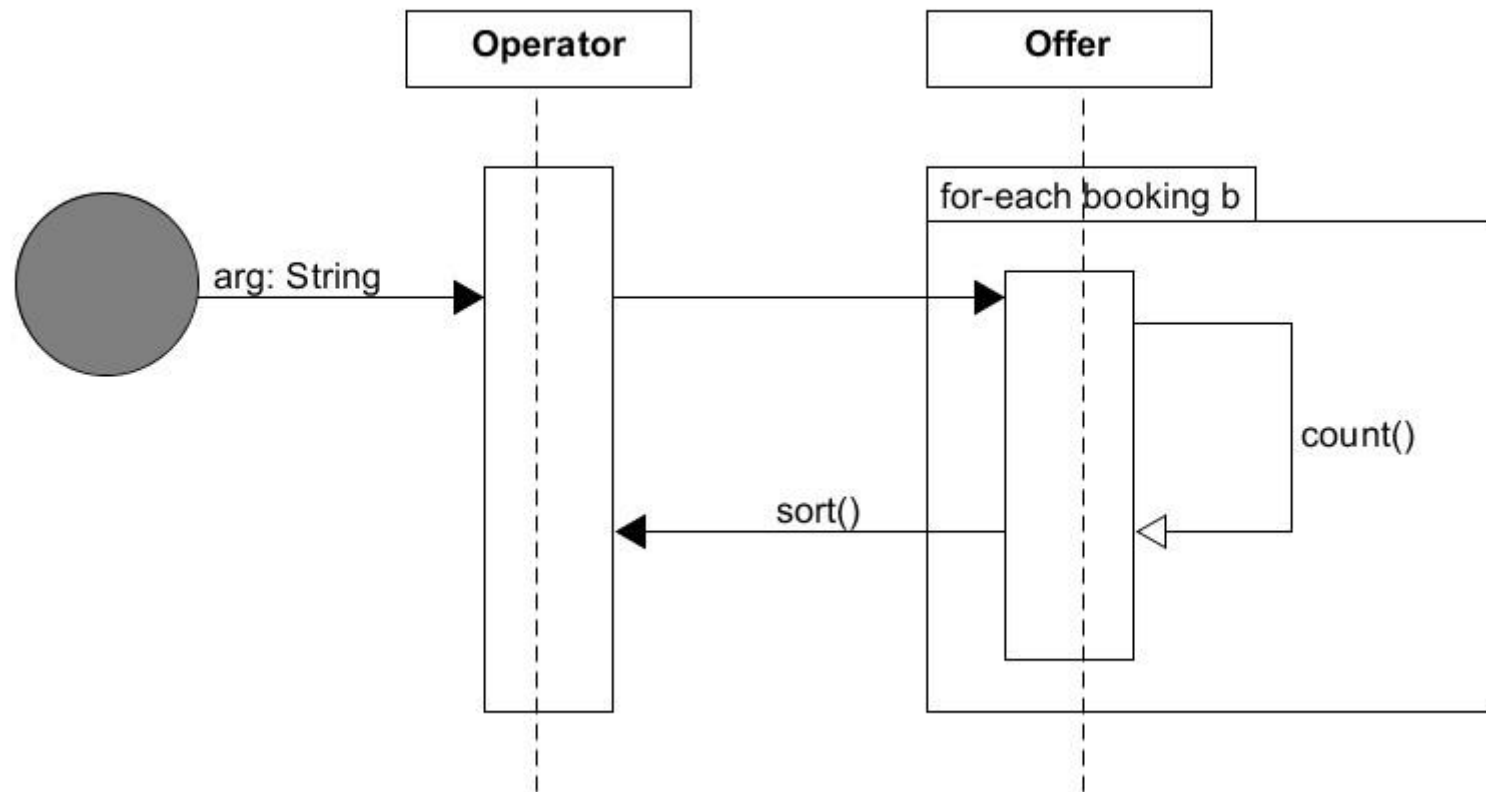
*Requerimiento funcional 4: Registrar una reserva*



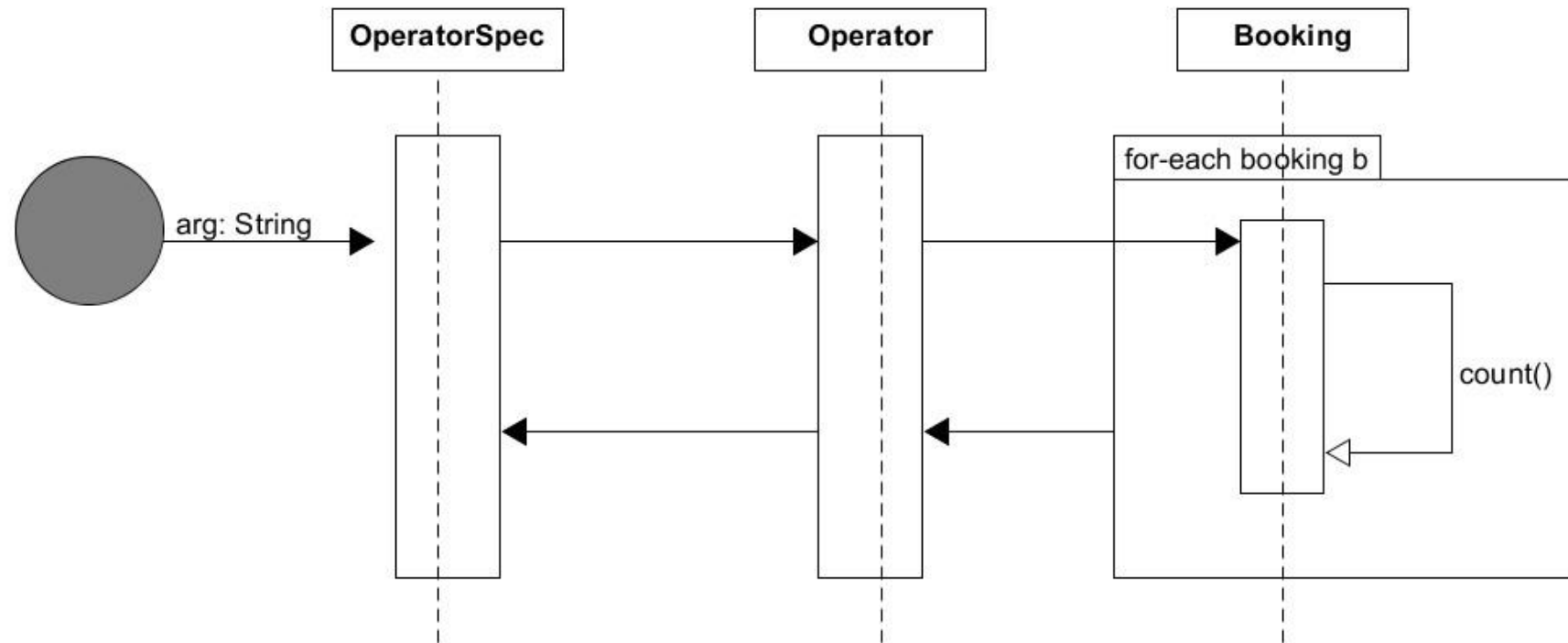
*Requerimiento funcional 6: Retirar oferta de alojamiento*



*Requerimiento funcional de consulta 2: Mostrar las 20 ofertas más populares*



*Requerimiento funcional de consulta 3: Mostrar el índice de ocupación de cada una de las ofertas de alojamiento*



Con base en los diagramas presentados y un análisis basado en el uso practico que se le dará al modelo de datos presentado, se llevo a corregir y verificar el mismo, de manera que se encuentra en un estado donde se asegura el manejo de los requerimientos no funcionales. Por un lado, la privacidad se maneja de manera correcta con el manejo de los conceptos **Client** y **Operator**, donde estos permiten una separación de las

funciones de ambos y del mismo modo el reconocimiento por medio de atributos de identificación (nuiip). De la misma forma, el modelo permite la persistencia de la información en la base de datos, sin contener redundancia, y así acceder a la información de manera persistente sin retrasar el tiempo de consulta. Por último, la distribución de los datos es buena, ya que se manejan diferentes conceptos identificados en el modelo conceptual, los cuales permiten un manejo optimo por separado y en unión, con el fin de mantener la durabilidad de los datos y el aislamiento.