

# “Proyecto del curso – iteración 1”

Juan A. Jaramillo Penagos, Santiago Sinisterra Arias

Este documento contiene el trabajo realizado para la primera iteración del proyecto del curso de Sistemas Transaccionales

Universidad de los Andes, Bogotá, Colombia

{ja.jaramillo, s.sinisterra}@uniandes.edu.co

Fecha de presentación: Marzo 26 de 2023

## Tabla de contenido

1	Introducción .....	1
1.1	General .....	1
1.2	AlohAndes.....	2
2	Requerimientos funcionales .....	3
3	Modelo conceptual .....	9
4	Diseño de la base de datos.....	9
5	Normalización .....	11
6	Control de calidad del modelo.....	11
6.1	Requerimientos funcionales .....	11
6.2	Requerimientos no funcionales .....	15
6.2.1	Privacidad.....	15
6.2.2	Persistencia.....	15
6.2.3	Concurrencia .....	15
6.2.4	Distribución.....	15
7	Bibliografía.....	15

## 1 Introducción

### 1.1 General

La creciente tendencia de internacionalización y regionalización de Uniandes ha llevado a una demanda importante de alojamiento para la comunidad, sus familias y los visitantes de la institución. Por ello, ha surgido la necesidad de diseñar mecanismos flexibles que permitan satisfacer las necesidades de corto, mediano y largo plazo de quienes llegan de otras ciudades. La inquietud es compartida por varias instituciones universitarias de la ciudad, quienes apoyan el proyecto ALOHA y están dispuestas a hacer parte del mismo.

ALOHA es un facilitador de opciones de alojamiento para la comunidad universitaria, sin ser por sí mismo un propietario de inmuebles. Por ello, ha convocado a hoteles reconocidos y a miembros de la comunidad y del vecindario para unirse al proyecto en calidad de operadores. Los operadores son, entonces, quienes ofrecen el servicio de alojamiento a los clientes que buscan una opción a su medida.

Los operadores que son empresas constituidas para ofrecer servicios de hotelería deben estar registrados como tales en la cámara de comercio y en la superintendencia de turismo. Usualmente son hoteles u hostales reconocidos en la ciudad por su calidad y servicio. Los hoteles pueden ofrecer el servicio de alojamiento por noche y pueden tener diversas opciones

en su oferta: habitaciones estándar, semi-suites y suites. Los precios de cada una de ellas pueden variar dependiendo del tipo y la categoría de la instalación, la capacidad del servicio ofrecido, la ubicación, el tamaño o los servicios como bañera, jacuzzi, sala, cocineta, etc. Además, pueden ofrecer servicio de restaurante, piscina, parqueadero, Wifi, TV cable, recepción 24 horas. Por otra parte, los hostales ofrecen el servicio de alojamiento por noche en habitaciones compartidas y usualmente no tienen recepción 24 horas, por lo cual deben indicar claramente en su oferta los horarios de apertura y cierre diarios.

Los operadores de ALOHA pueden ser también personas naturales que ofrecen a la comunidad habitaciones dentro de sus viviendas. El contrato mínimo debe ser de un mes y pueden establecer precios especiales por contratos de varios meses prepagados. Esto facilita a los estudiantes y empleados el contar con una vivienda estable durante el semestre académico. En este caso, la oferta debe indicar los servicios que se incluyen: comidas, acceso a la cocina, baño privado o compartido, habitación individual o compartida y esquema y costo de los servicios (luz, teléfono, agua, TV cable y servicio de internet).

Dentro de la oferta de ALOHA es también posible que se alquilen apartamentos, que pueden o no estar amoblados. El propietario del inmueble debe ser un miembro de la comunidad universitaria (profesores, empleados, egresados, estudiantes o padres de estudiantes). El contrato mínimo debe ser de un mes, debe indicarse si están incluidos los servicios públicos, TV, internet y la administración en el valor del contrato.

Los vecinos y las personas de la comunidad universitaria pueden ofrecer su vivienda para ser ocupada durante algunos días por visitantes esporádicos. Esto ocurre cuando hay eventos de gran magnitud (por ejemplo, un carnaval), que atraen un volumen importante de visitantes. En ese caso, los habitantes ceden el uso de toda la vivienda a los visitantes por algunas noches. Debe indicarse a los clientes las características básicas de la vivienda (número de habitaciones, ubicación, menaje) y el costo. El costo incluye un seguro de arrendamiento, que cubre el inmueble y su contenido. El sistema debe conocer las características de dicho seguro. Esta modalidad no permite que en un año se utilice la vivienda por más de 30 días, en total. No es necesario tener registro ante cámara de comercio ni tampoco ante la superintendencia, ya que no corresponde a un servicio hotelero.

## **1.2 AloHAndes**

Uniandes ha decidido implementar ALOHANDES, un servicio integrado de alojamiento para la comunidad uniandina, de acuerdo con el proyecto ALOHA. Para ello, ha establecido convenios con 10 hoteles reconocidos del sector, 12 hostales cercanos, un operador de residencias universitarias vecino, ha convocado a la comunidad vecina de Fenicia y ha hecho una convocatoria entre sus empleados, profesores y estudiantes, para ofrecer alojamiento en inmuebles de su propiedad, en cualquiera de las modalidades definidas por ALOHA.

Por razones de seguridad, todos los alojamientos deben estar relacionados con Uniandes a través de alguna de las vinculaciones establecidas. Además, los clientes de ALOHANDES deben, también, tener algún vínculo con la institución: estudiantes, egresados, empleados, profesores, padres de estudiantes, profesores invitados, personas registradas en eventos de Uniandes.

ALOHANDES ha establecido algunas reglas de convivencia y uso para asegurar la adecuada utilización de los recursos: una persona no puede reservar más de un alojamiento en un mismo día; un alojamiento no acepta reservas que superen su capacidad; el alojamiento en vivienda universitaria sólo está habilitado a estudiantes, profesores, empleados y profesores visitantes.

Los operadores pueden retirar su oferta de alojamiento. Para garantizar la certidumbre con respecto a la vivienda contratada, así como la disponibilidad de los inmuebles a los propietarios, una oferta de alojamiento sólo se puede retirar del sistema a partir del siguiente día de la finalización de la última reserva vigente. Informado ALOHANDES sobre esta decisión del propietario, el sistema no permite reservas posteriores a la fecha de retiro programada. Si la vivienda no tiene reservas vigentes, puede ser retirada de la oferta de forma inmediata.

Una vez se confirma una reserva, el cliente tiene un tiempo para cancelación oportuna. Si cancela antes del tiempo mínimo se cobra un gasto administrativo del 10%. Si se cancela entre el tiempo límite y el día anterior a la fecha de llegada, se cobra el 30% de la reserva. A partir de la fecha de inicio de la reserva, si el cliente decide no usar el inmueble o si lo desocupa antes de terminar el tiempo de reserva, se cobra el 50% del valor contratado. Para usos cobrados por días, el tiempo límite está fijado en 3 días. Para usos cobrados por semanas o meses, el tiempo límite está fijado en una semana.

## 2 Requerimientos funcionales

<b>Nombre</b>	RF1.
<b>Resumen</b>	Registrar un operador de alojamiento para AlohAndes Usuario: Admin.
<b>Entradas</b>	
Entidad de operador con toda su información.	
<b>Resultados</b>	
Instancia del operador persistida en la base de datos.	
<b>RNF asociados</b>	
Transaccionalidad, ya que se debe registrar en la base de datos las instancias de los diferentes operadores con toda su información asociada.	
Persistencia, ya que la información registrada debe persistir en la base de datos.	
Integridad, ya que no se debe permitir la creación de dos o más instancias de operadores que sean el mismo en el mundo real. Ni tampoco se debe permitir la creación incompleta de un operador. En otras palabras, los registros de las entidades deben ser precisos, consistentes con el mundo real, y completos.	
Facilidad de mantenimiento, la estructura del sistema debe crearse de tal forma que su mantenimiento sea fácil y eficiente.	
Adaptabilidad de los datos a cambios en las reglas de negocio (requisitos legales, servicios mínimos exigidos, etc.).	

Figura 1. Descripción del requerimiento funcional 1

<b>Nombre</b>	RF2.
<b>Resumen</b>	Registrar una propuesta de alojamiento para AlohAndes. Usuario: Operador.
<b>Entradas</b>	
Entidad de alojamiento con su respectivo requerimiento de información completo.	

<b>Resultados</b>
Instancia de la propuesta de alojamiento persistida en la base de datos.
<b>RNF asociados</b>
Transaccionalidad, ya que se debe registrar en la base de datos las instancias de las diferentes propuestas con toda su información asociada.
Persistencia, ya que la información registrada debe persistir en la base de datos.
Integridad, ya que se debe verificar el cumplimiento de todas las reglas de negocio asociadas al registro de las propuestas antes de persistirlas. Además, se tiene también que los registros de las entidades deben ser precisos, consistentes con el mundo real, y completos.
Eficiencia, ya que el registro de todas las propuestas se debe hacer en el menor tiempo posible.
Adaptabilidad de los datos a cambios en las reglas de negocio (requisitos legales, servicios mínimos exigidos, etc.).

Figura 2. Descripción del requerimiento funcional 2

<b>Nombre</b>	RF3.
<b>Resumen</b>	Registrar una persona habilitada para utilizar los servicios. Usuario: Admin.
<b>Entradas</b>	
	Entidad de la persona habilitada para utilizar los servicios con toda su información.
<b>Resultados</b>	
	Instancia de la persona persistida en la base de datos.
<b>RNF asociados</b>	
	Transaccionalidad, ya que se debe registrar en la base de datos las instancias de las diferentes personas con toda su información asociada.
	Persistencia, ya que la información registrada debe persistir en la base de datos.
	Integridad, ya que se debe verificar el cumplimiento de todas las reglas de negocio asociadas al registro de las personas antes de persistirlas. Además, se tiene también que los registros de las entidades deben ser precisos, consistentes con el mundo real, y completos.
	Eficiencia, ya que el registro de todas las personas se debe hacer en el menor tiempo posible.

Figura 3. Descripción del requerimiento funcional 3

<b>Nombre</b>	RF4.
<b>Resumen</b>	Registrar una reserva. Usuario: Persona habilitada por la universidad para utilizar los servicios (clientes).
<b>Entradas</b>	
	Una entidad de reserva con toda su información requerida completa.

<b>Resultados</b>
Instancia de la reserva persistida en la base de datos.
<b>RNF asociados</b>
Transaccionalidad, ya que se debe registrar en la base de datos la instancia de la reserva con toda su información asociada.
Persistencia, ya que la información registrada debe persistir en la base de datos.
Concurrencia, ya que se debe permitir que varios usuarios registren una reserva simultáneamente.
Integridad, ya que se debe verificar el cumplimiento de todas las reglas de negocio asociadas al registro de una reserva antes de persistirla. Además, se tiene también que los registros de las reservas deben ser precisos, consistentes con lo que la persona digitó, y completos.
Facilidad de uso, ya que debe ser fácil para el usuario crear una reserva.

Figura 4. Descripción del requerimiento funcional 4

<b>Nombre</b>	RF5.
<b>Resumen</b>	<p>Cancelar una reserva.</p> <p>Usuario: Persona habilitada por la universidad para utilizar los servicios (clientes).</p>
<b>Entradas</b>	
	El id de la reserva que se desea cancelar.
<b>Resultados</b>	
	Instancia de la reserva marcada como cancelada de la base de datos.
<b>RNF asociados</b>	
	Transaccionalidad, ya que la instancia de la reserva se debe marcar como cancelada en la base de datos (no se elimina para que se puedan seguir haciendo consultas históricas).
	Persistencia, ya que la reserva se debe quedar cancelada y no debe volver a aparecer como habilitada una vez se reinicie la aplicación.
	Concurrencia, ya que se debe permitir que varios usuarios cancelen su reserva simultáneamente.
	Privacidad, ya que no se debe permitir que un usuario pueda cancelar la reserva de otro usuario.
	Facilidad de uso, ya que debe ser fácil para el usuario cancelar una reserva.

Figura 5. Descripción del requerimiento funcional 5

<b>Nombre</b>	RF6.
<b>Resumen</b>	<p>Retirar una oferta de alojamiento.</p> <p>Usuario: Operador.</p>
<b>Entradas</b>	
	El id de la oferta de alojamiento que se desea retirar.

Resultados
Instancia de la oferta de alojamiento deshabilitada en la base de datos. (no se elimina para que se puedan seguir haciendo consultas históricas).
RNF asociados
Transaccionalidad, ya que se debe deshabilitar en la base de datos la instancia de la oferta de alojamiento con toda su información asociada.
Persistencia, ya que la información deshabilitada debe quedarse deshabilitada.
Concurrencia, ya que se debe permitir que varios operadores deshabiliten su oferta de alojamiento simultáneamente.
Privacidad, ya que no se debe permitir que un operador pueda deshabilitar la oferta de alojamiento de otro operador.
Facilidad de uso, ya que debe ser fácil para el operador deshabilitar su oferta de alojamiento.

Figura 6. Descripción del requerimiento funcional 6

Nombre	RFC1.
Resumen	Mostrar el dinero recibido por cada proveedor de alojamiento durante el año actual y el año corrido. Usuario: Admin.
Entradas	
Lista de id de proveedores.	
La fecha inicial y final, estableciendo el rango temporal a considerar.	
La lista de reservas.	
La lista de ofertas.	
Resultados	
Hash table de proveedor y sus respectivas ganancias en el periodo de tiempo.	
RNF asociados	
Seguridad al no permitir que se vean datos personales en el momento de realizar la operación.	
La distribución es necesaria para poder acceder a las ofertas por medio de las reservas, y conocer el proveedor de cada oferta respectiva.	
La concurrencia debe existir para que se puedan hacer múltiples consultas sin problema alguno.	
Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.	

Figura 7. Descripción del requerimiento funcional de consulta 1

Nombre	RFC2.
Resumen	Mostrar las 20 ofertas más populares. Usuario: Operador, Cliente, Admin.
Entradas	

Lista Reservas.
<b>Resultados</b>
Lista de las 20 ofertas más populares.
<b>RNF asociados</b>
Concurrencia, al poder múltiples usuarios saber las ofertas más populares.
Facilidad de uso, al ser algo que se deba poder observar con una sola solicitud.
Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.

Figura 8. Descripción del requerimiento funcional de consulta 2

<b>Nombre</b>	RFC3.
<b>Resumen</b>	Mostrar el índice de ocupación de cada una de las ofertas de alojamiento registradas. Usuario: Admin.
<b>Entradas</b>	
	Lista de ofertas.
	Lista de reservas.
<b>Resultados</b>	
	Hash map de ofertas y su correspondiente índice de ocupación.
<b>RNF asociados</b>	
	La concurrencia es vital para que se pueda consultar la información para múltiples ofertas simultáneamente sin problema alguno.
	Distribución, ya que la oferta debe poder acceder a las reservas que ocupen la oferta.
	Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.

Figura 9. Descripción del requerimiento funcional de consulta 3

<b>Nombre</b>	RFC4.
<b>Resumen</b>	Mostrar los alojamientos disponibles en un rango de fechas, que cumplen con un conjunto de requerimientos de dotación o servicios. Por ejemplo, cocineta, TV, cable, internet, sala. Usuario: Persona habilitada por la universidad para utilizar los servicios (clientes).
<b>Entradas</b>	
	Lista de ofertas disponibles.
	Lista de reservas.
	Rango inferior fecha.
	Rango superior fecha.
	Conjunto de requerimientos.
<b>Resultados</b>	

Lista de ofertas filtrada.
<b>RNF asociados</b>
Distribución, ya que la oferta debe poder acceder a las reservas que ocupen la oferta.
Concurrencia, los alojamientos deben poderse consultar en múltiples puntos de acceso (simultáneamente), o con múltiples filtros sin problemas algunos.
Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.

Figura 10. Descripción del requerimiento funcional de consulta 4

<b>Nombre</b>	RFC5.
<b>Resumen</b>	Mostrar el uso de AlohAndes para cada tipo de usuario de la comunidad Usuario: Admin.
<b>Entradas</b>	
	Lista de ofertas.
	Lista de Reservas.
<b>Resultados</b>	
	Uso que cada tipo de usuario le ha dado a la aplicación.
<b>RNF asociados</b>	
	Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.
	Facilidad de uso, ya que el administrador debe poder realizar esta consulta directamente, es decir, sin la necesidad de hacerla usuario por usuario.

Figura 11. Descripción del requerimiento funcional de consulta 5

<b>Nombre</b>	RFC6.
<b>Resumen</b>	Mostrar el uso de AlohAndes para un usuario dado (Número de noches o meses contratados, características del alojamiento utilizado, dinero pagado). Usuario: Persona habilitada por la universidad para utilizar los servicios (clientes).
<b>Entradas</b>	
	ID de cliente.
	Lista de reservas.
<b>Resultados</b>	
	Hash map de estadísticas.
<b>RNF asociados</b>	
	Privacidad, pues el usuario dado debe solo poder acceder a su información.
	Eficiencia, ya que la consulta no debería tardar más de unos cuantos segundos.



Figura 12. Descripción del requerimiento funcional de consulta 6

### 3 Modelo conceptual

A continuación se presenta la imagen del modelo conceptual plantado para la primera iteración del proyecto:

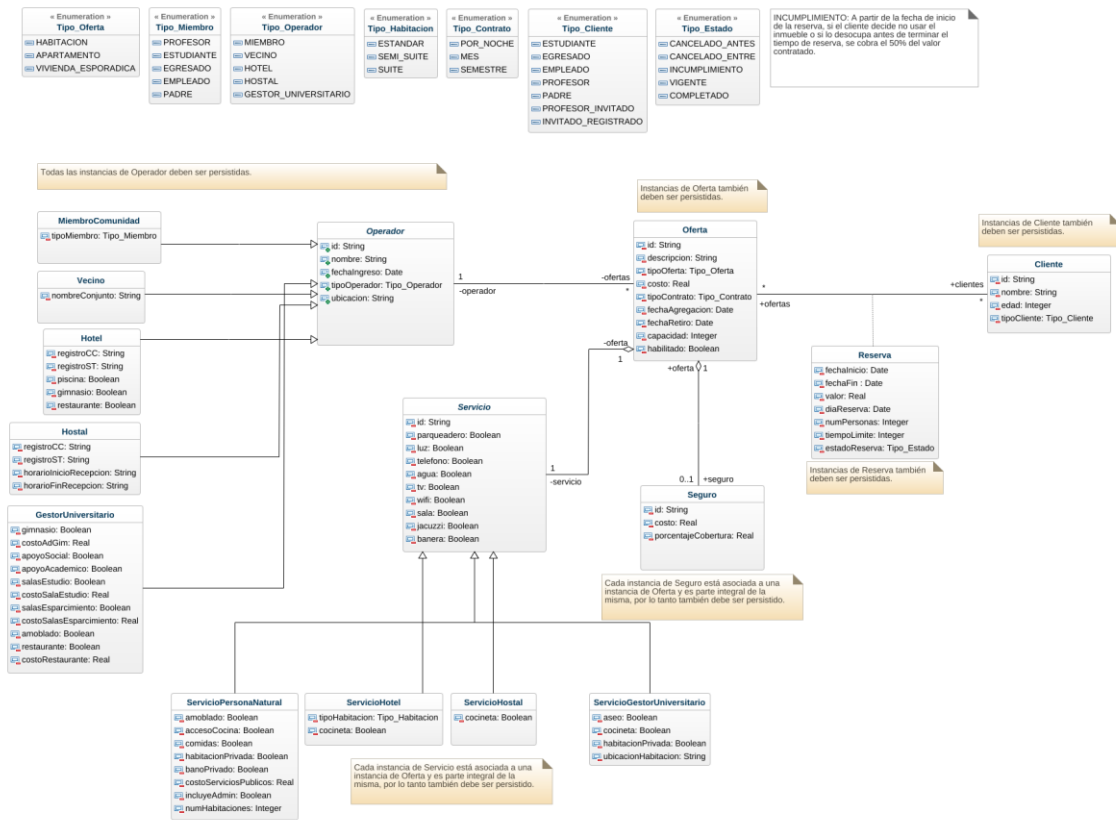


Figura 13. Modelo de clases para la iteración 1 del proyecto.

## 4 Diseño de la base de datos

Cliente			
id	nombre	edad	tipoCliente
PK,SA	NN	NN, CK[>15]	NN

Figura 14. Relación Cliente

id	costo	porcentajeCobertura
PK, SA	NN, CK[>=0]	NN, CK[0..100]

Figura 15. Relación Seguro

[illegible]

Figura 16. Relación Oferta

Reserva								
idOferta	idCliente	diaReserva	fechaInicio	fechaFin	valor	numPersonas	tiempoLimite	estadoReserva
PK, FK(Oferta.id), NN	PK, FK(Cliente.id), NN	PK, NN, SA	NN, UA	NN, UA	NN, CK[>=0]	NN, CK[>0]	NN, CK[>0]	NN

Figura 17. Relación Reserva

Operador				
id	nombre	fechaIngreso	tipoOperador	ubicacion
PK	NN	NN	NN	NN

Figura 18. Relación Operador

MiembroComunidad	
id	tipoMiembro
PK, FK(Operador.id), NN	NN

Figura 19. Relación MiembroComunidad

Vecino	
id	nombreConjunto
PK, FK(Operador.id), NN	NN

Figura 20. Relación Vecino

Hotel					
id	registroCC	registroST	piscina	gimnasio	restaurante
PK, FK(Operador.id), NN	NN, ND	NN, ND	NN	NN	NN

Figura 21. Relación Hotel

Hostal				
id	registroCC	registroST	horarioInicioRecepcion	horarioFinRecepcion
PK, FK(Operador.id), NN	NN, ND	NN, ND	NN	NN

Figura 22. Relación Hostal

GestorUniversitario											
id	gimnasio	costoAdGim	apoyoSocial	apoyoAcademico	salasEstudio	costoSalaEstudio	salasEsparcimiento	costoSalaEsparcimiento	amoblado	restaurante	costoRestaurante
PK, FK(Operador.id), NN	NN	NN, CK[>=0]	NN	NN	NN	NN, CK[>=0]	NN	NN, CK[>=0]	NN	NN	NN, CK[>=0]

Figura 23. Relación GestorUniversitario

Servicio									
id	parqueadero	luz	telefono	agua	tv	wifi	sala	jacuzzi	banera
PK	NN	NN	NN	NN	NN	NN	NN	NN	NN

Figura 24. Relación Servicio

ServicioPersonaNatural								
id	amoblado	accesoCocina	comidas	habitacionPrivada	banoPrivado	costoServiciosPublicos	incluyeAdmin	numHabitaciones
PK, FK(Servicio.id), NN	NN	NN	NN	NN	NN	NN, CK[>=0]	NN	NN, CK[>0]

Figura 25. Relación ServicioPersonaNatural

ServicioHotel		
id	tipoHabitacion	cocineta
PK, FK(Servicio.id), NN	NN	NN

Figura 26. Relación ServicioHotel

ServicioHostal	
id	cocineta
PK, FK(Servicio.id),NN	NN

Figura 27. Relación ServicioHostal

ServicioGestorUniversitario				
id	aseo	cocineta	habitacionPrivada	ubicacionHabitacion
PK, FK(Servicio.id),N N	NN	NN	NN	NN

Figura 28. Relación ServicioGestorUniversitario

## 5 Normalización

1FN: El modelo cumple la primera forma normal al no tener algún atributo multivalor.

2FN: El modelo es 1FN y no tiene dependencias parciales desde los atributos primos.

3FN: El modelo es 2FN y no tiene dependencias transitivas entre atributos no primos.

Forma normal Boyce-Codd: Está en BCNF al no tener varias llaves candidatas que sean compuestas y que se traslapen.

## 6 Control de calidad del modelo

### 6.1 Requerimientos funcionales

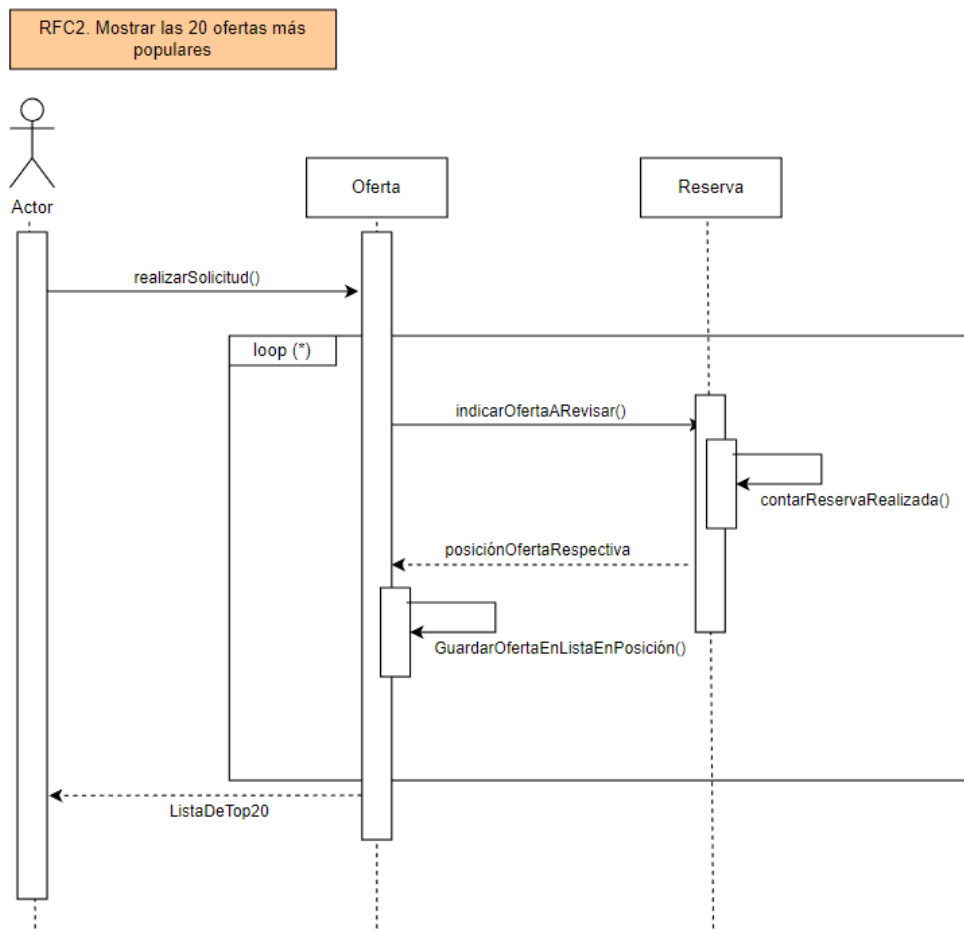


Figura 29. Diagrama de secuencia para el requerimiento funcional de consulta número 2

RFC3. Mostrar el índice de ocupación de cada una de las ofertas de alojamiento registradas.

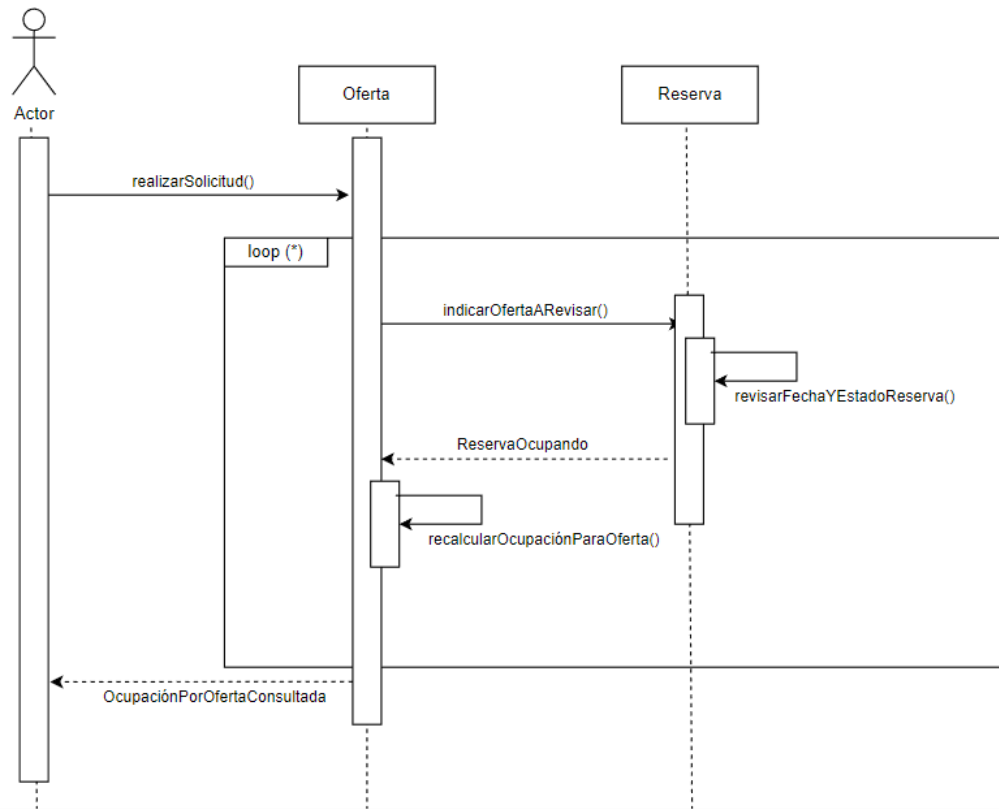


Figura 30. Diagrama de secuencia para el requerimiento funcional de consulta número 3

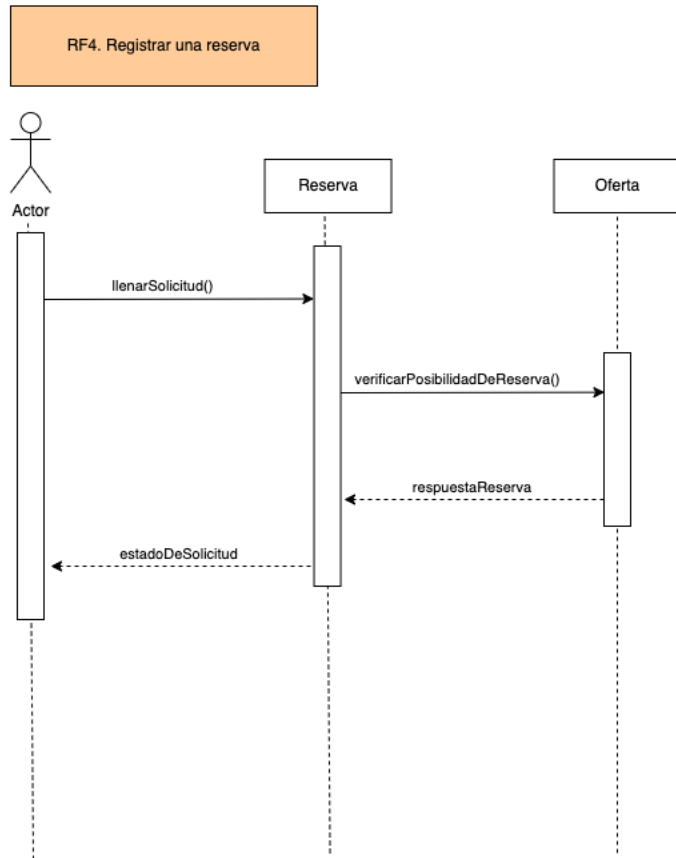
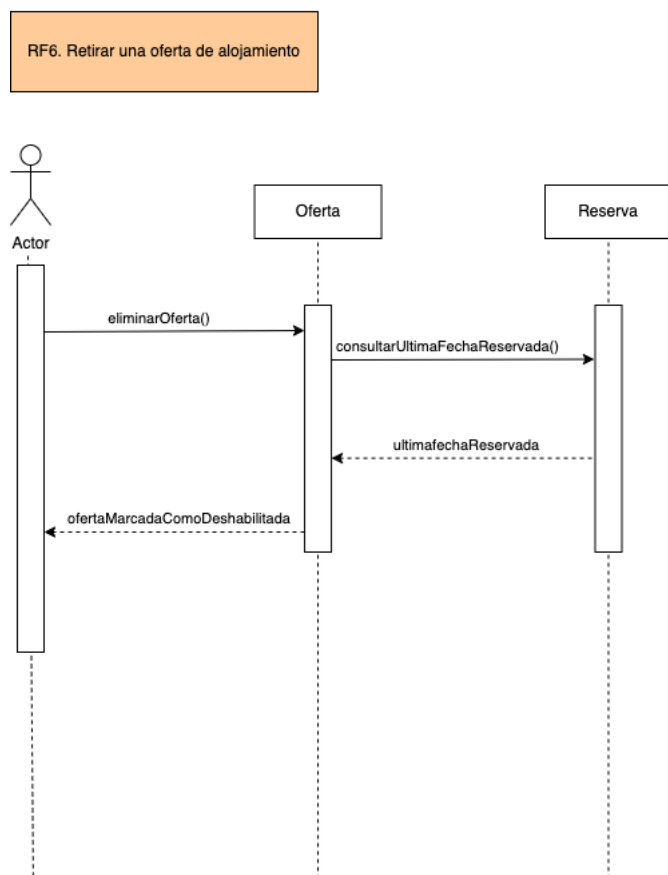


Figura 31. Diagrama de secuencia para el requerimiento funcional número 4



## 6.2 Requerimientos no funcionales

El modelo de datos facilita la implementación de los requerimientos no funcionales.

### 6.2.1 Privacidad

Un cliente tiene sus reservas, por lo que solo podría acceder a estas, y no hay clases que dependan de otras y pueda revelar información delicada por este hecho. Ya que cualquier usuario tiene acceso a las ofertas y como se relacionan a sí mismo.

### 6.2.2 Persistencia

La persistencia esta diseñada para que los datos no disponibles pero importantes se guarden, y que todo lo que se necesite eliminar elimina todo lo que requiere en el proceso, siendo persistente en el proceso.

### 6.2.3 Concurrencia

La concurrencia se encuentra en que las ofertas se actualizan siempre que el proveedor o el cliente realice alguna acción, que se debería mostrar para todos los otros usuarios.

### 6.2.4 Distribución

La base de datos esta centralizada en ofertas, las cuales tienen todos los componentes que necesitan, y de igual manera todos los usuarios pueden acceder a ellas en su manera.

## 7 Bibliografía

1. *www.xml.org*. [En línea] [Citado el: 28 de Abril de 2010.] <http://www.xml.org>.
2. The Institution of Engineering and Technology. *A Guide to Technical Report Writing*. [En línea] [www.theiet.org/students/resources/technicalreport.cfm](http://www.theiet.org/students/resources/technicalreport.cfm).
3. *Universidad de los Andes*. [En línea] [Citado el: 28 de Abril de 2010.] <http://uniandes.edu.co>.
4. IEEE. *Manual de estilo de documentos técnicos*. [En línea] [Citado el: 28 de Abril de 2010.] [http://standards.ieee.org/guides/style/2009\\_Style\\_Manual.pdf](http://standards.ieee.org/guides/style/2009_Style_Manual.pdf).
5. LNCS Springer Verlag. *Lecture Notes in Computer Science*. [En línea] [Citado el: 28 de Abril de 2010.] <http://www.springer.com/computer/lncs?SGWID=0-164-12-73062-0>.
6. Universidad de los Andes. *CARTILLA DE CITAS: Pautas para citar textos y hacer listas de referencias*. [En línea] Universidad de los Andes. [Citado el: 28 de Abril de 2010.] [http://decanaturadeestudiantes.uniandes.edu.co/Documentos/Cartilla\\_de\\_citas.pdf](http://decanaturadeestudiantes.uniandes.edu.co/Documentos/Cartilla_de_citas.pdf).