

## Documento Iteración 1

### Análisis y modelo conceptual

#### a. Requerimientos funcionales:

<b>Nombre</b>	RF1. Registrar los operadores en AlohAndes
<b>Resumen</b>	Debe permitir el registro de todos los operadores en AlohAndes.
<b>Entradas</b>	
Información del usuario como nombre de usuario y contraseña. Empresa: Información necesaria para registrar el operador como, identificación del operador, registro de la Cámara de Comercio y el registro ante la superintendencia de industria y comercio. <ul style="list-style-type: none"><li>• Hotel: (nombre, ubicación)</li><li>• Hostal: (nombre, ubicación, hora de apertura, hora de cierre)</li><li>• Empresa Vivienda Universitaria: (NIT, nombre)</li></ul> Persona Natural: Información necesaria para registrar el operador persona natural como, identificación del operador, canet Uniandes y tipo de vinculación.	
<b>Resultados</b>	
El registro del operador en la aplicación.	
<b>RNF asociados</b>	
Unicidad. No debe haber operadores repetidos.	
Persistencia. Los datos de registro de un operador deben mantenerse a través del tiempo.	
Precisión. La información de los operadores registrada debe ser exacta.	
Privacidad. La información de los operadores solo puede ser manipulada y consultada por los mismos operadores y los usuarios con permisos especiales.	

<b>Nombre</b>	RF2. Registrar ofertas de alojamientos para AlohAndes
<b>Resumen</b>	Debe permitir registrar todas las ofertas de alojamientos.
<b>Entradas</b>	
<ul style="list-style-type: none"><li>• Oferta Hotel: (tipo habitación, precio, id y servicios)</li><li>• Oferta Hostal: (horario hostel, precio, id, compartida y servicios)</li><li>• Oferta Empresa Vivienda Universitaria: (precio, id, tipo habitación y servicios)</li><li>• Oferta Vivienda Propia: (precio, id, comidas, acceso a cocina (Bool), baño privado (Bool), costo servicios y esquema)</li><li>• Oferta Apartamento: (precio, id, servicios incluidos (Bool), televisión (Bool), internet (Bool) y administración (Bool))</li><li>• Oferta Vivienda de la Comunidad: (precio, id, número habitaciones, ubicación, menaje, seguro arrendamiento)</li></ul>	
<b>Resultados</b>	

El registro de la oferta en la aplicación.
<b>RNF asociados</b>
Unicidad. No debe haber ofertas con el mismo id.
Persistencia. Los datos de registro de una oferta deben mantenerse a través del tiempo que esté disponible la oferta.
Precisión. La información de las ofertas registrada debe ser exacta.
Privacidad: La información de las ofertas solo puede ser manipulada por los mismos operadores.

<b>Nombre</b>	RF3. Registrar las personas habilitadas para usar los servicios
<b>Resumen</b>	Debe permitir registrar todas las personas habilitadas en AloHAndes.
<b>Entradas</b>	
Información necesaria para registrar la persona como, usuario, contraseña, carnet de Uniandes, tipo de vinculación con la Universidad y si es un operador (Bool).	
<b>Resultados</b>	
El registro de la persona en la aplicación.	
<b>RNF asociados</b>	
Unicidad. No debe haber personas con el mismo usuario.	
Persistencia. Los datos de registro de una persona autorizada deben mantenerse a través del tiempo.	
Precisión. La información de las personas autorizadas registrada debe ser exacta.	
Privacidad: La información de las personas autorizadas solo puede ser manipulada y consultada por las mismas personas y los usuarios con permisos especiales.	

<b>Nombre</b>	RF4. Registrar una reserva
<b>Resumen</b>	Debe permitir registrar las reservas en AloHAndes.
<b>Entradas</b>	
Información del usuario de la persona que realiza la reserva (usuario y contraseña); ID de la oferta que desea reservar.	
<b>Resultados</b>	
El registro de la reserva en la aplicación asociada al usuario.	
<b>RNF asociados</b>	
Unicidad. No debe haber reservas con el mismo id.	
Persistencia. Los datos de la reserva deben mantenerse a través del tiempo.	
Precisión. La información de la reserva registrada debe ser exacta.	
Privacidad: La información de la reserva solo puede ser manipulada y consultada por la misma persona, operador y/o los usuarios con permisos especiales.	
Transaccionalidad: Al registrar la reserva, se debe actualizar la oferta de alojamiento del operador y también se debe validar y actualizar el pago de la reserva por parte del usuario y el operador.	

<b>Nombre</b>	RF5. Cancelar una reserva
<b>Resumen</b>	Debe permitir cancelar las reservas en AloHAndes.
<b>Entradas</b>	
El id de la reserva a cancelar, usuario de la persona que realizo la reserva, contraseña del usuario que realizo la reserva.	
<b>Resultados</b>	
La cancelación de la reserva en la aplicación.	
<b>RNF asociados</b>	
Persistencia. La información de la cancelación de la reserva se debe mantener con el tiempo legal requerido en caso de reclamos, quejas y/o peticiones.	
Precisión. La información de la cancelación de la reserva debe ser exacta, en caso de presentar reclamos, quejas y/o peticiones.	
Privacidad: La cancelación de la reserva solo puede ser efectuada por el usuario propietario de la reserva, el operador propietario de la oferta y/o los usuarios con permisos especiales.	
Transaccionalidad: Al cancelar la reserva se debe eliminar tanto para el usuario como para el operador, además se requiere empezar el proceso de devolución y registrar el estatus del mismo para el operador como para el usuario.	

<b>Nombre</b>	RF6. Retirar una oferta de alojamiento
<b>Resumen</b>	Debe permitir retirar las ofertas de alojamiento de AloHAndes.
<b>Entradas</b>	
El id de la oferta a cancelar, usuario del operador que realizo la oferta, contraseña del operador que realizo la oferta.	
<b>Resultados</b>	
La cancelación de la oferta en la aplicación.	
<b>RNF asociados</b>	
Persistencia. La información de la cancelación de la oferta se debe mantener un tiempo, en caso de que el operador desee reactivar la oferta.	
Privacidad: La cancelación de la oferta solo puede ser efectuada por el operador propietario de la oferta y/o los usuarios con permisos especiales.	

<b>Nombre</b>	RFC1. Mostrar el dinero recibido por cada proveedor de alojamiento durante el año actual y el año corrido
<b>Resumen</b>	Debe permitir poder consultar el dinero total por proveedor durante el periodo de tiempo del año actual y el año corrido.
<b>Entradas</b>	
Inicio de sesión del usuario con permisos (usuario y contraseña); Opción de consulta de año actual o año corrido.	

<b>Resultados</b>	
HashMap con llave de usuario de cada proveedor y valor el dinero según la consulta del usuario.	
<b>RNF asociados</b>	
Precisión: La información brindada en la consulta debe ser exacta.	
Privacidad: Solo los usuarios con los permisos requeridos deberían tener acceso a esta información.	
Concurrencia: La información puede ser solicitada en cualquier momento y la cantidad de veces que sea necesaria.	

<b>Nombre</b>	RFC2. Mostrar las 20 ofertas más populares
<b>Resumen</b>	Debe permitir poder consultar las 20 ofertas más populares de la aplicación.
<b>Entradas</b>	
Inicio de sesión del usuario (usuario y contraseña)	
<b>Resultados</b>	
Lista con las 20 ofertas de alojamiento más populares.	
<b>RNF asociados</b>	
Precisión: La información brindada en la consulta debe ser exacta.	
Concurrencia: La información puede ser solicitada en cualquier momento y la cantidad de veces que sea necesaria.	
Privacidad: Solo las personas registradas en la aplicación deberían tener acceso a esta información.	

<b>Nombre</b>	RFC3. Mostrar el índice de ocupación de cada oferta mostrada
<b>Resumen</b>	Debe permitir poder consultar el índice de ocupación de cada oferta de la aplicación.
<b>Entradas</b>	
Inicio de sesión del usuario (usuario y contraseña)	
<b>Resultados</b>	
HashMap con llave de OfertaAlojamiento y valor el índice de ocupación de la oferta.	
<b>RNF asociados</b>	
Precisión: La información brindada en la consulta debe ser exacta.	
Privacidad: Solo los usuarios con los permisos requeridos deberían tener acceso a esta información.	

Concurrencia: La información puede ser solicitada en cualquier momento y la cantidad de veces que sea necesaria.

<b>Nombre</b>	RFC4. 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.
<b>Resumen</b>	Debe permitir consultar los alojamientos disponibles, qué cumplen los parámetros de fecha disponible y servicios de preferencia del cliente.
<b>Entradas</b>	
Inicio de sesión del usuario (usuario y contraseña); Rango de fechas que desea el cliente; Lista con los servicios que desea el cliente.	
<b>Resultados</b>	
Lista con los alojamientos que cumplen los criterios del cliente.	
<b>RNF asociados</b>	
Precisión: La información brindada en la consulta debe ser exacta.	
Concurrencia: La información puede ser solicitada en cualquier momento y la cantidad de veces que sea necesaria.	
Distribución: La consulta debe ser realizada por un mismo sistema de gestión de base de datos, ya que debe ser centralizada.	

<b>Nombre</b>	RFC5. Mostrar el uso de AloAandes para cada tipo de usuario de la comunidad.
<b>Resumen</b>	Debe permitir consultar la información del uso de la aplicación, por parte de un tipo de usuario particular
<b>Entradas</b>	
Inicio de sesión del usuario con permisos (usuario y contraseña); tipo de usuario a consultar información.	
<b>Resultados</b>	
Lista con la información del tipo de usuario.	
<b>RNF asociados</b>	
Precisión: La información brindada en la consulta debe ser exacta.	
Privacidad: Solo los usuarios con los permisos requeridos, deberían tener acceso a esta información.	
Concurrencia: La información puede ser solicitada en cualquier momento y la cantidad de veces que sea necesaria.	



- Es importante que todas las clases del modelo sean persistentes. Las que heredan de *OfertaAlojamiento* deben serlo porque de esta manera se sabe qué opciones tiene un cliente para elegir. Las clases que heredan de la clase *Usuario* también deben ser persistentes, ya que se necesita saber esta información para el manejo de autenticación y acceso a la aplicación. Además, la clase *Reserva* debe ser persistente, pues el sistema debe recordar las reservas hechas por los clientes. Finalmente, las clases relacionadas con *Servicio* necesitan persistencia, pues el sistema debe mostrarle al cliente el catálogo de servicios ofrecido por determinado operador u oferta.

## Diseño de la base de datos

### a. Modelo Relacional:

OperadorHotel					
identificacion	registroCamara	registroSuperintendencia	nombre	ubicacion	
PK, SA	PK, UA	PK, UA	NN	NN, NC, ND	
OperadorHostal					
identificacion	registroCamara	registroSuperintendencia	nombre	ubicacion	horaApertura
PK, SA	PK, UA	PK, UA	NN	NN, NC, ND	NN, CK
	horaCierre				NN, CK
OperadorEmpresaVivienda					
identificacion	registroCamara	registroSuperintendencia	nombre	nit	
PK, SA	PK, UA	PK, UA	NN	PK, UA	
UsuarioOperador					
user	password	registroSuperintendencia	registroCamara	identificacion	
PK, UA	NN	UK	UK	UK	
UsuarioPersona					
user	password	carnetUniaandes	vinculacion	esOperador	
PK, UA	NN	UK	NN, CK	NN	
OfertaViviendaComunidad					
id	costo	numHabitaciones	ubicacion	menaje	seguroArrendamiento
PK, SA	NN	NN, NC	NN	NN	NN, NC
	operador	personaAutorizada	fechaInicio	fechaFin	capacidad
	NC, FK UsuarioOperador.user	NC, FK UsuarioPersona.user	NC, CK	NC, CK	NC, CK
OfertaViviendaPropia					
id	costo	comidas	accesoCocina	banoPrivado	habIndividual
PK, SA	NN	NN	NN	NN	NN
	costoServicios	esquema	operador	personaAutorizada	fechaInicio
	NN	NN	NC, FK UsuarioOperador.user	NC, FK UsuarioPersona.user	NC, CK
					fechaFin
					capacidad
					NN, NC
OfertaApartamento					
id	costo	serviciosIncluidos	tv	internet	administracion
PK, SA	NN	NN	NN	NN	NN, NC
	apartamento	operador	personaAutorizada	fechaInicio	fechaFin
	NN, ND, NC, FK apartamento.(numero,con.junto,direccion)	NC, FK UsuarioOperador.user	NC, FK UsuarioPersona.user	NC, CK	NC, CK
Reserva					
id	fechaInicio	cliente	ofertaApartamento	ofertaPropia	OfertaComunidad
PK, SA	NN, NC, CK	NC, FK UsuarioPersona.user	FK apartamento.(numero,con.junto,direccion)	FK ofertaViviendaPropia.id	FK ofertaViviendaComunidad.id
					duracion
					fechaRegistro
					ocupacion
					NN, NC
					NN, NC
					NN, NC
Contrato					
id	valor	meses	ofertaApartamento	ofertaPropia	OfertaComunidad
PK, SA	NN	NN, NC	NC, FK apartamento.(numero,con.junto,direccion)	NC, FK ofertaViviendaPropia.id	NC, FK ofertaViviendaComunidad.id
	operador	cliente	noches		
	NN, NC, FK UsuarioOperador.user	NN, NC, FK UsuarioPersona.user	NN, NC		
OfertaHabitacion					
id	costo	tipo	tamano	piso	categoria
PK, SA	NN	NN, NC, CK	NN, NC	NN, NC	NN, NC, CK
	menaje	hotel	empresaVivienda	hostal	reserva
	NN	NN, NC, FK operadorHotel.identificacion	NN, NC, FK operadorEmpresaVivienda.identificacion	NN, NC, FK operadorHostal.identificacion	FK reserva.id
					capacidad
					NC, CK
Servicio					
nombre	proveedor	indice	capacidad	costoAdicional	
PK, UA	PK, UA, CK	NN, NC	NN, NC		
Apartamento					
numero	conjunto	direccion	amoblado	dueno	
PK, UA	PK, UA	PK, UA	NN	NN, NC, FK UsuarioPersona.user	
PrestacionesHotel					
id	servicio	proveedor			
PK, SA	NN, FK servicio.(nombre,proveedor)	NN, FK operadorHotel.identificacion			
PrestacionesHostal					
id	servicio	proveedor			
PK, SA	NN, FK servicio.(nombre,proveedor)	NN, FK operadorHostal.identificacion			
PrestacionesEmpresa					
id	servicio	proveedor			
PK, SA	NN, FK servicio.(nombre,proveedor)	NN, NC, FK operadorEmpresaVivienda.identificacion			
PrestacionesHabitacion					
id	servicio	habitacion			
PK, SA	NN, FK servicio.(nombre,proveedor)	NN, FK habitacion.id			

b. Nivel de normalización:

Tabla *OperadorHotel*:

$$T_{OperadorHotel}(ID, RC, RS, N, U)$$

$$F = \{(ID, RC, RS) \rightarrow N, (ID, RC, RS) \rightarrow U\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OperadorHostal*:

$$T_{OperadorHostal}(ID, RC, RS, N, U, HA, HC)$$

$$F = \{(ID, RC, RS) \rightarrow N, (ID, RC, RS) \rightarrow U, (ID, RC, RS) \rightarrow HA, (ID, RC, RS) \rightarrow HC\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OperadorEmpresaVivienda*:

$$T_{OperadorEmpresaVivienda}(ID, RC, RS, N, NI)$$

$$F = \{(ID, RC, RS, NI) \rightarrow N\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *UsuarioOperador*:

$$T_{UsuarioOperador}(U, P, RS, RC, ID)$$

$$F = \{U \rightarrow P, U \rightarrow RS, U \rightarrow RC, U \rightarrow ID\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).



Tabla *OperadorPersona*:

$$T_{\text{OperadorPersona}}(U, P, CU, V, O)$$

$$F = \{U \rightarrow P, U \rightarrow CU, U \rightarrow V, U \rightarrow O\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OfertaViviendaComunidad*:

$$T_{\text{OfertaViviendaComunidad}}(ID, CO, NH, U, M, SA, O, PA, FI, FF, CA)$$

$$F = \{ID \rightarrow CO, ID \rightarrow NH, ID \rightarrow U, ID \rightarrow M, ID \rightarrow SA, ID \rightarrow O, ID \rightarrow PA, ID \rightarrow FI, ID \rightarrow FF, ID \rightarrow CA\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OfertaViviendaPropia*:

$$T_{\text{OfertaViviendaPropia}}(ID, CO, C, AC, BP, HI, CS, E, O, PA, FI, FF, CA)$$

$$F = \{ID \rightarrow CO, ID \rightarrow C, ID \rightarrow AC, ID \rightarrow BP, ID \rightarrow HI, ID \rightarrow CS, ID \rightarrow E, ID \rightarrow O, ID \rightarrow PA, ID \rightarrow FI, ID \rightarrow FF, ID \rightarrow CA\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OfertaApartamento*:

$$T_{\text{OfertaApartamento}}(ID, CO, SI, TV, I, AD, CA, A, O, PA, FI, FF)$$

$$F = \{ID \rightarrow CO, ID \rightarrow SI, ID \rightarrow TV, ID \rightarrow I, ID \rightarrow AD, ID \rightarrow CA, ID \rightarrow A, ID \rightarrow O, ID \rightarrow PA, ID \rightarrow FI, ID \rightarrow FF, ID \rightarrow CA\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *Reserva*:

$$T_{Reserva}(ID, FI, C, OA, OP, OC, D, FR, O)$$

$$F = \{ID \rightarrow FI, ID \rightarrow C, ID \rightarrow OA, ID \rightarrow OP, ID \rightarrow OC, ID \rightarrow D, ID \rightarrow FR, ID \rightarrow O\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *Contrato*:

$$T_{Contrato}(ID, VA, M, OA, OP, OC, O, C, N)$$

$$F = \{ID \rightarrow FI, ID \rightarrow VA, ID \rightarrow M, ID \rightarrow C, ID \rightarrow OA, ID \rightarrow OP, ID \rightarrow OC, ID \rightarrow N, ID \rightarrow O\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *OfertaHabitacion*:

$$T_{OfertaHabitacion}(ID, CO, T, TA, P, CA, FI, FF, M, HO, EV, HS, R, C)$$

$$F = \{ID \rightarrow CO, ID \rightarrow T, ID \rightarrow TA, ID \rightarrow P, ID \rightarrow CA, ID \rightarrow FI, ID \rightarrow FF, ID \rightarrow M, ID \rightarrow HO, ID \rightarrow EV, ID \rightarrow HS, ID \rightarrow R, ID \rightarrow C\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *Servicio*:

$$T_{Servicio}(N, P, I, C, CA)$$

$$F = \{(N, P) \rightarrow I, (N, P) \rightarrow C, (N, P) \rightarrow CA\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *Apartamento*:

$$T_{Apartamento}(N, C, D, A, DU)$$

$$F = \{(N, P, D) \rightarrow A, (N, P, D) \rightarrow DU\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias

transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *PrestacionesHotel*:

$$T_{PrestacionesHotel}(ID, S, P)$$

$$F = \{ID \rightarrow S, ID \rightarrow P\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *PrestacionesHostal*:

$$T_{PrestacionesHostal}(ID, S, P)$$

$$F = \{ID \rightarrow S, ID \rightarrow P\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *PrestacionesEmpresa*:

$$T_{PrestacionesEmpresa}(ID, S, P)$$

$$F = \{ID \rightarrow S, ID \rightarrow P\}$$

- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

Tabla *PrestacionesHabitacion*:

$$T_{PrestacionesHabitacion}(ID, S, H)$$

$$F = \{ID \rightarrow S, ID \rightarrow H\}$$

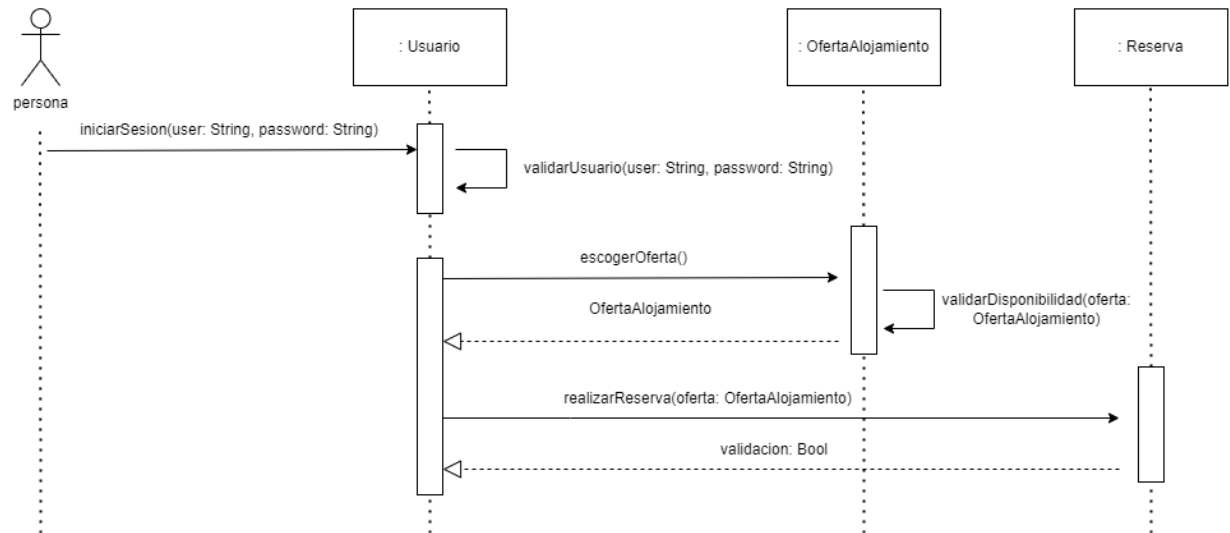
- La tabla se encuentra en FNBC, pues no existen atributos multivalor (1FN), no existen dependencias parciales desde atributos primos (2FN), no existen dependencias transitivas entre atributos no primos (3FN), y finalmente no hay dependencias parciales entre atributos primos (FNBC).

- ❖ Como todas las tablas contempladas en el modelo relacional están en FNBC, se puede entonces afirmar que el modelo relacional está en FNBC.

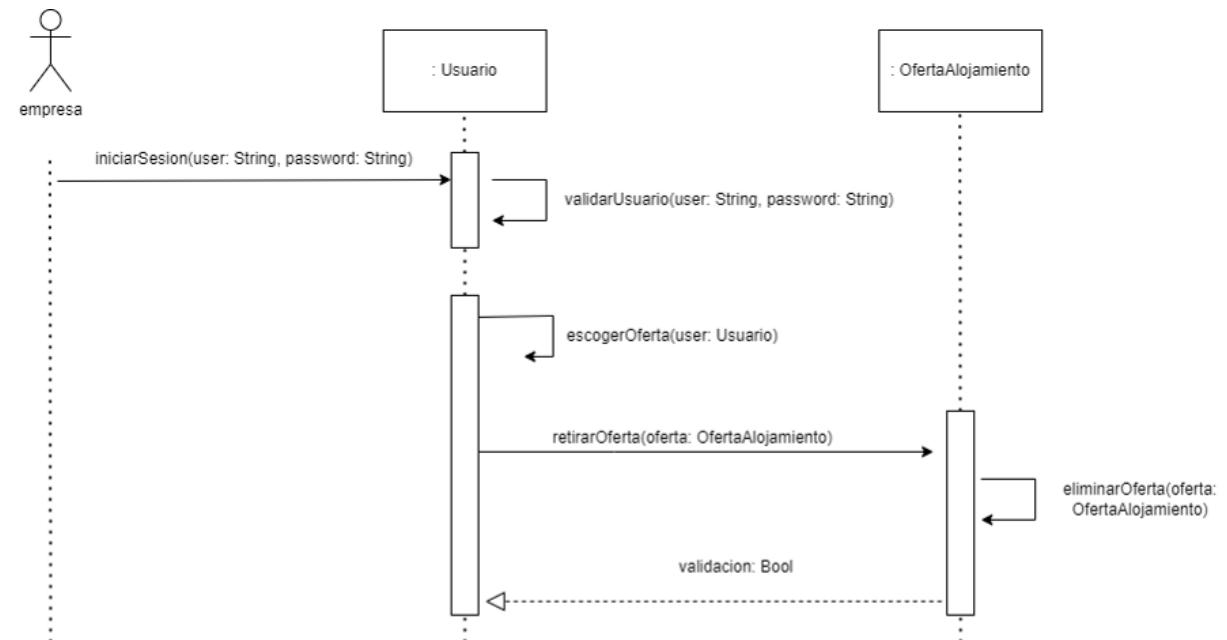
## Control de calidad del modelo

### a. Verificación requerimientos funcionales y de consulta

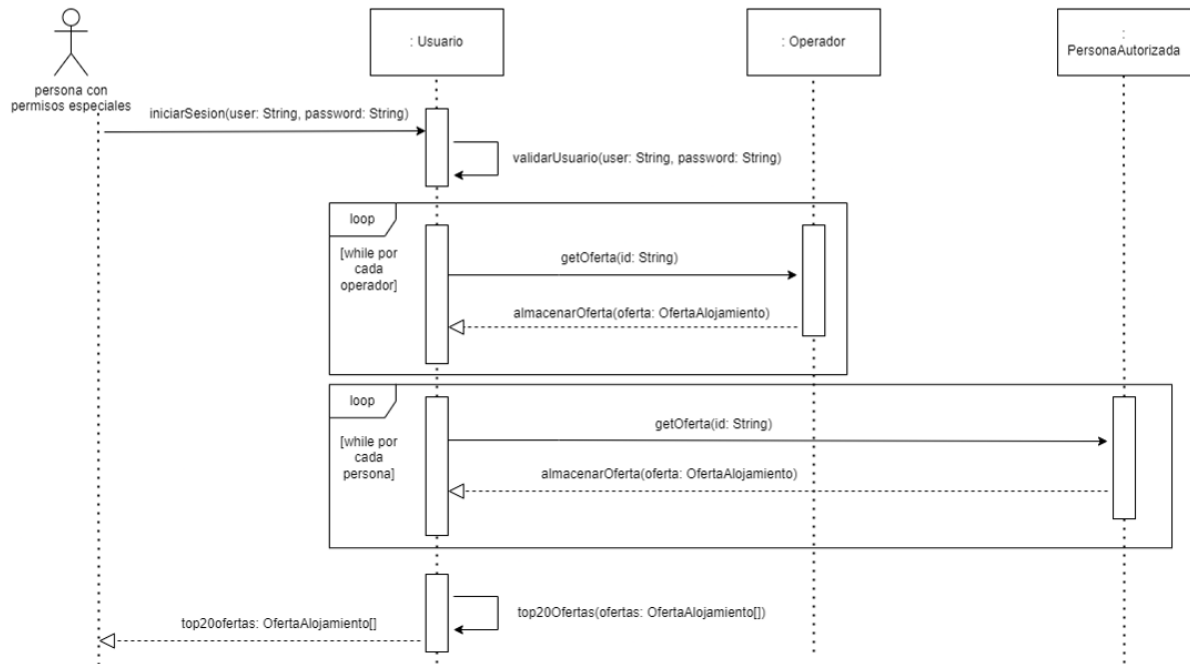
RF4



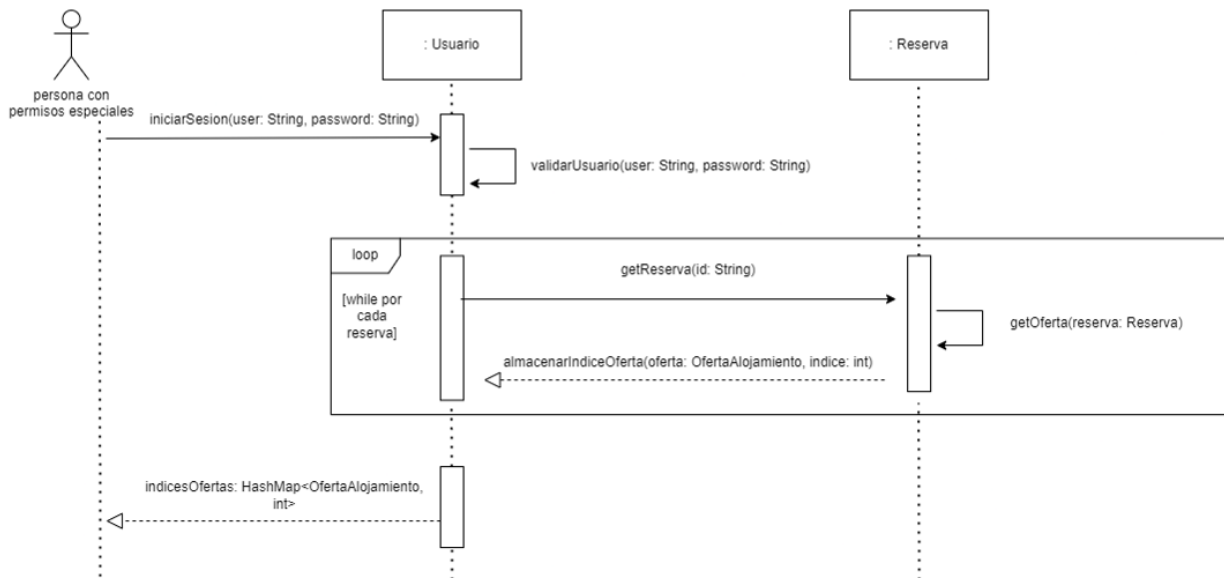
RF6



## RFC2



## RFC3



b. Verificación requerimientos no funcionales

RF / RNF	Privacidad	Persistencia	Concurrencia	Distribución
RF1	Si, ya que, requiere de la verificación del usuario para poder registrar al operador.	Si, ya que, los datos se van a mantener en la base de datos de forma permanente.	Si, ya que, se pueden crear varios operadores al mismo tiempo, sin generar conflictos; desde que sean con diferente usuario.	Si, ya que, para el registro de los operadores se requiere que la base de datos este accesible y funcionando todo el tiempo.
RF2	Si, ya que, requiere de la verificación del usuario para poder manipular las ofertas.	Si, ya que, los datos de las ofertas se van a mantener en la base de datos de forma permanente, hasta que esta sea retirada.	Si, ya que, se pueden registrar varias ofertas al mismo tiempo, sin generar conflictos.	Si, ya que, para el registro de las ofertas se requiere que la base de datos este accesible y funcionando todo el tiempo.
RF3	Si, ya que, requiere de verificación del usuario para poder registrar la información de persona autorizada.	Si, ya que, los datos se van a mantener en la base de datos de forma permanente.	Si, ya que, se pueden crear varias personas autorizadas al mismo tiempo, sin generar conflictos; desde que sean con diferente usuario.	Si, ya que, para el registro de las personas autorizadas se requiere que la base de datos este accesible y funcionando todo el tiempo.
RF4	Si, ya que, requiere de verificación del usuario para poder realizar reservas.	Si, ya que, los datos de una reserva se van a mantener en la base de datos de forma permanente.	Si, ya que, se pueden registrar varias reservas al mismo tiempo, sin generar conflictos.	Si, ya que, para el registro de las reservas se requiere que la base de datos este accesible y funcionando todo el tiempo.
RF5	Si, ya que, requiere de verificación del usuario para poder cancelar reservas.	Si, ya que, la información de la cancelación de la reserva se debe mantener con el tiempo	Si, ya que, se pueden cancelar varias reservas al mismo tiempo, sin generar	Si, ya que, para la cancelación de las reservas se requiere que la base de datos este accesible y

		legal requerido en caso de reclamos, quejas y/o peticiones.	conflictos; desde que sean diferentes reservas.	funcionando todo el tiempo.
RF6	Si, ya que, requiere de verificación del usuario para poder retirar las ofertas.	Si, ya que, la información de la cancelación de la oferta se debe mantener un tiempo, en caso de que el operador desee reactivar la oferta.	Si, ya que, se pueden retirar varias ofertas al mismo tiempo, desde que sean diferentes ofertas.	Si, ya que, para la cancelación de las reservas se requiere que la base de datos este accesible y funcionando todo el tiempo.
RFC1	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.
RFC2	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.
RFC3	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.
RFC4	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.

RFC5	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.
RFC6	Si, ya que, requiere de verificación del usuario para poder consultar la información.	Si, ya que, la información consultada debe ser almacenada hasta que el usuario decida eliminarla.	Si, ya que, se puede consultar la información varias veces al mismo tiempo, sin generar conflictos.	Si, ya que, para la consulta de la información se requiere que la base de datos este accesible y funcionando todo el tiempo.