AlohAndes

Nicolás Francisco Abondano Velasco, Andrés Felipe Orozco González

Documento de entrega para la Iteración 3 de sistemas transaccionales

Universidad de los Andes, Bogotá, Colombia

{nf.abondano, af.orozcog[}@uniandes.edu.co](mailto:%7d@uniandes.edu.co)

Fecha de presentación: Mayo 03 de 2020

Tabla de contenido

[1 Análisis 1](#_Toc39432251)

[1.1 Modelo del mundo 1](#_Toc39432252)

[1.2 Modelo relacional 2](#_Toc39432253)

[2 Diseño de la aplicación 3](#_Toc39432254)

[2.1 Información de las tablas 3](#_Toc39432255)

[2.2 BNCF 4](#_Toc39432256)

[2.3 Lógica de los requerimientos 4](#_Toc39432257)

[3 Resultados 6](#_Toc39432258)

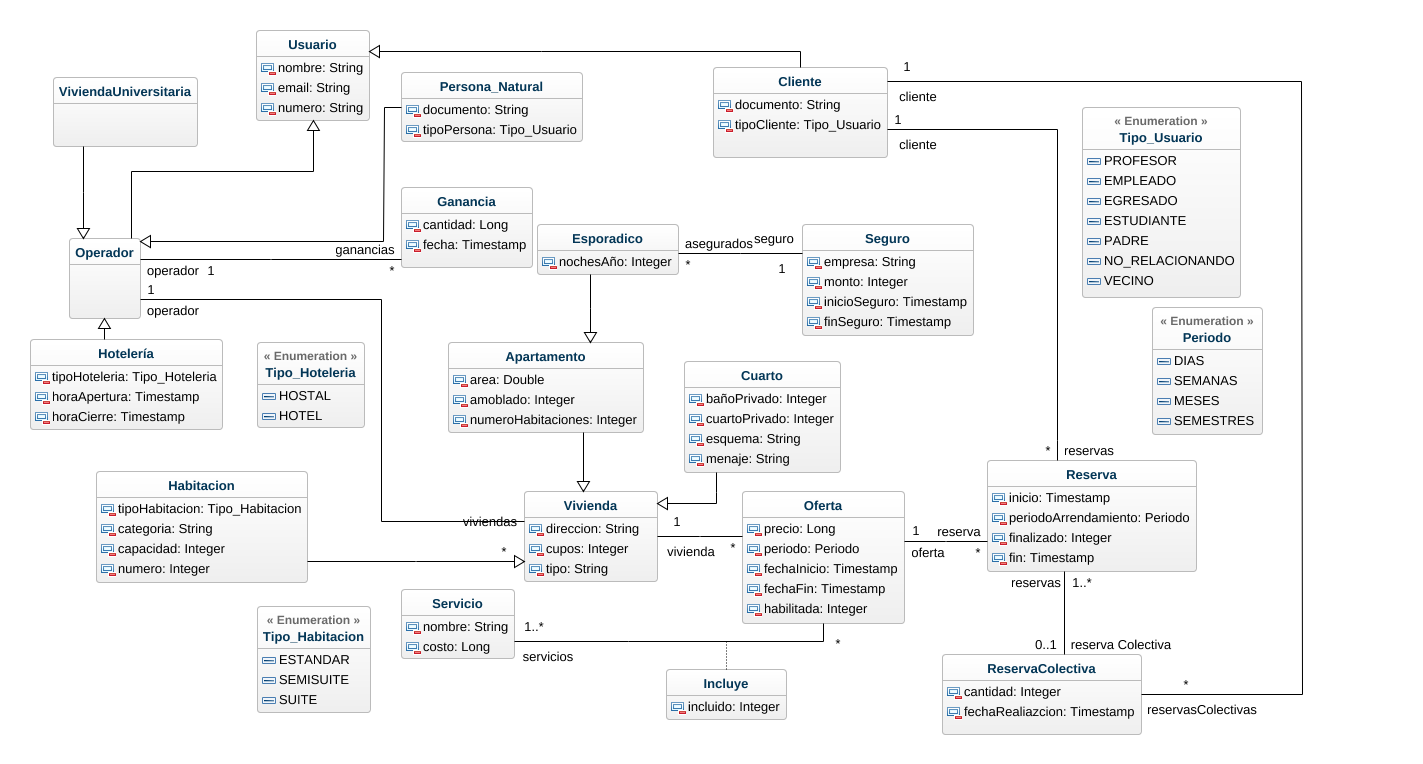
[3.1 Logrados: 6](#_Toc39432259)

[3.2 No logrados: 6](#_Toc39432260)

# Análisis

El modelo del mundo y relacional fue ajustado para incluir la clase “ReservaColectiva”, buscando permitir la implementación de los nuevos requerimientos funcionales RF7 y RF8. También se añadió el atributo “habilitada” a la clase “Oferta” para la implementación de los requerimientos RF9 y RF10. Adicionalmente todos los booleanos fueron cambiados a enteros para facilitar el guardado de los datos.

## Modelo del mundo



Se realizaron algunos cambios al modelo del mundo con respecto a la iteración pasada. Específicamente, cambiamos el formato en el que se registraban las ganancias que eran números que significaban el mes y año, para ahora ser un Timestamp. Además, se agrego a la clase oferta un entero que significa si esta o no esta habilitada. Más allá de lo anterior, se agrego una nueva clase, reserva colectiva, para modelar los nuevos requerimientos de esta iteración y se puso tipo a vivienda, para poder hacer consultas más rápido y modelar la herencia de mejor manera.

## Modelo relacional

Esquema de la BD





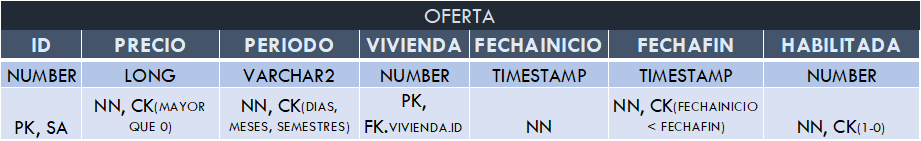




























# Diseño de la aplicación

## Información de las tablas

SELECT TABLE\_NAME NombreTabla, COLUMN\_NAME NombreCol, DATA\_TYPE TipoDeDato, CONSTRAINT\_TYPE TipoRestriccion

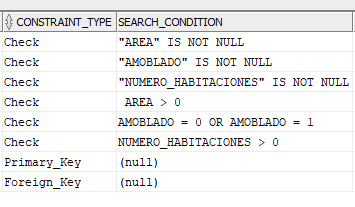
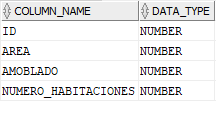
FROM ALL\_TAB\_COLUMNS NATURAL INNER JOIN ALL\_CONS\_COLUMNS

WHERE OWNER = 'ISIS2304A172010'

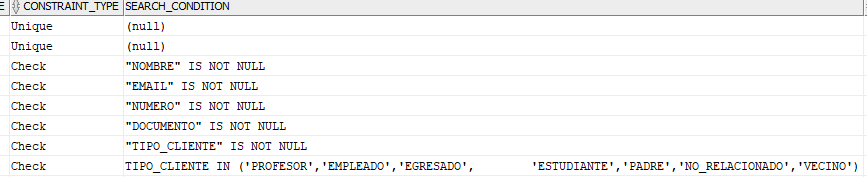
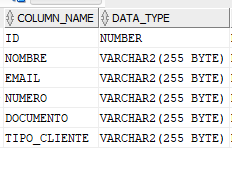
ORDER BY NombreTabla ASC,

NombreCol ASC;

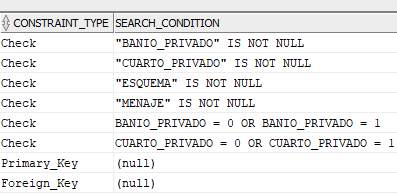
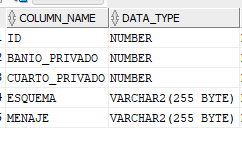
APARTAMENTO



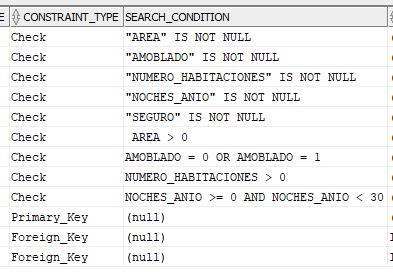
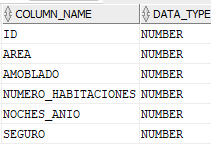
CLIENTE



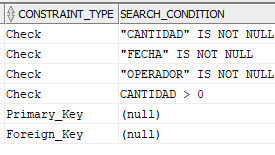
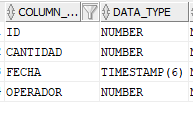
CUARTO



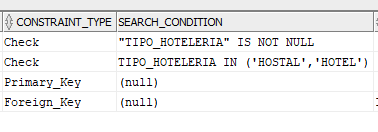
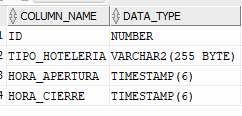
ESPORADICO



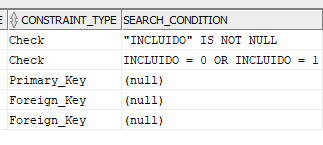
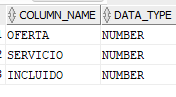
GANANCIAS



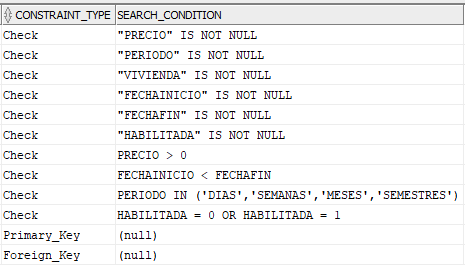
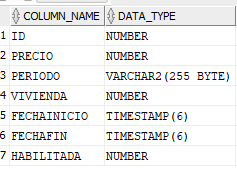
HOTELERIA



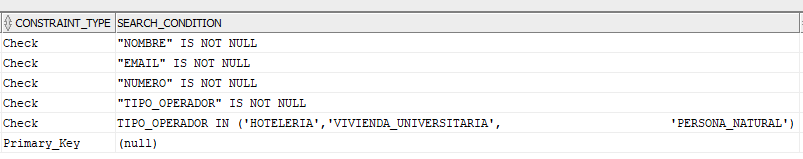
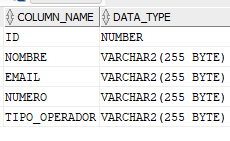
INCLUYE



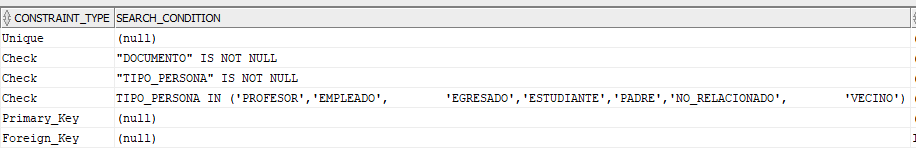
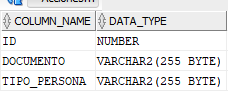
OFERTA



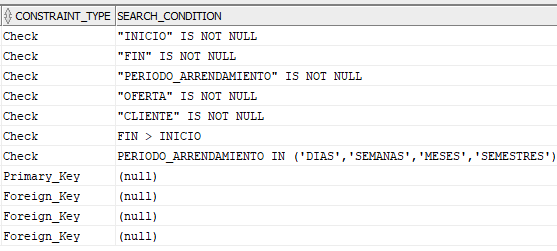
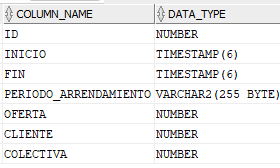
OPERADOR



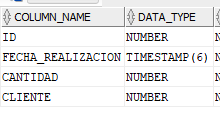
PERSONA NATURAL

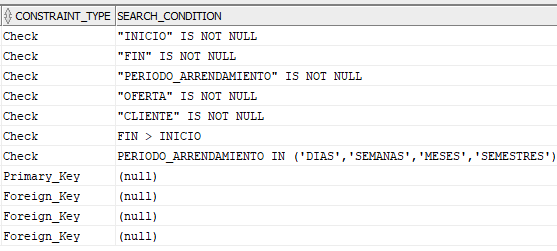


RESERVA

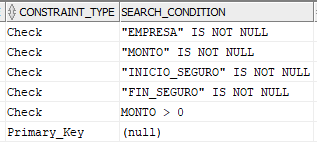
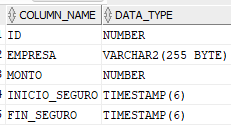


RESERVA\_COLECTIVA

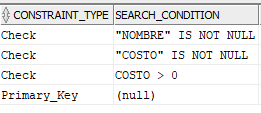
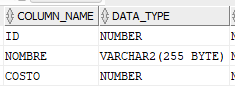




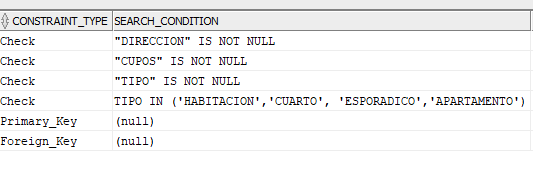
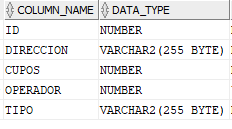
SEGURO



SERVICIO



VIVIENDA



## BNCF

El modelo ya se encontraba en BNCF, las pocas columnas añadidas directamente dependen de la llave primaria y no cambian la normalización. Por otro lado la tabla RESERVA\_COLECTIVA, tiene muy pocos atributos y todos dependen de esta, como una reserva colectiva tiene muchas reservas a RESERVA se le añadió un apuntador a RESERVA\_COLECTIVA. Esto no afecta la normalización.

## Lógica de los requerimientos

En los siguientes requerimientos se garantiza la transaccionalidad dado que aun cuando se hacen sub-transacciones estas realmente no hacen commit hasta que la transacción principal haga commit, y esta solo hace commit si todo resulta como se esperaba.

|  |  |
| --- | --- |
| Nombre | RF7. Registrar reserva colectiva |
| Resumen | Se busca que el sistema sea capaz de registrar numerosas reservas bajo el nombre de un solo cliente |
| Entradas | |
| El número de reservas que se necesitan | |
| Servicios que se están buscando en la reserva. | |
| El rango de fechas en el cual se quiere la reserva. | |
| Resultados | |
| Retorna verdadero si fue posible llevar a cabo la reserva colectiva. De lo contrario, se retorna falso y se anuncian las razones por las que fallo el sistema. | |
| RNF asociados | |
| Transaccionalidad, dado que no se puede registrar una reserva colectiva a medias. | |
| Concurrencia, dado que es posible que se registren varias reservas colectivas al mismo tiempo. | |
| Privacidad, dado que los usuarios solo pueden manipular la información correspondiente a sus privilegios. | |
| Persistencia, dado que los datos de las reservas colectivas deben mantenerse. | |

|  |  |
| --- | --- |
| Nombre | RF8. Cancelar reserva colectiva |
| Resumen | Se busca que el sistema sea capaz de cancelar una reserva colectiva teniendo en cuenta las restricciones que lleva consigo esto, especialmente en términos monetarios |
| Entradas | |
| La identificación de la persona que busca hacer la cancelación de la reserva colectiva. | |
| Resultados | |
| Se lleva a cabo la reserva colectiva y se actualizan las ganancias de los operadores correspondiente según aplique las reglas de negocio de alohAndes. | |
| RNF asociados | |
| Transaccionalidad, dado que no se puede cancelar una reserva colectiva a medias. | |
| Concurrencia, dado que es posible que se cancelen varias reservas colectivas al mismo tiempo. | |
| Privacidad, dado que los usuarios solo pueden manipular la información correspondiente a sus privilegios. | |
| Persistencia, dado que los datos de las cancelaciones se deben mantener. | |

|  |  |
| --- | --- |
| Nombre | RF10. Deshabilitar oferta de alojamiento. |
| Resumen | Se busca que el sistema sea capaz de deshabilitar, pero no borrar una oferta de alojamiento. Por lo que futuros clientes no tengan la posibilidad de verla. |
| Entradas | |
| La oferta que se quiere deshabilitar | |
| Resultados | |
| Se retornan las posibles reservas que pudieran estar asociadas a esa oferta, notificando cuales de estas fueron posible asignarles otra oferta y cuales no. | |
| RNF asociados | |
| Transaccionalidad, dado que no se puede deshabilitar una oferta a medias. | |
| Concurrencia, dado que es posible que se deshabiliten varias ofertas de alojamiento al mismo tiempo. | |
| Privacidad, dado que los usuarios solo pueden manipular la información correspondiente a sus privilegios. | |
| Persistencia, dado que los datos de las ofertas deshabilitadas tienen que mantenerse. | |

|  |  |
| --- | --- |
| Nombre | RF10. Rehabilitar una oferta de alojamiento. |
| Resumen | Se busca que el sistema sea capaz de rehabilitar una oferta de alojamiento que había sido previamente deshabilitada. |
| Entradas | |
| La oferta que se quiere rehabilitar. | |
| Resultados | |
| Ninguno. | |
| RNF asociados | |
| Transaccionalidad, dado que no se puede rehabilitar una oferta de alojamiento a medias. | |
| Concurrencia, dado que es posible que se rehabiliten varias ofertas al mismo tiempo. | |
| Privacidad, dado que los usuarios solo pueden manipular la información correspondiente a sus privilegios. | |
| Persistencia, dado que los datos de las ofertas deben persistir. | |

En los nuevos requerimientos de consulta encontramos los siguientes

|  |  |
| --- | --- |
| Nombre | RFC7. Analizar la operación de AlohAndes |
| Resumen | Se busca que el usuario tenga conocimiento para un periodo de tiempo determinado y un tipo de alojamiento especifico, las fechas con mayores ganancias, menor y mayor demanda. |
| Entradas | |
| El tipo de alojamiento | |
| El periodo de tiempo específico. | |
| Resultados | |
| La cantidad más baja de demanda con su respectiva fecha. La cantidad más alta de demanda con su respectiva fecha. La cantidad más grande de ganancias y su respectiva fecha. | |

|  |  |
| --- | --- |
| Nombre | RFC8. Encontrar los clientes frecuentes |
| Resumen | Para un alojamiento dado, se busca saber cuales son los clientes frecuentes de dicho alojamiento. Según si se han quedado en el lugar más de dos veces, o por un periodo mayor a 15 días en total. |
| Entradas | |
| El id del alojamiento del cual se quiere obtener la estadística | |
| Resultados | |
| Retorna una lista con el ID de los clientes frecuentes de dicho alojamiento. | |

|  |  |
| --- | --- |
| Nombre | RFC9. Encontrar las ofertas de alojamiento que no tienen mucha demanda. |
| Resumen | Se quiere mostrar al usuario aquellas ofertas que desde el inicio de funcionamiento de AlohAndes, nunca han tenido un periodo de menos de 30 días en el que estén ocupadas. |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Retorna una lista con las ofertas que tienen las características anteriormente mencionadas. | |

# Resultados

## Logrados:

Se logró la implementación correcta de los 4 nuevos requerimientos, además de las nuevas 4 consultas. Se logró hacer efectivo cambios en el modelo del mundo y relacional adicionando los atributos requeridos y la nueva clase.

## No logrados:

Se logró todo lo comprometido en clase, al menos hasta lo probado.