“Iteración 2”

Juan Sebastián Cabra, Daniela Rocha Torres

Departamento de Ingeniería de Sistemas y Computación

Universidad de los Andes, Bogotá, Colombia

{js.cabra, [d.rocha}@uniandes.edu.co](mailto:d.rocha%7d@uniandes.edu.co)

Fecha de presentación:Abril 24 de 2018

Tabla de contenido

[1 Introducción 1](#_Toc512322536)

[2 Caso de estudio 2](#_Toc512322537)

[2.1 Requerimientos funcionales 2](#_Toc512322538)

[2.1.1 RF1 2](#_Toc512322539)

[2.1.2 RF2 2](#_Toc512322540)

[2.1.3 RF3 2](#_Toc512322541)

[2.1.4 RF4 2](#_Toc512322542)

[2.1.5 RF5 2](#_Toc512322543)

[2.1.6 RF6 2](#_Toc512322544)

[2.2 Requerimientos funcionales de consulta 2](#_Toc512322545)

[2.2.1 RFC1 2](#_Toc512322546)

[2.2.2 RFC2 2](#_Toc512322547)

[2.2.3 RFC3 2](#_Toc512322548)

[2.2.4 RFC4 2](#_Toc512322549)

[2.2.5 RFC5 2](#_Toc512322550)

[2.2.6 RFC6 2](#_Toc512322551)

[2.2.7 RFC7 2](#_Toc512322552)

[2.2.8 RFC8 2](#_Toc512322553)

[2.3 Requerimientos funcionales de modificación 3](#_Toc512322554)

[3 Modelo conceptual 3](#_Toc512322555)

[3.1 Modelo UML 3](#_Toc512322556)

[3.2 Modelo relacional 4](#_Toc512322557)

[4 Análisis y Diseño de la aplicación 4](#_Toc512322558)

[5 Resultados 4](#_Toc512322559)

[5.1 Logrados 4](#_Toc512322560)

[5.2 No logrados 4](#_Toc512322561)

# Introducción

Este documento presentará el diseño de la aplicación transaccional basada en el caso de estudio “AlohAndes”, así como los modelos conceptuales asociados al mismo.

# Caso de estudio

## Requerimientos funcionales

### RF1

*Registrar los operadores de alojamiento para AlohAndes.*

-Se revisa si existe un operador/proveedor de alojamiento con el id del operador que se desea registrar; en caso de que ya exista, se envía una excepción informando que dicho operador ya se encuentra registrado, si este no existe en la base de datos, se añade a la base de datos con sus datos respectivos (id, nombre, tipo).

### RF2

*Registrar propuestas de alojamientos para AlohAndes*

-Se revisa si ya existe un alojamiento con el id del alojamiento del cual se desea registrar la propuesta; en caso de que ya exista, se revisa el estado de dicho alojamiento y si está Disponible, se envía una excepción informando que ya se encuentra habilitada dicha oferta, si está ‘No disponible’ se cambia su estado para que esté ‘Disponible’; en caso de no existir, se añade a la base de datos con sus datos respectivos dependiendo si es Apartamento, Vivienda o Habitación, declarando Estado como ‘Disponible’.

### RF3

*Registrar las personas habilitadas para utilizar los servicios*

-Se revisan si ya se encuentra un cliente/persona habilitada con dicho id, si ya lo hay se envía una excepción informando que este ya está registrado y habilitado. En caso de no estar, se añade a la base de datos con sus datos respectivos (*idcliente, nombre, apellido, login, passwordusuario).*

### RF4

*Registrar una reserva*

-Se revisan si ya existe un contrato con dicho id, si ya lo hay se envía una excepción informando que este ya existe. En caso de no estar, se añade un contrato a la base de datos con sus datos respectivos y aclarando que el estado es ‘En curso’, revisando primero reglas de negocio como el tiempo mínimo de estadía dependiendo del tipo de alojamiento para el cual se está reservando*.*

### RF5

*Cancelar una reserva*

-Se revisa que la reserva que se desee cancelar exista. Si no existe se envía una excepción informando que dicha reserva no puede cancelarse. Si existe, se revisan reglas de negocio, de manera que se apliquen las multas necesarias dependiendo del momento en el cual se está cancelando y se cambia el estado a ‘Cancelado’.

### RF6

*Retirar una oferta de alojamiento*

-Se revisa si existe el alojamiento del cual se desea retirar su oferta; en caso de que no exista, se envía una excepción informando que no se encuentra, si está ‘No disponible’ se envía una excepción informando que ya se encuentra deshabilitada dicha oferta. Si existe y se encuentra ‘Disponible’ se cambia su estado a ‘No disponible’.

## Requerimientos funcionales de consulta

### RFC2

*Mostrar las 20 ofertas más populares*

### RFC7

*Analizar la operación de AlohAndes*

### RFC8

*Encontrar los clientes frecuentes*

## Requerimientos funcionales de modificación

### RF7

*Registrar reserva colectiva*

-Se revisa si ya existe una reserva colectiva con dicho id. Si ya existe se envía una excepción informando que ya existe. En caso de no existir, se añade la reserva colectiva con su id y el id del cliente, creando un contrato individual por cada solicitud.

### RF8

*Cancelar reserva colectiva*

-Se revisa si existe la reserva colectiva con dicho id. Si no existe se envía una excepción informando que no se puede cancelar. En caso de existir, se cambia el estado de cada uno de los contratos asociados a la reserva colectiva con dicha id por ‘Cancelado’.

### RF9

*Deshabilitar oferta de alojamiento*

### RF10

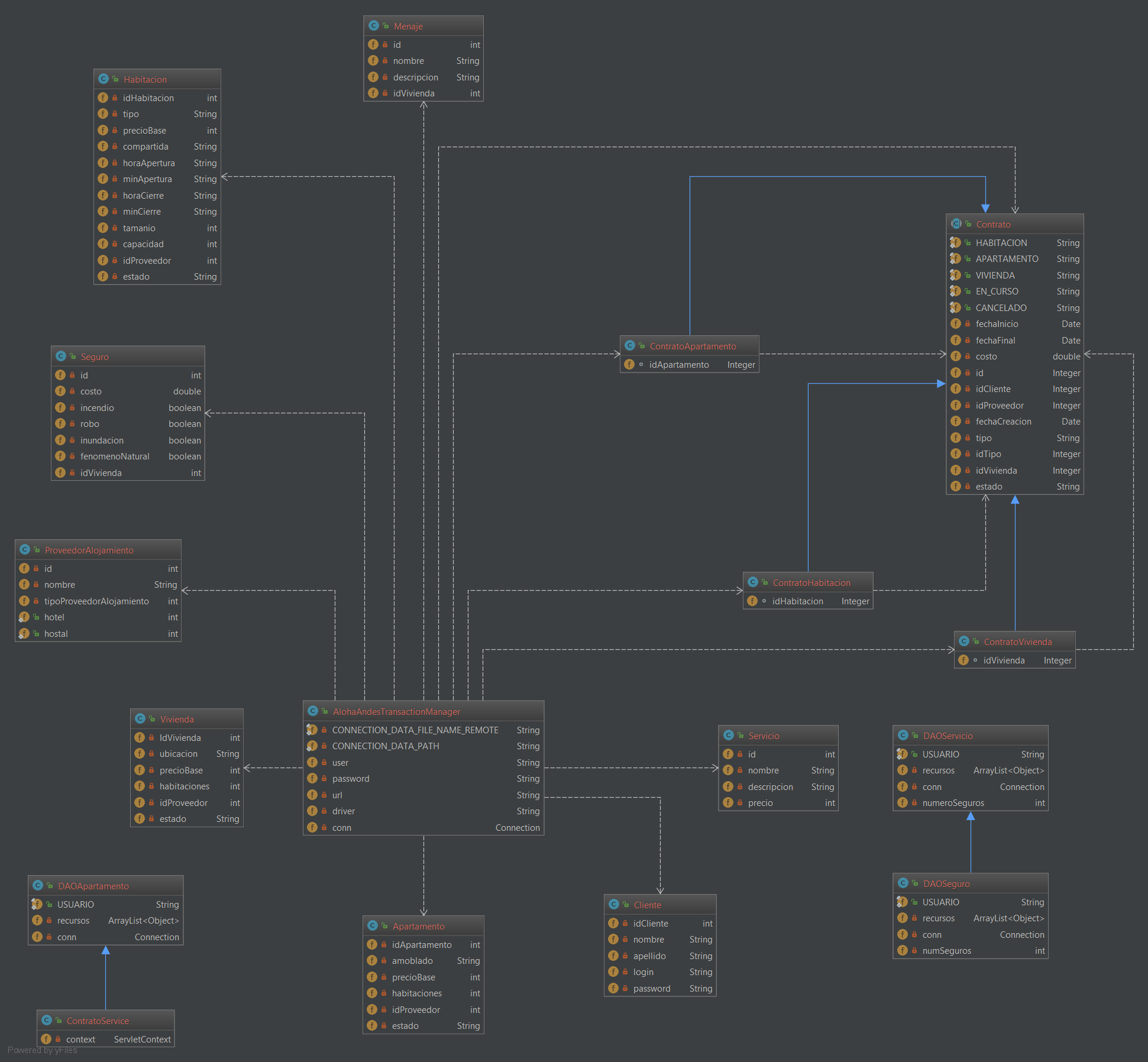
*Rehabilitar oferta de alojamiento*

# Análisis y Modelo conceptual

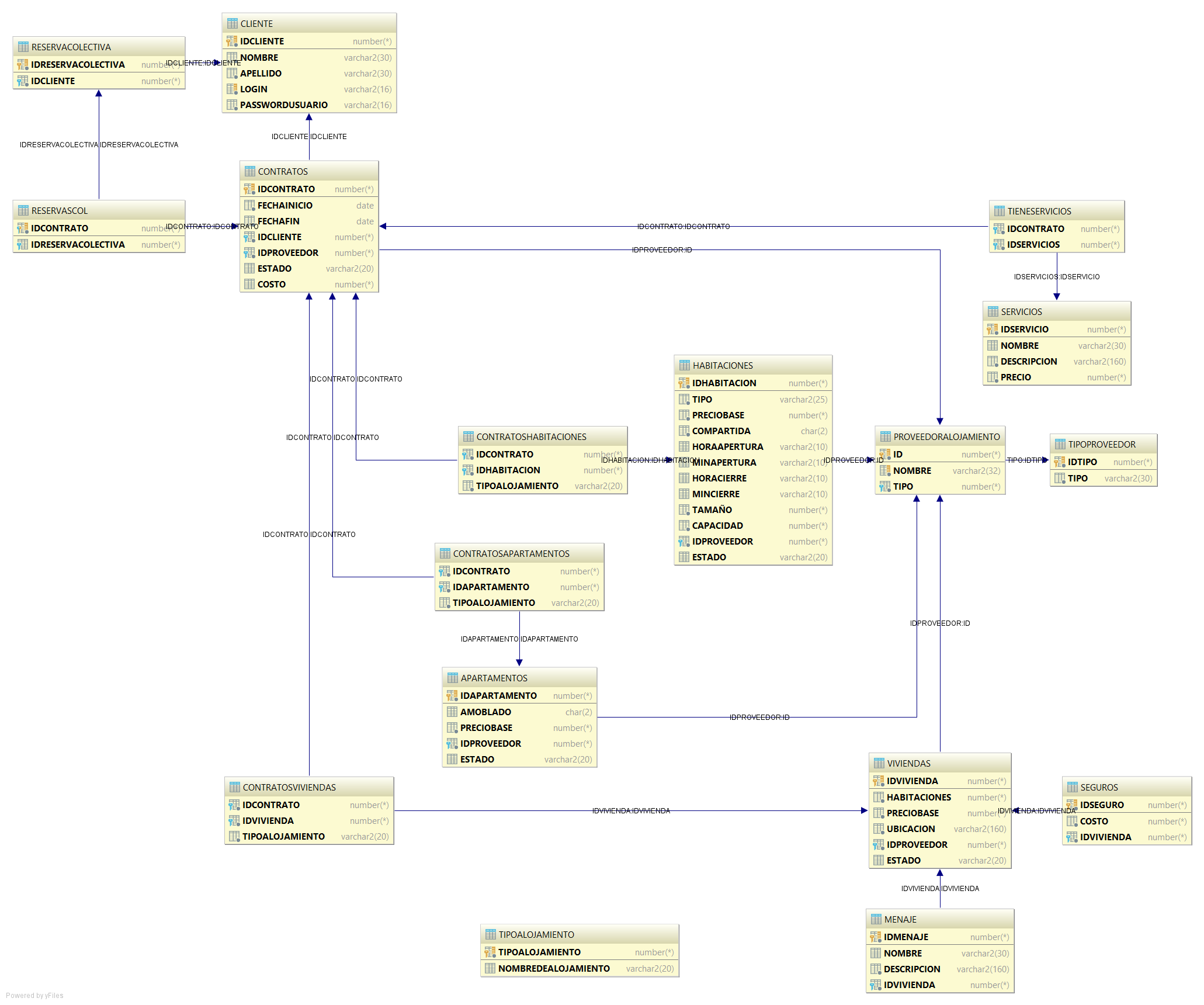
## Análisis

Para satisfacer con los nuevos requerimientos solicitados, fue necesario hacer modificaciones en el modelo conceptual propuesto para el primer marco propuesto para el negocio de AlohAndes, tales como añadir atributos que permitieran modelar reservas colectivas y ofertas de alojamiento. Y en el modelo relacional se creó una tabla que modela: una reserva colectiva, la cual tiene un usuario (el cual la creó) y un id asociado a dicha reserva colectiva, una tabla que relaciona las reservas individuales con cierta reserva colectiva (si aplica) y una tabla de tipoAlojamiento para identificar éstos más fácilmente.

## Modelo UML



## Modelo relacional



# Diseño de la aplicación

A

# Resultados

## Logrados

bjbj

## No logrados