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)

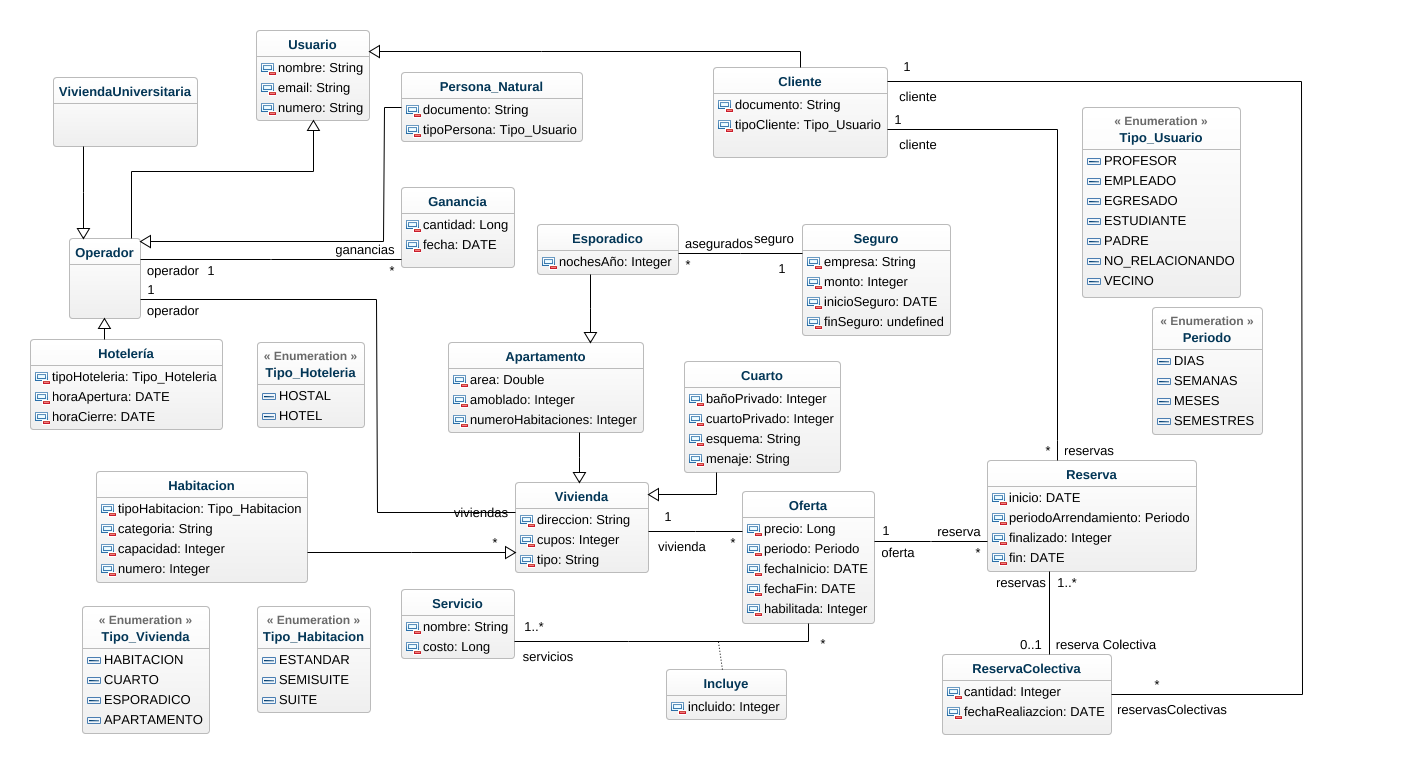
[3.1 Logrados: 6](#_Toc39432259)

[3.2 No logrados: 6](#_Toc39432260)

# Análisis

Se realizaron pequeños cambios pues se cambio de timestamp a date, esto debido a la facilidad de generar grandes cantidades de registros para el tipo date, lo cual el tipo timestamp no puede lograr.

## Modelo del mundo



# Diseño físico

Se le añadió índices a la tabla de reserva en las fechas, tanto en el inicio como fin de esta, para facilitar el filtrado cuando piden un rango de fechas.

## Indices

Cf

## Análisis

### Requerimiento de consulta 10

Sentencia sql:

SELECT DISTINCT c.id, c.nombre, c.email, c.numero, c.documento, c.tipo\_cliente

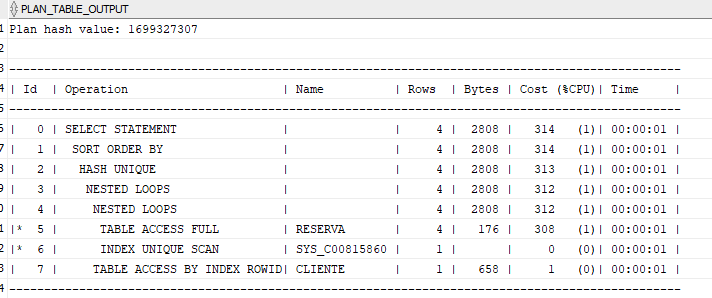
FROM CLIENTE c, RESERVA r, OFERTA o

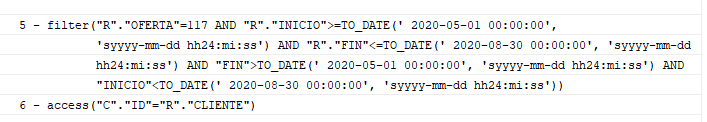
WHERE c.id = r.cliente AND o.id = r.oferta AND o.id = ? AND

r.inicio BETWEEN ? AND ? AND

r.fin BETWEEN ? AND ?

ORDER BY v.tipo, c.tipo\_cliente, c.nombre ASC;





Plan de ejecución de Oracle.

Los datos tienden a ser muy cortos, pues es ver que usuarios usaron una oferta especifica. Luego al haber tantas ofertas y reservas es muy raro que una oferta sea pedida muchas veces. Por lo que un simple scan de la tabla OFERTA y un nested join con RESERVA ya da la lista de reservas con los posibles usuarios que cumplan. Finalmente se aplica el filtro de las fechas, que con los índices creados debería ser rápido.

Aproximadamente 0,07 segundos para una oferta que devolvió 3 usuarios.

Nuestro plan de ejecución sería igual, ordenar reserva, aplicar hash para ver directamente que reservas son aptas en el rango pedido, realizar un nested join con la tabla reserva, habiendo escaneado previamente la fila de la oferta requerida. Luego con la tabla resultante buscar cada cliente usando su índice.