

Entrega final iteración 2 Sistemas Transaccionales

Juan Andrés Bernal Gil, Juan Felipe Serrano Martínez

Entrega iteración 2

Universidad de los Andes, Bogotá, Colombia

{ja.bernalg1, j.serrano}@uniandes.edu.co

Fecha de presentación: Abril 10 de 2023

Tabla de contenido

1	Introducción	1
2	Iteración.....	¡Error! Marcador no definido.
2.1	Modelos.....	¡Error! Marcador no definido.
2.2	Resultados logrados	¡Error! Marcador no definido.
2.3	Estructura del proyecto	¡Error! Marcador no definido.
2.4	Resultados no logrados	¡Error! Marcador no definido.
2.5	Balance funcionamiento de la aplicaicón	¡Error! Marcador no definido.
2.6	Diagramas de secuencia	¡Error! Marcador no definido.
2.7	Conclusiones	¡Error! Marcador no definido.
3	Bibliografía.....	¡Error! Marcador no definido.

1 Introducción.

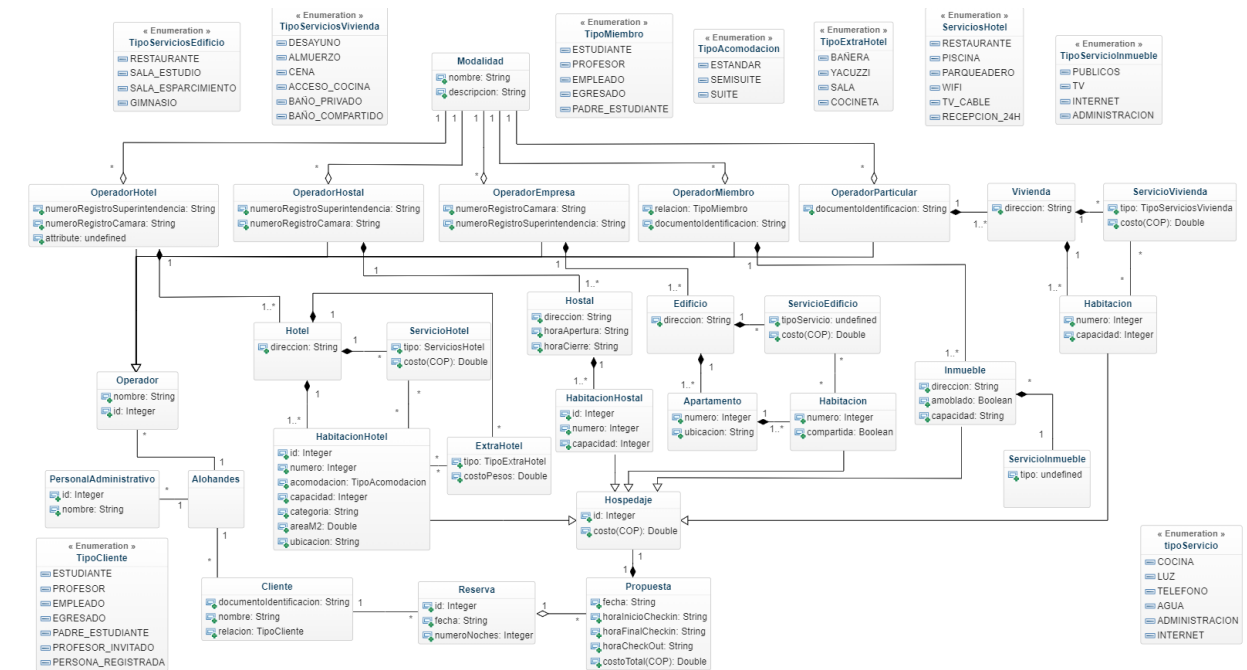
El presente documento presenta el análisis y e implementación requerida para la iteración número dos de la materia Sistemas Transaccionales (ISIS2403).

2 iteración

Inicialmente, en el negocio se identificaron los siguientes modelos, los cuales fueron los implementados:

2.1 Modelos:

Modelo conceptual:



2.2 Resultados logrados:

Se logró implementar toda la base de la persistencia necesaria para los requerimientos funcionales propuestos, al igual que el correcto cumplimiento de los requerimientos funcionales de consulta en base a los datos propuestos por nosotros.

2.3 Estructura del proyecto:

- En `docs/modelos/` podrá encontrar el modelo relacional y conceptual utilizado para este proyecto.
- En `docs/sql/create-tables/` podrá encontrar todas las sentencias de creación de tablas.
- En `docs/sql/data` podrá encontrar sentencias de limpieza de base de datos y scripts de creación del esquema completo.
- En `docs/sql/datos` podrá encontrar sentencias de inserción de registros de prueba para poblar la base de datos y soportar los requerimientos funcionales.
- En `docs/sql/rc` podrá encontrar sentencias de consulta para soportar los requerimientos de consulta.

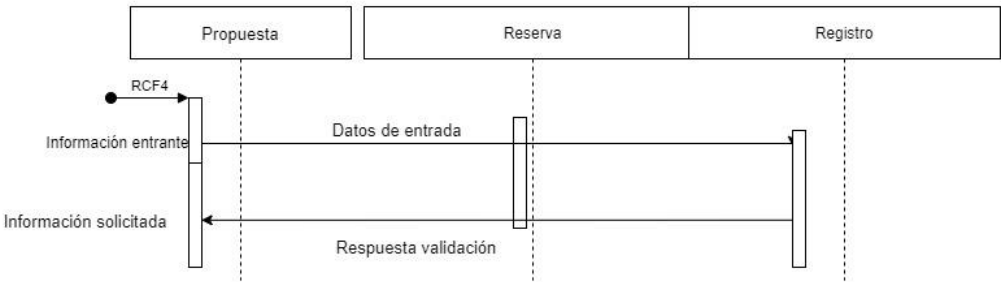
Se considera entidades propuestas por el enunciado deberían ser persistidas ya que proveen información necesaria para el correcto funcionamiento del modelo de datos en general. También facilitan el funcionamiento si se mantienen las dependencias para garantizar que no vayan a haber datos irrelevantes después de un borrado o modificación de algún otro dato del cual había una dependencia.

2.4 Resultados no logrados:

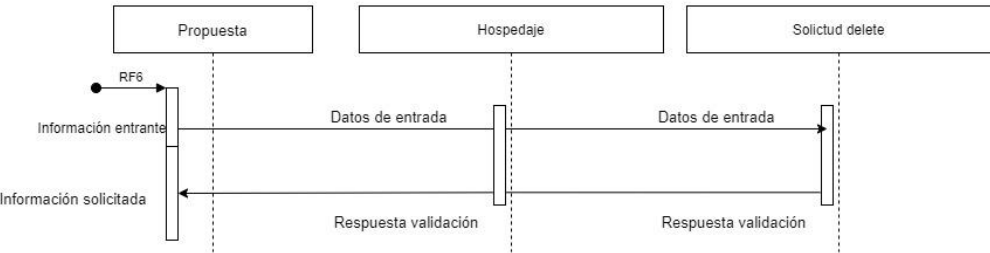
Desafortunadamente como grupo, quisimos haber tenido la oportunidad de profundizar más en las pruebas, haciéndolas en lo posible más específicas a cada caso, si bien las que propusimos responden a como tendrían que responder, consideramos que pudimos haber tenido un alcance mas amplio en ellas. Al igual, que no logramos implementar correctamente la interfaz propuesta en el proyecto de ejemplo parranderos.

2.6 Diagramas de secuencia iniciales:

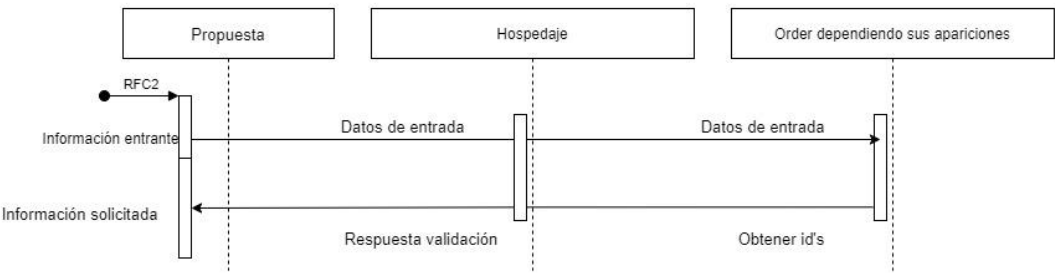
RF 4: Rol: Cliente



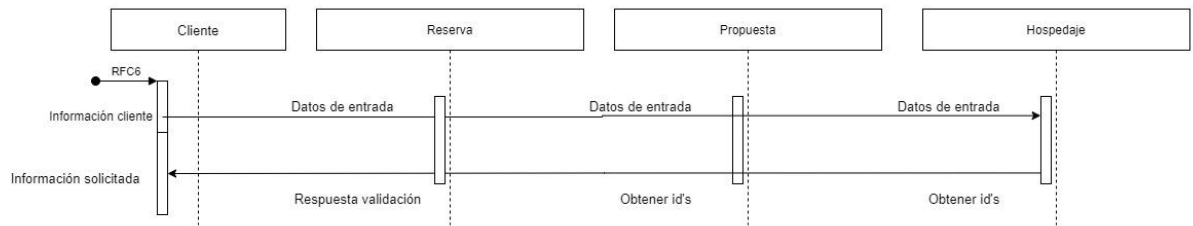
RF 6: Rol: Administrador del sistema



RFC 2: Rol: Cliente



RFC 6: Rol: Administrador



Consideramos que los requerimientos no funcionales lograron ser implementados en nuestro modelo de forma exitosa dado que, evidenciamos una diferencia clave entre los roles de los usuarios de la base de datos. Consideramos que, gracias a nuestro modelo relacional, se pueden guardar de forma exitosa los datos, dada la independencia de cada una de sus las clases con respecto a las otras. En base a cada pedido/solicitud, evidenciamos que es posible recibir más de una petición diferente al tiempo y finalmente se da a conocer que la base de datos está centrada en el dominio, los clientes y los administradores del sistema para obtener cada información necesitada.