

### **Usuario embebe su tipo de usuario:**

Simplicidad: Un usuario y su tipo están fuertemente relacionados y suelen ser consultados juntos, lo que justifica un modelo embebido para facilitar el acceso y la comprensión.

Query Atomicity: Dado que los datos se consultan juntos frecuentemente, embeberlos mejora el rendimiento de las consultas.

Update Complexity: Al actualizar un usuario, su tipo de usuario también podría necesitar actualización, lo que es más directo si los datos están embebidos.

RF5 & RF7: Al registrar y consultar la llegada y salida de un cliente, tener el tipo de usuario embebido permite un acceso rápido a la información necesaria para procesos de check-in y check-out.

### **Tipo habitación embebe una lista de elementos de habitación:**

Go Together: Los elementos de una habitación están directamente relacionados con su tipo y se benefician de estar contenidos dentro del mismo documento para operaciones de consulta y actualización.

RF1 & RF2: Esto simplifica las operaciones CRUD para los tipos de habitación, ya que los elementos relacionados se pueden gestionar directamente dentro de un mismo documento.

### **Habitación referencia su tipo de habitación:**

Cardinality: La cardinalidad entre habitaciones y tipos de habitación puede ser alta, y la referencia evita la duplicación excesiva de datos.

Data Duplication: Referenciar previene la duplicación de los elementos del tipo de habitación en cada habitación, lo cual sería ineficiente.

RFC2: Al mostrar el índice de ocupación, es más eficiente referenciar el tipo de habitación para evitar leer datos innecesarios que estarían embebidos.

### **Habitación embebe las reservas hechas en ella:**

Query Atomicity: Esto mejora la eficiencia al recuperar todas las reservas de una habitación en una sola consulta.

Archival: Las reservas pueden ser archivadas junto con la habitación, facilitando la gestión del ciclo de vida de los datos.

RF4: Esto facilita la consulta y actualización de reservas dentro de una habitación, ya que todas las operaciones CRUD se pueden realizar en el contexto de una sola habitación.

### **Las reservas referencian su plan de consumo:**

Workload: Los planes de consumo pueden ser escritos y leídos en diferentes cargas de trabajo, lo que justifica la referencia para evitar cuellos de botella en la escritura.

RF6 & RFC3: Referenciar permite gestionar los planes de consumo de forma independiente, lo cual es útil cuando se registran o consultan consumos para un rango de fechas específico.

#### **Las reservas embeben los usuarios que las reservan:**

Query Atomicity: Esto asegura que la consulta de una reserva traiga consigo los datos necesarios del usuario sin consultas adicionales.

RF4: Permite una gestión integral de la reserva, incluyendo la información del usuario, sin necesidad de realizar múltiples consultas.

#### **Los usuarios embeben los consumos que hicieron:**

Simplicity: Los consumos están estrechamente ligados a la actividad de un usuario y son más fáciles de manejar cuando están embebidos en el mismo documento.

Individuality: Los consumos no suelen existir sin un usuario asociado, lo que justifica su embebido.

RF6 & RFC3: Facilita el registro y la consulta de consumos por usuario, permitiendo un acceso directo a los consumos sin consultas adicionales.

#### **Las habitaciones embeben los consumos hechos por usuarios hospedados en esa habitación:**

Go Together: Estos consumos están intrínsecamente ligados a la habitación y su embebido permite un acceso y actualización eficiente.

RFC1: Permite calcular de manera eficiente el dinero recolectado por servicios en cada habitación, ya que los datos necesarios están contenidos dentro del mismo documento de la habitación.

#### **Los consumos referencian el servicio consumido:**

Individuality: Un servicio puede existir independientemente de un consumo y ser referenciado por muchos consumos, evitando así la duplicación innecesaria.

RF6 & RFC4: Al referenciar los servicios en los consumos, se facilita la actualización de los servicios sin tener que modificar cada consumo, y se simplifica la consulta de clientes que han consumido un servicio específico.

#### **Los servicios embeben los productos que ofrecen:**

Go Together: Los productos son parte integral del servicio y su relación cercana justifica el embebido para consultas y actualizaciones cohesivas.

RF3: Simplifica las operaciones CRUD en los servicios del hotel, ya que la información del producto está directamente disponible y no requiere consultas adicionales.