

Nicolás Rincón 202021963

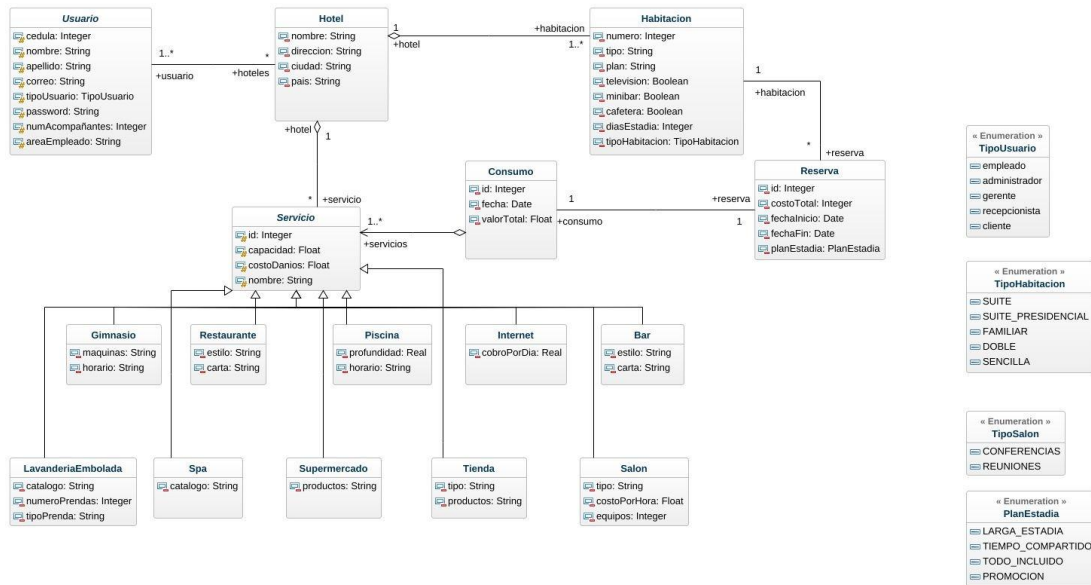
Santiago Najar 202021647

Juana Morales 202021591

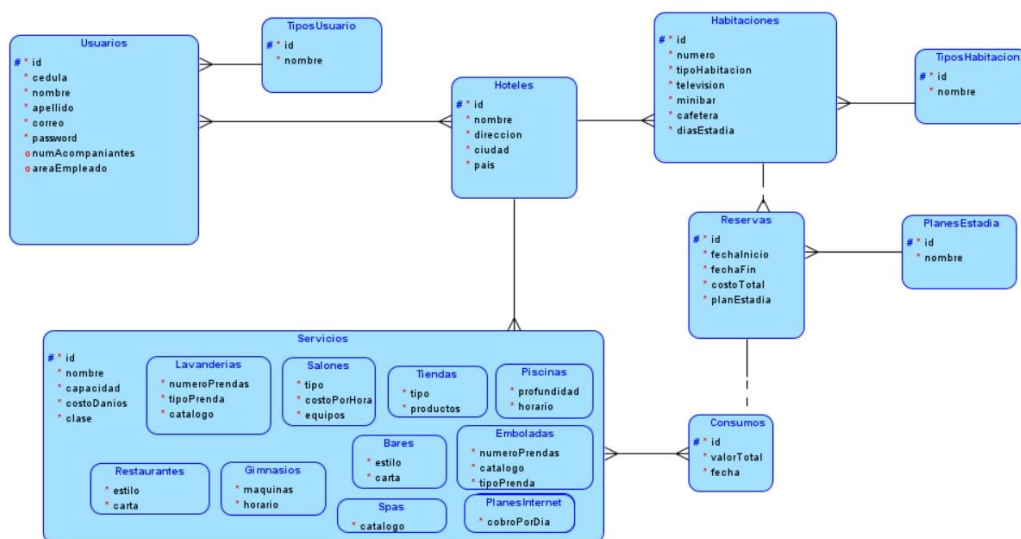
## Entrega 1 – Hotel de Los Andes

### Modelos solicitados

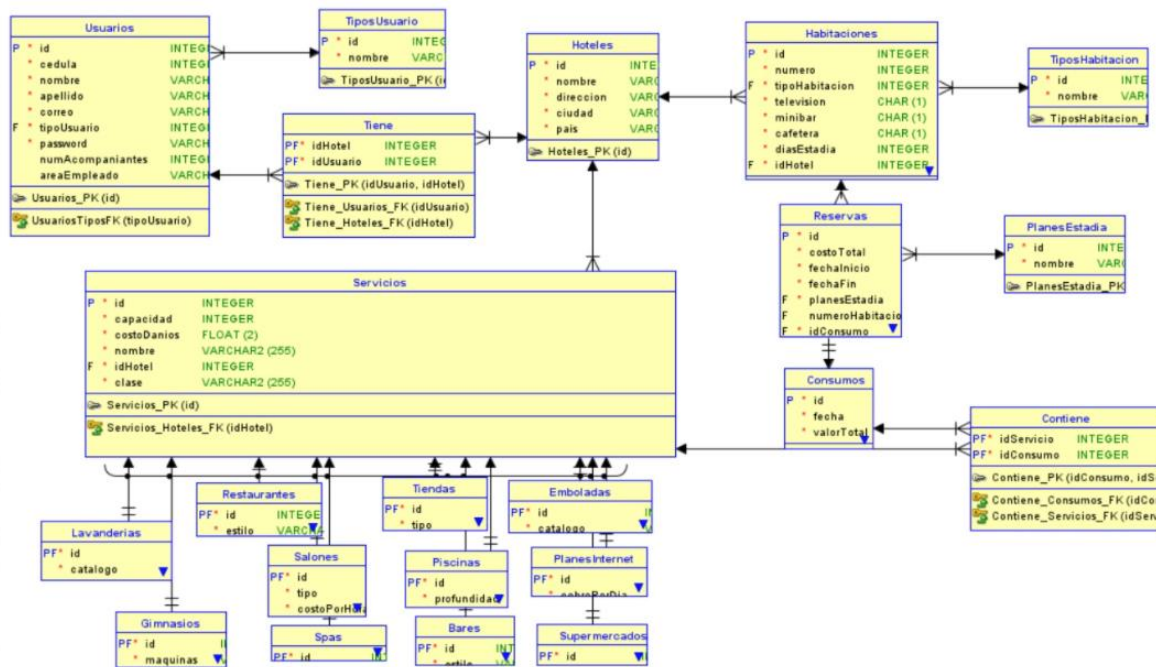
### UML



### E/R



## MODELO RELACIÓN



### Normalización.

#### Primera forma normal

Todas las tablas cumplen con atributos atómicos, todos los valores son de un mismo tipo y no hay filas duplicadas en las tablas.

#### Segunda forma normal

Todos los atributos de nuestras tablas dependen únicamente de la llave primaria

#### Tercera forma normal

Todos los atributos de nuestras tablas dependen únicamente de la llave primaria por lo que las transitividades son nulas.

### Resultados logrados

- Modelos conceptuales, implementación de DataModeler (realización de la ingeniería para obtener el modelo relacional), generación del DDL e implementación de esta en SQL.
- Ejecución de todo el back de la aplicación: creación de las entidades, controllers, repositorios, requerimientos funcionales (RF1, RF2, RF3, RF4, RF5, RF6, RF7)
- La persistencia en los datos se logró de manera exitosa.

### Resultados no logrados

- Requerimientos funcionales (9, 10, 11) en cuanto a la implementación del HTML para que se visualizarán en el front.

### **Supuestos adicionales sobre las reglas de negocio encontradas en el caso de estudio:**

- 1) Los usuarios saben y únicamente van a meterse a la parte correspondiente de los botones dependiendo del tipo de usuario que sea. Si es administrador por ejemplo solo se meterá a la parte de administrados.
- 2) La base de datos cuenta con memoria infinita por lo que no habrá colapso nunca de información dado a la cantidad de registros en un hotel.
- 3) Los usuarios únicamente van a inspeccionar la aplicación desde el frontend por lo que se espera que no vean y ni manipulen el backend.

### **Documentación de los Requerimientos Funcionales**

#### **RF1 - Tipos de Usuarios:**

El sistema otorga al administrador la habilidad de administrar completamente los tipos de usuarios. Esto significa que el administrador puede establecer, revisar, actualizar o eliminar cualquier tipo de usuario detallado en el enunciado. Recibe el nombre o el id necesario para la operación.

#### **RF2 - Usuarios:**

El sistema facilita la gestión de perfiles de usuario por parte del administrador de datos del hotel. Cada perfil tiene atributos clave como tipo de cedula, nombre, correo electrónico, etc. Esta funcionalidad puede recibir la info de los atributos de cada usuario necesaria para ejecutar las operaciones.

#### **RF3 - Tipos de Habitación:**

El administrador de datos del hotel tiene el poder de gestionar los diferentes tipos de habitaciones que el hotel ofrece. Esto incluye determinar las características y comodidades de cada tipo de habitación. Dicho esto esta funcionalidad recibe la información necesaria para para lograr estas funcionalidades básicas.

#### **RF4 - Habitación:**

La gestión detallada de cada habitación individual en el hotel es facilitada por el sistema y es responsabilidad del administrador de datos del hotel. Esta funcionalidad puede recibir información clave como el número de habitación, valores booleanos de los objetos que tiene entre otras.

#### **RF5 - Servicio del Hotel:**

El administrador de datos del hotel tiene la capacidad de detallar y gestionar todos los servicios que el hotel ofrece, desde piscinas hasta bares y restaurantes. Esta funcionalidad

recibe el id, nombre, el costo por posibles daños asociados a uno de estos servicios para lograr hacer estas operaciones básicas.

#### RF6 - Plan de Consumo:

Los planes de consumo del hotel, ya sean estándar o promocionales, son gestionados por el administrador de datos del hotel. Estos planes determinan las ofertas y paquetes disponibles para los huéspedes durante su estancia, lo que puede incluir comidas, servicios y otras comodidades. Recibe esta información.

#### RF7 - Reserva de Alojamiento:

Los clientes tienen la capacidad de reservar habitaciones para períodos específicos a través del sistema, siempre que la habitación deseada esté disponible.

Ningún requerimiento anterior devuelve un mensaje de exitoso, sino que simplemente agregar a la base de datos las nuevas modificaciones del código. En caso de haber un error si se mostrar en pantalla cual es el debido error.