

DOCUMENTO DE INFORME

Análisis del proyecto

Al terminar de leer enunciado se encontraron varias entidades claras:

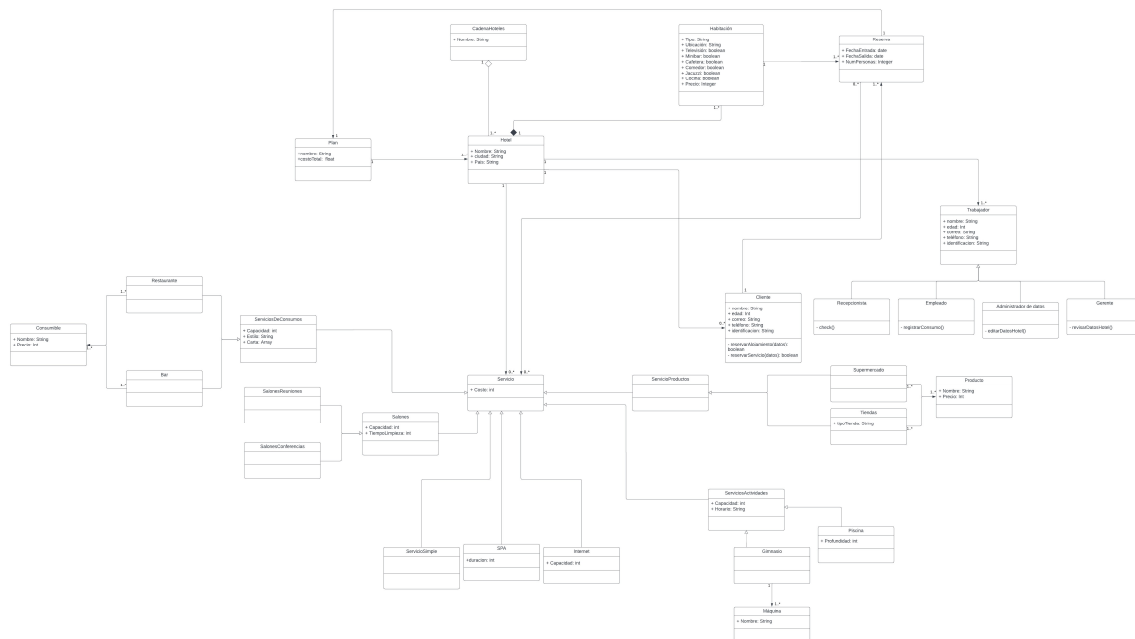
- La primera es el usuario, esta corresponde en un principio a una superclase que engloba tanto al cliente como a todos los tipos de empleado, después de analizar mejor el funcionamiento del programa se decidió separar en dos clases: cliente y empleado, para poder asignar mejor los métodos funcionales de cada uno.
- El hotel es la clase principal sobre la que gira la aplicación, por lo que esta es de vital importancia en el diseño.
- Al principio se menciona que existen cadenas de hoteles, por lo que esta clase también se debe modelar.
- Un hotel está compuesto de habitaciones, por lo que esta clase debe existir obligatoriamente.
- En el enunciado se mencionan planes a los que un usuario debe acceder para hospedarse en un hotel, entonces debe existir una clase plan.
- Una parte importante del hotel son los servicios que brinda, para lo que se crea una clase principal llamada servicios. Esta clase es una superclase de todos los servicios especificados, los cuales se buscó agrupar de la manera más eficiente posible. Para esto se crearon:
 - Servicios de productos, que son todos los servicios que dentro de los que se venden cosas, ya sea comida, ropa, etc. Para estos existe una clase adicional que representa los productos. Dentro de esta clase se encuentran restaurantes, bares, tiendas y supermercados.
 - Servicios simples, que son los servicios para los que solo se requieren el servicio y el costo, entre estos están préstamos de servicios, lavado planchado y embolado.
 - Gimnasio, piscina, SPA e internet son servicios con características diferentes a todos los demás, por lo que cada uno se modeló como una subclase separada. Además, para gimnasio se creó una clase para las máquinas de hacer ejercicio.
- Por último, hay una clase que permita tener registro de toda la información de la estadía de un cliente en un hotel. Para esto se modela una clase reserva, que almacena los consumos del cliente en el paso por el hotel.

Después de esto obtener todas las clases necesarias, hay que asociarlas para que todo tenga sentido. Para esto se utilizó la siguiente lógica:

- Una cadena de hoteles tiene uno o varios hoteles.
- Un hotel tiene una o varias habitaciones.
- Un hotel tiene cero, uno o muchos clientes.
- Un hotel tiene uno o muchos trabajadores.
- Un hotel puede tener muchos servicios.
- Un hotel tiene uno o varios planes.
- Una reserva solo puede tener una habitación y solo un cliente para el que se registra la reserva.
- Las reservas pueden acceder a muchos servicios, así como un servicio puede ser usado por muchas reservas.

- La clase Servicios debe heredar todos los tipos de servicios.
- La clase trabajadores hereda los 5 tipos de trabajadores.

Al tener toda la información anterior se puede modelar un UML para el problema. A continuación, se mostrará el modelo UML, que también se encuentra adjunto en la carpeta Docs. del repositorio.



Hay varias observaciones que se pueden hacer sobre el modelo:

- Todas las clases tienen los atributos que se mencionan en el enunciado, y son innecesarios de enumerar en este documento.
- La asociación entre hotel y habitación es de composición debido a que no pueden existir habitaciones sin un hotel.
- Se creó una enumeración para los tipos de habitaciones, ya que estos son específicos y no varían.
- Las características adicionales de las habitaciones, ya que estos pueden estar o no en cada habitación.
- Aunque en el enunciado se menciona que los servicios se cargan a la habitación, se decidió que es mejor que estos se carguen sobre cada reserva, ya que las habitaciones pueden alojar muchas reservas a lo largo del tiempo mientras que cada reserva es única.

Creación del modelo E/R

Para crear el modelo entidad-relación se tomó como base el UML diseñado anteriormente y se logró un modelo completo del enunciado del caso. En este modelo, como es natural, se van a encontrar ciertas diferencias con respecto al diagrama UML. Lo primero, es que en el diagrama existe una asociación de herencia entre trabajador y los tipos de trabajador, para esto se tomó la decisión de solo crear una entidad para trabajador, ya que estos en cuanto a atributos son iguales, para definir

los tipos de trabajador se creó una entidad adicional con los tipos. Para los tipos de habitación se realizó algo parecido con los tipos, lo que es crear una entidad de tipos de habitación. En el caso de todas las herencias de servicios se modelaron cada una de las subclases presentes.

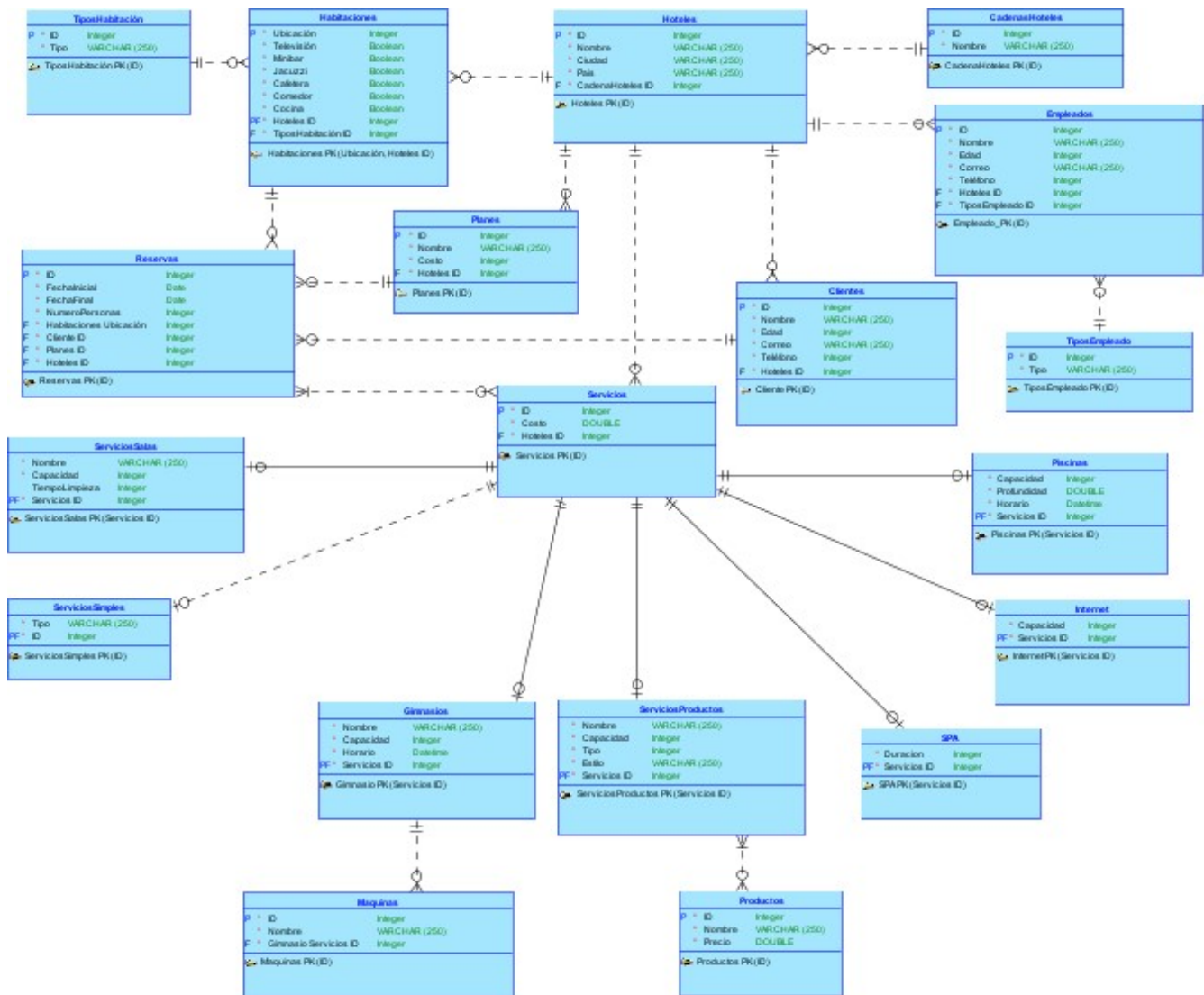
Hablando específicamente de cada entidad:

- Hoteles, tienen como llave primaria un ID asignado por el sistema, porque al revisar todos los elementos se llegó a la conclusión de que todos atributos pueden repetirse por lo que la mejor manera para identificar cada hotel debe haber un ID. Esta entidad tiene una relación cadena de hoteles, por lo que tiene una llave foránea para identificar la cadena de hoteles a la que pertenece.
- Cadena de hoteles, también se generó un ID ya que se asume que el nombre puede ser el mismo para varias cadenas.
- Habitaciones, tiene dos llaves, una por la relación con hoteles para saber a que hotel pertenece y otra para identificar el tipo de habitación. En esta entidad la llave primaria está compuesta por la ubicación (se asume que la ubicación corresponde a un número como en el caso de los salones de la universidad) y el id del hotel al que pertenecen.
- Planes, tiene como llave primaria un ID generado para evitar problemas con los nombres, ya que estos se pueden repetir. Este debe almacenar la información del hotel al que pertenece, por lo que en planes existe una llave foránea del hotel.
- Reservas, tiene varias llaves foráneas debido a las asociaciones que tiene, estas son: id de un cliente, id de un plan y la llave primaria de la habitación (con esta ya se tiene referencia sobre el hotel). Todas estas se deben a que las otras entidades tienen muchas reservas, por lo que es ineficiente que estas cuenten con el registro de reservas (también haría que no puedan cumplir con la primera forma normal). En esta se tomó como llave primaria un ID generado por el sistema, ya que una la única llave candidata estaba compuesta por tuplas que pueden llegar a ser complicadas de manejar (fechas, habitación).
- Clientes, en esta la llave primaria es un ID generado. Por la asociación con hotel, este debe tener una llave foránea para saber en que hotel está alojado, esto significa que cada vez que un cliente quiera ir a un hotel nuevo, este se debe registrar y se le generará un ID nuevo.
- Empleados, corresponde a trabajadores, tiene como llave primaria un ID generado. Este debe tener la información del hotel para el que trabaja, así como el cargo que tiene, por lo que hay una llave foránea del id del hotel y una con id del tipo de empleado.
- Servicios, esta cuenta con un ID asignado por el sistema. Este tiene una llave foránea que corresponde al hotel, ya que este pertenece solo a un hotel.
- Para todas las subclases de servicios estás tienen como llave primaria una llave foránea proveniente de servicios. Para cada uno de los posibles servicios se añaden los atributos adicionales de estos, y en caso de servicios que requieran de objetos adicionales, se crearon entidades que los representen.
- Para los productos, se hizo una asociación con servicios productos de muchos a muchos, ya que un producto lo pueden ofrecer varios servicios, así como un servicio puede ofrecer muchos productos.
- En el caso del gimnasio, se crearon las máquinas que almacenan el id del gimnasio porque estas son únicas en un gimnasio.
- Para almacenar la información de consumo para una habitación, como se mencionó anteriormente, se decidió que esta información debe estar relacionada con la reserva. Para lo que hay una asociación de muchos a muchos entre reservas y servicios. En el modelo relacional

esta asociación se puede ver mejor, así como la asociación entre productos y servicios de productos.

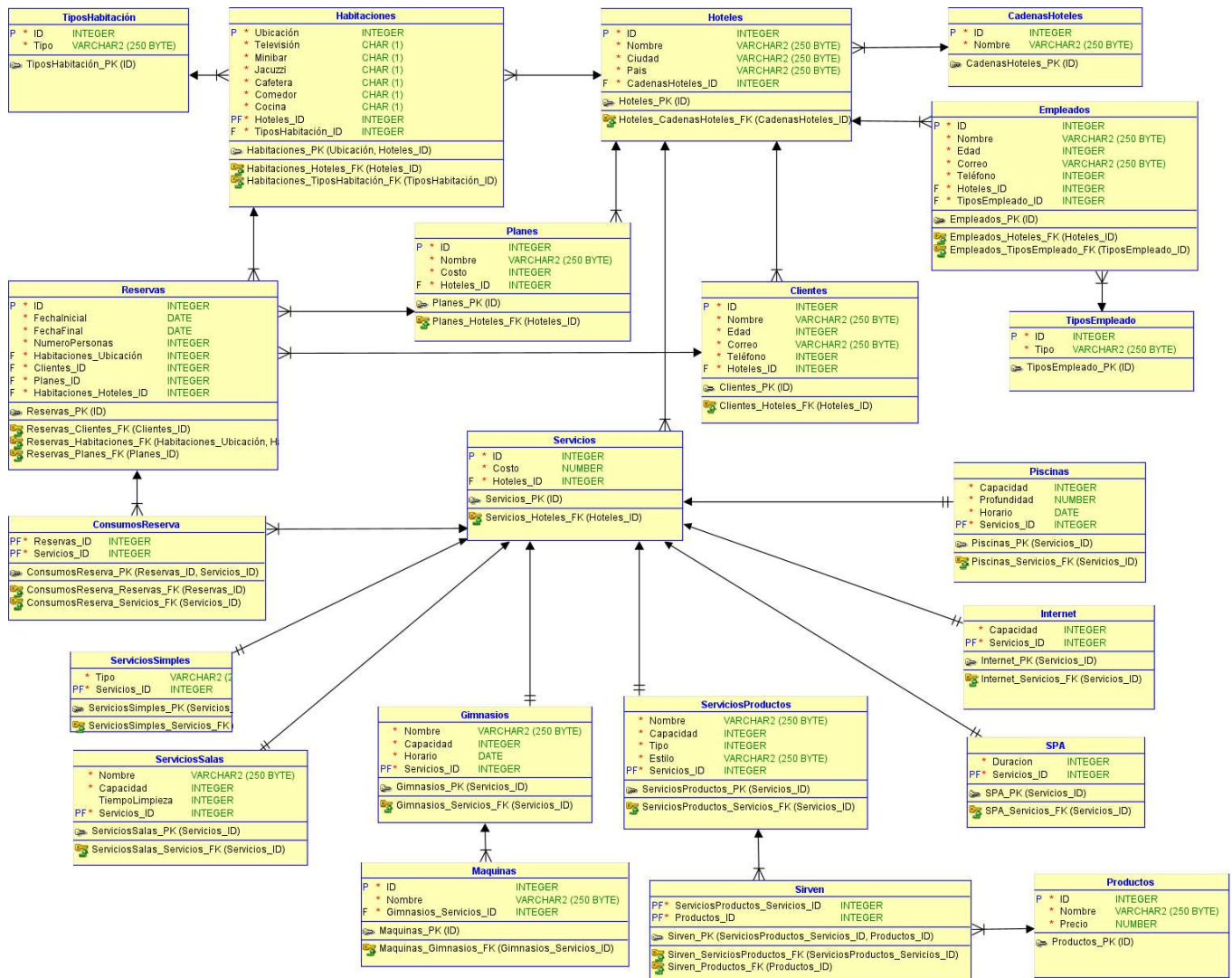
- Por último, las asociaciones que corresponden a los tipos de empleado y los tipos de habitación cuentan con un ID y el tipo correspondiente. La llave primaria de estas es el ID el cual llega a empleados y habitaciones respectivamente.

A continuación se encuentra el modelo Entidad-Relación que se desarrollo, que se encuentra adjunto por separado en esta misma carpeta, tanto en formato fuente como png:



Modelo relacional

Después de realizar el modelo de Entidad-Relación hay un gran avance para el diseñar un modelo relacional. Para este siguen faltando varias cosas; como la asociación entre reservas y servicios, o la asociación entre servicios de productos y productos. Para esto se diseñan tablas intermedias que guarden información de ambas entidades de la asociación. A continuación, se muestra el resultado del diseño del modelo relacional (también se encontrará adjunto por separado en la carpeta):



Normalización del modelo

A partir del modelo expuesto en este documento, podemos analizar la normalización.

1. Primera forma normal:

Todo el modelo se encuentra en la primera forma normal porque todos los atributos son atómicos. En los casos donde se pudieron crear listas como atributos, estos fueron descompuestos en tablas. Por ejemplo, las máquinas pudieron ser una lista dentro de gimnasio, pero se tomó la decisión de crear una entidad adicional.

2. Segunda forma normal:

El modelo se encuentra en segunda forma normal porque todos los atributos de cada entidad dependen completamente de la llave primaria. En el caso de habitaciones para determinar los atributos no primos, es decir todos los booleanos y el tipo de habitación dependen tanto de la ubicación como del hotel. El resto de las entidades que no corresponden a relaciones tienen llaves primarias atómicas por lo que es imposible descomponerlas. Por último, las tablas consumosReserva y sirven solo están compuestas por llaves primarias al corresponder a relaciones, por lo que toda la entidad se determina con la tupla de los dos atributos.

3. Tercera forma normal:

El modelo se encuentra en tercera forma normal porque no hay relaciones transitivas entre atributos primos. Para el caso de los tipos de habitación y los tipos de empleados, estas en un principio podían pertenecer a habitación y empleados respectivamente, pero para un modelo más ordenado se separaron en entidades que solo almacenan esta información. En el resto de las entidades no se encuentran atributos que puedan depender de atributos no-primos.

4. Forma normal de Boyce-Codd

El modelo se encuentra en forma normal de Boyce-Codd porque no existe superclaves de las entidades que se traslapen con otra superclaves. Esto se debe a que casi todas las superclaves en las entidades son atómicas, y en el único caso que no lo es, es decir en habitaciones, esta contiene íntegramente a la superclave de hoteles, por ende no se traslapan.