UNAM

FACULTAD DE INGENIERIA

INGENIERIA EN COMPUTACION

PROGRAMACION ORIENTADA OBJETOS

ANALISIS PREVIO A LA IMPLEMENTACION

CARRILLO SANCHEZ RICARDO
HERNANDEZ GOMEZ ALEJANDRO
ZARAZUA RAMIREZ JOHAN AXEL

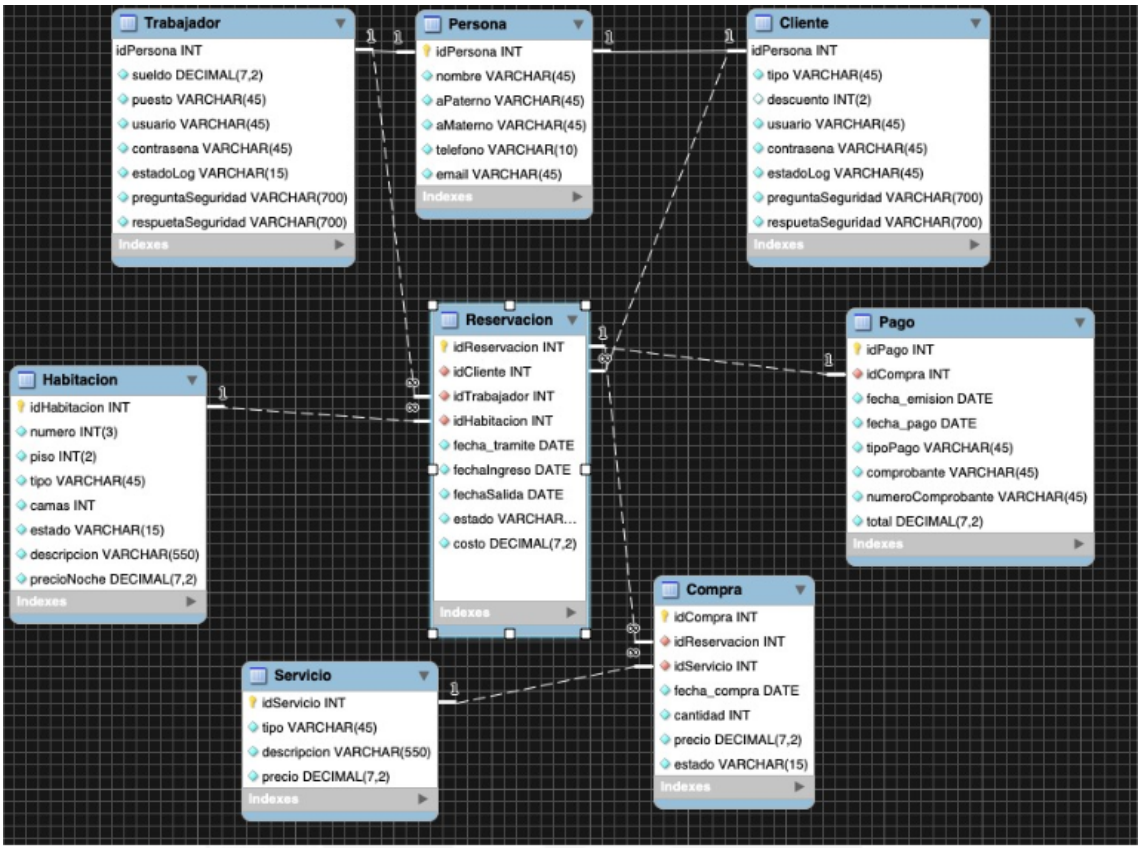
GRUPO: 04

PROFESOR: EDGAR TISTA GARCIA

Analisi previo a la implementacion de un sistema de administraciÃşn de hotel

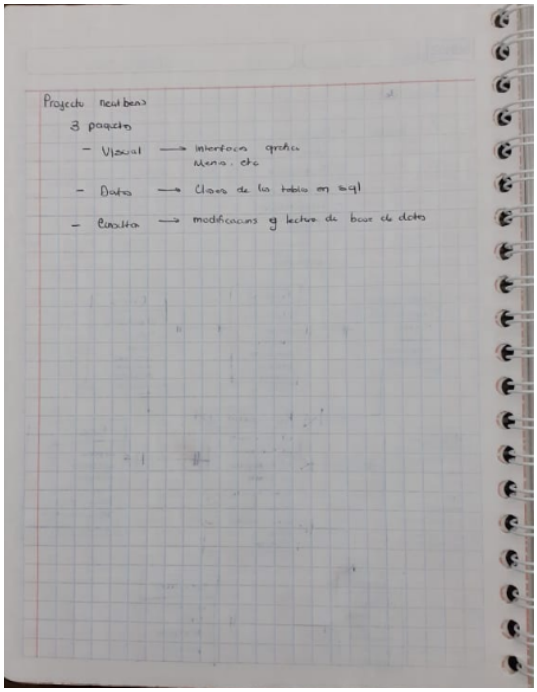
1 DiseÃśo

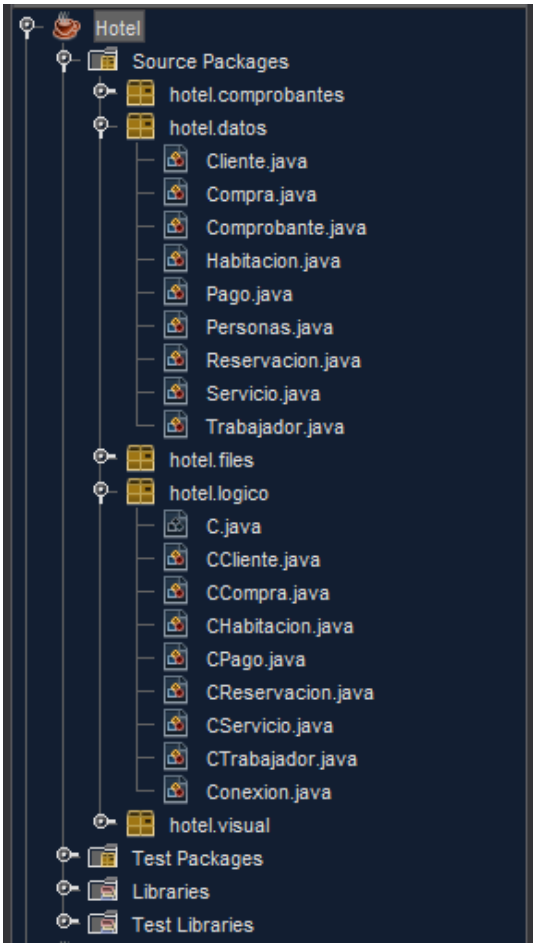
Para poder hacer el sistema se creo una base de datos con base en el siguiente esquema de entidad relaciÃșn



De esta manera podemos hacer que en una tabla tengamos un columna con la cual podamos ligar columnas de otra tabla, esto lo hicimos ya que detectamos que en el sistema podrÃąamos tener una composiciÃșn de objetos ya que una

reservaciÃșn tiene un cliente, un trabajador, una habitaciÃșn, etc. Por lo tanto pudimos hacer estas composiciÃșn a partir de la base datos haciendo que no exista una composiciÃșn de clases en el cÃ³digo de Java y favoreciendo a una cohesiÃșn baja.
En cuanto al codigo en java se divide se dividio en 3 paquetes distintos para poder separar nuestras clases y hacernos mas sencilla la solucion de un problema, mantenimiento, etc. En el paquete datos nos encargamos de crear una clase para cada una de las tablas en la base de datos, esto con el fin de almacenar la informaciÃșn necesaria en un objeto y despuÃ©s enviarla a la base de datos mediante este objeto.
En el paquete lÃ³gico se planeo crear las clases mediante las cuales se haran operaciones en la base de datos como pueden ser consultas, inserciones, actualizaciones o eliminaciÃșn de datos, en estas clases se penso en las posibles consultas que podrÃŁamos hacer por lo cual podemos ver sobrecarga de metodos en algunas de esta clases.
Por ultimo en el paquete visual se crearon cada una de las ventanas que vera el usuario, en ellas el podrÃą interactuar dependiendo el tipo de usuario que sea.





En conclusiÃșn podemos decir que las relaciones entre las tablas de la base de datos son muy importantes para el diseÃśo de nuestro problema ya que a partir de ellas podemos hacer que ciertos datos tengan una conexiÃșn pero sin depender de otros, por lo que podemos eliminar sin tanto problema ya que estaremos seguros de que solo se elimina el dato que nos interesa y no otros datos, esto lo podemos ver en la relaciÃșn de pagos y reservaciÃșn ya que podemos eliminar un pago sin eliminar una reservaciÃșn, en cambio si esto se realiza por composicion de objetos podrÃąa darse el caso de que al eliminar un pago tambiÃ©n eliminemos una reservaciÃșn, si esto lo vemos en un aspecto Real podrÃąa suceder que un trabajador se equivoque al registrar un pago y al eliminarlo, eliminarÃąa con el a la reservaciÃșn por lo que tendrÃąas que volver a crear la reservaciÃșn y posteriormente crear de nuevo el pago