Creación de la base de datos

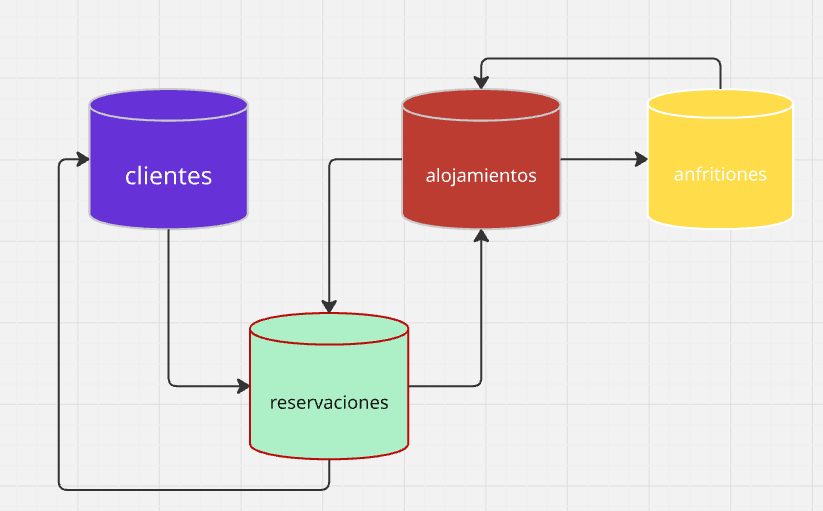
Analizando la naturaleza del sistema y los parámetros que tenemos que cumplir en sus funcionalidades, el mayor reto que se debe afrontar a lo largo de la solución del problema está sujeto a la replicación de datos innecesarios; ya que estos datos estarían ocupando gran cantidad de memoria mientras el programa se esté ejecutando. Dado esto y un análisis de las características del sistema se han definido los siguientes objetos. Los cuales son:

* Alojamientos
* Anfitriones
* Clientes (también se pueden llamar huéspedes)
* Reservaciones
* Fechas

Aunque podemos visualizar 5 objetos diferentes y cada uno de ellos poseen características diferentes; estas características poseen relaciones estrechas con las características de otros objetos; un ejemplo claro es el hecho de que un anfitrión puede tener varios alojamientos y sería un desperdicio tener una característica del objeto anfitrión y que se repita en varios alojamientos sabiendo que el anfitrión se conserva igual. Otro ejemplo, es como se relacionan las reservaciones con las fechas.

Teniendo en cuenta lo anterior, como conclusión; en el programa se debe evitar la replicación de datos ya que, en caso contrario, pueden existir fugas de memoria y además complica de gran manera la funcionalidad del programa. Por tanto, se pensó la siguiente base de datos que cumple con las funcionalidades requeridas por el sistema.

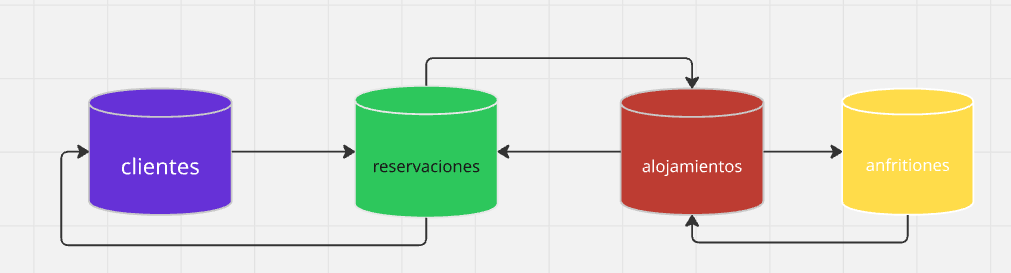
La base de datos forma tiene principalmente 3 arrays y su contenido son objetos. La primera las de los alojamientos, la segunda anfitriones, y tercera clientes. Después de eso tenemos una base de datos dentro del cada uno de los objetos alojamientos que será para corresponder a las reservas relacionadas para cada alojamiento. En el mundo de los dibujos se representa de la siguiente manera:

(*el contorno de la base de datos de reservaciones es el mismo que alojamientos para interpretar que está dentro de alojamientos.)*

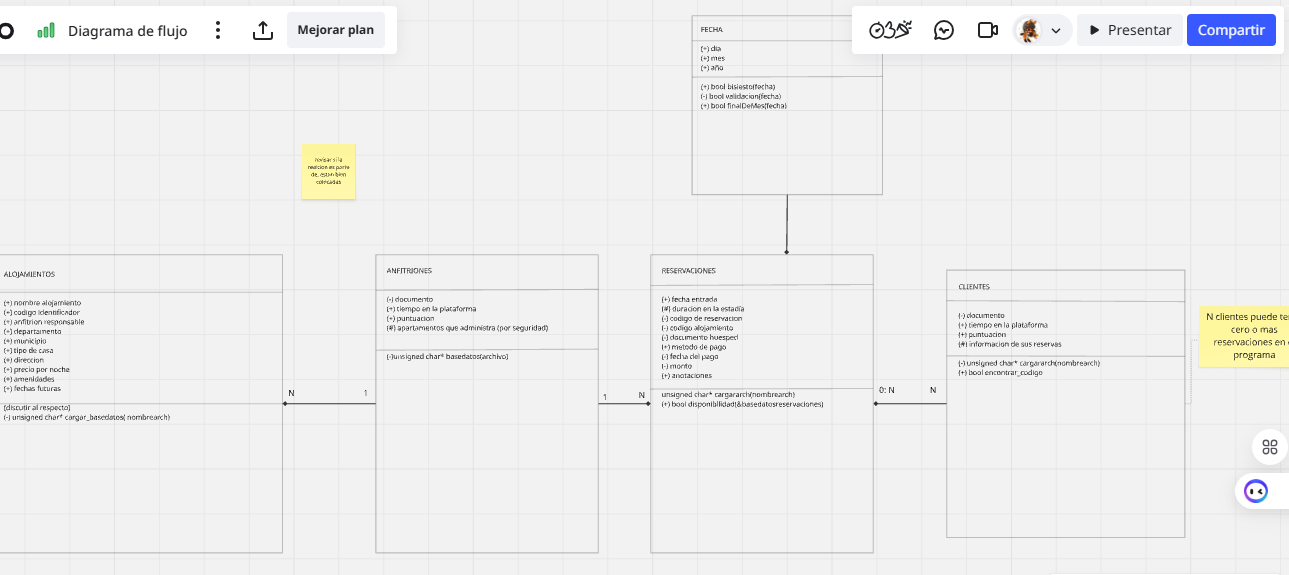
Con esta estructura se resuelve el problema sobre las relaciones entre los distintos objetos; y, por otro lado, el problema de la replicación de datos se resuelve utilizando punteros.

A continuación, un ejemplo enfocado en los anfitriones y los alojamientos.

Sabemos que tenemos un anfitrión puede tener 1 o N alojamientos. Para el objeto anfitrión unas de sus características es que tenga una lista con todos sus alojamientos, y para el objeto alojamiento debe tener una caracteriza anfitrión. Como pueden ver una opción sería replicar esos objetos en las características de del objeto, pero ya tenemos una base de datos que ya los tiene. La mejor opción sería cambiar esas características y que tengan un apuntador que apunte al objeto que se necesita. En el caso del objeto anfitrión tendría una lista de apuntadores que apuntan a sus alojamientos, y para el objeto alojamiento un apuntador que apunta a su anfitrión. Esto mismo ocurriría para el objeto reservas que está dentro del objeto alojamiento, que estaría en una lista de las reservas y las reservas apuntan al cliente y el cliente a punta a sus reservaciones.

Otra base de datos planteada es la de reservaciones ya que deberían estar en un array aparte y no estén dentro de la base de datos de alojamientos.

Teniendo en cuenta lo anterior. El diagrama de clases hasta el momento seria de la siguiente manera:



ETAPA DE PROCESO EN EL DESARROLLO DEL PROGRAMA: a medida que se avanzaba en la implementación del programa, nos percatamos que el diagrama de clases planteado inicialmente no era el correcto ya que faltaban métodos y/o cambios de tipos en los atributos.

Por esta razón, el diagrama de flujo quedó de la siguiente manera: (adjunto link porque es muy grande el diagrama): <https://miro.com/welcomeonboard/cW1jQWdIanBWSDlFUVpTRXMvYTl3SStTTVFxYk1tYWl0d0NDbUdtSUxIbnNXRWRjL0Y4c3BEQjlWN0NvRHlSZlFDb28rQit4YjJtbitnLzg0UmltY0pJM3JGUnc5MzA3MFNPWUNiQmNZMmprVTFOYkpZWFREa3lwVVJHclhMTTlhWWluRVAxeXRuUUgwWDl3Mk1qRGVRPT0hdjE=?share_link_id=935666561403>

PROBLEMAS AFRONTADOS DURANTE EL DESARROLLO DEL PROGRAMA:

Durante el desarrollo del programa, nos percatamos de la existencia de diversos factores problemas a la hora de cargar las bases de datos, ya que al existir varios alojamientos que pertenecen a un anfitrión en particular, a la hora de reservar la memoria, en muchas ocasiones se creaban “craheos” de la memoria y esto impedía poder seguir con el desarrollo del programa. Estos “crasheos” estaban directamente relacionados con la vinculación de los anfitriones por medio de punteros que tuvieran las direcciones de memoria de sus alojamientos (siendo este uno de los factores más importante para el desarrollo del programa). La memoria se fragmentaba constantemente. Sin embargo, nos percatamos que el error estaba en la mala utilización de los constructores (se pasaba al constructor equivocado) o, por otro lado, no se había creado el constructor que se hiciera cargo de este proceso, por tanto, la solución estuvo en la implementación de más sobrecargas y métodos necesarios para cada tarea en particular.