

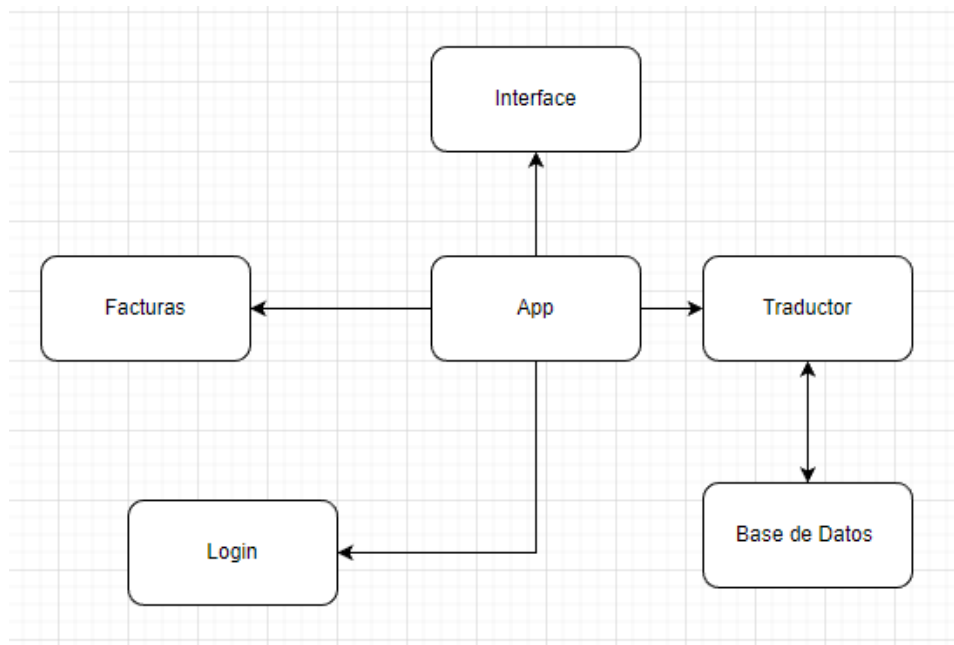
## Documento diseño

### Primera iteración

En la primera iteración creamos unos módulos encargados de aspectos específicos del programa. Decidimos crear estos módulos para tener un mayor entendimiento de la aplicación y para empezar a tener un mayor encapsulamiento en el programa.

Los módulos que creamos fueron:

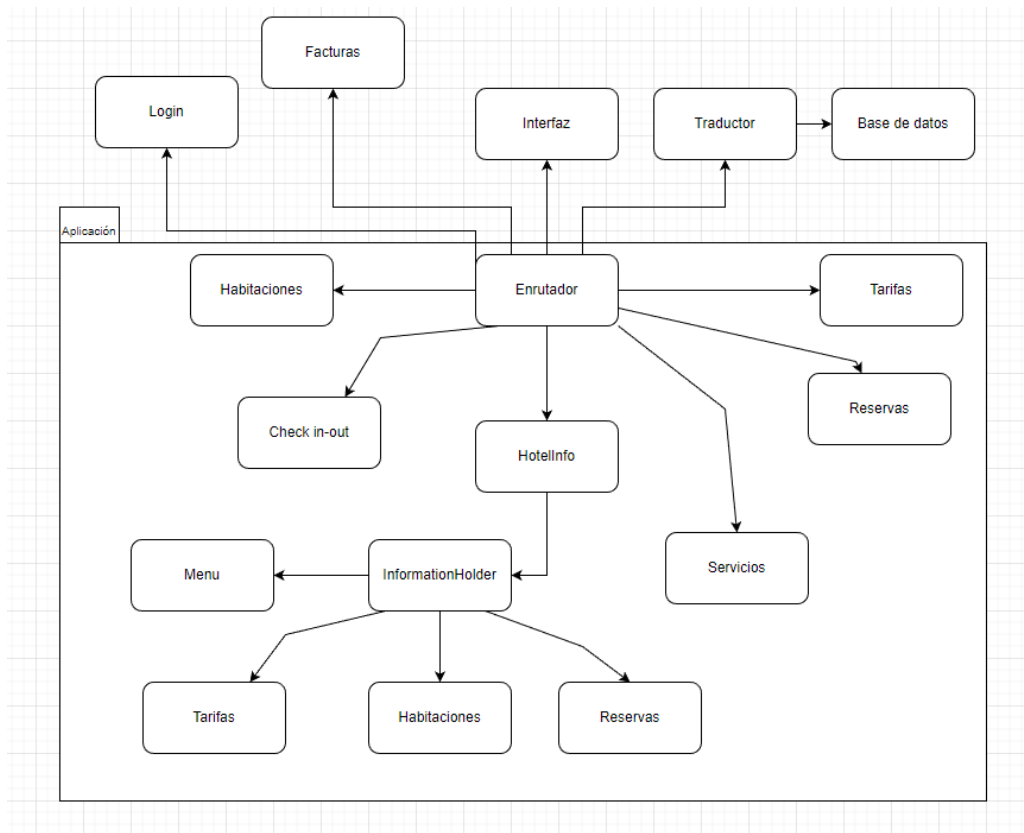
- Interfaz: Módulo con la función de mediar lo que el usuario recibe e ingresa en el programa.
- Facturas: Módulo con la función de crear y guardar las facturas de los huéspedes del hotel.
- Traductor: Módulo con la función de cambiar objetos a archivos y archivos a objetos, de forma que elementos en la base de datos puedan pasar de la base de datos al programa y del programa a la base de datos.
- Login: Módulo con la función de verificar los datos del usuario cuando vaya a entrar al programa
- App: Centro del programa. Encargado de lidiar con la lógica del programa (los requerimientos funcionales del programa). Debe almacenar y cambiar los datos del programa en forma de objetos.



### Segunda Iteración

Para la segunda iteración nos enfocamos en el módulo de aplicación, ya que este módulo era el más amplio, el más importante y el más abstracto en la primera iteración. En esta segunda iteración decidimos crear un primer módulo que estuviera encargado de la conexión e interacción entre los módulos fuera de

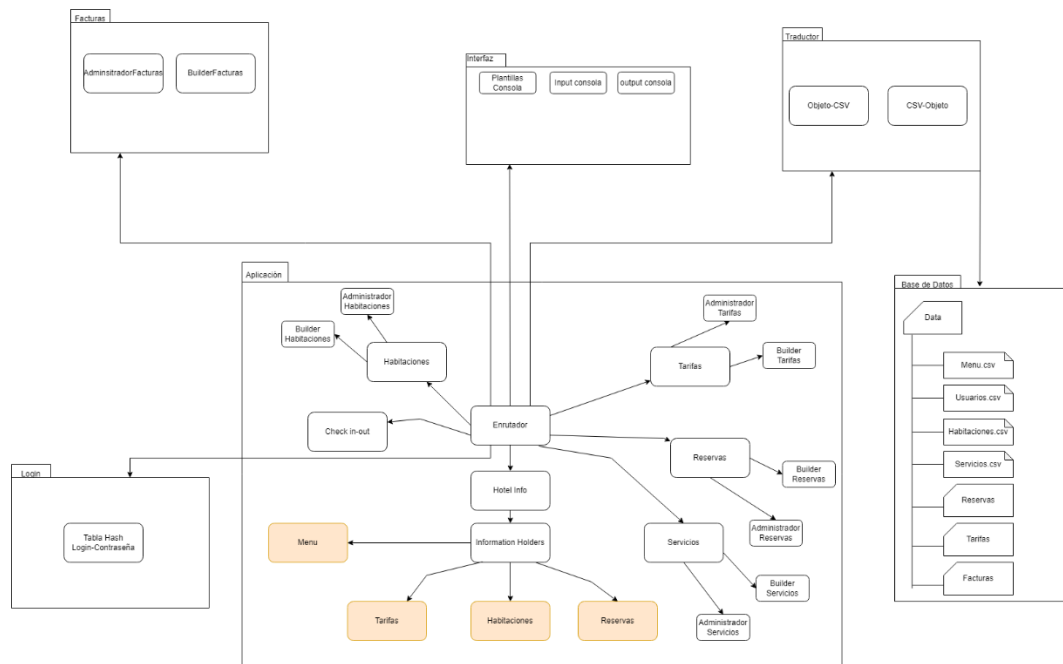
aplicación y los sub módulos por dentro. Para esto creamos un módulo con el rol de enrutador. A partir de este, empezamos a crear módulos con el propósito de administrar y crear cierto tipo de datos, como las reservas, las habitaciones, las tarifas y los huéspedes.



### Tercera Iteración

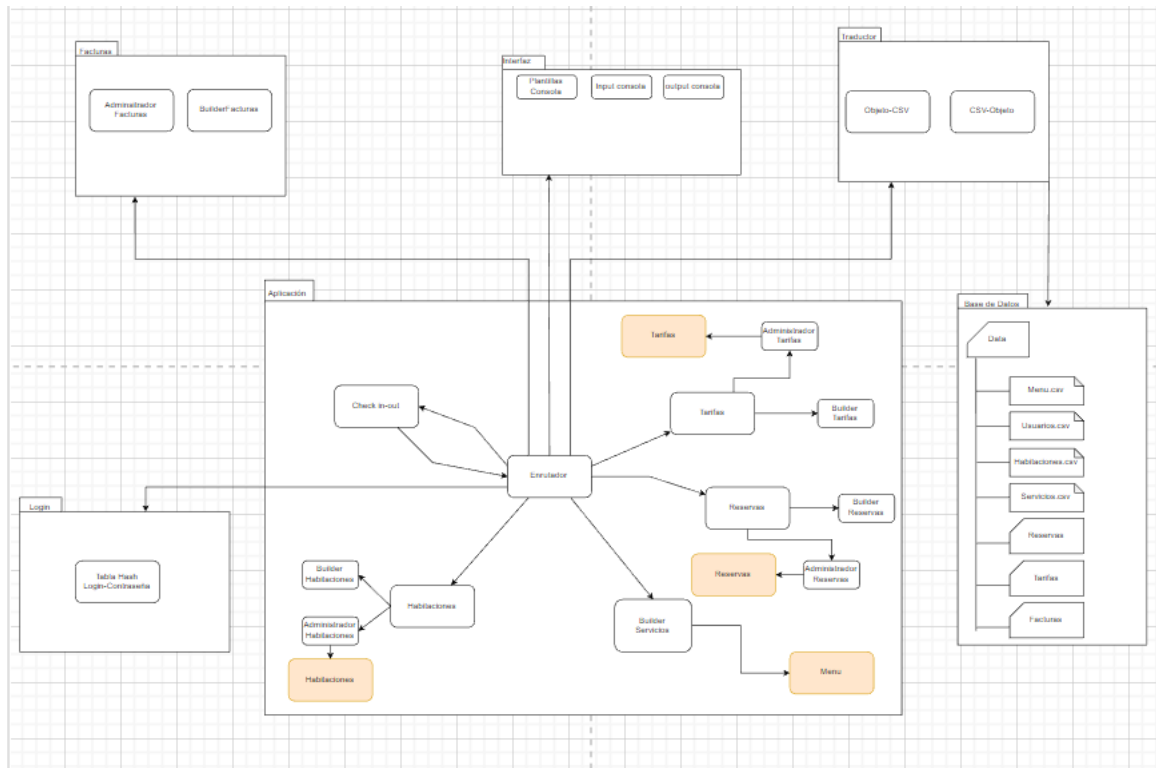
Para la tercera iteración decidimos empezar a centrarnos más en los módulos que principales del programa y los submódulos dentro de aplicación. Vimos que los submódulos dentro de aplicación tenían un componente de administrar y otro de crear. Para la parte de crear pretendemos usar el patrón builder, ya que los objetos dentro del programa tienen atributos que en un inicio no están definidos y que se van desarrollando a medida que crece el programa. Por otro lado, tenemos la parte que administra los objetos y la que se encarga de hacer los cambios en los objetos.

A parte, estructuramos la base de datos: Decidimos tener un archivo con el menú del restaurante, un archivo con la información del login de los usuarios y un archivo para las habitaciones y sus características. A su vez, tenemos 3 carpetas que contendrán los archivos de reservas, tarifas y facturas.



## Cuarta Iteración

En la cuarta iteración del diseño hicimos cambios grandes en la forma en que guardábamos la información dentro del programa. Antes teníamos un único módulo que se encargaba de guardar todos los datos, pero esto hacía complicado la guardar y cambiar información en el sistema si utilizábamos los módulos que tenían esta labor. Por ello, decidimos que cada módulo que administraba y creaba un tipo de datos también guardara la información. Esto hace más clara y simple la vida de los objetos dentro del programa.

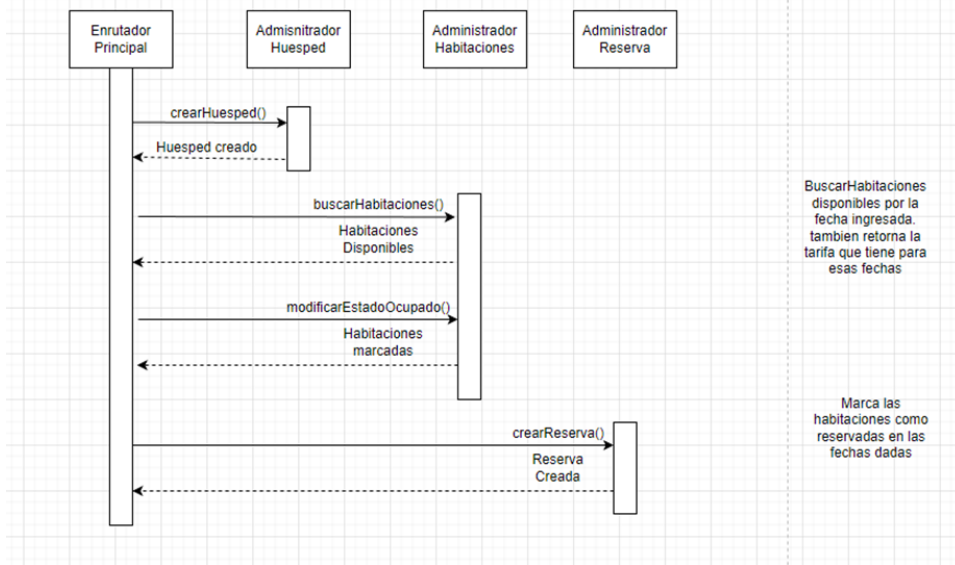


### Quinta iteración:

En nuestra quinta iteración ya empezamos a definir las clases y los métodos dentro de estas. Gracias a esto nos dimos cuenta de muchos módulos que no eran necesarios, otros que debían ser añadidos y otros que tenían que cambiar su relación y posicionamiento dentro del programa. Uno de estos principales fue que la interfaz ahora contiene al enrutador principal y no en la forma inversa. También añadimos un controlador para la base de datos, de forma que este se encargue de la base de datos como tal y de operar el traductor.

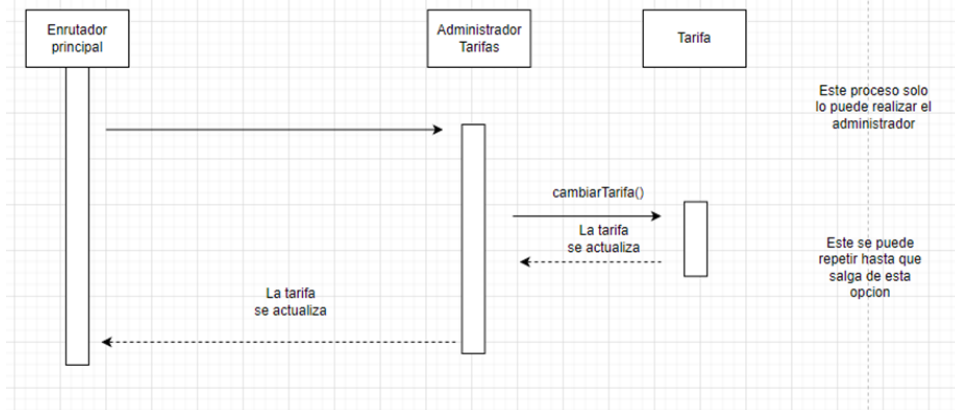
### Diagramas de secuencia:

Diagrama de secuencia para hacer una reserva



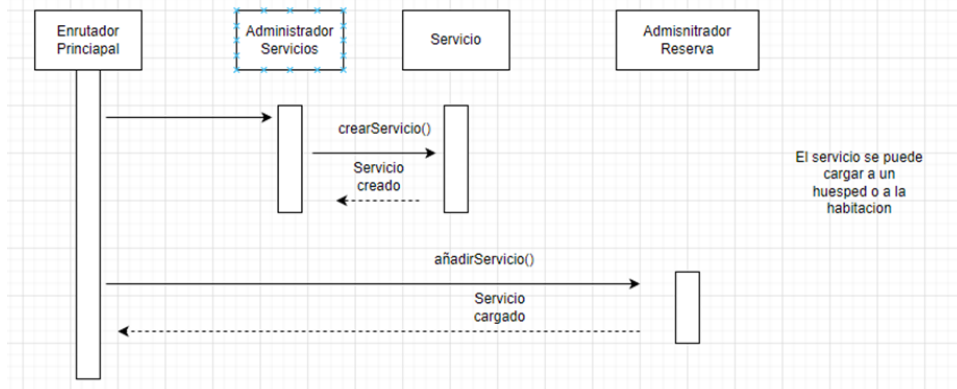
Lo primero que hace es crear el huésped principal (los acompañantes se crean cuando llegan y hacen el ingreso). Luego hace una búsqueda de habitaciones en las fechas que da el cliente, ahí informa de los precios. En cuanto eligen las habitaciones, se marcan en las fechas de reserva como ocupadas. Luego se crea la reserva.

Diagrama de secuencia para actualizar tarifas de habitaciones



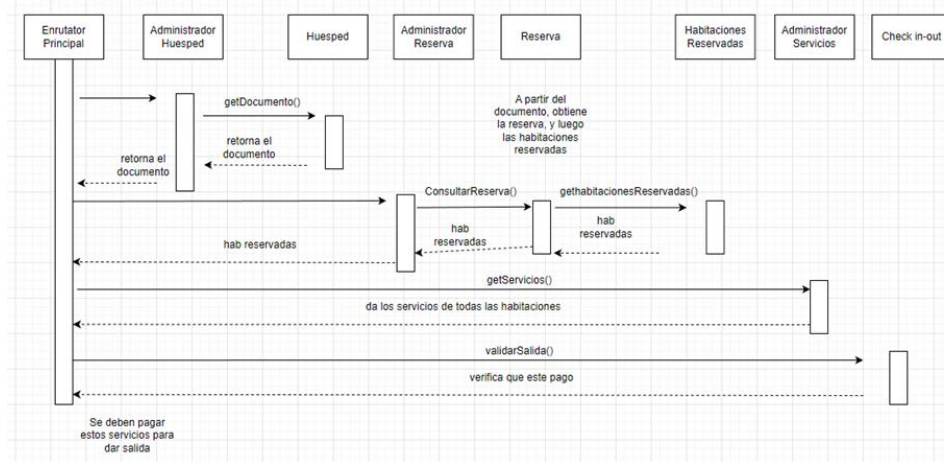
Esta opción solo la tiene el administrador. Entra al administrador de tarifas e ingresa los datos para cambiarlas. Esta se puede realizar varias veces. Importante, si se toman fechas que ya tienen tarifa, se escoge la mas barata.

Diagrama de secuencia para agregar servicios



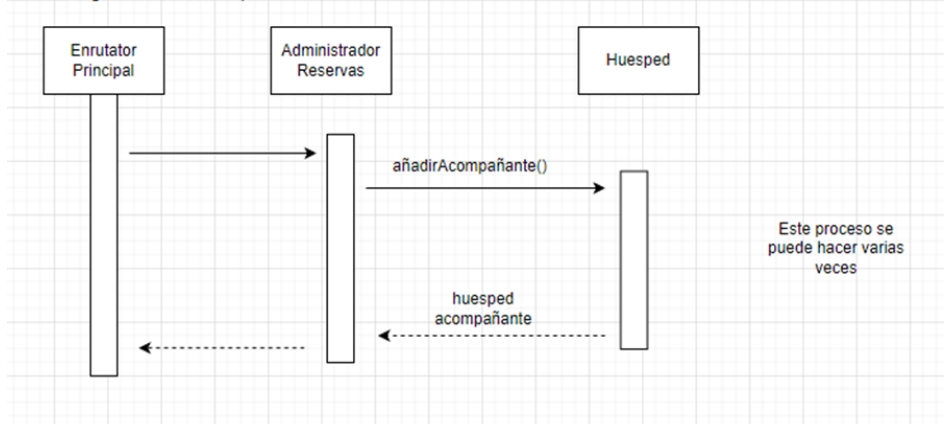
Primero se crea el servicio en el administrador, y luego se añade al huésped o a la habitación, dependiendo del tipo de servicio.

Diagrama de secuencia para hacer check - out



Primero se deben obtener todos los servicios de esa reserva a partir del documento del líder del grupo, para luego pagarlos y poder validar la salida

Diagrama de secuencia para hacer check - in



En el check-in se agregan los datos de los acompañantes de la reserva

**Restricciones:**

La información se almacenará en archivos csv

La consola se encarga de que a los empleados no les aparezca la opción de modificar tarifas, que es una función solo para el administrados

Los acompañantes solo se registrarán a la entrada, en la reserva solo se dará el dato de cuantos serán

Las fechas de reserva y tarifas se guardarán en listas de 365 doubles y booleanos.

Se guardarán tres listas de tarifas, una para cada tipo de habitación. El precio aumentara de manera fija por cada característica adicional que tenga, por ejemplo, cocina integrada. Se inician en negativo para indicar que no se ha definido una tarifa.