

Programación 4

Informe del Modelo de Diseño - Diagramas de Comunicación -

Grupo 49 Integrantes

Guillermo Coelho
Matias Rodriguez
Martina Rodriguez
Diego Salbarrey
Antonio Mures

CI: 5.043.831-6
CI: 4.630.803-4
CI: 5.039.353-8
CI: 4.377.137-7
CI: 5.158.394-4

Docente: Romina Parada

Diagramas de comunicación de los casos de uso:

Se hizo aclaraciones de las operaciones correspondientes a los diagramas para evitar copiar operaciones repetidas en varios diagramas por lo que se las mantuvo en un diagrama solo.

1. Alta de Usuario

Operaciones:

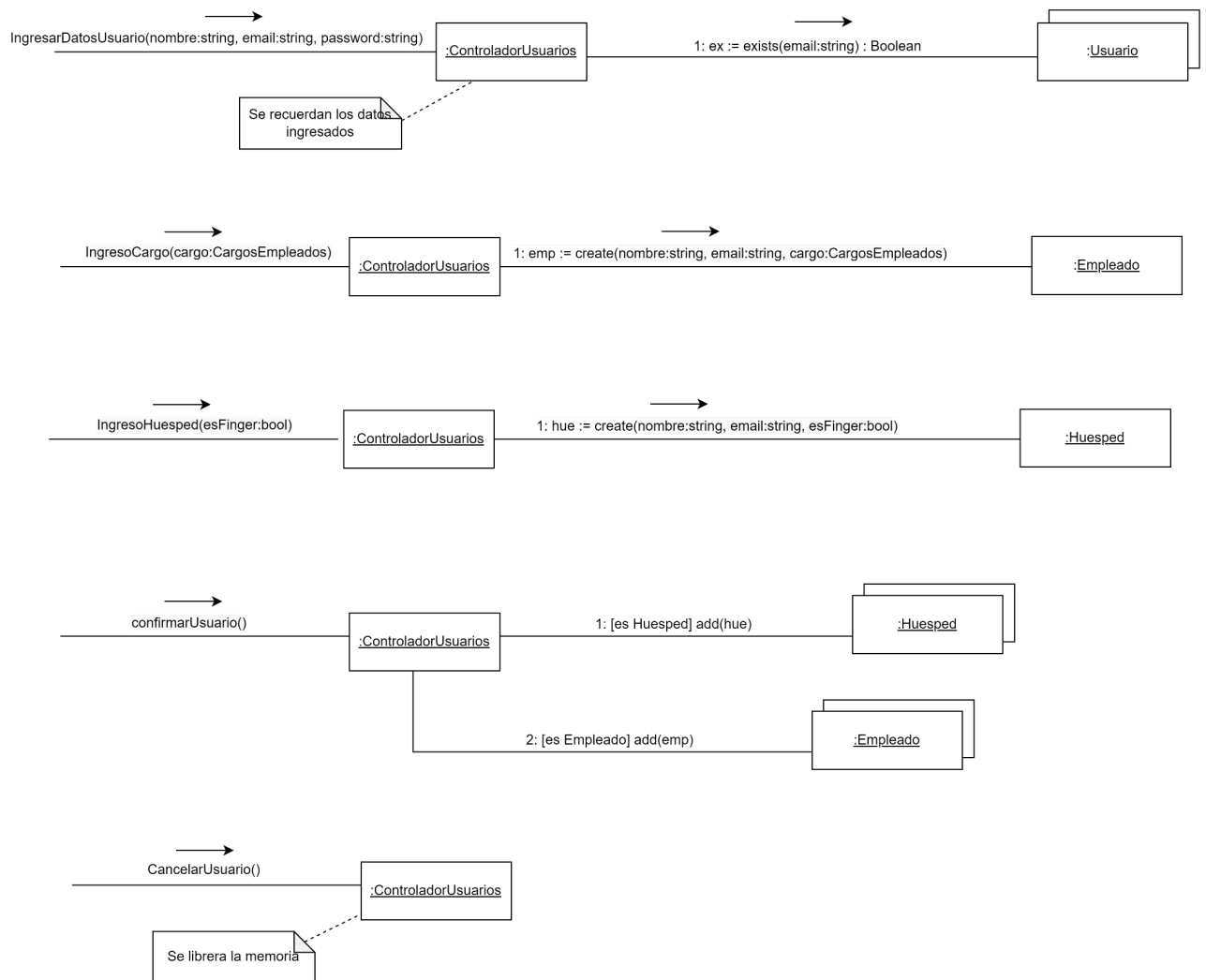
IngresarDatosUsuario

IngresoCargo

IngresoHuesped

confirmarUsuario

cancelarUsuario



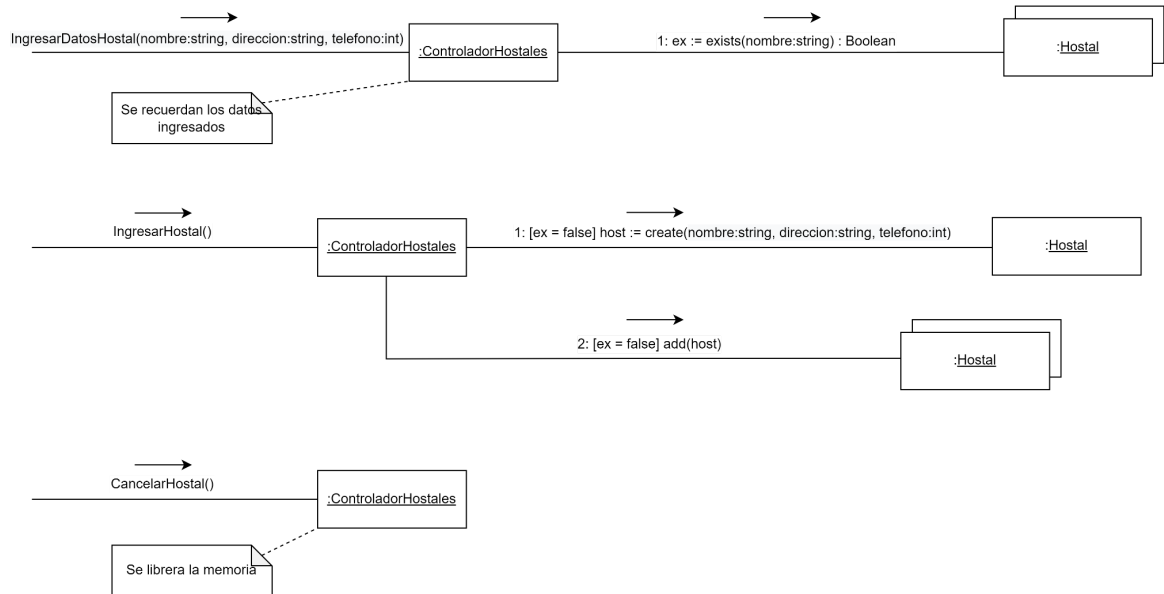
2. Alta de Hostal

Operaciones:

ingresarDatosHostal

ingresarHostal

cancelarHostal



3. Alta de Habitación

Operaciones:

listarHostales

elegirHostal

ingresarHabitacion

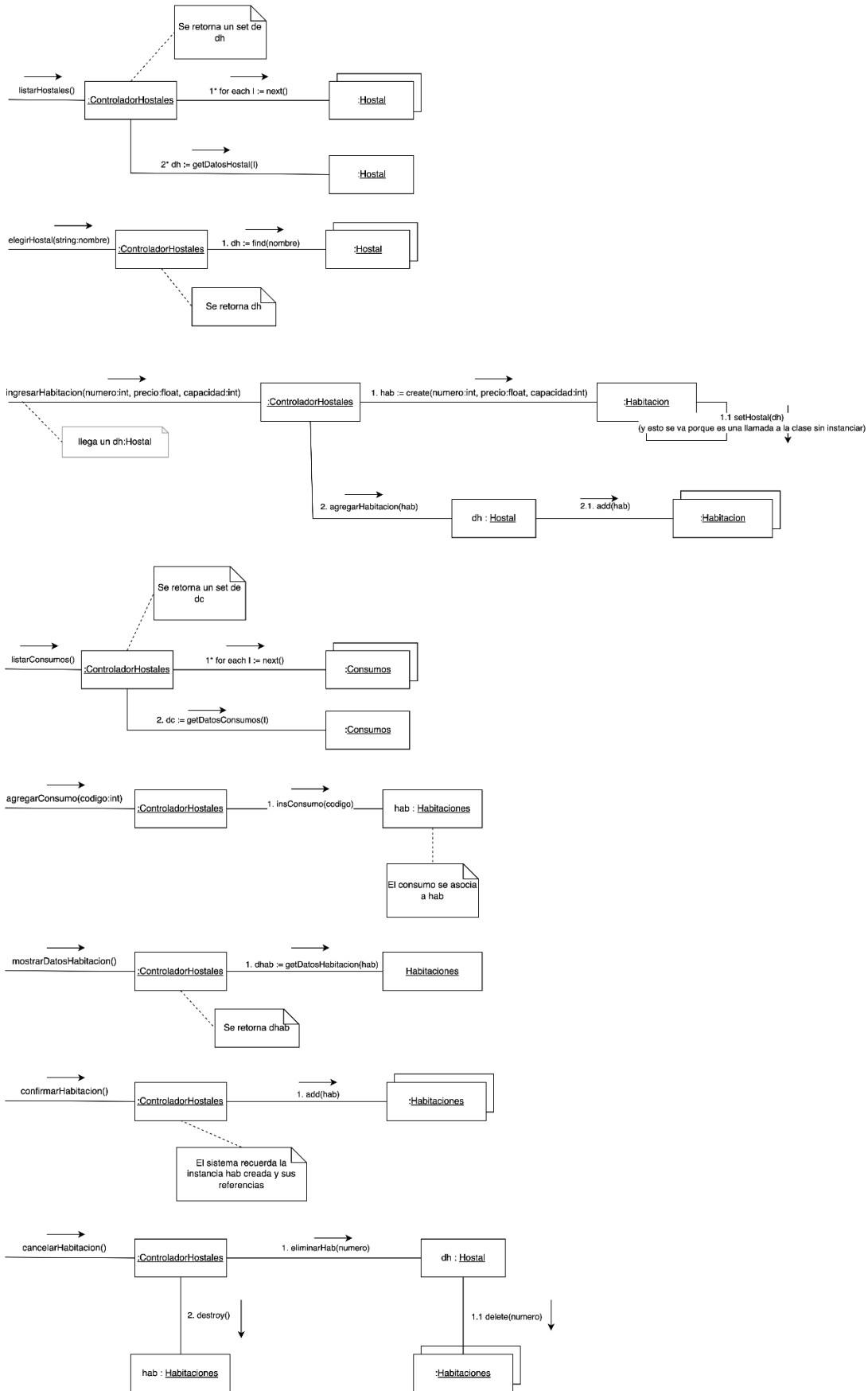
listarConsumos

agregarConsumo

mostrarDatosHabitacion

confirmarHabitacion

cancelarHabitacion



4. Asignar empleado a hostel

Operaciones:

listarHostales

elegirHostal

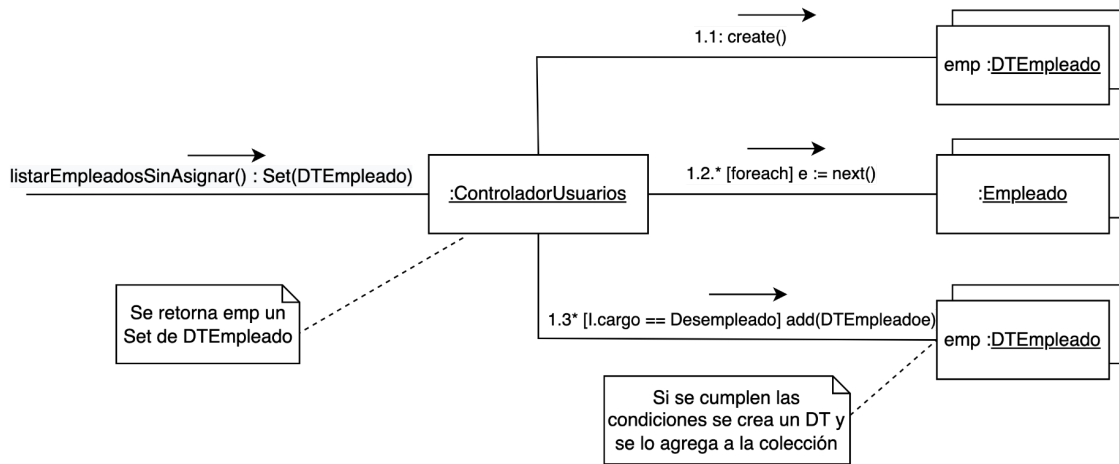
listarEmpleadosSinAsignar

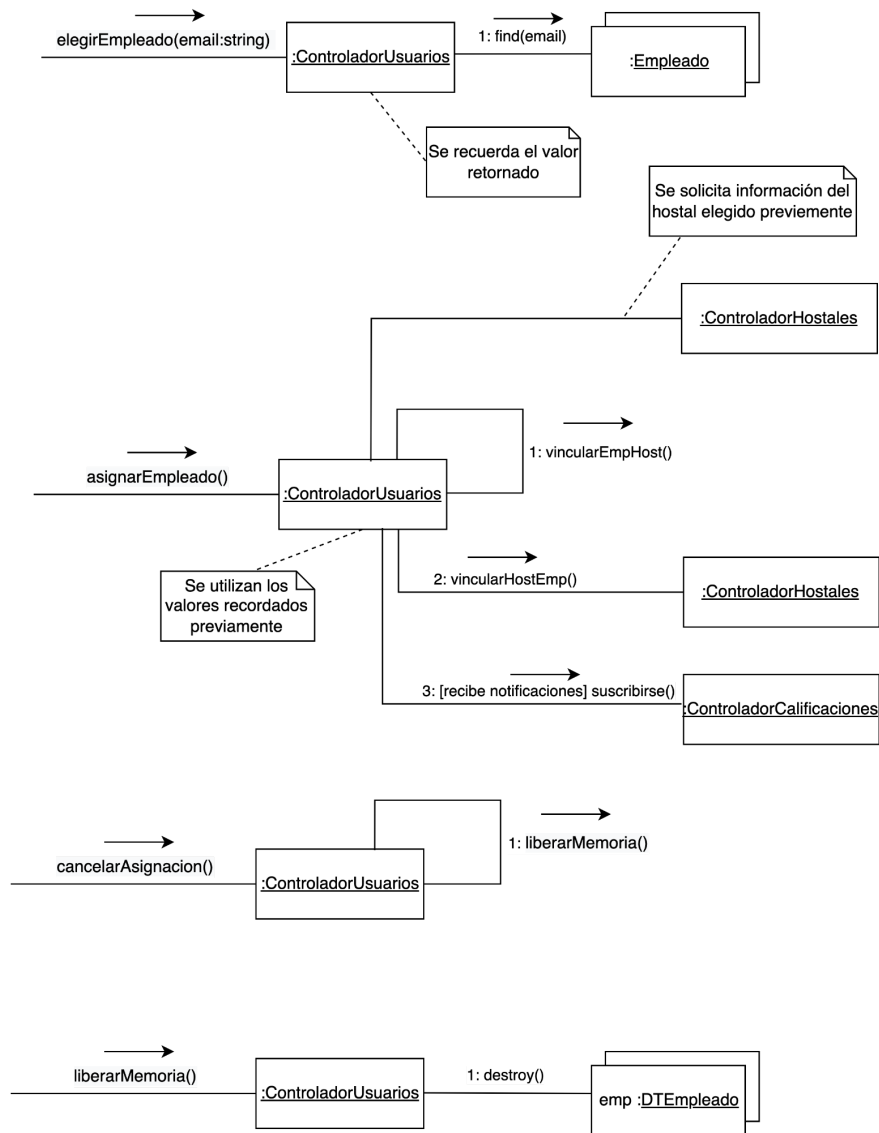
elegirEmpleado

asignarEmpleado

cancelarAsignacion

liberarMemoria





5. Realizar Reserva

Operaciones:

listarHostales

elegirHostal

ingresarDatosReserva

listarHabitaciones

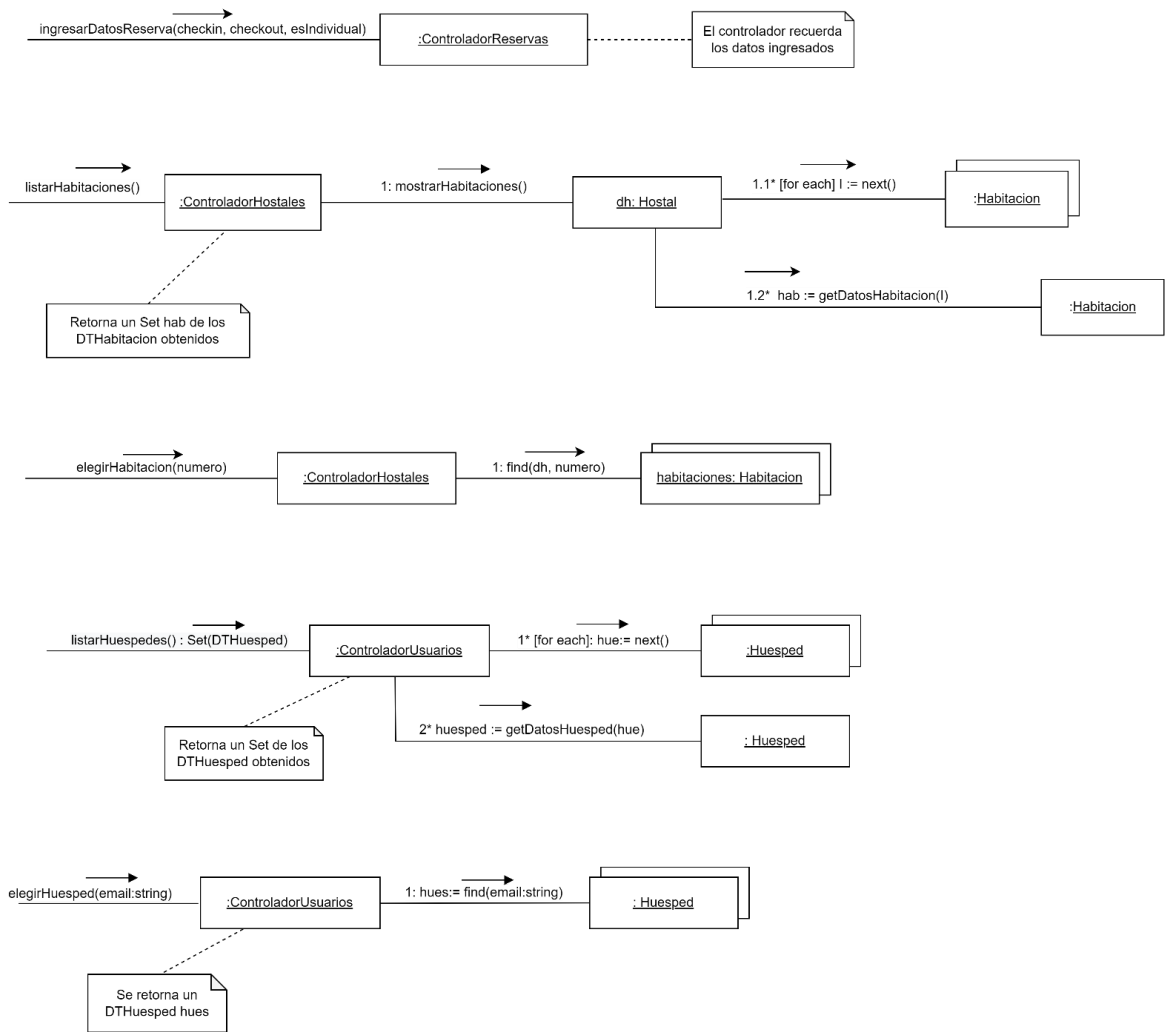
elegirHabitacion

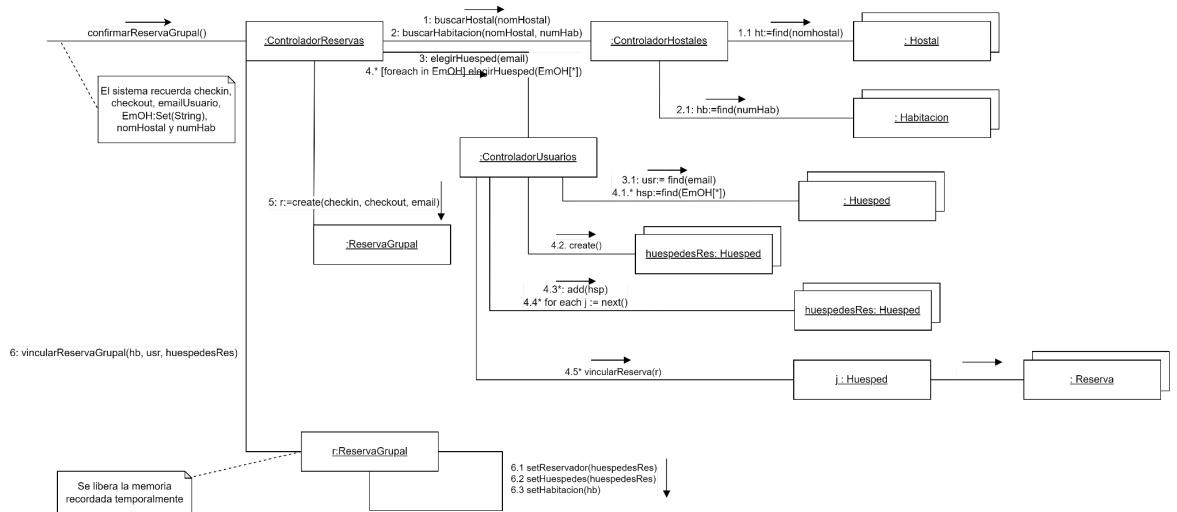
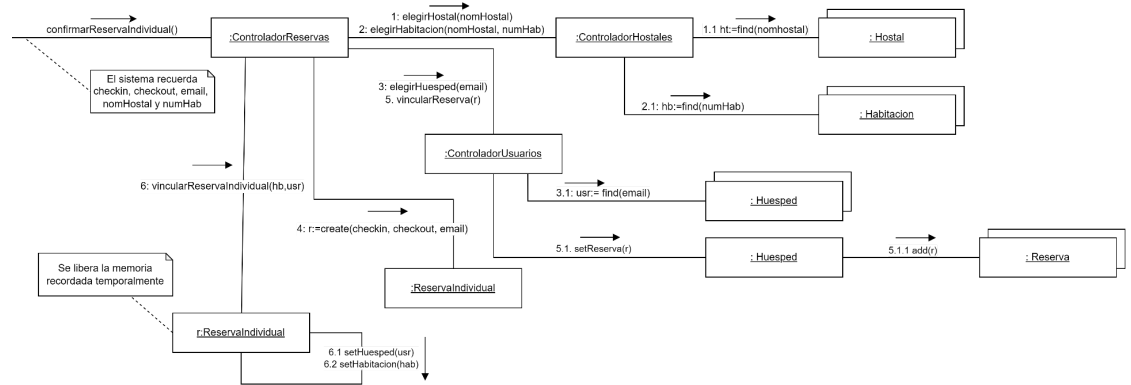
listarHuespedes

elegirHuesped

confirmarReservaIndividual

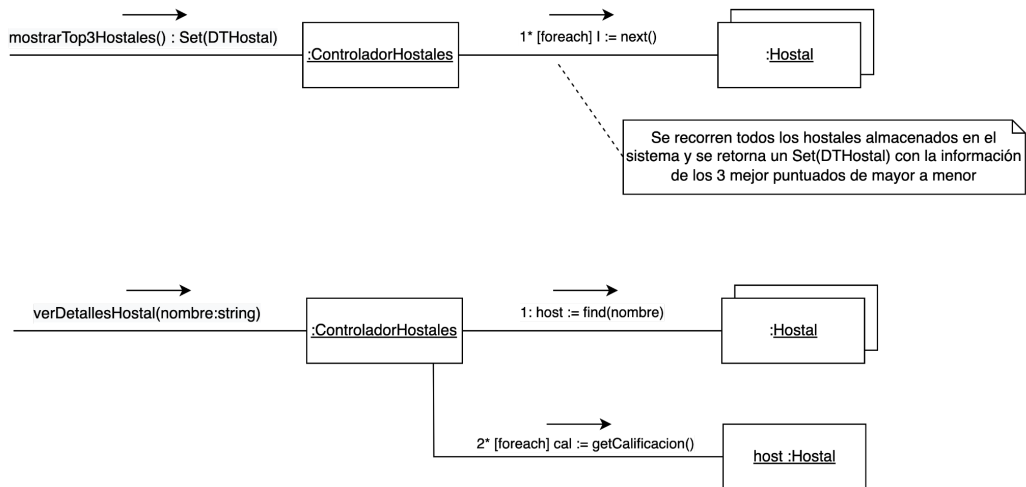
confirmarReservaGrupal





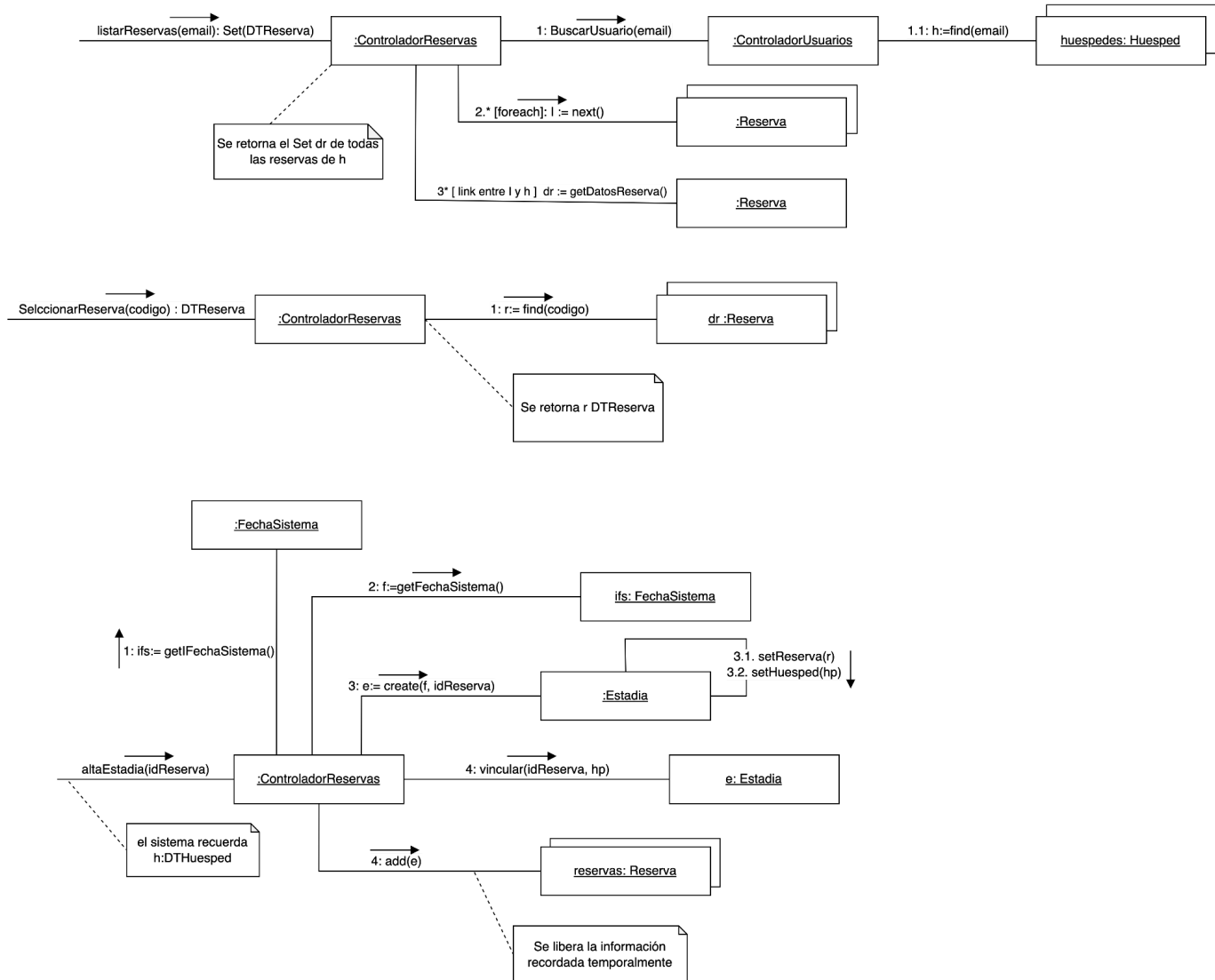
6. Consulta top 3 hostales

Operaciones:
 mostrarTop3Hostales
 verDetallesHostal



7. Registrar estadía

Operaciones:
 listarReservas
 seleccionarReserva
 altaEstadia



8. Finalizar estadia

Operaciones:

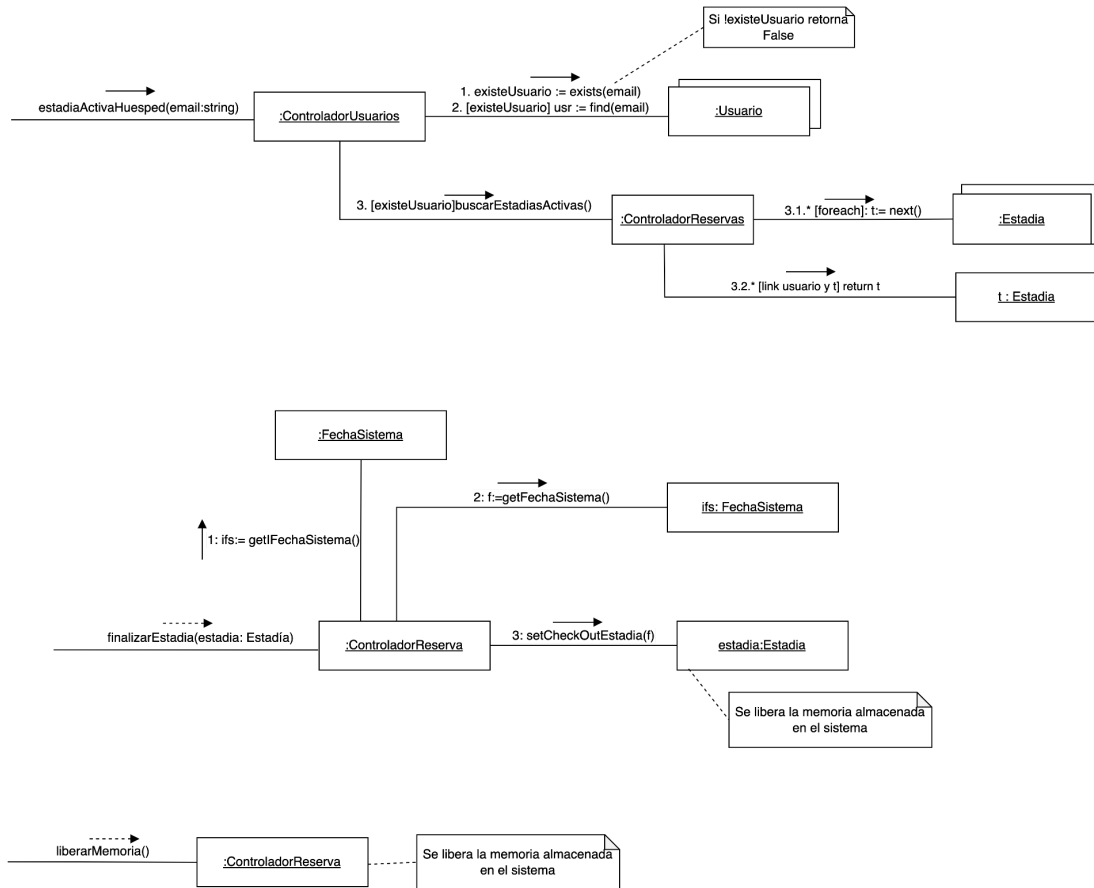
listarHostales

elegirHostal

estadiaActivaHuesped

finalizarEstadia

liberarMemoria



9. Calificar estadía

Operaciones:

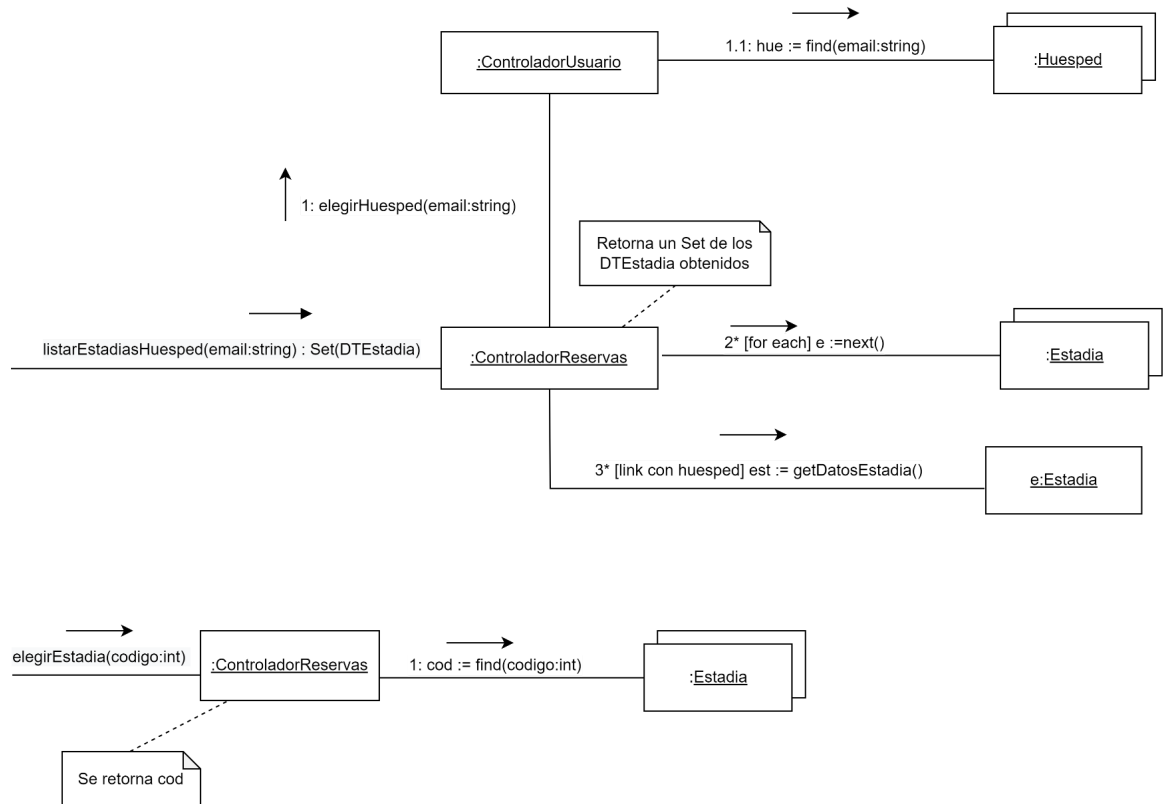
listarHostales

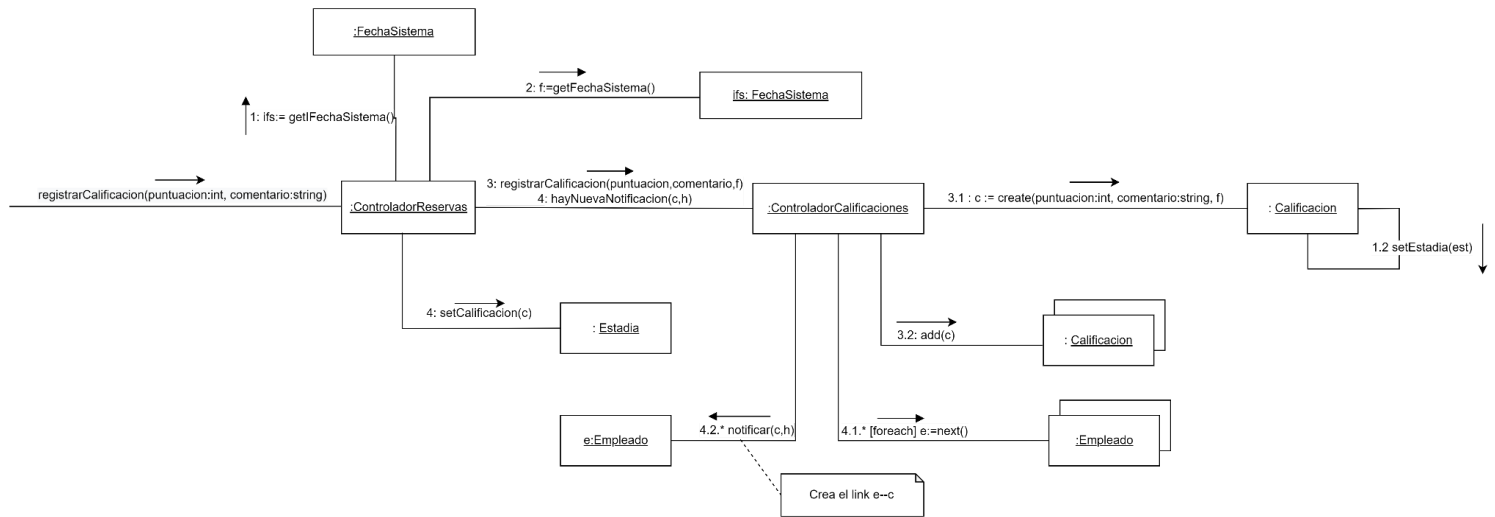
elegirHostal

listarEstadiasHuesped

elegirEstadia

registrarCalificacion





10. Comentar calificación

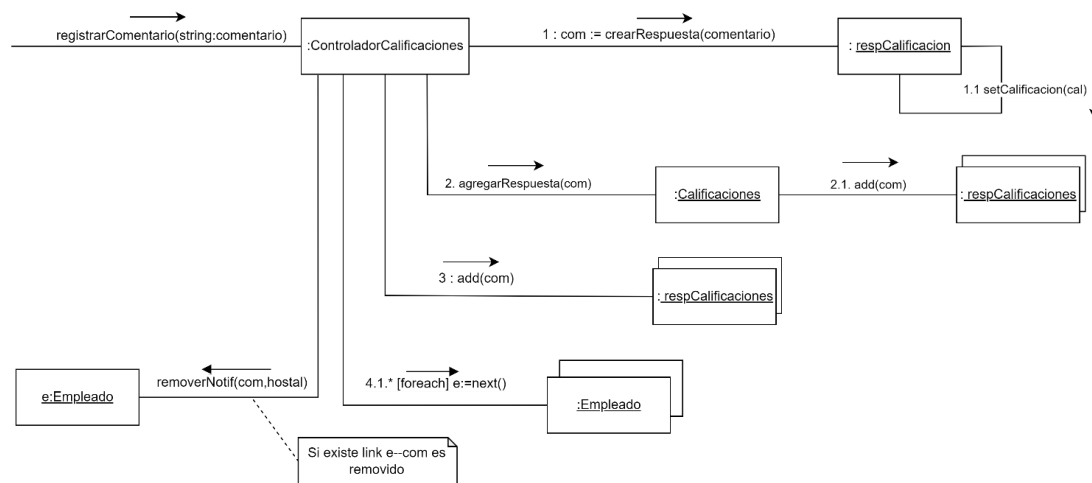
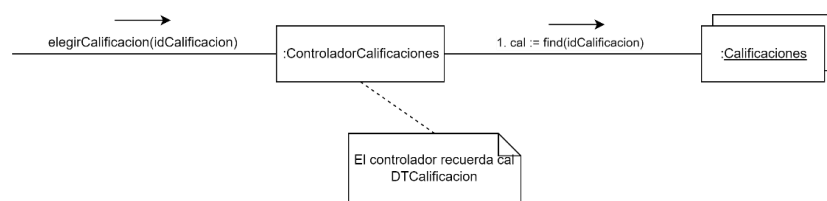
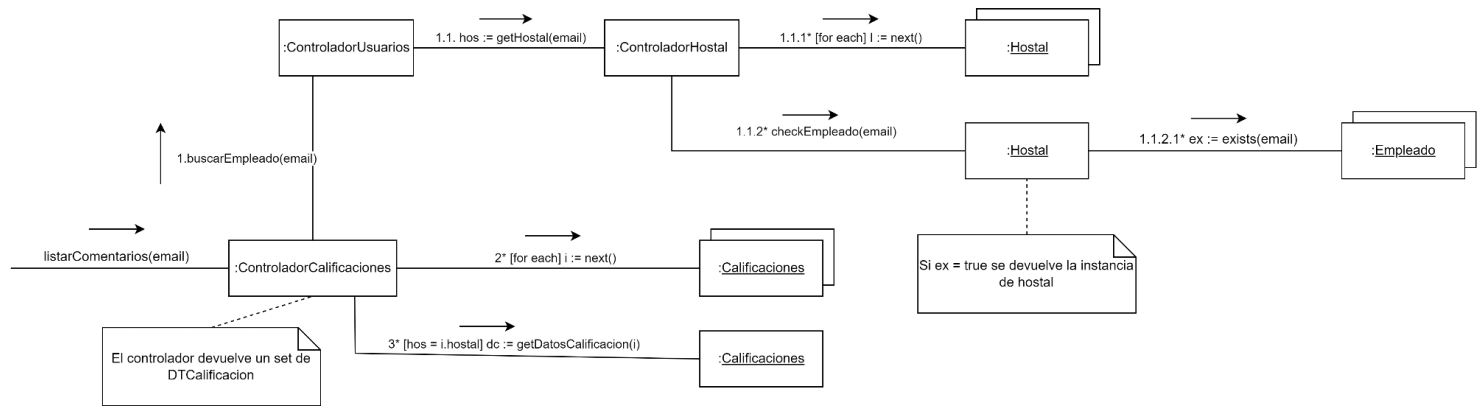
Operaciones:

listarComentarios

elegirCalificacion

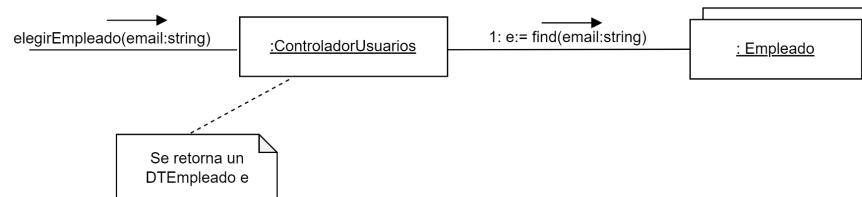
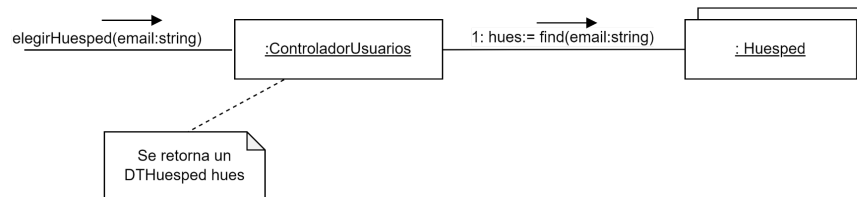
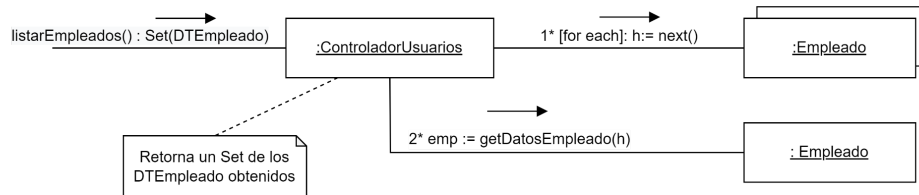
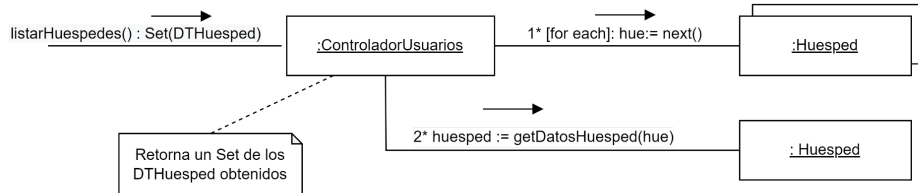
registrarComentario

liberarMemoriaComentar



11. Consulta de usuario

Operaciones:
 listarHuespedes
 listarEmpleados
 elegirHuesped
 elegirEmpleado



12. Consulta de hostel

Operaciones:

listarHostales

elegirHostal

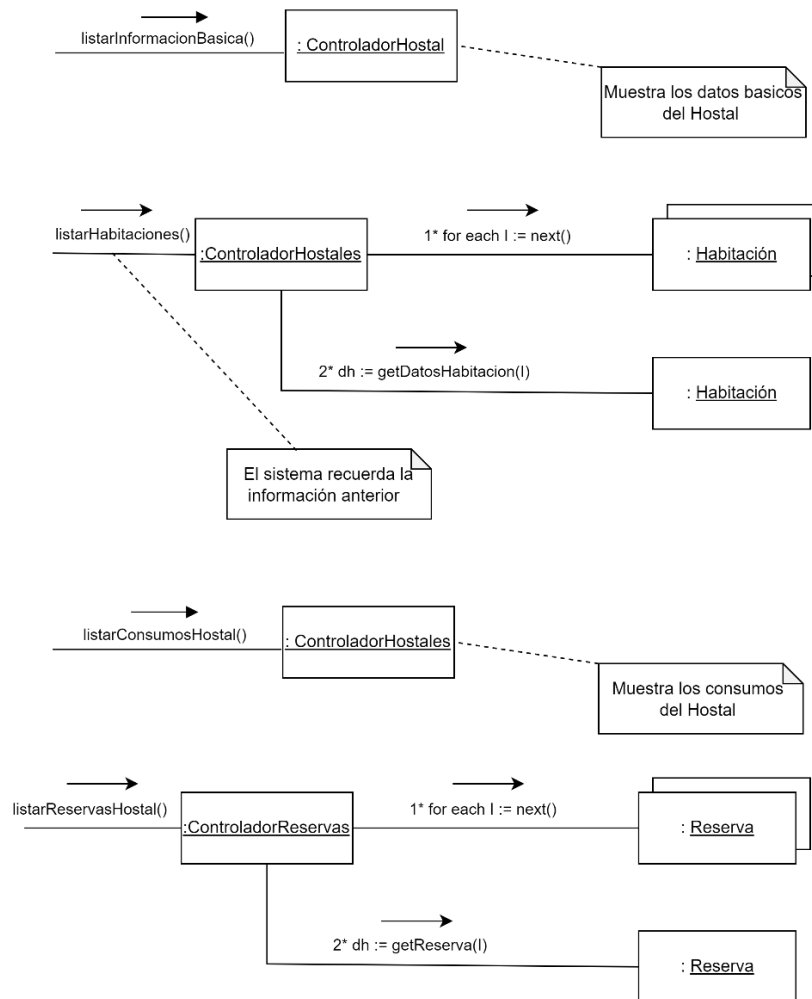
listarInformacionBasica

listarHabitaciones

listarConsumos

listarReservas

liberarMemoria



13. Consulta de reserva

Operaciones:

listarHostales

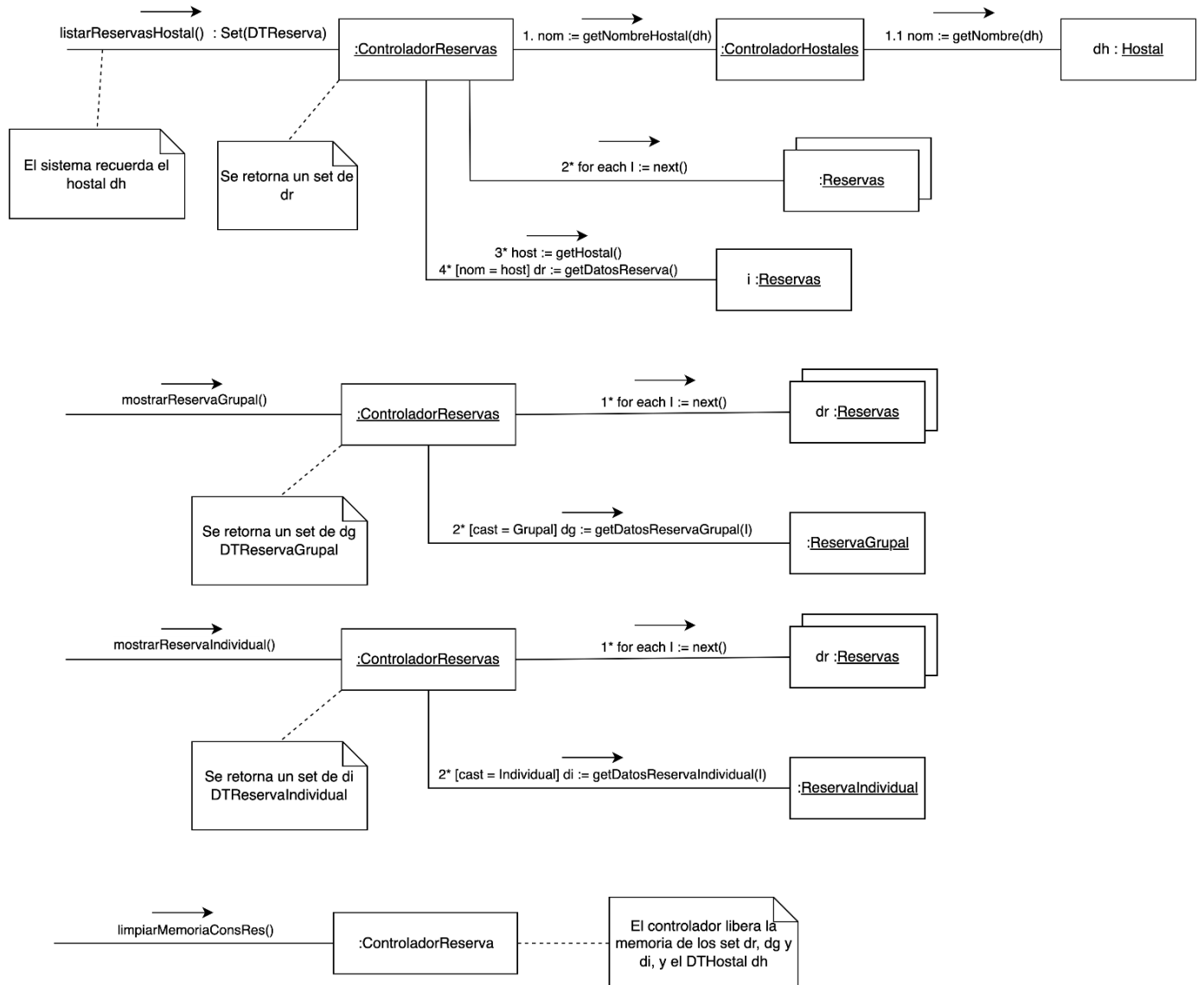
elegirHostal

listarReservasHostal

mostrarReservaIndividual

mostrarReservaGrupal

liberarMemoriaConsRes



14. Consulta de estadia

Operaciones:

listarHostales

elegirHostal

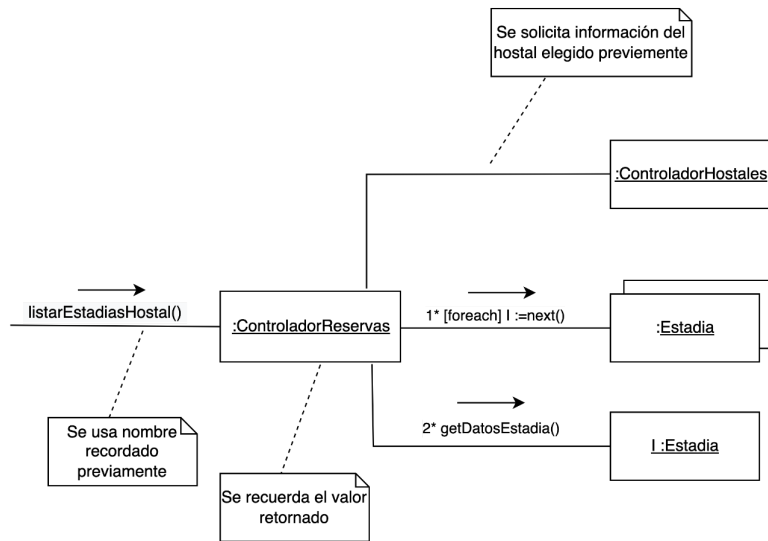
listarEstadias

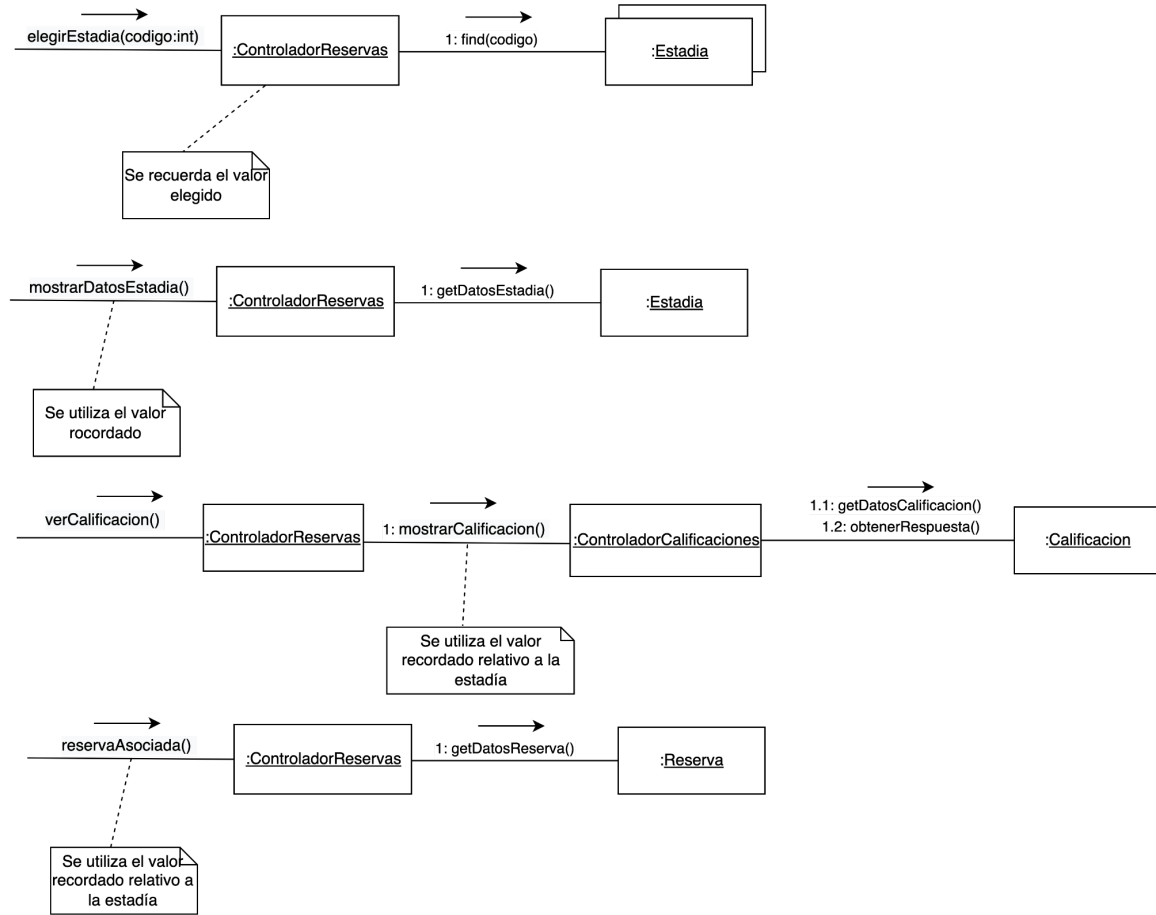
elegirEstadia

mostrarDatosEstadia

verCalificacion

reservaAsociada





15. Baja de reserva

Operaciones:

listarHostales

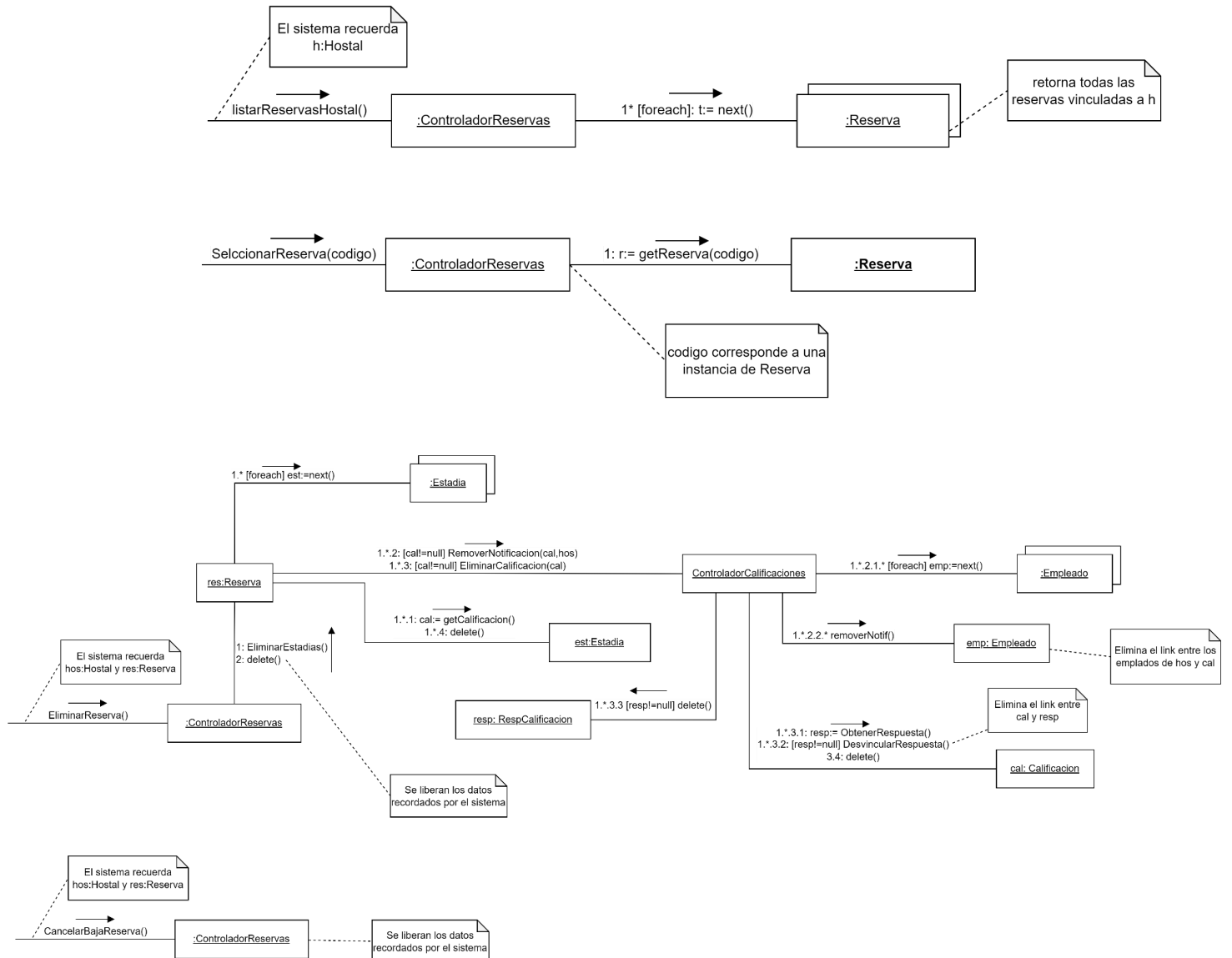
elegirHostal

listarReservasHostal

seleccionarReserva

eliminarReserva

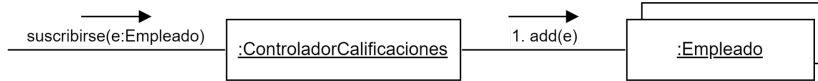
cancelarBajaReserva



16. Suscribirse a notificaciones

Operaciones:

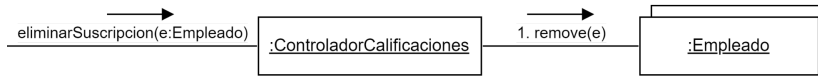
suscribirse



17. Consulta de notificaciones

Operaciones:

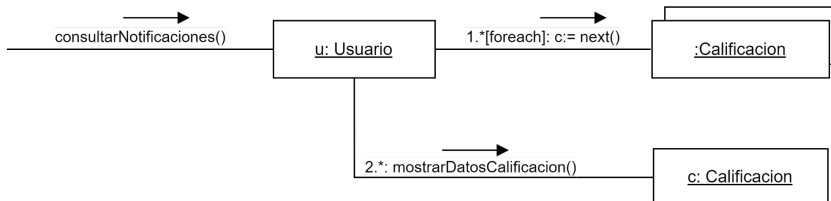
eliminarSuscripcion



18. Eliminar suscripción

Operaciones:

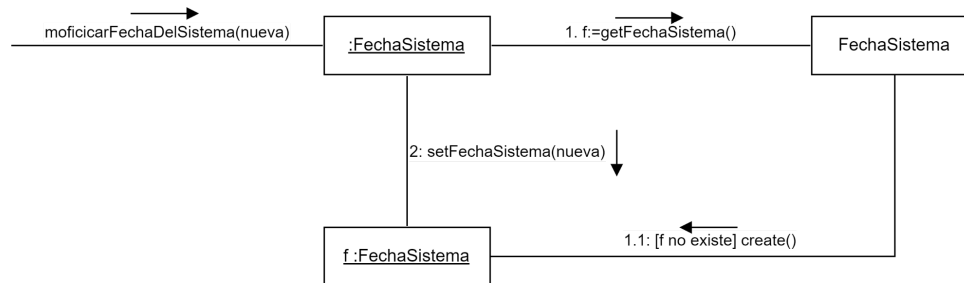
consultarNotificaciones



19. Modificar fecha de sistema

Operaciones:

modificarFechaDelSistema



Criterios Generales

Para comenzar se decidió crear controladores los cuales manejan cierto tipo de clases que tuvieran relaciones en común. Teniendo esto en cuenta construimos 4 controladores de esta forma: controladorUsuarios, controladorHostal, controladorReserva, controladorCalificacion. El pilar principal fue tener controladores que manejan ciertos tipos de clases en vez de funcionalidades. Cada controlador se implementó siguiendo un patrón Singleton para que estos sean globales. Las responsabilidades del manejo de las colecciones se asignó de la siguiente manera:

- ControladorUsuarios: Set(Usuario)
- ControladorHostales: Set(Hostal), Set(Habitación), Set(Consumo)
- ControladorCalificaciones: Set(Calificacion), Set(RespCalificacion)
- ControladorReservas: Set(Reserva), Set(Estadia)

Por otro lado se requería hacer un seguimiento de las calificaciones una vez eran ingresadas y respondidas por lo que se implementó en forma de notificación una forma de cada vez que se ingresaba una calificación nueva se le notifica a los empleados que deseen. Para esto se utilizó Observer.

Finalmente se hizo uso de Fábricas para crear interfaces que desacoplan los controladores de la capa de presentación que se espera luego.

En general la responsabilidad de retener la información necesaria para posteriores usos se la asignaba al controlador que estaba manejando dicho caso de uso. Por ejemplo: en Realizar Reserva en el que controladorReserva maneja los datos de hostal, habitación y usuario para lograr crear la reserva. A su vez la

creación y destrucción de instancias de los objetos se le designaba a su controlador correspondiente.

En cuanto a el uso de GRASP se hizo uso de todos ellos en diferentes contextos. Para los casos de alta, como usuario, hostel y habitación, se tuvieron en cuenta expert, creator y controller. Esto para que la responsabilidad de tener la información necesaria esté en el controlador que maneja el caso y la creación de las instancias está en el controlador. Los casos de consulta, reserva, estadía, usuario y hostel, funcionan de forma similar en donde se derivan el trabajo entre controladores para que cada cual maneja su información necesaria y no sobrecargar a quien lo esté llevando a cabo.

Luego para baja de reserva se implemento similar a los alta pero ahora la responsabilidad cambia de crear a destruir objetos y esta misma queda en el propio controlador quien tiene la información para saber cuál destruir y concluye la acción.

Lo que tuvo más impacto a la hora de hacer los diagramas fue la alta cohesión, bajo acoplamiento y no hables con extraños. Estos fueron un punto fuerte de apoyo a la hora de dividir el trabajo entre controladores y no tener clases que hablen con controladores que no corresponden.

Nuevos requerimientos y su impacto

A la hora de implementar las notificaciones y la posibilidad de modificar la fecha de sistema se debieron agregar estos a diagramas ya hechos, por lo que los diagramas con cambios son:

- Asignar empleado a hostel: a la hora de asociar un empleado a hostel se debe preguntar si este quiere suscribirse a la notificaciones.
- Calificar estadía: cada vez que se crea un huésped hace una calificación se debe notificar a los empleados que estén suscriptos.
- Comentar calificación y baja de reserva: deben eliminar las notificaciones que los empleados tengan pendiente.
- Registrar estadía y finalizar estadía: como hacen uso de la fecha del sistema se debe preguntar si los usuarios desean cambiarla antes de crear la instancia.

Cambios que se vieron necesarios para llegar a la solución

Se hicieron cambios en el Modelo de dominio y en algunos DSS para lograr un diseño más cohesivo.

Modelo de Dominio:

- Se agregó idCalificacion, un id autogenerado para identificar las diferentes

calificaciones.

- En la clase Estadia se agregó el atributo código.
- En la clase Hostal agregó el atributo calculado /promedio.

DSS

- En ComentarCalificacion la funcion elegirComentario() ahora tiene como parametro idCalificacion, quedando elegirComentario(idCalificacion).

Modelo de Clase de Diseño

