

# Programación 4

## Informe del Modelo de Diseño - Diagramas de Comunicación -

### **Grupo 42** **Integrantes**

Cristián González Núñez  
Federico Marchese Medone  
Mathias Francisco Estévez  
Claudia Arrieta Martinez

CI: 4894107-6  
CI: 4516935-4  
CI: 5206103-2  
CI: 3451565-3

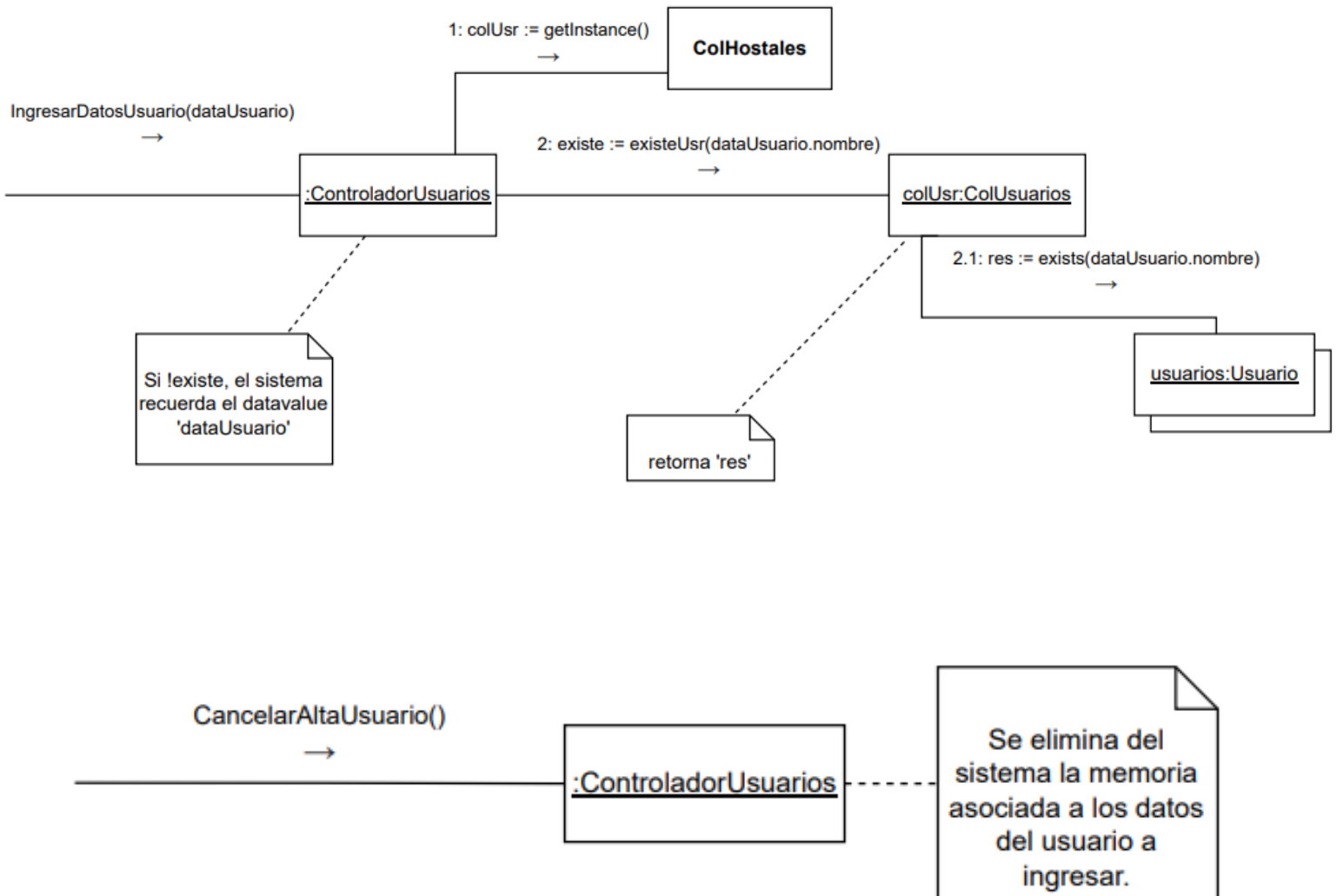
Docente:        Andrés Vasilev

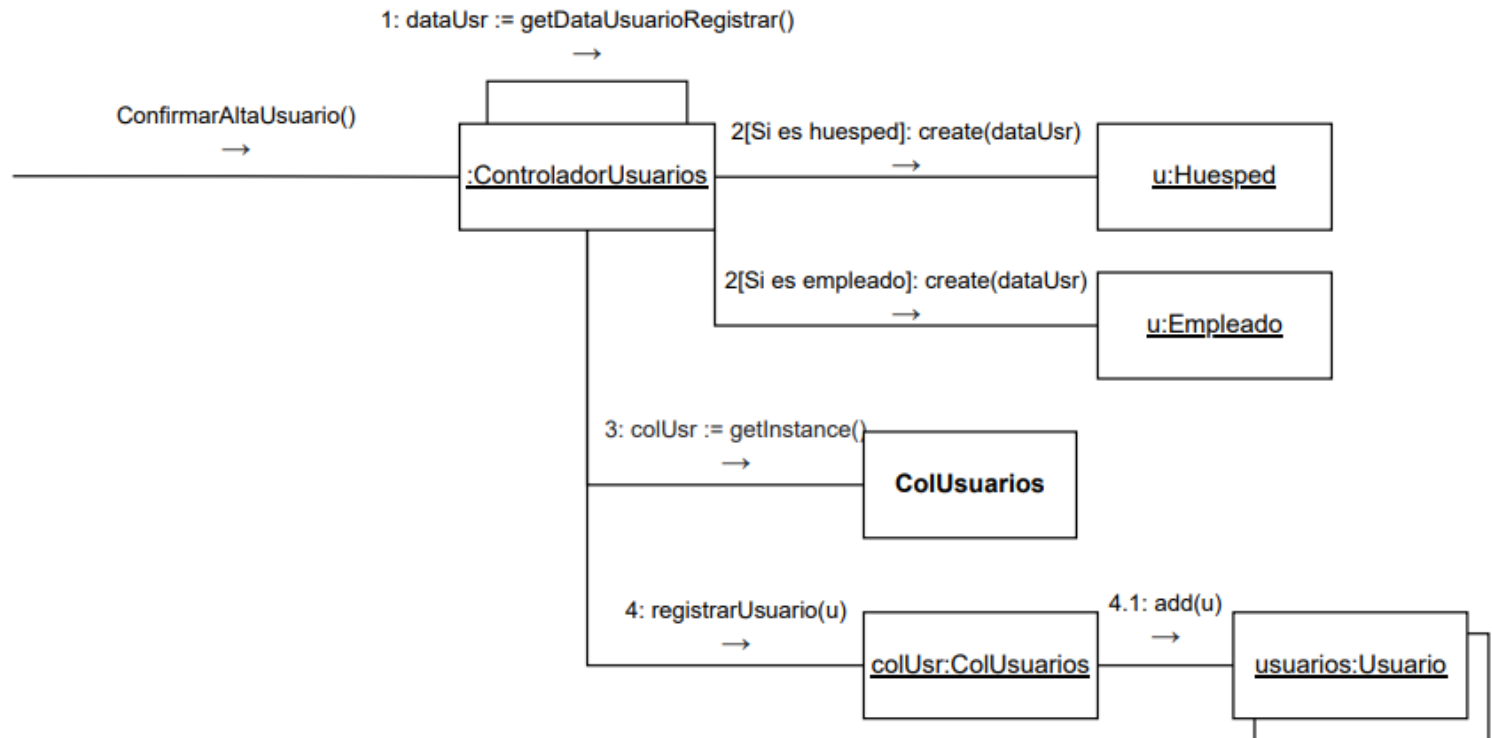
# 1 Realización de Casos de Uso

## 1.1. Interacciones

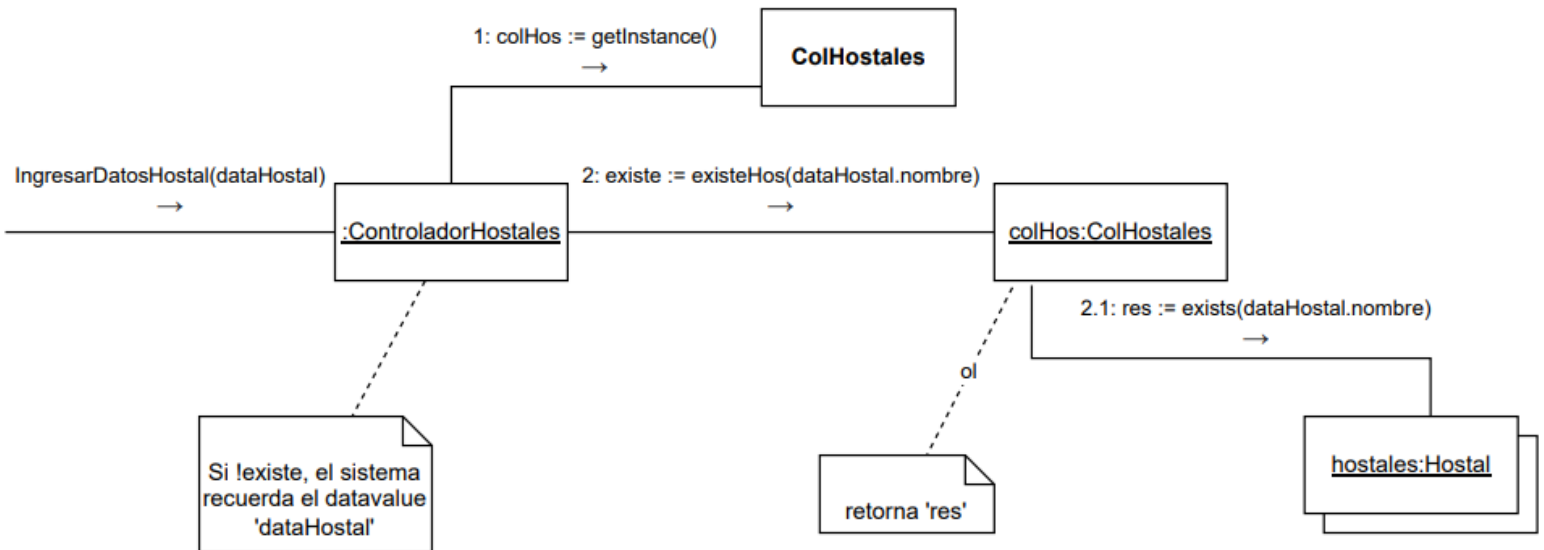
### Diagrama de comunicación

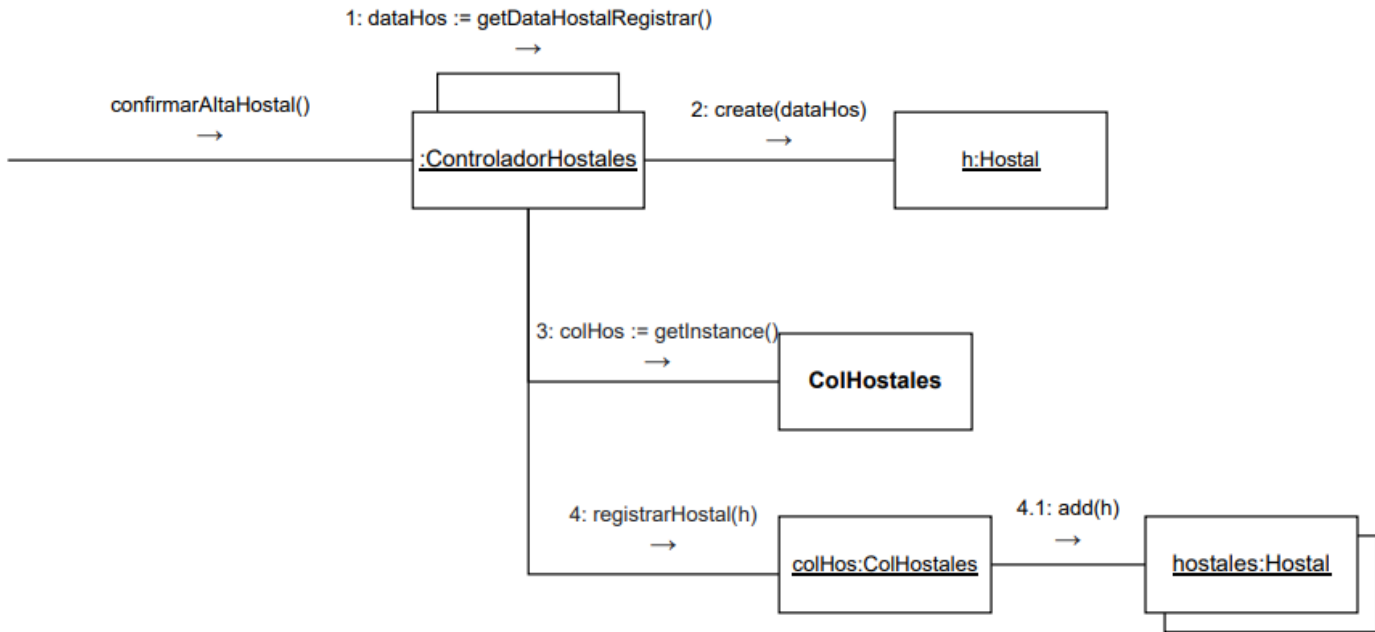
#### Alta de Usuario



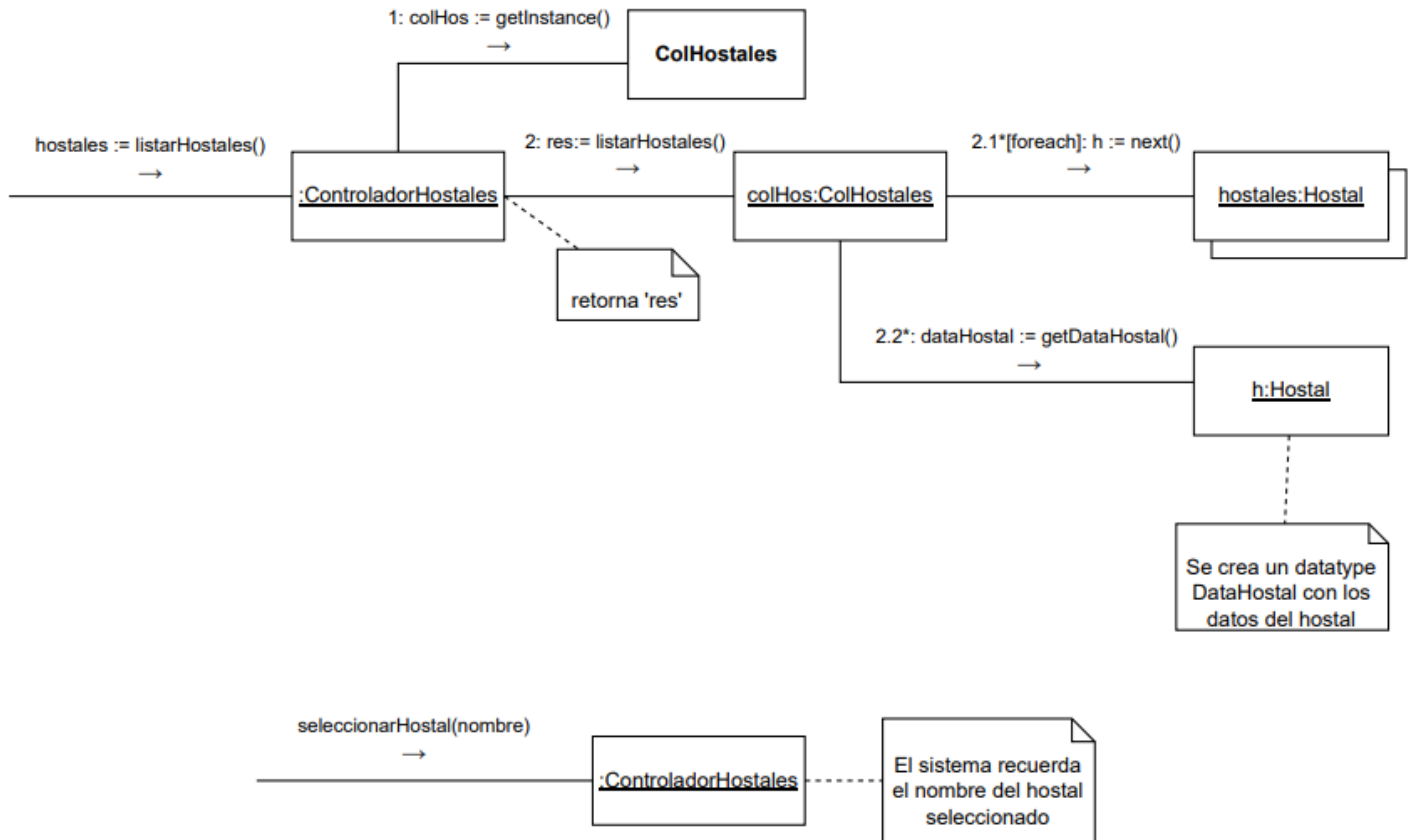


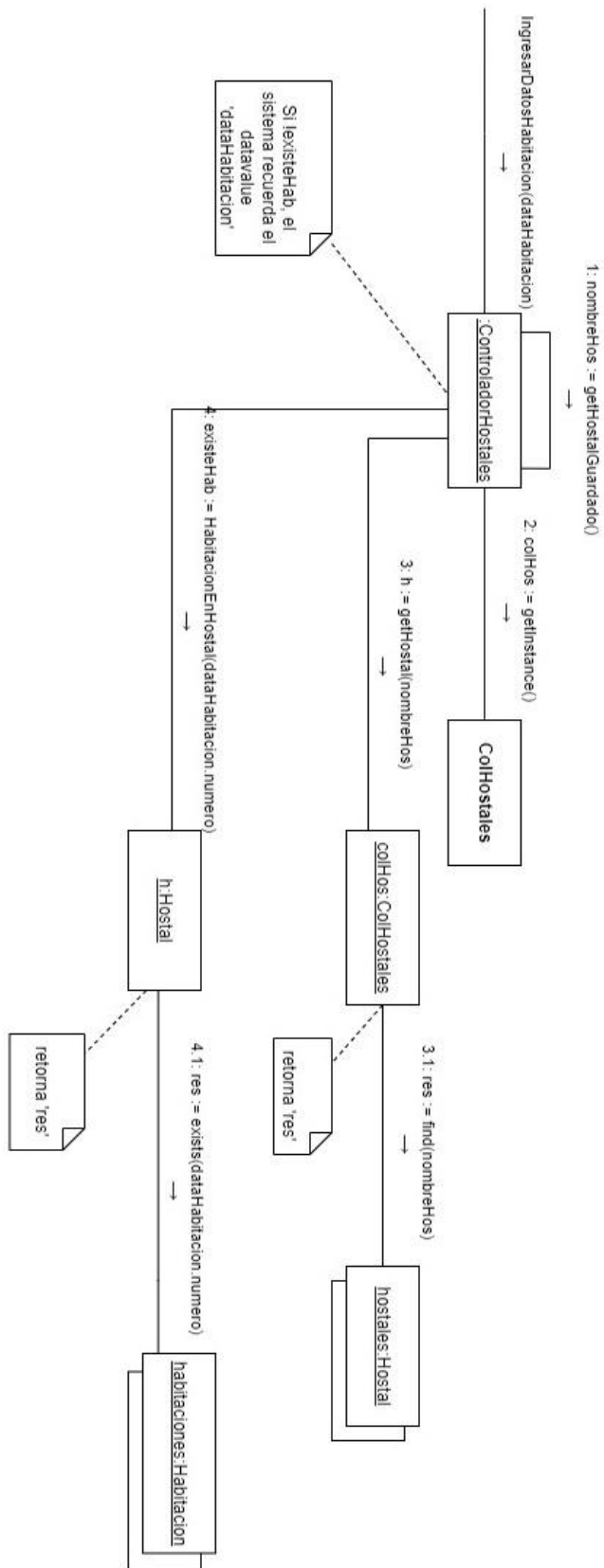
## Alta de Hostal

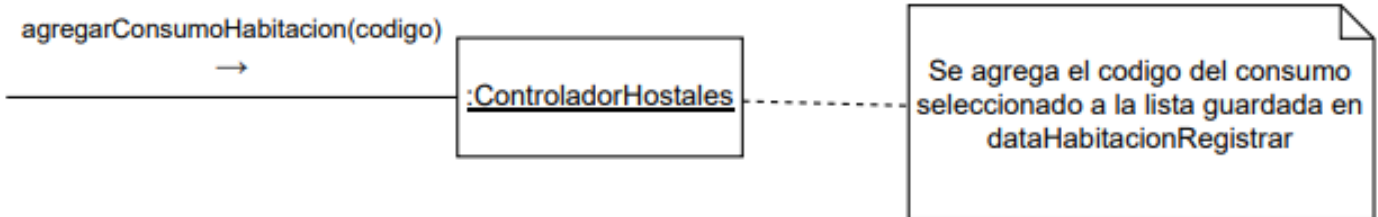
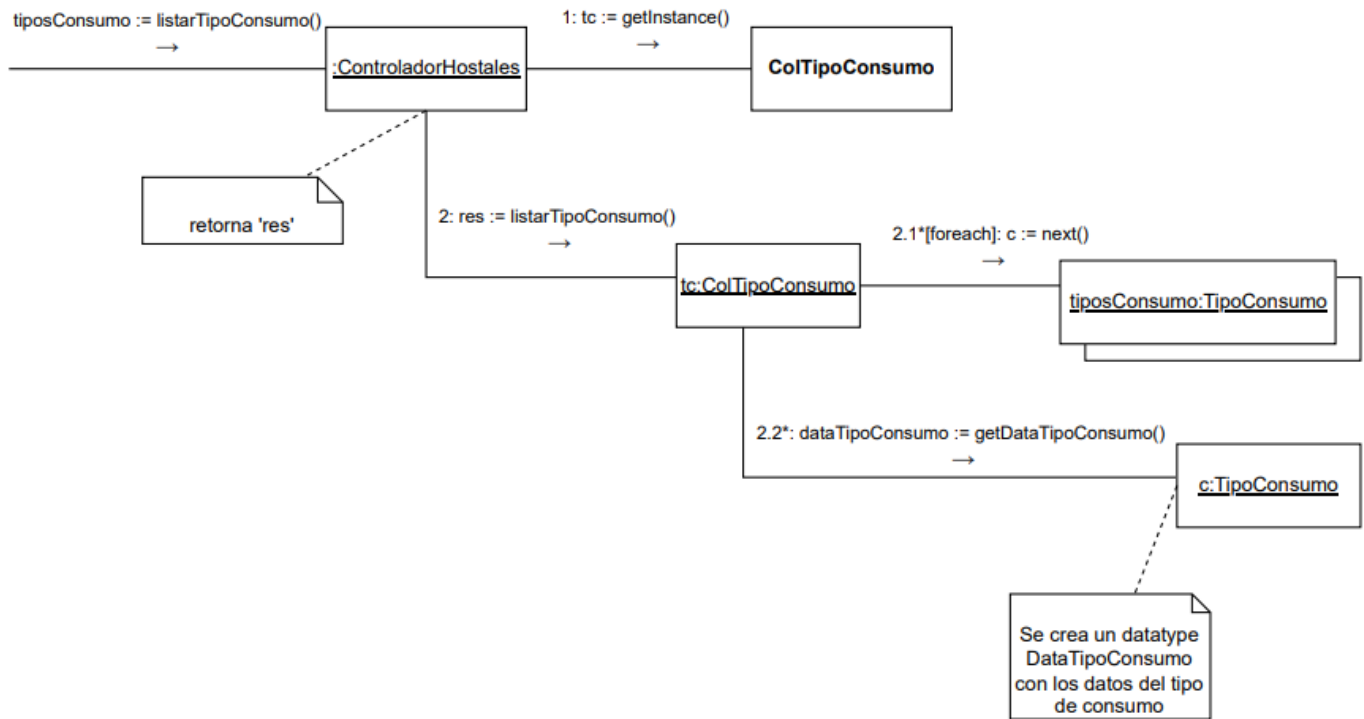




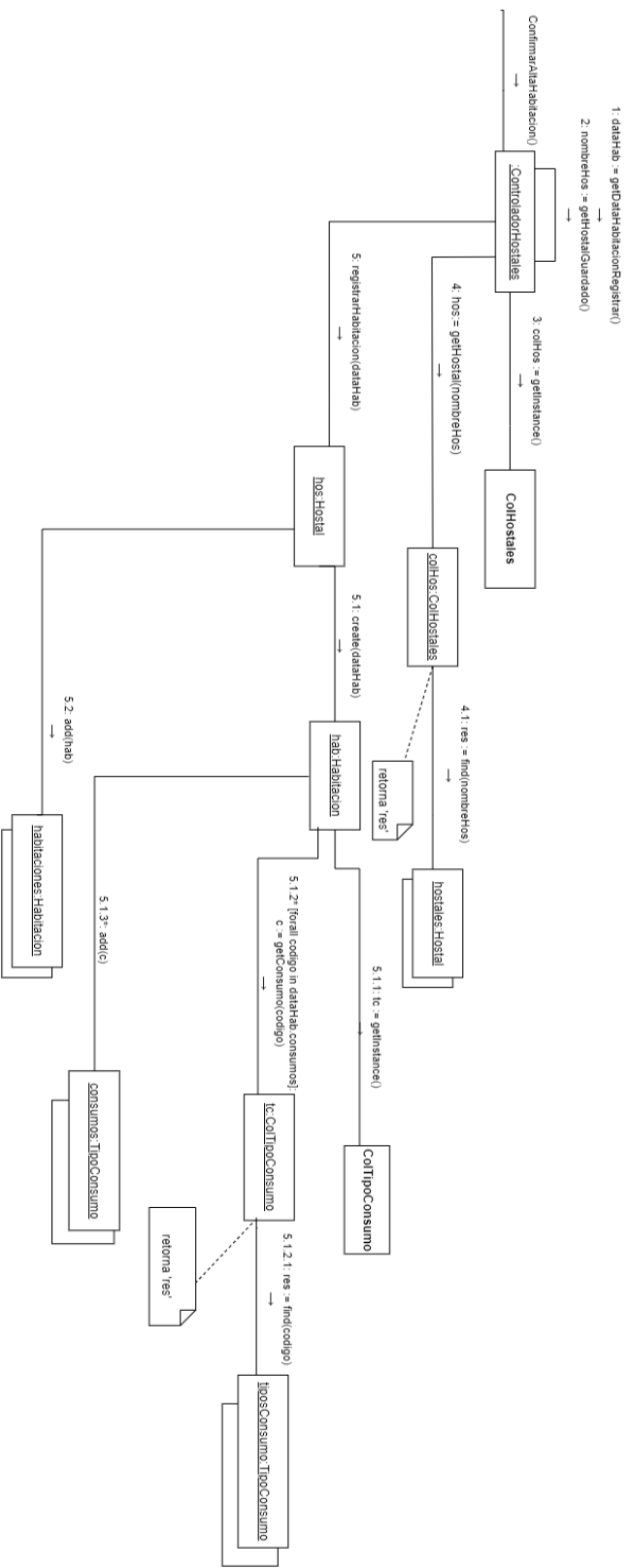
## Alta de Habitación





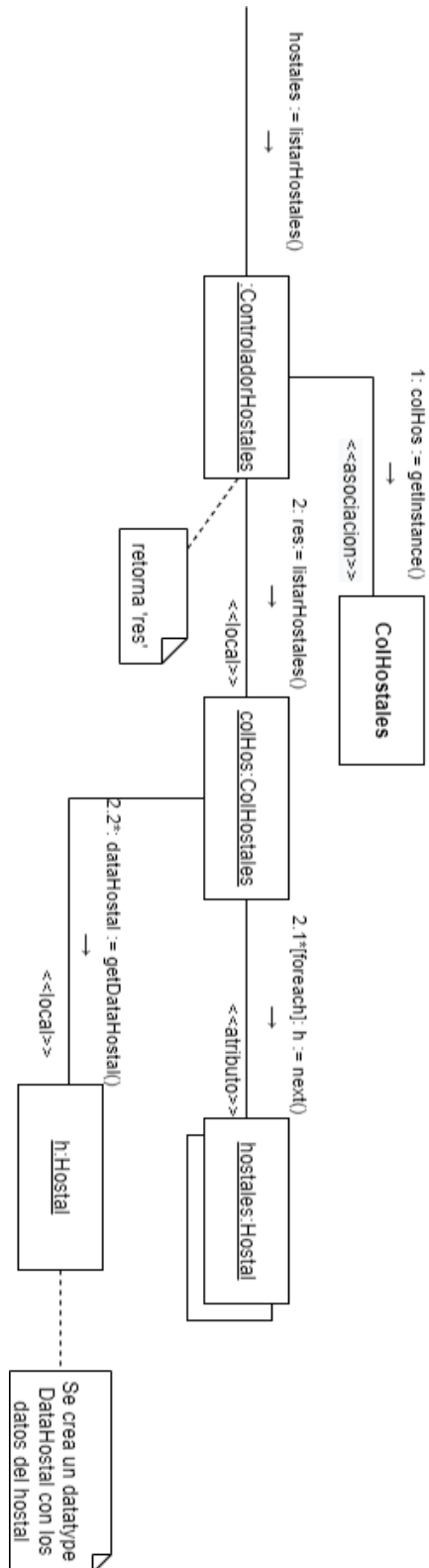


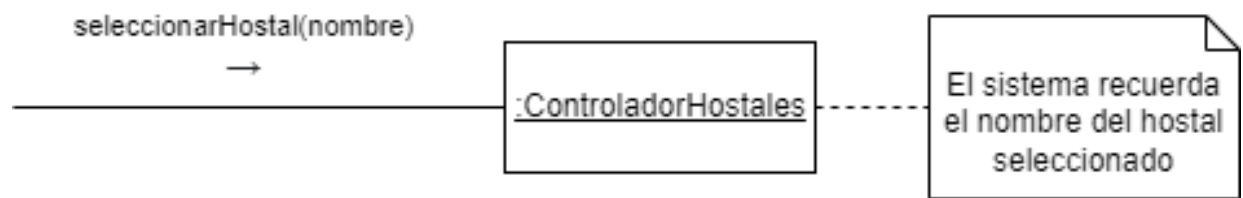


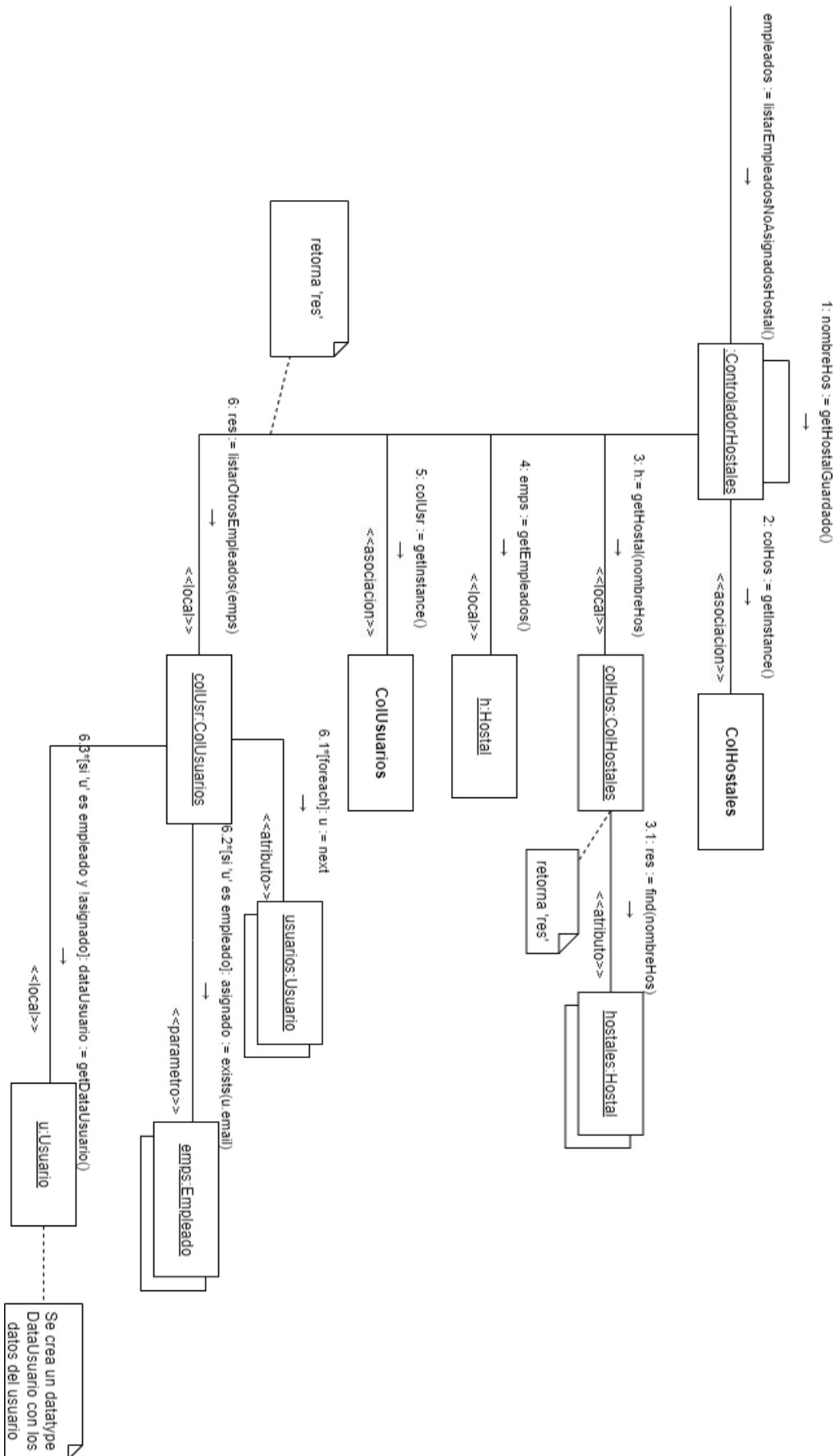


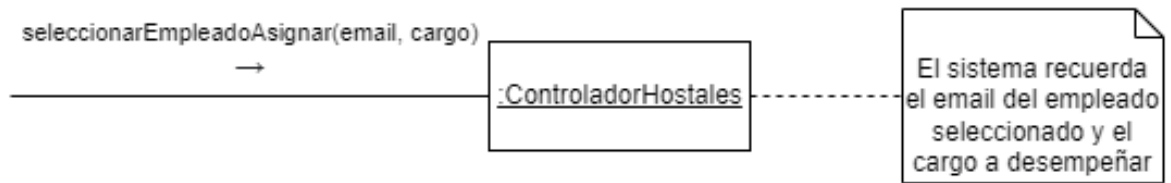


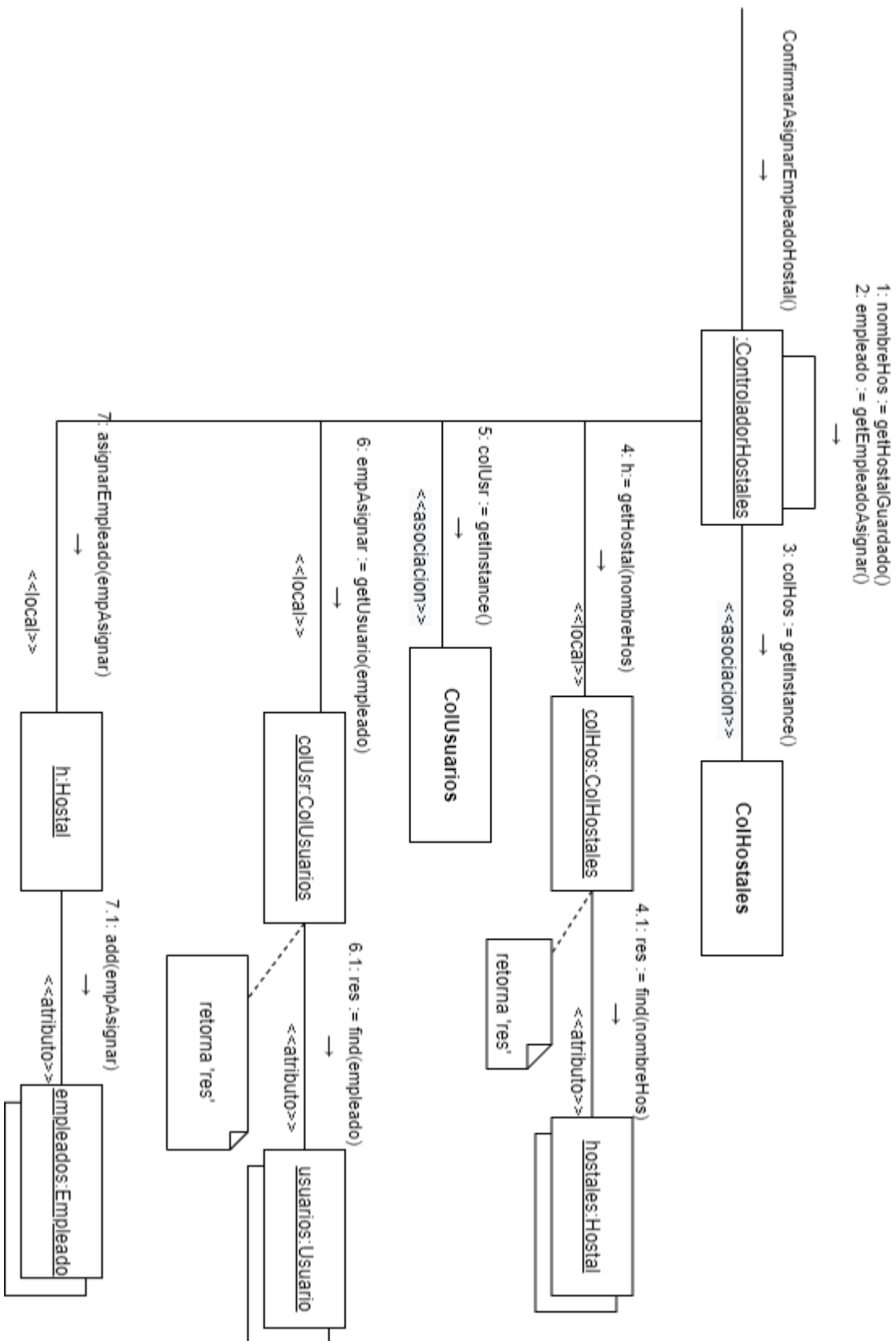
## Asignar empleado a hostel



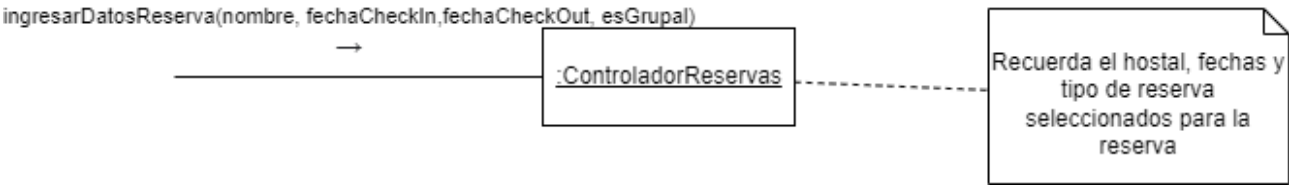


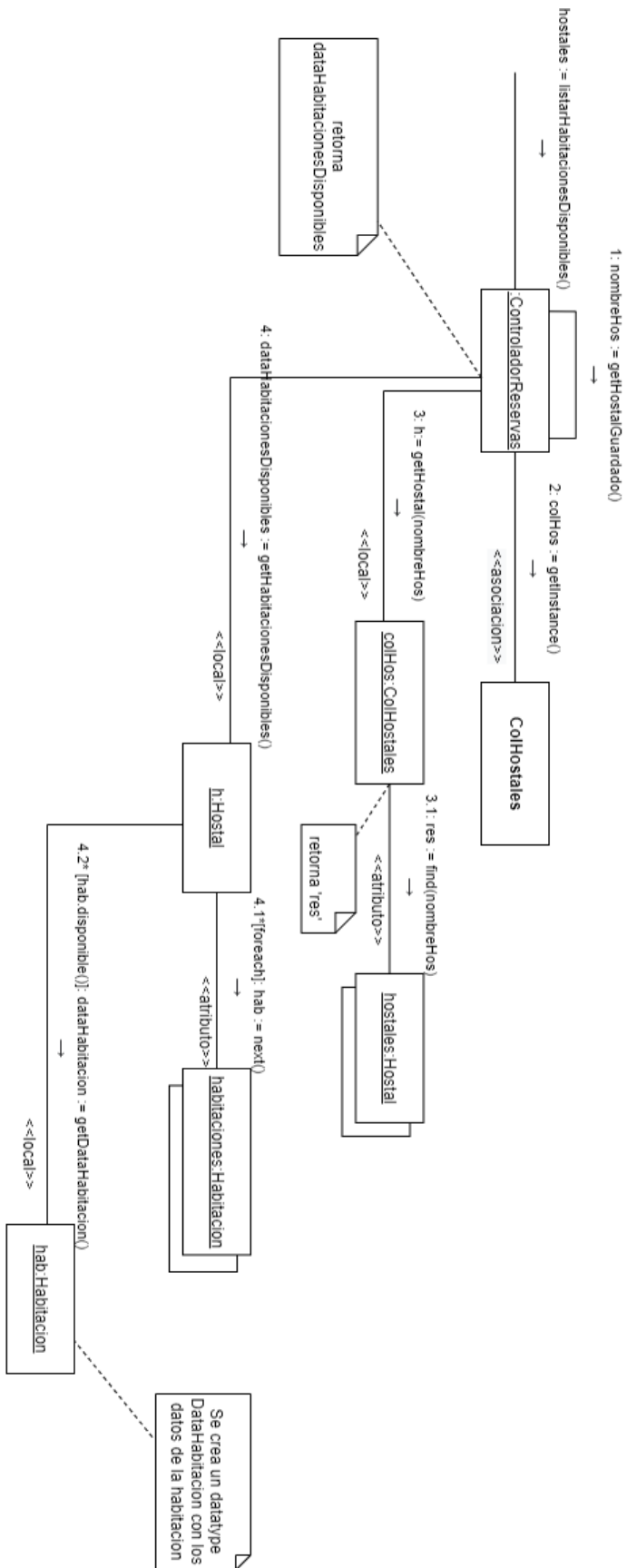


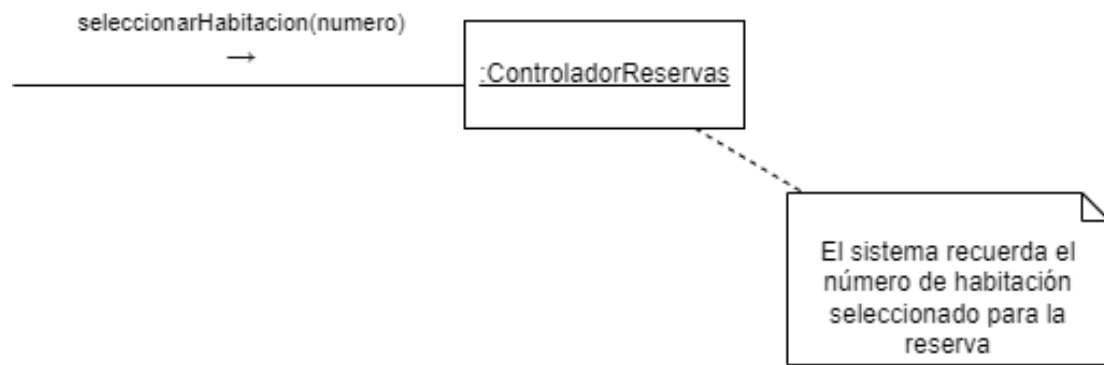




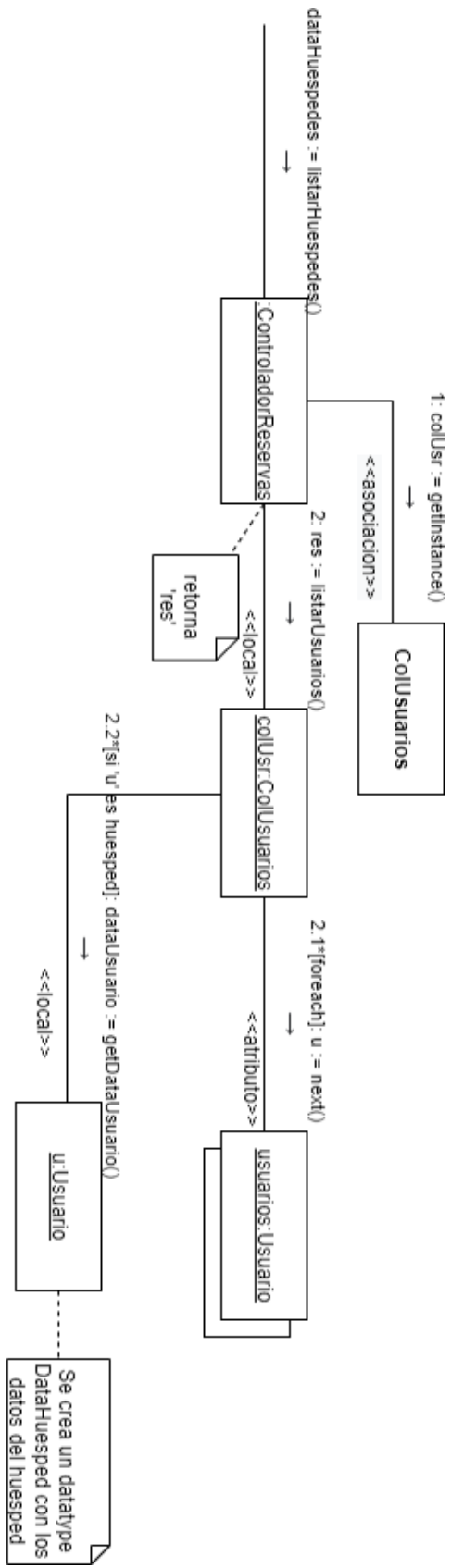
**Realizar Reserva**



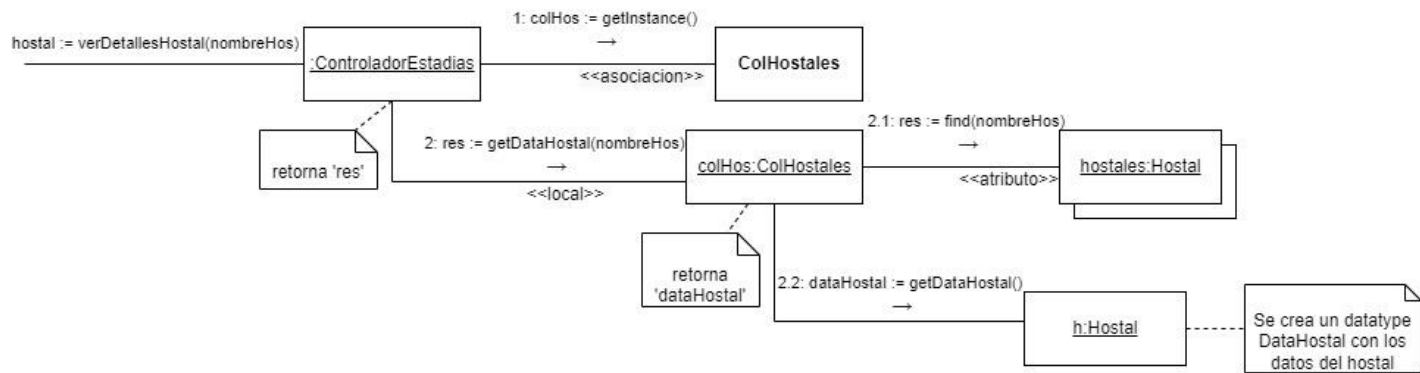
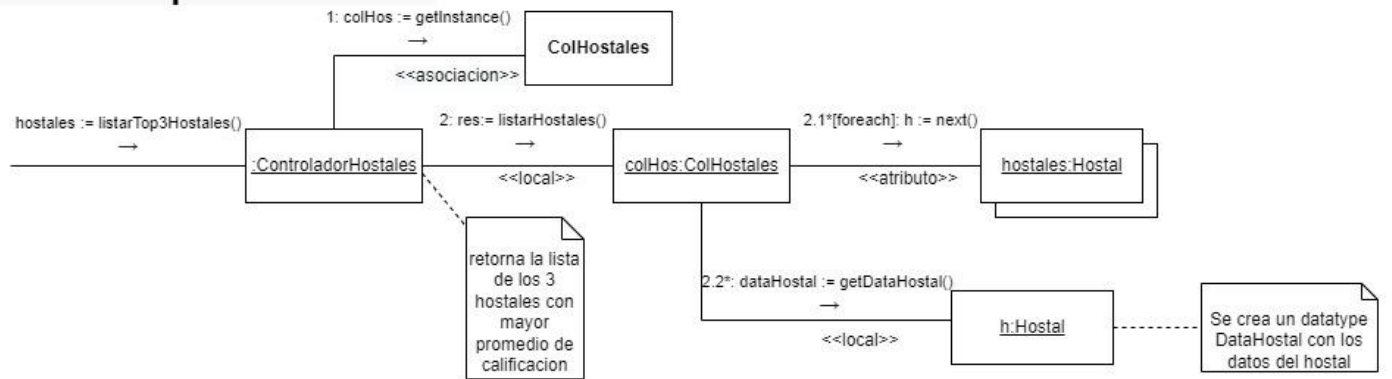


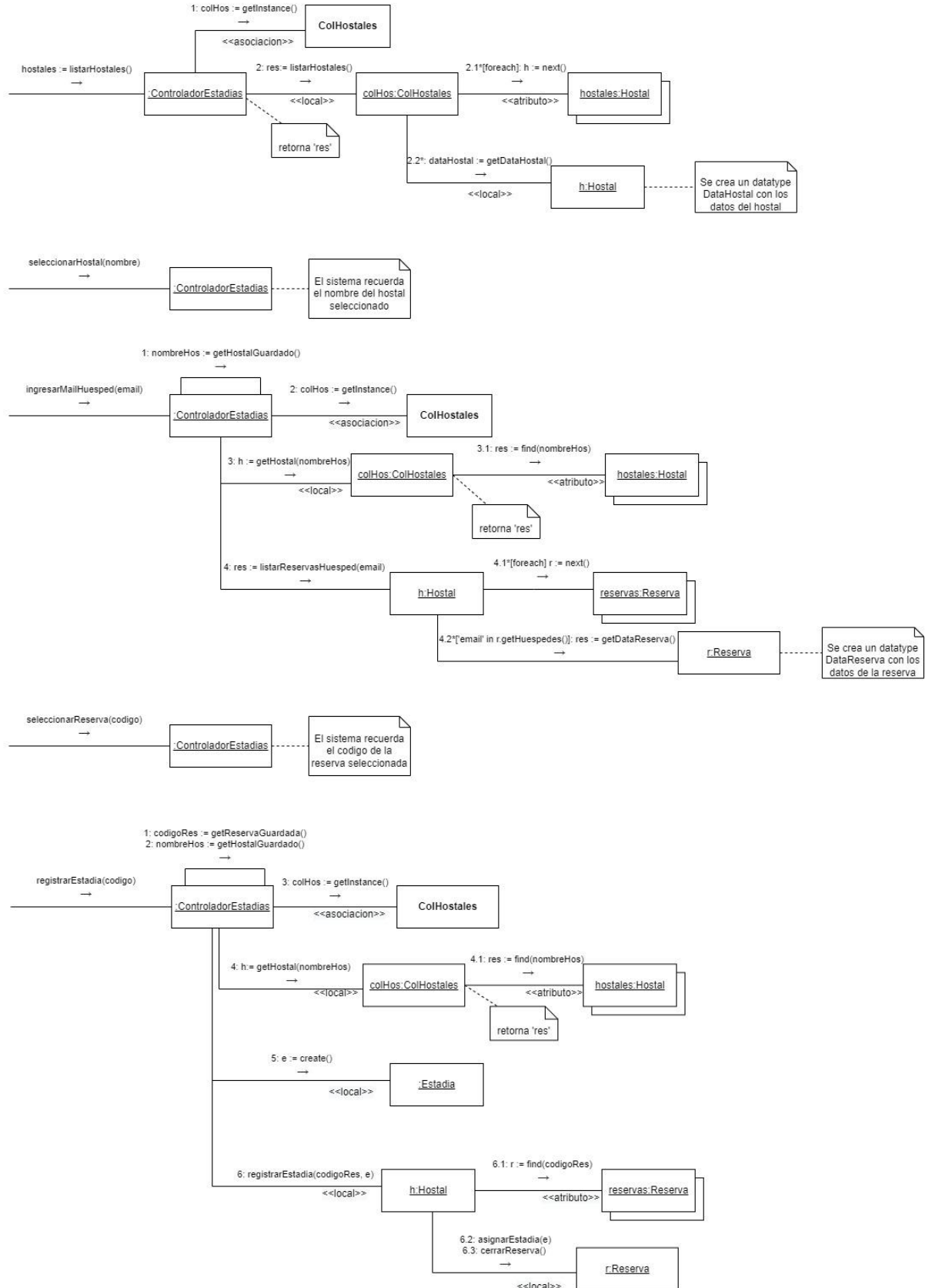




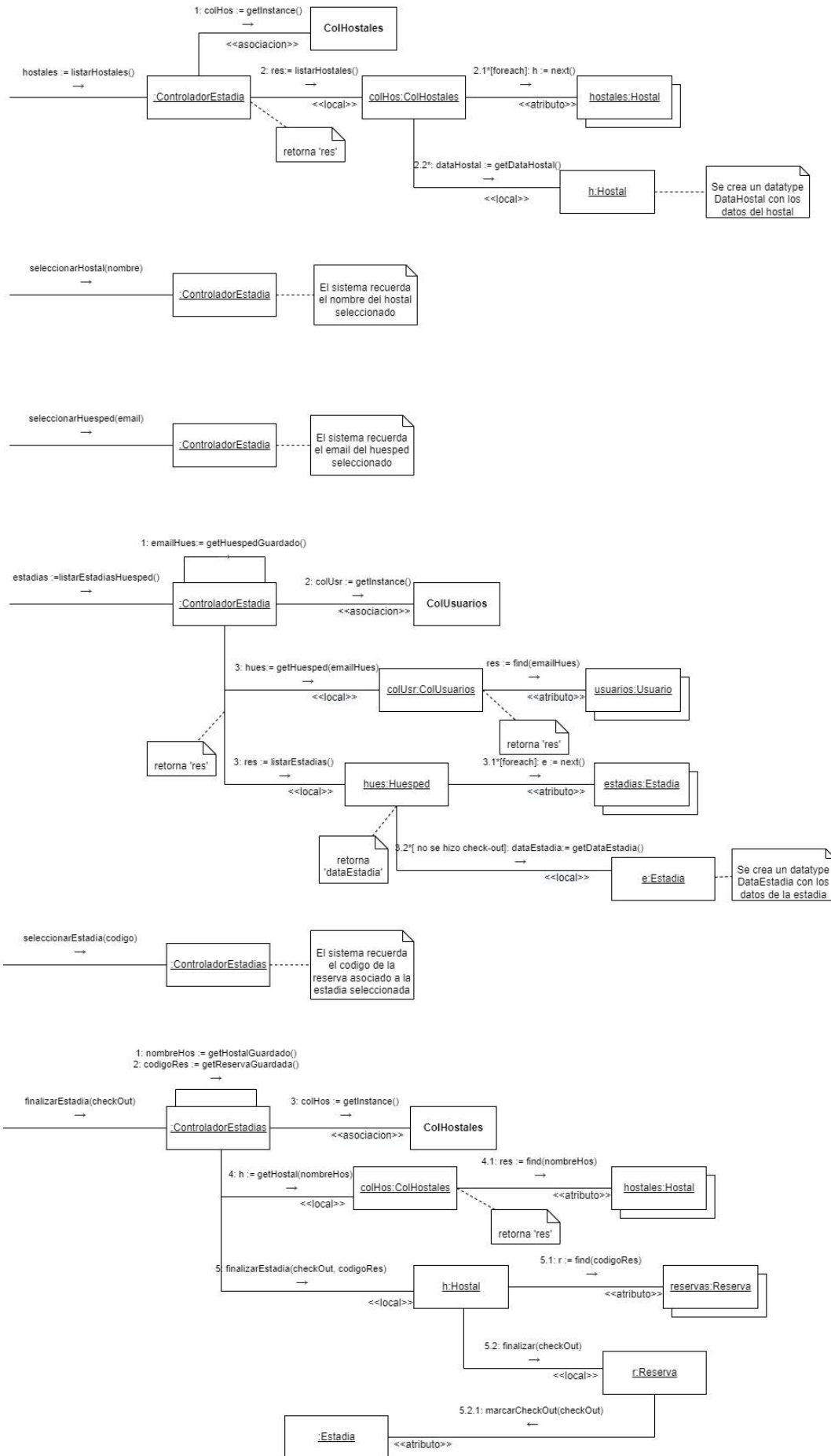


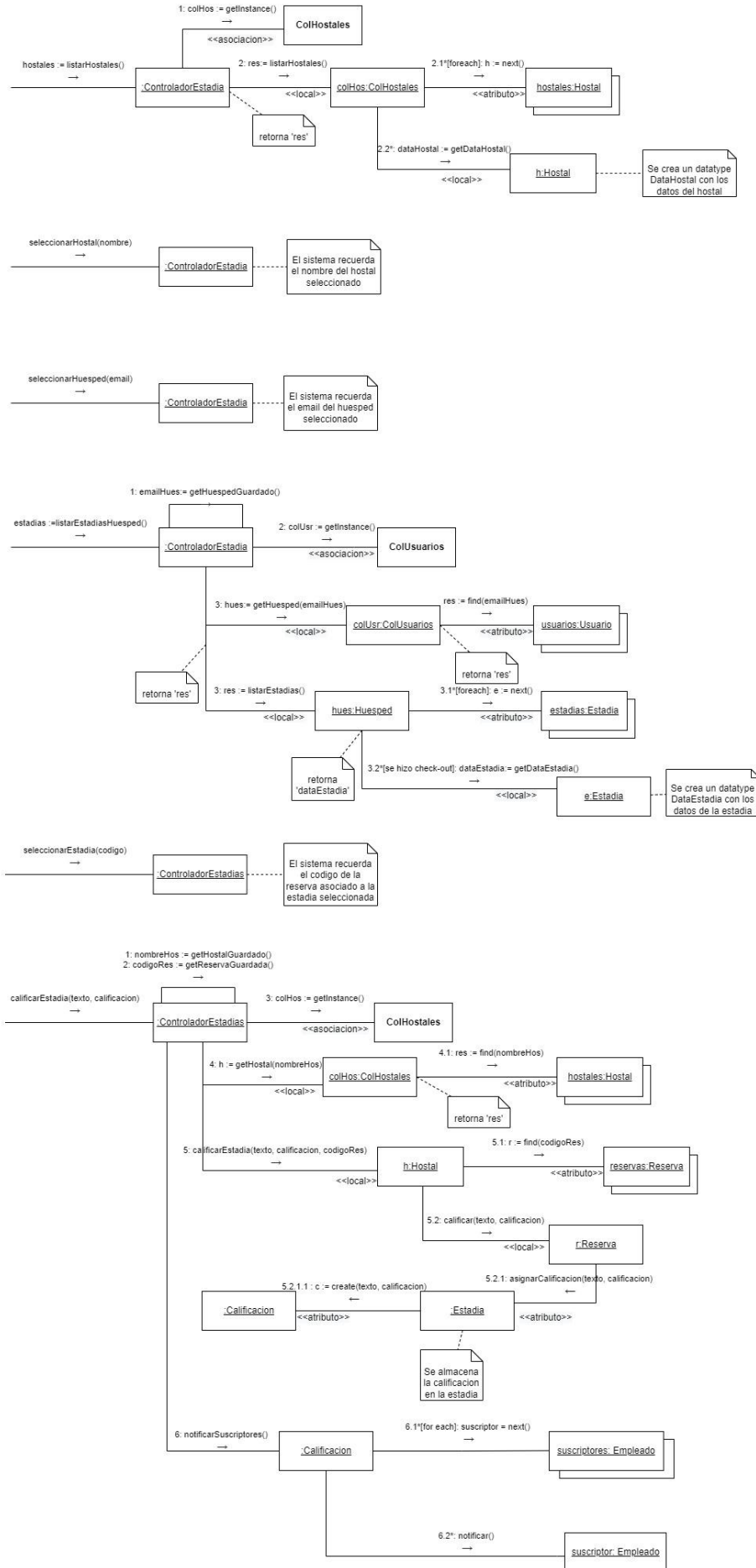
## Consultar top 3 de hostales



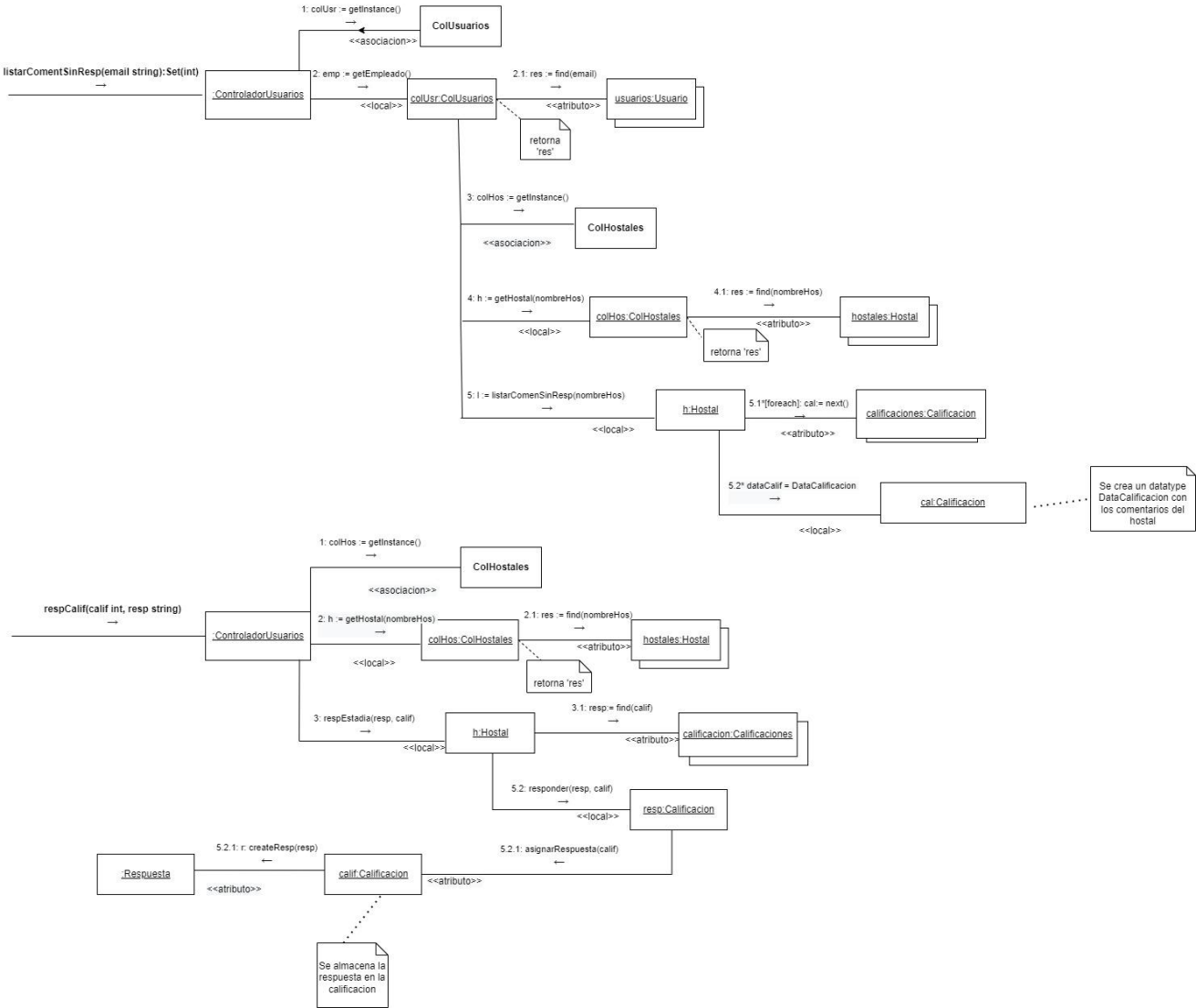
**Registrar Estadia**

## Finalizar Estadia

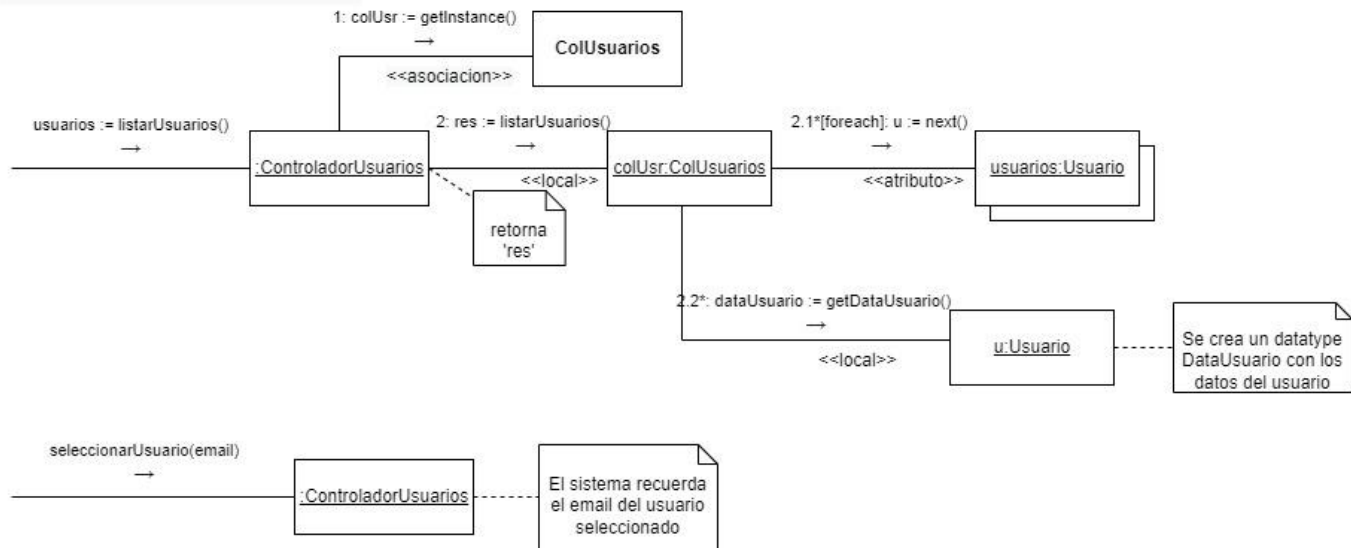


**Calificar Estadia**

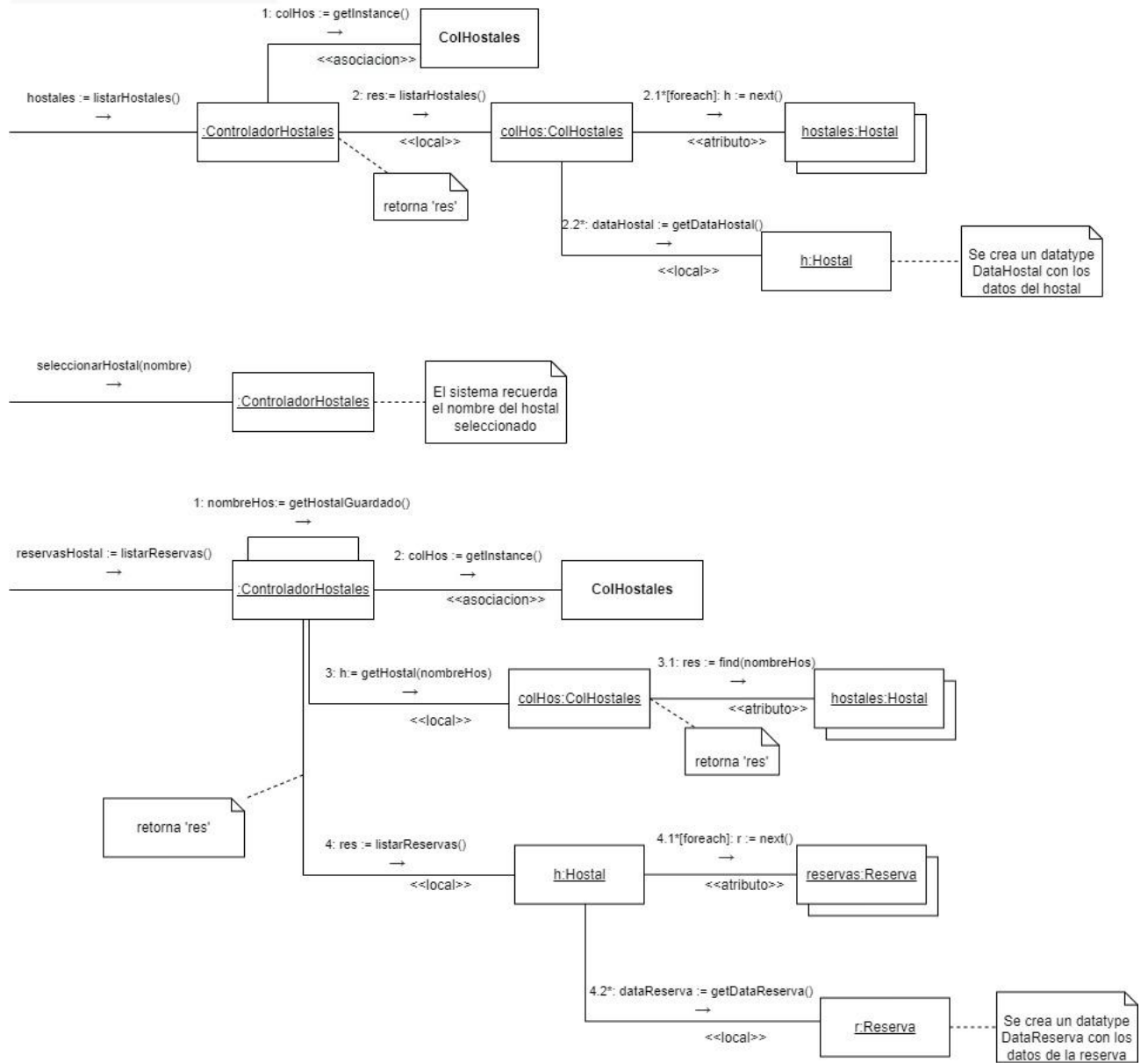
Comentar Calificacion



## Consulta de Usuario

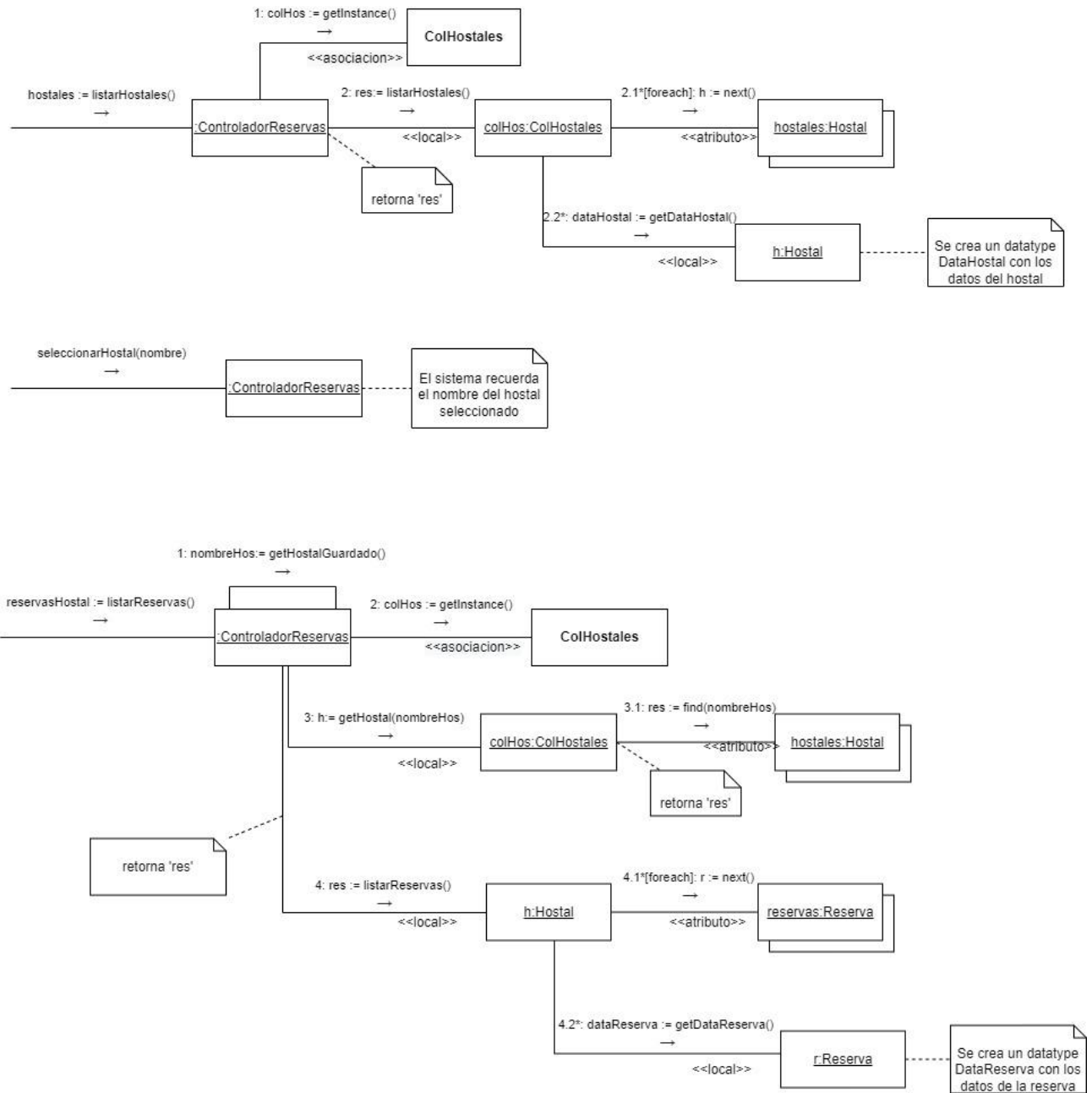


## Consulta de Hostal

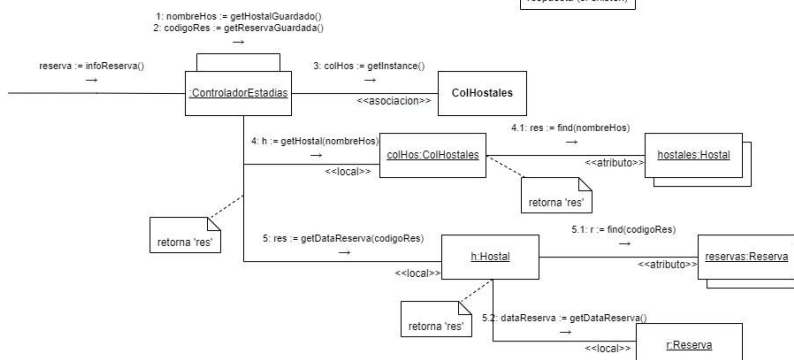
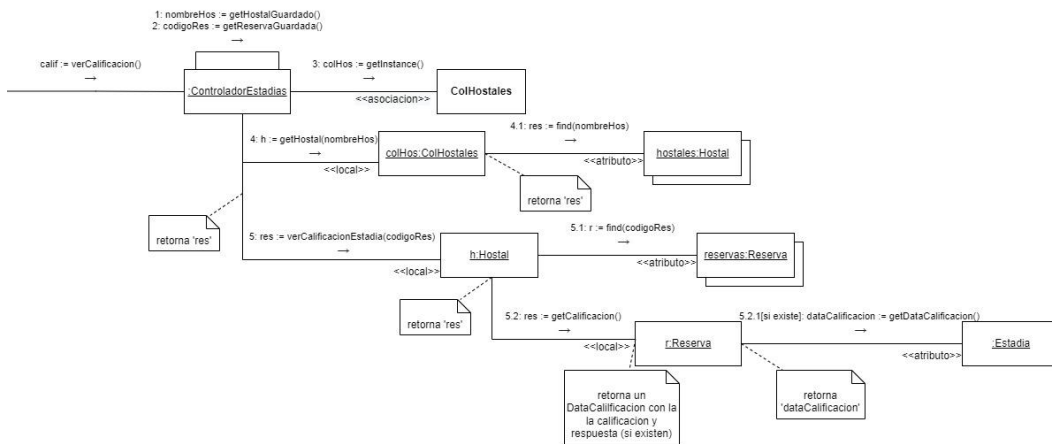
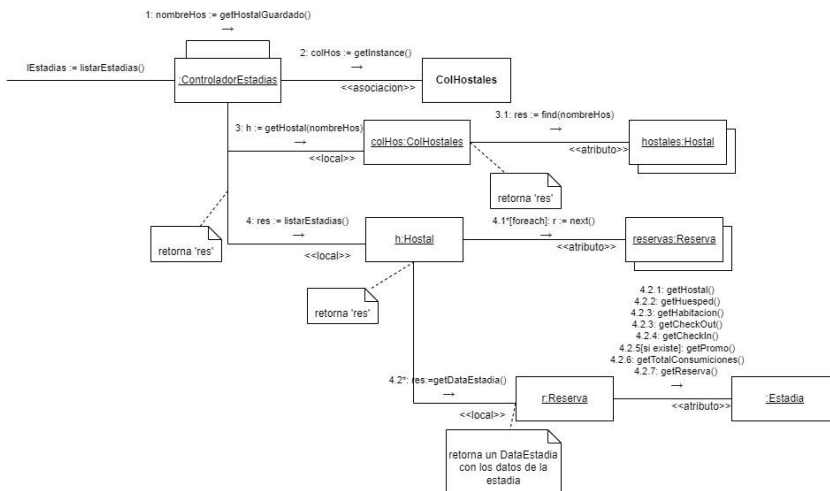
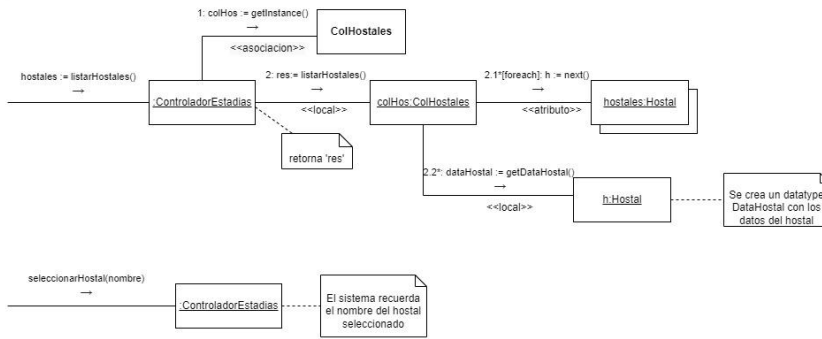


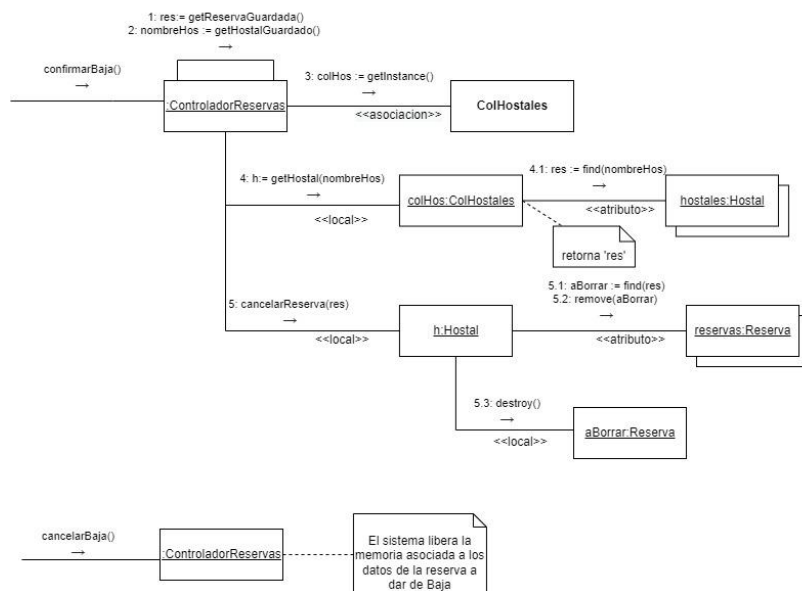
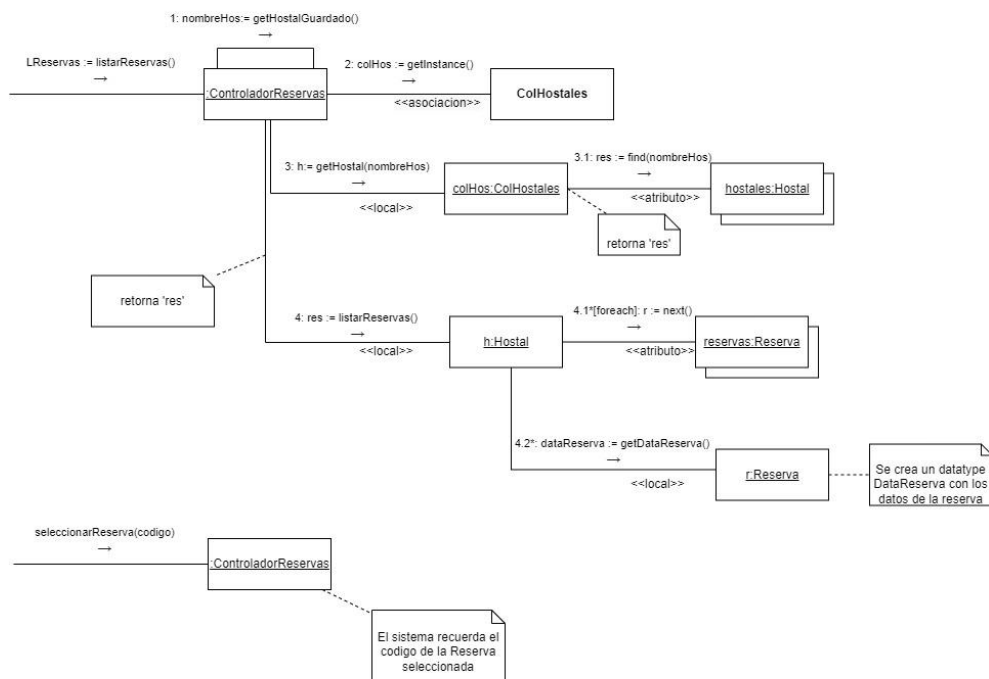
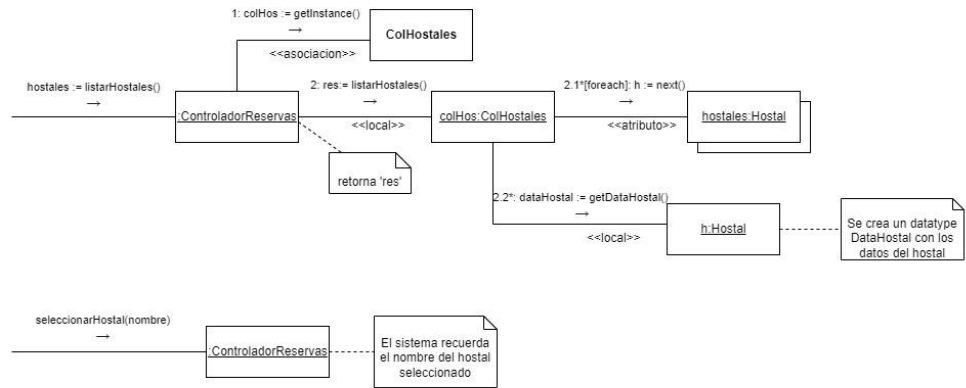


## Consulta de Reserva

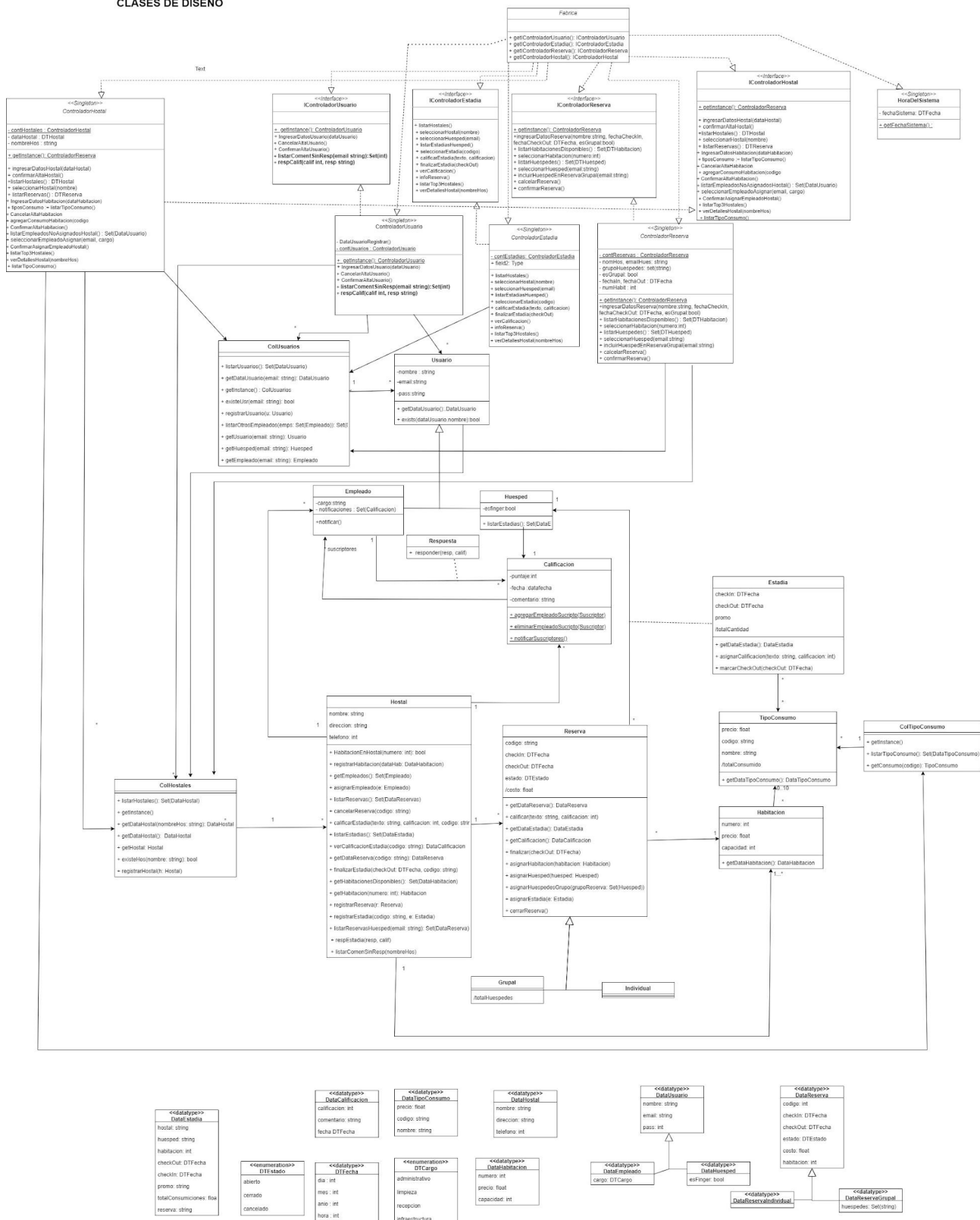


## Consulta de Estadía



**Baja de reserva**

### DIAGRAMA DE CLASES DE DISEÑO



## 2 Criterios Generales

### *2.1 Impactos de los nuevos requerimientos*

Para cumplir con el requerimiento de almacenamiento y modificación de la fecha y hora del sistema, se optó por definir una clase “Fecha del Sistema” que provee las operaciones “getFecha” y “modificarFechaSistema” y contiene la fecha actual del sistema. Esta clase es Singleton, para asegurar la existencia de una única fecha del sistema y para que provea acceso global a dichas operaciones.

Para cumplir con los nuevos requerimientos relativos a las notificaciones, se le asignó la responsabilidad de proveer las nuevas operaciones del sistema (las relacionadas con los casos Suscribirse a Notificaciones, Consulta de Notificaciones y Eliminar Suscripción) al controlador de usuarios. Además, debido a la necesidad de notificar Empleados suscriptos en el momento que se produce una nueva notificación, se decidió seguir el patrón de Diseño “Observer” para esta interacción entre las Calificaciones y Empleados, con una salvedad: puesto que siempre se notifica solamente a objetos de tipo “Empleado”, no se vio la necesidad de abstraer en una nueva clase Observador a los recipientes del mensaje; Empleado hace el rol de Observer y Calificación es el sujeto. Por ello es que se agrega una nueva relación entre Calificación y Empleado, donde la clase Calificación conoce los suscriptores a las calificaciones.

### *2.2 Sobre los criterios de asignación de responsabilidades y Patrones de diseño utilizados*

En primer lugar, para atender las operaciones del sistema se optó por agrupar casos de uso similares bajo un mismo controlador, y se decidió definir 4 controladores principales: uno (Usuarios) para las operaciones relacionadas con dar de alta, baja y listar usuarios; otro (Hostales) para las operaciones relacionadas con dar de alta y consultar hostales, así como dar de alta habitaciones en hostales y asignar empleados a éstos. Se definió también un controlador (Reservas) para las operaciones relacionadas con dar de alta, baja y consultar reservas en un Hostal y otro controlador (Estadías) para las operaciones relacionadas con dar de alta, baja, consultar y calificar estadías en un Hostal, así como comentar calificaciones y consultar el top 3 de Hostales. Todos estos controladores son Singleton.

Se definieron también 2 clases aparte para contener las colecciones de Usuarios y Hostales respectivamente, también Singleton para asegurar la existencia una única colección de cada uno. Se justifica pues dichas colecciones son accedidas por muchos controladores, y se deseó evitar que sea alguno de los controladores los que almacene dicha información.

El criterio GRASP utilizado para definir/asignar controladores fue “Controller”, pues manejan (la memoria de selecciones del usuario para) las operaciones relacionadas con un mismo caso de uso.