

Programación 4

Informe del Modelo de Casos de Uso

Grupo 01

Integrantes

Federico Bello	CI: 4.993.837-3
Javier Olveyra	CI: 4.971.974-7
Mathias Rodriguez	CI: 5.198.420-3
Danilo Giacometti	CI: 5.172.582-9
Diego Amorena	CI: 5.011.502-7

Docente: Martín Prino

Alta de Usuario

Diagramas de Secuencia del Sistema



Contratos

Nombre	Nuevo Empleado
Operación	<i>nuevoEmpleado(nomE:String; emailE:String; contE:String; cargoE:string):DTEmpleado</i>
Entrada	<i>nomE</i> representa el nombre del nuevo empleado, <i>emailE</i> representa el mail del nuevo empleado, <i>contE</i> representa la contraseña del nuevo empleado, <i>cargoE</i> representa el cargo del nuevo empleado
Salida	Retorna un DTEmpleado con los parámetros pasados
Descripción	Se crea una nueva instancia de DTEmpleado con los parámetros anteriores

Precondiciones y Postcondiciones

Post1: Crea una instancia de DTEmpleado con los parámetros dados.

Nombre	Nuevo Huésped
Operación	<i>nuevoHuesped(nomH:String; emailH:String; contH:String; esFinger:bool):DTHuesped.</i>
Entrada	<i>nomH</i> representa el nombre del nuevo huésped, <i>emailH</i> representa el mail del nuevo huésped, <i>contH</i> representa la contraseña del nuevo huésped, <i>esFinger</i> representa si el huésped es Finger o no, es true en caso de serlo y false de caso contrario.
Salida	Retorna un DTHuesped con los parámetros pasados.
Descripción	Se crea una nueva instancia de DTHuesped con los parámetros anteriores.

Precondiciones y Postcondiciones

Post1: Crea una instancia de DTUsuario con los parametros dados.

Nombre	Actualizar Mail
Operación	actuliazarMail(email:string ,us: DTUsuario)
Entrada	<i>us</i> representa el DT usuario al cual se le quiere modificar el mail, <i>email</i> representa el nuevo mail que se le quiere dar a <i>us</i> .
Salida	No hay
Descripción	Se modifica el email actual de un DTUsuario, por otro email otorgado como parámetro de entrada.

Precondiciones y Postcondiciones

Post1: Al DTUsuario *us* se le modificó el mail anterior por el nuevo pasado por parámetro.

Nombre	Cancelar agregar usuario
Operación	cancelarUsuario(us: DTUsuario)
Entrada	<i>us</i> representa el usuario el cual se quiere no dar de alta
Salida	No hay
Descripción	Se cancela la secuencia para agregar un usuario y se borran todos los datos ingresados hasta el momento

Precondiciones y Postcondiciones

Pre1: Se está agregando usuario.

Post1 : Se borran los datos ingresados y ya no se sigue con el procedimiento para agregar un nuevo usuario.

Nombre	Confirmar alta empleado
Operación	<i>confirmarAltaEmpleado(us :DTEmpleado)</i>
Entrada	<i>us</i> representa los datos de un nuevo empleado a ingresar en el sistema
Salida	No hay
Descripción	confirma el alta del empleado

Precondiciones y Postcondiciones

Pre1: el mail de *us* es único en el sistema.

Post1: Se guarda en el sistema la instancia de DTEmpleado pasada como parámetro de entrada.

Nombre	Confirmar alta huésped
Operación	<i>confirmarAltaHuesped(us :DTHuesped)</i>
Entrada	<i>us</i> representa los datos de un nuevo huésped a ingresar en el sistema
Salida	No hay
Descripción	Confirma el alta del huésped

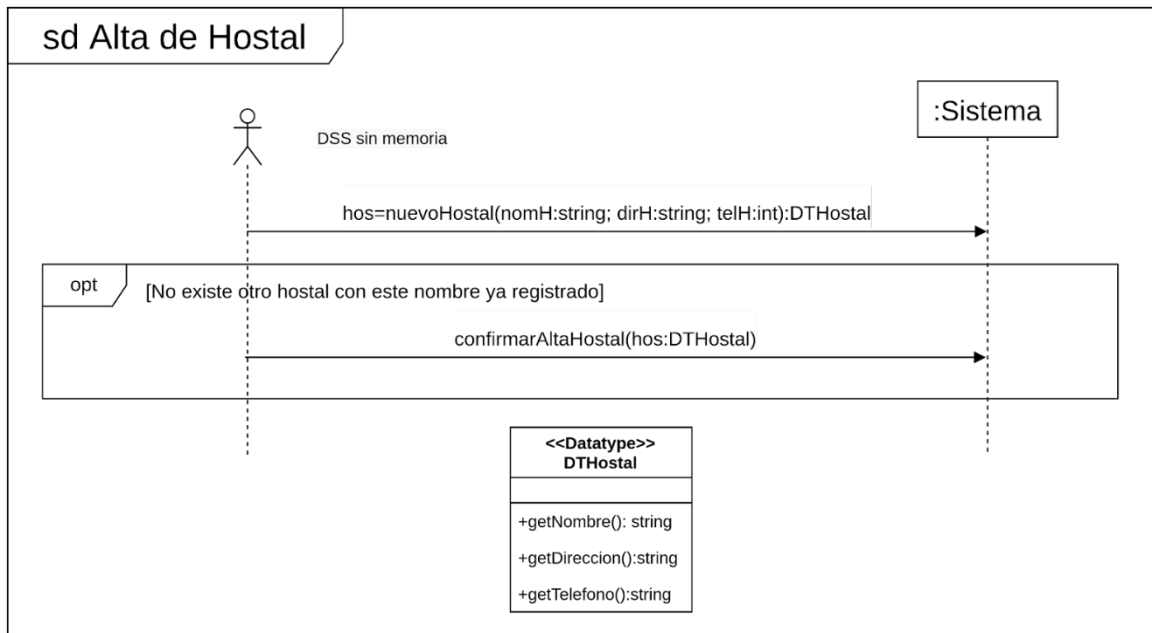
Precondiciones y Postcondiciones

Pre1: El mail de *us* es único en el sistema.

Post1: Se guarda en el sistema la instancia de DTEmpleado pasada como parámetro de entrada.

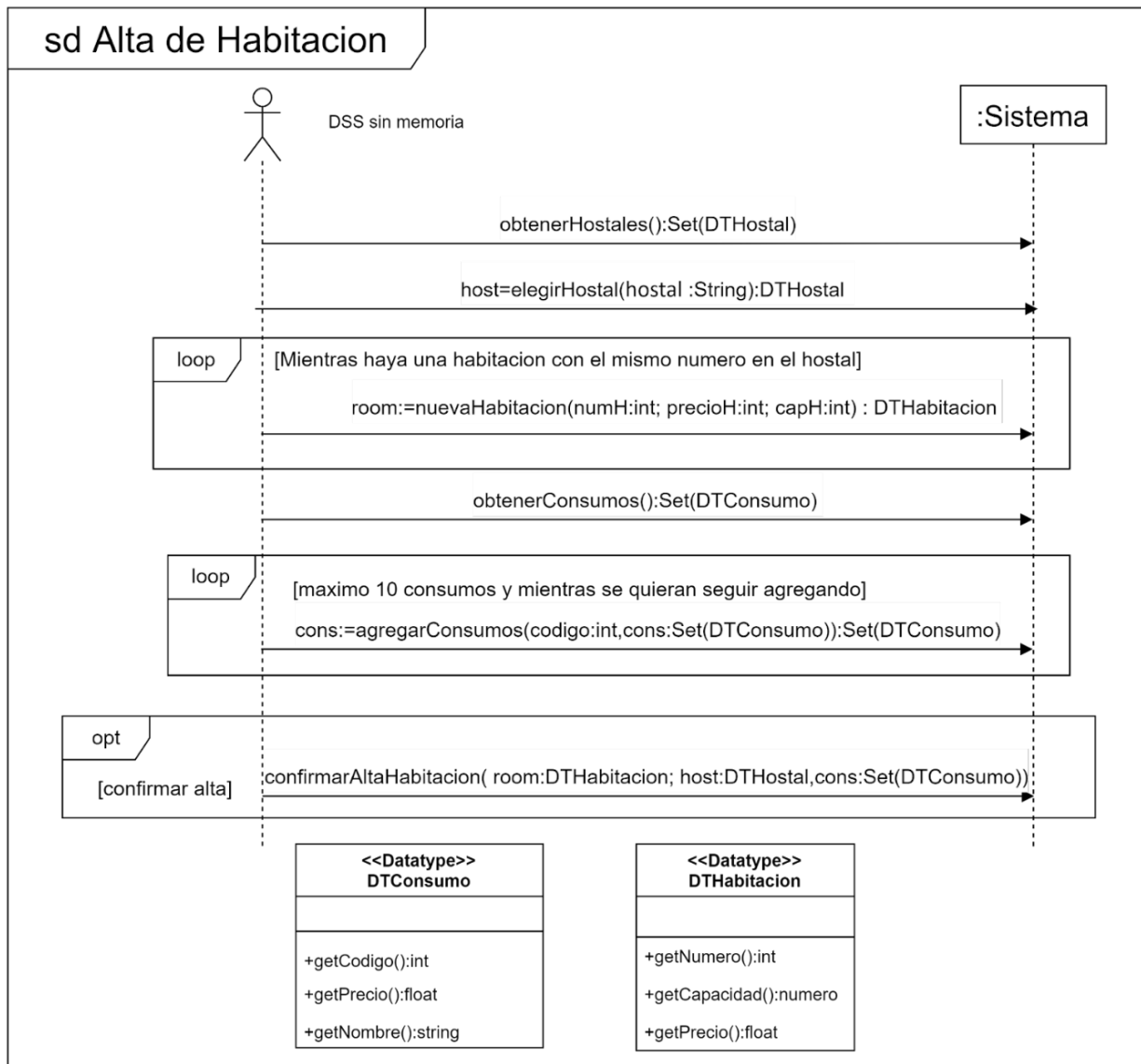
Alta de Hostal

Diagramas de Secuencia del Sistema



Alta de Habitación

Diagramas de Secuencia del Sistema



Contratos

Nombre	Obtener hostales
Operación	<i>obtenerHostales():Set(DTHostal)</i>
Entrada	No hay
Salida	Un conjunto de instancias de DTHostal que representa la información básica (nombre, dirección y teléfono) de los hostales del sistema
Descripción	Retorna los hostales registrados en el sistema

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un hostel.

Post1: Se construyó y retorno un conjunto de instancias de DTHostal con la información de los hostales existentes.

Nombre	Elegir hostel
Operación	<i>ElegirHostal(hostal :String):DTHostal</i>
Entrada	<i>hostal</i> representa el nombre del hostel el cual se quiere obtener
Salida	Devuelve el DTHostal en el sistema que se identifica con <i>hostal</i>
Descripción	Retorna los hostales registrados en el sistema

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un hostel.

Pre2: Existe en el sistema un hostel que se identifica con *hostal*.

Post1: Se devolvió el hostel que se identifica con *hostal*

Nombre	Nueva Habitación
Operación	<i>nuevaHabitacion(numH:int; precioH:int; capH:int): DTHabitacion</i>
Entrada	<i>numH</i> representa el número de la nueva habitación <i>precioH</i> representa el precio de la nueva habitación <i>capH</i> representa la capacidad de la nueva habitación
Salida	Devuelve un DTHabitacion con los parámetros dados
Descripción	Se crea una nueva instancia de DTHabitacion con los parámetros anteriores, guardandolo en el sistema

Precondiciones y Postcondiciones

Post1 Se crea una instancia de DTHabitacion con los parámetros otorgados.

Nombre	Obtener consumos
Operación	<code>obtenerConsumo():Set(DTConsumo)</code>
Entrada	No hay
Salida	Un conjunto de instancias de DTConsumo que representa la información de los consumos del sistema
Descripción	Retorna los consumos registrados en el sistema

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un consumo.

Post1: Se construyó y retorno un conjunto de instancias de DTConsumo con la información de los consumos existentes.

Nombre	Agregar consumos
Operación	<code>agregarConsumos(codigo:int, cons:Set(DTConsumo)):Set(DTConsumo)</code>
Entrada	<i>código</i> representa el identificador el cual se quiere agregar al set de DTConsumo <i>cons</i>
Salida	Devuelve un set de DTConsumo
Descripción	Agrega un consumo a un set de consumos y devuelve el mismo set con el consumo agregado

Precondiciones y Postcondiciones

Pre1: el consumo no está agregado en la habitación.

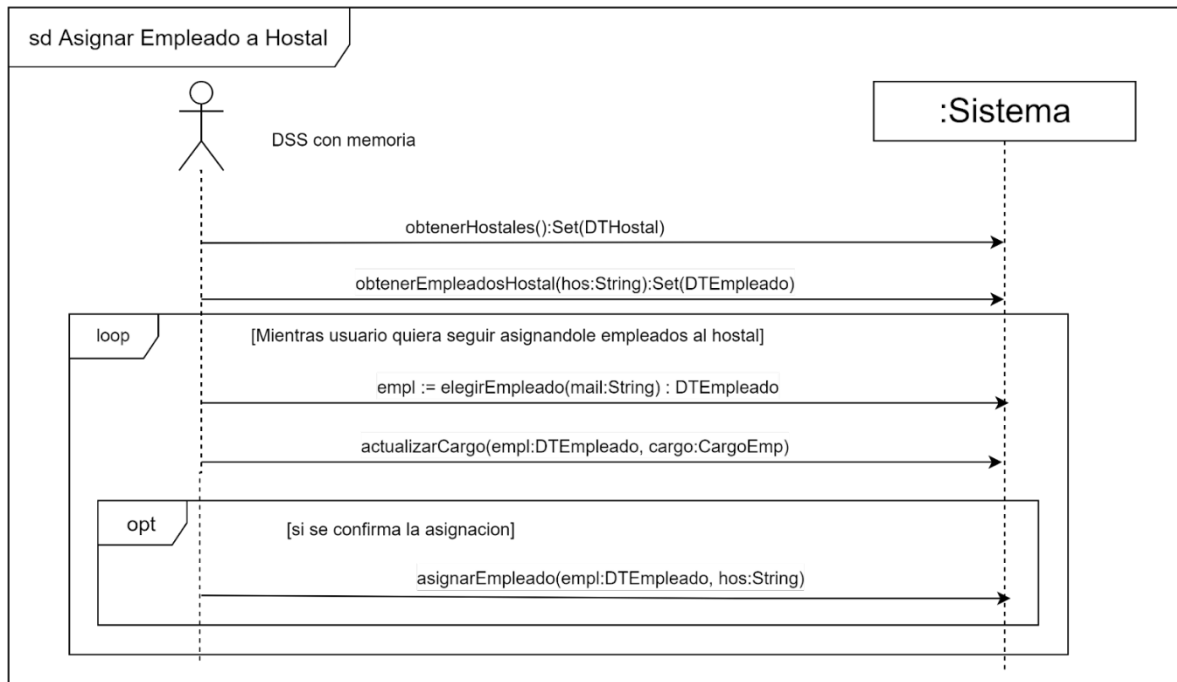
Post1: Se asociaron tipos de consumo a la habitación room.

Nombre	Confirmar Alta de Habitación
Operación	<i>confirmarAltaHabitacion(room:DTHabitacion; host:DTHostal,cons:Set(DTConsumo))</i>
Entrada	<i>room</i> representa los datos de la nueva habitación a ingresar en el sistema <i>host</i> representa el hostel donde se va a agregar esa nueva habitación <i>cons</i> representa el conjunto de consumos que se van a asociar a dicha habitación
Salida	No hay
Descripción	Se ingresan los datos de la nueva habitación con su hostel y consumos correspondientes y se agrega al sistema

Precondiciones y Postcondiciones
<p>Pre1: No hay una instancia DTHabitacion dentro de <i>host</i> con el mismo número que <i>room</i>.</p> <p>Post1: Se creó en el sistema una instancia de tipo DTHabitación, con los atributos pasados por parámetro, se asociada a un tipo DTHostal con los DTConsumo correspondientes.</p>

Asignar Empleado a Hostal

Diagramas de Secuencia del Sistema



Contratos

La operación obtenerHostales() está descrita en la sección Alta de Habitación.

Nombre	Obtener empleados Hostal
Operación	obtenerEmpleadosHostal(<i>hos</i> :string):Set(DTEmpleados)
Entrada	<i>nombre</i> representa el nombre de un hostel en el sistema
Salida	Un conjunto de instancias de DTEmpleado
Descripción	Retorna los empleados que no trabajan en <i>hos</i> .

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un empleado que no está asignado a *hos*.
 Pre2: el hostel de nombre *hos* existe en el sistema.

Post1: Se construyó y retorno un conjunto de instancias de DTEmpleado con la información básica (nombre, mail, password, cargo) de los empleados existentes que están en el sistema pero no asignados al hostel *hos*.

Nombre	Elegir empleado
Operación	<i>elegirEmpleado(mail:String) : DTEmpleado</i>
Entrada	<i>mail</i> representa el email del empleado el cual se quiere seleccionar
Salida	Un DTEmpleado con la información del empleado de email <i>mail</i>
Descripción	Devuelve una instancia de DTEmpleado guardado en el sistema de email <i>mail</i>

Precondiciones y Postcondiciones

Pre1: Existe en el sistema un empleado de email *mail*.

Post1: Se devolvió el DTEmpleado asociado con *mail*.

Nombre	Actualizar cargo
Operación	<i>actualizarCargo(empl:DTEmpleado; cargo: CargoEmp)</i>
Entrada	<i>empl</i> representa el DTEmpleado <i>cargo</i> representa el nuevo cargo de tipo CargoEmp
Salida	No hay
Descripción	Le modifica al DTEmpleado <i>empl</i> su atributo cargo. Independientemente de cual fuera el anterior el nuevo será <i>cargo</i>

Precondiciones y Postcondiciones

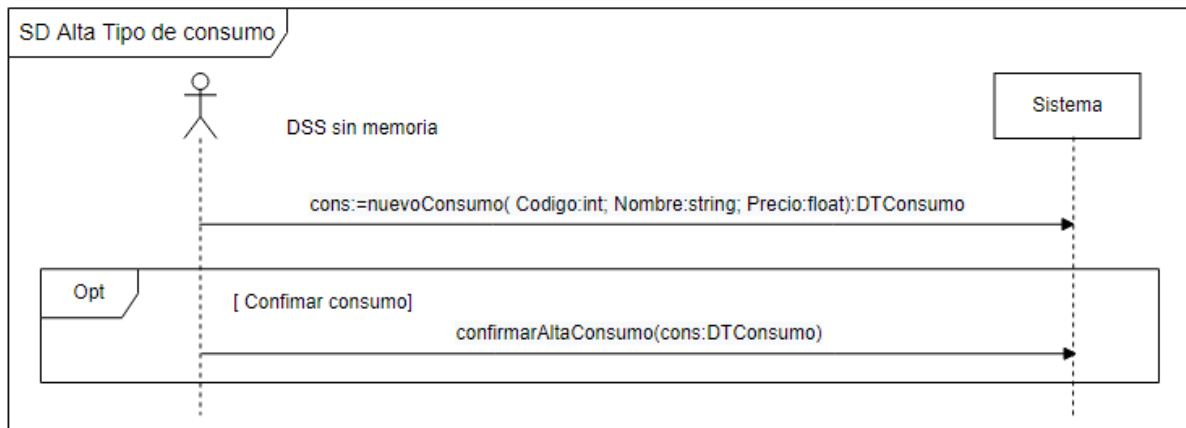
Post1: El nuevo atributo cargo de *empl* es *cargo*

Nombre	Asignar empleado a hostel
Operación	<i>asignarEmpleado(empl:DTEmpleado. hos:string)</i>
Entrada	<i>empl</i> representa una instancia de DTEmpleado <i>hos</i> representa el nombre de un hostel del sistema
Salida	No hay
Descripción	Se asocia el empleado <i>empl</i> con el hostel <i>hos</i> . Si el empleado ya está asociado a un hostel esta asociación se elimina antes de asociarlo al nuevo hostel

Precondiciones y Postcondiciones
<p>Pre1: el empleado <i>empl</i> ya existe en el sistema</p> <p>Pre2: el hostel de nombre <i>hos</i> existe en el sistema</p> <p>Post1: Se asoció el empleado <i>empl</i> con el hostel guardado en el sistema de nombre <i>hos</i></p>

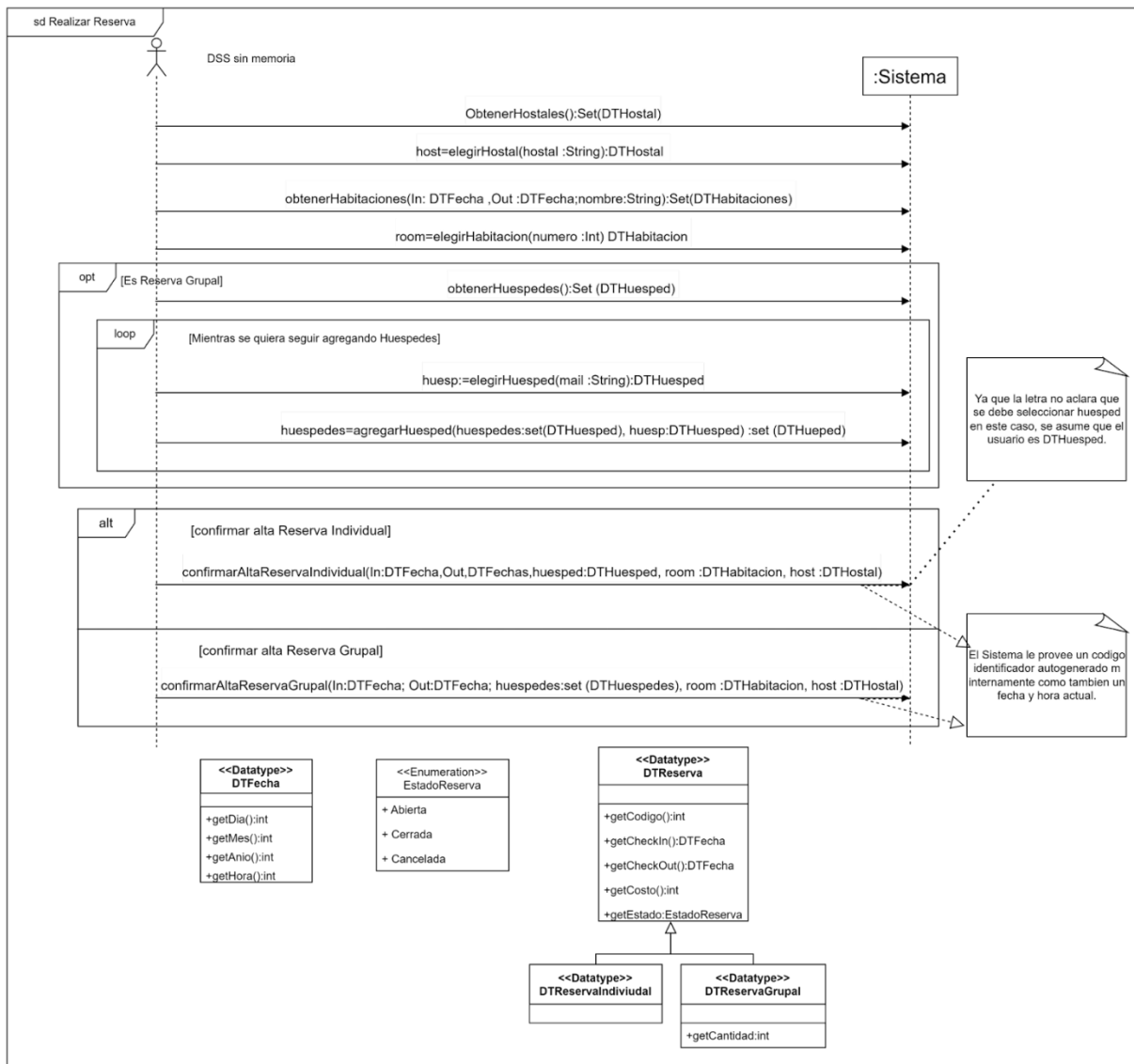
Alta Tipo de Consumo

Diagramas de Secuencia del Sistema



Realizar Reserva

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHostales()` está descrita en la sección Alta de Habitación.

La operación `elegirHostal()` está descrita en la sección Alta de Habitación.

Nombre	Obtener Habitaciones
Operación	<i>obtenerHabitaciones(In: DTFecha ,Out :DTFecha,hos:DThostal):Set(DTHa-bitacion)</i>
Entrada	<i>In</i> representa la DTFecha de checkIn <i>Out</i> representa la DTFecha de checkOut <i>hos</i> es el hostel del que se quieren obtener las habitaciones
Salida	Devuelve un conjunto de DTHabitacion
Descripción	Retorna todas las habitaciones disponibles entre la fecha <i>In</i> y <i>Out</i> que encuentran en el hostel <i>nombre</i>

Precondiciones y Postcondiciones	
Pre1: Existe en el sistema el hostel <i>hos</i> de nombre <i>nombre</i>	
Post1: Se construyó y retorno un conjunto de instancias de DTHabitacion con la información de las habitaciones disponibles entre las fechas <i>In</i> y <i>Out</i> en el Hostel <i>hos</i>	

Nombre	Elegir habitación
Operación	<i>elegirHabitacion(numero :Int) DTHabitacion</i>
Entrada	<i>numero</i> representa el identificador de una habitación
Salida	Un DTHabitacion con la información de la habitación identificada por <i>numero</i>
Descripción	Devuelve una instancia de DTHabitacion con la información correspondiente a una habitacion guardada en el sistema de identificador <i>numero</i>

Precondiciones y Postcondiciones	
Pre1: Existe en el sistema una habitación identificada por <i>numero</i>	
Post1: se devolvió el DTHabitacion con la información básica de la habitación identificada por mail	

Nombre	Obtener huéspedes
Operación	obtenerHuespedes():Set (DTHuesped)
Entrada	No hay
Salida	Un conjunto de instancias de DTHuesped que representa la información de los huéspedes del sistema
Descripción	Retorna los huéspedes registrados en el sistema.

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un huésped.

Post1: Se construyó y retorno un conjunto de instancias de DTHuesped con la información de los huéspedes existentes

Nombre	Elegir Huesped
Operación	<i>elegirHuesped(mail :String):DTHuesped</i>
Entrada	<i>mail</i> representa el email del huesped el cual se quiere elegir
Salida	Devuelve un DTHuesped de email <i>mail</i>
Descripción	Devuelve un DTHuesped guardado en el sistema bajo el email <i>mail</i>

Precondiciones y Postcondiciones

Pre1: Existe en el sistema un huesped de email *mail*

Post1: Se devolvió el Huesped de email *mail*

Nombre	Agregar Huésped
Operación	<i>agregarHuesped(huespedes:set(DTHuespedes), huesp:DTHuesped)</i>
Entrada	<i>huesp</i> representa una instancia de DTHuesped <i>huespedes</i> que es un set de DTHuesped
Salida	Conjunto de DTHuesped
Descripción	Se agrega <i>huesp</i> al set <i>huespedes</i>

Precondiciones y Postcondiciones
pre1: El huésped <i>huesp</i> existe en el sistema pre2: El huésped <i>huespedes</i> no pertenece al set huéspedes post1: Se agregó <i>huesp</i> al set <i>huespedes</i>

Nombre	Confirmar alta reserva individual
Operación	<i>confirmarAltaReservaIndividual(In:DTFecha, Out:DTFecha, huesped:DTHuesped, room :DTHabitacion, host :DTHostal)</i>
Entrada	<i>huesped</i> representa una instancia de DTHuesped <i>room</i> representa los datos de DTHabitacion <i>host</i> representa el hostel donde se realiza la reserva <i>In</i> representa el CheckIn de la reserva <i>Out</i> representa el CheckOut de la reserva
Salida	No hay
Descripción	Confirma el alta de la reserva individual en el Sistema

Precondiciones y Postcondiciones
pre1: En el sistema se encuentran almacenados el <i>huesped</i> , <i>habitación</i> y <i>host</i> . post1: Se creó en el sistema una reserva con los datos ingresados. Además se guarda la hora y fecha actual del Sistema asociado a dicha DTReserva, como también un identificador creado automáticamente por el sistema.

Nombre	Confirmar alta reserva grupal
Operación	<i>confirmarAltaReservaGrupal(In:DTFecha, Out:DTFecha, huespedes:set (DTHuespedes), room :DTHabitacion, host :DTHostal)</i>
Entrada	<i>huespedes</i> representa el conjunto de DTHuespedes <i>room</i> representa una instancia de DTHabitacion <i>host</i> representa una instancia de DTHostal <i>In</i> representa el CheckIn de la reserva <i>Out</i> representa el CheckOut de la reserva
Salida	No hay
Descripción	Confirma el alta de la reserva grupal en el sistema.

Precondiciones y Postcondiciones

Pre1: En el sistema se encuentran almacenados el conjunto de *huéspedes*, *room* y *host*.

Post1: Se guarda en el sistema una reserva con los datos ingresados. Además se guarda la hora y fecha actual del Sistema asociado a dicha DTReserva, como también un identificador creado automáticamente por el sistema.

Consulta top 3 hostales

Diagramas de Secuencia del Sistema



Contratos

Nombre	Mejores hostales
Operación	<i>mejoresTresHostales():Set(DTHostal)</i>
Entrada	No hay
Salida	Un conjunto de a lo sumo tres DTHostal
Descripción	Calcula el promedio de calificaciones de cada DTHostal en el sistema y posteriormente devuelve un conjunto con hasta los tres mejores hostales según su promedio de calificación. Si hay menos de 3 hostales ingresados en el sistema muestra los que haya.

Precondiciones y Postcondiciones

Pre1: Debe existir por lo menos un hostal en el sistema

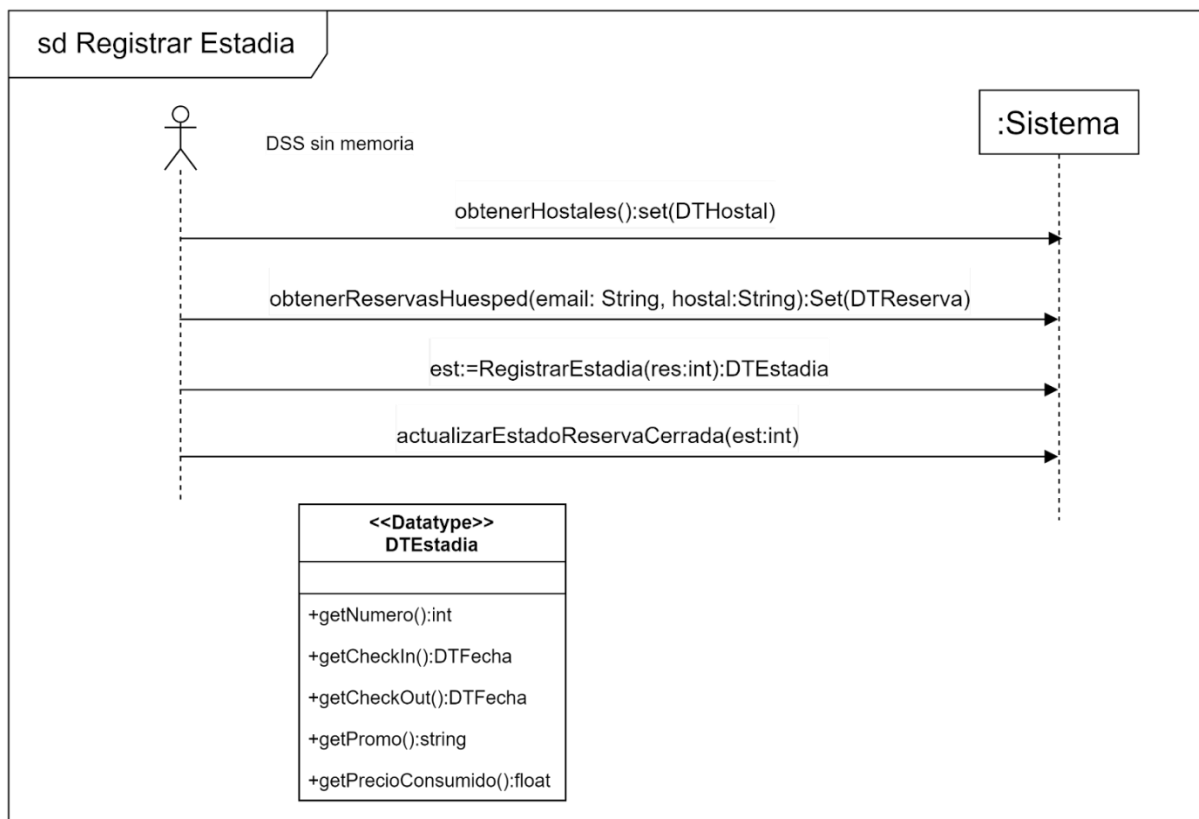
Post1: Se construyó y retorno un conjunto de hostales según el promedio de calificación, el orden de los hostales es decreciente según su calificación

Nombre	Detalles de hostales
Operación	<i>detallesHostal(nombre: String):Set (DTCalificaion)</i>
Entrada	<i>nombre</i> representa el nombre de un DTHostal
Salida	Conjunto de DTCalificacion
Descripción	Devuelve el conjunto de todas las calificaciones, en casos de haberlas, asignadas a un hostel identificado con <i>nombre</i>

Precondiciones y Postcondiciones
Pre1: Debe existir un hostel identificado por <i>nombre</i>
Post1: Se retornó el conjunto de calificaciones asociadas al hostel identificado por <i>nombre</i>

Registrar Estadía

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHostales()` está descrita en la sección Alta de Habitación.

Nombre	Obtener reservas
Operación	<i>obtenerReservasHuesped(email: String,hostal:String):Set(DTReserva)</i>
Entrada	<i>email</i> representa un string asociado a un único DTHuesped <i>hostal</i> representa el nombre de un DTHostal del sistema
Salida	Devuelve un conjunto de DTReserva
Descripción	Retorna todas las reservas no canceladas en donde el huésped con mail <i>email</i> está registrado como parte de la reserva y corresponden al hostal de nombre <i>hostal</i>

Precondiciones y Postcondiciones

Pre1: Existe en el sistema un huésped de email *email*
 Pre2: Existe en el sistema un hostel de nombre *hostal*

Post1: Se construyó y retorno un conjunto de instancias de DTReserva no canceladas en las que el huésped forma parte y se hospeda en el hostel *nombre*

Nombre	Registrar estadía
Operación	<i>RegistrarEstadia(res:int) :DTEstadia</i>
Entrada	<i>res</i> representa una reserva
Salida	Se devuelve una estadía de tipo DTEstadia
Descripción	Con los datos de la reserva y los datos de fecha y hora actuales brindados por el sistema se crea una DTEstadia de fecha de checkIn la actual

Precondiciones y Postcondiciones

Pre1: Existe la reserva *res* en el sistema

Post1: En la reserva se encuentra la información del check-in del huésped asociado a esta reserva

Nombre	Set estado reserva cerrada
Operación	<i>actualizarEstadoReservaCerrada(res:int)</i>
Entrada	<i>res</i> representa una reserva
Salida	No hay
Descripción	Se marca como cerrada la reserva <i>res</i>

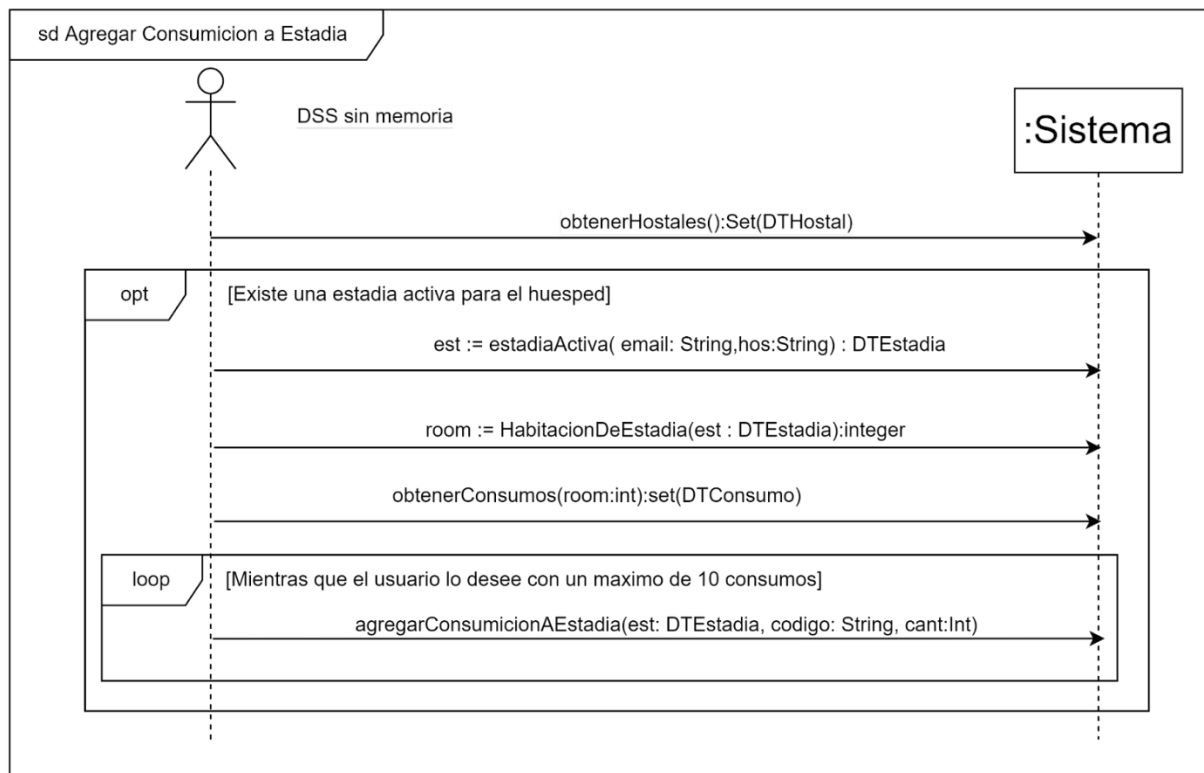
Precondiciones y Postcondiciones

Pre1: Existe una DTReserva con el código *res* en el sistema

Post1: La reserva *res* queda cerrada, tomados de ahora en adelante esa reserva como finalizada.

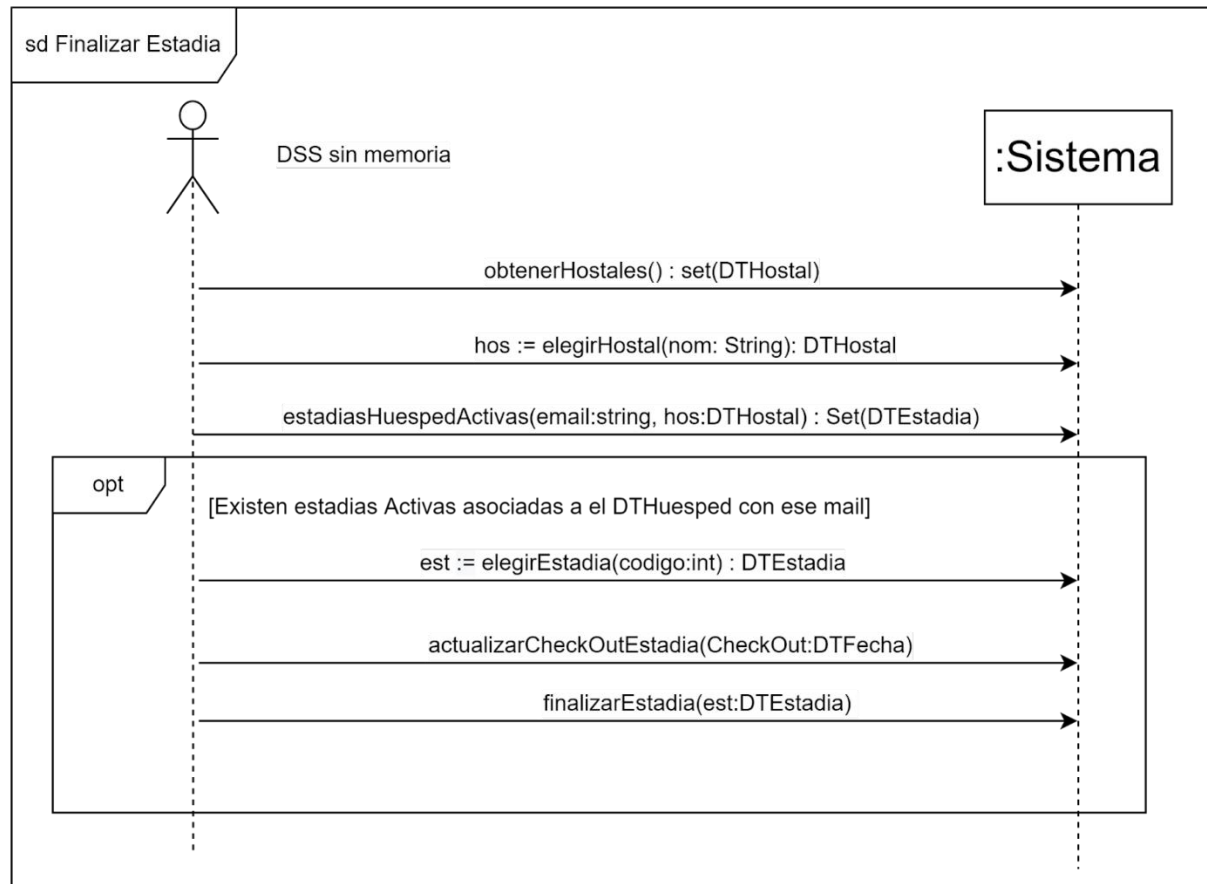
Agregar Consumición a Estadía

Diagramas de Secuencia del Sistema



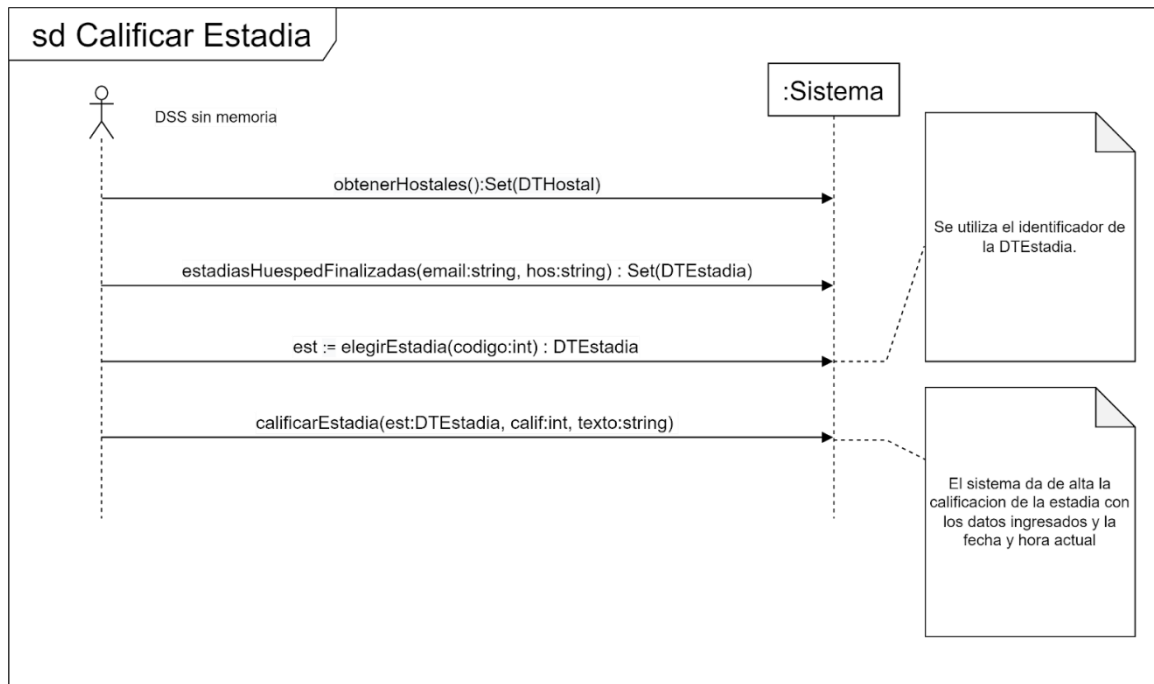
Finalizar Estadía

Diagramas de Secuencia del Sistema



Calificar Estadía

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHostales()` está descrita en la sección Alta de Habitación.

Nombre	Estadías de Huéspedes finalizadas
Operación	<i>estadiasHuespedFinalizadas(email:string, hos:string) : Set(DTEstadia)</i>
Entrada	<i>email</i> representa el email asociado a un único DTHuesped <i>hos</i> representa el nombre un hostel
Salida	Conjunto de DTEstadia
Descripción	Retorna un conjunto de estadías finalizadas asociadas a un DTHuesped identificado con <i>email</i> en en hostel <i>hos</i> .

Precondiciones y Postcondiciones

Pre1: Existe un huésped que se identifique con *email*

Pre2: Existe un hostel que se identifique con *hos*

Post1: Se retorna un conjunto de estadías finalizadas para un huésped en el hostel *hos*.

Nombre	Elegir Estadía
Operación	<i>elegirEstadia(codigo:Integer) : DTEstadia</i>
Entrada	<i>codigo</i> representa el código de la DTReserva asociada a la estadía que se quiere seleccionar.
Salida	Devuelve una DTEstadia
Descripción	Retorna la DTEstadia en el sistema asociada a la DTReserva identificada <i>codigo</i>

Precondiciones y Postcondiciones

Pre1: Existe en el sistema una reserva de código *código*
 Pre2: Existe en el sistema una estadía asociada a la reserva de código *código*
 Post1: Se devolvió la estadía asociada a la reserva identificada por *código*

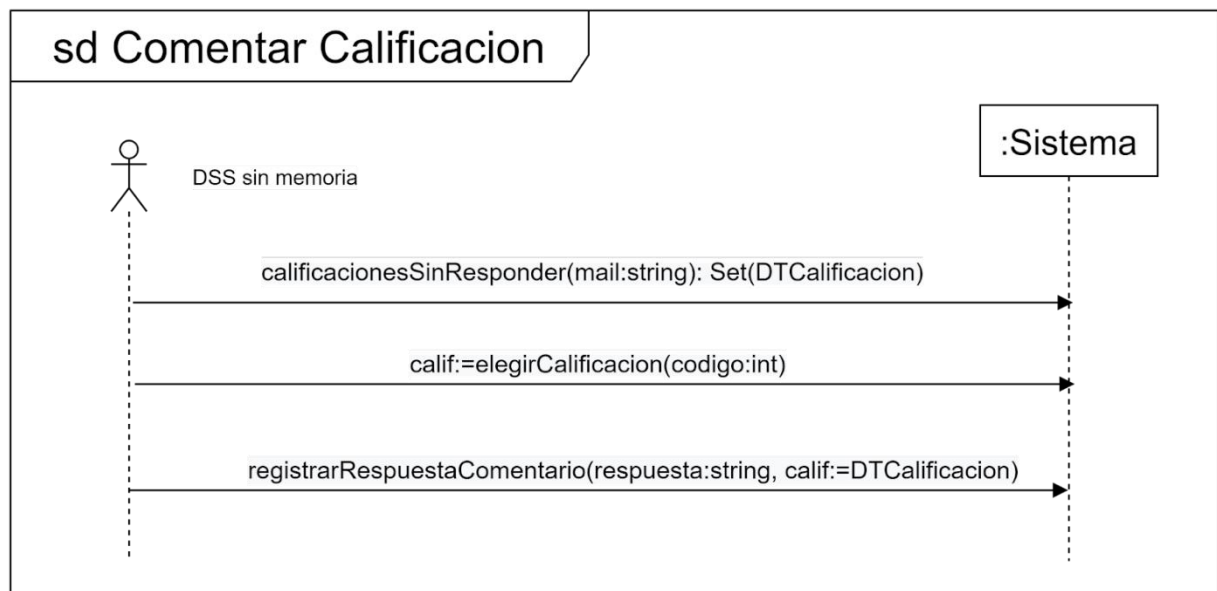
Nombre	Calificar Estadía
Operación	<i>calificarEstadia(est:DTEstadia, calif:DTCalificacion, texto:string)</i>
Entrada	<i>est</i> representa la estadía a calificar, <i>calif</i> representa la calificación que se va a asignar a la estadía, <i>texto</i> representa un comentario sobre la estadía
Salida	No hay
Descripción	El usuario califica una estadía con una calificación y un comentario.

Precondiciones y Postcondiciones

Pre1: La estadía *est* está registrada en el sistema
 Post1: La estadía *est* ahora tiene una calificación que contiene un número que la la puntúa y un comentario

Comentar Calificación

Diagramas de Secuencia del Sistema



Contratos

Nombre	Comentarios sin responder
Operación	<i>comentariosSinResponder(mail:string): Set(DTCalificacion)</i>
Entrada	<i>mail</i> representa el email de un empleado
Salida	Conjunto de calificaciones sin responder
Descripción	Devuelve un conjunto de calificaciones sin responder del hostel para el cual trabaja el empleado identificado por el email <i>mail</i>

Precondiciones y Postcondiciones

Pre1: Existe un empleado que se identifique por *mail*

Post1: Se construye y retorna un conjunto de instancias de DTCalificacion con la información de los comentarios sin responder del hostel para el cual trabaja el empleado identificado por *mail*

Nombre	Elegir calificación
Operación	<code>elegirCalificacion(codigo:int):DTCalificacion</code>
Entrada	<i>codigo</i> representa el identificador asociado a alguna calificación
Salida	Calificación de identificador <i>codigo</i>
Descripción	Devuelve la calificación del sistema la cual tenga de identificador el valor <i>codigo</i>

Precondiciones y Postcondiciones

Pre1: Existe una calificación que se identifique por *codigo*

Post1: Se retorna un DTCalificacion con los valores de la calificación de identificador *codigo* guardada en el sistema

Nombre	Registrar respuesta de comentario
Operación	<code>registrarRespuestaComentario(comentario:string, calif:DTCalificacion)</code>
Entrada	<i>comentario</i> representa una respuesta para la calificación <i>calif</i>
Salida	No hay
Descripción	Se agrega una respuesta a la calificación <i>calif</i>

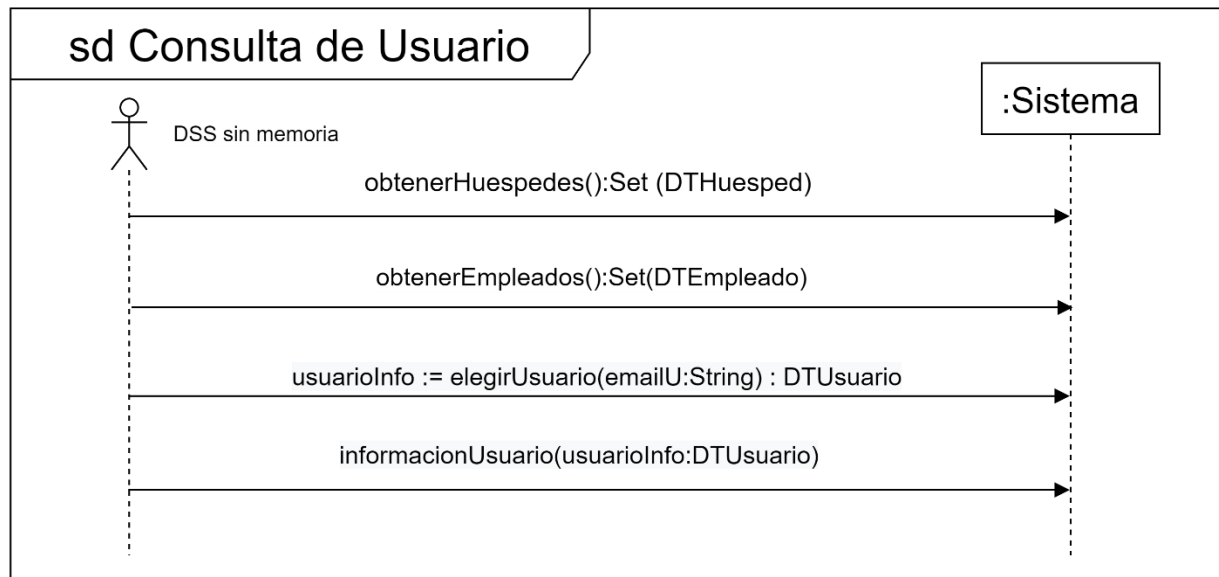
Precondiciones y Postcondiciones

Pre1: No hay

Post1: Se agrega una respuesta al DTCalificacion *calif*

Consulta de Usuario

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHuespedes()` está descrita en la sección Realizar Reserva.

Nombre	Obtener empleados
Operación	<i>obtenerEmpleados():Set(DTEmpleados)</i>
Entrada	No hay
Salida	Un conjunto de instancias de DTEmpleado
Descripción	Retorna todos los DTEmpleado en el Sistema

Precondiciones y Postcondiciones

Pre1: Existe en el sistema por lo menos un empleado.

Post1: Se devuelve un conjunto de instancias DTEmpleado dentro del sistema

Nombre	Elegir usuario
Operación	<i>elegirUsuario(emailU:string):DTUsuario</i>
Entrada	<i>emailU</i> representa el email de un usuario
Salida	Retorna un DTUsuario guardado en el sistema de email igual a <i>emailU</i>
Descripción	Se devuelve un DTUsuario de email <i>emailU</i>

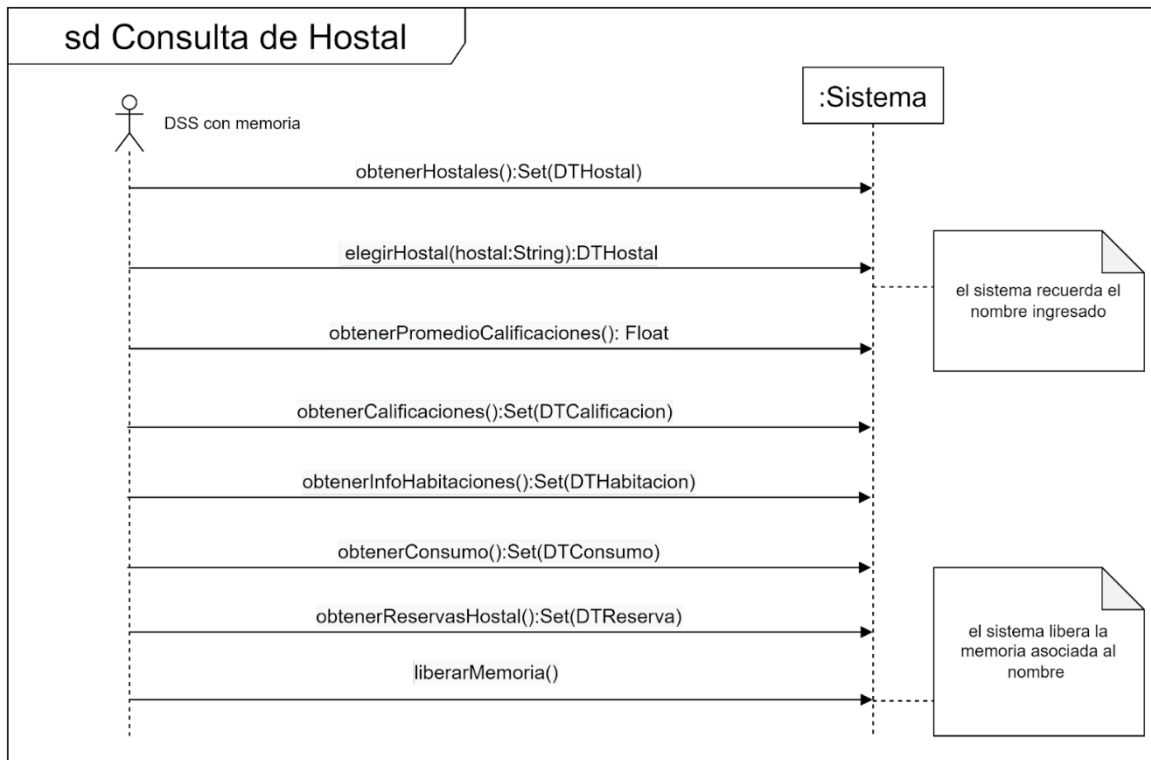
Precondiciones y Postcondiciones
Pre1: Existe un usuario en el sistema de email <i>emailU</i>
Post1: Se retorno una instancia de DTUsuario con atributo email igual a <i>emailU</i>

Nombre	Información usuario
Operación	<i>informacionUsuario(usuarioInfo:DTUsuarios)</i>
Entrada	<i>usuarioInfo</i> representa una instancia de tipo DTUsuario
Salida	Retorna la información de una empleado o huésped
Descripción	Se devuelve la información completa de un usuario. En caso de que sea un huésped se devuelve el campo de si es o no finger, en caso de que sea un empleado se devuelve el cargo y el lugar donde trabaja.

Precondiciones y Postcondiciones
Post1: Se retornó la información de una instancia de DT

Consulta de Hostal

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHostales()` está descrita en la sección Alta de Habitación.

La operación `elegirHostal()` está descrita en la sección Alta de Habitación.

Nombre	Promedio de calificaciones
Operación	<i>obtenerPromedioCalificaciones():Float</i>
Entrada	No hay
Salida	Retorna el promedio de todas calificaciones
Descripción	Devuelve un promedio de las calificaciones de un hostel

Precondiciones y Postcondiciones

Pre1: Existe en el sistema el nombre de un hostel recordado

Post1: se retorna el promedio de las calificaciones de hos

Nombre	Obtener calificaciones
Operación	<i>obtenerCalificaciones():Set(DTCalificacion)</i>
Entrada	No hay
Salida	Devuelve un conjunto de calificaciones de tipo DTCalificacion
Descripción	Devuelve un conjunto de calificaciones que contienen los comentarios y calificaciones de cada habitación

Precondiciones y Postcondiciones

Pre1: Existe en el sistema el nombre de un hostel recordado

Post1: Retorna un conjunto de tipo DTCalificacion con la información de cada calificación asociada a hos

Nombre	Obtener información habitaciones
Operación	<i>obtenerInfoHabitaciones():Set(DTHabitacion)</i>
Entrada	No hay
Salida	Devuelve un conjunto de DTHabitacion con todas las habitaciones de <i>hos</i>
Descripción	Retorna un conjunto de habitaciones con la información de cada una de ellas

Precondiciones y PostcondicionesPre1: Existe en el sistema el nombre de un hostel recordado por la variable *hos*.Post1: Se retorna un conjunto de tipo DTHabitacion que contiene la información de todas las habitaciones que pertenecen a *hos*

La operación obtenerConsumo() está descrita en la sección Alta de Habitación.

Nombre	Obtener reservas de hostel
Operación	obtenerReservasHostal():Set(DTReserva)
Entrada	No hay
Salida	Devuelve un conjunto de DTReserva con la información de las reservas asociadas a hos
Descripción	Retorna un conjunto de reservas de hos sin importar el estado en el que se encuentren, se devuelve la información básica de cada una el número de habitación y el o los huéspedes para cada reserva

Precondiciones y Postcondiciones

Post1: Se retorna un conjunto de tipo DTReserva que contiene la información de todas las reservas que pertenecen a hos

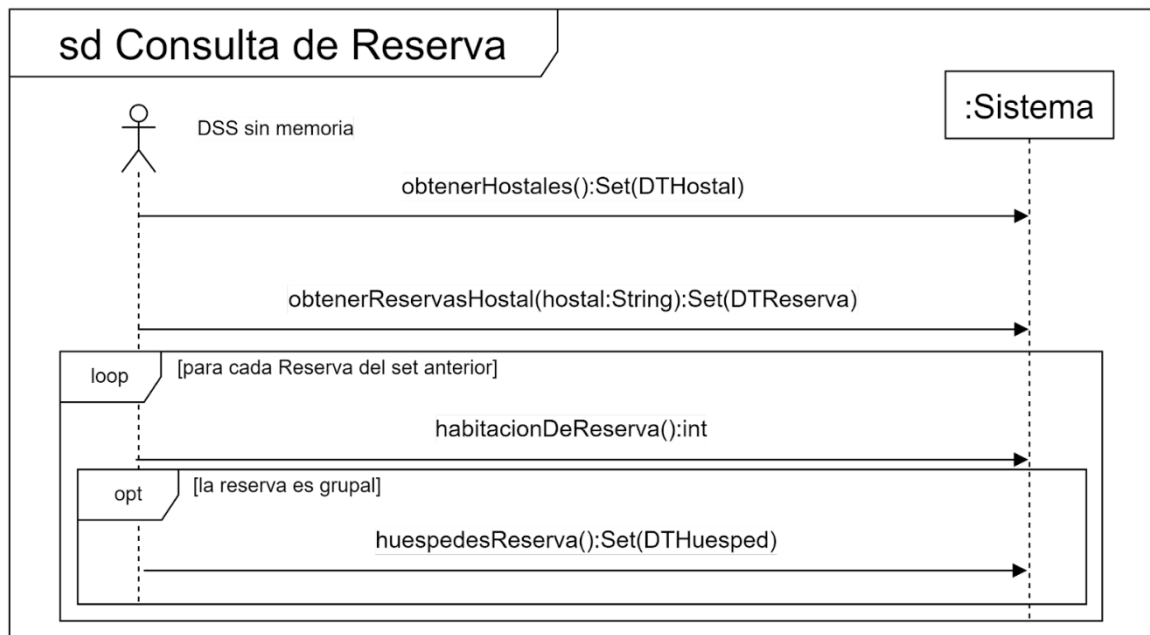
Nombre	Liberar memoria
Operación	liberarMemoria()
Entrada	No hay
Salida	No hay
Descripción	El sistema libera la memoria guardada hasta este punto

Precondiciones y Postcondiciones

Post1: se libero la memoria asignada al sistema

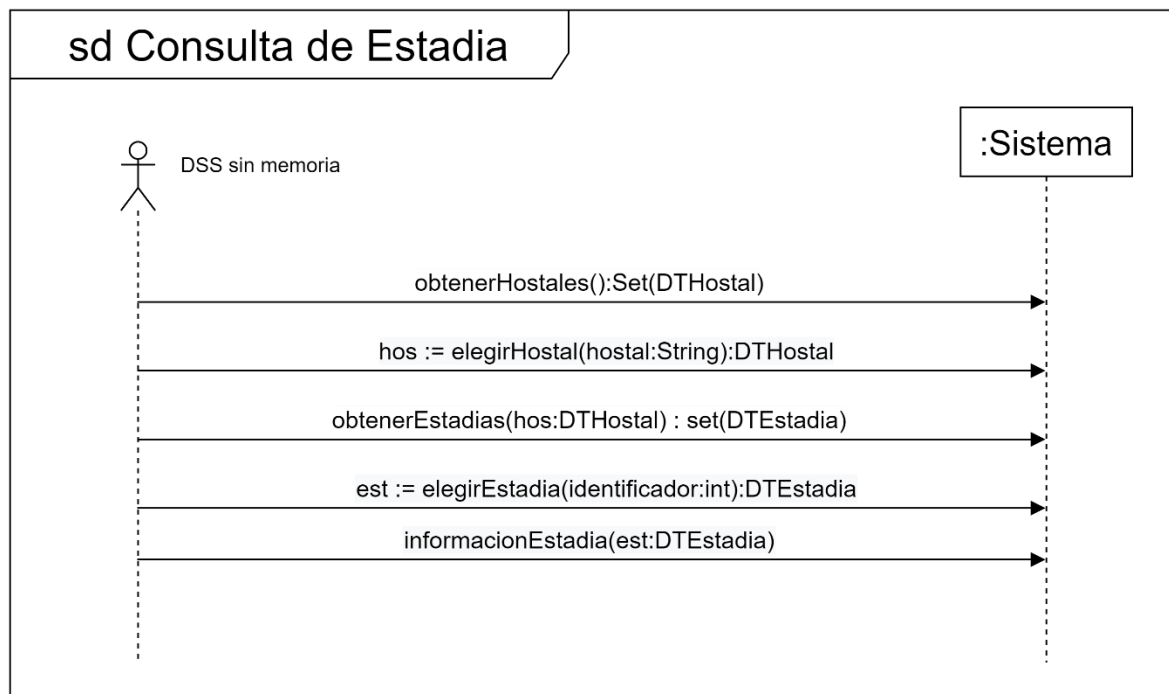
Consulta de Reserva

Diagramas de Secuencia del Sistema



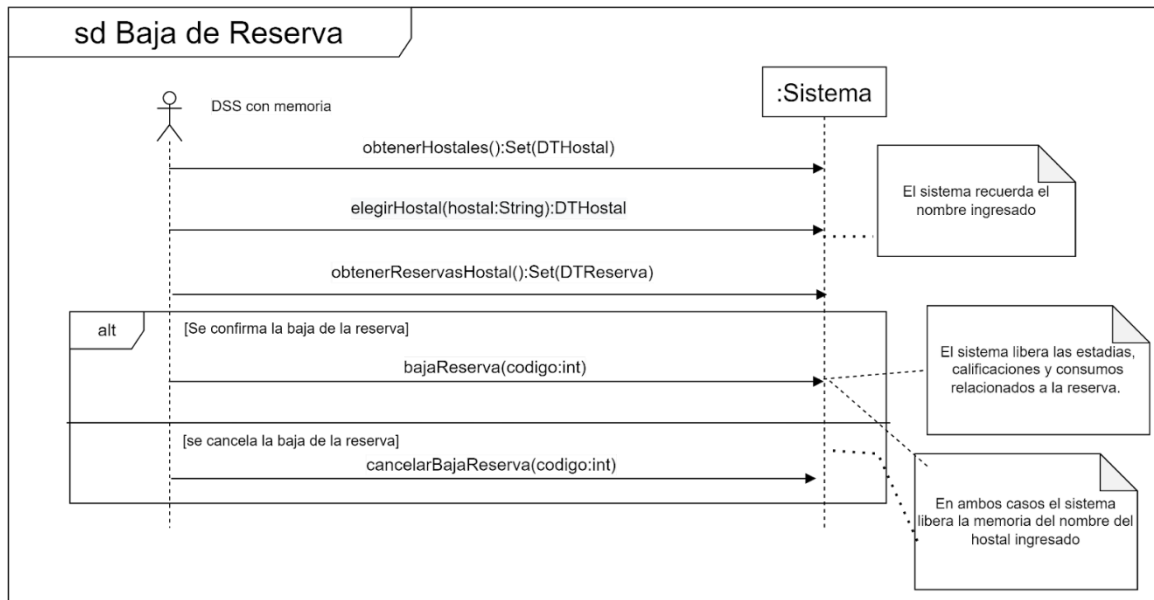
Consulta de Estadía

Diagramas de Secuencia del Sistema



Baja de Reserva

Diagramas de Secuencia del Sistema



Contratos

La operación `obtenerHostales()` está descrita en la sección Alta de Habitación.

La operación `elegirHostal()` está descrita en la sección Alta de Habitación.

La operación `obtenerReservasHostal()` está descrita en la sección Consulta de Hostal.

Nombre	Baja de reserva
Operación	<i>bajaReserva(codigo: Integer)</i>
Entrada	<i>código</i> representa el identificador de una reserva
Salida	No hay
Descripción	Da de baja la reserva de un hostel que está siendo recordado por el sistema

Precondiciones y Postcondiciones

Pre1: Existe una reserva que se identifique con *código*

Pre2: El Sistema recuerda el nombre de un hostel.

Post1: Se elimina una reserva del sistema junto con toda la información asociada a ella

Nombre	Cancelar Baja Reserva
Operación	cancelarBajaReserva(codigo:int)
Entrada	<i>codigo</i> representa el identificador de una reserva
Salida	No hay
Descripción	Se cancela la secuencia para dar de baja una reserva y se borran todos los datos ingresados hasta el momento

Precondiciones y Postcondiciones
<p>Pre1: Se está dando de baja una reserva identificada por <i>codigo</i>.</p> <p>Pre2: El Sistema recuerda el nombre de un hostel.</p> <p>Post1 : Se borran los datos ingresados y ya no se sigue con el procedimiento para dar de baja una reserva.</p>

CAMBIOS DEL MODELO DE DOMINIO Y RESTRICCIONES DEL LAB1

Cambios del modelo de dominio

Se agregó un identificador a las clases Estadías. Se agrega identificador a los DTCalificacion y DTEstadia.

Se agrega la asociación es Estadías [1___0..10]Consumo

Se agrega el campo hora:Int a las DTFecha