

CU02:

@startuml

actor Conserje

participant "UI - Gestionar Datos Personales" as UI

participant GestorHuesped as GestorHuesped

participant HuespedDAO as HuespedDAO

UI -> Conserje : se presenta la pantalla para completar datos

Conserje -> UI : se ingresan ninguno, uno o varios datos y presiona BUSCAR

UI -> GestorHuesped : se solicita lista de las personas

alt si no ingresa campos

GestorHuesped -> HuespedDAO: buscarTodosHuespedes

else ingresa campos

GestorHuesped -> HuespedDAO: buscarHuespedes (apellido, nombre, tipo , dni)

end

HuespedDAO -->GestorHuesped : listaHuespedes

GestorHuesped--> UI : listaHuespedes

opt si listaHuespedes es NULL

GestorHuesped -> UI : se ejecuta CU09

UI -> Conserje: navegar a CU09

end

UI --> Conserje : listaHuespedes

alt presiona siguiente sin seleccionar persona

UI -> Conserje : se ejecuta CU09

else

Conserje->UI : selecciona una persona y presiona boton SIGUIENTE

create Direccion

activate Direccion

GestorHuesped -> Direccion: <<create>>

GestorHuesped -> Direccion: setProvincia(huespedDTO.direccion.provincia)

GestorHuesped -> Direccion: setCalle(huespedDTO.direccion.calle)

GestorHuesped -> Direccion: setNumero(huespedDTO.direccion.numero)

GestorHuesped -> Direccion: setDepartamento(huespedDTO.direccion.departamento)

GestorHuesped -> Direccion: setPisoProvincia(huespedDTO.direccion.piso)

GestorHuesped -> Direccion: setCodigoPostal(huespedDTO.direccion.codigoPostal)

GestorHuesped -> Direccion: setLocalidad(huespedDTO.direccion.localidad)

GestorHuesped -> Direccion: setPais(huespedDTO.direccion.pais)

deactivate Direccion

create Huesped

activate Huesped

GestorHuesped -> Huesped: <<create>>

GestorHuesped -> Huesped : setApellido(DTO.apellido)

GestorHuesped -> Huesped : setNombre(DTO.nom)

GestorHuesped -> Huesped : setTipodoc(DTO.tipo)

GestorHuesped -> Huesped : setNumdoc(DTO.doc)

GestorHuesped -> Huesped : setFechaNac(DTO.fecha)

GestorHuesped -> Huesped : setTelefono(DTO.telefono)

GestorHuesped -> Huesped : setCuit(DTO.cuit)

GestorHuesped -> Huesped : setPosiva(DTO.posiva)
GestorHuesped -> Huesped : setDirec(direccion)
GestorHuesped -> Huesped : setEmail(DTO.email)
GestorHuesped -> Huesped : setOcupacion(DTO.ocupac)
GestorHuesped -> Huesped : setNacionalidad(DTO.nac)
Huesped -> GestorHuesped : Huesped
deactivate Huesped
GestorHuesped -> UI : Huesped
UI->Conserje : seejecutaCU10(Huesped)
end
@enduml

CU04:

@startuml
actor Conserje
participant "UI - Reservar Habitacion" as UI
participant "Gestor Reserva" as GestorReserva
participant "Reserva DAO" as ReservaDAO
Conserje -> UI : Se desea gestionar una reserva
activate UI '/activo aca la barra'/
UI -> Conserje: Se ejecuta CU05 mostrando estado Habitaciones
opt no existen habitaciones con las comodidades
UI->Conserje : mostrar mensaje "no existen habitaciones con las comodidades"
end
loop mientras el usuario no presione aceptar

Conserje -> UI : selecciona el periodo de la reserva
deactivate UI /'aca la desactvio '/

UI -> UI: validar disponibilidad de habitacion
alt habitacion no disponible

UI -> Conserje : mostrar mensaje habitaciones no disponibles
else

UI->Conserje : pintar habitacion con color "reservado"
end
end

Conserje -> UI: Aceptar()

UI -> GestorReserva : generarListadoReservar(fechaDesde,
fechaHasta,listaHabitacionesDTO)

GestorReserva -> UI : listado

UI -> Conserje : Se muestran datos reserva y botones ACEPTAR CANCELAR

Conserje -> UI : selecciona una opcion
/'

UI -> GestorReserva : reservarHabitacion()

activate GestorReserva

GestorReserva -> ReservaDAO : reservarHabitacion()

deactivate GestorReserva

activate ReservaDAO

ReservaDAO -> UI : datos reserva (tipo ingreso egreso)

deactivate ReservaDAO

UI -> Conserje : Se muestran datos reserva y botones aceptar cancelar

Conserje -> UI : selecciona una opcion

'/

opt opcion = cancelar

UI -> Conserje : se deshacen operaciones de la grilla

end

UI -> Conserje : se muestra "reserva a nombre de apellido nombre tel"

loop hasta que los datos esten correctos

Conserje -> UI : se completan y confirman los datos

UI -> UI: validar disponibilidad de habitacion

opt actor no completa un campo

UI -> Conserje : se muestra cartel de error

end

end

UI -> GestorReserva : reservarHabitacion(listaReservaDTO)

loop para cada reserva de listaReservaDTO

create Reserva

GestorReserva -> Reserva: <<create>>

GestorReserva -> Reserva: setNombre(reservaDTO.nombre)

GestorReserva -> Reserva: setApellido(reservaDTO.apellido)

GestorReserva -> Reserva: setTelefono(reservaDTO.telefono)

GestorReserva -> Reserva: setFechaDesde(reservaDTO.fechaDesde)

GestorReserva -> Reserva: setFechaHasta(reservaDTO.fechaHasta)

GestorReserva -> Reserva: setEstado("pendiente")

GestorReserva -> Reserva: setHabitacion(h)

GestorReserva -> ReservaDAO : persistirReserva(Reserva)

ReservaDAO --> GestorReserva : Reservas creada exitosamente

end

GestorReserva --> UI : Reservas creada exitosamente
UI --> Conserje : Mostrar mensaje "Reservas creada exitosamente"
@enduml

CU05:

```
@startuml
actor Conserje
participant "UI - Mostrar Estado de Habitaciones" as UI
participant "Gestor Habitacion" as GestorHabitacion
participant "Habitacion DAO" as HabitacionDAO
participant "Habitacion" as Habitacion

Conserje -> UI : desea verifi car el estado de las habitaciones
activate UI
UI -> Conserje : mostrar campos "Desde fecha" y "Hasta fecha"
loop mientras datos ingresados sean incorrectos
    Conserje -> UI : ingresa "desdeFecha"
    UI -> UI : validar desdeFecha
    opt fecha inválida
    UI -> Conserje : mostrar error
    deactivate UI
end
Conserje -> UI : ingresa "hastaFecha"
UI -> UI : validar hastaFecha
opt fecha inválida
UI -> Conserje : mostrar error
deactivate UI
```

```
end

end

opt Conserje presiona "CANCELAR"

UI -> Conserje : mensaje "¿Desea cancelar?"

alt confi rma SI

    UI -> Conserje : finalizar caso de uso

    deactivate UI

    else confi rma NO

        UI -> Conserje : volver a campos de fecha

        deactivate UI

    end

end

Conserje -> UI : presiona "ACEPTAR"

UI -> GestorHabitacion : obtenerEstadoHabitaciones(desdeFecha, hastaFecha)

activate GestorHabitacion

GestorHabitacion -> HabitacionDAO : read() // traer todas las habitaciones

activate HabitacionDAO

HabitacionDAO --> GestorHabitacion : datosHabitaciones[]

deactivate HabitacionDAO

loop por cada datoDeHabitacion

create Habitacion

GestorHabitacion -> Habitacion : <<create>>

activate Habitacion

GestorHabitacion -> Habitacion : setNumero(DTO.numero)

GestorHabitacion -> Habitacion : setEstado(DTO.estado)

GestorHabitacion -> Habitacion : setPrecio(DTO.precio)
```

```
GestorHabitacion -> Habitacion : setCapacidadPersonas(DTO.capacidad)
deactivate Habitacion

GestorHabitacion -> GestorHabitacion :
calcularEstadoYRetornarColumna(habitacion,fechaDesde,fechaHasta)
end

GestorHabitacion --> UI : grillaEstadoHabitaciones[]
deactivate GestorHabitacion

UI -> Conserje : mostrar grilla con habitaciones y reservas
deactivate UI

@enduml
```

CU09:

```
@startuml

actor Conserje

participant "UI - Alta Huesped" as UI

participant "Gestor Huesped" as GestorHuesped

participant "Huesped DAO" as HuespedDAO

Conserje -> UI : Se desea dar de alta un huésped nuevo
activate UI /'activo la UI'/

UI -> Conserje : Mostrar pantalla en blanco para carga de datos
loop hasta que los datos estén correctos y el usuario quiera corregir

Conserje -> UI : ingresa datos

UI -> Conserje

opt Conserje elige "CANCELAR"

Conserje -> UI : Cancelar()
```

UI -> Conserje : presentar confirmación "¿Desea cancelar el alta del huésped?"

alt actor elige "SI"

Conserje -> UI : SI

UI -> Conserje : El sistema finaliza el CU.

deactivate UI

else actor elige "NO"

Conserje -> UI : NO

UI -> Conserje : Regresa a la pantalla de carga mostrando los datos ya ingresados

deactivate UI

end

end

Conserje -> UI: presiona "SIGUIENTE"

UI -> UI : validar datos obligatorios y formato

opt faltan campos obligatorios

UI -> Conserje : mostrar mensaje de error detallando las omisiones y poner foco en los campos faltantes

deactivate UI /'vuelve a editar'/

end

UI -> GestorHuesped : verifi carExistenciaHuesped(huespedDTO.DNI)

GestorHuesped -> HuespedDAO : verifi carExistenciaHuesped(huespedDTO.DNI)

HuespedDAO --> GestorHuesped: huespedDTO

GestorHuesped --> UI: huespedDTO

opt huespedDTO !== null // ya existe el tipo y numero de documento

UI -> Conserje : mostrar cartel "¡CUIDADO! El tipo y número de documento ya existen" y botones "ACEPTAR IGUALMENTE", "CORREGIR"

alt Conserje elige "ACEPTAR IGUALMENTE"

Conserje -> UI : AceptarIgualmente()

UI -> GestorHuesped : registrarHuesped(huespedDTO)

create Huesped

GestorHuesped -> Huesped : <<create>>

Huesped -> GestorHuesped

activate Huesped

GestorHuesped -> Huesped : setApellido(huespedDTO.apellido)

GestorHuesped -> Huesped : setNombre(huespedDTO.nombre)

GestorHuesped -> Huesped : setTipoDocumento(huespedDTO.tipoDoc)

GestorHuesped -> Huesped : setNumeroDocumento(huespedDTO.numDoc)

GestorHuesped -> Huesped : setCUIL(huespedDTO.cuil)

GestorHuesped -> Huesped : setDireccion(huespedDTO.direccion)

GestorHuesped -> Huesped : setFechaNacimiento(huespedDTO.fechaNacimiento)

GestorHuesped -> Huesped : setTelefono(huespedDTO.telefono)

GestorHuesped -> Huesped : setEmail(huespedDTO.email)

deactivate Huesped

create Direccion

GestorHuesped -> Direccion: <<create>>

Direccion --> GestorHuesped

activate Direccion

GestorHuesped -> Direccion: setProvincia(huespedDTO.direccion.provincia)

GestorHuesped -> Direccion: setCalle(huespedDTO.direccion.calle)

GestorHuesped -> Direccion: setNumero(huespedDTO.direccion.numero)

GestorHuesped -> Direccion: setDepartamento(huespedDTO.direccion.departamento)

GestorHuesped -> Direccion: setPisoProvincia(huespedDTO.direccion.piso)

GestorHuesped -> Direccion: setCodigoPostal(huespedDTO.direccion.codigoPostal)

GestorHuesped -> Direccion: setLocalidad(huespedDTO.direccion.localidad)

GestorHuesped -> Direccion: setPais(huespedDTO.direccion.pais)

deactivate Direccion

'setear todos los otros campos

GestorHuesped -> Huesped:setDireccion(Direccion)

GestorHuesped --> UI : confirmación de alta

deactivate GestorHuesped

UI -> Conserje : "El huésped [apellido, nombre] ha sido cargado satisfactoriamente.\n¿Desea cargar otro?"

activate UI

Conserje -> UI : selecciona una opción

alt Conserje elige "NO"

UI -> Conserje : finalizar caso de uso

deactivate UI

else Conserje elige "SI"

UI -> Conserje : volver a pantalla inicial de carga (limpia o con campos listos)

deactivate UI

end

else Conserje elige "CORREGIR"

Conserje -> UI : Corregir()

UI -> Conserje : foco en campo "Tipo de documento"

end

end

end

' Si el flujo no terminó por cancelar, se procede a persistir

Conserje -> UI : presiona "SIGUIENTE"

activate UI

UI -> GestorHuesped : registrarHuesped(huespedDTO)

activate GestorHuesped

GestorHuesped -> Huesped : <<create>>

Huesped -> GestorHuesped

activate Huesped

GestorHuesped -> Huesped : setApellido(huespedDTO.apellido)

GestorHuesped -> Huesped : setNombre(huespedDTO.nombre)

GestorHuesped -> Huesped : setTipoDocumento(huespedDTO.tipoDoc)

GestorHuesped -> Huesped : setNumeroDocumento(huespedDTO.numDoc)

GestorHuesped -> Huesped : setCuil(huespedDTO.cuil)

GestorHuesped -> Huesped : setDireccion(huespedDTO.direccion)

GestorHuesped -> Huesped : setFechaNacimiento(huespedDTO.fechaNacimiento)

GestorHuesped -> Huesped : setTelefono(huespedDTO.telefono)

GestorHuesped -> Huesped : setEmail(huespedDTO.email)

deactivate Huesped

GestorHuesped -> Direccion: <<create>>

Direccion --> GestorHuesped

activate Direccion

GestorHuesped -> Direccion: setProvincia(huespedDTO.direccion.provincia)

GestorHuesped -> Direccion: setCalle(huespedDTO.direccion.calle)

GestorHuesped -> Direccion: setNumero(huespedDTO.direccion.numero)

GestorHuesped -> Direccion: setDepartamento(huespedDTO.direccion.departamento)

GestorHuesped -> Direccion: setPisoProvincia(huespedDTO.direccion.piso)

GestorHuesped -> Direccion: setCodigoPostal(huespedDTO.direccion.codigoPostal)

GestorHuesped -> Direccion: setLocalidad(huespedDTO.direccion.localidad)

GestorHuesped -> Direccion: setPais(huespedDTO.direccion.pais)

```

deactivate Direccion

'setear todos los otros cmapos

GestorHuesped -> Huesped:setDireccion(Direccion)

GestorHuesped --> UI : confi rmación de alta

deactivate GestorHuesped

UI -> Conserje : "El huésped [apellido, nombre] ha sido cargado
satisfactoriamente.\n¿Desea cargar otro?"

Conserje -> UI : selecciona una opción

alt Conserje elige "NO"

UI -> Conserje : volver a pantalla principal

else Conserje elige "SI"

UI -> Conserje : volver a pantalla inicial de carga (limpia o con campos listos)

deactivate UI

end

@enduml

```

CU15:

```

@startuml

participant Conserje as Conserje

participant "UI-Ocupar Habitacion" as UIOcuparHabitacion

participant "Gestor Huesped" as GestorHuesped

participant "Gestor Estadia" as GestorEstadia

participant "Estadia-DAO" as EstadiaDAO

participant "Huesped-DAO" as HuespedDAO

'paso 1

```

Conserje -> UIOcuparHabitacion : Sigue la asignación de Habitaciones

loop mientras el usuario presione cargarOtraHabitacion()

'paso 2

UIOcuparHabitacion --> Conserje : ejecuta CU05

opt no existen habitaciones con las comodidades

UIOcuparHabitacion --> Conserje : mostrar mensaje "no existen habitaciones con las comodidades"

end

'paso 3

loop hasta que el usuario haga una selección válida o presione volver

Conserje -> UIOcuparHabitacion : selecciona Habitación

UIOcuparHabitacion -> UIOcuparHabitacion : validar el Estado de las Habitaciones

alt habitación seleccionada o rango de habitaciones con estado "ocupada"

UIOcuparHabitacion --> Conserje : mostrarMensaje("habitación ocupada, por favor seleccione otra")

'paso 3.D'

else habitación seleccionada tiene estado "reservada"

UIOcuparHabitacion --> Conserje : muestra por pantalla días y de quién es la reserva

UIOcuparHabitacion --> Conserje : muestra los botones "OCUPAR IGUAL" Y "VOLVER"

Conserje --> UIOcuparHabitacion : presionaBoton()

alt Botón="volver"

Conserje --> UIOcuparHabitacion : presionaVolver()

UIOcuparHabitacion --> Conserje : devuelveControl

else Botón="ocupar igual"

Conserje --> UIOcuparHabitacion : presionaOcuparIgual()

end

end

end

'paso 4

UIOcuparHabitacion --> Conserje : muestra habitaciones pintadas

UIOcuparHabitacion --> Conserje : muestra por pantalla "PRESIONE CUALQUIER TECLA Y CONTINUA"

'paso 5

Conserje --> UIOcuparHabitacion : presionaCualquierTecla()

loop mientras usuario presione seguirCargando()

'paso 6'

UIOcuparHabitacion --> Conserje : muestra pantalla para cargar datos de huespedes y acompañantes

loop mientras el usuario no ingrese ningun dato o los ingrese con formato invalido

Conserje --> UIOcuparHabitacion : Ingresa criterios de busqueda

UIOcuparHabitacion -> UIOcuparHabitacion : validar datos ingresados

opt datos son incorrectos

UIOcuparHabitacion -> Conserje : mostrarMensaje("los datos tienen formato invalido")

end

end

Conserje -> UIOcuparHabitacion : presionaAceptar()

'paso 7'

UIOcuparHabitacion -> GestorHuesped : se solicitan los Huespedes

GestorHuesped -> HuespedDAO : buscarHuespedes(apellido,nombre,tipodoc,dni)

HuespedDAO -> GestorHuesped : listaHuespedes

GestorHuesped -> UIOcuparHabitacion : grilla con informacion de huespedes

loop mientras el actor presione aceptar sin seleccionar un huesped

UIOcuparHabitacion --> Conserje : muestra grilla con datos de personas que cumplen los criterios

alt presiona Aceptar() sin especifi car huesped
Conserje -> UIOcuparHabitacion : presionaAceptar()
UIOcuparHabitacion -> Conserje : mostrarMensaje("seleccione al menos un huesped")
Conserje -> UIOcuparHabitacion : devuelveControl
else el actor selecciona al menos un huesped
'paso 8'
Conserje -> UIOcuparHabitacion : selecciona huesped y sus acompañantes, y
presionaAceptar()
end
end
' PASO 10'
UIOcuparHabitacion -> Conserje : muestra botones seguirCargando(),
cargarOtraHbitacion(), salir()
' PASO 11 '
alt presiona seguirCargando()
'sistema vuelve al punto 6'
Conserje -> UIOcuparHabitacion : presionaSeguirCargando()
else presiona cargarOtraHbitacion()
'vuelve al punto 2 ==> otro loop al 2'
Conserje -> UIOcuparHabitacion : presionaCargarOtraHabitacion()
else presiona salir()
'termina el CU'
Conserje -> UIOcuparHabitacion : presionaSalir()
end

end
UIOcuparHabitacion -> GestorEstadia : guardarEstadia(estadiaDTO)

create Estadia

GestorEstadia -> Estadia : <<create>>

Estadia -> GestorEstadia: devuelveControl

GestorEstadia -> Estadia: setHuespedes(estadiaDTO.huespedes)

GestorEstadia -> Estadia: setCheckIn(estadiaDTO.fecha)

GestorEstadia -> Estadia: setCheckOut(estadiaDTO.fecha)

GestorEstadia -> EstadiaDAO: persistirEstadia(Estadia)

EstadiaDAO -> GestorEstadia: estadia creada con exito

GestorEstadia -> UIOcuparHabitacion: estadia creada con exito

UIOcuparHabitacion -> Conserje: mostrarMensaje("estadia creada con exito")

end

'UIOcuparHabitacion -> GestorEstadia : crearEstadia(EstadiaDTO)

'UIOcuparHabitacion -> GestorHabitacion : actualizar estados de habitaciones

'GestorHabitacion -> HabitacionDAO : actualizar datos

@enduml