

Taller 1 - Corte 3

Link del repositorio https://github.com/JerssonF/AN-LISIS_DE_SISTEMAS/tree/main/Taller_1Corte_III

Instrucciones

Lee detenidamente el ejercicio asignado a continuación y procede a realizar las siguientes actividades:

1. Construye un diagrama de clases, un diagrama de secuencia y un diagrama de casos de us, según el ejercicio asignado.
2. Envía tu trabajo asignado con estricto cumplimiento del tiempo establecido.
3. Crea una carpeta en un repositorio público para cada ejercicio y proporciona el enlace junto con los archivos en la plataforma Moodle.
4. En el repositorio Git, asegúrate de previsualizar cada uno de los diagramas y el código utilizado para crearlos.
5. Para esto último crea un archivo ;arkdown donde hagas referencia a cada uno de los diagramas contruidos y su respectivo código en UMLPlant.

Ejercicio 2 : Sistema de Reservas de Hotel

RF1: Reserva de Habitaciones

- El sistema debe permitir a los usuarios buscar habitaciones disponibles por fechas específicas.
- Debe mostrar información detallada sobre cada habitación disponible, incluyendo tarifas, comodidades y ubicación.
- Debe permitir a los usuarios seleccionar la habitación deseada y proporcionar detalles de contacto para la reserva.
- Debe confirmar la reserva inmediatamente después de completar el proceso de recserva.
- Debe enviar una confirmación por correo electrónico al usuario con los detalles de la reserva.

RF2: Gestión De Reservas

- El sistema debe permitir a los usuarios ver y administrar sus reservas existentes.
- Debe proporcionar opciones para modificar fechas de reserva, número de habitaciones y servicios adicionales.
- Debe permitir la cancelación de reservas dentro de un plazo especificado sin cargos adicionales.
- Debe enviar recordatorios automáticos antes de la fecha de llegada para confirmar la reserva.
- Debe generar facturas detalladas para cada reserva realizada.

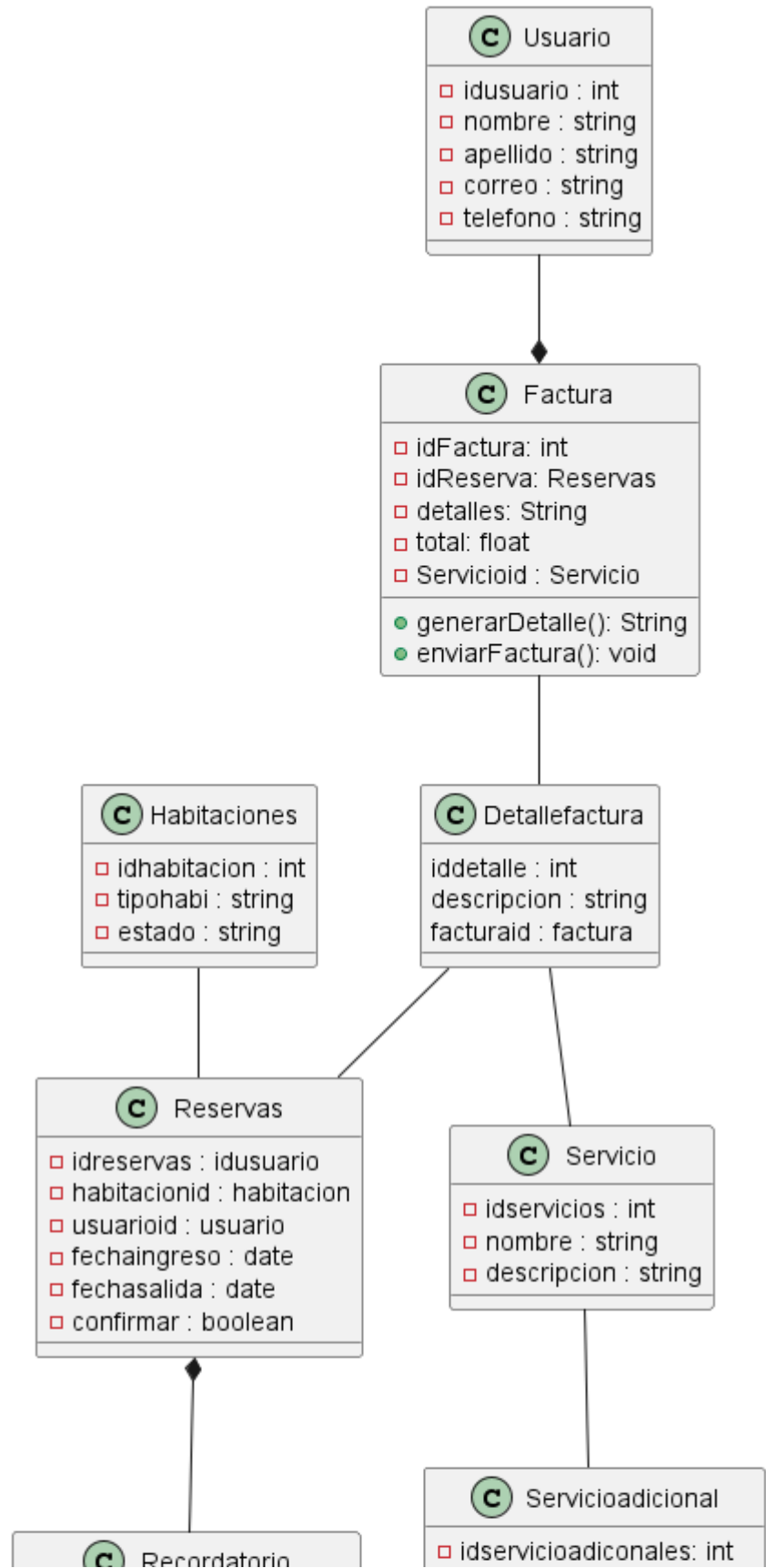
RF3: Servicios de conserjería

- El sistema debe ofrecer servicios de conserjería adicionales, como reservas en restaurantes o transporte local.
- Debe mostrar información actualizada sobre los servicios disponibles en la ubicación del hotel.
- Debe permitir a los usuarios solicitar servicios de conserjería como parte de su reserva o posteriormente.

- Debe confirmar las solicitudes de servicios de conserjería y proporcionar detalles adicionales si es necesario.

Diagramas

DIAGRAMA DE CLASES





Código Uml

@startuml Reserva

```
class Usuario {
- idusuario : int
- nombre : string
- apellido : string
- correo : string
- telefono : string
}

class Reservas {
- idreservas : idusuario
- habitacionid : habitacion
- usuarioid : usuario
- fechaingreso : date
- fechasalida : date
```

- confirmar : boolean

- }

class Habitaciones {

- idhabitacion : int

- tipohabi : string

- estado : string

- }

class Servicioadicional {

- idservicioadiconales: int

- serviciosid : Servicio

- nombre: String

- descripcion: String

- precio: float

- facturaid : factura

- + añadirAServicio(): void

- + eliminarDeServicio(): void

- }

class Factura {

- idFactura: int

- idReserva: Reservas

- detalles: String

- total: float

- Servicioid : Servicio

- + generarDetalle(): String

- + enviarFactura(): void

```
}
```

```
class Detallefactura {
```

```
-iddetalle : int
```

```
-descripcion : string
```

```
-facturaid : factura
```

```
}
```

```
class Recordatorio {
```

```
- idRecordatorio: int
```

```
- idReserva: int
```

```
- fechaEnvio: Date
```

```
+ enviarRecordatorio(): void
```

```
}
```

```
class Consenjeria {
```

```
-idconsejeria : id
```

```
-nombre : string
```

```
-descripcionconse : string
```

```
-precioconse : float
```

```
-reservasid : Reservas
```

```
-Consenjeriaid : Consenjeria
```

```
+ agregar_servicios(): void
```

```
+ confirmar_reserva(): void
```

```
}
```

```
class Servicio {
```

```
- idservicios : int
```

```
- nombre : string
```

- descripcion : string

}

Usuario --* Factura

Detallefactura -- Servicio

Factura -- Detallefactura

Detallefactura -- Reservas

Habitaciones -- Reservas

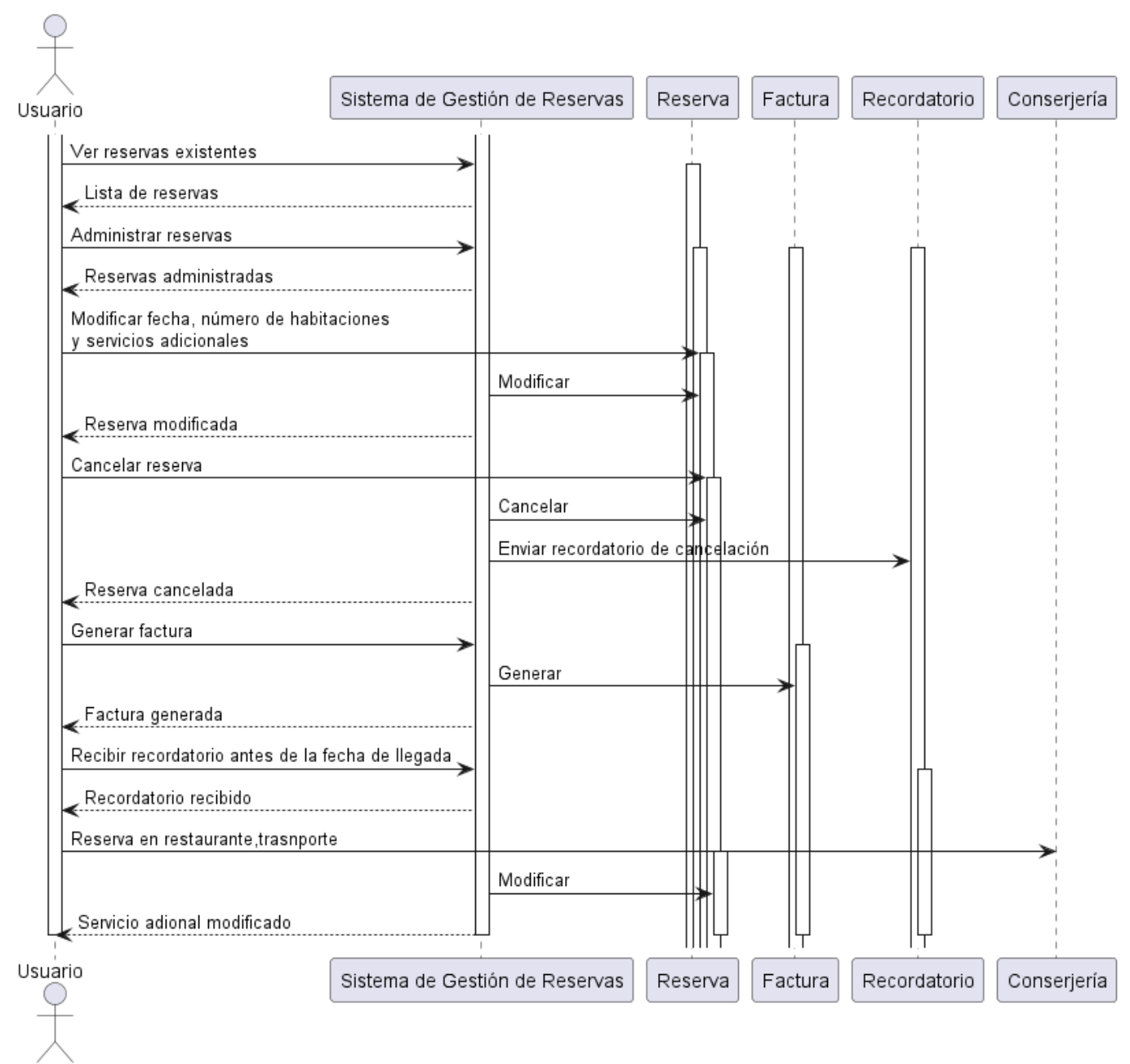
Servicio -- Servicioadicional

Reservas *-- Recordatorio

Servicioadicional -- Consenjeria

@enduml

DIAGRAMA DE SECUENCIA



Código Uml

@startuml

actor Usuario

participant "Sistema de Gestión de Reservas" as Sistema

participant "Reserva" as Reserva

participant "Factura" as Factura

participant "Recordatorio" as Recordatorio

participant "Conserjería" as Conserjería

activate Usuario

activate Sistema

Usuario -> Sistema: Ver reservas existentes

activate Reserva

Sistema --> Usuario: Lista de reservas

Usuario -> Sistema: Administrar reservas

activate Reserva

activate Factura

activate Recordatorio

Sistema --> Usuario: Reservas administradas

Usuario -> Reserva: Modificar fecha, número de habitaciones\ny servicios adicionales

activate Reserva

Sistema -> Reserva: Modificar

Sistema --> Usuario: Reserva modificada

Usuario -> Reserva: Cancelar reserva

activate Reserva

Sistema -> Reserva: Cancelar

Sistema -> Recordatorio: Enviar recordatorio de cancelación

Sistema --> Usuario: Reserva cancelada

Usuario -> Sistema: Generar factura

activate Factura

Sistema -> Factura: Generar

Sistema --> Usuario: Factura generada

Usuario -> Sistema: Recibir recordatorio antes de la fecha de llegada

activate Recordatorio

Sistema --> Usuario: Recordatorio recibido

Usuario -> Conserjería: Reserva en restaurante,trasnporte

activate Reserva

Sistema -> Reserva: Modificar

Sistema --> Usuario:Servicio adional modificado

deactivate Recordatorio

deactivate Factura

deactivate Reserva

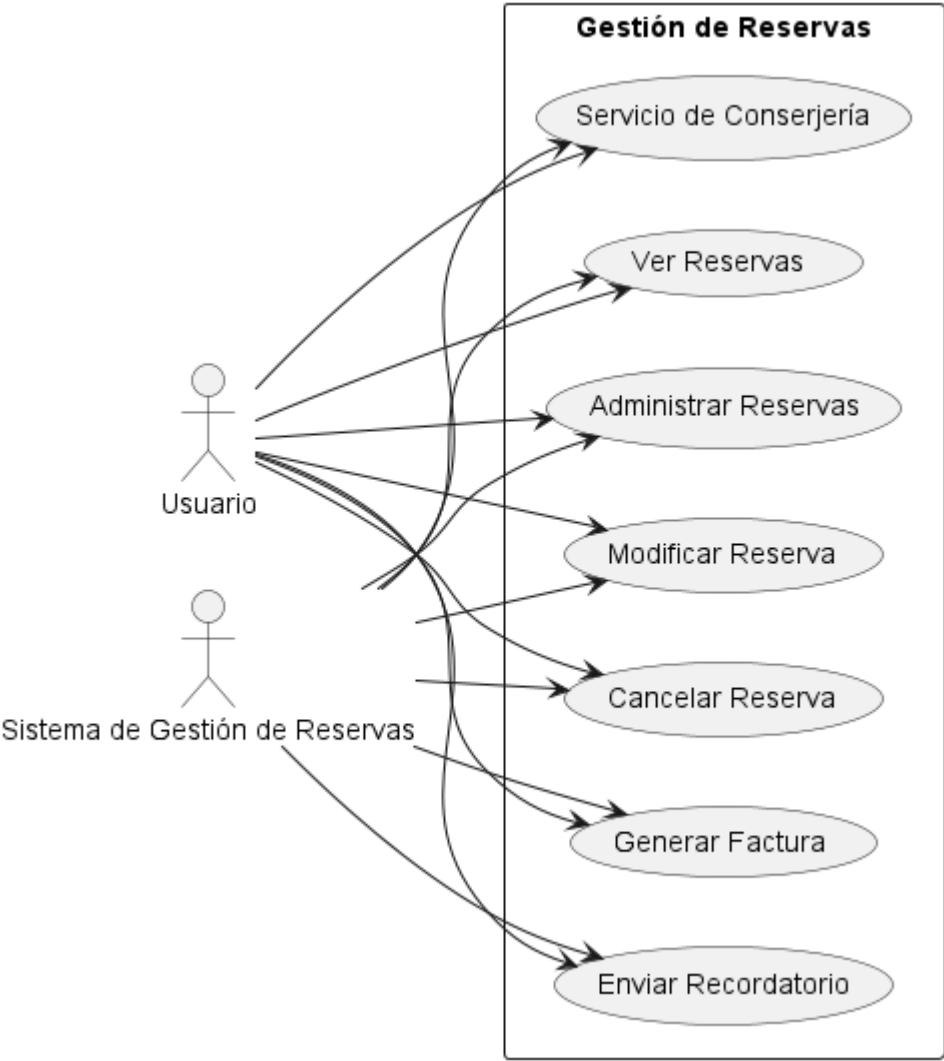
deactivate Sistema

deactivate Usuario

deactivate Conserjería

@enduml

DIAGRAMA DE CASOS DE USO



Código Uml

@startuml

left to right direction

actor Usuario as User

actor "Sistema de Gestión de Reservas" as System

rectangle "Gestión de Reservas" {

usecase "Ver Reservas" as VerReservas

usecase "Administrar Reservas" as AdminReservas

usecase "Modificar Reserva" as ModificarReserva

usecase "Cancelar Reserva" as CancelarReserva

usecase "Generar Factura" as GenerarFactura

```
usecase "Enviar Recordatorio" as EnviarRecordatorio

usecase "Servicio de Conserjería" as ServicioConserjeria

}

User --> GenerarFactura

User --> VerReservas

User --> AdminReservas

User --> ModificarReserva

User --> CancelarReserva

User --> EnviarRecordatorio

User --> ServicioConserjeria

System --> VerReservas

System --> AdminReservas

System --> ModificarReserva

System --> CancelarReserva

System --> GenerarFactura

System --> EnviarRecordatorio

System --> ServicioConserjeria

@enduml
```