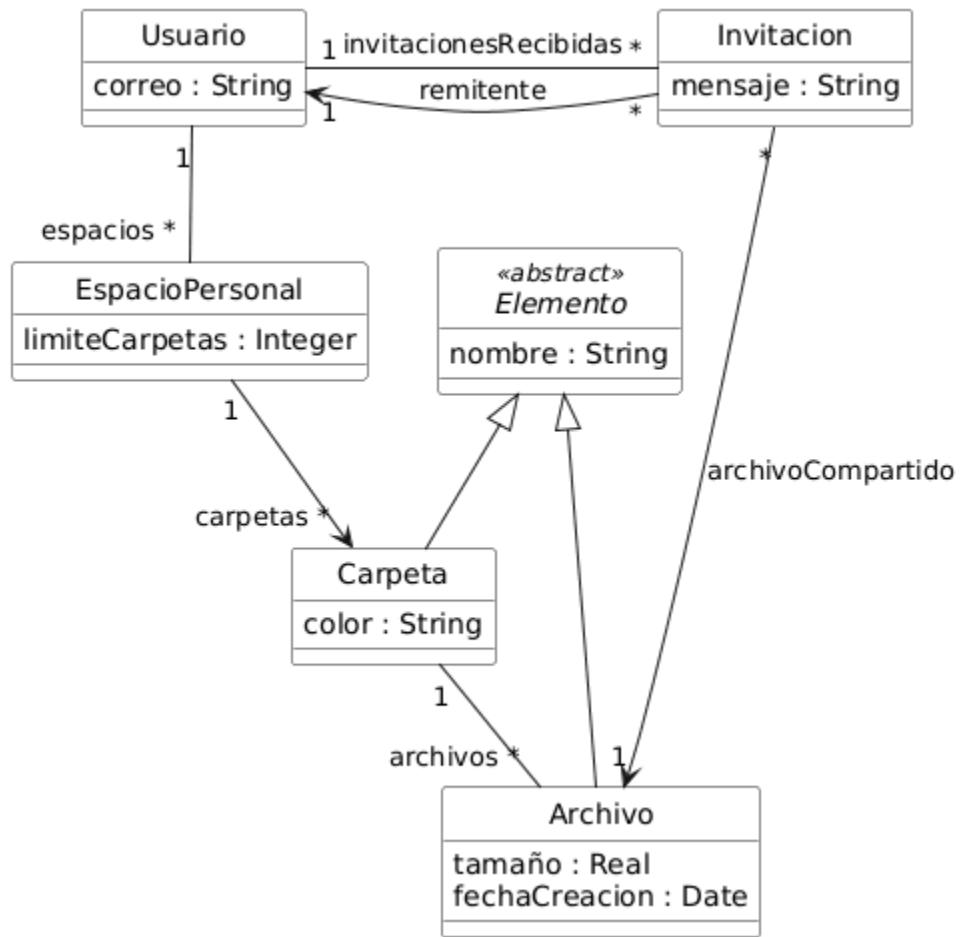
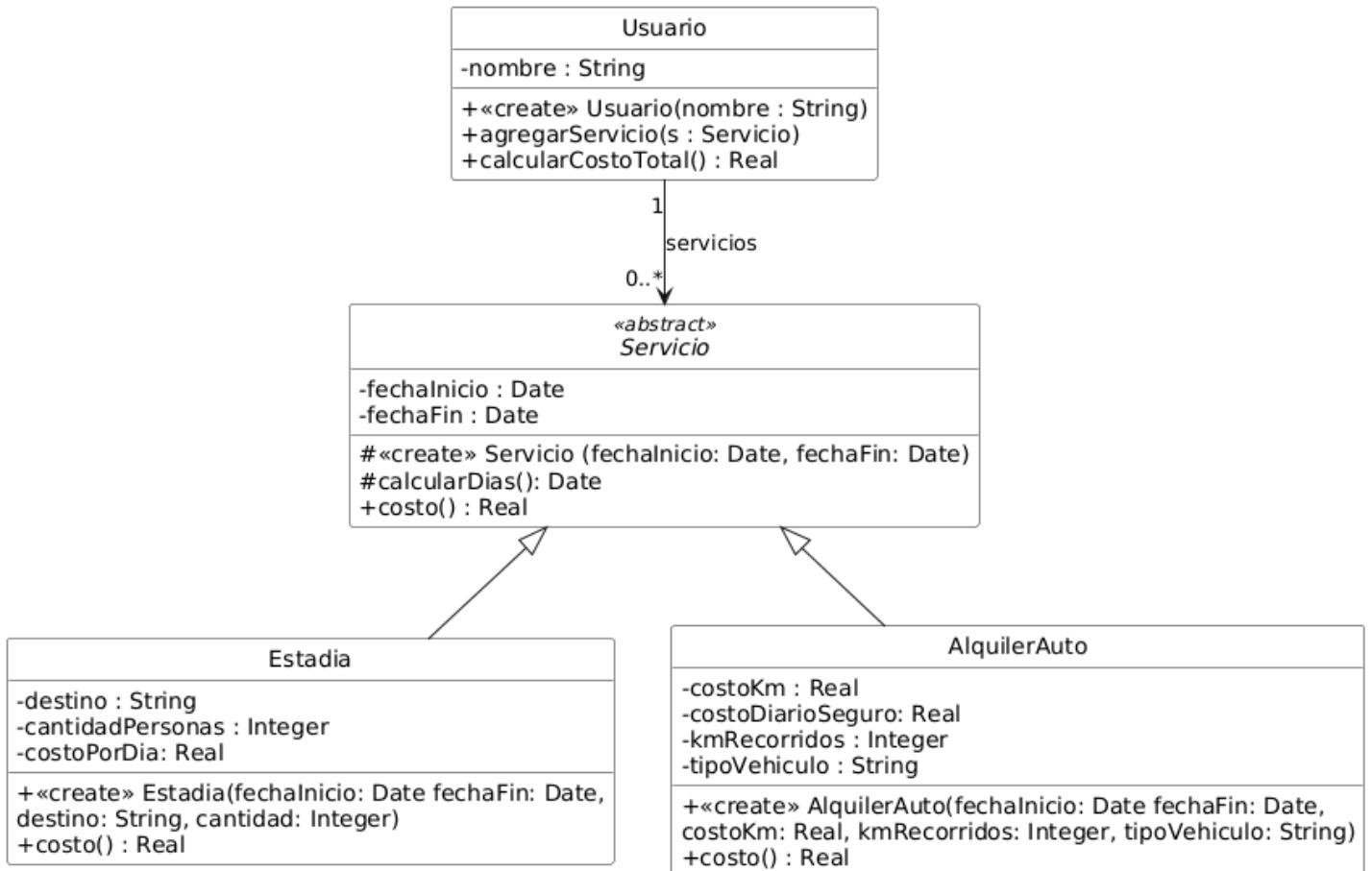


Solución posible
aceptada como correcta al examen parcial del 8/11/2025

Ejercicio 1)



Ejercicio 2)



```

public class Usuario {

    private String nombre;
    private List<Servicio> servicios;

    public Usuario(String nombre) {
        this.nombre = nombre;
        this.servicios = new ArrayList<>();
    }

    public void agregarServicio(Servicio s) {
        servicios.add(s);
    }

    public double calcularCostoTotal() {
        return servicios.stream()
            .mapToDouble(s -> s.costo())
            .sum();
    }
}

public abstract class Servicio {
    private LocalDate fechalinicio;
    private LocalDate fechaFin;
}
  
```

```
protected Servicio(LocalDate fechalinicio, LocalDate fechaFin) {  
    this.fechalinicio = fechalinicio;  
    this.fechaFin = fechaFin;  
}  
protected long calcularDias() {  
    return ChronoUnit.DAYS.between(fechalinicio, fechaFin);  
}  
public abstract double costo();  
}
```

```
public class Estadia extends Servicio {  
    private String destino;  
    private double costoPorDia;  
    private int cantidadPersonas;  
  
    public Estadia(LocalDate fechalinicio, LocalDate fechaFin,  
                   String destino, int cantidadPersonas, double costoPorDia) {  
        super(fechalinicio, fechaFin);  
        this.destino = destino;  
        this.cantidadPersonas = cantidadPersonas;  
        this.costoPorDia = costoPorDia;  
    }  
    protected double costo() {  
        return this.calcularDias() * costoPorDia * cantidadPersonas;  
    }  
}
```

```
public class AlquilerAuto extends Servicio {  
  
    private double costoKm;  
    private double kmRecorridos;  
    private double costoDiarioSeguro;  
    private String tipoVehiculo;  
  
    public AlquilerAuto(LocalDate fechalinicio, LocalDate fechaFin,  
                        double costoKm, double kmRecorridos, String tipoVehiculo, double  
costoDiarioSeguro) {  
        super(fechalinicio, fechaFin);  
        this.costoKm = costoKm;  
        this.kmRecorridos = kmRecorridos;  
        this.tipoVehiculo = tipoVehiculo;  
        this.costoDiarioSeguro = costoDiarioSeguro;  
    }  
  
    protected double costo() {  
        return (costoPorKm * kilometrosRecorridos)  
               + (costoDiarioSeguro * this.calcularDias());  
    }  
}
```

Ejercicio 4)

Solución 1: Un empleado siempre tiene un contrato.

```
public Empleado(String nombre, LocalDate fechaNacimiento, int cantidadHijos, boolean tieneConyuge, Contrato c) {
    this.nombre = nombre;
    this.fechaNacimiento = fechaNacimiento;
    this.cantidadHijos = cantidadHijos;
    this.tieneConyuge = tieneConyuge;
    this.contratos = new ArrayList<>();
    this.contratos.add(c);
}
```

Solución 2: En este caso, también aceptada. Un empleado no tiene contratos aún.

```
public Empleado(String nombre, LocalDate fechaNacimiento, int cantidadHijos, boolean tieneConyuge) {
    this.nombre = nombre;
    this.fechaNacimiento = fechaNacimiento;
    this.cantidadHijos = cantidadHijos;
    this.tieneConyuge = tieneConyuge;
    this.contratos = new ArrayList<>();
}
```

Instanciación utilizando el constructor de la solución 2

```
Empleado juan = new Empleado( "Juan Pérez", LocalDate.of(1985, 5, 10), 2, true);

ContratoDeHoras contratoHorasJuan = new ContratoDeHoras(LocalDate.of(2020, 6, 1), 900, 160,
LocalDate.of(2023, 1, 31));

juan.agregarContrato(contratoHorasJuan);

ContratoDePlanta contratoPlantaJuan = new ContratoDePlanta(LocalDate.of(2023, 2, 1), 120000, 6000,
10000);

juan.agregarContrato(contratoPlantaJuan);

Empleado ana = new Empleado( "Ana García", LocalDate.of(1990, 8, 15), 1, false);

ContratoDePlanta contratoPlantaAna = new ContratoDePlanta( LocalDate.of(2022, 7, 1),      80000, 0,
8000);

ana.agregarContrato(contratoPlantaAna);
```

```
ContratoDeHoras contratoHorasAna = new ContratoDeHoras(LocalDate.of(2024, 10, 1), 1100, 150,  
LocalDate.of(2025, 12, 15));
```

```
ana.agregarContrato(contratoHorasAna);
```

Diseño de los casos de test

Diseñe los casos de prueba para el método **calcularSueldoBasico** de **ContratoDePlanta** teniendo en cuenta los conceptos de valores de borde y particiones equivalentes vistos en la materia

Componentes en el cálculo del sueldo

- Antiguedad:
 - Identificamos dos particiones: si tiene más de 10 años de antigüedad o no.
 - Valores de borde: 9, 10, 11 años de antigüedad.
- Estado civil:
 - Identificamos dos particiones: si es casado o no.
- Cantidad de hijos:
 - Identificamos dos particiones: si tiene hijos o no.
 - Valores de borde: 0, 1 y 2 hijos.

Los casos a testar son las combinaciones de los valores de borde identificados.