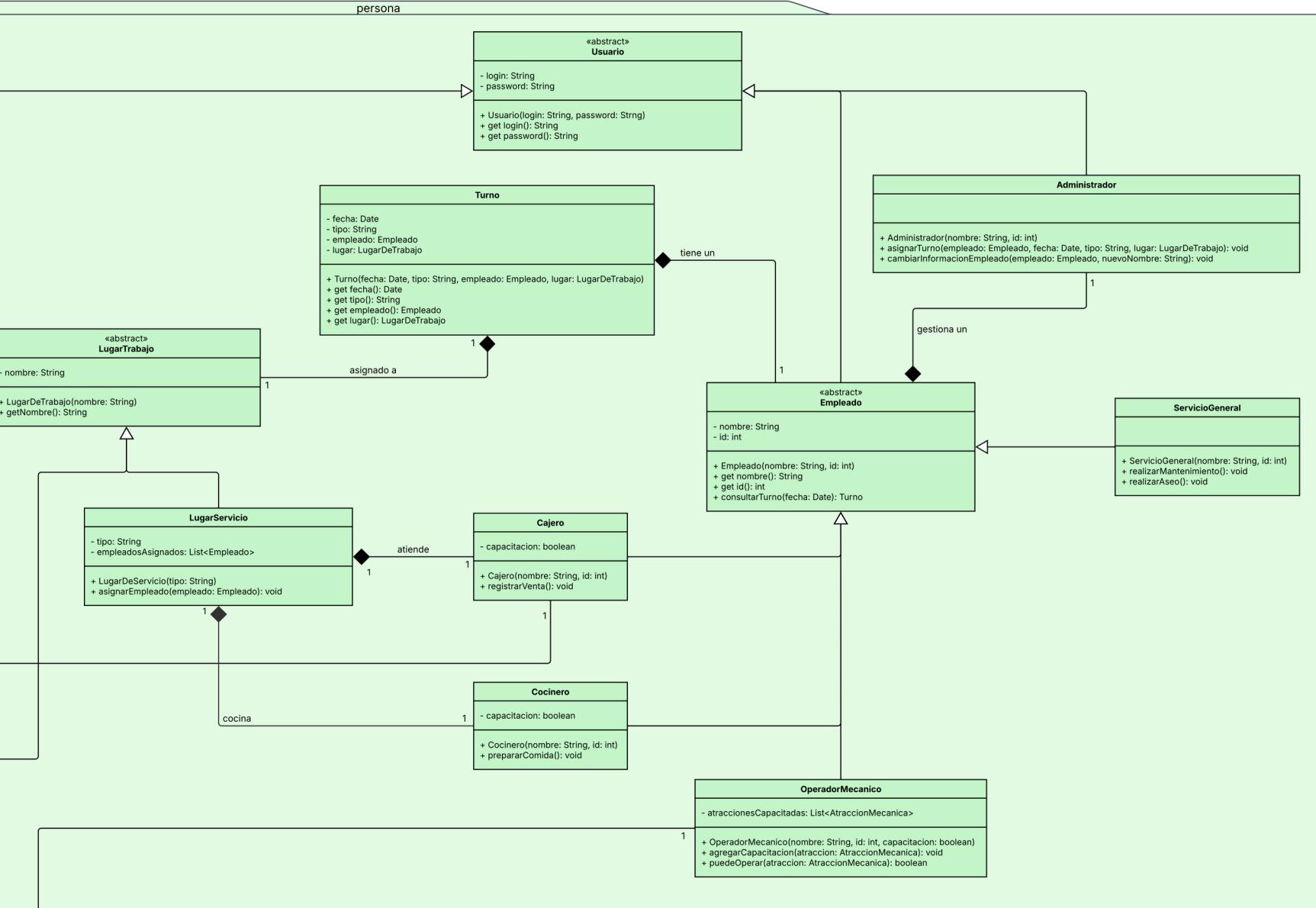
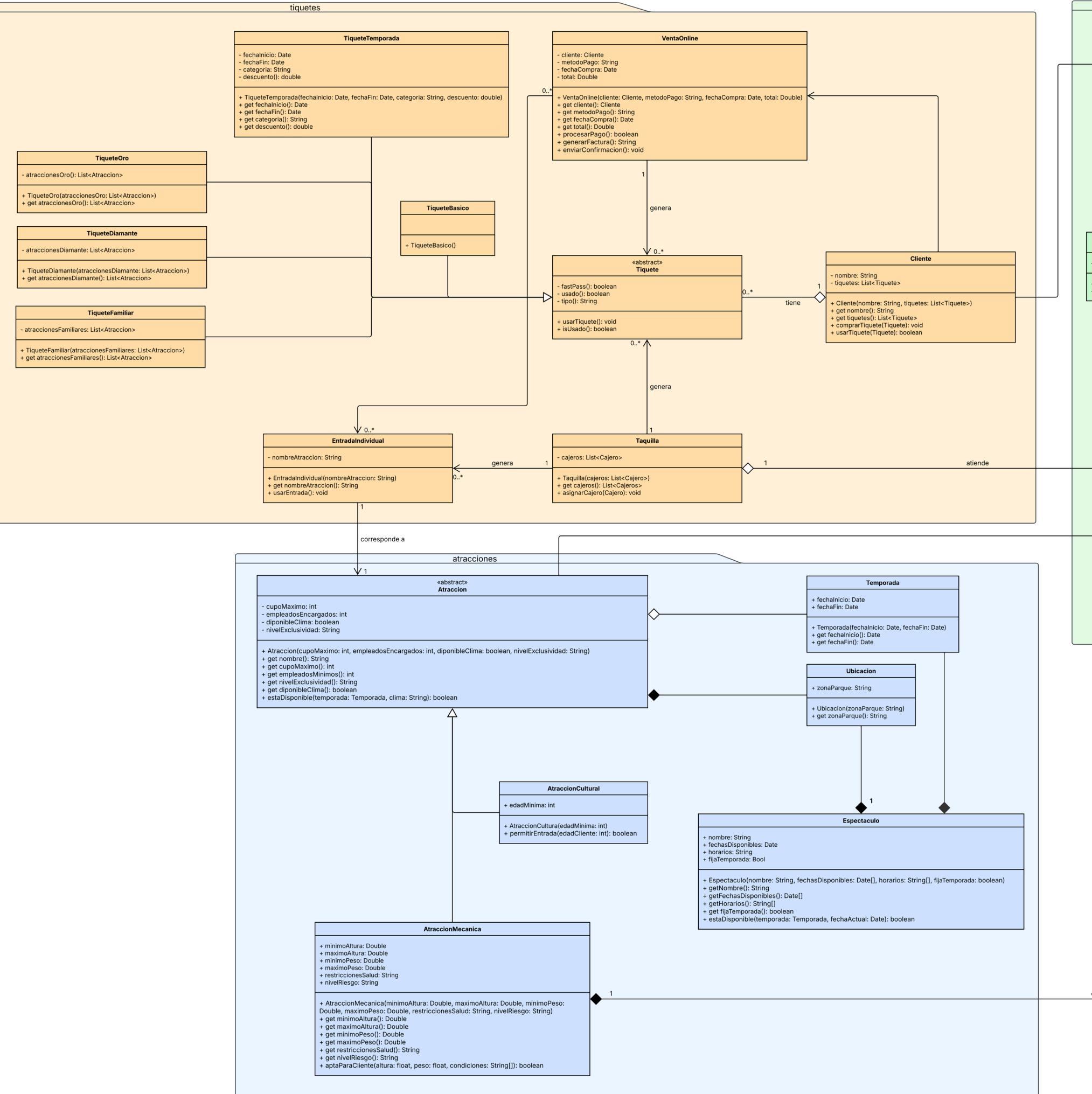


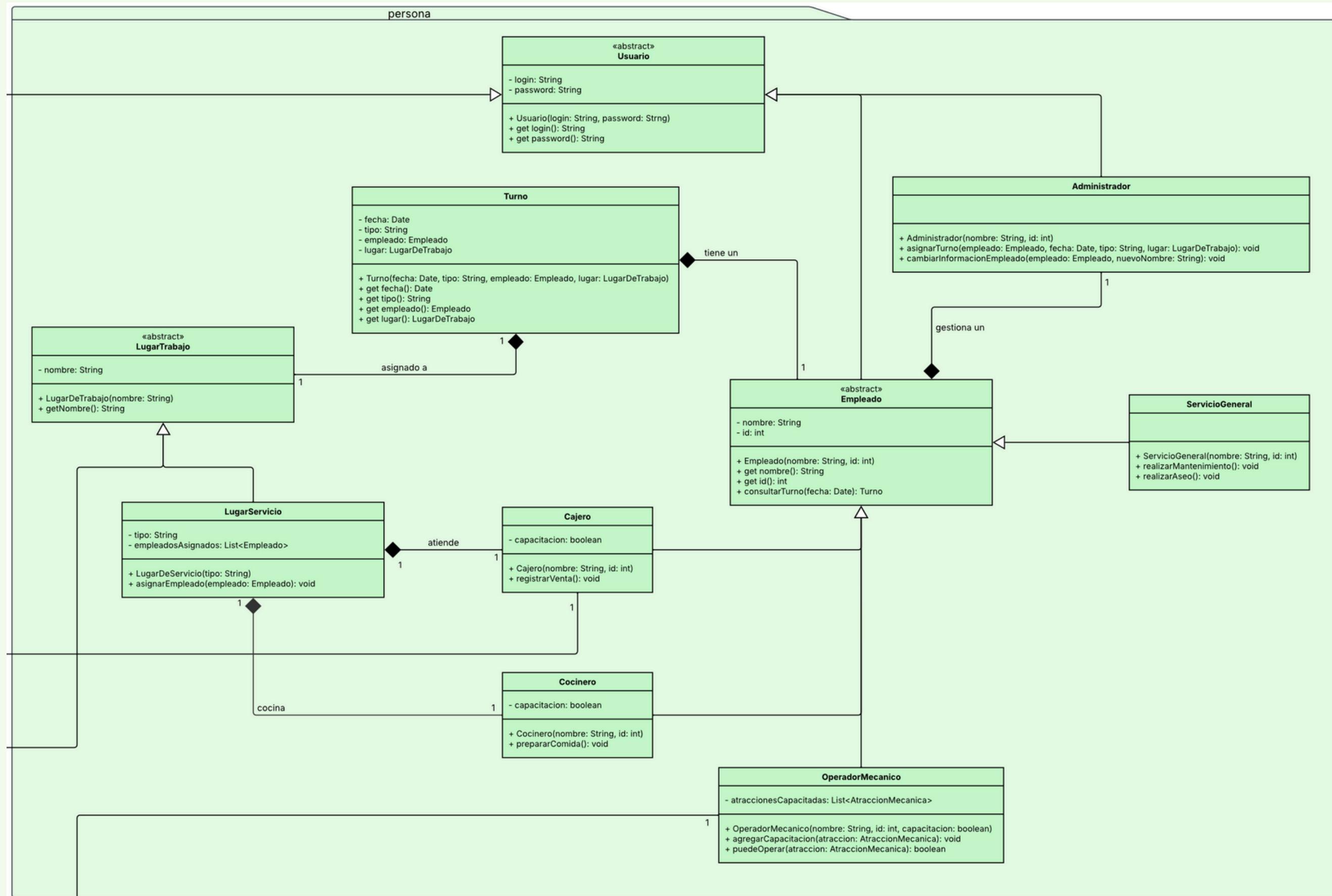
ANALISIS PROYECTO 1

DANIEL DIAB,
JERÓNIMO LÓPEZ,
JUAN ESTEBAN PINEROS

DIAGRAMA UML

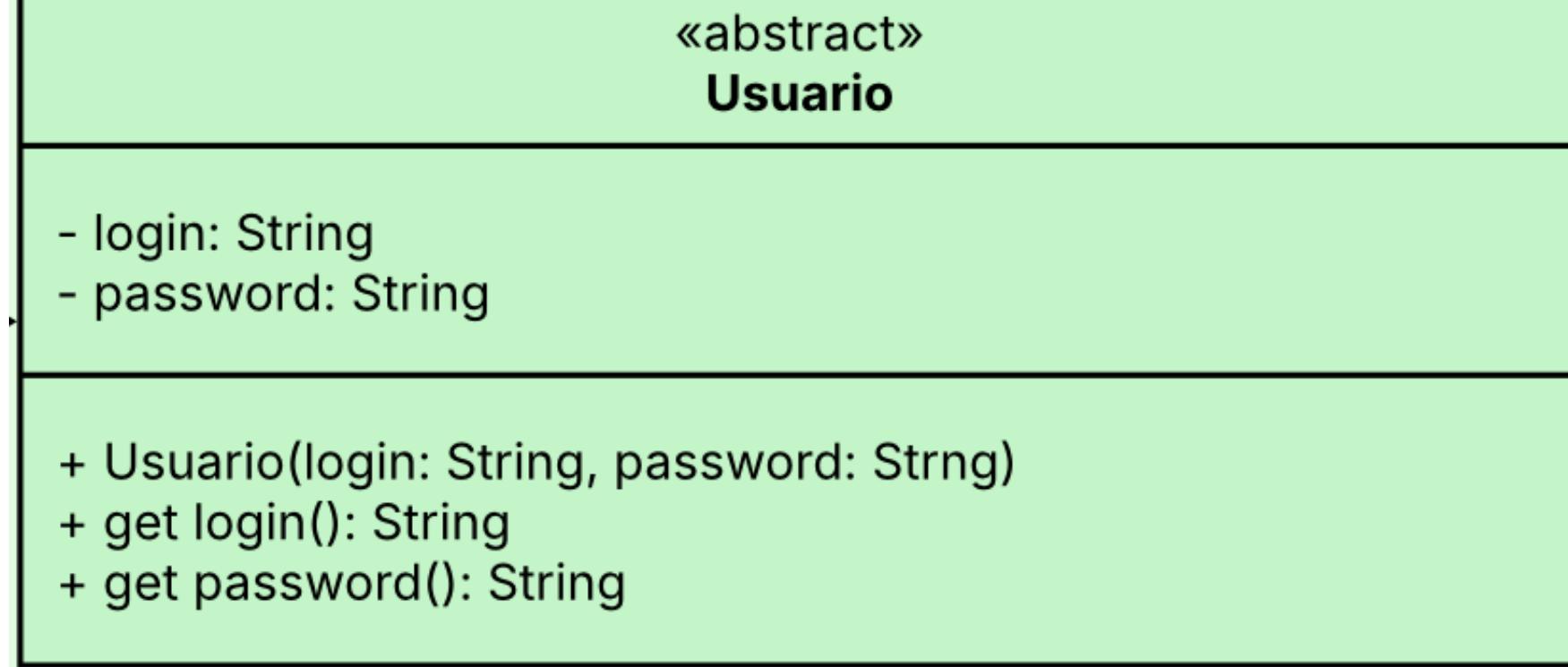


PAQUETE PERSONA



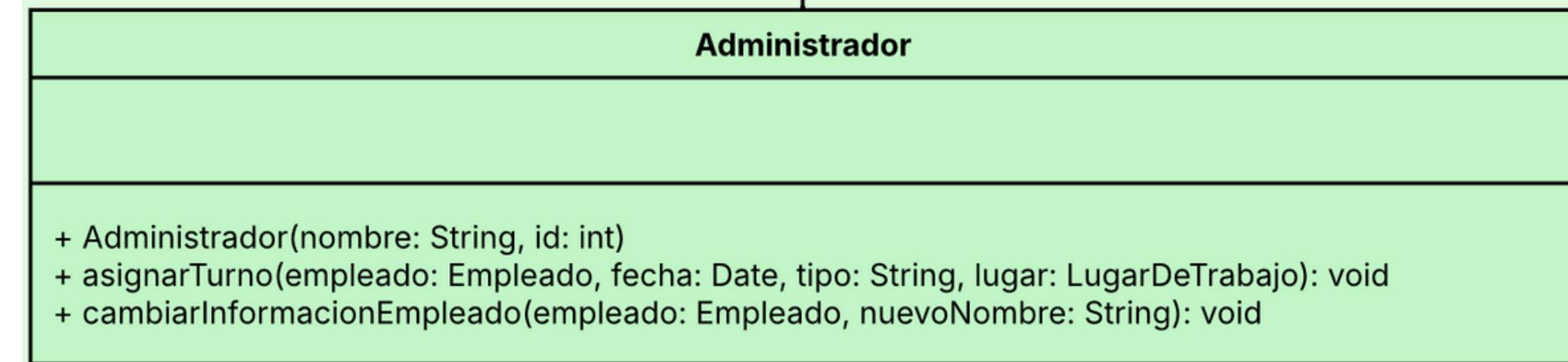
CLASE USUARIO

Clase abstracta base para todos los usuarios del sistema, la cual almacena información privada como el login y la contraseña. Sus métodos permiten acceder a estas credenciales, sirviendo de base para otras clases como Cliente, Administrador y Empleado



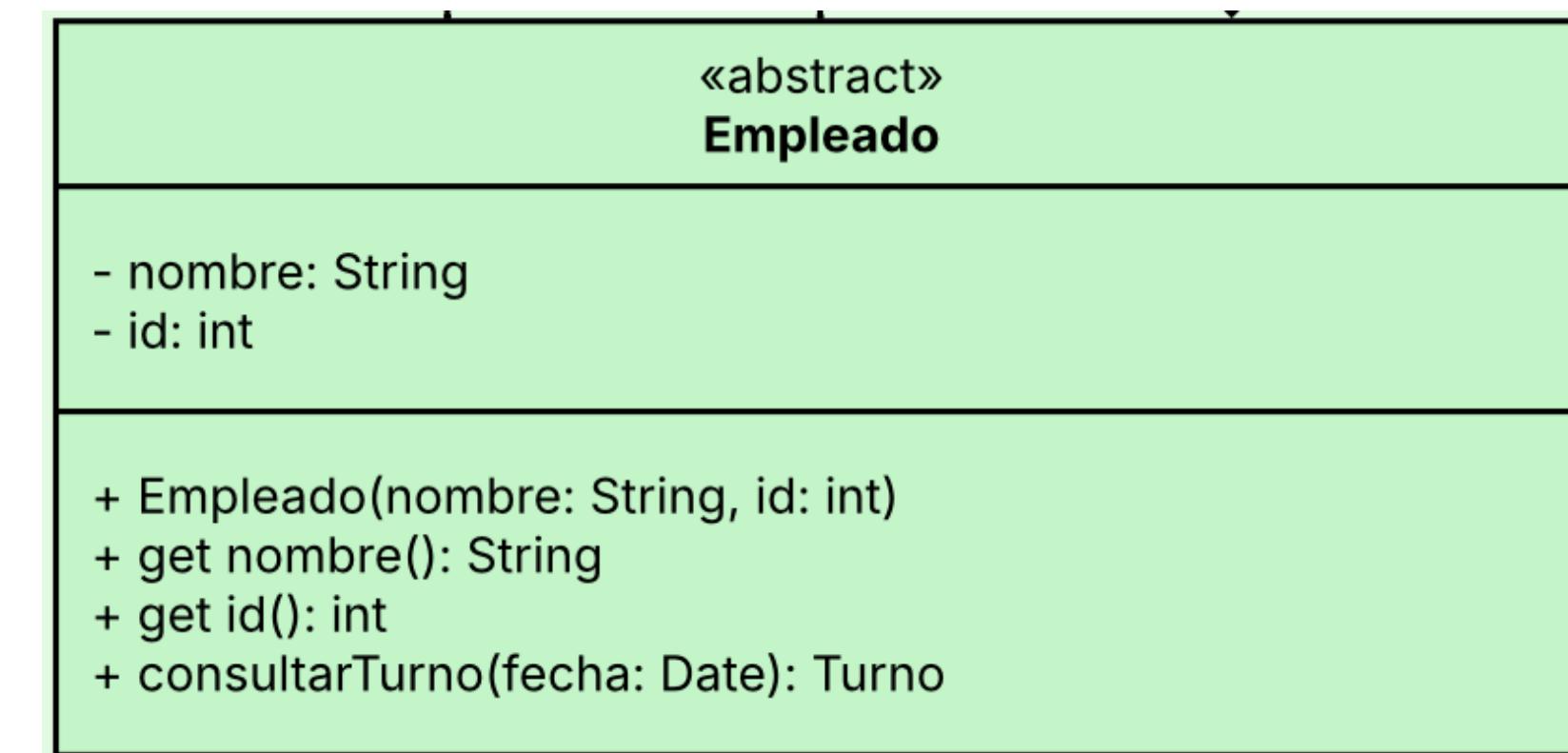
CLASE ADMINISTRADOR

Empleado con funciones administrativas dentro del parque. Sus métodos permiten asignar turnos a los empleados, modificar información de los mismos y gestionar el personal dentro del parque



CLASE EMPLEADO

Clase base para todos los empleados del parque. Contiene atributos como un nombre y un identificador único. Sus métodos permiten consultar el turno asignado a un empleado en una fecha específica



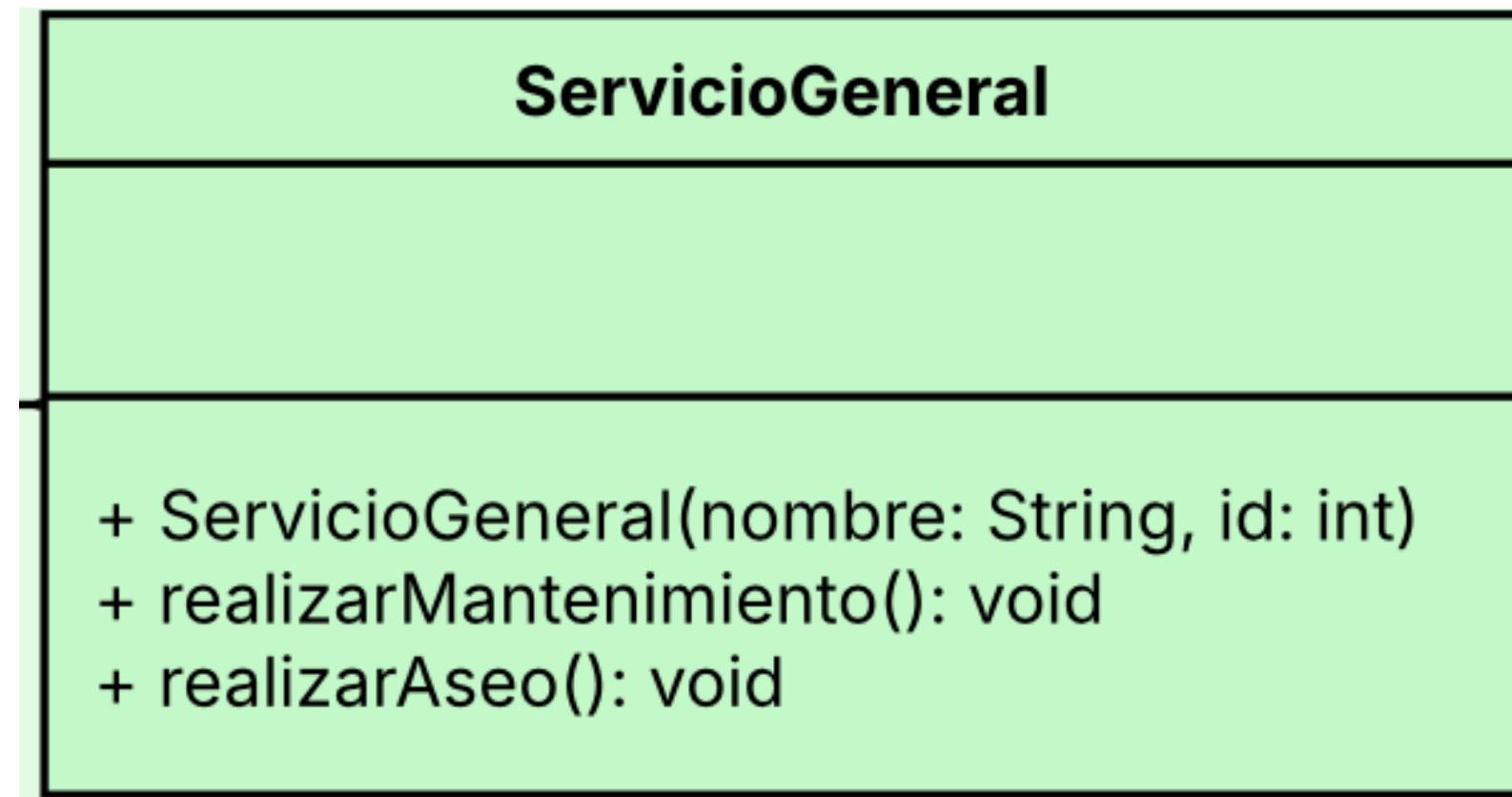
CLASE TURNO

Clase que modela los turnos de trabajo de los empleados. Almacena información sobre la fecha del turno, el tipo de turno (mañana o tarde), el empleado asignado y el lugar donde trabajará. Sus métodos permiten consultar estos datos y asignar turnos

Turno
- fecha: Date - tipo: String - empleado: Empleado - lugar: LugarDeTrabajo
+ Turno(fecha: Date, tipo: String, empleado: Empleado, lugar: LugarDeTrabajo) + get fecha(): Date + get tipo(): String + get empleado(): Empleado + get lugar(): LugarDeTrabajo

CLASE SERVICIO GENERAL

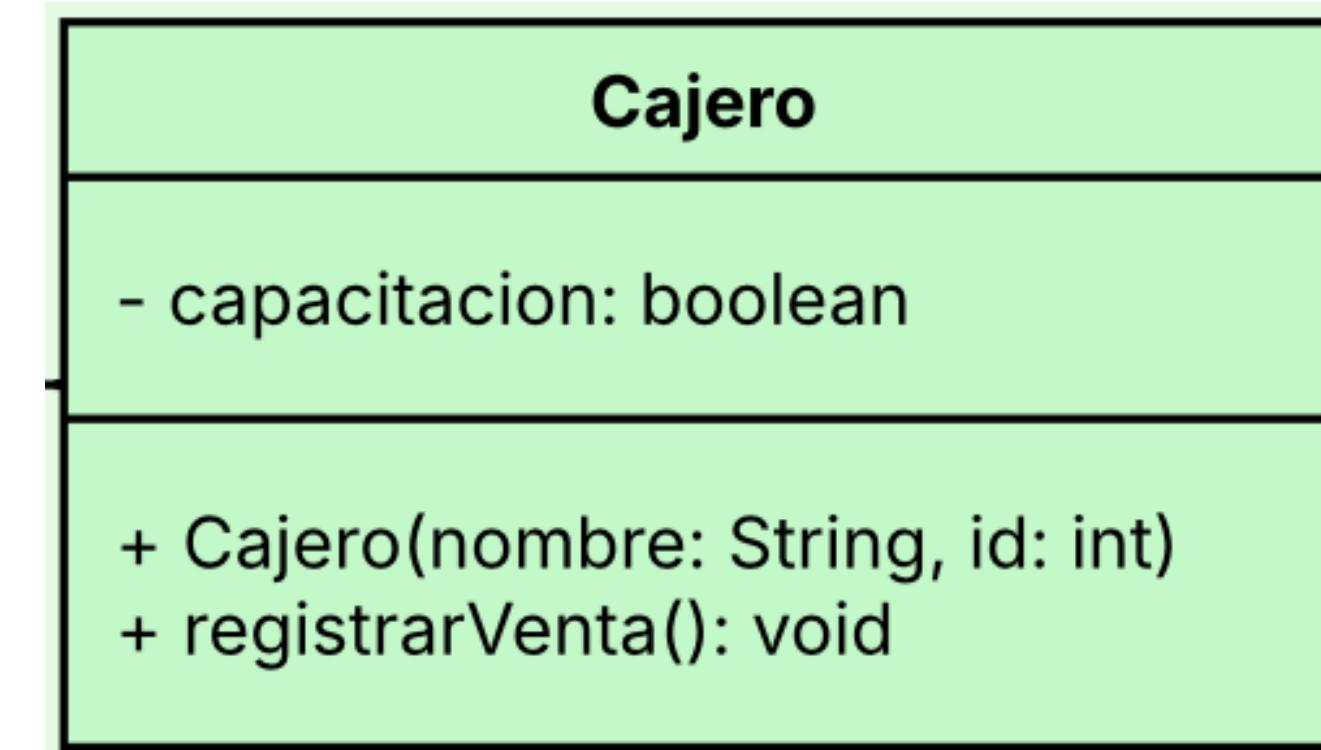
Empleado responsable del mantenimiento y limpieza del parque. Sus métodos permiten realizar tareas de mantenimiento y aseo dentro de las distintas instalaciones del parque



CLASE CAJERO

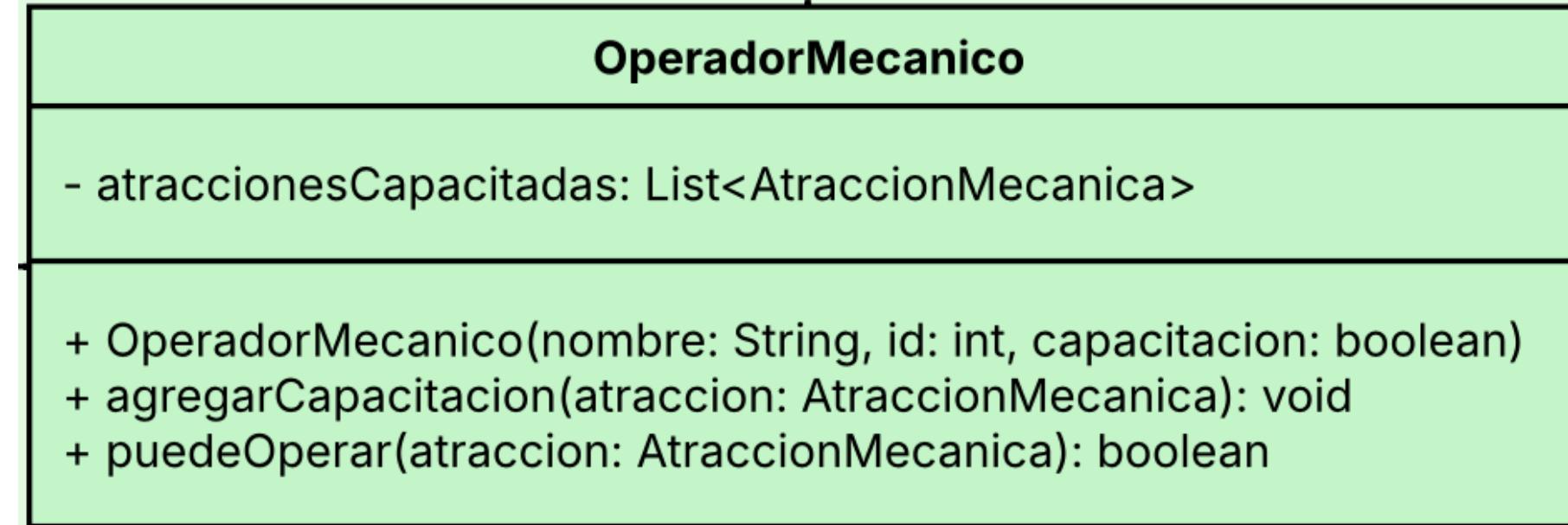
Empleado encargado de registrar ventas en las taquillas del parque.

Cuenta con métodos que permiten verificar su capacitación y registrar transacciones de venta de entradas y tiquetes



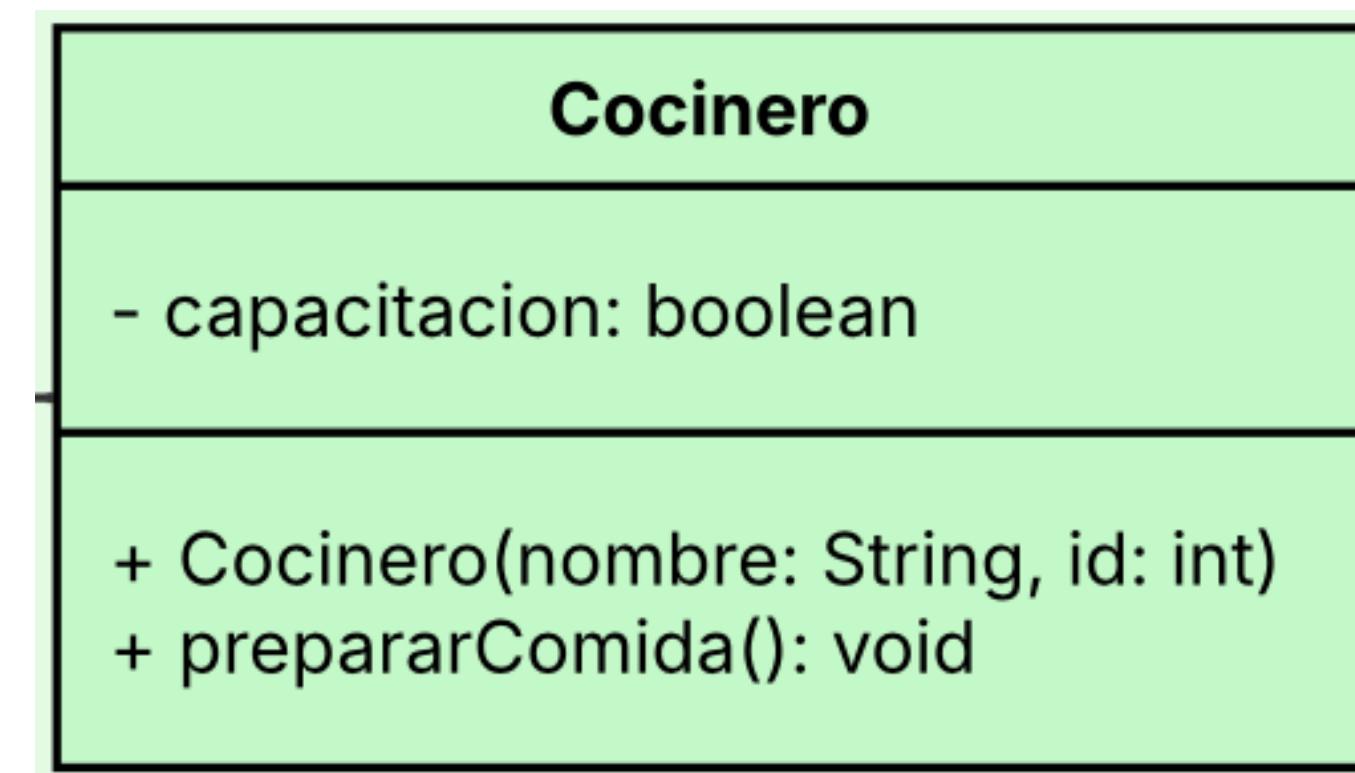
CLASE OPERADOR MECANICO

Empleado encargado de operar atracciones mecánicas. Puede recibir capacitaciones sobre diferentes atracciones y sus métodos permiten registrar nuevas capacitaciones y verificar si está autorizado para operar una atracción específica



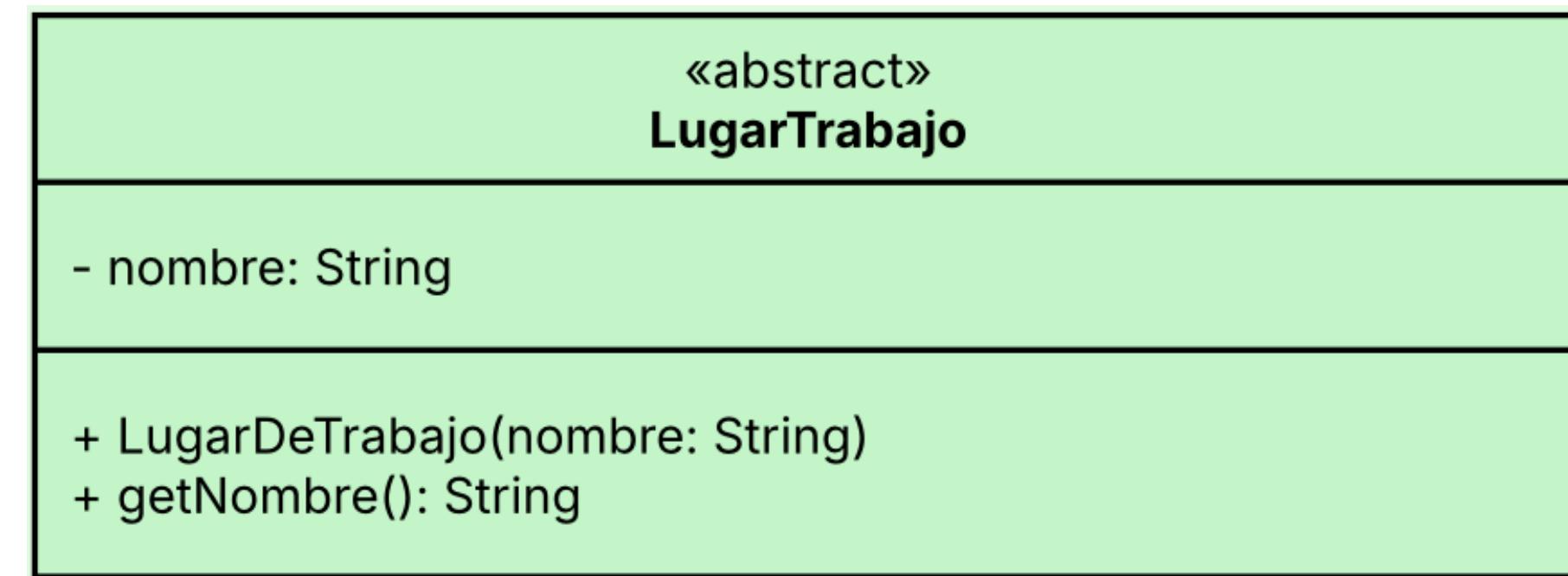
CLASE COCINERO

Empleado encargado de la preparación de alimentos dentro del parque. Cuenta con un atributo que indica si ha recibido la capacitación necesaria y métodos que permiten verificar su capacitación y realizar la preparación de alimentos



CLASE LUGAR TRABAJO

Clase abstracta que define un lugar dentro del parque donde se asignan empleados. Contiene un nombre identificador del lugar y métodos para obtener esta información

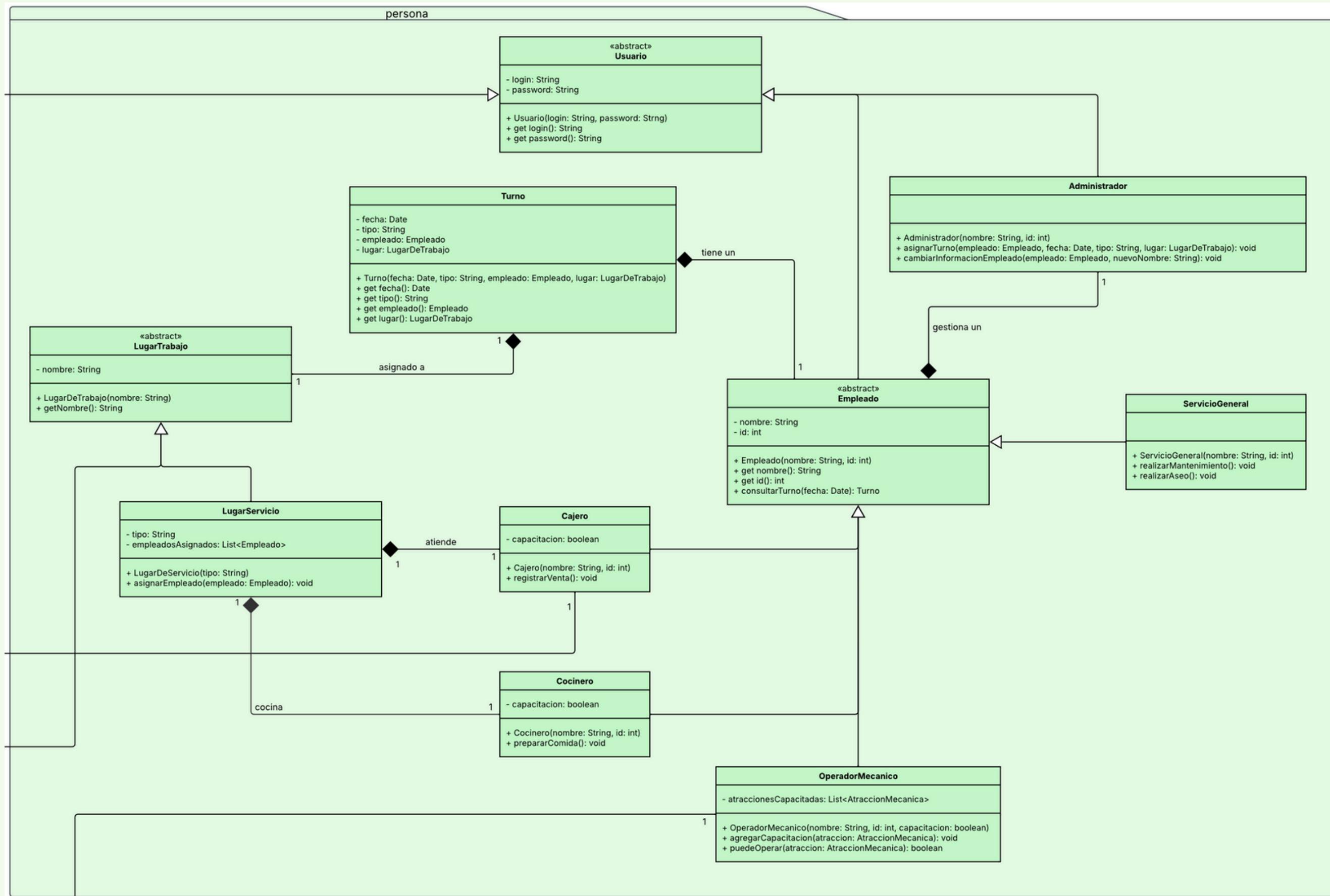


CLASE LUGAR SERVICIO

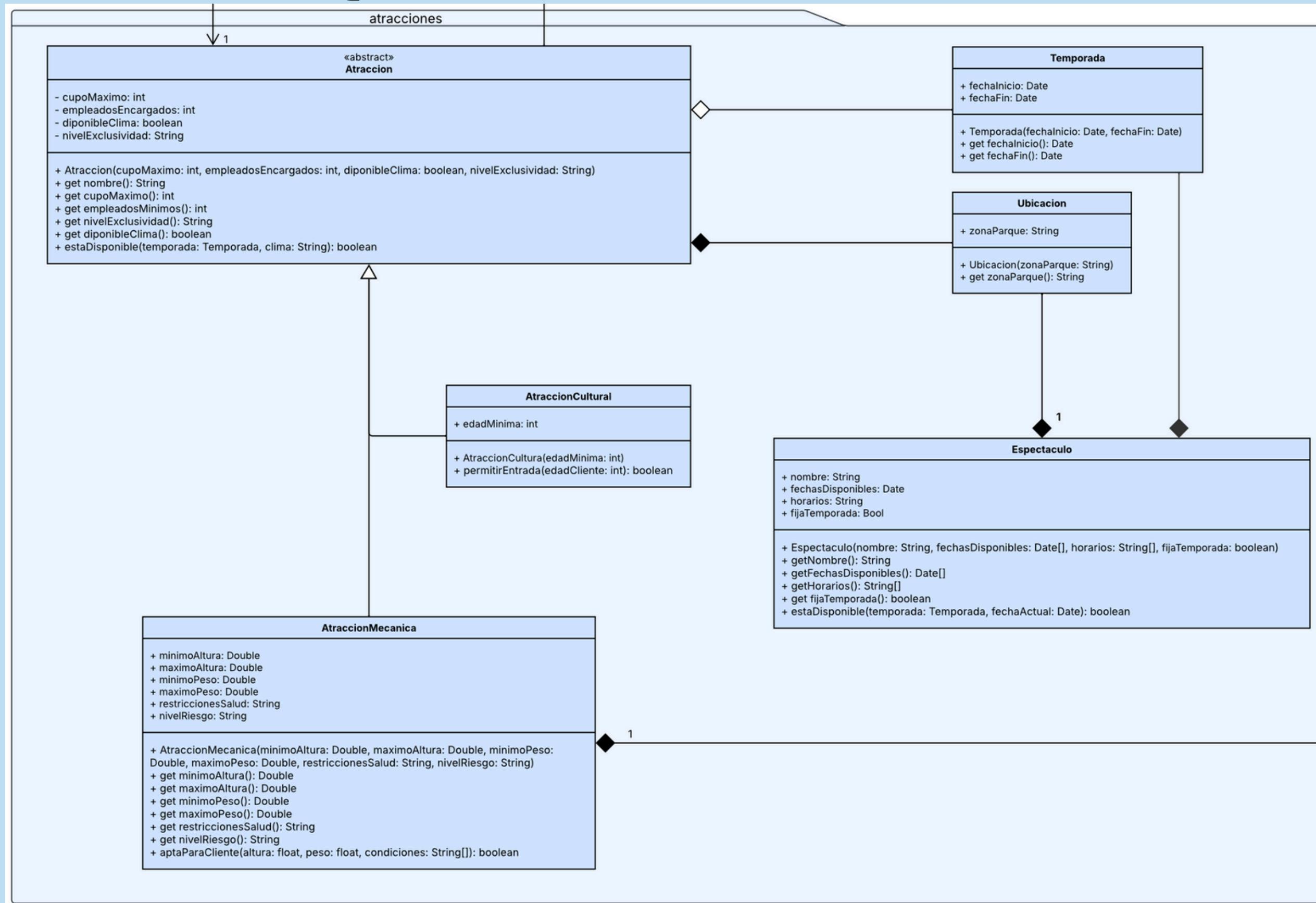
Clase que representa un área dentro del parque donde se asignan empleados. Permite gestionar la asignación de personal en estos lugares de servicio

LugarServicio
- tipo: String - empleadosAsignados: List<Empleado>
+ LugarDeServicio(tipo: String) + asignarEmpleado(empleado: Empleado): void

RELACIONES ENTRE CLASES

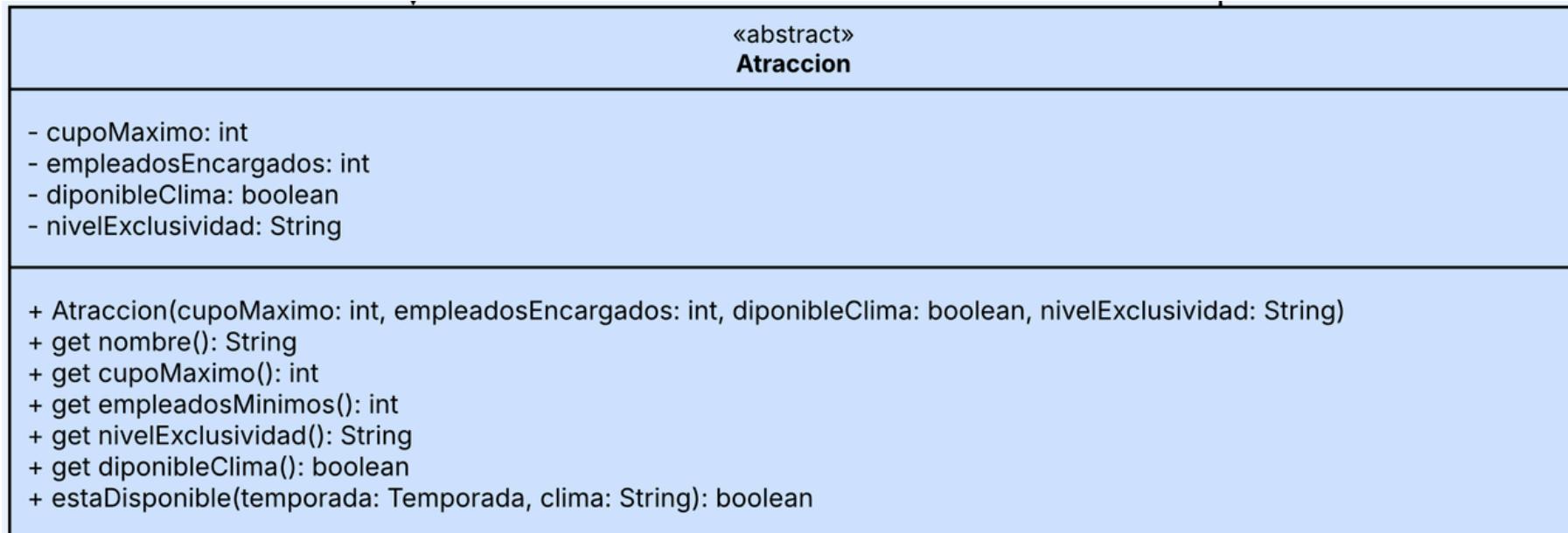


PAQUETE ATRACCIONES



CLASE ATRACCION

Clase abstracta que modela las atracciones del parque. Contiene atributos como un cupo máximo, el número de empleados encargados, un indicador de disponibilidad por clima y un nivel de exclusividad. Sus métodos permiten verificar si la atracción está disponible en función de la temporada y las condiciones climáticas



CLASE ATRACCION CULTURAL

Subclase de Atraccion que representa atracciones de tipo cultural. Contiene una restricción de edad mínima y un método que permite verificar si un cliente cumple con este requisito para ingresar a la atracción

AtraccionCultural

- + edadMinima: int
- + AtraccionCultura(edadMinima: int)
- + permitirEntrada(edadCliente: int): boolean

CLASE ATRACCION MECANICA

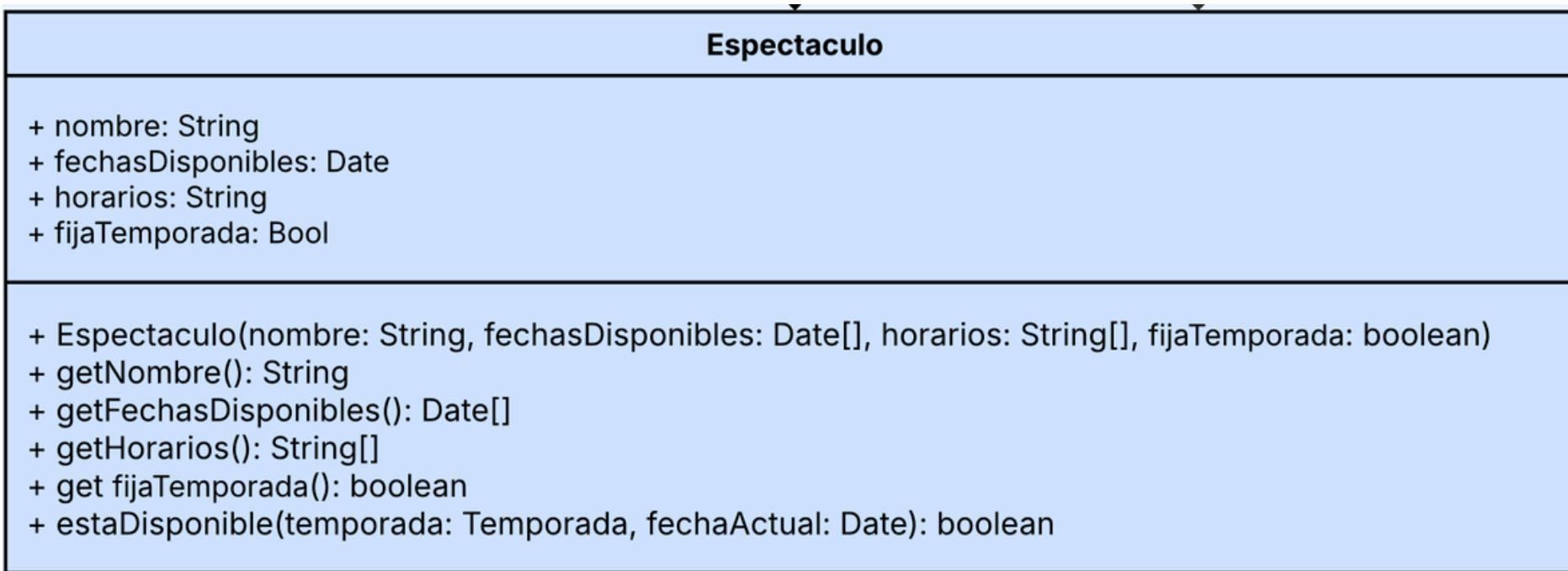
Subclase de Atraccion que agrega restricciones de altura y peso, restricciones de salud y un nivel de riesgo. Incluye métodos que permiten verificar si un cliente puede acceder a la atracción según sus características físicas y condiciones de salud

AtraccionMecanica
+ minimoAltura: Double + maximoAltura: Double + minimoPeso: Double + maximoPeso: Double + restriccionesSalud: String + nivelRiesgo: String
+ AtraccionMecanica(minimoAltura: Double, maximoAltura: Double, minimoPeso: Double, maximoPeso: Double, restriccionesSalud: String, nivelRiesgo: String) + get minimoAltura(): Double + get maximoAltura(): Double + get minimoPeso(): Double + get maximoPeso(): Double + get restriccionesSalud(): String + get nivelRiesgo(): String + aptaParaCliente(altura: float, peso: float, condiciones: String[]): boolean

CLASE ESPECTACULO

Clase que modela un espectáculo dentro del parque.

Contiene atributos como su nombre, fechas disponibles, horarios y si está vinculado a una temporada específica. Sus métodos permiten verificar la disponibilidad de un espectáculo en una fecha dada



CLASE UBICACION

Clase que modela la localización de elementos dentro del parque. Contiene un atributo que indica la zona en la que se encuentra el elemento y métodos para obtener esta información

Ubicacion

+ zonaParque: String

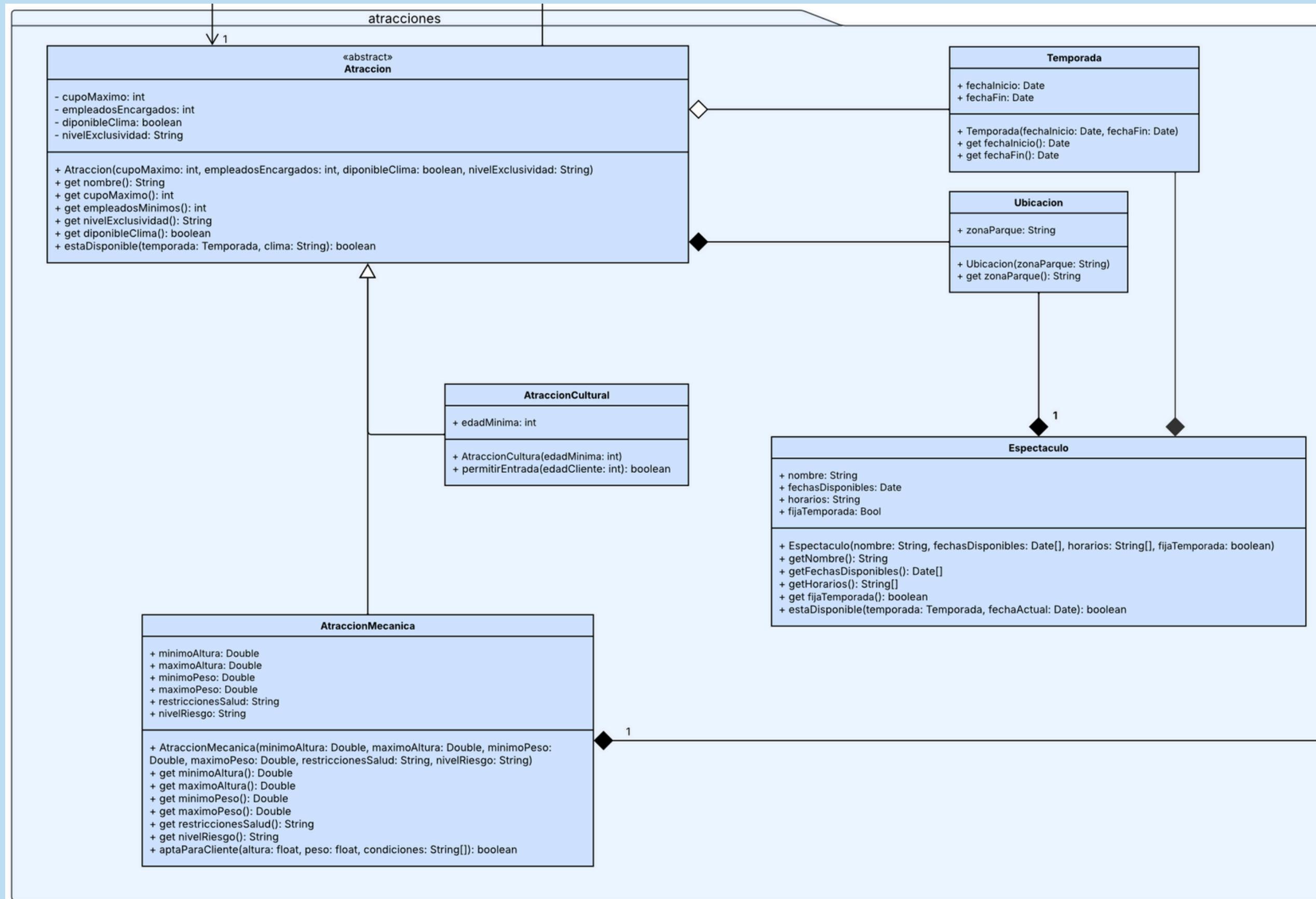
+ Ubicacion(zonaParque: String)
+ get zonaParque(): String

CLASE TEMPORADA

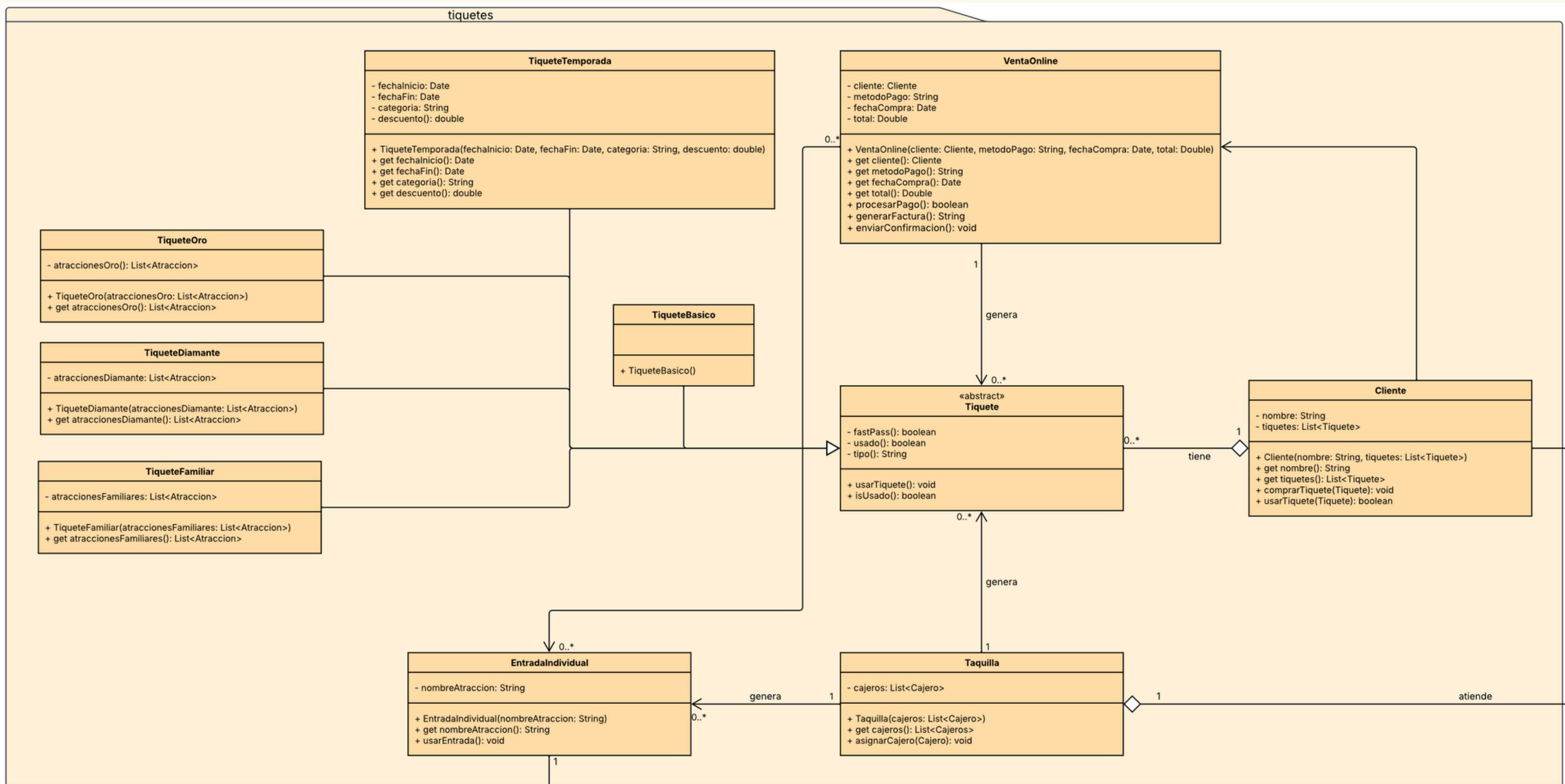
Clase que define un período de tiempo en el parque. Contiene una fecha de inicio y una fecha de fin, y sus métodos permiten consultar estos valores para determinar la disponibilidad de ciertas atracciones o eventos

Temporada
+ fechalinicio: Date + fechaFin: Date + Temporada(fechalinicio: Date, fechaFin: Date) + get fechalinicio(): Date + get fechaFin(): Date

RELACIONES ENTRE CLASES

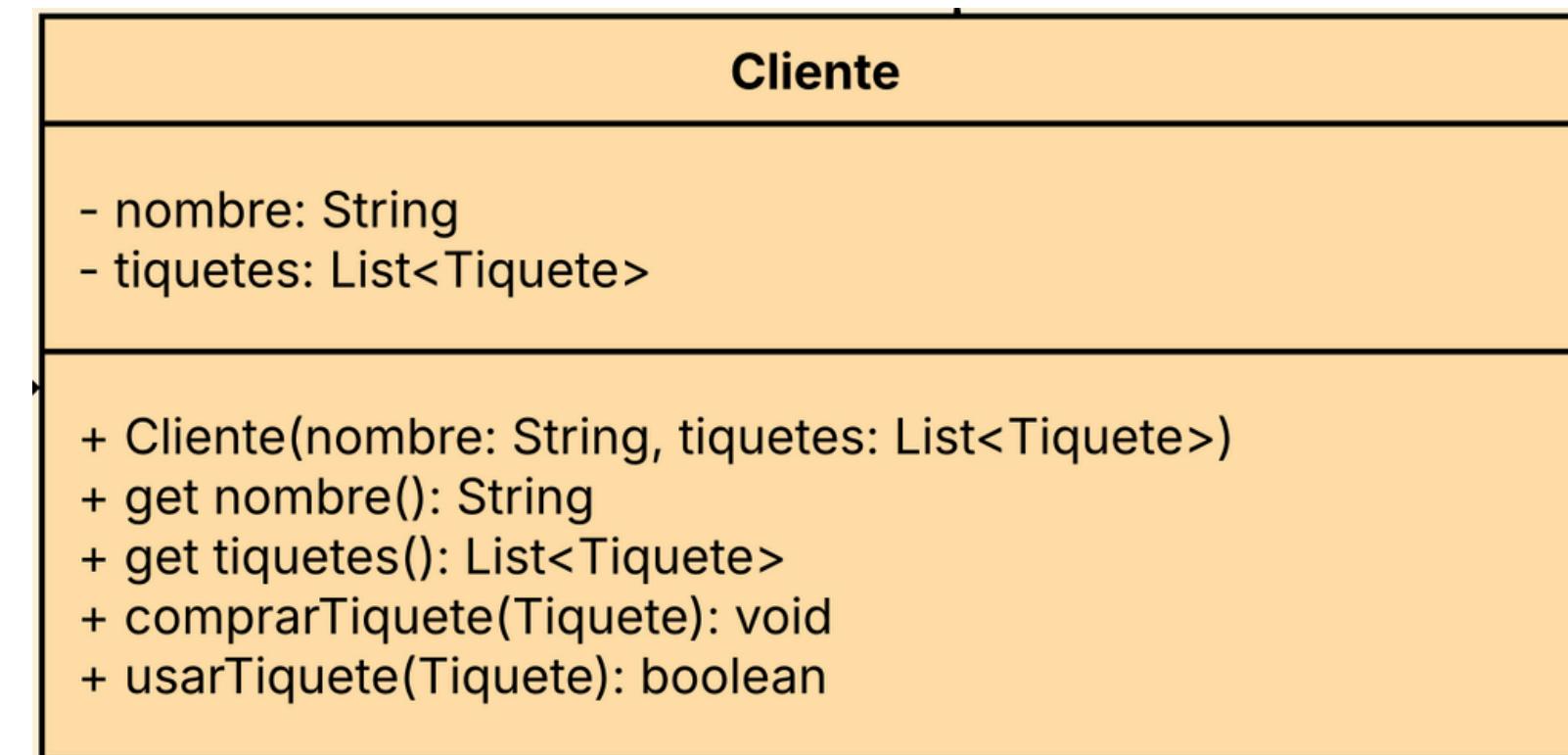


PAQUETE TIQUETES



CLASE CLIENTE

Representa a un visitante del parque que puede adquirir y utilizar tiquetes. Contiene su nombre y una lista de tiquetes comprados. Sus métodos permiten gestionar la compra de tiquetes y el uso de los mismos dentro del parque



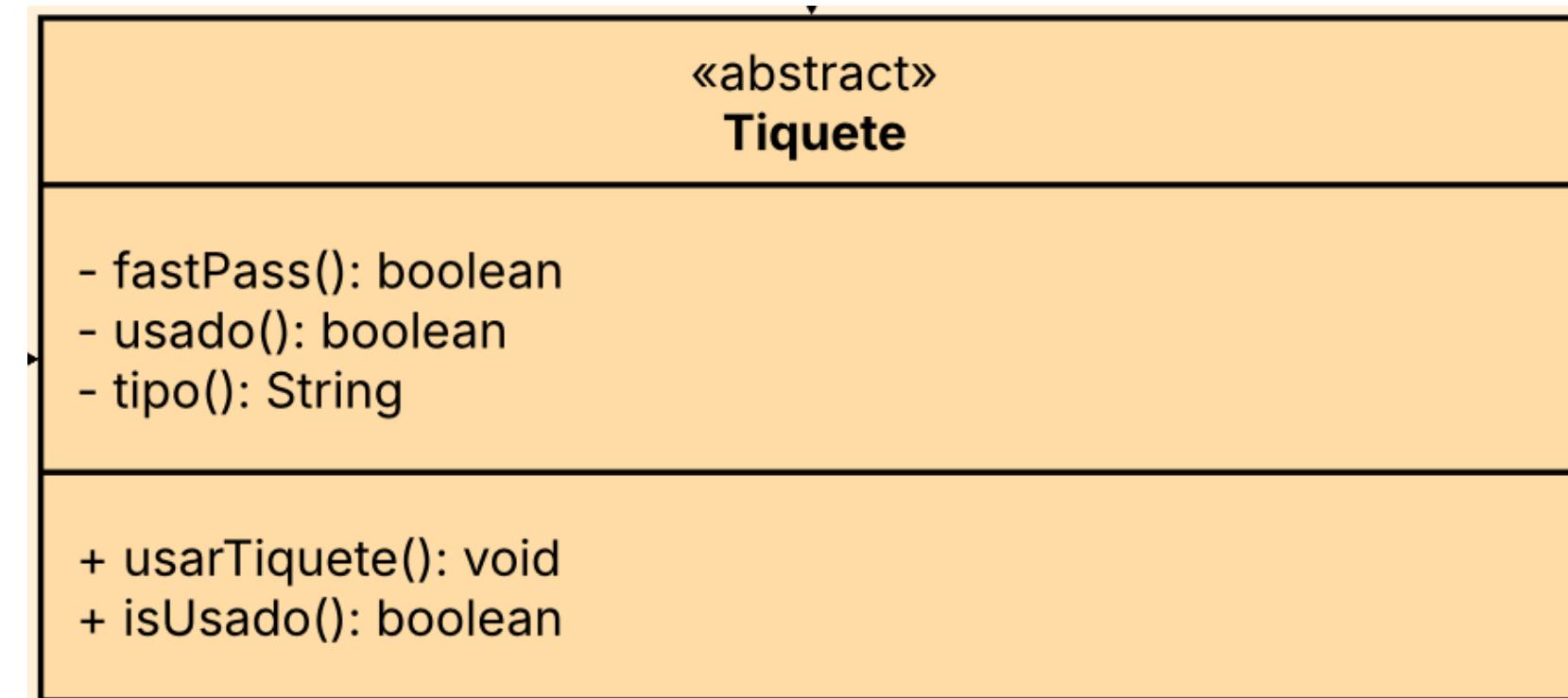
CLASE VENTA ONLINE

Clase que gestiona la compra de tiquetes a través de internet. Almacena información del cliente, el método de pago, la fecha de compra y el total de la transacción. Sus métodos permiten procesar el pago, generar la factura y enviar la confirmación al cliente

VentaOnline
<ul style="list-style-type: none">- cliente: Cliente- metodoPago: String- fechaCompra: Date- total: Double
<ul style="list-style-type: none">+ VentaOnline(cliente: Cliente, metodoPago: String, fechaCompra: Date, total: Double)+ get cliente(): Cliente+ get metodoPago(): String+ get fechaCompra(): Date+ get total(): Double+ procesarPago(): boolean+ generarFactura(): String+ enviarConfirmacion(): void

CLASE TIQUETE

Clase abstracta base para los distintos tipos de tiquetes. Define atributos como si el tiquete cuenta con fast pass, si ha sido utilizado y el tipo de tiquete. Sus métodos permiten consultar esta información y modificar el estado del tiquete



CLASE TAQUILLA

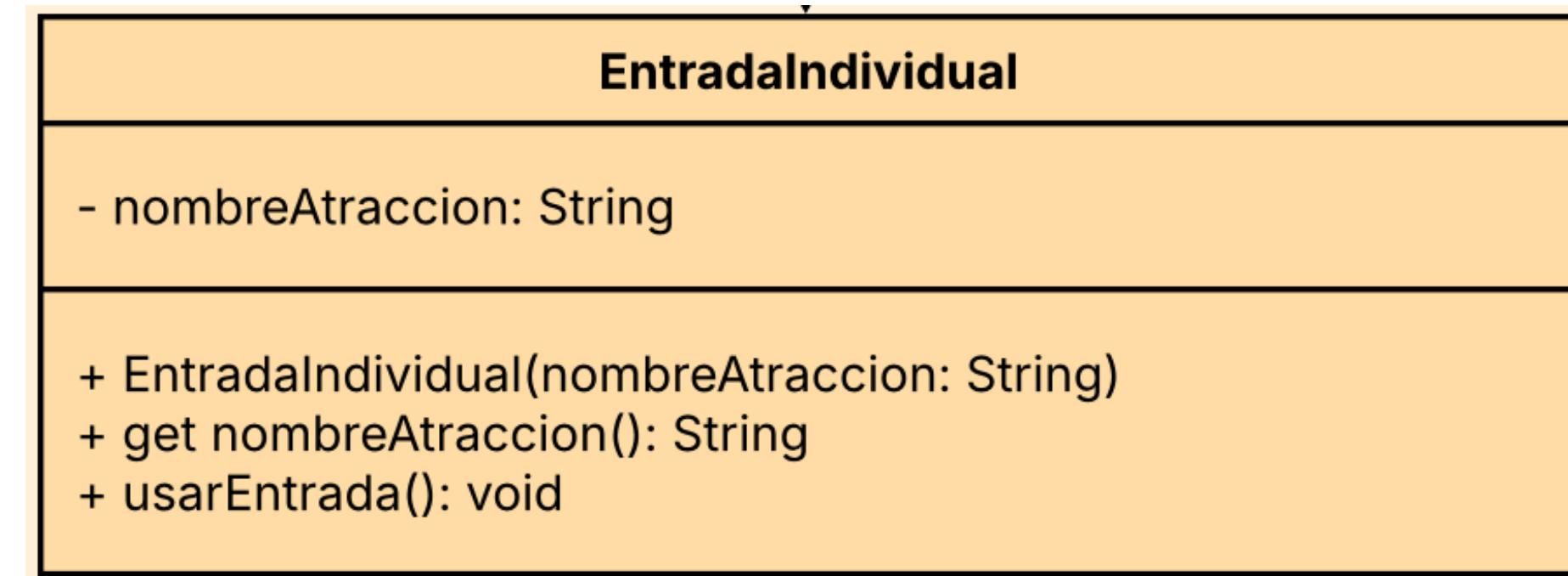
Clase que representa una taquilla del parque donde se venden entradas. Contiene una lista de cajeros asignados y permite gestionar la asignación de estos empleados a la taquilla

Taquilla
- cajeros: List<Cajero>
+ Taquilla(cajeros: List<Cajero>) + get cajeros(): List<Cajeros> + asignarCajero(Cajero): void

CLASE ENTRADA INDIVIDUAL

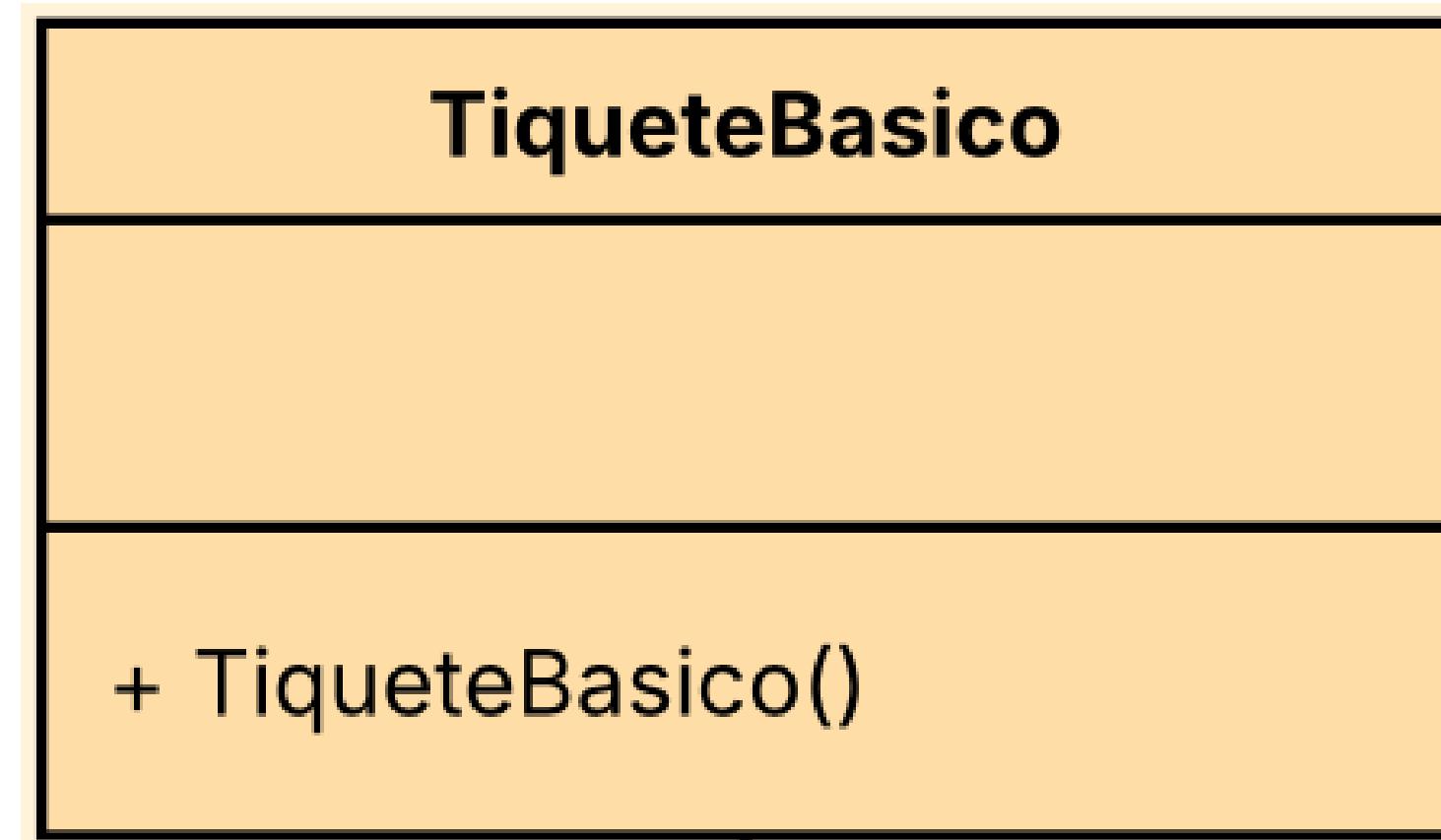
Clase que representa una entrada para una atracción específica dentro del parque.

Contiene el nombre de la atracción a la que da acceso y permite verificar si la entrada ha sido utilizada. Sus métodos permiten consultar la atracción asociada y registrar el uso de la entrada



CLASE TIQUETE BASICO

Clase que representa un tiquete estándar dentro del parque, este no incluye el acceso a ninguna atracción. Hereda de la clase Tiquete y mantiene la funcionalidad básica sin características adicionales



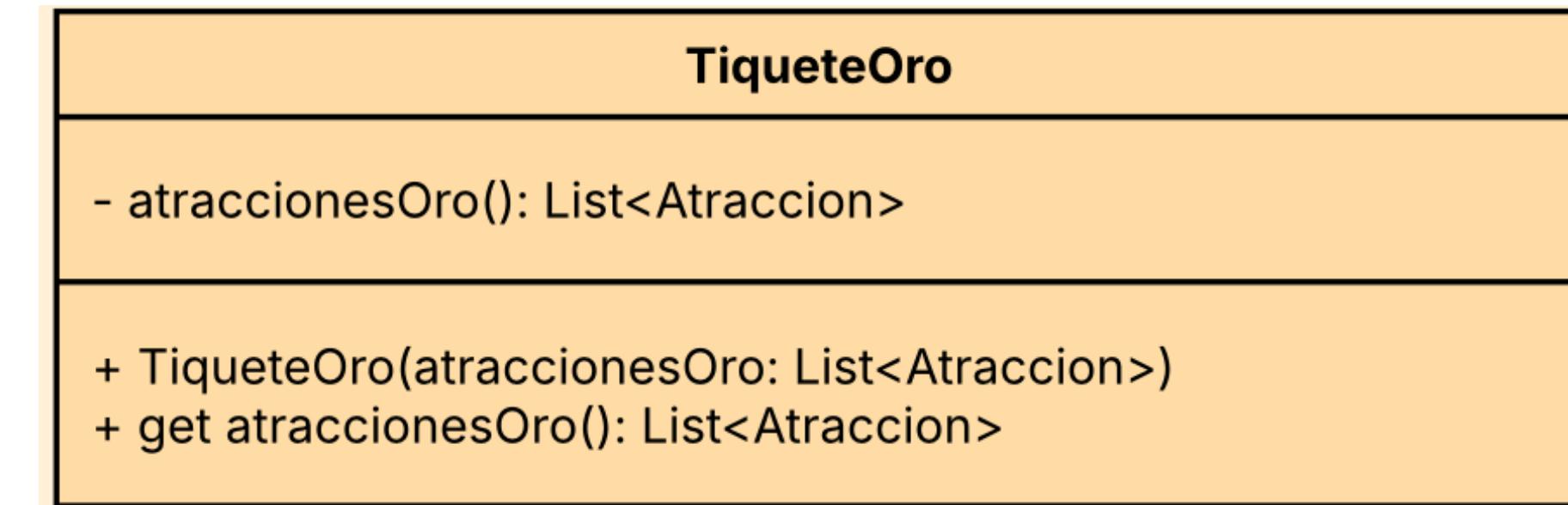
CLASE TIQUETE FAMILIAR

Clase que representa un tiquete diseñado para acceso familiar. Contiene una lista de atracciones familiares a las que permite ingresar y métodos que permiten consultar esta información

TiqueteFamiliar
- atraccionesFamiliares: List<Atraccion>
+ TiqueteFamiliar(atraccionesFamiliares: List<Atraccion>) + get atraccionesFamiliares(): List<Atraccion>

CLASE TIQUETE ORO

Tiquete con beneficios adicionales, que permite el acceso a atracciones de categoría oro. Contiene una lista de atracciones exclusivas para este tipo de tiquete y métodos que permiten consultar esta lista



CLASE TIQUETE DIAMANTE

Tiquete premium que ofrece acceso exclusivo a atracciones de categoría diamante.

Contiene una lista de atracciones diamante y métodos que permiten consultar estas atracciones para los clientes que poseen este tipo de tiquete

TiqueteDiamante

- atraccionesDiamante: List<Atraccion>
- + TiqueteDiamante(atraccionesDiamante: List<Atraccion>)
- + get atraccionesDiamante(): List<Atraccion>

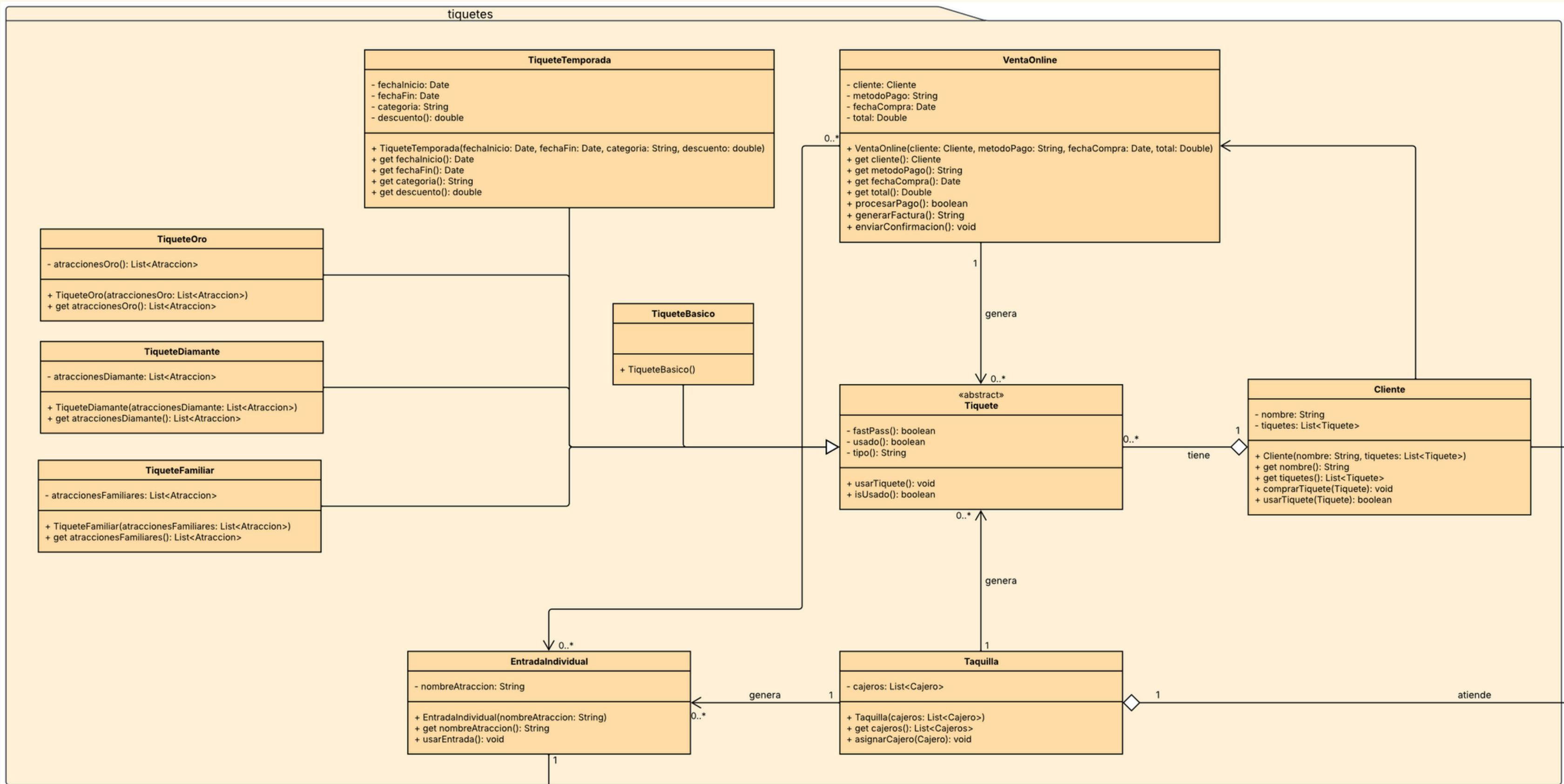
CLASE TIQUETE TEMPORADA

Esta clase representa un tiquete con validez limitada dentro de un rango de fechas.

Contiene atributos como la fecha de inicio y fin del tiquete, su categoría y un descuento aplicable. A través de sus métodos, permite consultar esta información y aplicar el descuento correspondiente

TiqueteTemporada
<ul style="list-style-type: none">- fechalinicio: Date- fechaFin: Date- categoria: String- descuento(): double
<ul style="list-style-type: none">+ TiqueteTemporada(fechalinicio: Date, fechaFin: Date, categoria: String, descuento: double)+ get fechalinicio(): Date+ get fechaFin(): Date+ get categoria(): String+ get descuento(): double

RELACIONES ENTRE CLASES



GRACIAS