

Clases para crear en Lucidchart

CasaRural

- Atributos:
 - id : uuid
 - nombre : String
 - direccion : String
 - telefono : String
 - Métodos:
 - agregarHabitacion(h: Habitacion) : void
 - listarHabitaciones() : List<Habitacion>
-

Habitacion

- Atributos:
 - id : uuid
 - codigo : String
 - descripcion : String
 - capacidad : int
 - precioNoche : BigDecimal
 - activa : boolean
-

Cliente

- Atributos:
 - id : uuid
 - nombre : String
 - documento : String

- email : String
 - telefono : String
-

Reserva

- Atributos:

- id : uuid
- fechaInicio : LocalDate
- fechaFin : LocalDate
- noches : int
- importeTotal : BigDecimal
- estado : EstadoReserva

- Métodos:

- calcularNoches() : void
 - calcularImporte() : void
 - cambiarEstado(e: EstadoReserva) : void
-

Pago (*abstracta*)

- Atributos:

- id : uuid
 - fecha : LocalDateTime
 - importe : BigDecimal
 - tipo : String ("PagoTarjeta"/"PagoEfectivo")
-

PagoTarjeta

- Atributos:

- titular : String
- ultimos4 : String

- autorizacion : String
 - marca : String
-

PagoEfectivo

- Atributos:
 - recibo : String
-

EstadoReserva (enumeración)

- PENDIENTE
 - CONFIRMADA
 - CANCELADA
-

Relaciones para conectar en Lucidchart

- CasaRural 1 → 1..* Habitación (contiene)
 - Cliente 1 → 0..* Reserva (realiza)
 - Habitación 1 → 0..* Reserva (es reservada por)
 - Reserva 1 → 0..* Pago (recibe)
 - Pago (abstracta) → PagoTarjeta y PagoEfectivo (herencia)
 - Reserva usa EstadoReserva (enum)
-

Con esto, puedes ir a **Lucidchart** → **Nuevo diagrama UML** → **Clases** y:

1. Crear las cajas de clase (con nombre, atributos y métodos).

2. Dibujar las flechas con multiplicidad.
3. Aplicar herencia desde **Pago** hacia **PagoTarjeta** y **PagoEfectivo**.
4. Añadir el enum **EstadoReserva** y enlazarlo con **Reserva**.