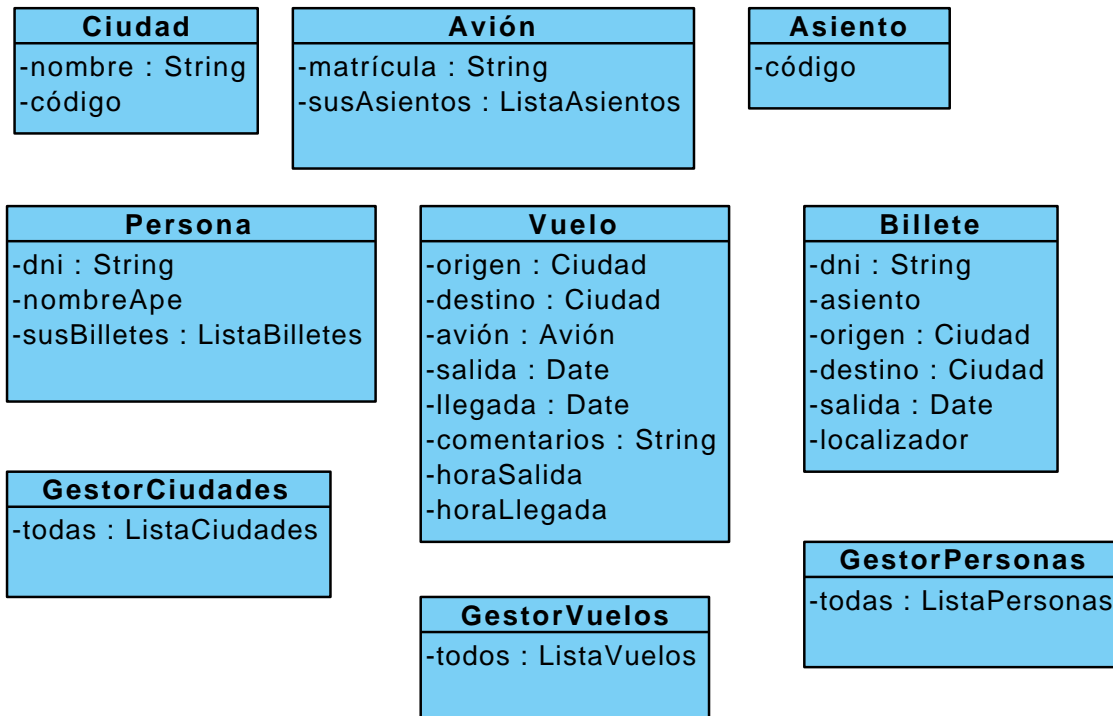


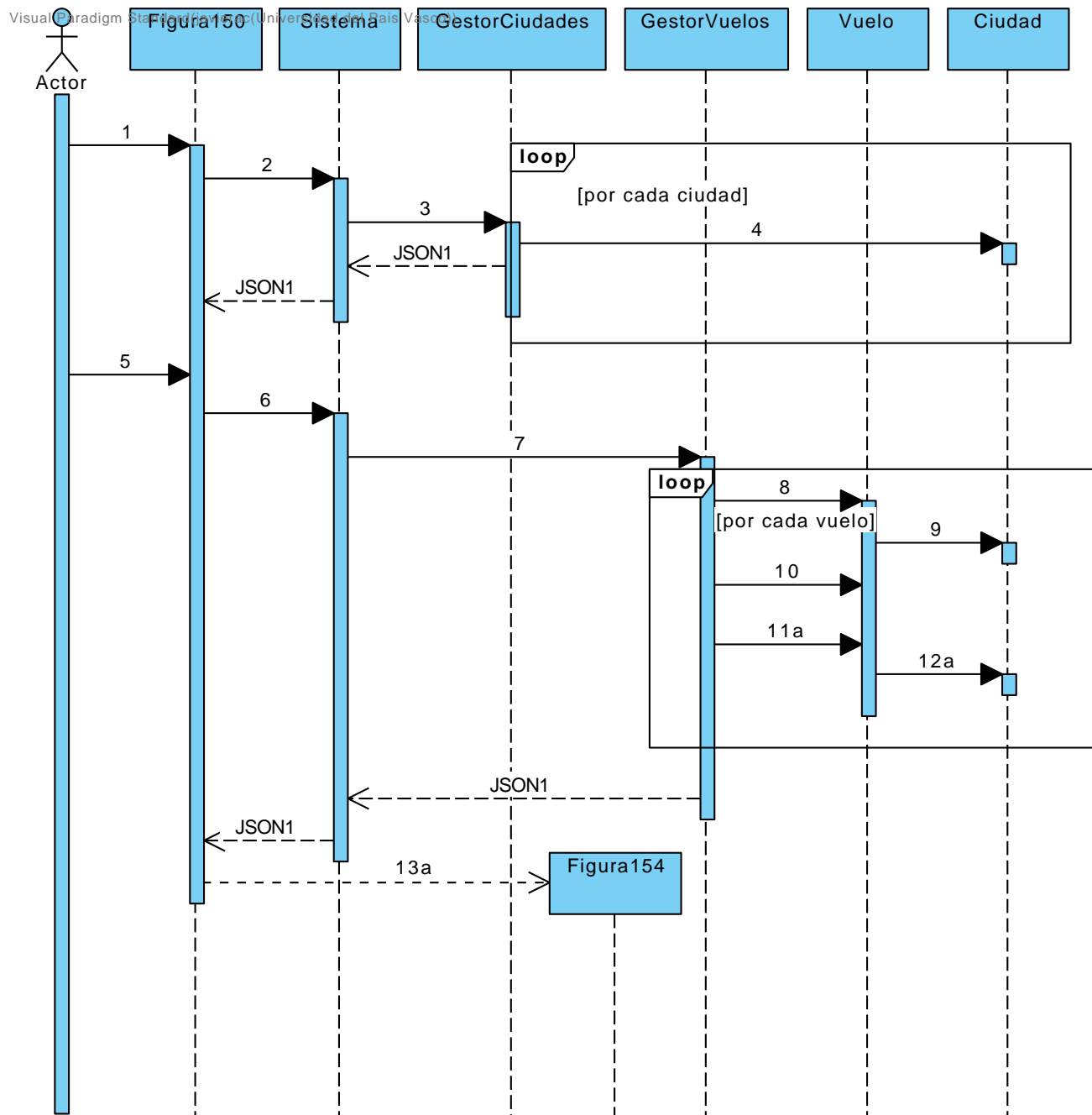
DB:

Ciudad(nombre, **código**)Avión(**matrícula**)Asiento(**código**, *matrículaAvión*)Persona(**dni**, nombreApellidos)Billete(*dni, asiento, origen, destino*, localizador, *fecha, hora*)Vuelo(*origen, destino, avión, fechaSalida, horaSalida*, llegada, comentarios)**Observaciones:**

En algunos casos se ha separado el campo FechaHora en dos campos: Fecha y Hora. No tener en cuenta esta apreciación.

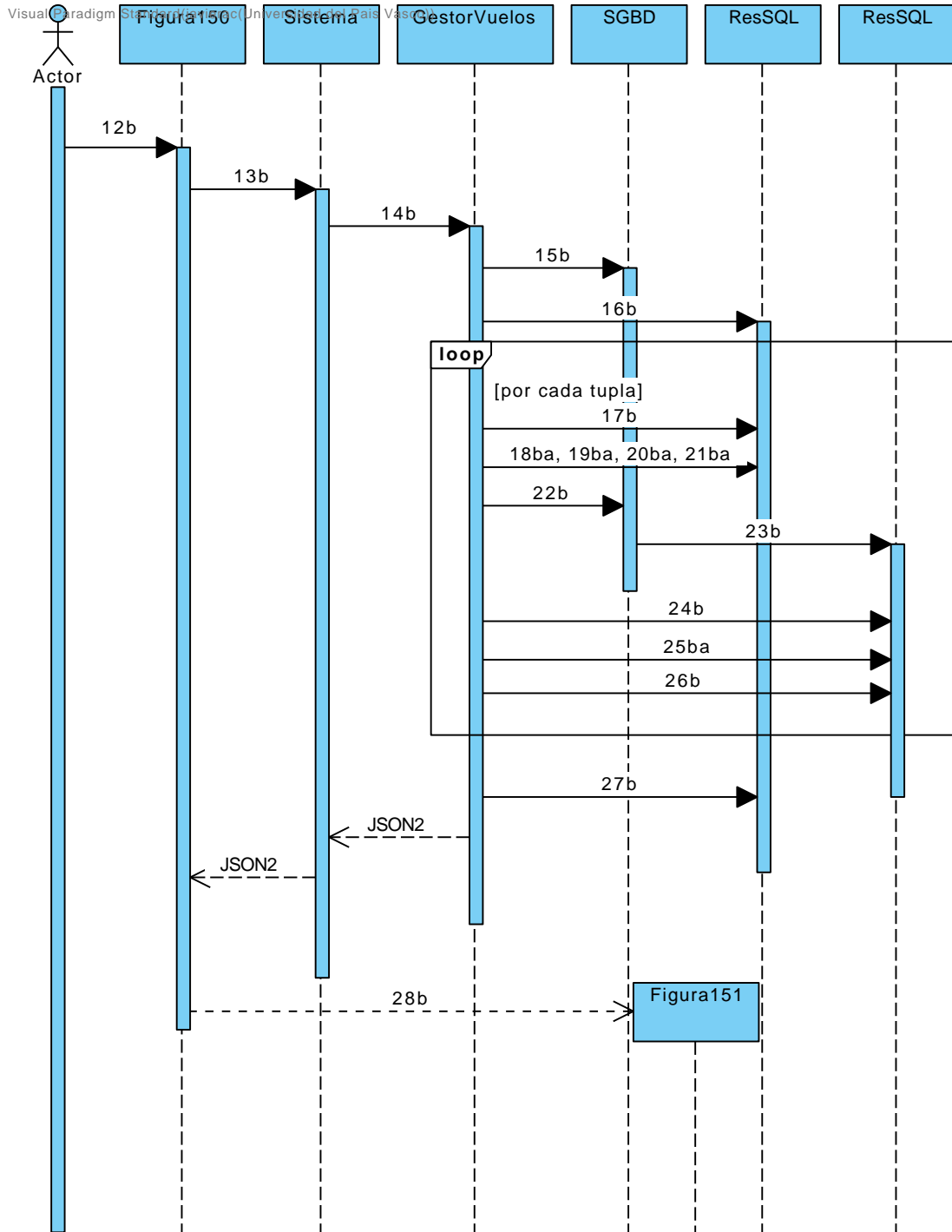
Algunas partes se han realizado con DB y otras con objetos.

Suponer que cuando se hace una natural join entre Billete y Vuelo, también coge los campos Fecha con FechaSalida y Hora con HoraSalida.

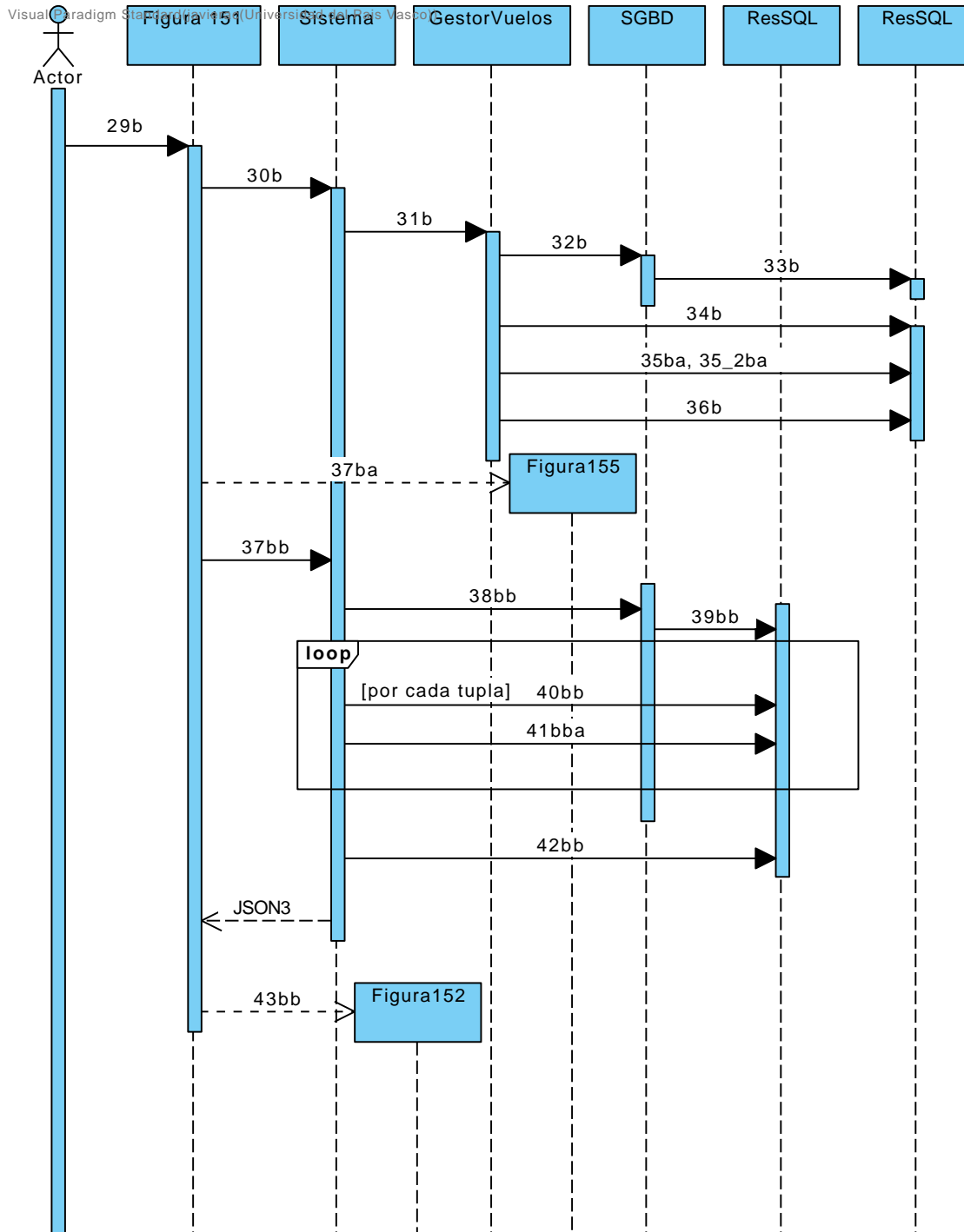


1. Accede a la aplicación  
 2. obtenerCiudades(): JSON1: {[nombreCiudad: String, ...]}  
 3. obtenerCiudades(): JSON1  
 4. getNombre(): String

5. Abre el desplegable de ciudades y selecciona una, y elige una fecha para viajar.  
 6. obtenerDestinos(fecha, origen): JSON1  
 7. obtenerDestinos(fecha, origen): JSON1  
 8. getOrigen(): String  
 9. getOrigen(): String  
 10. getDate(): String  
 [if origen and fecha coinciden]  
 11a. getDestino: String  
 12a. getDestino: String  
 [si no hay destinos]  
 13a. new Figura154(fecha, origen)  
 [si hay]



12b. Selecciona destino y pulsa "Buscar vuelos".  
 13b. obtenerVuelos(fecha, origen, destino): JSON2:  
 {[{horaSalida: String, horaLlegada: String, plazas: int, comentarios: String},{...},...]}  
 14b. obtenerVuelos(fecha,origen,destino): JSON2  
 15b. execSQL("SELECT horaSalida, horaLlegada, comentarios, COUNT(Asiento.código) FROM Vuelos NATURAL JOIN Asiento WHERE origen LIKE '%origen%' AND destino LIKE '%destino%' AND fechaSalida LIKE '%fecha%' GROUP BY horaSalida, horaLlegada, comentarios")  
 16b. new ResSQL()  
 17b. next(): bool  
 [si T]  
 18ba. getString("horaSalida")  
 19ba. getString("horaLlegada")  
 20ba. getString("comentarios")  
 21ba. getInt("asientos")  
 22b. execSQL("SELECT COUNT(dni) FROM Billeto WHERE fecha LIKE '%fecha%' AND origen LIKE '%origen%' AND destino LIKE '%destino%' AND hora LIKE '%horaSalida%'")  
 23b. new ResSQL()  
 24b. next(): bool  
 [si T]  
 25ba. getInt("númeroReservas")  
 26b. close()  
 27b. close()  
 28b. new Figura151(JSON2)



29b. "Adquirir billete"

30b. quedanPlazas(origen, destino, fecha, hora): boolean

31b. quedanPlazas(origen, destino, fecha, hora): boolean

32b. execSQL("SELECT COUNT(b.dni), COUNT (Asiento.código) FROM Billete AS b NATURAL JOIN Vuelo NATURAL JOIN Asiento WHERE b. origen LIKE '%origen%' AND b.destino LIKE '% destino%' AND b.fecha LIKE '%fecha%' AND b. hora LIKE '%hora%'")

33b. new ResSQL()

34b. next(): boolean

[si T]

35ba. getInt("vendidos")

35\_2ba. getInt("total")

36b. close()

[si no quedan plazas, si F]

37ba. new Figura155()

[si T]

37bb. obtenerAsientosOcupados(origen, destino, fecha, hora): JSON3: {[código: sString, ...]}

38bb. execSQL("SELECT asiento FROM Billete WHERE origen LIKE '%origen%' AND destino LIKE '%destino%' AND fecha LIKE '% fecha%' AND hora LIKE '%hora%'")

39bb. new ResSQL()

40bb. next(): boolean

[si T]

41bba. getString("asiento")

42bb. close()

43bb. new Figura152(origen, destino,

