/isual Paradigm Standard(javierac(Universidad del Pais Vasco))

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)

### Ciudad

-nombre : String -código

### Avión

-matrícula : String

-susAsientos : ListaAsientos

### Asiento

-código

### Persona

-dni : String -nombreApe

-susBilletes : ListaBilletes

# GestorCiudades

-todas : ListaCiudades

### Vuelo

-origen : Ciudad -destino : Ciudad -avión : Avión

-salida : Date

-llegada : Date

-comentarios : String

GestorVuelos -todos : ListaVuelos

-horaSalida

-horaLlegada

## **GestorPersonas**

-todas : ListaPersonas

### Billete

-dni : String -asiento

-origen : Ciudad

-destino : Ciudad -salida : Date

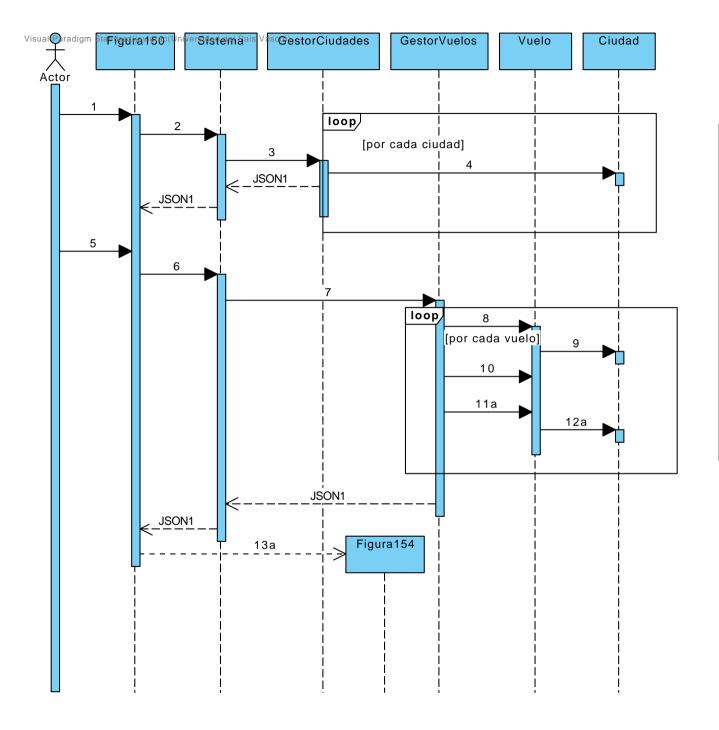
l-localizador

# 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]

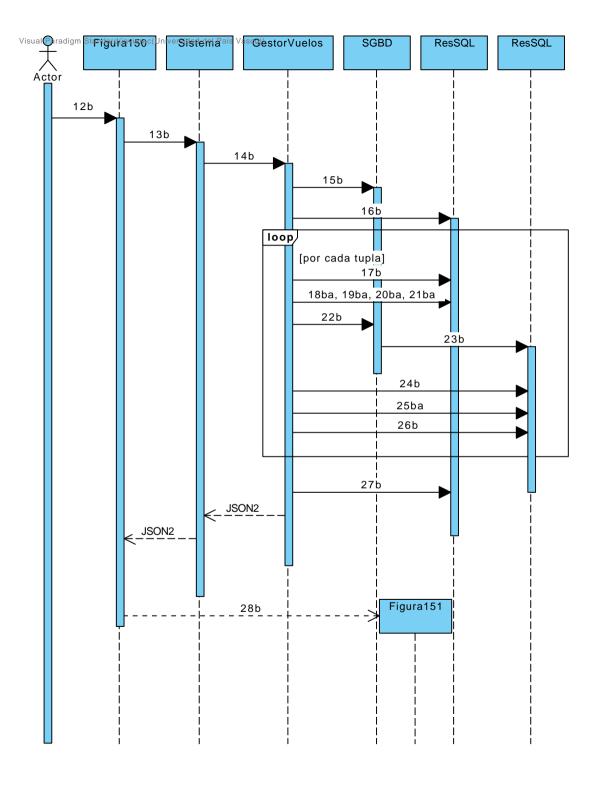
11a. getDestino: String

12a. getDestino: String

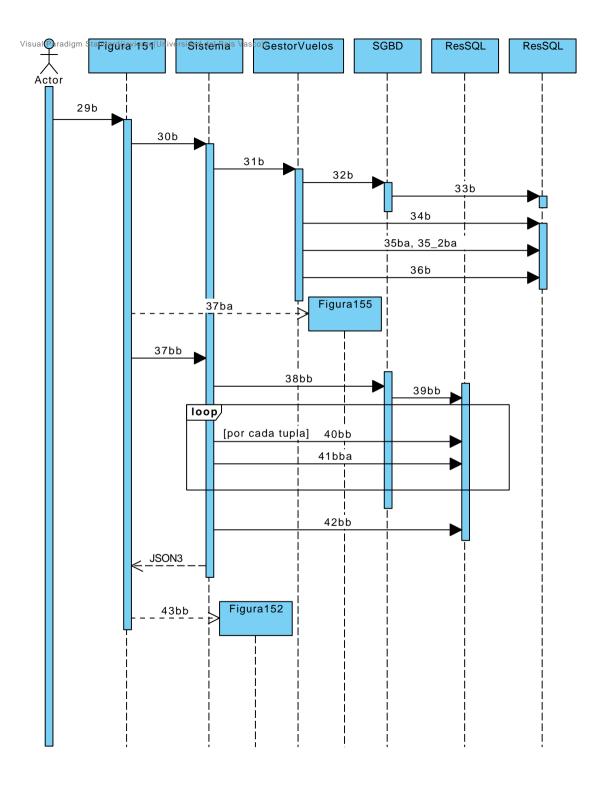
[si no hay destinos]

13a. new Figura154(fecha, origen)

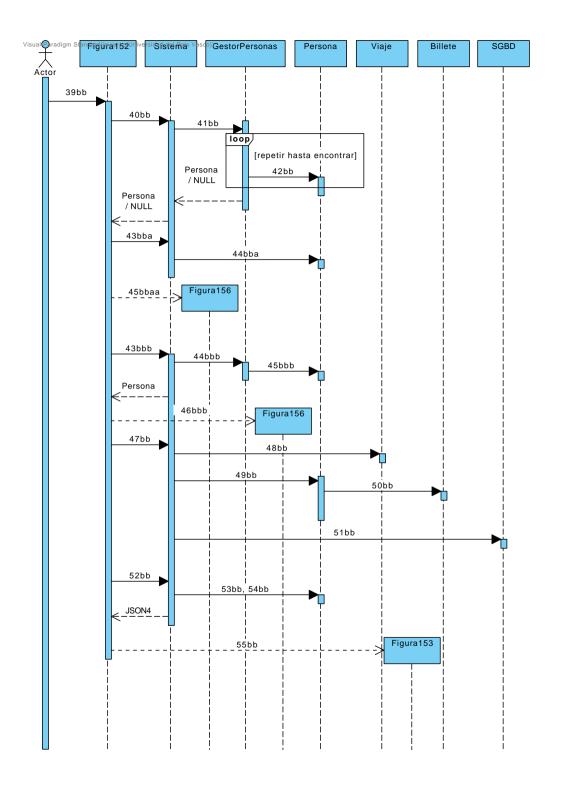
[si hay]



```
12b. Selecciona destino y pulsa "Buscar vuelos".
13b. obtenerVuelos(fecha, origen, destino): ISON2:
{[{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 ResSOL()
17b. next(): bool
[si T]
     18ba. getString("horaSalida")
     19ba. getString("horaLlegada")
    20ba. getString("comentarios")
     21ba. getInt("asientos")
22b. execSQL("SELECT COUNT(dni) FROM Billete
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 Figura 151 (ISON2)
```



```
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 Figura 155()
[si T]
     37bb. obtenerAsientosOcupados(origen,
destino, fecha, hora): JSON3:{[código: sTring,
...]}
     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 Figura 152 (origen, destino,
```



```
39bb. Introduce datos y "confirmar"
40bb. buscarPersona(dni): Persona
41bb. buscarPersona(dni):Persona
42bb. getDNI():String
[si se ha encontrado]
    43bba. tieneDistintoNombre(Persona, nombreApe):
     44bba. getNombre(): String
    [si T, tiene distinto nombre]
          45bbaa. new Figura156()
[40. si no se ha encontrado]
     43bbb. añadirPersona(dni, nombreApe): Persona
    44bbb. añadirPersona(dni, nombreApe): Persona
     45bbb. new Persona(dni, nombreApe)
     46bbb. new Figura157(dni, nombreApe)
[si no existe usuario, 0]
47bb. añadirReserva(Persona, asiento, origen, destino, fecha,
hSalida)
48bb. loc = getLocalizador(dni, asiento, origen, destino,
fechaHora): String
49bb. añadirReserva(asiento, origen, destino, fecha, hSalida,
50bb. new Billete(dni, asiento, origen, destino, fecha, hSalida,
51bb. execSQL("INSERT INTO Billete VALUES('%dni%',
'%asiento%', '%origen%', '%destino%', '%localizador%',
'%fecha%', '%hSalida%'")
52bb. getInfo(Persona): JSON4: {DNI: String, nombreApe:
String}
53bb. getDNI(): String
54bb. getNombreApe(): String
55bb. new Figura153(dni, asiento, origen, destino, fecha,
hora, localizador, nombreApe)
```