

Índice de contenidos

| | |
|---|-----------|
| 1. Descripción del problema | 2 |
| 2. Análisis de requisitos | 2 |
| 2.1. Requisitos de datos | 2 |
| 2.2. Requisitos funcionales | 6 |
| Gestión de eventos | 6 |
| Gestión de servicios | 7 |
| Gestión web | 8 |
| Administración | 8 |
| 2.3. Restricciones semánticas | 9 |
| 2.4. Validación cruzada de requisitos | 9 |
| 3. Diseño | 12 |
| 3.1. Esquema de caja negra | 12 |
| 3.2. Esquema almacén F, esquema almacén D y esquemas externos (0) | 12 |
| Esquema almacén F (DFD0) | 12 |
| Esquema almacén D (ED0) | 13 |
| Esquemas externos (EE0) | 13 |
| 3.3. Esquema almacén F, esquema almacén D y esquemas externos (1) | 15 |
| Esquema almacén F (DFD1) | 15 |
| Proceso 1 - Gestión de eventos | 15 |
| Proceso 2 - Gestión de servicios | 16 |
| Proceso 3 - Gestión web | 16 |
| Proceso 4 - Administración | 16 |
| Esquema almacén D (ED1) | 17 |
| Esquemas externos (EE1) | 17 |
| 3.4. Esquema almacén F, esquema almacén D y esquemas externos (2) | 19 |
| Esquema almacén F (DFD2) | 19 |
| Proceso 1.1 | 19 |
| Proceso 1.2 | 19 |
| Proceso 2.1 | 20 |
| Proceso 3.1 | 20 |
| Proceso 3.2 | 21 |
| Proceso 4.1 | 21 |
| Proceso 4.2 | 22 |
| Esquema almacén D (ED2) | 22 |
| Esquema externo (EE2) | 23 |
| 3.5. Esquema almacén F y E/R final | 25 |
| Esquema almacén F (DFD2) | 25 |
| E/R | 25 |
| 3.6. Paso a tablas | 26 |

| | |
|---------------------------|-----------|
| 4. Implementación | 27 |
| 4.1. Tablas | 27 |
| 4.2. Triggers | 30 |
| 1. Aforo completo | 30 |
| 2. Vehículo no disponible | 30 |
| 3. Personal ya asignado | 31 |
| 4. Escenario ocupado | 31 |

1. Descripción del problema

La empresa BoopFest nos ha encargado el desarrollo, gestión y mantenimiento de su sistema para los festivales que quieren organizar. Su función consiste en almacenar y gestionar los datos relacionados con sus eventos, servicios, personal, página web, etc.

En cada evento habrá varios escenarios entre los que se repartirán los diferentes grupos contratados, dando cada uno un concierto en un horario específico en base a la confirmación del cartel.

En cuanto al trámite de servicios hay que administrar la creación de líneas de transporte y las paradas de cada una, además de gestionar los distintos vehículos que compondrán la flota para facilitar a los asistentes la llegada al recinto donde tienen lugar los conciertos. También se diseñará un mapa del recinto con sus distintos escenarios, puntos de información, aseos, etc. Por otro lado, se administrará la página web, en la que será posible la venta de entradas, el listado de los horarios de los distintos conciertos, junto con el mapa del recinto. Además de la venta de abonos, existirá la posibilidad de comprar un bonobús para utilizar una de las líneas para llegar al recinto.

Finalmente, la administración del personal se llevará a cabo permitiendo la alta o baja de los empleados, la asignación a uno de los festivales disponibles y la gestión de sus horarios laborales.

2. Análisis de requisitos

2.1. Requisitos de datos

- **RD01. Datos del festival.** Se componen de:
 - Nombre
 - Lugar
 - Fecha inicio
 - Fecha fin
 - Aforo máximo
- **RD02. Datos del festival almacenados.** Se componen de:
 - Nombre
 - Lugar
 - Fecha inicio
 - Fecha fin
 - Aforo máximo
 - Grupos
 - Escenarios
 - Personal

- **RD03. Datos de un grupo.** Se componen de:
 - Nombre del grupo
 - Integrantes
 - Género
- **RD04. Datos de un escenario.** Se componen de:
 - Nombre de escenario
 - Identificador de escenario
- **RD05. Datos de una actuación.** Se componen de:
 - Grupo
 - Fecha
 - Hora
 - Duración
 - Escenario
- **RD06. Datos una actuación almacenados.** Se componen de:
 - Grupo
 - Fecha
 - Hora
 - Duración
 - Escenario
- **RD07. Datos de línea de transporte.** Se componen de:
 - Nombre de línea
 - Hora inicio servicio
 - Hora fin servicio
 - Frecuencia de paso
- **RD08. Datos de línea de transporte almacenados.** Se componen de:
 - Nombre de línea
 - Hora inicio servicio
 - Hora fin servicio
 - Frecuencia de paso
 - Paradas
 - Vehículos
- **RD09. Datos de parada.** Se componen de:
 - Nombre de parada
 - Código de parada
 - Ubicación
- **RD10. Datos de vehículo.** Se componen de:
 - Matrícula
 - Tipo
 - Marca
 - Modelo
 - Capacidad

- **RD11. Datos de mapa.** Se componen de:
 - Mapa de ciudad
- **RD12. Datos de mapa almacenados.** Se componen de:
 - Mapa de ciudad
 - Líneas de transporte
 - Escenarios
- **RD13. Datos de abono.** Se componen de:
 - Nombre de asistente
 - D.N.I.
 - Tarjeta de crédito
 - Tipo de entrada
- **RD14. Datos de abono almacenados.** Se componen de:
 - Nombre de asistente
 - D.N.I.
 - Tipo de entrada
 - Precio
 - Abonos vendidos
- **RD15. Datos de abono salida.** Se componen de:
 - Nombre de asistente
 - Tipo de entrada
 - Precio
- **RD16. Datos del horario.** Se componen de:
 - Nombre festival
- **RD17. Datos del horario de salida.** Se componen de:
 - Días
 - Horas
 - Escenarios
 - Grupos
- **RD18. Datos de visualización mapa.** Se compone de:
 - Nombre festival
- **RD19. Datos de visualización mapa salida.** Se componen de:
 - Mapa final
- **RD20. Datos de bonobús.** Se componen de:
 - Nombre
 - D.N.I.
 - Tarjeta de crédito
 - Línea de bus

- **RD21. Datos de bonobús almacenados.** Se componen de:
 - Nombre
 - D.N.I.
 - Línea de bus
 - Precio
- **RD22. Datos de bonobús salida.** Se componen de:
 - Nombre
 - D.N.I.
 - Línea de bus
 - Precio
- **RD23. Datos de alta de empleados.** Se componen de:
 - Nombre
 - Apellidos
 - DNI
 - Tipo de contrato
 - Función
 - Sueldo
- **RD24. Datos de empleados almacenados.** Se componen de:
 - Nombre
 - Apellidos
 - DNI
 - Código de empleado
 - Contraseña
 - Tipo de contrato
 - Finiquito
 - Nombre festival
- **RD25. Datos de baja de empleados.** Se componen de:
 - Nombre
 - Apellidos
 - DNI
 - Código de empleado
 - Finiquito
- **RD26. Datos de asignar personal a festival.** Se componen de:
 - Código de empleado
 - Nombre festival
- **RD27. Datos de asignar horarios a personal.** Se componen de:
 - Código de empleado
 - Horario laboral
- **RD28. Datos de horarios de personal almacenados.** Se componen de:
 - Código de empleado
 - Días de la semana laborales
 - Horario laboral

- **RD29. Datos de mostrar horarios de personal.** Se componen de:
 - Código de empleado
- **RD30. Datos de mostrar horarios de personal salida.** Se componen de:
 - Código de empleado
 - Días de la semana laborales
 - Horario laboral
- **RD31. Datos almacenados de escenario:** Se componen de:
 - Nombre de escenario
 - Identificador de escenario
- **RD32. Datos almacenados de un grupo.** Se componen de:
 - Nombre del grupo
 - Integrantes
 - Género
- **RD33. Datos almacenados de parada.** Se componen de:
 - Nombre de parada
 - Código de parada
 - Ubicación
- **RD34. Datos almacenados de vehículo.** Se componen de:
 - Matrícula
 - Tipo
 - Marca
 - Modelo
 - Capacidad

2.2. Requisitos funcionales

Gestión de eventos

- **RF01. Crear festival.** Se crea un festival proporcionando los siguientes datos:
 - RD01
 El sistema, en respuesta, almacenará los siguientes datos:
 - RD02
- **RF02. Añadir grupo a festival.** Añade un grupo a un festival proporcionando los siguientes datos:
 - RD03
 El sistema, en respuesta, consultará los siguientes datos:
 - RD32
 A continuación, en respuesta, modificará los siguientes datos:
 - RD02

- **RF03. Designar escenario.** Crea un escenario del recinto proporcionando los siguientes datos:
 - RD04
 El sistema, en respuesta, consultará los siguientes datos:
 - RD31
 A continuación, en respuesta, modificará los siguientes datos:
 - RD02

- **RF04. Añadir grupo a escenario.** Asigna un grupo al escenario donde va a actuar proporcionando los siguientes datos:
 - RD05
 El sistema, en respuesta, consultará los siguientes datos:
 - RD32
 A continuación, modificará los siguientes datos:
 - RD06

Gestión de servicios

- **RF05. Crear línea de transporte.** Se traza una línea de transporte de llegada al festival proporcionando los siguientes datos:
 - RD07
 El sistema, en respuesta, almacenará los siguientes datos:
 - RD08

- **RF06. Añadir parada a línea.** Agrega un lugar de parada para la recogida de pasajeros proporcionando los siguientes datos:
 - RD09
 El sistema, en respuesta, consultará los siguientes datos:
 - RD33
 A continuación, en respuesta, modificará los siguientes datos:
 - RD08

- **RF07. Añadir vehículo a línea.** Asocia un nuevo vehículo a cada línea para realizar el transporte proporcionando los siguientes datos:
 - RD10
 El sistema, en respuesta, consultará los siguientes datos:
 - RD34
 A continuación, en respuesta, modificará los siguientes datos:
 - RD08

- **RF08. Crear mapa.** Origina un mapa del recinto y líneas proporcionando los siguientes datos:
 - RD11
 El sistema, en respuesta, consultará los siguientes datos:
 - RD08
 A continuación, el sistema almacenará los siguientes datos:
 - RD12

Gestión web

- **RF09. Vender abono.** Gestiona la venta de entradas (normales y VIP) proporcionando los siguientes datos:
 - RD13El sistema, en respuesta, almacenará los siguientes datos:
 - RD14A continuación, el sistema devolverá los siguientes datos:
 - RD15
- **RF10. Mostrar horarios.** Muestra los horario de los conciertos del festival proporcionando los siguientes datos:
 - RD16El sistema, en respuesta, consultará los siguientes datos:
 - RD06A continuación, el sistema devolverá los siguientes datos:
 - RD17
- **RF11. Mostrar mapa.** Muestra el mapa del festival proporcionando los siguientes datos:
 - RD18El sistema, en respuesta, consultará los siguientes datos:
 - RD12A continuación, el sistema devolverá los siguientes datos:
 - RD19
- **RF12. Vender bonobús.** Gestiona la venta de tickets para las distintas líneas de autobús para trasladarse de la ciudad al festival proporcionando los siguientes datos:
 - RD20El sistema, en respuesta, almacenará los siguientes datos:
 - RD21A continuación, el sistema devolverá los siguientes datos:
 - RD22

Administración

- **RF13. Alta personal.** Gestiona el alta de un/a trabajador/a proporcionando los siguientes datos:
 - RD23El sistema, en respuesta, almacenará los siguientes datos:
 - RD24
- **RF14. Baja personal** Gestiona la baja en el sistema de un empleado proporcionando los siguientes datos:
 - RD25El sistema, en respuesta, eliminará los siguientes datos:
 - RD24

- **RF15. Asignar personal a festival.** Añade al festival un/a empleado/a proporcionando los siguientes datos:
 - RD26
 El sistema, en respuesta, modificará los siguientes datos:
 - RD02
- **RF16. Asignar horario a empleado.** Establece el horario laboral del personal durante el festival, previo a éste y a posteriori, proporcionando los siguientes datos:
 - RD27
 El sistema, en respuesta, almacenará los siguientes datos:
 - RD28
- **RF17. Mostrar horario de un empleado.** Cualquier empleado puede consultar su horario laboral, proporcionando los siguientes datos:
 - RD29
 El sistema, en respuesta, consultará los siguientes datos:
 - RD28
 A continuación, el sistema devolverá los siguientes datos:
 - RD30

2.3. Restricciones semánticas

- **RS1.** No se pueden asignar varios grupos a un mismo escenario y a la misma hora.
- **RS2.** Un grupo no puede estar en dos actuaciones a la vez.
- **RS3.** No se pueden vender más entradas que el aforo máximo permitido del recinto.

2.4. Validación cruzada de requisitos

| RD | Entrada | Manejo | Salida |
|------|---------|------------------------|--------|
| RD01 | RF01 | | |
| RD02 | | RF01, RF02, RF03, RF15 | |
| RD03 | RF02 | | |
| RD04 | RF03 | | |
| RD05 | RF04 | | |
| RD06 | | RF04, RF10 | |
| RD07 | RF05 | | |
| RD08 | | RF05, RF06, RF07, RF08 | |
| RD09 | RF06 | | |

| | | | |
|------|------|------------|------|
| RD10 | RF07 | | |
| RD11 | RF08 | | |
| RD12 | | RF08, RF11 | |
| RD13 | RF09 | | |
| RD14 | | RF09 | |
| RD15 | | | RF09 |
| RD16 | RF10 | | |
| RD17 | | | RF10 |
| RD18 | RF11 | | |
| RD19 | | | RF11 |
| RD20 | RF12 | | |
| RD21 | | RF12 | |
| RD22 | | | RF12 |
| RD23 | RF13 | | |
| RD24 | | RF13, RF14 | |
| RD25 | RF14 | | |
| RD26 | RF15 | | |
| RD27 | RF16 | | |
| RD28 | | RF16, RF17 | |
| RD29 | | RF17 | |
| RD30 | | | RF17 |
| RD31 | | RF03 | |
| RD32 | | RF02, RF04 | |
| RD33 | | RF06 | |
| RD34 | | RF07 | |

Tabla 1. Requisitos de datos.

| RF | Entrada | Manejo | Salida |
|------|---------|------------|--------|
| RF01 | RD01 | RD02 | |
| RF02 | RD03 | RD32, RD02 | |

| | | | |
|------|------|------------|------|
| RF03 | RD04 | RD31, RD02 | |
| RF04 | RD05 | RD32, RD06 | |
| RF05 | RD07 | RD08 | |
| RF06 | RD09 | RD33, RD08 | |
| RF07 | RD10 | RD34, RD08 | |
| RF08 | RD11 | RD12 | |
| RF09 | RD13 | RD14 | RD15 |
| RF10 | RD16 | RD06 | RD17 |
| RF11 | RD18 | RD12 | RD19 |
| RF12 | RD20 | RD21 | RF22 |
| RF13 | RD23 | RD24 | |
| RF14 | RD25 | RD24 | |
| RF15 | RD26 | RD02 | |
| RF16 | RD27 | RD28 | |
| RF17 | RD29 | RD28 | RD30 |

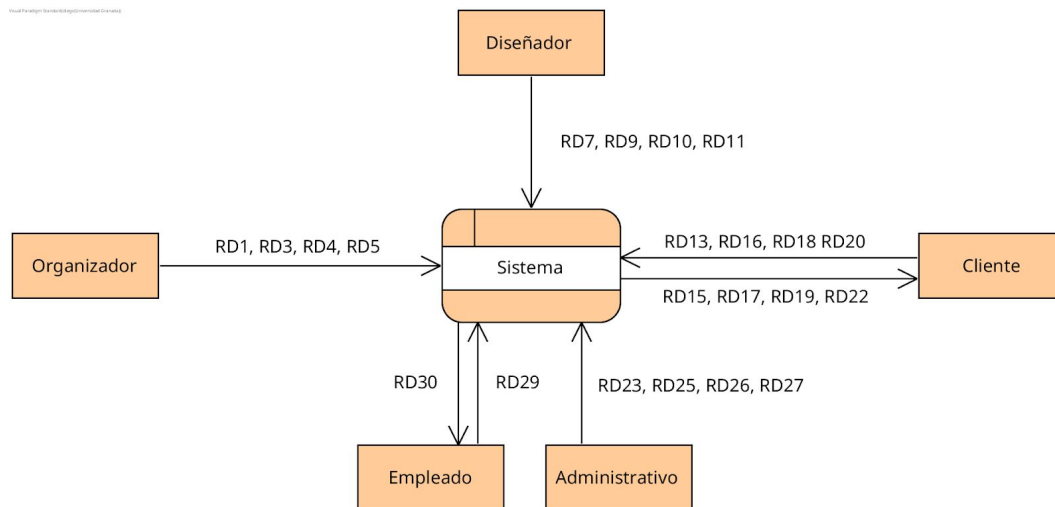
Tabla 2. Requisitos funcionales

| RS | RF | RDs |
|-----|------|------------|
| RS1 | RF03 | RD07, RD09 |
| RS2 | RF04 | RD05, RD09 |
| RS3 | RF09 | RD02 |

Tabla 3. Restricciones semánticas.

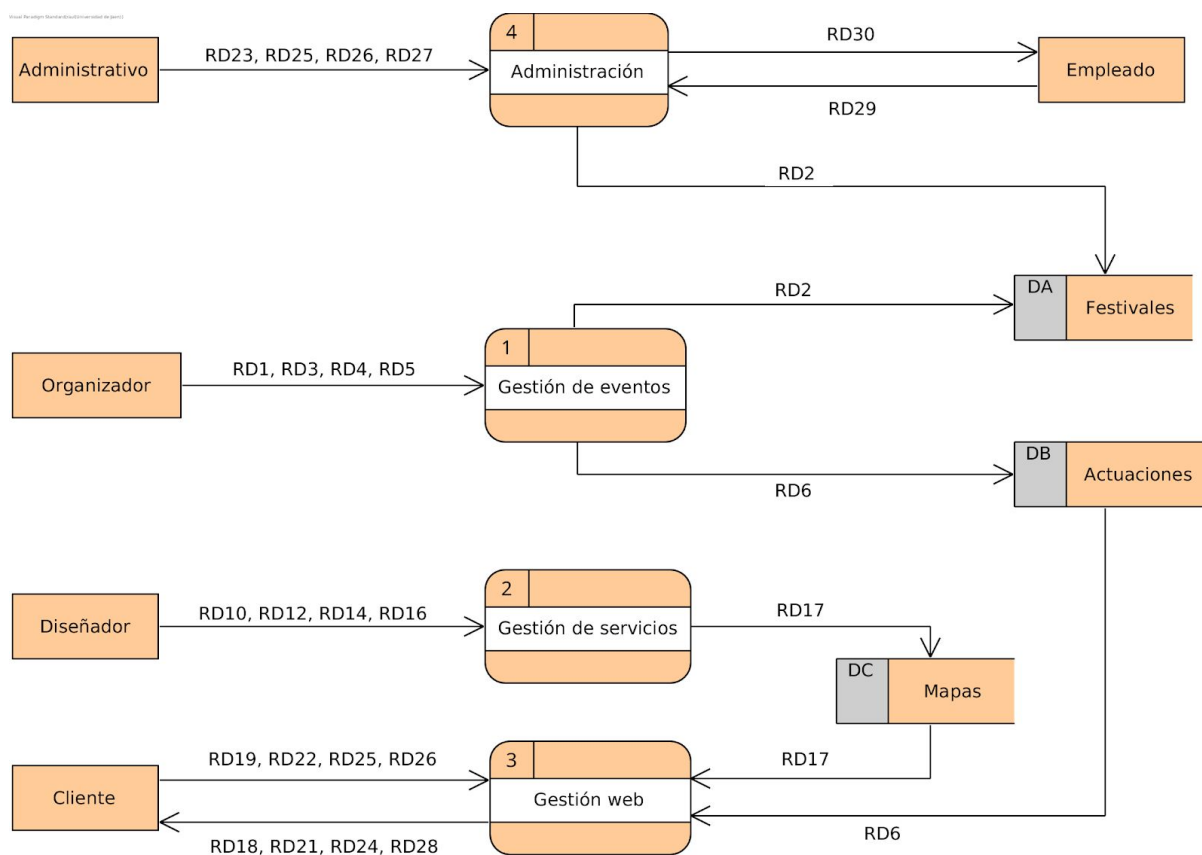
3. Diseño

3.1. Esquema de caja negra

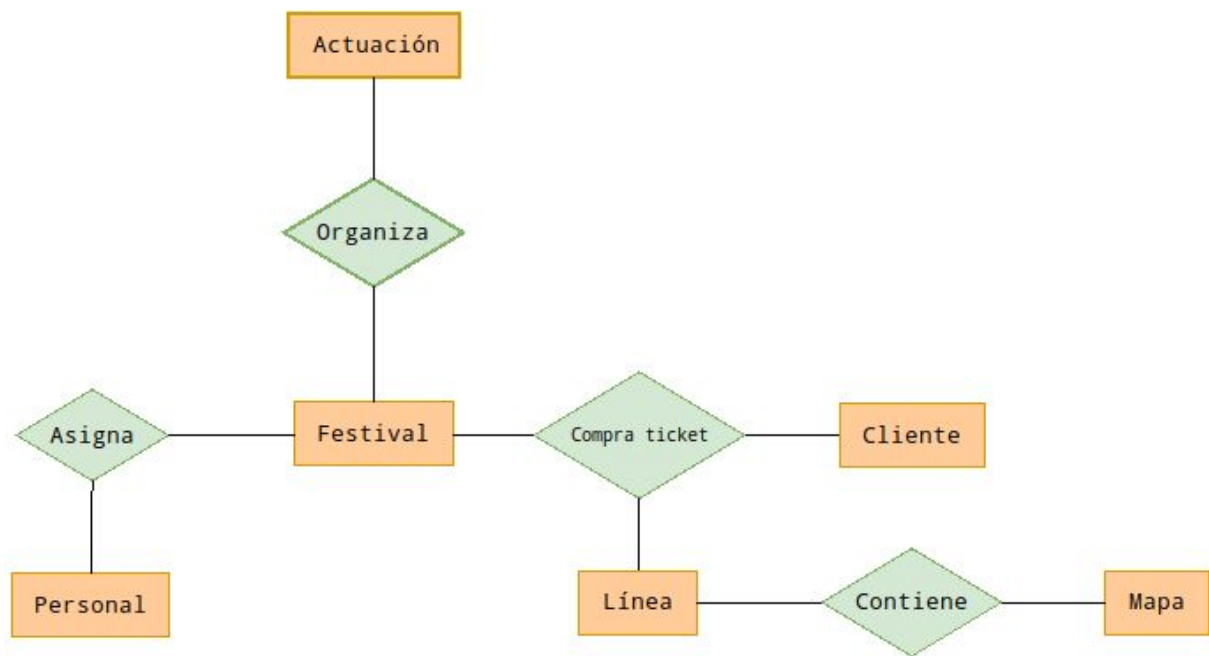


3.2. Esquema almacén F, esquema almacén D y esquemas externos (0)

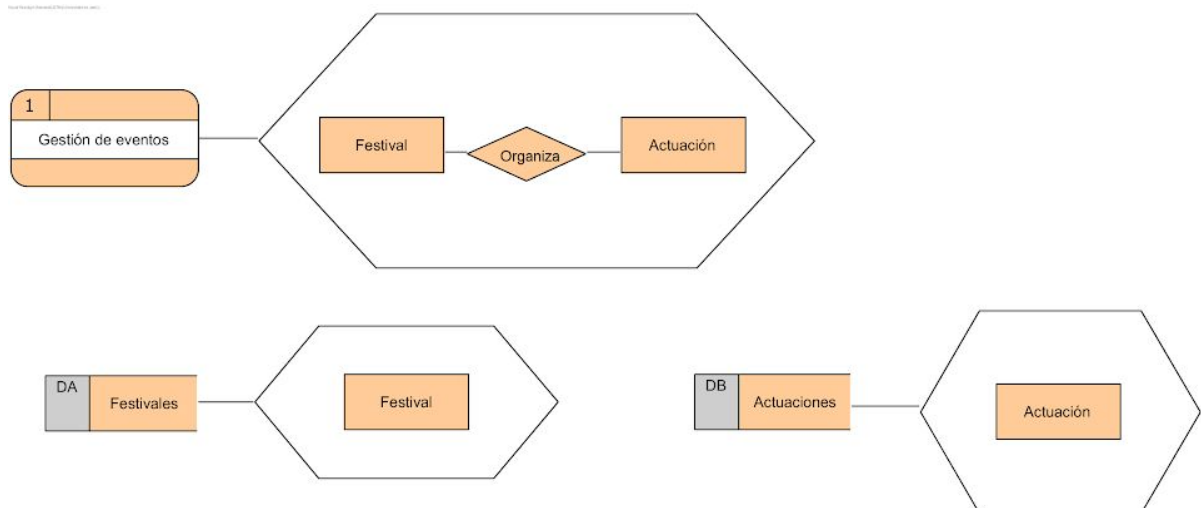
Esquema almacén F (DFD0)

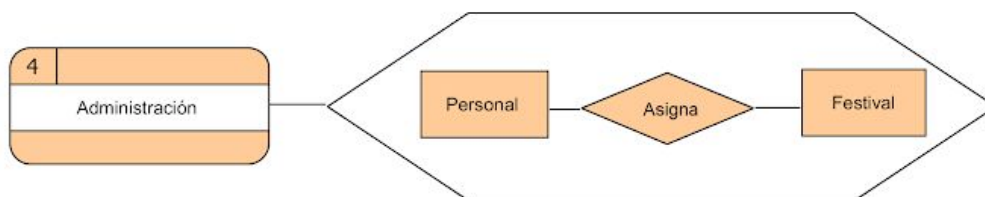
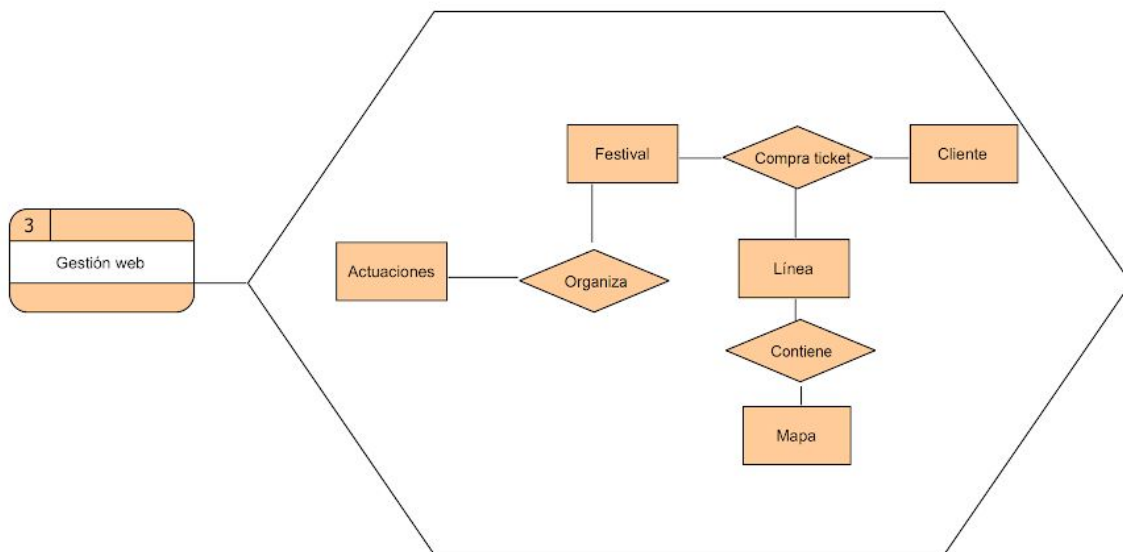
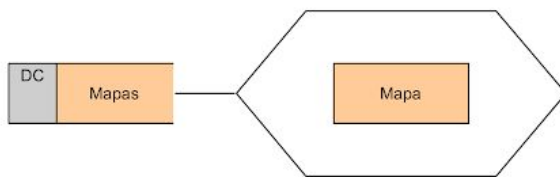
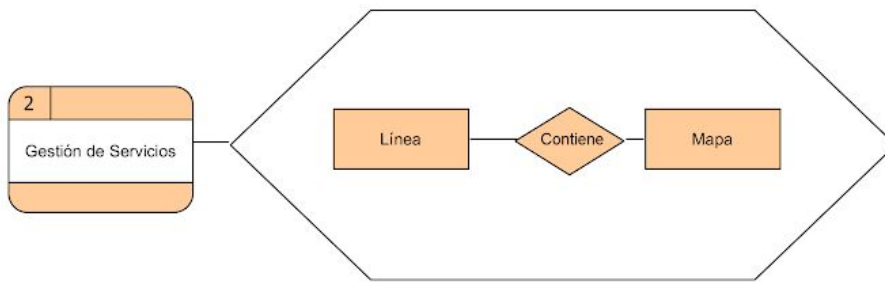


Esquema almacén D (ED0)



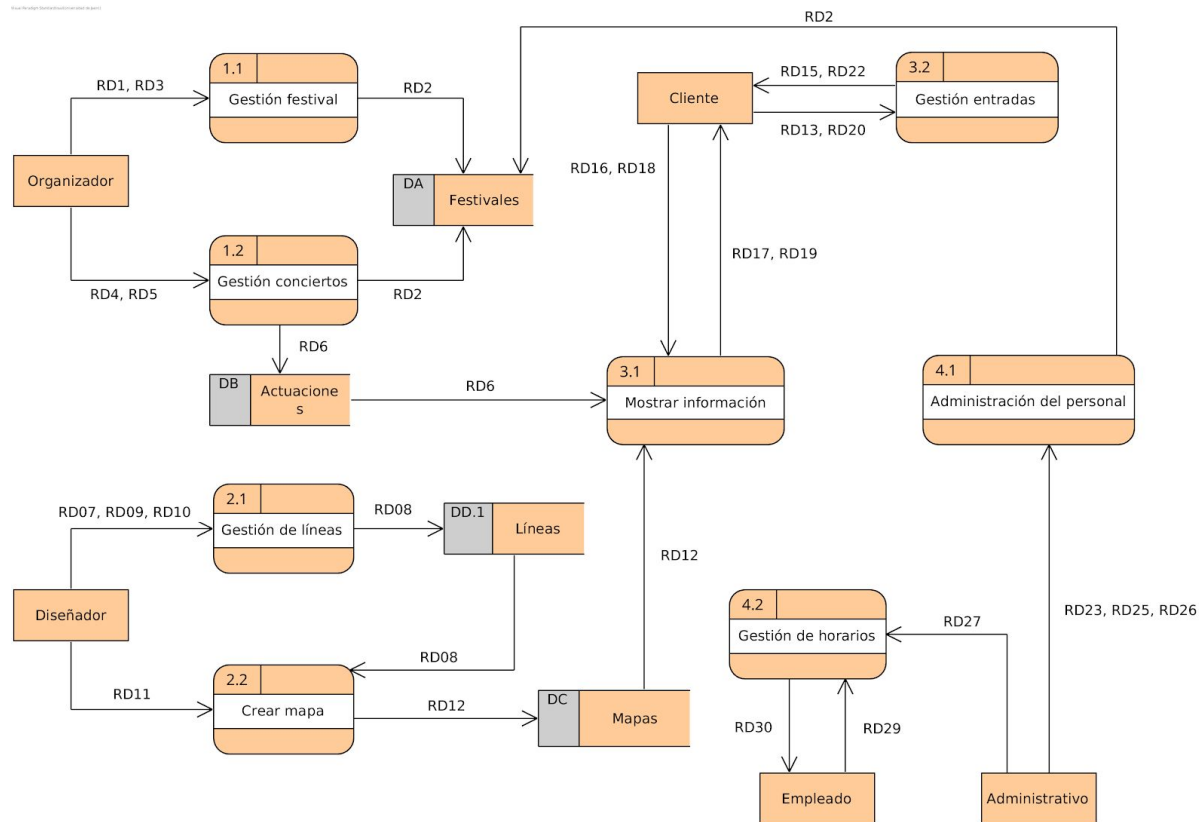
Esquemas externos (EE0)



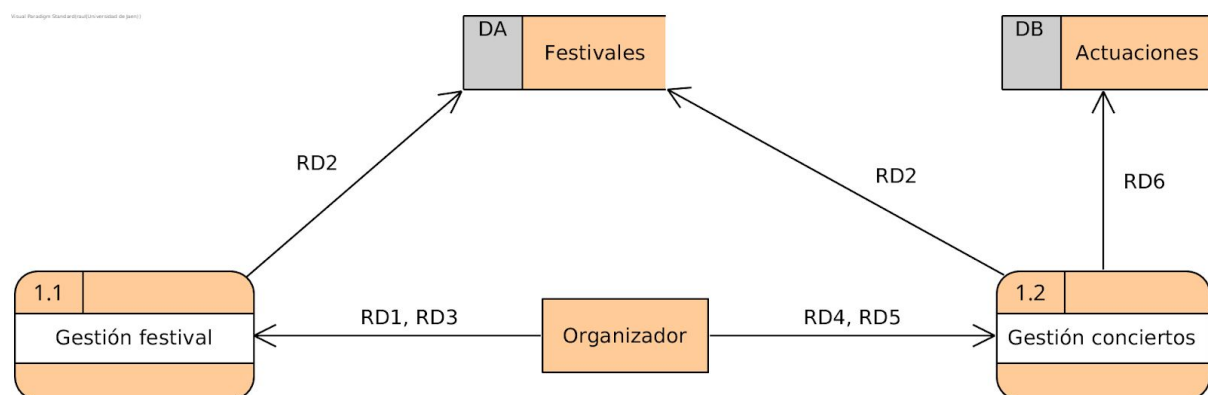


3.3. Esquema almacén F, esquema almacén D y esquemas externos (1)

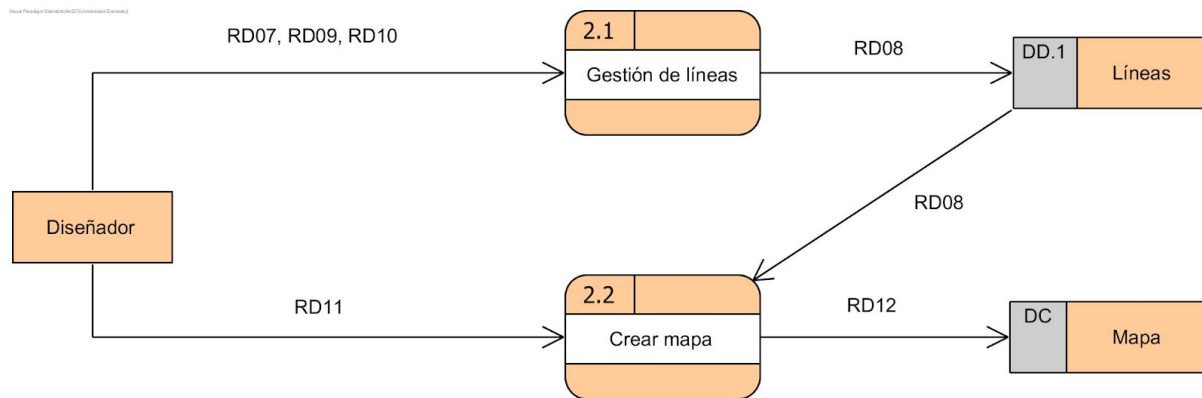
Esquema almacén F (DFD1)



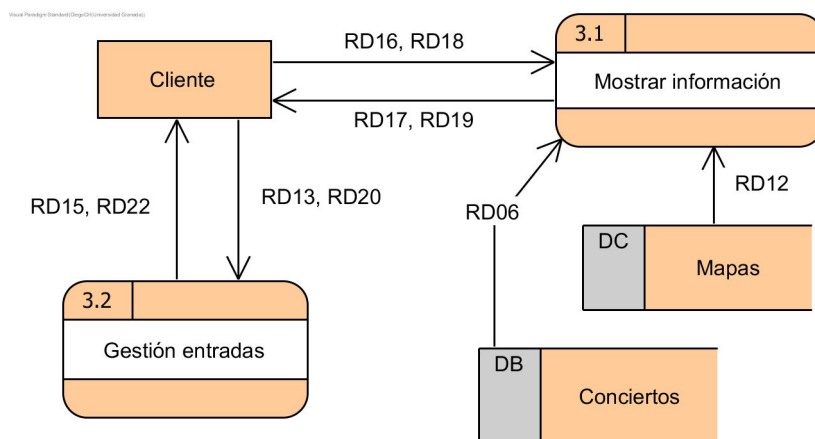
Proceso 1 - Gestión de eventos



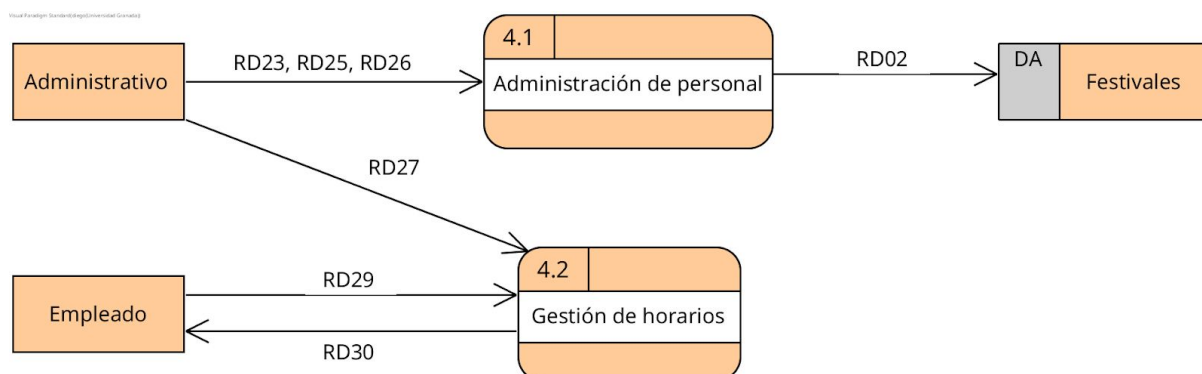
Proceso 2 - Gestión de servicios



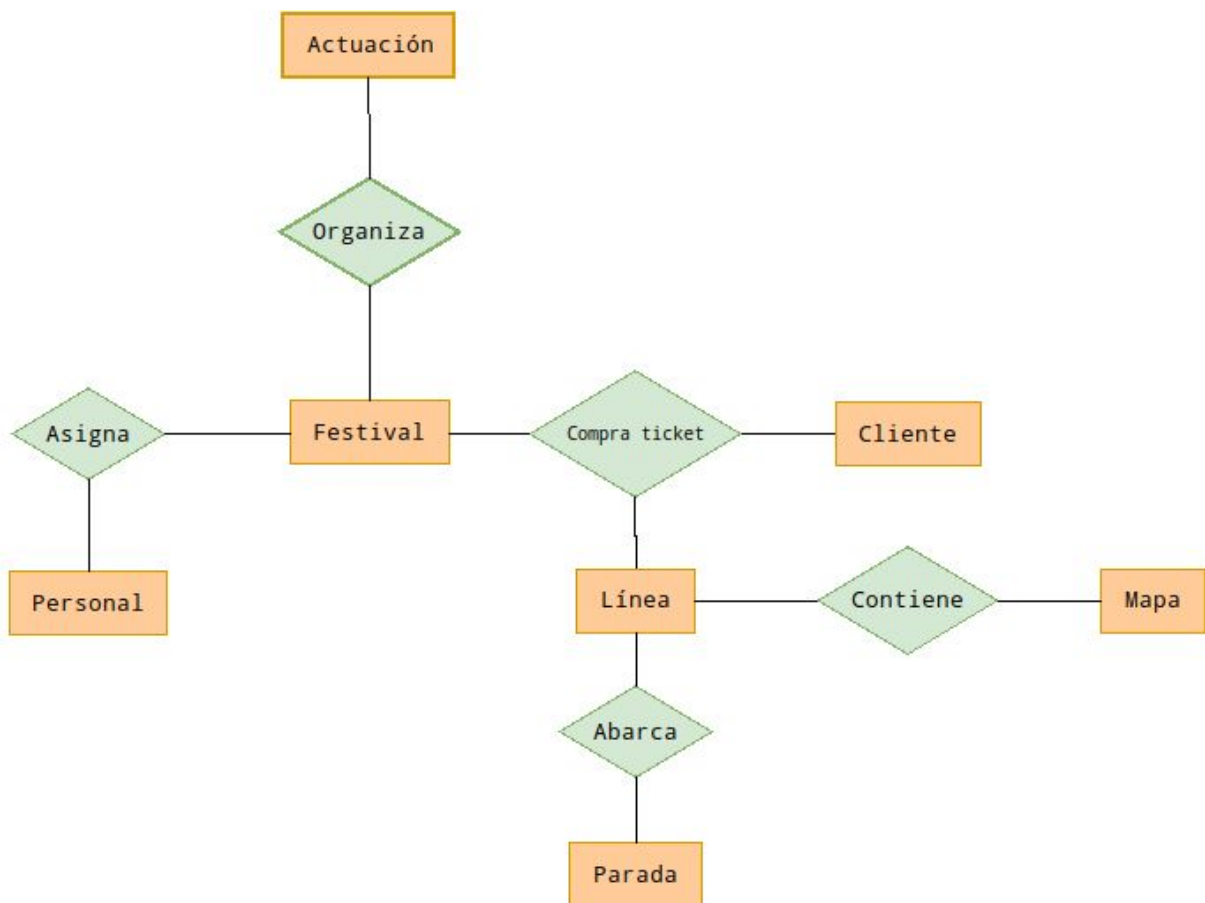
Proceso 3 - Gestión web



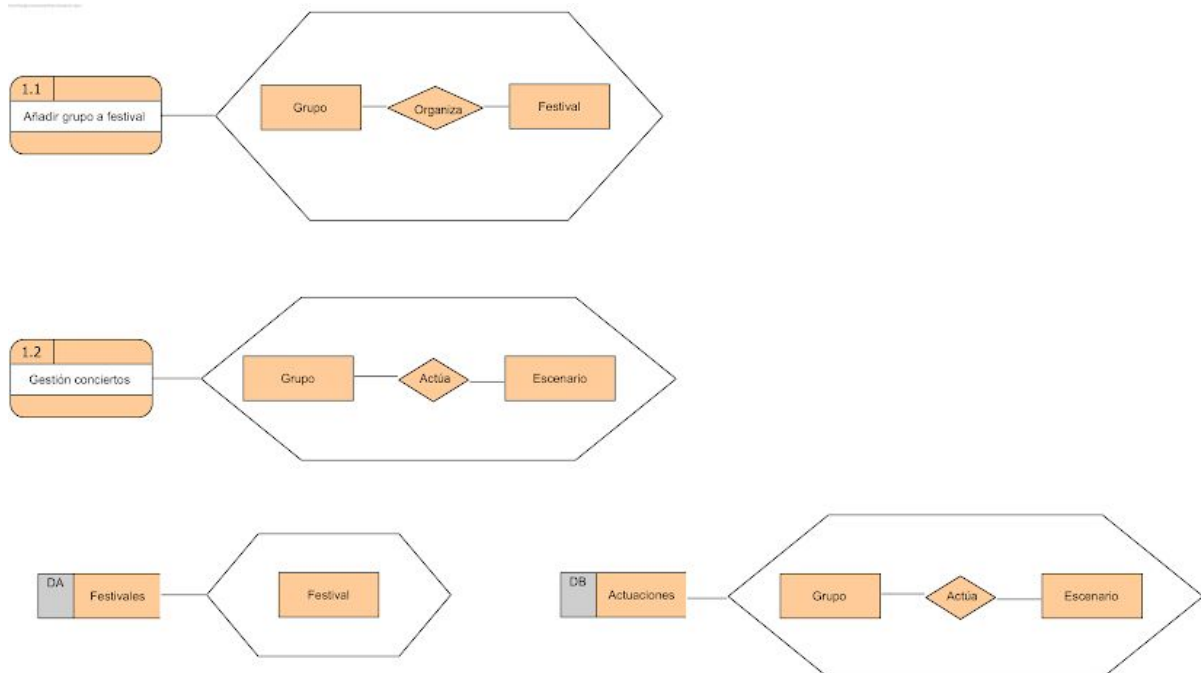
Proceso 4 - Administración

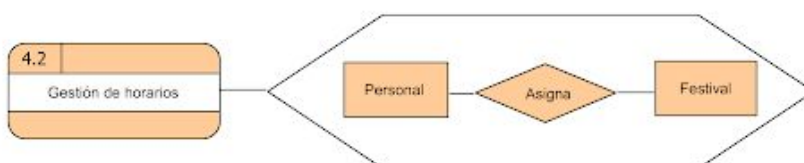
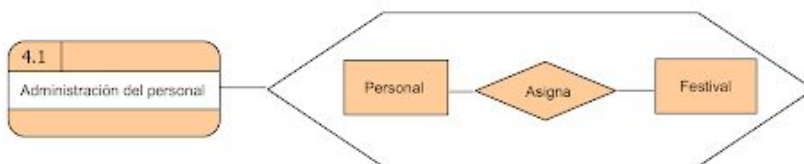
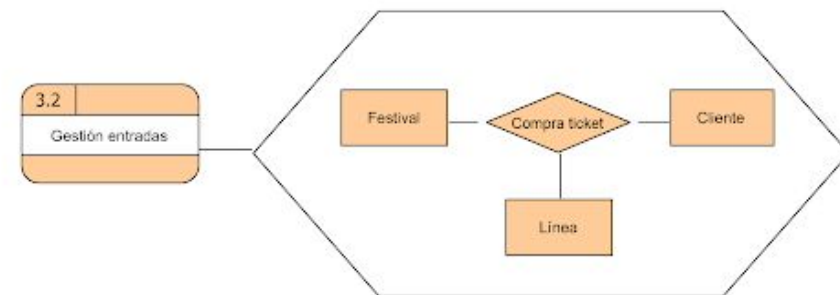
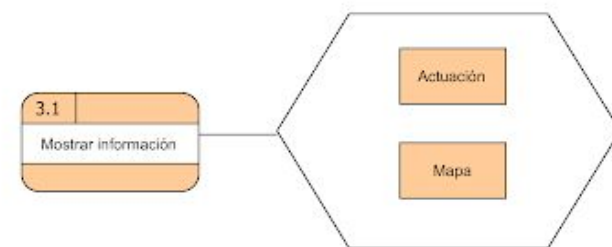
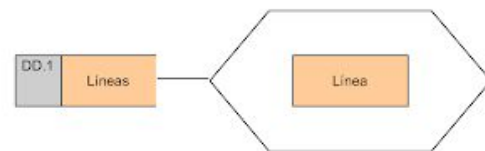
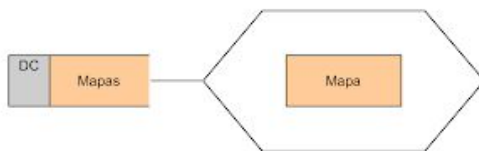
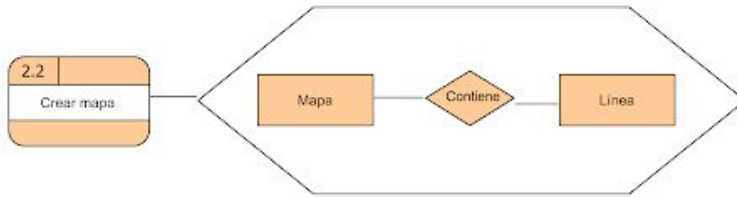
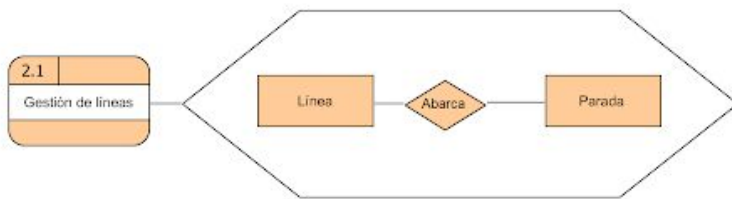


Esquema almacén D (ED1)



Esquemas externos (EE1)



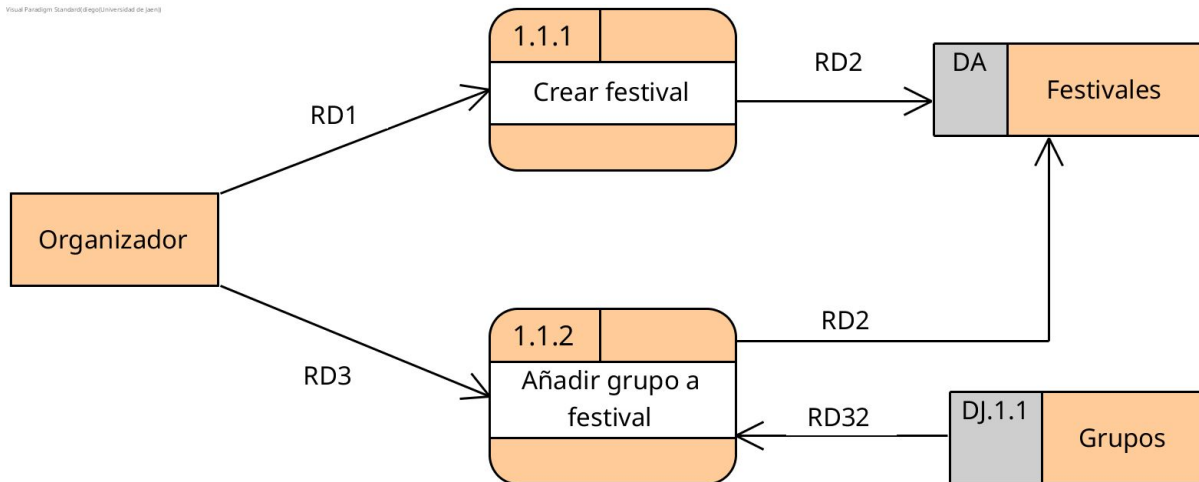


3.4. Esquema almacén F, esquema almacén D y esquemas externos (2)

Esquema almacén F (DFD2)

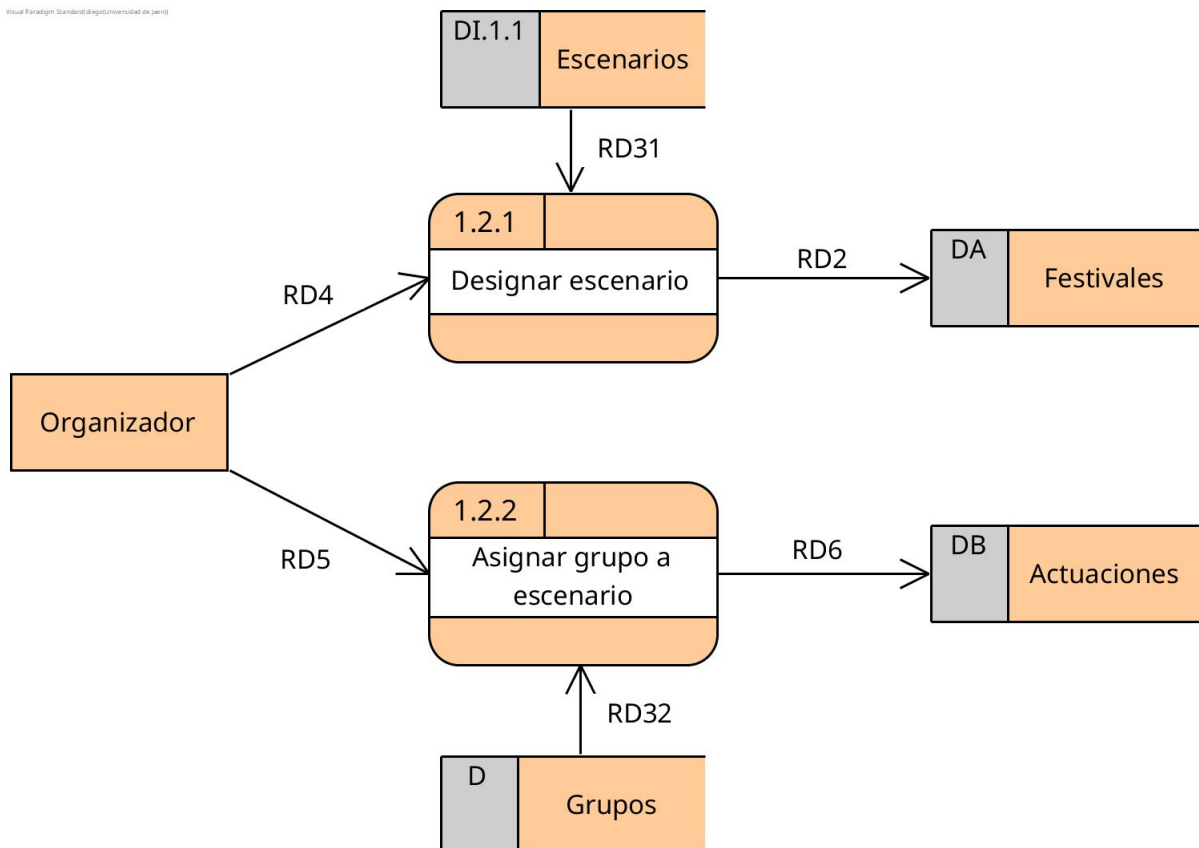
Proceso 1.1

Visual Paradigm Standard (desplazamiento de jarrón)



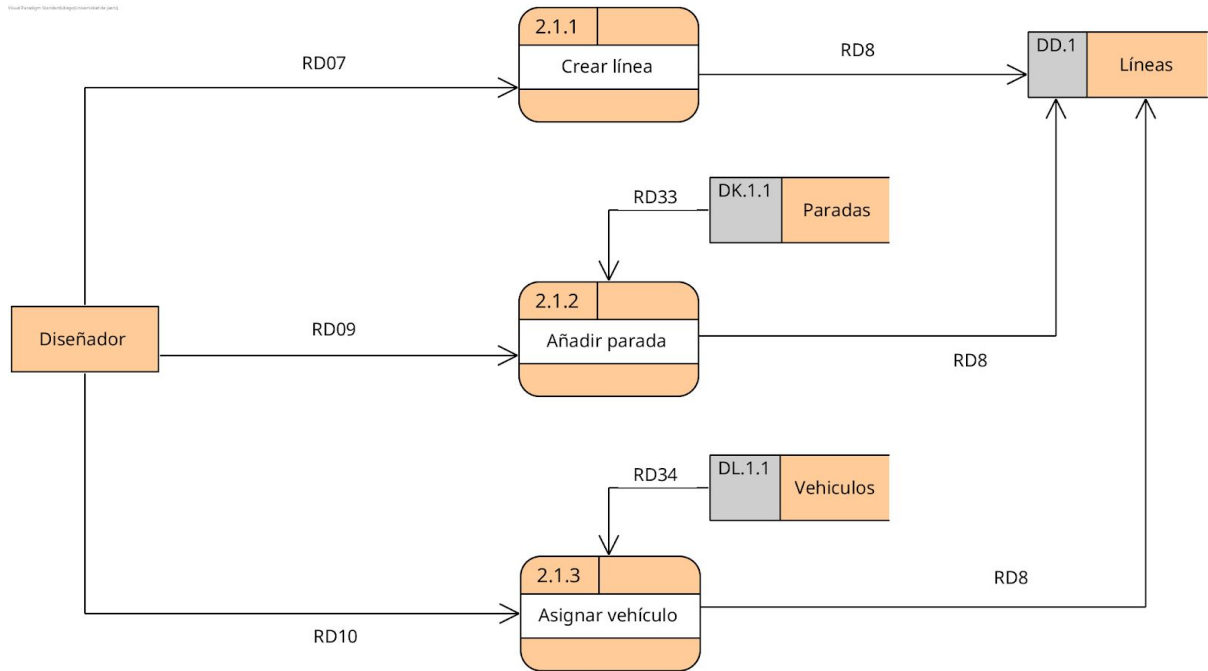
Proceso 1.2

Visual Paradigm Standard (desplazamiento de jarrón)



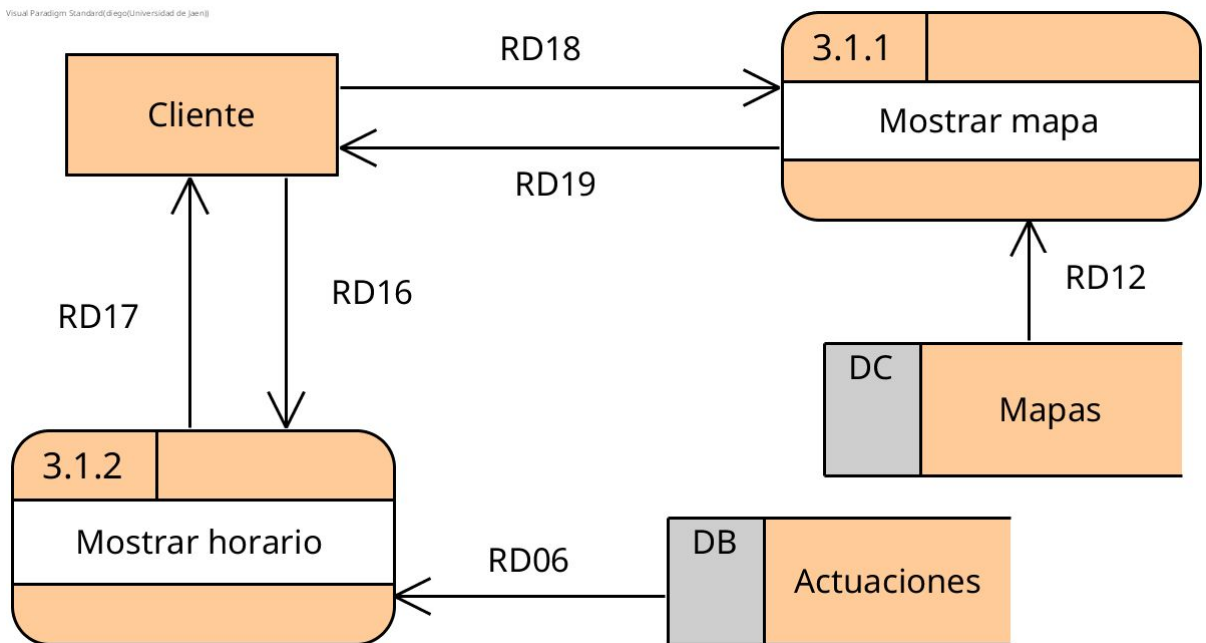
Proceso 2.1

Visual Paradigm Standard (diego(Universidad de Jaén))



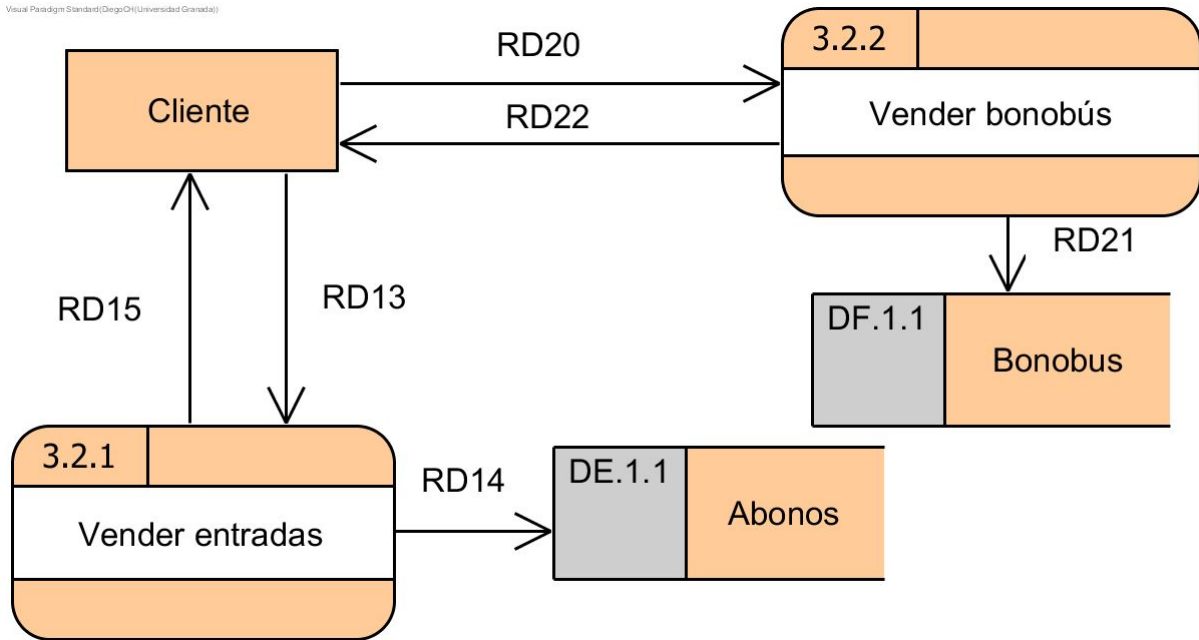
Proceso 3.1

Visual Paradigm Standard (diego(Universidad de Jaén))



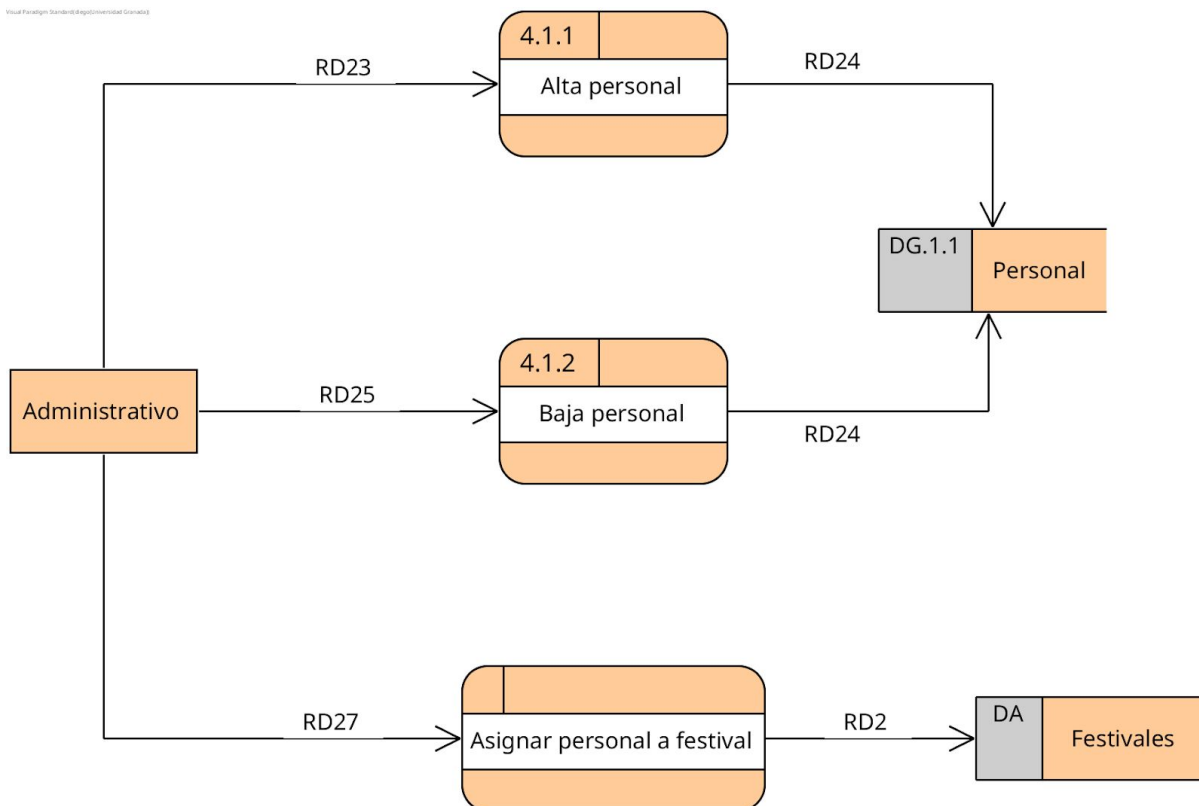
Proceso 3.2

Visual Paradigm Standard (Diego Quijano/Universidad Granada)



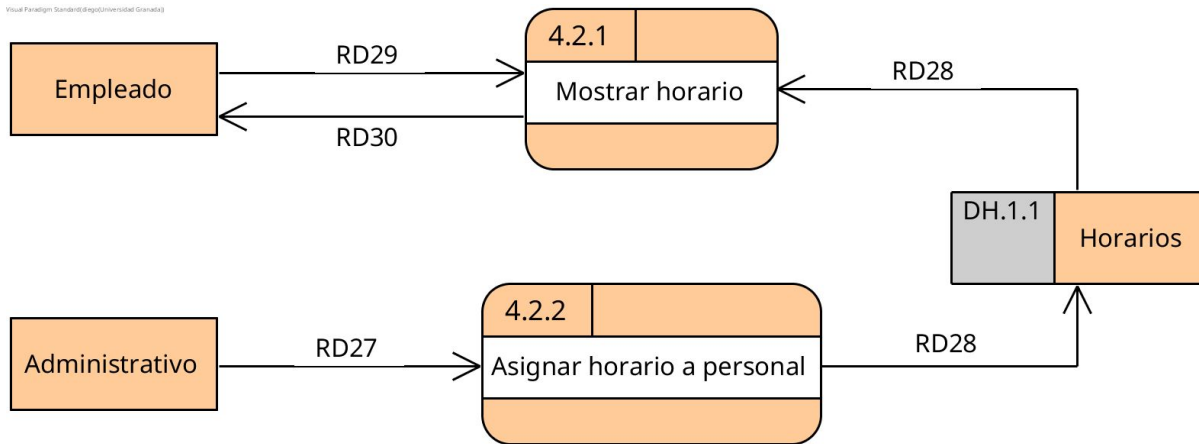
Proceso 4.1

Visual Paradigm Standard (Diego Quijano/Universidad Granada)

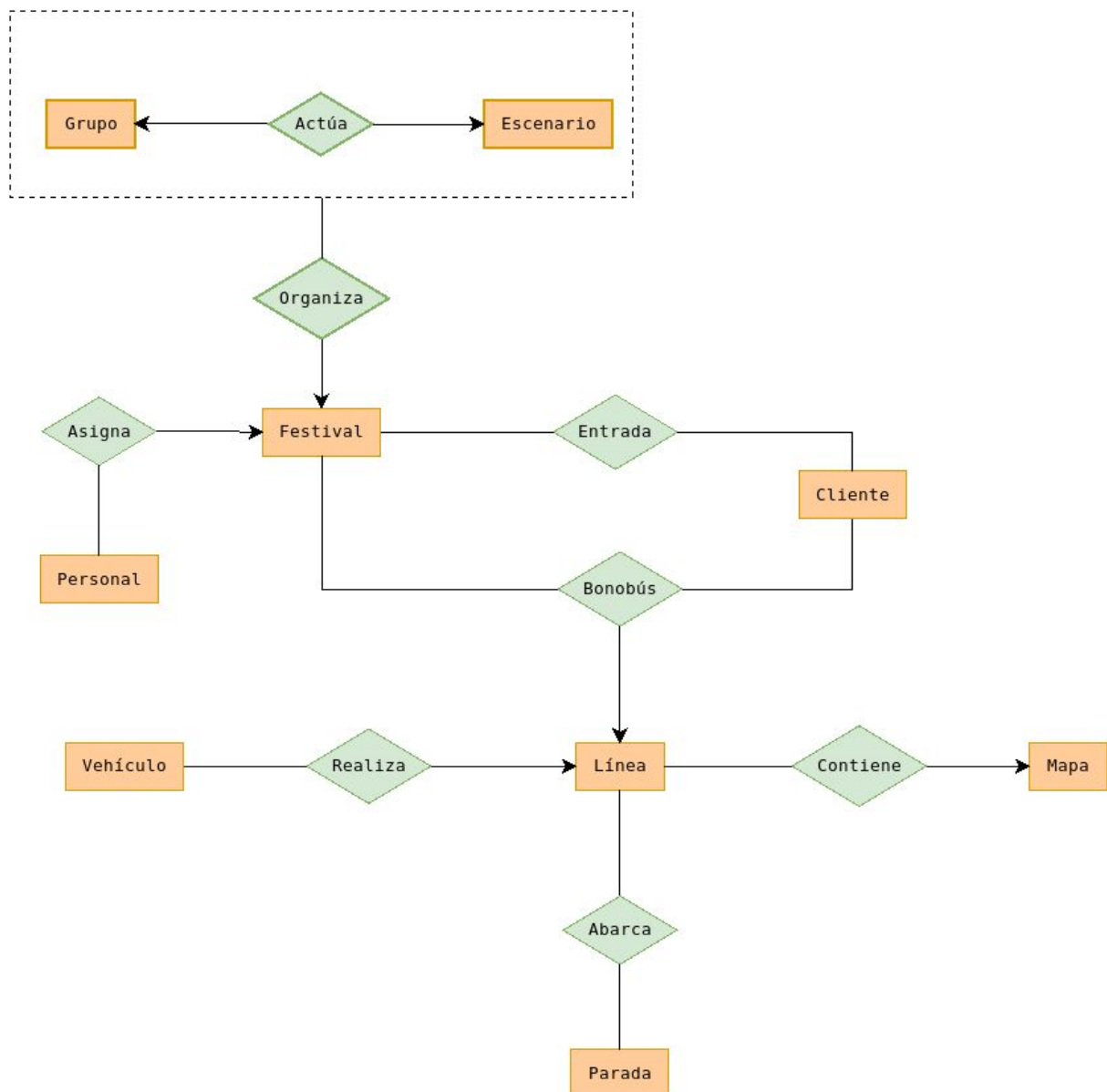


Proceso 4.2

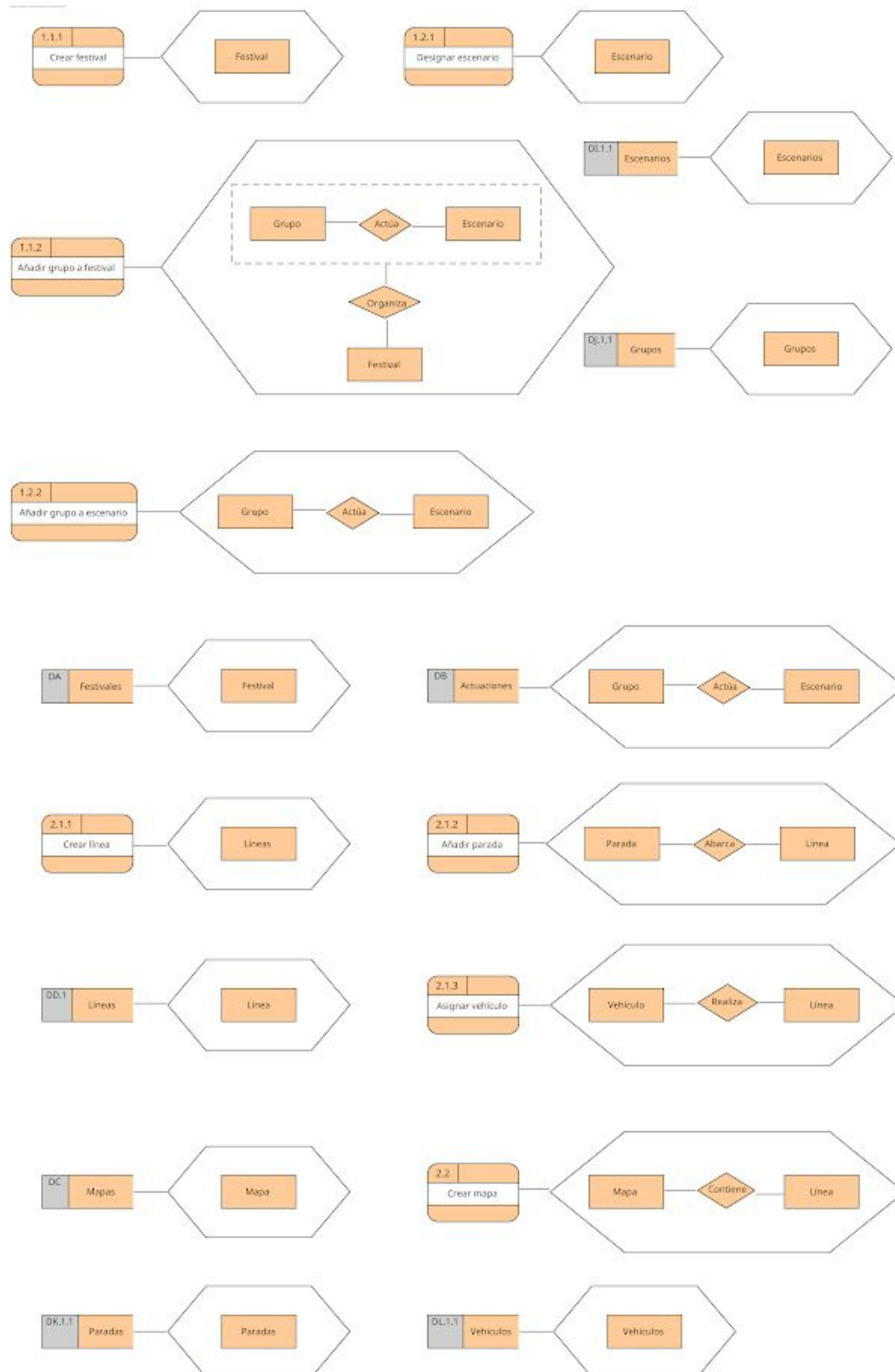
Visual Paradigm Standard (dmg)(Universidad Granada)

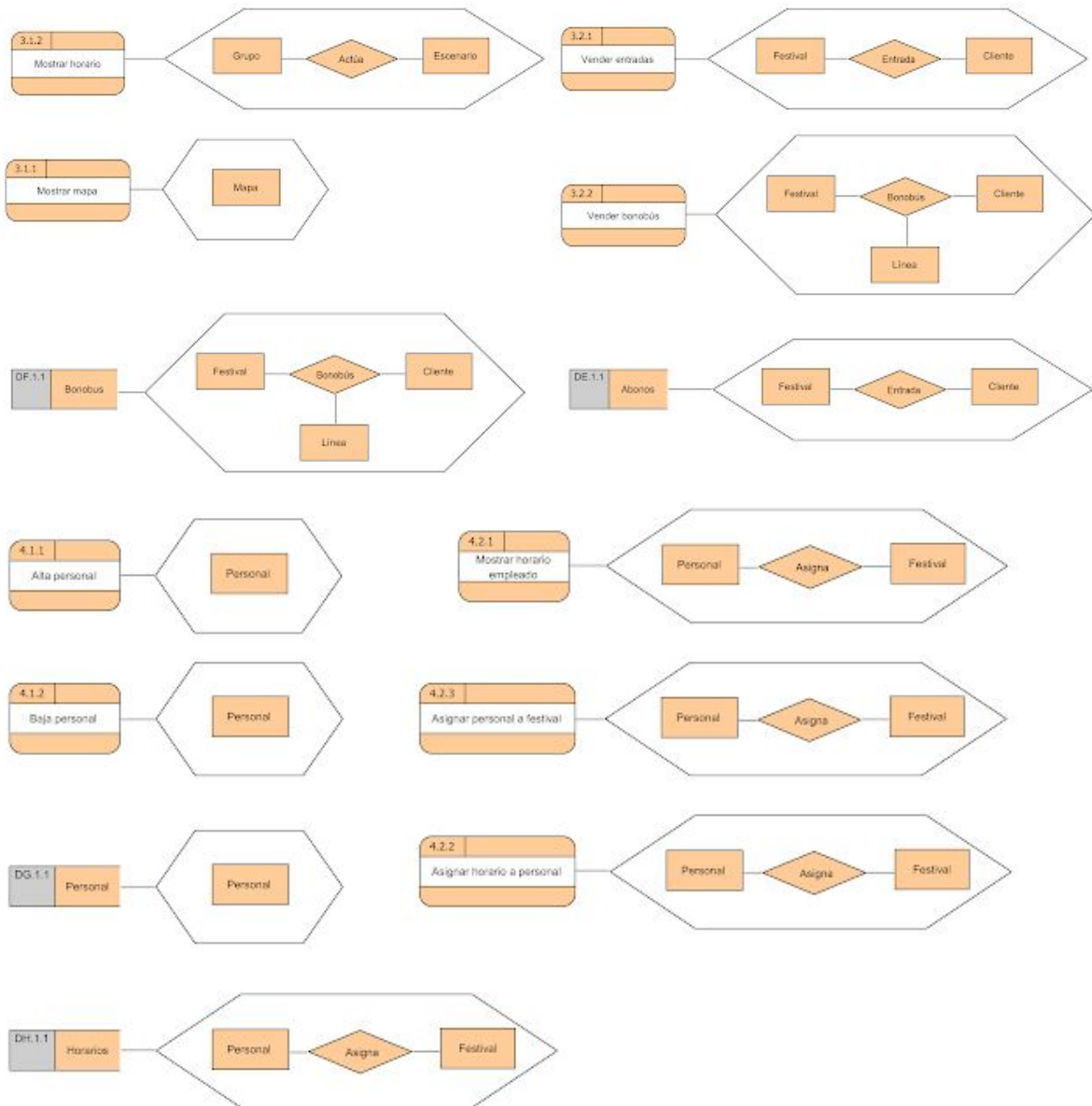


Esquema almacén D (ED2)



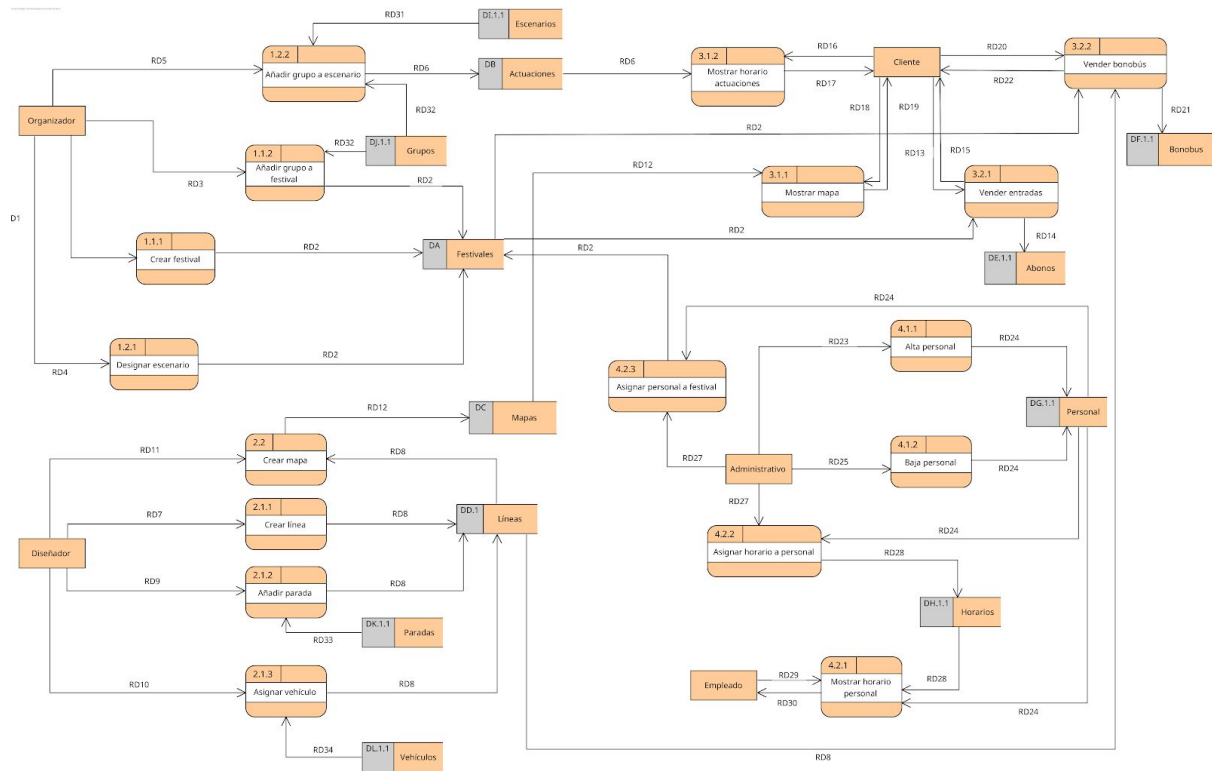
Esquema externo (EE2)



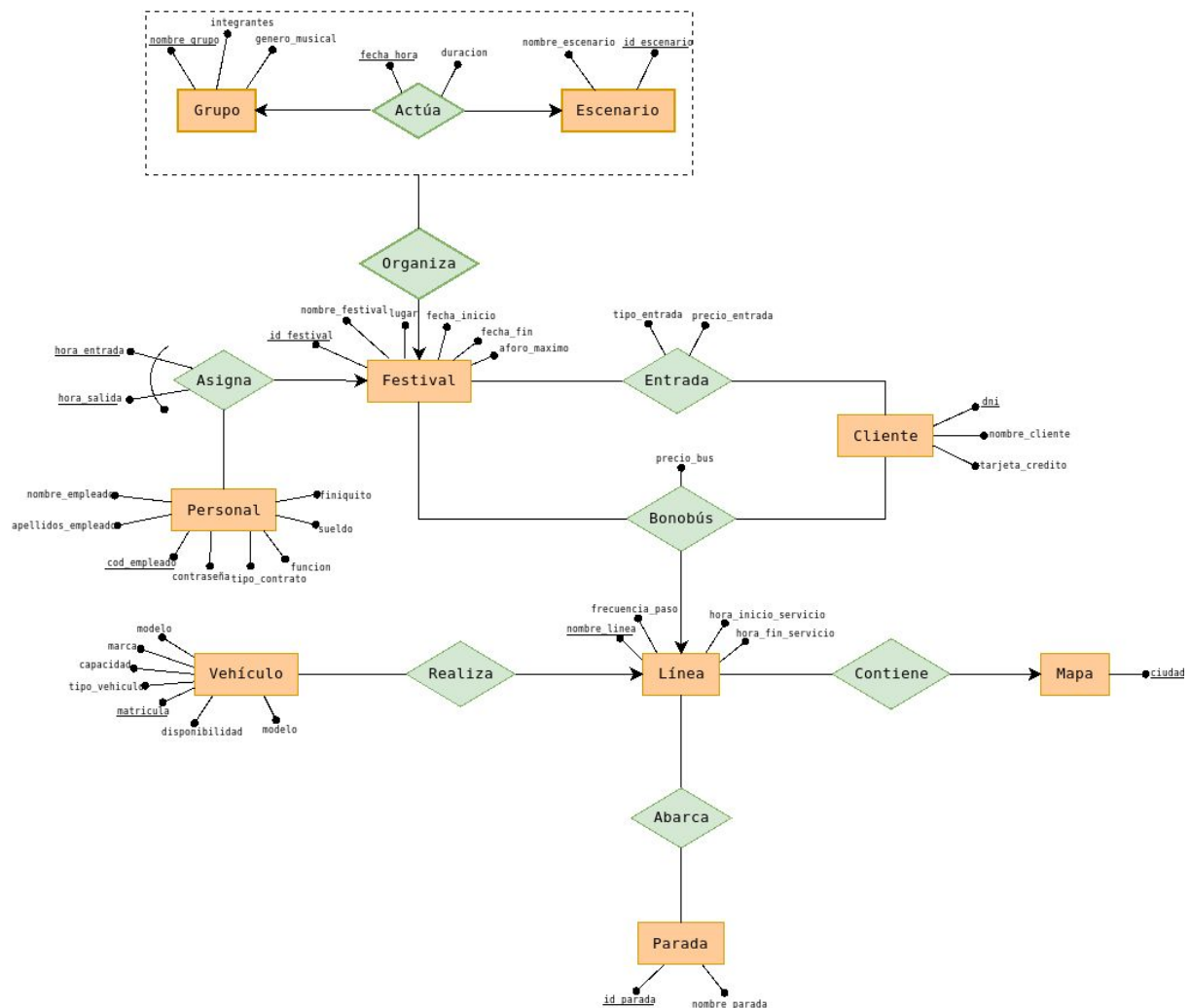


3.5. Esquema almacén F y E/R final

Esquema almacén F (DFD2)



E/R



3.6. Paso a tablas

- (1) Grupo (nombre_grupo, integrantes, genero_musical)
- (2) Escenario (id_escenario, nombre_escenario)
- (3) Actúa (nombre_grupo^(CE 1), id_escenario^(CE2), fecha_hora, duracion)
- (4) Festival (id_festival, nombre_festival, lugar, fecha_inicio, fecha_fin, aforo_maximo)
- (5) Organiza (nombre_grupo, fecha_hora^(CE3), id_festival^(CE4))
- (3-5) - (5') Actúa-Organiza(nombre_grupo^(CE 1), fecha_hora, id_escenario^(CE2), duracion, id_festival^(CE4))
- (6) Personal (cod_empleado, nombre_empleado, apellidos_empleado, contraseña, tipo_contrato, funcion, sueldo, finiquito)
- (7) Asigna(id_festival^(CE4), cod_empleado^(CE6), hora_entrada, hora_salida)
- (8) Cliente(dni, nombre_cliente, tarjeta_credito)
- (9) Entrada(id_festival^(CE4), dni^(CE8), tipo_entrada, precio_entrada)
- (10) Línea(nombre_linea, frecuencia_paso, hora_inicio_servicio, hora_fin_servicio)
- (11) Mapa(ciudad)
- (12) Contiene(nombre_linea^(CE 10), ciudad^(CE 16))
- (10-12)-(10') Línea-Contiene(nombre_linea, frecuencia_paso, hora_inicio_servicio, hora_fin_servicio, ciudad^(CE11))

- (13) Bonobus(id_festival^(CE4), dni^(CE8), nombre_linea^(CE10'), precio_bus)
- (14) Vehiculo(matricula, tipo_vehiculo, capacidad, marca, modelo, estado, disponibilidad)
- (15) Realiza(matricula^(CE14), nombre_linea^(CE10'))
- (14-15)-(14') Vehiculo-Realiza(matricula, tipo_vehiculo, capacidad, marca, modelo, estado, disponibilidad, nombre_linea^(CE10'))
- (16) (id_parada, nombre_parada)
- (17) Abarca(nombre_linea^(CE 10'), id_parada^(CE 16))

4. Implementación

4.1. Tablas

```
--1--
create table grupo(
    nombre_grupo varchar2(30) primary key,
    integrantes varchar2(150),
    genero_musical varchar2(15)
);

--2--
create table escenario(
    id_escenario varchar2(5) primary key,
    nombre_escenario varchar2(30)
);

--4--
create table festival(
    id_festival varchar2(5) primary key,
    nombre_festival varchar2(30),
    lugar varchar2(30),
    fecha_inicio date,
    fecha_fin date,
    aforo_maximo number(5)
);

-- 3-5 -> 3' --
create table actua_organiza(
    nombre_grupo references grupo(nombre_grupo),
    id_escenario references escenario(id_escenario),
    fecha_hora timestamp,
    duracion number,
    id_festival references festival(id_festival),
    primary key (nombre_grupo, fecha_hora)
);
```

```

--6--
create table personal(
    cod_empleado varchar2(7) primary key,
    nombre_empleado varchar2(20),
    apellidos_empleado varchar2(50),
    contraseña varchar2(20),
    tipo_contrato varchar2(10),
    funcion varchar2(15),
    sueldo number(6,2),
    finiquito number(5,2)
);

--7--
create table asigna(
    id_festival references festival(id_festival),
    cod_empleado references personal(cod_empleado),
    hora_entrada timestamp,
    hora_salida timestamp,
    primary key (cod_empleado,hora_entrada,hora_salida)
);

--8--
create table cliente(
    dni varchar2(9) primary key,
    nombre_cliente varchar2(20),
    tarjeta_credito number(16)
);

--9--
create table entrada(
    id_festival references festival(id_festival),
    dni references cliente(dni),
    tipo_entrada varchar2(7),
    precio_entrada number(4,2),
    primary key (id_festival,dni)
);

--11--
create table mapa(
    ciudad varchar2(30) primary key
);

-- 10-12 -> 10' --
create table linea_contiene(
    nombre_linea varchar2(10) primary key,
    frecuencia_paso number,
    hora_inicio_servicio timestamp,
    hora_fin_servicio timestamp,
    ciudad references mapa(ciudad)
);

```

--13--

```
create table bonobus(  
    id_festival references festival(id_festival),  
    dni references cliente(dni),  
    nombre_linea references linea_contiene(nombre_linea),  
    precio_bus number(4,2),  
    primary key (id_festival,dni)  
);
```

-- 14-15 -> 14' --

```
create table vehiculo_realiza(  
    matricula varchar2(7) primary key,  
    tipo_vehiculo varchar2(10),  
    capacidad number(3),  
    marca varchar2(20),  
    modelo varchar2(20),  
    estado varchar2(15),  
    disponibilidad varchar2(15),  
    nombre_linea references linea_contiene(nombre_linea)  
);
```

--16--

```
create table parada(  
    id_parada varchar2(5) primary key,  
    nombre_parada varchar2(30)  
);
```

--17--

```
create table abarca(  
    nombre_linea references linea_contiene(nombre_linea),  
    id_parada references parada(id_parada),  
    primary key (nombre_linea,id_parada)  
);
```

4.2. Triggers

1. Aforo completo

```
CREATE OR REPLACE TRIGGER aforo_completo
BEFORE
  INSERT
ON entrada
FOR EACH ROW
DECLARE
  num_entradas_festival number(4);
  aforo_max number(10);
BEGIN
  SELECT COUNT (*) INTO num_entradas_festival FROM entrada WHERE id_festival = :new.id_festival;
  SELECT AFORO_MAXIMO INTO aforo_max FROM FESTIVAL WHERE id_festival = :new.id_festival;

  IF num_entradas_festival >= aforo_max THEN
    RAISE_APPLICATION_ERROR(-20001, 'Aforo completo');
  END IF;
END;
```

Su función es impedir la venta de entradas una vez superado el número máximo de entradas posibles (aforo). Primero se cuenta el número de filas que tiene la tabla 'entrada' filtrando el 'id_festival' con el identificador del festival concreto, lo cual proporciona el número de entradas vendidas hasta el momento. Se obtiene el aforo máximo del festival y si es mayor o igual se lanza el error.

2. Vehículo no disponible

```
CREATE OR REPLACE TRIGGER vehiculo_no_disponible
BEFORE
  UPDATE OR INSERT OF NOMBRE_LINEA
ON vehiculo_realiza
FOR EACH ROW
BEGIN
  IF :new.disponibilidad != 'disponible' THEN
    RAISE_APPLICATION_ERROR(-20002, 'No se puede asignar vehiculo a linea porque no esta disponible');
  END IF;
END;
```

Impide asociar un vehículo a una línea si éste no se encuentra disponible. Para ello, comprueba si la nueva disponibilidad es distinta de 'disponible' y, si lo es, actúa el disparador.

3. Personal ya asignado

```
CREATE OR REPLACE TRIGGER personal_ya_asignado
BEFORE
INSERT
ON asigna
FOR EACH ROW
DECLARE
    CURSOR fecha_inicio1 IS SELECT festival.FECHA_INICIO FROM FESTIVAL, ASIGNA WHERE :new.cod_empleado = asigna.cod_empleado AND festival.
        id_festival = asigna.id_festival;
    CURSOR fecha_fin1 IS SELECT festival.FECHA_FIN FROM FESTIVAL, ASIGNA WHERE :new.cod_empleado = asigna.cod_empleado AND festival.
        id_festival = asigna.id_festival;
    fecha_i DATE;
    fecha_f DATE;
    fecha_i2 DATE;
    fecha_f2 DATE;
    personal_en_festival NUMBER(4);
BEGIN
    SELECT COUNT (*) INTO personal_en_festival FROM asigna WHERE :new.cod_empleado = cod_empleado;

    IF personal_en_festival > 0 THEN
        OPEN fecha_inicio1;
        OPEN fecha_fin1;

        SELECT FECHA_INICIO INTO fecha_i2 FROM FESTIVAL WHERE :new.id_festival = id_festival;
        SELECT FECHA_FIN INTO fecha_f2 FROM FESTIVAL WHERE :new.id_festival = id_festival;

        LOOP
            FETCH fecha_inicio1 INTO fecha_i;
            FETCH fecha_fin1 INTO fecha_f;
            EXIT WHEN fecha_inicio1%NOTFOUND AND fecha_fin1%NOTFOUND;
            IF(((fecha_i2 <= fecha_i) AND ((fecha_f2 >= fecha_i) OR (fecha_f2 >= fecha_f)) OR ((fecha_i2 >= fecha_i) AND (fecha_i2 <= fecha_f))))
            THEN
                RAISE_APPLICATION_ERROR(-20003,'El empleado ya esta asignado en un festival durante estas fechas');
            END IF;
        END LOOP;
        CLOSE fecha_inicio1;
        CLOSE fecha_fin1;
    END IF;
END;
```

Imposibilita asignar un empleado a un festival si ya está asociado a otro y coinciden las fechas entre ellos. Primero se comprueba si ya tiene algún festival asignado para poder comparar. Se declaran dos cursores y en ellos se almacenan la fecha de inicio y fin de los festivales que ya están asignados al empleado (haciendo coincidir los códigos de empleado y de festival). Por cada festival que detecte se compara que no coinciden las fechas con las del nuevo festival.

4. Escenario ocupado

```
CREATE OR REPLACE TRIGGER escenario_ocupado
BEFORE
INSERT
ON actua_organiza
FOR EACH ROW
DECLARE
    CURSOR inicio_1 IS SELECT fecha_hora, duracion from actua_organiza where :new.id_festival = id_festival AND :new.id_escenario=
        id_escenario;
    hora_inicio_1 TIMESTAMP;
    hora_fin_1 TIMESTAMP;
    hora_inicio_2 TIMESTAMP;
    hora_fin_2 TIMESTAMP;
    duracion_1 number;
    num_actuaciones number;
BEGIN
    SELECT COUNT(*) INTO num_actuaciones FROM actua_organiza where :new.id_festival = id_festival AND :new.id_escenario= id_escenario;

    IF num_actuaciones > 0 THEN
        OPEN inicio_1;
        hora_fin_2 := :new.fecha_hora + NUMTODSINTERVAL( :new.duracion, 'MINUTE' );
        LOOP
            FETCH inicio_1 INTO hora_inicio_1, duracion_1;
            EXIT WHEN inicio_1%NOTFOUND;
            hora_fin_1 := hora_inicio_1 + NUMTODSINTERVAL( duracion_1, 'MINUTE' );
            IF(((new.fecha_hora <= hora_inicio_1) AND ((hora_fin_2 >= hora_inicio_1) OR (hora_fin_2 >= hora_fin_1)) OR ((:new.fecha_hora >=
                hora_inicio_1) AND (:new.fecha_hora <= hora_fin_1)))) THEN
                RAISE_APPLICATION_ERROR(-20004,'Escenario ya asignado en ese intervalo de tiempo');
            END IF;
        END LOOP;
        CLOSE inicio_1;
    END IF;
END;
```

Este disparador actúa al insertar una actuación en un mismo escenario si coincide en el intervalo de tiempo de otra actuación.

Para empezar, se comprueba si ya hay actuaciones en el escenario de un festival concreto; si las hay, a continuación, se recorren con un cursor del cual obtenemos su hora de inicio y su duración, con los que calculamos también su hora final. Por último, con la hora de inicio y la de fin se comprueba si los intervalos coinciden.