

Índice de contenido

<i>Descripción en lenguaje natural.....</i>	<i>2</i>
<i>Requisitos de datos.....</i>	<i>3</i>
<i>Requisitos funcionales.....</i>	<i>6</i>
<i>Diagramas de Flujo de Datos (DFDs).....</i>	<i>10</i>
<i>Esquema D (conceptual).....</i>	<i>13</i>
<i>Esquemas externos por área funcional.....</i>	<i>14</i>
<i>Operaciones de datos (por área funcional).....</i>	<i>21</i>
<i>Esquemas de operación (por área funcional).....</i>	<i>24</i>
<i>Esquemas de navegación (por área funcional).....</i>	<i>28</i>
<i>Paso a tablas.....</i>	<i>32</i>
<i>Implementación.....</i>	<i>33</i>
<i>Procedimiento cálculo del balance.....</i>	<i>43</i>
<i>Trigger (Disparador por INSERT).....</i>	<i>46</i>

Sistema de Información para Gestión de un Partido Político

Descripción en lenguaje natural

El sistema de información se divide en cuatro grandes áreas: financiera, agenda, inmuebles y recursos humanos:

Área financiera

El área financiera se va a encargar de gestionar ingresos, gastos y nóminas (por separado). También debe ser capaz de calcular balances, listar las deudas (pagos que aún no se han realizado) y créditos (ingresos que aún no se han efectuado).

Área de agenda

Este área es necesaria ya que los miembros del partido deben gestionar y notificar adecuadamente un gran número de eventos.

- Gestionar implica:
 - creación,
 - modificación
 - y eliminación de citas y eventos.
- Notificar implica
 - que los usuarios implicados en una cita reciban notificaciones adecuadas,
 - poder consultar cierta información sobre la agenda de otro usuario en función de los permisos que disponga.

El sistema contempla los **eventos** como un suceso delimitado en el tiempo, y las **citas** como el compromiso de un usuario a acudir a un evento con un determinado rol. Además, existen **usuarios**, **grupos de usuarios** y **notificaciones**.

Un **evento** está descrito por:

- Nombre.
- Lugar: puede ser un edificio en propiedad, una calle, un enlace a Google Maps, unas coordenadas GPS...
- Fecha: hora, día, mes, año, y duración.
- Usuarios implicados: una lista de usuarios del sistema de información o de personas ajenas a él.

Una **cita** está descrita por:

- Usuario que tiene la cita.
- Evento al que se acude.
- Privacidad: público, privado, o reservado para ciertos grupo.
- Estado durante la cita: disponible, ocupado, o disponibilidad para ciertos grupos.
- Nivel de prioridad: Opcional, normal, importante.
- Rol desempeñado: espectador, exponente, comitiva...
- Notificación: Recordatorio programado.

Un **usuario** consta de:

- Identificación (nombre, apellidos, lugar de nacimiento, DNI...).
- Información de contacto, ya sea fuera de la plataforma (correo electrónico, teléfono...) o dentro (identificador).
- Lista de citas que tiene.
- Grupos de usuarios a los que pertenece.

Un **grupo de usuarios** consta de:

- Lista de usuarios.
- Grupo común (cualquier usuario puede hacer referencia a él) o personalizado (grupos creados por usuarios para gestionar sus propias citas).

Una **notificación** está descrita por:

- Usuario destinatario.
- Fecha de la notificación.
- Descripción de la notificación.
- Prioridad.

Área de activos

Se consideran activos aquellos bienes propiedad del partido, entre los que consideraremos inmuebles, mobiliario y material de oficina.

El objetivo del área de activos es el de gestionar el inventario, con las adquisiciones y salidas de bienes:

- Inventario de activos.
- Adquisición de activos:
 - Compra de activos.
 - Donación de activos.
 - Traspaso de mobiliario y materiales de oficina (entre los distintos inmuebles del partido).
- Salida de activos:
 - Venta de activos.
 - Donación de activos.
 - Traspaso de activos (a otras organizaciones).

Área de recursos humanos

El área de Recursos Humanos tiene como misión principal el diseño y aplicación de una política de personal que mezcle los intereses de los empleados con los propios del partido, con el fin de conseguir los objetivos de la corporación.

Las funciones del sistema de información del departamento de Recursos Humanos son las siguientes:

- Reclutamiento, selección y acogida de los empleados.
- Gestión de incidencias de tipo laboral (ceses, situaciones administrativas, permisos, licencias, etc.).
- Análisis, descripción e inventario de plazas y puestos de trabajo.
- Control horario, de presencia del cumplimiento de la normativa de incompatibilidades.

Los empleados se dividen en políticos (miembros del partido con funciones políticas) y administrativos (miembros del partido con funciones administrativas).

Requisitos de datos

RD1: Datos de los gastos realizados. Se trata de registrar todos los gastos que se producen o se van a producir en el futuro y su concepto. Cada gasto debe tener como mínimo la siguiente información:

- Identificador del gasto
- Fecha del gasto (cuando se tiene constancia de él, no cuando se realiza efectivamente)
- Concepto del gasto
- Cantidad gasto
- ¿Se ha efectuado?

RD2: Datos de los ingresos recibidos: Se llevará el control de todos los ingresos que se hayan recibido o todavía estén por recibir. Cada ingreso debe contener, al menos, la siguiente información:

- Identificador del ingreso
- Fecha del ingreso (cuando se tiene constancia de él, no cuando se realiza efectivamente)
- Tipo de ingreso
- Concepto del ingreso
- Cantidad del ingreso
- ¿Se ha efectuado?

RD3: Balance. Positivo: Beneficios, Negativo: Pérdidas.

RD4: Nóminas. Se trata de registrar la nómina de un determinado empleado. Necesitamos:

- Tipo de empleado,
- DNI del empleado
- Sueldo empleado

RD5: Los datos de identificación de un empleado:

- Nombre (una cadena de hasta 20 caracteres no vacía).
- Apellidos (una cadena de hasta 50 caracteres no vacía).
- Nacionalidad (una cadena de hasta 20 caracteres no vacía).
- Lugar de nacimiento (una cadena de hasta 20 caracteres no vacía).
- Fecha de nacimiento (un dato de tipo fecha).
- DNI (una cadena de 8 números seguida de una letra mayúscula de verificación).

RD6: Los datos de contacto de un empleado:

- Domicilio (una cadena de hasta 50 caracteres no vacía).
- Correo electrónico (una cadena de hasta 50 caracteres con una dirección de correo válida).
- Teléfono (una cadena con un carácter '+' inicial opcional seguido de hasta 20 caracteres numéricos, con un teléfono válido).
- Identificador dentro de la plataforma (una cadena con el símbolo '@' seguida de hasta 20 caracteres).

RD7: Los datos de un empleado:

- Identificación.
- Contacto.
- Tipo (AD: administrativo / PO: político).

RD8: Los datos de un evento:

- Nombre (una cadena de hasta 20 caracteres no vacía).
- Lugar (una cadena de hasta 100 caracteres).
- Fecha (un dato de tipo fecha)
- Duración (un dato que puede ser de tipo fecha o de caracteres numéricos acabado con una cadena tipo 'horas', 'minutos'...).
- Empleados implicados (una lista de usuarios del sistema de información).

RD9: Los datos de una cita:

- Empleado.
- Evento.
- Notificación.
- Privacidad (un conjunto de datos booleanos que indican la visibilidad de la cita para diferentes grupos).
- Estado durante la cita (un conjunto de datos booleanos que indican disponibilidad para diferentes grupos).
- Nivel de prioridad (un dato enumerado con valores opcional, normal, importante).
- Rol desempeñado (una cadena de hasta 100 caracteres).

RD10: Los datos de un grupo de empleados:

- Usuarios.
- Usuario creador (clave de un usuario; puede ser nula, en cuyo caso el grupo es común).

RD11: Los datos de una notificación.

- Empleado destinatario.
- Fecha de la notificación (un dato de tipo fecha).
- Descripción de la notificación (una cadena de hasta 100 caracteres).
- Prioridad (un dato numérico).

RD12: Puesto ofertado:

- Nombre que lo identifica (cadena no vacía).
- Tipo (cadena no vacía).
- Requerimientos (cadena no vacía).
- Salario (cadena numérica).

RD13: Horarios:

- Hora de entrada.
- Hora de salida.
- IDEmp

RD14: Datos de una incidencia de trabajo:

- Descripción de la incidencia en cuestión (cadena no vacía).
- Tipo (cadena no vacía).
- Empleado asociado a ella (cadena no vacía).

RD15: Datos del inmueble:

- Dirección (País, Provincia, Ciudad, Calle...).
- Nombre.
- Código.
- Precio.

RD16: Datos del mobiliario:

- Tipo.
- Nombre.
- Código.
- Precio.
- Cantidad.
- Peso.

RD17: Datos del material de oficina:

- Tipo.
- Nombre.
- Código.
- Precio.
- Cantidad.
- Peso.

Requisitos funcionales

Área financiera

RF1: Dar de alta un ingreso: se almacenará un ingreso proporcionando los datos referentes a:

- RD2

RF2: Dar de alta un gasto: se almacenará un gasto proporcionando los datos referentes a:

- RD1

RF3: Buscar un ingreso: Se buscan ingresos por fecha o identificador. Para ello hay que proporcionar los datos fecha o identificador que contiene

- RD2

RF4: Buscar un gasto: Se buscan los gastos por fecha o identificador. Para ello hay que proporcionar los datos fecha o identificador que contiene

- RD1

RF5: Borrar un ingreso: Se borrará un ingreso proporcionando el identificador (ya que si se proporciona una fecha podrían borrarse varios). Para ello hay que proporcionar el dato identificador de

- RD2

que puede ser obtenido mediante el uso de RF3. Este requisito debería estar bloqueado solo para un administrador. No estoy seguro si se deberían borrar los ingresos.

RF6: Borrar un gasto: Se borrará un gasto proporcionando el identificador (ya que si se proporciona una fecha podrían borrarse varios). Para ello hay que proporcionar el dato identificador de

- RD1

que puede ser obtenido mediante el uso de RF4. Este requisito debería estar bloqueado solo para un administrador. No estoy seguro si se deberían borrar los gastos.

RF7: Modificar un ingreso: Se trata de cambiar los datos de un determinado ingreso, identificado por el identificador contenido en:

- RD2

que se puede obtener mediante el uso de RF3. Al igual que RF5, debería estar bloqueado o reservado a administradores.

RF8: Modificar un gasto: Se trata de cambiar los datos de un determinado gasto, identificado por el identificador contenido en:

- RD1

que se puede obtener mediante el uso de RF4. Al igual que RF6, debería estar bloqueado o reservado a administradores.

RF9: Balance de ingresos/gastos: Se trata de obtener el balance instantáneo de dinero virtual que se tiene, es decir, ingresos - gastos. Computan:

- RD1
- RD2

hayan sido o no efectivos.

RF10: Líquido disponible: Se trata de conocer el *cash-flow* actual, es decir, serán computables:

- RD1
- RD2

sólo aquéllos que ya hayan sido efectuados.

RF11: Líquido no disponible: Se trata de saber la cantidad de dinero que debemos o se nos adeuda. Para ello será necesario:

- RD1
- RD2

Solo computan los que hayan sido efectuados.

RF12: Deudas: Se trata de saber qué deudas tenemos:

- RD1
- RD2

Sólo computan los que hayan sido efectuados

RF13: Créditos: Se trata de saber qué y cuánto dinero se nos debe:

- RD1
- RD2

Sólo computan los que hayan sido efectuados

RF14: Solicitar balance.

RF15: Calcular balance.

RF16: Devolver balance. Se usa RD3

Área de agenda

RF17: Crear un evento: Esta función registra un evento dentro del sistema. Requiere de los datos proporcionados por el usuario, entre otros los usuarios implicados (RD5) y opcionalmente de notificaciones resultantes (RD9). Produce un nuevo evento (RD6) dentro del sistema, y opcionalmente notificaciones (RD9). El sistema comunica la correcta creación del evento.

RF18: Modificar un evento: Esta función modifica un evento dentro del sistema. Requiere de un evento (RD6) determinado, y de los datos de actualización (entre otros los usuarios implicados (RD5) y las notificaciones (RD9)), ambos proporcionados por el usuario. El sistema comunica la correcta modificación del evento.

RF19: Eliminar un evento: Esta función elimina un evento existente dentro del sistema. Requiere de un evento (RD6) determinado por el usuario. Elimina dicho evento del sistema, así como las notificaciones (RD9) asociadas a dicho evento. El sistema comunica la correcta eliminación del mismo.

RF20: Crear una cita: Esta función registra una cita dentro del sistema. Requiere de un evento (RD6) proporcionado por el usuario (puede ser creado en el momento o puede referirse a un evento existente) así como información sobre el usuario que la crea (RD5), los grupos de usuarios con permiso para verla (RD8), y opcionalmente las notificaciones asociadas (RD9), entre otros datos. Produce una nueva cita (RD7) dentro del sistema, y opcionalmente notificaciones (RD9). El sistema comunica la correcta creación de la cita.

RF21: Modificar una cita: Esta función modifica una cita dentro del sistema. Requiere de una cita (RD7) determinada, y de los datos de actualización (como información sobre el usuario que la crea (RD5), los grupos de usuarios con permiso para verla (RD8), o las notificaciones asociadas (RD9), entre otros), ambos proporcionados por el usuario. El sistema comunica la correcta modificación de la cita.

RF22: Eliminar una cita: Esta función elimina una cita existente dentro del sistema. Requiere de una cita (RD7) determinada por el usuario. Elimina dicha cita del sistema. El sistema comunica la correcta eliminación del mismo.

RF23: Notificar citas: El sistema envía notificaciones (por correo electrónico, en la plataforma, por teléfono...) a los usuarios cuando queda poco tiempo para que ocurran sus citas. Requiere de las notificaciones del sistema (RD9). Cuando es el momento indicado por la notificación, envía el mensaje de la notificación al usuario (RD5) destinatario indicado en la notificación.

RF24: Consultar citas de otros usuarios: Permite a un usuario consultar las citas de otros usuarios, si tiene los permisos adecuados. Requiere los usuarios implicados (RD5), citas del usuario consultado (RD7) y grupos a los que pertenece el usuario consultor (RD8). No se producen cambios en el sistema. Si el usuario consultor tiene los permisos adecuados, el sistema muestra las citas del usuario

consultado. Si no tiene los permisos, sólo muestra cuándo está ocupado el usuario consultado.

Área de activos

RF25: Dar de alta un activo: Se da de alta un activo en el inventario. El usuario deberá introducir datos relevantes con RD15, RD16 o RD17.

RF26: Dar de baja un activo: Se da de baja un activo en el inventario. El usuario deberá introducir datos relevantes con RD15, RD16 o RD17.

RF27: Modificar un activo: Se modifica el activo en el inventario. El usuario deberá introducir datos relevantes con RD15, RD16 o RD17.

RF28: Buscar un activo: Función utilizada para buscar un activo en el inventario. Es necesario haber dado antes de alta el activo. El usuario deberá introducir datos relevantes con RD15, RD16 o RD17.

Área de recursos humanos

RF29: Reclutamiento de los empleados. Necesario RD7.

RF30: Gestión de incidencias. Necesario RD14.

RF31: Dar de alta a un empleado en el departamento. Necesario RD7.

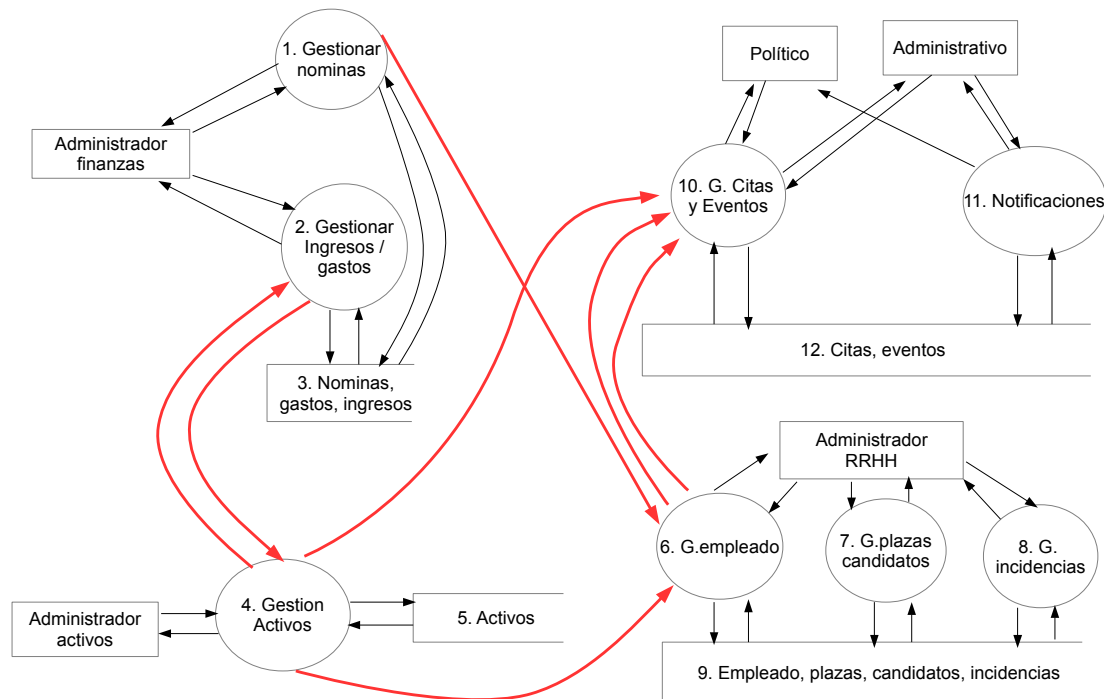
RF32: Dar de baja un empleado en el departamento. Necesario RD7.

RF33: Modificar un empleado. Necesario RD7.

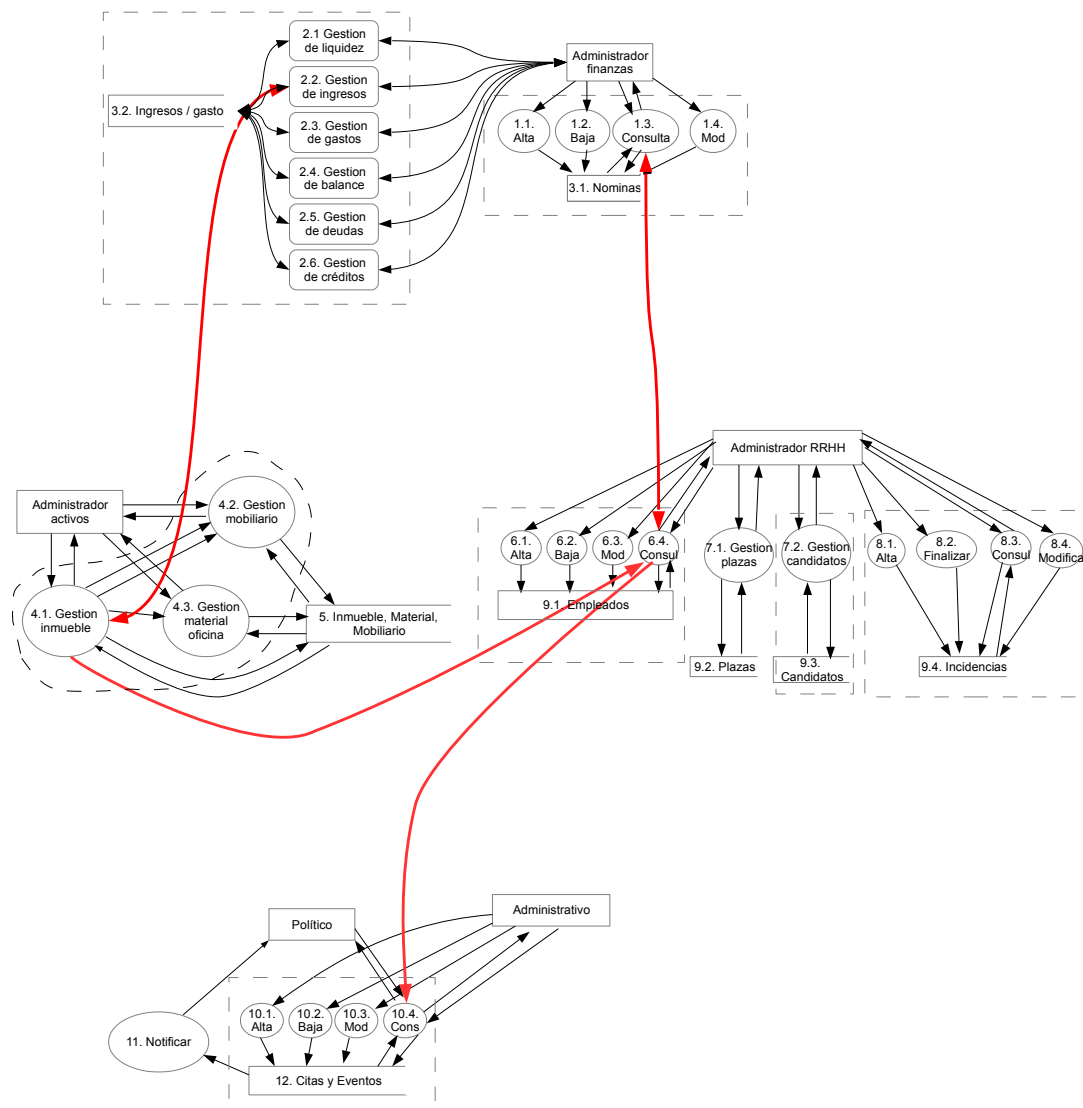
RF34: Gestión de plazas ofertadas. Necesario RD12.

Diagramas de Flujo de Datos (DFDs)

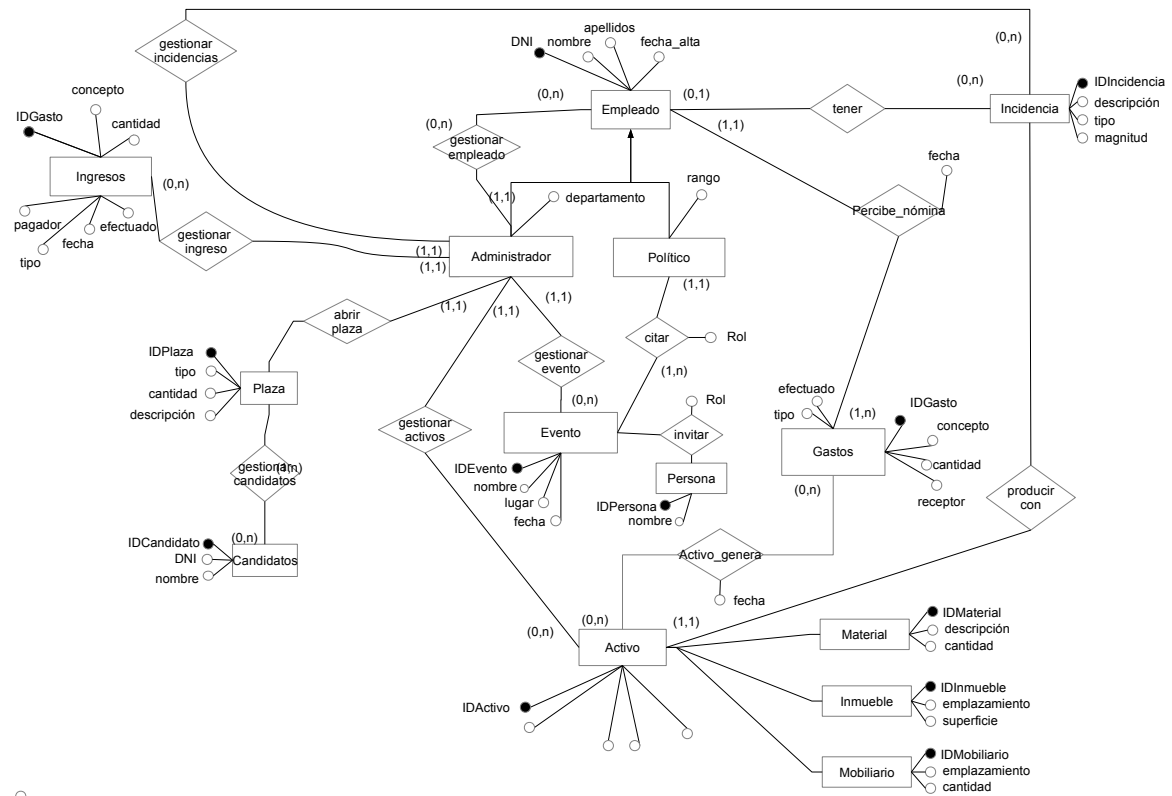
DFD armazón



DFD de nivel 1

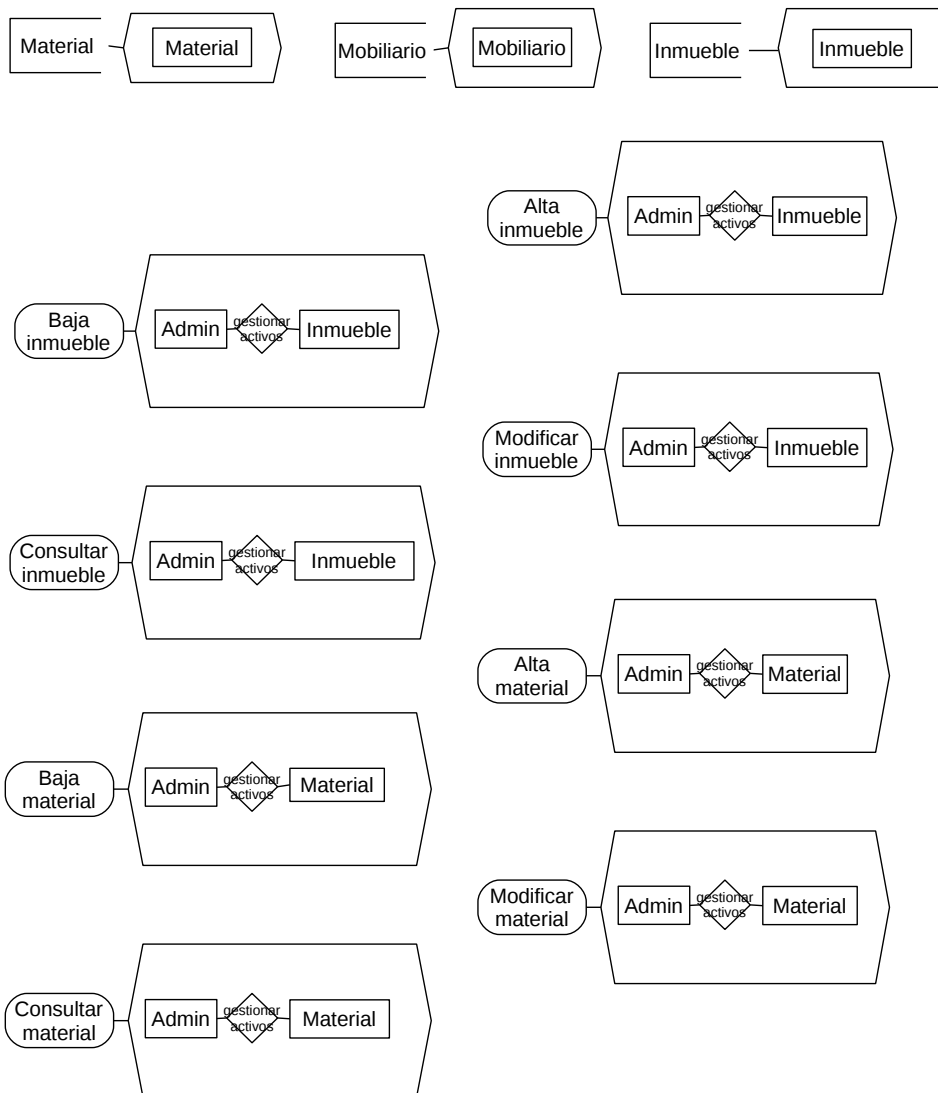


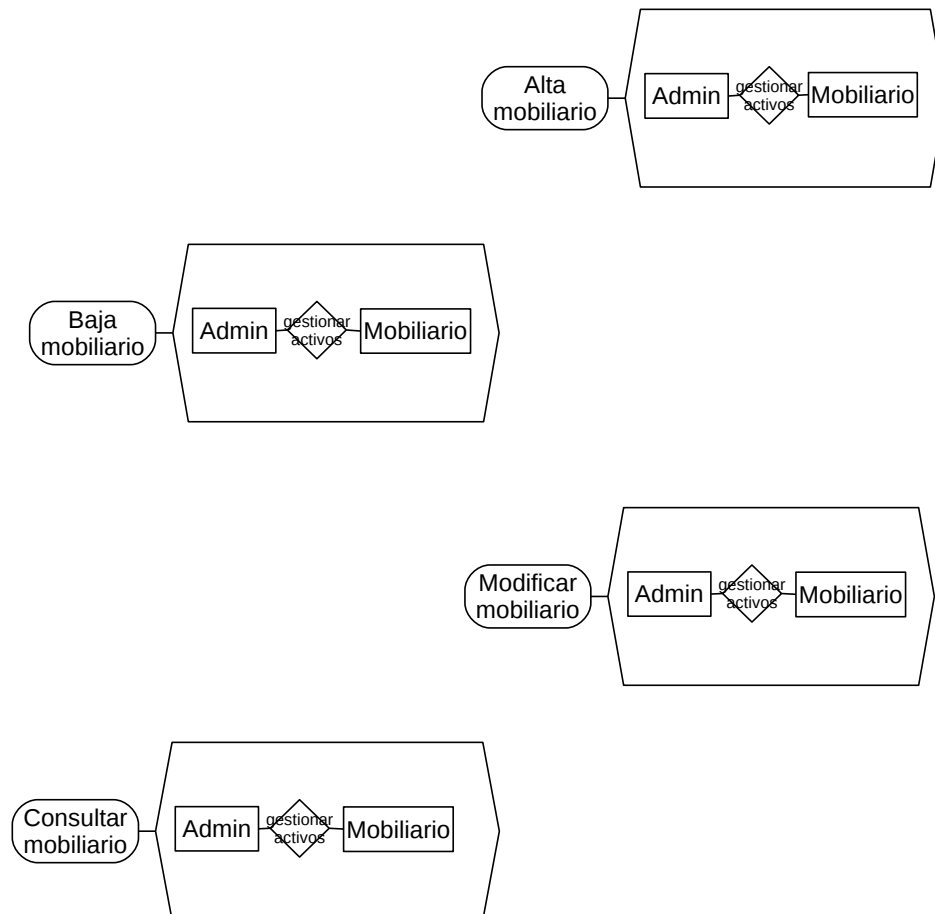
Esquema D (conceptual)



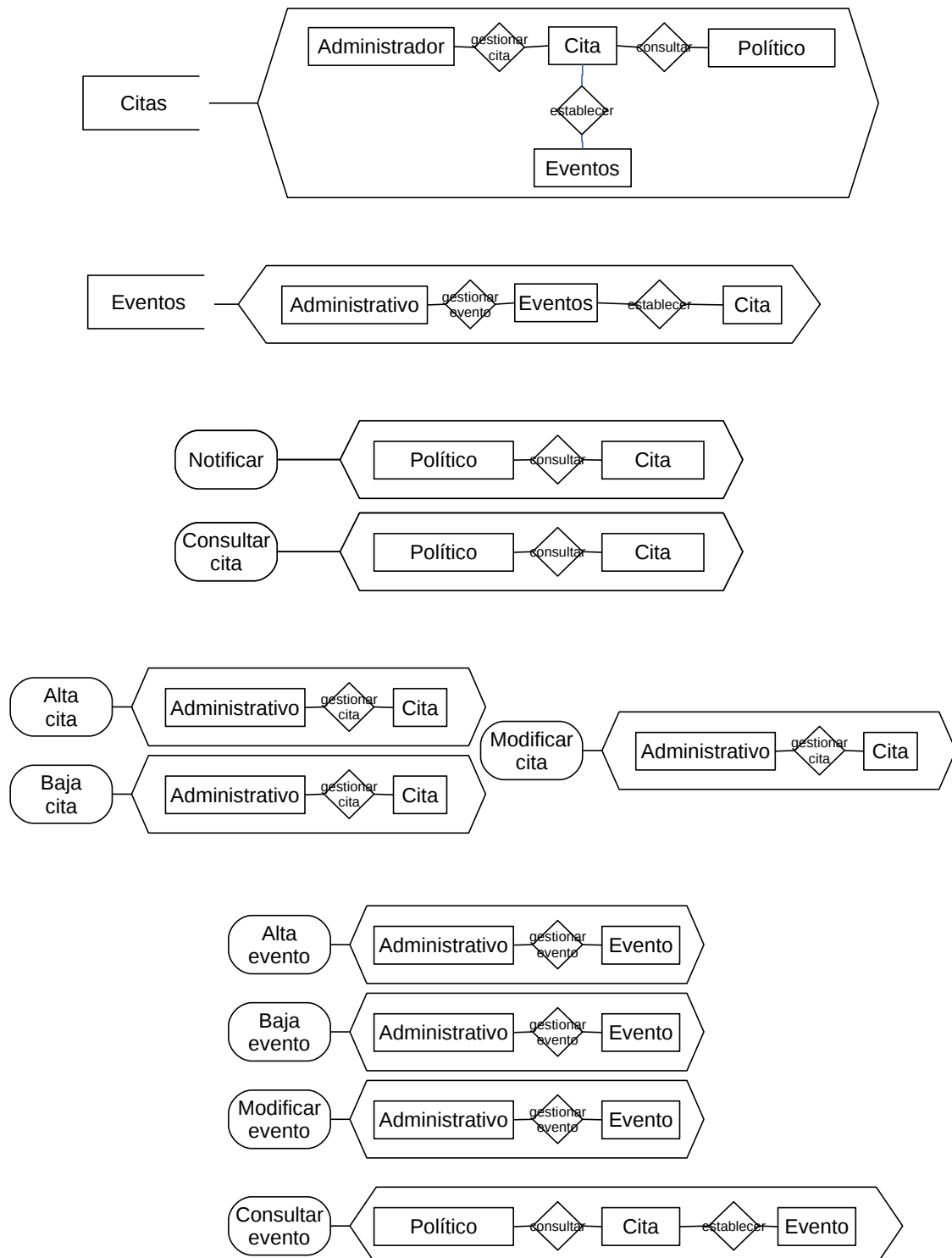
Esquemas externos por área funcional

Área de activos

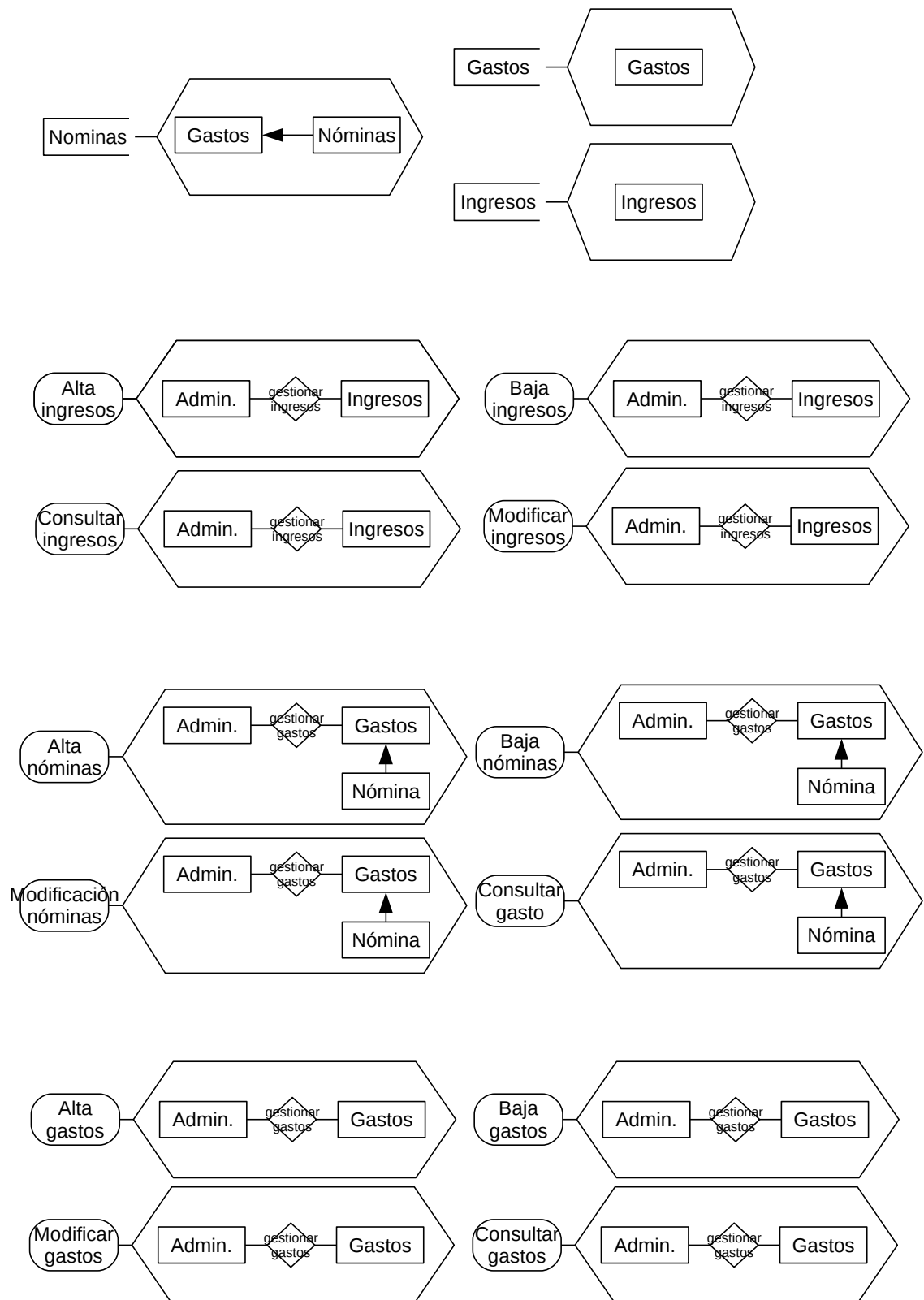


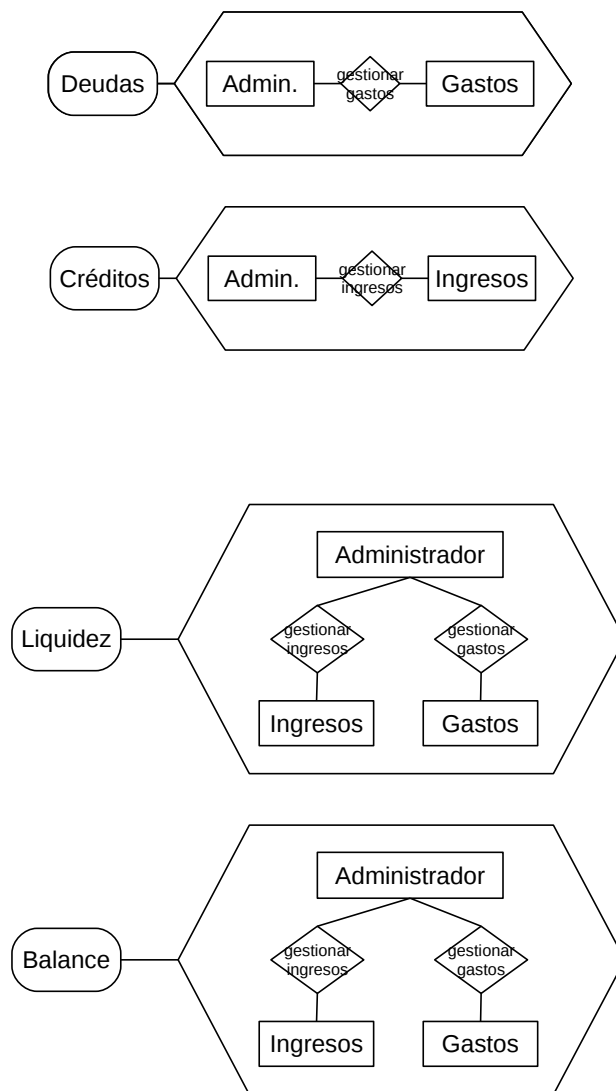


Área de agenda

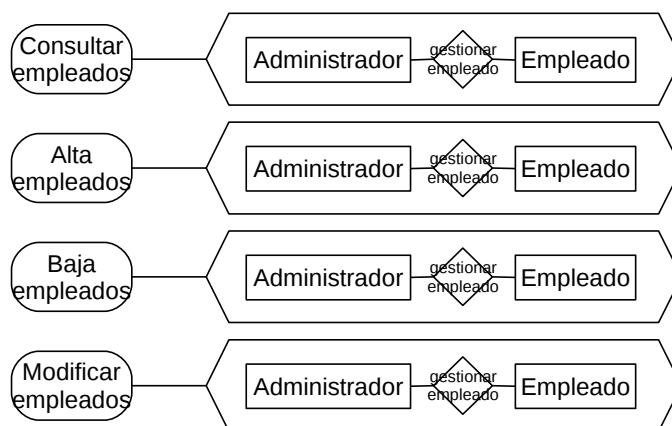
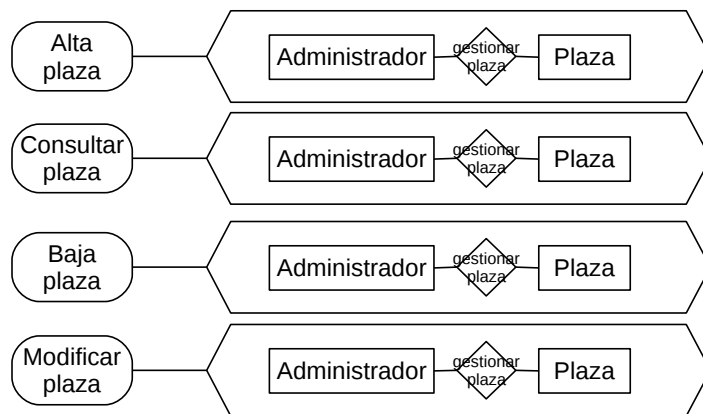
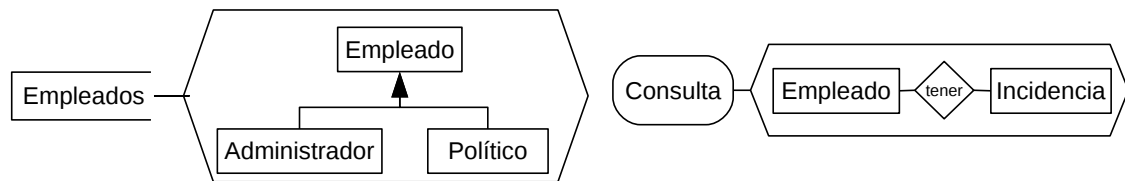


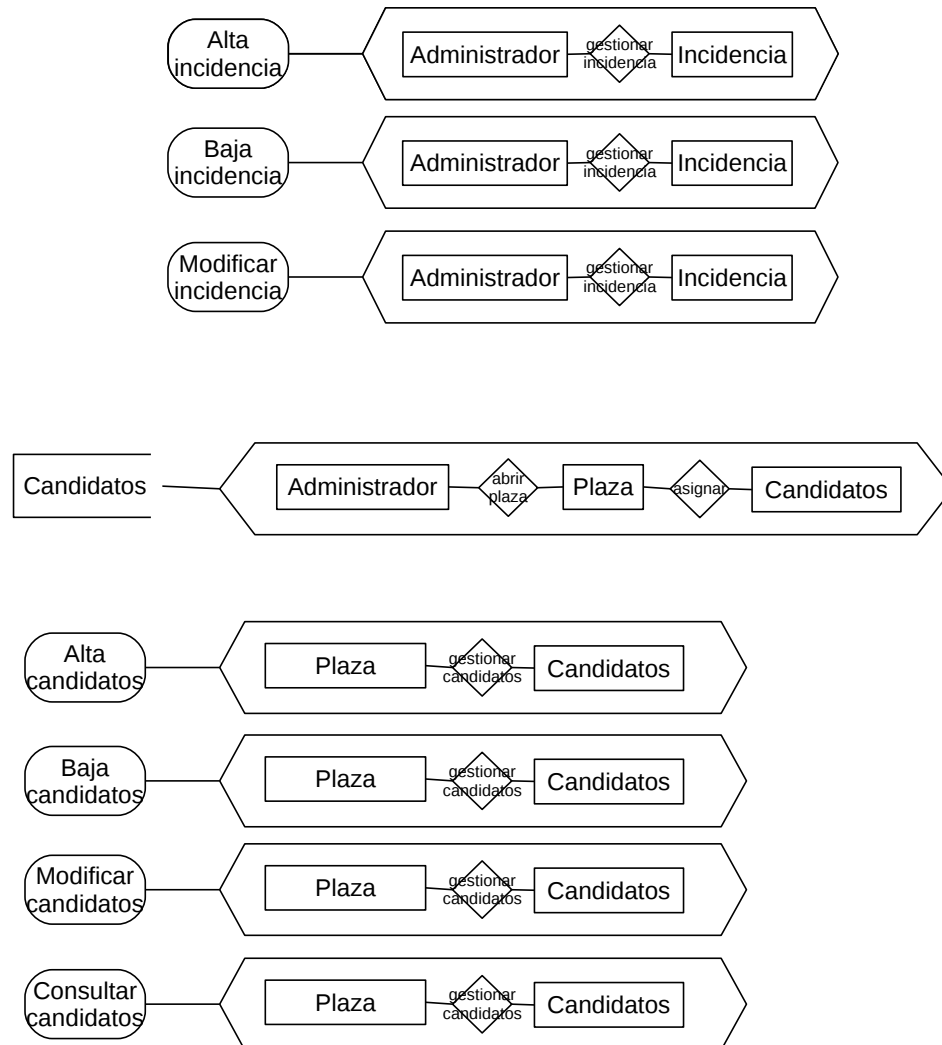
Área financiera





Área de recursos humanos





Operaciones de datos (por área funcional)

Área financiera

- Op.1. Dar de alta un ingreso: IDIngreso, concepto, cantidad, pagador, tipo, fecha, efectuado.
- Op.2. Consultar un ingreso por IDIngreso, pagador o fecha.
- Op.3. Modificar un ingreso identificado por su IDIngreso.
- Op.4. Borrar un ingreso identificado por su IDIngreso.
- Op.5. Dar de alta un gasto: IDGasto, concepto, cantidad, receptor, tipo, fecha, efectuado.
- Op.6. Consultar un gasto por IDGasto, receptor o fecha.
- Op.7. Modificar un gasto identificado por su IDGasto.
- Op.8. Borrar un gasto identificado por su IDGasto.
- Op.9. Dar de alta una nómina: DNI, IDGasto, mes, año.
- Op.10. Consultar una nómina identificada por DNI.
- Op.11. Modificar una nómina identificada por DNI.
- Op.12. Borrar una nómina identificada por DNI.
- Op.13. Calcular Balance.
- Op.14. Consultar Balance.
- Op.15. Calcular Líquido disponible.
- Op.16. Consultar Líquido disponible.
- Op.17. Calcular Deudas.
- Op.18. Consultar Deudas.
- Op.19. Calcular Créditos (lo que nos deben).
- Op.20. Consultar Créditos.

Área de activos

- Op.21. Dar de alta un activo: IDActivo.
- Op.22. Dar de alta un material: IDMaterial, descripción, cantidad.
- Op.23. Consultar un material identificado por IDMaterial.
- Op.24. Modificar un material identificado por IDMaterial.

- Op.25. Borrar un material identificado por IDMaterial.
- Op.26. Dar de alta un Inmueble: IDInmueble, emplazamiento, superficie.
- Op.27. Consultar un Inmueble identificado por IDInmueble.
- Op.28. Modificar un Inmueble identificado por IDInmueble.
- Op.29. Borrar un Inmueble identificado por IDInmueble.
- Op.30. Dar de alta Mobiliario: IDMobiliario, emplazamiento, cantidad.
- Op.31. Consultar Mobiliario identificado por IDMobiliario.
- Op.32. Modificar Mobiliario identificado por IDMobiliario.
- Op.33. Borrar Mobiliario identificado por IDMobiliario.

Área de agenda

- Op.34. Consultar citas por DNI de Político.
- Op.35. Consultar un evento por IDEvento.
- Op.36. Dar de alta una cita: DNI de Político, IDEvento, rol.
- Op.37. Modificar una cita identificada por DNI de Político e IDEvento.
- Op.38. Borrar una cita identificada por DNI de Político e IDEvento.
- Op.39. Borrar una invitación por IDPersona e IDEvento.
- Op.40. Dar de alta un evento: IDEvento, nombre, lugar, fecha.
- Op.41. Modificar un evento identificado por IDEvento.
- Op.42. Borrar un evento identificado por IDEvento.

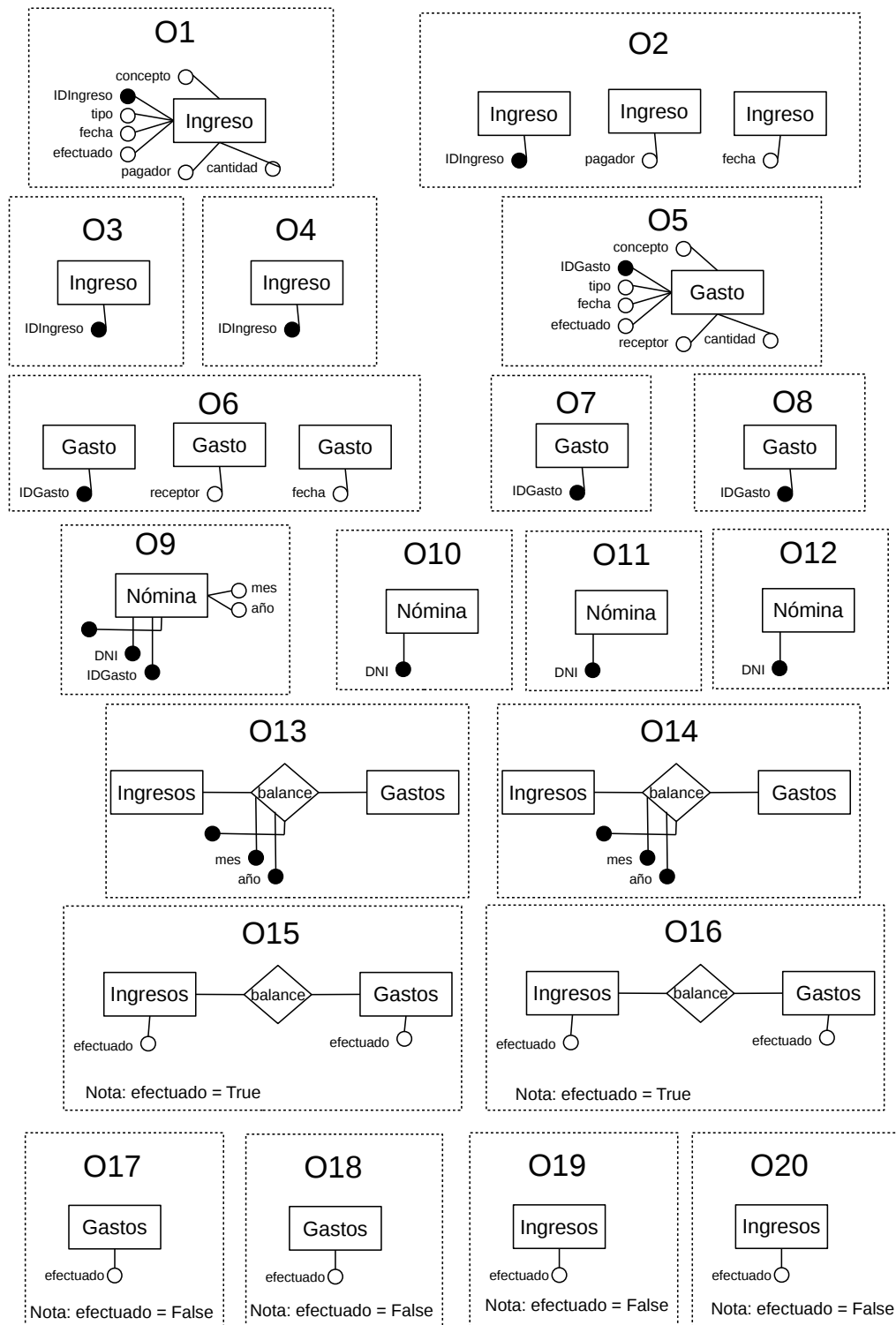
Área de recursos humanos

- Op.43. Dar de alta una plaza: IDPlaza, tipo, descripción, cantidad
- Op.44. Consultar una plaza identificado por IDPlaza, tipo, descripción.
- Op.45. Dar de baja una plaza identificado por IDPlaza
- Op.46. Modificar una plaza identificado por IDPlaza
- Op.47. Consultar empleado identificado por IDEmpleado
- Op.48. Dar de alta empleado: IDEmpleado, DNI, departamento, nombre.
- Op.49. Dar de baja empleado identificado por IDEmpleado
- Op.50. Modificar empleado identificado por IDEmpleado
- Op.51. Dar de alta incidencia: IDIncidencia, tipo, descripción, magnitud.

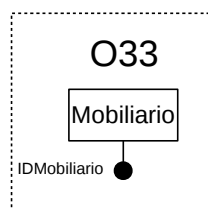
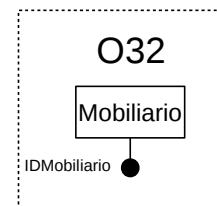
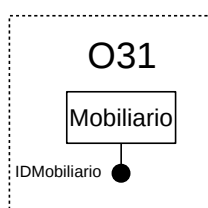
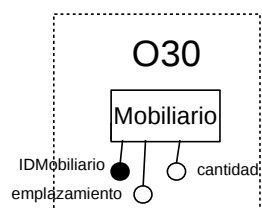
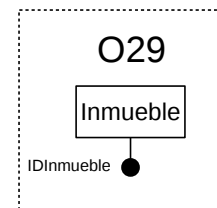
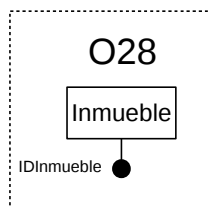
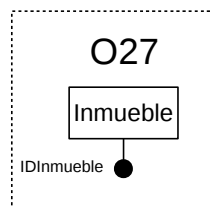
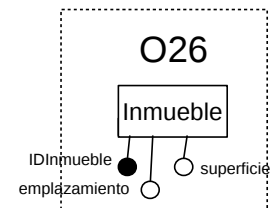
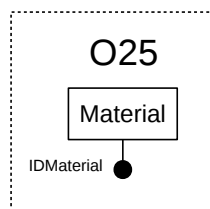
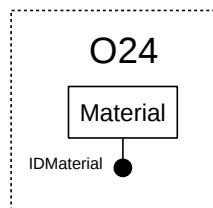
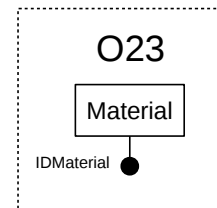
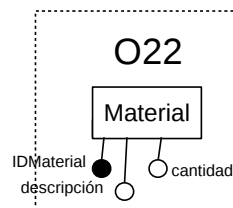
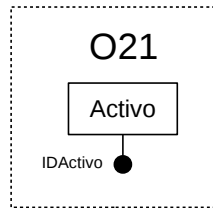
- Op.52. Dar de baja incidencia identificado por IDIncidencia.
- Op.53. Modificar incidencia identificado por IDIncidencia.
- Op.54. Dar de alta candidatos: IDCandidato, nombre, DNI.
- Op.55. Dar de baja candidatos identificado por IDCandidato.
- Op.56. Modificar candidatos identificado por IDCandidato.
- Op.57. Consultar candidatos identificado por IDCandidato.

Esquemas de operación (por área funcional)

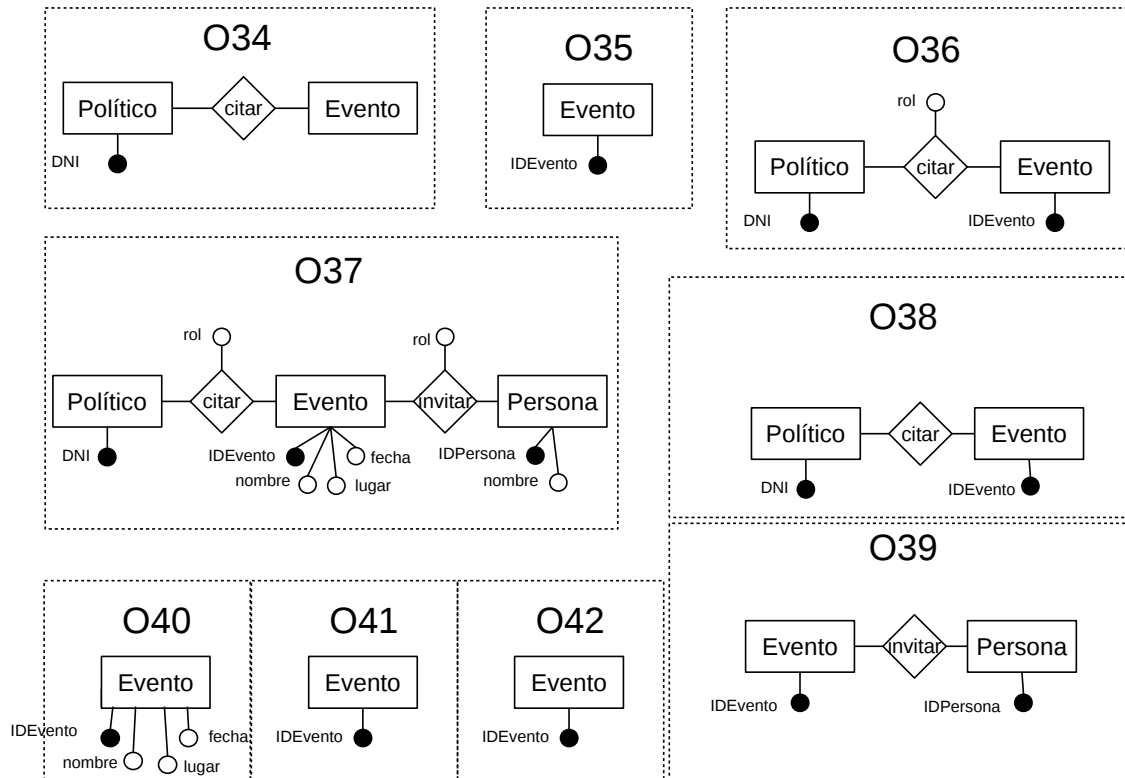
Área financiera



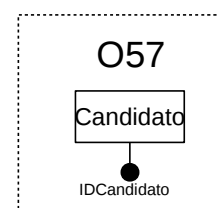
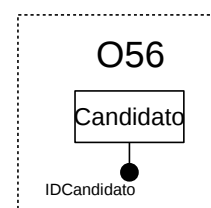
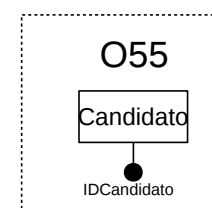
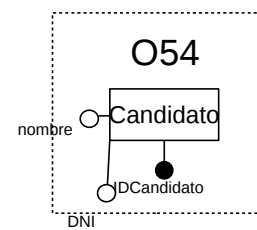
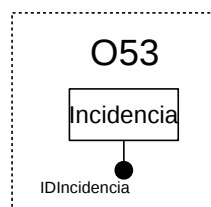
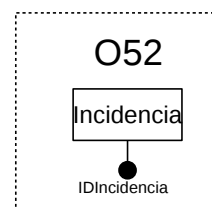
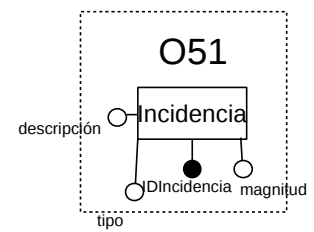
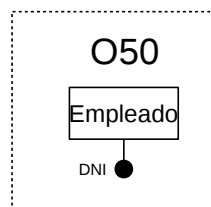
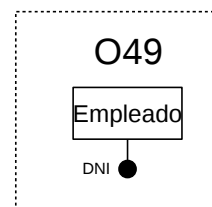
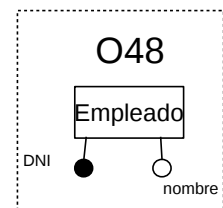
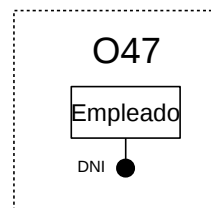
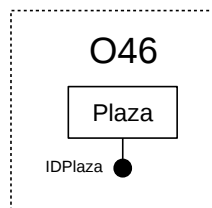
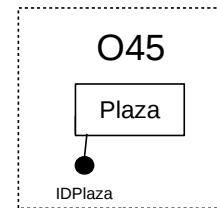
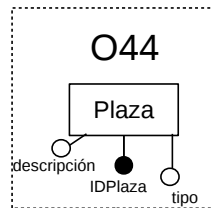
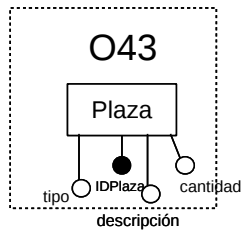
Área de activos



Área de agenda

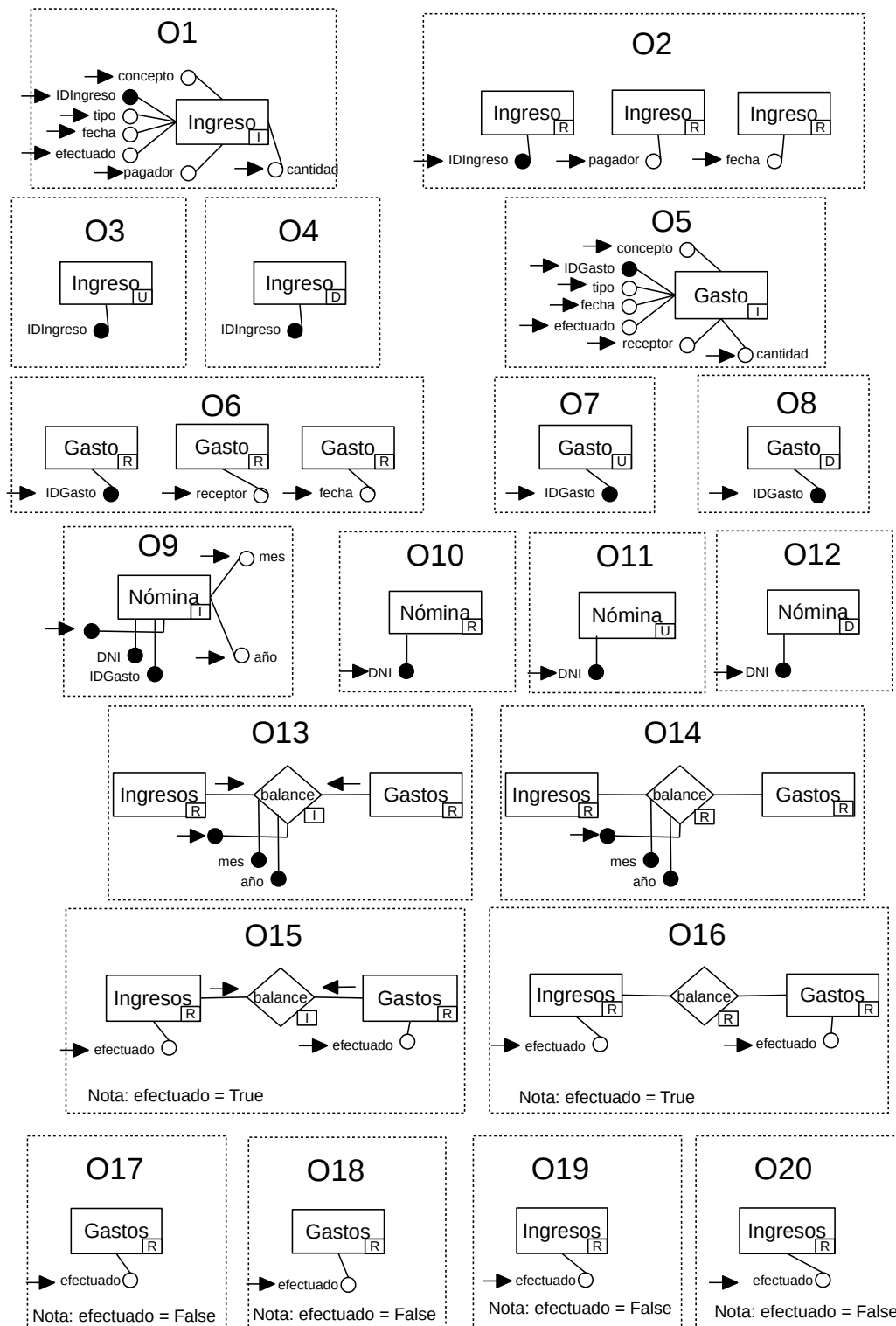


Área de recursos humanos

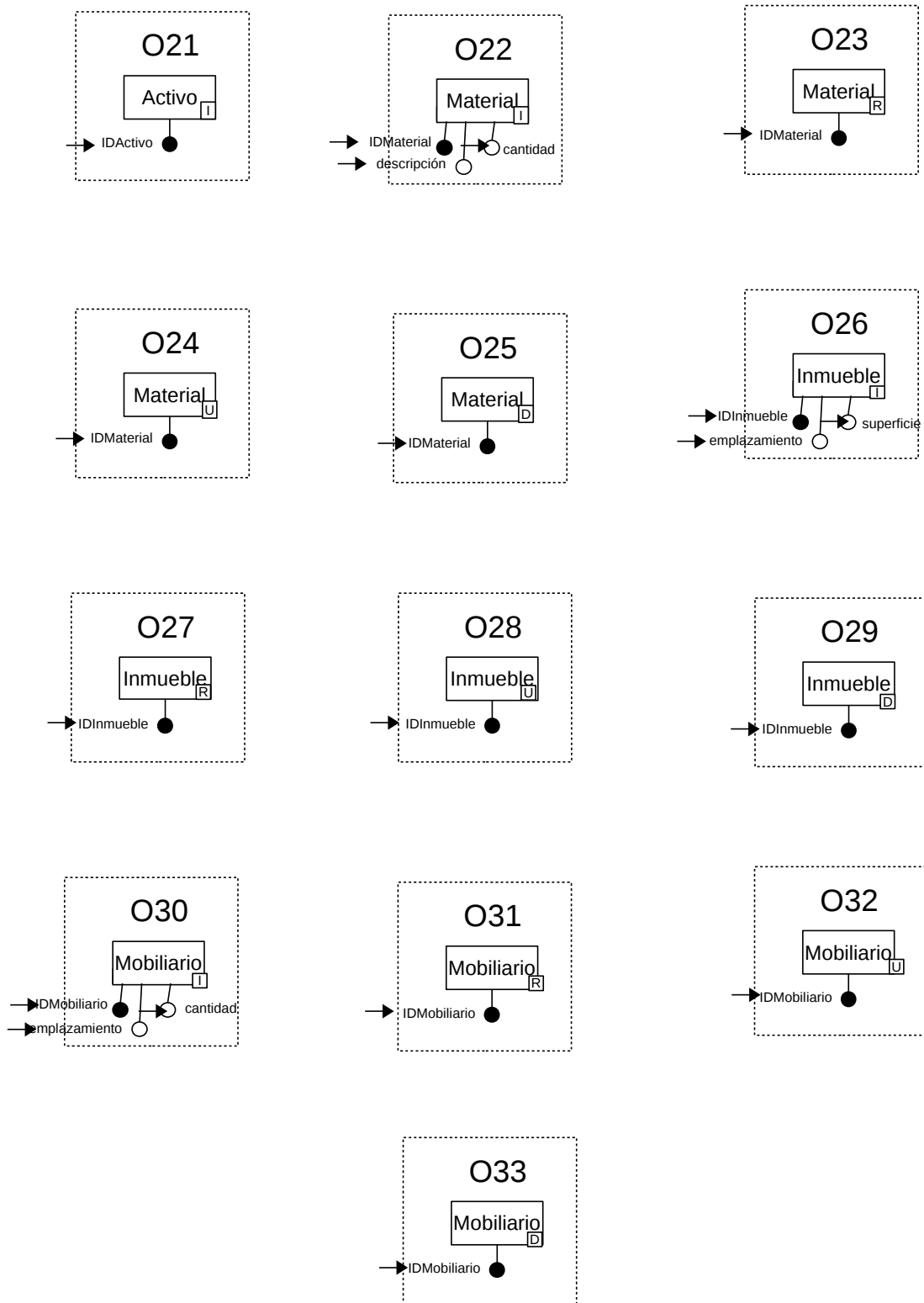


Esquemas de navegación (por área funcional)

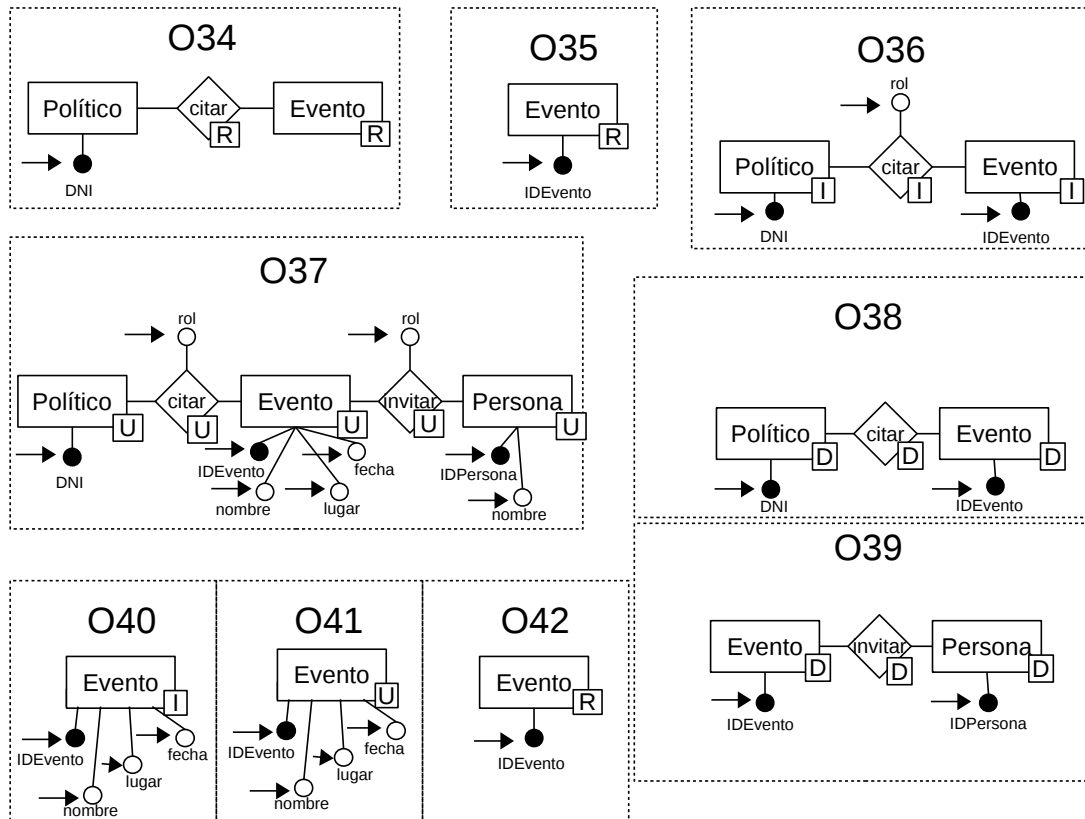
Área financiera



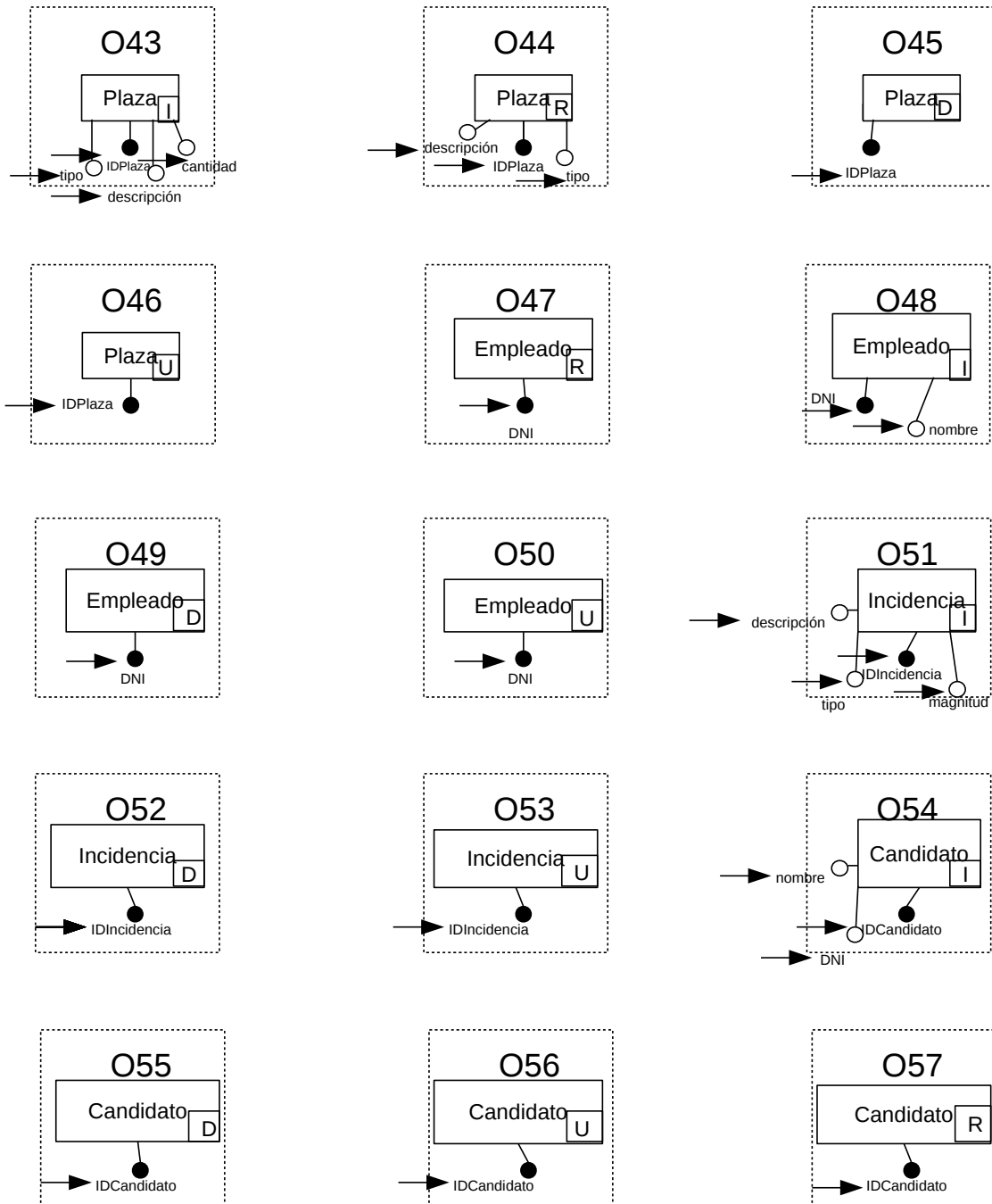
Área de activos



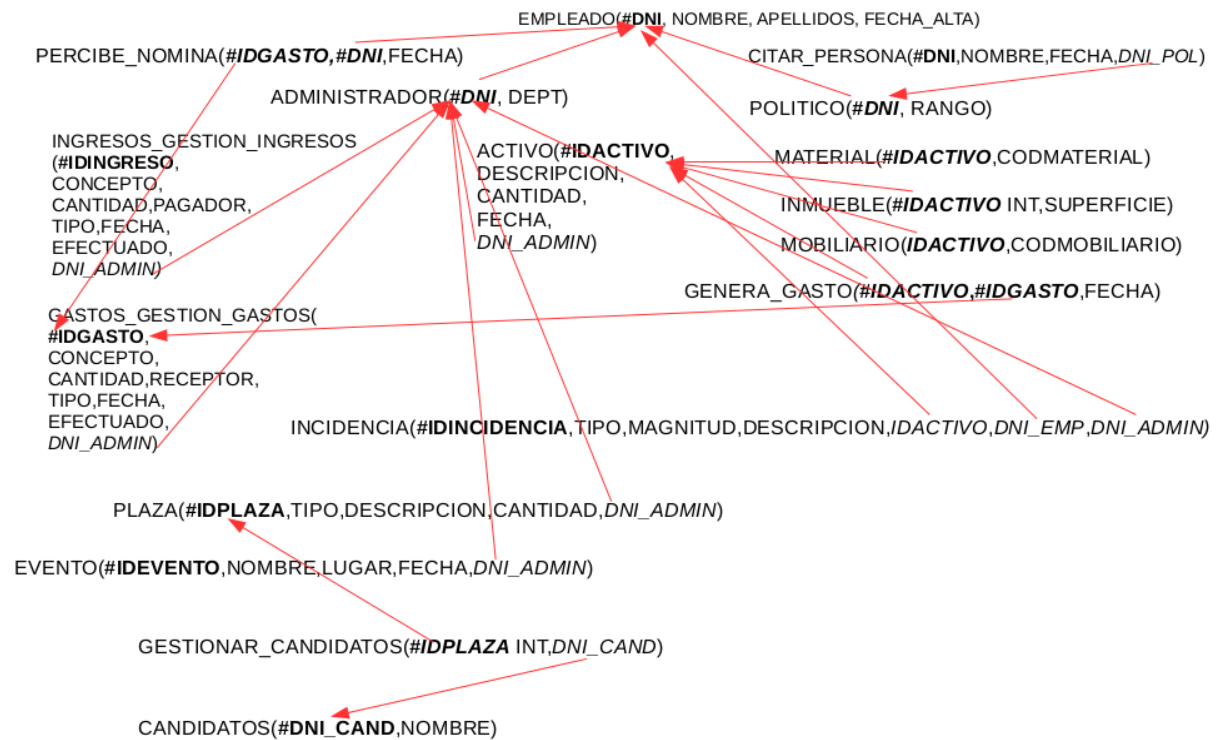
Área de agenda



Área de recursos humanos



Paso a tablas



Implementación

```
CREATE TABLE EMPLEADO (
    DNI VARCHAR(9) PRIMARY KEY,
    NOMBRE VARCHAR(20),
    APELLIDOS VARCHAR(20),
    FECHA_ALTA DATE
);

INSERT INTO EMPLEADO VALUES ('11111111A', 'UN', 'TIO', STR_TO_DATE('1/01/2012', '%d/%m/%Y'));
INSERT INTO EMPLEADO VALUES ('11111111B', 'OTRO', 'TIO', CURDATE());
INSERT INTO EMPLEADO VALUES ('22222222B', 'UN', 'POLITICO', CURDATE());
INSERT INTO EMPLEADO VALUES ('22222222C', 'OTRO', 'POLITICO', CURDATE());

CREATE TABLE ADMINISTRADOR (
    DNI VARCHAR(9) PRIMARY KEY,
    DEPT VARCHAR(20),
    FOREIGN KEY (DNI) REFERENCES EMPLEADO(DNI) ON DELETE CASCADE
);

INSERT INTO ADMINISTRADOR VALUES ('11111111A', 'FINANZAS');
INSERT INTO ADMINISTRADOR VALUES ('11111111B', 'RRHH');
INSERT INTO ADMINISTRADOR VALUES ('22222222B', 'SECRETARIO GENERAL');
INSERT INTO ADMINISTRADOR VALUES ('22222222c', 'ADJUNTO');
```

```
CREATE TABLE POLITICO(  
    DNI VARCHAR(9) PRIMARY KEY,  
    RANGO VARCHAR(20),  
    FOREIGN KEY (DNI) REFERENCES EMPLEADO(DNI) ON DELETE CASCADE  
);
```

```
INSERT INTO POLITICO VALUES('22222222B','SECRETARIO GENERAL');  
INSERT INTO POLITICO VALUES('22222222C','ADJUNTO');
```

```
CREATE TABLE INGRESOS_GESTION_INGRESOS(  
    IDINGRESO INT PRIMARY KEY,  
    CONCEPTO VARCHAR(30),  
    CANTIDAD DEC(10,2),  
    PAGADOR VARCHAR(40),  
    TIPO VARCHAR(10),  
    FECHA DATE,  
    EFECTUADO BOOL,  
    DNI_ADMIN VARCHAR(9) NOT NULL,  
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE  
);
```

```
INSERT INTO INGRESOS_GESTION_INGRESOS VALUES(1,'UN CONCEPTO',300,'UNA  
PERSONA','DONACION',CURDATE(),FALSE,'22222222B');
```

```
INSERT INTO INGRESOS_GESTION_INGRESOS VALUES (2, 'SEGUNDO
CONCEPTO', 10.54, 'PEPE LOPEZ', 'DONACION', CURDATE(), TRUE, '22222222B');
```

```
CREATE TABLE GASTOS_GESTION_GASTOS (
    IDGASTO INT PRIMARY KEY,
    CONCEPTO VARCHAR(30),
    CANTIDAD DEC(10,2),
    RECEPTOR VARCHAR(40),
    TIPO VARCHAR(20),
    FECHA DATE,
    EFECTUADO BOOL,
    DNI_ADMIN VARCHAR(9) NOT NULL,
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE
);
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (1, 'UN CONCEPTO INGRESO', 10.54, 'UN
RECEPTOR', 'FAVORES', CURDATE(), TRUE, '22222222B');
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (2, 'OTRO CONCEPTO
INGRESO', 1032.27, NULL, 'NOMINA', CURDATE(), FALSE, '22222222B');
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (3, 'REPARACION INMUEBLE', 100, 'OBRAS
PEPE', 'MANTENIMIENTO', CURDATE(), FALSE, '22222222B');
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (4, 'REPARACION INMUEBLE', 200, 'OBRAS
PEPE', 'MANTENIMIENTO', CURDATE(), FALSE, '22222222B');
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (5, 'REPARACION INMUEBLE', 300, 'OBRAS
PEPE', 'MANTENIMIENTO', CURDATE(), FALSE, '22222222B');
```

```
INSERT INTO GASTOS_GESTION_GASTOS VALUES (6, 'LA NOMINA DE
NOSEQUIEN', 2347.12, NULL, 'NOMINA', CURDATE(), FALSE, '22222222B');
```

```
CREATE TABLE ACTIVO(  
    IDACTIVO INT PRIMARY KEY,  
    DESCRIPCION VARCHAR(50),  
    CANTIDAD DEC(10,1),  
    FECHA DATE,  
    DNI_ADMIN VARCHAR(9),  
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE  
);  
  
INSERT INTO ACTIVO VALUES(1,'CASA EN CALLE FALSA',1,CURDATE(),'11111111A');  
INSERT INTO ACTIVO VALUES(2,'CASA EN OTRA CALLE',1,CURDATE(),'11111111A');  
INSERT INTO ACTIVO VALUES(3,'MESA DE OFICINA',7,CURDATE(),'11111111A');  
INSERT INTO ACTIVO VALUES(4,'COMPUTADOR',14,CURDATE(),'11111111A');  
INSERT INTO ACTIVO VALUES(5,'BOLIGRAFO',128,CURDATE(),'11111111A');  
INSERT INTO ACTIVO VALUES(6,'TRAPO',72,CURDATE(),'11111111A');  
  
CREATE TABLE GENERA_GASTO(  
    IDACTIVO INT,  
    IDGASTO INT,  
    FECHA DATE,  
    PRIMARY KEY (IDACTIVO,IDGASTO)  
);  
  
INSERT INTO GENERA_GASTO VALUES(1,3,CURDATE());  
INSERT INTO GENERA_GASTO VALUES(1,4,CURDATE());  
INSERT INTO GENERA_GASTO VALUES(1,5,CURDATE());
```

```
CREATE TABLE PERCIBE_NOMINA(  
    IDGASTO INT,  
    DNI VARCHAR(9),  
    FECHA DATE,  
    PRIMARY KEY(IDGASTO,DNI),  
    FOREIGN KEY (DNI) REFERENCES EMPLEADO(DNI) ON DELETE CASCADE,  
    FOREIGN KEY (IDGASTO) REFERENCES GASTOS_GESTION_GASTOS(IDGASTO) ON DELETE  
    CASCADE  
);
```

```
INSERT INTO PERCIBE_NOMINA VALUES(2,'11111111A',CURDATE());
```

```
INSERT INTO PERCIBE_NOMINA VALUES(6,'11111111B',CURDATE());
```

```
CREATE TABLE MATERIAL(  
    IDACTIVO INT,  
    CODMATERIAL VARCHAR(15),  
    FOREIGN KEY (IDACTIVO) REFERENCES ACTIVO(IDACTIVO) ON DELETE CASCADE  
);
```

```
INSERT INTO MATERIAL VALUES(5,'BIC0742');
```

```
INSERT INTO MATERIAL VALUES(6,'VILFF67');
```

```
CREATE TABLE INMUEBLE(  
    IDACTIVO INT,  
    SUPERFICIE INT,  
    FOREIGN KEY (IDACTIVO) REFERENCES ACTIVO(IDACTIVO) ON DELETE CASCADE  
);
```

```
INSERT INTO MATERIAL VALUES(1,523);  
INSERT INTO MATERIAL VALUES(2,1024);
```

```
CREATE TABLE MOBILIARIO(  
    IDACTIVO INT,  
    CODMOBILIARIO VARCHAR(15),  
    FOREIGN KEY (IDACTIVO) REFERENCES ACTIVO(IDACTIVO) ON DELETE CASCADE  
);
```

```
INSERT INTO MATERIAL VALUES(3,'HOYLFRONT');  
INSERT INTO MATERIAL VALUES(4,'MEDION124');
```

```
CREATE TABLE INCIDENCIA(  
    IDINCIDENCIA INT PRIMARY KEY,  
    TIPO VARCHAR(20),  
    MAGNITUD VARCHAR(20),  
    DESCRIPCION VARCHAR(100),  
    IDACTIVO INT,  
    DNI_EMP VARCHAR(9),  
    DNI_ADMIN VARCHAR(9),  
    FOREIGN KEY (IDACTIVO) REFERENCES ACTIVO(IDACTIVO) ON DELETE CASCADE,  
    FOREIGN KEY (DNI_EMP) REFERENCES EMPLEADO(DNI) ON DELETE CASCADE,  
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE  
);
```

```
INSERT INTO INCIDENCIA VALUES(1,  
    'ROTURA',  
    'LEVE',  
    'SE HA ROTO EL CRISTAL DE UNA MESA EN LA SEDE MADRID',  
    3,  
    '22222222B',  
    '11111111B'  
);
```

```
INSERT INTO INCIDENCIA VALUES(2,  
    'INCENDIO',  
    'GRAVE',  
    'SE INICIO UN FUEGO EN LA SALA DE CALLE FALSA',  
    1,  
    NULL,  
    '11111111B'
```



```
);
```

```
CREATE TABLE PLAZA(  
    IDPLAZA INT PRIMARY KEY,  
    TIPO VARCHAR(20),  
    DESCRIPCION VARCHAR(100),  
    CANTIDAD INT,  
    DNI_ADMIN VARCHAR(9),  
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE  
);
```

```
INSERT INTO PLAZA VALUES(  
    1,  
    'ADMINISTRATIVO',  
    'SE NECESITA UN ADMINISTRADOR DE FINANZAS',  
    1,  
    '11111111A'  
);
```

```
INSERT INTO PLAZA VALUES(  
    2,  
    'PRACTICAS',  
    'POLITICO BECARIO EN PRACTICAS EN MADRID',  
    1,  
    '11111111A'  
);
```

```
CREATE TABLE CANDIDATOS (
```

```
    DNI_CAND VARCHAR(9) PRIMARY KEY,
```

```
    NOMBRE VARCHAR(20)
```

```
);
```

```
INSERT INTO CANDIDATOS VALUES('12345678A','PEPE LOPEZ');
```

```
INSERT INTO CANDIDATOS VALUES('12345678B','ALMUDENA PEREZ');
```

```
CREATE TABLE GESTIONAR_CANDIDATOS (
```

```
    IDPLAZA INT,
```

```
    DNI_CAND VARCHAR(9),
```

```
    FOREIGN KEY (IDPLAZA) REFERENCES PLAZA(IDPLAZA) ON DELETE CASCADE,
```

```
    FOREIGN KEY (DNI_CAND) REFERENCES CANDIDATOS(DNI_CAND) ON DELETE CASCADE
```

```
);
```

```
INSERT INTO GESTIONAR_CANDIDATOS VALUES(1,'12345678A');
```

```
INSERT INTO GESTIONAR_CANDIDATOS VALUES(2,'12345678B');
```

```
CREATE TABLE EVENTO (
```

```
    IDEVENTO INT,
```

```
    NOMBRE VARCHAR(20),
```

```
    LUGAR VARCHAR(20),
```

```
    FECHA DATE,
```

```
    DNI_ADMIN VARCHAR(9),
```

```
    FOREIGN KEY (DNI_ADMIN) REFERENCES ADMINISTRADOR(DNI) ON DELETE CASCADE
```

```
);
```

```
INSERT INTO EVENTO VALUES(1,'CONGRESO','MADRID',CURDATE(),'11111111A');
```

```
INSERT INTO EVENTO VALUES (2, 'MITIN', 'LEON', CURDATE(), '11111111A');
```

```
CREATE TABLE CITAR_PERSONA (  
    DNI VARCHAR(9) PRIMARY KEY,  
    NOMBRE VARCHAR(20),  
    FECHA DATE,  
    DNI_POL VARCHAR(9),  
    FOREIGN KEY (DNI_POL) REFERENCES POLITICO(DNI) ON DELETE CASCADE  
);
```

```
INSERT INTO CITAR_PERSONA  
VALUES ('DNIP1', 'PEPERSONA', CURDATE(), '22222222B');
```

```
INSERT INTO CITAR_PERSONA VALUES ('DNIP2', 'OTRA  
PERSONA', CURDATE(), '22222222B');
```

Procedimiento cálculo del balance

Este procedimiento calcula el balance de aquéllos ingresos/gastos que ya han sido efectivos (SATISFECHO=TRUE) o que todavía no lo han sido (SATISFECHO=false). Se implementan los cursores dentro del procedimiento.

```
/*Esta línea es para propósitos de desarrollo, cada vez que se hacen
cambios en el SP, hay que ejecutarla para borrar lo que teníamos.*/

/*DROP PROCEDURE IF EXISTS SP_CALCULAR_BALANCE;*/

DELIMITER //

CREATE PROCEDURE SP_CALCULAR_BALANCE(SATISFECHO TINYINT/*, OUT BALANCE
DECIMAL(10,2)*/)

BEGIN

    DECLARE cantidad_actual_gasto DECIMAL(10,2) DEFAULT 0.00;

    DECLARE cantidad_actual_ingreso DECIMAL(10,2) DEFAULT 0.00;

    DECLARE efectuado_actual_gasto TINYINT(1) DEFAULT 0;

    DECLARE efectuado_actual_ingreso TINYINT(1) DEFAULT 0;

    DECLARE ingreso_total DECIMAL(10,2) DEFAULT 0.00;

    DECLARE gasto_total DECIMAL(10,2) DEFAULT 0.00;

    DECLARE num_ingresos INT DEFAULT 0;

    DECLARE num_gastos INT DEFAULT 0;

    DECLARE contador_ingresos INT DEFAULT 0;

    DECLARE contador_gastos INT DEFAULT 0;

    DECLARE cursor_gastos CURSOR FOR SELECT CANTIDAD,EFECTUADO FROM
gastos_gestion_gastos;

    DECLARE cursor_ingresos CURSOR FOR SELECT CANTIDAD,EFECTUADO FROM
ingresos_gestion_ingresos;

    SELECT count(*) FROM ingresos_gestion_ingresos INTO num_ingresos;

    SELECT count(*) FROM gastos_gestion_gastos INTO num_gastos;
```

```
/*Proceso los gastos*/

OPEN cursor_gastos;

lectura_gastos:LOOP

    SET contador_gastos = contador_gastos+1;

    IF contador_gastos>num_gastos THEN

        LEAVE lectura_gastos;

    ELSE

        FETCH cursor_gastos INTO
cantidad_actual_gasto,efectuado_actual_gasto;

        IF efectuado_actual_gasto=SATISFECHO THEN

            SET gasto_total=gasto_total+cantidad_actual_gasto;

        END IF;

    END IF;

END LOOP lectura_gastos;

CLOSE cursor_gastos;


/*Proceso los ingresos*/

OPEN cursor_ingresos;

lectura_ingresos:LOOP

    SET contador_ingresos = contador_ingresos+1;

    IF contador_ingresos>num_ingresos THEN

        LEAVE lectura_ingresos;

    ELSE

        FETCH cursor_ingresos INTO
cantidad_actual_ingreso,efectuado_actual_ingreso;

        IF efectuado_actual_ingreso=SATISFECHO THEN

            SET
ingreso_total=ingreso_total+cantidad_actual_ingreso;

        END IF;
```

```
        END IF;

    END LOOP lectura_ingresos;

    CLOSE cursor_ingresos;

    /*La siguiente línea podría hacerse si tuviéramos un parámetro OUT
    BALANCE*/

    /*SET BALANCE=ingreso_total-gasto_total;*/

    /*ingresos - gastos*/

    SELECT ingreso_total-gasto_total;

END;
```

La llamada al procedimiento se hace con la siguiente línea.

```
CALL SP_CALCULAR_BALANCE(TRUE);
```

A continuación se detallan las pruebas diseñadas para comprobar que el procedimiento funciona como es debido.

```
/*Por si se quiere comprobar, éstas serían las consultas.*/
```

```
SELECT GGG.CANTIDAD AS C_G
FROM GASTOS_GESTION_GASTOS GGG
WHERE GGG.EFFECTUADO=0;
```

```
SELECT IGI.CANTIDAD AS C_I
FROM INGRESOS_GESTION_INGRESOS IGI
WHERE IGI.EFFECTUADO=0;
```

```
SELECT GGG.CANTIDAD AS C_G
FROM GASTOS_GESTION_GASTOS GGG
WHERE GGG.EFFECTUADO=1;
```

```
SELECT IGI.CANTIDAD AS C_I
```

```
FROM INGRESOS_GESTION_INGRESOS IGI
```

```
WHERE IGI.EFFECTUADO=1;
```

Trigger (Disparador por INSERT)

El siguiente trigger comprueba que no se dé de alta un empleado en una fecha posterior a hoy. El trigger ya comprende el manejo de excepciones.

```
DELIMITER //
```

```
DROP TRIGGER IF EXISTS TR_INSERT_EMPLEADO//
```

```
CREATE TRIGGER TR_INSERT_EMPLEADO BEFORE INSERT ON EMPLEADO
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE mensaje_error VARCHAR(128);
```

```
    IF NEW.FECHA_ALTA > CURDATE() THEN
```

```
        SET mensaje_error = concat('TR_INSERT_EMPLEADO: No se puede dar de  
alta un empleado en fecha futura: ', cast(NEW.FECHA_ALTA as char));
```

```
        SIGNAL SQLSTATE '46001' SET MESSAGE_TEXT = mensaje_error;
```

```
    end if;
```

```
end
```

```
//
```

Se diseña la siguiente prueba para comprobar que el trigger funciona como es debido (Mientras no hayamos alcanzado el 1 de Enero de 2017, esta sentencia INSERT no funcionará.

```
/*Esta es una prueba para ver que salta el trigger*/
```

```
INSERT INTO EMPLEADO VALUES('25429841A','UNA','PRUEBA','2017-1-01');
```