

# GranaCar

## *Empresa de alquiler de coches*

### Grupo 2

Ángela Izquierdo García  
Pilar Navarro Ramírez  
Maximino Suarez van Gelderen  
Víctor Castro Serrano

### Departamentos/subsistemas/unidades funcionales

- Contabilidad: Pilar Navarro Ramírez (Subsistema 1)
- Recursos Humanos: Maximino Suárez van Gelderen (Subsistema 2)
- Almacén: Víctor Castro Serrano (Subsistema 3)
- Logística: Ángela Izquierdo García (Subsistema 4)

## Modelo de negocio

Ofrecemos servicio de alquiler de coches para particulares que es posible contratar en la oficina o de forma online; en la web o una aplicación, antes de llegar a la oficina para la recogida y permitiendo al cliente ahorrar tiempo. Generamos ingresos con el alquiler de dichos coches.

- El departamento de **contabilidad** controla y registra los gastos e ingresos y demás operaciones económicas que realiza la empresa. Aquí se encargará del registro de las facturas de los productos adquiridos por la empresa, registro de las ganancias obtenidas a través de los alquileres de vehículos a los clientes, el control de los bienes que posee la empresa, así como de un seguimiento de la situación económica en ciertos periodos de tiempo y la realización de un balance financiero con el fin de tener una visión general de la economía de la empresa y su desarrollo durante la actividad de la misma.
- El departamento de **recursos humanos**, o RRHH (también llamado departamento de personal) es el responsable de la selección y contratación del personal idóneo para cada puesto laboral vacante, y también del mantenimiento de un plantel laboral con posibilidades y comodidades, en buenos ambientes, con respeto y

tolerancia. En nuestro caso debemos contratar personal para ofrecer los servicios de alquiler en las oficinas y atender las llamadas, personal para el mantenimiento de la página web y personal de limpieza de coches.

- El departamento de **almacén** gestiona el estado de la flota de vehículos. Se encarga de revisar los vehículos para decidir si necesitan reparaciones para llevarlos al taller. Controla los vehículos que tiene la empresa sabiendo si se encuentran disponibles o en el parking de la empresa. Gestiona la adquisición de nuevos vehículos y la venta de los mismos cuando se decide que deben ser dados de baja de la flota de la empresa.
- El departamento de **logística** se encarga del transporte de los vehículos del garaje al taller y viceversa, la localización de los mismos, su disponibilidad, así como de la gestión de los alquileres de estos a los clientes.

## Requisitos

### 1. Contabilidad

#### **RF1.1 Elaborar y registrar facturas**

Entrada: Agente externo: Contable. Acción: solicitar inserción factura

Requisito de datos de entrada: RDE1.1

BD: Requisito de datos de escritura RDW1.1.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE1.1:** Datos de entrada de una nueva factura

IDfactura: Cadena de caracteres (formato Fxxxx donde cada x corresponde a un dígito)

Fecha factura: Cadena de caracteres de la forma DD/MM/YY

Proveedor: Cadena de caracteres (50 caracteres máximo)

Productos: Lista de datos de la forma (Nombre Producto, Cantidad, Precio) donde

- IDProducto: Cadena de caracteres (formato Pxxxx donde cada x corresponde a un dígito)

- Nombre Producto: Cadena de caracteres (60 caracteres máximo)

- Cantidad: Número real o entero (expresada en número de piezas o peso según corresponda)

- Precio: Número real con dos decimales (expresado en euros)

Precio total: Número real con dos decimales (expresado en euros)

**RDW1.1:** Datos almacenados de factura

Los mismos datos que RDE1.1.

## **RF1.2 Elaborar informes contables**

Entrada: Agente externo: Contable. Acción: solicitar inserción informe

Requisito de datos de entrada: RDE1.2

BD: Requisito de datos de escritura RDW1.2.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE1.2:** Datos de entrada de un informe

IDinforme: Cadena de caracteres (formato lxxx donde cada x corresponde a un dígito)

Fecha realización: Cadena de caracteres de la forma DD/MM/YY

Bienes: Lista de bienes de que dispone la empresa, donde cada bien tiene la forma

- ID: Cadena de caracteres (formato Wxxx donde cada x corresponde a un dígito)
- Nombre: Cadena de caracteres (máximo 50 caracteres)
- Descripción: Cadena de caracteres (máximo 200 caracteres)
- Valor: Número real con dos decimales (expresado en euros)

**RDW1.2:** Datos almacenados de informe

Los mismos datos que RDE1.2.

## **RF1.3 Registrar ganancias**

Entrada: Agente externo: Contable. Acción: solicitar inserción ganancia

Requisito de datos de entrada: RDE1.3

BD: Requisito de datos de escritura RDW1.3.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE1.3:** Identificador del alquiler cuya ganancia asociada se quiere registrar y la ganancia.

IDalquiler: Cadena de caracteres (formato Axxxx donde cada x corresponde a un dígito)

Ganancia: Número real con dos decimales (expresado en euros)

**RDW1.3:** Datos almacenados de alquiler

Ganancia: Número real con dos decimales (expresado en euros)

## **RF1.4 Consultar documentos financieros**

Entrada: Agente externo: Contable.

Acción: solicitar consulta facturas, informes y alquileres

Requisito de datos de entrada: RDE1.4

BD: Requisito de datos de lectura RDR1.4.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: RDS1.4

**RDE1.4:** Identificadores de los documentos a consultar

Identificadores: Lista de cadenas de caracteres (con el formato adecuado de los identificadores de los documentos que se desean consultar)

**RDR1.4:** Datos almacenados de facturas, informes y alquileres cuyos identificadores corresponden a los introducidos (RDE1.4).

Los mismos datos que RDW1.1, RDW1.2 y RDW1.3 respectivamente.

**RDS1.4:** Listado de facturas, informes y alquileres, con los mismos datos que RDR1.4.

### **RF1.5 Realizar balance financiero**

Entrada: Agente externo: Contable. Acción: solicitar inserción balance

Requisito de datos de entrada: RDE1.5

BD: Requisito de datos de escritura RDW1.5.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE1.5:** Datos de entrada de un documento de balance financiero

IDbalance: Cadena de caracteres (formato Bxxxx donde cada x corresponde a un dígito)

Fecha realización: Cadena de caracteres de la forma DD/MM/YY

Total gastos: Número real con dos decimales (expresado en euros)

Total ingresos: Número real con dos decimales (expresado en euros)

Bienes actuales: Número real con dos decimales (expresado en euros)

Deudas pendientes: Número real con dos decimales (expresado en euros)

**RDW1.5:** Datos almacenados del balance

Los mismos datos que RDE1.5.

### **RF1.6 Consultar situación financiera**

Entrada: Agente externo: contable.

Acción: solicitar consulta balance financiero

Requisito de datos de entrada: RDE1.6

BD: Requisito de datos de lectura RDR1.6.

Salida: Agente externo: Contable. Acción: confirmación resultado

Requisito de datos de salida: RDS1.6

**RDE1.6**: Identificadores de los documentos de balance a consultar

Identificadores: Lista de cadenas de caracteres (con el formato Bxxx donde cada x corresponde a un dígito)

**RDR1.6**: Datos almacenados de los documentos de balance cuyos identificadores corresponden a los introducidos (RDE1.6).

Los mismos datos que RDW1.5.

**RDS1.6**: Datos de los documentos de balance consultados, con los mismos datos que RDR1.6.

**RS1.1**: Ningún campo puede ser vacío.

RF: RF 1.1, RF 1.2, RF 1.3, RF 1.5

RD(s): RDE 1.1, RDE 1.2, RDE 1.3, RDE 1.5, RDW 1.1, RDW 1.2, RDW 1.3, RDW 1.5

Descripción: Si alguno de los campos es vacío, no se realiza la inserción y se pide que se introduzca el campo faltante

**RS1.2**: El formato de los datos introducidos debe ser correcto

RF: RF 1.1, RF 1.2, RF 1.3, RF 1.5

RD(s): RDE 1.1, RDE 1.2, RDE 1.3, RDE 1.5, RDW 1.1, RDW 1.2, RDW 1.3, RDW 1.5

Descripción: Si alguno de los datos introducidos tiene un formato incorrecto, no se realiza la inserción y se pide que se corrija

**RS1.3**: El identificador de factura es único en el sistema

RF: RF1.1

RD(s): RDE1.1, RDW1.1

Descripción: El identificador de factura introducido ya existe en la base de datos, no se realiza la inserción y se devuelve un error

**RS1.4**: El identificador de informe es único en el sistema

RF: RF1.2

RD(s): RDE1.2, RDW1.2

Descripción: El identificador de informe introducido ya existe en la base de datos, no se realiza la inserción y se devuelve un error

**RS1.5**: El identificador de alquiler no existe

RF: RF1.3

RD(s): RDE1.3

Descripción: El identificador de alquiler introducido no se corresponde con un identificador existente en la base de datos, no se realiza la inserción y se devuelve un error

**RS1.6:** El identificador de balance es único en el sistema

RF: RF1.5

RD(s): RDE1.5, RDW1.5

Descripción: El identificador de balance introducido ya existe en la base de datos, no se realiza la inserción y se devuelve un error

**RS1.7:** El identificador no existe

RF: RF1.4, RF1.6

RD(s): RDE1.4, RDE1.6

Descripción: Alguno de los identificadores introducidos no se corresponde con un identificador existente en la base de datos, se devuelve un error asociado a dicho identificador.

## **2. Recursos humanos**

### **RF2.1 Dar de alta a un empleado**

Entrada: Agente externo: Trabajador de RRHH. Acción: solicitar inserción de un nuevo empleado

Requisito de datos de entrada: RDE2.1

BD: Requisito de datos de escritura RDW2.1.

Salida: Agente externo:Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE2.1:** Datos de entrada de un nuevo empleado

ID: Cadena alfanumérica. (Formato DNI- 9 numero y una letra)

Nombre: Cadena de 20 caracteres

Apellidos: Cadena de 40 caracteres

Departamento: Cadena de 40 caracteres

Puesto: Cadena de 40 caracteres

Fecha de alta: Cadena de caracteres de la forma DD/MM/YY

Sueldo: Cadena numérica con dos decimales

Cuenta bancaria: Cadena alfanumérica (24 digitos)

Estado de empleado: Cadena de 20 caracteres

**RDW2.1:** Datos almacenados de empleado

Los mismos datos que RDE2.1.

## **RF2.2 Dar de baja a un empleado**

Entrada: Agente externo: Trabajador de RRHH. Acción: solicitar borrado de un empleado

Requisito de datos de entrada: RDE2.2

BD: Requisito de datos de escritura RDW2.2

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE2.2:** ID del empleado a dar de baja

Nombre de puesto

Estado de empleado: Cadena de 40 caracteres

Fecha de baja: Cadena de caracteres de la forma DD/MM/YY

**RDW2.2:** Estado de empleado, Fecha de baja

## **RF2.3 Pagar nómina**

Entrada: Agente externo: Trabajador de RRHH. Acción: pagar nómina de empleado

Requisito de datos de entrada: RDE2.3

BD: Requisito de datos de escritura RDW2.3

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE2.3:** ID del empleado

Fechas de pago: Array de Cadena de caracteres de la forma DD/MM/YY

**RDR2.3:** Sueldo , Cuenta bancaria

**RDW2.3:** Datos de realización de una nueva paga de nómina

Añadir nueva entrada con fecha del pago a RDE2.3

## **RF2.4 Añadir puesto vacante**

Entrada: Agente externo: Trabajador de RRHH. Acción: añadir un puesto vacante en la empresa

Requisito de datos de entrada: RDE2.4

BD: Requisito de datos de escritura RDW2.4

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE2.4:** Datos de entrada de puestos vacantes

Nombre del puesto: Cadena de 40 caracteres

Departamento: Cadena de 40 caracteres

Número de vacantes: Cadena numérica

Breve descripción del puesto: Cadena de 250 caracteres

Aptitudes interesantes para el puesto: Cadena de 250 caracteres

Requisitos para el puesto: Cadena de 500 caracteres

**RDW2.4:** Datos almacenados de puestos vacantes

Los mismos datos que RDE2.4

### **RF2.5 Eliminar puesto vacante**

Entrada: Agente externo: Trabajador de RRHH. Acción: eliminar un puesto vacante en la empresa

Requisito de datos de entrada: RDE2.5

BD: Requisito de datos de escritura RDW 2.5

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: ninguno

**RDE2.5:** Nombre del puesto

ID de empleado

Los mismos que RDE2.4

**RDW2.5:** Datos almacenados de puestos vacantes

Crear vínculo entre puesto y empleado

y modificar el 'número de vacantes' de un puesto para reducirlo.

### **RF2.6 Consultar puestos vacantes**

Entrada: Agente externo: Trabajador de RRHH. Acción: consultar los puestos vacantes

Requisito de datos de entrada: ninguno

BD: Requisito de datos de lectura RDR2.6

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: RDS2.6

**RDR2.6:** Datos almacenados de puestos vacantes

Los mismos datos que RDW2.4

**RDS2.6:** Listado de puestos vacantes y su información.

Los mismos datos que RDR2.

### **RF2.7 Consultar empleados**

Entrada: Agente externo: Trabajador de RRHH. Acción: visualizar información de uno o más empleados

Requisito de datos de entrada: RDE2.7



BD: Requisito de datos de lectura RDR2.7

Salida: Agente externo: Trabajador de RRHH. Acción: confirmación resultado

Requisito de datos de salida: RDS2.7

**RDE2.7**: Identificador de los empleados a consultar

Identificadores: Lista de DNIs

**RDR2.7**: Datos almacenados de empleado/s

Los mismos datos que RDW2.1

**RDS2.7**: Datos de empleado/s.

Los mismos datos que RDR2.7

**RS2.1**: Ningún campo puede ser vacío.

RF: RF 2.1, RF 2.2, RF 2.3, RF 2.4

RD(s): RDE 2.1, RDE 2.2, RDE 2.3, RDE 2.5, RDW 2.1, RDW 2.2, RDW 2.3, RDW 2.4

Descripción: Si alguno de los campos es vacío, no se realiza la inserción y se pide que se introduzca el campo faltante

**RS2.2**: El formato de los datos introducidos debe ser correcto

RF: RF 2.1, RF 2.2, RF 2.3, RF 2.4

RD(s): RDE 2.1, RDE 2.2, RDE 2.3, RDE 2.4, RDW 2.1, RDW 2.2, RDW 2.3, RDW 2.4

Descripción: Si alguno de los datos introducidos tiene un formato incorrecto, no se realiza la inserción y se pide que se corrija

**RS2.3**: El ID del empleado es único en el sistema

RF: RF2.1

RD(s): RDE2.1, RDW2.1

Descripción: El ID del empleado introducido ya existe en la base de datos, no se realiza la inserción y se devuelve un error

**RS2.4**: La fecha de baja de empleado debe ser posterior a la fecha de alta

RF: RF2.2

RD(s): RDE2.2, RDW2.2

Descripción: La fecha de baja del empleado es anterior a la fecha de alta que ya existe en la base de datos, no se realiza la inserción y se devuelve un error.

**RS2.5**: Los datos de borrado, modificación o consulta deben existir en la base de datos

RF: RF2.4, RF2.6 RF2.7

RD(s): RDE2.4, RDE2.6, RDE2.7

Descripción: El identificador seleccionado para la modificación o consulta no está en la base de datos, no se realiza el cambio o consulta y se devuelve error.

### **3. Almacén**

#### **RF 3.1. Consultar disponibilidad de vehículo (está alquilado, o en almacén, o en el taller..)**

Entrada: Agente externo: operador almacén. Acción: solicitar disponibilidad. Requisito de datos de entrada: RDE 3.1

BD: Requisito de datos de lectura RDR3.1

Salida: Agente externo: operador almacén. Acción: ninguna. Requisito de datos de salida: RDR3.1

**RDE3.1:** datos de entrada de vehículo a consultar

-Matrícula: cadena de caracteres (mínimo 7, máximo 9)

**RDR3.1:** disponibilidad (estado) del vehículo

- Estado: cadena de caracteres entre 4 opciones: disponible, mantenimiento necesario, no disponible, en taller

**RDS3.1:** lo mismo que RDR3.1

#### **RF3.2. Dar de alta un vehículo.**

Entrada: Agente externo: operador almacén. Acción: solicitar inserción. Requisito de datos de entrada: RDE 3.2

BD: Requisito de datos de escritura RDW3.2

Salida: Agente externo: operador almacén. Acción: confirmar. Requisito de datos de salida: ninguno

**RDE3.2:** datos de entrada de un nuevo vehículo

- Matrícula: cadena de caracteres (mínimo 7, máximo 9)

- Número máximo de pasajeros: número entero

- Combustible: cadena de caracteres (máximo 30)

- Transmisión: cadena de caracteres (máximo 30)

- Tipo de vehículo: cadena de caracteres (máximo 30)

- Estado: cadena de caracteres entre 3 opciones: disponible, no disponible, en taller

- Mantenimiento: cadena de caracteres, dos opciones: necesario, no necesario

**RDW3.2:** Datos almacenados de vehículo

Los mismos datos que RDE 3.2

#### **RF 3.3. Dar de baja un vehículo**

Entrada: Agente externo: operador almacén. Acción: solicitar borrado de vehículo.

Requisito de datos de entrada: RDE 3.3

BD: Requisito de datos de escritura RDR3.3

Salida: Agente externo: operador almacén. Acción: confirmar. Requisito de datos de salida: ninguno

**RDE3.3:** matrícula de vehículo a eliminar

- Los mismos datos que RDE 3.1

**RDW3.3:** eliminación del vehículo de la base de datos. Se elimina la entrada de la base de datos.

### **RF 3.4. Consulta si un coche necesita hacer mantenimiento**

Entrada: Agente externo: operador almacén. Acción: consultar necesidad de mantenimiento. Requisito de datos de entrada: RDE 3.4

BD: Requisito de datos de lectura RDR3.4

Salida: Agente externo: operador almacén. Acción: ninguna. Requisito de datos de salida: RDS3.4

**RDE3.4:** Matrícula de vehículo a consultar

Los mismos datos que RDE3.1

**RDR3.4:** mantenimiento del vehículo

- Mantenimiento: cadena de caracteres, dos opciones: [necesario, no necesario]

**RDS3.4:** mantenimiento del vehículo

- Mantenimiento: cadena de caracteres, dos opciones: [necesario, no necesario]

### **RF 3.5. Cambiar estado vehículo**

Entrada: Agente externo: operador almacén. Acción: cambiar estado de un vehículo.

Requisito de datos de entrada: RDE 3.5

BD: Requisito de datos de escritura RDR3.5

Salida: Agente externo: operador almacén. Acción: confirmar. Requisito de datos de salida: ninguno

**RDE3.5:** datos de entrada de coche a editar y nuevo estado

- Matrícula: cadena de caracteres (mínimo 7, máximo 9)

- Estado: cadena de caracteres, entre 3 opciones [disponible, en taller, no disponible]. Opcional, si no se introduce no cambia.

- Mantenimiento: cadena de caracteres, dos opciones: [necesario, no necesario]. Opcional, si no se introduce no cambia.

**RDW3.5:** mantenimiento del vehículo. Para la matrícula dada,

- Estado: cadena de caracteres, entre 3 opciones [disponible, en taller, no disponible]. Opcional, si no se introduce no cambia.
- Mantenimiento: cadena de caracteres, dos opciones: [necesario, no necesario]. Opcional, si no se introduce no cambia.

### **RF 3.6. Consultar listado total de vehículos disponibles.**

Entrada: Agente externo: operador almacén. Acción: solicitud de vehículos disponibles.

Requisito de datos de entrada: ninguno

BD: Requisito de datos de lectura RDR3.6

Salida: Agente externo: operador almacén. Acción: ninguna. Requisito de datos de salida: RDS3.6

**RDR3.6:** consulta de listado de todos los vehículos con estado “disponible”

- Por cada elemento de la lista, un vehículo, con los datos como en RDE3.2

**RDS3.6:** listado de vehículos, con los datos como en RDR3.6

**RS 3.1:** El formato de matrícula debe ser correcto.

RF: RF 3.1

RD(s): RDE 3.1

Descripción: Si el formato de matrícula no es correcto o vacío, no se realiza la consulta y se devuelve un error.

**RS 3.2:** Ningún campo vacío o incorrecto.

RF: RF 3.2

RD(s): RDE 3.2

Descripción: Si algún campo es vacío o incorrecto, no se realiza la inserción del vehículo

**RS3.2:** Una matrícula corresponde a un único vehículo

RF: RF 3.2

RD(s): RD W3.2

Descripción: Si ya había un vehículo con esa matrícula, no se inserta el vehículo y se devuelve un error

**RS 3.3:** El formato de matrícula debe ser correcto.

RF: RF 3.3, RF 3.4, RF 3.5

RD(s): RDE 3.3, RDE 3.4, RDE 3.5, RDW 3.5

Descripción: Si el formato de matrícula no es correcto o vacío, no se realiza la eliminación y se devuelve un error.

**RS 3.4:** El estado debe ser correcto.

RF: RF 3.5

RD(s): RDE 3.5, RDW 3.5

Descripción: Si el estado no es correcto o vacío, no se realiza la modificación y se devuelve un error.

#### **4. Logística**

##### **RF 4.1 Llevar coche a taller**

Entrada: Agente externo: operador logística. Acción: llevar coche al taller . Requisito de datos de entrada RDE4.1.

BD: Requisito de datos de escritura RDW4.1.

Salida: Agente externo: operador logística. Acción: confirmación resultado y poner a false el campo de disponibilidad del vehículo (en el taller) .Requisito de datos de salida: ninguno

**RDE4.1:** Datos de entrada de llevar coche a taller.

Mismos que en RDE3.5

- Matrícula: cadena de caracteres (mínimo 7, máximo 9)
- Estado: cadena de caracteres, entre 3 opciones [disponible, en taller, no disponible].
- Mantenimiento: cadena de caracteres, dos opciones: [necesario, no necesario].

**RDW4.1.** Cambiar estado del vehículo

Estado: cadena de caracteres, entre 3 opciones [ disponible, en taller, no disponible]  
Cambiar a en taller.

Disponibilidad : Campo booleano , poner a false ( ya que no se puede alquilar)

La diferencia entre Estado y Disponibilidad es que uno indica si el automóvil está en condiciones para ser utilizado, mientras que la Disponibilidad indica que está libre para ser alquilado (además de estar en condiciones para ser utilizado)

##### **RF 4.2 Traer coche del taller**

Entrada: Agente externo: operador logística. Acción: recoger coche del taller . Requisito de datos de entrada RDE4.2.

BD: Requisito de datos de escritura RDW4.2.

Salida: Agente externo: operador logística. Acción: confirmación resultado y poner a true el campo de disponibilidad del vehículo .Requisito de datos de salida: ninguno

**RDE4.2:** Datos de entrada de traer coche de taller.

Mismos que en RDE.3.5

**RDW4.2.** Nuevo estado del vehículo

Estado: cadena de caracteres, entre 3 opciones [disponible, en taller, no disponible]

Disponibilidad : Campo booleano (cambiarlo si está disponible para alquiler o no, dependiendo si se pudo reparar)

### **RF 4.3 Solicitar servicio de coche**

Entrada: Agente externo: operador logística . Acción: solicitar servicio de coche.  
Requisito de datos de entrada RDE4.3.

BD: Requisito de datos de escritura RDW4.3.

Salida: Agente externo: operador logística . Acción: confirmación resultado y número de referencia del pedido. Requisito de datos de salida: RDS 4.1

#### **RDE4.3:**

IDalquiler: Cadena de caracteres (formato Axxxx donde cada x corresponde a un dígito)

Datos de entrada de solicitar servicio coche

Dni del cliente : Cadena de caracteres (9 caracteres alfanuméricos)

Edad : Cadena de caracteres ( 2 caracteres )

Nombre cliente: Cadena de caracteres (máximo 40 caracteres)

Duración: Número entero que indica los días

#### **RDW4.3 : Datos de escritura de solicitar servicio coche**

Nº identificador alquiler IDalquiler especificado en [RDE1.3](#)  
Matrícula: cadena de caracteres (mínimo 7, máximo 9)

#### **RDS4.3 : Datos de salida de solicitar servicio coche**

Nº identificador alquiler IDalquiler especificado en [RDE1.3](#)

### **RF 4.4 Cancelar pedido de servicio de coche**

Entrada: Agente externo: operador logística. Acción: solicitar borrado. Requisito de datos de entrada RDE4.4.

BD: Requisito de datos de escritura RDW4.4.

Salida: Agente externo: usuario. Acción: confirmación resultado y poner a true el campo de disponibilidad del vehículo (marcarlo como libre ) .Requisito de datos de salida: ninguno.

#### **RDE4.4: Datos de entrada de solicitar cancelación de servicio coche.**

Nº identificador alquiler IDalquiler especificado en [RDE1.3](#)

**RDW4.4.** Disponibilidad de vehículo

Campo de estado del vehículo ( true o false, libre o alquilado)

#### **RF 4.5 Geolocalización de automóviles**

Entrada: Agente externo: operador logística . Acción: localizar situación de un vehículo.

Requisito de datos de entrada: RDE4.5

BD: Requisito de datos de escritura: RDW4.5

Salida: Agente externo: operador logística . Acción: recibo de la la localización en la que se encuentra . Requisito de datos de salida: RDS4.5

**RDE4.5:** datos de entrada de coche a localizar

- Matrícula: cadena de caracteres (mínimo 7, máximo 9)

**RDW4.5:** actualiza la situación del vehículo

- Calle-Ciudad-Código postal : Cadena de caracteres (100)

**RDR4.5:** datos del vehículo sobre dónde está

**RDS4.5 :** datos de la localización del vehículo

#### **RF 4.6 Devolver vehículo :**

Entrada: Agente externo: operador logística. Acción: Devolver el vehículo a nuestro inventario de coches disponibles, cambiar estado del vehículo de no disponible a disponible. Requisito de datos de entrada RDE4.6.

BD: Requisito de datos de escritura: RDW4.6

Salida: Agente externo: operador logística . Acción: confirmación de la operación y marcar vehículo como libre. Requisito de datos de salida: ninguno

**RDE4.6:** Datos de entrada devolver vehículo

Nº identificador alquiler IDalquiler especificado en **RDE1.3**

**RDW4.6:** Campo de estado del vehículo (true o false, libre o alquilado, cambiar a libre)

**RS 4.1:** La matrícula debe estar registrada en el sistema y corresponder a uno de los coches del inventario .

RF: RF 4.1, RF 4.5

RD(s): RDE4.1,RDE4.5

Descripción: Si no, se devuelve error.

**RS 4.2:** Para llevar el coche al taller , mantenimiento debe estar marcado como necesario.

RF: RF 4.1,

RD(s): RDE4.1

Descripción: Si no, se devuelve error.

**RS 4.3:** La hora y fechas introducidas en RF4.3 debe estar dentro de la franja horaria de la compañía

RF: RF 4.3

RD(s): RDE4.3

Descripción: Si no, se devuelve error.

**RS 4.3:** La edad de la persona que lo va a alquilar debe ser mayor de 18 años

RF: RF 4.3

RD(s): RDE4.3

Descripción: Si no, se devuelve error.

**RS 4.4:** El nº identificador alquiler IDalquiler debe estar registrado en el sistema

y existir.

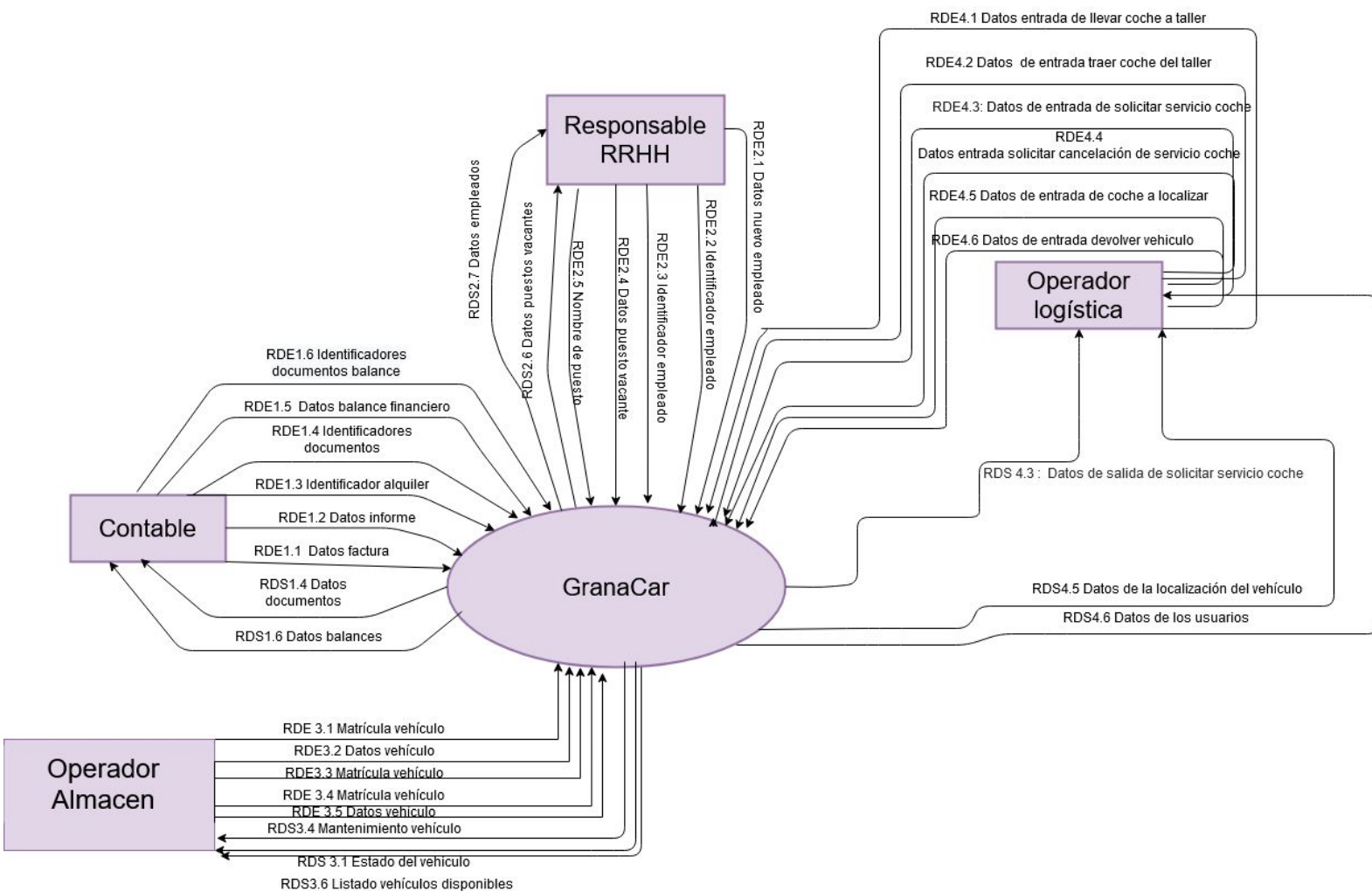
RD(s): RDE4.3,RDE4.6

Descripción: Si no, se devuelve error.

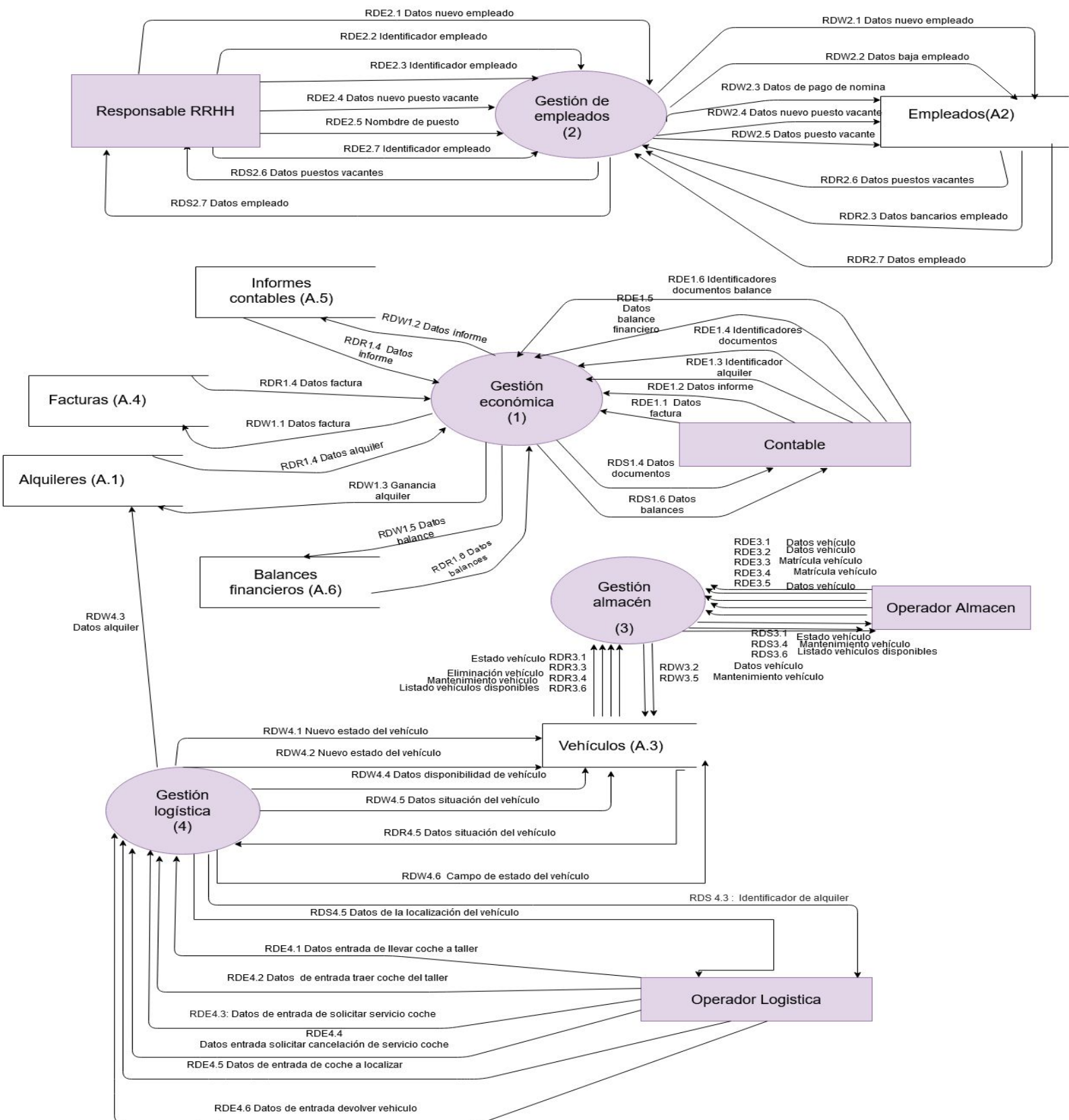


# Diagramas de flujo de datos

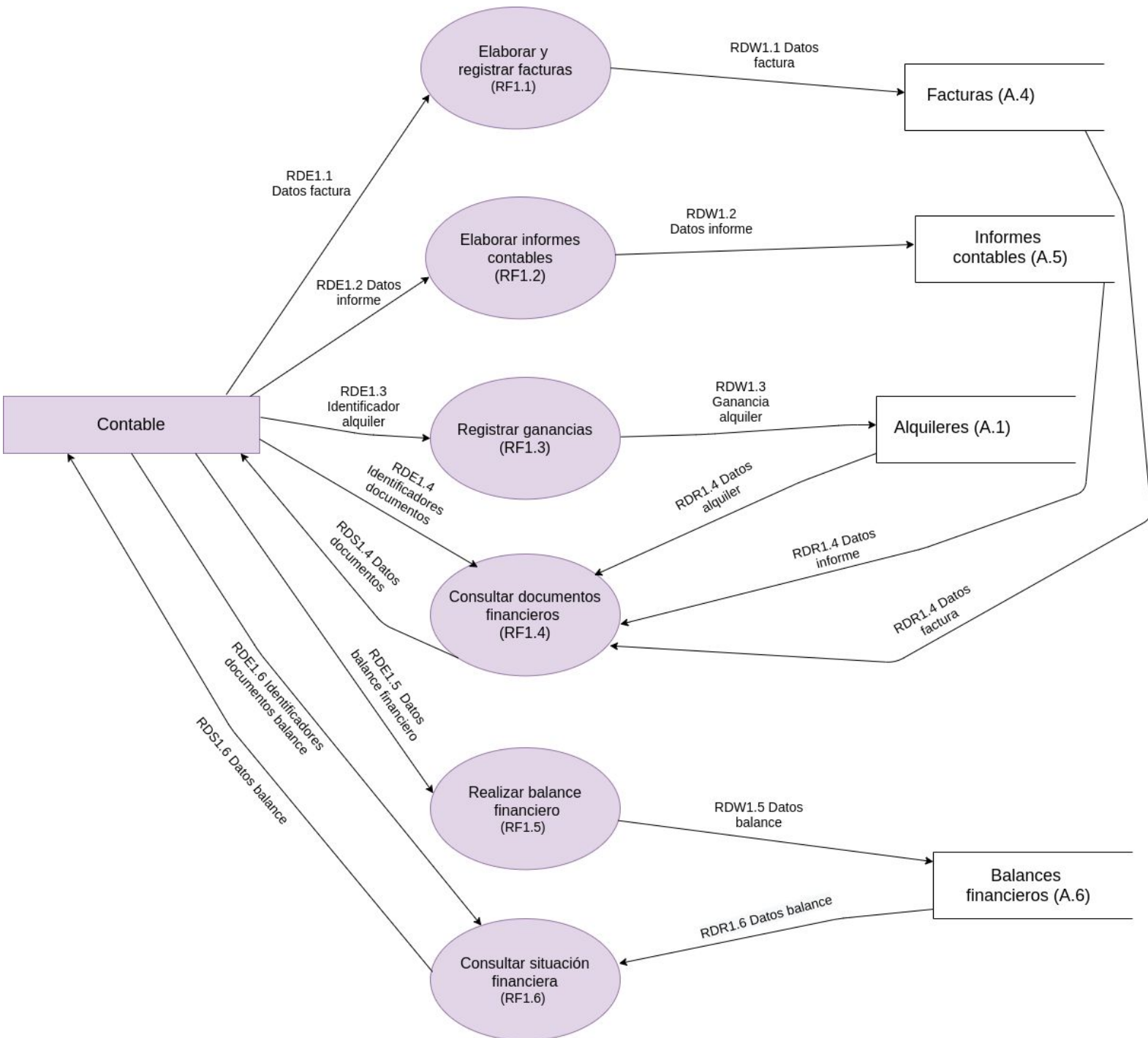
## Esquema caja negra



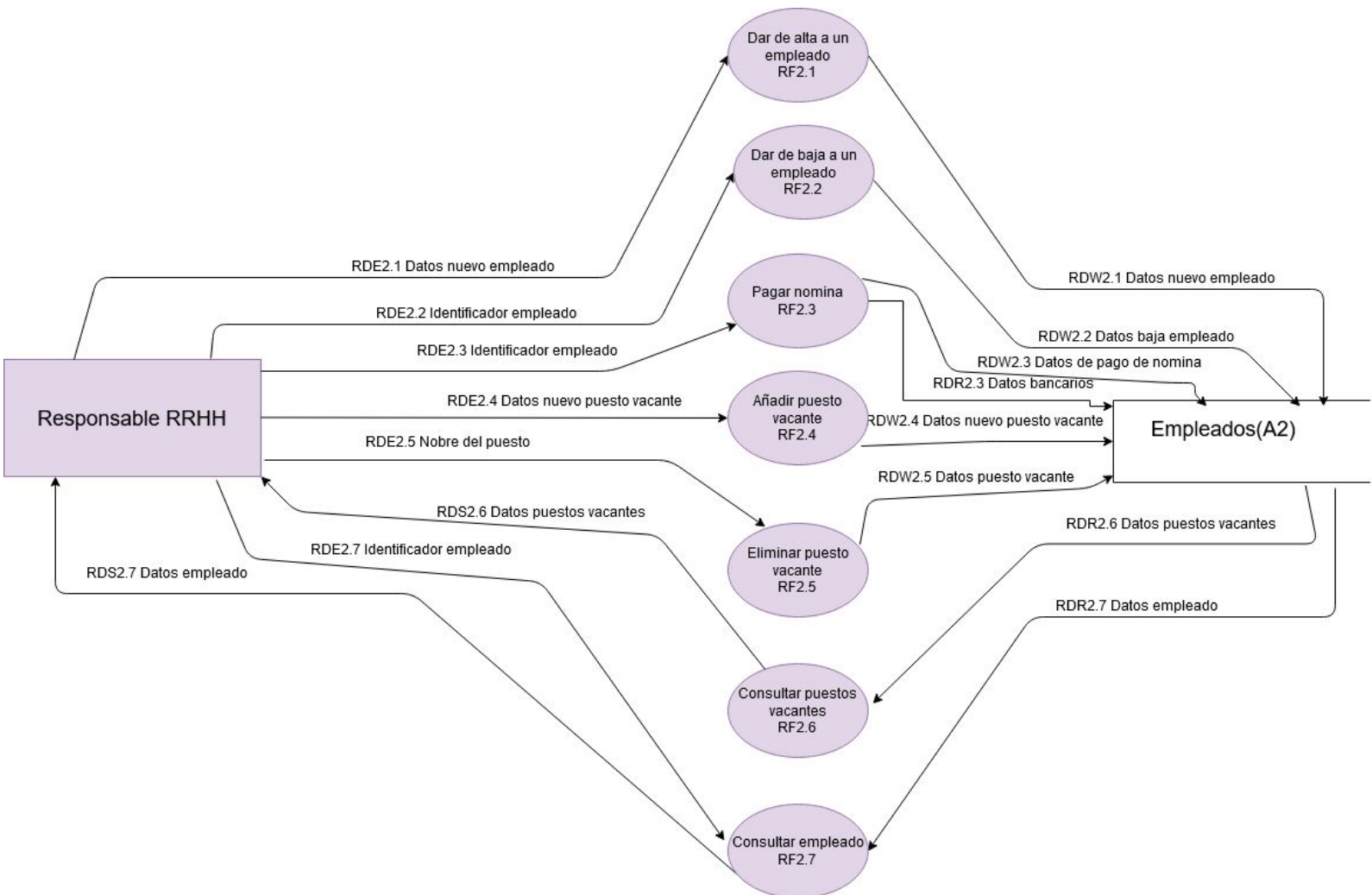
## Esquema almacén



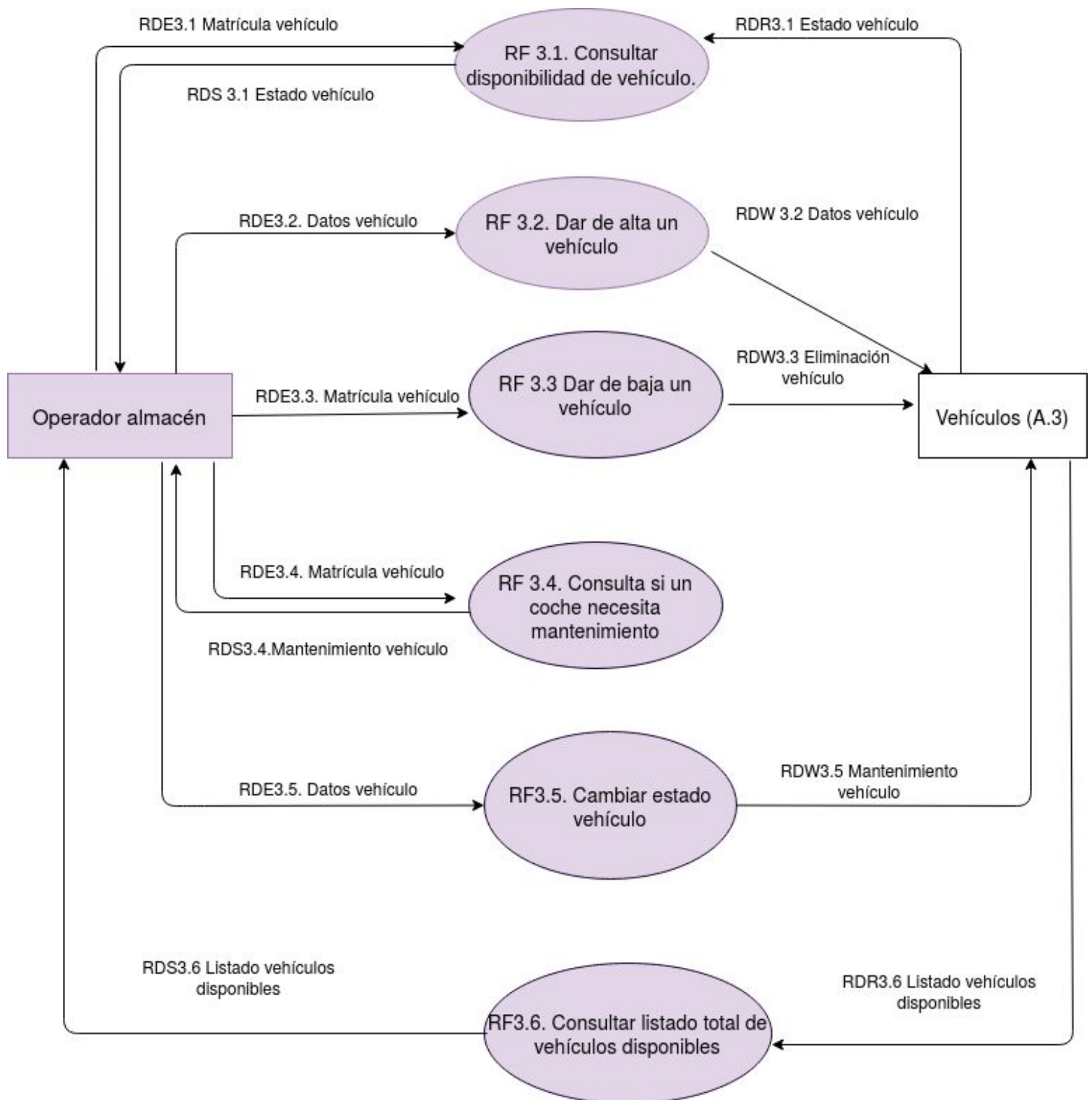
## DFD Contabilidad



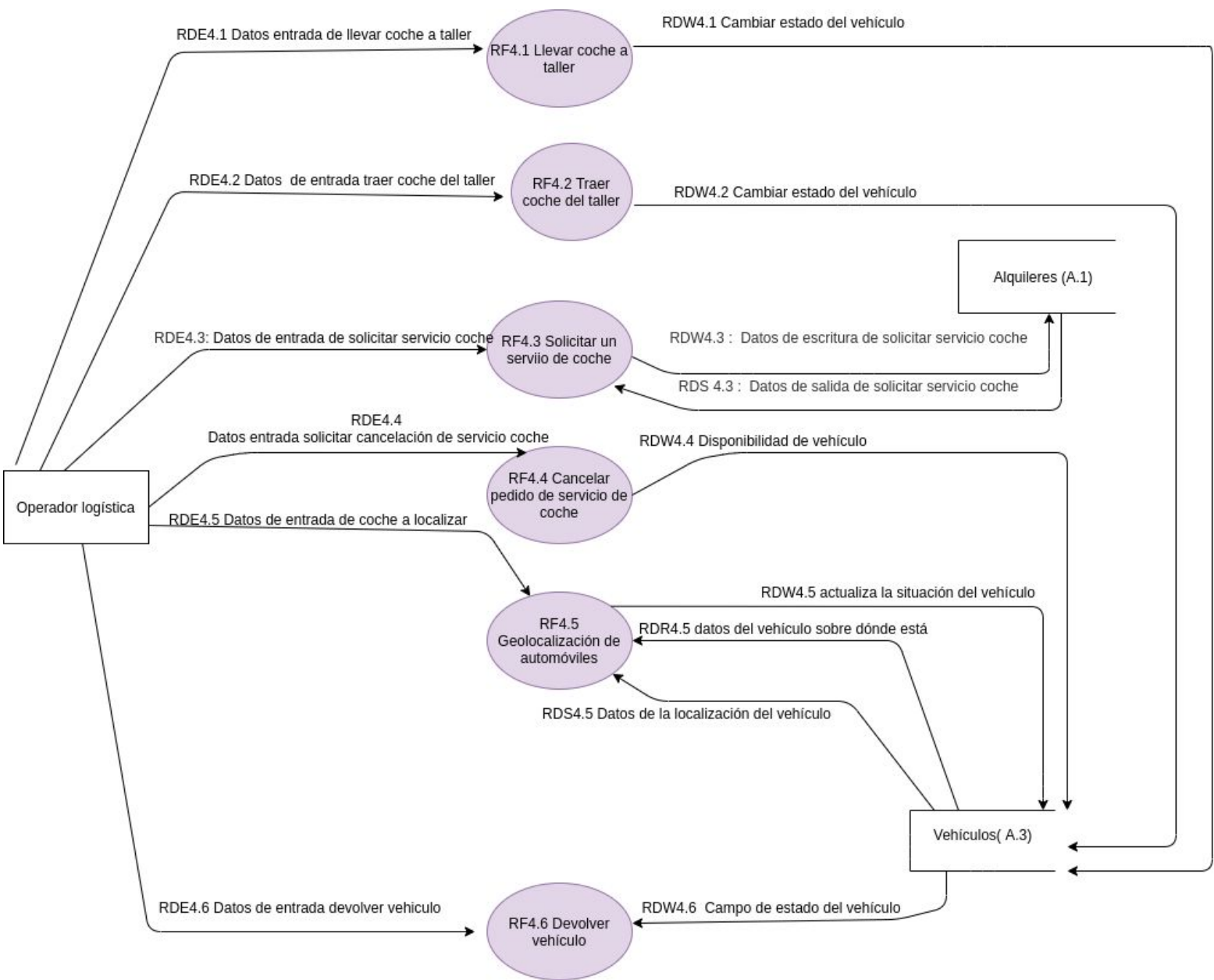
## DFD Recursos humanos



## DFD Almacén



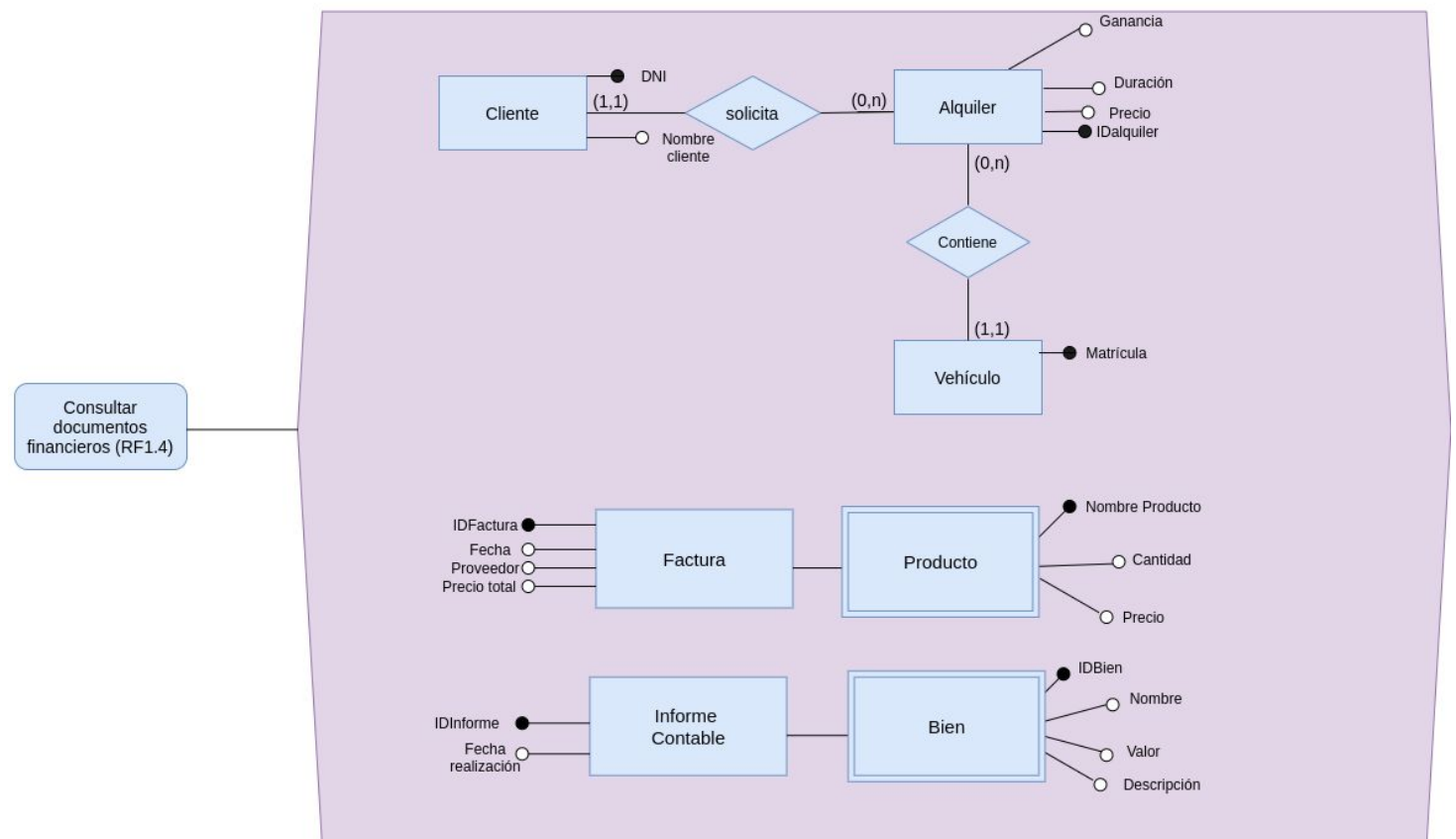
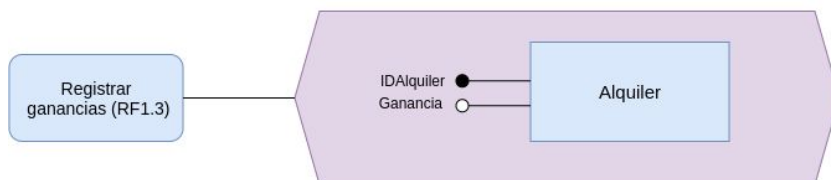
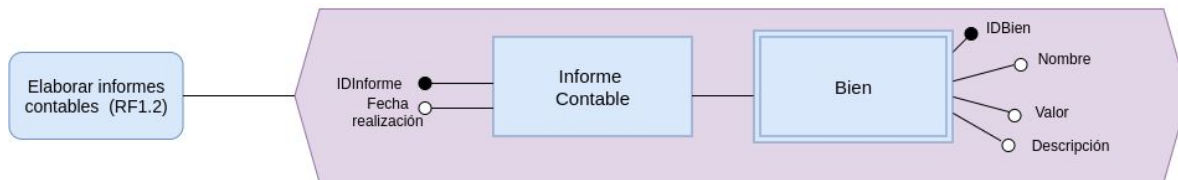
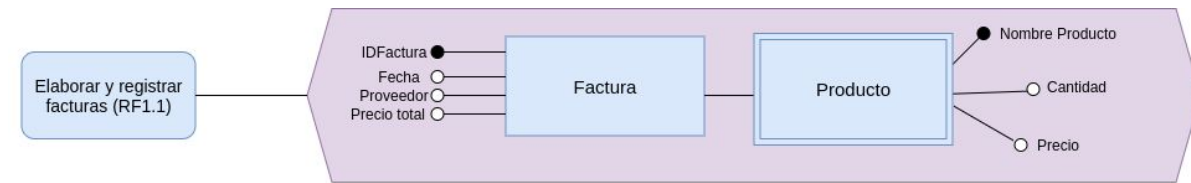
DFD Logística

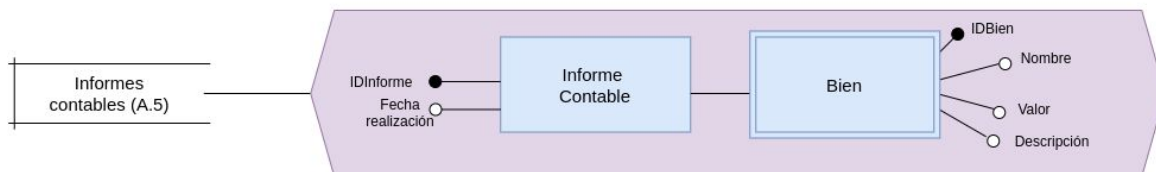
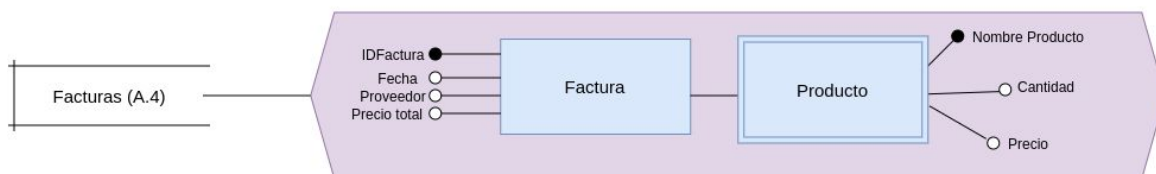
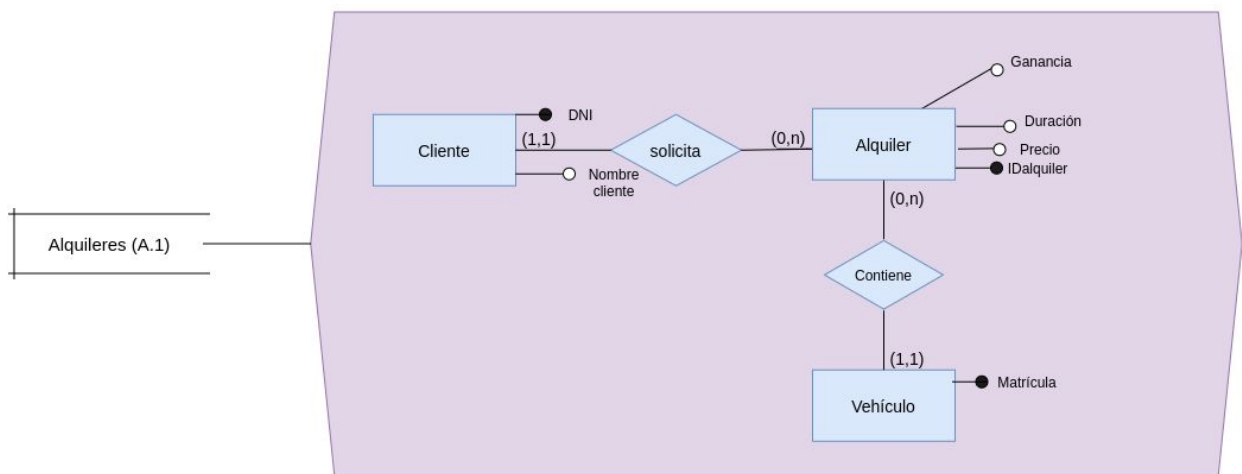
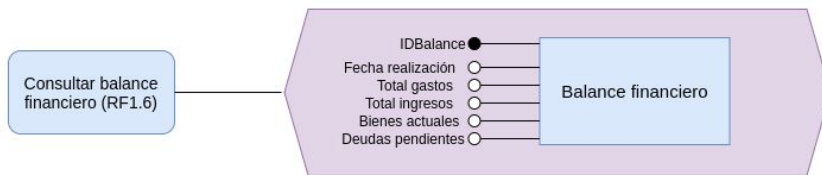




# Diagramas externos

## Contabilidad

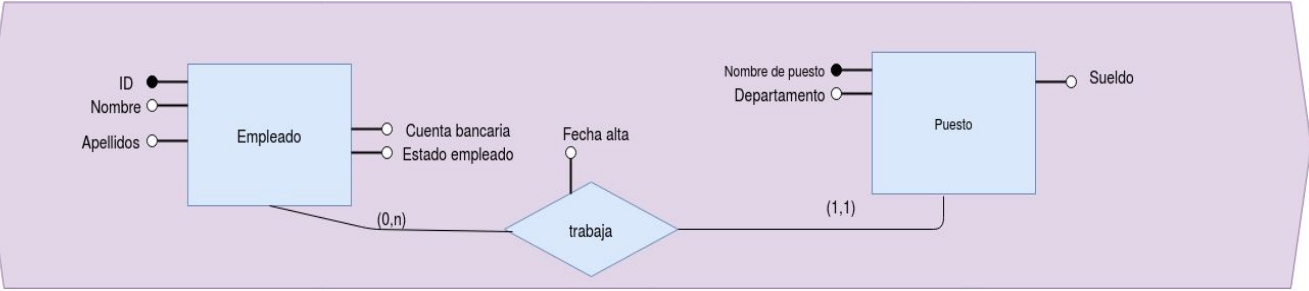




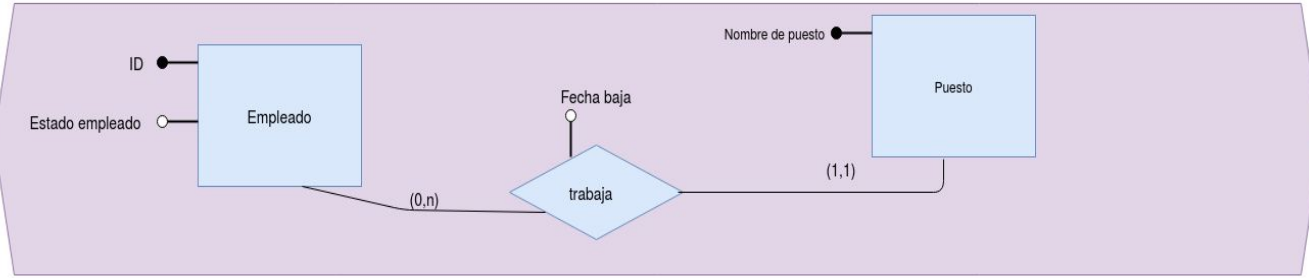


# Recursos humanos

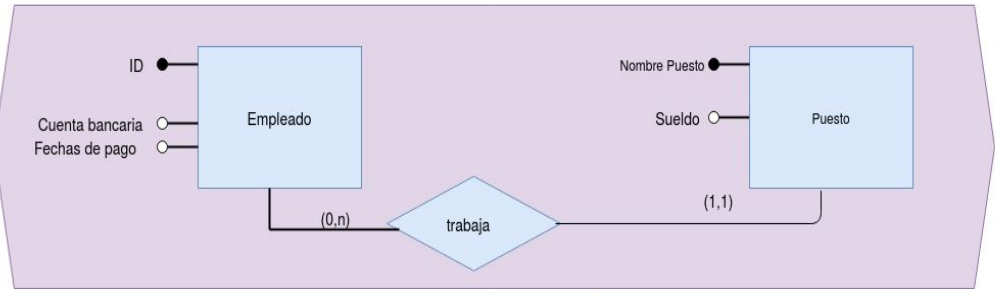
Dar de alta a un empleado  
(2.1)



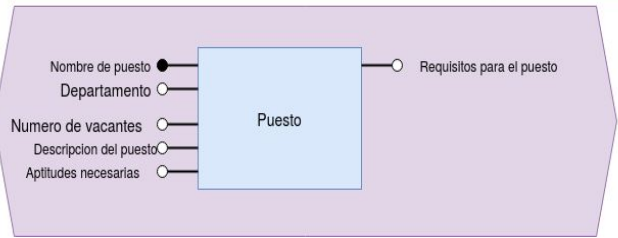
Dar de baja a un empleado  
(2.2)

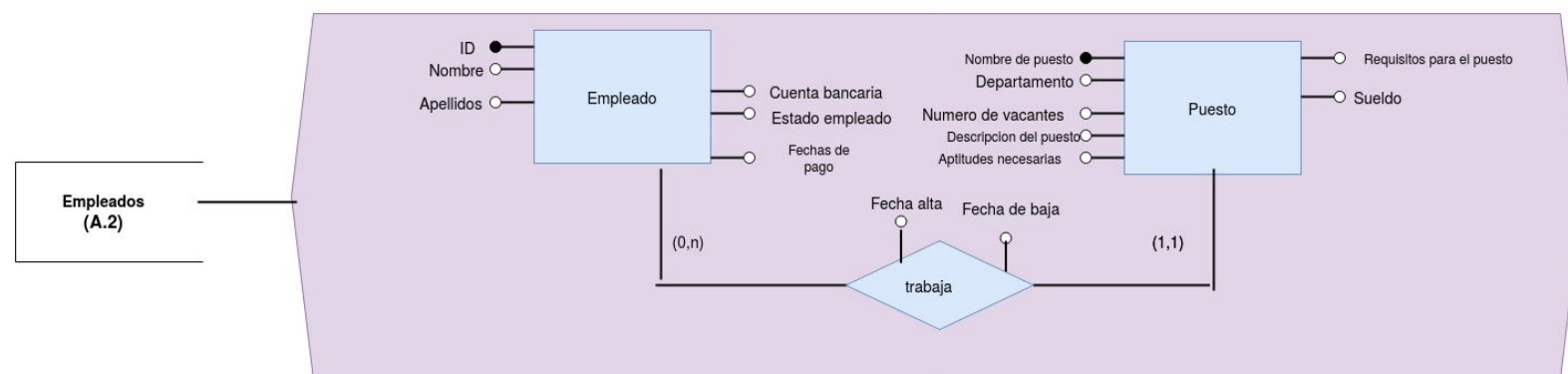
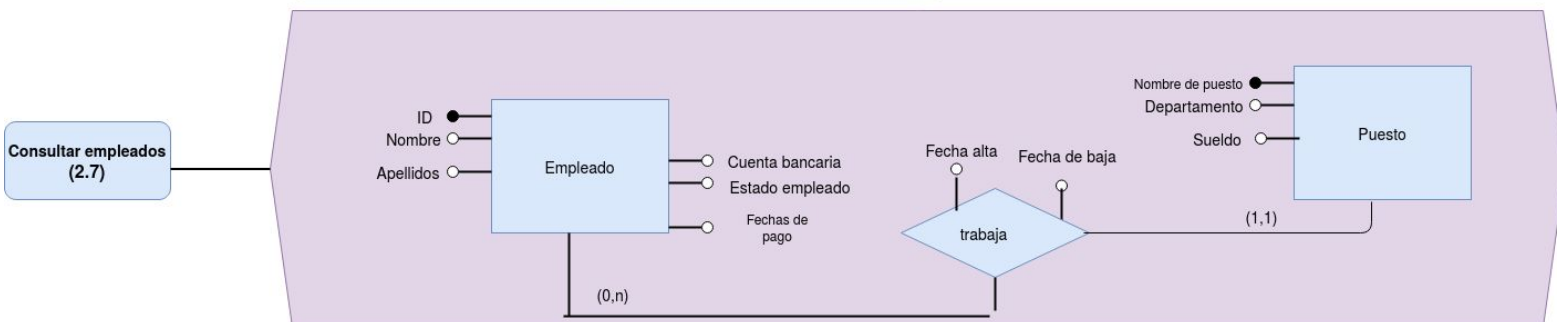
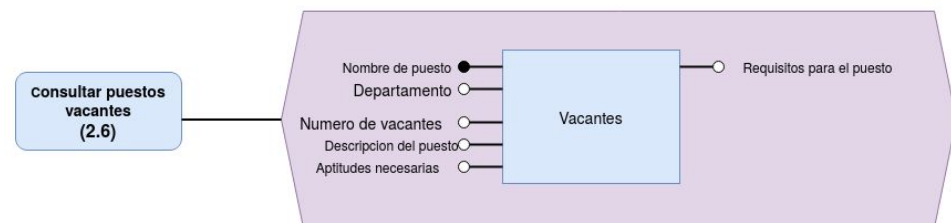
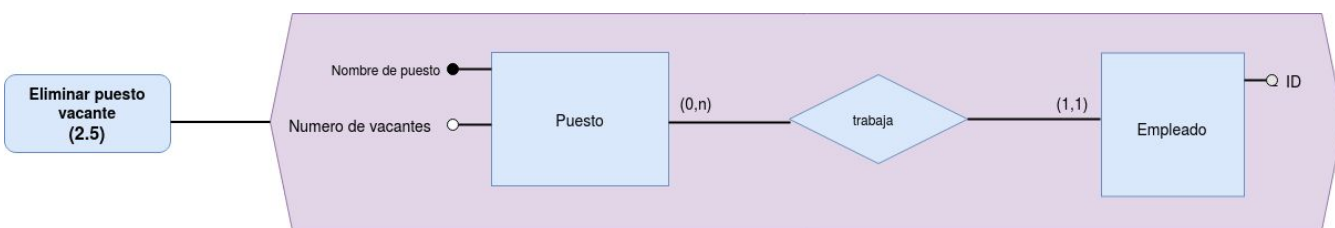


Pagar nomina  
(2.3)

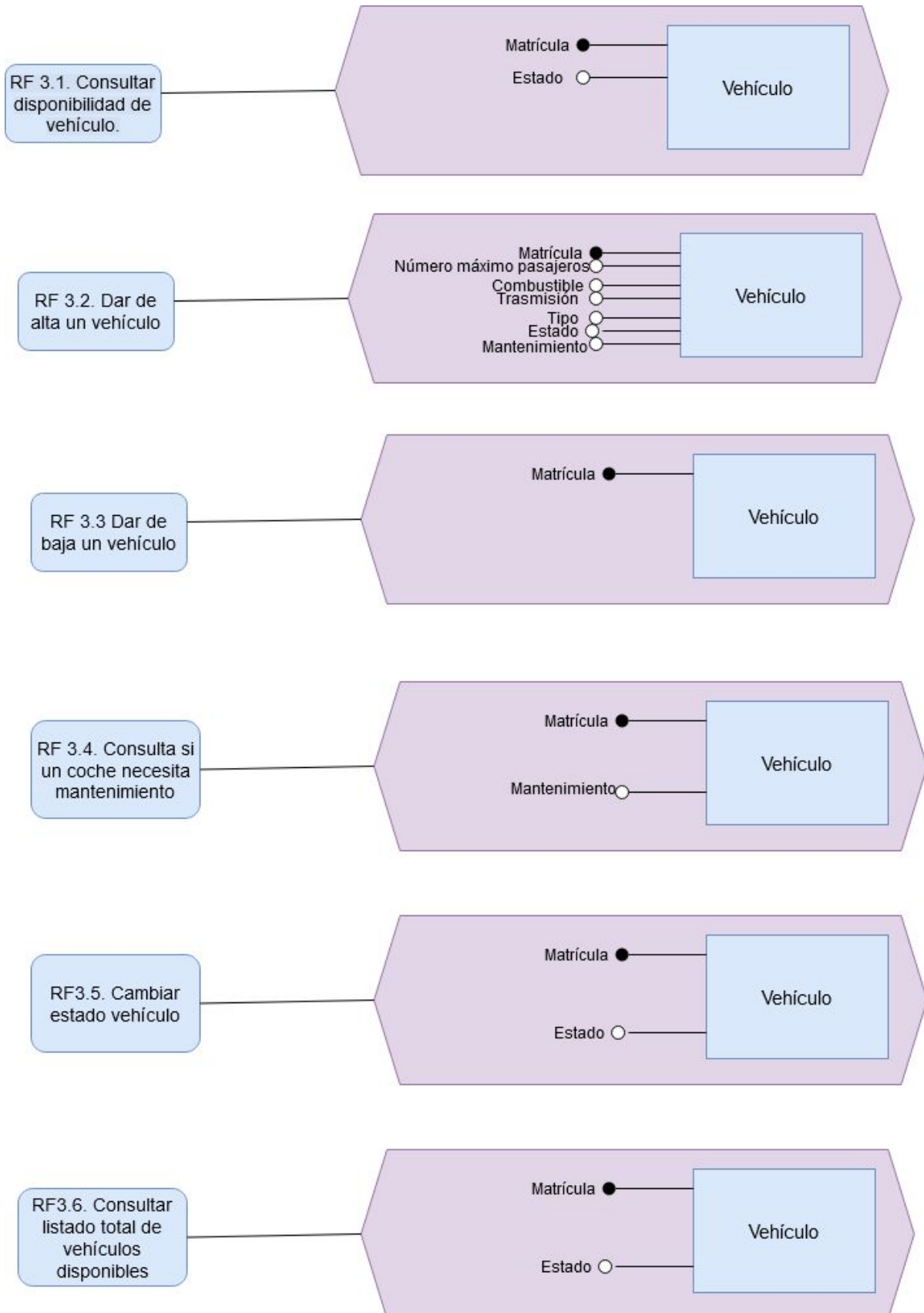


Añadir puesto  
vacante  
(2.4)



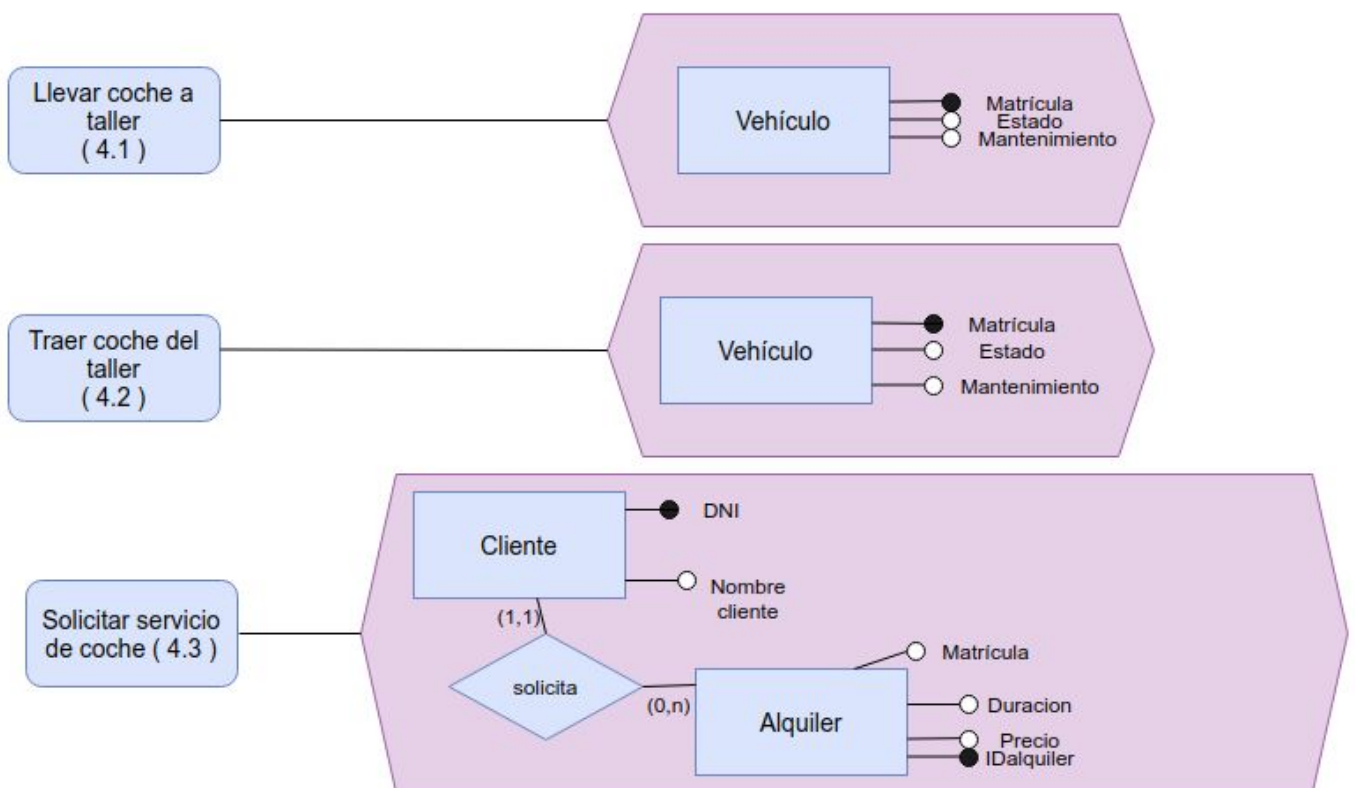


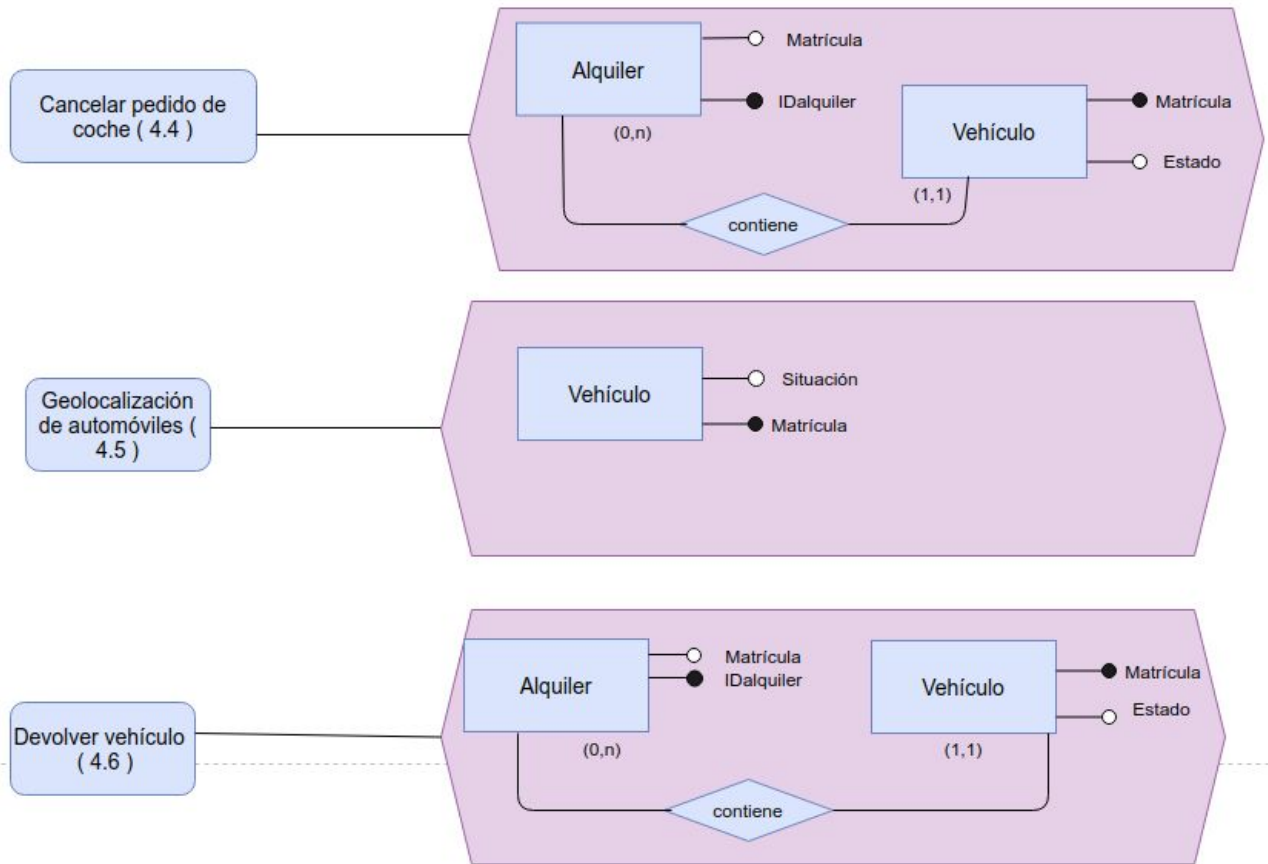
## Almacén



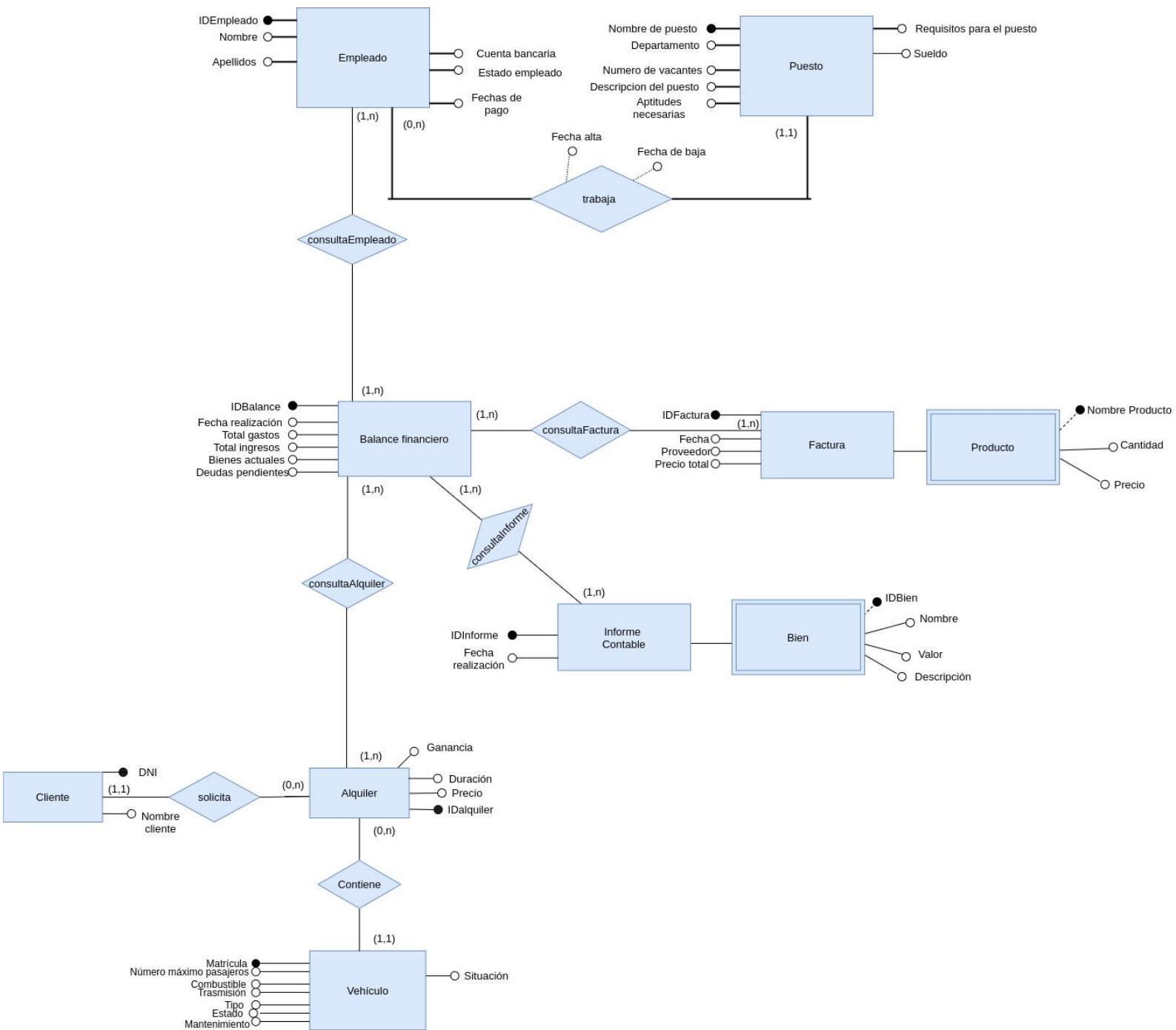


## Logística

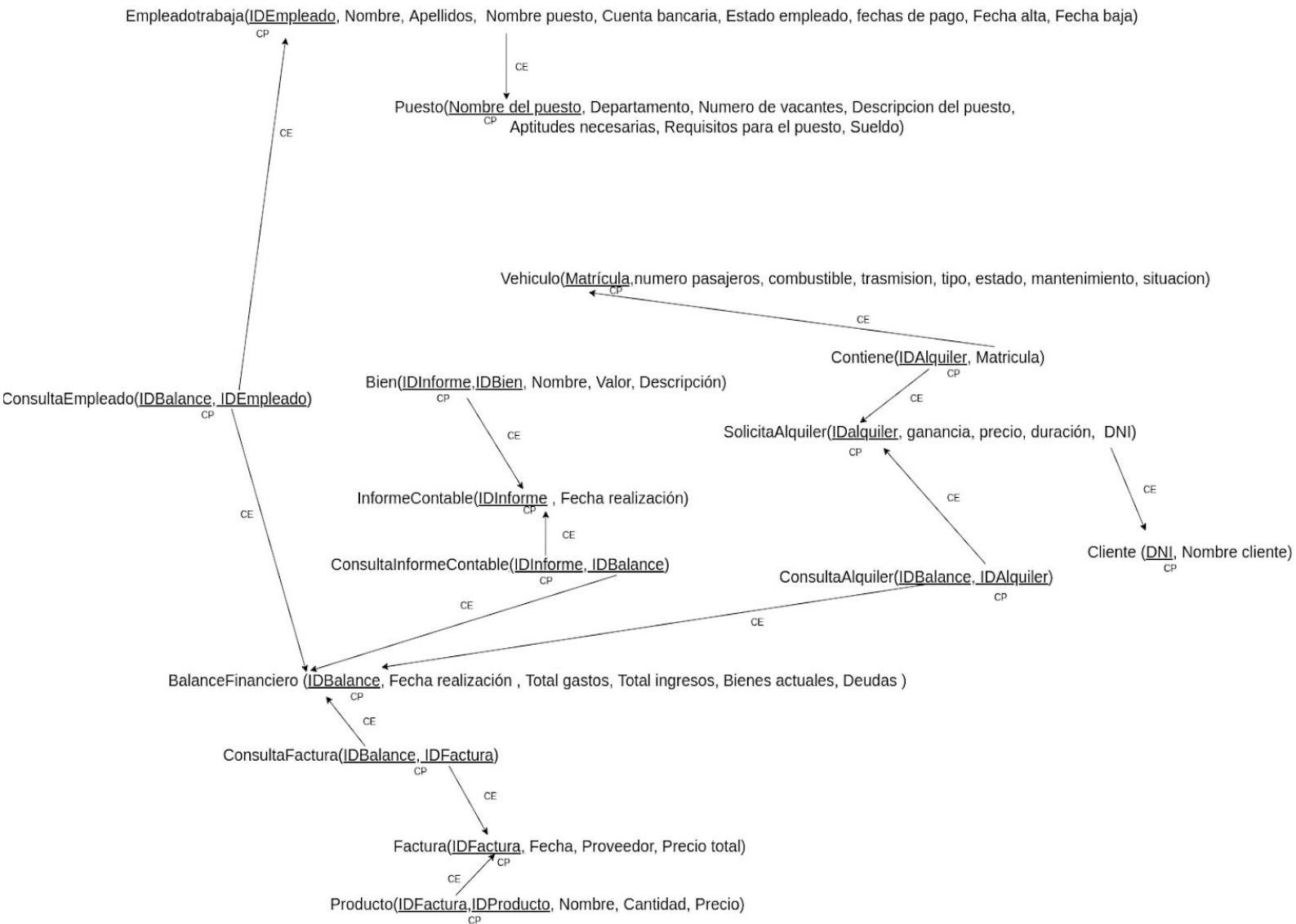




# Diagrama entidad-relación



# Diseño lógico relacional



# Proceso de normalización

## 1. Dependencias funcionales

**Nota:** No hemos incluido las dependencias triviales.

**EmpleadoTrabaja:** {IDEmpleado -> Nombre, Apellidos, Nombre puesto, Cuenta bancaria, Estado empleado, Fechas de pago, Fecha alta, Fecha baja }

(hemos considerado que puede haber coincidencia de dos personas que tengan el mismo nombre y apellido, pero no serían el mismo empleado, de manera que nombre y apellidos no determina al resto de atributos)

**Puesto:** {Nombre de puesto -> Departamento, Número de vacantes, Descripción del puesto, Aptitudes necesarias, Requisitos para el puesto, Sueldo }

**ConsultaEmpleado:** { }

**BalanceFinanciero:** {IDBalance -> Fecha realización, Total gastos, Total ingresos, Bienes actuales, Deudas}

**ConsultaFactura:** { }

**Factura:** {IDFactura-> Fecha, Proveedor, Precio total}

**Producto:** {IDFactura, IDProducto -> Cantidad, Precio}

**Vehículo:** { Matrícula -> número pasajeros, combustible, transmisión, tipo, estado, mantenimiento, situación }

**SolicitaAlquiler**{ IDalquiler -> ganancia, precio, duración, DNI }

**ConsultaAlquiler** { }

**Cliente** { DNI -> Nombre cliente }

**Informe Contable:** {IDInforme -> Fecha realización}

**Bien** { IDInforme, IDBien -> Nombre, Valor, Descripción; IDBien-> Nombre, Descripción }

**ConsultaInformeContable** { }



## 2. Segunda forma normal

Procedemos a la normalización de las relaciones. Vemos que todas las relaciones están en 2FN, puesto que no hay ningún subconjunto de atributos de una clave candidata que determine funcionalmente a un atributo no primo, excepto la relación **Bien** ya que el nombre y la descripción pueden ser determinados por *IDBien*.

Aplicamos el teorema de Heath para normalizar dicha relación y obtenemos las siguientes relaciones con sus correspondientes conjuntos de dependencias funcionales:

Bien\_1(IDInforme, IDBien, Valor)

**F1** = { IDInforme, IDBien -> Valor }

Bien\_2( IDBien , Nombre, Descripción)

**F2** = { IDBien -> Nombre, Descripción }

Parece que hemos perdido una dependencia funcional:

IDInforme, IDBien -> Nombre, Descripción

así que vamos a calcular el cierre de {IDInforme, IDBien} en base al conjunto de dependencias funcionales (F1 U F2) para ver si efectivamente lo hemos perdido:

$\{IDInforme, IDBien\}^+_{(F1 \cup F2)} = \{IDInforme, IDBien, Valor, Nombre, Descripción\}$

Como Nombre y Descripción están en el cierre de atributos, se pueden deducir de ellos y la dependencia no se ha perdido.

## 3. Forma normal de Boyce-Codd

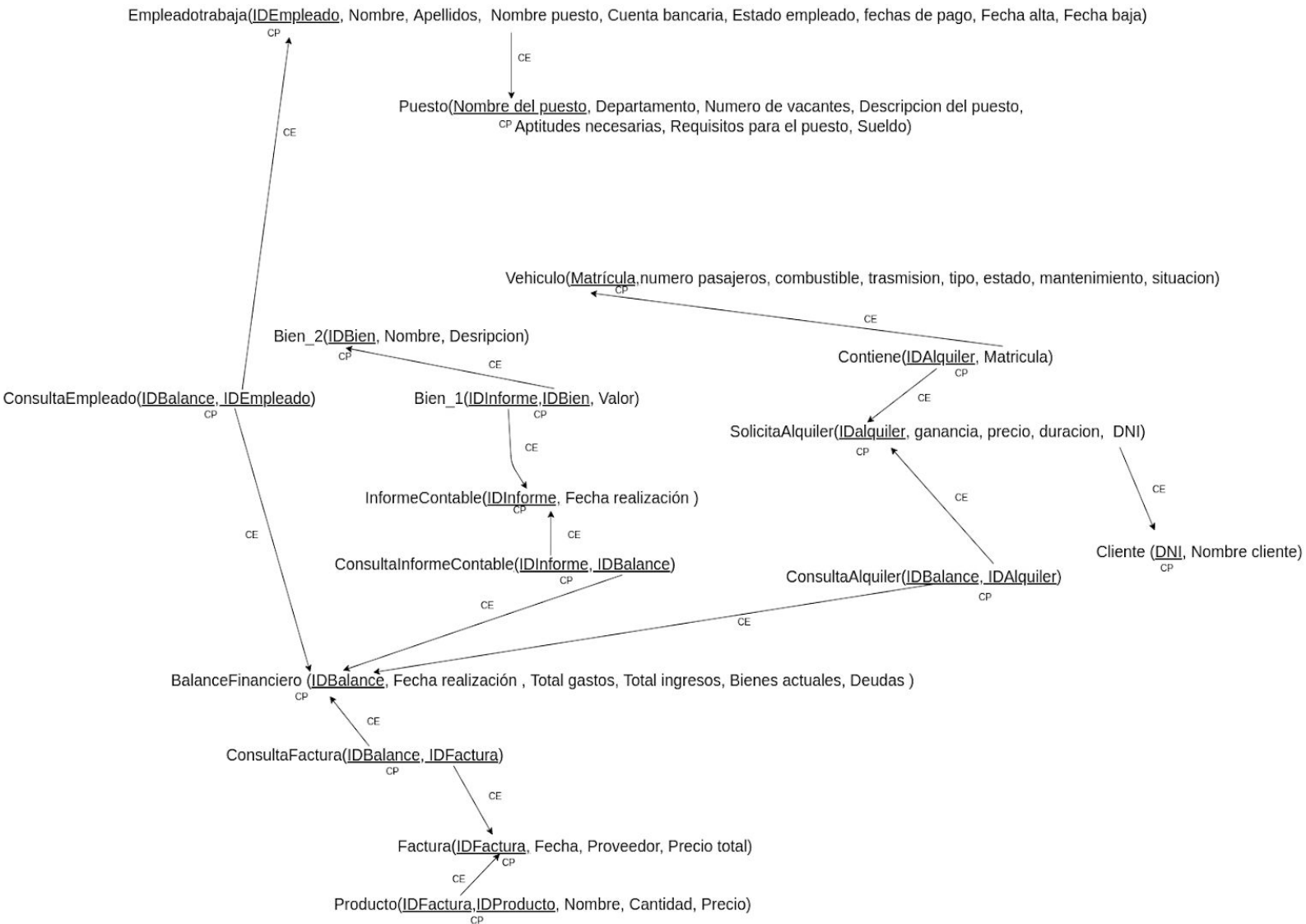
Comprobamos ahora si nuestras relaciones están en FNBC. Todas lo están ya que sus determinantes son todas claves candidatas.

En efecto, *IDBien* es una clave candidata, pues aparece sólo a la izquierda de las dependencias funcionales de F2 y

$\{IDBien\}^+_{F2} = \{IDBien, Nombre, Descripción\}$

contiene a todos los atributos de la relación.

Así, las tablas resultantes tras la normalización son las siguientes:



## Creación de tablas e inserción de tuplas.

### Sentencias de creación de tablas:

```
CREATE TABLE "rentsite_consultainformecontable" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "balance_id" integer NOT NULL REFERENCES "rentsite_balancefinanciero" ("id") DEFERRABLE INITIALLY DEFERRED, "informe_id" varchar(5) NOT NULL REFERENCES "rentsite_informecontable" ("id") DEFERRABLE INITIALLY DEFERRED);
```

```
CREATE TABLE "rentsite_consultafactura" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "balanceFinanciero_id" integer NOT NULL REFERENCES "rentsite_balancefinanciero" ("id") DEFERRABLE INITIALLY DEFERRED, "factura_id" varchar(5) NOT NULL REFERENCES "rentsite_factura" ("id") DEFERRABLE INITIALLY DEFERRED);
```

```
CREATE TABLE "rentsite_consultaalquiler" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "idAlquiler_id" varchar(5) NOT NULL REFERENCES "rentsite_solicitaalquiler" ("idAlquiler") DEFERRABLE INITIALLY DEFERRED, "idBalance_id" integer NOT NULL REFERENCES "rentsite_balancefinanciero" ("id") DEFERRABLE INITIALLY DEFERRED);
```

```
CREATE TABLE "rentsite_puesto" ("nombre_puesto" varchar(40) NOT NULL PRIMARY KEY, "departamento" varchar(40) NOT NULL, "numero_de_vacantes" integer unsigned NOT NULL CHECK ("numero_de_vacantes" >= 0), "aptitudes_necesarias" varchar(250) NULL, "requisitos_puesto" varchar(250) NULL, "sueldo" integer unsigned NOT NULL CHECK ("sueldo" >= 0));
```

```
CREATE TABLE "rentsite_balancefinanciero" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "fecha_realizacion" date NOT NULL, "total_gastos" decimal NOT NULL, "total_ingresos" decimal NOT NULL, "bienes_actuales" decimal NOT NULL, "deudas" decimal NOT NULL, "idBalance" varchar(5) NOT NULL);
```

```
CREATE TABLE "rentsite_cliente" ("nombrecliente" varchar(40) NOT NULL, "dni" varchar(9) NOT NULL PRIMARY KEY);
```

```
CREATE TABLE "rentsite_factura" ("fecha" date NOT NULL, "proveedor" varchar(50) NULL, "total" decimal NOT NULL, "id" varchar(5) NOT NULL PRIMARY KEY);
```

**CREATE TABLE** "rentsite\_informecontable" ("fecha" date NOT NULL, "id" varchar(5) NOT NULL PRIMARY KEY);

**CREATE TABLE** "rentsite\_producto" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre" varchar(10) NOT NULL, "cantidad" integer unsigned NOT NULL CHECK ("cantidad" >= 0), "precio" integer unsigned NOT NULL CHECK ("precio" >= 0), "factura\_id" varchar(5) NOT NULL REFERENCES "rentsite\_factura" ("id") DEFERRABLE INITIALLY DEFERRED, "idProducto" varchar(5) NOT NULL);

**CREATE TABLE** "rentsite\_solicitaalquiler" ("ganancia" decimal NOT NULL, "precio" decimal NOT NULL, "duracion" integer unsigned NOT NULL CHECK ("duracion" >= 0), "cliente\_id" varchar(9) NOT NULL REFERENCES "rentsite\_cliente" ("dni") DEFERRABLE INITIALLY DEFERRED, "idAlquiler" varchar(5) NOT NULL PRIMARY KEY);

**CREATE TABLE** "rentsite\_bien" ("id" varchar(5) NOT NULL PRIMARY KEY, "nombre" varchar(50) NOT NULL, "descripcion" text NULL, "informe\_id" varchar(5) NOT NULL REFERENCES "rentsite\_informecontable" ("id") DEFERRABLE INITIALLY DEFERRED, "valor" decimal NOT NULL);

**CREATE TABLE** "rentsite\_empleadotrabaja" ("nombre" varchar(20) NOT NULL, "apellidos" varchar(20) NOT NULL, "cuenta\_bancaria" varchar(24) NOT NULL, "fecha\_pago" date NULL, "fecha\_alta" date NOT NULL, "fecha\_baja" date NULL, "nombre\_puesto\_id" varchar(40) NOT NULL REFERENCES "rentsite\_puesto" ("nombre\_puesto") DEFERRABLE INITIALLY DEFERRED, "idEmpleado" varchar(9) NOT NULL PRIMARY KEY);

**CREATE TABLE** "rentsite\_consultaempleado" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "balance\_id" integer NOT NULL REFERENCES "rentsite\_balancefinanciero" ("id") DEFERRABLE INITIALLY DEFERRED, "empleado\_id" varchar(9) NOT NULL REFERENCES "rentsite\_empleadotrabaja" ("idEmpleado") DEFERRABLE INITIALLY DEFERRED);

**CREATE TABLE** "rentsite\_vehiculo" ("numero\_pasajeros" integer unsigned NOT NULL CHECK ("numero\_pasajeros" >= 0), "combustible" varchar(30) NOT NULL, "trasmision" varchar(30) NOT NULL, "tipo" varchar(30) NOT NULL, "estado" varchar(30) NOT NULL, "mantenimiento" varchar(30) NOT NULL, "situacion" varchar(100) NOT NULL, "matricula" varchar(7) NOT NULL PRIMARY KEY);

**CREATE TABLE** "rentsite\_contiene" ("alquiler\_id" varchar(5) NOT NULL PRIMARY KEY REFERENCES "rentsite\_solicitaalquiler" ("idAlquiler"))

DEFERRABLE INITIALLY DEFERRED, "vehiculo\_id" varchar(7) NOT NULL  
REFERENCES "rentsite\_vehiculo" ("matricula") DEFERRABLE INITIALLY  
DEFERRED);

### Sentencias de insercción de tuplas:

#### **Vehículos:**

```
insert into vehiculo values('2312QJG', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('6740HNI', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('4260PPS', 5, 'diesel', 'automatico', 'turismo', 'disponible',  
'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('0793DCP', 5, 'diesel', 'automatico', 'turismo', 'disponible',  
'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('2628JVS', 5, 'diesel', 'automatico', 'turismo', 'disponible',  
'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('9452XLH', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('8355ROV', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('8957LEJ', 5, 'diesel', 'automatico', 'turismo', 'disponible',  
'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('0386VGO', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');  
insert into vehiculo values('1039WUH', 5, 'gasolina', 'automatico', 'turismo',  
'disponible', 'no necesario', 'Calle Aguila, Granada, 18002');
```

#### **Productos:**

```
insert into producto values ('F5693', 'P6352', 'Alfombrilla', 5, 30);  
insert into producto values ('F3698', 'P6930', 'Cambio aceite', 6, 50);  
insert into producto values ('F9685', 'P3690', 'Reemplazo bombillas', 8, 100);  
insert into producto values ('F2587', 'P9874', 'Limpieza interior', 8, 125);  
insert into producto values ('F8520', 'P5632', 'Llantas', 2, 123);  
insert into producto values ('F1236', 'P4152', 'Lavado', 1, 9);  
insert into producto values ('F1234', 'P7844', 'Correa distribucion', 1, 630);  
insert into producto values ('F6349', 'P0022', 'Reemplazo frenos', 1, 500);  
insert into producto values ('F8745', 'P2471', 'Pegatina logo parabrisas', 2, 12);  
insert into producto values ('F0025', 'P1111', 'Etiqueta parabrisas', 1, 20);
```

#### **ConsultaFacturas:**

```
insert into consultafactura values('B7410', 'F5693');  
insert into consultafactura values('B8523', 'F1236');  
insert into consultafactura values('B7896', 'F6349');  
insert into consultafactura values('B6547', 'F2587');  
insert into consultafactura values('B3366', 'F0025');
```

### Facturas:

```
insert into factura values ('F5693', to_date('08/12/2020','dd/mm/yyyy'), 'Hermanos Gámiz SL', 30.00);
insert into factura values ('F3698', to_date('08/12/2020','dd/mm/yyyy'), 'Repuestos Granada', 50.00);
insert into factura values ('F9685', to_date('02/12/2020','dd/mm/yyyy'), 'Repuestos Granada', 100.00);
insert into factura values ('F2587', to_date('02/12/2020','dd/mm/yyyy'), 'Limpialo', 125.00);
insert into factura values ('F8520', to_date('25/11/2020','dd/mm/yyyy'), 'Repuestos Granada', 123.00);
insert into factura values ('F1236', to_date('10/11/2020','dd/mm/yyyy'), 'Limpialo', 9.00);
insert into factura values ('F1234', to_date('04/11/2020','dd/mm/yyyy'), 'Repuestos Granada', 630.00);
insert into factura values ('F6349', to_date('04/11/2020','dd/mm/yyyy'), 'Desguaces Granada', 500.00);
insert into factura values ('F8745', to_date('30/11/2020','dd/mm/yyyy'), 'Hermanos Gámiz SL', 12.00);
insert into factura values ('F0025', to_date('22/11/2020','dd/mm/yyyy'), 'Hermanos Gámiz SL', 20.00);
```

### Bienes:

```
insert into bien values ('W4578', 'Cochera', 'Cochera en Granada ciudad', 'I8765', 120000.00);
insert into bien values ('W8742', 'Cochera', 'Cochera en Albolote', 'I6513', 100000.00);
insert into bien values ('W6572', 'Cochera', 'Cochera en Loja', 'I7845', 90000.00);
insert into bien values ('W1245', 'Oficina', 'Oficina en Granada', 'I2584', 70000.00);
insert into bien values ('W1847', 'Oficina', 'Oficina en Loja', 'I1478', 50000.00);
insert into bien values ('W4976', 'Taller', 'Taller de reparación de vehículos en Granada', 'I4851', 150000.00);
insert into bien values ('W5735', 'Taller', 'Taller de reparación de vehículos en Loja', 'I1753', 130000.00);
insert into bien values ('W3765', 'Vehículo', 'Turismo con matrícula 2312QJG', 'I3895', 20000.00);
insert into bien values ('W3857', 'Vehículo', 'Turismo con matrícula 1039WUH', 'I6789', 15000.00);
insert into bien values ('W3135', 'Vehículo', 'Turismo con matrícula 8355ROV', 'I7596', 40000.00);
```

### ConsultainformeContable:

```
insert into consultainformecontable values ('I8765', 'B7410');
insert into consultainformecontable values ('I6513', 'B8523');
insert into consultainformecontable values ('I7845', 'B7896');
insert into consultainformecontable values ('I2584', 'B6547');
insert into consultainformecontable values ('I1478', 'B8412');
```

### Balance Financiero:

```
INSERT INTO BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
'B7410',
'to_date("25/11/2020","dd/mm/yyyy")',
'48123',
'48129',
'21323',
'4123'
);
```

```
INSERT INTO BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
'B8523",
'to_date("26/11/2020","dd/mm/yyyy")',
'423423',
'1329',
'2121353',
'41234'
);
```

```
INSERT INTO BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
'B7896',
'to_date("27/11/2020","dd/mm/yyyy")',
'423',
'489',
'223',
'4123'
);
```

```
INSERT INTO BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
'B6547',
'to_date("28/11/2020","dd/mm/yyyy")',
'4223',
'54129',
'254323',
'223'
);
```

```

INSERT                                INTO                                BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
    'B8412',
    'to_date("30/11/2020","dd/mm/yyyy")',
    '4233',
    '4859',
    '6323',
    '4323'
);

```

```

INSERT                                INTO                                BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
    'B3366",
    'to_date("24/11/2020","dd/mm/yyyy")',
    '5123',
    '1259',
    '2223',
    '4163'
);

```

```

INSERT                                INTO                                BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
    'B5410',
    'to_date("28/11/2020","dd/mm/yyyy")',
    '43253',
    '48959',
    '2623',
    '4436"
);

```

```

INSERT                                INTO                                BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
    'B7453',
    'to_date("25/11/2020","dd/mm/yyyy")',
    '4723',
    '5129',
    '2823',
    '8123'
);

```

```

INSERT                                INTO                                BalanceFinanciero
(IDBalance,Fecharealización,Totalgastos,Totalingresos,Bienesactuales,Deudas)
VALUES (
    'B7481',

```



```
'to_date("20/11/2020","dd/mm/yyyy"),',
'48723',
'48659',
'6323',
'4423'
);
```

```
INSERT INTO BalanceFinanciero
(IDBalance, Fecharealización, Totalgastos, Totalingresos, Bienesactuales, Deudas)
VALUES (
'B7520',
'to_date("26/11/2020","dd/mm/yyyy"),',
'421423',
'4629',
'53323',
'5123'
);
```

### EmpleadoTrabaja:

```
INSERT INTO EmpleadoTrabaja
(IDEmpleado, Nombre, Apellidos, Nombre_puesto, Cuenta_Bancaria, Fecha_alta)
VALUES (
'672872735D',
'Jose',
'Lopez',
'Jefe de marketing',
'894128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
```

```
INSERT INTO EmpleadoTrabaja
(IDEmpleado, Nombre, Apellidos, Nombre_puesto, Cuenta_Bancaria, Fecha_alta)
VALUES (
'672872325R',
'Manuel',
'Dominguez',
'Asesor de marketing',
'4372128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
```

```
INSERT INTO EmpleadoTrabaja
(IDEmpleado, Nombre, Apellidos, Nombre_puesto, Cuenta_Bancaria, Fecha_alta)
VALUES (
'543872735K',
'Angela',
```

```

'Torres',
'Asesor de marketing',
'432128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'4928432735F',
'Manuela',
'Perez',
'Mecanico ',
'342128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'322872735R',
'Pablo',
'Smith',
'Mecanico',
'434128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'412872735T',
'Patricia',
'Lopez',
'Teleoperador',
'844128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'672982735Z',
'Pilar',
'Picón',
'',
'894128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);

```

```

INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'674372735P',
'Rocio',
'Lopez',
'Informatico web',
'456128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'672872735D',
'Victor',
'Alvarez',
'Jefe de mantenimiento de coches',
'894128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'673872735Y',
'Francisco',
'Cyrus',
'Jefe de atencion al cliente',
'894128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);
INSERT INTO EmpleadoTrabaja
(IDEmpleado,Nombre,Apellidos,Nombre_puesto,Cuenta_Bancaria,Fecha_alta)
VALUES (
'672877835K',
'Mario',
'Serrano',
'Jefe de logistica',
'894128091283918909',
'to_date("10/11/2020","dd/mm/yyyy"),'
);

```

### Puestos:

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Jefe de marketing',
'Marketing',

```

```

'0',
'Encargado de hacer marketing de la empresa',
",
",
'Grado en Marketing o finanzas',
'2200'
);
INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Asesor de marketing',
'Marketing',
'1',
'Encargado de hacer marketing de la empresa',
",
",
'Grado en Marketing o finanzas',
'2000'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Mecanico',
'Logistica",
'0',
'Encargado de hacer arreglar cosas de los coches',
",
",
'FP mecanica o Grado universitario',
'2200'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Informatico web',
'Atencion cliente",
'2',
'Encargado de mantener la pagina web',
",
",
'Grado en Marketing o finanzas',
'2200'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Teleoperador',
'Atencion cliente',
'0',

```

```

'Encargado de atender llamadas',
",
'Grado universitario o FP',
'2200'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Inspector de daños',
'Logística',
'0',
'Encargado de inspeccionar daños de los coches',
",
'Grado universitario o FP',
'2200'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Contable',
'Contabilidad',
'0',
'Encargado de contabilidad',
",
'Grado universitario o FP',
'2200'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Jefe de atención al cliente',
'Atención cliente',
'0',
'Encargado de supervisar la atención al cliente',
",
'Grado universitario o FP',
'2300'
);

```

```

INSERT INTO Puesto
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Jefe de logística',
'Logística',
'0',
'Encargado de hacer la logística',
",
'Grado universitario o FP',

```

```

'2200'
);
INSERT INTO
(Nombre_puesto,Departamento,Numero_de_vacantes,Descripcion_puesto,Aptitud
es_necesarias,Requisitos_puesto,Sueldo) VALUES (
'Jefe de mantenimiento de coches',
'Logistica',
'0',
'Encargado de mantener los coches y buscar nuevos"',
",
'Grado universitario o FP',
'2200'
);

```

### ConsultaEmpleado:

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B7410',
'672872735D'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B8523',
'672872325R'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B7896',
'543872735K'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B6547',
'4928432735F'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B8412',
'322872735R'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B3366"',
'412872735T'
);

```

```

INSERT INTO ConsultaEmpleado (IDBalance,IDEpleado) VALUES (
'B5410',

```

```
'674372735P'  
);
```

```
INSERT INTO ConsultaEmpleado (IDBalance,IDEmpleado) VALUES (  
'B7453',  
'672982735Z'  
);
```

```
INSERT INTO ConsultaEmpleado (IDBalance,IDEmpleado) VALUES (  
'B7481',  
'672872735D'  
);
```

```
INSERT INTO ConsultaEmpleado (IDBalance,IDEmpleado) VALUES (  
'B7520',  
'673872735Y'  
);
```

### Solicita Alquiler:

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A5963' , '1981','2829', '3' , '24628498W');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A6358' , '197','2829', '2' , '24692498Q')
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A9864' , '881','1229', '1' , '24692326M');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A7852' , '981','3829', '3' , '24628494W');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A0036' , '191','289', '2' , '24628494W');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A3463' , '191','2829', '2' , '25628498W');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A1358' , '311','529', '3' , '24592498Q')
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A1864' , '981','1229', '4' , '20692325M');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A3352' , '111','2829', '3' , '54628494T');
```

```
INSERT INTO SolicitaAlquiler( IDalquiler, ganancia , precio, duracion, dni  
)VALUES( 'A0936' , '281','2829', '2' , '84628494Y');
```

### ConsultaAlquiler:

```
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B8523','A5963');
```

```
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B7896','A6358');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B6547','A9864');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B6547','A7852');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B8412','A0036');
```

```
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B7410','A3463');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B7520','A1358');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B7481','A1864');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B5410','A3352');
INSERT INTO ConsultaAlquiler(IDBalance , IDAlquiler) VALUES ('B3366','A0936');
```

### Cientes:

```
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24628498W','Jose García');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24692498Q','María García');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24692326M','María Morales');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24628494W','Federico
Ramirez');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24628432T','Federico
Ramirez');
```

```
INSERT INTO Cliente(dni , Nombrecliente )VALUES('25628498W','Luis Ter');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('24592498Q','Marina Mar');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('20692325M','Marta Morales');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('54628494T','Fernando
Ramirez');
INSERT INTO Cliente(dni , Nombrecliente )VALUES('84628494Y','Juan Ramirez');
```

### Informes Contables:

```
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'18765',to_date('08/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'16513',to_date('08/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'17845',to_date('18/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'12584',to_date('18/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'11478',to_date('18/12/2020','dd/mm/yyyy'));
```



```

INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'I1765',to_date('03/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'I2513',to_date('08/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'I3845',to_date('01/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'I4584',to_date('06/12/2020','dd/mm/yyyy'));
INSERT INTO InformeContable(IDInforme , Fecharealizacion )VALUES(
'I5478',to_date('19/12/2020','dd/mm/yyyy'));

```

Veamos ahora los valores que contienen nuestras tablas finales:

```

sqlite> select * from rentsite_balancefinanciero;
3|2020-11-25|48123|48129|21323|4123|B7410
4|2020-11-26|423423|1329|2121353|41234|B8523
5|2020-11-27|423|489|223|4123|B7896
6|2020-11-28|4223|54129|254323|223|B6547
7|2020-11-30|4233|4859|6323|4323|B8412
8|2020-11-24|5123|1259|2223|4163|B3366
9|2020-11-28|43253|48959|2623|4436|B5410
10|2020-11-25|4723|5129|2823|8123|B7453
11|2020-11-20|48723|48659|6323|4423|B7481
12|2020-11-26|421423|4629|53323|5123|B7520

```

```

sqlite> select * from rentsite_bien;
W4578|Cochera|Cochera en Granada ciudad|I8765|120000
W8742|Cochera|Cochera en Albolote|I6513|100000
W6572|Cochera|Cochera en Loja|I7845|90000
W1245|Oficina|Oficina en Granada|I2584|70000
W1847|Oficina|Oficina en Loja|I1478|50000
W4976|Taller|Taller de reparación de vehículos en Granada|I2513|150000
W5735|Taller|Taller de reparación de vehículos en Loja|I1765|130000
W3765|Vehículo|turismo con matrícula 8355ROV|I3845|20000
W3857|Vehículo|turismo con matrícula 1039WUH|I4584|15000
W3135|Vehículo|turismo con matrícula 8355ROV|I5478|40000

```

```

sqlite> select * from rentsite_cliente;
Jose García|24628498W
María García|24692498Q
María Morales|24692326M
Federico Ramirez|24628494W
Federico Ramirez|24628432T
Luis Ter|25628498W
Marina Mar|24592498Q
Marta Morales|20692325M
Fernando Ramirez|54628494T
Juan Ramirez|84628494Y

```

```
sqlite> select * from rentsite_consultaalquiler;
2|A5963|4
3|A6358|5
4|A9864|6
5|A7852|6
6|A0036|7
7|A3463|3
8|A1358|12
9|A1864|11
10|A3352|9
11|A0936|8
```

```
sqlite> select * from rentsite_consultaempleado;
1|3|672872735D
2|4|672872325R
3|5|543872735K
4|6|492843273F
5|7|322872735R
6|8|412872735T
7|9|672982735Z
8|11|672872734D
9|12|673872735Y
```

```
sqlite> select * from rentsite_consultafactura;
4|3|F5693
5|4|F1236
6|5|F6349
7|6|F2587
8|8|F0025
```

```
sqlite> select * from rentsite_consultainformecontable;
3|3|I8765
4|4|I6513
5|5|I7845
6|6|I2584
7|7|I1478
```

```
sqlite> select * from rentsite_contiene;
A5963|1039WUH
A6358|8957LEJ
A1864|1039WUH
A7852|8957LEJ
```

```
sqlite> select * from rentsite_empleadotrabaja;
Jose|Lopez|894128091283918909||2020-11-10||Jefe de marketing|672872735D
Manuel|Dominguez|4372128091283918909||2020-11-10||Asesor de marketing|672872325R
Angela|Torres|432128091283918909||2020-11-10||Asesor de marketing|543872735K
Manuela|Perez|342128091283918909||2020-11-10||Mecanico|492843273F
Pablo|Smith|434128091283918909||2020-11-10||Mecanico|322872735R
Patricia|Lopez|844128091283918909||2020-11-10||Teleoperador|412872735T
Pilar|Picón|894128091283918909||2020-11-10||Contable|672982735Z
Rocio|Lopez|456128091283918909||2020-11-10||Informatico web|674372735P
Victor|Alvarez|894128091283918909||2020-11-10||Jefe de mantenimiento de coches|672872734D
Francisco|Cyrus|894128091283918909||2020-11-10||Jefe de atencion al cliente|673872735Y
Mario|Serrano|894128091283918909||2020-11-10||Jefe de logistica|672877835K
```

```
sqlite> select * from rentsite_factura;
2020-12-08|Hermanos Gámiz SL|30|F5693
2020-12-08|Repuestos Granada|50|F3698
2020-12-02|Repuestos Granada|100|F9685
2020-12-02|Limpialo|125|F2587
2020-11-25|Repuestos Granada|123|F8520
2020-11-10|Limpialo|9|F1236
2020-11-04|Repuestos Granada|630|F1234
2020-11-04|Desguaces Granada|500|F6349
2020-11-30|Hermanos Gámiz SL|12|F8745
2020-11-22|Hermanos Gámiz SL|20|F0025
```

```
sqlite> select * from rentsite_informecontable;
2020-12-08|I8765
2020-12-08|I6513
2020-06-04|I7845
1998-08-21|I2584
2000-08-01|I1478
2019-07-10|I1765
2018-10-09|I2513
2012-03-23|I3845
2020-12-06|I4584
2013-12-19|I5478
```

```
sqlite> select * from rentsite_producto;
3|P6352|5|30|F5693|Alfombrill
4|P6930|6|50|F3698|Cambio aceite
5|P3690|8|100|F9685|Reemplazo bombillas
6|P9874|8|125|F2587|Limpieza interior
7|P5632|2|123|F8520|Llantas
8|P4152|1|9|F1236|Lavado
9|P7844|1|630|F1234|Correa distribucion
10|P0022|1|500|F6349|Reemplazo frenos
11|P2471|1|12|F8745|Pegatina logo parabrisas
12|P1111|1|20|F0025|Etiqueta parabrisas
```

```
sqlite> select * from rentsite_puesto;
Jefe de marketing|Marketing|0||'Grado en Marketing o finanzas'|2200|Encargado de hacer marketing de la empresa
Asesor de marketing|Marketing|1||Grado en Marketing o finanzas|2000|Encargado de hacer marketing de la empresa
Mecanico|Logistica|0||FP mecanica o Grado universitario|2200|Encargado de hacer arreglar cosas de los coches
Informatico web|Atencion cliente|2||Grado en Marketing o finanzas|2200|Encargado de mantener la pagina web
Teleoperador|Atencion cliente|0||Grado universitario o FP|2200|Encargado de atender llamadas
Inspector de daños|Logistica|0||Grado universitario o FP|2200|Encargado de inspeccionar daños de los coches
Contable|Contabilidad|0||Grado universitario o FP|2200|Encargado de contabilidad
Jefe de atencion al cliente|Atencion cliente|0||Grado universitario o FP|2300|Encargado de supervisar la atencion al cliente
Jefe de logistica|Logistica|0||Grado universitario o FP|2200|Encargado de hacer la logistica
Jefe de mantenimiento de coches|Logistica|0||Grado universitario o FP|2200|Encargado de mantener los coches y buscar nuevos
```



```

sqlite> select * from rentsite_solicitaalquiler;
19281|2829|3|24628498W|A5963
19281|2829|2|24692498Q|A6358
92881|1229|1|24692326M|A9864
19281|2829|3|24628494W|A7852
19281|2829|2|24628494W|A0036
19281|2829|2|25628498W|A3463
811|129|3|24592498Q|A1358
92881|129|4|20692325M|A1864
18111|2829|3|54628494T|A3352
3281|2829|2|84628494Y|A0936
sqlite> select * from rentsite_vehiculo;
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|2312QJG
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|6740HNI
5|diesel|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|4260PPS
5|diesel|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|0793DCP
5|diesel|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|2628JVS
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|9452XLH
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|8355R0V
5|diesel|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|8957LEJ
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|0386VGO
5|gasolina|automatico|turismo|Disponible|No necesario|Calle Aguila, Granada, 18002|1039WUH

```

## Implementación

Hemos creado una página web utilizando el framework de Django con python, HTML para la interfaz, y como base de datos SQLite y la hemos subido al servidor de pythonanywhere por lo que podemos acceder desde cualquier navegador con acceso a internet y manipular la base de datos, así como llevar a cabo cualquier funcionalidad.

La página web es la siguiente:

<http://maxiddsi2020.pythonanywhere.com>

Todo el código escrito para que esto funcionase podemos encontrarlo en el siguiente repositorio:

[https://github.com/maxisuarez/Proyecto\\_DDSI](https://github.com/maxisuarez/Proyecto_DDSI)

## Requisitos funcionales implementados

Todas las tablas constan de una página como la siguiente:

## Lista de facturas

Añadir factura

Id	<input type="text"/>	Fecha mayor o igual que	<input type="text"/>
Fecha menor o igual que	<input type="text"/>	Precio total mayor o igual que	<input type="text"/>
Precio total menor o igual que	<input type="text"/>	Proveedor contiene	<input type="text"/>
<p>Buscar</p>			

ID	Fecha	Proveedor	Precio total (en €)	Productos		
F1234	4 de Noviembre de 2020	Repuestos Granada	630,00	Correa distribucion con identificador P7844 -----	<a href="#">Editar</a>	<a href="#">Borrar</a>
F6349	4 de Noviembre de 2020	Desguaces Granada	500,00	Reemplazo frenos con identificador P0022 -----	<a href="#">Editar</a>	<a href="#">Borrar</a>
F1236	10 de Noviembre de 2020	Limpialo	9,00	Lavado con identificador P4152 -----	<a href="#">Editar</a>	<a href="#">Borrar</a>
F0025	22 de Noviembre de 2020	Hermanos Gámiz SL	20,00	Etiqueta parabrisas con identificador P1111 -----	<a href="#">Editar</a>	<a href="#">Borrar</a>
F8520	25 de Noviembre de 2020	Repuestos Granada	123,00	Llantas con identificador P5632 -----	<a href="#">Editar</a>	<a href="#">Borrar</a>

Donde podemos crear, editar, borrar y buscar según criterios lo que necesitemos de dicha tabla. Gracias a esto es posible conseguir todas las funcionalidad que necesitábamos (y muchas más) como detallamos a continuación.

Despliegue del menú contabilidad :

En el apartado “Facturas” hemos llevado a cabo el

**RF1.1 Elaborar y registrar facturas.** También se nos ofrece la opción de buscar facturas.

También encontramos el **RF1.2 Elaborar informes contables** satisfecho en el apartado “Informes”, similarmente .

Las ganancias se registran a la hora de registrar un alquiler, en “Logística” → “Alquileres” , cumpliendo así con **RF1.3 Registrar ganancias.**

**RF1.4 Consultar documentos financieros** también es satisfactorio ya que podemos consultar como hemos dicho anteriormente, facturas, informes y alquileres.

En “Contabilidad”--> “Balances financieros” , se nos ofrece la oportunidad de realizar balances financieros y consultarlos : **RF1.5 Realizar balance financiero , RF1.6 Consultar situación financiera.**

En el apartado de “Recursos Humanos” → “Empleados” , podemos llevar a cabo la acción de **RF2.1 Dar de alta a un empleado** y **RF2.2 Dar de baja a un**

**empleado**, al crear un empleado o añadirle una fecha de baja, también podemos modificarlo y además en esa página se nos presenta la lista de los empleados, (**RF2.7 Consultar empleados** )

En esta lista , hay un apartado que indica la fecha de pago , cambiamos cada vez que se le ha pagado la nómina al empleado, así satisfacemos **RF2.3 Pagar nómina**

Similarmente en “Recursos Humanos” → “Puestos” , se nos despliega una lista para ver los distintos puestos ( **RF2.6 Consultar puestos vacantes**) Además , en cada entrada del puesto podemos **RF2.5 Eliminar puesto vacante** y además hay un botón para poder introducir un nuevo puesto, **RF2.4 Añadir puesto vacante**.

En el despliegue : “Almacén” → “Vehículos”, nos aparece una lista de los vehículos de los que consta la empresa , sus distintas características, estado , mantenimiento, situación, etc , con estos distintos campos , damos por satisfechos: **RF 3.1. Consultar disponibilidad de vehículo** ( en el campo estado, está no disponible , o en el taller.), **RF 3.4. Consulta si un coche necesita hacer mantenimiento** ( mantenimiento : necesario o no )

**RF 4.1 Llevar coche a taller**( editar el campo estado a en taller ) **RF 4.2 Traer coche del taller** ( editar el campo estado a disponible ) . **RF 4.6 Devolver vehículo** ( marcar como disponible ) . En el apartado “Situación” , introducimos la geolocalización que recibamos y la actualizamos editando (**RF 4.5 Geolocalización de automóviles**) .

Además, con el botón de “Añadir vehículo” , cumplimos con **RF3.2. Dar de alta un vehículo**.

En cada entrada de vehículo, dentro de la lista, encontramos los botones para editar o borrar vehículo y así cumplimos con **RF 3.5. Cambiar estado vehículo**, **RF 3.3. Dar de baja un vehículo** .

Otra facilidad a la hora de consultar, es el filtrado que se nos permite hacer según distintos campos , con lo que conseguimos hacer una búsqueda personalizada y por tanto, satisfacer **RF 3.6. Consultar listado total de vehículos disponibles**.

También, realizando la creación de alquileres conseguimos **RF 4.3 Solicitar servicio de coche** e igualmente borrando dicho alquiler , **RF 4.4 Cancelar pedido de servicio de coche**.

## Restricciones implementadas y control de fallos

**RS1.1, RS2.1, RS 3.2:** Ningún campo puede ser vacío.

Los campos obligatorios aparecen con un asterisco, si se intenta crear o editar un campo que no puede estar vacío dejándolo vacío al intentar hacerlo nos saltará un error que no nos dejará hacerlo.

En el ejemplo siguiente intentamos crear un informe con campos vacíos, no nos deja y nos dice que los campos no pueden estar vacíos.

maxiddsi2020.pythonanywhere.com/crearInforme

GranaCar Recursos humanos Contabilidad Almacen Logística Relaciones

## Registro nuevo informe

Introduce la información del informe

Id\*

Fecha de realización del informe\*

Guardar

**RS1.2, RS2.2, RS 3.1, RS 3.3:** El formato de los datos introducidos debe ser correcto

Los formatos deben ser correctos, si alguna vez intentamos crear un id, fecha o dato con un formato incorrecto, nos saltará un error y no podremos crearlo.

En el ejemplo intentamos crear un informe con el formato de ID incorrecto y formato de fecha incorrecta, salta error y no nos deja.

maxiddsi2020.pythonanywhere.com/crearInforme

GranaCar Recursos humanos Contabilidad Almacen Logística Relaciones

**Error:**

- fecha
  - Introduzca una fecha válida.
- id
  - El ID debe tener un formato IXXXX con X un número.

## Registro nuevo informe

Introduce la información del informe

Id\*

fsadf

El ID debe tener un formato IXXXX con X un número.

Fecha de realización del informe\*

5216436223

**RS1.3, RS 1.4, RS1.6, RS2.3, RS3.2, RS 3.3:** El identificador es único en el sistema. Se cumple al ser una clave, si intentamos crear un nuevo objeto con el mismo identificador nos saldrá error y no podremos crearlo.

En el ejemplo intentamos crear una factura con un id que ya existe, nos salta error y no nos deja crearlo.

The screenshot shows a web browser window with the URL `maxiddsi2020.pythonanywhere.com/crearFactura`. The page has a dark header with the logo "GranaCar" and a navigation menu with items: "Recursos humanos", "Contabilidad", "Almacen", "Logística", and "Relaciones". A red error message box at the top states: "Error: id" followed by a bullet point "id" and a sub-point "Ya existe Factura con este Id.". Below the error, the main heading is "Registro nueva factura" with the subtitle "Introduce la información de la factura". The form contains several input fields: "Id\*" with the value "F1233", "Fecha\*" with the value "8/8/2020", and "Proveedor" with the value "Mannolo". A red error message "Ya existe Factura con este Id." is displayed below the "Id\*" field.

RS1.5, RS1.7, RS 4.1, RS 4.4: El identificador seleccionado no existe.  
Solo podemos elegir identificadores que ya están creados en el sistema.

En el ejemplo, si queremos poner un informe solo podemos seleccionar los que salgan al desplegar la lista de informes que existe.

The screenshot shows a form with a label "Informe\*" above a dropdown menu. The dropdown menu is open, showing a list of report names. Below the dropdown menu is a green button labeled "Guardar".

RS2.5: Los datos de borrado, modificación o consulta deben existir en la base de datos  
Se pueden ver todos en su respectiva página.



## Lista de puestos

Añadir puesto

Departamento	<input type="text"/>	Nombre puesto contiene	<input type="text"/>
Numero de vacantes mayor o igual que	<input type="text"/>	Numero de vacantes menor o igual que	<input type="text"/>
Sueldo mayor o igual que	<input type="text"/>	Sueldo menor o igual que	<input type="text"/>
Aptitudes necesarias contiene	<input type="text"/>	Requisitos puesto contiene	<input type="text"/>

Buscar

Nombre	Departamento	Nº Vacantes	Empleados	Aptitudes	Requisitos	Sueldo(€)		
Jefe de marketing	Marketing	0	Jose con identificador 672872735D -----	None	'Grado en Marketing o finanzas',	2200	<a href="#">Editar</a>	<a href="#">Borrar</a>
Asesor de marketing	Marketing	1	Manuel con identificador 672872325R ----- Angela con identificador 543872735K	None	Grado en Marketing o finanzas	2000	<a href="#">Editar</a>	<a href="#">Borrar</a>

RS 3.4: El estado debe ser correcto.

Solo es posible seleccionar entre la lista de los posibles estados.

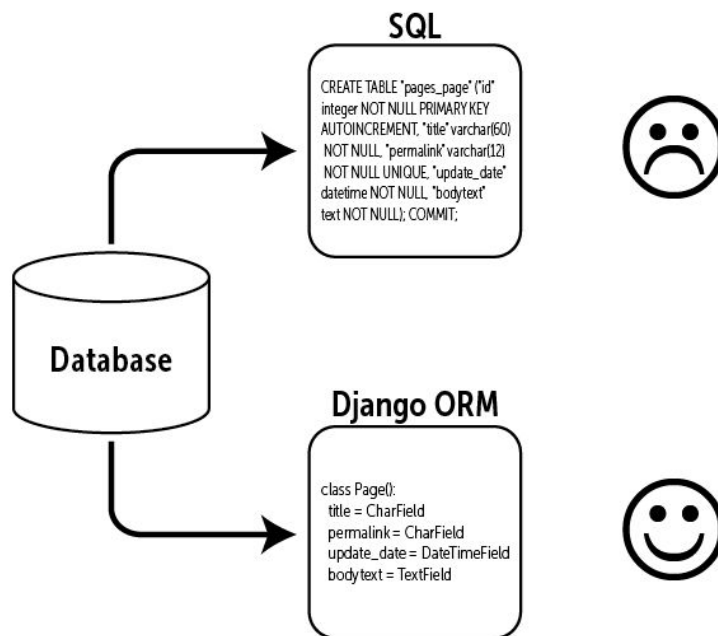
Estado\*

Disponibile ▾

### NOTA: Cómo trabaja Django con las bases de datos

En Django, aunque existe la posibilidad, no se trabaja directamente con SQL para manejar la base de datos, sino con la clase *models* de Django. Un modelo es una fuente finita de información de nuestros datos. Contiene los campos esenciales y comportamientos de los datos que estamos almacenando. En general, cada modelo se identifica con una única tabla en la base de datos.

Los modelos en Django proveen un Object-relational mapping (ORM) o asignación objeto-relacional en español para la base de datos subyacente. Es una técnica que facilita trabajar con datos y bases relacionales. En vez de utilizar SQL, se utiliza el lenguaje del ORM disponible, mucho más intuitivo, sencillo y robusto ante errores.



Cuando creamos un modelo, Django ejecuta la sentencia SQL correspondiente para crear una tabla en la base de datos que se corresponda con ese modelo. También se encarga de enlazar toda la información necesaria en la base de datos. Por ejemplo, si hay una clave externa, referencia a la tabla correspondiente. De manera similar, se realizan consultas y se modifican tuplas de las tablas.

Django soporta oficialmente 5 bases de datos:

- PostgreSQL
- MySQL
- SQLite
- Oracle
- MariaDB (solo en Django 3)

Podemos ajustar la configuración en el documento settings.py. Es posible almacenar la base de datos de manera local en nuestro servidor, pero también se dispone del mecanismo necesario para conectarse con una base de datos alojada en un servidor externo. Basta con modificar el settings.py para ello.

Nosotros utilizamos la base de datos en local, en el mismo servidor que desplegamos la aplicación y servimos el frontend, por lo que no es necesario conectarnos a un servidor externo para tener acceso a la base de datos.