

UNIVERSIDAD DE GRANADA

INGENIERÍA INFORMÁTICA

---

# Movilízate

---

Autores: GUILLERMO GÓMEZ TRENADO

ADRIÁN PELAEZ VEGAS

JOSÉ ANTONIO RUIZ MILLÁN

MOHAMMAD YA'AFAR UASANI VALVERDE

Asignatura: Diseño y Desarrollo de Sistemas de Información

15 de enero de 2018



# Índice

<b>1. P1: Descripción y análisis</b>	<b>2</b>
1.1. Descripción del sistema . . . . .	2
1.2. Análisis de todos los subsistemas . . . . .	3
1.2.1. Requisitos funcionales . . . . .	3
1.2.2. Requisitos de datos . . . . .	6
1.2.3. Restricciones semánticas . . . . .	31
<b>2. P2: Diagrama de flujo de datos y esquema conceptual ER de la base de datos</b>	<b>32</b>
2.1. Esquema de caja negra . . . . .	32
2.2. Esquema almacén . . . . .	32
2.3. Refinamiento de cada subsistema . . . . .	34
2.3.1. Gestión de multas, clientes y carnés (Guillermo) . . . . .	34
2.3.2. Gestión de clientes . . . . .	35
2.3.3. Gestión de carnés (Guillermo) . . . . .	36
2.3.4. Gestión de multas (Guillermo) . . . . .	37
2.3.5. Gestión de reservas (Guillermo) . . . . .	38
2.3.6. Gestión de Recursos Humanos (Jose Antonio) . . . . .	40
2.3.7. Gestión de altercados (Jose Antonio) . . . . .	41
2.3.8. Gestión de empleados (Jose Antonio) . . . . .	42
2.3.9. Gestión de tareas (Jose Antonio) . . . . .	43
2.3.10. Gestión de flota (Adrián) . . . . .	44
2.3.11. Gestión de motos (Adrián) . . . . .	45
2.3.12. Gestión de batería (Adrián) . . . . .	46
2.3.13. Gestión de limpieza (Adrián) . . . . .	47
2.3.14. Gestión de disponibilidad (Adrián) . . . . .	48
2.3.15. Gestión de posición geográfica (Adrián) . . . . .	49
2.4. Esquemas externos de procesos y almacenes al máximo nivel de refinamiento . . . . .	50
2.4.1. Esquema externo G.1 . . . . .	50
2.4.2. Esquema externo G.2 . . . . .	50
2.4.3. Esquema externo G.3 . . . . .	50
2.4.4. Esquema externo G.4 . . . . .	51
2.4.5. Esquema externo J.1 . . . . .	51
2.4.6. Esquema externo J.2 . . . . .	52
2.4.7. Esquema externo J.3 . . . . .	52
2.4.8. Esquema externo J.4 . . . . .	53
2.4.9. Esquema externo A.1 . . . . .	54
2.4.10. Esquema externo A.2 . . . . .	54
2.4.11. Esquema externo A.3 . . . . .	55
2.4.12. Esquema externo A.4 . . . . .	56
2.4.13. Esquema externo A.5 . . . . .	56
2.5. Esquema conceptual ER para cada subsistema . . . . .	57
2.5.1. Subsistemas clientes, reservas, carnés y multas . . . . .	57
2.5.2. Subsistema Gestion de Recursos Humanos . . . . .	58
2.5.3. Subsistema Gestion de Flota . . . . .	59

2.6.	Esquema conceptual ER de todo el sistema . . . . .	60
<b>3.</b>	<b>P3: Operaciones de datos</b>	<b>61</b>
3.1.	Subsistemas clientes, reservas, carnés y multas (Guillermo) . . . . .	61
3.2.	Subsistema Gestion de Recursos Humanos (Jose Antonio) . . . . .	66
3.3.	Subsistema Gestion de Flota (Adrián) . . . . .	74
<b>4.</b>	<b>Implementación del sistema</b>	<b>78</b>
4.1.	Paso a tablas . . . . .	78
4.2.	Normalización . . . . .	79
4.2.1.	Empleado (Guillermo) . . . . .	79
4.2.2.	Usuario (Guillermo) . . . . .	80
4.2.3.	Microalquiler (Guillermo) . . . . .	80
4.2.4.	Tecnico (Jose Antonio) . . . . .	81
4.2.5.	Administrador (Jose Antonio) . . . . .	81
4.2.6.	Se_asigna (Jose Antonio) . . . . .	81
4.2.7.	Flota (Adrián) . . . . .	82
4.2.8.	Tarea (Adrián) . . . . .	82
4.3.	Creacion de tablas (SQL) . . . . .	82
4.4.	Triggers en PL-SQL . . . . .	84
4.4.1.	Tabla Flota . . . . .	85
4.4.2.	Tabla Microalquiler . . . . .	86
4.4.3.	Tabla Se Asigna . . . . .	88
4.5.	Interfaz de administración: Oracle Application Builder . . . . .	89
4.5.1.	Página de inicio . . . . .	90
4.5.2.	Insertar nueva moto . . . . .	91
4.5.3.	Mostrar motos sucias y mostrar motos con batería baja . . . . .	91
4.5.4.	Tareas asignadas técnico y microalquileres . . . . .	92
4.5.5.	Actualizar motos y tareas . . . . .	93
4.6.	Interfaz de usuario: NodeJS . . . . .	94

# P1: Descripción y análisis

## Descripción del sistema

Queremos diseñar un sistema de información para la gestión de una empresa de alquileres rápidos de motos, las motos están repartidas por la ciudad, y cualquier usuario registrado puede activar una desde su aplicación del móvil. La aplicación de gestión de esta plataforma se divide en cuatro módulos, uno para la gestión de las finanzas -pagos y cobros-, otro para la gestión de usuarios, un tercero para la gestión de la flota y el último para la gestión de recursos humanos.

- Gestión de las finanzas. Yafar
  - Cobrar el total de minutos utilizados a los usuarios.
  - Pagar a los empleados (Nóminas).
  - Dar de alta y baja a empresas de alquiler de motos.
  - Contratación del seguro.
  - Facturas.
- Gestión de usuarios. Guillermo
  - Dar de alta y baja a usuarios.
  - Verificar el carné de conducir.
  - Contabilizar el tiempo de uso.
  - Derivar multas de tráfico a los usuarios.
- Gestión de la flota. Adrián
  - Alta y baja de las motos.
  - Controlar estado de limpieza (Avisar a recursos humanos).
  - Controlar batería (Avisar a recursos humanos).
  - Definir posición geográfica de las motos.
  - Control de disponibilidad.
- Gestión de recursos humanos. Jose Antonio
  - Dar de alta y baja empleados.
  - Asignar tareas (limpieza, mover moto y cambio de batería).
  - Comunicar altercado o problema entre distintos empleados.

# **Análisis de todos los subsistemas**

## **Requisitos funcionales**

### **Guillermo**

#### **RF-1 CRUD Usuarios (Clientes)**

- RF-1.1 Crear usuarios. El sistema ha de permitir dar de alta nuevos usuarios. RD-1, y RD-2.
- RF-1.2 Leer usuarios. El sistema debe proporcionar los datos de los usuarios atendiendo a distintos criterios de búsqueda. Optimizado para buscar por DNI, por teléfono y por tarjeta de crédito. RD-3,4,5,6.
- RF-1.3 Actualizar usuarios. El sistema debe permitir modificar los datos de los usuarios en las mismas condiciones que RD-7,8.
- RF-1.4 Eliminar usuarios. El sistema debe permitir eliminar un usuario sólo en casos estrictamente necesarios (errores administrativos), en los demás casos cambiará su estado a DE BAJA. RD-9.
- RF-1.5 Dar de baja usuario. El sistema debe permitir cambiar el estado de un usuario a de baja. Conservando su historia pero sin permitirle realizar operaciones. RD-10,11.

#### **RF-2 Verificación de carnés**

- RF-21. El sistema debe almacenar la información de usuario y las fotos sobre el carné a fin de que los administradores puedan verificarlo. RD-76,77.
- RF-22. Si un administrador no verifica un carné se notificará al usuario con el motivo del rechazo. RD-78,79.

#### **RF-3 Gestión de minialquileres**

- RF-31.
- RF-32. El sistema debe permitir a los usuarios reservar y alquilar las motos que reunan las conficiones de alquilerabilidad. RD-80,81
- RF-33. Las reservas durarán exclusivamente 15 minutos. Pasado el tiempo la moto vuelve a estar a disposición del resto de usuarios para su reserva (Incluido éste al que la reserva se le caducó) y la reserva se anota como cancelada. RD-82,83.
- RF-34. Las reservas se pueden cancelar. RD-82,83.
- RF-35. Una vez comenzado el alquiler se debe almacenar la información relativa al momento de inicio y punto inicial. RD-109,110
- RF-36. Al finalizar el alquiler se almacena el punto de fin y la hora y se registrá en el historial del alquileres del usuario. RD-84,111

#### **RF-4 Gestión de multas**

- RF-41. El sistema debe permitir establecer un mecanismo con la DGT y los ayuntamientos mediante el cual las multas recibidas por las motos sean respondidas con la información

del usuario que la tenía alquilada en ese momento o el último que la alquiló si no está activa. RD-85,86,87.

## **Jose Antonio**

### **RF-5 C.R.U.D Usuarios (Empleados)**

- RF-5.1 Crear usuarios (tipo empleado). El sistema ha de permitir dar de alta nuevos empleados con los requisitos de entrada de RD-12 , requisitos de manejo RD-13 (Todos obligatorios a excepción del atributo Evaluación).
- RF-5.2 Leer empleados. El sistema debe proporcionar los datos de los empleados atendiendo a distintos criterios de búsqueda. Con requisitos de entrada RD-14, requisitos de manejo RD-15 y requisitos de salida RD-16
- RF-5.3 Actualizar empleados. El sistema debe permitir modificar los datos de los empleados con requisitos de entrada RD-17 y requisitos de manejo RD-13.
- RF-5.4 Eliminar empleados. El sistema debe permitir eliminar un empleado sólo en casos estrictamente necesarios (errores administrativos), en los demás casos cambiará su estado a DE BAJA. Requisitos de entrada RD-18 y requisitos de manejo RD-19

### **RF-6 C.R.U.D Tareas:**

- RF-6.1 Crear tarea. El sistema debe permitir crear una tarea con los requisitos de datos de entrada de RD-20, con los requisitos de manejo de RD-21 Cada tarea debe estar asociada a un empleado obligatoriamente en su creación.
- RF-6.2 Leer tarea. El sistema debe proporcionar los datos de las tareas atendiendo a distintos criterios de búsqueda. Requisitos de entrada RD-22 , requisitos de manejo RD-23 y requisitos de salida RD-24
- RF-6.3 Actualizar tareas. El sistema debe permitir modificar los datos de las tareas teniendo como requisitos de entrada RD-25 , requisitos de manejo RD-21.Teniendo en cuenta las restricción semántica 10.
- RF-6.4 Eliminar tareas. El sistema debe permitir eliminar una tarea sólo en casos estrictamente necesarios (errores administrativos), en los demás casos cambiará su estado a FINALIZADA. Requisitos de entrada RD-26 , requisitos de manejo RD-27

### **RF-7 C.R.U Altercados:**

- RF-7.1 Crear altercado. El sistema debe permitir crear un altercado con los requisitos de datos de entrada RD-28 y los requisitos de manejo RD-29
- RF-7.2 Leer altercado. El sistema debe proporcionar los datos de los altercados atendiendo a distintos criterios de búsqueda. Requisitos de entrada RD-30 , requisitos de manejo RD-31 , requisitos de salida RD-32
- RF-7.3 Actualizar altercado. El sistema debe permitir modificar los datos de los altercados con requisitos de entrada RD-33 , requisitos de manejo RD-34 .Teniendo en cuenta la restricción semántica 11.

## Adrián

### RF-8 C.R.U.D Motos

- RF-8.1 Crear moto. El sistema ha de permitir dar de alta nuevas motos que se unan a la flota. RD-35 de entrada, RD-36 de manejo.
- RF-8.2 Leer moto. El sistema debe proporcionar los datos de las motos atendiendo a distintos criterios de búsqueda. Optimizado para buscar por matrícula. RD-37 de entrada, RD-38 de manejo, RD-39 de salida.
- RF-8.3 Actualizar moto. El sistema debe permitir modificar los datos de las motos. RD-40 de entrada, RD-41 de manejo.
- RF-8.4 Eliminar moto. El sistema debe permitir eliminar motos solo en casos en los que se tenga la total seguridad de que dicha información es redundante siempre habrá que mantener la información de dicha moto un margen de tiempo para posibles multas o problemas ocasionados con dicha moto). En cualquier otro caso, la moto no será eliminada y su estado pasará a DE BAJA. RD-42 de entrada, RD-43 de manejo.

### RF-9 Gestión batería

- RF-91. Baterías bajas. El sistema ha de permitir mostrar un listado de aquellas motos cuya batería es baja para que se proceda a su carga. A su vez, el sistema deberá dar un aviso a recursos humanos para que cree la tarea de recarga de batería correspondiente. RD-44 de manejo, RD-45 de salida.
- RF-92. Consultar estado batería. El sistema ha de permitir mostrar el estado de la batería de una moto indicada por su matrícula. RD-88 de entrada, RD-89 de manejo, RD-90 de salida.
- RF-93. Cambiar estado batería. El sistema debe permitir cambiar el estado de la batería a ‘alto’ de una moto identificada por su matrícula. RD-91 de entrada, RD-92 de manejo.

### RF-10 Gestión limpieza

- RF-101. Motos sucias. El sistema ha de permitir mostrar un listado de aquellas motos cuyo estado de limpieza sea sucio. Esta información será recogida a través de avisos que los clientes podrán dar a través de la aplicación de usuario. A su vez, el sistema deberá dar un aviso a recursos humanos para que cree la tarea de limpiar la moto correspondiente. RD-46 de manejo, RD-47 de salida
- RF-102. Consultar estado limpieza. El sistema ha de permitir mostrar el estado de limpieza de una moto indicada por su matrícula. RD-93 de entrada, RD-94 de manejo, RD-95 de salida.
- RF-103. Cambiar estado limpieza. El sistema ha de permitir cambiar el estado de limpieza de una moto indicada por su matrícula. RD-96 de entrada, RD-97 de manejo.

### RF-11 Gestión disponibilidad

- RF-111. El sistema debe ser capaz de mostrar un listado de las motos disponibles en cada momento. RD-48 de manejo, RD-49 de salida.

RF-112. Consultar estado disponibilidad. El sistema ha de permitir mostrar el estado de disponibilidad de una moto indicada por su matrícula. RD-98 de entrada, RD-99 de manejo, RD100- de salida.

RF-113. Cambiar estado disponibilidad. El sistema ha de permitir cambiar el estado de disponibilidad de una moto indicada por su matrícula. RD-101 de entrada, RD-102 de manejo.

#### RF-12 Gestión posición geográfica

RF-121. Listado posición. El sistema debe ser capaz de mostrar un listado de la geolocalización de cada moto en cada momento que se requiera. RD-50 de manejo, RD-51 de salida.

RF-122. El sistema debe poder calcular distancias entre motos y entre motos y usuarios para poder facilitar esta información a los clientes. RD-106 de entrada, RD-107 de manejo, RD-108 de salida.

RF-123. Consulta posición. El sistema debe ser capaz de mostrar la posición de una moto identificada por su matrícula. RD-103 de entrada, RD-104 de manejo, RD-105 de salida.

### Requisitos de datos

#### RD-1 Crear usuario: (entrada)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Alta y caducidad del carnet de conducir. Dos fechas
- h) Estado de verificación. Booleano
- i) Datos de la tarjeta de crédito (Número, fecha y ccv2). Caracteres alfanum

#### RD-2 Crear usuario: (manejo)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Alta y caducidad del carnet de conducir. Dos fechas



- h) Estado de verificación. Booleano
- i) Datos de la tarjeta de crédito (Número, fecha y ccv2). Caracteres alfanum
- j) Estado de actividad. Enum
- k) Fecha de baja. Fecha

RD-3 Leer usuario: (Entrada)

- a) DNI. Secuencia alfanumérica

RD-4 Leer usuario: (Entrada)

- a) Teléfono. Secuencia alfanumérica

RD-5 Leer usuario: (Entrada)

- a) Tarjeta de crédito. Secuencia alfanumérica

RD-6 Leer usuario: (Salida)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Alta y caducidad del carnet de conducir. Dos fechas
- h) Estado de verificación. Booleano
- i) Datos de la tarjeta de crédito (Número, fecha y ccv2). Caracteres alfanum

RD-7 Actualizar usuario (Entrada)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Alta y caducidad del carnet de conducir. Dos fechas
- h) Estado de verificación. Booleano
- i) Datos de la tarjeta de crédito (Número, fecha y ccv2). Caracteres alfanum

RD-8 Actualizar usuario (Manejo)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Alta y caducidad del carnet de conducir. Dos fechas
- h) Estado de verificación. Booleano
- i) Datos de la tarjeta de crédito (Número, fecha y ccv2). Caracteres alfanum

RD-9 Eliminar usuarios (Entrada)

- a) DNI. Secuencia alfanumérica

RD-10 Datos sobre las bajas de usuarios (Entrada)

- a) DNI. Secuencia alfanumérica
- b) Fecha de baja
- c) Motivo de baja

RD-11 Datos sobre las bajas de usuarios (Manejo)

- a) DNI. Secuencia alfanumérica
- b) Fecha de baja
- c) Motivo de baja

RD-12 Datos de creación de los empleados (entrada)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Cuenta corriente. secuencia alfanumérica
- h) Alta y baja del contrato. secuencia alfanumérica
- i) Tipo de contrato. secuencia alfanumérica
- j) Salario. Numero
- k) Estado (Activo o de baja). Enum\*

- l) Puesto de trabajo
- m) Evaluación

RD-13 Datos de creación/modificación de los empleados (manejo)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Cuenta corriente. secuencia alfanumérica
- h) Alta y baja del contrato. secuencia alfanumérica
- i) Tipo de contrato. secuencia alfanumérica
- j) Salario. Numero
- k) Estado (Activo o de baja). Enum\*
- l) Puesto de trabajo
- m) Evaluación

RD-14 Datos sobre consulta de empleados (entrada)

- a) DNI
- b) Nombre y apellidos
- c) Cuenta Corriente
- d) Estado

RD-15 Datos sobre consulta de empleados (manejo)

- a) DNI
- b) Nombre y apellidos
- c) Cuenta Corriente
- d) Estado

RD-16 Datos de consulta de los empleados (salida)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Teléfono. Secuencia alfanumérica
- d) Email. Secuencia alfanumérica

- e) Fecha de nacimiento. Fecha
- f) Cuenta corriente. secuencia alfanumérica
- g) Alta y baja del contrato. secuencia alfanumérica
- h) Tipo de contrato. secuencia alfanumérica
- i) Salario. Numero
- j) Estado (Activo o de baja). Enum\*
- k) Puesto de trabajo
- l) Evaluación

RD-17 Datos de actualización de los empleados (entrada)

- a) Nombre y apellidos, secuencia alfanumérica
- b) DNI. Secuencia alfanumérica
- c) Clave de acceso. Secuencia cifrada de caracteres
- d) Teléfono. Secuencia alfanumérica
- e) Email. Secuencia alfanumérica
- f) Fecha de nacimiento. Fecha
- g) Cuenta corriente. secuencia alfanumérica
- h) Alta y baja del contrato. secuencia alfanumérica
- i) Tipo de contrato. secuencia alfanumérica
- j) Salario. Numero
- k) Estado (Activo o de baja). Enum\*
- l) Puesto de trabajo
- m) Evaluación

RD-18 Datos sobre eliminación de empleados (entrada)

- a) DNI
- b) Nombre y apellidos

RD-19 Datos sobre eliminación de empleados (manejo)

- a) DNI
- b) Nombre y apellidos
- c) Estado

RD-20 Datos sobre creación las tareas (entrada)

- a) Fecha y hora de realización

- b) Estado (en cola, en proceso, finalizada)
- c) Tiempo máximo
- d) Descripción
- e) Empleado que la realiza (DNI)
- f) Moto sobre la que se realiza

RD-21 Datos sobre creación/modificación de las tareas (manejo)

- a) Fecha y hora de realización
- b) Estado (en cola, en proceso, finalizada)
- c) Tiempo máximo
- d) Descripción
- e) Empleado que la realiza (DNI)
- f) Moto sobre la que se realiza

RD-22 Datos sobre la consulta de las tareas (entrada)

- a) Fecha y hora de realización
- b) Tipo
- c) Estado (en cola, en proceso, finalizada)
- d) Empleado que la realiza (DNI)
- e) Moto sobre la que se realiza

RD-23 Datos sobre la consulta de las tareas (manejo)

- a) Fecha y hora de realización
- b) Estado (en cola, en proceso, finalizada)
- c) Empleado que la realiza (DNI)
- d) Moto sobre la que se realiza

RD-24 Datos sobre consulta de las tareas (salida)

- a) Fecha y hora de realización
- b) Estado (en cola, en proceso, finalizada)
- c) Tiempo máximo
- d) Descripción
- e) Empleado que la realiza (DNI)
- f) Moto sobre la que se realiza

RD-25 Datos sobre actualización de las tareas (entrada)

- a) Fecha y hora de realización
- b) Estado (en cola, en proceso, finalizada)
- c) Descripción
- d) Tiempo máximo
- e) Empleado que la realiza (DNI)
- f) Moto sobre la que se realiza

RD-26 Datos sobre la eliminación de las tareas (entrada)

- a) Fecha y hora de realización
- b) Empleado que la realiza (DNI)
- c) Moto sobre la que se realiza
- d) Tipo

RD-27 Datos sobre la eliminación de las tareas (manejo)

- a) Fecha y hora de realización
- b) Estado (en cola, en proceso, finalizada)
- c) Tiempo máximo
- d) Descripción
- e) Empleado que la realiza (DNI)
- f) Moto sobre la que se realiza

RD-28 Datos sobre creación los altercados (entrada)

- a) Usuario/s que emite/n el altercado
- b) Usuario/s que está/n involucrados
- c) Descripción del problema
- d) Estado (en proceso, parado, resuelto)

RD-29 Datos sobre creación de los altercados (manejo)

- a) Usuario/s que emite/n el altercado
- b) Usuario/s que está/n involucrados
- c) Descripción del problema
- d) Estado (en proceso, parado, resuelto)

RD-30 Datos sobre consulta de los altercados (entrada)

- a) Usuario que emite el altercado
- b) Usuario que está involucrados

c) Estado (en proceso, parado, resuelto)

RD-31 Datos sobre consulta de los altercados (manejo)

a) Usuario que emite el altercado

b) Usuario que está involucrados

c) Estado (en proceso, parado, resuelto)

RD-32 Datos sobre consulta de los altercados (salida)

a) Usuario/s que emite/n el altercado

b) Usuario/s que está/n involucrados

c) Descripción del problema

d) Estado (en proceso, parado, resuelto)

e) Resolución

RD-33 Datos sobre actualización de los altercados (entrada)

a) Usuario que emite el altercado

b) Usuario que está involucrados

c) Estado (en proceso, parado, resuelto)

RD-34 Datos sobre actualización de los altercados (manejo)

a) Usuario/s que emite/n el altercado

b) Usuario/s que está/n involucrados

c) Descripción del problema

d) Estado (en proceso, parado, resuelto)

e) Resolución

RD-35 Crear moto (entrada)

a) Matrícula

b) Coste de alquiler

c) Empresa de alquiler

d) Estado de la batería

e) Estado de limpieza

f) Localización

g) Disponibilidad moto (En uso, reservada o libre)

h) Estado moto (Activa o de baja)

i) Cilindrada

RD-36 Datos crear moto (manejo)

- a) Matrícula
- b) Coste de alquiler
- c) Empresa de alquiler
- d) Estado de la batería
- e) Estado de limpieza
- f) Localización
- g) Disponibilidad moto (En uso, reservada o libre)
- h) Estado moto (Activa o de baja)
- i) Cilindrada

RD-37 Datos leer moto (entrada)

- a) Matrícula

RD-38 Datos leer moto (manejo)

- a) Matrícula
- b) Coste de alquiler
- c) Empresa de alquiler
- d) Estado de la batería
- e) Estado de limpieza
- f) Localización
- g) Disponibilidad moto (En uso, reservada o libre)
- h) Estado moto (Activa o de baja)
- i) Cilindrada

RD-39 Datos leer moto (salida)

- a) Matrícula
- b) Coste de alquiler
- c) Empresa de alquiler
- d) Estado de la batería
- e) Estado de limpieza
- f) Localización
- g) Disponibilidad moto (En uso, reservada o libre)
- h) Estado moto (Activa o de baja)



i) Cilindrada

RD-40 Datos actualizar moto (entrada)

a) Matrícula

b) Coste de alquiler

c) Empresa de alquiler

d) Estado de la batería

e) Estado de limpieza

f) Localización

g) Disponibilidad moto (En uso, reservada o libre)

h) Estado moto (Activa o de baja)

i) Cilindrada

RD-41 Datos actualizar moto (manejo)

a) Matrícula

b) Coste de alquiler

c) Empresa de alquiler

d) Estado de la batería

e) Estado de limpieza

f) Localización

g) Disponibilidad moto (En uso, reservada o libre)

h) Estado moto (Activa o de baja)

i) Cilindrada

RD-42 Datos eliminar moto (entrada)

a) Matrícula

RD-43 Datos eliminar moto (manejo)

a) Matrícula

b) Coste de alquiler

c) Empresa de alquiler

d) Estado de la batería

e) Estado de limpieza

f) Localización

g) Disponibilidad moto (En uso, reservada o libre)

- h) Estado moto (Activa o de baja)
  - i) Cilindrada
- RD-44 Datos motos batería baja(manejo)
  - a) Matricula
  - b) Estado de la batería
- RD-45 Datos motos batería baja(salida)
  - a) Matrícula
  - b) Estado de la batería
- RD-46 Datos motos sucias(manejo)
  - a) Matrícula
  - b) Estado de limpieza
- RD-47 Datos motos sucias(salida)
  - a) Matrícula
- RD-48 Datos motos disponibles(manejo)
  - a) Matrícula
  - b) Disponibilidad moto
- RD-49 Datos motos disponibles(salida)
  - a) Matrícula
- RD-50 Listado posición geográfica (entrada)
  - a) Matrícula
  - b) Localización
- RD-51 Listado posición geográfica (salida)
  - a) Matrícula
  - b) Localización
- RD-52 Creación minialquileres (entrada)
  - a) Estado del minialquiler (Reservado, activo, finalizado o cancelado)
  - b) Usuario que lo alquila
  - c) Moto alquilada
  - d) Hora de inicio
  - e) Hora de fin
  - f) Punto de inicio

g) Punto de fin

h) Tarifa

RD-53 Actualización minialquileres (entrada)

a) Estado del minialquiler (Reservado, activo, finalizado o cancelado)

b) Usuario que lo alquila

c) Moto alquilada

d) Hora de inicio

e) Hora de fin

f) Punto de inicio

g) Punto de fin

h) Tarifa

RD-54 Creación/actualización minialquileres (manejo)

a) Estado del minialquiler (Reservado, activo, finalizado o cancelado)

b) Usuario que lo alquila

c) Moto alquilada

d) Hora de inicio

e) Hora de fin

f) Punto de inicio

g) Punto de fin

h) Tarifa

RD-55 Creación Nóminas (Entrada):

a) Identificador: Secuencia alfanumérica

b) Cantidad: Numerica

c) Estado (pagada o no): alfanumérica

d) Cuenta corriente: Secuencia alfanumérica

e) Id empleado: numerica

f) Fecha

RD-56 Creación Nóminas (Manejo):

a) Identificador: Secuencia alfanumérica

b) Cantidad: Numerica

c) Estado (pagada o no): alfanumérica

- d) Cuenta corriente: Secuencia alfanumérica
- e) Id empleado: numerica
- f) Fecha

RD-57 Modificación Nóminas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Cuenta corriente: Secuencia alfanumérica
- c) Id empleado: numerica
- d) Fecha

RD-58 Modificación de Nóminas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Cantidad: Numerica
- c) Estado (pagada o no): alfanumérica
- d) Cuenta corriente: Secuencia alfanumérica
- e) Id empleado: numerica
- f) Fecha

RD-59 Consulta Nominas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Cuenta corriente: Secuencia alfanumérica
- c) Id empleado: numerica
- d) Fecha

RD-60 Consulta Nominas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Cantidad: Numerica
- c) Estado (pagada o no): alfanumérica
- d) Cuenta corriente: Secuencia alfanumérica
- e) Id empleado: numerica
- f) Fecha

RD-61 Consulta Nominas (Salida):

- a) Identificador: Secuencia alfanumérica
- b) Cantidad: Numerica
- c) Estado (pagada o no): alfanumérica

- d) Cuenta corriente: Secuencia alfanumérica
- e) Id empleado: numerica
- f) Fecha

RD-62 Eliminacion Nominas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Cuenta corriente: Secuencia alfanumérica
- c) Id empleado: numerica
- d) Fecha

RD-63 Eliminacion Nominas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Cantidad: Numerica
- c) Estado (pagada o no): alfanumérica
- d) Cuenta corriente: Secuencia alfanumérica
- e) Id empleado: numerica
- f) Fecha

RD-64 Creacion Leasing (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-65 Creacion Leasing (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-66 Modificacion Leasing (Entrada):

- a) Identificador: Secuencia alfanumérica

- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica

RD-67 Modificación Leasing (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos: Secuencia alfanumérica.

RD-68 Consulta Leasing (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica

RD-69 Consulta Leasing (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos: Secuencia alfanumérica.

RD-70 Consulta Leasing (Salida):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos: Secuencia alfanumérica.

RD-71 Eliminacion Leasing (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica

RD-72 Eliminacion Leasing (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-73 Creación Aseguradoras (Entrada)

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-74 Creación Aseguradoras (Manejo)

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-75 Modificación Aseguradoras (Entrada)

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica

- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos: Secuencia alfanumérica.

RD-76 Verificar carné (Entrada):

- a) DNI

RD-77 Verificar carné (Manejo):

- a) DNI
- b) Estado de verificación

RD-78 Rechazar carné (Entrada):

- a) DNI
- b) Motivo de rechazo

RD-79 Rechazar carné (Manejo):

- a) Nombre: Secuencia alfanumérica
- b) DNI: Secuencia alfanumérica
- c) Correo electrónico: Secuencia alfanumérica

RD-80 Reservar moto (Entrada):

- a) DNI: Secuencia alfanumérica
- b) Fecha de reserva: Fecha
- c) Matrícula: Secuencia alfanumérica

RD-81 Reservar moto (Manejo):

- a) Disponibilidad moto
- b) Minialquiler
- c) Estado del minialquiler (Reservado, activo, finalizado o cancelado)
- d) Usuario que lo alquila
- e) Moto alquilada
- f) Fecha y hora de reserva
- g) Fecha y hora de inicio
- h) Fecha y hora de fin
- i) Punto de inicio
- j) Punto de fin
- k) Tarifa



RD-82 Cancelar reserva (Entrada):

- a) DNI: Secuencia alfanumérica
- b) Fecha de reserva: Fecha
- c) Matrícula

RD-83 Cancelar reserva (Manejo):

- a) DNI: Secuencia alfanumérica
- b) Fecha de reserva: Fecha
- c) Minialquiler
- d) Estado del minialquiler (Reservado, activo, finalizado o cancelado)

RD-84 Finalizar reserva (Entrada):

- a) DNI: Secuencia alfanumérica
- b) Fecha de reserva: Fecha

RD-85 Obtener datos moto en uso (Entrada):

- a) Matrícula: Secuencia alfanumérica
- b) Fecha de incidente: Fecha

RD-86 Obtener datos motos en uso (Manejo):

- a) Usuario: Objeto
- b) Minialquiler: Objeto

RD-87 Obtener datos motos en uso (Salida):

- a) DNI: Secuencia alfanumérica
- b) Teléfono: Secuencia alfanumérica

RD-88 Consultar estado batería (Entrada):

- a) Matrícula

RD-89 Consultar estado batería (Manejo):

- a) Matrícula
- b) Estado de batería

RD-90 Consultar estado batería (Salida):

- a) Estado de batería

RD-91 Cambiar estado batería(Entrada):

- a) Matrícula

RD-92 Cambiar estado batería(Manejo):

- a) Matrícula
  - b) Estado de batería.
- RD-93 Consultar estado limpieza (Entrada):
  - a) Matrícula
- RD-94 Consultar estado limpieza (Manejo):
  - a) Matrícula
  - b) Estado de limpieza
- RD-95 Consultar estado limpieza (Salida):
  - a) Estado de limpieza
- RD-96 Cambiar estado limpieza (Entrada):
  - a) Matrícula
  - b) Estado de limpieza
- RD-97 Cambiar estado limpieza (Manejo):
  - a) Matrícula
  - b) Estado de limpieza
- RD-98 Consultar estado disponibilidad(Entrada):
  - a) Matrícula
- RD-99 Consultar estado de disponibilidad(Manejo):
  - a) Matrícula
  - b) Estado de disponibilidad
- RD-100 Consultar estado de disponibilidad (Salida):
  - a) Estado de disponibilidad
- RD-101 Cambiar estado de disponibilidad (Entrada):
  - a) Matrícula
  - b) Estado de disponibilidad
- RD-102 Cambiar estado de disponibilidad (Manejo):
  - a) Matrícula
  - b) Estado de disponibilidad
- RD-103 Consulta posición (entrada):
  - a) Matrícula
- RD-104 Consulta posición (manejo):

a) Matrícula

b) Posición

RD-105 Consulta posición (salida):

a) Posición

RD-106 Distancia (entrada):

a) Matrícula

RD-107 Distancia (manejo):

a) Matrícula

b) Posición

RD-108 Distancia (salida):

a) Distancia

RD-109 Comenzar uso moto(Entrada):

a) DNI: Secuencia alfanumérica

b) Fecha de reserva: Fecha

c) Matrícula: Secuencia alfanumérica

RD-110 Reservar moto (Manejo):

a) Disponibilidad moto

b) Minialquiler

c) Estado del minialquiler (Reservado, activo, finalizado o cancelado)

d) Hora de inicio

e) Punto de inicio

RD-111 Finalizar reserva (Manejo):

a) Disponibilidad moto

b) Minialquiler

c) Estado del minialquiler (Reservado, activo, finalizado o cancelado)

d) Hora de fin: Fecha

e) Punto de fin: Fecha

RD-112 Subir Carné (Entrada):

a) DNI

b) Fotocopia carné

RD-113 Subir Carné (Manejo):

a) Fotocopia carné

RD-114 Leer Carné (Entrada):

a) DNI

RD-115 Leer Carné (Salida):

a) Fotocopia carné

b) Rechazar carné (Salida):

c) Motivo de rechazo

d) Coreo electrónico: Secuencia alfanumérica

RD-116 Modificación Aseguradoras (Manejo)

a) Identificador: Secuencia alfanumérica

b) Nombre: Secuencia alfanumérica

c) Dirección: Secuencia alfanumérica

d) Correo: Secuencia alfanumérica

e) Teléfono: Secuencia Numérica

f) Otros datos: Secuencia alfanumérica.

RD-117 Consulta Aseguradoras (Entrada):

a) Identificador: Secuencia alfanumérica

b) Nombre: Secuencia alfanumérica

c) Dirección: Secuencia alfanumérica

d) Correo: Secuencia alfanumérica

e) Teléfono: Secuencia Numérica

RD-118 Consulta Aseguradoras (Manejo)

a) Identificador: Secuencia alfanumérica

b) Nombre: Secuencia alfanumérica

c) Dirección: Secuencia alfanumérica

d) Correo: Secuencia alfanumérica

e) Teléfono: Secuencia Numérica

f) Otros datos: Secuencia alfanumérica.

RD-119 Consulta Aseguradoras (Salida)

a) Identificador: Secuencia alfanumérica

b) Nombre: Secuencia alfanumérica

- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-120 Eliminación Aseguradoras (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica

RD-121 Eliminación Aseguradoras (Manejo)

- a) Identificador: Secuencia alfanumérica
- b) Nombre: Secuencia alfanumérica
- c) Dirección: Secuencia alfanumérica
- d) Correo: Secuencia alfanumérica
- e) Teléfono: Secuencia Numérica
- f) Otros datos:Secuencia alfanumérica.

RD-122 Creación Facturas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-123 Creación Facturas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-124 Modificación Facturas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-125 Modificación Facturas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-126 Consulta Facturas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-127 Consulta Facturas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-128 Consulta Facturas (Salida):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha

- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-129 Eliminación Facturas (Entrada):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-130 Eliminación Facturas (Manejo):

- a) Identificador: Secuencia alfanumérica
- b) Fecha:Formato Fecha
- c) Concepto:Secuencia alfanumérica
- d) Importe:Dato Numerico
- e) Receptor: Secuencia alfanumérica
- f) Pagada:Booleano

RD-131 Contabilización Tiempo (Entrada):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

RD-132 Contabilización Tiempo (Manejo):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico

f) Pagado: Booleano

RD-133 Modificación Tiempo (Entrada):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

RD-134 Modificación Tiempo (Manejo):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

RD-135 Consulta Facturacion Tiempo (Entrada):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

RD-136 Consulta Facturacion Tiempo (Manejo):

- a) Identificador Usuario: Secuencia alfanumérica
- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

RD-137 Consulta Facturacion Tiempo (Salida):

- a) Identificador Usuario: Secuencia alfanumérica



- b) Identificador Vehículo: Secuencia alfanumérica
- c) Fecha: Formato Fecha
- d) Tiempo: Formato Hora
- e) Importe: Dato Numerico
- f) Pagado: Booleano

## Restricciones semánticas

En nuestro caso, hemos visto oportuno añadir estas restricciones semánticas, que intentaremos aplicarlas todas en la propia **base de datos** para así no depender del sistema externo que se conecte a la base de datos, sino que directamente la base de datos incorpore estas comprobaciones (siempre que se pueda).

### Restricciones:

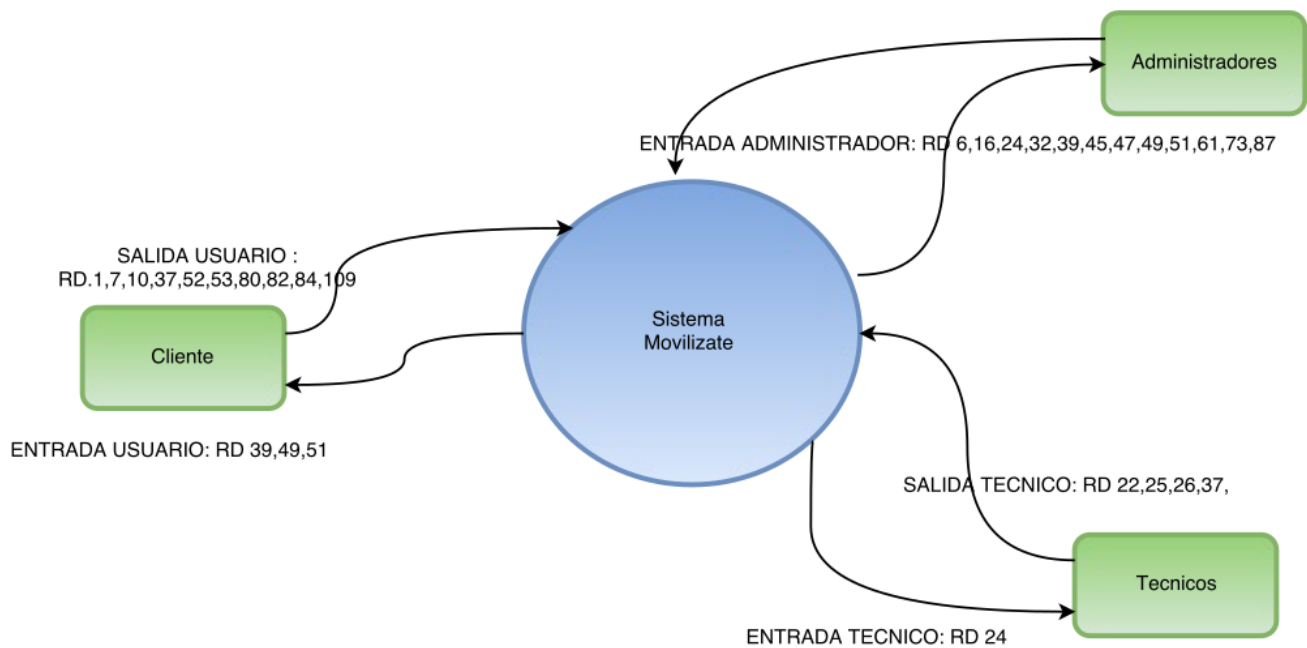
1. Una moto no se puede alquilar si está en uso o de baja
2. Un usuario no puede alquilar sin el carné verificado
3. Un usuario no se puede dar de alta sin alguno de los datos
4. Una moto no se puede dejar fuera del área designada
5. Una moto no se puede alquilar con la batería baja
6. Una moto no se puede alquilar sucia
7. Un usuario no puede alquilar dos motos a la vez
8. La hora de fin no puede ser anterior que la de inicio
9. Una moto con una tarea asignada y no terminada, esta obligatoriamente de baja hasta que todas las tareas asignadas estén terminadas.
10. No puede haber varias tareas el mismo día a la misma hora para la misma moto ni una tarea antes de que expire el tiempo máximo de la tarea anterior.
11. Un altercado con estado resuelto debe tener información en el atributo de Resolución del altercado.
12. No puede haber dos nóminas con el mismo identificador
13. No puede haber dos facturas con el mismo identificador

## P2: Diagrama de flujo de datos y esquema conceptual ER de la base de datos

### Esquema de caja negra

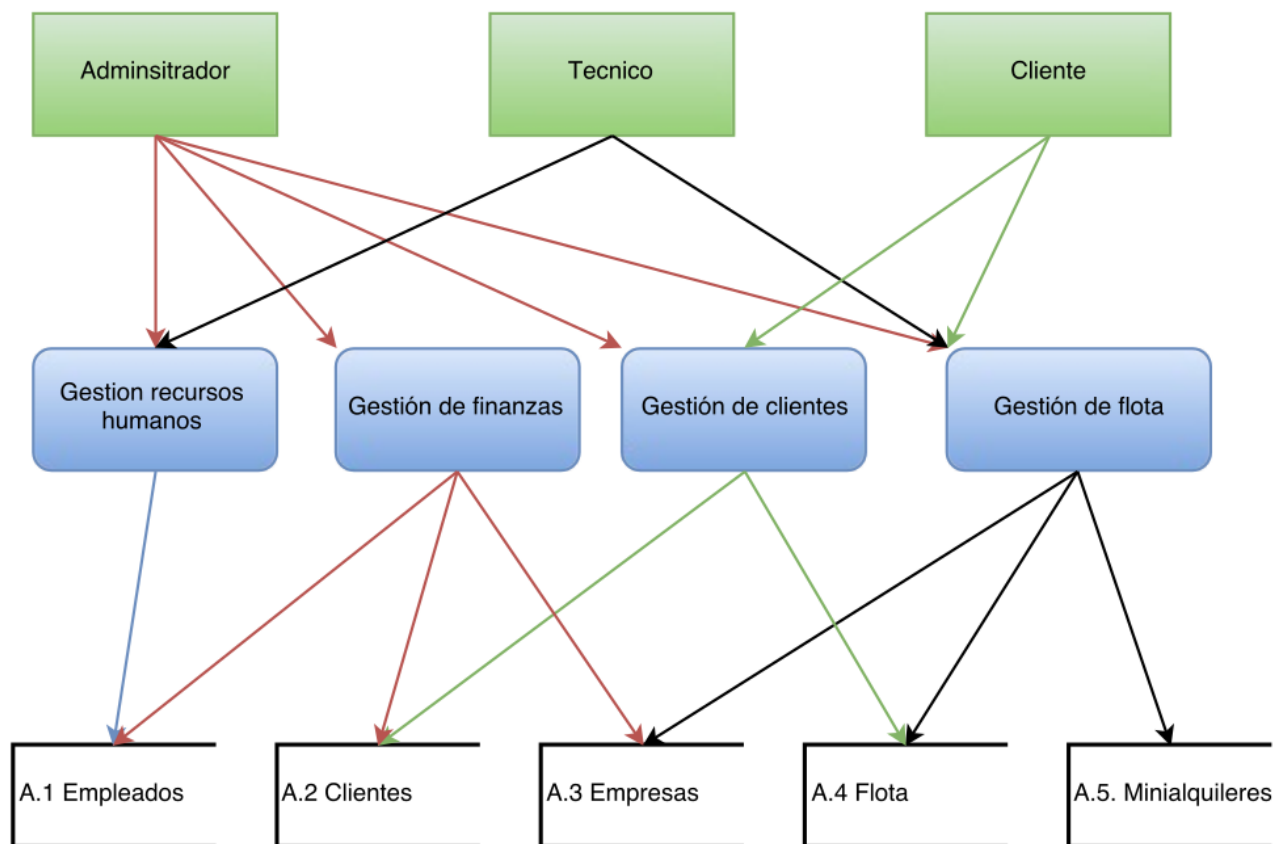
En este diagrama mostramos las distintas operaciones que realizan diferentes elementos (en nuestro caso personas) a **nivel abstracto** con el **sistema completo**.

SALIDA ADMINISTRADOR: RD 3,4,5,7,9,10,12,14,17,18,20,22,25,26,28,30,33,35,37,40,42,52,53,55,57,59,62,64,65,67,69,71,74,76,78,85



### Esquema almacén

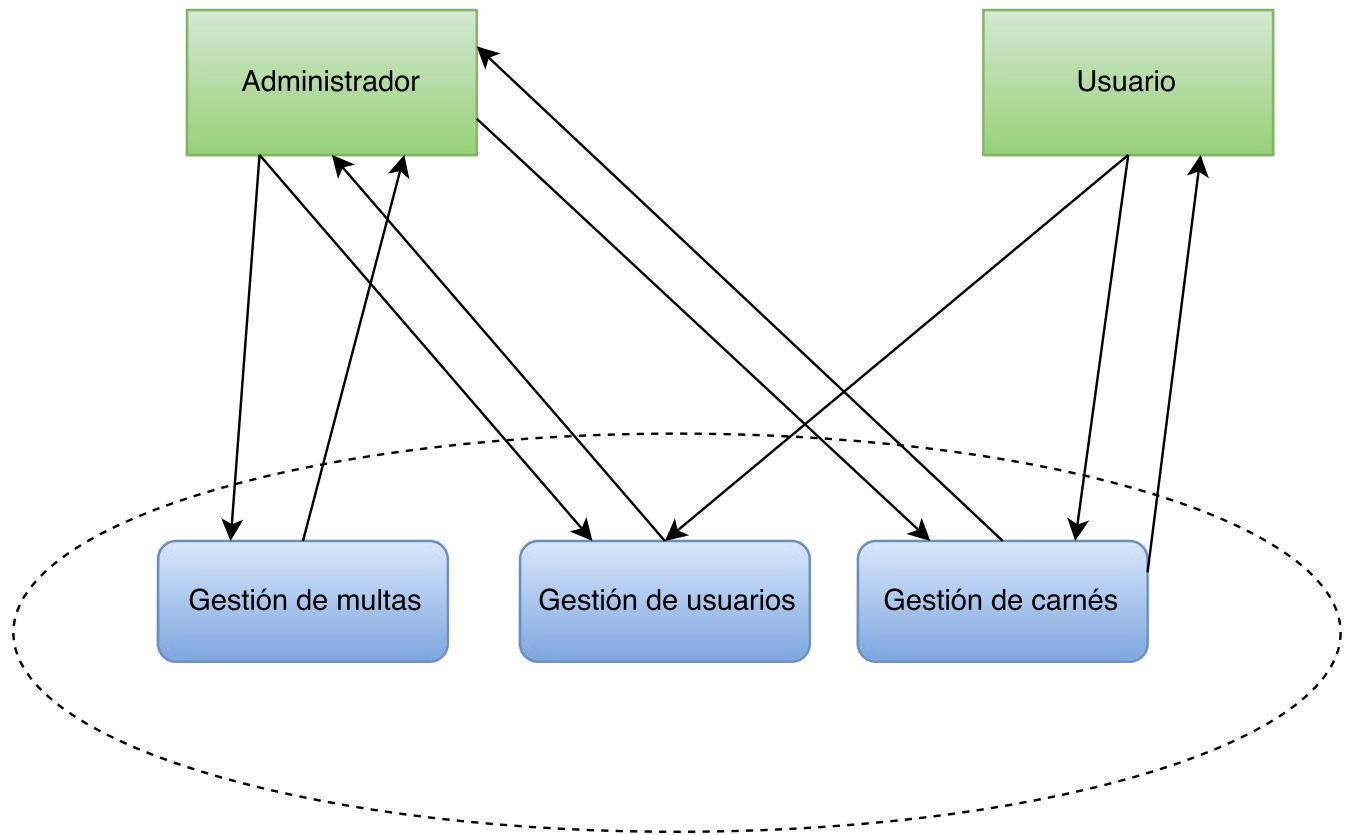
En este diagrama **refinamos** un poco más el esquema de caja negra y podemos diferenciar claramente los 4 subsistemas a los que nos vamos a dedicar.



## Refinamiento de cada subsistema

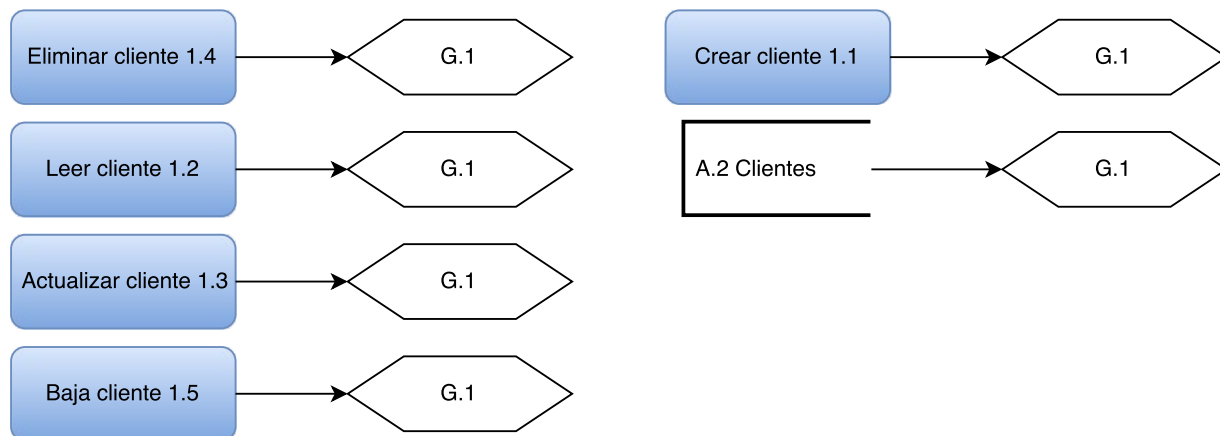
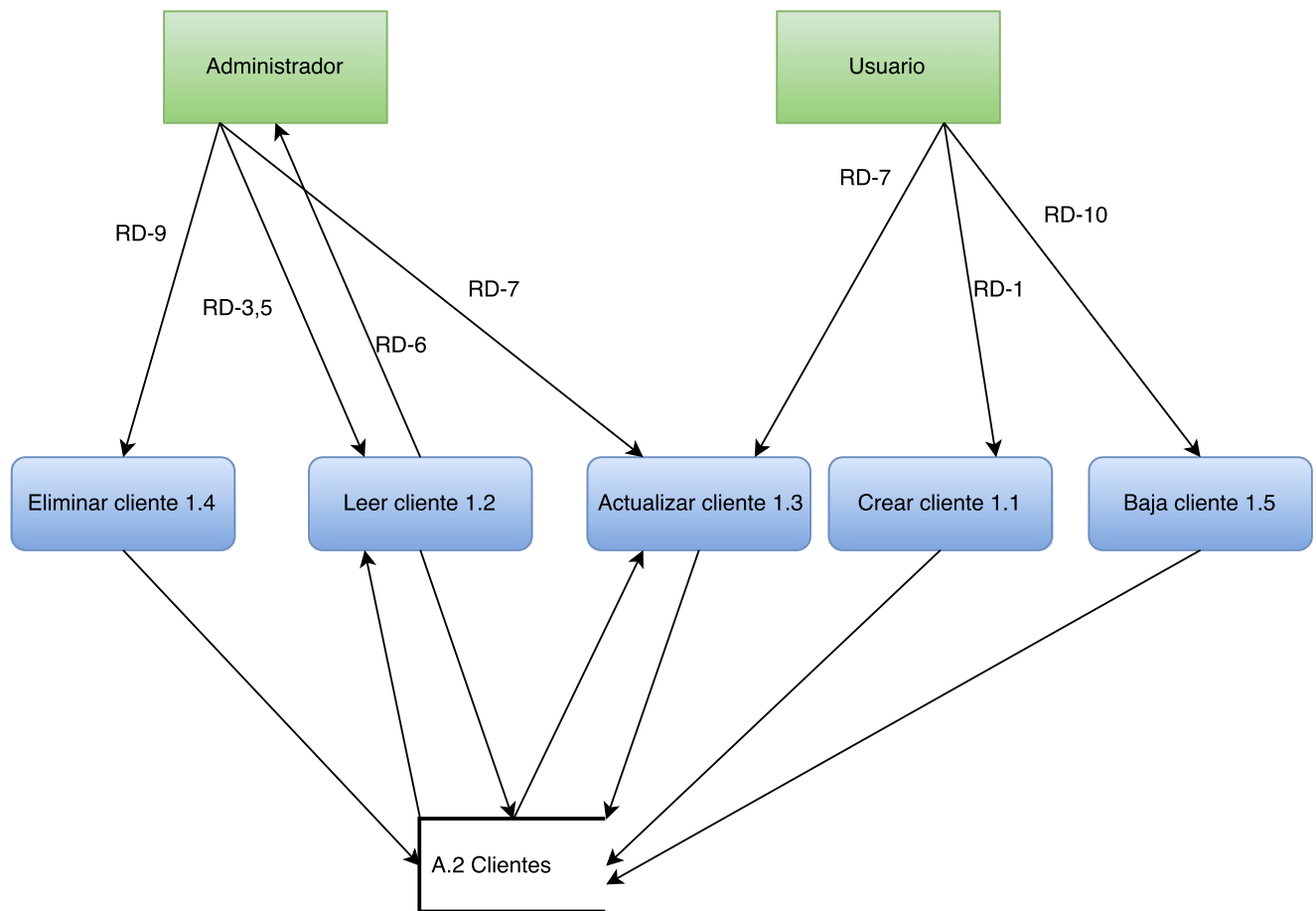
### Gestión de multas, clientes y carnés (Guillermo)

#### Nivel 1



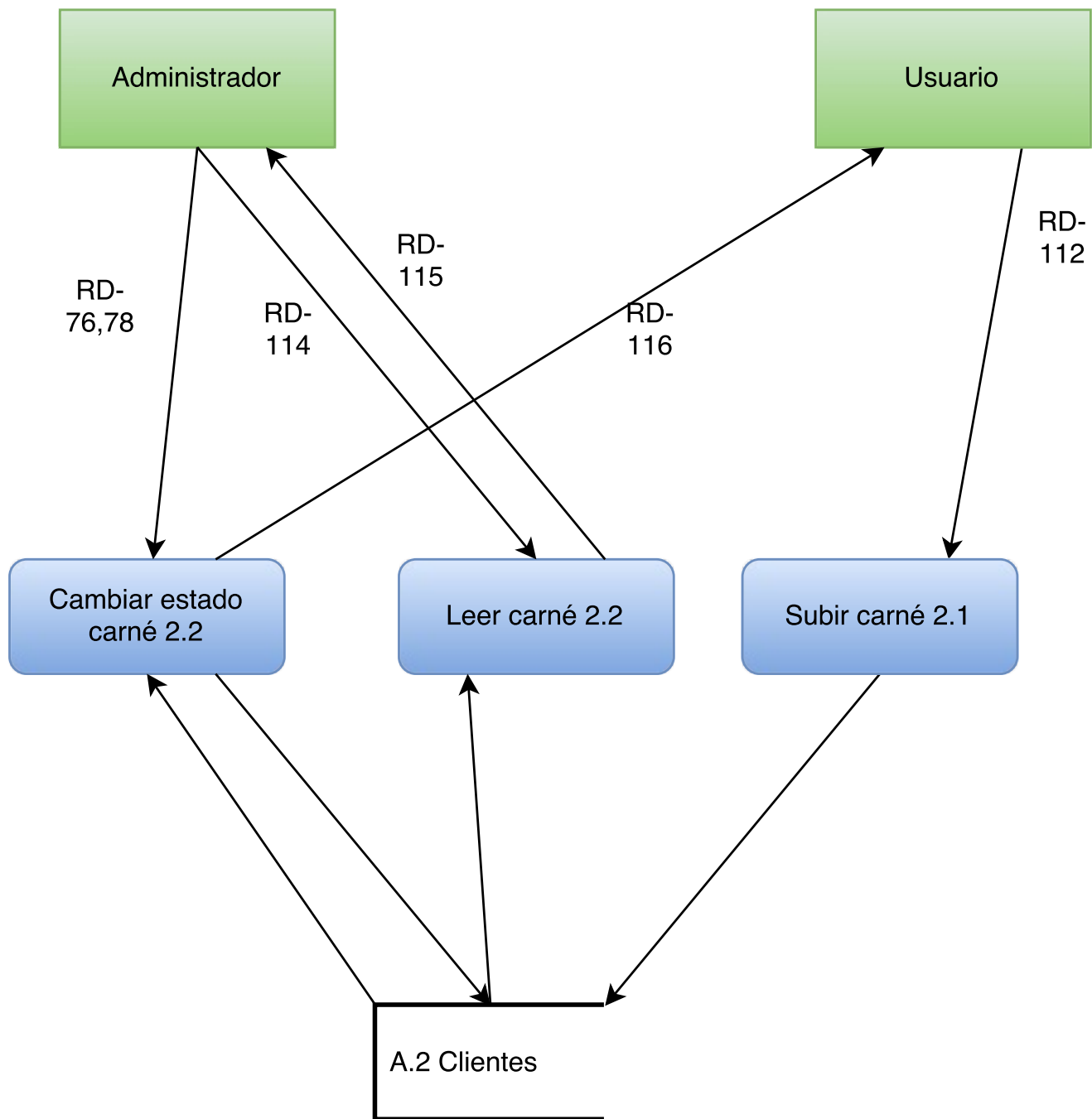
## Gestión de clientes

### Nivel 2



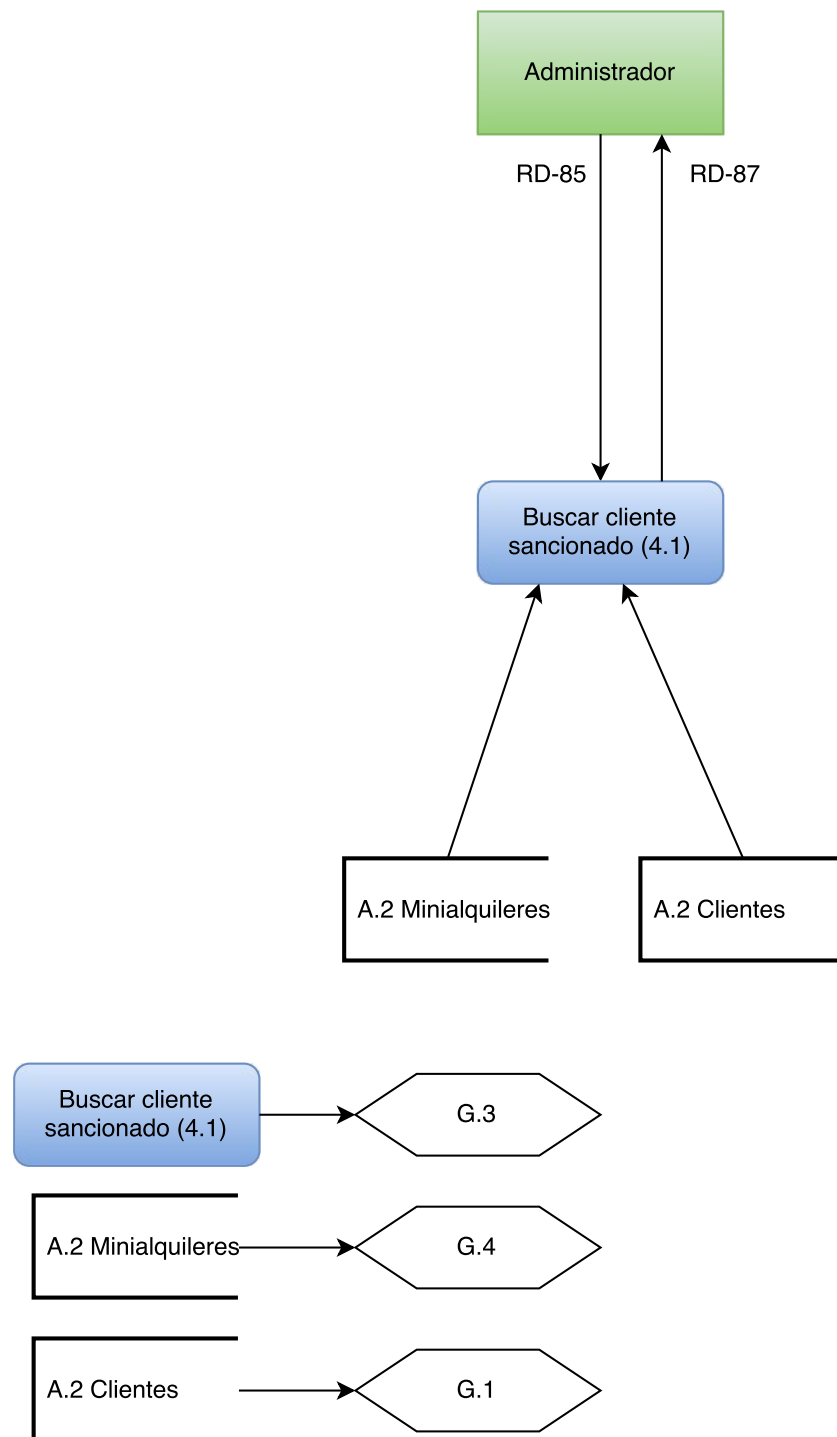
## Gestión de carnés (Guillermo)

### Nivel 2



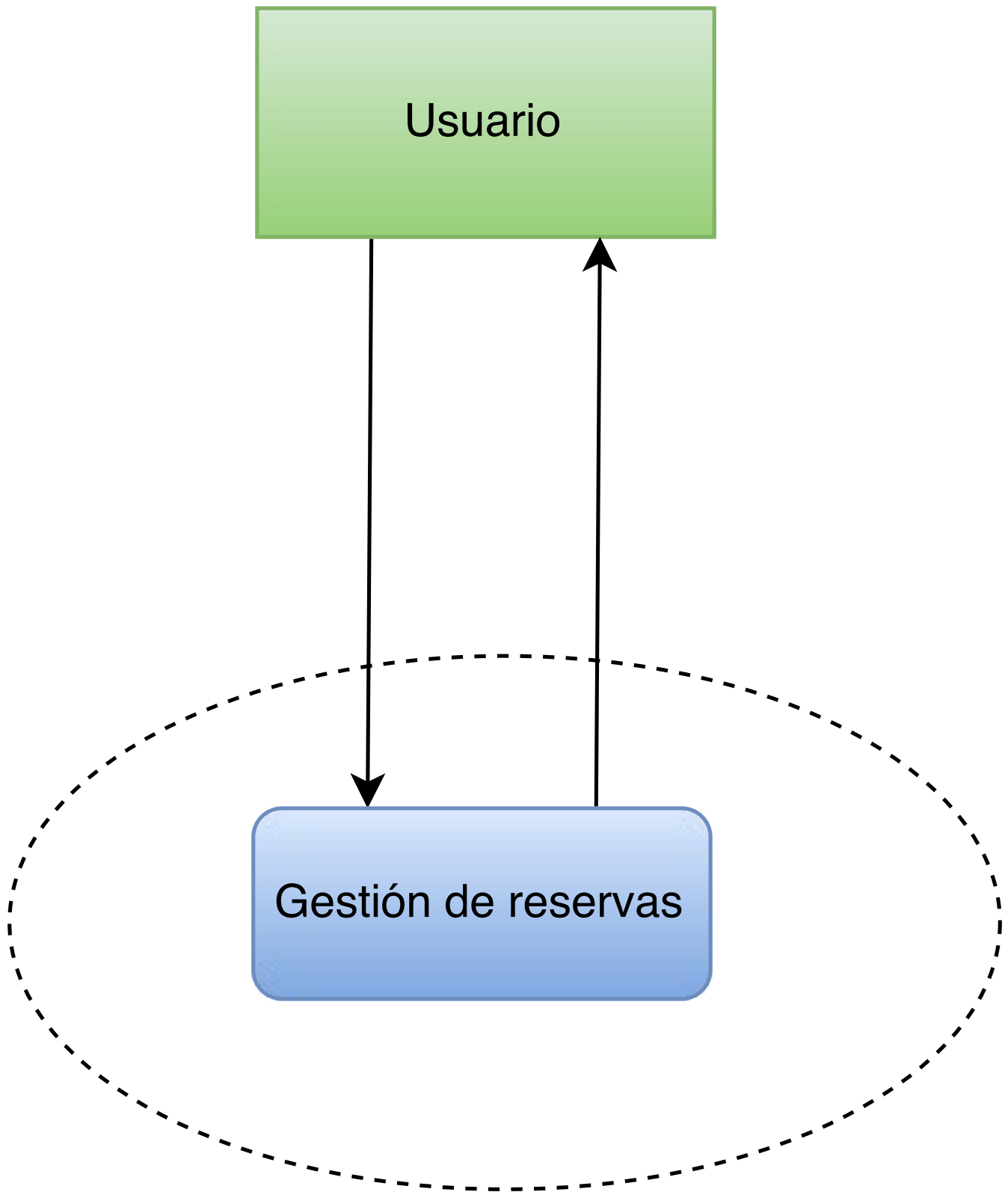
## Gestión de multas (Guillermo)

### Nivel 2



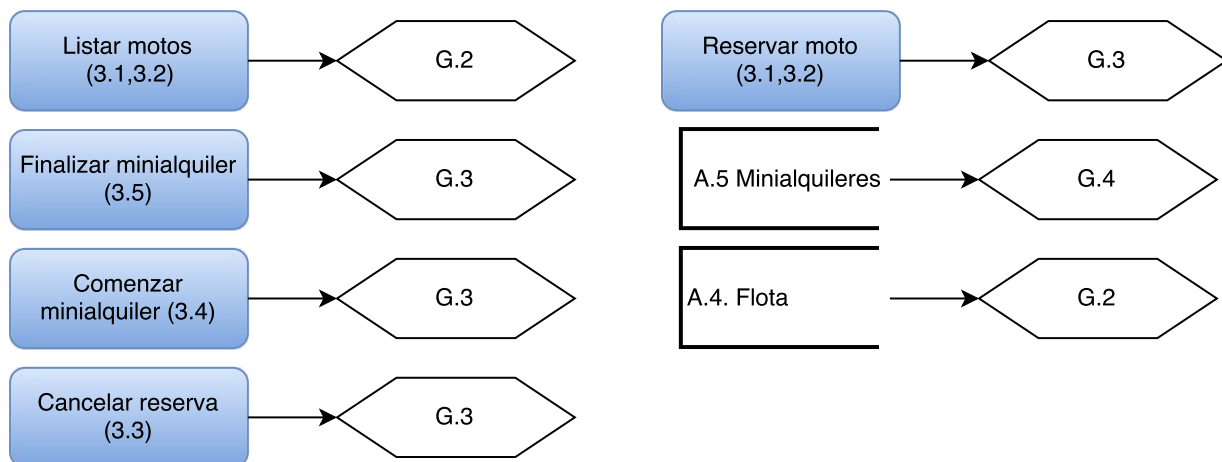
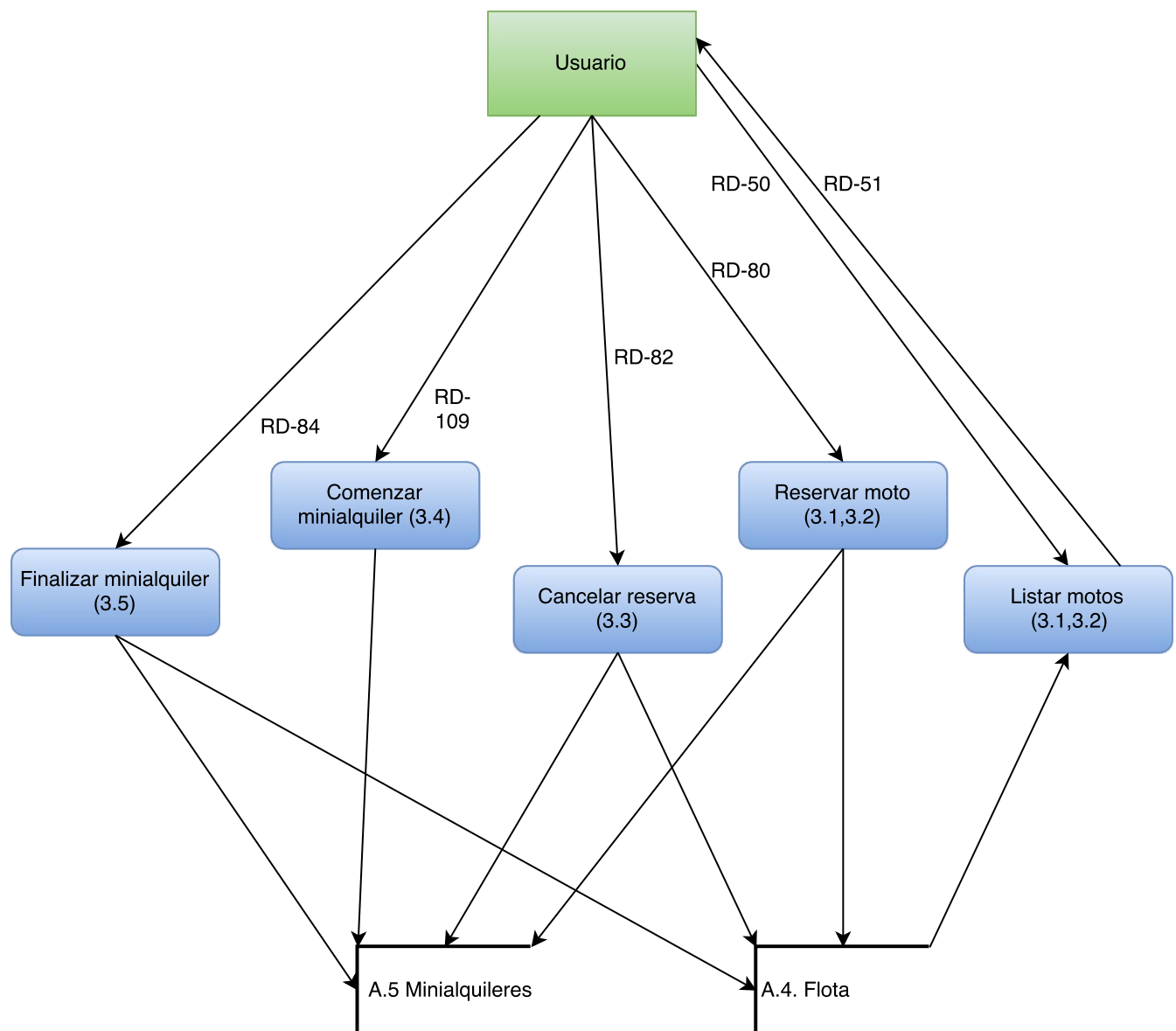
## Gestión de reservas (Guillermo)

Nivel 1



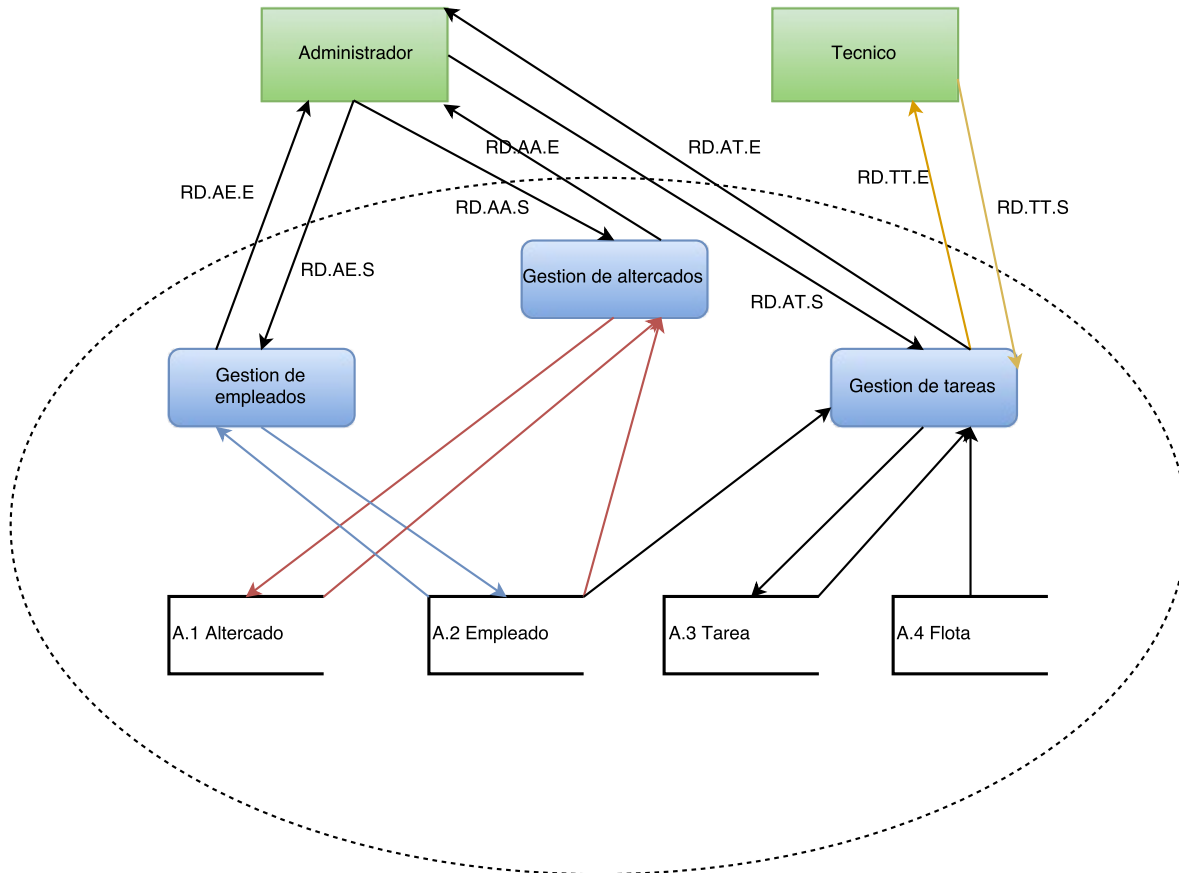


## Nivel 2



## Gestión de Recursos Humanos (Jose Antonio)

### Nivel 1



RD.AE.S = RD 12,14,17,18

RD.AE.E = RD 16

RD.AA.S = RD 28,30,33

RD.AA.E = RD 32

RD.AT.S = RD 20,22,25,26

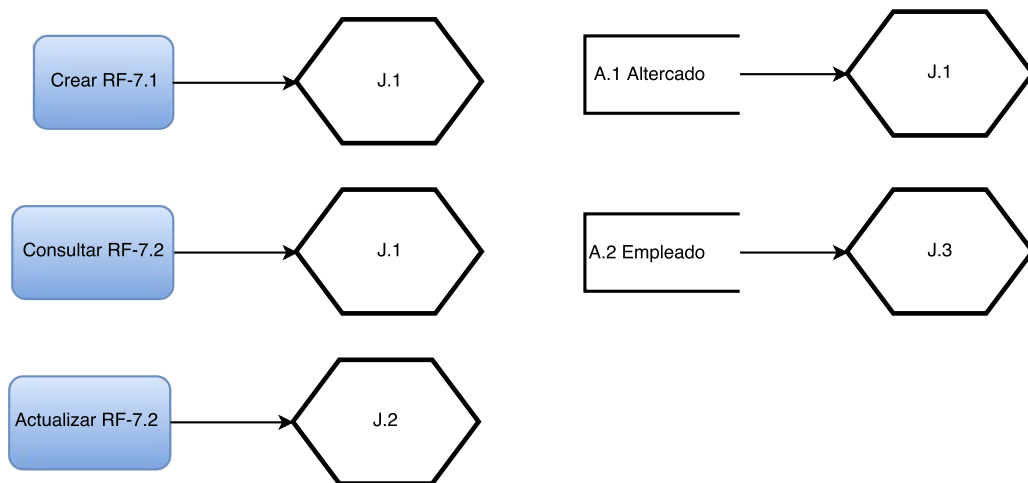
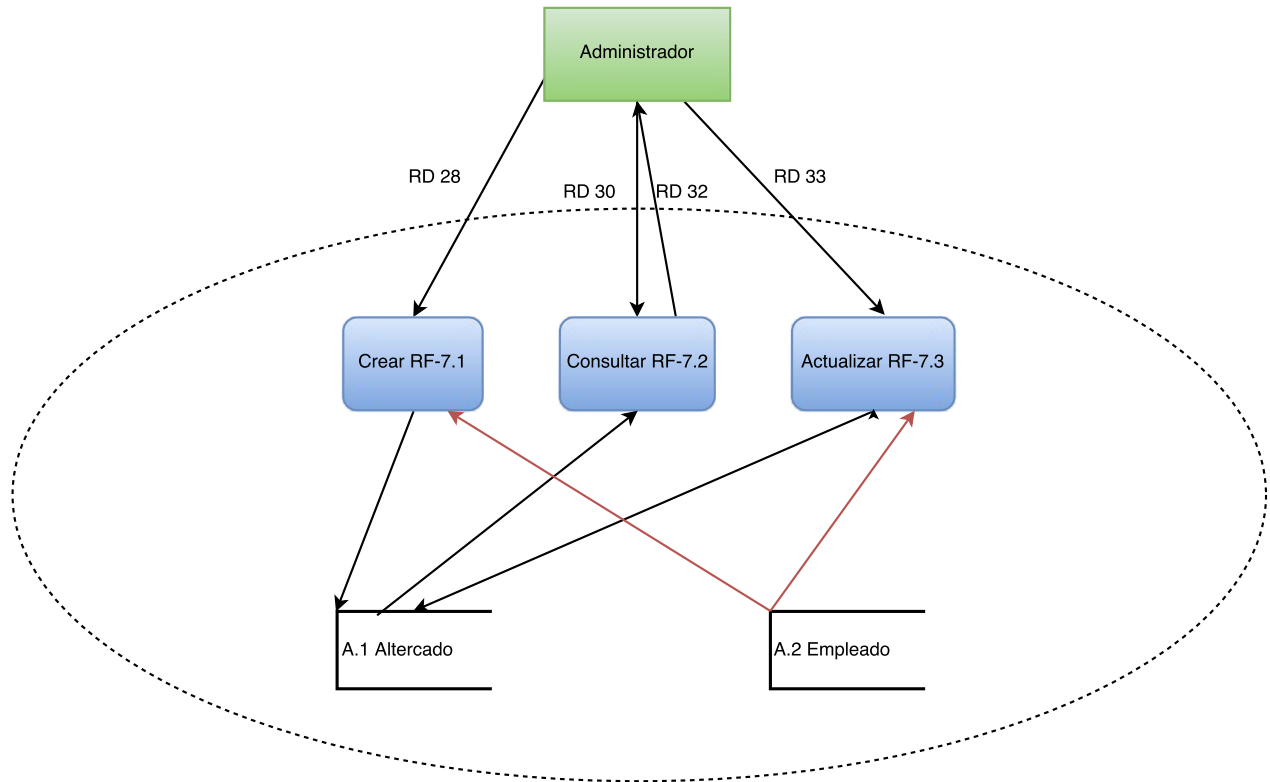
RD.AT.E = RD 24

RD.TT.S = RD 22,25

RD.TT.E = RD 24

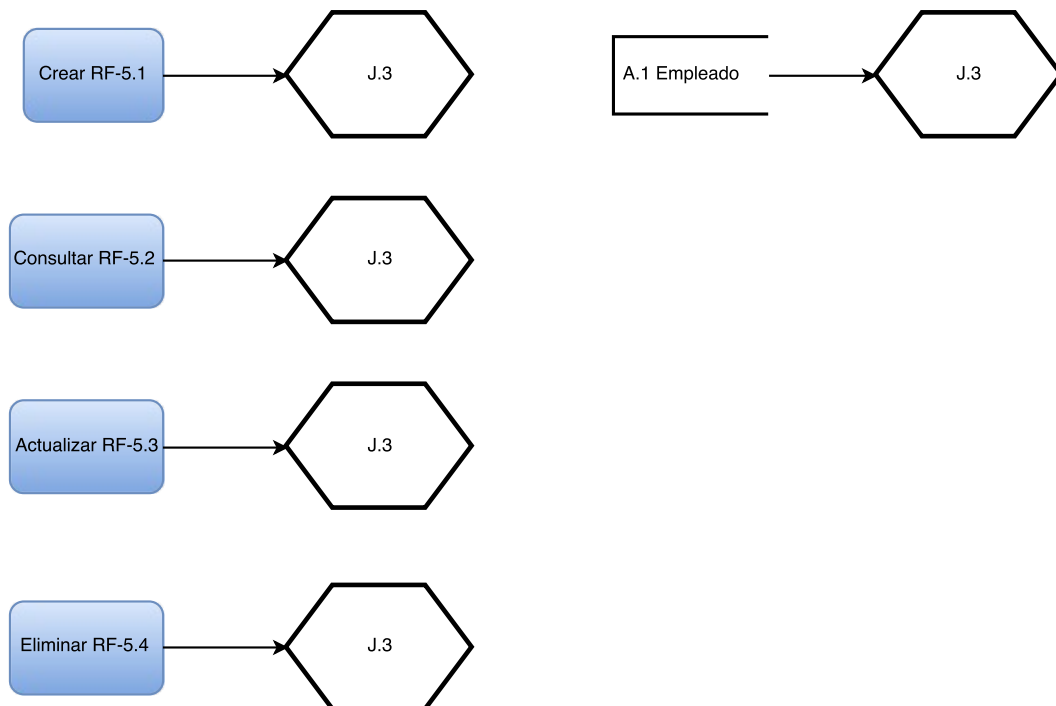
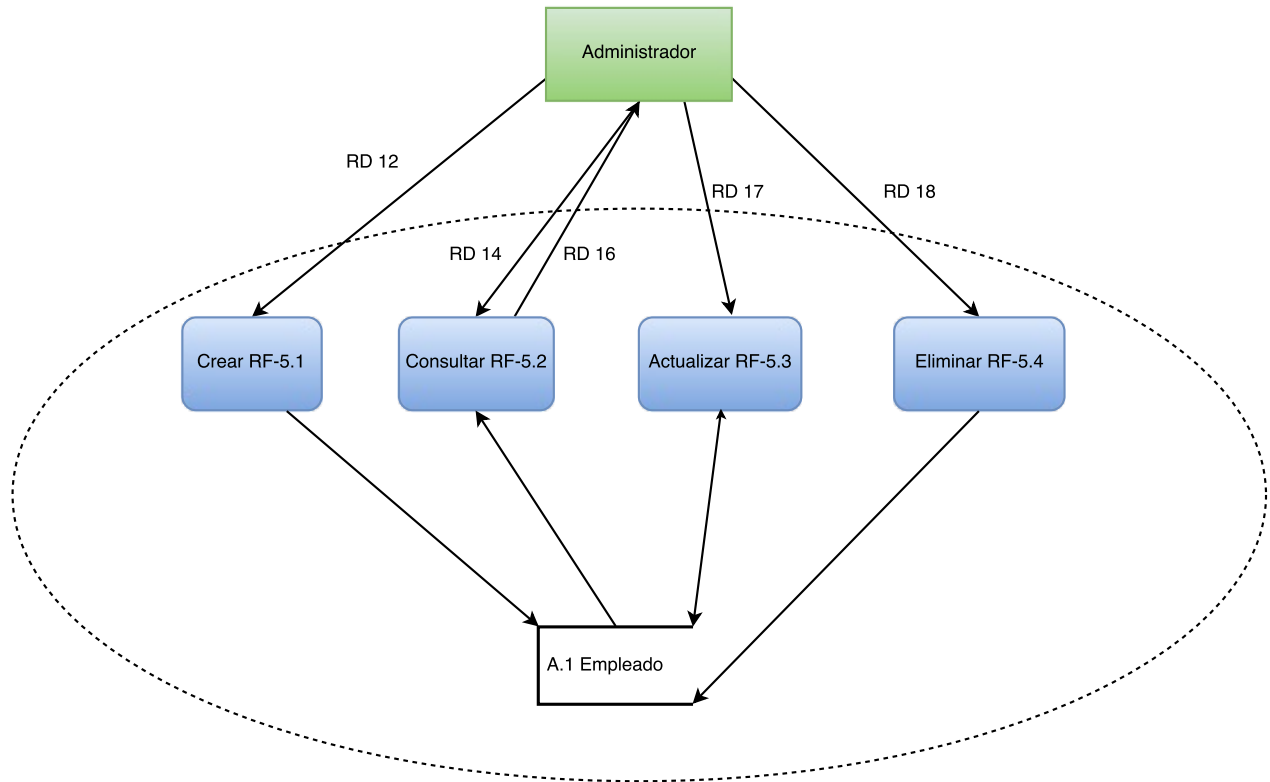
## Gestión de altercados (Jose Antonio)

### Nivel 2



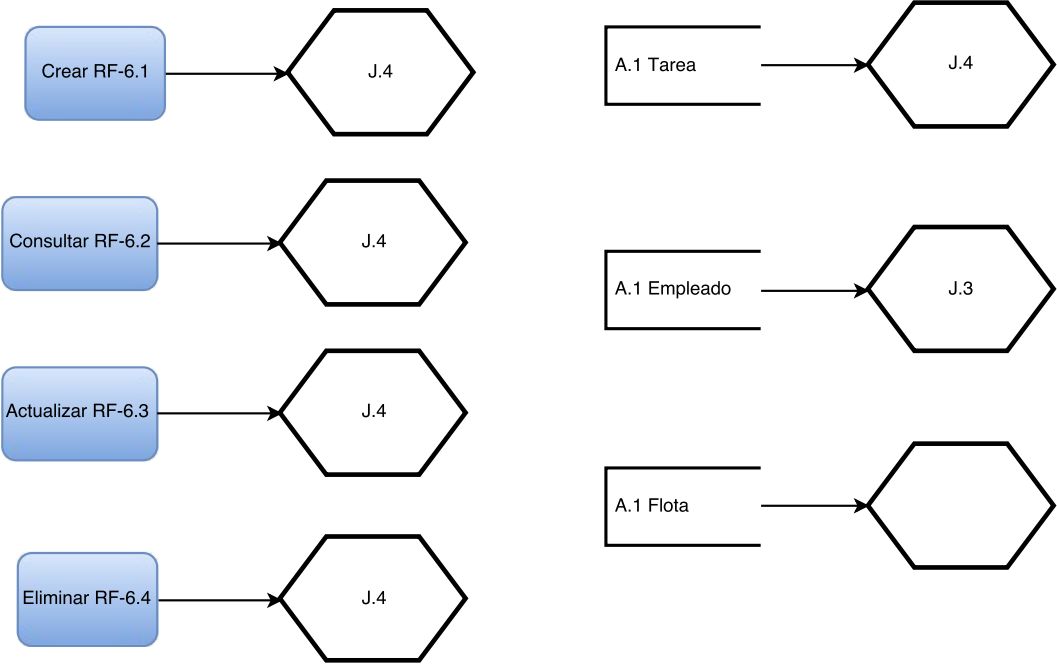
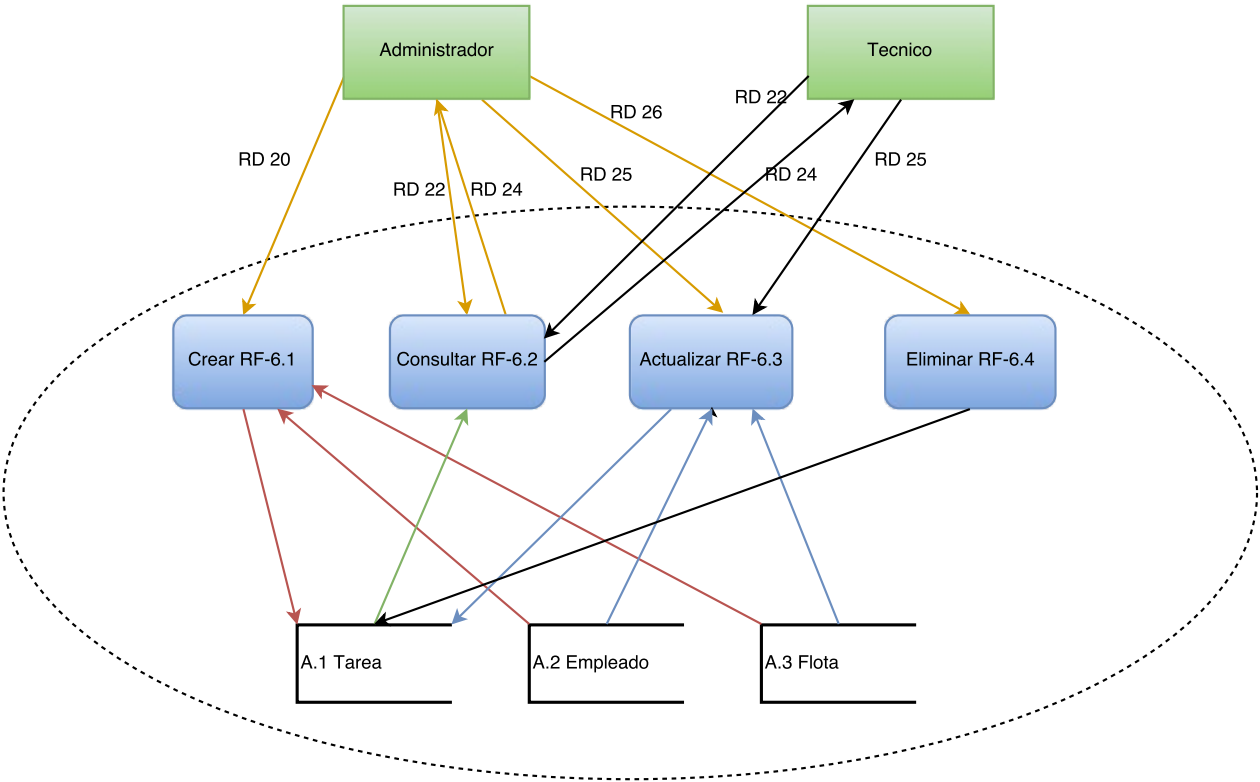
## Gestión de empleados (Jose Antonio)

### Nivel 2



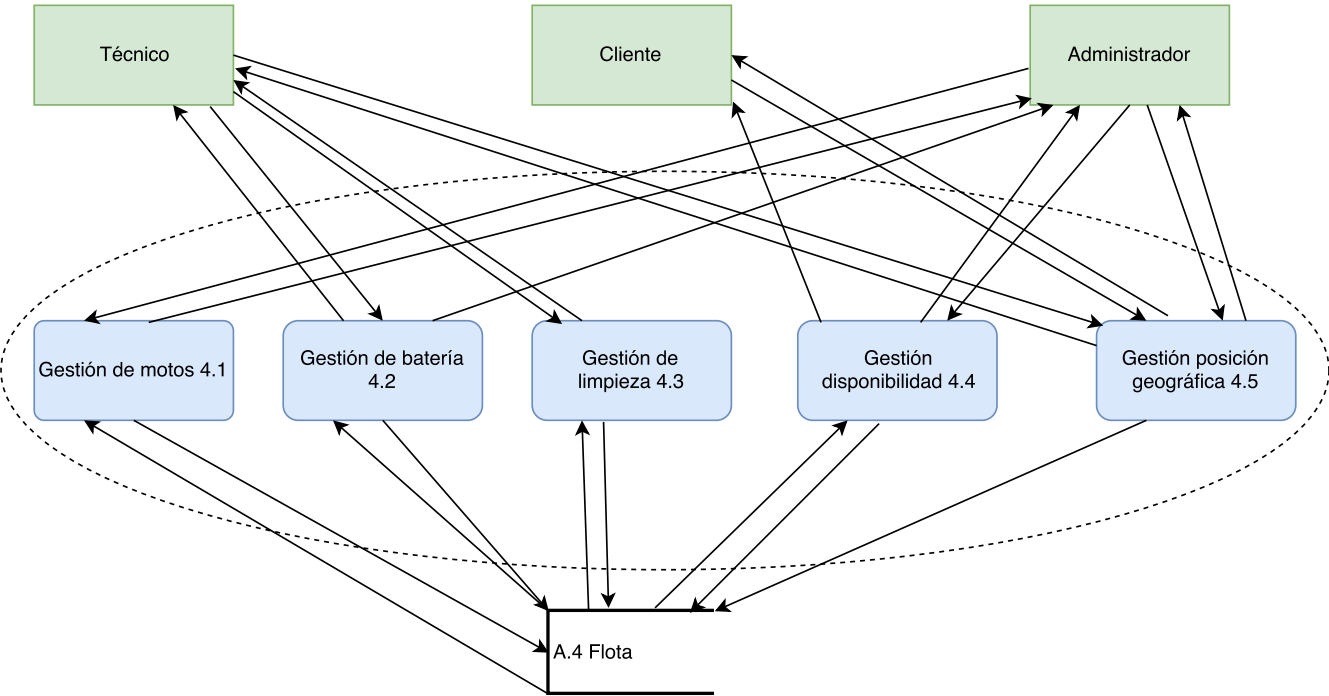
Gestión de tareas (Jose Antonio)

Nivel 2



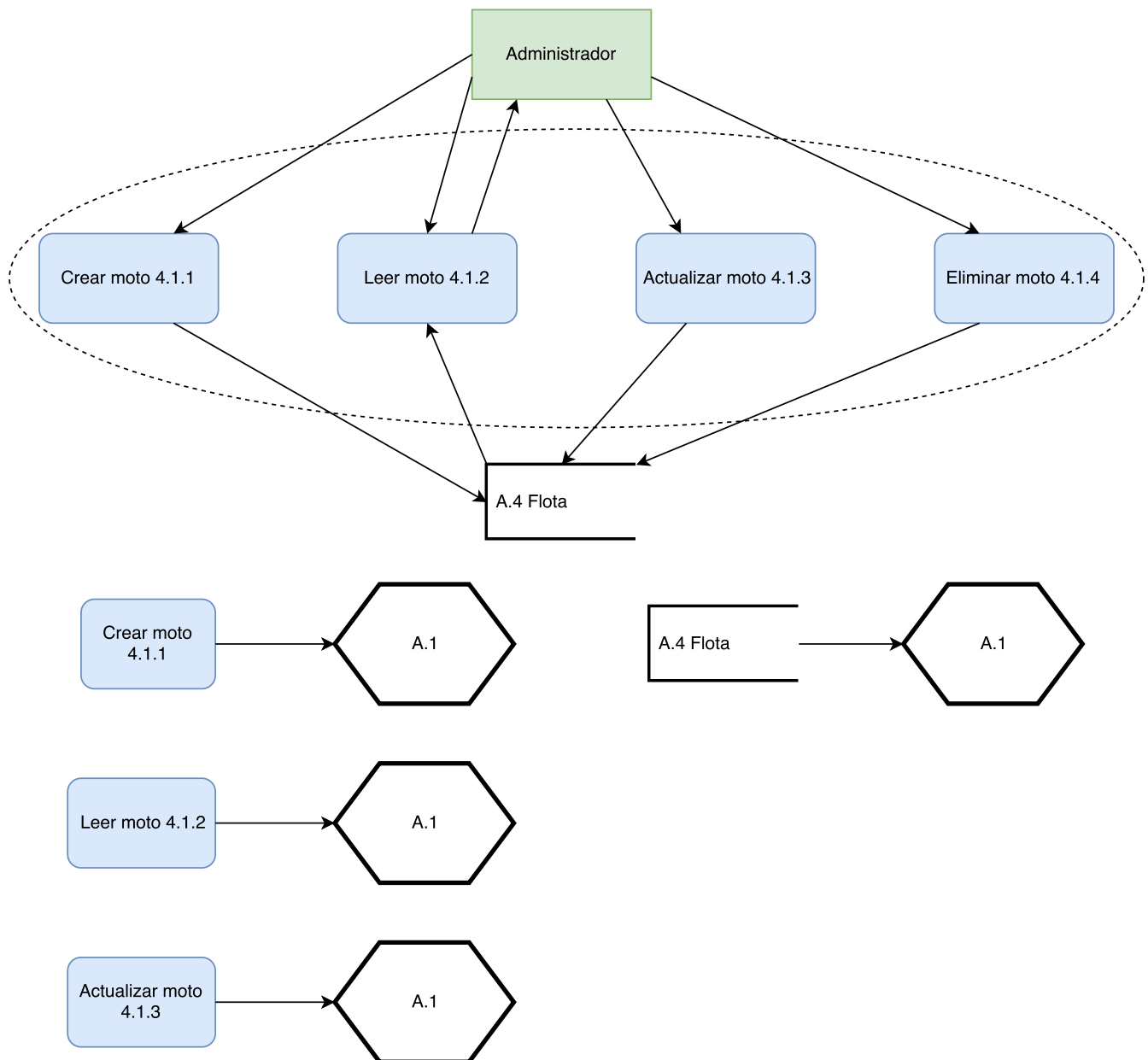
Gestión de flota (Adrián)

Nivel 1



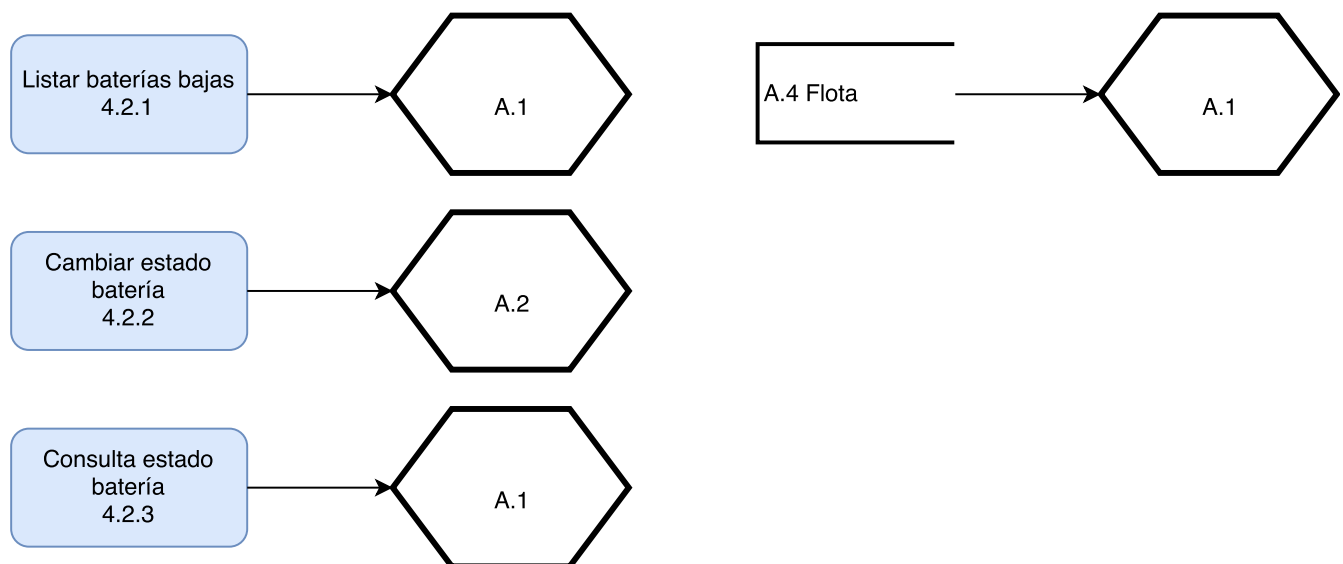
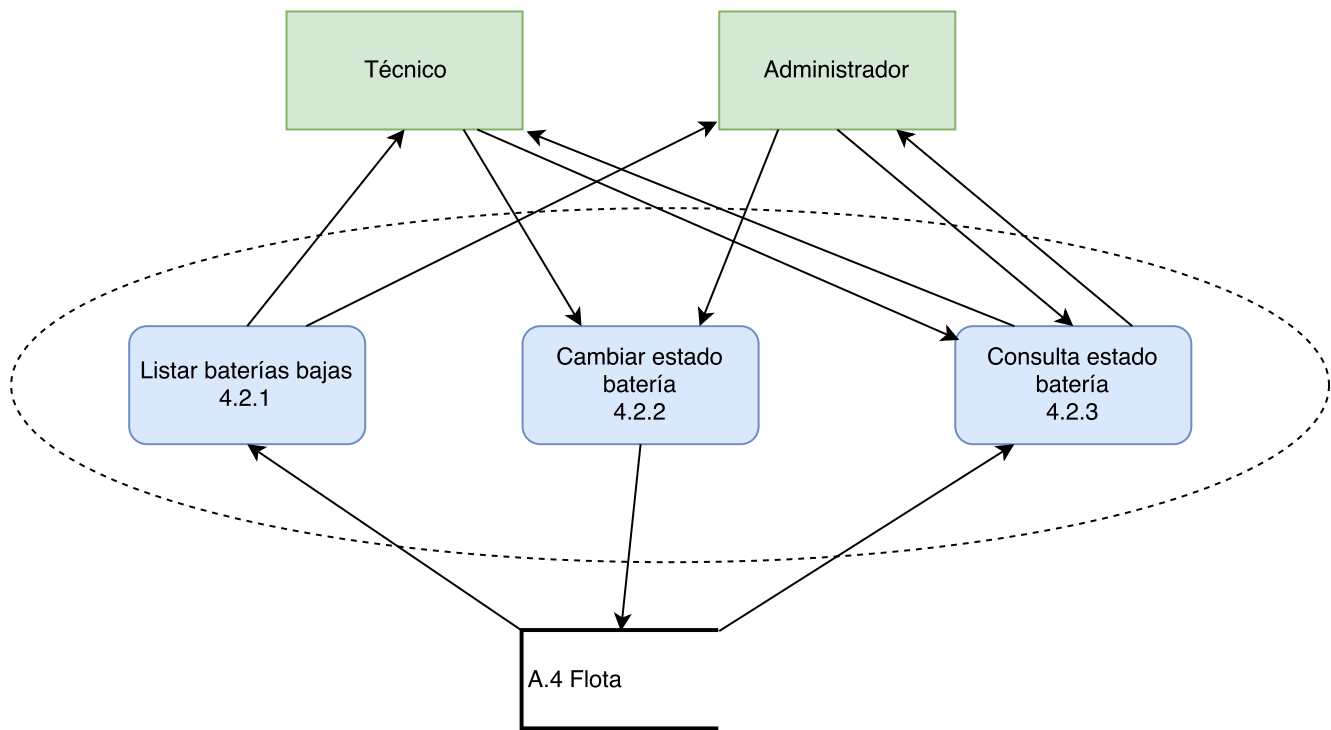
## Gestión de motos (Adrián)

### Nivel 2



## Gestión de batería (Adrián)

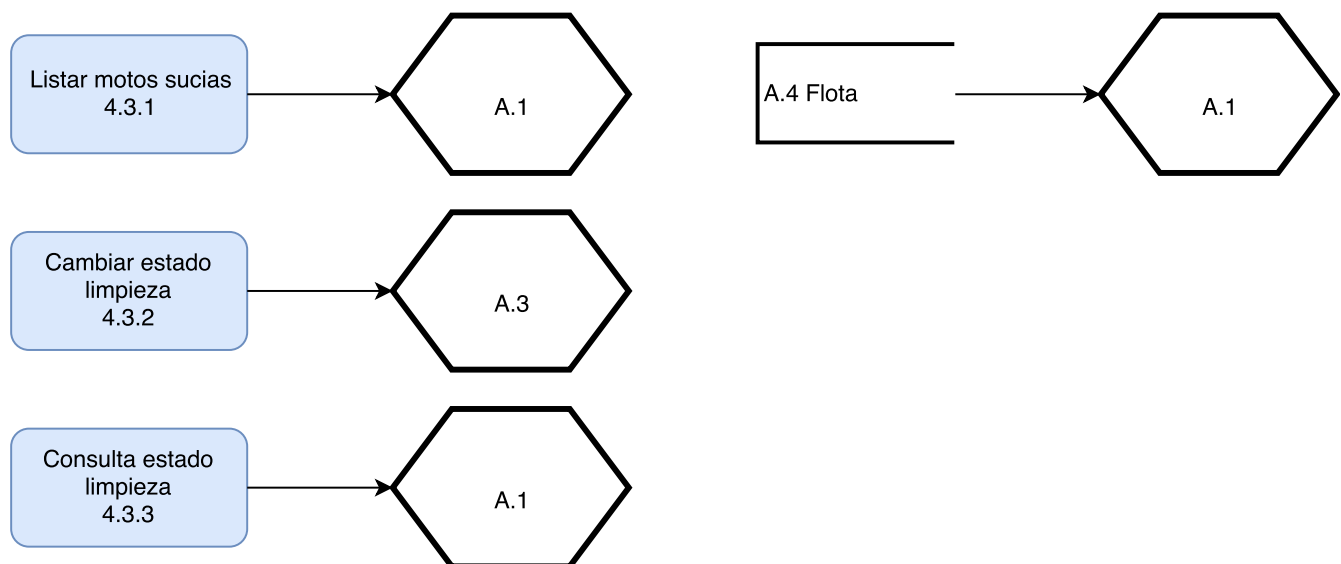
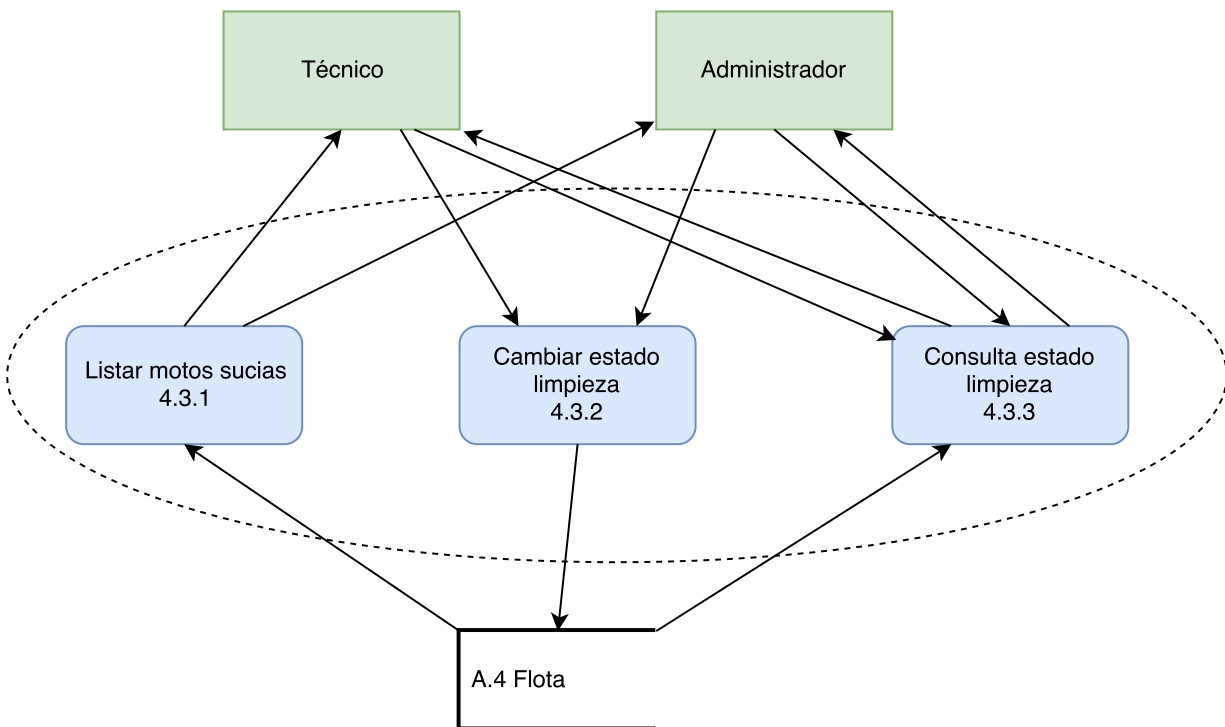
### Nivel 2





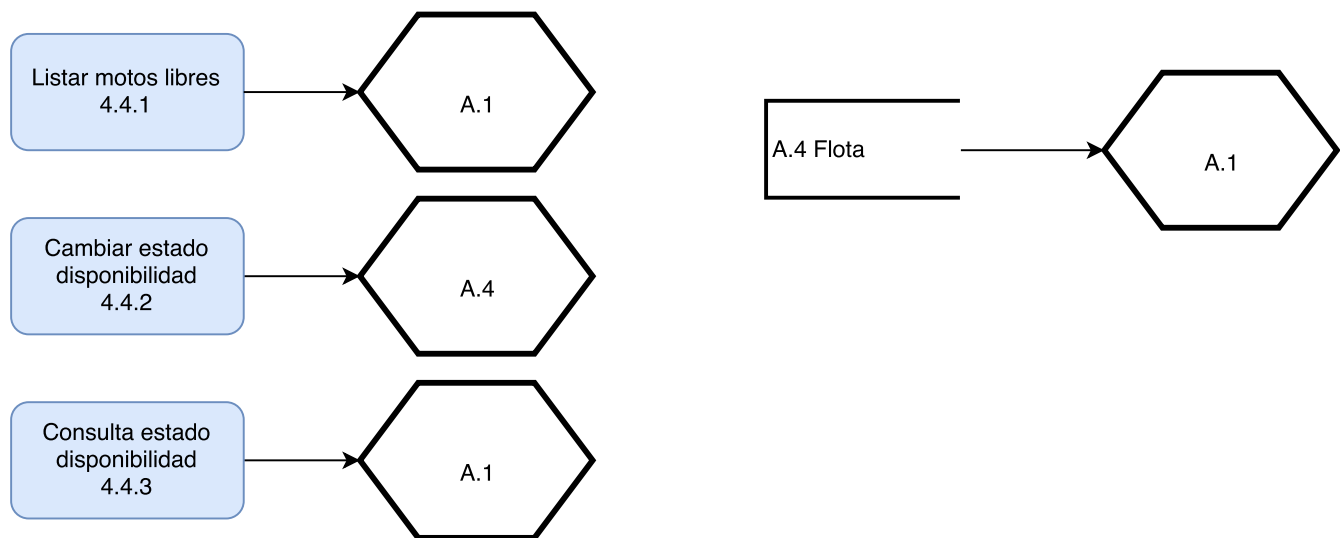
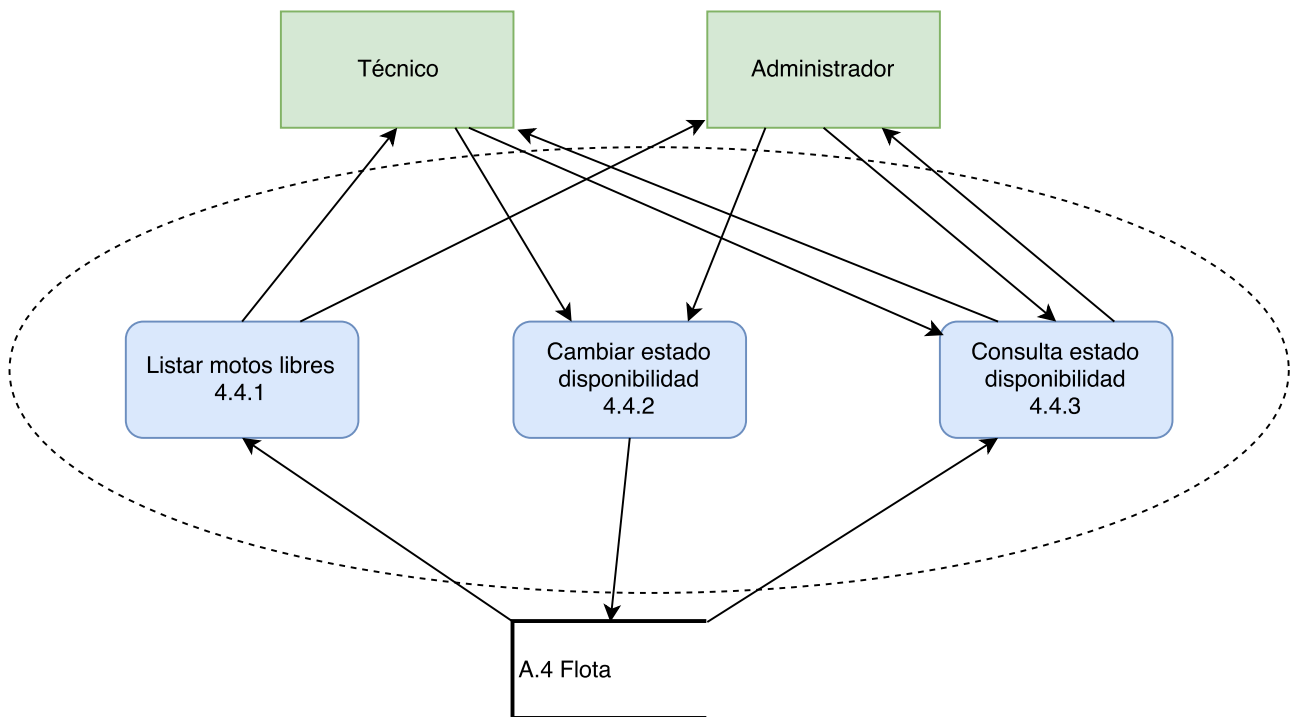
## Gestión de limpieza (Adrián)

### Nivel 2



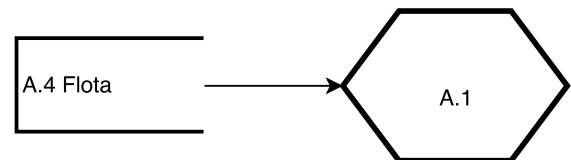
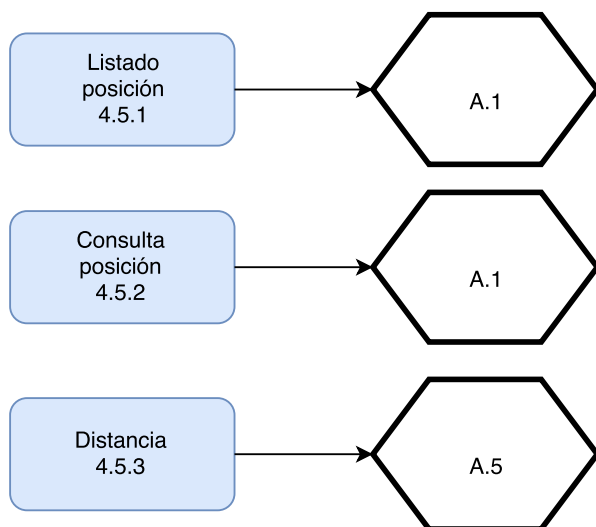
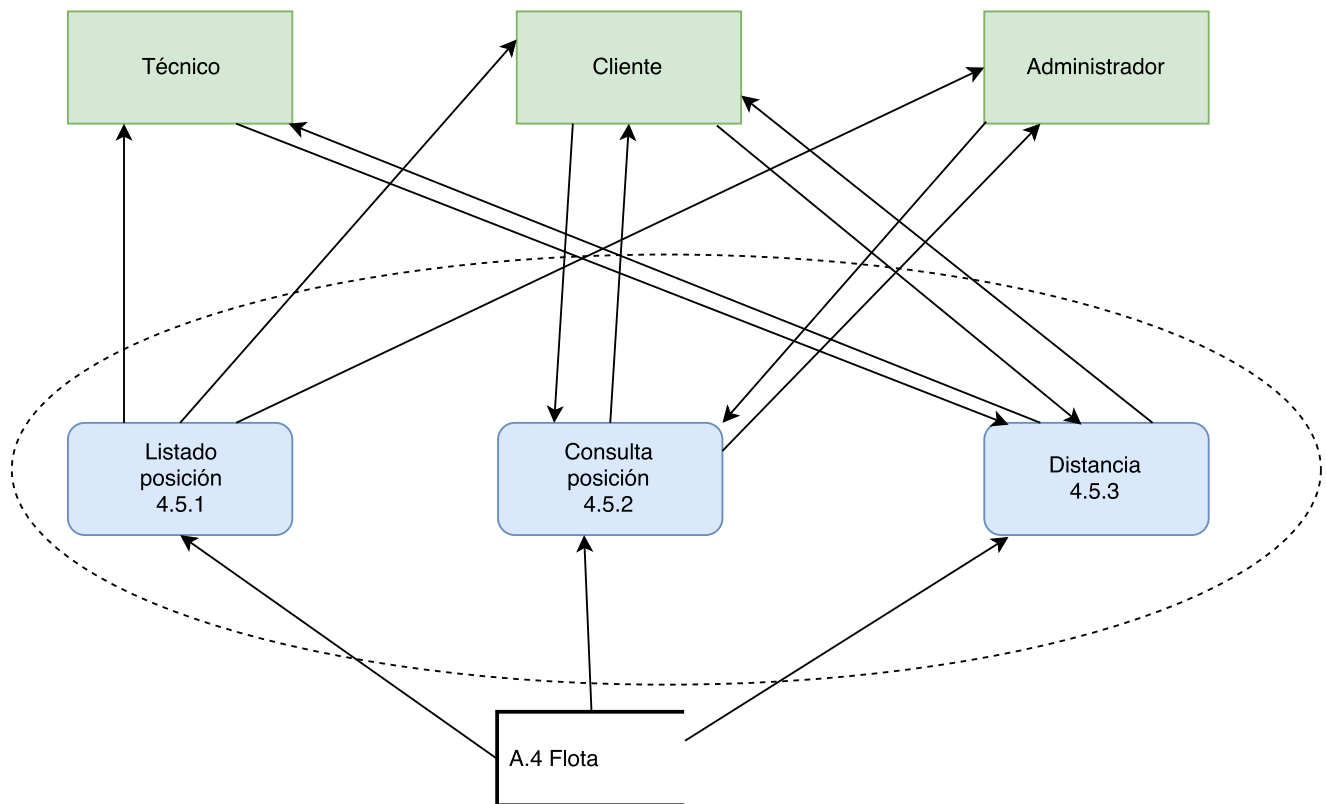
## Gestión de disponibilidad (Adrián)

### Nivel 2



## Gestión de posición geográfica (Adrián)

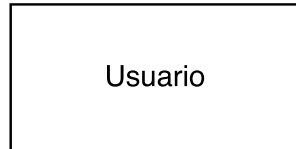
### Nivel 2



## Esquemas externos de procesos y almacenes al máximo nivel de refinamiento

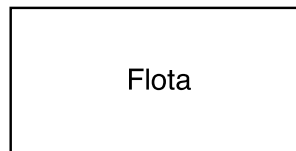
### Esquema externo G.1

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



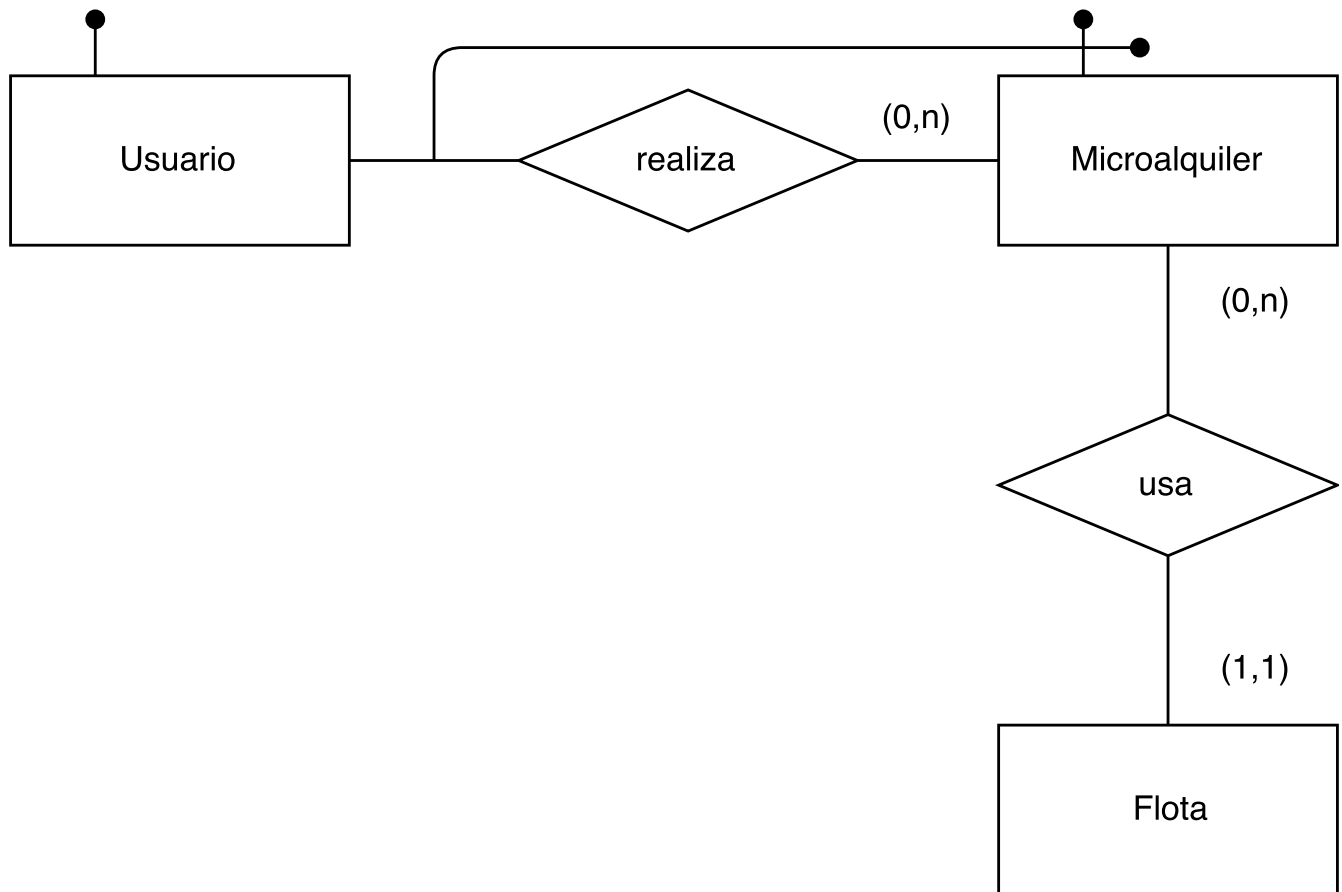
### Esquema externo G.2

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



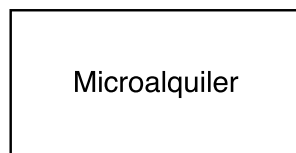
### Esquema externo G.3

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



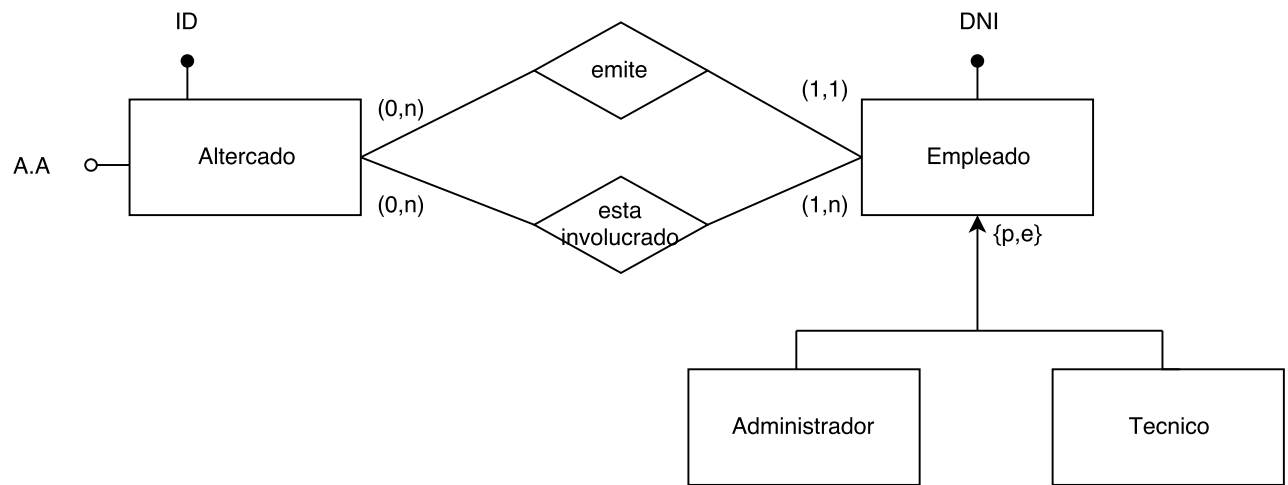
#### Esquema externo G.4

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



#### Esquema externo J.1

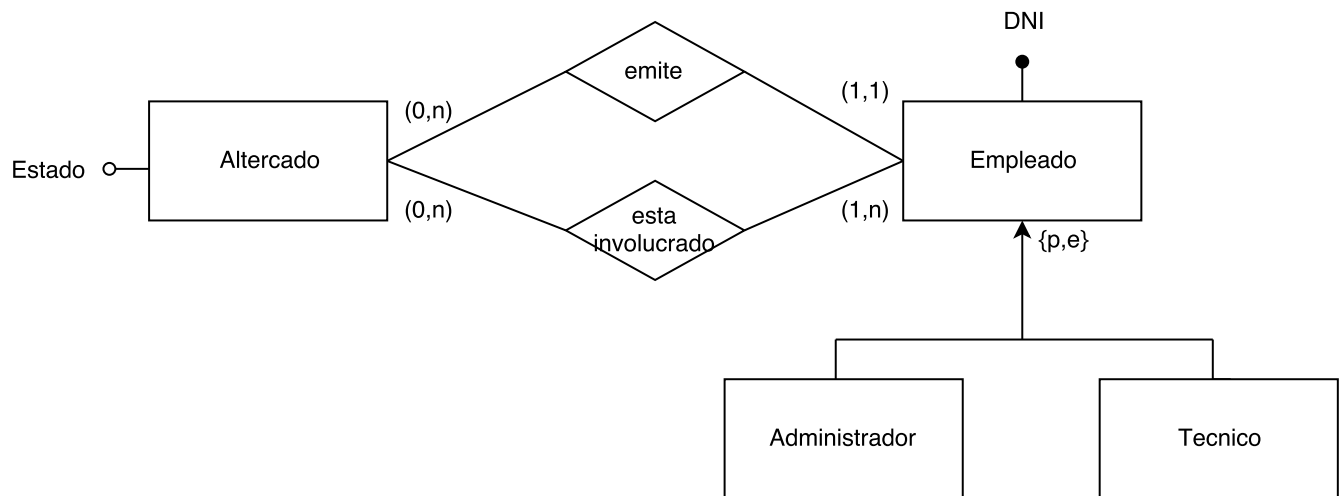
Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



A.A = Descripción, Estado (en proceso, parado, resuelto)

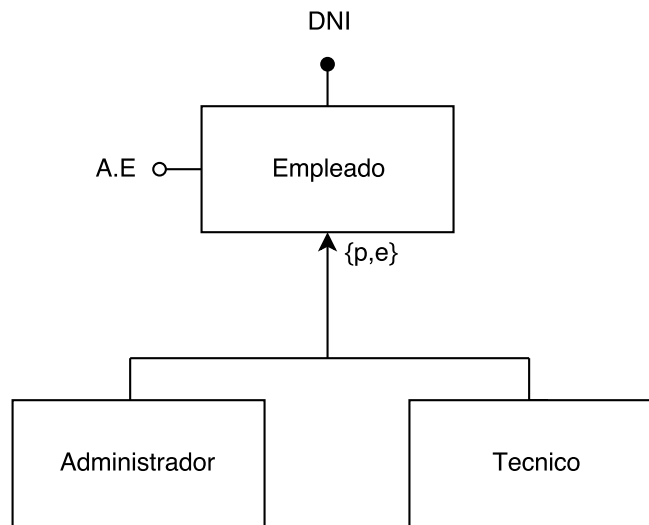
## Esquema externo J.2

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



## Esquema externo J.3

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.

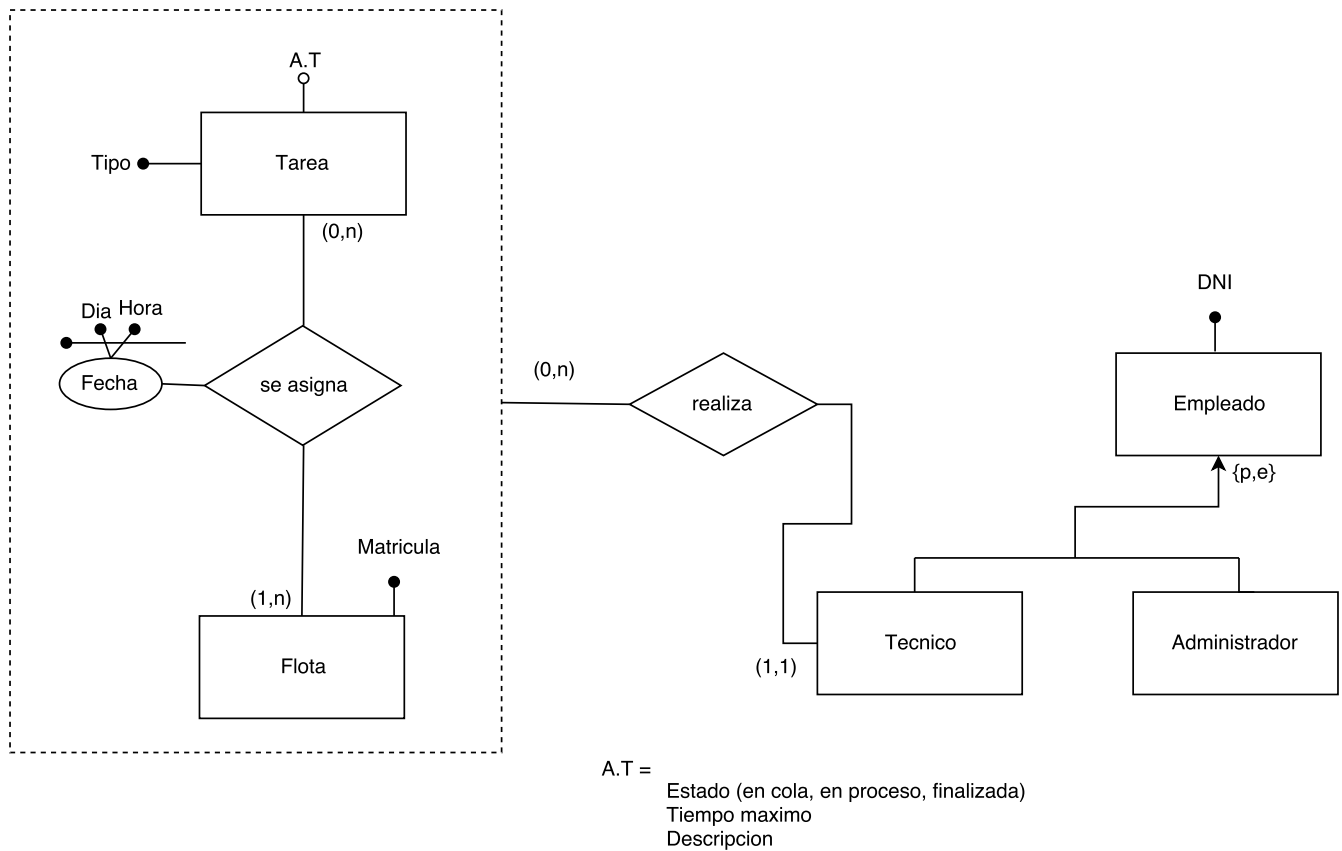


A.E =

Nombre y apellidos.  
Clave de acceso  
telefono  
email  
fecha de nacimiento  
cuenta corriente  
alta y baja del contrato  
tipo de contrato  
salario  
estado (activo o de baja)  
puesto de trabajo  
evaluación

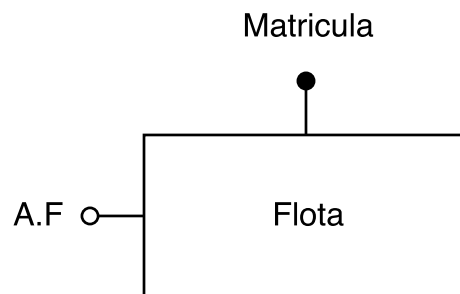
## Esquema externo J.4

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



### Esquema externo A.1

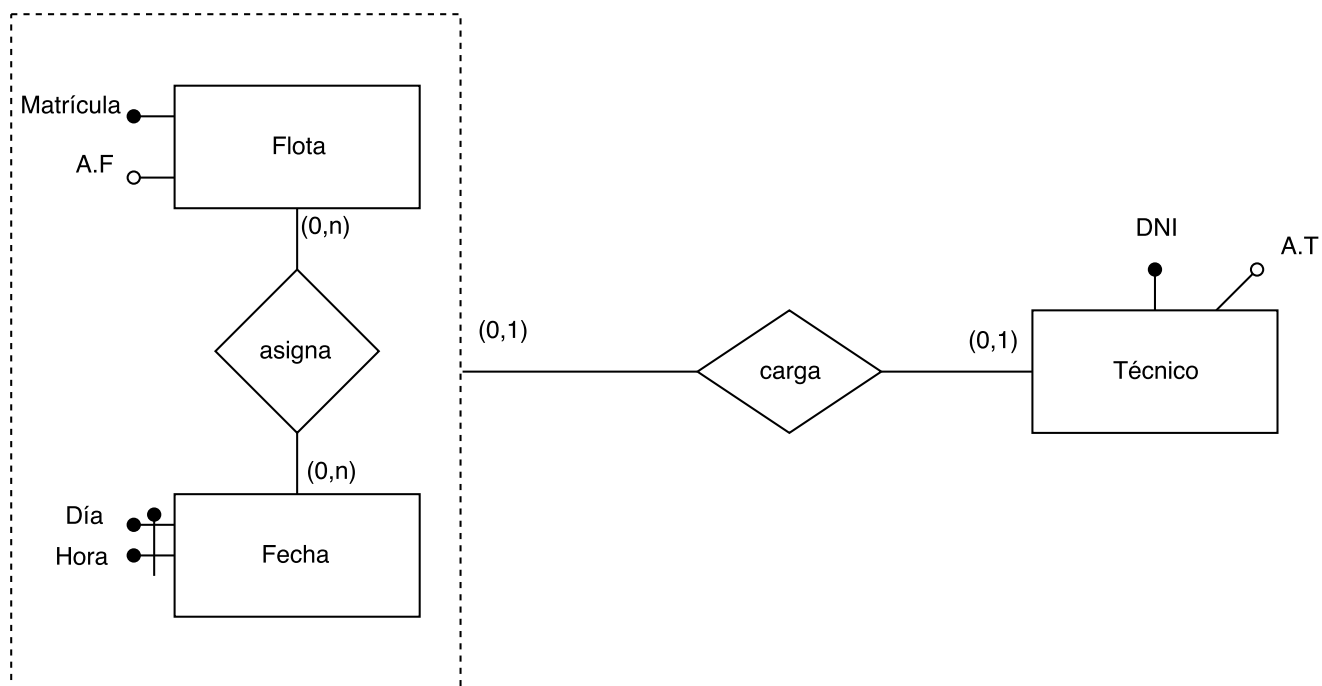
Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



### Esquema externo A.2

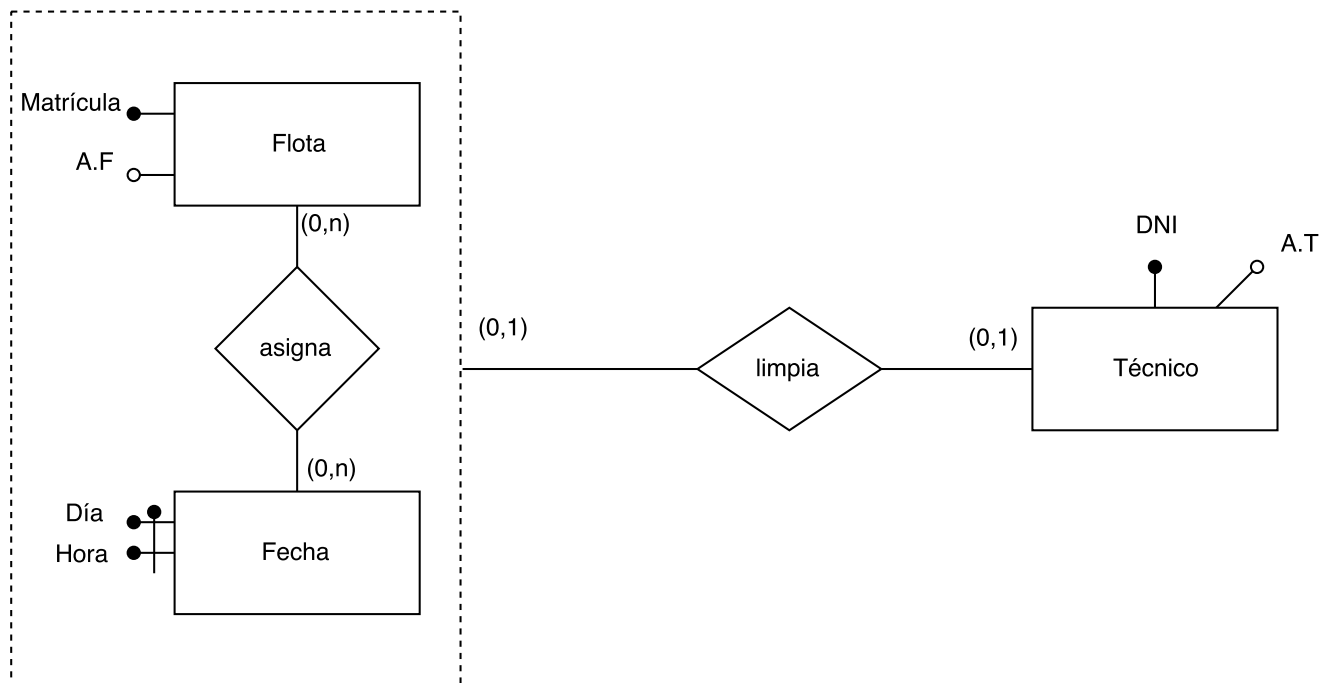
Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.





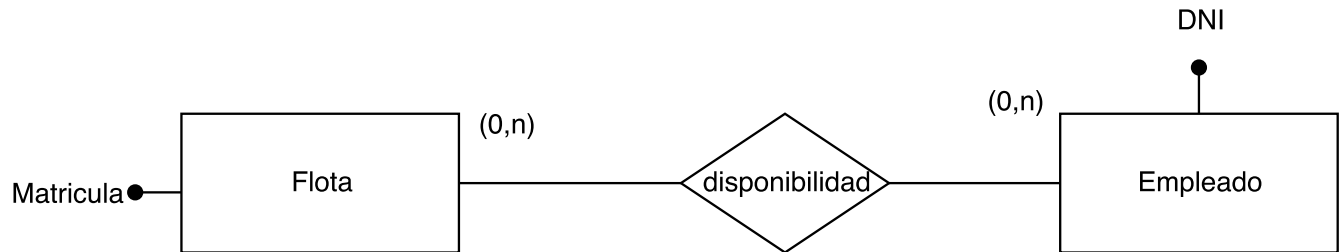
### Esquema externo A.3

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



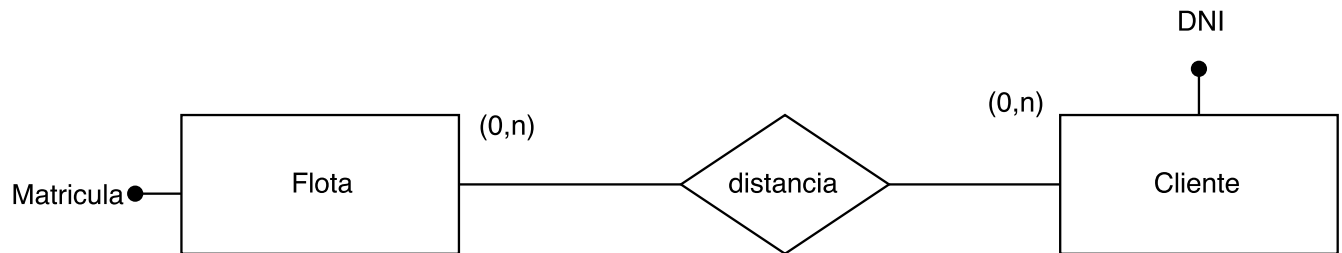
### Esquema externo A.4

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.



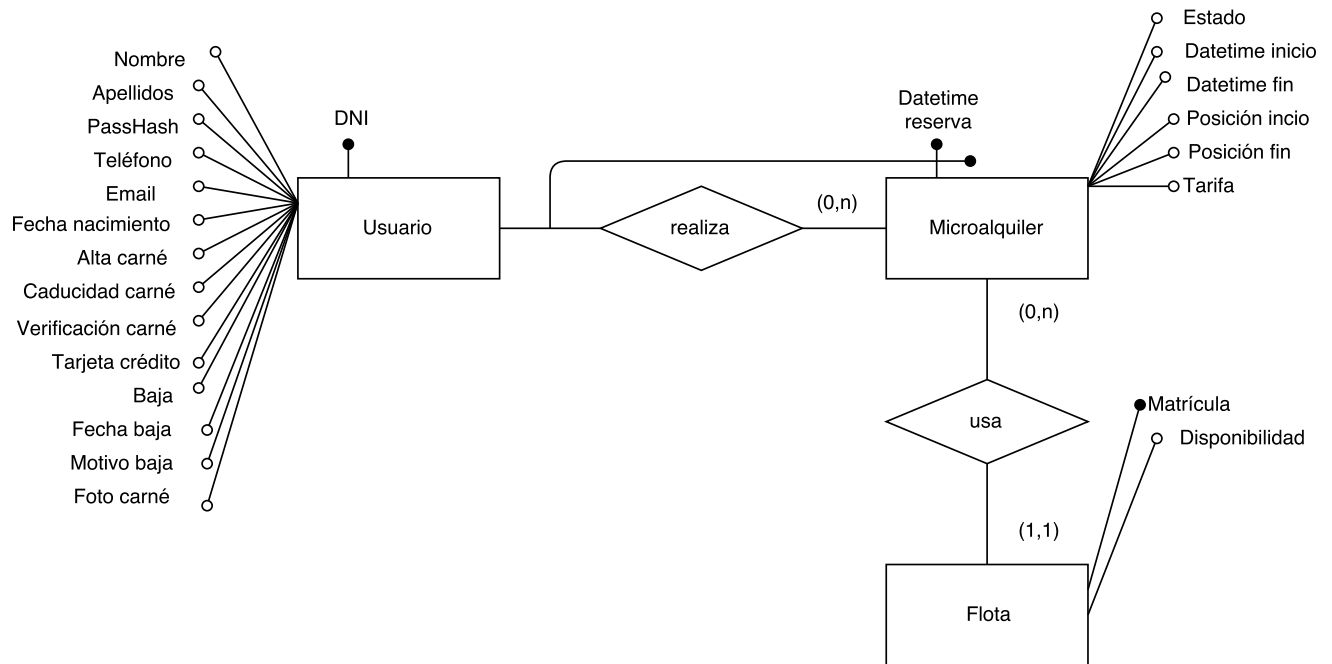
### Esquema externo A.5

Por simplicidad las entidades no tienen atributos y sólo muestran sus relaciones. Éstos se encuentran en el esquema externo del subsistema completo.

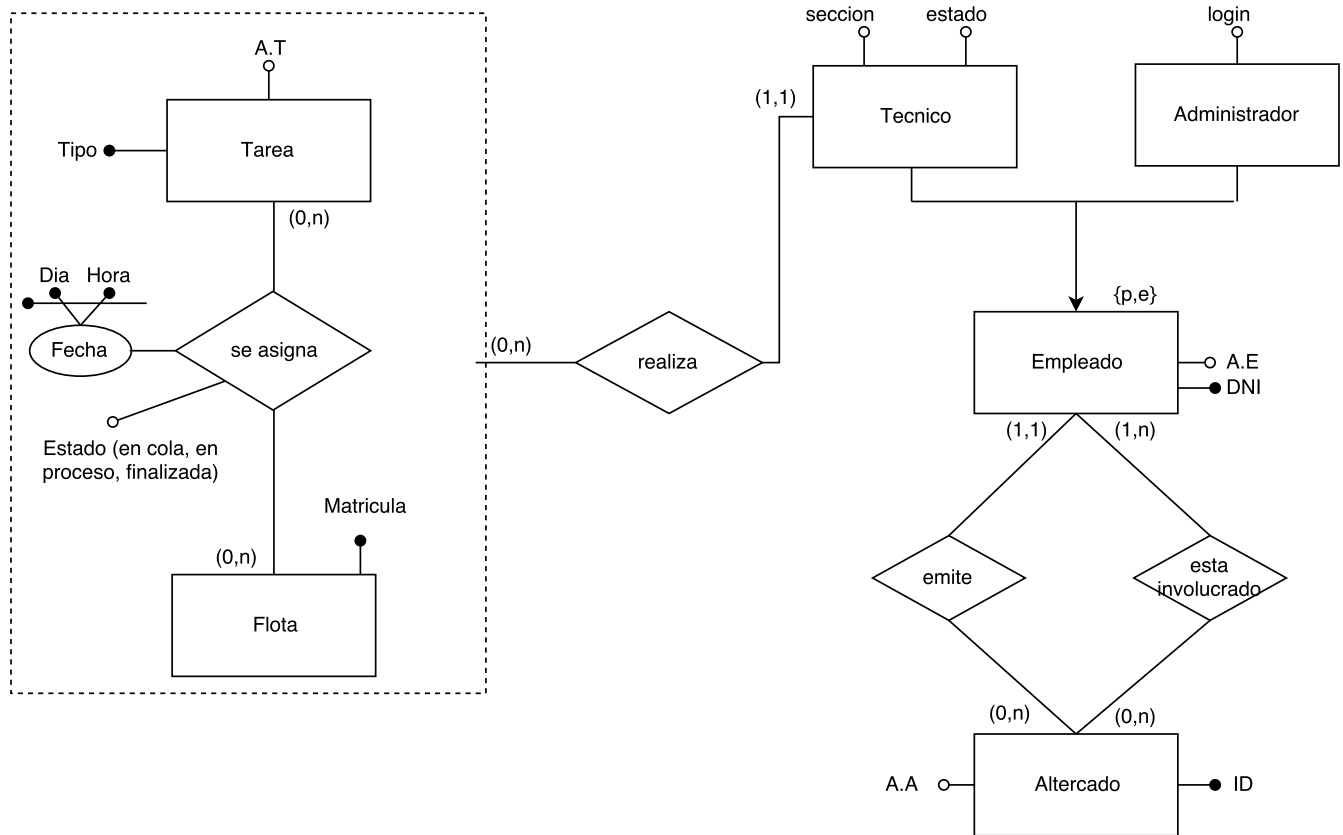


## Esquema conceptual ER para cada subsistema

### Subsistemas clientes, reservas, carnés y multas



## Subsistema Gestion de Recursos Humanos

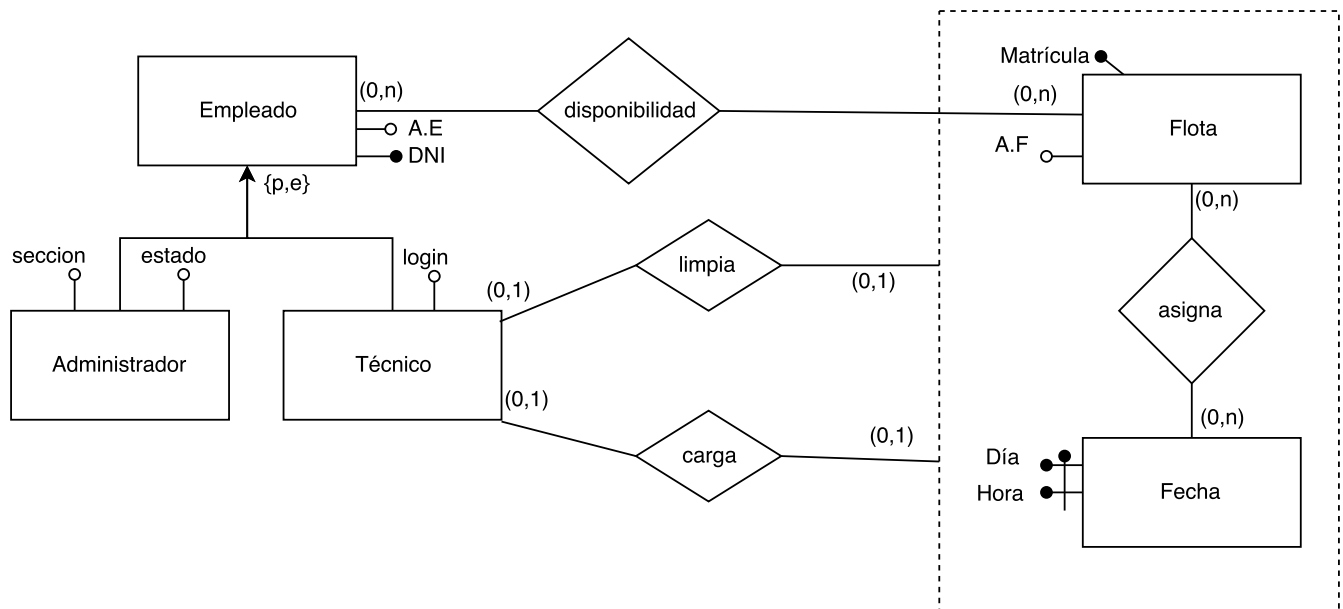


A.T = Tiempo maximo  
Descripcion

A.A = Descripción, Estado (en proceso, parado, resuelto)

A.E = Nombre y apellidos.  
Clave de acceso  
telefono  
email  
fecha de nacimiento  
cuenta corriente  
alta y baja del contrato  
tipo de contrato  
salario  
estado (activo o de baja)  
puesto de trabajo

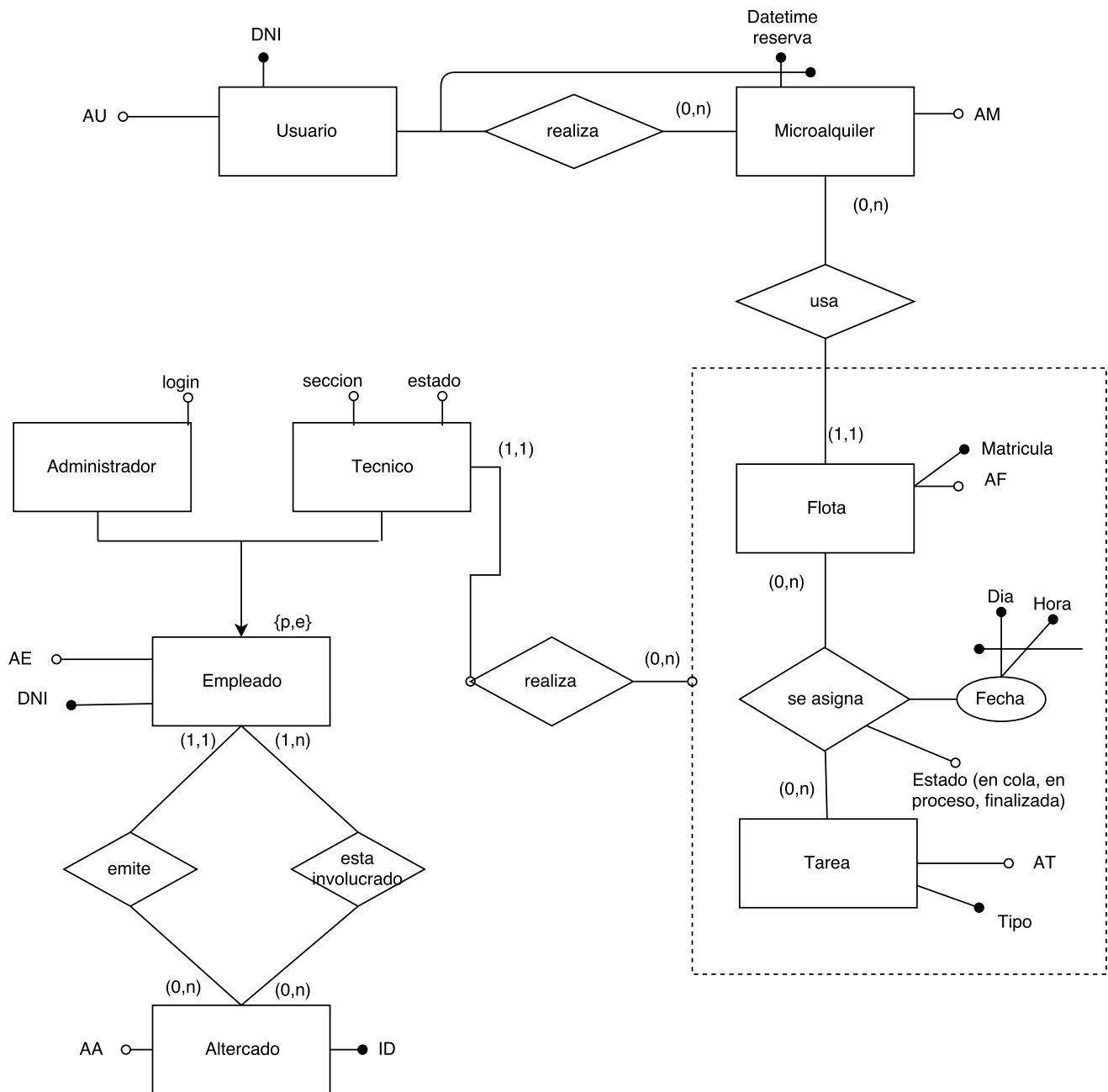
## Subsistema Gestion de Flota



A.F = Matrícula  
 Coste de alquiler  
 Empresa de alquiler  
 Estado de batería  
 Estado de limpieza  
 Localización  
 Disponibilidad moto  
 Estado moto

A.E = Nombre y apellidos.  
 Clave de acceso  
 telefono  
 email  
 fecha de nacimiento  
 cuenta corriente  
 alta y baja del contrato  
 tipo de contrato  
 salario  
 estado (activo o de baja)  
 puesto de trabajo

## Esquema conceptual ER de todo el sistema



AT = Tiempo maximo  
Descripcion

AA = Descripción,  
Estado (en proceso, parado, resuelto)

AE = Nombre y apellidos.  
Clave de acceso  
telefono  
email  
fecha de nacimiento  
cuenta corriente  
alta y baja del contrato  
tipo de contrato  
salario  
estado (activo o de baja)  
puesto de trabajo

AM = Estado  
Datetime inicio  
Datetime fin  
Posición inicio  
Posición fin  
Tarifa

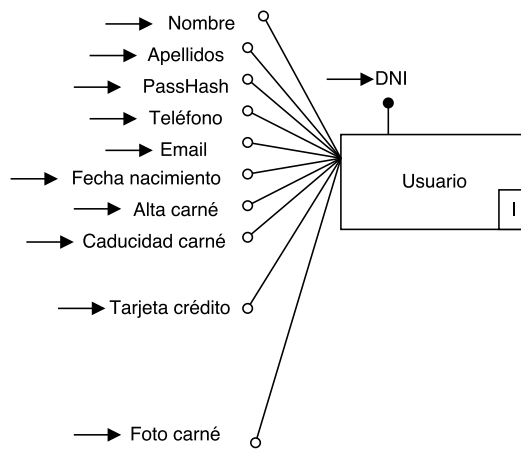
AF = Matrícula  
Coste de alquiler  
Empresa de alquiler  
Estado de batería  
Estado de limpieza  
Localización  
Disponibilidad moto  
Estado moto

AU = Nombre  
Apellidos  
PassHash  
Teléfono  
Email  
Fecha nacimiento  
Alta carné  
Caducidad carné  
Verificación carné  
Tarjeta crédito  
Baja  
Fecha baja  
Motivo baja  
Foto carné

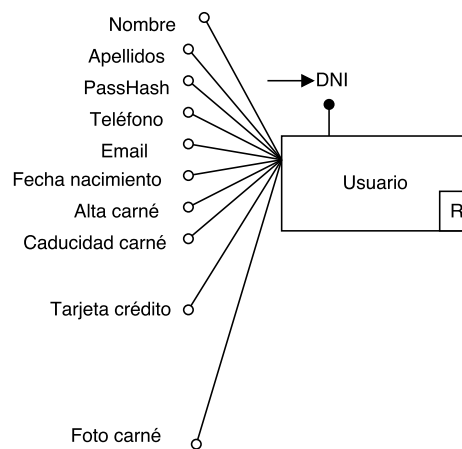
## P3: Operaciones de datos

### Subsistemas clientes, reservas, carnés y multas (Guillermo)

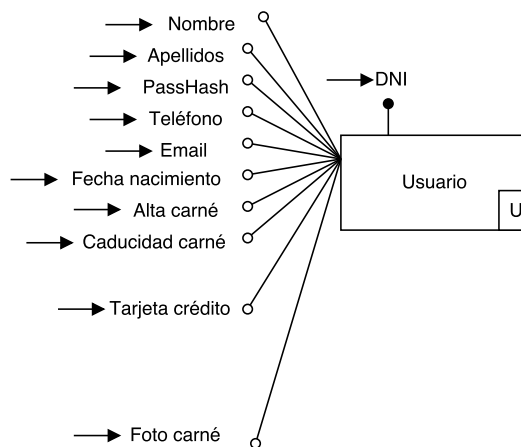
GO-1 Crear un nuevo usuario



GO-2 Leer usuario por DNI

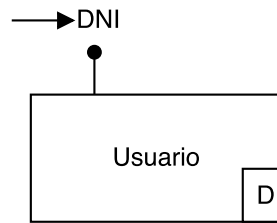


GO-3 Actualizar usuario

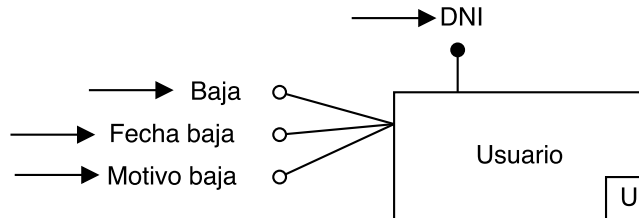


GO-4 Eliminar usuario

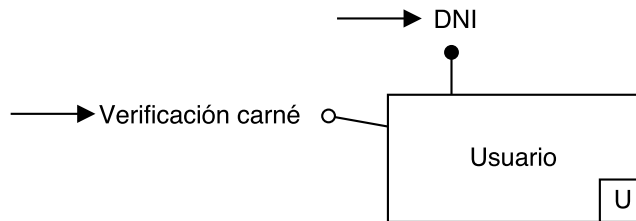




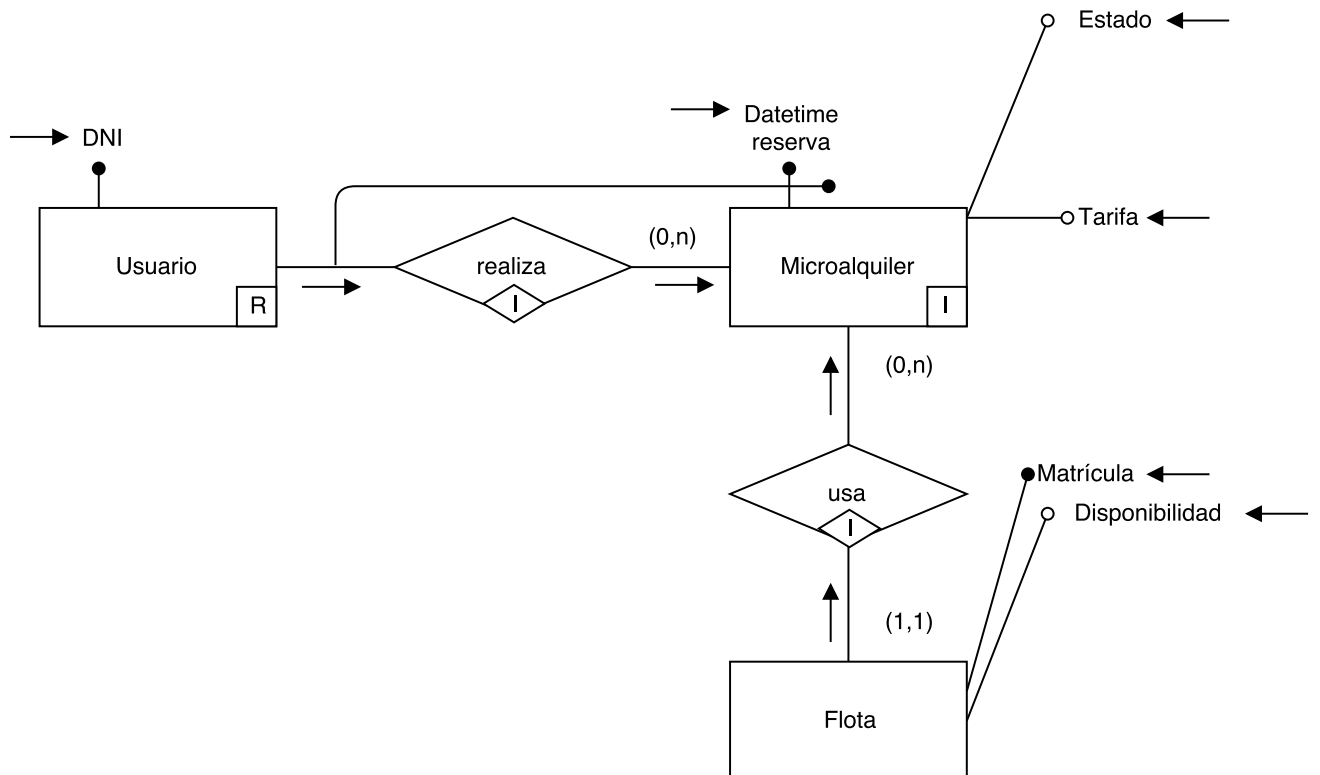
GO-5 Dar de baja un usuario sin eliminar su información



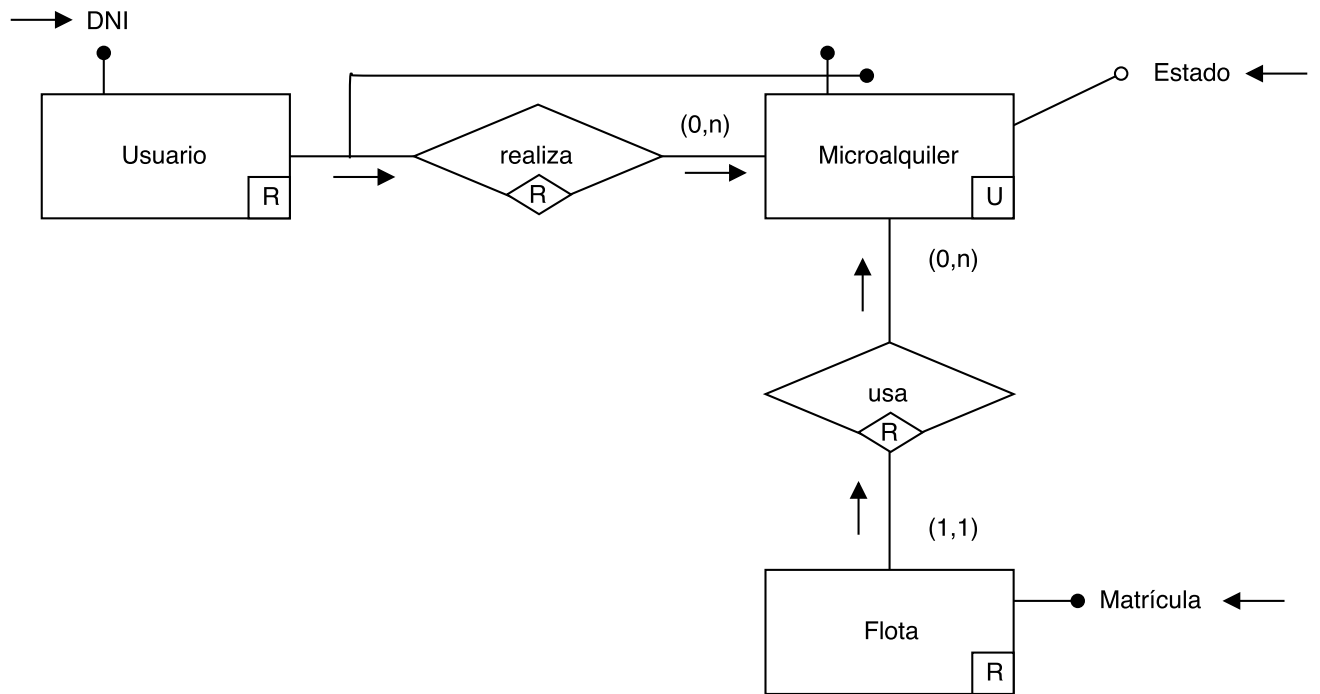
GO-6 Verificar carné de conducir



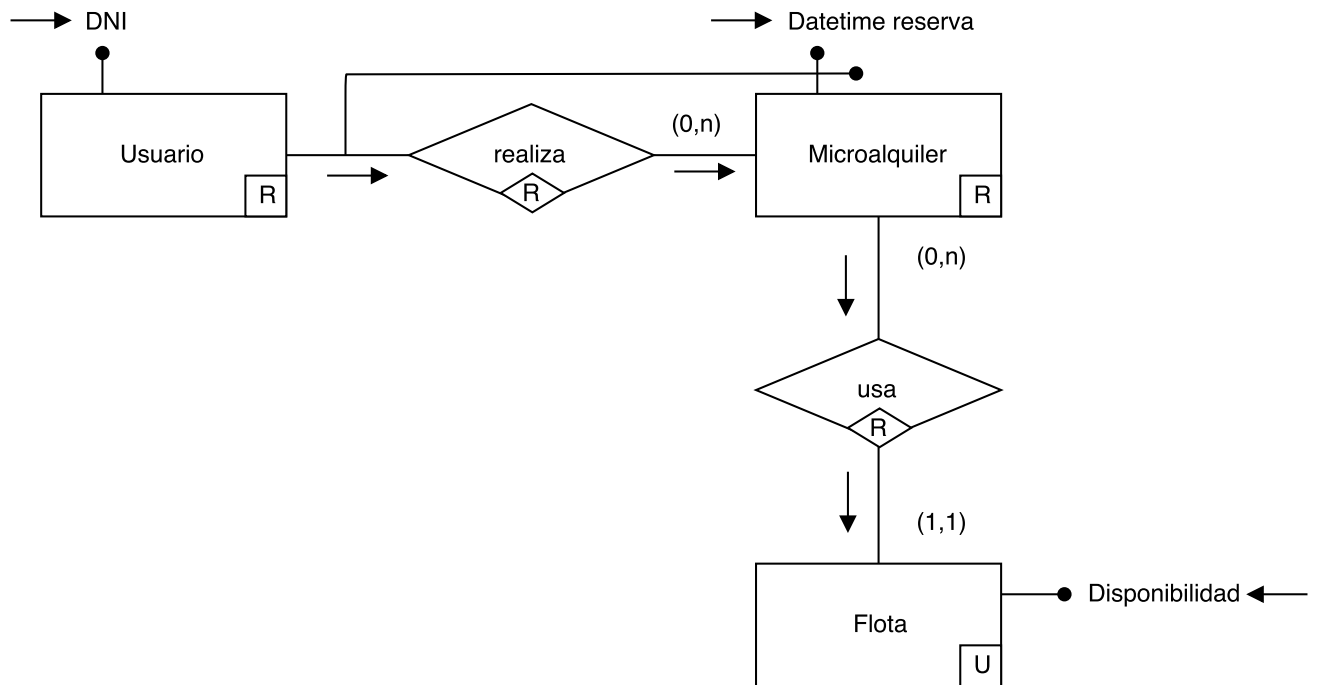
GO-7 Reservar una moto



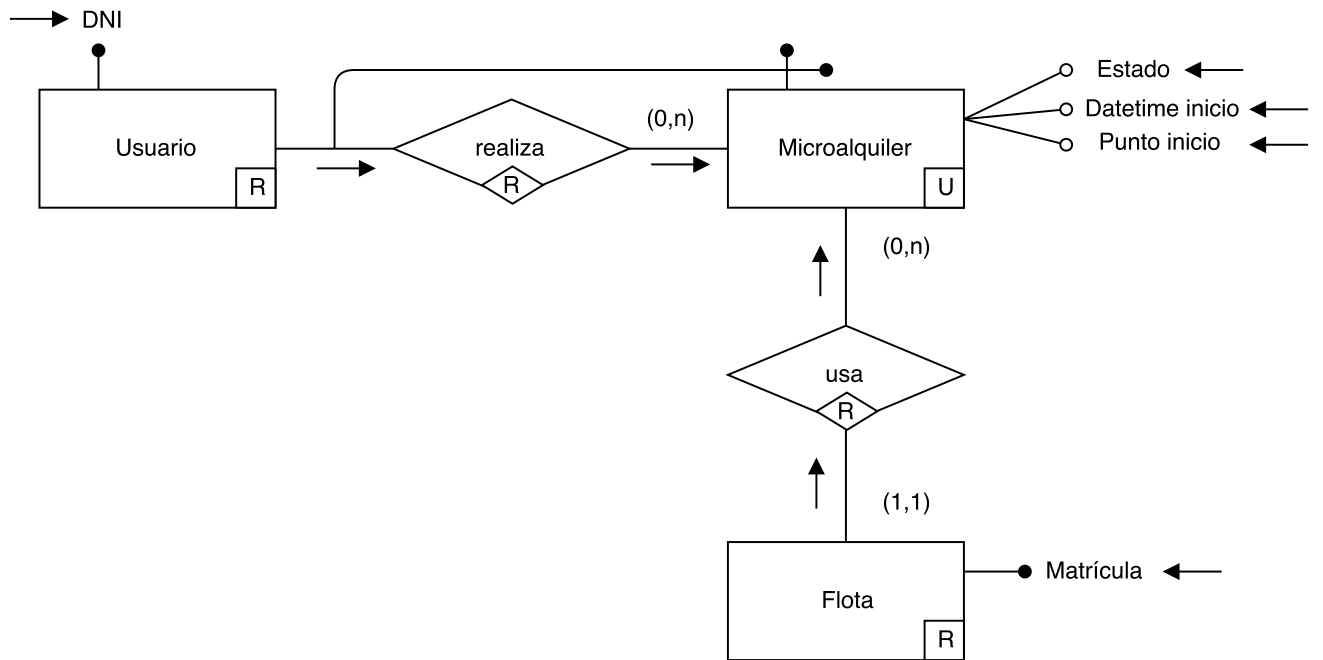
GO-8 Cancelar reserva. Modificar el estado de la reserva



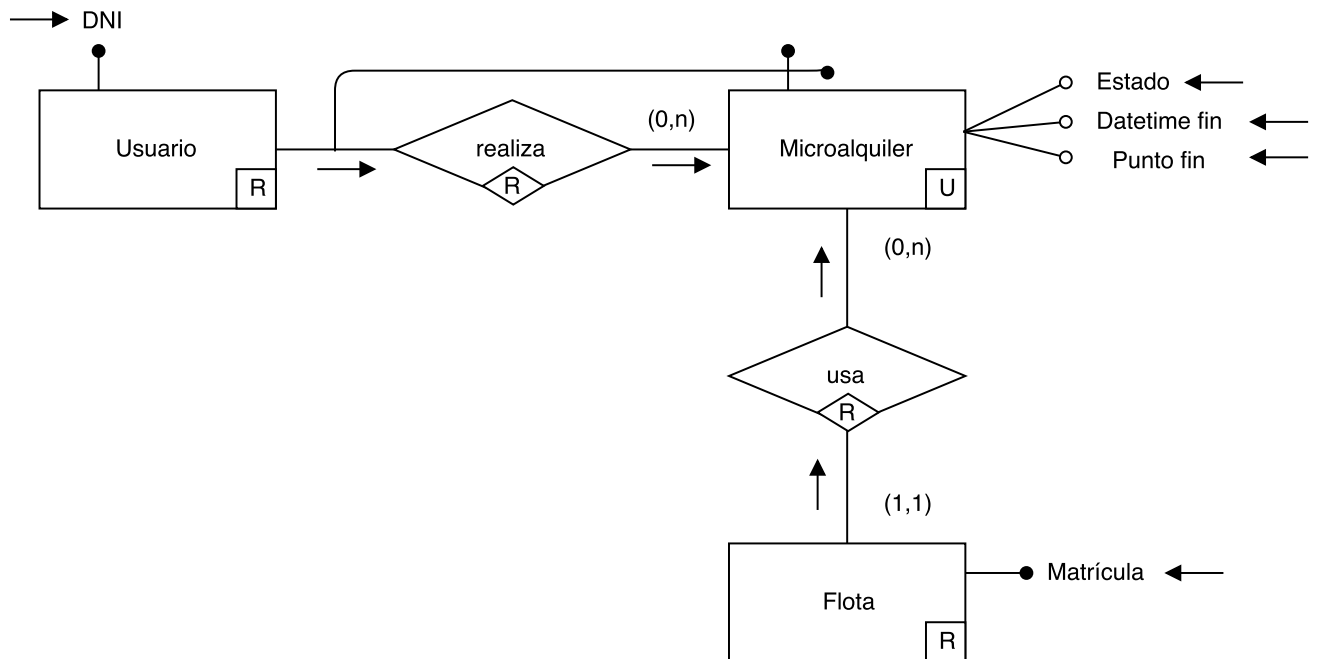
GO-9 Cancelar reserva. Modificar la disponibilidad de la moto



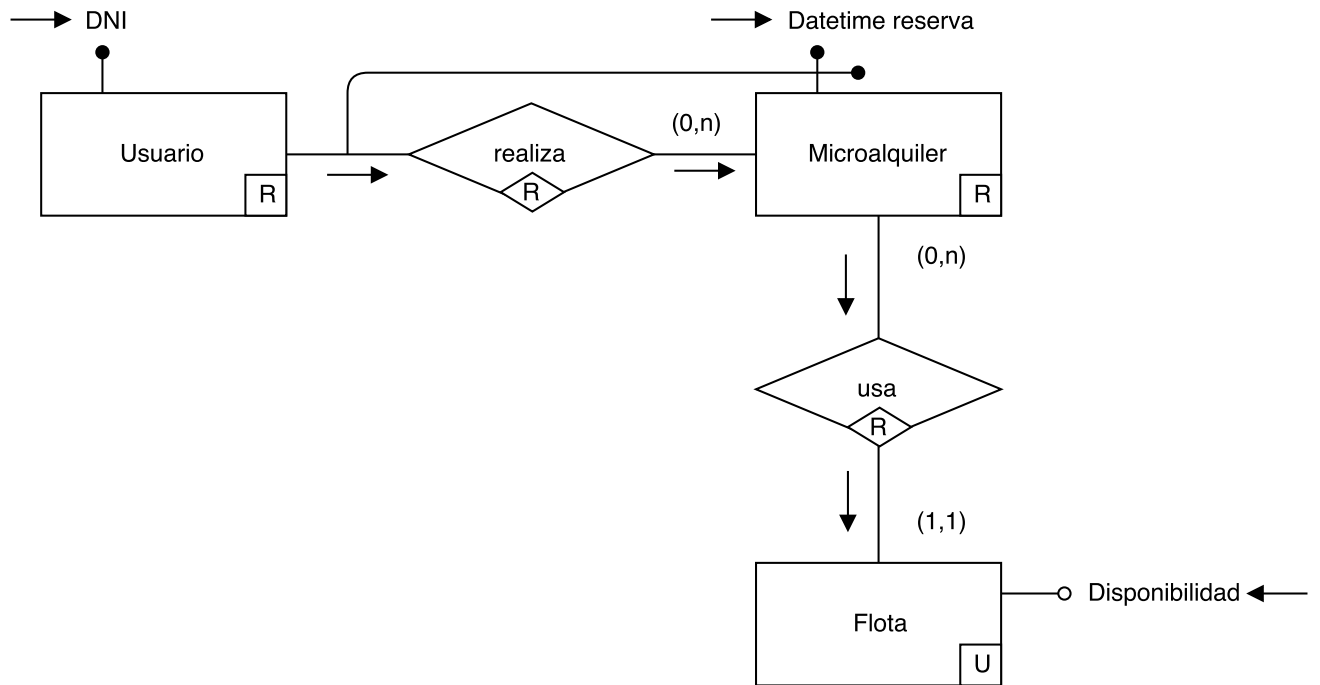
GO-10 Iniciar la reserva



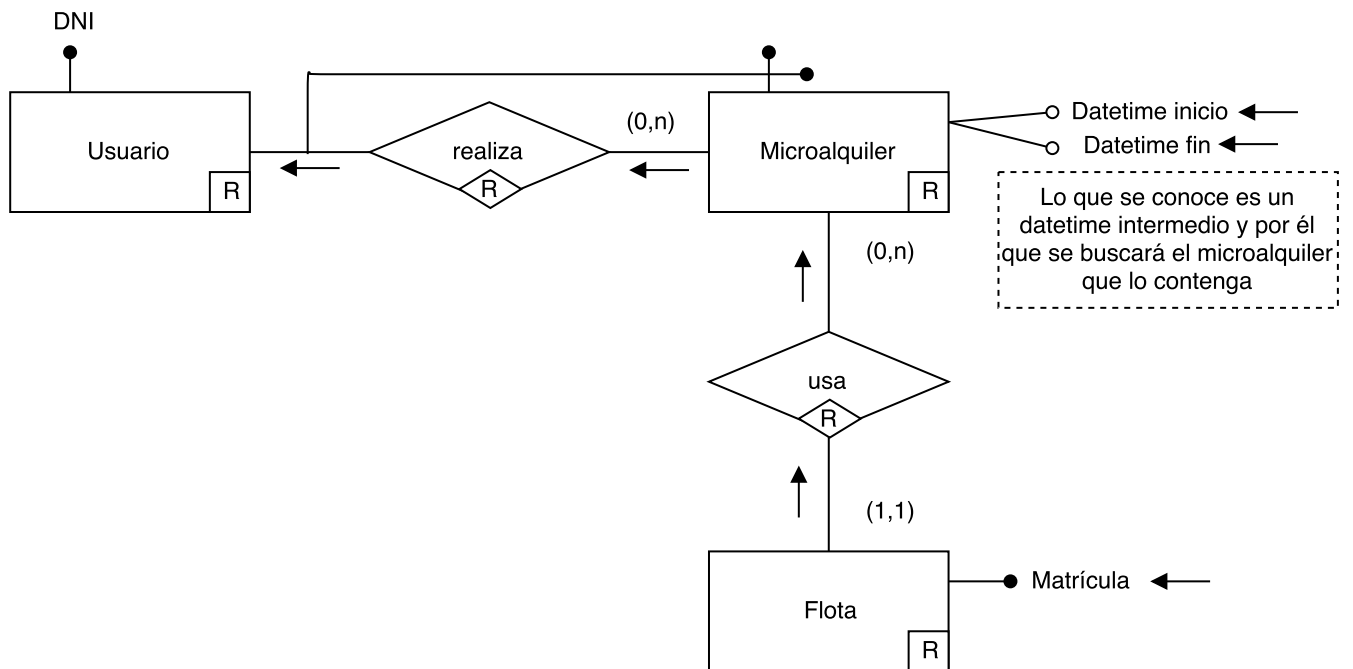
GO-11 Finalizar la reserva. Modificar el estado de la reserva



GO-12 Finalizar la reserva. Modificar la disponibilidad de la moto



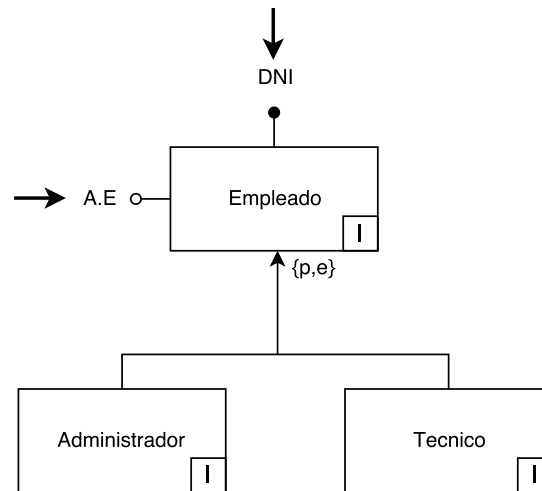
GO-13 Buscar usuario en una moto dada en una fecha y hora determinada



## Subsistema Gestion de Recursos Humanos (Jose Antonio)

JAO-1 Empleados

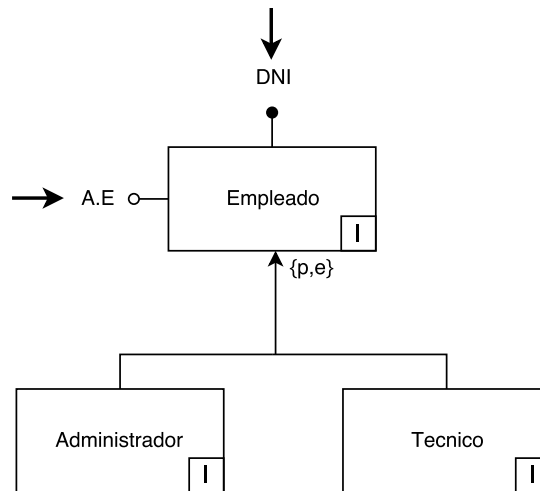
JAO-1.1 Crear nuevo empleado



A.E =

Nombre y apellidos.  
 Clave de acceso  
 telefono  
 email  
 fecha de nacimiento  
 cuenta corriente  
 alta y baja del contrato  
 tipo de contrato  
 salario  
 estado (activo o de baja)  
 puesto de trabajo

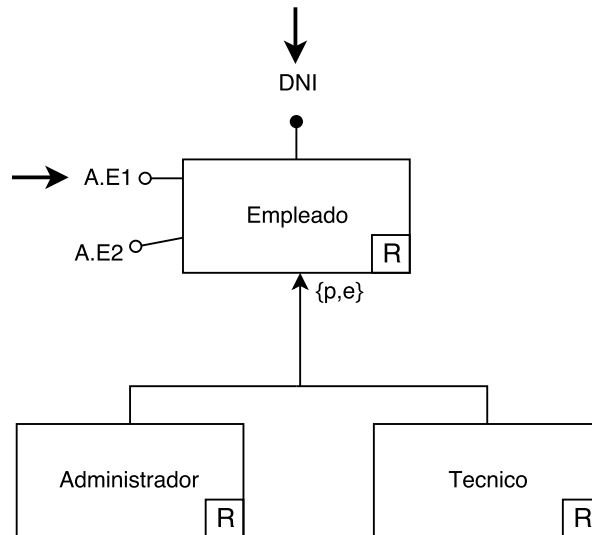
JAO-1.2 Actualizar empleado



A.E =

- Nombre y apellidos.
- Clave de acceso
- telefono
- email
- fecha de nacimiento
- cuenta corriente
- alta y baja del contrato
- tipo de contrato
- salario
- estado (activo o de baja)
- puesto de trabajo
- evaluacion

JAO-1.3 Consultar empleado



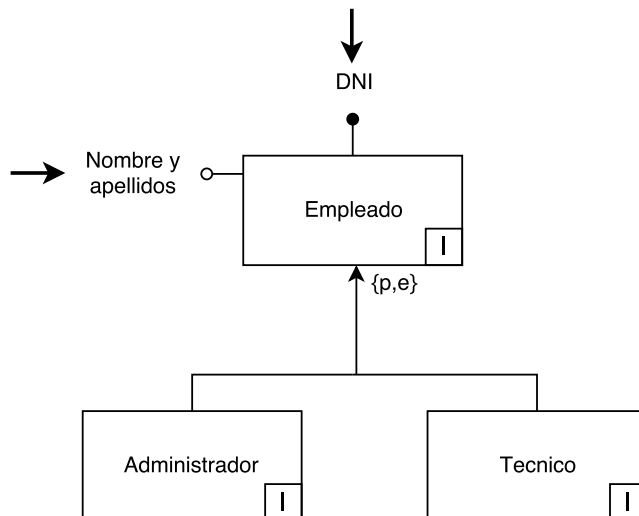
A.E1 =

Nombre y apellidos.  
cuenta corriente  
estado (activo o de baja)

A.E2 =

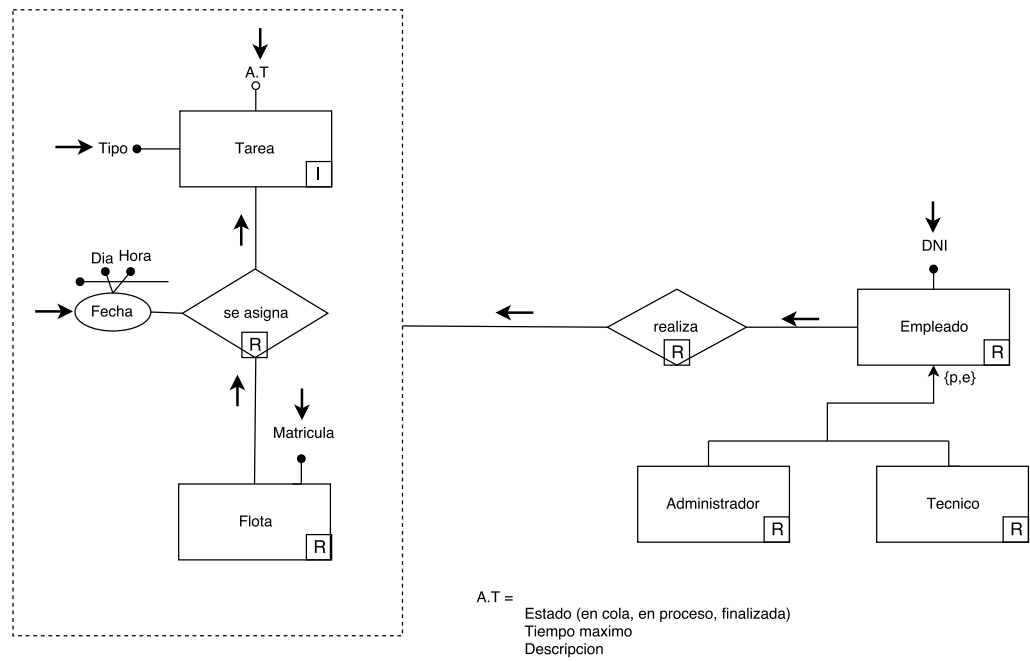
Clave de acceso  
telefono  
email  
fecha de nacimiento  
alta y baja del contrato  
tipo de contrato  
salario  
puesto de trabajo

#### JAO-1.4 Eliminar empleado

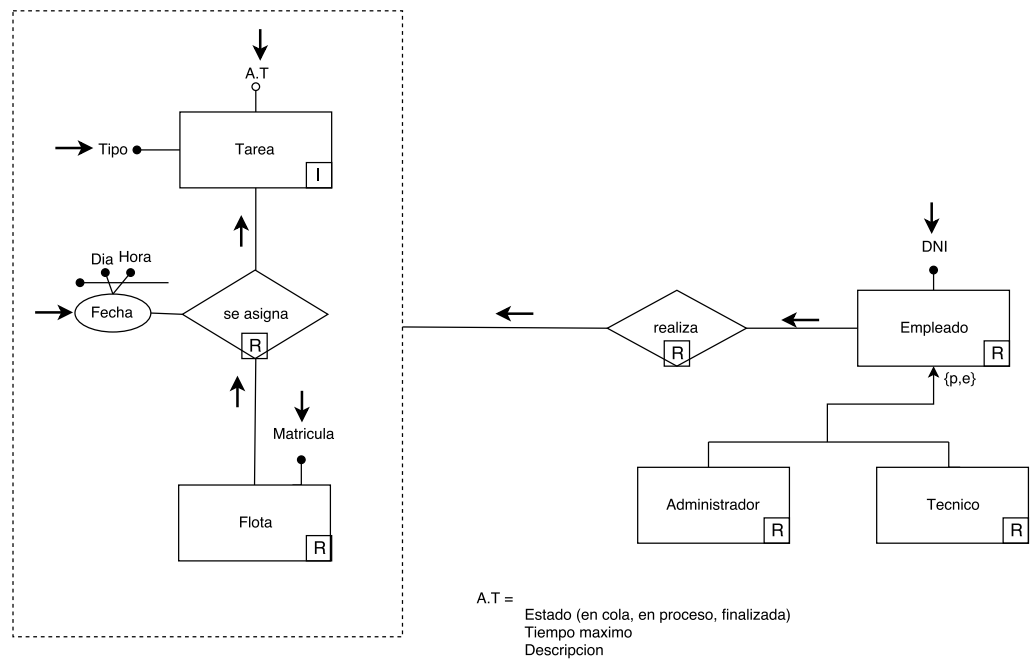


#### JAO-2 Tareas

##### JAO-2.1 Crear tarea



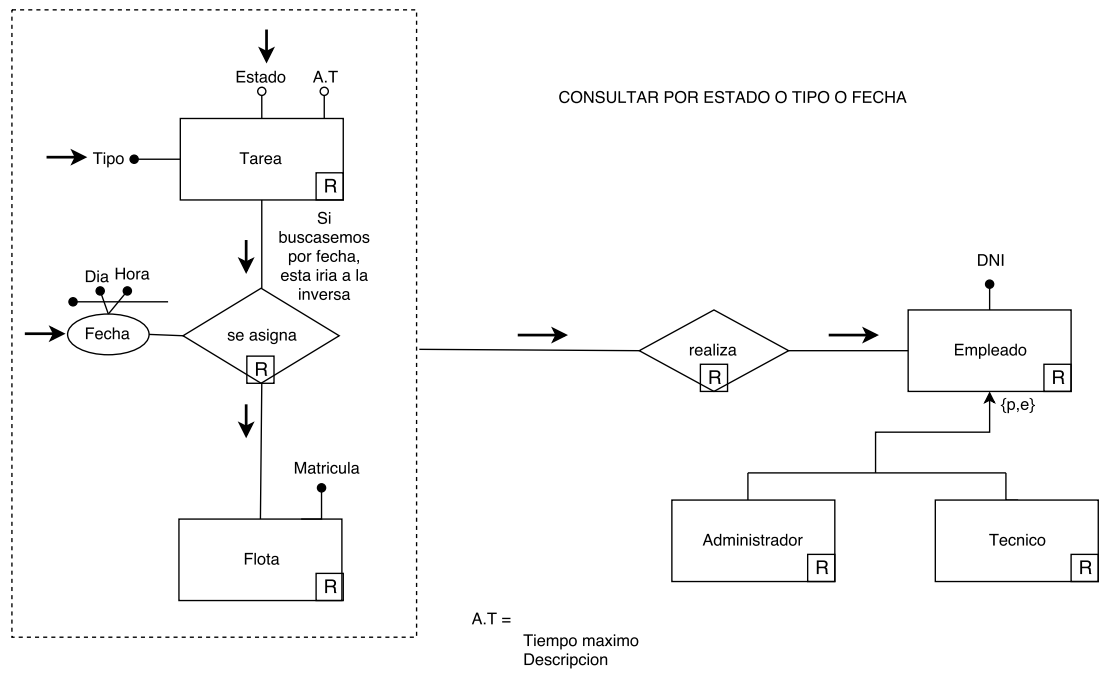
## JAO-2.2 Actualizar tarea



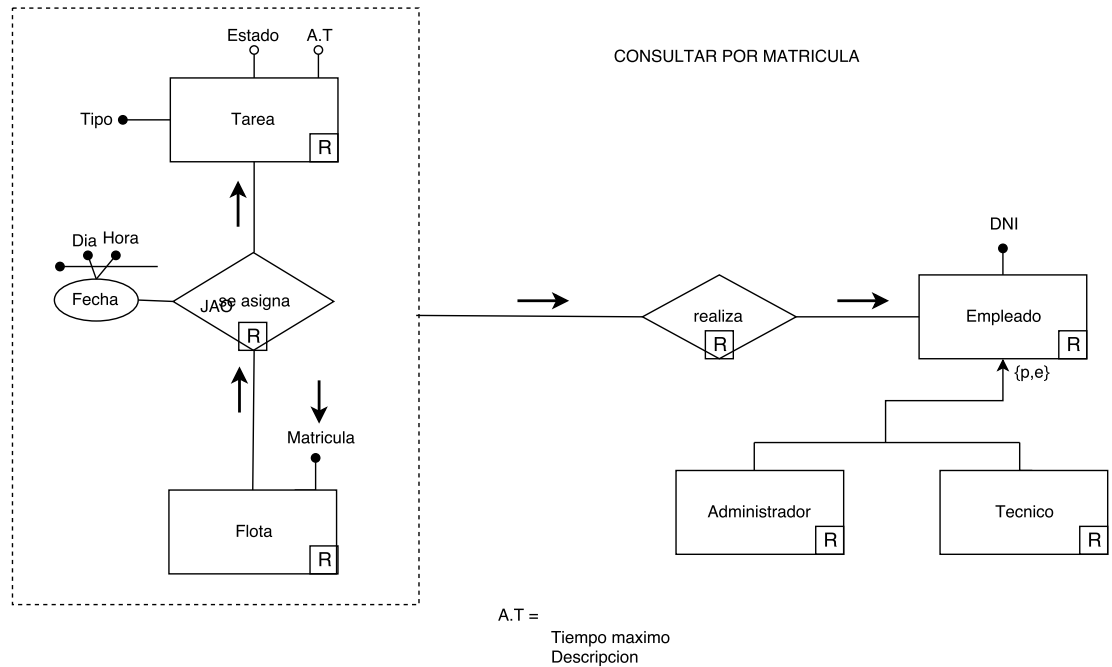
## JAO-2.3 Consultar tarea

### JAO-2.3.1 Consultar tarea por estado o tipo o fecha

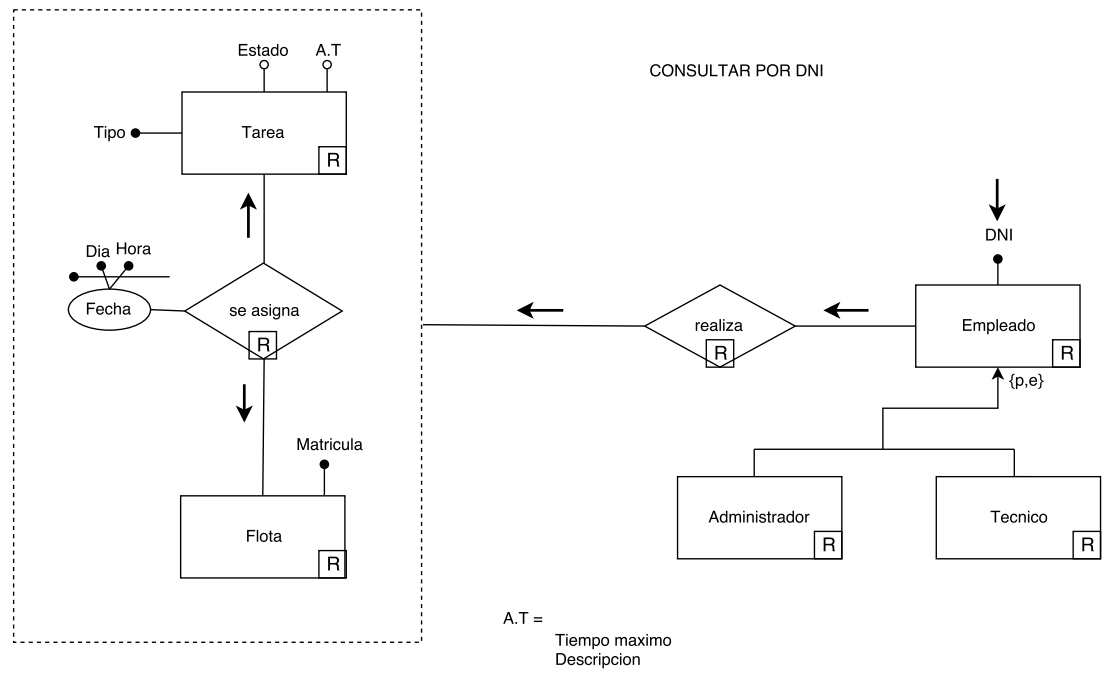




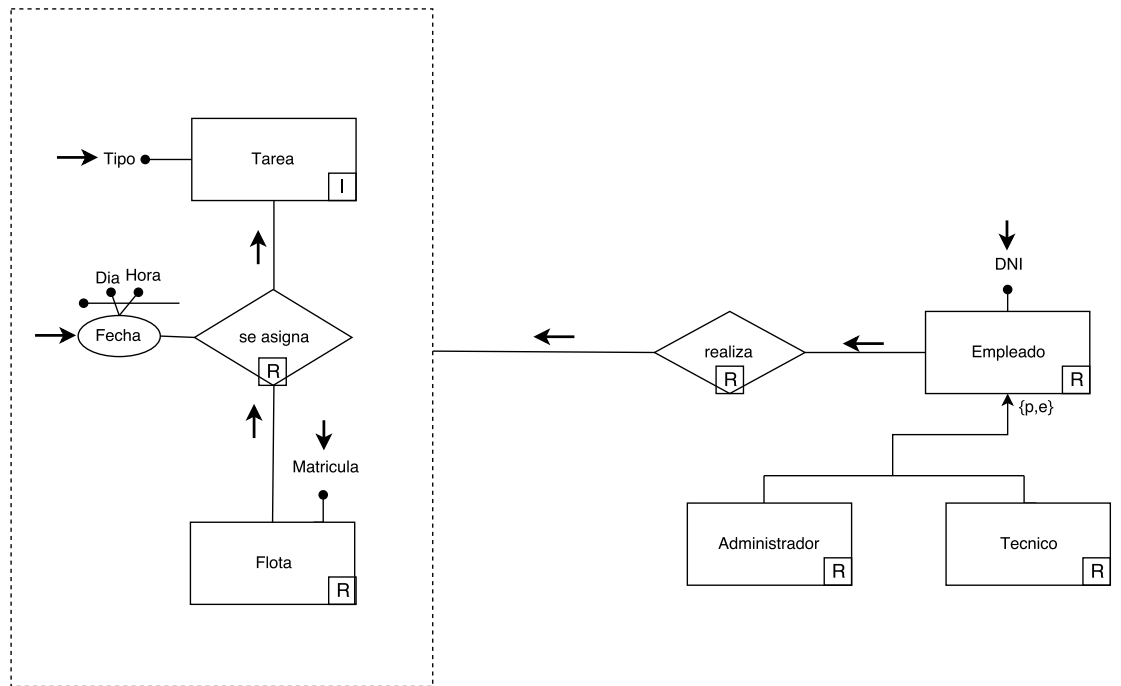
### JAO-2.3.2 Consultar tarea por matricula



### JAO-2.3.3 Consultar tarea por dni

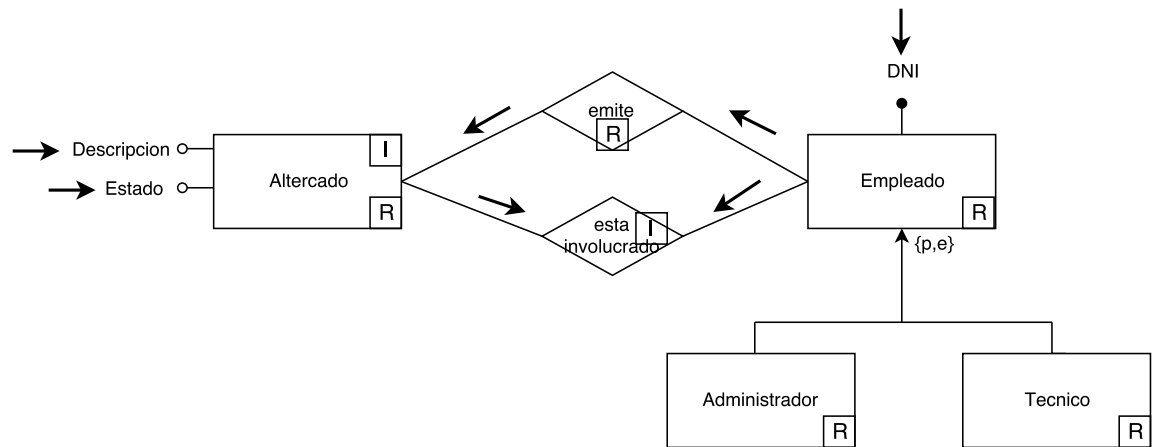


## JAO-2.4 Eliminar tarea

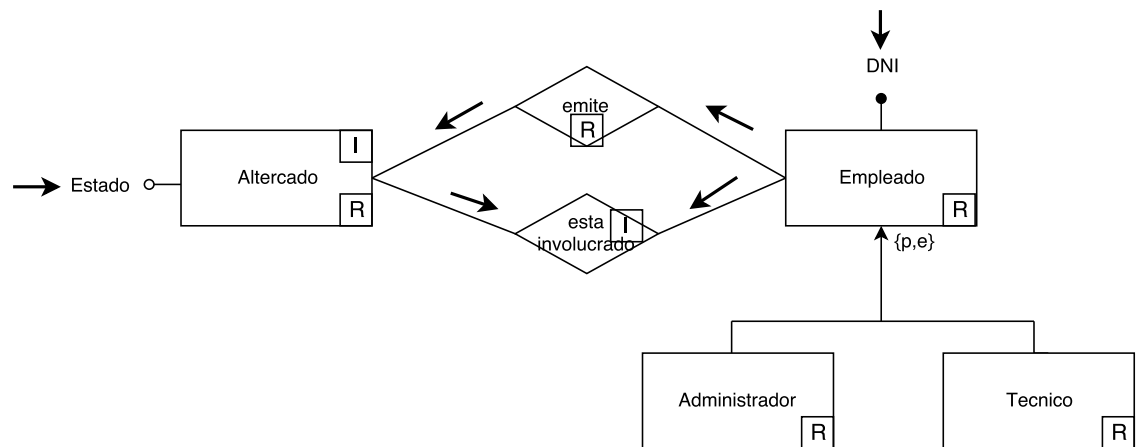


## JAO-3 Altercados

### JAO-3.1 Crear altercado

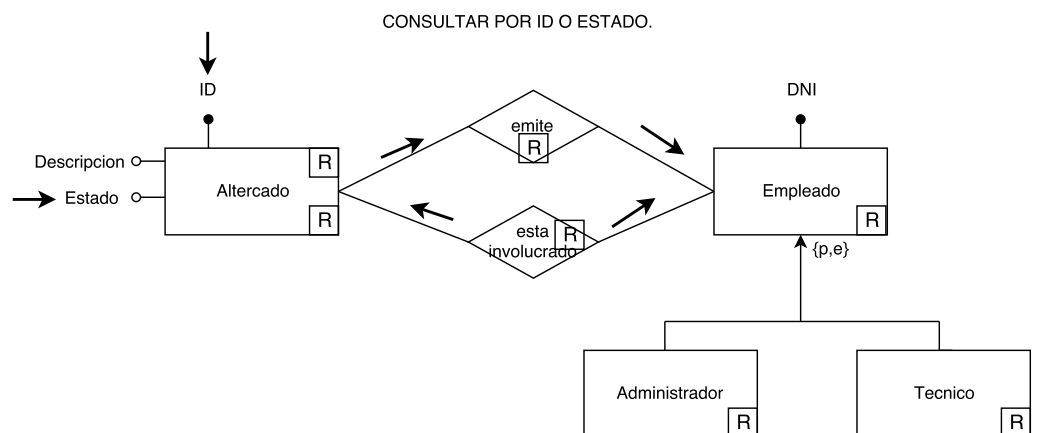


### JAO-3.2 Actualizar altercado

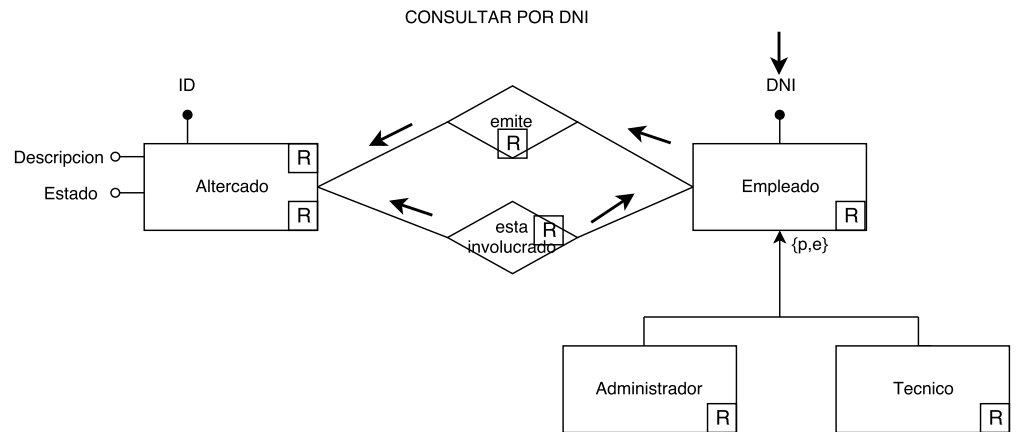


### JAO-3.3 Consultar altercado

#### JAO-3.3.1 Consultar altercado por estado o id

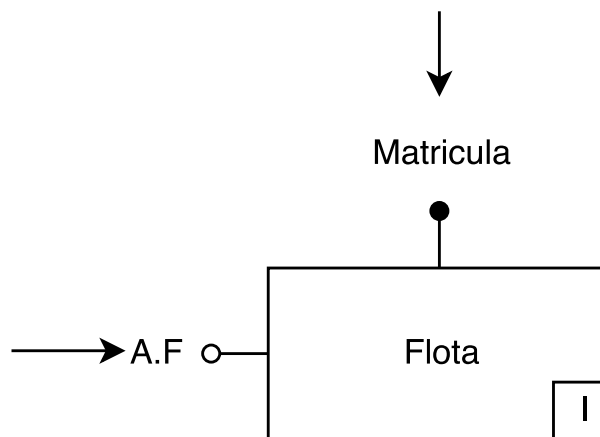


#### JAO-3.3.2 Consultar altercado por dni

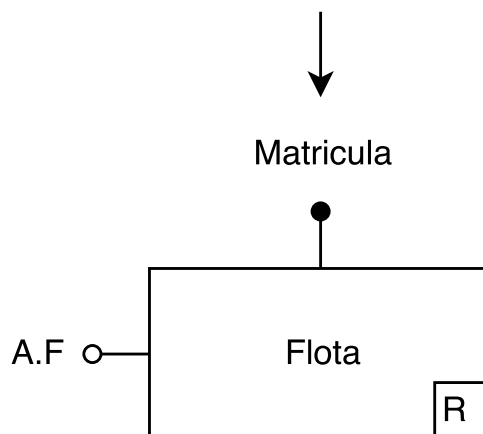


## Subsistema Gestion de Flota (Adrián)

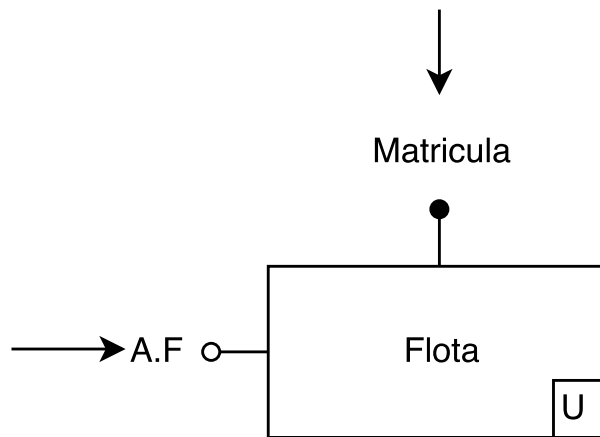
AO-1 Insertar nueva moto identificada por su matrícula



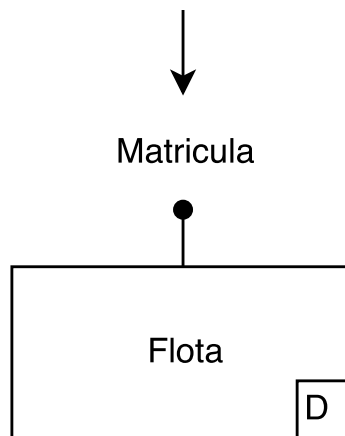
AO-2 Buscar moto identificada por su matrícula



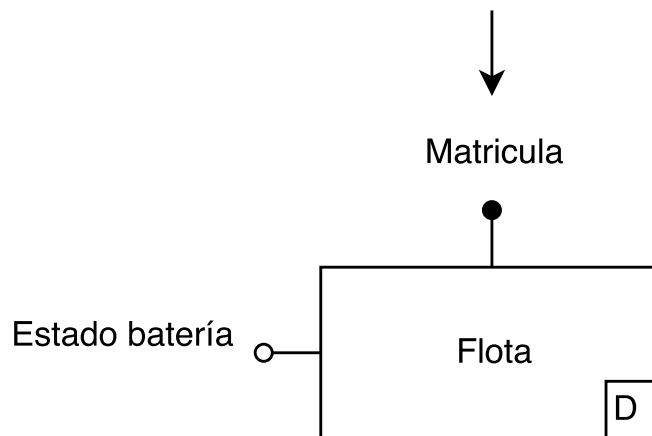
AO-3 Modificar moto identificada por su matrícula con los nuevos atributos que se indican



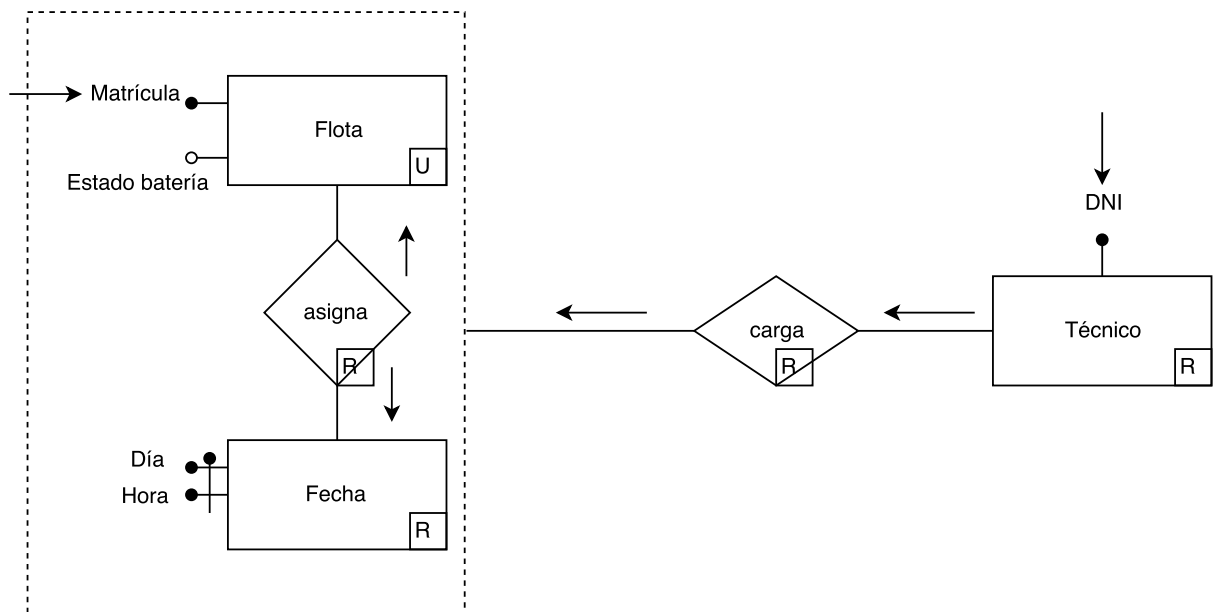
AO-4 Eliminar moto identificada por su matrícula



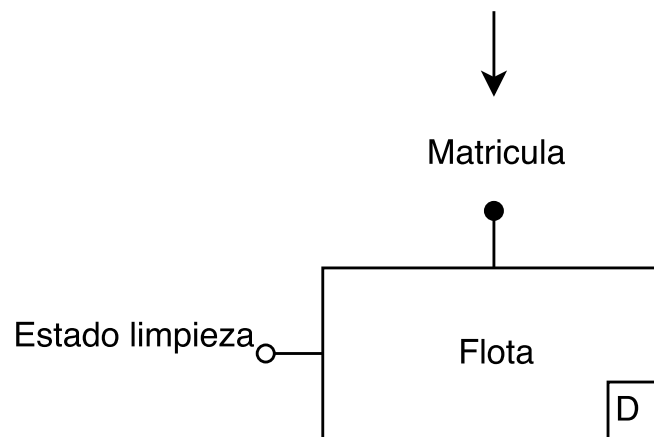
AO-5 Consultar estado de batería de moto identificada por su matrícula



AO-6 Modificar estado de batería de moto identificada por su matrícula

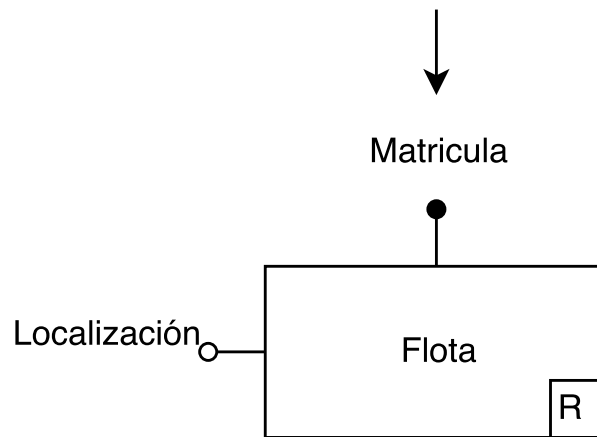


AO-7 Consultar estado de limpieza de moto identificada por su matrícula



AO-8 Modificar estado de limpieza de moto identificada por su matrícula





## Implementación del sistema

Para la implementación de nuestra idea hemos tomado tres funcionalidades relativamente interconectadas y las hemos integrado en un sistema de información. Para realizar la prueba de concepto en un principio intentamos utilizar Odoo[1], ERP sobre el que hablamos en clase y con el que tuvimos la oportunidad de entrar en contacto y comprender con cierta profundidad en el desarrollo del seminario, sin embargo, al empezar a ver cómo concretar algunas características concretas las dificultades que surgían para implementar las funcionalidades que habíamos decidido eran más que las ventajas.

Así que finalmente decidimos utilizar dos tecnologías que o bien nos gustaban o bien queríamos aprender, en primer lugar el sistema de bases de datos de Oracle (**Oracle Database 11g Express Edition**[2]), que tiene tanto un intérprete SQL online como un editor de triggers asociados a tablas, así como la herramienta Application Builder, que permite crear interfaces para la representación de la información contenida en la base de datos. Y por otro lado, NodeJS[3] para implementar la interfaz web del cliente, aprovechando que Oracle tiene implementado una API sobre Node para ejecutar comandos SQL sobre la base de datos.

Decidimos alquilar un VPS de OVH por un mes para implementar el servidor CentOS que alojaría tanto la base de datos como el servidor ExpressJS[4]. La definición de las tablas fue inmediata, únicamente tomamos las tablas que estaban relacionadas con las funcionalidades que queríamos implementar (prácticamente todas) y les aplicamos las restricciones definidas, unas directamente en la definición de la columna y otras mediante triggers como veremos más adelante. Una vez hechas éstas y habiendo poblado la base implementamos el resto del sistema.

## Paso a tablas

**Empleado**(DNI, Nombre, Apellidos, Clave\_acceso, telefono, email, fecha\_nacimiento, cuenta\_corriente, alta\_contrato, baja\_contrato, tipo\_contrato, salario, estado, puesto).

*Donde DNI es clave principal de la tabla Empleado.*

**Administrador**(DNI, login).



*Donde DNI es clave principal de la tabla Administrador y a su vez es clave externa al DNI de la tabla Empleado.*

**Tecnico**(DNI, seccion, estado).

*Donde DNI es clave principal de la tabla Administrador y a su vez es clave externa al DNI de la tabla Empleado.*

**Flota**(Matrícula, Coste\_alquiler, Empresa\_alquiler, Estado\_limpieza, Localización, Disponibilidad, Estado\_moto, Estado\_bateria).

*Donde Matrícula es clave principal de la tabla Flota.*

**Tarea**(Tipo, Tiempo\_máximo, Descripción).

*Donde Tipo es clave principal de la tabla Tarea.*

**Se Asigna**(Matrícula, Tipo, Fecha, DNI, Estado).

*Donde Matrícula, Tipo y fecha son clave principal de la tabla Se Asigna, Matrícula es clave externa de Flota, tipo es clave externa de Tarea y DNI es clave externa de Tecnico.*

**Usuario**(DNI, Nombre, Apellidos, PassHash, telefono, email, fecha\_nacimiento, alta\_carnet, caducidad\_carnet, verificacion\_carnet, tarjeta\_credito, baja, fecha\_baja, motivo\_baja, foto\_carnet).

*Donde DNI es clave principal de la tabla Usuario*

**Microalquiler**(datetime\_reserva, DNI, matricula, Estado, datetime\_inicio, datetime\_fin, posicion\_inicio, posicion\_fin, tarifa).

*Donde Datetime y DNI son clave principal de la tabla Microalquiler, DNI es clave externa al DNI de la tabla Usuario y matricula es clave externa a la matricula de la tabla Flota.*

## Normalización

Como era de esperar, gracias al procedimiento de diseño todas las tablas parecen encontrarse en FNBC. Vamos a ir analizando las dependencias funcionales y comprobarlo. Es inmediato ver que todas están en 1FN

### Empleado (Guillermo)

**Empleado**(DNI, Nombre, Apellidos, Clave\_acceso, telefono, email, fecha\_nacimiento, cuenta\_corriente, alta\_contrato, baja\_contrato, tipo\_contrato, salario, estado, puesto).

La única llave candidata sólida es DNI pues ni email, ni teléfono ni cuenta\_corriente pueden considerarse como unívocamente identificadora en un sentido estricto, tanto es así que no comprobamos que ninguna de ellas no se repita en la tabla, el empleado es responsable de dar estos datos válidos a fin de poder ser contactado y pagado.

Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos.

2. (2FN) Todos los atributos no primos, es decir todos menos DNI, dependen de forma completa de DNI.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan.

### Usuario (Guillermo)

**Usuario**(DNI, Nombre, Apellidos, PassHash, telefono, email, fecha\_nacimiento, alta\_carnet, caducidad\_carnet, verificacion\_carnet, tarjeta\_credito, baja, fecha\_baja, motivo\_baja, foto\_carnet).

Esta es prácticamente idéntica a la tabla empleado, la llave primaria y única llave candidata es DNI, y la única diferencia en esta tabla es que podríamos haber creado una tabla baja\_usuario, dependencia débil de esta tabla y consecuentemente identificada por DNI, pero no hay ni dependencias funcionales distintas de aquellas derivada de la llave primaria ni dependencias transitivas.

Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos.
2. (2FN) Todos los atributos no primos, es decir todos menos DNI, dependen de forma completa de DNI.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan.

### Microalquiler (Guillermo)

Las tablas que representan relaciones suelen ser más problemáticas que aquellas sobre entidades, vamos a analizar esta detenidamente.

**Microalquiler**(datetime\_reserva, DNI, matricula, Estado, datetime\_inicio, datetime\_fin, posicion\_inicio, posicion\_fin, tarifa).

Aquí sí tenemos una llave primaria (y única llave candidata) compuesta, DNI y datetime\_reserva, sin embargo, es inmediatamente comprobable lo siguiente:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos, hay atributos que pueden ser nulos y se irán cumplimentando en las distintas etapas del proceso pero ninguno pertenece a una llave candidata.
2. (2FN) Todos los atributos no primos, dependen de forma completa de DNI y datetime\_reserva.
3. (3FN) Como por definición, en este diseño, la tarifa viene *hardcodeada* no hay ningún otro atributo conflictivo y consecuentemente no tiene dependencias transitivas.

4. (FNBC) En este caso si hay llaves candidatas compuestas, la primaria, pero se comprueba que para todo atributo, si  $\alpha \rightarrow \beta$ ,  $\alpha$  es llave candidata y  $\beta$  no es un subconjunto de  $\alpha$  siendo ambas distintas, es decir, la llave candidata es mínima y no hay dependencias transitivas.

### **Tecnico (Jose Antonio)**

**Tecnico**(DNI, seccion, estado).

La llave primaria y única llave candidata es DNI, que a su vez es clave externa de la tabla Administrador y no hay ni dependencias funcionales distintas de aquellas derivada de la llave primaria ni dependencias transitivas.

Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos.
2. (2FN) Todos los atributos no primos, dependen de forma completa de DNI.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan..

### **Administrador (Jose Antonio)**

**Administrador**(DNI, login).

Esta es prácticamente idéntica a la tabla tecnico, la llave primaria y única llave candidata es DNI, y no hay ni dependencias funcionales distintas de aquellas derivada de la llave primaria ni dependencias transitivas.

Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos.
2. (2FN) Todos los atributos no primos, es decir, login, dependen de forma completa de DNI.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan.

### **Se.asigna (Jose Antonio)**

**Se.Asigna**(Matrícula, Tipo, Fecha, DNI, Estado).

Aquí sí tenemos una llave primaria (y única llave candidata) compuesta, siendo matricula, tipo y fecha, sin embargo, es inmediatamente comprobable lo siguiente:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos.

2. (2FN) Todos los atributos no primos, dependen de forma completa de matricula, tipo y fecha.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) En este caso si hay llaves candidatas compuestas, la primaria, pero se comprueba que para todo atributo, si  $\alpha \rightarrow \beta$ ,  $\alpha$  es llave candidata y  $\beta$  no es un subconjunto de  $\alpha$  siendo ambas distintas, es decir, la llave candidata es mínima y no hay dependencias transitivas.

### Flota (Adrián)

**Flota**(Matrícula, Coste\_alquiler, Empresa\_alquiler, Estado\_limpieza, Localización, Disponibilidad, Estado\_moto, Estado\_batería). La llave primaria y única llave candidata es Matrícula, y no hay ni dependencias funcionales distintas de aquellas derivada de la llave primaria ni dependencias transitivas. Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos, hay atributos que pueden ser nulos, pero no pertenecen a ninguna llave candidata.
2. (2FN) Todos los atributos no primos dependen de forma completa de Matrícula.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan.

### Tarea (Adrián)

**Tarea**(Tipo, Tiempo\_máximo, Descripción). La llave primaria y única llave candidata es Tipo, y no hay ni dependencias funcionales distintas de aquellas derivada de la llave primaria ni dependencias transitivas. Por lo tanto:

1. (1FN) Todas las columnas son regulares y no tienen, por diseño, atributos que pertenezcan a claves candidatas y que puedan ser nulos, hay atributos que pueden ser nulos, pero no pertenecen a ninguna llave candidata.
2. (2FN) Todos los atributos no primos dependen de forma completa de Tipo.
3. (3FN) No tiene dependencias transitivas.
4. (FNBC) No hay varias llaves candidatas, ni son compuestas ni se solapan.

## Creacion de tablas (SQL)

### Empleado:

```
CREATE TABLE "EMPLEADO"
( "DNI" VARCHAR2(9) PRIMARY KEY,
  "NOMBRE" VARCHAR2(30) NOT NULL,
  "APELLIDOS" VARCHAR2(50) NOT NULL,
  "CLAVE_ACCESO" VARCHAR2(40) NOT NULL,
  "TELEFONO" NUMBER(9,0) NOT NULL,
  "EMAIL" VARCHAR2(30),
```

```

"FECHA_NACIMIENTO" DATE,
"CUENTA_CORRIENTE" VARCHAR2(24) NOT NULL,
"ALTA_CONTRATO" DATE DEFAULT sysdate NOT NULL,
"BAJA_CONTRATO" DATE,
"TIPO_CONTRATO" VARCHAR2(7) NOT NULL,
"SALARIO" NUMBER(5,0) NOT NULL,
"ESTADO" VARCHAR2(6) NOT NULL,
    CHECK (tipo_contrato IN('fijo','prueba','parcial')),
    CHECK (estado IN('activo','baja')),
CHECK (email LIKE '%_@__%.__%')
) ;

```

### Administrador:

```

CREATE TABLE "ADMINISTRADOR"
( "DNI" PRIMARY KEY REFERENCES "EMPLEADO",
  "LOGIN" CHAR(1) DEFAULT 'N' NOT NULL,
  CHECK (login IN('S','N'))
) ;

```

### Tecnico:

```

CREATE TABLE "TECNICO"
( "DNI" PRIMARY KEY REFERENCES "EMPLEADO",
  "SECCION" VARCHAR2(8) NOT NULL,
  "ESTADO" VARCHAR2(7) DEFAULT 'libre' NOT NULL,
  CHECK (seccion IN('recambio','limpieza','posicion')),
  CHECK (estado IN('libre','ocupado'))
) ;

```

### Flota:

```

CREATE TABLE "FLOTA"
( "MATRICULA" VARCHAR2(7),
  "COSTE_ALQUILER" NUMBER(4,0) NOT NULL,
  "EMPRESA_ALQUILER" VARCHAR2(30) NOT NULL,
  "ESTADO_LIMPIEZA" VARCHAR2(6) NOT NULL,
  "LOCALIZACION" VARCHAR2(40) NOT NULL,
  "DISPONIBILIDAD" VARCHAR2(1) DEFAULT 'S',
  "ESTADO_MOTO" VARCHAR2(30),
  "ESTADO_BATERIA" NUMBER(3,0) NOT NULL,
  CHECK (estado_limpieza IN('sucia','limpia')),
  CHECK (disponibilidad IN('S','N')),
  PRIMARY KEY ("MATRICULA"),
  CHECK (estado_bateria BETWEEN 0 AND 100)
) ;

```

### Tarea:

```

CREATE TABLE "TAREA"
( "TIPO" VARCHAR2(10),
  "TIEMPO_MAXIMO" NUMBER(3,0),
  "DESCRIPCION" VARCHAR2(100),
  PRIMARY KEY ("TIPO"),
  CONSTRAINT "AAA" CHECK ( "TIPO" IN('limpiar','recambio','posicion','carga'))
) ;

```

### Se Asigna:

```

CREATE TABLE "SE_ASIGNA"
( "MATRICULA" VARCHAR2(7),
  "TIPO" VARCHAR2(10),
  "FECHA" TIMESTAMP,
  "DNI" VARCHAR2(9) NOT NULL,
  "ESTADO" VARCHAR2(10) NOT NULL,
  CHECK (estado IN('en_cola','en_proceso','finalizada')),
  PRIMARY KEY ("MATRICULA", "TIPO", "FECHA")
) ;ALTER TABLE "SE_ASIGNA" ADD FOREIGN KEY ("MATRICULA") REFERENCES "FLOTA" ("MATRICULA");
↪ ALTER TABLE "SE_ASIGNA" ADD FOREIGN KEY ("TIPO") REFERENCES "TAREA" ("TIPO");ALTER
↪ TABLE "SE_ASIGNA" ADD FOREIGN KEY ("DNI") REFERENCES "TECNICO" ("DNI");

```

## Usuario:

```

CREATE TABLE "USUARIO"
( "DNI" VARCHAR2(9) NOT NULL,
  "NOMBRE" VARCHAR2(50) NOT NULL,
  "APELLIDOS" VARCHAR2(50) NOT NULL,
  "CLAVE_ACCESO" VARCHAR2(50) NOT NULL,
  "TELEFONO" VARCHAR2(20) NOT NULL,
  "EMAIL" VARCHAR2(50) NOT NULL,
  "FECHA_NACIMIENTO" DATE NOT NULL,
  "ALTA_CARNET" DATE,
  "CADUCIDAD_CARNET" DATE,
  "VERIFICACION_CARNET" NUMBER(1,0) NOT NULL,
  "TARJETA_CREDITO" VARCHAR2(19) NOT NULL,
  "BAJA" NUMBER(1,0) NOT NULL,
  "FECHA_BAJA" DATE,
  "MOTIVO_BAJA" VARCHAR2(100),
  "FOTO_CARNET" BLOB,
  CONSTRAINT "USUARIO_PK" PRIMARY KEY ("DNI"),
  CONSTRAINT "USUARIO_BOOL" CHECK (VERIFICACION_CARNET IN (0,1) AND BAJA IN (0,1))
) ;

```

## Microalquiler:

```

CREATE TABLE "MICROALQUILER"
( "DNI" VARCHAR2(9) NOT NULL,
  "FECHA_RESERVA" DATE NOT NULL,
  "ESTADO" NUMBER(1,0) NOT NULL,
  "FECHA_INICIO" DATE,
  "FECHA_FIN" DATE,
  "POSICION_INICIO" VARCHAR2(40),
  "POSICION_FIN" VARCHAR2(40),
  "TARIFA" NUMBER(10,3) NOT NULL,
  "MATRICULA" VARCHAR2(7) NOT NULL,
  CONSTRAINT "MICROALQUILER_PK" PRIMARY KEY ("DNI", "FECHA_RESERVA")
) ;ALTER TABLE "MICROALQUILER" ADD CONSTRAINT "MICROALQUILER_DNI" FOREIGN KEY ("DNI")
↪ REFERENCES "USUARIO" ("DNI");ALTER TABLE "MICROALQUILER" ADD CONSTRAINT "
↪ MICROALQUILER_MATRICULA" FOREIGN KEY ("MATRICULA") REFERENCES "FLOTA" ("MATRICULA");

```

## Triggers en PL-SQL

Para la definición de los triggers[6] decidimos implementar algunos relevantes que afecten a la parte del sistema que hemos implementado. En este caso, voy a mostrar los triggers mas complejos o que mas tareas realizan, y algunos como los que controlan la inserción de fechas y demás casos simples

los podreis ver en el codigo adjuntado a la practica. No obstante, los códigos complejos tambien estarán adjuntados por si desean realizar la implementacion de ellos.

## Tabla Flota

Dicho esto, uno de los problemas que nos encontramos en nuestro sistema es en la tabla Flota, cada vez que una moto está con estado 'sucia' o el porcentaje de batería es bajo (menor que 10 %), el estado de la moto debe ser N (no disponible), por lo que para solucionar esto, creamos el siguiente trigger.

```
create or replace trigger "FLOTA_ACT_DISPO"
BEFORE
update of disponibilidad on "FLOTA"
for each row
begin
  IF (:new.estado_bateria < 10 OR :new.estado_limpieza='sucia') THEN
    :new.disponibilidad:='N';
  ELSE
    :new.disponibilidad:=:new.disponibilidad;
  END IF;
end;
```

*Este trigger lo que comprueba es que cada vez que se actualiza la disponibilidad en flota, verifica si el estado y la bateria de la moto que se esta actualizando para insertar los datos correctamente, ya que si el administrador pone que la moto está disponible (por equivocacion) y la moto está sucia, seguirá como no disponible.*

El trigger anterior, su principal cometido es que el administrador tenga algun fallo al actualizar el estado de la moto, pero realmente ese trigger sólo tiene funcionabilidad de seguridad, el siguiente trigger es el verdadero encargado de actualizar automáticamente la disponibilidad de la moto fijandose en la batería y el estado de la moto.

```
create or replace trigger "FLOTA_DISPONIBILIDAD"
BEFORE
insert or update of estado_bateria, estado_limpieza on "FLOTA"
for each row
DECLARE N NUMBER(1);
begin
  SELECT COUNT(*) INTO N FROM microalquiler WHERE matricula=:new.matricula AND (estado=0 OR
    ⇐ estado=1);
  IF (:new.estado_bateria < 10 OR :new.estado_limpieza='sucia' OR N!=0) THEN
    :new.disponibilidad:='N';
  ELSE
    :new.disponibilidad:='S';
  END IF;
end;
```

*Como vemos, este trigger se activa cuando se actualiza el estado de la bateria o bien el estado de limpieza. Para definir el nuevo estado de la disponibilidad tambien revisa que no exista ningun microalquiler en el que se esté utilizando esta moto. Una vez realizados todos los pasos que vemos en el código, pone la disponibilidad a 'S' o 'N' según convenga.*

Un problema que vimos fué cuando una moto se pone como sucia o el estado de la batería es insuficiente, el administrador tiene que estar atento para asignarle una tarea a esa moto y asignarsela

a un empleado que se dedique a esa tarea para que la moto pueda limpiarse o cargarse lo antes posible. Por lo que decidimos crear un nuevo trigger que realizase esta tarea automáticamente.

```
create or replace trigger "FLOTA_AUTO_TAREA"
AFTER
insert or update of estado_bateria, estado_limpieza on "FLOTA"
for each row
DECLARE N1 NUMBER(3); N2 NUMBER(3); N3 NUMBER(3); N4 NUMBER(3); D VARCHAR2(9);
begin
  SELECT COUNT(*) INTO N1 FROM TECNICO WHERE estado='libre' AND seccion='limpieza';
  SELECT COUNT(*) INTO N2 FROM TECNICO WHERE estado='libre' AND seccion='recambio';
  SELECT COUNT(*) INTO N3 FROM TECNICO WHERE seccion='limpieza';
  SELECT COUNT(*) INTO N4 FROM TECNICO WHERE seccion='recambio';
  IF (:new.estado_bateria<10 AND N2>0) THEN
    SELECT MAX(dni) INTO D FROM TECNICO WHERE estado='libre' AND seccion='recambio';
    INSERT INTO se_asigna VALUES(:new.matricula,'carga',SYSTIMESTAMP,D,'en_proceso');
    UPDATE TECNICO SET estado='ocupado' WHERE dni=D;
  ELSIF (:new.estado_bateria<10 AND N4>0) THEN
    SELECT MAX(dni) INTO D FROM TECNICO WHERE seccion='recambio';
    INSERT INTO se_asigna VALUES(:new.matricula,'carga',SYSTIMESTAMP,D,'en_colas');
  END IF;
  IF (:new.estado_limpieza='sucia' AND N1>0) THEN
    SELECT MAX(dni) INTO D FROM TECNICO WHERE estado='libre' AND seccion='limpieza';
    INSERT INTO se_asigna VALUES(:new.matricula,'limpiar',SYSTIMESTAMP,D,'en_proceso');
    UPDATE TECNICO SET estado='ocupado' WHERE dni=D;
  ELSIF (:new.estado_limpieza='sucia' AND N3>0) THEN
    SELECT MAX(dni) INTO D FROM TECNICO WHERE seccion='limpieza';
    INSERT INTO se_asigna VALUES(:new.matricula,'limpiar',SYSTIMESTAMP,D,'en_colas');
  END IF;
end;
```

*Este trigger es algo más complejo por lo que intentaré resumir su funcionamiento. Básicamente este trigger cuando se inserta o se actualizan los campos estado de batería o estado de limpieza se activa y comprueba para cada tarea existente (limpieza y cargado) hay empleados que estén libres o no los hay. Sabiendo esto lo que hace es crear una tarea y asignarle el empleado libre claramente que pertenezca a la sección que se encarga de realizar la determinada tarea, y si no existen empleados libres, le añade la tarea a la cola a un empleado de todos los ocupados para que cuando no esté ocupado la realice.*

## Tabla Microalquiler

Una de las tareas a realizar era controlar automáticamente la disponibilidad de las motos sin que ningún usuario o administrador tenga que modificarlas. Por lo que tuvimos que añadir el siguiente trigger.

```
create or replace trigger "MICROALQUILER_DISP_FLOTA"
AFTER
insert or update on "MICROALQUILER"
for each row
begin
  IF (:new.estado=0 or :new.estado=1) THEN
    UPDATE flota
    SET disponibilidad='N'
    WHERE matricula=:new.matricula;
  ELSIF (:new.estado=2 or :new.estado=3) THEN
    UPDATE flota
```



```

        SET disponibilidad='S'
        WHERE matricula=:new.matricula;
    END IF;
end;

```

*Por la lógica y la definición de nuestro sistema, en la tabla microalquiler, el estado 0 o 1 indican que la moto esta reservada o utilizandose respectivamente por lo que en ese caso la disponibilidad de la moto es 'N' y en el caso de que el estado sea 2 o 3 que son finalizado o cancelado respectivamente, la moto ya puede estar disponible de nuevo.*

Otro comprobacion que realizamos en la propia base de datos para abstraerla de la implementacion de la interfaz es si un usuario intenta reservar una moto que no esta disponible pero por algun motivo puede visualizarla, por lo que:

```

create or replace trigger "MICROALQUILER_MOTO_EN_USO"
BEFORE
insert or update on "MICROALQUILER"
for each row
DECLARE N number(1);
begin
    SELECT COUNT(*) INTO N FROM flota WHERE matricula=:new.matricula AND disponibilidad='N';
    IF INSERTING THEN
        IF(N!=0) THEN
            RAISE_APPLICATION_ERROR(-102,'LA_MOTO_YA_ESTA_EN_USO');
        END IF;
    END IF;
    IF UPDATING THEN
        IF ((:old.estado=2 OR :old.estado=3) AND (:new.estado=0 OR :new.estado=1) AND N!=0) THEN
            RAISE_APPLICATION_ERROR(-102,'LA_MOTO_YA_ESTA_EN_USO');
        END IF;
    END IF;
end;

```

También decidimos comprobar desde el propio sistema de base de datos que un mismo usuario no pueda estar con dos motos reservadas a la vez, por lo que:

```

create or replace trigger "MICROALQUILER_USUARIO_USO"
BEFORE
insert on "MICROALQUILER"
for each row
DECLARE N NUMBER(1);
begin
    SELECT COUNT(*) INTO N FROM MICROALQUILER WHERE dni=:new.dni AND (estado=0 OR estado=1);
    IF N!=0 THEN
        RAISE_APPLICATION_ERROR(-103,'USUARIO_YA_TIENE_UNA_RESERVA');
    END IF;
end;

```

Por último creamos un trigger que permite facilitar un poco la implementacion ya que automáticamente asigna a una moto en la tabla flota su posicion cuando el microalquiler finaliza:

```

create or replace trigger "MICROALQUILER_ACT_ZONA"
AFTER
insert or update of estado on "MICROALQUILER"
for each row
begin
    IF (:new.estado=2) THEN
        UPDATE flota

```

```

        SET localizacion=:new.posicion_fin
        WHERE matricula=:new.matricula;
    END IF;
end;

```

## Tabla Se Asigna

En esta tabla lo primero que tuvimos en cuenta fué que a un técnico que no sea de la seccion correcta para realizar una determinada tarea, no podía relacionarse con dicha tarea, por lo que creamos el siguiente trigger:

```

create or replace TRIGGER "SE_ASIGNA_DEPARTAMENTOS"
BEFORE
insert or update on "SE_ASIGNA"
for each row
DECLARE T1 NUMBER(1); T2 NUMBER(1); T3 NUMBER(1);
begin
    SELECT COUNT(*) INTO T1 FROM TECNICO WHERE dni=:new.dni AND seccion='limpieza';
    SELECT COUNT(*) INTO T2 FROM TECNICO WHERE dni=:new.dni AND seccion='recambio';
    SELECT COUNT(*) INTO T3 FROM TECNICO WHERE dni=:new.dni AND seccion='posicion';
    IF :new.tipo='limpiar' AND T1=0 THEN
        RAISE_APPLICATION_ERROR(-104,'EL TECNICO NO PERTENECE AL DEPARTAMENTO');
    ELSIF :new.tipo='recambio' AND T2=0 THEN
        RAISE_APPLICATION_ERROR(-104,'EL TECNICO NO PERTENECE AL DEPARTAMENTO');
    ELSIF :new.tipo='posicion' AND T3=0 THEN
        RAISE_APPLICATION_ERROR(-104,'EL TECNICO NO PERTENECE AL DEPARTAMENTO');
    ELSIF :new.tipo='carga' AND T2=0 THEN
        RAISE_APPLICATION_ERROR(-104,'EL TECNICO NO PERTENECE AL DEPARTAMENTO');
    END IF;
end;

```

*Como podemos ver en el código, este trigger comprueba cuando se va a insertar una tupla que el tecnico pueda realmente realizarla porque pertenezca a la seccion correspondiente, cancelando esta inserción en caso de que no sea correcta.*

Como hemos comentado en toda la parte de los triggers, nuestra principal tarea era controlar la disponibilidad de las motos automáticamente por lo que tuvimos que crear el siguiente trigger:

```

create or replace trigger "SE_ASIGNA_DISP_MOTO"
AFTER
update of estado on "SE_ASIGNA"
for each row
begin
    IF :new.estado='en_proceso' THEN
        UPDATE flota
        SET disponibilidad='N'
        WHERE matricula=:new.matricula;
        UPDATE TECNICO
        SET estado='ocupado'
        WHERE dni=:new.dni;
    ELSIF :new.estado='finalizada' THEN
        UPDATE flota
        SET disponibilidad='S'
        WHERE matricula=:new.matricula;
        UPDATE tecnico
        SET estado='libre'
        WHERE dni=:new.dni;
    end if;
end;

```

```

IF :new.tipo='limpiar' THEN
    UPDATE flota
    SET estado_limpieza='limpia'
    WHERE matricula=:new.matricula;
ELSIF :new.tipo='carga' THEN
    UPDATE flota
    SET estado_bateria=100
    WHERE matricula=:new.matricula;
END IF;
END IF;
end;

```

*Este trigger controla el estado en la tabla se asigna, que puede ser 'en cola', 'en proceso' y 'finalizada'. Lo que hace es actualizar la disponibilidad de la flota dependiendo del estado en el que se encuentre la tarea ya que si una tarea de limpieza o de carga esta en proceso, la moto no debe estar disponible, y cuando una tarea pasa a 'finalizada', se actualiza tanto el estado de la moto como la propia definicion de la tarea ya que si la tarea es limpiar, pone el estado de la moto en flota como limpia y si es de carga de bateria, pone la bateria al 100.*

Por último, tambien decidimos controlar en la base de datos que no se pueda asignar una tarea para ponerla como en proceso a un tecnico que este ocupado, ya que si el tecnico esta ocupado, solo se le pueden asignar tareas a la cola no ponerlas en proceso.

```

create or replace trigger "SE_ASIGNA_TECNICO_OCUPADO"
BEFORE
insert or update of estado on "SE_ASIGNA"
for each row
DECLARE E NUMBER(1);
begin
SELECT COUNT(*) INTO E FROM TECNICO WHERE dni=:new.dni AND estado='ocupado';
IF (E>0 and :new.estado='en_proceso') THEN
RAISE_APPLICATION_ERROR(-106,'EL_TECNICO_ESTA_OCUPADO');
END IF;
end;

```

## Interfaz de administración: Oracle Application Builder

Para implementar la interfaz de administración usamos Oracle Application Builder. Esta herramienta nos da la posibilidad de crear aplicaciones con interfaz gráfica para interactuar con nuestra base de datos de Oracle Database, de una forma gráfica e intuitiva.

Para nuestra interfaz de administración hemos elegido una serie de funcionalidades de varios subsistemas, cuya implementación nos ayudará a imaginar la interfaz completa del sistema. Estas funcionalidades son:

- **Insertar nueva moto:** esta funcionalidad nos permite insertar mediante una sencilla interfaz una nueva tupla a nuestra tabla de flota, especificando los atributos que tendrá nuestra nueva moto. De forma similar, se implementarían el resto de funcionalidades para añadir tuplas al resto de nuestras tablas, de cara a la implementación de nuestro sistema completo.
- **Mostrar motos sucias y mostrar motos con batería baja:** estas dos funcionalidades nos permite consultar las matrículas de las motos con estado de limpieza 'sucia' y estado















de batería menor a 20, con objetivo de gestionar dichos recursos. De forma similar a esta, podremos implementar cualquier funcionalidad de consulta sobre nuestra base de datos.

- **Tareas asignadas técnico y microalquileres:** estas funcionalidades, al igual que las dos anteriores nos permite realizar una consulta sobre las tareas y los microalquileres respectivamente, en el caso de las tareas nos muestra todas las tareas asignadas a técnicos existentes en nuestra base de datos, y en el caso de microalquileres, todos los microalquileres. Además, permite realizar subconsultas de forma interactiva con la interfaz para filtrar los resultados por cualquier atributo de cada tabla, con el objetivo de que los empleados puedan realizar consultas más personalizadas, por ejemplo que un técnico pueda consultar las tareas que le quedan pendientes, o que un administrador pueda ver las tareas que se encuentran en proceso en cualquier momento, entre otras muchas utilidades.
- **Actualizar motos y actualizar tareas:** en este caso, se nos muestra una tabla con los respectivos atributos de flota o de tarea, y además, se nos permite modificar el valor de sus atributos, o directamente eliminar tuplas de nuestra tabla, de forma interactiva con nuestra interfaz. El objetivo es que los empleados puedan modificar atributos concretos cuando realicen alguna acción, por ejemplo cambiar el estado de limpieza de una moto cuando esta se limpie, o cambiar el estado de una tarea cuando esta sea realizada.

## Página de inicio

Esta es la página que nos aparecerá cuando ejecutamos nuestra aplicación. Su función es mostrarnos las diferentes funcionalidades que la aplicación nos ofrece y darnos acceso a ellas.

Para su implementación usamos uno de los objetos que nos ofrece Oracle Application Builder, una lista de navegación, la cual es una lista con una serie de etiquetas y con un número de pagina asociado a cada etiqueta que será la página a la que nos direccionará cuando pulsemos en dicha etiqueta. Podemos ver la forma de dicha lista en la siguiente imagen:

Sequence	Name	Parent Entry	Target	Conditional	Last Updated	Level	Copy
	10 <a href="#">Introducir nueva moto</a>	-	f?p=&APP_ID.:4:&SESSION.::&DEBUG.::::	-	3 hours ago	1	
	20 <a href="#">Motos batería baja</a>	-	f?p=&APP_ID.:5:&SESSION.::&DEBUG.::::	-	3 hours ago	1	
	30 <a href="#">Tareas asignadas técnico</a>	-	f?p=&APP_ID.:8:&SESSION.::&DEBUG.::::	-	3 hours ago	1	
	40 <a href="#">Motos sucias</a>	-	f?p=&APP_ID.:9:&SESSION.::&DEBUG.::::	-	3 hours ago	1	
	50 <a href="#">Actualizar motos</a>	-	f?p=&APP_ID.:10:&SESSION.::&DEBUG.::::	-	86 minutes ago	1	
	60 <a href="#">Actualizar tareas</a>	-	f?p=&APP_ID.:11:&SESSION.::&DEBUG.::::	-	56 minutes ago	1	
	70 <a href="#">Microalquileres</a>	-	f?p=&APP_ID.:12:&SESSION.::&DEBUG.::::	-	48 minutes ago	1	

1 - 7

Una vez creada la lista, la asociamos a la página de inicio, y el resultado final será el siguiente:



## Insertar nueva moto

En este caso, crearemos una página con un formulario en modo tabla, y sus respectivos botones, lo cual nos permite introducir los atributos de una moto y añadir esta tupla a la tabla flota. El resultado de esta página sería el siguiente:

 A form titled "Introducir Nueva Moto" with the following fields:
 

- Matricula:
- \*Coste Alquiler:
- \*Empresa Alquiler:
- \*Estado Limpieza:
- \*Localizacion:
- Disponibilidad:
- Estado Moto:
- \*Estado Bateria:

## Mostrar motos sucias y mostrar motos con batería baja

En este caso, creamos una nueva página con un informe estático el cual nos permite visualizar en una tabla la consulta que le indiquemos sobre nuestra base de datos. En este caso buscamos mostrar las motos cuyo estado de limpieza es 'sucio' y, por otra parte, las motos cuyo estado de batería es inferior a 10. Para ello, hemos realizado las siguientes consultas en sql:

```

select  "FLOTA"."MATRICULA" as "MATRICULA",
        "FLOTA"."ESTADO_LIMPIEZA" as "ESTADO_LIMPIEZA"
  from  "FLOTA" "FLOTA"
 where  "FLOTA"."ESTADO_LIMPIEZA" ='sucia'

select  "FLOTA"."MATRICULA" as "MATRICULA",
        "FLOTA"."ESTADO_BATERIA" as "ESTADO_BATERIA"
  from  "FLOTA" "FLOTA"
 where  "FLOTA"."ESTADO_BATERIA" < 20

```

Finalmente, el estado final de nuestra página con el informe sería el siguiente:

Motos sucias	
<u>MATRICULA</u> ▼	<u>ESTADO LIMPIEZA</u>
0123ABC	sucia
321	sucia
1 - 2	

Motos batería baja	
<u>MATRICULA</u> ▼	<u>ESTADO BATERIA</u>
123	12
123456	13
1 - 2	

## Tareas asignadas técnico y microalquileres

En este caso, en otras nuevas páginas, haremos uso del informe dinámico, el cual añade la funcionalidad de realizar filtrados en base a los atributos de la tabla que tengamos, al informe estático. En este caso esta opción cumple satisfactoriamente con la funcionalidad que buscamos, que es poder

ofrecer a los empleados un método de consulta mas personalizado sobre alguno de los recursos que les concierne. Estas páginas quedarían de la siguiente forma:

Tareas asignadas a técnicos

	Dni	Tipo	Estado	Matricula	Fecha
	9658785	carga	en cola	123	2018-11-01
	12345678J	limpiar	en cola	0123ABC	2018-01-09
	12345678J	limpiar	en proceso	321	2018-01-09

1 - 3

Microalquileres

Volver

	Dni	Fecha Reserva	Estado	Fecha Inicio	Fecha Fin	Posicion Inicio	Posicion Fin	Tarifa	Matricula
	654654654	2018-01-07	2	2018-01-07	2018-01-07	37.18702305656495 -3.6162185668945312	37.17874880934861 -3.6090946197509766	.25	0125ABC
	654654654	2018-01-07	2	2018-01-07	2018-01-07	37.17922750950913 -3.6084938049316406	37.18702305656495 -3.6162185668945312	.25	0125ABC

1 - 2

## Actualizar motos y tareas

En este caso, usaremos otro tipo de formulario, el formulario tabular, el cual nos permite tener una visión genérica sobre las tuplas de nuestras tablas y poder modificar atributos y borrar tuplas de una forma muy directa. El estado final sería el siguiente:

Actualizar motos

	Matricula	Coste Alquiler	Empresa Alquiler	Estado Limpieza	Localizacion	Disponibilidad	Estado Moto	Estado Bateria
<input type="checkbox"/>	0126ABC	2	asdd	limpia	37.18093712817645 -3	S	bien	100
<input type="checkbox"/>	0127ABC	2	asdsdd	limpia	37.196578342147134 -	S	bien	100
<input type="checkbox"/>	123	9	sadfas	limpia	sdfgasdf	S	bien	12
<input type="checkbox"/>	123456	5	sdfg	limpia	asd345	S	bien	13
<input type="checkbox"/>	0123ABC	2	asdfas	sucia	37.19023677593392 -3	N	bien	100
<input type="checkbox"/>	0124ABC	2	asdfas	limpia	37.170542049160446 -	S	bien	100
<input type="checkbox"/>	0125ABC	1	asdfas	limpia	37.17874880934861 -3	S	bien	100
<input type="checkbox"/>	321	4	asd	sucia	asdfasfd234234	N	bien	96

1 - 8

Actualizar tareas

Cancelar

Borrar

Guardar

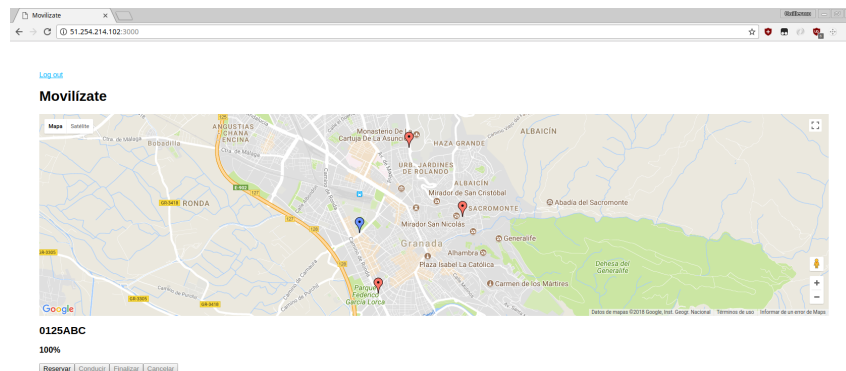
<input type="checkbox"/>	Matricula	Tipo	Fecha	Dni	Estado
<input type="checkbox"/>	0123ABC	limpiar	2018-01-09	12345678J	en cola
<input type="checkbox"/>	321	limpiar	2018-01-09	12345678J	en proceso

1 - 2

## Interfaz de usuario: NodeJS

Para implementar el servidor web utilizamos NodeJS V8, con dos módulos esenciales, ExpressJS[4] (framework web para Node) y node-oracledb[5], un módulo desarrollado por Oracle para abstraer la lógica de la conexión con la base de datos, y en el estilo de Javascript ejecutar instrucciones sobre la DB.

La funcionalidad elegida es alquilar motos, el usuario debe poder autenticarse, y una vez verificada su identidad en base a la tabla de usuarios, se le mostrarán en un mapa las motos disponibles (en base a la definición de disponibilidad definida en las restricciones) con la información correspondiente (matrícula y batería) y aquí el usuario podrá, en este orden, reservarlas, confirmar la reserva o cancelarla y finalizar la conducción. Para la prueba de concepto, en vez de utilizar el GPS del dispositivo para implementar el desplazamiento del usuario porque sería terriblemente molesto, hemos implementado una función de drag&drop sobre la moto que estamos conduciendo.



Utiliza cookies para determinar cuando te conectas a la web si estás verificado, en caso contrario te dirige a la venta de identificación. Una vez en al ventana de usuario se realizan dos consultas como máximo.

1. Se comprueba si el usuario tiene alguna moto reservada o en uso

```
select DNI,ESTADO,MATRICULA,LOCALIZACION,ESTADO_BATERIA from MICROALQUILER natural
↪ join FLOTA where DNI = '${dni}' and ESTADO < 2;
```

2. En caso contrario se listan las motos disponibles

```
select MATRICULA,LOCALIZACION,ESTADO_BATERIA from FLOTA WHERE DISPONIBILIDAD = 'S'
↪ ;
```



La localización se almacena como una cadena con dos coordenadas en grados separadas por un espacio. Cada uno de los botones se implementa como un formulario con campos invisibles que apunta a una dirección que acepta peticiones POST, hay cuatro de estas **/reservar**, **/conducir**, **/cancelar**, **/finalizar**, veamos por ejemplo el código de alguna de estos *endpoints*.

```
router.post('/conducir',function(req,res,next){
  var req_body = {
    dni: req.cookies.user_id,
    matricula: req.body.matricula
  }
  var q_str = 'update MICROALQUILER set ESTADO=1, FECHA_INICIO=sysdate, POSICION_INICIO=(
    ↪ select LOCALIZACION from FLOTA where MATRICULA = '${req_body.matricula}') where (DNI,
    ↪ FECHA_RESERVA) in (select DNI,FECHA_RESERVA from MICROALQUILER where ESTADO = 0 and
    ↪ DNI='${req_body.dni}')';
  //console.log(req_body,q_str);
  queryDB(
    q_str,
    {

  },
  function(re){
    //console.log(re);
    res.redirect(req.get('Referer'));
  }
  )
});
```

Como se puede observar el funcionamiento es muy sencillo, una vez recibida la petición POST se forma una cadena que será la sentencia SQL a ejecutar, se realiza y una vez finalizada se redirige al usuario otra vez a la página original (desde la que venía). No se manda al usuario antes de completar la operación para evitar que la información no se haya actualizado aún. La función queryDB es relativamente sencilla también.

```
function queryDB(query_text,n_args,cb,er_cb){
  oracledb.getConnection(
    {
      user : dbConfig.user,
      password : dbConfig.password,
      connectString : dbConfig.connectString
    },
    function(err, connection)
    {
      if (err) {
        console.error("1" + err.message);
        return;
      }
      connection.execute(
        // The statement to execute
        query_text,
        n_args,
        { autoCommit: true },
        function(err, result)
        {
          if (err) {
            console.error("2" + err.message);
            doRelease(connection);
            if(er_cb !== undefined) er_cb(err);
            return;
          }
        }
      )
    }
  )
}
```

```

        cb(result.rows);
        doRelease(connection);
    });
}

```

Además de la cadena, le mandamos los argumentos para sustituir las variables (evitando la inyección sql), la función a la que llamar una vez completado y la función de error. Vemos un ejemplo de uso a continuación, las variables a sustituir se prefijan con `::`

```

router.post('/reservar',function(req,res,next){
    var req_body = {
        dni: req.cookies.user_id,
        matricula: req.body.matricula
    }
    var q_str = "insert into MICROALQUILER (DNI,FECHA_RESERVA,ESTADO,TARIFA,MATRICULA) values (:
        ↪ dni,sysdate,0,0.25,:matricula)";
    queryDB(
        q_str,
        {
            dni: {val: req_body.dni},
            matricula: {val: req_body.matricula}
        },
        function(re){
            res.redirect(req.get('Referer'));
        }
    )
});

```

Como se puede apreciar en queryDB la información de autenticación se almacena en un archivo separado que contiene lo siguiente:

```

module.exports = {
    user : process.env.NODE_ORACLEDB_USER || "MOVILIZATE",

    // Instead of hard coding the password, consider prompting for it,
    // passing it in an environment variable via process.env, or using
    // External Authentication.
    password : process.env.NODE_ORACLEDB_PASSWORD || "practicas",

    // For information on connection strings see:
    // https://github.com/oracle/node-oracledb/blob/master/doc/api.md#connectionstrings
    connectString : process.env.NODE_ORACLEDB_CONNECTIONSTRING || "//vps497554.ovh.net:1521/XE:
        ↪ DEDICATED",

    // Setting externalAuth is optional. It defaults to false. See:
    // https://github.com/oracle/node-oracledb/blob/master/doc/api.md#extauth
    externalAuth : process.env.NODE_ORACLEDB_EXTERNALAUTH ? true : false
};

```

La aplicación al ser un prototipo tiene muchos puntos de crítica, el primero, por simplicidad no hasheadamos las contraseñas; las cookies no están firmadas, entonces cualquiera puede manipularlas y suplantar la identidad de un usuario; en algunas funciones se utiliza la sustitución de variables segura y en otras no, permitiendo la inyección de código SQL, esto se debe a que esta funcionalidad la descubrí a mitad del desarrollo y como el objetivo era sencillamente crear algo funcional dejé las anteriores como estaban; se podrían sustituir las peticiones POST por llamadas AJAX, que permitirían actualizar sólo la porción de página de interés y no tener que cargar toda la página por

completo, pero añadía una dificultad que transcendía de nuevo el objetivo de la práctica. Por lo demás la interfaz de usuario emula relativamente bien las condiciones de un sistema de producción y podemos pensarla como un frame dentro de una interfaz más compleja.

He decidido no poner todos los detalles del código al relativamente extenso y no ser de relevancia para la práctica. De todos modos, el código se puede consultar en el archivo adjunto al PDF y la página web se puede montar bien en el servidor local instalando NodeJS y montando la DB o visitarla hasta el 25 de Enero del 2018 en el enlace 51.254.214.102:3000, con el usuario **12345678Z** y la contraseña **practicassddsi**.

## Referencias

- [1] ERP y CRM de código abierto — Odoo, Odoo.com, [https://www.odoo.com/es\\_ES/](https://www.odoo.com/es_ES/), Accedido el 15 de enero de 2018.
- [2] Oracle Database 11g Express Edition, Oracle Corp, <http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/index.html>, Accedido el 15 de enero de 2018.
- [3] Node.js, Node.js Foundation, <https://nodejs.org/es/>, Accedido el 15 de enero de 2018.
- [4] Express - Infraestructura de aplicaciones web Node.js, Node.js Foundation, <http://expressjs.com/es/>, Accedido el 15 de enero de 2018.
- [5] oracle/node-oracledb: Oracle Database driver for Node.js, Oracle Corp, <https://github.com/oracle/node-oracledb>, Accedido el 15 de enero de 2018.
- [6] Oracle PL/SQL, Triggers documentation, Oracle Corp, <https://docs.oracle.com/cloud/latest/db112/LNPLS/triggers.htm#LNPLS020> Accedido el 15 de enero de 2018