

Práctica 1: Descripción del sistema, análisis y especificación de requisitos

Descripción del problema

Deseamos crear Modest, un sistema de información para gestionar el Hotel Modesto. A continuación se explican pormenorizadamente los subsistemas que se crearán y sus funcionalidades.

Los subsistemas creados y la organización de esta práctica es la siguiente.

1. Subsistema de gestión de empleados: Paula Villanueva Núñez.
2. Subsistema de gestión de clientes y reservas: José Luis Ruiz Benito.
3. Subsistema de mantenimiento: Ana Buendía Ruiz-Azuaga.
4. Subsistema de gestión de eventos: Antonio Merino Gallardo.

Subsistema de gestión de empleados

De cada empleado se almacenarán los siguientes datos: DNI (8 dígitos y un carácter), nombre y apellidos (cadenas de 20 y 40 caracteres), nº de la seguridad social (8 dígitos), teléfono (9 dígitos), puesto de trabajo, una cuenta bancaria donde ingresar el salario, fecha de nacimiento y si está dado de alta. El puesto del empleado puede ser: recepcionista, empleado de limpieza, empleado de mantenimiento, gestor de eventos y gerente. Un empleado debe ser mayor de 16 años y sólo puede haber un gerente en el sistema.

Para dar de alta un empleado, se incluirán todos estos datos en el sistema. La información de un empleado podrá ser también modificada, aportando al sistema la información actualizada. Naturalmente, se podrá consultar la ficha de un empleado. Cuando el empleado deje de trabajar en el hotel, será dado de baja del sistema. Todas estas acciones serán realizadas por el gerente del hotel.

Subsistema de gestión de clientes y reservas

En cuanto a las reservas, se guardará el DNI del cliente (9 dígitos y un carácter), el tipo de habitación, la fecha de entrada y la fecha de salida. El tipo de habitación puede ser simple, doble y suite.

Cuando un cliente solicita una reserva, el recepcionista comprueba si está registrado en el sistema y, en caso de no estarlo, procede a registrarlo. Seguidamente, procederá a realizar la reserva con los datos de la misma. Para cancelar una reserva existente el cliente deberá informar al recepcionista. Cuando el cliente llegue al hotel, se realizará el check-in, esto es, el recepcionista anotará en el sistema su llegada y se le asignará una habitación concreta de entre las que estén libres y se marcará como ocupada. Cuando finalice la estancia del

cliente, se hará el check-out correspondiente, marcando como libre la habitación en la que se ha alojado. Un check-out debe tener siempre un check-in asociado. Por otra parte, el recepcionista podrá consultar la disponibilidad de cualquier tipo de habitación en todo momento, aportando al sistema el tipo de habitación y las fechas entre las cuales se desea consultar su disponibilidad.

Subsistema de mantenimiento

El sistema guardará información acerca de los avisos de reparación. Un aviso de reparación se compone de un identificador de habitación en la que se realizará la reparación (entero de tres dígitos), una descripción (cadena de hasta 500 caracteres), la fecha en la que se ha notificado el aviso (tipo date) y si la reparación está resuelta. Por otra parte, se llevará un registro de limpieza de habitaciones, que contendrá información de cada turno de limpieza, es decir, empleado que la realiza, habitación y fecha. Un empleado de la limpieza se encargará de un máximo de 10 habitaciones al día. Asimismo, el sistema contará con un registro de provisiones. De cada producto se almacenará su nombre (50 caracteres) y cantidad (entero).

Cuando un cliente descubre una avería, el recepcionista registrará un aviso de reparación en el sistema y dicha reparación la realizará el empleado de mantenimiento con la mayor brevedad posible. Cuando el problema sea subsanado, el recepcionista lo indicará al sistema, que marcará la reparación como “resuelta”.

El gerente, planificará periódicamente la limpieza de las habitaciones del hotel. Esto consistirá en indicarle al sistema el empleado, habitación y fecha que corresponda para cada acción de limpieza.

En cuanto al registro de provisiones, cuando el empleado de mantenimiento solicite la inclusión de nuevos productos, se añadirán al sistema si no había anteriormente o se incrementará su cantidad, según corresponda.

Subsistema de gestión de eventos

En el sistema se almacenará información de los grupos de excursión que se organicen, incluyendo los DNIs de los clientes que conforman el grupo, el identificador del guía turístico que va a dirigir la excursión (cadena 9 caracteres), una descripción de la misma (cadena de 500 caracteres) y la fecha y hora en que se va a realizar. Del mismo modo, se llevará un registro de los distintos eventos que se celebren en el hotel, indicando el identificador de la sala en la que se llevará a cabo (entero de tres dígitos), los identificadores de los empleados que lo controlarán (DNI), una descripción (cadena de hasta 500 caracteres) y la fecha y hora de su comienzo.

Cuando se organizan dichos grupos de excursión y eventos es el gestor de eventos el que los introduce en el sistema indicando la información ya detallada. Asimismo, cuando el gestor de eventos desee promocionar un evento determinado, lo indicará al sistema para así componer un correo con los datos del mismo que será enviado a todos los clientes que están registrados en el sistema.

Análisis de requisitos

Requisitos funcionales

1. Subsistema de gestión de empleados

- RF1.1: Dar de alta a un empleado.
 - Entrada: Agente externo: gerente. Acción: solicitar inserción. Requisito de datos de entrada RDE1.1
 - BD: Requisito de datos de escritura RDW1.1
 - Salida: Agente externo: gerente. Acción: confirmación resultado. Requisito de datos de salida: ninguno.
- RF1.2: Modificar ficha de un empleado.
 - Entrada: Agente externo: gerente. Acción: solicitar modificación. Requisito de datos de entrada RDE1.2.
 - BD: Requisito de datos de lectura RDR1.2 y escritura RDW1.2.
 - Salida: Agente externo: gerente. Acción: confirmación resultado. Requisito de datos de salida: ninguno.
- RF1.3: Consultar ficha de un empleado.
 - Entrada: Agente externo: gerente. Acción: solicitar ficha. Requisito de datos de entrada RDE1.3.
 - BD: Requisito de datos de lectura RDR1.1.
 - Salida: Agente externo: gerente. Acción: confirmación resultado. Requisito de datos de salida: RDS1.1.
- RF1.4: Dar de baja a un empleado.
 - Entrada: Agente externo: gerente. Acción: solicitar borrado. Requisito de datos de entrada RDE1.4.
 - BD: Requisito de datos de lectura RDR1.3 y escritura RDW1.3.
 - Salida: Agente externo: gerente. Acción: confirmación resultado. Requisito de datos de salida: ninguno

2. Subsistema de gestión de clientes y reservas

- RF2.1: Registrar una reserva
 - Entrada: Agente externo: cliente. Acción: Incluir una reserva en el sistema de información. Requisito de datos de entrada RDE2.1.
 - BD: Requisito de datos de escritura RDW2.1
 - Salida: Agente externo: cliente. Acción: confirma el resultado. Requisito de datos de salida: RDS2.1.
- RF2.2: Realizar check in
 - Entrada: Agente externo: cliente. Acción: confirmar la llegada de un cliente que tenía una reserva. Requisito de datos de entrada RDE2.2.
 - BD: Requisito de datos de lectura RDR2.2 y escritura RDW2.2
 - Salida: Agente externo: cliente. Acción: Confirmar el resultado. Requisito de datos de salida RDS2.2.
- RF2.3: Realizar check out

- Entrada: Agente externo: cliente. Acción: confirmar la salida de un cliente que se ha hospedado en el hotel. Requisito de datos de entrada RDE2.3.
- BD: Requisito de datos de lectura RDR2.3 y escritura RDW2.3.
- Salida: Agente externo: cliente. Acción: Confirmar el resultado.
- **RF2.4:** Cancelar una reserva
 - Entrada: Agente externo: cliente. Acción: cancelar una reserva. Requisito de datos de entrada RDE2.4.
 - BD: Requisitos de datos de lectura RDR2.4 y escritura RDW2.4
 - Salida: Agente externo: cliente. Acción: confirmar resultado. Requisito de datos de salida: ninguno.
- **RF2.5:** Consultar disponibilidad de un tipo de habitación
 - Entrada: Agente externo: Recepcionista. Acción: Solicitar consulta de disponibilidad de un tipo de habitación en una fecha determinada. Requisito de datos de entrada RDE2.5
 - BD: Requisito de datos de lectura: RDR2.1
 - Salida: Agente externo: Recepcionista. Acción: Comunicar disponibilidad. Requisito de datos de salida: RDS2.3

3. Subsistema de gestión de mantenimiento

- **RF3.1:** Registrar aviso de reparación
 - Entrada: Agente externo: cliente. Acción: Solicitar inserción de reparación necesaria. Requisito de datos de entrada RDE3.1.
 - BD: Requisito de datos de lectura RDR3.1 y escritura RDW3.1.
 - Salida: Agente externo: cliente. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.
- **RF3.2:** Marcar reparación como resuelta.
 - Entrada: Agente externo: empleado de mantenimiento. Acción: Marcar una reparación como resuelta. Requisito de datos de entrada RDE3.2.
 - BD: Requisito de datos de lectura RDR3.2 y escritura RDW3.2.
 - Salida: Agente externo: empleado de mantenimiento. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.
- **RF3.3:** Planificación de limpieza por habitación
 - Entrada: Agente externo: Gerente. Acción: Asignar a un empleado la limpieza de una habitación a una hora determinada. Requisito de datos de entrada RDE3.3.
 - BD: Requisito de datos de lectura RDR3.3 y RDR3.4 y escritura RDW3.3.
 - Salida: Agente externo: Gerente. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.
- **RF3.4:** Registro de provisiones
 - Entrada: Agente externo: empleado de mantenimiento. Acción: Solicitar inserción de nuevas provisiones. Requisito de datos de entrada RDE3.4.
 - BD: Requisito de datos de lectura RDR3.5 y escritura RDW3.4.
 - Salida: Agente externo: empleado de mantenimiento. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.

4. Subsistema de gestión de eventos

- RF4.1: Organizar grupo de excursión
 - Entrada: Agente externo: Gestor de eventos. Acción: Incluir grupo de excursión. Requisito de datos de entrada RDE4.1
 - BD: Requisito de datos de lectura RDR4.3 y RDR4.4 y escritura RDW4.1.
 - Salida: Agente externo: Gestor de eventos. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.
- RF4.2: Organizar evento en el hotel
 - Entrada: Agente externo: Gestor de eventos. Acción: Incluir evento en el hotel. Requisito de datos de entrada RDE4.2
 - BD: Requisito de datos de lectura RDR4.5 y RDR4.6 y escritura RDW4.2.
 - Salida: Agente externo: Gestor de eventos. Acción: Confirmación del resultado. Requisito de datos de salida: ninguno.
- RF4.3: Publicitar evento: envío de un folleto publicitarios a todos los clientes por email.
 - Entrada: Agente externo: Gestor de eventos. Acción: Requisito de datos de entrada RDE4.3
 - BD: Requisito de datos de lectura RDR4.1 y RDR4.2
 - Salida: Agente externo: Gestor de eventos. Acción: Mandar datos del evento a los correos de los clientes. Requisito de datos de salida: ninguna

Requisitos de datos

1. Subsistema de gestión de empleados

- RDE1.1: Datos de entrada de alta de empleado
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadenas de caracteres (40)
 - Teléfono: Cadena de caracteres (9)
 - DNI: Cadena de caracteres (9)
 - Alta: boolean
 - Fecha de nacimiento: date
 - Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
 - Número de la seguridad social: Cadena de caracteres (8)
 - Cuenta bancaria: Cadena de caracteres (24)
- RDW1.1: Datos almacenados de empleado.
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadenas de caracteres (40)
 - Teléfono: Cadena de caracteres (9)
 - DNI: Cadena de caracteres (9)
 - Alta: boolean
 - Fecha de nacimiento: date

- Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
 - Número de la seguridad social: Cadena de caracteres (8)
 - Cuenta bancaria: Cadena de caracteres (24)
- RDE1.2: Datos de entrada de modificación de empleado.
 - DNI: Cadena de caracteres (9)
 - Datos a modificar:
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadenas de caracteres (40)
 - Teléfono: Cadena de caracteres (9)
 - DNI: Cadena de caracteres (9)
 - Alta: boolean
 - Fecha de nacimiento: date
 - Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
 - Número de la seguridad social: Cadena de caracteres (8)
 - Cuenta bancaria: Cadena de caracteres (24)
- RDW1.2: Datos almacenados de empleado.
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadenas de caracteres (40)
 - Teléfono: Cadena de caracteres (9)
 - DNI: Cadena de caracteres (9)
 - Alta: boolean
 - Fecha de nacimiento: date
 - Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
 - Número de la seguridad social: Cadena de caracteres (8)
 - Cuenta bancaria: Cadena de caracteres (24)
- RDE1.3: Datos de entrada de empleado.
 - DNI: Cadena de caracteres (9)
- RDR1.1: Datos de empleado almacenado.
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadenas de caracteres (40)
 - Teléfono: Cadena de caracteres (9)
 - DNI: Cadena de caracteres (9)
 - Alta: boolean
 - Fecha de nacimiento: date
 - Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
 - Número de la seguridad social: Cadena de caracteres (8)
 - Cuenta bancaria: Cadena de caracteres (24)
- RDS1.1: Ficha de un empleado.
 - Nombre: Cadena de caracteres (20)

- Apellidos: Cadenas de caracteres (40)
- Teléfono: Cadena de caracteres (9)
- DNI: Cadena de caracteres (9)
- Alta: boolean
- Fecha de nacimiento: date
- Puesto de trabajo: Enumeración (gerente, recepcionista, gestor de eventos, limpieza, mantenimiento)
- Número de la seguridad social: Cadena de caracteres (8)
- Cuenta bancaria: Cadena de caracteres (24)
- RDE1.4: Datos de entrada de baja de contacto.
 - DNI: Cadena de caracteres (9)
- RDW1.3: Datos almacenados de baja de contacto.
 - Alta: boolean
- RDR1.2: Consulta de empleado.
 - DNI: Cadena de caracteres (9)
- RDR1.3: Consulta de empleado.
 - DNI: Cadena de caracteres (9)

2. Subsistema de gestión de clientes y reservas

- RDE2.1: Datos de entrada del registro de una reserva.
 - DNI cliente: cadena de caracteres (9)
 - Tipo de habitación reservada: cadena de caracteres (20)
 - Fecha de entrada: date
 - Fecha de salida: date
- RDW2.1: Datos almacenados del registro de reserva.
 - DNI cliente: cadena de caracteres (9)
 - Tipo de habitación reservada: cadena de caracteres (20)
 - Fecha de entrada: date
 - Fecha de salida: date
 - Identificador de la reserva (lo establece el sistema): cadena de caracteres (20)
- RDS2.1: Datos asociados al registro de la reserva.
 - Identificador de la reserva: cadena de caracteres (20)
- RDE2.2: Datos de entrada al realizar check in
 - Fecha entrada de la reserva: date
 - Identificador de reserva: cadena de caracteres (20)
- RDW2.2: Datos almacenados al realizar el check in.
 - Datos almacenado en el registro de reserva
 - Check-in: boolean
 - Identificador de la habitación (lo establece el sistema): entero de 3 dígitos

- Datos almacenados en la ficha de la habitación:
 - Ocupada: boolean
- RDS2.2: Datos asociados a la reserva de un cliente en el check in.
 - Identificador de la habitación: entero de 3 dígitos.
- RDE2.3: Datos de entrada al realizar el check out.
 - Identificador de la habitación: entero de 3 dígitos.
- RDW2.3: Datos almacenado en la ficha de la habitación
 - Ocupada: boolean
 - Check-out: boolean
- RDE2.4: Datos de entrada al cancelar una reserva.
 - Identificador de la reserva: cadena de caracteres (20)
- RDW2.4: Datos almacenados para modificar los datos de una reserva
 - Anulada: boolean
- RDE2.5: Datos de entrada para consultar la disponibilidad de una habitación.
 - Identificador de tipo de habitación
 - Fecha: tipo date
- RDR2.1: Datos almacenados de las reservas asociadas al tipo de habitación introducido.
 - Fecha: date.
 - Identificador de tipo de habitación: cadena de caracteres (20).
- RDS2.3: Disponibilidad de la habitación en la fecha introducida. Número de habitaciones disponibles en las fechas indicadas del tipo requerido.
 - Número de habitaciones: entero.
- RDR2.2: Consulta de reserva
 - Identificador de reserva: cadena de caracteres(20).
- RDR2.3: Consulta de reserva
 - Identificador de habitación: cadena de caracteres(20).
- RDR2.4: Consulta de reserva
 - Identificador de reserva: cadena de caracteres(20).

3. Subsistema de gestión de mantenimiento

- RDE3.1: Datos de entrada de alta de la reparación
 - Identificador de habitación: número entero (3).
 - Descripción: Cadena de caracteres (300).
 - Fecha: Tipo date.
- RDW3.1: Datos almacenados de reparación.
 - Identificador de habitación: número entero (3).
 - Descripción: Cadena de caracteres (300).
 - Fecha: Tipo date.
 - Reparación resuelta: boolean.
- RDE3.2: Datos de entrada para resolver una reparación.
 - Identificador de reparación: cadena de caracteres (30)
- RDW3.2: Datos almacenados de reparación.
 - Identificador de reparación: cadena de caracteres (30)

- Reparación resuelta: boolean
- RDE3.3: Datos de entrada de planificación de limpieza.
 - Identificador de habitación: número entero (3).
 - DNI empleado: 8 enteros y 1 carácter.
 - Fecha: date.
- RDW3.3: Datos almacenados de planificación de limpieza.
 - Identificador de habitación: número entero (3).
 - Identificador de empleado: 8 enteros y 1 carácter.
 - Fecha y hora: date.
- RDE3.4: Datos de entrada de registro de provisiones.
 - Identificador de producto: Cadena de caracteres (50).
 - Cantidad: Número entero positivo.
- RDW3.4: Datos almacenados de registro de provisiones.
 - Identificador de producto: Cadena de caracteres (50).
 - Cantidad: Número entero positivo.
- RDR3.1: Consulta de habitación.
 - Identificador de habitación: número entero(3).
- RDR3.2: Consulta de reparación.
 - Identificador de reparación: cadena de caracteres (30).
- RDR3.3: Consulta de habitación
 - Identificador de habitación: número entero(3).
- RDR3.4: Consulta empleado.
 - DNI: Cadena de caracteres(9)
- RDR3.5: Consulta producto
 - Identificador de producto: Cadena de caracteres(50).

4. Subsistema de gestión de eventos

- RDE4.1: Datos de entrada para crear un grupo
 - Identificadores de los clientes.
 - Descripción de la actividad: Cadena de caracteres (500)
 - Identificador del guía turístico: Cadena de caracteres (9)
 - Fecha y hora: tipo date.
- RDW4.1: Datos almacenados sobre el grupo.
 - Identificadores de los clientes.
 - Descripción de la actividad: Cadena de caracteres (500)
 - Identificador del guía turístico: Cadena de caracteres (9)
 - Fecha y hora: tipo date
 - Se crea una asociación entre los clientes, la actividad, el guía turístico y el grupo creado.
- RDE4.2: Identificadores y datos del evento.
 - Identificador de la sala: número entero (3)
 - Identificadores de los empleados.

- Descripción del evento: Cadena de caracteres (500)
 - Precio: número entero y dos decimales.
 - Fecha y hora: tipo date.
- RDW4.2: Datos almacenados sobre el evento.
 - Identificador de la sala: Cadena de caracteres (20)
 - Identificadores de los empleados.
 - Descripción del evento: Cadena de caracteres (500)
 - Precio: número entero y dos decimales.
 - Fecha y hora: tipo date.
 - Se crea una asociación entre la sala, los empleados y el evento creado.
- RDE4.3: Datos de entrada de la publicidad asociada a un evento.
 - Identificador del evento: Cadena de caracteres (20)
- RDR4.1: Lectura de los correos de cada ficha de todos los clientes del sistema.
 - Correo Electrónico: Cadena de caracteres (30)
- RDR4.2: Datos del evento.
 - Identificador de la sala: Cadena de caracteres (20)
 - Descripción del evento: Cadena de caracteres (500)
 - Precio: número entero y dos decimales.
 - Fecha y hora: tipo date.
- RDR4.3: Consulta de clientes.
 - Identificador de clientes: Cadena de caracteres(9)
- RDR4.4: Consulta de guía.
 - Identificador de guía: Cadena de caracteres(9)
- RDR4.5: Consulta de sala
 - Identificador de sala: enteros(3).
- RDR4.6: Consulta de empleados
 - Identificador de empleados: Cadena de caracteres(9)

Restricciones semánticas

1. Subsistema de gestión de empleados

- RS1.1: Un DNI corresponde a un único empleado.
 - RF: RF1.1
 - RD(s): RDW1.1
 - Descripción: "Si ya había un empleado con el mismo DNI, no se inserta el nuevo contacto y se devuelve un error."
- RS1.2: Un empleado debe ser mayor de 16 años.
 - RF: RF1.1
 - RD(s): RDW1.1
 - Descripción: "Si la edad de un empleado no supera los 16 años, no se insertan los datos y se devuelve un error."
- RS1.3: En el sistema hay un único gerente.
 - RF: RF1.1
 - RD(s): RDW1.1

- Descripción: “Si ya existe un empleado de tipo gerente en el sistema, no se insertan los datos y se devuelve un error.”
- RS1.4: Un DNI corresponde a un único empleado.
 - RF: RF1.2
 - RD(s): RDW1.2
 - Descripción: “Si ya había un empleado con el mismo DNI, no se inserta el nuevo contacto y se devuelve un error.”
- RS1.5: Un empleado debe ser mayor de 16 años.
 - RF: RF1.2
 - RD(s): RDW1.2
 - Descripción: “Si la edad de un empleado no supera los 16 años, no se insertan los datos y se devuelve un error.”
- RS1.6: En el sistema hay un único gerente.
 - RF: RF1.2
 - RD(s): RDW1.2
 - Descripción: “Si ya existe un empleado de tipo gerente en el sistema, no se insertan los datos y se devuelve un error.”

2. Subsistema de gestión de clientes y reservas

- RS2.1: El DNI corresponde a un único cliente.
 - RF: RF2.1
 - RD(s): RDW2.1
 - Descripción: “Si ya había un cliente con el mismo DNI, no se inserta el nuevo contacto y se devuelve un error.”
- RS2.2: La fecha de entrada debe ser posterior a la fecha actual.
 - RF: RF2.1
 - RD(s): RDE2.1
 - Descripción: “Si se introduce una fecha de entrada que no sea posterior a la fecha actual, no se inserta y se devuelve un error.”
- RS2.3: La fecha de salida debe ser posterior a la fecha de entrada.
 - RF: RF2.1
 - RD(s): RDE2.1
 - Descripción: “Si se introduce una fecha de salida que no sea posterior a la fecha de entrada, no se inserta y se devuelve un error.”
- RS2.4: El check out debe estar asociado a un check in.
 - RF: RF2.3
 - RD(s): RDE2.3
 - Descripción: “Si se introduce un check out sin un check in asociado, no se inserta y se devuelve un error, es decir, no se podrá marcar un check-out como realizado si no se ha hecho el check-in.”
- RS2.5: Una reserva para la que ya se ha realizado un check-in no se puede cancelar.
 - RF: RF2.4
 - RD(s): RDE2.4
 - Descripción: Si se intenta cancelar una reserva con un check-in previamente realizado, el sistema no la cancelará y devolverá un error

3. Subsistema de gestión de mantenimiento

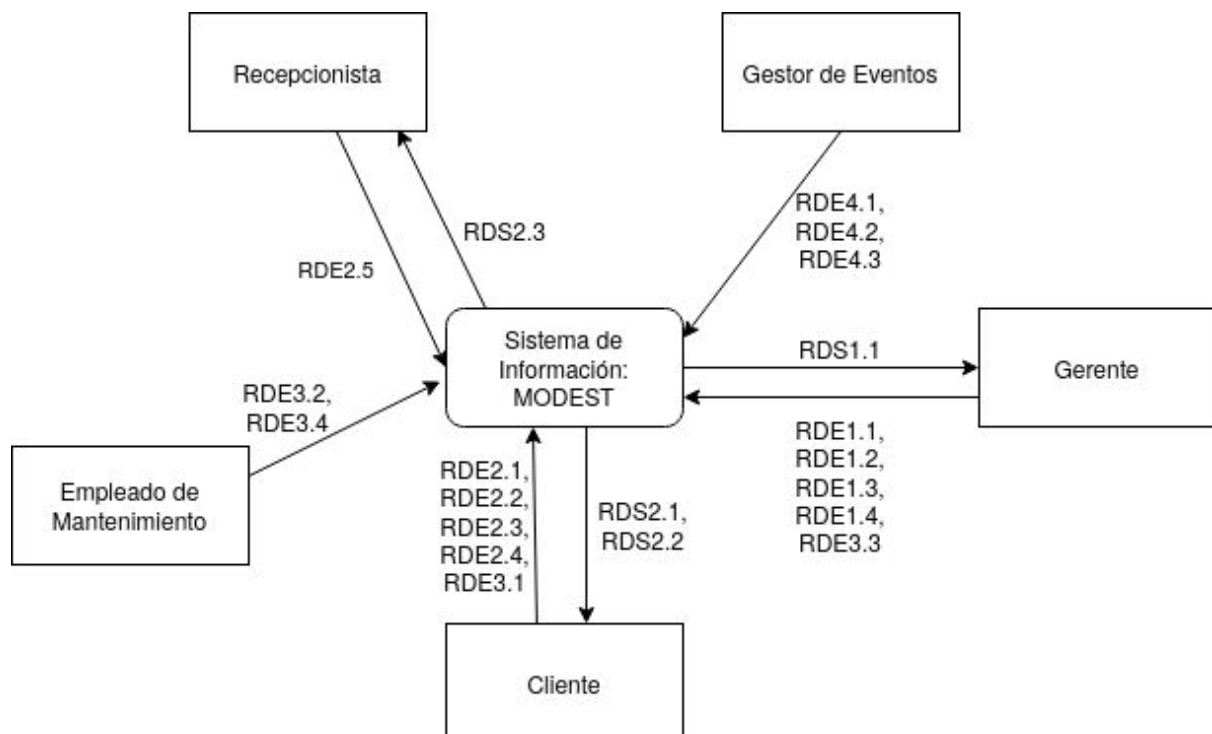
- RS3.1: Un empleado no puede limpiar más de 10 habitaciones al día
 - RF: RF3.3
 - RD(s): RDE3.3
 - Descripción: Si se intenta adjudicar una habitación nueva a un empleado que ya tenga asignadas 10 habitaciones en un día concreto se mostrará un error.
- RS3.2: La fecha debe ser posterior o igual a la fecha actual.
 - RF: RF3.3
 - RD(s): RDE3.3
 - Descripción: Si se introduce una fecha que no sea posterior o igual a la actual no se introduce y se muestra un error.

4. Subsistema de gestión de eventos

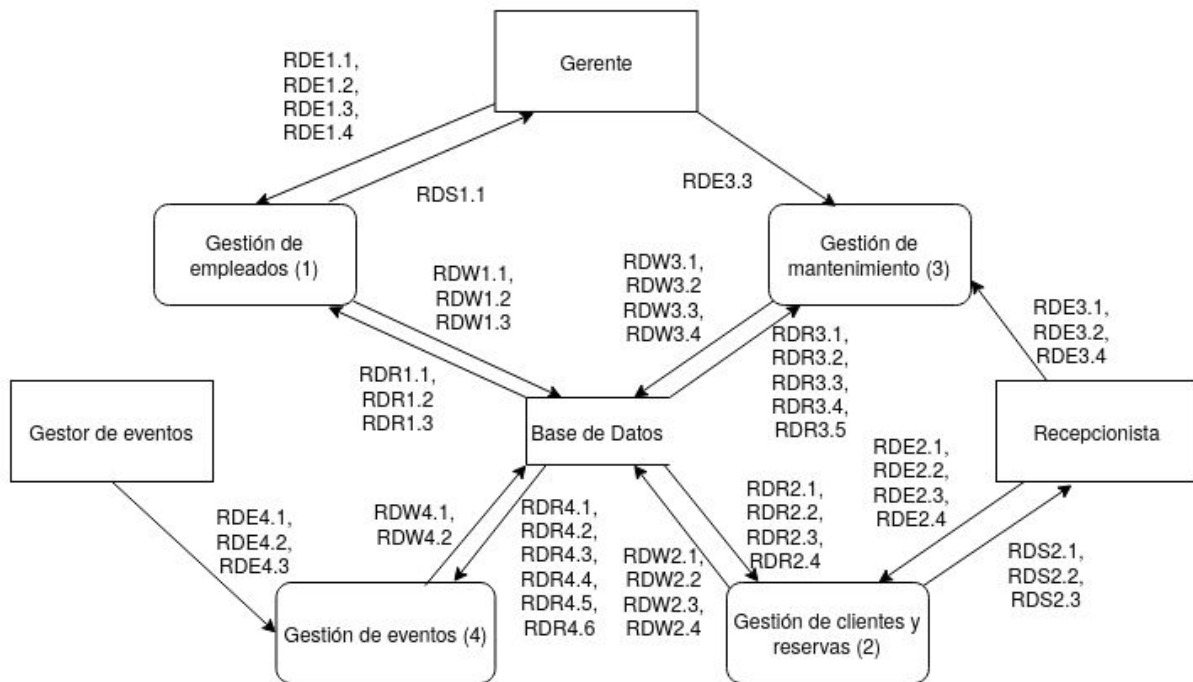
- RS4.1: La sala debe estar disponible.
 - RF: RF4.2
 - RD(s): RDE4.2
 - Descripción: “La sala identificada por el identificador proporcionado debe estar disponible en la fecha y hora introducidas.”

Diagramas

DFD Esquema de Caja Negra

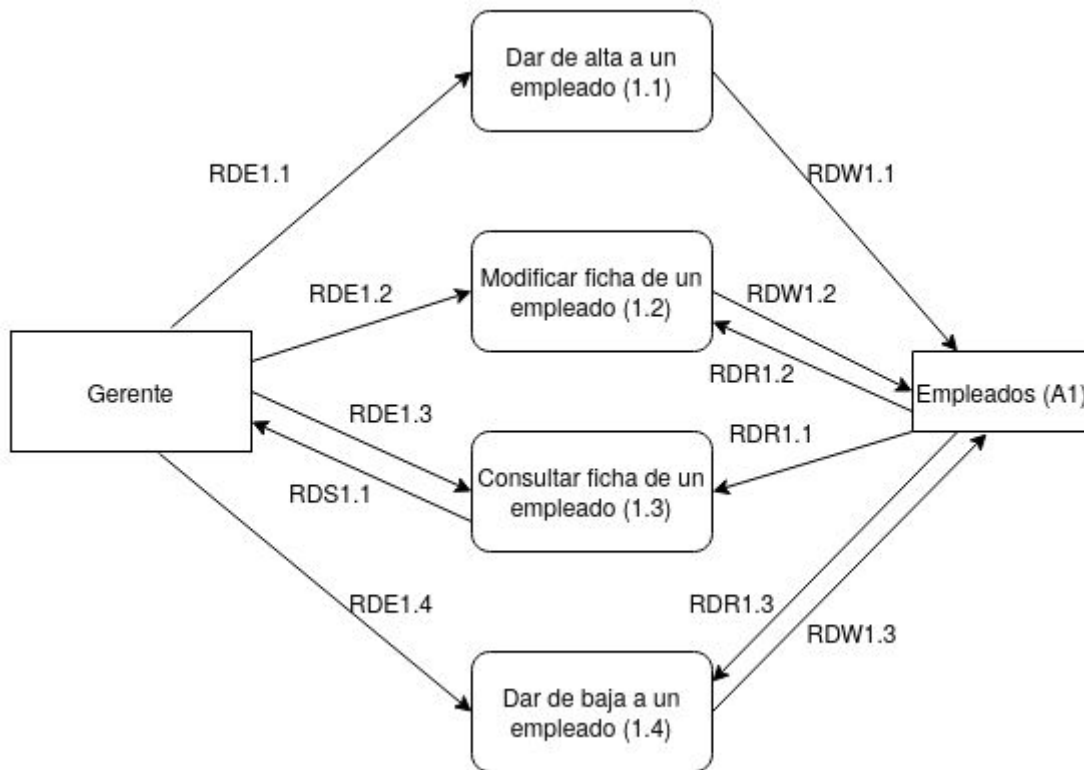


DFD Armazón (DFD0)

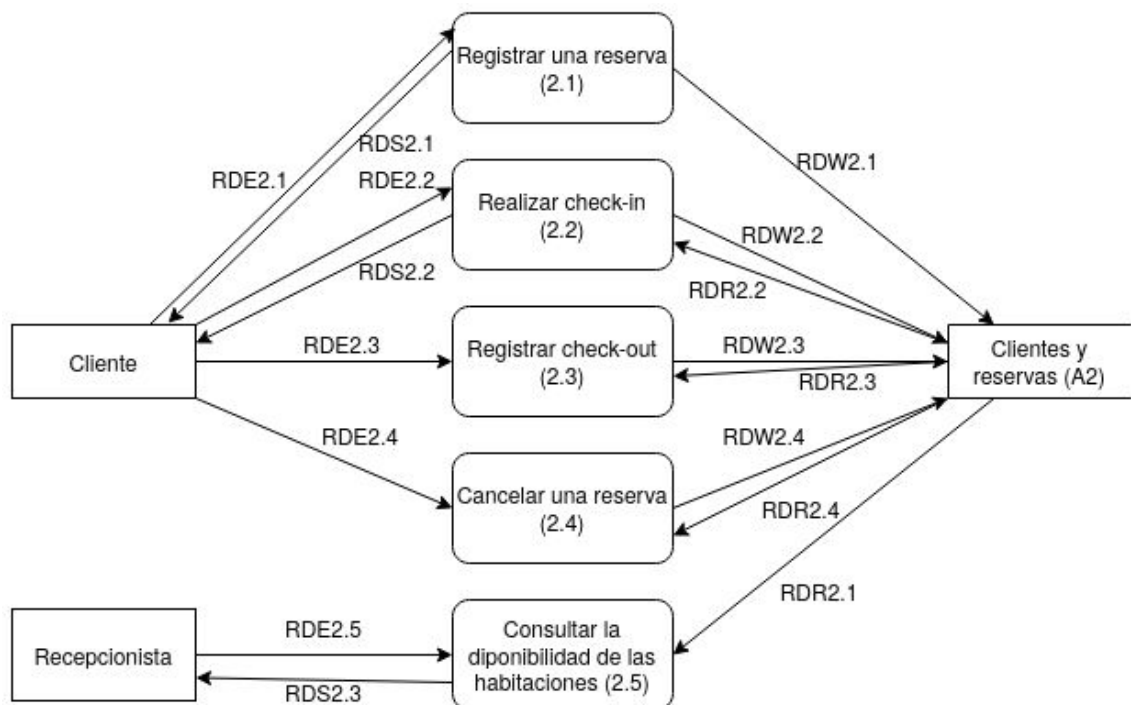


DFD1

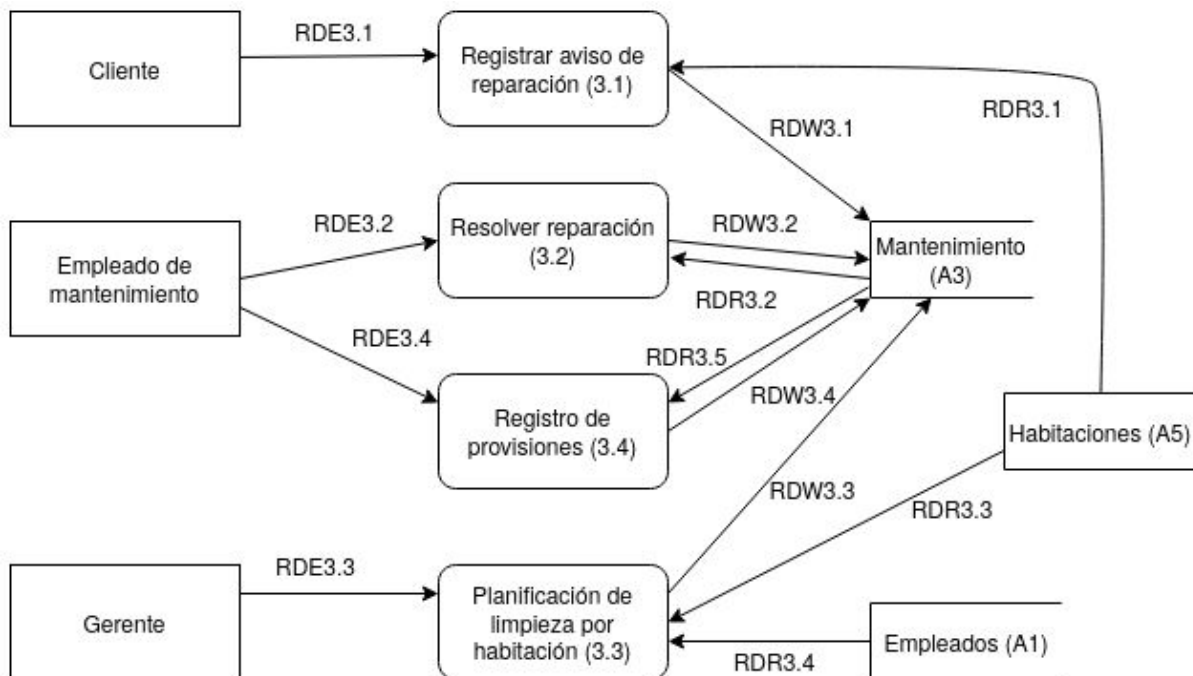
1. DFD1 Subsistema de gestión de empleados



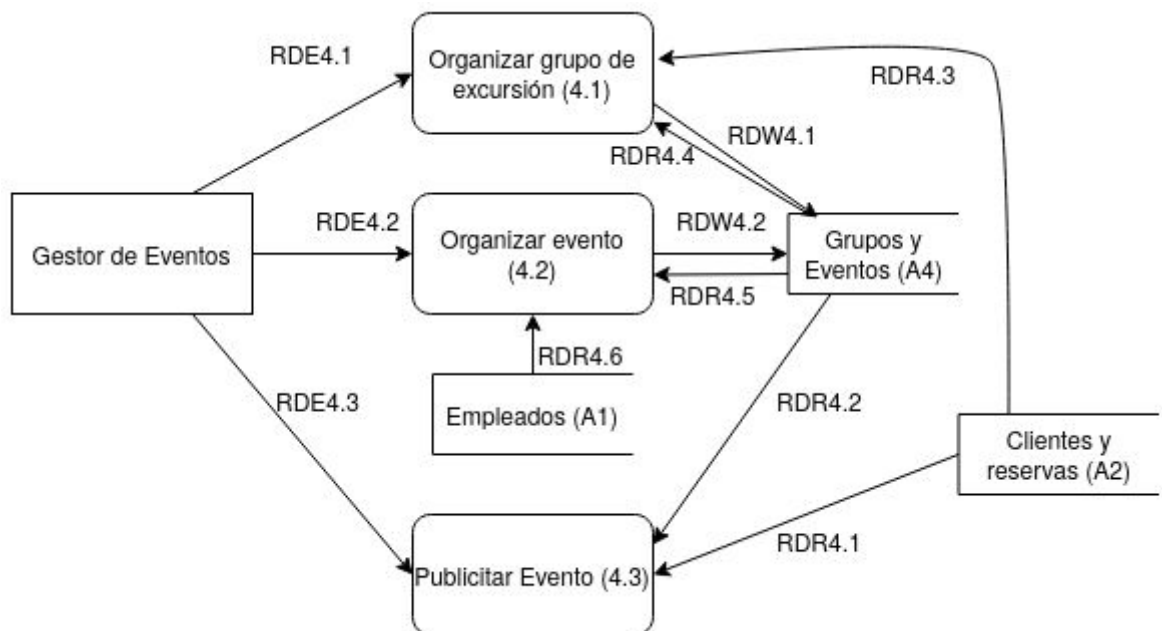
2. DFD1 Subsistema de clientes y reservas



3. DFD1 Subsistema de gestión de mantenimiento



4. DFD1 Subsistema de gestión de eventos



Práctica 2: Diseño de datos

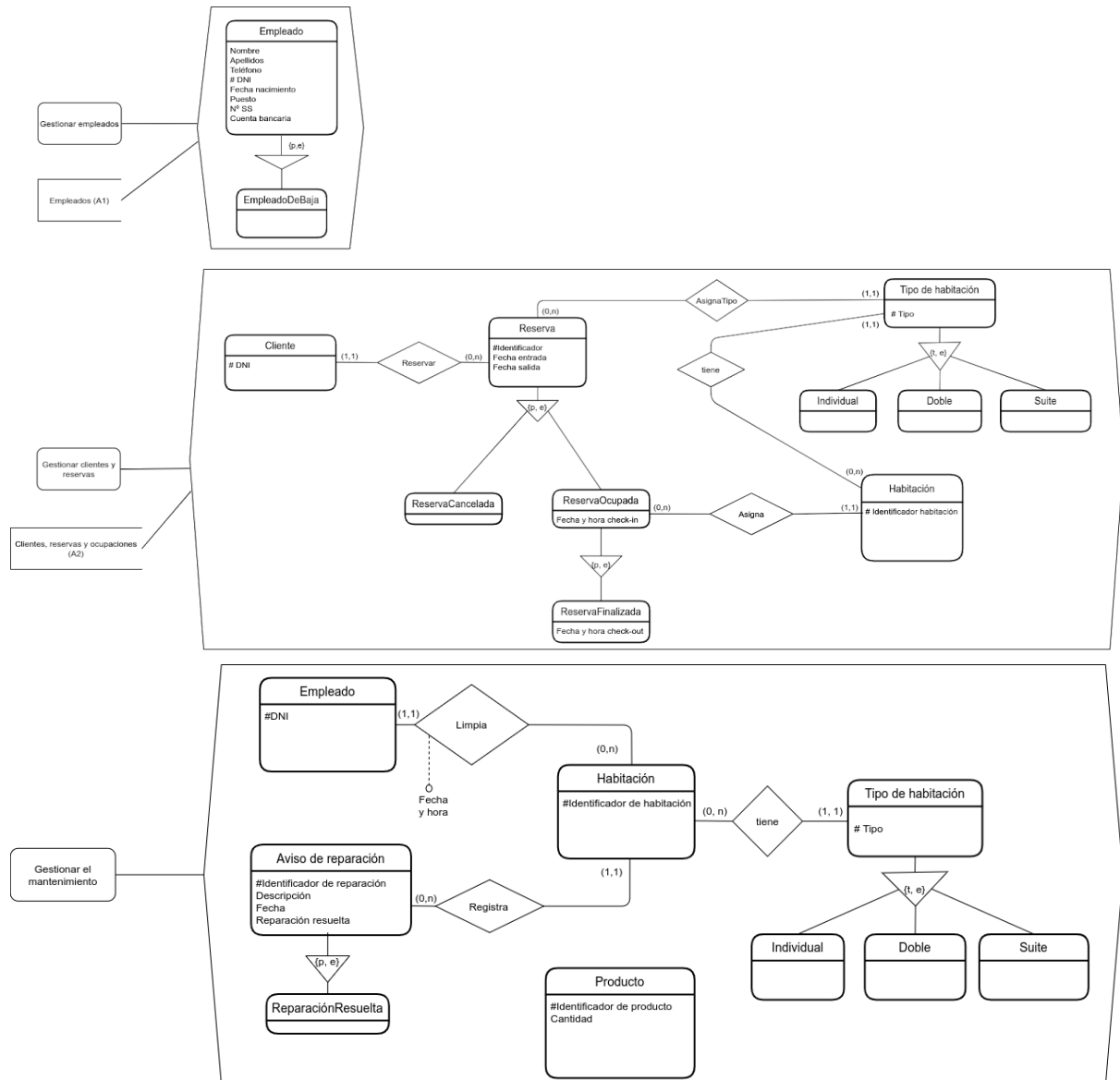
Revisión de la Práctica 1

A continuación se exponen las modificaciones de la práctica 1.

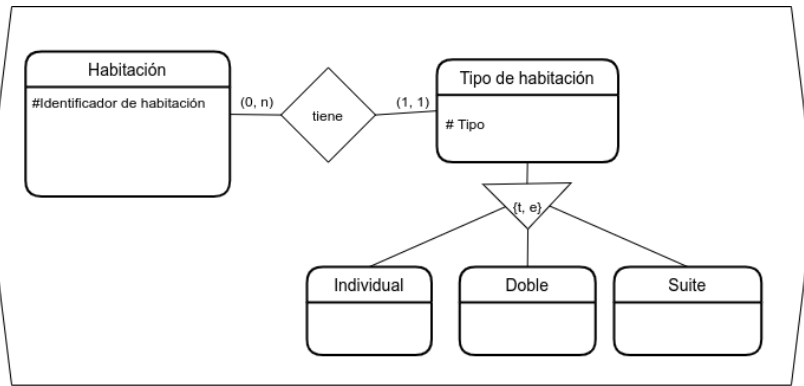
- Adecuar los agentes externos a las funcionalidades, tanto en los requisitos funcionales como en los esquemas.
 - En el subsistema 2 hemos cambiado el agente externo Recepcionista por Cliente, salvo en el último requisito funcional.
 - En el subsistema 3 hemos cambiado el agente externo Recepcionista por Cliente o Empleado de mantenimiento, según corresponda.
- Se ha añadido una restricción semántica (RS 2.5): una reserva para la que ya se ha realizado un check-in no se puede cancelar.
- Se ha modificado el DFD1 Subsistema de gestión de mantenimiento sustituyendo los almacenes "Reparaciones", "Provisiones" y "Limpieza y empleados" por el almacén "Mantenimiento (A3)".
- Se ha incluido un nuevo almacén en el sistema, "Habitaciones (A5)".
- Se han modificado algunos Requisitos Funcionales, a los que se les ha añadido requisitos de lectura para cuando acceden a la base de datos a consultar algunos datos. También se han modificado los Diagramas con esta información.

Esquemas externos de todos los procesos y almacenes

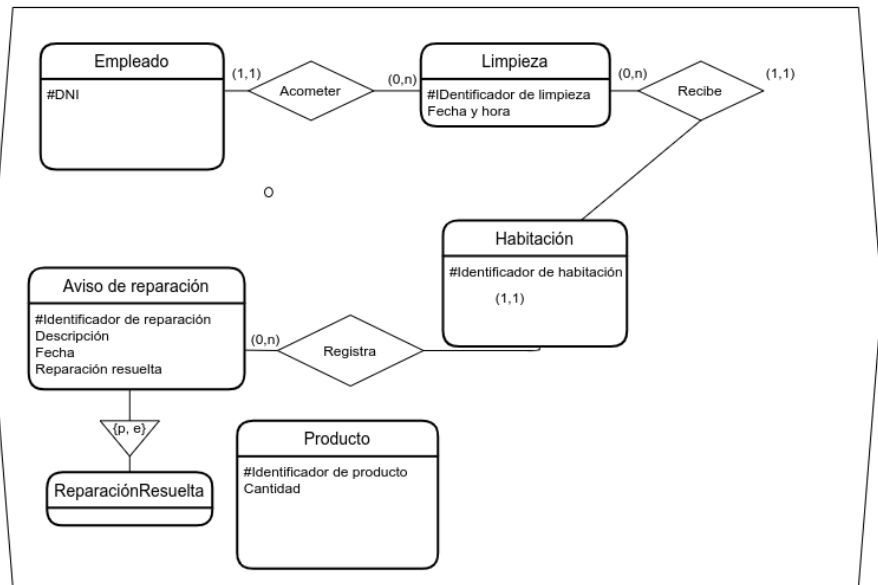
DFD0



Habitaciones y tipos (A5)

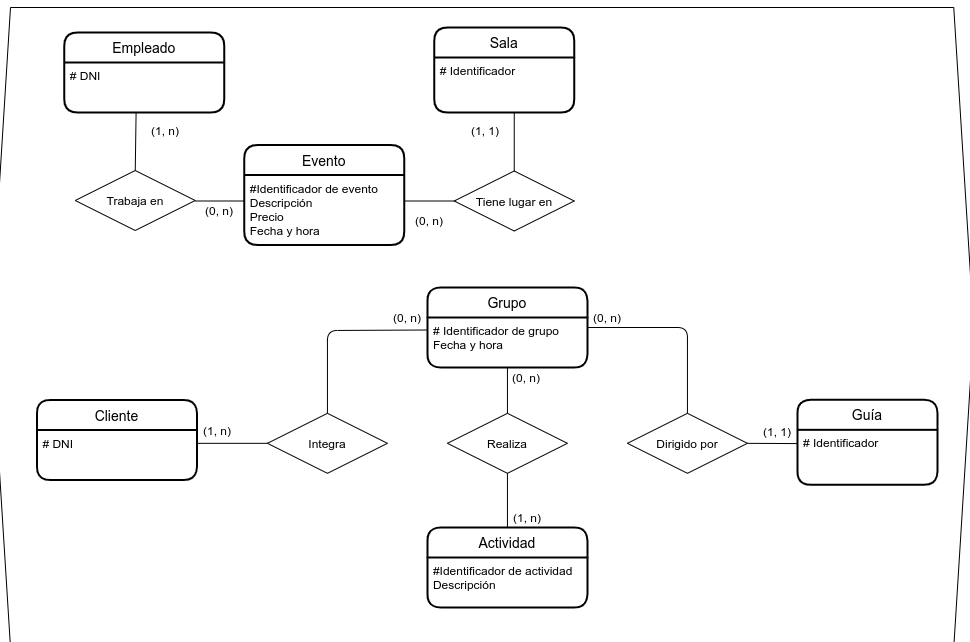


Mantenimiento (A3)

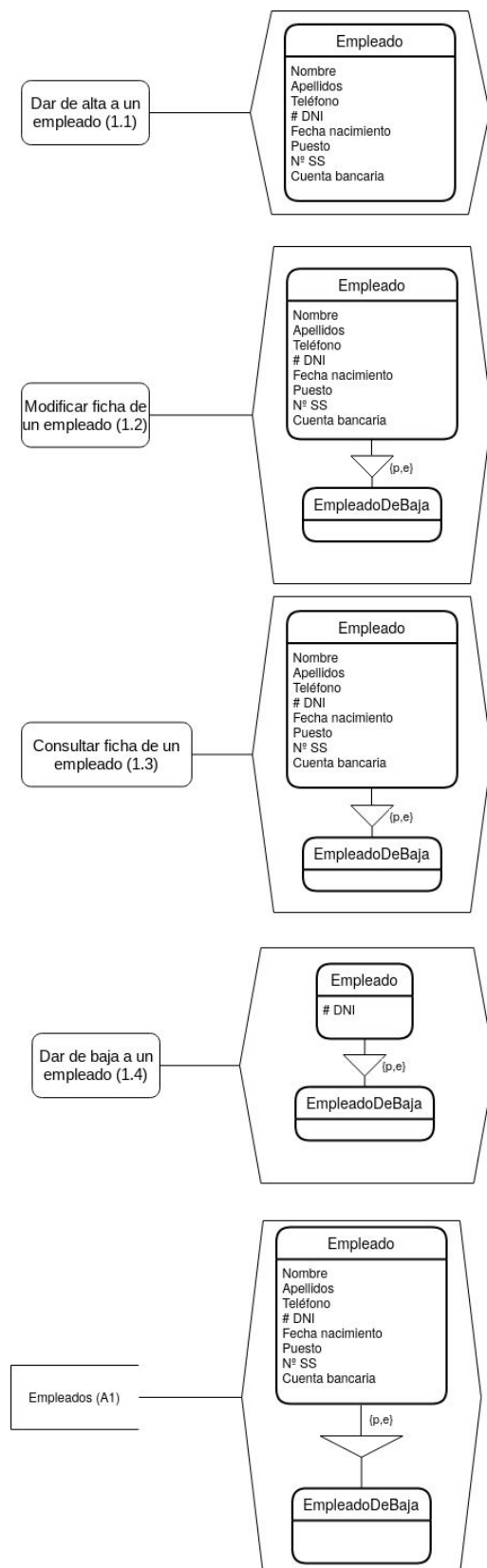


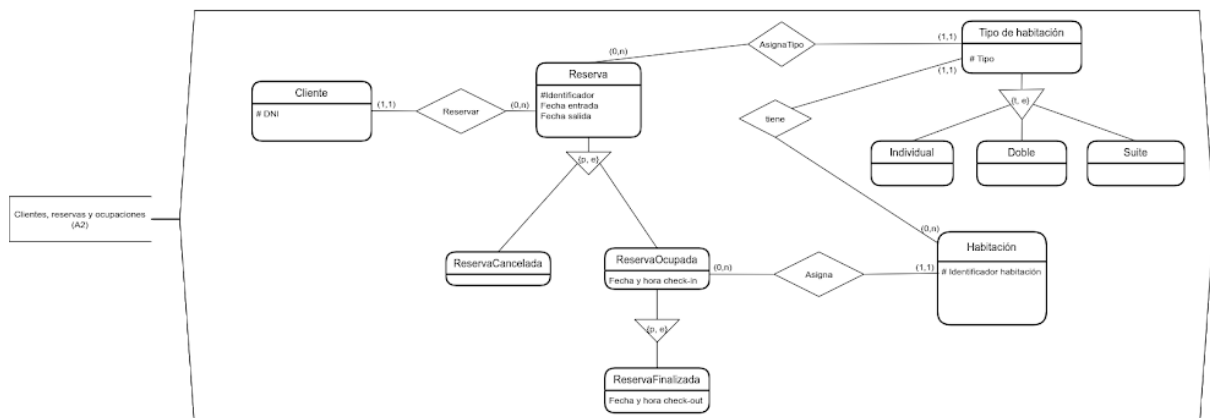
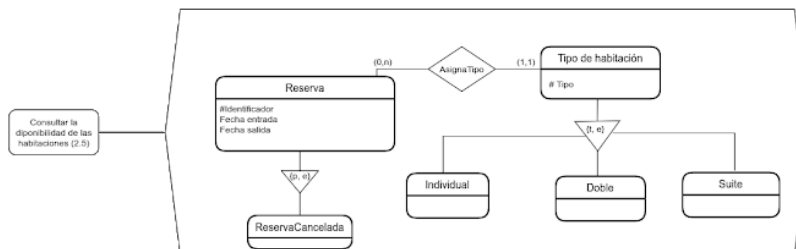
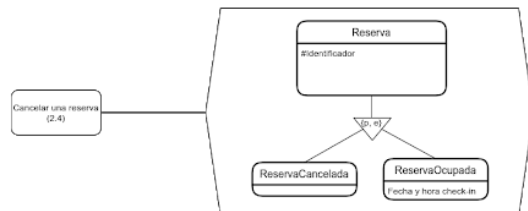
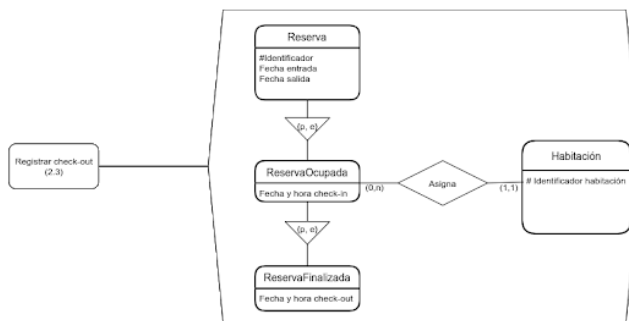
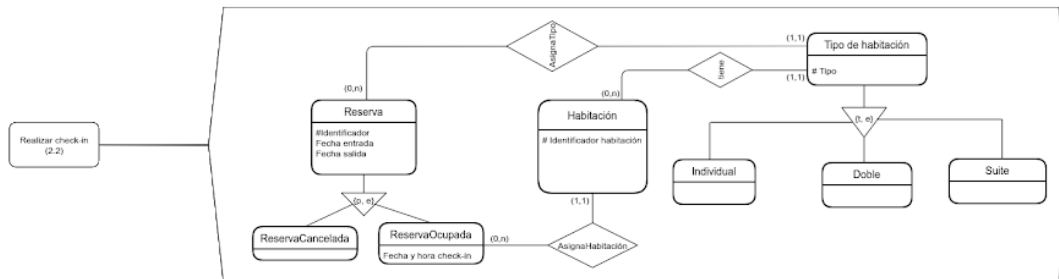
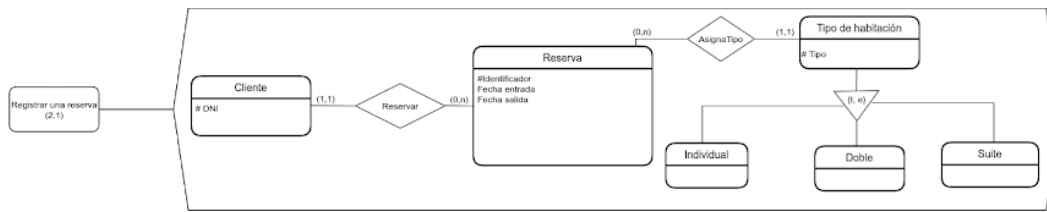
Gestionar eventos

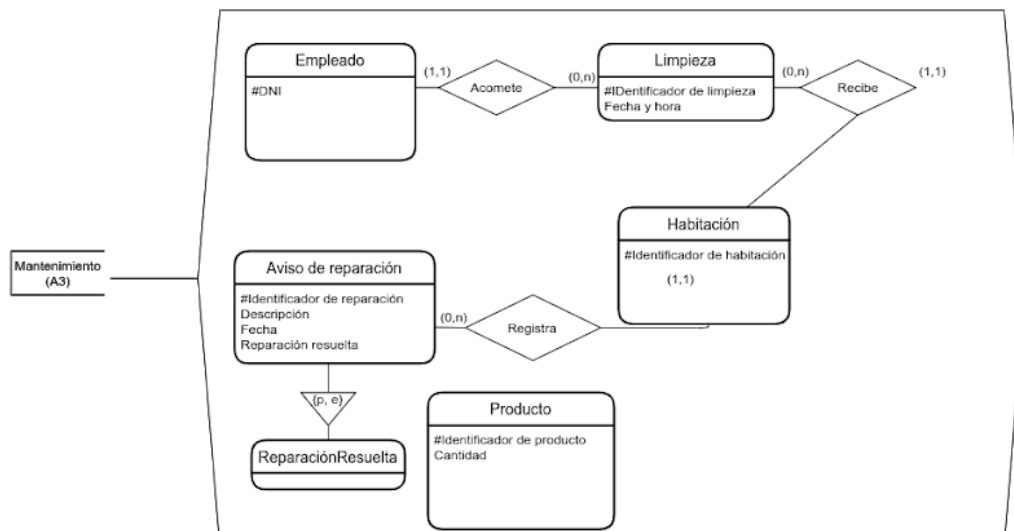
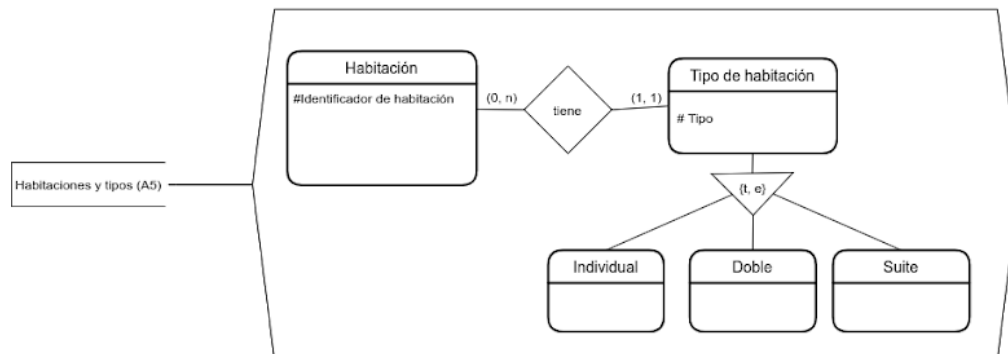
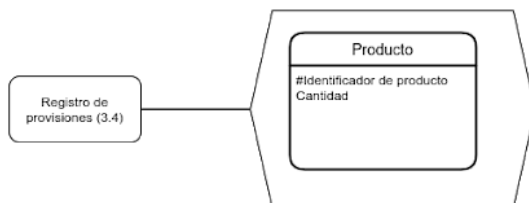
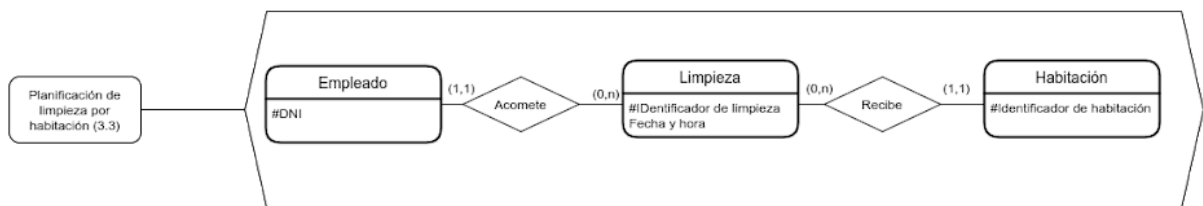
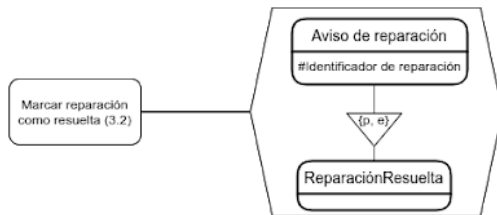
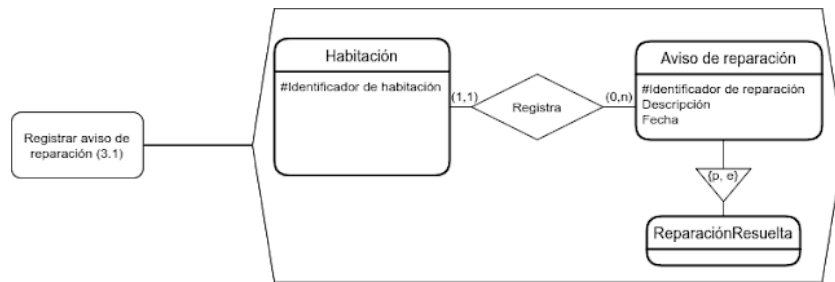
Grupos y Eventos (A4)



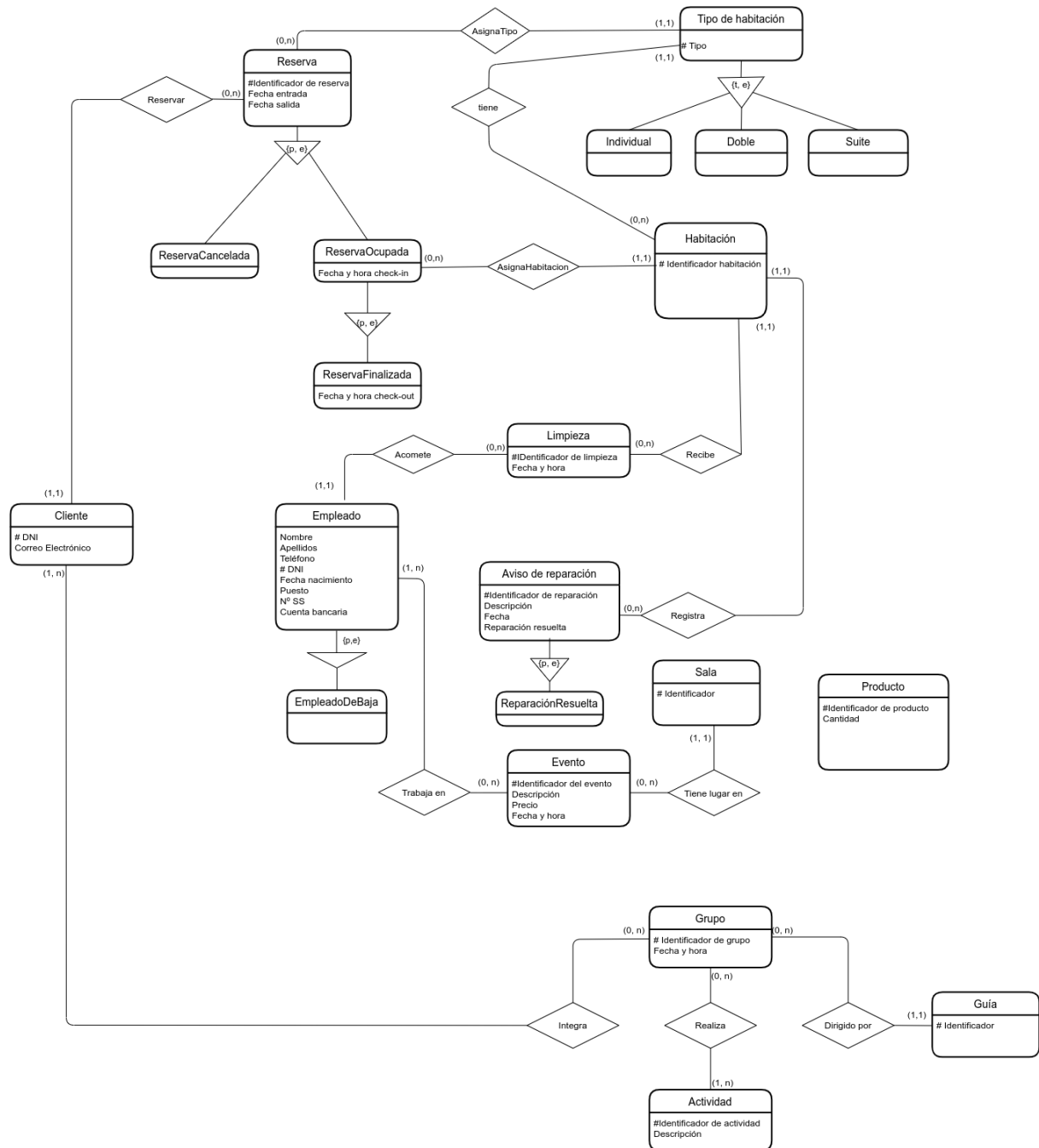
DFD1

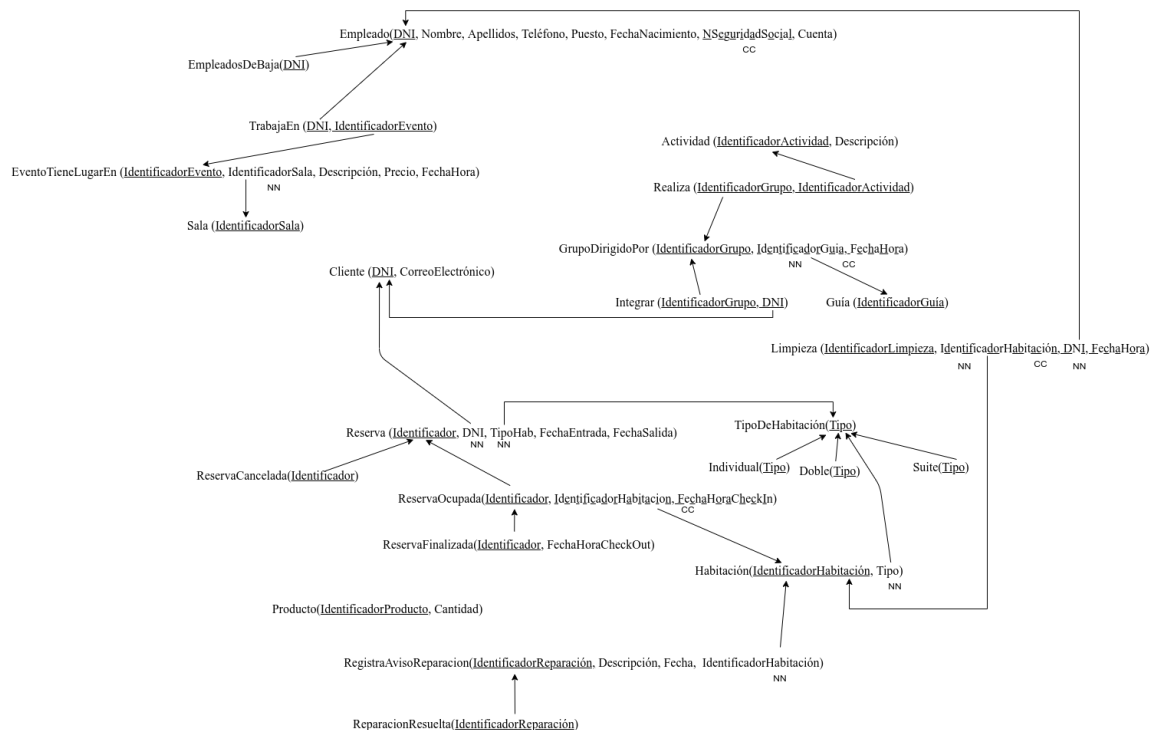






Esquema E/R completo del sistema





- Empleado (DNI, Nombre, Apellidos, Teléfono, Puesto, FechaNacimiento, NSeguridadSocial, Cuenta):
 - DNI -> Nombre, Apellidos, Teléfono, Puesto, FechaNacimiento, NSeguridadSocial, Cuenta.
 - NSeguridadSocial -> DNI, Nombre, Apellidos, Teléfono, Puesto, FechaNacimiento, Cuenta.Está en forma normal de Boyce-Codd.
- EmpleadosDeBaja (DNI):
Está en forma normal de Boyce-Codd.
- TrabajaEn (DNI, IdentificadorEvento):
Está en forma normal de Boyce-Codd.
- EventoTieneLugarEn (IdentificadorEvento, IdentificadorSala, Descripción, Precio, FechaHora):

- IdentificadorEvento->IdentificadorSala, Descripcion, Precio, FechaHora

Está en forma normal de Boyce-Codd.
- Sala (IdentificadorSala):
Está en forma normal de Boyce-Codd.
- Actividad (IdentificadorActividad, Descripción):
 - IdentificadorActividad->Descripción

Está en forma normal de Boyce-Codd.
- Realiza (IdentificadorGrupo, IdentificadorActividad):
Está en forma normal de Boyce-Codd.
- GrupoDirigidoPor (IdentificadorGrupo, IdentificadorGuia, FechaHora):
 - IdentificadorGrupo->IdentificadorGuia, FechaHora
 - IdentificadorGuia, FechaHora->IdentificadorGrupo.

Está en forma normal de Boyce-Codd.
- Integrar (IdentificadorGrupo, DNI):
Está en forma normal de Boyce-Codd.
- Guía (IdentificadorGuía):
Está en forma normal de Boyce-Codd.
- Reserva (Identificador, DNI, TipoHab, FechaEntrada, FechaSalida):
 - Identificador -> DNI, TipoHab, FechaEntrada, FechaSalida

Está en forma normal de Boyce-Codd.
- ReservaOcupada (Identificador, IdentificadorHabitacion, FechaHoraCheckIn):
 - Identificador->IdentificadorHabitacion, FechaHoraCheckIn
 - IdentificadorHabitacion, FechaHoraCheckIn->Identificador

Está en forma normal de Boyce-Codd.
- ReservaFinalizada (Identificador, FechaHoraCheckOut):
 - Identificador->FechaHoraCheckOut

Está en forma normal de Boyce-Codd.
- ReservaCancelada (Identificador):
Está en forma normal de Boyce-Codd.
- TipoDeHabitación (Tipo):
Está en forma normal de Boyce-Codd.
- Individual (Tipo):
Está en forma normal de Boyce-Codd.

- Doble (Tipo):
Está en forma normal de Boyce-Codd.
- Suite (Tipo):
Está en forma normal de Boyce-Codd.
- Cliente (DNI, CorreoElectrónico):
 - DNI->CorreoElectrónico
 Está en forma normal de Boyce-Codd.
- Habitación (IdentificadorHabitación, Tipo):
 - IdentificadorHabitación->Tipo
 Está en forma normal de Boyce-Codd.
- RegistraAvisoReparacion (IdentificadorReparacion, Descripcion, Fecha, IdentificadorHabitacion):
 - IdentificadorReparacion->Descripcion, Fecha, IdentificadorHabitacion
 Está en forma normal de Boyce-Codd.
- ReparacionResuelta(IdentificadorReparacion):
Está en forma normal de Boyce-Codd.
- Limpieza (IdentificadorLimpieza, IdentificadorHabitación, DNI, FechaHora):
 - IdentificadorLimpieza->IdentificadorHabitación, DNI, FechaHora
 - IdentificadorHabitación, DNI, FechaHora->IdentificadorLimpieza
 Está en forma normal de Boyce-Codd.
- Producto (IdentificadorProducto, cantidad):
 - IdentificadorProducto->cantidad
 Está en forma normal de Boyce-Codd.

Sentencias SQL de creación de las tablas obtenidas y de inserción de tuplas

```
CREATE TABLE Empleado(
  DNI VARCHAR2(9) PRIMARY KEY,
  Nombre VARCHAR2 NOT NULL,
  Apellidos VARCHAR2 NOT NULL,
  Telefono INT(9),
```

```

        Puesto VARCHAR2 CHECK(Puesto='Gerente' OR
Puesto='Recepcionista' OR Puesto='Mantenimiento' OR
Puesto='Limpieza' OR Puesto='Eventos') NOT NULL,
        FechaNacimiento DATE,
        NSeguridadSocial VARCHAR2(8) NOT NULL UNIQUE,
        Cuenta INT(20)
    );

CREATE TABLE EmpleadoDeBaja(
    DNI VARCHAR2(9) PRIMARY KEY REFERENCES Empleado(DNI)
);

CREATE TABLE Guia(
    IdentificadorGuia VARCHAR2(9) PRIMARY KEY
);

CREATE TABLE Actividad(
    IdentificadorActividad VARCHAR2(9) PRIMARY KEY,
    Descripcion VARCHAR2(100)
);

CREATE TABLE Cliente(
    DNI VARCHAR2(9) PRIMARY KEY,
    CorreoElectronico VARCHAR2(30)
);

CREATE TABLE Sala(
    IdentificadorSala VARCHAR2(9) PRIMARY KEY
);

CREATE TABLE EventoTieneLugarEn(
    IdentificadorEvento VARCHAR2(9) PRIMARY KEY,
    IdentificadorSala VARCHAR2(9) REFERENCES
Sala(IdentificadorSala) NOT NULL,
    Descripcion VARCHAR2(100),
    Precio DOUBLE(3, 2) NOT NULL CHECK(Precio > 0),
    FechaHora DATE NOT NULL
);

CREATE TABLE TrabajaEn(
    DNI VARCHAR2(9) REFERENCES Empleado(DNI),

```

```

        IdentificadorEvento VARCHAR2(9) REFERENCES
EventoTieneLugarEn(IdentificadorEvento),
        PRIMARY KEY(DNI, IdentificadorEvento)
);
CREATE TABLE GrupoDirigidoPor(
        IdentificadorGrupo VARCHAR2(9) PRIMARY KEY,
        IdentificadorGuia VARCHAR2(9) REFERENCES
Guia(IdentificadorGuia) NOT NULL,
        FechaHora DATE NOT NULL,
        UNIQUE(IdentificadorGuia, FechaHora)
);

CREATE TABLE Realiza(
        IdentificadorGrupo VARCHAR2(9) REFERENCES
GrupoDirigidoPor(IdentificadorGrupo),
        IdentificadorActividad VARCHAR2(9) REFERENCES
Actividad(IdentificadorActividad),
        PRIMARY KEY (IdentificadorGrupo, IdentificadorActividad)
);

CREATE TABLE Integrar(
        IdentificadorGrupo VARCHAR2(9) REFERENCES
GrupoDirigidoPor(IdentificadorGrupo),
        DNI VARCHAR2(9) REFERENCES Cliente(DNI),
        PRIMARY KEY (IdentificadorGrupo, DNI)
);

CREATE TABLE Reserva(
        Identificador VARCHAR2(9) PRIMARY KEY,
        DNI VARCHAR2(9) REFERENCES Cliente(DNI) NOT NULL,
        TipoHab VARCHAR(1) REFERENCES TipoDeHabitacion(Tipo) NOT NULL,
        FechaEntrada DATE NOT NULL,
        FechaSalida DATE NOT NULL
);

CREATE TABLE ReservaCancelada(
        Identificador VARCHAR2(9) REFERENCES Reserva(Identificador)
PRIMARY KEY
);

CREATE TABLE ReservaFinalizada(

```

```

        Identificador VARCHAR2(9) REFERENCES
ReservaOcupada(Identificador) PRIMARY KEY,
        FechaHoraCheckOut DATE NOT NULL
    );

CREATE TABLE Habitacion(
        IdentificadorHabitacion INT(3) PRIMARY KEY,
        Tipo VARCHAR(1) REFERENCES TipoDeHabitacion(Tipo) NOT NULL
    );

CREATE TABLE ReservaOcupada(
        Identificador VARCHAR2(9) REFERENCES Reserva(Identificador)
PRIMARY KEY,
        IdentificadorHabitacion INT(3) REFERENCES
Habitacion(IdentificadorHabitacion) NOT NULL,
        FechaHoraCheckIn DATE NOT NULL,
        UNIQUE (IdentificadorHabitacion, FechaHoraCheckIn)
    );

CREATE TABLE Limpieza(
        IdentificadorLimpieza VARCHAR2(9) PRIMARY KEY,
        IdentificadorHabitacion INT REFERENCES
Habitacion(IdentificadorHabitacion) NOT NULL,
        DNI VARCHAR2(9) REFERENCES Empleado(DNI) NOT NULL,
        FechaHora DATE NOT NULL,
        UNIQUE (IdentificadorHabitacion,DNI, FechaHora)
    );

CREATE TABLE RegistraAvisoReparacion(
        IdentificadorReparacion VARCHAR2(9) PRIMARY KEY,
        IdentificadorHabitacion INT(3) REFERENCES
Habitacion(IdentificadorHabitacion) NOT NULL
        Descripcion VARCHAR2(100),
        Fecha DATE
    );

CREATE TABLE ReparacionResuelta(
        IdentificadorReparacion VARCHAR2(9) REFERENCES
RegistraAvisoReparacion(IdentificadorReparacion) PRIMARY KEY
    );

```

```
CREATE TABLE TipoDeHabitacion(  
    Tipo CHAR PRIMARY KEY CHECK (Tipo='I' OR Tipo='D' OR Tipo='S')  
);
```

```
CREATE TABLE Individual(  
    Tipo CHAR REFERENCES TipoDeHabitacion(Tipo) PRIMARY KEY  
);
```

```
CREATE TABLE Doble(  
    Tipo CHAR REFERENCES TipoDeHabitacion(Tipo) PRIMARY KEY  
);
```

```
CREATE TABLE Suite(  
    Tipo CHAR REFERENCES TipoDeHabitacion(Tipo) PRIMARY KEY  
);
```

```
CREATE TABLE Producto(  
    IdentificadorProducto VARCHAR2(9) PRIMARY KEY,  
    Cantidad INT CHECK(Cantidad >= 0) NOT NULL  
);
```

```
INSERT INTO Empleado (DNI, Nombre, Apellidos, Telefono, Puesto,  
FechaNacimiento, NSeguridadSocial, Cuenta) VALUES ('11111111A',  
'Pepe', 'Espagnol', '611111111', 'Gerente', TO_DATE('1970-10-10',  
'YYYY-MM-DD'), '11111111', 'ES662100041840123456789');
```

```
INSERT INTO Empleado (DNI, Nombre, Apellidos, Telefono, Puesto,  
FechaNacimiento, NSeguridadSocial, Cuenta) VALUES ('22222222B',  
'Carmen', 'San Diego', '622222222', 'Recepcionista',  
TO_DATE('1985-12-11', 'YYYY-MM-DD'), '22222222',  
'ES6621000418401233456789');
```

```
INSERT INTO Empleado (DNI, Nombre, Apellidos, Telefono, Puesto,  
FechaNacimiento, NSeguridadSocial, Cuenta) VALUES ('33333333C',  
'Manuel', 'Carrero', '633333333', 'Limpieza', TO_DATE('1965-01-10',  
'YYYY-MM-DD'), '33333333', 'ES6621507418401233456365');
```

```
INSERT INTO EmpleadoDeBaja (DNI) VALUES ('22222222B');
```

```
INSERT INTO Sala (IdentificadorSala) VALUES ('S00000001');
```

```
INSERT INTO Sala (IdentificadorSala) VALUES ('S00000002');
```

```
INSERT INTO EventoTieneLugarEn (IdentificadorEvento,  
IdentificadorSala, Descripción, Precio, FechaHora) VALUES  
( 'E00000001', 'S00000001', 'Boda', '200', TO_DATE('2020-10-02  
10:00', 'YYYY-MM-DD HH:MI'));
```

```
INSERT INTO Actividad (IdentificadorActividad, Descripcion) VALUES  
( 'A00000001', 'Playa');
```

```
INSERT INTO Guía (IdentificadorGuia) VALUES ('12345678B');
```

```
INSERT INTO GrupoDirigidoPor (IdentificadorGrupo, IdentificadorGuia,  
FechaHora) VALUES ('G00000001', '12345678B', TO_DATE('2020-12-23',  
'YYYY-MM-DD'));
```

```
INSERT INTO Realiza (IdentificadorGrupo, IdentificadorActividad)  
VALUES ('G00000001', 'A00000001');
```

```
INSERT INTO Cliente (DNI, CorreoElectronico) VALUES ('12345678S',  
'pepitojd97@go.ugr.es');
```

```
INSERT INTO Cliente (DNI, CorreoElectronico) VALUES ('72435678S',  
'rosalinda@go.ugr.es');
```

```
INSERT INTO Integrar (IdentificadorGrupo, DNI) VALUES ('G00000001',  
'12345678S');
```

```
INSERT INTO TipoDeHabitacion (Tipo) VALUES ('I');
```

```
INSERT INTO TipoDeHabitacion (Tipo) VALUES ('D');
```

```
INSERT INTO TipoDeHabitacion (Tipo) VALUES ('S');
```

```
INSERT INTO Individual (Tipo) VALUES ('I');
```

```
INSERT INTO Doble (Tipo) VALUES ('D');
```

```
INSERT INTO Suite (Tipo) VALUES ('S');
```

```
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES  
( '100', 'I');
```

```

INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('101','I');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('102','I');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('103','I');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('104','I');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('105','D');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('106','D');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('107','D');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('108','D');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('200','S');
INSERT INTO Habitacion (IdentificadorHabitacion, Tipo) VALUES
('201','S');

```

```

INSERT INTO Reserva (Identificador, DNI, TipoHab, FechaEntrada,
FechaSalida) VALUES ('R00000001', '12345678S', 'I',
TO_DATE('2021-12-11', 'YYYY-MM-DD'), TO_DATE('2021-12-14',
'YYYY-MM-DD'));

```

```

INSERT INTO Reserva (Identificador, DNI, TipoHab, FechaEntrada,
FechaSalida) VALUES ('R00000002', '72435678S', 'D',
TO_DATE('2021-12-13', 'YYYY-MM-DD'), TO_DATE('2021-12-17',
'YYYY-MM-DD'));

```

```

INSERT INTO Reserva (Identificador, DNI, TipoHab, FechaEntrada,
FechaSalida) VALUES ('R00000003', '72435678S', 'S',
TO_DATE('2021-12-18', 'YYYY-MM-DD'), TO_DATE('2021-12-19',
'YYYY-MM-DD'));

```

```

INSERT INTO ReservaOcupada (Identificador, IdentificadorHabitacion,
FechaHoraCheckIn) VALUES ('R00000001', 101, TO_DATE('2021-12-7
12:30', 'YYYY-MM-DD HH:MI'));

```



```
INSERT INTO ReservaOcupada (Identificador, IdentificadorHabitacion,
FechaHoraCheckIn) VALUES ('R00000002', 102, TO_DATE('2021-12-7
12:30', 'YYYY-MM-DD HH:MI'));
```

```
INSERT INTO ReservaFinalizada (Identificador, FechaHoraCheckOut)
VALUES ('R00000001', TO_DATE('2021-12-12 12:30', 'YYYY-MM-DD HH:MI'
));
```

```
INSERT INTO ReservaCancelada (Identificador) VALUES ('R00000003');
```

```
INSERT INTO RegistraAvisoReparacion (IdentificadorReparacion,
Descripcion, Fecha, IdentificadorHabitacion) VALUES ('RE0000001',
'Arreglar grifo', TO_DATE('2020-12-12 11:30', 'YYYY-MM-DD HH:MI'),
101);
```

```
INSERT INTO ReparacionResuelta (IdentificadorReparacion)VALUES
('RE0000001');
```

```
INSERT          INTO          Limpieza          (IdentificadorLimpieza,
IdentificadorHabitacion, DNI, FechaHora) VALUES ('L0000001','100',
'33333333C', TO_DATE('2021-12-12 10:30', 'YYYY-MM-DD HH:MI'));
```

```
INSERT INTO Producto (IdentificadorProducto, cantidad) VALUES
('P00000001', 10);
```

```
INSERT INTO Producto (IdentificadorProducto, cantidad) VALUES
('P00000002', 5);
```

Práctica 3: Implementación

Revisión de la Práctica 2

A continuación se exponen las modificaciones de la práctica 2 que hemos realizado:

- Eliminar los nombres repetidos en las relaciones de los diagramas externos, concretamente los del esquema 2.2.
- Utilizar la especificación para representar la resolución de las reparaciones. Esquemas 3.1 y 3.2.
- Renombrar almacenes: A2->Clientes, reservas y ocupaciones; A5->Habitaciones y tipos.
- Representar la limpieza de habitaciones mediante una entidad.
- Incluir Reservas Canceladas en el diagrama 2.5.
- Propagar todos los cambios.

Descripción de las transacciones identificadas (secuencia de operaciones lógicas)

Las operaciones que modifican el estado de la base de datos se han implementado de la siguiente forma: Primero se recogen en la aplicación todos los datos necesarios para la operación en cuestión y después se realiza el envío de los mismos a la base de datos para que se produzcan los cambios pertinentes de manera definitiva. En este sentido, no hemos necesitado implementar una política de puntos de guardado para controlar las transacciones.

Las operaciones que producen cambios sobre la base de datos son precisamente una por cada uno de los requisitos funcionales que hemos analizado, excepto los requisitos RF1.3, RF2.5 y RF4.3, que solo interaccionan con la base de datos para obtener información.

Código de los disparadores implementados en el SGBD

1. SUBSISTEMA DE GESTIÓN DE EMPLEADOS:

1.1.Los empleados deben ser mayores de 16 años:

```
CREATE OR REPLACE TRIGGER edadEmpleado
BEFORE INSERT OR UPDATE ON Empleado
FOR EACH ROW
BEGIN
```

```

        IF (FLOOR(MONTHS_BETWEEN(SYSDATE, :new.FechaNacimiento)/12) <
16) THEN
            RAISE_APPLICATION_ERROR(-20111, 'Error: el empleado debe
ser mayor de 16 años');
        END IF;
    END;

```

1.2.En el sistema hay un único gerente.

```

CREATE OR REPLACE TRIGGER unicoGerente
    BEFORE INSERT OR UPDATE
    ON Empleado
    FOR EACH ROW
DECLARE
    num_gerentes INTEGER;
BEGIN
    IF (:new.Puesto='Gerente') THEN
        SELECT count(*) INTO num_gerentes FROM Empleado WHERE
Puesto='Gerente';
        IF(num_gerentes > 0) THEN
            RAISE_APPLICATION_ERROR(-20134, 'Error: solo puede haber un
gerente en el sistema');
        END IF;
    END IF;
END;

```

2. SUBSISTEMA DE GESTIÓN CLIENTES Y RESERVAS:

2.1 La fecha de entrada debe ser posterior a la fecha actual:

```

CREATE OR REPLACE TRIGGER fechaEntradaPosteriorR
    BEFORE INSERT OR UPDATE ON Reserva
    FOR EACH ROW
    BEGIN
        IF (FLOOR(MONTHS_BETWEEN(:new.FechaEntrada, SYSDATE)/12) < 0)
THEN
            RAISE_APPLICATION_ERROR(-20111, 'Error: la fecha de
entrada debe ser posterior a la fecha actual');
        END IF;
    END;

```

2.2 La fecha de salida debe ser posterior a la fecha de entrada:

```

CREATE OR REPLACE TRIGGER fechaSalidaPosterior
BEFORE INSERT OR UPDATE ON Reserva
FOR EACH ROW
    BEGIN
        IF
            (FLOOR(MONTHS_BETWEEN(:new.FechaSalida,
:new.FechaEntrada)/12) < 0) THEN
            RAISE_APPLICATION_ERROR(-20111, 'Error: la fecha de
salida debe ser posterior a la fecha de entrada');
        END IF;
    END;

```

2.3 Una reserva para la que ya se ha realizado un check-in no se puede cancelar:

```

CREATE OR REPLACE TRIGGER NoCancelarReservaOcupada
BEFORE INSERT
ON ReservaCancelada
FOR EACH ROW
DECLARE
    reservas INTEGER;
BEGIN
    SELECT COUNT(*)
    INTO reservas
    FROM ReservaOcupada
    WHERE ReservaOcupada.identificador = :new.identificador;

    IF (reservas > 0) THEN
        raise_application_error(-20700, 'No se puede cancelar una
reserva ocupada. ');
    END IF;

END;

```

2.4 No se puede ocupar una habitación ya ocupada:

```

CREATE OR REPLACE TRIGGER habitacionYaOcupada
BEFORE INSERT OR UPDATE
ON ReservaOcupada
FOR EACH ROW
DECLARE
    ocupaciones INTEGER;
BEGIN

```

```

SELECT count(*)
INTO ocupaciones
FROM (
    SELECT Identificador
    FROM ReservaOcupada
    WHERE IdentificadorHabitacion = :new.IdentificadorHabitacion
    MINUS
    SELECT Identificador
    FROM ReservaFinalizada
);

IF (ocupaciones > 0) THEN
    raise_application_error(-20750, 'Habitación ya ocupada.');
```

END IF;

END;

3. SUBSISTEMA DE GESTIÓN DE MANTENIMIENTO:

3.1 Un empleado no puede limpiar más de 10 habitaciones al día:

```

CREATE OR REPLACE TRIGGER maximoHabitacionesDia
BEFORE INSERT OR UPDATE ON Limpieza
FOR EACH ROW
DECLARE
    numlimp INTEGER;
BEGIN
    SELECT COUNT(*) INTO numlimp FROM Limpieza l WHERE( l.DNI =
:new.DNI AND TO_CHAR(l.FechaHora, 'YYYY-MM-DD') =
TO_CHAR(:new.FechaHora, 'YYYY-MM-DD'));
    IF (numlimp >= 10) THEN
        raise_application_error(-20600, :new.DNI || ' no puede
limpiar mas de 10 habitaciones en un dia');
    END IF;
END;
```

3.2 La fecha debe ser posterior o igual a la fecha actual:

```

CREATE OR REPLACE TRIGGER fechaEntradaPosteriorL
BEFORE INSERT ON Limpieza
FOR EACH ROW
BEGIN
```

```

        IF (FLOOR(MONTHS_BETWEEN(:new.FechaHora, SYSDATE)/12) < 0)
THEN
        RAISE_APPLICATION_ERROR(-20111, 'Error: la fecha de
entrada debe ser posterior a la fecha actual');
        END IF;
        END;

```

4. SUBSISTEMA DE GESTIÓN DE EVENTOS:

4.1 La sala debe estar disponible (máximo 2 eventos por sala):

```

CREATE OR REPLACE TRIGGER salaDisponible
BEFORE INSERT OR UPDATE ON EventoTieneLugarEn
FOR EACH ROW
DECLARE
    numevent INTEGER;
BEGIN
    SELECT count(*) INTO numevent FROM EventoTieneLugarEn e WHERE
(e.IdentificadorSala = :new.IdentificadorSala AND e.FechaHora =
:new.FechaHora);
    IF (numevent > 2) THEN
        raise_application_error(-20601, :new.IdentificadorSala || '
no puede albergar mas eventos en esa fecha y hora');
    END IF;
END;

```

Breve motivación de la elección de software

Hemos utilizado el mismo software que usamos en el Seminario 2. La aplicación se ha desarrollado en Python utilizando para acceder a la base de datos la biblioteca pyodbc a través del driver ODBC. Como base de datos se ha usado la proporcionada por la ETSIIT, en Oracle.

De este modo, hacemos uso de una base de datos relacional que resulta conveniente para la implementación de nuestro sistema y a la que tenemos fácil acceso, así como de un método de conexión sencillo que ya habíamos probado con anterioridad.