

---

# PRÁCTICA 3:

# IMPLEMENTACIÓN

---

Alejandro Cárdenas Barranco

Mónica Calzado Granados

Jesús García León

Álvaro Rodríguez Gallardo

Juan Manuel Rodríguez Gómez

Doble Grado en Ingeniería Informática y Matemáticas

Diseño y Desarrollo de Sistemas de Información (Grupo 2)

Curso 2022 – 2023



UNIVERSIDAD  
DE GRANADA

# Índice

## Contenido de la Práctica 1

1. Nombre del Sistema y Distribución de los Subsistemas .....	1
2. Descripción del Sistema .....	1
3. Listado de Requisitos Funcionales, Requisitos de Datos y Restricciones Semánticas .....	9
4. DFD Esquema de Caja Negra .....	29
5. DFD Armazón (DFD0).....	30
6. DFD Refinado de cada Subsistema (DFD1) .....	31

## Contenido de la Práctica 2

7. Esquemas Externos de DFD0 .....	34
8. Esquemas Externos de DFD1 .....	37
9. Esquema E/R Completo del Sistema .....	48
10. Tablas Obtenidas a partir del Esquema E/R .....	49
11. Dependencias Funcionales y Normalización de las Tablas .....	50
12. Esquema de Tablas tras la Normalización .....	59
13. Sentencias SQL para la Creación de las Tablas y la Inserción de Tuplas.....	59

## Contenido de la Práctica 3

14. Descripción de las Transacciones Identificadas.....	¡Error! Marcador no definido.
15. Sentencias SQL para la Creación de los Disparadores .....	¡Error! Marcador no definido.
16. Elección del Software .....	¡Error! Marcador no definido.

## **1. Nombre del Sistema y Distribución de los Subsistemas**

- **Nombre del Sistema:**

*DDSI Hotel Paradise*

- **Distribución de los Subsistemas:** Dentro de nuestro sistema se diferencian principalmente 5 subsistemas, los cuales se mencionan a continuación junto a sus correspondientes encargados:
  - **Trabajadores** (Encargada: Mónica Calzado Granados).
  - **Reservas del Hotel** (Encargado: Alejandro Cárdenas Barranco).
  - **Actividades de Ocio** (Encargado: Álvaro Rodríguez Gallardo).
  - **Parking** (Encargado: Juan Manuel Rodríguez Gómez).
  - **Mantenimiento de las Instalaciones** (Encargado: Jesús García León).

**NOTA: Los cambios realizados tras la corrección de la práctica 1 y 2 se han remarcado con el fondo amarillo.**

## **2. Descripción del Sistema**

Este sistema es un sistema de información multiusuario para la gestión de un hotel. El sistema será gestionado y usado principalmente por administradores. Además, también lo usarán los trabajadores del hotel para ofrecer sus servicios a los clientes.

Por un lado, para poder usar el sistema siendo administrador/trabajador, este deberá proporcionar su DNI/Pasaporte (en una serie de hasta 15 caracteres) y una contraseña (en una serie de hasta 20 caracteres). Para cada administrador/trabajador, se almacenará en el sistema su nombre (en una serie de hasta 20 caracteres), sus apellidos (en una serie de hasta 30 caracteres), su DNI/Pasaporte (en una serie de hasta 15 caracteres), su fecha de nacimiento (date), su correo electrónico (en una serie de hasta 50 caracteres), su teléfono móvil (en una serie de hasta 15 caracteres), su cargo (en una serie de hasta 20 caracteres, aquí se indicará si es un administrador o un trabajador del hotel) y su departamento (en una serie de hasta 20 caracteres). El trabajador no podrá darse de baja del sistema salvo que deje su trabajo, en cuyo caso será un administrador el que le dé de baja del sistema.

Por otro lado, para cada cliente, se almacenará en el sistema los siguientes datos: su nombre (en una serie de hasta 20 caracteres), sus apellidos (en una serie de hasta 30 caracteres), su DNI/Pasaporte (en una serie de hasta 15 caracteres), su correo electrónico (en una serie de hasta 50 caracteres), su teléfono móvil (en una serie de hasta 15 caracteres) y su cuenta bancaria (en una serie de hasta 30 caracteres).

Dentro del sistema, podemos diferenciar principalmente 5 subsistemas:

- 1) **Trabajadores**: Se encarga de todo lo relacionado con las contrataciones y los datos de los trabajadores del hotel. De cada trabajador se almacenará en el sistema lo mencionado anteriormente.

Sus principales funcionalidades son las siguientes:

- 1.1) **Dar de alta a un trabajador**: Para ello, un administrador deberá proporcionar al sistema todos los datos relativos al trabajador, es decir, su nombre, sus apellidos, su DNI/Pasaporte, su fecha de nacimiento, su correo electrónico, su teléfono móvil, su cargo y su departamento. Tras esto, el sistema confirmará la inserción de los datos o dará un error.
- 1.2) **Dar de baja a un trabajador**: Para realizar esta acción, un administrador proporcionará el DNI de este, y el sistema confirmará el borrado o dará un error. Además, se debe realizar un borrado en cascada pues podría haber alguna actividad de ocio que se quede sin un organizador, por tanto, se borran todos los datos de dicha actividad de ocio.
- 1.3) **Mostrar trabajadores de un departamento**: Un administrador o un trabajador proporcionará un departamento al sistema y este desplegará una lista de los trabajadores pertenecientes a dicho departamento, mostrando el nombre, los apellidos y el cargo de cada trabajador.
- 1.4) **Modificar datos de un trabajador**: Un administrador proporcionará el DNI del trabajador y el sistema desplegará toda la información almacenada de dicho trabajador. Luego, el administrador modificará los datos necesarios y finalmente el sistema confirmará la modificación o dará un error.
- 1.5) **Consultar datos de un trabajador**: Un administrador proporcionará el DNI del trabajador y el sistema desplegará toda la información almacenada de dicho trabajador excepto su contraseña para introducirse en el sistema.

- 2) **Reservas del Hotel**: Se encarga de la gestión de las diferentes reservas del hotel realizadas por los clientes. De cada reserva del hotel se almacenará:

DNI/Pasaporte del cliente (en una serie de hasta 15 caracteres), tipo de habitación (en una serie de hasta 10 caracteres), número de cuenta bancaria del cliente (en una serie de hasta 30 caracteres), número de teléfono del cliente (en una serie de hasta 15 caracteres), fecha de entrada (date), fecha de salida (date), código identificador de la reserva (en una serie de hasta 10 caracteres), número de habitación (número entero positivo de 3 dígitos), check-in (bool) que será true o false en función de si el cliente ha realizado el check-in, check-out (bool) y cancelada (bool) que será true si la reserva ha sido cancelada.

De cada habitación se almacenará: el número de habitación (número entero positivo de 3 dígitos) que será el identificador de la habitación y el tipo de habitación (en una serie de hasta 10 caracteres) y ocupada (bool) que será true si la habitación está siendo hospedada.

Sus principales funcionalidades son las siguientes:

**2.1) Registrar reserva de habitación:** El registro de la reserva en el sistema la realizará el trabajador cuando el cliente la solicite acudiendo al hotel presencialmente o llamando por teléfono. De una forma u otra, se le proporcionará al sistema todos los datos relativos al cliente, es decir, su nombre, sus apellidos, su DNI/Pasaporte, su teléfono móvil y su cuenta bancaria. También se registrará el tipo de habitación que el cliente desea (individual, doble o suite), aunque no se le asignará una habitación específica hasta el momento en que el cliente llegue al hotel. Por último, se especificará al sistema la fecha de llegada del cliente al hotel y la de salida. El sistema creará un identificador asociado a dicha reserva que será único y se le dará al cliente para gestionar su reserva. Un cliente solamente podrá tener una reserva a su nombre. Una reserva sólo podrá estar a nombre de un cliente.

**2.2) Realizar Check-in:** Una vez que el cliente llegue al hotel, este tendrá que comentarle al trabajador de recepción que tiene una reserva a su nombre, indicando su código de reserva y su DNI/Pasaporte. Tras ello, el trabajador le proporcionará al sistema el DNI/Pasaporte código de reserva, comprobando si hay una reserva hecha por parte del cliente. Tras confirmar que la reserva está hecha se marcará con true el atributo check-in del registro reserva para indicar que el cliente ha realizado el check-in, se comprobará la fecha de entrada

de dicha reserva. Se le asignará una habitación al cliente del tipo requerido por este, quedando registrado en el sistema el número de habitación que se le ha asignado a dicho cliente, ~~incluyéndose dentro del registro reserva. Además, esta habitación se marcará como ocupada.~~

**2.3) Realizar Check-out:** Cuando el cliente vaya a marcharse de su habitación, éste acudirá a recepción para informárselo al trabajador. El cliente le indicará al trabajador el código de su reserva. Tras ello, el trabajador le proporcionará dicho código al sistema, que le devolverá el número de habitación en la que se ha hospedado el cliente, ~~así como la fecha de salida.~~ Se marcará como true el atributo check-out de la reserva ~~y se marcará como false el atributo ocupada en habitación.~~

**2.4) Listar disponibilidad de habitación:** El trabajador le proporcionará al sistema el tipo de habitación de la que se desea comprobar la disponibilidad, ~~así como la fecha de entrada y la fecha de salida del cliente del hotel.~~ El sistema mostrará un listado de las habitaciones del tipo especificado disponibles ~~en la fecha indicada,~~ indicando únicamente el número de habitación.

**2.5) Cancelar reserva de habitación:** La eliminación de la reserva en el sistema la realizará el trabajador cuando el cliente la solicite acudiendo al hotel presencialmente o llamando por teléfono. De una forma u otra, se le proporcionará al sistema el código identificador de la reserva. Se marcará como true el atributo cancelada de dicha reserva.

**3) Actividades de Ocio:** Se encarga de la gestión de las diferentes actividades de ocio organizadas por el hotel para que las realicen los clientes. De cada actividad de ocio se almacenará: código identificador (en una serie de hasta 10 caracteres), nombre (en una serie de hasta 20 caracteres), ~~DNI/Pasaporte del trabajador responsable de dicha actividad (en una serie de hasta 15 caracteres), teléfono de dicho trabajador (en una serie de hasta 15 caracteres),~~ descripción (en una serie de hasta 100 caracteres), la fecha en la que estará disponible (date), el horario en el que se realizará (~~time~~ date), el número total de plazas disponibles (número entero positivo de 2

dígitos). Dichas actividades de ocio estarán relacionadas con las opiniones (puntuación sobre 10 y comentario de hasta 200 caracteres) realizadas por los clientes. También, cada actividad de ocio estará relacionada con el trabajador que la organiza.

Sus principales funcionalidades son las siguientes:

**3.1) Registrar actividad de ocio:** El registro de una actividad de ocio del hotel será realizado por un administrador. Para ello, el administrador le proporcionará al sistema el nombre de la actividad, ~~DNI/Pasaporte del trabajador responsable de dicha actividad, teléfono de dicho trabajador~~, una descripción de esta, la fecha en la que estará disponible, el horario en el que se realizará dicha actividad y el número de plazas de la actividad (número entero positivo de 2 dígitos). Si el momento de realización no está disponible, devuelve error. El sistema creará un identificador asociado a dicha actividad que será único. Se pueden hacer registros de la misma actividad en diferentes momentos, pero esta operación solo registrará una de ellas, por lo que, si se realizará N veces la actividad, cada vez en un momento diferente, tendrá que realizarse esta acción N veces. **Tras ello, se relacionará dicha actividad de ocio con el DNI/Pasaporte del trabajador responsable de realizar dicha actividad.**

**3.2) Reservar actividad de ocio:** Para ello, el cliente acudirá presencialmente al trabajador y este último le proporcionará al sistema el nombre **y el código identificador** de la actividad que quiere reservar, así como el nombre y el DNI/Pasaporte del cliente. Dicho cliente deberá tener asignada una habitación del hotel. En caso de no ser así, el sistema mostrará un error. Si la condición anterior se cumple, el sistema desplegará las diferentes fechas disponibles con el correspondiente horario en las que se realiza la actividad que se encuentre dentro de las fechas de estancia del cliente. Siempre que no se supere el límite de plazas disponibles para la actividad que se realiza en un momento determinado (el trabajador introducirá también, por petición del cliente, el número de personas que reservaran la actividad, ya que el cliente podría ir acompañado en su estancia del hotel), el sistema permitirá realizar la reserva. En otro caso, dará error.

**3.3) Consultar disponibilidad de una actividad de ocio:** Para ello, el cliente acudirá presencialmente al trabajador y este último le proporcionará al sistema el nombre de la actividad de la cual quiere consultar su disponibilidad y la fecha en la que quiere consultar

dicha disponibilidad. Tras ello, el sistema desplegará un listado de la disponibilidad de la actividad en la fecha introducida. Dicho listado contendrá los diferentes horarios (de la fecha indicada) en los que se realiza la actividad que se encuentre dentro de las fechas de indicadas.

**3.4) Opinar sobre actividad de ocio:** Para ello, el cliente rellenará un papel con su opinión sobre la actividad y se lo entregará al trabajador responsable. Luego, el trabajador le proporcionará al sistema el nombre y el código identificador de la actividad de la cual el cliente ha opinado y la opinión correspondiente, la cual se almacenará en el sistema de forma anónima (no se registrará ningún dato personal del cliente, aunque se deberá proporcionar al sistema el DNI/Pasaporte del cliente solo se almacenará en dicha opinión su DNI/Pasaporte para evitar opiniones de personas externas al hotel). Finalmente, el sistema confirmará si se ha registrado la opinión correctamente o se ha producido algún error.

**3.5) Consultar opiniones de una actividad de ocio:** A petición de un cliente, el trabajador proporcionará al sistema el nombre y el código identificador de la actividad de la cual el cliente quiere visualizar las opiniones. Tras ello, el sistema desplegará todas las opiniones de la actividad especificada.

**3.6) Cancelar reserva de una actividad de ocio:** La cancelación de la reserva de la actividad la realiza el trabajador cuando el cliente acude presencialmente a la recepción del hotel. De una forma u otra, se le proporcionará al sistema el DNI/Pasaporte del cliente que quiere cancelarla, y se mostrarán todas las actividades reservadas por el cliente. El cliente le comentará al trabajador la reserva de actividad que quiere cancelar, el trabajador seleccionará dicha actividad para cancelarla y finalmente el sistema eliminará dicha reserva.

**4) Parking:** Se encarga de la gestión de las plazas del parking del hotel, así como del pago de su uso. Se almacenará en el sistema de cada plaza: su número identificativo (número entero positivo de 3 dígitos) y si la plaza está ocupada o no en ese momento (boolean, true si la plaza está ocupada y false si no lo está), el tiempo que lleva ocupando el cliente dicha plaza



(time). En caso de estar ocupada, dicha plaza estará relacionada con el DNI/Pasaporte del cliente que la ocupa (en una serie de hasta 15 caracteres) y se contará el tiempo en el que el coche del cliente se encuentra en la plaza (número entero).

Sus principales funcionalidades son las siguientes:

- 4.1) Registrar reserva de plaza de parking:** El cliente debe realizar la reserva presencialmente y será el trabajador el que le proporcione al sistema el DNI/Pasaporte del cliente. Dicho cliente deberá tener asignada una habitación del hotel. En caso de no ser así, el sistema mostrará un error. Si la condición anterior se cumple, el sistema le asignará al cliente una plaza del parking que se encuentre libre en ese momento como máximo hasta el último día de estancia del cliente en el hotel.
- 4.2) Consultar reserva de plaza de parking:** Un trabajador proporcionará al sistema el DNI/Pasaporte de un determinado cliente. En caso de tener reservada una plaza, se mostrará el DNI/Pasaporte del cliente, así como el número de plaza que tiene asignada.
- 4.3) Consultar plazas disponibles del parking:** El trabajador hará que el sistema despliegue un listado con el número de plazas del parking que están disponibles (es decir, que no están ocupadas).
- 4.4) Mostrar cantidad a pagar del parking:** Cuando el cliente abandone el hotel, deberá realizar el pago del parking en función del tiempo que haya estado ocupando la plaza (para lo cual hay detectores que recogen en cada momento la hora a la que entra el coche al parking y la hora a la que sale de él, llevando así un contador del tiempo total en minutos que el coche se encuentra en el parking durante la estancia del cliente en el hotel). Para realizar el pago, el cliente le proporcionará al trabajador su DNI/Pasaporte y este lo introducirá en el sistema, el cual le mostrará el tiempo que ha estado el coche del cliente en la plaza de parking y la cantidad de dinero que debe abonar (esta se calcula multiplicando el tiempo que ha estado ocupando la plaza por 2 céntimos el minuto). En caso de que el cliente haya usado la plaza de parking más de 1 hora, deberá realizar el pago correspondiente. En otro caso, no deberá pagar nada.
- 4.5) Cancelar reserva de plaza de parking:** La cancelación de la reserva de la plaza de parking en el sistema será realizada por el trabajador

cuando el cliente acuda presencialmente a la recepción del hotel. Se le proporcionará al sistema el DNI/Pasaporte del cliente que quiere cancelarla. En caso de cancelarla tras ocupar la plaza durante un tiempo, el cliente deberá pagar lo que corresponda. Tras la cancelación, el atributo ocupada pasa a estar a false (ya que la plaza deja de estar ocupada) y se eliminará el DNI/Pasaporte del cliente la reserva asociado a dicha plaza.

- 5) **Mantenimiento de las Instalaciones:** Se encarga de mantener actualizadas las diferentes reparaciones que necesitan las habitaciones del hotel. Por un lado, de cada desperfecto se almacenará: código identificador (en una serie de hasta 10 caracteres), descripción (en una serie de hasta 100 caracteres), lugar donde se ha observado (en una serie de hasta 20 caracteres), fecha en la que se ha observado (date), precio de su reparación (número en coma flotante) y prioridad de su arreglo (número entero positivo de 3 dígitos, cuanto menor sea dicho número, mayor prioridad tendrá el desperfecto para ser arreglado). El desperfecto estará relacionado con el número de habitación donde se ha producido.

Por otro lado, de cada producto de limpieza/herramienta para arreglar los desperfectos se almacenará: código identificador (en una serie de hasta 10 caracteres), nombre (en una serie de hasta 20 caracteres), cantidad disponible (número entero positivo de 3 dígitos) y compañía distribuidora (en una serie de hasta 20 caracteres). Cada desperfecto estará relacionado con una serie de utensilios.

Sus principales funcionalidades son las siguientes:

- 5.1) **Añadir desperfecto del hotel:** El trabajador responsable introducirá en el sistema una descripción sobre los daños existentes, el lugar número de habitación en el que se han producido dichos daños, la fecha en la que se ha visto dicho desperfecto y su prioridad de arreglo. El sistema creará un identificador asociado a dicho desperfecto que será único. No se introducirá en este momento el precio de reparación del desperfecto, ya que este debe pensarse primero para introducirlo posteriormente.
- 5.2) **Asignar precio a reparación de desperfecto:** El administrador le asignará un precio u otro a la reparación de un desperfecto de una habitación en función de su prioridad. Si a la hora de asignar el precio a la reparación, esto hace que la suma total de todos los precios de las diferentes reparaciones de los desperfectos sea mayor que el presupuesto disponible (el cual es un número decimal fijo y

conocido por el administrador), entonces el sistema devolverá error y no podrá asignarse dicho precio. En caso de no cumplirse lo anterior, el sistema asigna dicho precio a la reparación sin proporcionar ningún error

- 5.3) Consultar precio de una reparación de desperfecto:** El administrador le proporcionará al sistema el código identificador del desperfecto del hotel y el número de la habitación donde se encuentra dicho desperfecto. El sistema mostrará el precio asignado a dicho desperfecto (si no estuviera asignado el precio, el sistema no mostrará nada).
- 5.4) Consultar desperfectos del hotel:** El trabajador responsable hará que el sistema muestre un listado que contendrá el código identificador del desperfecto, su descripción, su ubicación, la fecha en la que se observó y su prioridad. Los desperfectos de dicho listado estarán ordenados de mayor a menor prioridad.
- 5.5) Consultar stock de productos de limpieza y herramientas:** El trabajador responsable hará que el sistema muestre un listado que contendrá el código identificador, el nombre, la descripción y la cantidad disponible de cada producto. Los productos de dicho listado estarán ordenados alfabéticamente.

### **3. Listado de Requisitos Funcionales, Requisitos de Datos y Restricciones Semánticas**

#### **Requisitos Funcionales y Requisitos de Datos**

## Subsistema 1: Trabajadores

### Encargada: Mónica Calzado Granados

**RF1.1:** Dar de alta a un trabajador.

**Entrada:** Agente externo: administrador. Acción: solicitar inserción de los datos de un nuevo trabajador. Requisito de datos de entrada: **RDE1.1**.

**BD:** Requisito de datos de escritura: **RDW1.1**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE1.1:** Datos de entrada de alta de trabajador.

- Nombre: cadena de caracteres (20)
- Apellidos: cadena de caracteres (30)
- DNI/Pasaporte: cadena de caracteres (15)
- Fecha de nacimiento: date
- Correo electrónico: cadena de caracteres (50)
- Teléfono móvil: cadena de caracteres (15)
- Cargo: cadena de caracteres (20)
- Departamento: cadena de caracteres (20)

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

- Los mismos que RDE1.1

**RF1.2:** Dar de baja a un trabajador.

**Entrada:** Agente externo: administrador. Acción: solicitar borrado de los datos de un trabajador. Requisito de datos de entrada: **RDE1.2**.

**BD:** Requisitos de datos de escritura: **RDW1.2**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE1.2:** Datos de entrada de baja de trabajador.

- DNI/Pasaporte: cadena de caracteres (15)

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

- Los mismos que RDW1.1

**Datos almacenados de las actividades de ocio asociadas al trabajador.**

- Los mismos que RDE3.1.

**RF1.3:** Mostrar trabajadores de un departamento.

**Entrada:** Agente externo: administrador/trabajador. Acción: solicitar listado de trabajadores de un departamento. Requisito de datos de entrada **RDE1.3**.

**BD:** Requisito de datos de lectura: **RDR1.3**.

**Salida:** Agente externo: administrador/trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS1.3**.

**RDE1.3:** Datos de entrada de consulta de trabajadores.

- Departamento: cadena de caracteres (20)

**RDR1.3:** Datos almacenados de trabajador.

- Nombre: Cadena de caracteres (20)
- Apellidos: Cadena de caracteres (30)
- Cargo: cadena de caracteres (20)

**RDS1.3:** Listado de registros, cada uno de ellos con la información de cada trabajador, la cual contiene los mismos datos que RDR1.3.

**RF1.4:** Modificar datos de un trabajador.

**Entrada:** Agente externo: administrador. Acción: solicitar modificación de los datos de un trabajador. Requisito de datos de entrada: **RDE1.4**.

**BD:** Requisito de datos de escritura: **RDW1.4**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE1.4:** Datos de entrada de modificación de trabajador.

- Los mismos que RDE1.1

**RDW1.4:** Datos almacenados de trabajador (los mismos que RDE1.4 excepto DNI/Pasaporte).

- Nombre: Cadena de caracteres (20)
- Apellidos: Cadena de caracteres (30)
- Fecha de nacimiento: date
- Correo electrónico: cadena de caracteres (50)
- Teléfono móvil: cadena de caracteres (15)
- Cargo: cadena de caracteres (20)
- Departamento: cadena de caracteres (20)

- Los mismos que RDE1.4

**RF1.5:** Consultar datos de un trabajador.

**Entrada:** Agente externo: administrador. Acción: solicitar datos de un trabajador. Requisito de datos de entrada: **RDE1.5**.

**BD:** Requisito de datos de lectura: **RDR1.5**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: **RDS1.5**.

**RDE1.5:** Datos de entrada de consulta de trabajador.  
- DNI/Pasaporte: cadena de caracteres (15)

**RDR1.5:** Datos almacenados de trabajador.  
- Los mismos que RDW1.1

**RDS1.5:** Datos del trabajador.  
- Los mismos que RDR1.5

## **Subsistema 2: Reservas del Hotel**

**Encargado: Alejandro Cárdenas Barranco**

**RF2.1:** Registrar reserva de habitación.

**Entrada:** Agente externo: trabajador. Acción: solicitar inserción de reserva en el sistema de información. Requisito de datos de entrada: **RDE2.1**.

**BD:** Requisito de datos de escritura: **RDW2.1**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS2.1**.

**RDE2.1:** Datos de entrada al reservar una habitación.

- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Tipo de habitación: cadena de caracteres (10)
- Número de cuenta bancaria cliente: cadena de caracteres (30)
- Número de teléfono cliente: cadena de caracteres (15)
- Fecha de entrada: date
- Fecha de salida: date

**RDW2.1:** Datos almacenados en el sistema.

- DNI/Pasaporte cliente: cadena de caracteres (15)
- Tipo de habitación: cadena de caracteres (10)
- Número de cuenta bancaria cliente: cadena de caracteres (30)
- Número de teléfono cliente: cadena de caracteres (15)
- Fecha de entrada: date.
- Fecha de salida: date.
- Código identificador de la reserva: cadena de caracteres (10)

**RDS2.1:** Datos asociados al registro de la reserva

- Código identificador de la reserva: cadena de caracteres (10)

**RF2.2:** Realizar check-in.

**Entrada:** Agente externo: trabajador. Acción: confirmar la llegada de un cliente al hotel. Requisito de datos de entrada: **RDE2.2**.

**BD:** Requisito de datos de escritura: **RDW2.2** y de lectura: **RDR2.2**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS2.2**.

**RDE2.2:** Datos de entrada para el check-in.

- Código identificador de la reserva: cadena de caracteres (10)
- ~~DNI/Pasaporte del cliente: cadena de caracteres (15)~~

**RDR2.2:** Datos de reserva almacenados:

- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Tipo de habitación: cadena de caracteres (10)
- Fecha de entrada: date.

**RDW2.2:** Datos almacenados en el sistema al realizar el check-in.

- Registro de reserva
  - + Check-in: boolean
  - + Número de habitación: número entero positivo de 3 dígitos
- Registro de la habitación:
  - + Ocupada: boolean

**RDS2.2:** Datos asociados a la reserva de un cliente en el check-in.

- Número de habitación: número entero positivo de 3 dígitos

**RF2.3:** Realizar check-out.

**Entrada:** Agente externo: trabajador. Acción: confirmar la salida de un cliente del hotel. Requisito de datos de entrada: **RDE2.3**.

**BD:** Requisito de datos de escritura: **RDW2.3** y lectura: **RDR2.3**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE2.3:** Datos de entrada para el check-out.

- Código identificador de la reserva: cadena de caracteres (10)

**RDR2.3:** Datos asociados a la reserva.

- Número de habitación: número entero positivo de 3 dígitos
- Fecha de entrada: date.

**RDW2.3:** Datos almacenados en el sistema al realizar el check-out.

- Registro de reserva
  - + Check-out: boolean
- Registro de habitación
  - + Ocupada: boolean

**RF2.4:** Consultar disponibilidad de un tipo de habitación.

**Entrada:** Agente externo: trabajador. Acción: consultar habitaciones de un tipo disponibles en la fecha indicada. Requisito de datos de entrada: **RDE2.4**.

**BD:** Requisito de datos de lectura: **RDR2.4**.



**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida **RDS2.4**.

**RDE2.4:** Datos de entrada para consultar si una habitación está disponible.

- Tipo de habitación: cadena de caracteres (10)

- Fecha de entrada: date

- Fecha de salida: date

**RDR2.4:** Datos almacenados de las reservas asociadas al tipo de habitación establecido.

- Tipo de habitación: cadena de caracteres (10)

- Fecha de entrada: date

- Fecha de salida: date

- Número de habitación: número entero positivo de 3 dígitos

- Código identificador de la reserva: cadena de caracteres (10)

**RDS2.4:** Listado de habitaciones del tipo indicado que están disponibles en la fecha establecida, cada uno de ellos con:

- Número de habitación: número entero positivo de 3 dígitos

**RF2.5:** Cancelar reserva de habitación.

**Entrada:** Agente externo: trabajador. Acción: cancelar reserva. Requisito de datos de entrada: **RDE2.5**.

**BD:** Requisito de datos de escritura: **RDW2.5**.

**Salida:** Agente externo: trabajador. Acción: Confirma el resultado. Requisito de datos de salida: ninguno

**RDE2.5:** Datos de entrada para cancelar una reserva.

- Código identificador de la reserva: cadena de caracteres (10)

**RDW2.5:** Datos almacenados para modificar los datos de una reserva.

- Cancelada: boolean

### Subsistema 3: Actividades de Ocio

**Encargado: Álvaro Rodríguez Gallardo**

**RF3.1:** Registrar actividad de ocio.

**Entrada:** Agente externo: administrador. Acción: solicitar inserción de una nueva actividad de ocio. Requisito de datos de entrada: **RDE3.1**.

**BD:** Requisito de datos de escritura: **RDW3.1**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: **RDS3.1**.

**RDE3.1:** Datos de entrada de la actividad de ocio.

- Nombre de la actividad: cadena de caracteres (20)
- DNI/Pasaporte del trabajador responsable: cadena de caracteres (15)
- Teléfono del trabajador responsable: cadena de caracteres (15)
- Descripción: cadena de caracteres (100)
- Fecha de realización de la actividad: date
- Horario de realización de la actividad: time date
- Número de plazas: número entero positivo de 2 dígitos
- Código identificador de la actividad: cadena de caracteres (10)

**RDW3.1:** Datos almacenados de la actividad de ocio.

- Los mismos que RDE3.1
- Código identificador de la actividad: cadena de caracteres (10)

**RDS3.1:** Datos asociados al registro de la actividad de ocio.

- Código identificador de la actividad: cadena de caracteres (10)

**RF3.2:** Reservar actividad de ocio.

**Entrada:** Agente externo: trabajador. Acción: solicitar inserción de una reserva de una actividad de ocio. Requisito de datos de entrada: **RDE3.2**.

**BD:** Requisito de datos de escritura: **RDW3.2**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE3.2:** Datos de entrada al reservar una actividad de ocio.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (20)
- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Nombre del cliente: cadena de caracteres (20)
- Descripción: cadena de caracteres (100)

- Fecha de reserva de la actividad: date
- Horario de reserva de la actividad: time date
- Número de personas que reservan la actividad: número entero positivo de 1 dígito

**RDW3.2:** Datos almacenados de la reserva de la actividad de ocio.

- DNI/Pasaporte del cliente: cadena de caracteres (15)
- ~~Nombre del cliente: cadena de caracteres (20)~~
- Número de personas que reservan la actividad: número entero Positivo de 1 dígito

**RF3.3:** Consultar disponibilidad de una actividad de ocio.

**Entrada:** Agente externo: trabajador. Acción: solicitar fechas y horarios disponibles de una actividad de ocio. Requisito de datos de entrada: **RDE3.3**.

**BD:** Requisito de datos de lectura: **RDR3.3**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS3.3**.

**RDE3.3:** Datos de entrada de consulta de disponibilidad de la actividad de ocio.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (20)
- Fecha de realización de la actividad: date

**RDR3.3:** Datos almacenados de la actividad de ocio.

- Descripción: cadena de caracteres (100)
- Fecha de realización la actividad: cadena de caracteres (10)
- Horario de realización la actividad: time date
- Número de plazas que admite la actividad: número entero positivo de 2 dígitos
- Teléfono del trabajador responsable: cadena de caracteres (15)

**RDS3.3:** Listado de registros, cada uno de ellos con la información de la actividad de ocio en la fecha y horario correspondiente.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (20)
- Descripción: cadena de caracteres (100)
- Fecha de la actividad: date
- Horario de la actividad: time date

- Número de plazas que admite la actividad: número entero positivo de 2 dígitos
- Teléfono del trabajador responsable: cadena de caracteres (15)

**RF3.4:** Opinar sobre actividad de ocio.

**Entrada:** Agente externo: trabajador. Acción: solicitar inserción de opinión de una actividad de ocio. Requisito de datos de entrada: **RDE3.4.**

**BD:** Requisito de datos de escritura: **RDW3.4.**

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE3.4:** Datos de entrada de la opinión de la actividad de ocio.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (20)
- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Opinión: cadena de caracteres (200)

**RDW3.4:** Datos almacenados de la opinión de la actividad de ocio.

- Opinión: cadena de caracteres (200)
- Puntuación: número entero del 0 al 10.

**RF3.5:** Consultar opiniones de una actividad de ocio.

**Entrada:** Agente externo: trabajador. Acción: solicitar opiniones de una actividad de ocio. Requisito de datos de entrada: **RDE3.5.**

**BD:** Requisito de datos de lectura: **RDR3.5.**

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS3.5.**

**RDE3.5:** Datos de entrada de consulta de opiniones.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (15)

**RDR3.5:** Datos almacenados de la opinión de la actividad de ocio.

- Opinión: cadena de caracteres (200)
- Puntuación: número entero del 0 al 10
- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDS3.5:** Listado de registros, cada uno de ellos con la opinión de la actividad de ocio correspondiente.

- Lo mismo que RDR3.5

**RF3.6:** Cancelar reserva de una actividad de ocio.

**Entrada:** Agente externo: trabajador. Acción: solicitar borrado de los datos de una reserva de una actividad de ocio. Requisito de datos de entrada: **RDE3.6**.

**BD:** Requisito de datos de escritura **RDW3.6**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE3.6:** Datos de entrada al cancelar la reserva de la actividad de ocio.

- Código identificador de la actividad: cadena de caracteres (10)
- Nombre de la actividad: cadena de caracteres (20)
- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Fecha de reserva de la actividad: cadena de caracteres (10)
- Horario de reserva de la actividad: **time date**
- Número de personas que realizan la actividad: número entero positivo de 1 dígito

**RDW3.6:** Datos almacenados de la reserva de la actividad de ocio.

~~- Lo mismo que RDW3.2~~

- Código identificador de la actividad: cadena de caracteres (10)

## **Subsistema 4: Parking**

**Encargado: Juan Manuel Rodríguez Gómez**

**RF4.1:** Registrar reserva de plaza de parking.

**Entrada:** Agente externo: trabajador. Acción: solicitar inserción de una nueva reserva de plaza de parking. Requisito de datos de entrada: **RDE4.1**.

**BD:** Requisito de datos de escritura: **RDW4.1** y de lectura: **RDR4.1**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS4.1**.

**RDE4.1:** Datos de entrada al reservar la plaza de parking.

- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDR4.1:** Datos almacenados de la plaza de parking.

- Número de la plaza del parking: número entero positivo de 3 dígitos
- ~~Ocupada: boolean.~~

**RDW4.1:** Datos almacenados de la reserva de la plaza de parking.

- ~~Ocupada: boolean.~~
- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDS4.1:** Datos asociados a la reserva de la plaza de parking.

- Número de la plaza del parking: número entero positivo de 3 dígitos
- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RF4.2:** Consultar reserva de plaza de parking.

**Entrada:** Agente externo: trabajador. Acción: consultar reserva de plaza de parking. Requisito de datos de entrada: **RDE4.2**.

**BD:** Requisito de datos de lectura: **RDR4.2**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS4.2**.

**RDE4.2:** Datos de entrada de consulta de la reserva de la plaza del parking.

- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDR4.2:** Datos almacenados de la reserva de la plaza del parking.

- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Número de la plaza del parking: número entero positivo de 3 dígitos
- Tiempo que lleva ocupando la plaza el cliente: ~~time~~ número entero positivo

**RDS4.2:** Datos de la reserva de la plaza del parking.  
- Los mismos que RDR4.2

**RF4.3:** Consultar plazas disponibles del parking.

**Entrada:** Agente externo: trabajador. Acción: consultar plazas disponibles del parking. Requisito de datos de entrada: ninguno.

**BD:** Requisito de datos de lectura: **RDR4.3**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS4.3**.

**RDR4.3:** Datos almacenados de la reserva de la plaza del parking.  
- Número de la plaza del parking: número entero positivo de 3 dígitos  
- Ocupada: boolean

**RDS4.3:** Listado de registros de las plazas disponibles, cada uno de ellos con la siguiente información:  
- Número de la plaza del parking: número entero positivo de 3 dígitos

**RF4.4:** Mostrar cantidad a pagar del parking.

**Entrada:** Agente externo: trabajador. Acción: mostrar cantidad que debe pagar un cliente la ocupación de su plaza de parking. Requisito de datos de entrada: **RDE4.4**.

**BD:** Requisito de datos de lectura: **RDR4.4**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS4.4**.

**RDE4.4:** Datos de entrada de la reserva de la plaza del parking.  
- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDR4.4:** Datos almacenados de la reserva de la plaza del parking.  
- DNI/Pasaporte del cliente: cadena de caracteres (15)  
- Número de la plaza del parking: número entero positivo de 3 dígitos  
- Ocupada: boolean  
- Tiempo que lleva ocupando la plaza el cliente: ~~time~~ número entero positivo

**RDS4.4:** Datos de la cantidad que debe pagar el cliente por el uso de la plaza de parking.

- Dinero (en euros) a pagar por el cliente por el uso de la plaza de parking: número en coma flotante (este se calculará como el tiempo que el cliente ha ocupado la plaza, expresado en minutos, por una tarifa de 2 céntimos por minuto).

**RF4.5:** Cancelar reserva de plaza de parking.

**Entrada:** Agente externo: trabajador. Acción: cancelar reserva de la plaza de parking. Requisito de datos de entrada: **RDE4.5**.

**BD:** Requisito de datos de escritura: **RDW4.5**

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno

**RDE4.5:** Datos de entrada para cancelar una reserva de una plaza del parking.

- DNI/Pasaporte del cliente: cadena de caracteres (15)

**RDW4.5:** Datos almacenados para cancelar la reserva de la plaza del parking.

- DNI/Pasaporte del cliente: cadena de caracteres (15)
- Ocupada: boolean
- Tiempo que lleva ocupando la plaza el cliente: time número entero positivo



## Subsistema 5: Mantenimiento de Instalaciones

Encargado: Jesús García León

**RF5.1:** Añadir desperfecto del hotel.

**Entrada:** Agente externo: trabajador. Acción: solicitar inserción de un desperfecto del hotel. Requisito de datos de entrada: **RDE5.1**.

**BD:** Requisito de datos de escritura: **RDW5.1**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE5.1:** Datos de entrada del desperfecto del hotel.

- Código identificador del desperfecto: cadena de caracteres (10)
- Descripción: cadena de caracteres (200)
- Lugar donde se ha producido el desperfecto: cadena de caracteres (20)

- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

- Fecha en la que se ha visto el desperfecto: date
- Prioridad de arreglo: número entero positivo de 3 dígitos

**RDW5.1:** Datos almacenados del desperfecto del hotel.

- Lo mismo que RDE5.1

**RF5.2:** Asignar precio a reparación de desperfecto.

**Entrada:** Agente externo: administrador. Acción: solicitar inserción de precio de reparación de un desperfecto. Requisito de datos de entrada: **RDE5.2**.

**BD:** Requisito de datos de escritura: **RDW5.2**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: ninguno.

**RDE5.2:** Datos de entrada del desperfecto del hotel.

- Código identificador del desperfecto: cadena de caracteres (10)
- Precio de la reparación del desperfecto: número en coma flotante
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RDR5.2:** Datos almacenados del desperfecto del hotel.

- Código identificador del desperfecto: cadena de caracteres (10)
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RDW5.2:** Datos almacenados del desperfecto del hotel.

- Precio de la reparación del desperfecto: número en coma flotante
- Código identificador del desperfecto: cadena de caracteres (10)
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RF5.3:** Consultar precio de una reparación de desperfecto.

**Entrada:** Agente externo: administrador. Acción: solicitar precio de una reparación del desperfecto del hotel. Requisito de datos de entrada: **RDE5.3**.

**BD:** Requisito de datos de lectura: **RDR5.3**.

**Salida:** Agente externo: administrador. Acción: confirmar resultado. Requisito de datos de salida: **RDS5.3**.

**RDE5.3:** Datos de entrada del desperfecto del hotel.

- Código identificador del desperfecto: cadena de caracteres (10)
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RDR5.3:** Datos almacenados del desperfecto del hotel.

- Precio de la reparación del desperfecto: número en coma flotante
- Código identificador del desperfecto: cadena de caracteres (10)
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RDS5.3:** Datos acerca del desperfecto.

- Código identificador del desperfecto: cadena de caracteres (10)
- Precio de la reparación del desperfecto: número en coma flotante.
- Número de habitación donde se ha producido el desperfecto: número entero positivo de 3 dígitos

**RF5.4:** Consultar desperfectos del hotel.

**Entrada:** Agente externo: trabajador. Acción: solicitar datos de los diferentes desperfectos del hotel. Requisito de datos de entrada: ninguno.

**BD:** Requisito de datos de lectura: **RDR5.4**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS5.4**.

**RDR5.4:** Datos almacenados del desperfecto del hotel.

- Lo mismo que RDE5.1

- Código identificador del desperfecto: cadena de caracteres (10)
- Descripción: cadena de caracteres (200)
- Fecha en la que se ha visto el desperfecto: date
- Prioridad de arreglo: número entero positivo de 3 dígitos

**RDS5.4:** Listado de registros, cada uno de ellos con la información de cada desperfecto, la cual contiene los mismos datos que RDR5.4.

**RF5.5:** Consultar stock de productos de limpieza y herramientas.

**Entrada:** Agente externo: trabajador. Acción: solicitar datos de la cantidad de los diferentes productos de limpieza y herramientas disponibles para arreglar los desperfectos. Requisito de datos de entrada: ninguno.

**BD:** Requisito de datos de lectura: **RDR5.5**.

**Salida:** Agente externo: trabajador. Acción: confirmar resultado. Requisito de datos de salida: **RDS5.5**.

**RDR5.5:** Datos almacenados de los diferentes productos de limpieza y herramientas.

- Código identificador: cadena de caracteres (10)
- Nombre: cadena de caracteres (20)
- Cantidad disponible: Número entero positivo de 3 dígitos
- ~~Compañía distribuidora: cadena de caracteres (20)~~

**RDS5.5:** Listado de registros, cada uno de ellos con la información acerca de los diferentes productos de limpieza y herramientas de los que dispone el hotel con los mismos datos que RDR5.5.

## Restricciones Semánticas

### Subsistema 1: Trabajadores

Encargada: Mónica Calzado Granados

**RS1.1:** El trabajador debe ser mayor de 16 años.

**RF:** RF1.1

**RD(s):** RDE1.1

**Descripción:** “Solo se podrá dar de alta un nuevo trabajador si este cumple la mayoría de edad legal para trabajar, es decir, tiene más de 16 años. Si no lo cumple, el sistema no dejará insertar los datos del nuevo trabajador y dará un error.”

### Subsistema 2: Reservas del Hotel

Encargado: Alejandro Cárdenas Barranco

**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 crea una nueva reserva y se devuelve un error.”

**RS2.2:** 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 anterior a la fecha de entrada, no se crea una nueva reserva y se devuelve un error.”

**RS2.3:** El check-out debe estar asociado a un check-in

**RF:** RF2.3

**RD(s):** RDE2.3

**Descripción:** “Si se intenta realizar un check-out en una reserva en la que no se haya realizado un check-in, no se podrá marcar el check-out y se devolverá un error.”

**RS2.4:** Una reserva en la que se haya realizado un check-in no puede ser cancelada.

**RF:** RF2.5

**RD(s):** RDE2.5

**Descripción:** “Si se intenta cancelar una reserva en la que previamente se ha realizado un check-in, el sistema no la cancelará y devolverá un error.”

### Subsistema 3: Actividades de Ocio

#### Encargado: Álvaro Rodríguez Gallardo

**RS3.1:** El cliente tiene asignada una habitación.

**RF:** RF3.2, RF4.1

**RD(s):** RDW3.2, RDW4.1

**Descripción:** “Para poder realizar una reserva de una actividad de ocio o de una plaza de parking, el cliente debe estar ocupando una habitación del hotel. En caso de no ser así, no podrá reservar la actividad o la plaza.”

**RS3.2:** No se puede superar el máximo de plazas permitidas.

**RF:** RF3.2

**RD(s):** RDE3.2, RDW3.2

**Descripción:** “Solo se podrá reservar una actividad si el cupo de plazas máximo no ha sido superado. Si sumando a las personas que quieren reservar la actividad se llega a dicho máximo, se devuelve un error.”

**RS3.3:** ~~No se puede superar el máximo de plazas permitidas.~~

~~No puede indicar eliminar más plazas de las reservadas.~~

**RF:** RF3.2

**RD(s):** RDE3.2, RDW3.2

~~**Descripción:** “Solo se podrá reservar una actividad si el cupo de plazas máximo no ha sido superado. Si sumando a las personas que quieren reservar la actividad se llega a dicho máximo, se devuelve un error.”~~

~~“En el momento en que un cliente quiera eliminar la reserva total o parcial de plazas en una actividad, se indicará el número de plazas a eliminar, que deberá ser menor o igual a las plazas reservadas en un principio. Tras la eliminación, se actualizan las plazas asociadas a un DNI/Pasaporte. Así, se evita que el contador de plazas disponibles sea mayor al máximo de plazas de la actividad.”~~

**RS3.4:** La opinión solo puede ser realizada por clientes del hotel tras realizar la actividad.

**RF:** RF3.4

**RD(s):** RDE3.4, RDW3.4

**Descripción:** “Para evitar falsas opiniones de personas externas al hotel y la actividad, solo se dejará opinar a clientes del hotel que tenían una reserva de la actividad hecha. Para ello, se comprueba que su DNI/Pasaporte se encuentran en las reservas de las actividades de ocio. Si no es así, no se permite realizar la opinión.”

**RS3.5:** La puntuación de la opinión realizada sobre una actividad de ocio debe ser un número entero comprendido entre el 0 y el 10.

**RF:** RF3.4

**RD(s):** RDE3.4, RDW3.4

**Descripción:** “Cuando el cliente opine sobre una actividad de ocio, la puntuación otorgada por el cliente sobre dicha actividad no podrá ser ni un número negativo ni un número mayor que 10.”

## **Subsistema 4: Parking**

**Encargado: Juan Manuel Rodríguez Gómez**

**RS4.1:** El cliente paga el tiempo que ha ocupado la plaza de parking si dicho tiempo es mayor a 1 hora.

**RF:** RF4.4

**RD(s):** RDR4.4

**Descripción:** “A la hora de que el cliente vaya a pagar el tiempo que ha estado ocupando la plaza del parking, dicho pago se realizará si el tiempo es mayor a 1 hora. En otro caso, el cliente no tendrá que pagar nada”.

## **Subsistema 5: Mantenimiento de Instalaciones**

**Encargado: Jesús García León**

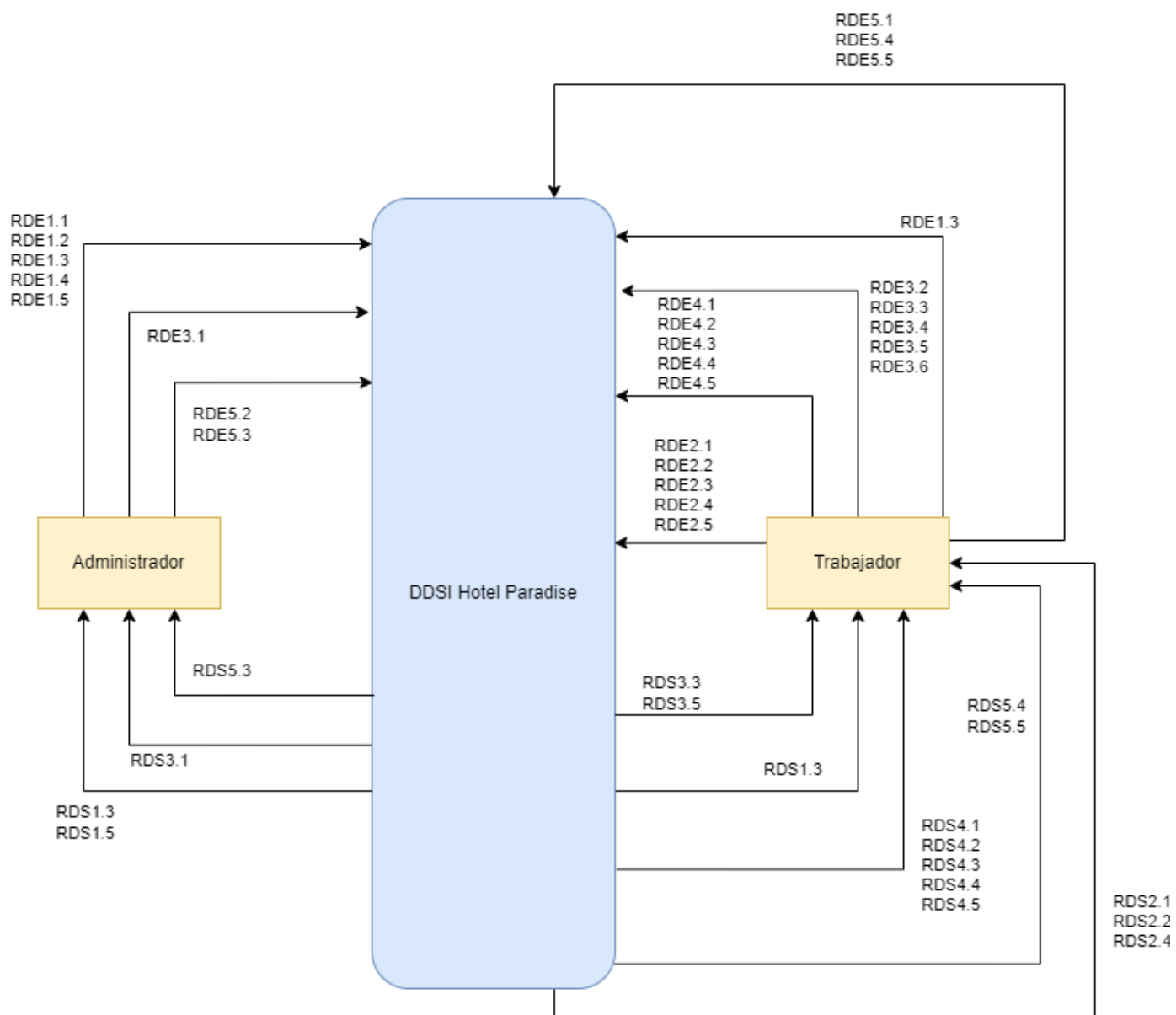
**RS5.1:** La suma de los diferentes precios asignados a reparaciones hasta la fecha no puede superar el presupuesto total disponible.

**RF:** RF5.2

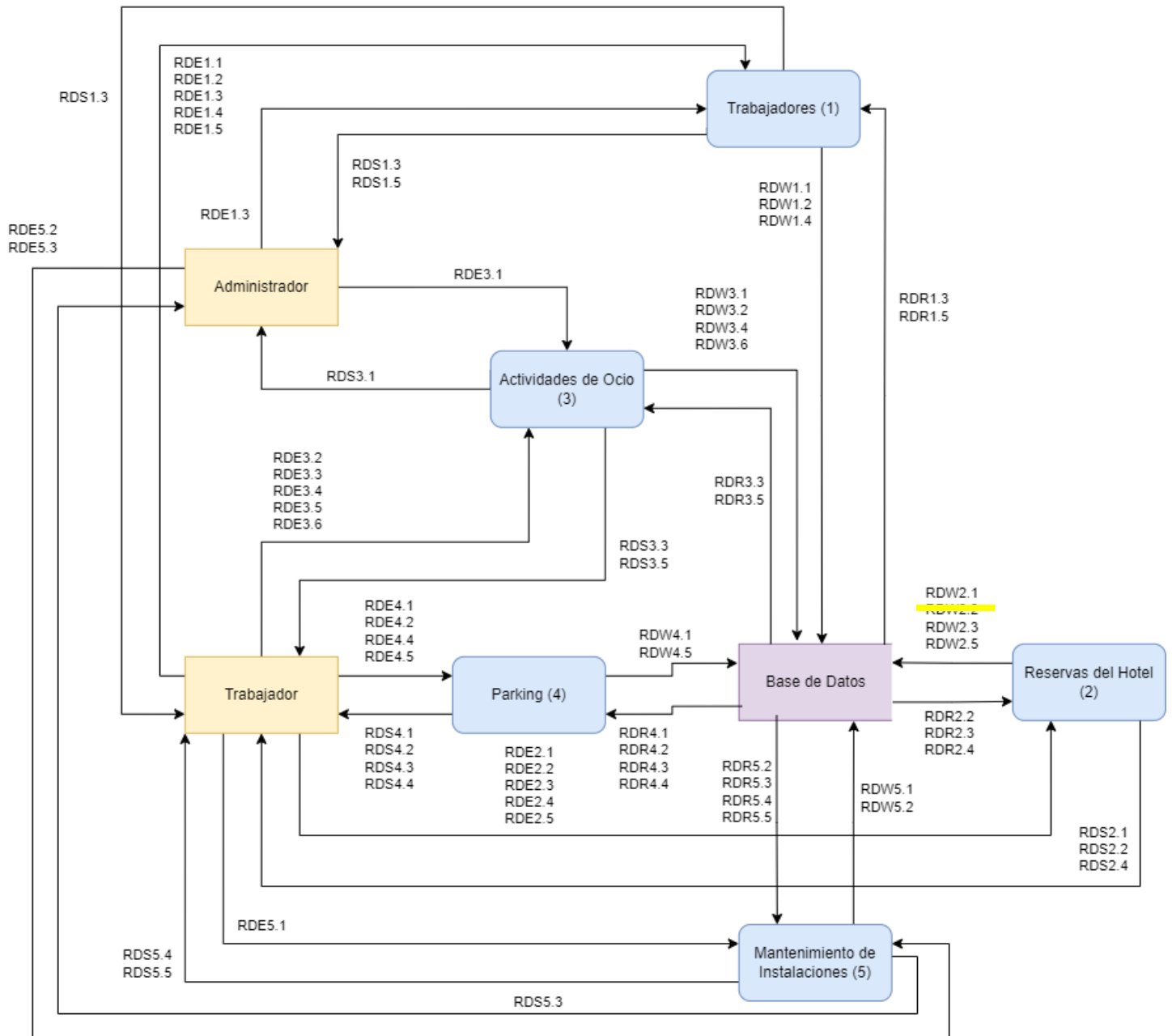
**RD(s):** RDW5.2

**Descripción:** “Si a la hora de asignar el precio a la reparación, esto hace que la suma total de todos los precios de las diferentes reparaciones de los desperfectos sea mayor que el presupuesto disponible (el cual es un número decimal fijo y conocido por el administrador), entonces el sistema devolverá error y no podrá asignarse dicho precio. En caso de no cumplirse lo anterior, el sistema asigna dicho precio a la reparación sin proporcionar ningún error.”

#### 4. DFD Esquema de Caja Negra



## 5. DFD Armazón (DFD0)

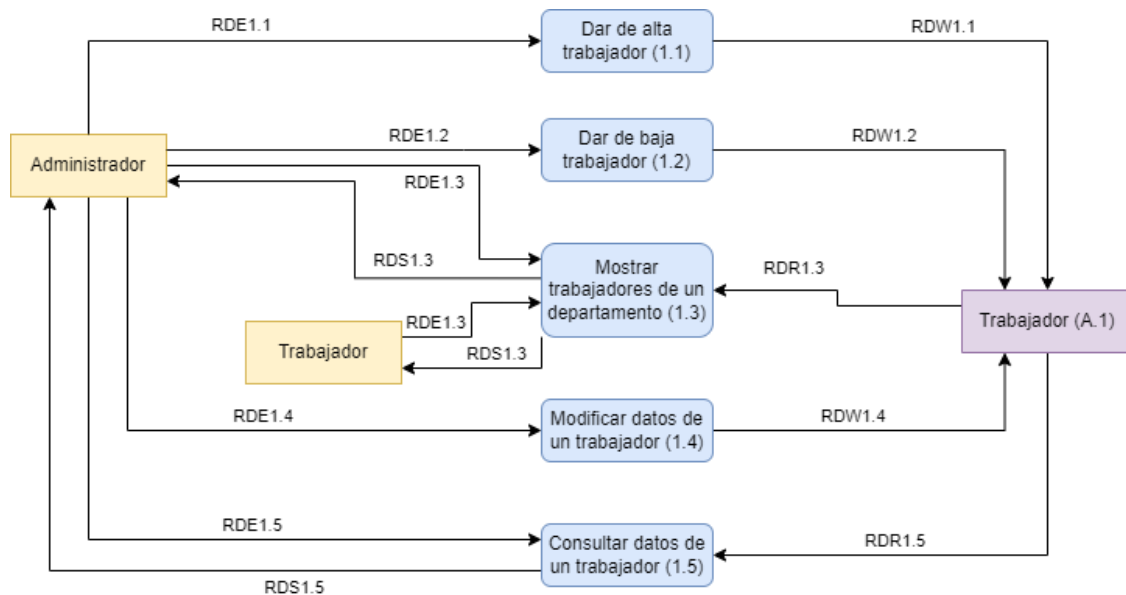




## 6. DFD Refinado de cada Subsistema (DFD1)

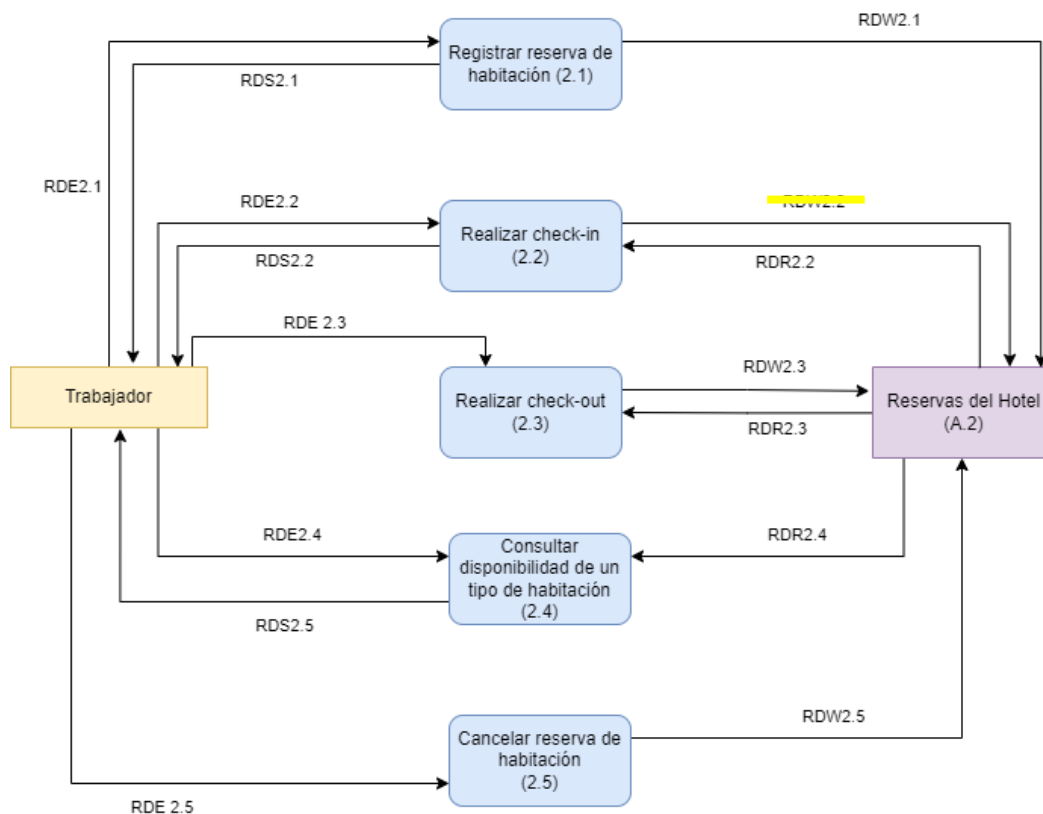
### Subsistema 1: Trabajadores

Encargada: Mónica Calzado Granados



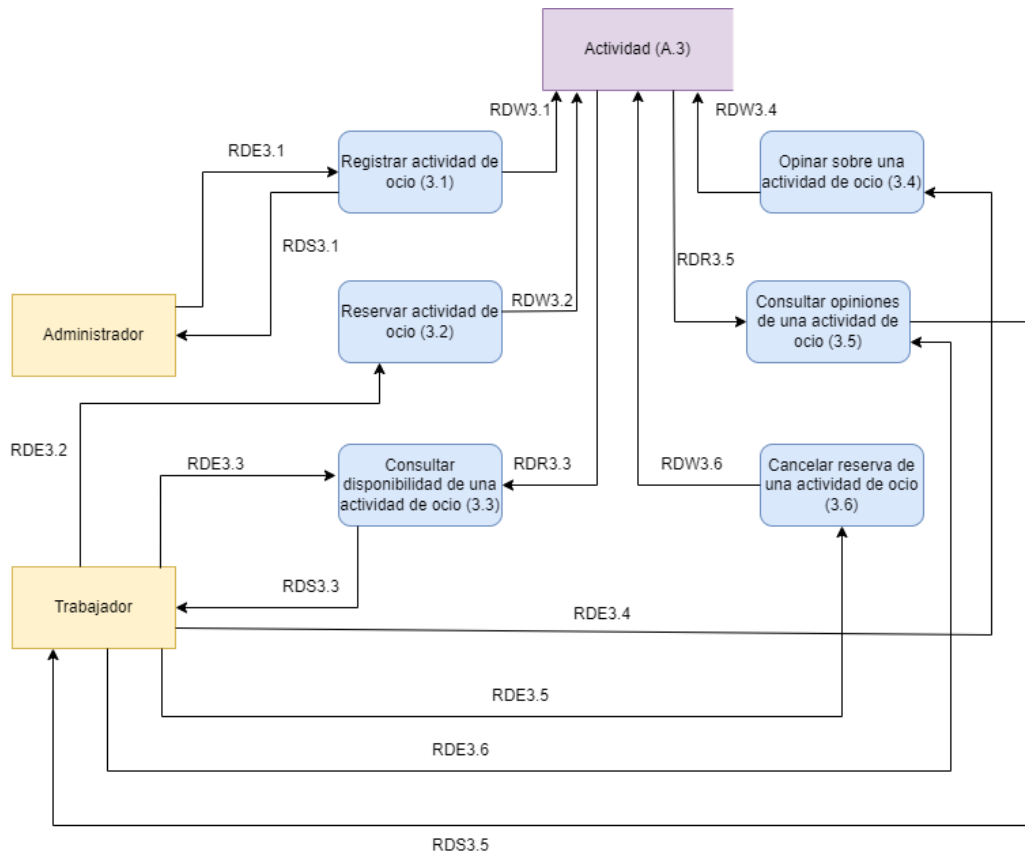
### Subsistema 2: Reservas del Hotel

Encargado: Alejandro Cárdenas Barranco



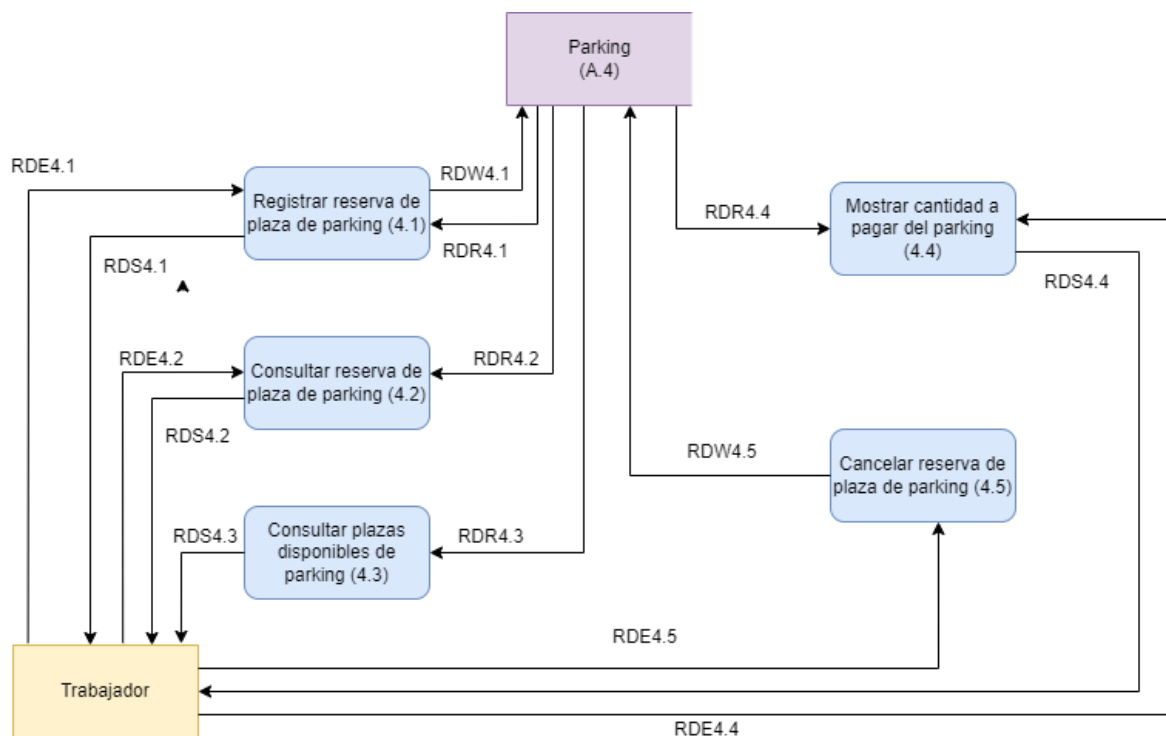
### Subsistema 3: Actividades de Ocio

Encargado: Álvaro Rodríguez Gallardo



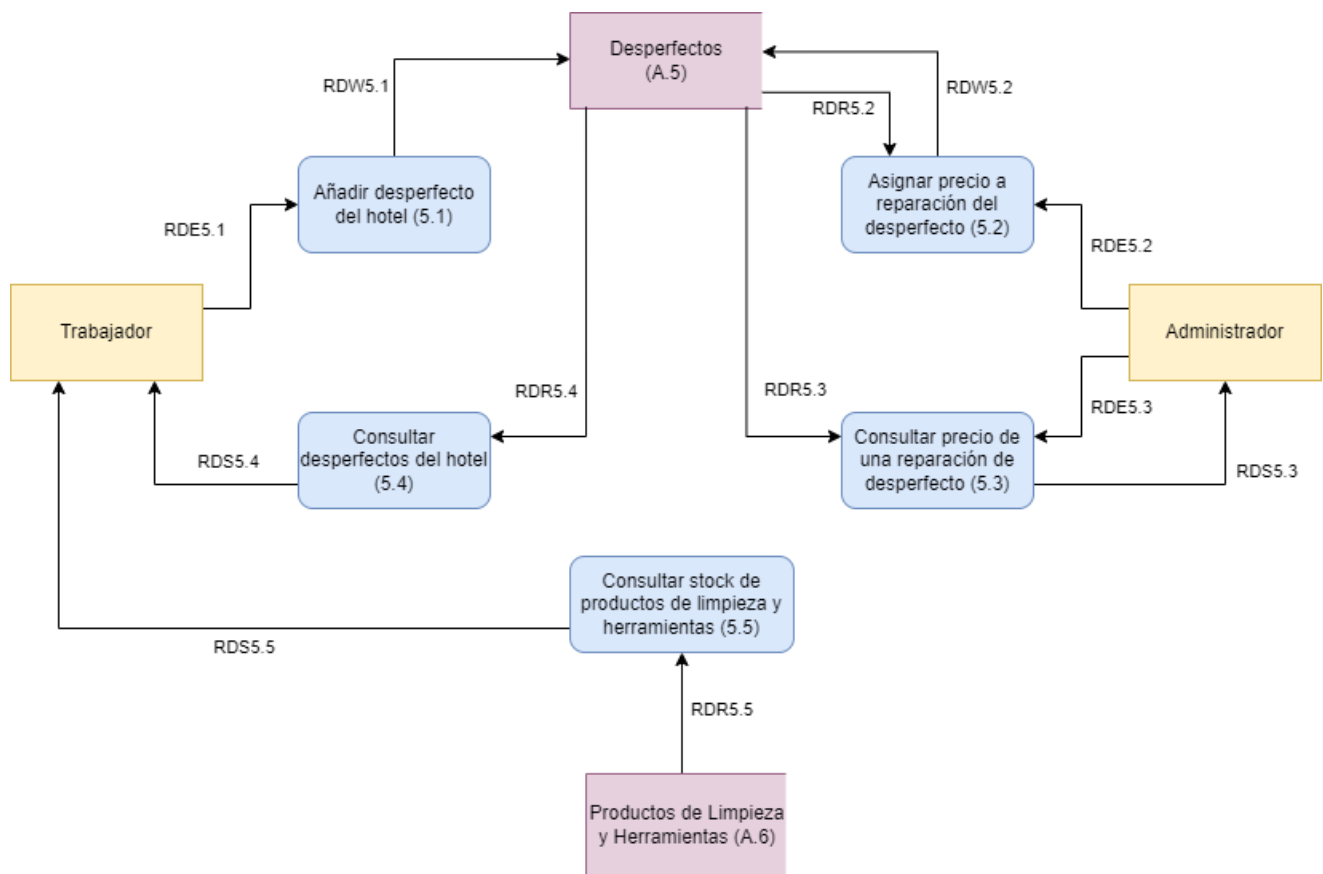
### Subsistema 4: Parking

Encargado: Juan Manuel Rodríguez Gómez



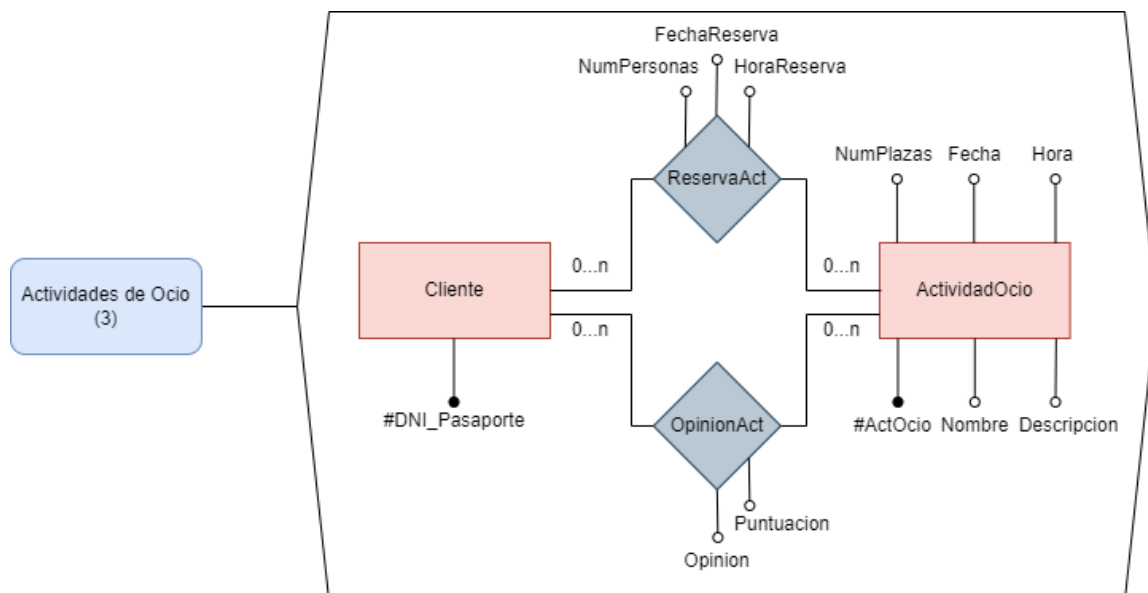
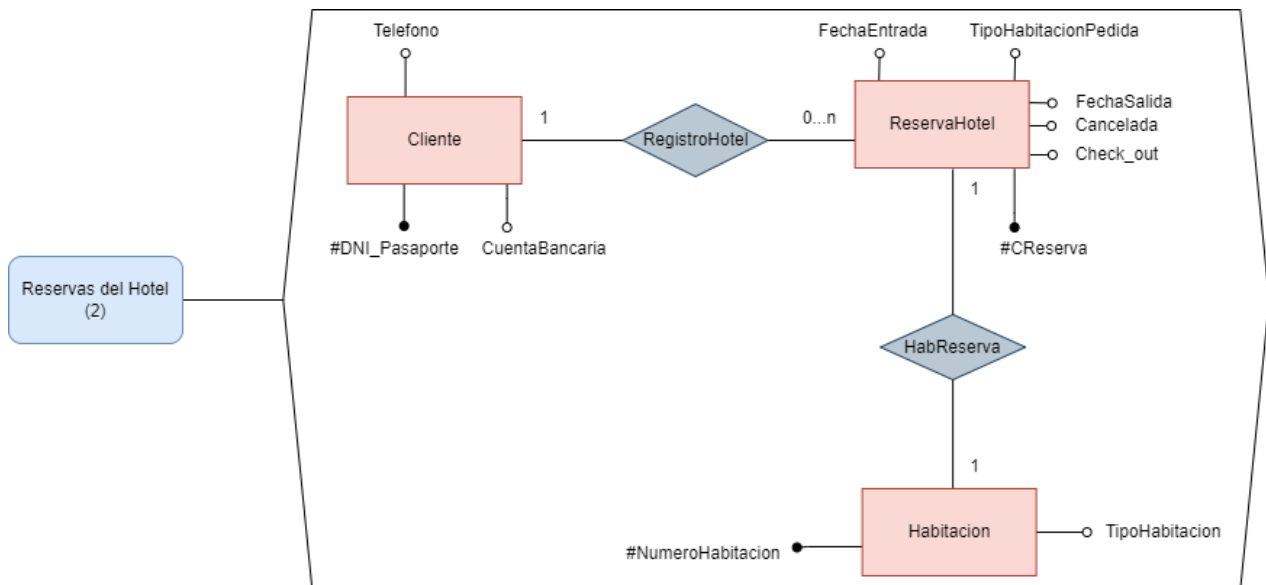
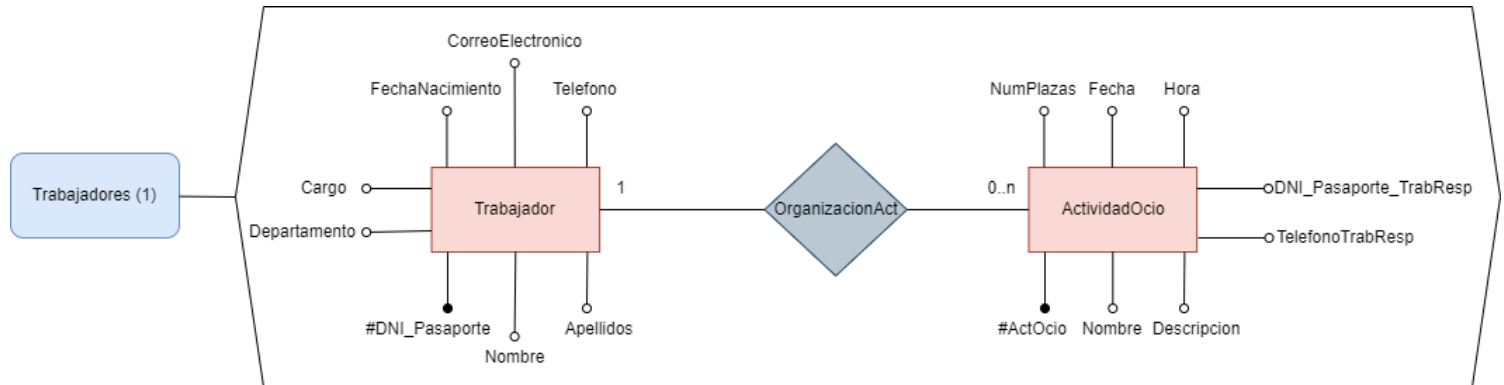
## Subsistema 5: Mantenimiento de Instalaciones

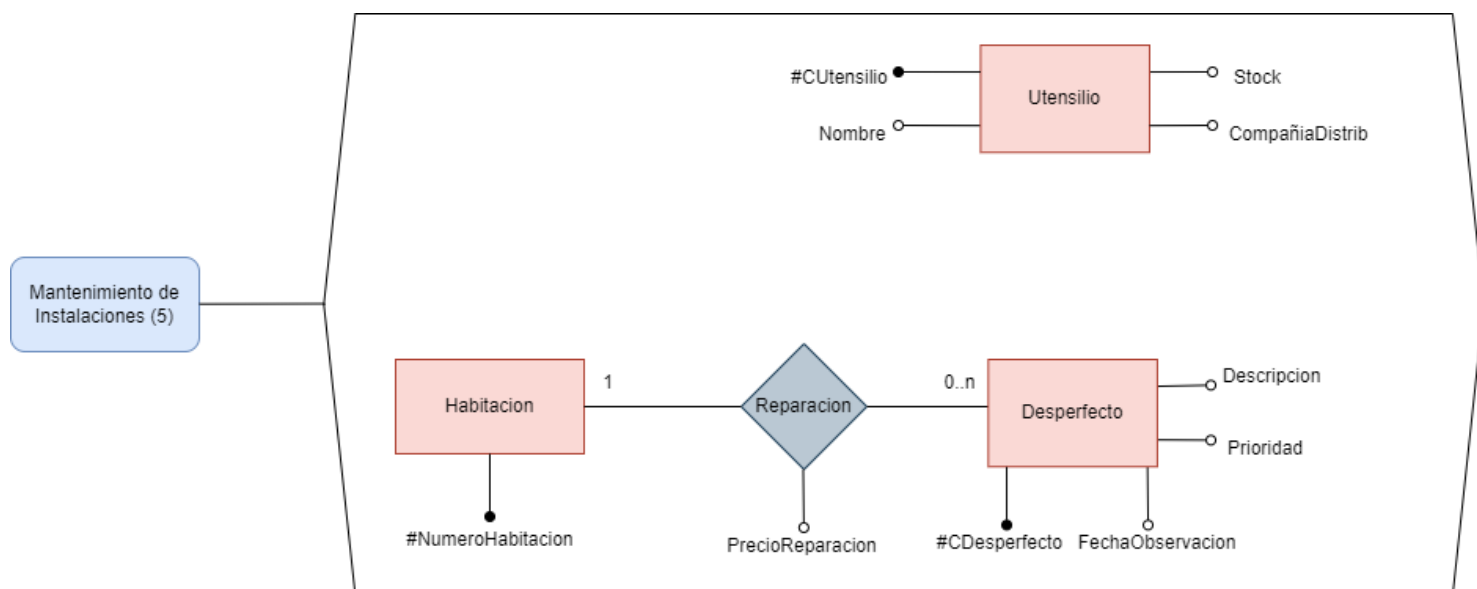
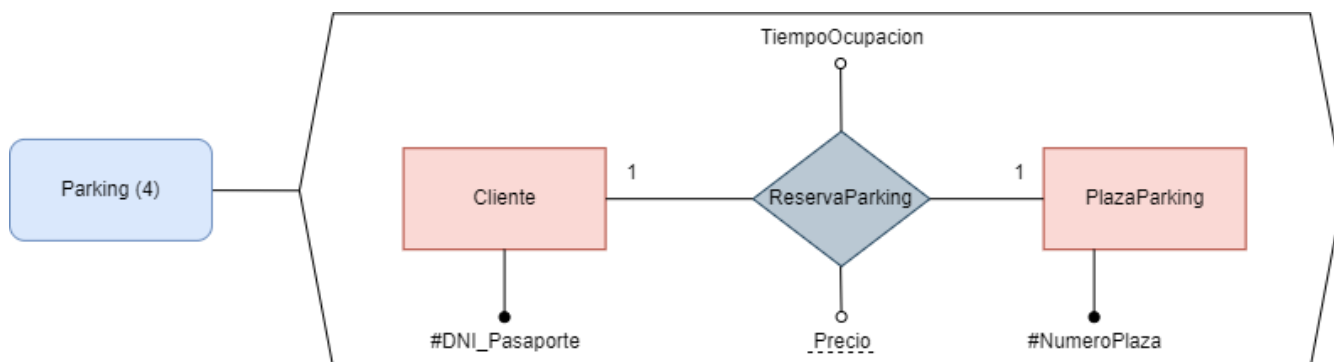
Encargado: Jesús García León



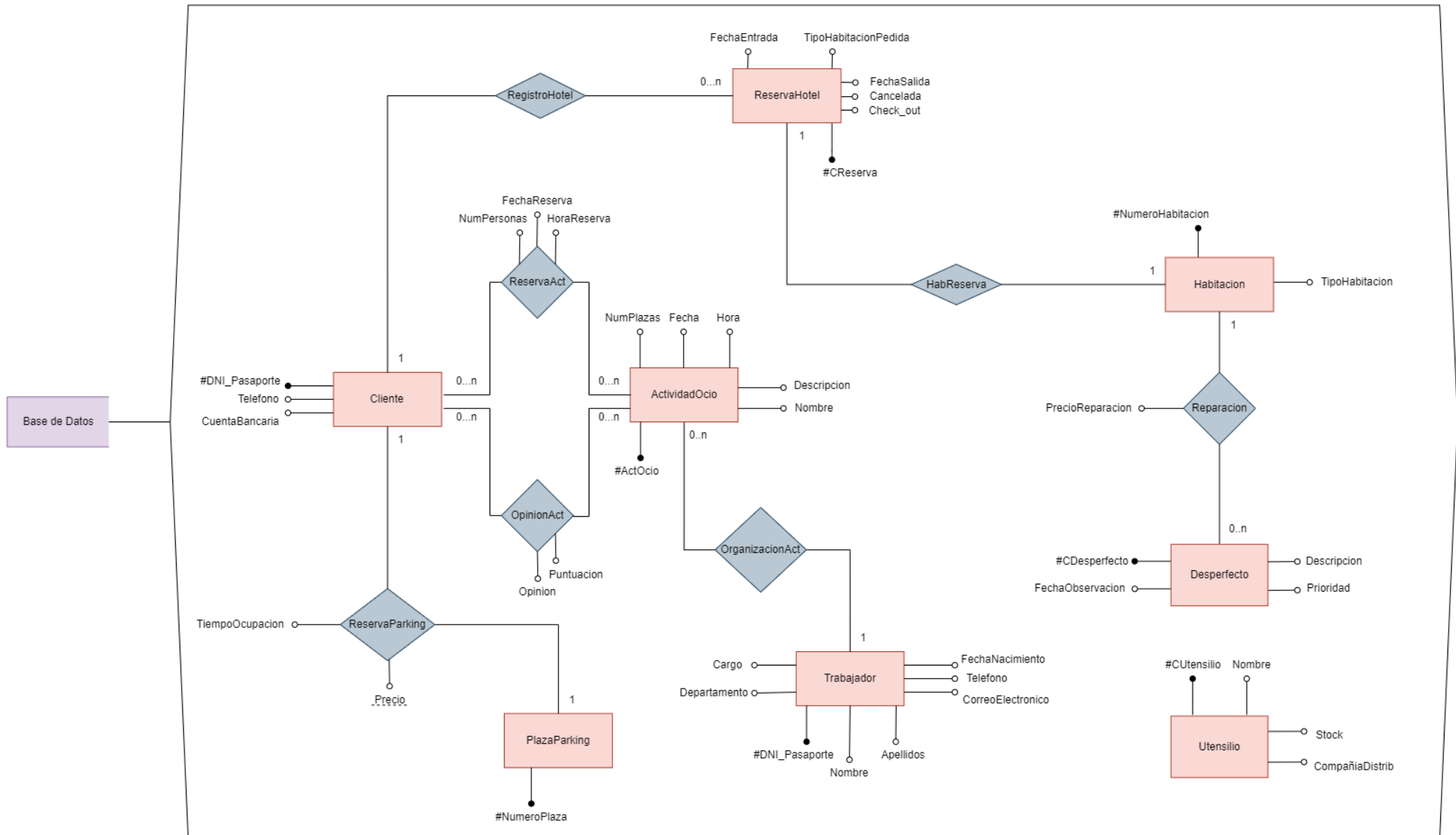
## 7. Esquemas Externos de DFD0

### Procesos





## Almacén

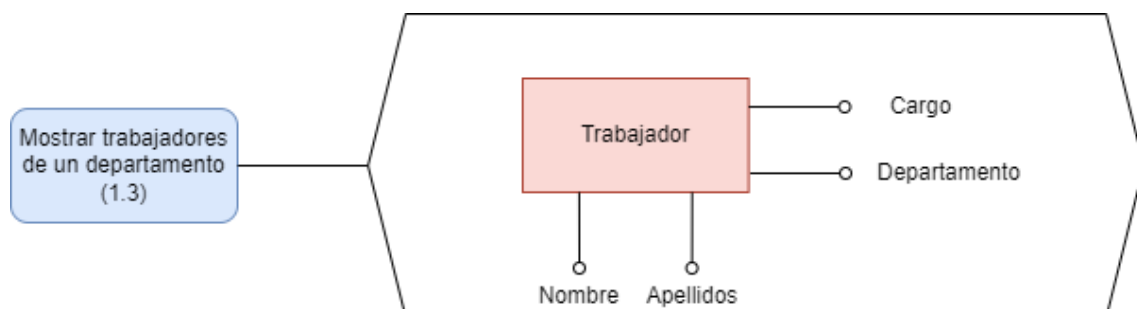
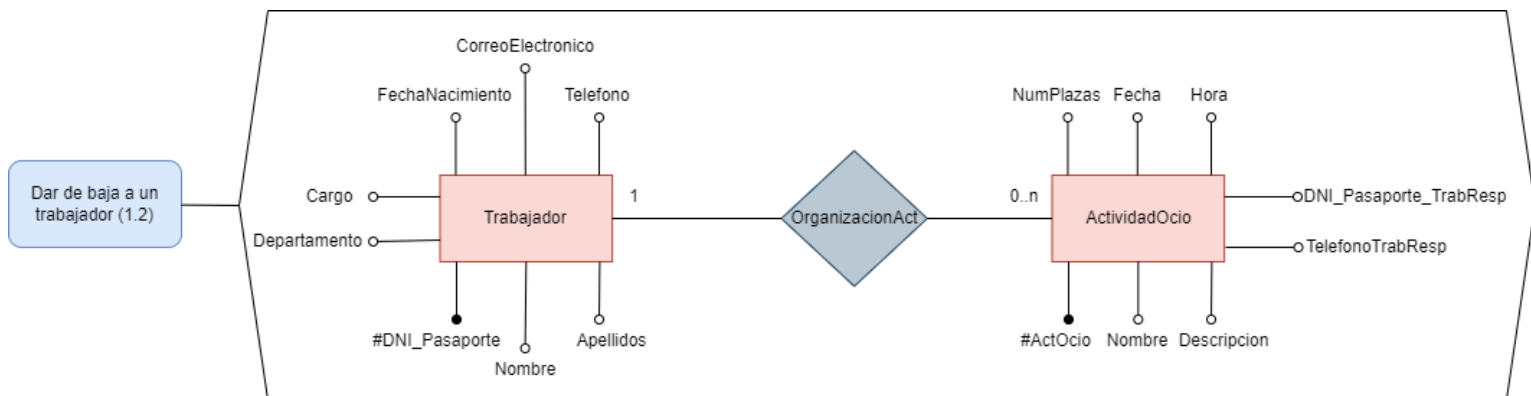
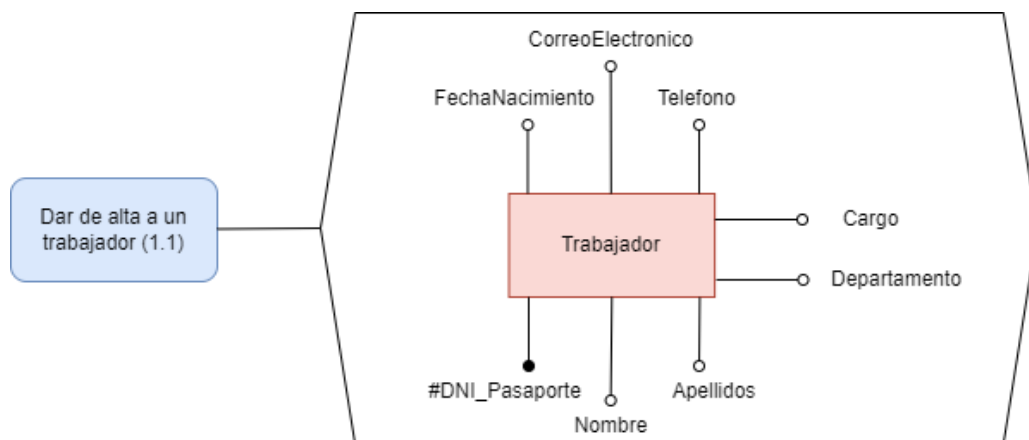


## 8. Esquemas Externos de DFD1

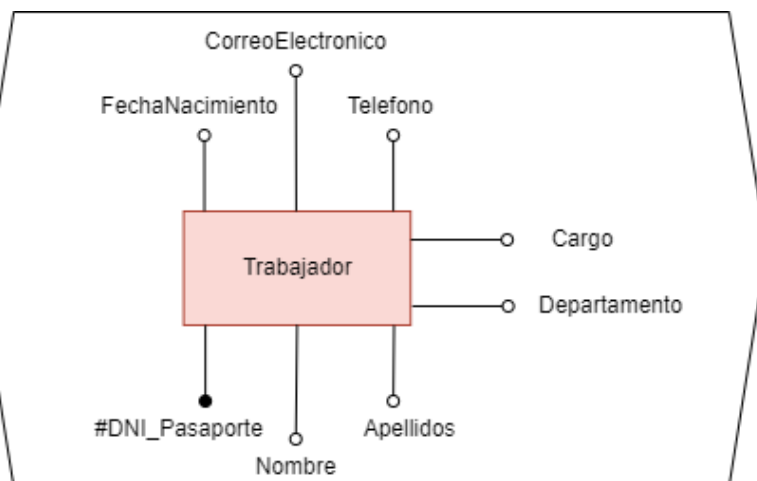
### Subsistema 1: Trabajadores

Encargada: Mónica Calzado Granados

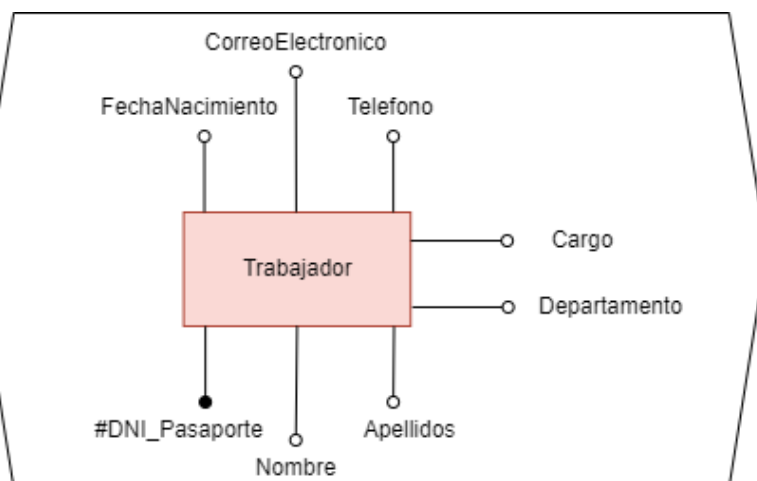
#### Procesos



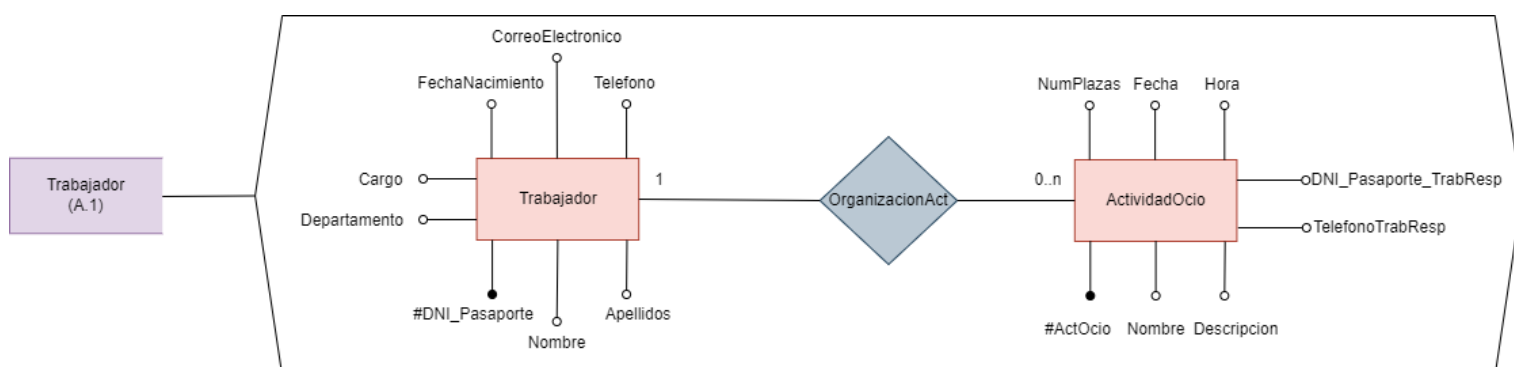
Modificar datos de un trabajador (1.4)



Consultar datos de un trabajador (1.5)



## Almacén

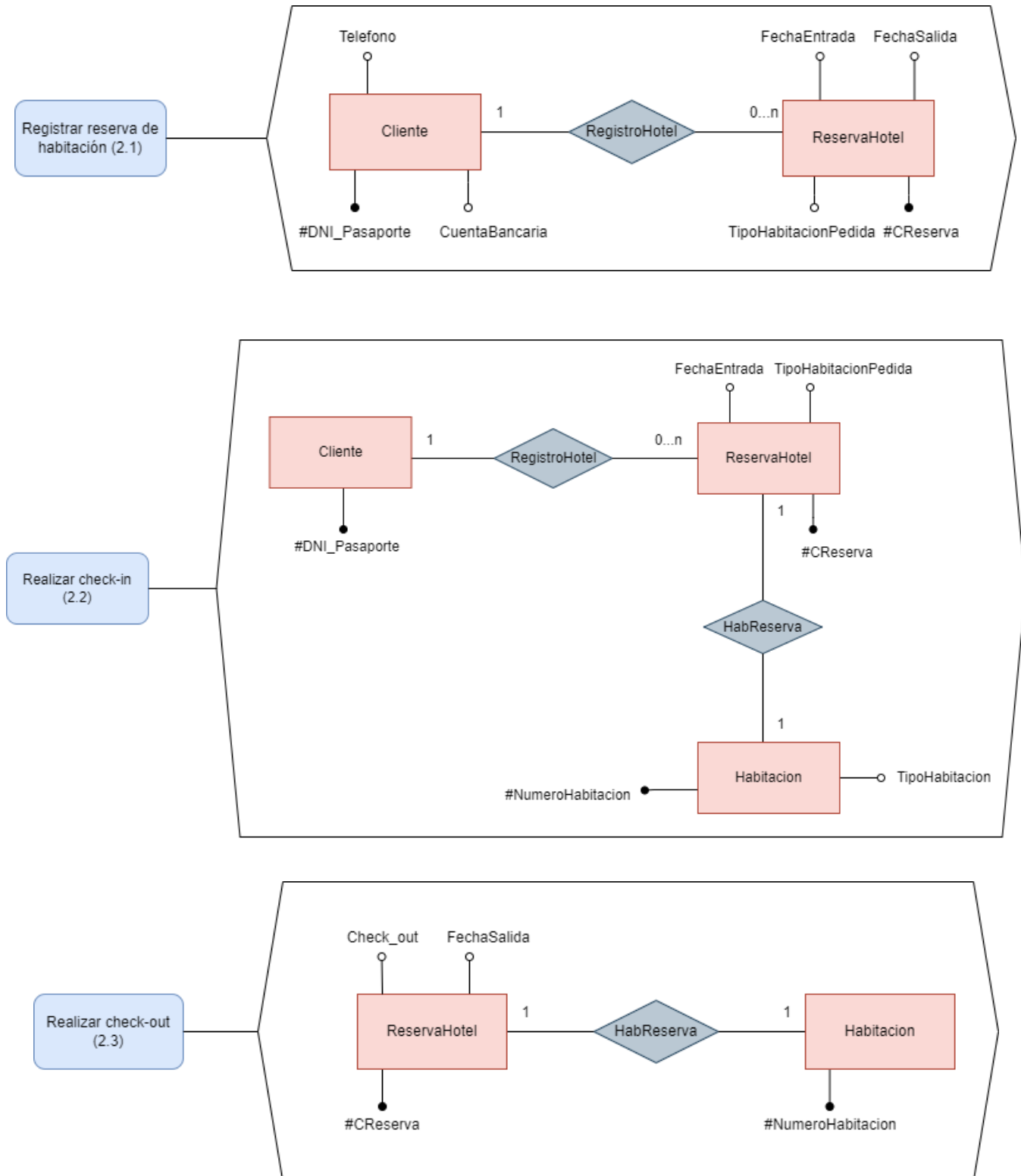


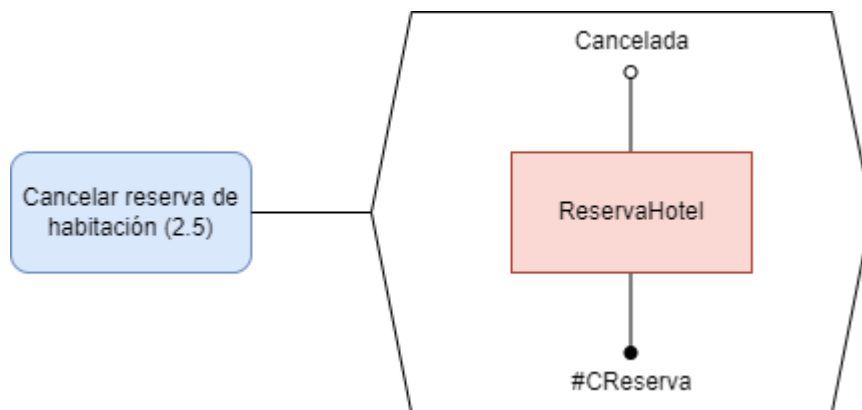
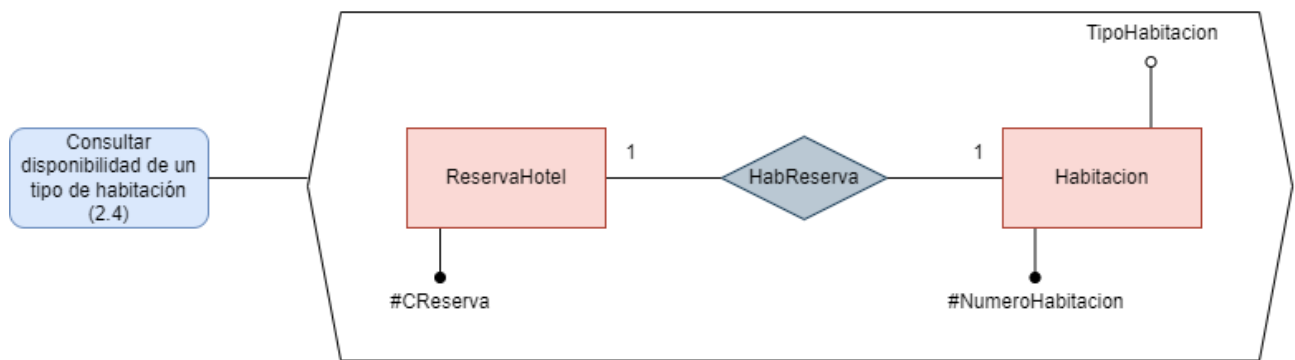


## Subsistema 2: Reservas del Hotel

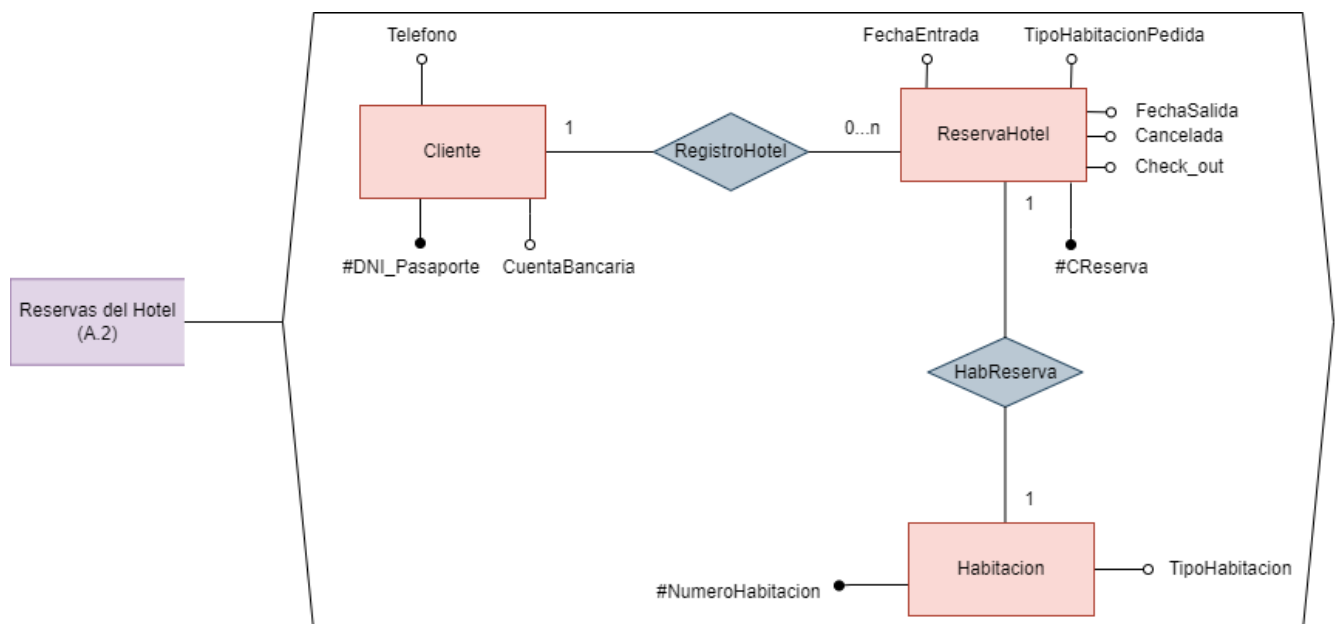
Encargado: Alejandro Cárdenas Barranco

### Procesos



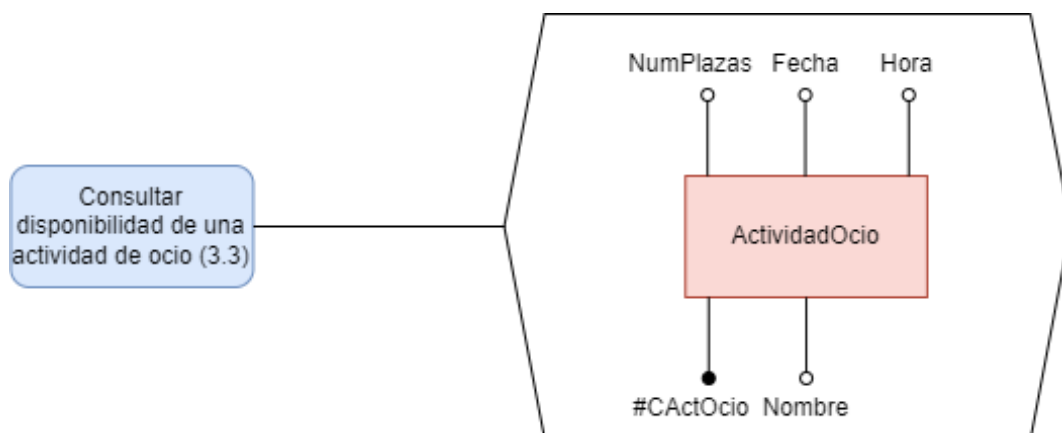
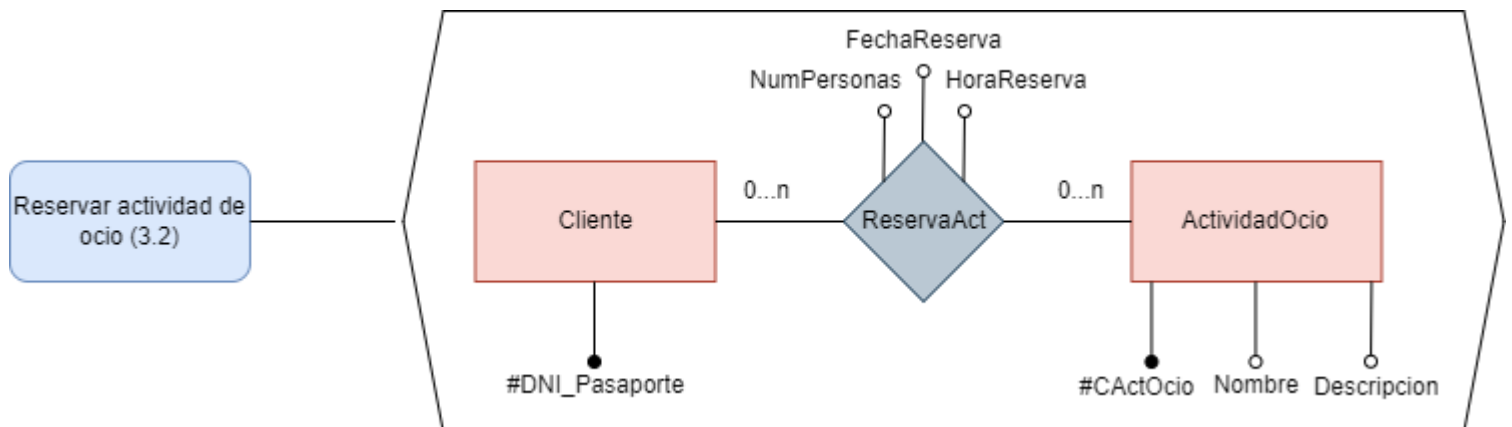
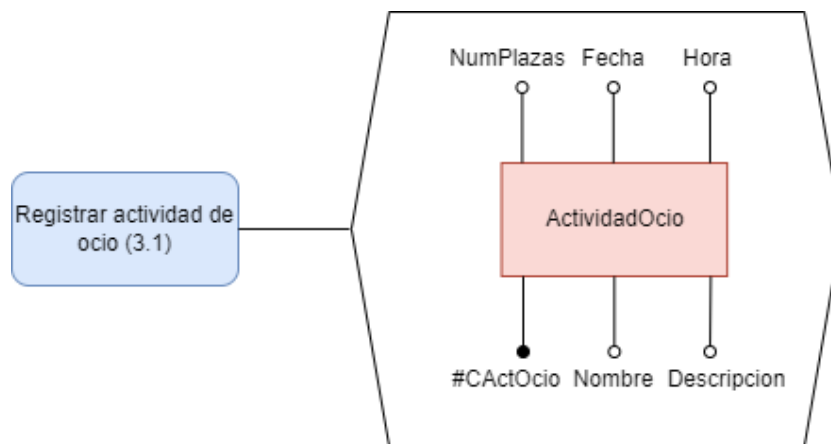


## Almacén

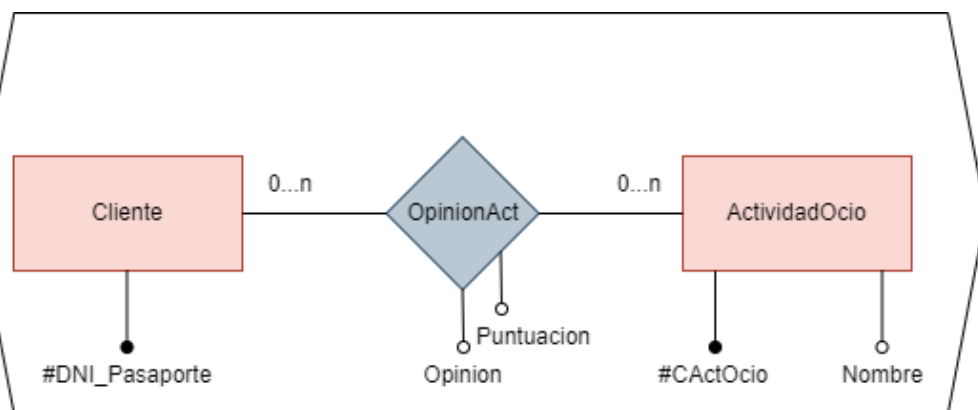


**Subsistema 3: Actividades de Ocio**  
**Encargado: Álvaro Rodríguez Gallardo**

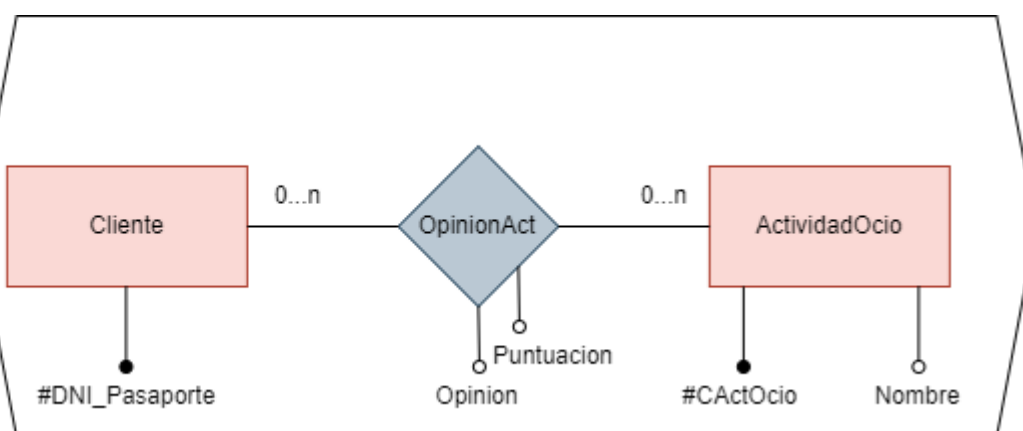
**Procesos**



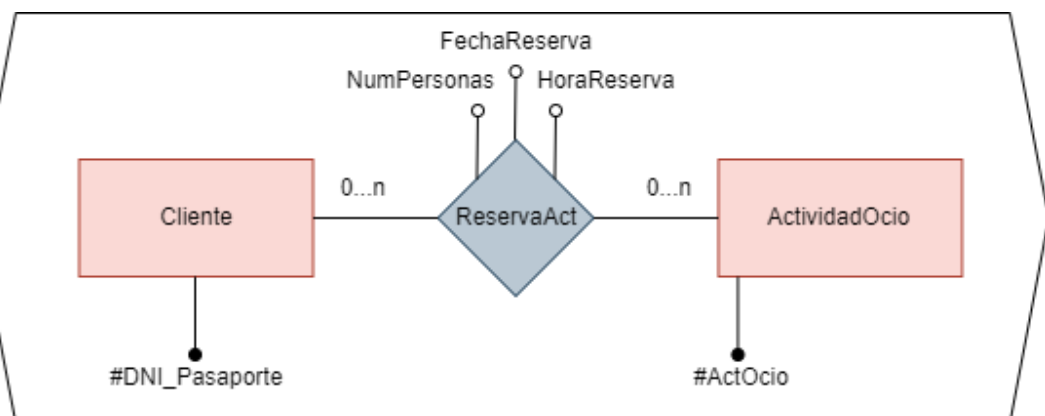
Opinar sobre una actividad de ocio (3.4)



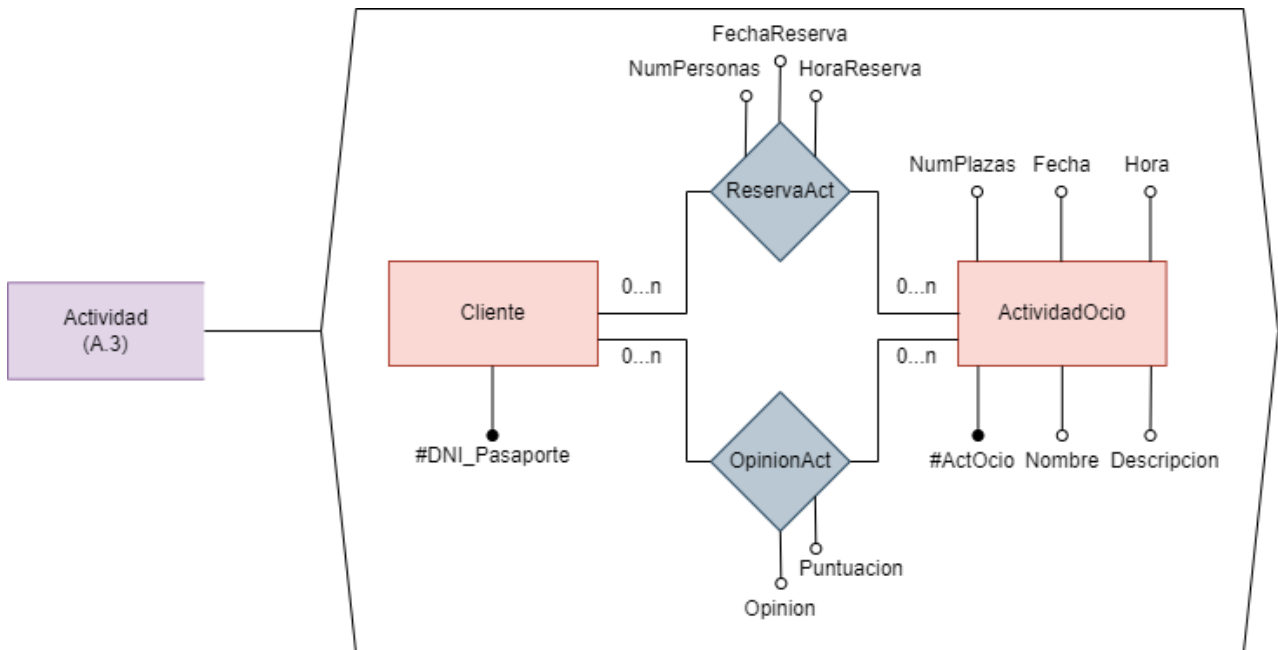
Consultar opiniones de una actividad de ocio (3.5)



Cancelar reserva de una actividad de ocio (3.6)



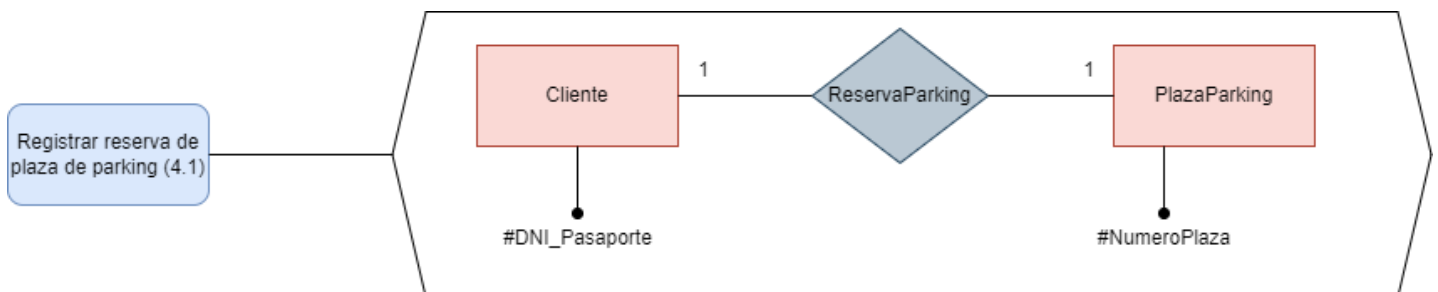
## Almacén



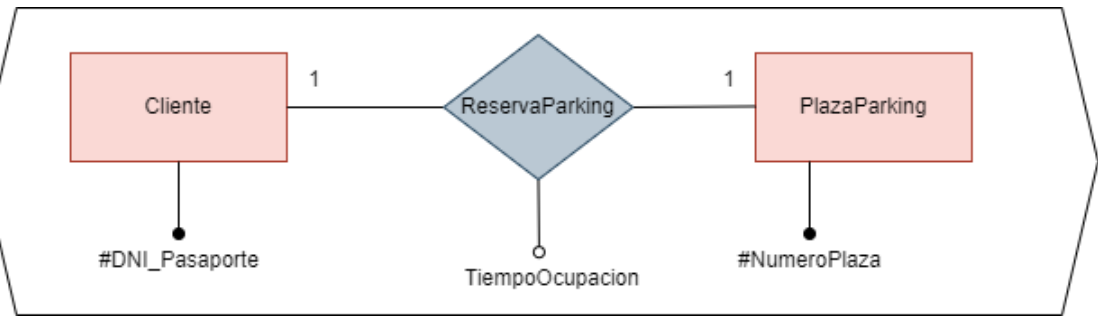
## Subsistema 4: Parking

Encargado: Juan Manuel Rodríguez Gómez

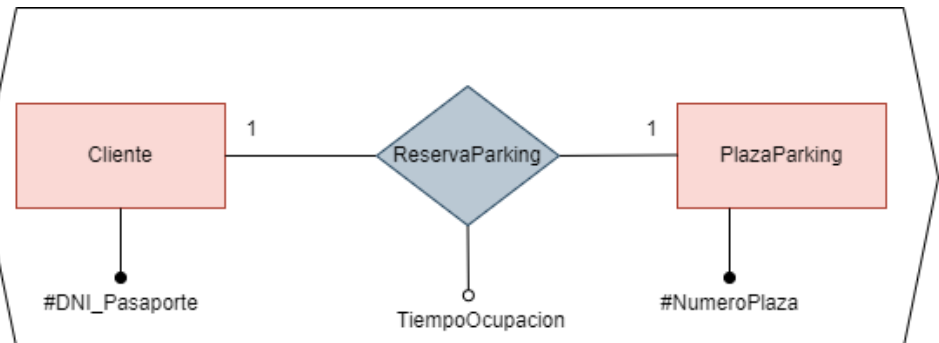
## Procesos



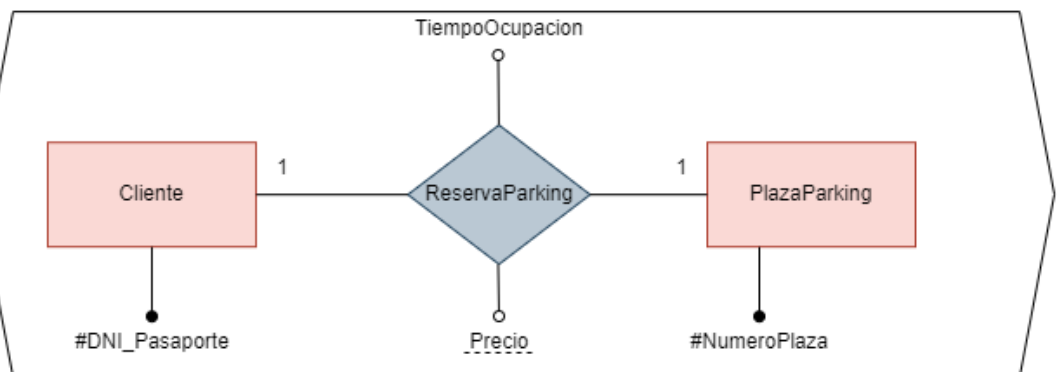
Consultar reserva de plaza de parking (4.2)



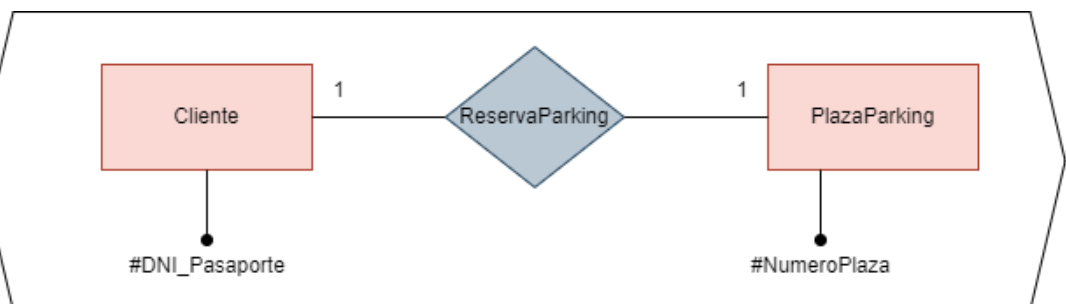
Consultar plazas disponibles de parking (4.3)



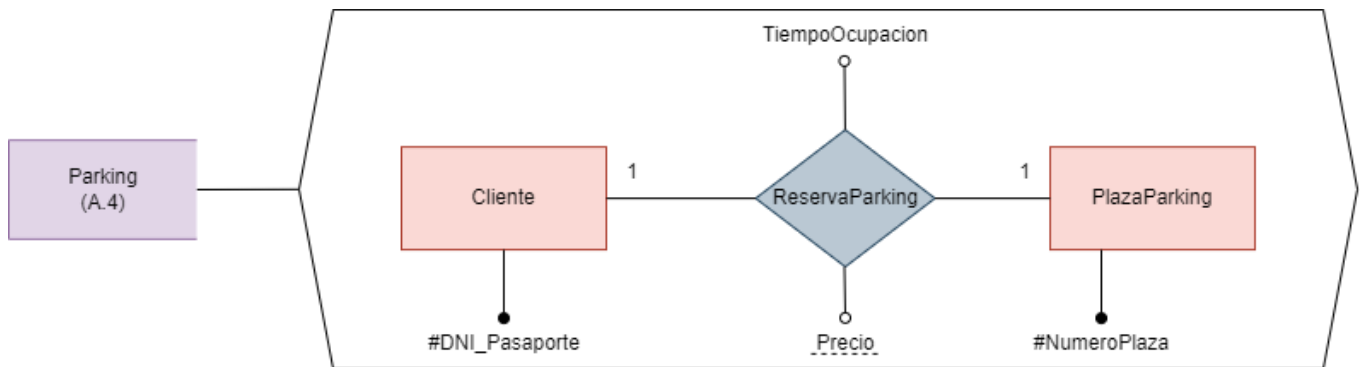
Mostrar cantidad a pagar del parking (4.4)



Cancelar reserva de plaza de parking (4.5)



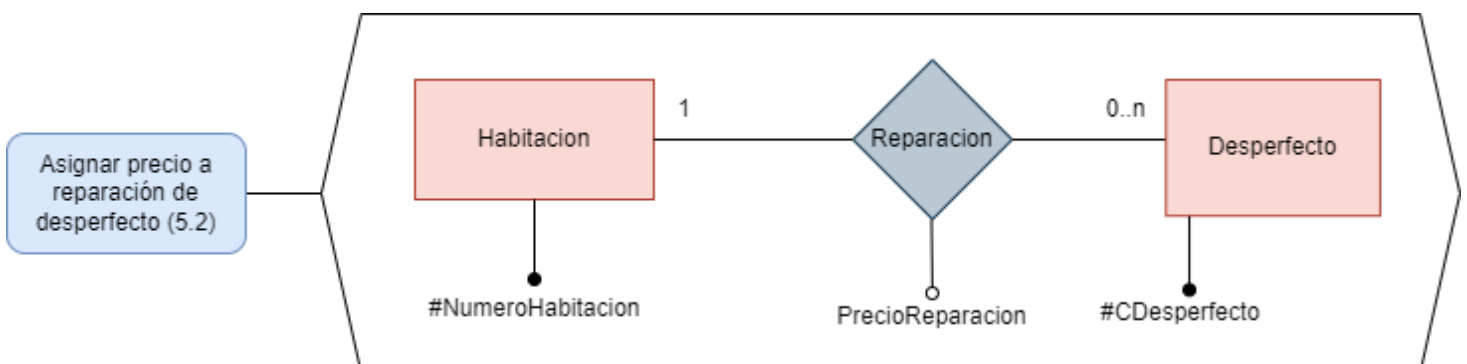
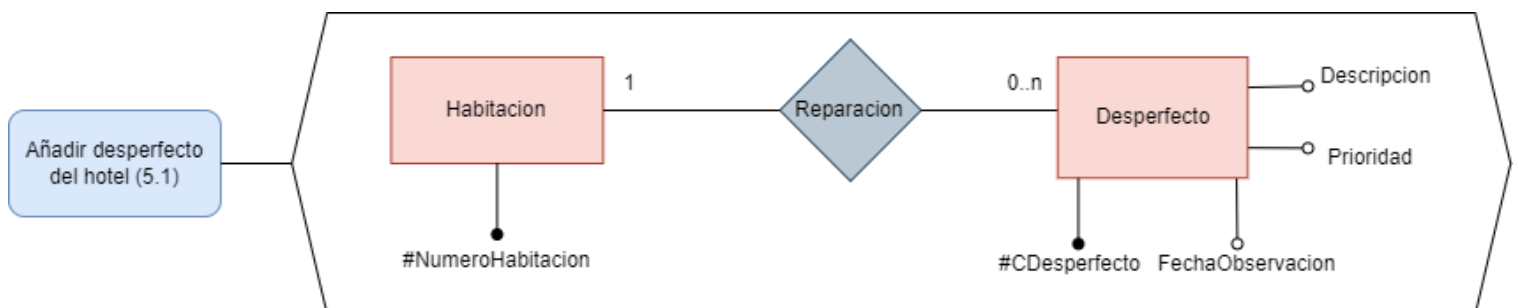
## Almacén



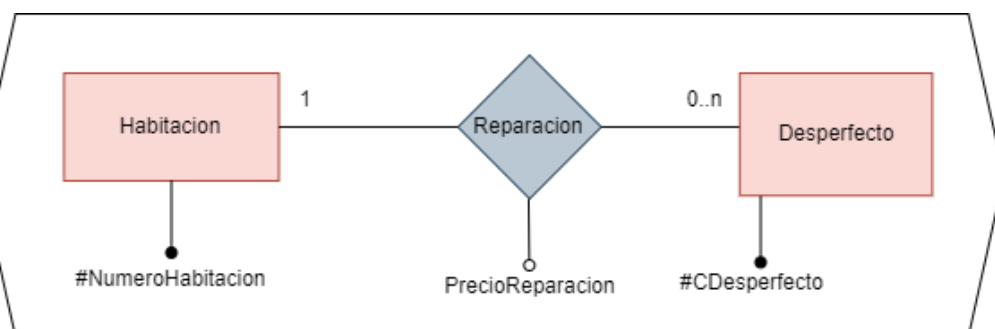
## Subsistema 5: Mantenimiento de Instalaciones

Encargado: Jesús García León

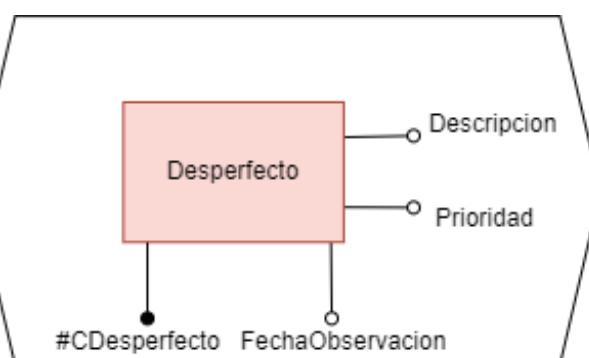
## Procesos



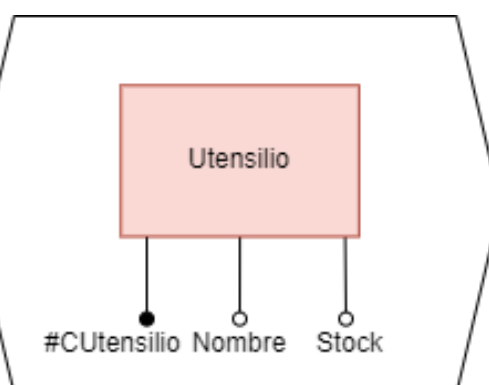
Consultar precio de  
reparación de  
desperfecto (5.3)



Consultar  
desperfectos del hotel  
(5.4)

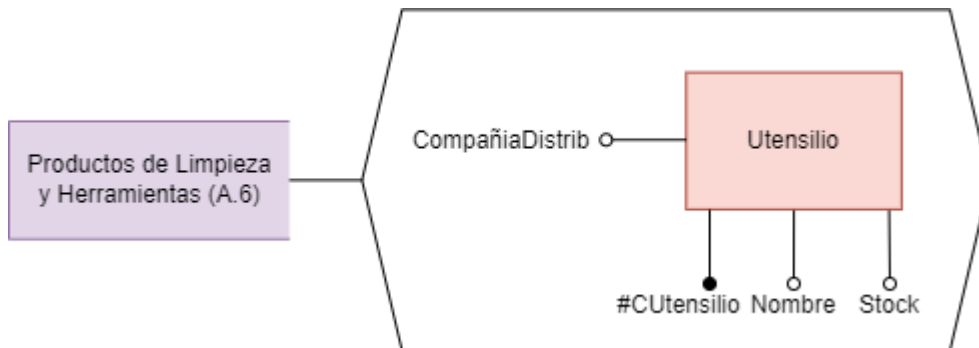
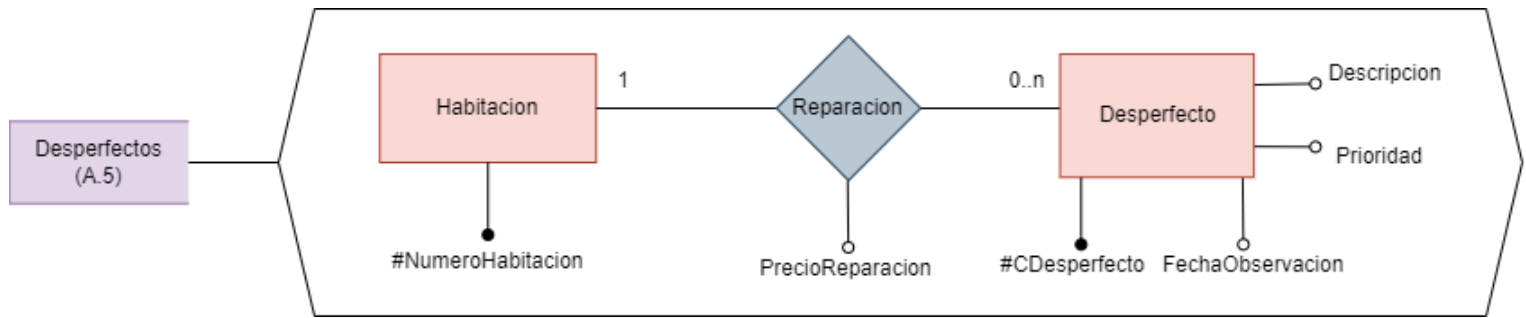


Consultar stock de  
producto de limpieza  
y herramientas (5.5)

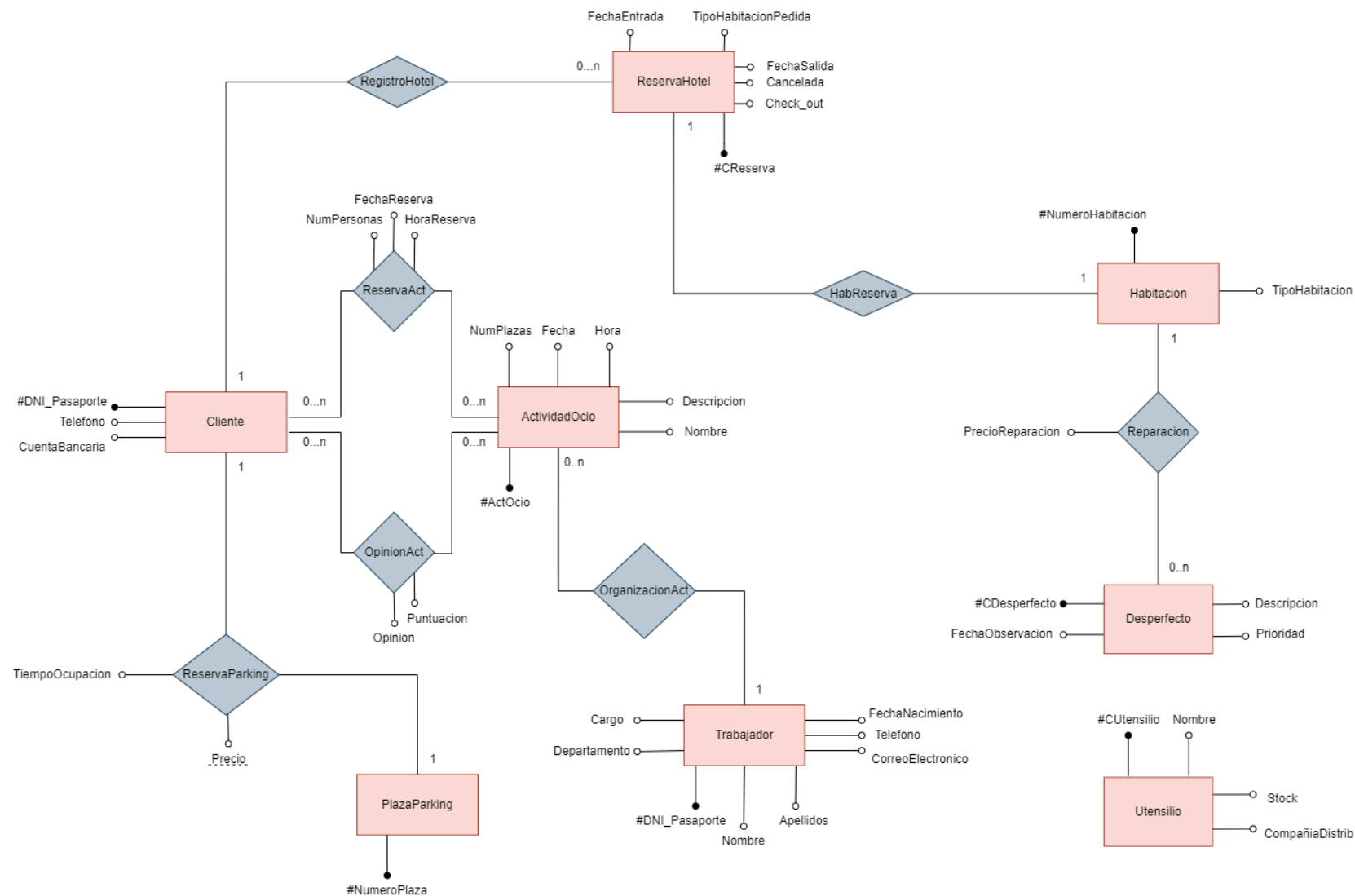




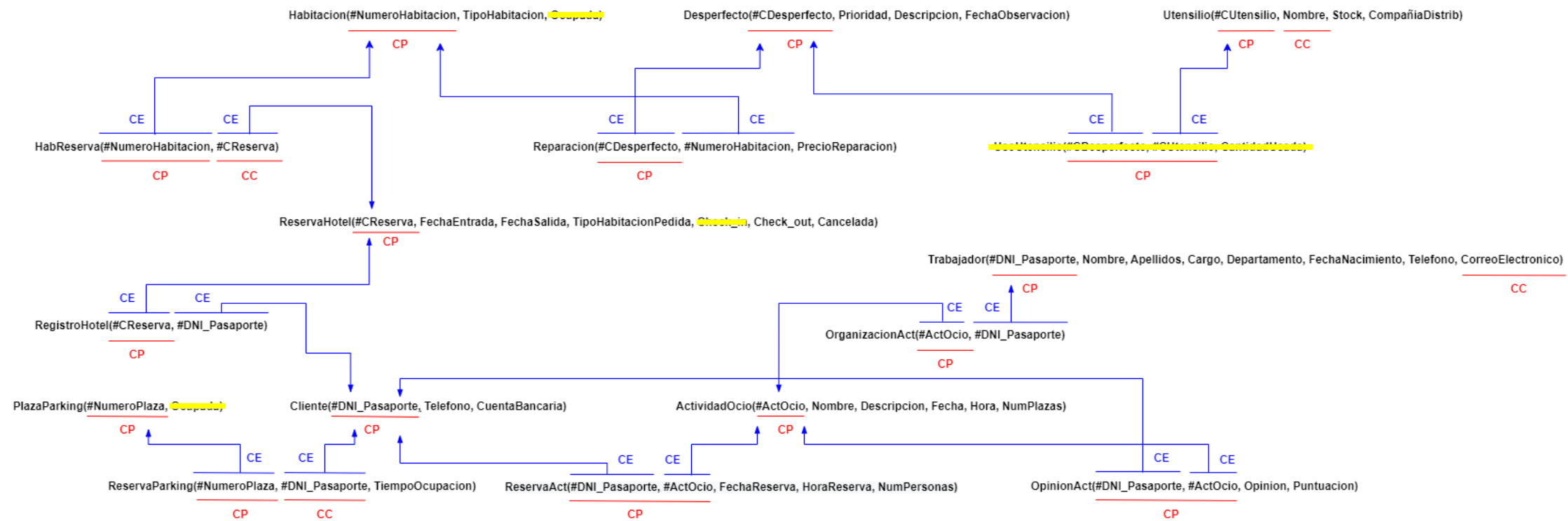
## Almacenes



## 9. Esquema E/R Completo del Sistema



## 10. Tablas Obtenidas a partir del Esquema E/R



## 11. Dependencias Funcionales y Normalización de las Tablas

- Tabla 1:

***Trabajador(#DNI\_Pasaporte, Nombre, Apellidos, Cargo, Departamento, FechaNacimiento, Telefono, CorreoElectronico)***

- **Claves Candidatas:** {#DNI\_Pasaporte, CorreoElectronico}
- **Dependencias Funcionales:**
  - {#DNI\_Pasaporte → Nombre}
  - {#DNI\_Pasaporte → Apellidos}
  - {#DNI\_Pasaporte → Cargo}
  - {#DNI\_Pasaporte → Departamento}
  - {#DNI\_Pasaporte → FechaNacimiento}
  - {#DNI\_Pasaporte → Telefono}
  - {#DNI\_Pasaporte → CorreoElectronico}
  - {CorreoElectronico → #DNI\_Pasaporte}
  - {CorreoElectronico → Nombre}
  - {CorreoElectronico → Apellidos}
  - {CorreoElectronico → Cargo}
  - {CorreoElectronico → Departamento}
  - {CorreoElectronico → FechaNacimiento}
  - {CorreoElectronico → Telefono}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#DNI\_Pasaporte, que es la clave primaria, y CorreoElectronico que también es superclave), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.
- **Aclaración:** No consideramos Telefono como clave candidata ya que, por ejemplo, un mismo matrimonio pueden trabajar los dos en el hotel y ambos ponen como teléfono de contacto el de su vivienda.

- **Tabla 2:**

***ActividadOcio(#ActOcio, Nombre, Descripcion, Fecha, Hora, NumPlazas)***

- **Claves Candidatas:** {#ActOcio}
- **Dependencias Funcionales:**
  - {#ActOcio → Nombre}
  - {#ActOcio → Descripcion}
  - {#ActOcio → Fecha}
  - {#ActOcio → Hora}
  - {#ActOcio → NumPlazas}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#ActOcio), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 3:**

***OrganizacionAct(#ActOcio, #DNI\_Pasaporte)***

- **Claves Candidatas:** {#ActOcio}
- **Dependencias Funcionales:**
  - {#ActOcio → #DNI\_Pasaporte}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#ActOcio), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 4:**

***Cliente(#DNI Pasaporte, Telefono, CuentaBancaria)***

- **Claves Candidatas:** {#DNI\_Pasaporte}
- **Dependencias Funcionales:**
  - {#DNI\_Pasaporte → Telefono}
  - {#DNI\_Pasaporte → CuentaBancaria}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#DNI\_Pasaporte), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.
- **Aclaración:** No consideramos *CuentaBancaria* como clave candidata ya que, por ejemplo, podría ocurrir que viniera un matrimonio en el que una persona hace una reserva con su DNI, su teléfono y una cuenta bancaria compartida entre los dos y luego, la otra persona en el mismo momento hace otra reserva a su nombre con su DNI, su teléfono y dicha cuenta bancaria compartida. Por el mismo motivo, tampoco consideramos *Telefono* como clave candidata (puede ocurrir que una persona del matrimonio ponga su teléfono en la reserva y la otra persona ponga el mismo teléfono).

- **Tabla 5:**

***ReservaAct(#DNI\_Pasaporte, #ActOcio, FechaReserva, HoraReserva, NumPersonas)***

- **Claves Candidatas:** { {#DNI\_Pasaporte, #ActOcio} }
- **Dependencias Funcionales:**
  - { {#DNI\_Pasaporte, #ActOcio} → FechaReserva }
  - { {#DNI\_Pasaporte, #ActOcio} → HoraReserva }
  - { {#DNI\_Pasaporte, #ActOcio} → NumPersonas }
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave ( {#DNI\_Pasaporte, #ActOcio} ), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 6:**

***OpinionAct(#DNI\_Pasaporte, #ActOcio, Opinion, Puntuacion)***

- **Claves Candidatas:** { {#DNI\_Pasaporte, #ActOcio} }
- **Dependencias Funcionales:**
  - { {#DNI\_Pasaporte, #ActOcio} → Opinion }
  - { {#DNI\_Pasaporte, #ActOcio} → Puntuacion }
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave ( {#DNI\_Pasaporte, #ActOcio} ), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 7:**

***PlazaParking(#NumeroPlaza, Ocupada)***

- **Claves Candidatas:** {#NumeroPlaza}
- **Dependencias Funcionales:**
  - ~~{#NumeroPlaza → Ocupada}~~
- **Normalización:** ~~Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#NumeroPlaza), luego, No se tiene ninguna dependencia funcional, luego, es trivial que ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.~~

- **Tabla 8:**

***ReservaParking(#NumeroPlaza, #DNI\_Pasaporte, TiempoOcupacion)***

- **Claves Candidatas:** {#NumeroPlaza, #DNI\_Pasaporte}
- **Dependencias Funcionales:**
  - {#NumeroPlaza → #DNI\_Pasaporte}
  - {#NumeroPlaza → TiempoOcupacion}
  - {#DNI\_Pasaporte → #NumeroPlaza}
  - {#DNI\_Pasaporte → TiempoOcupacion}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#NumeroPlaza, que es la clave primaria, y #DNI\_Pasaporte que también es superclave), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.



- **Tabla 9:**

***ReservaHotel(#CReserva, FechaEntrada, FechaSalida, TipoHabitacionPedida, Check\_in, Check\_out, Cancelada)***

- **Claves Candidatas:** {#CReserva}
- **Dependencias Funcionales:**
  - {#CReserva → FechaEntrada}
  - {#CReserva → FechaSalida}
  - {#CReserva → TipoHabitacionPedida}
  - ~~{#CReserva → Check\_in}~~
  - {#CReserva → Check\_out}
  - {#CReserva → Cancelada}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#CReserva), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 10:**

***RegistroHotel(#CReserva, #DNI\_Pasaporte)***

- **Claves Candidatas:** {#CReserva}
- **Dependencias Funcionales:**
  - {#CReserva → #DNI\_Pasaporte}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#CReserva), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 11:**

***Habitacion(#NumeroHabitacion, TipoHabitacion, Ocupada)***

- **Claves Candidatas:** {#NumeroHabitacion}
- **Dependencias Funcionales:**
  - {#NumeroHabitacion → TipoHabitacion}
  - {#NumeroHabitacion → Ocupada}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#NumeroHabitacion), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 12:**

***HabReserva(#NumeroHabitacion, #CReserva)***

- **Claves Candidatas:** {#NumeroHabitacion, #CReserva}
- **Dependencias Funcionales:**
  - {#NumeroHabitacion → #CReserva}
  - {#CReserva → #NumeroHabitacion}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#NumeroHabitacion, que es la clave primaria, y #CReserva que también es superclave), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 13:**

***Desperfecto(#CDesperfecto, Prioridad, Descripcion, FechaObservacion)***

- **Claves Candidatas:** {#CDesperfecto}
- **Dependencias Funcionales:**
  - {#CDesperfecto → Prioridad}
  - {#CDesperfecto → Descripcion}
  - {#CDesperfecto → FechaObservacion}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#CDesperfecto), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 14:**

***Reparacion(#CDesperfecto, #NumeroHabitacion, PrecioReparacion)***

- **Claves Candidatas:** {#CDesperfecto}
- **Dependencias Funcionales:**
  - {#CDesperfecto → #NumeroHabitacion}
  - {#CDesperfecto → PrecioReparacion}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#CDesperfecto), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.
-

- **Tabla 15:**

***Utensilio(#CUtensilio, Nombre, Stock, CompañíaDistrib)***

- **Claves Candidatas:** {#CUtensilio, Nombre}
- **Dependencias Funcionales:**
  - {#CUtensilio → Nombre}
  - {#CUtensilio → Stock}
  - {#CUtensilio → CompañíaDistrib}
  - {Nombre → #CUtensilio}
  - {Nombre → Stock}
  - {Nombre → CompañíaDistrib}
- **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave (#CUtensilio, que es la clave primaria, y #Nombre que también es superclave), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

- **Tabla 16:**

***UsoUtensilio(#CDesperfecto, #CUtensilio, CantidadUsada)***

○ **Dependencias Funcionales:**

▪ { {#CDesperfecto, #CUtensilio} → CantidadUsada }

○ **Normalización:** Para toda dependencia funcional no trivial descrita en el punto anterior, a la izquierda hay una superclave ({#CDesperfecto, #CUtensilio}), luego, ya está en FNBC y, por tanto, también está en 3FN, 2FN y 1FN.

## 12. Esquema de Tablas tras la Normalización

Como se puede observar en el apartado anterior, las 16 tablas que forman nuestro sistema de información están en FNBC, por tanto, el esquema de tablas es el mismo que el del apartado 11.

## 13. Sentencias SQL para la Creación de las Tablas y la Inserción de Tuplas

A la hora de crear las tablas e insertar tuplas en cada tabla, estas acciones se realizarán en el orden que se describe a continuación (evitando así problema de referencias de claves externas). Hemos modificado algunos valores de tipo cadena de caracteres de varias tuplas, quitándole el acento a las vocales que lo tenían ya que esto daba problemas a la hora de, por ejemplo, eliminar dichas tuplas al no codificarse bien los caracteres acentuados. Por ello, aunque en las imágenes inferiores de inserción de tuplas aparezca una vocal acentuada, en realidad no lo está.

- Tabla 1:

*Trabajador(#DNI\_Pasaporte, Nombre, Apellidos, Cargo, Departamento, FechaNacimiento, Telefono, CorreoElectronico)*

- Creación de la Tabla:

```
CREATE TABLE Trabajador (  
    DNI_Pasaporte VARCHAR2(15) PRIMARY KEY,  
    Nombre VARCHAR2(20) NOT NULL,  
    Apellidos VARCHAR2(30) NOT NULL,  
    Cargo VARCHAR2(20),  
    Departamento VARCHAR2(20),  
    FechaNacimiento DATE,  
    Telefono VARCHAR2(15) NOT NULL,  
    CorreoElectronico VARCHAR2(50) UNIQUE  
);
```

- **Inserción de Tuplas:**

```
INSERT INTO Trabajador VALUES ('79212958G', 'Antonio', 'Morales Reinos', 'Recepcionista', 'Recepción', TO_DATE('24/11/1984','dd/mm/yyyy'), '630847299', 'antoniomr@gmail.com');
INSERT INTO Trabajador VALUES ('18371847A', 'Andrea', 'Gutierrez Alfaro', 'Directora', 'Dirección', TO_DATE('08/01/2001','dd/mm/yyyy'), '638492483', 'andreagalf@gmail.com');
INSERT INTO Trabajador VALUES ('18371856B', 'Benito', 'Pérez García', 'Camarero', 'Restauración', TO_DATE('10/11/1999','dd/mm/yyyy'), '628393849', 'benitopg@gmail.com');
INSERT INTO Trabajador VALUES ('18371833C', 'Sara', 'Sánchez Torres', 'Cocinera', 'Restauración', TO_DATE('18/11/1978','dd/mm/yyyy'), '628474839', 'sarast@gmail.com');
INSERT INTO Trabajador VALUES ('18371822D', 'Ana', 'Núñez Quesada', 'Limpiadora', 'Limpieza', TO_DATE('21/11/1988','dd/mm/yyyy'), '684235252', 'ananq@gmail.com');
```

- **Tabla 2:**

***ActividadOcio(#ActOcio, Nombre, Descripcion, Fecha, Hora, NumPlazas)***

- **Creación de la Tabla:**

```
CREATE TABLE ActividadOcio (
    ActOcio VARCHAR2(10) PRIMARY KEY,
    Nombre VARCHAR2(20) NOT NULL,
    Descripcion VARCHAR2(100),
    Fecha DATE,
    Hora DATE,
    NumPlazas NUMBER(2,0) CHECK(NumPlazas >= 0)
);
```

- **Inserción de Tuplas:**

```
INSERT INTO ActividadOcio VALUES ('#A00000001', 'Fiesta Ibicenca', 'Fiesta donde todo el mundo va vestido de blanco', TO_DATE('01/12/2022','DD/MM/YYYY'), TO_DATE('12:00','HH24:MI'), 40);
INSERT INTO ActividadOcio VALUES ('#A00000002', 'Aquagym', 'Variante del aeróbic que se realiza en la piscina', TO_DATE('02/11/2022','DD/MM/YYYY'), TO_DATE('18:00','HH24:MI'), 18);
INSERT INTO ActividadOcio VALUES ('#A00000003', 'Gymkana', 'Diferentes juegos de competición en equipo', TO_DATE('14/10/2022','DD/MM/YYYY'), TO_DATE('11:30','HH24:MI'), 30);
INSERT INTO ActividadOcio VALUES ('#A00000004', 'Yoga', 'Conjunta de técnicas de concentración para conseguir un mayor control físico y mental', TO_DATE('16/12/2022','DD/MM/YYYY'), TO_DATE('12:45','HH24:MI'), 14);
INSERT INTO ActividadOcio VALUES ('#A00000005', 'Concurso de Cocina', 'Evento para poner a prueba las habilidades culinarias', TO_DATE('01/02/2023','DD/MM/YYYY'), TO_DATE('13:00','HH24:MI'), 14);
```

- Tabla 3:

*OrganizacionAct(#ActOcio, #DNI\_Pasaporte)*

- Creación de la Tabla:

```
CREATE TABLE OrganizacionAct (  
    ActOcio VARCHAR2(10) REFERENCES ActividadOcio(ActOcio) PRIMARY KEY,  
    DNI_Pasaporte VARCHAR2(15) REFERENCES Trabajador(DNI_Pasaporte)  
);
```

- Inserción de Tuplas:

```
INSERT INTO OrganizacionAct VALUES ('#A00000001', '18371847A');  
INSERT INTO OrganizacionAct VALUES ('#A00000002', '79212958G');  
INSERT INTO OrganizacionAct VALUES ('#A00000003', '18371847A');  
INSERT INTO OrganizacionAct VALUES ('#A00000004', '79212958G');  
INSERT INTO OrganizacionAct VALUES ('#A00000005', '18371833C');
```

- Tabla 4:

*Cliente(#DNI\_Pasaporte, Telefono, CuentaBancaria)*

- Creación de la Tabla:

```
CREATE TABLE Cliente (  
    DNI_Pasaporte VARCHAR2(15) PRIMARY KEY,  
    Telefono VARCHAR2(15) NOT NULL,  
    CuentaBancaria VARCHAR2(30)  
);
```

- Inserción de Tuplas:

```
INSERT INTO Cliente VALUES ('77654234H', '603345683', 'ISPB12341223456');
INSERT INTO Cliente VALUES ('77614234S', '603345681', 'ISPB12341231111');
INSERT INTO Cliente VALUES ('77623434L', '678094345', 'ISPB12341232233');
INSERT INTO Cliente VALUES ('77611224A', '603345681', 'ISPB12123234123');
INSERT INTO Cliente VALUES ('77645678D', '612367894', 'ISPB12112345123');
```

- Tabla 5:

*ReservaAct(#DNI Pasaporte, #ActOcio, FechaReserva, HoraReserva, NumPersonas)*

- Creación de la Tabla:

```
CREATE TABLE ReservaAct (
    DNI_Pasaporte VARCHAR2(15) REFERENCES Cliente(DNI_Pasaporte),
    ActOcio VARCHAR2(10) REFERENCES ActividadOcio(ActOcio),
    FechaReserva DATE,
    HoraReserva DATE,
    NumPlazas INT CHECK(NumPlazas > 0),

    PRIMARY KEY(DNI_Pasaporte, ActOcio)
);
```

- Inserción de Tuplas:

```
INSERT INTO ReservaAct VALUES ('77623434L', '#A000000001', TO_DATE('01/12/2022', 'DD/MM/YYYY'), TO_DATE('12:00', 'HH24:MI'), 4);
INSERT INTO ReservaAct VALUES ('77611224A', '#A000000002', TO_DATE('02/11/2022', 'DD/MM/YYYY'), TO_DATE('18:00', 'HH24:MI'), 2);
INSERT INTO ReservaAct VALUES ('77623434L', '#A000000005', TO_DATE('01/02/2023', 'DD/MM/YYYY'), TO_DATE('13:00', 'HH24:MI'), 1);
```



- Tabla 6:

*OpinionAct(#DNI Pasaporte, #ActOcio, Opinion, Puntuacion)*

- Creación de la Tabla:

```
CREATE TABLE OpinionAct (
  DNI_Pasaporte VARCHAR2(15) REFERENCES Cliente(DNI_Pasaporte),
  ActOcio VARCHAR2(10) REFERENCES ActividadOcio(ActOcio),
  Opinion VARCHAR2(200),
  Puntuacion INT CHECK(Puntuacion >= 0 AND Puntuacion <= 10) NOT NULL
);
```

- Inserción de Tuplas:

```
INSERT INTO OpinionAct VALUES ('77623434L', '#A00000001', 'Perfecto', 10);
INSERT INTO OpinionAct VALUES ('77611224A', '#A00000002', 'Mejorable', 4);
```

- Tabla 7:

*PlazaParking(#NumeroPlaza, Ocupada)*

- Creación de la Tabla:

```
CREATE TABLE PlazaParking (
  NumeroPlaza NUMBER(3,0) PRIMARY KEY CHECK(NumeroPlaza >= 0) NOT NULL,
  Ocupada NUMBER(1,0) CHECK(Ocupada = 0 OR Ocupada = 1)
);
```

- Inserción de Tuplas:

```
INSERT INTO PlazaParking VALUES ('17', 0);
INSERT INTO PlazaParking VALUES ('253', 0);
INSERT INTO PlazaParking VALUES ('467', 1);
INSERT INTO PlazaParking VALUES ('682', 0);
INSERT INTO PlazaParking VALUES ('947', 1);
```

- Tabla 8:

*ReservaParking(#NumeroPlaza, #DNI\_Pasaporte, TiempoOcupacion)*

- Creación de la Tabla:

```
CREATE TABLE ReservaParking (  
    NumeroPlaza INT PRIMARY KEY REFERENCES PlazaParking(NumeroPlaza),  
    DNI_Pasaporte VARCHAR2(15) REFERENCES Cliente(DNI_Pasaporte) UNIQUE,  
    TiempoOcupacion INT  
);
```

- Inserción de Tuplas:

```
INSERT INTO ReservaParking VALUES ('947', '77623434L', 98);
```

- Tabla 9:

*ReservaHotel(#CReserva, FechaEntrada, FechaSalida, TipoHabitacionPedida,  
Check\_in, Check\_out, Cancelada)*

- Creación de la Tabla:

```
CREATE TABLE ReservaHotel (  
    CReserva VARCHAR2(10) PRIMARY KEY,  
    FechaEntrada DATE NOT NULL,  
    FechaSalida DATE NOT NULL,  
    TipoHabitacionPedida VARCHAR2(10) NOT NULL,  
    Check_out NUMBER(1,0) CHECK(Check_out = 0 OR Check_out = 1),  
    Cancelada NUMBER(1,0) CHECK(Cancelada = 0 OR Cancelada = 1)  
);
```

- **Inserción de Tuplas:**

```
INSERT INTO ReservaHotel VALUES ('R00000001', TO_DATE('01/11/2022','dd/mm/yyyy'), TO_DATE('10/12/2022','dd/mm/yyyy'), 'Suite', 1, 0);
INSERT INTO ReservaHotel VALUES ('R00000002', TO_DATE('30/11/2022','dd/mm/yyyy'), TO_DATE('04/12/2022','dd/mm/yyyy'), 'Doble', 1, 0);
INSERT INTO ReservaHotel VALUES ('R00000003', TO_DATE('12/01/2023','dd/mm/yyyy'), TO_DATE('02/02/2023','dd/mm/yyyy'), 'Individual', 0, 0);
INSERT INTO ReservaHotel VALUES ('R00000004', TO_DATE('13/01/2023','dd/mm/yyyy'), TO_DATE('26/01/2023','dd/mm/yyyy'), 'Suite', 0, 0);
INSERT INTO ReservaHotel VALUES ('R00000005', TO_DATE('02/01/2023','dd/mm/yyyy'), TO_DATE('05/01/2023','dd/mm/yyyy'), 'Doble', 0, 1);
```

- **Tabla 10:**

***RegistroHotel(#CReserva, #DNI\_Pasaporte)***

- **Creación de la Tabla:**

```
CREATE TABLE RegistroHotel (  
    CReserva VARCHAR2(10) REFERENCES ReservaHotel(CReserva) PRIMARY KEY,  
    DNI_Pasaporte VARCHAR2(15) REFERENCES Cliente(DNI_Pasaporte)  
);
```

- **Inserción de Tuplas:**

```
INSERT INTO RegistroHotel VALUES ('R00000001', '77614234S');  
INSERT INTO RegistroHotel VALUES ('R00000002', '77654234H');  
INSERT INTO RegistroHotel VALUES ('R00000003', '77623434L');  
INSERT INTO RegistroHotel VALUES ('R00000004', '77611224A');  
INSERT INTO RegistroHotel VALUES ('R00000005', '77645678D');
```

- **Tabla 11:**

*Habitacion(#NumeroHabitacion, TipoHabitacion, Ocupada)*

- Creación de la Tabla:

```
CREATE TABLE Habitacion (
    NumeroHabitacion NUMBER(3,0) PRIMARY KEY CHECK(NumeroHabitacion > 0),
    TipoHabitacion VARCHAR2(10) NOT NULL
);
```

- Inserción de Tuplas:

```
INSERT INTO Habitacion VALUES (8, 'Individual', 0);
INSERT INTO Habitacion VALUES (123, 'Doble', 0);
INSERT INTO Habitacion VALUES (369, 'Suite', 1);
INSERT INTO Habitacion VALUES (401, 'Individual', 1);
INSERT INTO Habitacion VALUES (500, 'Doble', 0);
INSERT INTO Habitacion VALUES (673, 'Doble', 0);
INSERT INTO Habitacion VALUES (837, 'Individual', 0);
```

- **Tabla 12:**

*HabReserva(#NumeroHabitacion, #CReserva)*

- Creación de la Tabla:

```
CREATE TABLE HabReserva (
    NumeroHabitacion INT REFERENCES Habitacion(NumeroHabitacion) PRIMARY KEY,
    CReserva VARCHAR2(10) REFERENCES ReservaHotel(CReserva) UNIQUE
);
```

- Inserción de Tuplas:

```
INSERT INTO HabReserva VALUES (369, '#R00000004');
INSERT INTO HabReserva VALUES (401, '#R00000003');
```

- **Tabla 13:**

***Desperfecto(#CDesperfecto, Prioridad, Descripcion, FechaObservacion)***

- **Creación de la Tabla:**

```
CREATE TABLE Desperfecto (  
    CDesperfecto VARCHAR2(10) PRIMARY KEY,  
    Prioridad NUMBER(3,0) CHECK(Prioridad >= 0),  
    Descripcion VARCHAR2(200) NOT NULL,  
    FechaObservacion DATE  
);
```

- **Inserción de Tuplas:**

```
INSERT INTO Desperfecto VALUES ('#D00000001', NULL, 'Mancha de humedad en la habitación 673', TO_DATE('29/11/2022','dd/mm/yyyy'));  
INSERT INTO Desperfecto VALUES ('#D00000002', 2, 'No hay luz en la habitación 837', TO_DATE('30/11/2022','dd/mm/yyyy'));  
INSERT INTO Desperfecto VALUES ('#D00000003', 1, 'No funciona la cadena del váter en la habitación 123', TO_DATE('05/01/2023','dd/mm/yyyy'));
```

- **Tabla 14:**

***Reparacion(#CDesperfecto, #NumeroHabitacion, PrecioReparacion)***

- **Creación de la Tabla:**

```
CREATE TABLE Reparacion (  
    CDesperfecto VARCHAR2(10) REFERENCES Desperfecto(CDesperfecto) PRIMARY KEY,  
    NumeroHabitacion INT REFERENCES Habitacion(NumeroHabitacion),  
    PrecioReparacion NUMBER(8,2) CHECK(PrecioReparacion > 0)  
);
```

- **Inserción de Tuplas:**

```
INSERT INTO Reparacion VALUES ('#D00000003', 123, 56.50);  
INSERT INTO Reparacion VALUES ('#D00000002', 837, NULL);
```

- **Tabla 15:**

***Utensilio(#CUtensilio, Nombre, Stock, CompañíaDistrib)***

- **Creación de la Tabla:**

```
CREATE TABLE Utensilio (
    CUtensilio VARCHAR2(10) PRIMARY KEY,
    Nombre VARCHAR2(20) UNIQUE,
    Stock INT CHECK(Stock > 0),
    CompaniaDistrib VARCHAR2(20)
);
```

- **Inserción de Tuplas:**

```
INSERT INTO Utensilio VALUES ('#U00000001', 'Tornillo Plano', 500, 'Screws Company');
INSERT INTO Utensilio VALUES ('#U00000002', 'Tornillo Hexagonal', 329, 'Screws Company');
INSERT INTO Utensilio VALUES ('#U00000003', 'Tornillo Redondo', 446, 'Screws Company');
INSERT INTO Utensilio VALUES ('#U00000004', 'Bombilla', 97, 'Lamps Company');
INSERT INTO Utensilio VALUES ('#U00000005', 'Tirador Váter', 49, 'WC Products');
INSERT INTO Utensilio VALUES ('#U00000006', 'Bote Limpiacristales', 15, 'Cleaning S.A.');
```

- **Tabla 16:**

***UsoUtensilio(#CDesperfecto, #CUtensilio, CantidadUsada)***

- **Creación de la Tabla:**

```
CREATE TABLE UsoUtensilio (
    CDesperfecto CHAR(10) REFERENCES Desperfecto(CDesperfecto),
    CUtensilio CHAR(10) REFERENCES Utensilio(CUtensilio),
    CantidadUsada INT CHECK(CantidadUsada > 0),
    PRIMARY KEY (CDesperfecto, CUtensilio)
);
```

- **Inserción de Tuplas:**

```
INSERT INTO UsoUtensilio VALUES ('#D00000003', '#U000000005', 1);
INSERT INTO UsoUtensilio VALUES ('#D00000003', '#U000000003', 4);
```

## 14. Descripción de las Transacciones Identificadas

Hemos identificado como **transacciones** aquellas **operaciones que conllevan la inserción, modificación o eliminación de tuplas en las tablas (no se considera como transacción aquellas operaciones de consulta, pues no se modifica la base de datos)**, debido a que pueden interactuar varios trabajadores al mismo tiempo en dichas operaciones.

Al inicio de cada operación considerada como una transacción hemos colocado un *savepoint*, de manera que al finalizar dicha operación se le preguntará al usuario si desea guardar los cambios realizados en la base de datos. En caso afirmativo, se hace un *commit* en la base de datos para guardar permanentemente dichos cambios y, en caso negativo, se deshace dicha operación mediante un *rollback* al *savepoint* mencionado anteriormente.

## 15. Sentencias SQL para la Creación de los Disparadores

### Subsistema 1: Trabajadores

Encargada: Mónica Calzado Granados

- **Disparador 1:**

```
CREATE OR REPLACE TRIGGER edad_min_laboral
BEFORE INSERT OR UPDATE ON Trabajador
FOR EACH ROW
BEGIN
IF (trunc(months_between(to_date(to_char(SYSDATE, 'dd/mm/yyyy'), 'dd/mm/yyyy'), to_date(to_char(:new.FechaNacimiento, 'dd/mm/yyyy'), 'dd/mm/yyyy'))/12) < 16) THEN
RAISE_APPLICATION_ERROR(-20001, 'El trabajador debe ser mayor de 16 años');
END IF;
END;
```

- **Disparador 2:**

```
CREATE OR REPLACE TRIGGER borrado_cascada_tr_act1
BEFORE DELETE ON Trabajador
FOR EACH ROW
BEGIN
--Borramos OrganizacionAct antes que la Actividad de ocio pues la primera referencia a la segunda
DELETE FROM OrganizacionAct WHERE DNI_Pasaporte= :OLD.DNI_Pasaporte;
END;
```

- **Disparador 3:**

```
CREATE OR REPLACE TRIGGER borrado_cascada_tr_act2
  AFTER DELETE ON OrganizacionAct
  FOR EACH ROW
BEGIN
  --Este trigger se ejecuta como consecuencia de borrado_cascada_tr_act1
  --Tras borrar OrganizacionAct eliminamos las otras dos referencias al campo ActOcio de ActividadOcio
  --y por último borramos la Actividad de ocio
  DELETE FROM ReservaAct WHERE ActOcio = :OLD.ActOcio;
  DELETE FROM OpinionAct WHERE ActOcio = :OLD.ActOcio;
  DELETE FROM ActividadOcio WHERE ActOcio = :OLD.ActOcio;
END;
```

## Subsistema 2: Reservas del Hotel

Encargada: Alejandro Cárdenas Barranco

- **Disparador 4:**

```
CREATE OR REPLACE TRIGGER tr_no_aceptar_reservas
BEFORE INSERT ON HabReserva
FOR EACH ROW
DECLARE
  c_out NUMBER(1,0);
  cancela NUMBER(1,0);
BEGIN
  SELECT Check_out,cancelada INTO c_out,cancela FROM ReservaHotel WHERE CReserva = :NEW.CReserva;
  IF (c_out = 1 or cancela = 1) THEN
    RAISE_APPLICATION_ERROR(-20001, 'No se puede registrar una habitación para una reserva que ya ha realizado el check-out o se ha cancelado');
  END IF;
END;
```

- **Disparador 5:**

```
CREATE OR REPLACE TRIGGER tr_comprobacion_cancelacion
BEFORE UPDATE ON ReservaHotel
FOR EACH ROW
DECLARE
  l_count INTEGER;
BEGIN
  -- Comprobar si existen reservas en HabReserva para la reserva en cuestión
  SELECT COUNT(*) INTO l_count FROM HabReserva WHERE CReserva = :new.CReserva;

  -- Comprobar que se está estableciendo Cancelada en true
  IF :new.Cancelada = 1 AND :old.Cancelada = 0 THEN
    -- Comprobar si existen reservas en HabReserva para la reserva en cuestión
    IF l_count > 0 THEN
      RAISE_APPLICATION_ERROR(-20001, 'No se puede cancelar una reserva en la que se haya realizado un check-in');
    END IF;
  END IF;
END;
```



- **Disparador 6:**

```
CREATE OR REPLACE TRIGGER tr_comprobar_fechas
BEFORE INSERT OR UPDATE ON ReservaHotel
FOR EACH ROW
BEGIN
    IF :new.FechaSalida <= :new.FechaEntrada THEN
        RAISE_APPLICATION_ERROR(-20001, 'La fecha de salida debe ser posterior a la fecha de entrada');
    END IF;
END;
```

### Subsistema 3: Actividades de Ocio

Encargada: Álvaro Rodríguez Gallardo

- **Disparador 7:**

```
CREATE OR REPLACE TRIGGER no_reserva_act_ocio
BEFORE INSERT ON ReservaAct
FOR EACH ROW
DECLARE
    contador NUMBER(5,0);
BEGIN
    -- Verifica si el DNI/Pasaporte del cliente de la tupla que se va a insertar tiene una habitación asignada
    SELECT COUNT(*) INTO contador FROM RegistroHotel r, HabReserva h WHERE r.CReserva = h.CReserva AND DNI_Pasaporte = :new.DNI_Pasaporte;
    IF (contador = 0) THEN
        -- Si dicho DNI/Pasaporte no tiene ninguna habitación asignada, se lanza una excepción
        RAISE_APPLICATION_ERROR(-20000, 'No se puede realizar la reserva de la actividad de ocio (El DNI/Pasaporte introducido no tiene ninguna habitación asignada)');
    END IF;
END;
```

- **Disparador 8:**

```
CREATE OR REPLACE TRIGGER opinion_dni_cliente
BEFORE
INSERT ON OpinionAct
FOR EACH ROW
DECLARE
    numero NUMBER(5,0);
BEGIN
    -- Si el dni y código están en una misma fila de ReservaAct, se permite opinar
    SELECT COUNT(*) INTO numero FROM ReservaAct ra WHERE ra.DNI_Pasaporte=:new.DNI_Pasaporte AND ra.ActOcio=:new.ActOcio;
    IF (numero = 0) THEN
        RAISE_APPLICATION_ERROR(-20600, :new.DNI_Pasaporte || 'El DNI/Pasaporte introducido no corresponde con ninguna reserva asociada a tal código');
    END IF;
END;
```

## Subsistema 4: Parking

Encargada: Juan Manuel Rodríguez Gómez

- Disparador 9:

```
CREATE OR REPLACE TRIGGER no_reserva_parking
BEFORE INSERT ON ReservaParking
FOR EACH ROW
DECLARE
    contador NUMBER(5,0);
BEGIN
    -- Verifica si el DNI/Pasaporte del cliente de la tupla que se va a insertar tiene una habitación asignada
    SELECT COUNT(*) INTO contador FROM RegistroHotel r, HabReserva h WHERE r.CReserva = h.CReserva AND DNI_Pasaporte = :new.DNI_Pasaporte;
    IF (contador = 0) THEN
        -- Si dicho DNI/Pasaporte no tiene ninguna habitación asignada, se lanza una excepción
        RAISE_APPLICATION_ERROR(-20000, 'No se puede realizar la reserva de la plaza de parking (El DNI/Pasaporte introducido no tiene ninguna habitación asignada)');
    END IF;
END;
```

## Subsistema 5: Mantenimiento de Instalaciones

Encargada: Jesús García León

- Disparador 10:

```
CREATE OR REPLACE TRIGGER precio_rep_no_modif
BEFORE
UPDATE OF PrecioReparacion ON Reparacion
FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR (-20001, 'Esta tabla no se puede modificar, pruebe a borrar e insertar la tupla con el nuevo precio');
END;
```

- Disparador 11:

```
CREATE OR REPLACE TRIGGER no_superar_presupuesto
BEFORE
INSERT ON Reparacion
FOR EACH ROW
DECLARE
    precio_total NUMBER(8,2);
    presupuesto_total NUMBER(12,2) := 10000;
BEGIN
    SELECT SUM(PRECIOREPARACION) INTO PRECIO_TOTAL FROM REPARACION;
    precio_total := precio_total + :new.PRECIOREPARACION;
    IF (precio_total > presupuesto_total) THEN
        RAISE_APPLICATION_ERROR (-20600, 'La suma de los precios de reparación no puede superar el presupuesto total (10000€)');
    END IF;
END;
```

## 16. Elección del Software

Para la realización de esta práctica, hemos elegido usar **Windows 10** como sistema operativo y **Python** como lenguaje de programación. La elección de usar Python se basa en que es un lenguaje muy popular, con una sintaxis muy simple y fácil de entender y que, como característica más destacable, cuenta con una amplia variedad de paquetes para programar en diferentes ámbitos de la informática como pueden ser Machine Learning, Ciencia de Datos, etc. Para descargar Python en Windows 10, basta con instalarlo desde Microsoft Store o, como hemos hecho en nuestro grupo, descargar el instalador que se ajuste a nuestro sistema operativo (32 o 64 bits) de forma gratuita mediante siguiente enlace:

<https://www.python.org/downloads/windows/>

Cabe destacar que para comprobar que las acciones realizadas por nuestro código funcionaban correctamente, se ha usado **SQLdeveloper** para establecer una conexión cliente con la base de datos (la instalación de SQLdeveloper es un poco laboriosa, pero se pueden encontrar vídeos con todo el proceso de instalación explicado detalladamente).

Finalmente, para conectarnos a la base de datos desde la aplicación, lo intentamos con el paquete pyodbc en un primer momento, el cual es el que usamos en el seminario 1. Sin embargo, solo te otorgan un mes de prueba para tener activado el driver a partir del cual podemos conectarnos a la base de datos de Oracle con pyodbc, y dicho mes de prueba se nos caducó. Por lo tanto, buscamos otro paquete como alternativa para realizar la conexión a la base de datos, el cual se llama **oracledb**. Dicho paquete es similar a pyodbc, la única diferencia es que pyodbc nos permitía conectarnos a bases de datos de todo tipo y oracledb solo nos permite conectarnos a bases de datos de Oracle. Lo interesante del paquete oracledb es que no necesitamos ningún driver para el cual solo tengamos un mes de prueba, sino que lo podremos usar siempre de forma gratuita. Para instalarlo basta con ejecutar el siguiente comando en la consola de Windows:

***py -m pip install oracledb***