

# Práctica de Diseño de Software

Grado en Ingeniería Informática  
Mención Ingeniería del Software  
Universidad de Valladolid

curso 2022-2023.  
28/04/2023.  
versión 1.0

# Índice general

<b>1. Supuesto práctico. Requisitos</b>	<b>2</b>
1.1. Fragmento del documento de especificación de requisitos . . . . .	3
1.1.1. Requisitos funcionales . . . . .	4
1.1.2. Reglas de negocio . . . . .	4
1.1.3. Requisitos no funcionales y restricciones . . . . .	5
1.1.4. Requisitos de información . . . . .	5
1.1.5. Restricciones adicionales . . . . .	5
1.2. Casos de uso: especificación (parcial) de escenarios . . . . .	6
1.2.1. Identificarse . . . . .	6
1.2.2. Consultar vehículos de la flota del punto que están situados en otro punto .	7
1.2.3. Registrar devolución de vehículo en alquiler . . . . .	8
1.2.4. Registrar recepción de vehículo del taller . . . . .	10
<b>2. Documentación parcial de Análisis</b>	<b>11</b>
2.1. Vista parcial del diagrama de Casos de Uso . . . . .	11
2.2. Modelo del dominio . . . . .	13
2.3. Restricciones a tener en cuenta en la implementación . . . . .	16
2.4. Análisis arquitectónico . . . . .	16
<b>3. Diseño de Datos</b>	<b>17</b>
3.1. Diseño lógico (datamodel style) . . . . .	17
3.2. Scripts de creación de la base de datos . . . . .	20
<b>4. Normas de entrega y criterios de evaluación</b>	<b>27</b>
4.1. Indicaciones para la entrega . . . . .	27
4.2. Estructura y contenidos de la entrega . . . . .	27
4.3. Criterios de Evaluación . . . . .	28
4.3.1. Reglas del Concurso . . . . .	30
4.3.2. Hitos en la realización del proyecto . . . . .	31

# Capítulo 1

## Supuesto práctico. Requisitos

PucelaCar es una compañía de alquiler de automóviles en la que se manejan distintas categorías de vehículos (pequeño, mediano, grande, gama alta). Cada vehículo tiene una marca y un modelo, una matrícula y un color. Todos los vehículos del mismo modelo comparten una serie de características como la cantidad de puertas, los metros de largo, ancho y alto, así como la capacidad del maletero en litros y la capacidad en cuanto a pasajeros. Las categorías caracterizan a los modelos y definen unos tamaños mínimos y máximos de largo y ancho para considerar un vehículo en dicha categoría y en el caso de gama alta una descripción de las prestaciones que debe tener el vehículo para considerarse de gama alta. La compañía dispone de una serie de puntos en los que se realiza la entrega y recogida de vehículos (generalmente situados en estaciones, aeropuertos, etc.). Estos puntos tienen una flota de vehículos asignada que puede cambiar. Cada vehículo no alquilado y que no se ha enviado al taller debe estar situado en un punto de entrega-recogida.

Cada vehículo se puede reservar (y alquilar) todos los días del año. La recogida se puede hacer a partir de las 8h del primer día de alquiler y la entrega antes de las 20h del último día. Los vehículos están disponibles en una fecha y hora dadas cuando en ese momento no se encuentran averiados, reservados, en alquiler, o en preparación para el siguiente alquiler durante una hora después de ser devueltos. Un vehículo puede llegar a estar propuesto para baja y no está disponible en ningún momento a partir de la fecha en que se ha propuesto para baja.

Las reservas se deben realizar con un día de antelación como mínimo al alquiler deseado. Al reservar, el cliente indica los puntos de recogida y devolución que necesita, así como también las fechas y horas correspondientes, y elige una categoría del vehículo que desea alquilar, y opcionalmente indica algunas restricciones como capacidad en litros del maletero y cantidad de ocupantes. Esta reserva solicitada puede pasar a ser firme si hay un vehículo disponible en las fechas deseadas. Es necesario que el cliente sea identificado y que en el sistema se conozca el número de su tarjeta de crédito. Para confirmar la reserva también se comprueba con un sistema externo de pagos seguros mediante tarjeta que hay autorización de crédito para al menos un día del alquiler. Se envía un email al cliente con los datos de la reserva confirmada. Cada reserva tiene asociado un identificador único que permite utilizarlo para realizar operaciones con la reserva, como la anulación.

Si el vehículo no se recoge el día previsto, se factura un día a la tarjeta de crédito indicada en la reserva. La reserva se anula automáticamente 1h después del horario de cierre (como se cierra a las 20h, la anulación automática es a las 21h) del día previsto de recogida. Si el vehículo se entrega después de las 20h del último día de la reserva, se facturará como mínimo un día más. Las reservas realizadas se pueden anular hasta dos días antes de la fecha prevista de recogida sin gastos para el cliente.

El precio básico del alquiler se calcula en función del número de días de alquiler y de la categoría del vehículo. Aparte del alquiler en sí, habrá un coste extra si el cliente solicita algún tipo de seguro adicional (seguro personal de accidentes, antirrobo, etc.) además del seguro obligatorio (daños a terceros). El precio de cada tipo de seguro adicional es fijo por día según la categoría del vehículo. Se añade un extra establecido si el punto de entrega no es el mismo que el punto

de recogida. Podrá haber descuentos en función de los puntos (entre 10 y 40 puntos por día de alquiler, según la categoría del vehículo) acumulados por los clientes, siempre y cuando se hayan inscrito en el programa de premios. En ese caso, el sistema mantiene un perfil del cliente con las preferencias de tipo de seguro, categoría del vehículo habitualmente preferido, los puntos acumulados, etc. De los clientes el sistema siempre almacena nombre completo, número de teléfono, e-mail y le asigna un número de cliente.

El pago del alquiler se hace siempre utilizando la tarjeta de crédito asociada al cliente en el momento de la recogida del vehículo (inicio del alquiler). En ese momento se registra el pago y, opcionalmente, se imprime la factura correspondiente que estará formada por diferentes líneas con los distintos conceptos que se cobrarán (básico: alquiler por día, uno por cada extra y descuentos, de aplicarse).

En los puntos de entrega-recogida, los empleados de atención al público realizan los registros de entrega (inicio de alquiler) y recogida respectivamente. La compañía requiere conocer qué empleado ha realizado la entrega/recogida.

Al realizar la recepción de un vehículo, cuando el cliente hace entrega de este, un empleado técnico del punto realizará una revisión del vehículo y registrará cualquier incidencia (accidente, avería, no se entrega a tiempo, etc.). En caso de incidencia por accidente o avería, los vehículos son enviados a un taller asociado en la localidad en la que se encuentra el punto de entrega-recogida con una orden de reparación que crea el técnico del punto. Estos talleres no pertenecen a la compañía de alquiler de coches, pero al ser taller asociado podrá registrar la entrada y salida del vehículo. Al salir del taller, el vehículo puede registrarse como propuesto para baja, o estará para continuar su servicio normal, aunque inicialmente no quedará disponible hasta transcurrido el tiempo de una hora (desde su llegada al punto de entrega-recogida) previsto para su revisión por parte del técnico del punto.

En las oficinas centrales de la compañía se ocupan de la gestión de la flota de vehículos, de la gestión de los puntos de entrega-recogida, de la gestión de los empleados, de las redes de talleres asociados en cada localidad donde tiene la empresa puntos de entrega-recogida y de las aseguradoras con las que trabaja la compañía, así como de la facturación asociada a los talleres y aseguradoras. Esta parte queda excluida de este desarrollo.

Se comenzó con el desarrollo de un proyecto siguiendo una adaptación del Proceso Unificado de Desarrollo. En sus primeras fases (Inicio y Elaboración) ya se ha realizado la elicitación y especificación de requisitos, el análisis de estos requisitos, en particular el modelo de casos de uso y el modelado del dominio. A partir de los artefactos obtenidos de las tareas ya realizadas, vuestro equipo ha sido subcontratado para diseñar la arquitectura general del sistema, así como el diseño e implementación de una pequeña parte de éste, teniendo en cuenta la arquitectura general y la futura integración del resto de funcionalidad del sistema completo.

Podría decirse que sois un equipo en pruebas. Probablemente, fruto de la satisfacción de la parte contratante con la ejecución de vuestro subproyecto está en juego una futura contratación del diseño e implementación de una parte mucho mayor del resto del sistema.

La parte contratante os ha pasado el documento de especificación de requisitos del que se expone a continuación un fragmento.

## **1.1. Fragmento del documento de especificación de requisitos**

Identificación de algunos roles de usuarios del sistema:

- Gerente de un punto de recogida-entrega
- Empleado de atención al público de un punto de recogida-entrega
- Técnico de un punto de recogida-entrega
- Cliente
- ... Se omiten otros posibles roles por brevedad

### **1.1.1. Requisitos funcionales**

**RF-1** El sistema deberá permitir a los empleados identificarse.

**RF-2** El sistema deberá permitir al gerente de un punto de recogida-entrega conocer el estado de la flota asignada a su punto, en particular:

**RF-2a** deberá permitir consultar los vehículos de la flota que se encuentran situados en otro punto.

**RF-2b** deberá permitir consultar los vehículos de la flota propuestos para baja.

**RF-2...** ...

**RF-3** El sistema deberá permitir a los empleados de Atención al Cliente de un punto crear un alquiler cuando un cliente llegue a hacer efectiva su reserva teniendo en cuenta el punto de recogida.

**RF-5** El sistema deberá permitir a los empleados de Atención al Cliente de un punto mostrar la factura básica de un alquiler y sus detalles, e imprimirla (opcionalmente).

**RF-4** El sistema deberá realizar el cobro del alquiler en el momento de la recogida del coche mediante la tarjeta de crédito del cliente.

**RF-6** El sistema deberá permitir a los empleados de atención al público de un punto conocer las reservas del día para el punto en el que trabajan.

**RF-7** El sistema deberá permitir al técnico de un punto añadir incidencias por avería o accidente al registro de devolución de un alquiler.

**RF-8** El sistema deberá permitir al técnico de un punto crear una orden de reparación de un vehículo para un taller asociado.

**RF-9** El sistema deberá permitir al técnico de un punto registrar la recepción de un vehículo que llega del taller y aceptar la propuesta de baja del vehículo, si fuese el caso.

Etc., etc., etc.

### **1.1.2. Reglas de negocio**

**RN-1** Los empleados que no se encuentran en activo [RN-2] no podrán identificarse en el sistema.

**RN-2** Un empleado está en activo cuando se encuentra “Contratado” en cuando a Vinculación actual (Contratado, En ERTE, Despedido) y cuando se encuentra “Trabajando” en cuando a Disponibilidad actual (Trabajando, de vacaciones, de baja temporal).

**RN-3** Para poder alquilar un vehículo se necesita una reserva previa.

**RN-4** Las reservas se deben realizar con un día de antelación como mínimo al alquiler deseado.

**RN-5** Los vehículos no alquilados y que no se han enviado al taller deben estar situado en algún punto de entrega-recogida.

**RN-6** Los vehículos se puede reservar (y alquilar) todos los días del año.

**RN-7** La recogida de un vehículo reservado se puede hacer a partir de las 8h del primer día de alquiler y la entrega antes de las 20h del último día en el punto indicado en la reserva.

**RN-8** Los vehículos están disponibles en una fecha y hora dadas cuando en ese momento no se encuentran averiados, reservados, en alquiler, o en preparación para el siguiente alquiler.

- RN-9** El tiempo de preparación para el siguiente alquiler se establece en 1h siempre que no haya incidencia que produzca que el coche no vaya a estar disponible.
- RN-10** A partir de que un vehículo queda propuesto para baja ya no estará disponible.
- RN-11** Para poder reservar un cliente debe identificarse en el sistema.
- RN-12** Para poder hacer efectiva la reserva en el sistema deberá conocerse la tarjeta de crédito del cliente y se contará con la autorización de un sistema de pagos externos de que el cliente puede pagar el importe de la reserva con esa tarjeta.
- RN-13** A las 21h del día de la reserva sin que se haya recogido (1h después del cierre) , la reserva se anula automáticamente y se cobra un cargo por anulación automática.
- RN-14** El cargo por anulación automática será equivalente al coste de 1 día de alquiler para el vehículo reservado.
- RN-15** Se debe garantizar que la factura no contiene descuentos si el cliente no tiene perfil (si no está inscrito en el programa de puntos).

Etc., etc., etc.

### **1.1.3. Requisitos no funcionales y restricciones**

#### **Requisitos no funcionales**

- RNF-1** El sistema deberá trabajar con una base de datos relacional centralizada en un servidor dedicado.

Se omiten otros requisitos no funcionales en este trabajo para no condicionar al equipo (dado el contexto académico).

### **1.1.4. Requisitos de información**

Los requisitos de información se omiten por brevedad debido a que ya quedan recogidos en el modelado del dominio que se muestra en el Capítulo 2.

### **1.1.5. Restricciones adicionales**

Se ha decidido separar el sistema en al menos dos subsistemas, el que será desarrollado para el uso de los empleados de la empresa (podría dividirse en más subsistemas) y el que será desarrollado para los clientes.

El subsistema dedicado a los clientes será una aplicación web y no será abordada en esta asignatura sino probablemente en otra (¿?).

Para el subsistema dedicado a los empleados de la empresa se ha decidido realizar una aplicación de escritorio con acceso a base de datos (BD). El sistema deberá utilizar una BD, cuyo diseño se aporta, implementada con Derby.

Deberá asegurarse que todos los usuarios están previamente identificados en el sistema para acceder a cualquier función, y que las funciones serán las que correspondan al usuario según su rol en la empresa.

En teoría la BD debe ser centralizada en modo cliente-servidor, pero para facilitar las prácticas en este caso supondremos la conexión a una BD en local. El acceso a los datos deberá ser diseñado e implementado de forma que sea inmediato realizar los cambios necesarios para que la BD pase a ser remota y centralizada, o para facilitar la transición si se decide utilizar otro sistema gestor de bases de datos relacionales.

(...)

Se han realizado las fases de Elicitación y Especificación de Requisitos. Se han descrito los requisitos funcionales, no funcionales, de interacción y de información. Los requisitos de interacción (especificación textual de los casos de uso y sus escenarios) correspondientes a cuatro casos de uso se incluyen en este documento en la siguiente sección (Sec. 1.2).

A partir de los requisitos, en la fase de análisis se ha realizado el Modelado del Dominio y la realización de casos de uso en análisis (se omite en este enunciado). En el Capítulo 2 se aporta una documentación parcial de análisis que incluye: un fragmento del diagrama de casos de uso y el modelo del dominio.

En el Capítulo 3 se aporta el diseño de la base de datos (diseño lógico y físico), realizado en la primera fase del Diseño a partir del modelado conceptual expresado en los diagramas del modelo de dominio.

## **1.2. Casos de uso: especificación (parcial) de escenarios**

Se especifican parcialmente (solamente 4 casos de uso) los escenarios de interacción con el sistema.

La especificación de casos de uso que aquí se describe no se corresponde exactamente con lo esperado en un caso real. Se han realizado muchas simplificaciones para que no sea necesario trabajar con demasiadas clases de dominio ni tablas en la base de datos.

### **1.2.1. Identificarse**

**Actor:** Empleado

**Caso de Uso:** Identificarse

**Precondición:** El actor no se encuentra identificado en el sistema

**Secuencia principal:**

1. El actor Empleado introduce nif y contraseña.
2. El sistema comprueba que dicho par nif-contraseña corresponden con un empleado en activo y muestra las opciones correspondientes a su rol en el sistema.

**Alternativas y excepciones:**

- ( 2 a ) Si no existe empleado con nif indicado, el sistema muestra un mensaje de error, a continuación el caso de uso queda sin efecto.
- ( 2 b ) Si la contraseña es incorrecta para el empleado con el nif indicado, el sistema muestra un mensaje de error, a continuación el caso de uso queda sin efecto.
- ( 2 c ) Si el empleado no está en activo (según [RN-2], activo es: contratado y trabajando), el sistema muestra un mensaje de error y a continuación el caso de uso queda sin efecto.

**Postcondición:** Escenario de éxito: El empleado está identificado en el sistema y sus opciones de trabajo mostradas.

Aquí va la especificación de los otros 3 casos de uso, cuyo texto estoy revisando en detalle y se irán subiendo.

### **1.2.2. Consultar vehículos de la flota del punto que están situados en otro punto**

**Actor:** Gerente de Punto

**Caso de Uso:** Consultar vehículos de la flota del punto que están situados en otro punto

**Precondición:** El actor Gerente de Punto se encuentra identificado en el sistema

#### **Secuencia principal:**

1. El actor solicita ver los vehículos del punto del que es gerente que no se encuentran en el mismo.
2. El sistema muestra los vehículos de la flota asignada al punto del actor cuya situación es un punto diferente. En particular muestra matrícula, estado, color, nombre del modelo y nombre de la marca de cada vehículo.
3. El actor selecciona un vehículo para ver los detalles del punto en el que se encuentra.
4. El sistema le muestra los detalles del punto en el que se encuentra el vehículo seleccionado (localización, email y teléfono) y el caso de uso vuelve al paso 3.

#### **Alternativas y excepciones:**

- ( 1a ) Si el actor cancela, el caso de uso queda sin efecto.
- ( 2a ) Si el sistema encuentra que no hay ningún vehículo de la flota situado en otro punto, muestra un mensaje informativo y el caso de uso finaliza.
- ( 3a ) Si el actor cancela, el caso de uso queda sin efecto.

**Postcondición:** El sistema muestra los vehículos de la flota asignada al punto del que es gerente el actor que se encuentran situados en otro punto.

**Indicaciones de usabilidad:** el actor podrá elegir al seleccionar un vehículo y ver los detalles del punto en el que se encuentra que los resultados se muestren filtrados por dicho punto, es decir mostrar solamente los vehículos que se encuentran en el mismo punto que el vehículo seleccionado, o volver a ver todos los vehículos sin filtrar.



### 1.2.3. Registrar devolución de vehículo en alquiler

**Actor:** Atención al Cliente de un punto

**Caso de Uso:** Registrar devolución de vehículo en alquiler

**Precondición:** El actor Atención al Cliente de un punto se encuentra identificado en el sistema

**Secuencia principal:**

1. El actor introduce el id de la reserva que identifica el alquiler.
2. El sistema comprueba que hay un alquiler en curso para dicha reserva, comprueba que el punto en el que se encuentra el actor es el punto de entrega previsto en la reserva, muestra los datos del vehículo (al menos la matrícula) y del cliente (al menos el dni), y solicita confirmación al usuario para continuar con la devolución.
3. El actor confirma.
4. El sistema crea la información sobre la devolución (la Entrega) en la fecha y hora actual en el punto, realizada por el empleado de atención al cliente del punto que está identificado.
5. El sistema pregunta si se ha producido alguna incidencia.
6. El actor indica que no se han producido incidencias.
7. El sistema pasa la reserva a “finalizada” y el vehículo a “enPreparación”, indicando que está situado en el punto en el que se está produciendo la entrega.
8. El sistema registra toda la información y el caso de uso finaliza

**Alternativas y excepciones:**

- ( 1a ), ( 3a ), ( 6a ) Si el actor cancela el caso de uso queda sin efecto.
- ( 2a ) Si el sistema comprueba no hay una reserva con ese id, muestra mensaje de error y el caso de uso queda sin efecto.
- ( 2b ) Si el sistema comprueba que la reserva con ese id no se encuentra en estado “enAlquiler”, muestra mensaje de error y el caso de uso queda sin efecto.
- ( 3b ) Si el actor no confirma el caso de uso vuelve al paso 1.
- ( 6b ) Si el actor indica que se ha producido alguna incidencia se ejecuta el caso de uso Añadir incidencias y el caso de uso finaliza.
- ( 8a ) Si ha habido incidencias, el sistema registra toda la información incluida las incidencias creadas y el caso de uso finaliza.

**Postcondición:** Escenario de éxito: Se ha creado la información de la entrega del vehículo (devolución de alquiler), la reserva queda en estado “finalizada” y el vehículo queda en estado “enPreparación”, el vehículo queda situado en el punto en el que está realizando la entrega. Si se ha indicado que hay incidencias, se habrán creado y registrado las incidencias vinculadas a la entrega.

**Nota importante:** Debemos asumir que se satisface por el sistema que si la reserva se encuentra en estado “enAlquiler”, existe un alquiler vinculado con dicha reserva y el vehículo asociado con dicha reserva se encuentra también en estado “enAlquiler”. También asumimos que si el vehículo se encuentra en estado “enAlquiler”, su situación es null (no está situado en ningún punto). Esto quiere decir que asumimos que los datos están bien almacenados porque el caso de uso en el que se realiza la recogida de la reserva y se crea el alquiler se realiza correctamente, respetando todas las restricciones, o porque hemos creado todos los datos de prueba correctamente.

A continuación se describe el caso de uso auxiliar “Añadir incidencias”.

## **Caso de uso** Añadir incidencias

**Precondición:** El actor está identificado en el sistema.  
Hay una entrega (devolución de alquiler) en curso

### **Secuencia principal:**

1. El actor introduce un tipo de incidencia y una descripción de la incidencia.
2. El sistema comprueba que se trata de uno de los Tipos de Incidencia previstos y que la descripción está vacía.
3. El sistema pregunta si hay cargo extra por la incidencia (se asume que el cargo asociado a algunas incidencias es 0 por defecto).
4. El actor indica que no hay cargo (se asume valor 0 por defecto del cargo asociado).
5. El sistema crea la nueva incidencia asociada a la entrega en curso, muestra los datos de la incidencia (junto a los datos de las anteriores incidencias que se hayan introducido), y pregunta si no hay más incidencias que reflejar.
6. El actor indica que no hay más incidencias a reflejar.
7. El caso de uso finaliza

### **Alternativas y excepciones:**

- ( 1a ), ( 4a ), ( 6a ) Si el actor cancela el caso de uso queda sin efecto.
- ( 2a ) Si el sistema comprueba que la descripción está vacía, muestra un mensaje de error y vuelve al paso 1.
- ( 2b ) Si el sistema comprueba que el tipo de incidencia no es uno entre: accidente, avería, retraso en la entrega, entrega en otro punto no previsto, muestra un mensaje de error y vuelve al paso 1.
- ( 6b ) Si el actor indica que quedan incidencias por reflejar, el caso de uso se repite desde el paso 1.

#### 1.2.4. Registrar recepción de vehículo del taller

**Actor:** Técnico del punto

**Caso de Uso:** Registrar recepción de vehículo del taller

**Precondición:** El actor Técnico del punto se encuentra identificado en el sistema

**Secuencia principal:**

1. El actor introduce la matrícula del vehículo.
2. El sistema comprueba que hay un vehículo con esa matrícula en estado “enTaller” y una orden de reparación a la que están asociados los datos de llegada al taller, salida del taller pero no la llegada al punto, muestra los datos del vehículo (matrícula, estado, color), la orden de reparación (fecha, hora, razones), los datos de entrada y salida del taller (fecha y hora), y si está propuesto para baja, la causa, y solicita confirmación del actor.
3. El actor confirma.
4. El sistema comprueba que el vehículo no está propuesto para baja y pasa el vehículo al estado “enPreparación”.
5. El sistema registra la fecha y hora de entrada al punto, actualiza todos los cambios y el caso de uso finaliza.

**Alternativas y excepciones:**

- ( 1a ), ( 3a ) Si el actor cancela, el caso de uso queda sin efecto.
- ( 2a ) Si el sistema comprueba que no hay un vehículo con la matrícula indicada, muestra un mensaje de error y el caso de uso queda sin efecto.
- ( 2b ) Si el sistema comprueba que no hay una orden de reparación del vehículo pendiente de registrar la entrega en el punto, muestra un mensaje de error y el caso de uso queda sin efecto.
- ( 3b ) Si el actor no confirma, el caso de uso vuelve al paso 1.
- ( 4a ) Si el sistema comprueba que el vehículo ha sido propuesto para baja, pasa el vehículo al estado “propuestoParaBaja”.

**Postcondición:** Escenario de éxito: Se ha registrado la llegada del vehículo al punto, en caso de venir del taller propuesto para baja, el vehículo ha pasado al estado “propuestoParaBaja”, en caso de no venir del taller propuesto para baja, el vehículo ha pasado al estado “enPreparación”

**Nota importante:** Se asume que para que el vehículo esté en el estado enTaller, hay una orden de reparación y se ha registrado la llegada al taller. Esto quiere decir que se han realizado bien los correspondientes casos de uso o que se han introducido correctamente los datos de prueba.

## Capítulo 2

# Documentación parcial de Análisis

### 2.1. Vista parcial del diagrama de Casos de Uso

Se aporta una vista parcial del diagrama de Casos de Uso. Para esta práctica se han escogido aquellos que se encuentran resaltados y para los que se adjunta una especificación.

Se han definido distintos actores. Los que vamos a considerar en este trabajo son:

- el actor generalización Empleado
- el actor Gerente de un punto
- el actor Atención al Cliente de un punto
- el actor Técnico de un punto

En la Figura 2.1 se muestra el mencionado fragmento del diagrama de casos de uso. Tal como se aprecia (resaltados) en dicha figura, para este trabajo vamos a tener en cuenta los siguientes casos de uso:

**Empleado** CU Identificarse

**Gerente** CU Consultar vehículos de la flota del punto que están situados en otro punto

**Atención al público** CU Registrar devolución de vehículo en alquiler

**Técnico** CU Registrar recepción de vehículo del taller

Aclaraciones sobre el caso de uso “Identificarse”: El empleado se identifica con su nif y contraseña. Si todo va bien, se carga su perfil con las opciones que puede realizar este tipo de empleado.

Para esta práctica, a cada tipo de actor se le mostrará una lista de opciones. Esta lista debe contener al menos el CU que se pide realizar para ese actor en este supuesto práctico. Para listar el resto de opciones se utilizarán los otros ejemplos de CU que se muestran en la Figura 2.1. La funcionalidad asociada a estas opciones será mostrar un mensaje que informe de “Opción aún no implementada”.

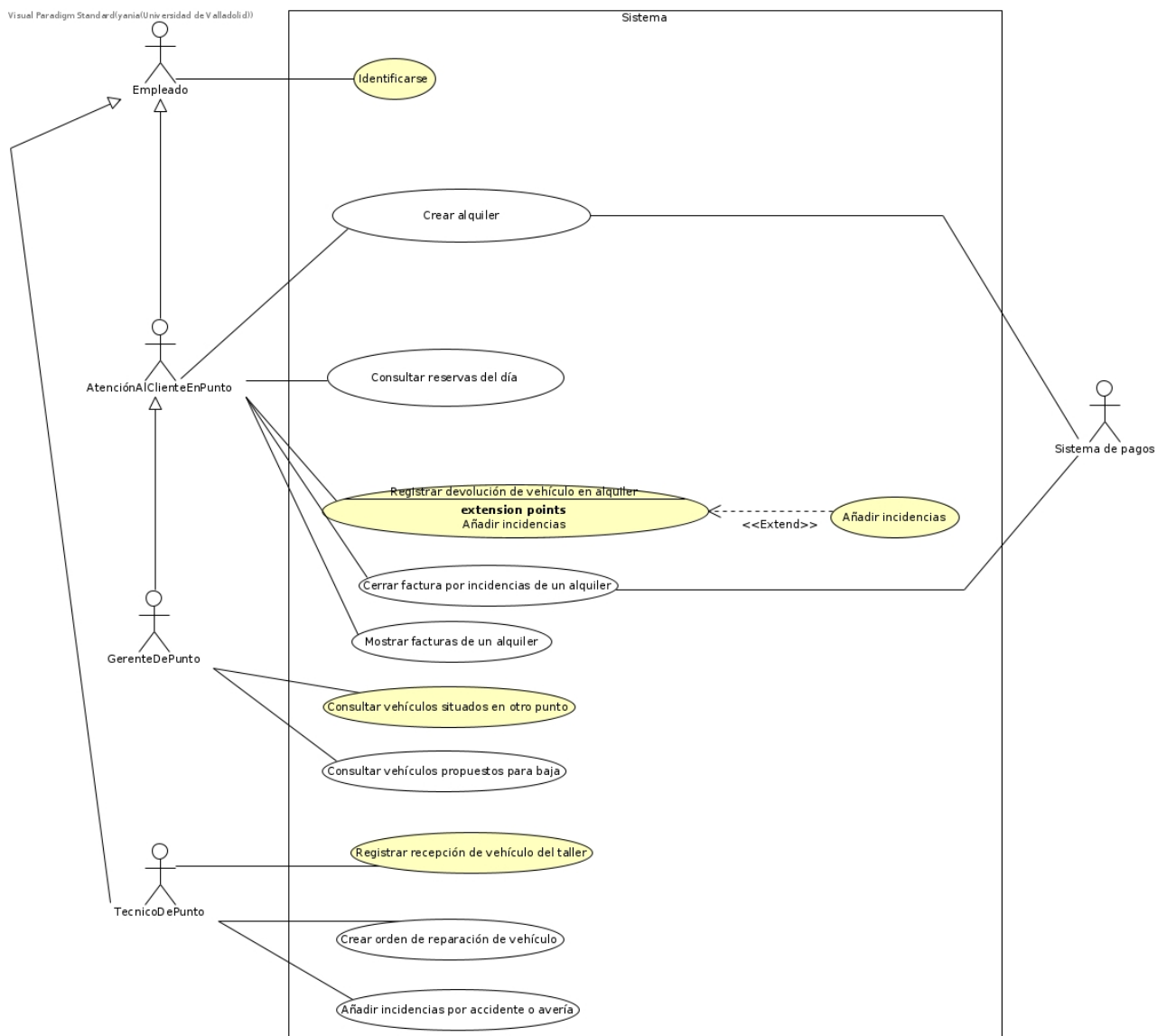


Figura 2.1: Fragmento del diagrama de casos de uso. Se muestran sólo algunos ejemplos por actor.

## **2.2. Modelo del dominio**

En la fase de análisis se ha obtenido el modelo del dominio que se muestra en las Figuras 2.2 y 2.3. En dicho diagrama se ha omitido a propósito la especificación de restricciones OCL y la especificación de los detalles de algunas clases que no participan en ninguno de los casos de uso a diseñar e implementar.

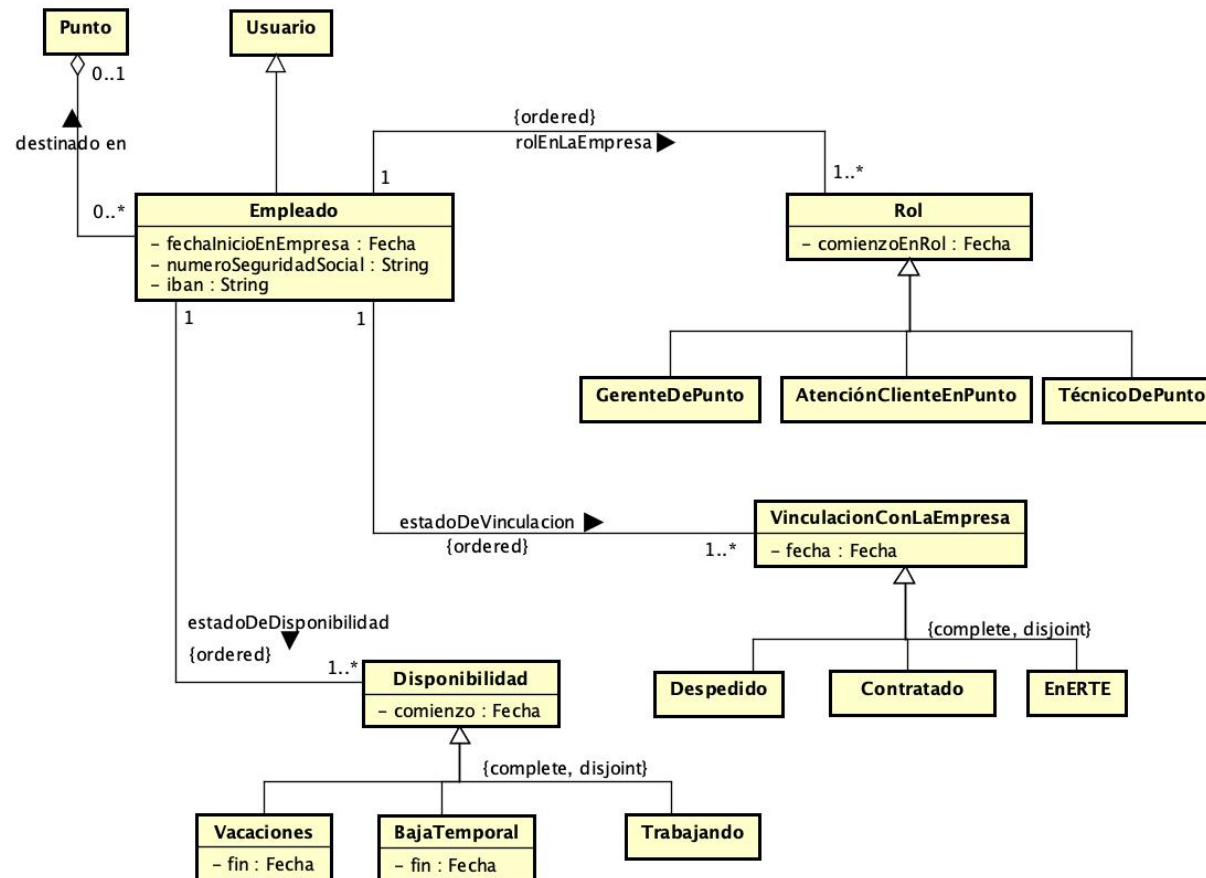


Figura 2.2: Modelo del Dominio relativo a Empleados de un Punto de recogida y entrega de vehículos.

Figura 2.3: Modelo del Dominio



## **2.3. Restricciones a tener en cuenta en la implementación**

El sistema deberá utilizar una base de datos cuyo esquema (diseño lógico) se aporta en las páginas 18 y 19, implementada con Derby. Deberá utilizarse el script de creación de la base de datos (diseño físico) se aporta en la página 20. El sistema será una aplicación de escritorio independiente del sistema operativo que se desarrollará en JAVA, jdk 11 utilizando Swing y JDBC para db-derby-10.14.2.0. El sistema será desarrollado en NETBEANS en la misma versión que se encuentra instalada actualmente en los laboratorios de la Escuela (actualmente Netbeans 12.6).

## **2.4. Análisis arquitectónico**

En el subsistema para el trabajo interno de la empresa se ha decidido aplicar una arquitectura de capas con tres capas estricta y una relajada de servicios útiles a todas las capas, MVC y DAO+DTO como patrón de acceso a datos. El DTO será una cadena JSON.

## Capítulo 3

# Diseño de Datos

### 3.1. Diseño lógico (datamodel style)

En las Figuras 3.1 y 3.2 se muestra el diseño de los datos como esquema relacional (diseño lógico), utilizando una representación basada en UML con estereotipos. Los estereotipos «PK» y «FK» representan claves primarias y claves foráneas, respectivamente. Los clasificadores con el estereotipo «table» representan una tabla en el esquema relacional. Las tablas que representan un elemento del dominio se identifican con el estereotipo «entity». Las tablas que representan asociaciones se identifican con el estereotipo «relation». Las tablas que representan valores enumerados se identifican con el estereotipo «enum» y aparecen sombreadas. La transformación al modelo relacional de las relaciones de herencia presentes en el Modelo del Dominio se ha realizado mediante referencias entre tablas. El modelado de roles temporales presentes en el modelo del dominio para empleados se ha realizado mediante tablas asociación y tablas enum.

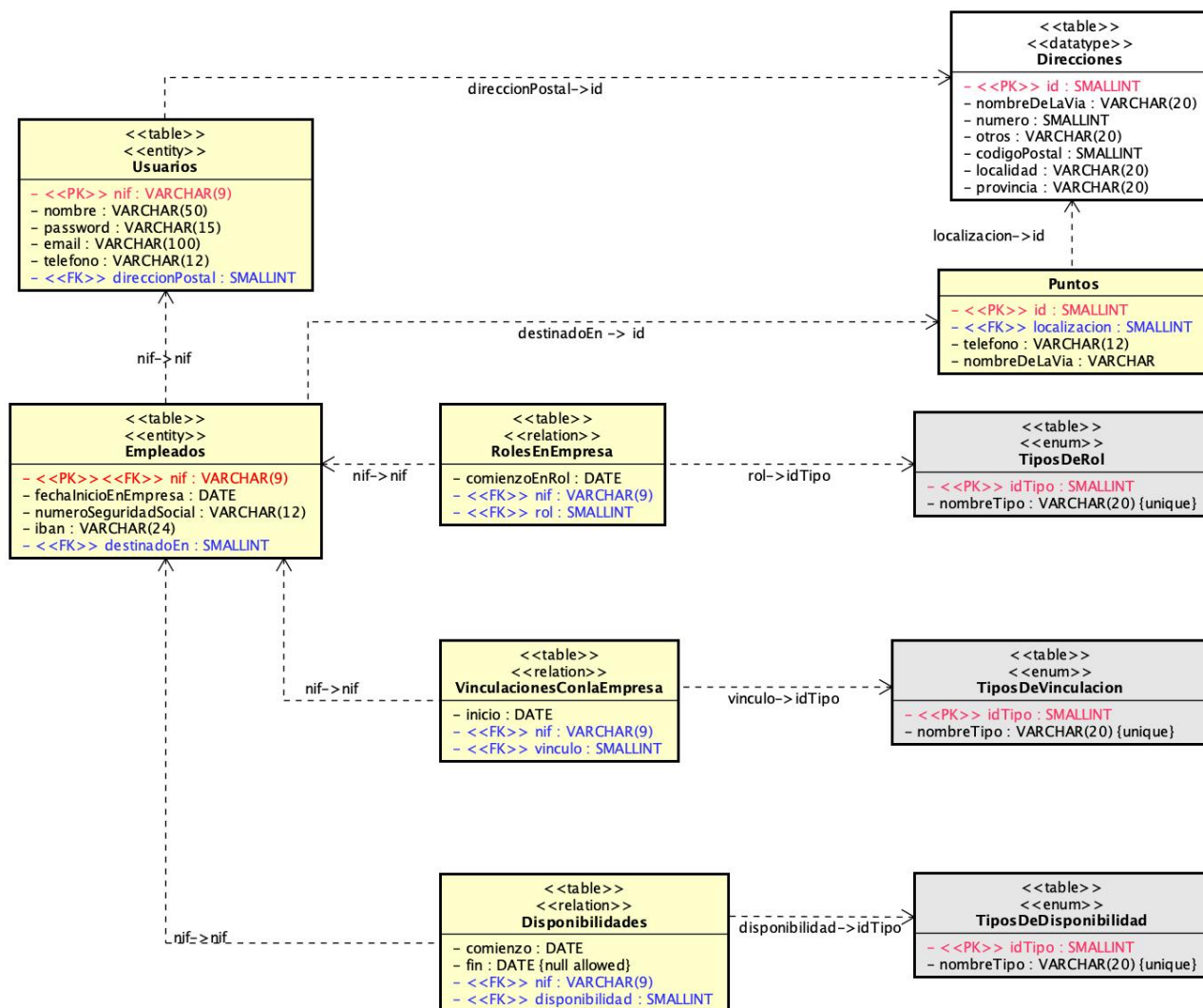


Figura 3.1: Diseño lógico de la base de datos relacional (parte relativa a empleados)

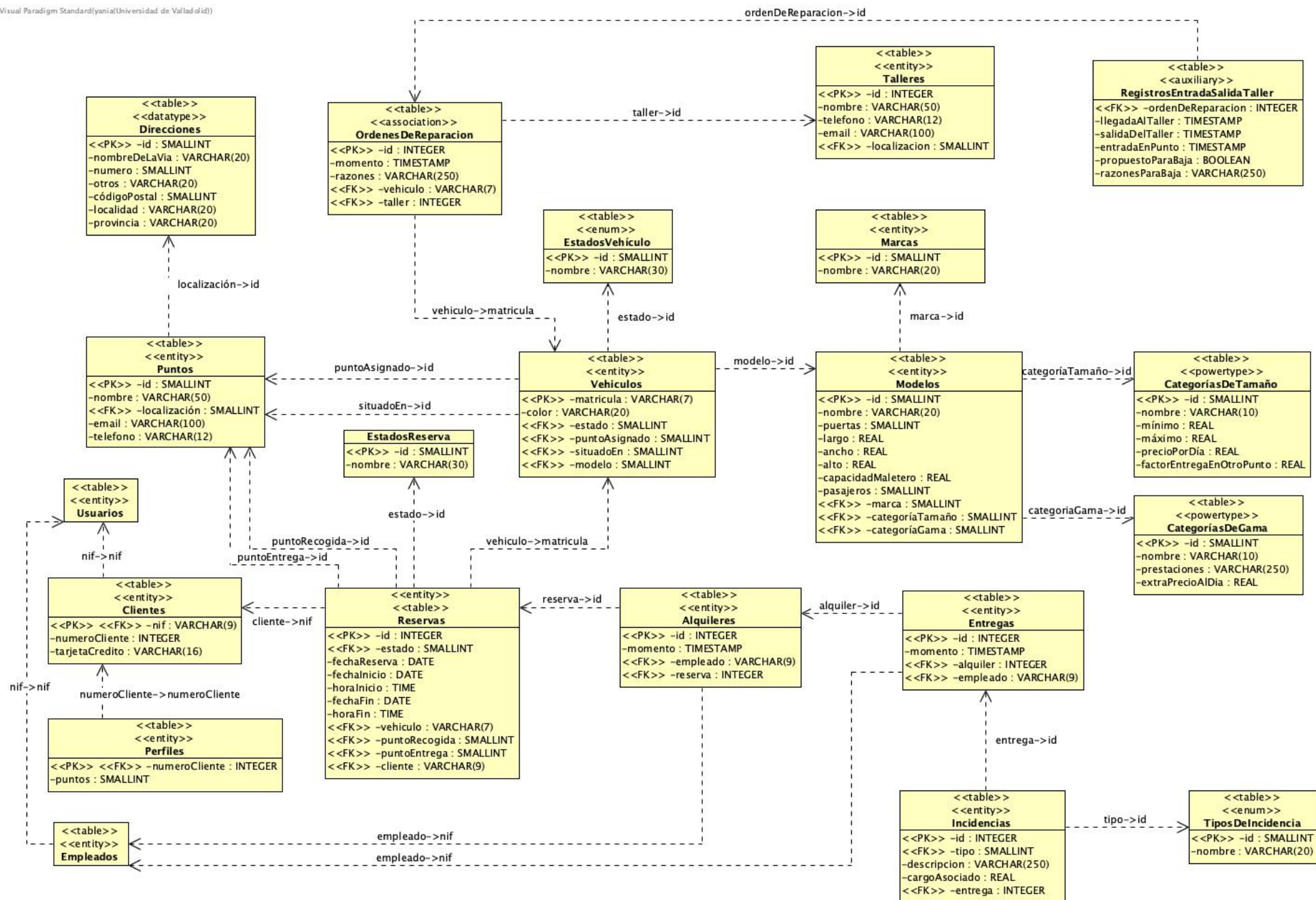


Figura 3.2: Diseño lógico de la base de datos relacional (Fragmento)

## 3.2. Scripts de creación de la base de datos

Este script puede obtenerse en un archivo almacenado como recurso en el aula virtual siguiendo el enlace: [https://campusvirtual.uva.es/pluginfile.php/3705499/mod\\_resource/content/8/createTables.sql](https://campusvirtual.uva.es/pluginfile.php/3705499/mod_resource/content/8/createTables.sql).

```
-- Reset database
--Derby does not support DROP TABLE IF EXISTS

DROP TABLE REGISTROSENTRADASALIDATALLER;
DROP TABLE ORDENESDEREPARACION;
DROP TABLE TALLERES;
DROP TABLE INCIDENCIAS;
DROP TABLE TIPOSDEINCIDENCIA;
DROP TABLE ENTREGAS;
DROP TABLE ALQUILERES;
DROP TABLE RESERVAS;
DROP TABLE ESTADOSRESERVA;
DROP TABLE VEHICULOS;
DROP TABLE MODELOS;
DROP TABLE MARCAS;
DROP TABLE ESTADOSVEHICULO;
DROP TABLE CATEGORIASDETAMANO;
DROP TABLE CATEGORIASDEGAMA;

DROP TABLE DISPONIBILIDADES;
DROP TABLE VINCULACIONESCONLAEMPRESA;
DROP TABLE ROLESENEMPRESA;
DROP TABLE TIPOSDEDISPONIBILIDAD;
DROP TABLE TIPOSDEVINCULACION;
DROP TABLE TIPOSDEROL;

DROP TABLE PERFILES;
DROP TABLE CLIENTES;
DROP TABLE EMPLEADOS;
DROP TABLE USUARIOS;
DROP TABLE PUNTOS;
DROP TABLE DIRECCIONES;

-- Enum
create table TIPOSDEROL
(
    IdTipo SMALLINT not null,
    NombreTipo VARCHAR(20) not null unique,
    PRIMARY KEY(IdTipo)
);

INSERT INTO TIPOSDEROL
VALUES (1, 'GerenteDePunto'),
(2, 'AtencionEnPunto'),
(3, 'TecnicoEnPunto');

-- Enum
create table TIPOSDEVINCULACION
(
    IdTipo SMALLINT not null,
    NombreTipo VARCHAR(20) not null unique,
    PRIMARY KEY(IdTipo)
);
```

```

INSERT INTO TIPOSEVINCULACION
VALUES (1, 'Contratado'),
       (2, 'Despedido'),
       (3, 'EnERTE');

-- Enum
create table TIPOSEDISPONIBILIDAD
(
    IdTipo SMALLINT not null,
    NombreTipo VARCHAR(20) not null unique,
    PRIMARY KEY(IdTipo)
);

INSERT INTO TIPOSEDISPONIBILIDAD
VALUES (1, 'Vacaciones'),
       (2, 'BajaTemporal'),
       (3, 'Trabajando');

-- Datatype
create table DIRECCIONES
(
    Id SMALLINT not null,
    NombreDeLaVia VARCHAR(20) not null,
    Numero SMALLINT,
    Otros VARCHAR(20),
    CodigoPostal SMALLINT not null,
    Localidad VARCHAR(20) not null,
    Provincia VARCHAR(20) not null,
    PRIMARY KEY(Id)
);

-- Entity
create table PUNTOS
(
    Id SMALLINT not null primary key,
    Nombre VARCHAR(50) not null,
    Email VARCHAR(100) not null,
    Telefono VARCHAR(12) not null,
    Localizacion SMALLINT not null,
    FOREIGN KEY(Localizacion) REFERENCES DIRECCIONES(Id)
);

-- Entity
create table USUARIOS
(
    Nif VARCHAR(9) not null primary key,
    Nombre VARCHAR(50) not null,
    Password VARCHAR(15) not null,
    Email VARCHAR(100) not null,
    Telefono VARCHAR(12) not null,
    DireccionPostal SMALLINT not null,
    FOREIGN KEY(DireccionPostal) REFERENCES DIRECCIONES(Id)
);

-- Entity
create table EMPLEADOS
(
    Nif VARCHAR(9) not null primary key,
    NumeroSeguridadSocial VARCHAR(12) not null,
    Iban VARCHAR(24) not null,
    FechaInicioEnEmpresa DATE not null,
    DestinadoEn SMALLINT not null,
    FOREIGN KEY(Nif) REFERENCES USUARIOS(Nif),
    FOREIGN KEY(DestinadoEn) REFERENCES PUNTOS(Id)
);

```

```

-- Association
create table ROLESNEMPRESA
(
    ComienzoEnRol DATE not null ,
    Empleado VARCHAR(9) not null ,
    Rol SMALLINT not null ,
    FOREIGN KEY(Empleado) REFERENCES EMPLEADOS(Nif) ,
    FOREIGN KEY(Rol) REFERENCES TIPOSDEROL(IdTipo)
);

-- Association
create table VINCULACIONESCONLAEMPRESA
(
    inicio DATE not null ,
    Empleado VARCHAR(9) not null ,
    Vinculo SMALLINT not null ,
    FOREIGN KEY(Empleado) REFERENCES EMPLEADOS(Nif) ,
    FOREIGN KEY(Vinculo) REFERENCES TIPOSDEVINCULACION(IdTipo)
);

-- Association
create table DISPONIBILIDADES
(
    Comienzo DATE not null ,
    FinalPrevisto DATE ,
    Empleado VARCHAR(9) not null ,
    Disponibilidad SMALLINT not null ,
    FOREIGN KEY(Empleado) REFERENCES EMPLEADOS(Nif) ,
    FOREIGN KEY(Disponibilidad) REFERENCES TIPOSDEDISPONIBILIDAD(IdTipo)
);

-- Entity
create table CLIENTES
(
    Nif VARCHAR(9) not null ,
    NumeroCliente INTEGER not null unique ,
    TarjetaCredito VARCHAR(16) not null ,
    PRIMARY KEY(Nif) ,
    FOREIGN KEY(Nif) REFERENCES USUARIOS(Nif)
);

-- Entity
create table PERFILES
(
    NumeroCliente INTEGER not null ,
    Puntos SMALLINT not null ,
    PRIMARY KEY(NumeroCliente) ,
    FOREIGN KEY(NumeroCliente) REFERENCES CLIENTES(NumeroCliente)
);

-- Powertype
create table CATEGORIASDEGAMA
(
    Id SMALLINT not null ,
    Nombre VARCHAR(10) not null unique ,
    Prestaciones VARCHAR(250) not null ,
    ExtraPrecioAlDia REAL not null ,
    PRIMARY KEY(Id)
);

INSERT INTO CATEGORIASDEGAMA
VALUES (1, 'Media', 'Una descripci n de gama media', 0),
(2, 'Alta', 'Una descripci n de gama alta', 10);

-- Powertype
create table CATEGORIASDETAMANO

```

```

(
    Id SMALLINT not null ,
    Nombre VARCHAR(10) not null unique ,
    Minimo REAL not null ,
    Maximo REAL not null ,
    PrecioPorDia REAL not null ,
    FactorEntregaEnOtroPunto REAL not null ,
    PRIMARY KEY(Id)
);

INSERT INTO CATEGORIASDETAMANO
VALUES (1, 'Peque o', 2.4, 3.5, 10, 1.3),
        (2, 'Mediano', 3.6, 4.5, 15, 1.7),
        (3, 'Grande', 4.6, 6, 20, 2.0);

-- Enum
create table ESTADOSVEHICULO
(
    Id SMALLINT not null ,
    Nombre VARCHAR(30) not null unique ,
    PRIMARY KEY(Id)
);

INSERT INTO ESTADOSVEHICULO
VALUES (1, 'Disponible'),
        (2, 'EnAlquiler'),
        (3, 'EnReserva'),
        (4, 'EnTaller'),
        (5, 'Averiado'),
        (6, 'EnPreparaci n'),
        (7, 'PropuestoParaBaja');

-- Entity
create table MARCAS
(
    Id SMALLINT not null ,
    Nombre VARCHAR(20) not null unique ,
    PRIMARY KEY(Id)
);

-- Entity
create table MODELOS
(
    Id SMALLINT not null ,
    Nombre VARCHAR(20) not null unique ,
    Puertas SMALLINT not null ,
    Largo REAL not null ,
    Ancho REAL not null ,
    Alto REAL not null ,
    CapacidadMaletero REAL not null ,
    Pasajeros SMALLINT not null ,
    Marca SMALLINT not null ,
    CategoriaTamano SMALLINT not null ,
    CategoriaGama SMALLINT not null ,
    PRIMARY KEY(Id) ,
    FOREIGN KEY(Marca) REFERENCES MARCAS(Id) ,
    FOREIGN KEY(CategoriaTamano) REFERENCES CATEGORIASDETAMANO(Id) ,
    FOREIGN KEY(CategoriaGama) REFERENCES CATEGORIASDEGAMA(Id)
);

-- Entity
create table VEHICULOS
(
    Matricula VARCHAR(7) not null ,
    Color VARCHAR(20) not null ,
    Estado SMALLINT not null ,

```



```

        PuntoAsignado SMALLINT not null ,
        SituadoEn SMALLINT ,
        Modelo SMALLINT not null ,
            PRIMARY KEY( Matricula ) ,
            FOREIGN KEY( Estado ) REFERENCES ESTADOSVEHICULO ,
            FOREIGN KEY( PuntoAsignado ) REFERENCES PUNTOS( Id ) ,
            FOREIGN KEY( SituadoEn ) REFERENCES PUNTOS( Id ) ,
            FOREIGN KEY( Modelo ) REFERENCES MODELOS( Id )
    );

-- Enum
create table ESTADOSRESERVA
(
    Id SMALLINT not null ,
    Nombre VARCHAR(30) not null unique ,
        PRIMARY KEY( Id )
);

INSERT INTO ESTADOSRESERVA
VALUES (1, 'Solicitada' ) ,
        (2, 'EnFirme' ) ,
        (3, 'Anulada' ) ,
        (4, 'Facturada' ) ,
        (5, 'EnAlquiler' ) ,
        (6, 'Finalizada' );

-- Entity
create table RESERVAS
(
    Id INTEGER not null ,
    Estado SMALLINT not null ,
    FechaReserva DATE not null ,
    FechaInicio DATE not null ,
    HoraInicio TIME not null ,
    HechaFin DATE not null ,
    HoraFin TIME not null ,
    Vehiculo VARCHAR(7) not null ,
    PuntoRecogida SMALLINT not null ,
    PuntoEntrega SMALLINT not null ,
    Cliente VARCHAR(9) not null ,
        PRIMARY KEY( Id ) ,
        FOREIGN KEY( Estado ) REFERENCES ESTADOSRESERVA( Id ) ,
        FOREIGN KEY( Vehiculo ) REFERENCES VEHICULOS( Matricula ) ,
        FOREIGN KEY( PuntoRecogida ) REFERENCES PUNTOS( Id ) ,
        FOREIGN KEY( PuntoEntrega ) REFERENCES PUNTOS( Id ) ,
        FOREIGN KEY( Cliente ) REFERENCES CLIENTES( Nif )
);

-- Entity. Transaction.
create table ALQUILERES
(
    Id INTEGER not null ,
    Momento TIMESTAMP not null ,
    Empleado VARCHAR(9) not null ,
    Reserva INTEGER not null ,
        PRIMARY KEY( Id ) ,
        FOREIGN KEY( Empleado ) REFERENCES EMPLEADOS( Nif ) ,
        FOREIGN KEY( Reserva ) REFERENCES RESERVAS( Id )
);

-- Entity. Transaction.
create table ENTREGAS
(
    Id INTEGER not null ,

```

```

        Momento TIMESTAMP not null ,
        Alquiler INTEGER not null ,
        Empleado VARCHAR(9) not null ,
                PRIMARY KEY(Id) ,
                FOREIGN KEY(Alquiler) REFERENCES ALQUILERES(Id) ,
                FOREIGN KEY(Empleado) REFERENCES EMPLEADOS(Nif)
);

-- Enum
create table TIPOSDEINCIDENCIA
(
        Id SMALLINT not null ,
        Nombre VARCHAR(20) not null unique ,
                PRIMARY KEY(Id)
);

INSERT INTO TIPOSDEINCIDENCIA
VALUES (1, 'Accidente' ) ,
        (2, 'Aver a ' ) ,
        (3, 'SuciedadExtrema' ) ,
        (4, 'RetrasoEnLaEntrega' );

-- Entity
create table INCIDENCIAS
(
        Id INTEGER not null ,
        Tipo SMALLINT not null ,
        Descripcion VARCHAR(250) not null ,
        CargoAsociado REAL ,
        Entrega INTEGER not null ,
                PRIMARY KEY(Id) ,
                FOREIGN KEY(Tipo) REFERENCES TIPOSDEINCIDENCIA(Id) ,
                FOREIGN KEY(Entrega) REFERENCES ENTREGAS(Id)
);

-- Entity
create table TALLERES
(
        Id INTEGER not null ,
        Nombre VARCHAR(50) not null ,
        Telefono VARCHAR(12) not null ,
        Email VARCHAR(100) not null ,
        Localizacion SMALLINT not null ,
                PRIMARY KEY(Id) ,
                FOREIGN KEY(Localizacion) REFERENCES DIRECCIONES(Id)
);

-- Association + Data
create table ORDENESDEREPARACION
(
        Id INTEGER not null ,
        Momento TIMESTAMP not null ,
        Razones VARCHAR(250) not null ,
        Vehiculo VARCHAR(7) not null ,
        Taller INTEGER not null ,
                PRIMARY KEY(Id) ,
                FOREIGN KEY(Vehiculo) REFERENCES VEHICULOS(Matricula) ,
                FOREIGN KEY(Taller) REFERENCES TALLERES(Id)
);

-- Auxiliary
create table REGISTROSENTRADASALIDATALLER
(
        OrdenDeReparacion INTEGER not null ,
        LlegadaAlTaller TIMESTAMP ,
        SalidaDelTaller TIMESTAMP ,

```

```
EntradaEnPunto TIMESTAMP,  
PropuestoParaBaja BOOLEAN,  
RazonesParaBaja VARCHAR(250),  
    FOREIGN KEY(OrdenDeReparacion) REFERENCES ORDENESDEREPARACION( Id )  
);
```

## Capítulo 4

# Normas de entrega y criterios de evaluación

### 4.1. Indicaciones para la entrega

El seguimiento del proyecto se realizará a través de la aplicación PIVOTAL TRACKER:

<https://www.pivotaltracker.com>.

Todos los estudiantes se habrán creado una cuenta en PIVOTAL TRACKER y habrán sido añadidos al proyecto correspondiente según su equipo de trabajo. El estudiante deberá estar identificado con su correo institucional (estudiantes.uva.es) y su identificación como login en los laboratorios de la Escuela.

La comunicación entre el equipo, y el equipo con los profesores, puede hacerse mediante PIVOTAL TRACKER.

Para realizar la entrega se preparará una *release* en PIVOTAL TRACKER. En dicha *release* se indicará la url de un repositorio en el GITLAB de la Escuela. El contenido del repositorio debe cumplir estrictamente la estructura y contenidos que se especifican en el apartado 4.2).

Para esto último deberán seguirse las siguientes normas:

- La entrega se realizará añadiendo, con permisos de tipo *Reporter*, a los profesores (usuario *yania*, en todo caso, y usuarios *marga/marcorr*, según corresponda) al repositorio que contiene el proyecto en el GITLAB de la Escuela cuando ya no se vaya a realizar ningún *commit & push*. Cualquier *push* al repositorio una vez vencido el plazo de entrega será penalizado con 0 en la Práctica en la convocatoria correspondiente.
- El enlace (url) al repositorio y cualquier documentación necesaria al respecto serán anotadas en la *release* en PIVOTAL TRACKER.
- La versión final correspondiente a la release deberá estar integrada en la rama master.

### 4.2. Estructura y contenidos de la entrega

La entrega tendrá la siguiente estructura de carpetas:

```
models/  
app/
```

En la carpeta `models` se encontrará un archivo ASTAH PROFESSIONAL o VISUAL PARADIGM (a elección del equipo). Los modelos y diagramas contenidos en dicho archivo ASTAH PROFESSIONAL o VISUAL PARADIGM que son evaluables corresponden a lo realizado en la fase de diseño. Se tendrá cuidado de alojar todo lo realizado en un modelo llamado Diseño organizado con sus respectivos submodelos y diagramas.

Se contará con los diagramas necesarios para representar la arquitectura de referencia, los estilos universales *decomposition style*, *uses style*, *inheritance style* y *data model* así como los

diagramas de clases de diseño detallado y los diagramas de secuencia con la realización en diseño de los casos de uso que se piden en este enunciado. Adicionalmente, se aportará un diagrama de estados para modelar la interfaz de usuario del sistema (en el que se modelará únicamente lo necesario para los casos de uso que se especifican en este documento).

Los diagramas tendrán que ser legibles y comprensibles. Los diagramas no pueden ser excesivamente grandes por lo que se deberán utilizar los elementos que ofrece UML para reducir el tamaño y la complejidad de los diagramas.

En la carpeta `app` se espera una estructura como la siguiente:

```
app/netbeansProject/  
app/db/
```

La carpeta `app/netbeansProject` contendrá, como su nombre indica, el proyecto NETBEANS que implementa el caso de uso. La implementación no debe incumplir el diseño propuesto. El proyecto de la implementación deberá estar gestionado con maven.

La carpeta `app/db` contendrá un archivo de propiedades llamado `config.db` con lo necesario para conectarse con la BD. Este archivo indicará la url con el puerto, el nombre de la base de datos, el usuario y el password exactamente como se indica a continuación:

```
url=jdbc:derby://localhost:1527/pucelacar  
user=adminpucelacar  
password=pucelacar2023
```

En dicha carpeta (`app/db`) también se encontrarán unos scripts que permitirán la regeneración de la base de datos (el suministrado con este enunciado para crear las tablas) así como tantos scripts SQL como sean necesarios para poblar automáticamente la base de datos de cara a probar la aplicación (puede ser uno o varios separados por casos de uso o escenarios) siempre que se documente apropiadamente el propósito de cada uno.

En el control de versiones, y por tanto en los archivos de entrega, no debe residir la base de datos Derby como tal. Tampoco deberá encontrarse ninguna librería puesto que la gestión de dependencias será realizada por maven. El `.gitignore` deberá indicar target para que ningún binario que se pueda volver a generar resida en el repositorio.

### Fecha límite para la entrega

A continuación se especifican las **fechas límite de entrega para cada convocatoria**. Esta será la fecha de la *release* en PIVOTAL TRACKER. Si no se cumple la fecha límite, el equipo será penalizado con 0 en la Práctica en la convocatoria correspondiente.

**convocatoria ordinaria:** 28 de mayo de 2023

**convocatoria extraordinaria:** 29 de junio de 2023

Será obligatorio realizar una presentación y defensa de la práctica por parte de todo el equipo (ver Sección 4.3.2 para hitos en el desarrollo y la defensa del proyecto).

## 4.3. Criterios de Evaluación

La evaluación de la práctica tiene un peso total del 45 % de la asignatura.

Respecto del proyecto entregado se valorará:

- (a) la aplicación y consistencia en el diseño de la arquitectura de 3 capas (capas estrictas) combinada con MVC, se puede considerar además añadir una capa transversal de servicios (capa relajada);
- (b) la aplicación de los patrones GRASP y algunos patrones de diseño conocidos;

(c) la corrección y completitud de los modelos UML.

(d) la calidad de la solución.

Los criterios anteriores tendrán el mayor peso en la evaluación de la práctica (70 %) y se desglosan de la siguiente forma:

- diagramas de la arquitectura de referencia y descomposición modular- 0,5/10
- diagramas de dependencias entre capas - 0,4/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Identificarse? 0,1/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Registrar devolución de vehículo en alquiler? - 0,1/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Registrar recepción de vehículo del taller? - 0,1/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Consultar vehículos de la flota del punto que están situados en otro punto? - 0,1/10
- diagramas de clases de diseño detallado - 0,5/10
- diagramas de secuencia con la Realización en Diseño de los CU-
  - ¿Se diseña adecuadamente la realización del CU Identificarse? - 1/10
  - ¿Se diseña adecuadamente la realización del CU Registrar devolución de vehículo en alquiler?- 1,5/10
  - ¿Se diseña adecuadamente la realización del CU Registrar recepción de vehículo del taller? - 1,5/10
  - ¿Se diseña adecuadamente la realización del CU Consultar vehículos de la flota del punto que están situados en otro punto? - 1,5/10
- diagrama de estados que modela la interfaz - 0,1/10

Respecto del código de la aplicación (30 %) se valorará:

- (e) que sea consistente con el diseño arquitectónico (1/10);
- (f) que sea consistente con el modelo dinámico diseñado (es decir, con los diagramas de secuencia que describen la realización en diseño de los casos de uso) (1/10);
- (g) que se comporte según lo esperado en las situaciones válidas y que no acepte situaciones inválidas comunicando al usuario los errores que se controlan (pruebas de aceptación superadas) (1/10).

La forma de uso de PIVOTAL TRACKER no será evaluable, pero el uso será obligatorio. En caso de no utilizarse, la práctica no podrá ser considerada entregada ya que las *releases* parciales y la *release* final se realizan por esa vía.

Las releases parciales no son evaluables. Solamente marcan el ritmo del proyecto.

El uso de rocket no será obligatorio, aunque es recomendable. Se bonificará la participación de los estudiantes en el canal de la asignatura <https://rocket.inf.uva.es/group/ds22-23>, tanto en relación con la parte de teoría como con la parte práctica. Los estudiantes realizarán preguntas, podrán aportar contribuciones a las preguntas de los compañeros. Tanto los compañeros como los profesores podrán “premiar” la participación con una reacción. La cantidad y calidad de la participación en el canal será valorada con hasta 0,5 puntos.

Adicionalmente es importante el uso de este canal para notificar de algún error u omisión detectado en el enunciado, los modelos, el script sql para la creación de la BD, etc..

La calidad y la seguridad del código de la aplicación será analizada mediante SonarQube como se explica en el campus virtual (<https://campusvirtual.uva.es/course/view.php?id=44324&section=17#tabs-tree-start>).

Los proyectos con buenos indicadores de calidad y seguridad, detectados por esta herramienta, podrán optar a un premio.

El premio a repartir son puntos en la asignatura. Se otorgará 0,9, 0,6 y 0,3 como premio al primero, segundo, y tercer lugar en el concurso, respectivamente. No habrá empates en ninguna de las tres posiciones del concurso. Al sumar los premios obtenidos en la nota de los ganadores, no podrá sobrepasarse los 10 puntos como nota global de la asignatura.

En ningún caso la nota de la asignatura podrá resultar mayor de 10.

#### 4.3.1. Reglas del Concurso

Se elaborará un ranking entre los equipos participantes.

Para entrar en el ranking se requiere:

- pasar el QualityGate DS definido en el servidor Sonarqube (marca verde Passed por el contrario de marca roja Failed). Este QualityGate se puede consultar en: [https://sonarqube.inf.uva.es/quality\\_gates/show/2](https://sonarqube.inf.uva.es/quality_gates/show/2).
- en las fases iniciales del concurso no se tendrá en cuenta la restricción anterior.

Una vez dentro del ranking se establecerá un orden inverso (de mayor a menor puntuación, es decir, menos puntos mejor posición en el ranking), asignando una puntuación a cada equipo:  $RDT + DD + VRE + BRE$

Donde *RDT* es la ratio de deuda técnica acumulada, *DD* es la densidad de duplicados, *VRE* es la tasa de esfuerzo para remediar las vulnerabilidades y *BRE* es la tasa de esfuerzo para remediar bugs potenciales. Todos estos indicadores se pueden consultar en Sonarqube.

Los empates se resolverán aplicando los siguientes criterios en este orden:

1. el que tenga menor ratio de deuda técnica,
2. el que tenga menor severidad de los code smells, es decir, el que tenga menor la ratio (blocking+major)/violations
3. el que tenga menor porcentaje de código duplicado,
4. el que tenga menos bugs potenciales
5. el que tenga menos vulnerabilidades
6. el que tenga menor complejidad ciclomática (Cyclomatic complexity)
7. el que tenga menor complejidad cognitiva (Cognitive complexity)
8. finalmente el que tenga mayor densidad de comentarios. Los comentarios deberán ser todos de tipo javadoc. Si bien los comentarios javadoc no son evaluables en esta práctica, todos sabemos lo importante que son para la mantenibilidad (principio de autodocumentación). Al no permitir otro tipo de comentarios en el código se aplica el principio: Si tienes necesidad de escribir un comentario escribe una función.

Las clases que representan las Vistas (en el patrón MVC) no participan en el concurso. Estas clases, además de tener una gran parte de código automáticamente generado, tienen una herencia de nivel 5 que no podemos quitarnos porque forma parte de `javax.swing`. En el campus virtual se explica cómo excluirlas del análisis automatizado.

**Regla fundamental:**

- haber implementado toda la funcionalidad especificada en los casos de uso de este enunciado.

El incumplimiento de la regla fundamental será motivo de exclusión del ranking.

Debido a las reglas definidas, el resultado del concurso deberá ser refrendado por una revisión manual de los profesores. Se ejecutarán de nuevo los análisis de todos los proyectos para garantizar que se han analizado en las mismas condiciones. Se revisará la implementación para asegurar que se implementa correctamente toda la funcionalidad especificada (incluidas alternativas y excepciones).

**4.3.2. Hitos en la realización del proyecto**

A continuación se indican los hitos que se han definido para el seguimiento y realización de este proyecto.

- 16/4 - Release 1. Comienza el concurso con el código implementado para el Caso de Uso Identificarse.
- 30/4 - Release 2.
- 14/5 - Release 3.
- 28/5 - Release 4. Se cierra el concurso. Entrega de la práctica en convocatoria ordinaria.
- Desde la entrega final de la práctica hasta el 16 de junio los equipos podrán realizar las defensas, previa concertación de una fecha y hora con los profesores.

Las fechas se han definido de forma holgada. Desde el 27 de marzo, considerado fecha de inicio del proyecto, hasta el 16 de abril, hay tiempo más que suficiente para haber conseguido la Release 1 y adelantar tareas necesarias para conseguir la Release 2. Los hitos son puntos de control límite.

En la defensa, el equipo presentará la demostración de la implementación desarrollada utilizando sus pruebas preparadas para ello, los diagramas de diseño realizados y la correspondencia entre el diseño presentado y el código implementado. El equipo es libre de decidir la organización, quién presenta qué parte. Asimismo, el equipo contestará a las preguntas que se les realicen. Para la defensa no se prepara una presentación con diapositivas. Se utilizará en todo momento Astah/Visual Paradigm, según sea el caso, y Netbeans. Es necesario que todos los miembros del equipo participen en la defensa.

En convocatoria extraordinaria se podrá entregar la práctica hasta el día del examen que es el 29 de junio. Hasta el 3 de julio los equipos podrán realizar las defensas, previa concertación de una fecha y hora con los profesores.