

Boletín de ejercicios resueltos Tema 4

DIAGRAMA DE CLASES Y CASOS DE USO

ANTONIO GARRIDO; ELISEO MARZAL; SOLEDAD VALERO; JOAN
JOSEP FONS CORS



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica
Superior de Ingeniería
Informática

Boletín de ejercicios resueltos Tema 4

Contenido

NOTA IMPORTANTE	2
1. Gestión tareas a través de Web.....	2
2. Gestión proyectos investigación.....	4
3. Comisión Nacional de Emergencias	10
4. Aplicación “MensajeYa”	15
5. Gestión de una Clínica	19
6. Secretaría Virtual.....	22
7. Gestión biblioteca universidad	24
8. Hoteles Morfeo (ejercicio del boletín).....	27
9. El portal ISGOO (ejercicio del boletín)	30
10. Editor colaborativo (ejercicio del boletín)	33
11. Sistema de apoyo a la docencia (ejercicio del boletín)	36
12. Banquetes Spress (ejercicio del boletín).....	39
13. Gestión de una piscina.....	42
14. Gestión de un AMPA.....	45
15. Gestión del Servicio Técnico de un Concesionario.....	48
16. La Fábrica ACERUPV	51
17. Servicio de Alertas Medioambientales	52

NOTA IMPORTANTE

En este boletín se ofrecen resultados de ejercicios de modelado que plantean una solución a una colección de problemas. Para cada ejercicio se ofrece un modelo solución, pero dicha solución no es única y, muy posiblemente, existirán otras soluciones que abordarán el mismo problema de forma distinta. Hay que tener en cuenta que estas soluciones no necesariamente serán extrapolables a otros problemas, pues un pequeño cambio en el problema puede cambiar sustancialmente el modelo resultante.

1. Gestión tareas a través de Web

Queremos desarrollar un sistema de gestión de tareas similar al que define el poliformat para su funcionamiento a través de la Web.

El sistema a desarrollar debe permitir trabajar con distintas asignaturas, que están organizadas en grupos de teoría, laboratorio y seminario. De cada asignatura nos interesa conocer su nombre y código. Para cada grupo necesitaremos un código y su horario. Cada asignatura y grupo cuenta con muchos alumnos matriculados, pero, en una asignatura dada, un alumno solo puede estar en un grupo de teoría, uno de laboratorio y uno de seminario, respectivamente. Los profesores de una asignatura pueden definir tantas tareas como deseen y pueden decidir para qué grupo(s) estarán disponibles. También nos interesa conocer qué profesor(a) es el responsable de una asignatura.

Cada tarea tiene un identificador, título, condiciones de evaluación, fecha tope de entrega, profesor que la crea y, posiblemente, un anexo en forma de archivo .pdf inferior a 5Mb. Obviamente, solo un profesor de la asignatura puede crear una tarea en dicha asignatura y un alumno solo verá las tareas de su grupo y asignatura en la que esté matriculado.

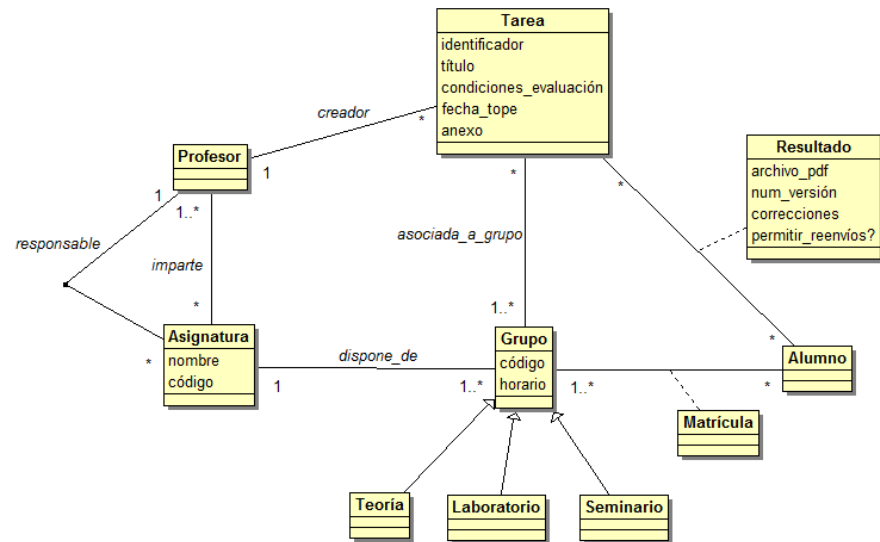
Una vez definida una tarea, y siempre antes de la fecha tope de entrega, cada alumno deberá subir sus resultados a la tarea, adjuntando un archivo .pdf. Una vez subido el resultado se debe enviar un mail de confirmación automáticamente, tanto al alumno como al profesor responsable de la asignatura.

Finalmente, el profesor que creó la tarea calificará la tarea de cada alumno y podrá escribir una serie de correcciones en la misma, que serán enviadas automáticamente al alumno. Si el profesor lo estima oportuno, puede permitir que el alumno suba una versión modificada de sus resultados que será evaluada de nuevo.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

- R1. En una misma asignatura, un alumno sólo puede estar en un grupo de teoría, uno de laboratorio y uno de seminario, respectivamente
- R2. Sólo un profesor de la asignatura puede crear una tarea en dicha asignatura



2. Gestión proyectos investigación

Queremos desarrollar un sistema de gestión de proyectos de investigación y sus actividades para su manejo a través de la Web.

El sistema a desarrollar debe permitir trabajar con distintos proyectos, que están organizados en proyectos de alto riesgo y proyectos de I+D. De cada uno de estos proyectos nos interesa conocer su nombre, duración (siempre inferior a tres años), descripción y temáticas asociadas. Asimismo, nos interesa mantener la persona responsable del proyecto que, para el caso de un proyecto de alto riesgo es única, y para un proyecto de I+D puede ser una o dos personas. Cada proyecto tiene además un equipo de personas trabajando en él. Para focalizar más el trabajo de cada persona solo se puede estar involucrado en cuatro proyectos como máximo, independientemente de su tipo. Además de la información básica sobre una persona (nombre, apellidos y teléfono) nos interesa conocer su antigüedad.

La persona (o personas responsables) de un proyecto se encargarán de la redacción de la memoria del proyecto. Como la creación de la memoria es un proceso iterativo nos interesa almacenar todas las versiones de la misma y cuál es la versión definitiva. Una vez evaluado el proyecto se debe añadir el informe de evaluación, tanto si es favorable como desfavorable; en el caso de ser favorable el proyecto se ha aceptado y nos interesa conocer también el importe de financiación.

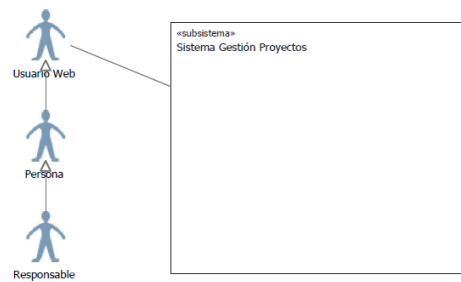
Cualquier persona que trabaja en un proyecto puede añadir actividades al proyecto, como por ejemplo artículos, ponencias, charlas informales, etc. Un proyecto puede tener muchas actividades, pero una actividad es solo de un proyecto. Cada actividad tiene un nombre, una descripción y una fecha de realización. Obviamente, nos interesa conocer qué personas han estado involucradas en dicha actividad ya que el sistema enviará una notificación automática a las personas involucradas, que deberán validar la actividad (también nos interesa conocer cuándo la han validado). Una vez validada dicha actividad por todas las personas involucradas se enviará una notificación automática a los responsables del proyecto de dicha actividad que podrán realizar correcciones si lo consideran oportuno. Finalmente, cualquiera de los responsables de dicho proyecto validará la actividad para que esté disponible a través de la Web.

Se pretende dar la máxima difusión a través de la Web, por lo que se necesita ofrecer públicamente la información de todos los proyectos que se están realizando con su(s) responsable(s). Adicionalmente, para cada proyecto se debe ofrecer el conjunto de actividades que se han realizado (siempre y cuando hayan sido validadas por el responsable) y las personas involucradas en dichas actividades. También a través de la Web se debe ofrecer en formato PDF, pero esta vez solo a los responsables de cada proyecto, un resumen de todas las personas que trabajan en ese proyecto y las actividades realizadas (tanto las validadas como las que no). A partir de la descripción: anterior, se pide:

- a) Obtener el diagrama de contexto en UML
- b) Obtener el diagrama de casos de uso en UML.

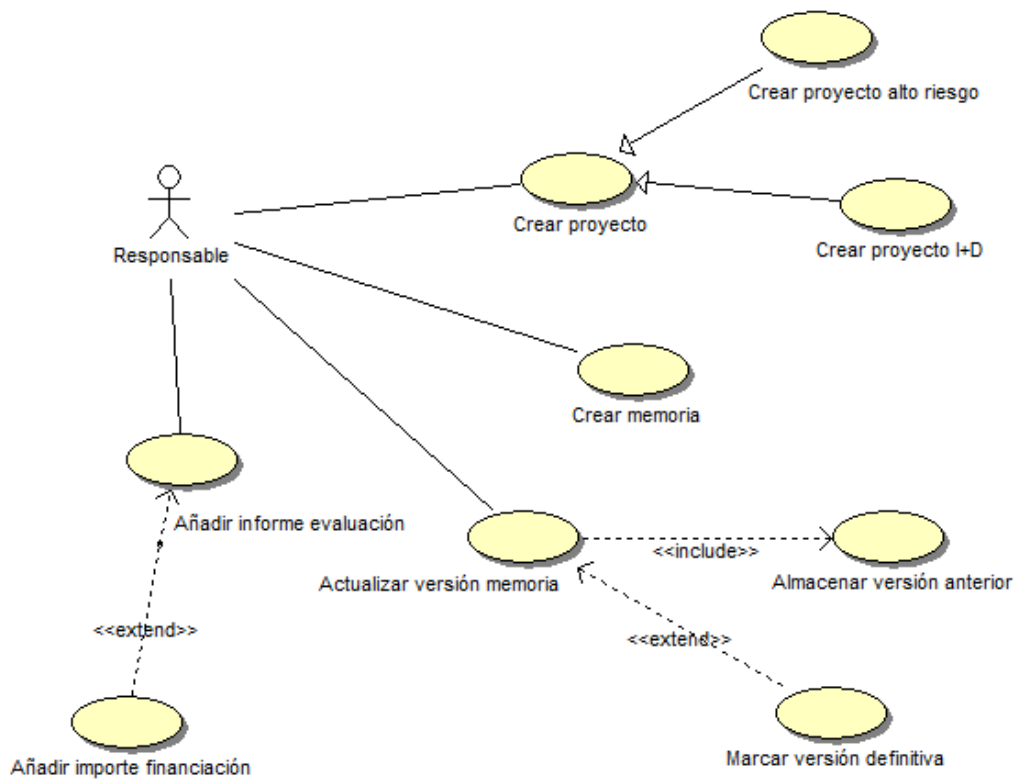
Solución 1:

Diagrama de contexto

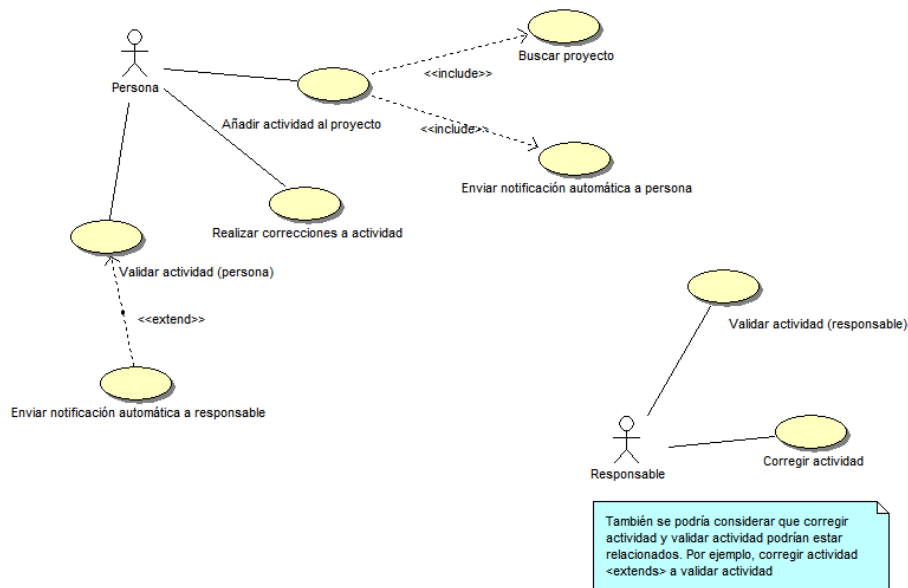


Diagramas casos de uso

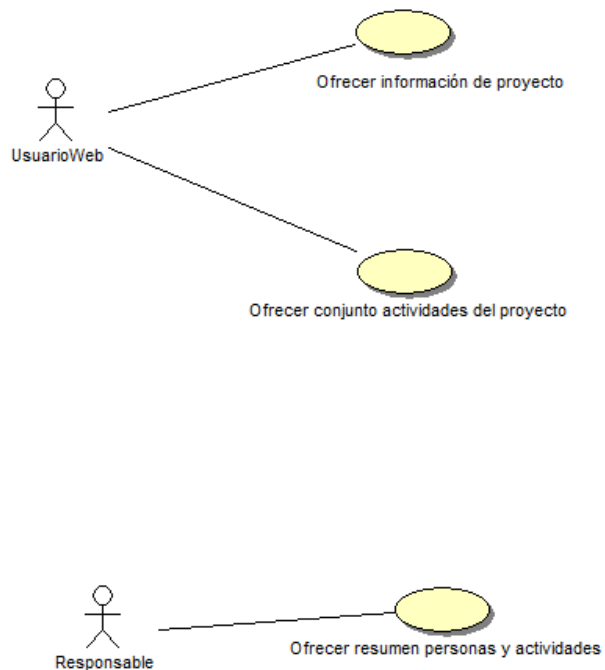
Proyectos: [suponemos que el responsable se encarga de la parte correspondiente al informe de evaluación]



Actividades:



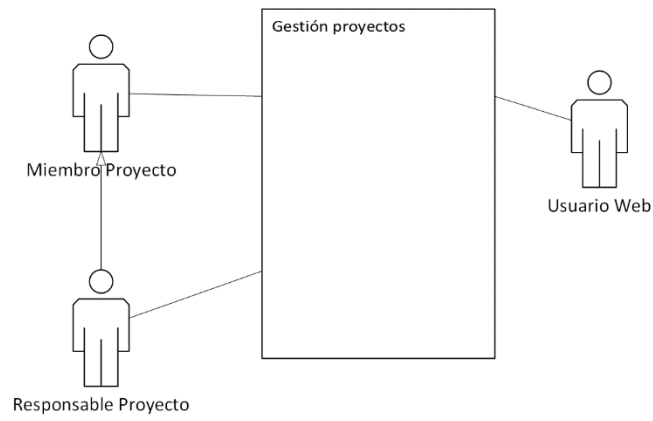
Informes/Listados:



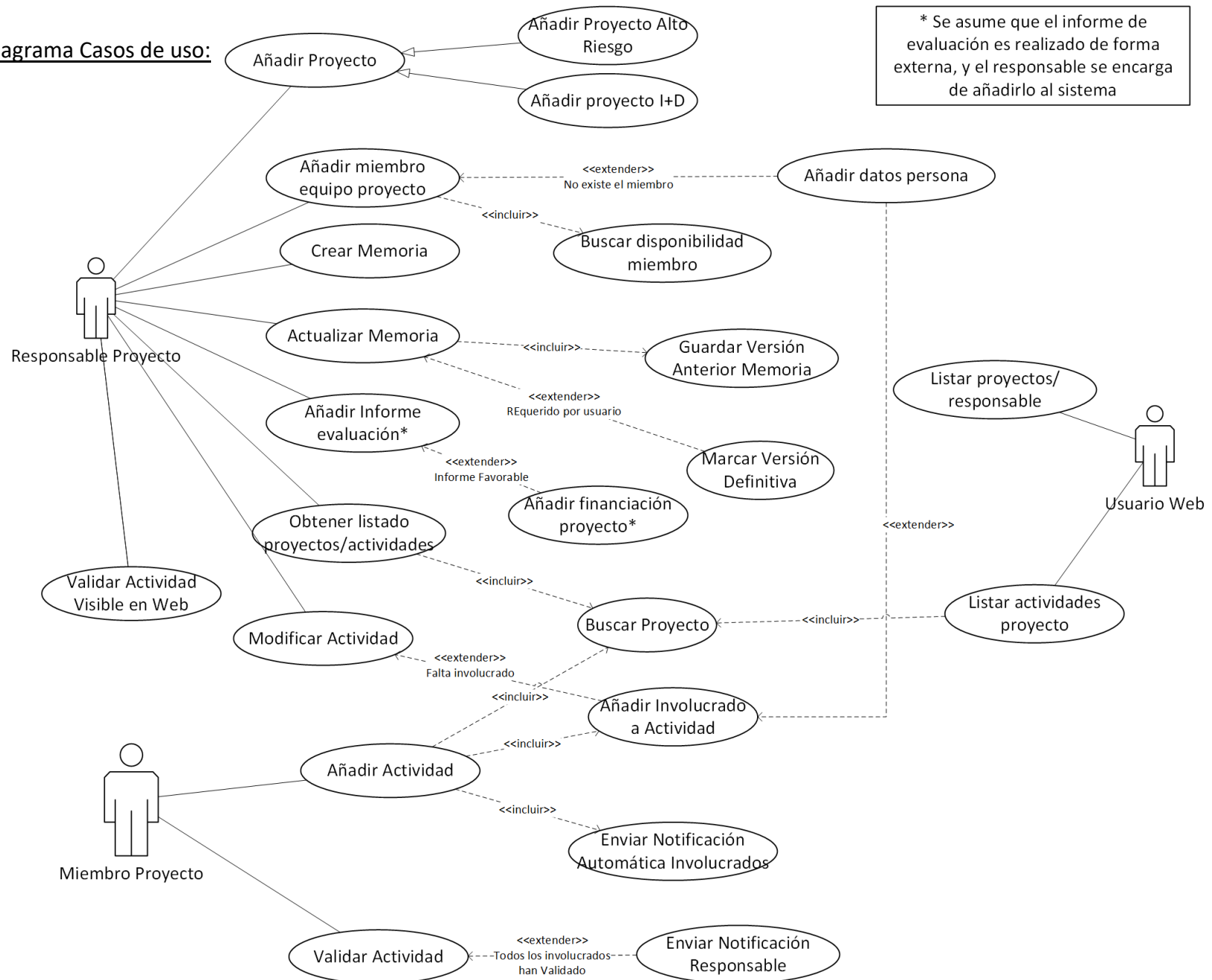
Solución 2:

La solución 2 contempla toda la parte de gestión de los miembros de un proyecto, que realmente no está pedida como funcionalidad (sí en la parte estática)

Diagrama de contexto:



Solución 2. Diagrama Casos de uso:



3. Comisión Nacional de Emergencias

La Comisión Nacional de Emergencias quiere ofrecer un servicio de llamadas de emergencias eficiente que optimice, cuando ocurra una emergencia, la llegada de las ambulancias, y el posterior transporte de los pacientes a los hospitales más apropiados. Para abordar este problema, se ha facilitado la descripción de los siguientes elementos:

- Hospitales. Cada hospital tiene un nombre, una dirección y unas coordenadas geográficas (latitud y longitud, p.e. 39.451, -0.358).
- Especialidades médicas. Cada especialidad tiene un nombre (p.e. dermatología, cardiología, cirugía, etc.). No todas las especialidades son ofrecidas por todos los hospitales; pero cada hospital ofrece al menos una especialidad, y todas las especialidades son ofrecidas por al menos un hospital.
- Ambulancias. Cada ambulancia tiene una placa o número de registro, un tipo de equipo instalado y las coordenadas geográficas dadas por un GPS para saber dónde se encuentra en cada momento. Hay que tener en cuenta que las ambulancias pueden moverse y sus coordenadas pueden cambiar a lo largo del tiempo, así como su disponibilidad; evidentemente si una ambulancia está ocupada por un paciente no está disponible. Hay dos tipos de ambulancias: ambulancias privadas y ambulancias con base en los hospitales. Las primeras no se asignan a ningún hospital particular y pueden transportar a los pacientes a cualquier hospital del sistema. Las últimas se asignan sólo a un hospital y pueden transportar a los pacientes sólo a ese hospital en particular. En el sistema habrá hospitales que tienen ambulancias y hospitales que no las tienen.
- Pacientes. Cada persona que se almacena en el sistema ha marcado el número de emergencias al menos una vez, con lo que se convierte realmente en un paciente. De lo contrario, aún no forma parte del sistema. Los campos más interesantes que describen a un paciente son su DNI, nombre y apellidos, sexo, edad, número de teléfono y dirección.
- Llamadas de emergencia. Los pacientes marcan el número de teléfono de emergencias en caso de emergencia. Se quiere almacenar las coordenadas geográficas del lugar donde ocurre la emergencia, proporcionadas automáticamente por la compañía telefónica, para averiguar la combinación óptima de ambulancia y hospital para transportar al paciente. Una emergencia significa que el paciente muestra algunos signos y síntomas relacionados con alguna de las especialidades (el paciente presenta al menos un síntoma de alguna de las especialidades médica, pero podría ser más de uno). Para cada uno de los síntomas que presenta un paciente es necesario conocer su gravedad (alta, media o baja) y durante cuánto tiempo (en horas) el paciente está sufriendo ese síntoma.
- Asignación de un paciente a una ambulancia y a un hospital. Después de recibir una llamada de emergencia, el sistema tiene que asignar automáticamente la ambulancia y hospital óptimos para ese paciente. La asignación óptima se calcula

en función de la distancia más corta entre una ambulancia disponible (es decir, no ocupada en ese momento por un paciente) el paciente y un hospital al que sea posible transportarlo, teniendo en cuenta que las ambulancias con base en un hospital, sólo pueden ir a ese hospital.

Funcionamiento del sistema:

Cuando una persona marca el número de emergencia, un operador especialmente preparado contesta la llamada y completa un formulario electrónico con la información sobre: i) el paciente y sus datos, si se trata de la primera vez que utiliza el sistema; ii) las coordenadas geográficas donde se encuentra el paciente y se ha producido la emergencia; y iii) todos los síntomas, sus respectivas especialidades e información adicional acerca de su gravedad y duración. Una vez que toda esta información se ha introducido en el sistema, éste tiene que proporcionar automáticamente una ambulancia y un hospital para hacer frente a esa emergencia de una manera muy eficiente. Es importante tener en cuenta las siguientes restricciones:

- El hospital asignado a una emergencia tiene que ofrecer todas las especialidades necesarias para tratar dicha emergencia.
- La ambulancia asignada a la emergencia debe estar disponible y debe poder transportar al paciente a ese hospital, bien porque es una ambulancia privada (y puede, consecuentemente, transportar pacientes a cualquier hospital del sistema), o bien porque tiene base en ese hospital.
- Las ambulancias pueden circular con o sin pacientes, por lo que sus coordenadas geográficas pueden variar en cualquier momento. Por lo tanto, es necesario que el conductor de la ambulancia sea capaz de cambiar fácilmente esas coordenadas de forma manual. Además, se asumirá que una ambulancia está siempre disponible, salvo que esté asignada en ese momento a una emergencia. Igualmente, es necesario que el conductor disponga de una opción para cambiar la disponibilidad de su ambulancia, por ejemplo, cuando una ambulancia ha entregado un paciente en un hospital y, por lo tanto, vuelve a estar disponible.
- La asignación óptima de la ambulancia y el hospital se calcula por medio de la distancia euclídea, ya que se considera que todas las ambulancias tienen la misma velocidad media. De esta manera, el sistema devolverá la tupla $\langle \text{ambulancia}_i, \text{hospital}_j \rangle$ que minimice la siguiente expresión:

$$\sum_{\forall \text{ ambulancia}_i, \text{hospital}_j} \text{distancia}(\text{coordenadas_ambulancia}_i, \text{coordenadas_emergencia}) + \text{distancia}(\text{coordenadas_emergencia}, \text{coordenadas_hospital}_j)$$

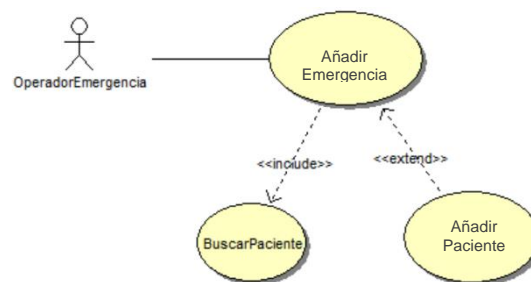
El sistema debe gestionar todo lo descrito anteriormente. Además, debe proporcionar al administrador del sistema las operaciones CRUD habituales (Create, Read, Update y Delete) para hospitales, especialidades médicas, ambulancias, pacientes, etc. Las búsquedas de hospitales, especialidades, ambulancias y pacientes con respecto a sus atributos más relevantes también serán necesarias. Por último, el sistema proporcionará los siguientes informes para el responsable de la Comisión, los directores de hospitales y los conductores de las ambulancias:

- Especialidades médicas de cada hospital (para el responsable de la Comisión).
- Especialidades médicas del hospital (para el director del hospital).
- Pacientes del sistema (para el responsable de la Comisión) y pacientes que han sido asignados a cada hospital (para el director del hospital).
- Ambulancias con base en hospitales para cada hospital (para el director del hospital).
- Llamadas de emergencia, con la información sobre el paciente, los síntomas y especialidades, y con la asignación de ambulancia y hospital (para el responsable de la Comisión).
- Llamadas de emergencia asignadas y hospital destino (para los conductores de ambulancias).
- Hospitales y ambulancias que han servido el mayor y/o el menor número de llamadas de emergencia (para el responsable de la Comisión). Esta información se considera esencial para mejorar la prestación del servicio.

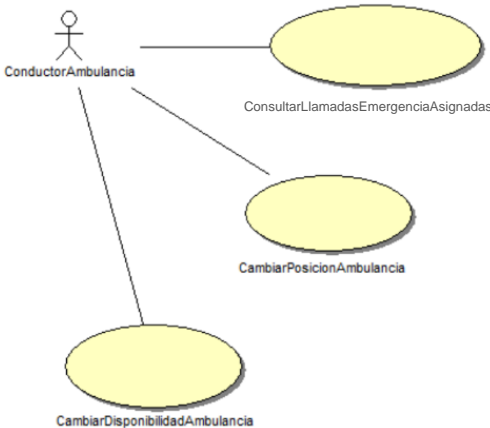
A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de casos de uso en UML.

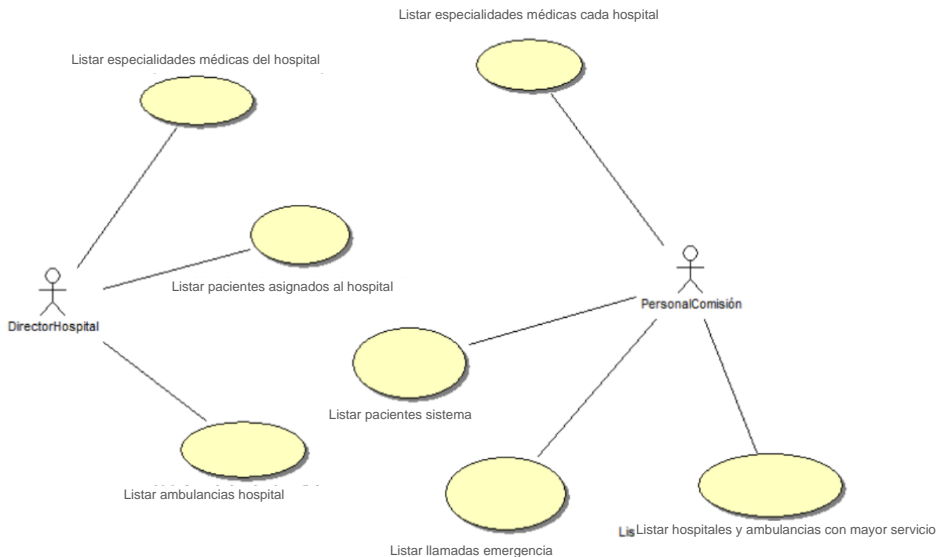
Gestión de emergencias



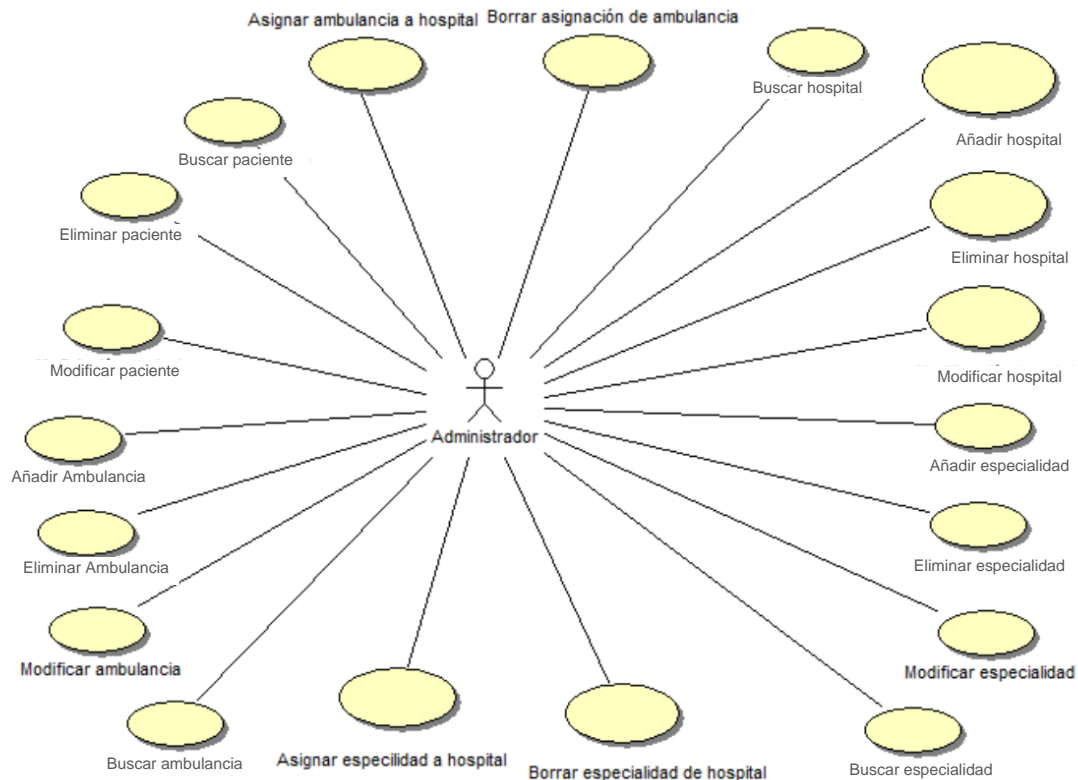
Operaciones del conductor



Listados de personal de comisión y hospital

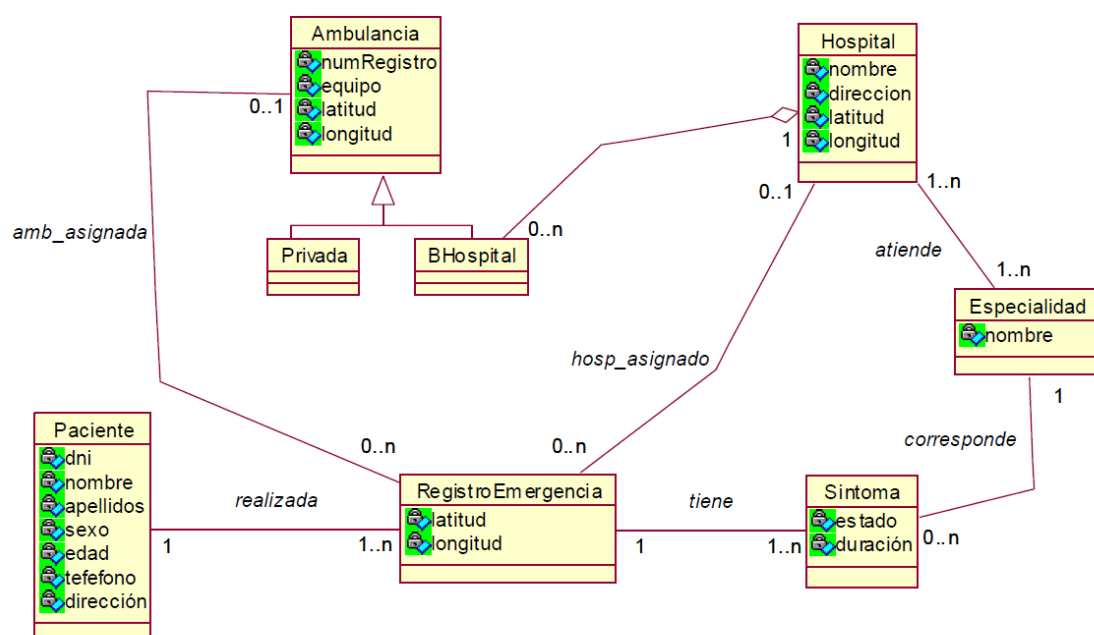


Operaciones de mantenimiento (tipo CRUD)



Nota: no se muestra el alta de paciente ya que suponemos que únicamente los puede dar de alta el personal de emergencia como resultado de una llamada de emergencia.

- b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.



4. Aplicación “MensajeYa”

Queremos desarrollar una aplicación “MensajeYa” para la integración de diferentes aplicaciones de mensajería instantánea del tipo Whatsapp, Telegram, Skype, etc. y sus contactos asociados. El objetivo es proporcionar a los usuarios una vista única de lectura de los mensajes/contactos a partir de su interacción en diversas aplicaciones de mensajería.

Una aplicación de mensajería viene descrita por su nombre, descripción, Web del desarrollador y una URL para el acceso a su API. Por otro lado, un usuario de “MensajeYa” posee un nombre, un número de teléfono, una dirección de correo electrónico para envío de notificaciones importantes y una fecha de nacimiento. Dicho usuario puede ser miembro de una o más aplicaciones de mensajería, a las que accede mediante una cuenta de usuario y una contraseña. Obviamente, la cuenta de usuario y contraseña variarán en función de la aplicación de mensajería. Por ejemplo, en el caso de Whatsapp la cuenta de usuario estará asociada al número de teléfono del usuario, mientras que en Skype puede ser un nombre y no un número de teléfono.

Cada usuario mantiene una lista de contactos, que habrá que dar de alta, con los contactos de sus aplicaciones de mensajería. Un requisito importante en el sistema es poder listar la totalidad de contactos existentes agrupados por aplicación de mensajería, y también agrupados por contactos “reales”, ya que puede darse el caso de que el mismo contacto esté repetido en más de una aplicación (por ejemplo, en Telegram y Whatsapp, con cuentas de usuario distintas representando al mismo contacto “real”). Por lo tanto, el sistema debe permitir añadir cuentas de usuario distintas de dichos contactos procedentes de las distintas aplicaciones de mensajería.

Cuando un usuario se da de alta en “MensajeYa”, debe proporcionar sus datos de perfil, e identificar de qué aplicaciones de mensajería es usuario, indicando los datos correspondientes. Para ello, el sistema ofrece al usuario una lista de las aplicaciones registradas previamente y el usuario, al seleccionar una de ellas, debe proporcionar la cuenta de usuario y la contraseña para acceder a la misma. A partir de ahí, el sistema debe construir una red de contactos que se almacenará para su uso posterior.

Cuando un usuario inicia sesión en “MensajeYa” se le ofrece la red de contactos; es decir, un contacto en “MensajeYa” es el definido como contacto en al menos una aplicación de mensajería. Para cada contacto se muestra al usuario las aplicaciones de mensajería en las que está presente y relacionado con el usuario. Al seleccionar uno de los contactos el sistema debe ofrecer dos posibilidades: 1) mostrar los últimos 4 mensajes de cada aplicación de mensajería en la que está el contacto, o 2) mostrar los últimos 10 mensajes del total de aplicaciones de mensajería, indicando mediante un icono el nombre de la aplicación de mensajería de la que procede cada mensaje. Con el fin de obtener esta información, se supone que existe un agente de mensajería para cada una de las aplicaciones registradas que extrae la información requerida sobre un contacto y la proporciona al sistema. Adicionalmente, al finalizar cada día se deberá generar un informe con los 5 contactos que más mensajes nos han enviado y enviárselo al usuario por correo electrónico. Nos interesa almacenar un histórico de esos informes, con la fecha y los 5 contactos.

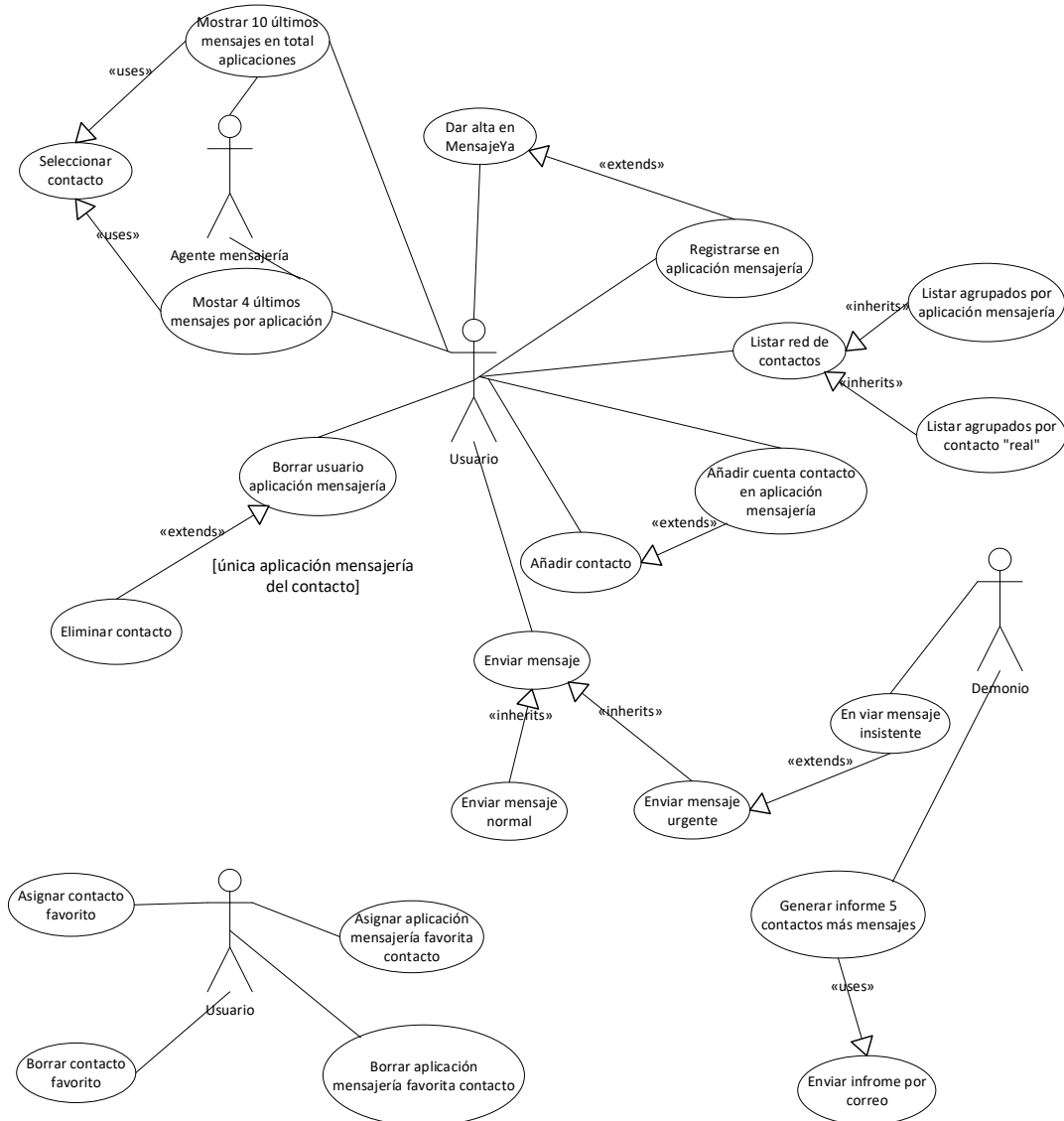
El sistema debe permitir borrarse de una aplicación de mensajería cuando se estime una cantidad de información desmesurada. Esto implicará la pérdida de los contactos de dicha aplicación, aunque pueden seguir siendo contactos a través de otra aplicación. Evidentemente, si se trataba de la última aplicación de mensajería que me relacionaba con ese contacto, se perderá (es decir se deberá eliminar) dicho contacto

El sistema también debe gestionar (altas y bajas) cuál es la aplicación de mensajería favorita de cada contacto y una serie de contactos favoritos. De esta forma, el sistema debe ofrecer dos formas de envío de mensajes a un contacto favorito: 1) envío normal, donde se enviará un mensaje mediante la aplicación de mensajería favorita del contacto, y 2) envío urgente, donde se enviará el mismo mensaje utilizando todas las aplicaciones de mensajería en las que esté registrado el contacto. Además, en el caso de envío urgente se puede programar una repetición automática *insistente* para que se re-envíe el mismo mensaje cada hora hasta que el contacto responda. Para los contactos no favoritos solo se puede realizar el envío normal de mensajes.

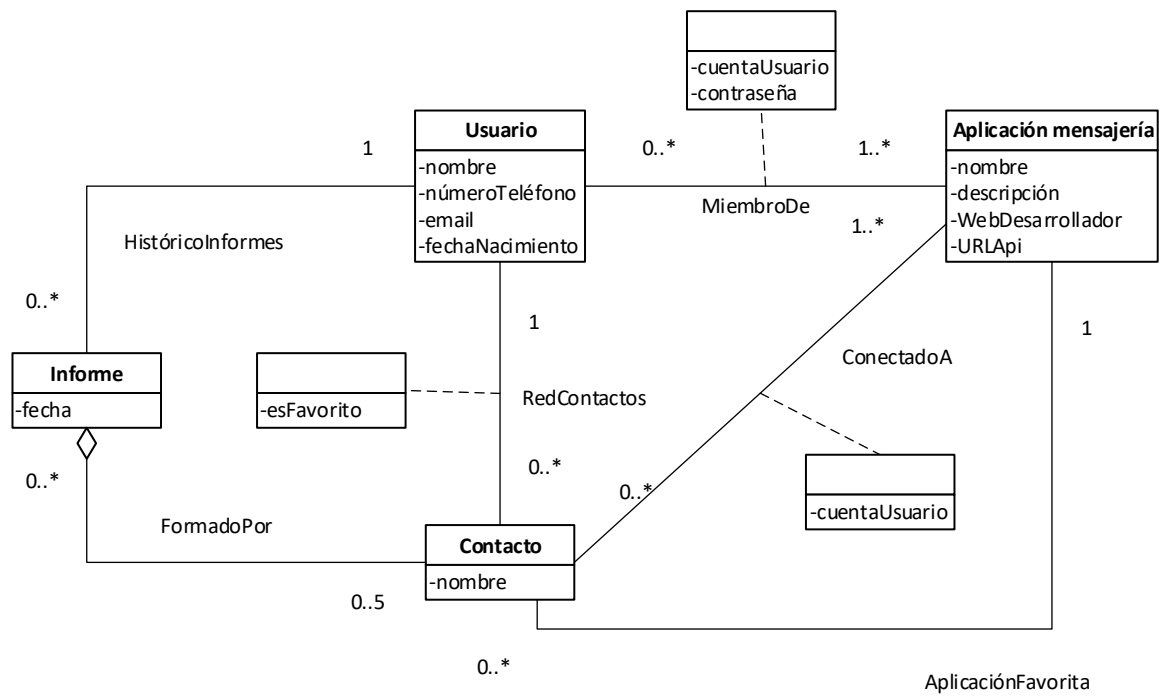
Por el momento no se permite generar grupos de contactos con la idea de enviar mensajes (normales y urgentes) a dichos grupos, aunque está previsto que se incorpore en futuras versiones.

A partir de la descripción anterior, se pide:

a) Obtener el diagrama de casos de uso en UML.



b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.



5. Gestión de una Clínica

Una clínica que atiende a pacientes de diferentes aseguradoras ha encargado que se le desarrolle una sencilla aplicación de gestión de las visitas de los clientes, que de soporte a su forma actual de actuar.

De cada paciente necesita conocer sus datos personales (nombre, dirección, teléfono, correo electrónico y NIF) así como su aseguradora y número de póliza. Los datos de los pacientes solo pueden eliminarse si no han tenido ninguna visita en el último año.

Del mismo modo, de los diferentes médicos que pasan consulta también desean mantener sus datos personales (nombre, dirección, teléfono, correo electrónico y NIF) y su número de colegiado. Los médicos podrán eliminarse del sistema sólo si no han tenido ninguna visita en el último año.

Cuando un cliente llama a la clínica para pedir cita, el auxiliar verifica la identidad del cliente (realizando una acreditación de su identidad) y comprueba si ya es paciente de la clínica. En ese caso busca un médico y una consulta disponible dentro de una franja horaria y le abre una ficha de visita. En ésta le anota la fecha en la que se realizará, la consulta asignada y un tiempo de inicio y final estimado para ésta. Posteriormente, el médico, al atender al cliente, actualizará dicha ficha con los síntomas del cliente, el tratamiento recetado y el importe de la visita.

Si la persona que llama no es cliente, el auxiliar le pide todos sus datos y procede a abrirle una ficha. Tras ello, realiza el mismo paso descrito anteriormente.

A partir de la descripción anterior, se pide:

- Obtener el diagrama de contexto.
- Obtener el diagrama de casos de uso en UML.
- Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

Diagrama de contexto:

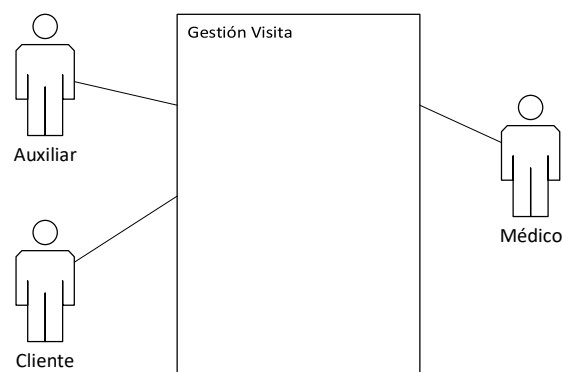


Diagrama de casos de uso:

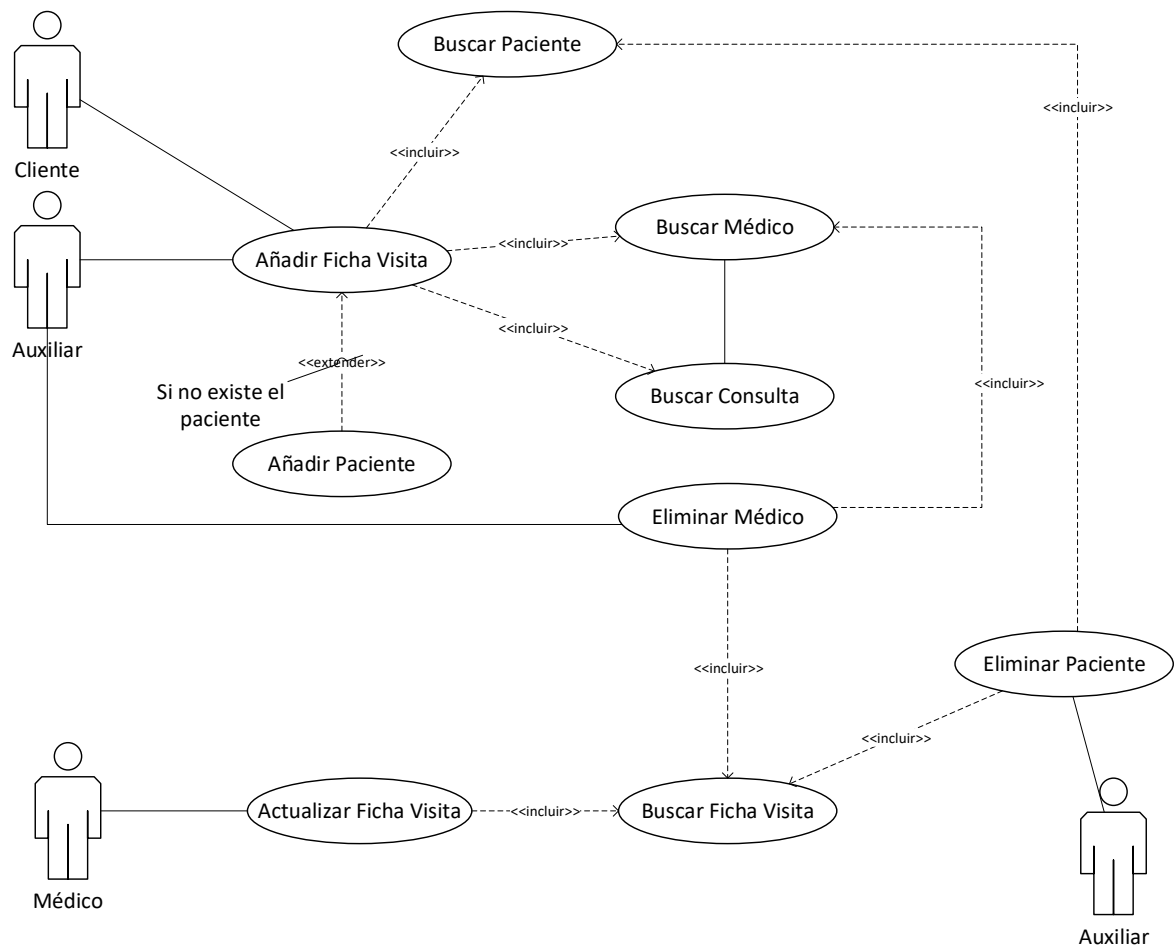
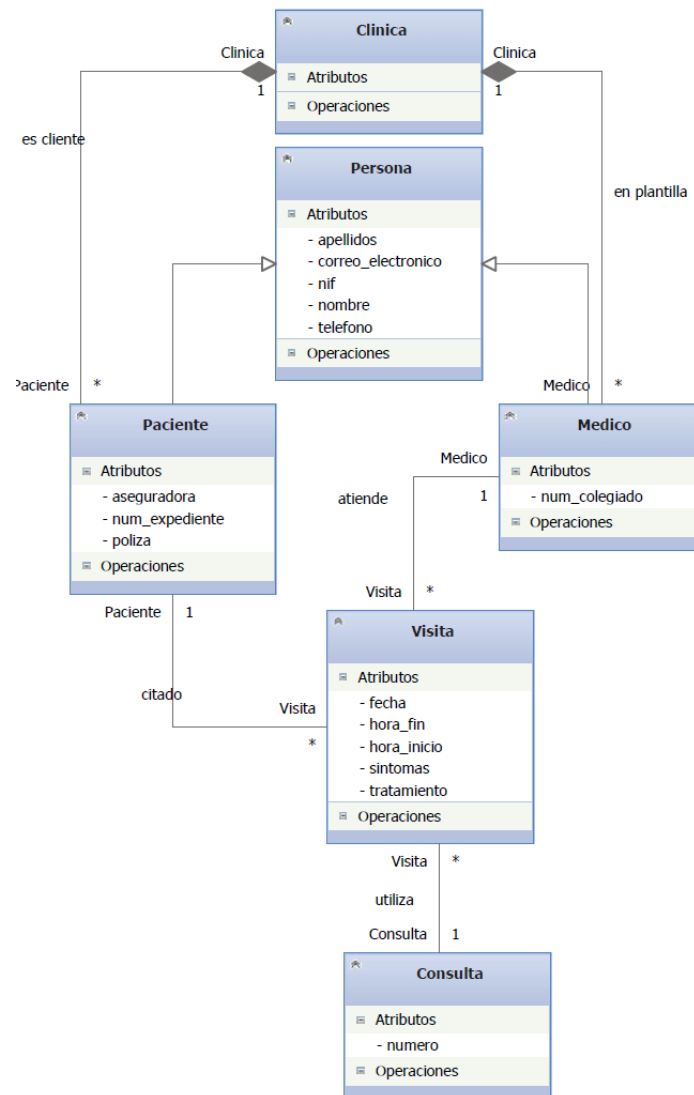


Diagrama de clases:



6. Secretaría Virtual

En una universidad privada están comenzando a desarrollar un nuevo sistema de secretaría virtual que permitirá que el alumno pueda realizar las gestiones más demandadas en la actualidad a través de una página web, autenticándose adecuadamente antes de acceder a cualquier servicio.

Así, el alumno podrá consultar su expediente, con la relación de asignaturas del mismo, con la nota obtenida y el curso académico en que se superó. Además, en el expediente aparece el nombre y apellidos del alumno, su NIF, el número de expediente y el nombre de la secretaria que lo emite.

Además, podrá realizar el pago de los recibos pendientes con la universidad. Cada recibo tiene un precio y un concepto.

Por último, podrá consultar algunos datos de las asignaturas ofertadas, como el título, una breve descripción y número de créditos ECTS de que consta.

Teniendo en cuenta la descripción anterior, se pide

- Obtener el diagrama de contexto y el diagrama de casos de uso en UML.
- Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que aparezcan, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

Diagrama de contexto:

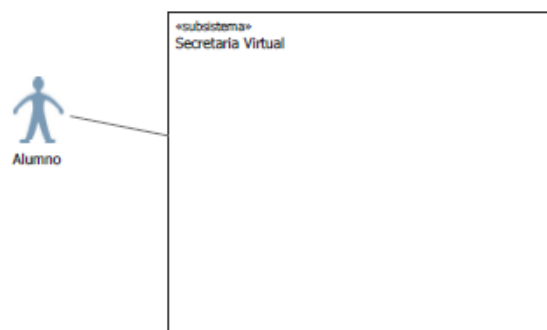


Diagrama de casos de uso:

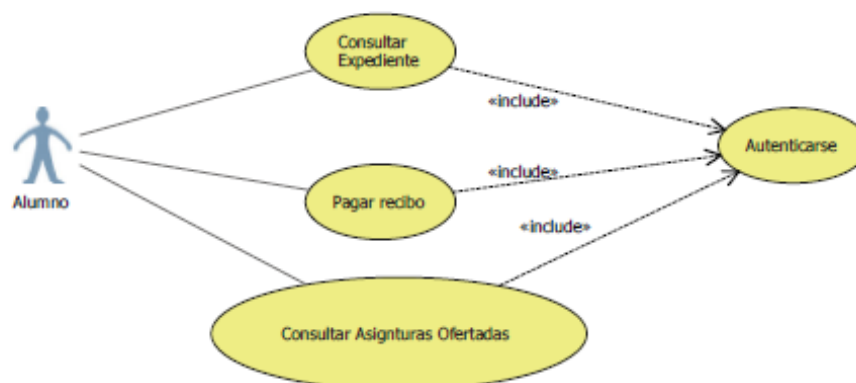
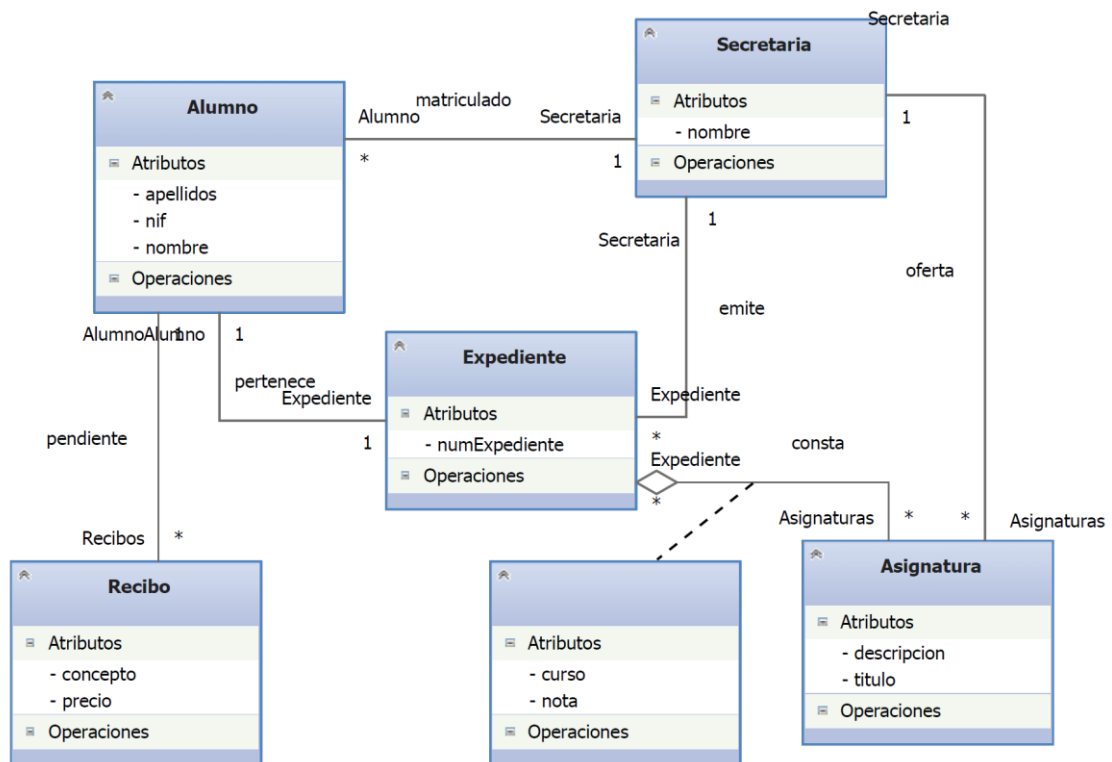


Diagrama de Clases:



7. Gestión biblioteca universidad

Se desea desarrollar un sistema informático para la biblioteca de una universidad. La especificación dada por el cliente es:

En la biblioteca hay libros y revistas. Puede haber varias copias de un libro. Algunos libros se prestan a corto plazo y el resto se pueden prestar a cualquier socio de la biblioteca durante 3 semanas. Solo los socios del personal pueden tomar prestadas revistas. Los socios de la biblioteca normalmente pueden tomar prestadas hasta 6 artículos de una sola vez. Regularmente llegan nuevos libros y revistas, y a veces hay que deshacerse de los antiguos. Las revistas del año actual se envían para encuadernarlas en volúmenes al finalizar el año.

El sistema debe llevar un control de cuándo se prestan y devuelven los libros y revistas. Además, deberá avisar cuándo un libro ha sido prestado, pero no devuelto. En un futuro se puede requerir que los usuarios puedan ampliar el préstamo de un libro si no está reservado.

El sistema debería permitir a los usuarios buscar un libro por tema, por autor, etc., para comprobar si hay una copia del libro disponible para ser prestado, y si no, lo reserva. Cualquiera puede hojear los libros de la biblioteca.

Teniendo en cuenta la descripción anterior, se pide

- Obtener el diagrama de contexto y el diagrama de casos de uso en UML.
- Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que aparezcan, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

Diagrama de contexto:

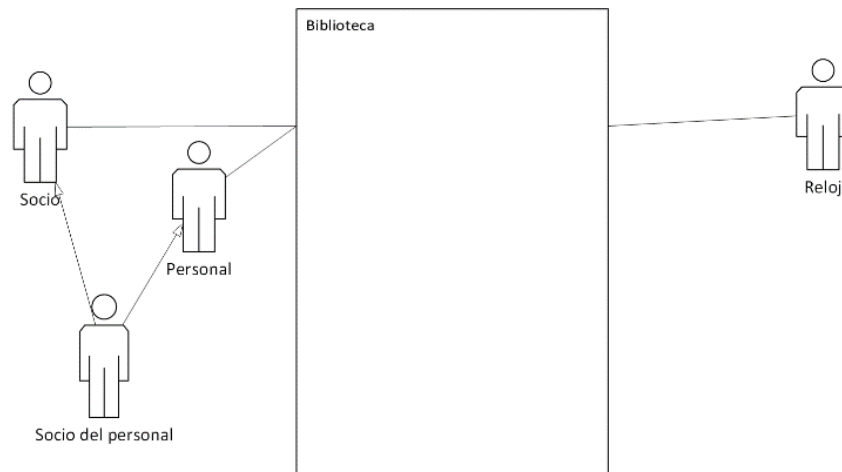


Diagrama de casos de uso en UML:

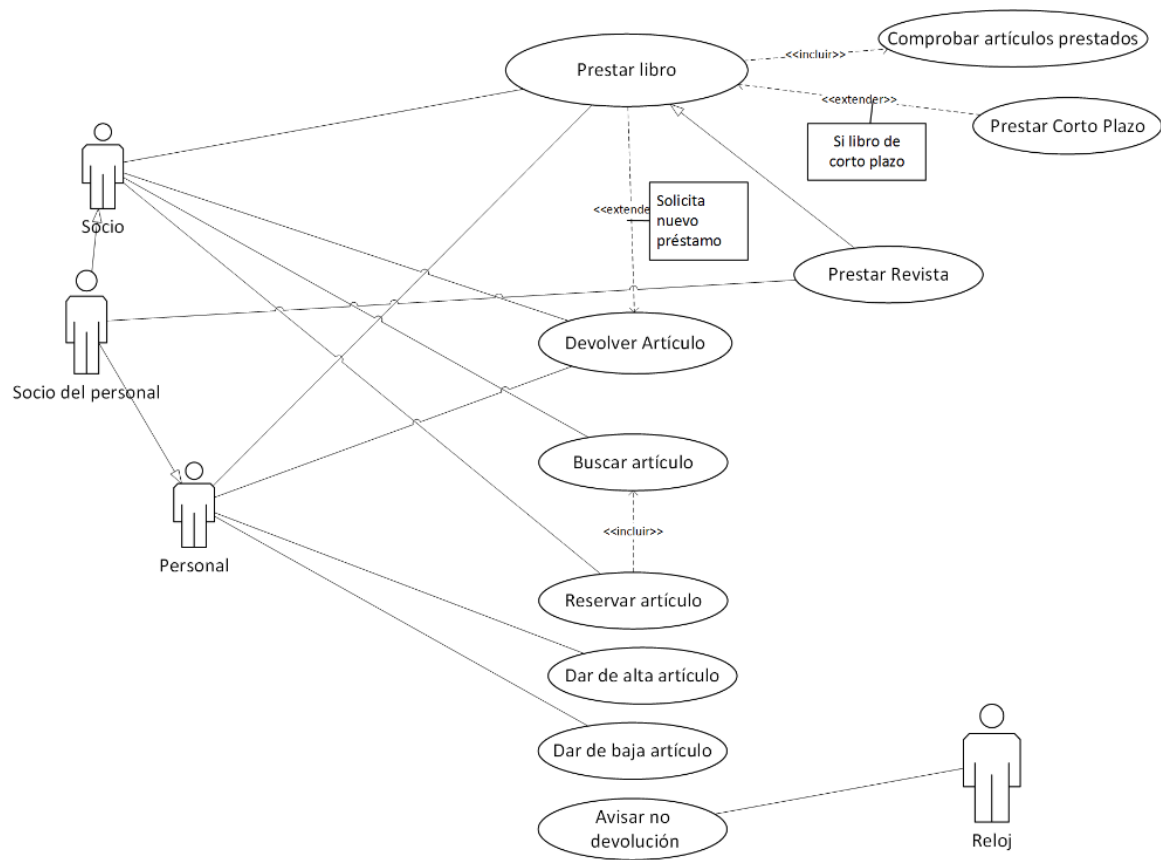
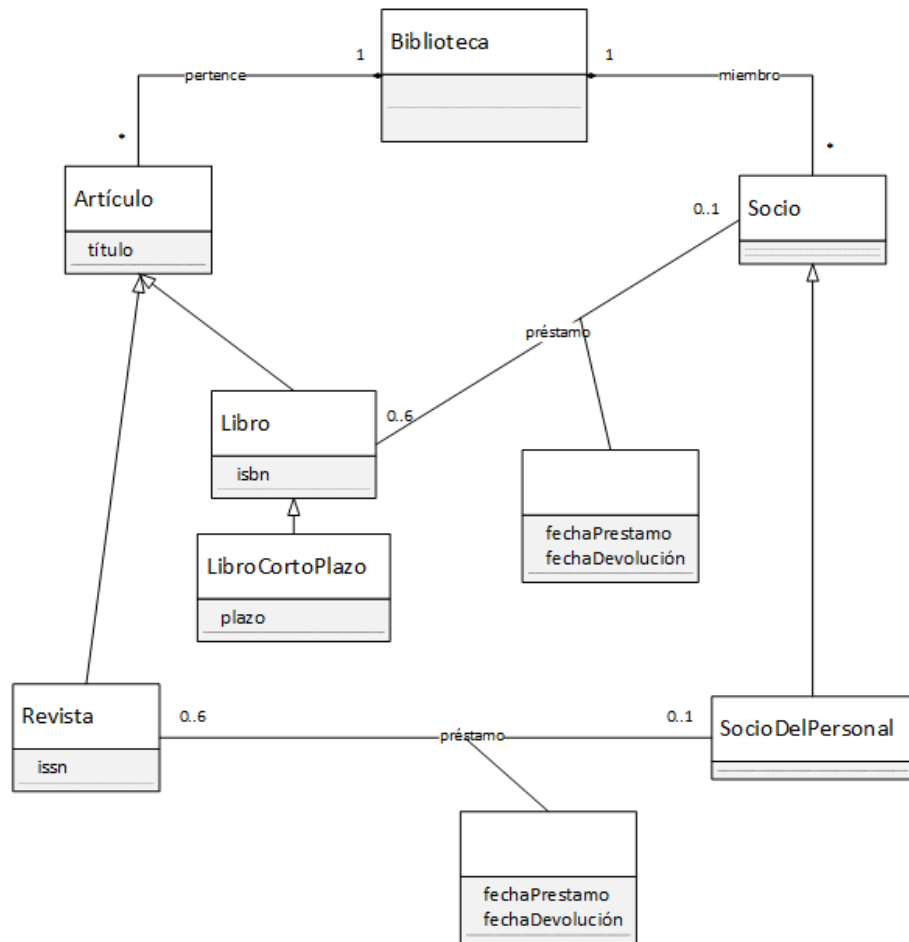


Diagrama de clases:



8. Hoteles Morfeo (ejercicio del boletín)

La cadena de hoteles Morfeo ha decidido reorganizar sus procesos de negocio con el fin de aumentar la automatización. Para ello, ha contactado con ISGSoft y le ha encargado el desarrollo del proyecto. La que sigue es una descripción del negocio.

La cadena posee hoteles situados en diferentes ciudades de distintos países. Cada hotel tiene una categoría definida de la forma usual (1-5 estrellas). En cada hotel hay un número de habitaciones individuales, dobles y triples, pudiéndose dar casos en los cuales no exista oferta de alguna de ellas. Todas las habitaciones están equipadas con aire acondicionado, televisión, minibar y caja fuerte. En los hoteles de 4 y 5 estrellas, además, existe una consola de videojuegos disponible para los huéspedes sin coste.

Como característica principal de la cadena Morfeo, sus hoteles disponen de servicios relacionados con la salud, como gimnasio, masajes e hidroterapia. Además, existen los típicos servicios de lavandería, alquiler de coches y golf. Todos estos servicios tienen un coste no incluido en el precio de la habitación.

Se desea automatizar los siguientes procesos:

1. Reserva de habitaciones: un cliente, por teléfono, puede realizar una reserva; para ello, debe indicar el tipo de habitación que desea, el día de entrada, el de salida. Si existe disponibilidad para las fechas solicitadas, se le comunica el precio, y si es aceptado, se le solicita un número de tarjeta de crédito, su nombre y un número de teléfono de contacto. Al crear la reserva, se genera un número de reserva, que se comunica al cliente para que lo indique en el momento de hacer la entrada.
2. Entrada (check-in). Cuando el cliente llega al hotel, se le solicita el número de reserva y un documento de identificación. Si es la primera vez que visita el hotel, se le abre una ficha de cliente. Si no, se recupera su ficha y se añade un nuevo registro de estancia, en el que se anotan fechas de entrada y salida, número de la habitación asignada.
3. Gestión de servicios extra. Cada vez que el cliente solicita (ya sea por teléfono, ya sea personalmente en la recepción) algún servicio extra, se crea un registro de servicio en el que se indica el servicio solicitado, así como el importe, con el fin de añadirlo posteriormente a la cuenta de la habitación.
4. Salida (check-out). Cuando el cliente abandona la habitación, se genera la factura que, además del importe de la habitación, debe incluir todos los gastos extra realizados por el cliente. La factura puede ser pagada en efectivo o con tarjeta de crédito. El importe de la misma es apuntado en el registro de estancia.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de contexto y modelo inicial del sistema en UML (Modelo de Casos de Uso).
- b) Obtener el Diagrama de Clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto:

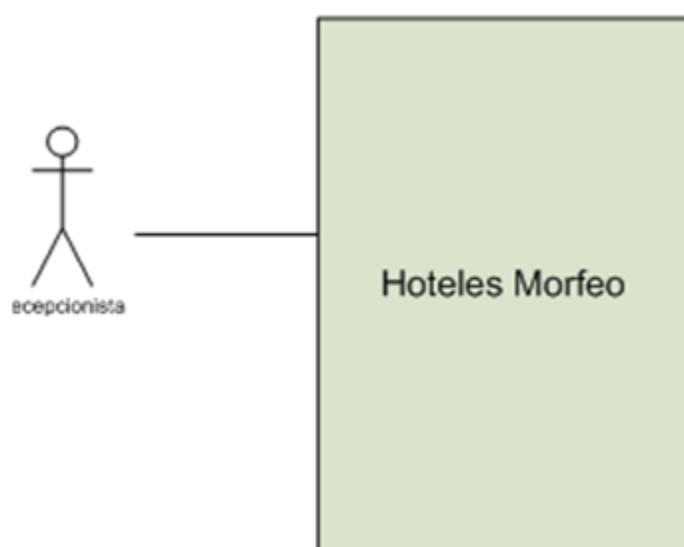


Diagrama de casos de uso

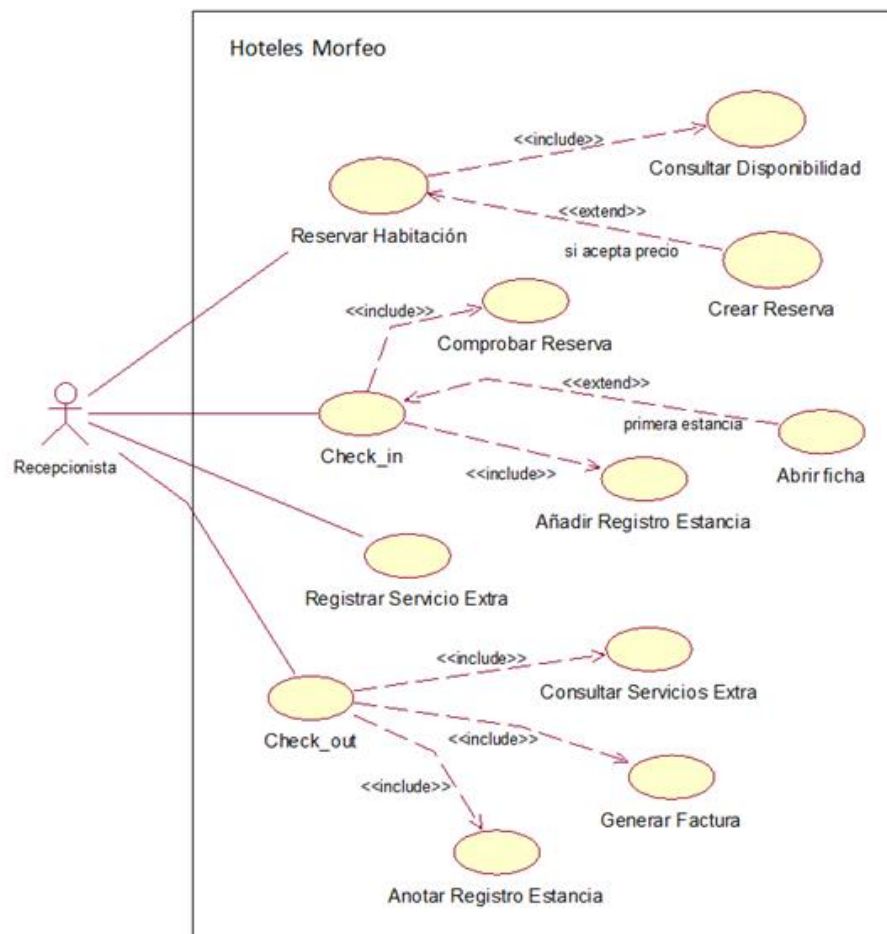
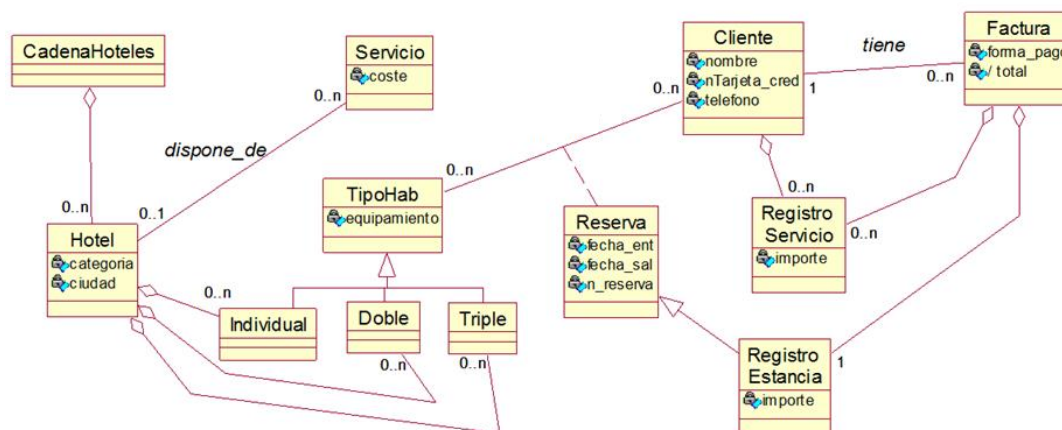


Diagrama de clases



9. El portal ISGOO (ejercicio del boletín)

El portal ISGOO!! quiere extender los servicios que presta a sus usuarios creando un gestor de imágenes a semejanza de los realizados por sus competidores. Para ello, ha elaborado la siguiente descripción del problema:

Se trata de ofrecer un servicio mediante el cual los usuarios, a través de una interfaz Web, puedan depositar sus imágenes. Se aceptarán imágenes en formato JPEG, TIFF, GIF y PNG. De cada imagen interesa conocer su autor, la fecha en que se tomó, la fecha en que se deposita en el portal, su tamaño y un conjunto de descriptores o palabras clave para facilitar su clasificación. Cuando un usuario se registra, se crea un espacio privado donde irá depositando sus imágenes. Sólo el usuario propietario del espacio está autorizado a subir imágenes. Al subir una imagen, el usuario puede añadir un comentario sobre la misma de una longitud máxima de 300 caracteres. Con el fin de facilitar un control de accesos más rico al sistema, el usuario puede definir colecciones en su espacio, y asignar permisos de acceso a las mismas a otros usuarios del sistema. La intersección de las colecciones no tiene por qué ser vacía.

Una colección puede visitarse de dos maneras. En el modo vista general, se muestra al usuario una pantalla con las miniaturas de todas las fotos que componen la colección. Aunque puede particionarse para facilitar su recorrido, estas mejoras no se tendrán en consideración para el presente ejercicio. Cuando un usuario hace clic con el ratón sobre una miniatura, el sistema le muestra la imagen junto con los metadatos mencionados anteriormente (lo que se conoce como modo vista miniatura). Además, el usuario puede a su vez depositar un comentario sobre la imagen; dicho comentario se mostrará la próxima vez que se visualice la imagen. Una determinada imagen puede tener diferentes comentarios en cada una de las colecciones en las que se sitúe.

Además de las vistas personalizadas, el portal puede explotar la colección global de imágenes. Para ello, los propietarios de las imágenes deben especificar, en el momento de depositarlas, si autorizan al sistema a utilizarlas en la vista global. Para construir la vista global, el sistema empleará los descriptores que acompañan a cada imagen para ofrecer una especie de directorio por categorías que se podrá explorar por los diferentes usuarios, tanto registrados como no registrados. En este caso no se permitirá depositar comentarios asociados a las imágenes.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de contexto y diagrama de Casos de Uso del sistema en UML.
- b) Obtener el Diagrama de Clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto

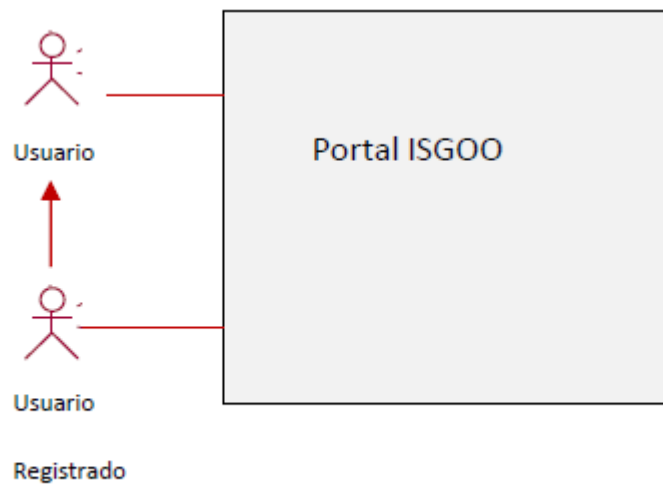


Diagrama de casos de uso

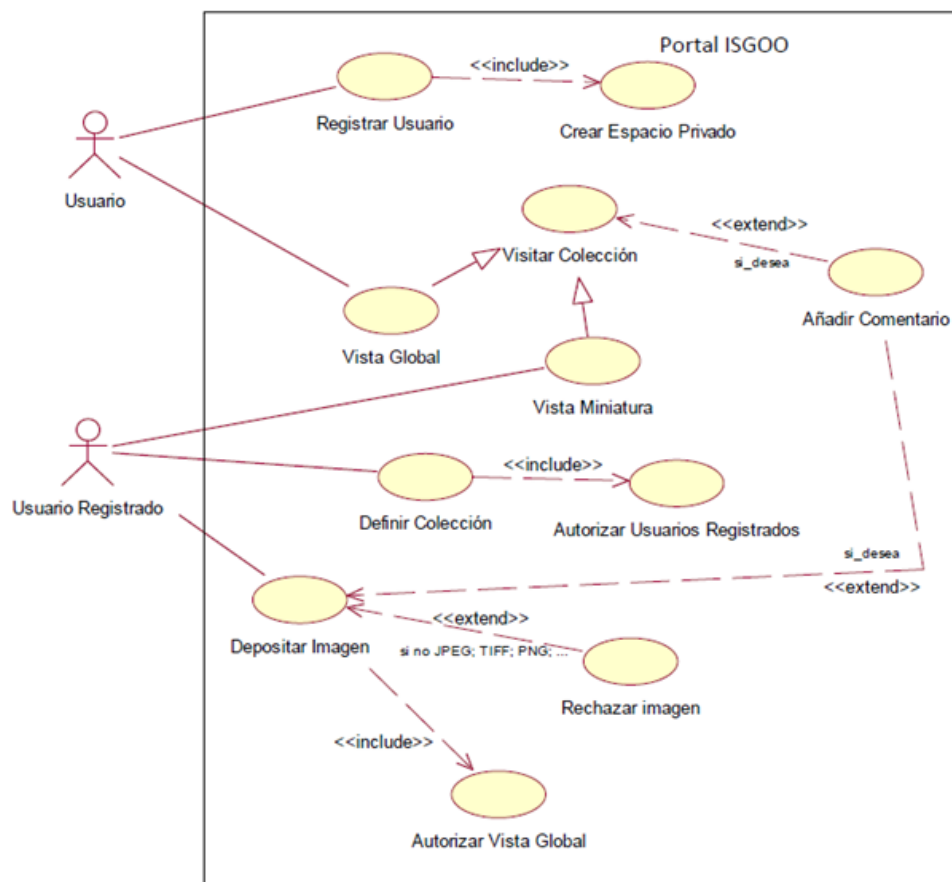
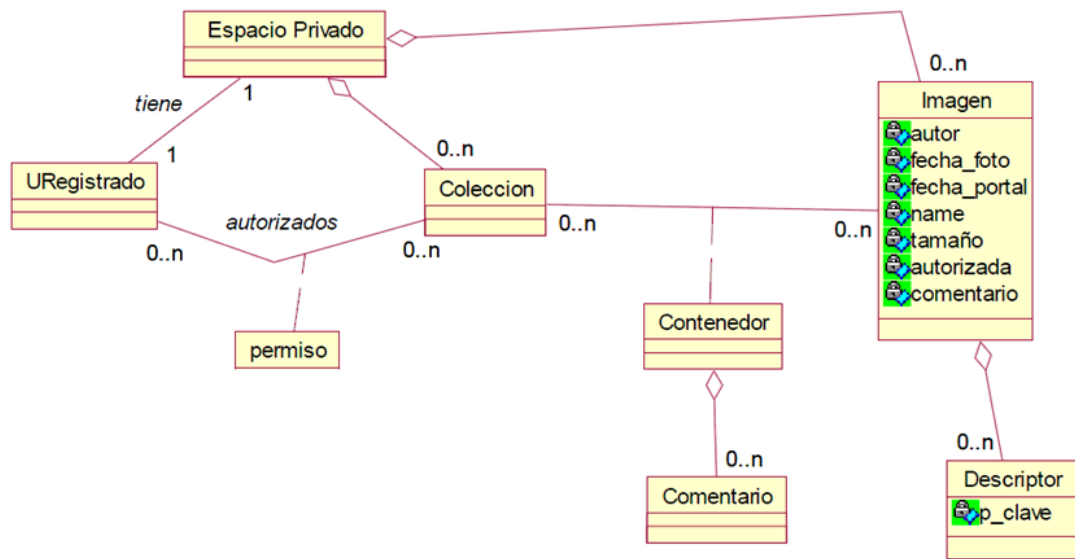


Diagrama de clases



10. Editor colaborativo (ejercicio del boletín)

Se desea implementar un editor colaborativo de textos que permita editar, organizar y compartir una serie de documentos entre uno o más usuarios. Para ello, los analistas de ISG Soft han realizado una primera descripción informal del problema:

El editor permite la manipulación de documentos por uno o más usuarios, que accederán a la funcionalidad del mismo mediante su navegador Web. Para comenzar a utilizar el editor, cualquier usuario debe autenticarse. Un documento es la unidad básica de trabajo, y se caracteriza por un conjunto reducido de metadatos: nombre, fecha de creación y propietario. El editor es colaborativo porque otros usuarios del sistema pueden ser autorizados a trabajar con el documento por el propietario, que es el usuario que creó el documento. En cualquier momento, el propietario de un documento puede invitar a otros usuarios del sistema a participar en la edición del documento con diferentes privilegios (lectura y escritura, sólo lectura).

Cada usuario dispone de un espacio privado de trabajo. Dicho espacio se puede organizar en carpetas, cada una de las cuales puede contener documento u otras carpetas. Igual que con los documentos, existirán facilidades para manipular carpetas (crear, borrar, renombrar).

Cuando un usuario desea modificar un documento para el que está autorizado, se crea una nueva versión del mismo, que es modificada y cuyos cambios no afectan al documento desde el cual se creó la versión. Cuando el usuario ha terminado de modificar, debe guardar la versión, que se convierte en la más reciente, pero que no elimina del sistema las anteriores versiones. Sólo el propietario del documento está autorizado a eliminar versiones no deseadas del documento. Por defecto, la versión más reciente se convierte en la “versión actual” del mismo, la cual se supone que representa el contenido actual del documento. Existe la opción para el propietario de cambiar en cualquier momento la versión actual del documento. Si se elimina una versión actual, su predecesora se convierte en la nueva versión actual. Finalmente, sólo el propietario puede eliminar el documento, lo cual supone la eliminación de todas sus versiones.

El editor ofrecerá las funciones que se esperan en cualquier editor de textos, y que se detallarán en documentos posteriores.

A partir de la descripción anterior, se pide:

- a) Obtener el modelo de contexto y el modelo de casos de uso en UML.
- b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto

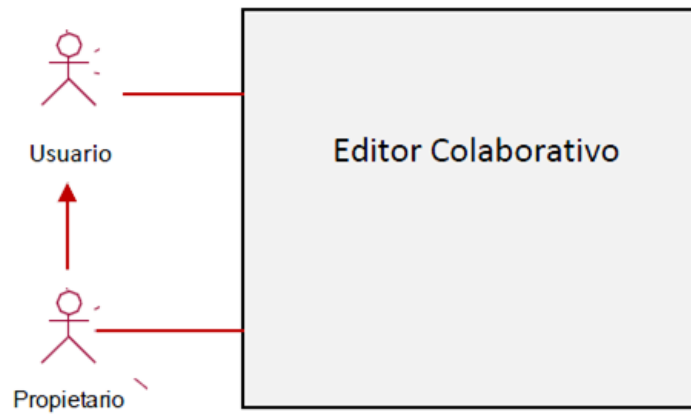


Diagrama de casos de Uso:

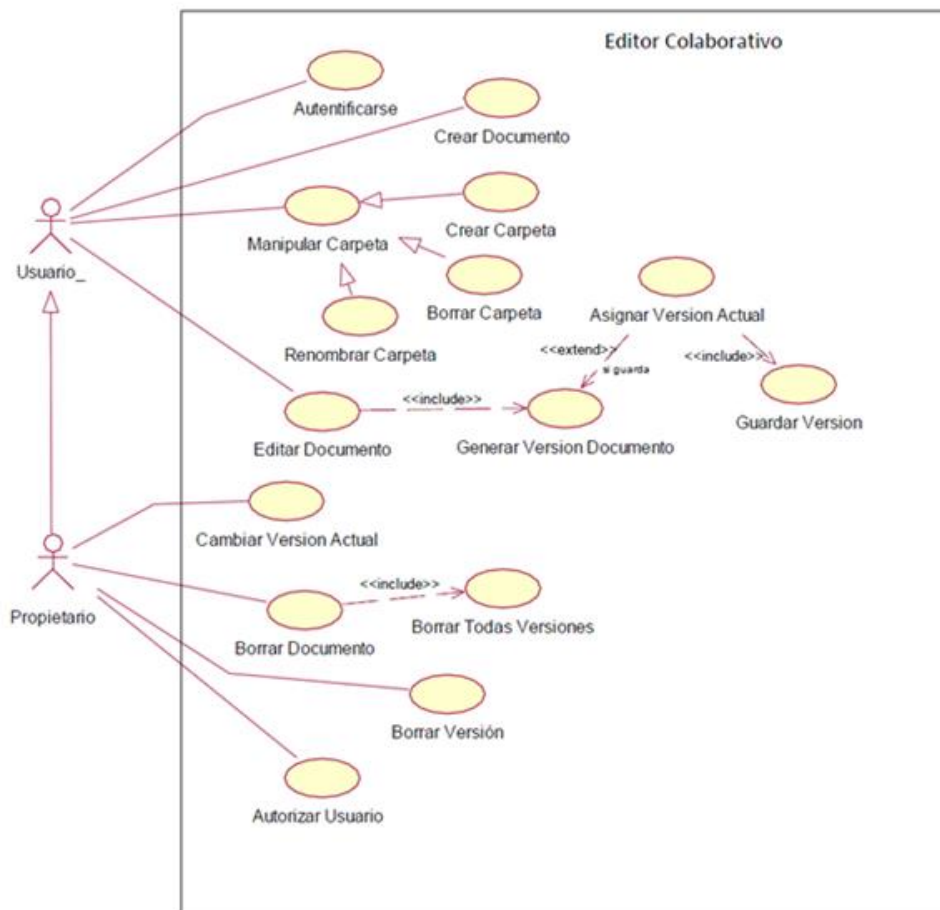
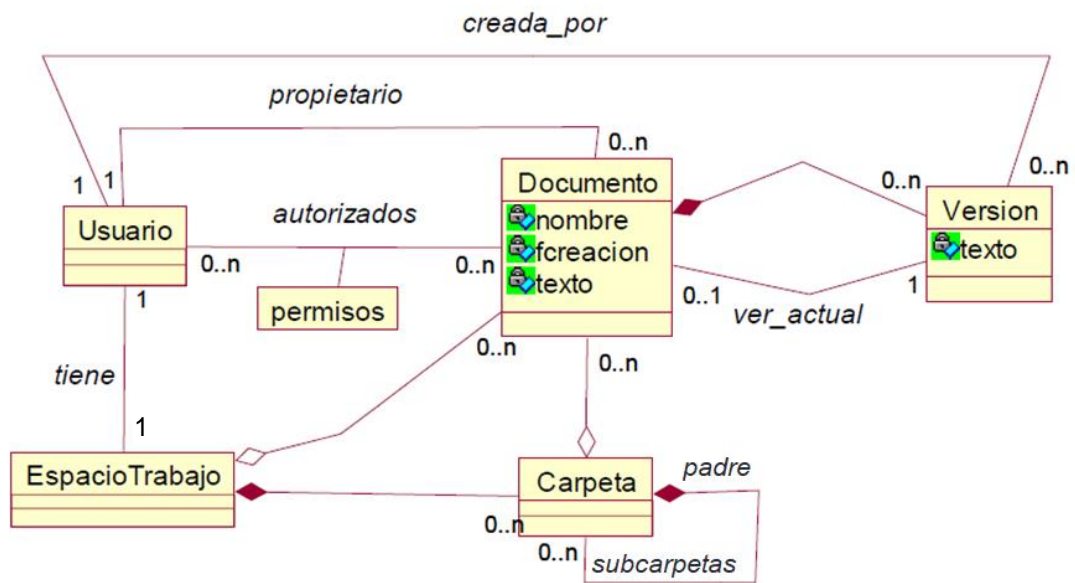


Diagrama de clases:



11. Sistema de apoyo a la docencia (ejercicio del boletín)

Se desea desarrollar un sistema de apoyo a la docencia que permita a los profesores y alumnos de una universidad la preparación, difusión y acceso a materiales de estudio a través de la Web. A continuación, se describe una versión simplificada del sistema:

El Sistema de Apoyo a la Docencia (SAD) ofrece soporte tanto a profesores como a alumnos. A los primeros les permite publicar avisos relacionados con las asignaturas que imparten y materiales docentes, a los que denominaremos recursos. A los segundos les permite el acceso a los dichos recursos y avisos de las asignaturas en que están matriculados. Cada usuario, ya sea profesor o alumno, tiene asignado un espacio de trabajo personal al que se accede mediante un proceso de autenticación basado en el modelo identificador/contraseña.

El espacio de un usuario está dividido en zonas, cada una correspondiente a una asignatura. Una zona puede contener, además de otros elementos no descritos para simplificar, recursos y avisos. Los recursos se organizan en carpetas, que pueden contener archivos u otras carpetas. Las carpetas pueden ser creadas exclusivamente por los profesores, de la misma forma que los archivos pueden ser depositados en las carpetas exclusivamente por ellos. Toda carpeta tiene un nombre, una descripción y una fecha de creación. Al depositar un archivo, el sistema puede, si así lo desea el profesor, enviar una notificación por correo electrónico al resto de profesores o a todos los usuarios autorizados a acceder a la zona. Igualmente, cada recurso puede tener un periodo de vigencia que determina el profesor y que puede modificarse en cualquier momento. Además, el recurso puede ser visible sólo para profesores, o para profesores y alumnos.

Los avisos son creados por los profesores y suelen tener como destinatario al conjunto de los alumnos matriculados más los profesores de la asignatura. Sin embargo, en ocasiones se permitirá enviar avisos sólo a profesores, si así lo desea el autor. Los avisos se publican en la zona correspondiente de SAD, si bien una copia de los mismos es enviada a la dirección de correo electrónico de cada uno de los destinatarios.

Los alumnos tienen acceso a los recursos y a los avisos a través del SAD. De cada recurso se visualiza una breve descripción, que incluye, en el caso de los archivos, un enlace para acceder al contenido de los mismos. Obviamente, sólo aquellos recursos cuya visibilidad está permitida son mostrados al usuario.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de contexto, el modelo de casos de uso en UML (incluir ejemplos de inclusión y extensión).
- b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto:

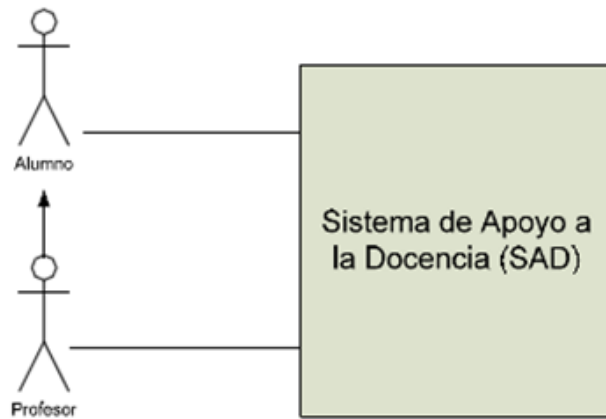


Diagrama casos de uso:

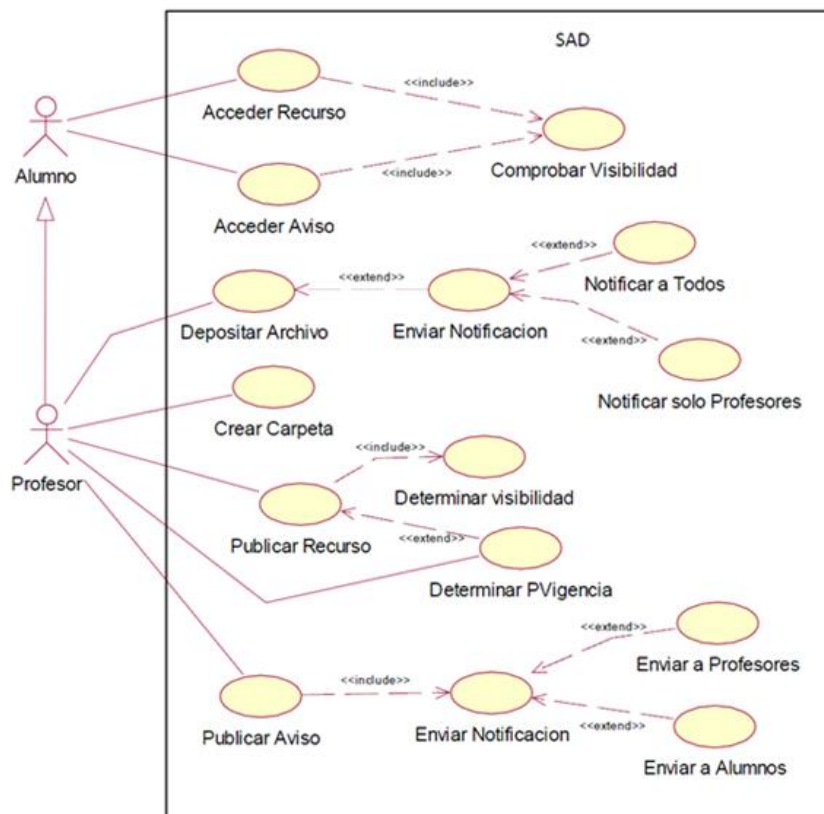
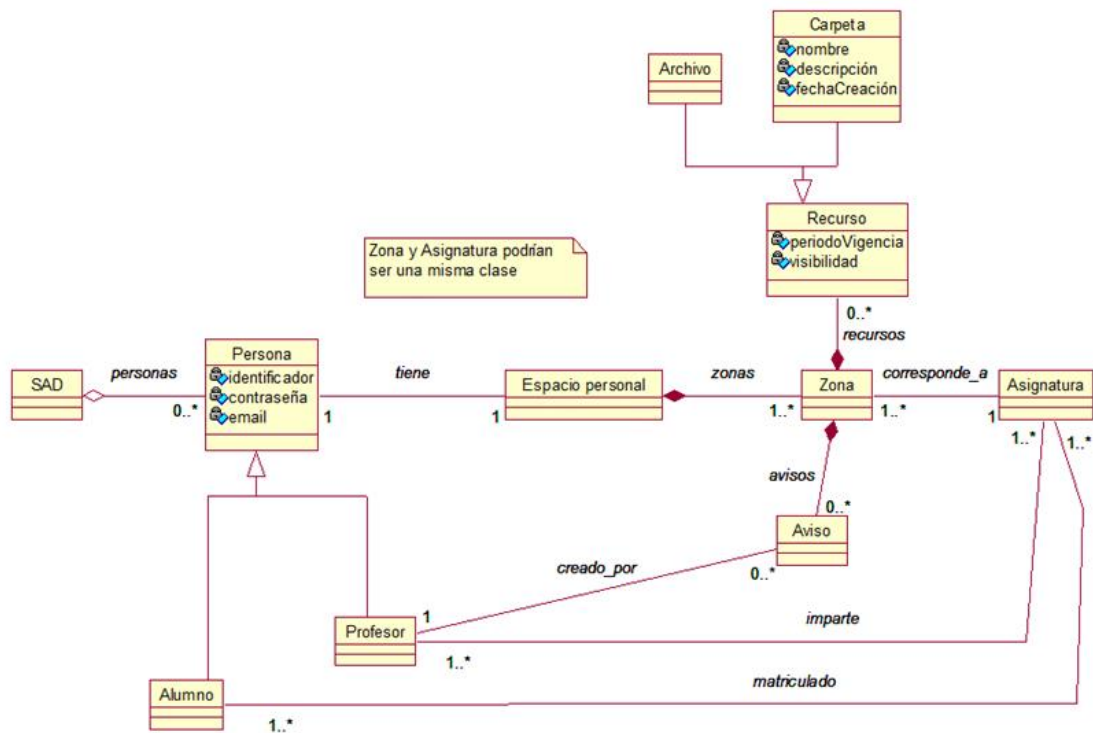


Diagrama de clases:



12. Banquetes Spress (ejercicio del boletín)

La Empresa “Banquetes Spress” ha contactado con la empresa “Soluciones ISG” para que sus ingenieros desarrollen una aplicación. El objetivo de la aplicación es dar servicio a la gestión de banquetes de bodas, bautizos, comuniones y comidas de empresas.

La empresa “Banquetes Spress” se dedica a la organización de banquetes de diferentes tipos, concretamente en bodas, bautizos, comuniones y comidas de empresa. Para ello, dispone de un local que se distribuye en diferentes salas, en la que cada sala se identifica con un nombre y tiene una capacidad para albergar un número determinado de comensales.

Cuando un cliente contacta con “Banquetes Spress” vía telefónica o personalmente, el comercial de la empresa toma los datos del cliente en el sistema (nombre, apellidos, e-mail, teléfono) y se cita con él un día a una hora determinada, para enseñarle las diferentes salas de las que se disponen en el local. También ofrece al cliente el servicio propio de menús que la empresa dispone, así como también servicios externos, que se pueden subcontratar por catálogo (discoteca móvil, barra libre, guardería, cocktails), informando en todos los casos del coste de cada servicio y la empresa que lo provee. Por ello, el sistema a desarrollar debe gestionar las diferentes citas que el comercial realiza con sus clientes, y de esa forma evitar solapamientos y poder atender al cliente de forma adecuada.

Si el cliente queda conforme con las instalaciones y servicios que se le han ofrecido, se realiza la reserva de un banquete, a celebrar en la sala escogida por el propio cliente. Para ello el sistema comprueba que la sala no está ocupada para el día que se hace la reserva. La reserva se formaliza mediante el pago de una señal y se indica la fecha y tipo de banquete que se va a contratar.

Una vez realizada la reserva, el cliente debe contratar de forma obligatoria un menú para el banquete con la propia empresa, y de forma opcional el resto de servicios, si lo desea.

Para la elección del menú, el cliente tiene de tiempo desde la fecha de reserva hasta un mes antes del banquete. Para ello, la empresa emplaza a su cliente a una fecha para la realización de las pruebas de menús, en las que el propio cliente prueba diferentes menús y finalmente selecciona el que quiere para su banquete, quedando así registrado en el sistema.

En cambio, para la contratación del resto de servicios opcionales el cliente tiene de tiempo hasta 10 días antes del evento. No es necesario contratarlos con la propia empresa, de forma que el propio cliente puede aportar la empresa que ofrece dicho servicio, quedando ésta registrada.

Una vez producido el evento, el cliente tiene 15 días para abonar la factura de todos los servicios prestados. Para ello previamente el gestor de “Banquetes Spress” ha generado la factura y se la ha enviado al cliente.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de contexto y el modelo de casos de uso en UML (incluir ejemplos de inclusión y extensión)

b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto:

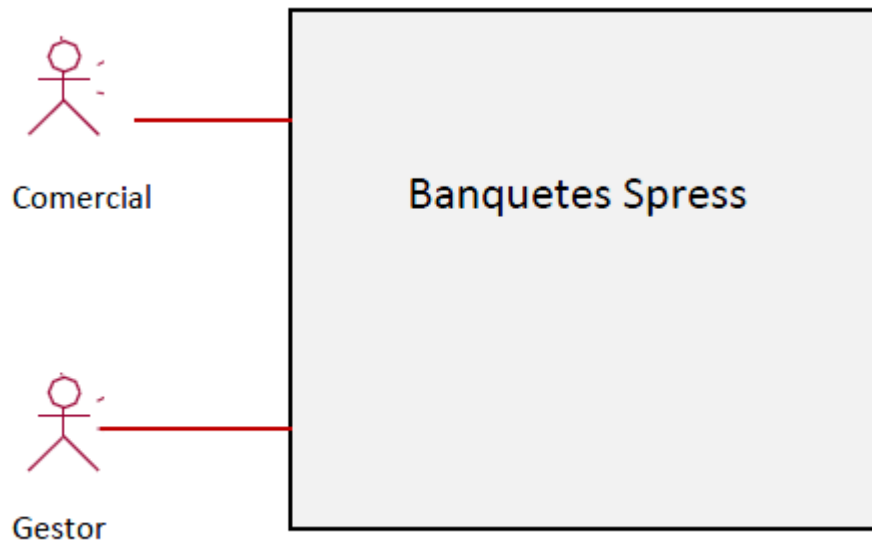
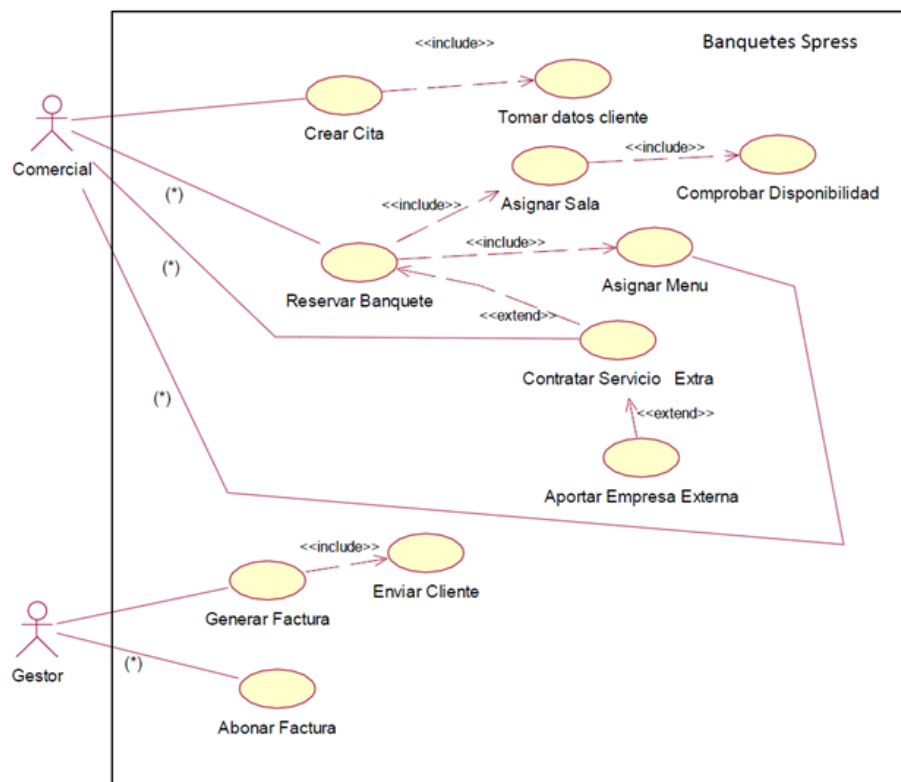
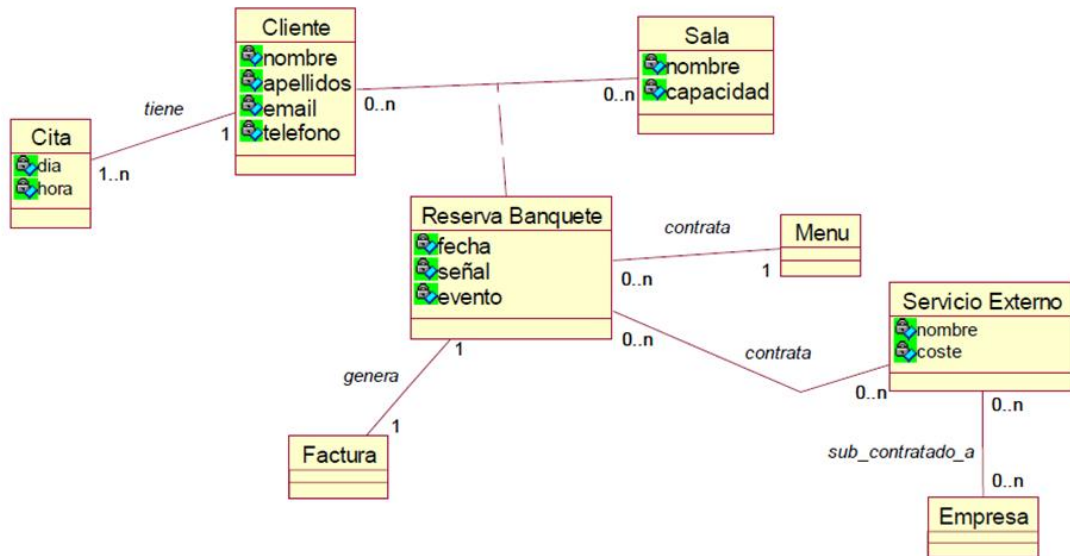


Diagrama de casos de uso:



(*) El problema no deja claro si lo hace el comercial o el gestor.

Diagrama de clases:



13. Gestión de una piscina

Se dispone de una piscina en las que se imparten cursillos y se permite el baño libre si hay calles disponibles. Los cursillos son impartidos por monitores, en un horario determinado, durante unos meses concretos y ocupan unas calles de la piscina. Un monitor puede impartir tantos cursillos como quiera siempre y cuando no coincidan en el horario.

Las personas pueden matricularse de los cursillos indicándoselo al recepcionista de la piscina. Cada cursillo tiene un número mínimo de inscritos, que si no se alcanza hace que no se imparta, y también un número máximo de alumnos. Los precios de los cursillos son mensuales y diferentes en función de si la persona está jubilada y/o pertenece al pueblo. En el momento de la formalización de la matrícula se debe de pagar una cuota correspondiente al primer mes del cursillo. Las personas pueden desmatricularse de un cursillo en cualquier momento vía el recepcionista.

Si por alguna razón un cursillo no se imparte, las personas inscritas podrán recuperar el dinero que se cobró en el momento de la formalización de la matrícula reclamándolo en la piscina. De las personas que se inscriben a los cursillos (llamados usuarios) se desea conocer su nombre completo, fecha de nacimiento, dirección completa, nº de cuenta en el que se cargan los recibos (no el primero que se cobra en el momento, pero sí los siguientes). De los monitores se desea conocer su nombre completo, dirección, nº de la seguridad social y nº de cuenta en el que se les ingresa la nómina.

Se desea conocer las calles libres que quedan en la piscina. Por supuesto, estas dependen de los cursillos que se impartan en cada momento. El listado debe mostrar para una semana dada cuántas calles quedan libres cada día a cada hora.

En cualquier momento, una persona cualquiera puede hacer uso de las calles que haya libres en ese momento. Obviamente, solo se podrá hacer si hay calles libres en ese momento.

Se necesita un listado para conocer cuánto dinero se recauda por los conceptos de baño libre y cuotas de los cursillos. Si ha habido devoluciones de cuotas también deberá aparecer en dicho listado.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de contexto y el modelo de casos de uso en UML (incluir ejemplos de inclusión y extensión)
- b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis.

Diagrama de contexto:

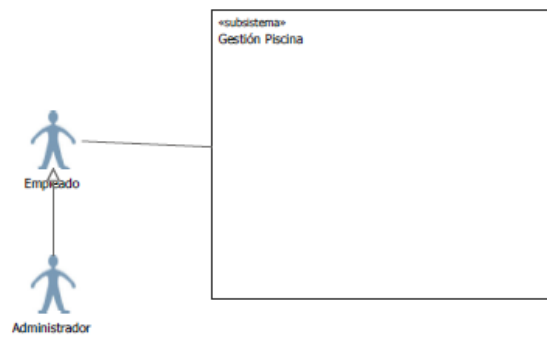


Diagrama de casos de uso

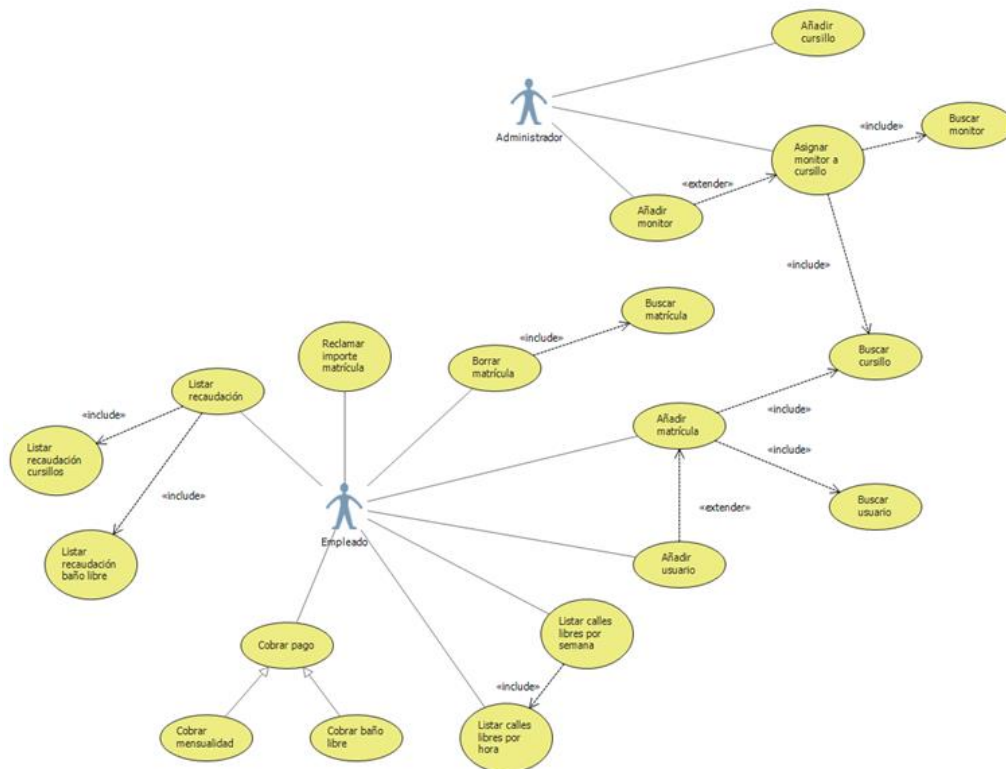
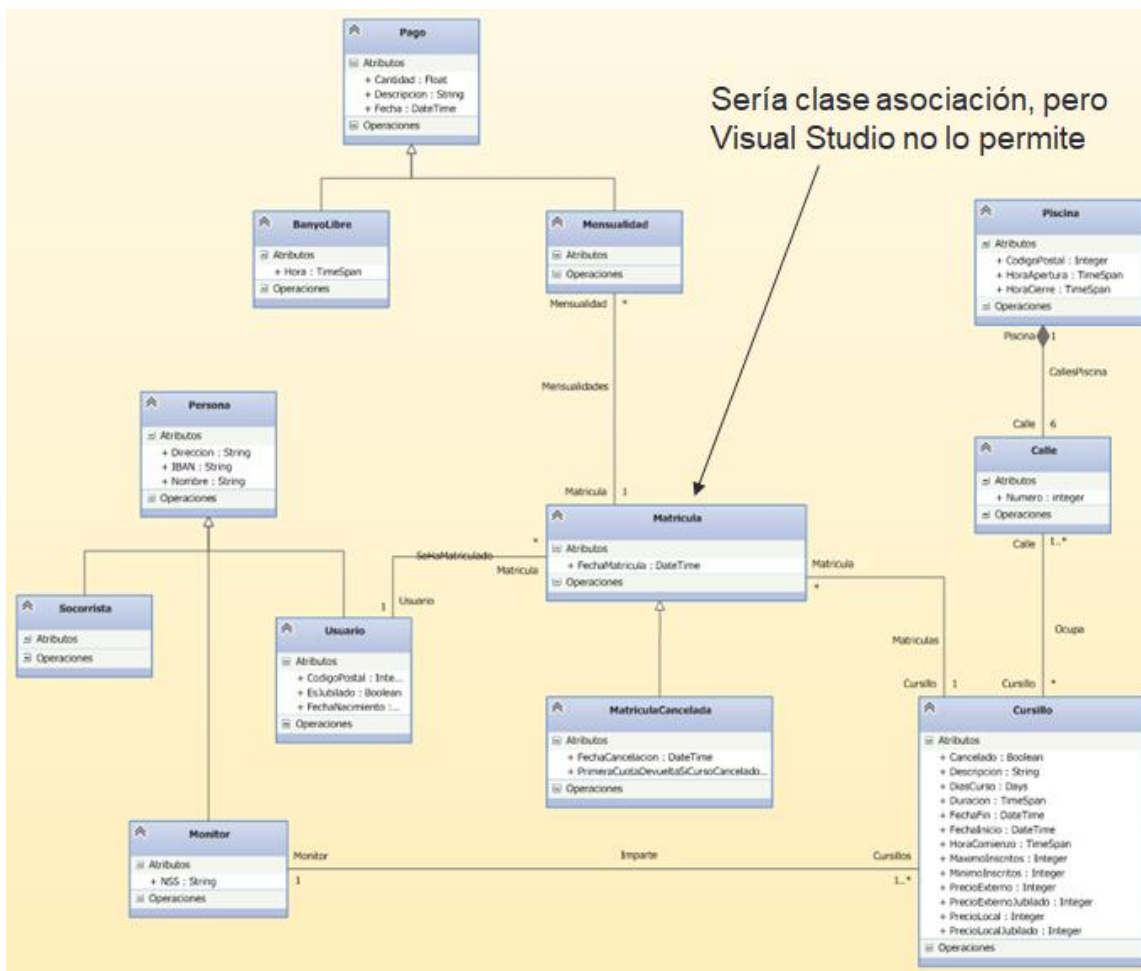


Diagrama de clases:



14. Gestión de un AMPA

Un AMPA de un colegio desea obtener una aplicación web de gestión tanto para poder gestionar los datos de sus asociados como para poder mantener adecuadamente la gestión de las actividades extraescolares que oferta en el centro.

Al AMPA puede inscribirse cualquier familia del centro, pagando para ello una cuota anual de 15 euros. El AMPA quiere que en su sistema pueda inscribirse a través de su web cualquier modelo de familia, tanto monoparentales como aquellas con progenitores del mismo sexo. Para inscribirse, una familia debe aportar nombre, apellidos, DNI, email y teléfono de contacto de cada progenitor. Además, deben introducir el nombre y apellidos de los alumnos a su cargo, así como del curso al que asiste en el centro. El AMPA quiere poder obtener en cualquier momento un listado actualizado de todas las familias asociadas, así como un listado de todos alumnos cuyas familias son socias, agrupados por curso.

Por otra parte, la mayor parte del trabajo del AMPA reside en la gestión de las actividades extraescolares que los alumnos/as pueden realizar en el centro. Hasta ahora, eran los voluntarios del AMPA quien registraban todos los datos, pero ahora desean que las familias puedan inscribir directamente a los niños en la web. Para ello se abre un periodo inicial de matrícula, tras el cual los alumnos/as solo pueden inscribirse si quedan plazas, puesto según la actividad existe un número mínimo y máximo de plazas.

Las actividades son fundamentalmente de dos tipos: gestionadas por las escuelas deportivas municipales (EEDD) y gestionadas por una empresa externa. Cada actividad se realiza en un tipo de instalación diferente (aula, pista exterior, pabellón, trinquet, etc), con diferentes características: interior o exterior, capacidad, etc. Las actividades de EEDD se realizan únicamente a mediodía, y solo pueden inscribirse alumnos/as de tercer curso en adelante. La cuota es anual y no es posible darse de baja.

En la inscripción del alumno/a se debe incluir su nombre y apellidos, su NIA (Número de Identificación del Alumno), DNI (si lo tiene), su curso, su fecha de nacimiento, su SIP, un número de cuenta bancaria (IBAN), un email y un teléfono de contacto.

Por otra parte, las actividades gestionadas por empresas se realizan por la tarde, pudiendo inscribirse cualquier alumno/a del centro, independientemente de su curso. Los alumnos pueden inscribirse o borrarse de una actividad en el momento que deseen, siendo la cuota de pago de carácter mensual.

En el caso de inscribirse a una actividad de tarde, es necesario pagar un único cargo anual de 25 euros para las familias socias del AMPA. Las familias no socias del AMPA deben pagar 50 euros. Las familias deberán adjuntar su recibo de pago en el proceso de inscripción, indicando el importe del mismo.

Además, el AMPA necesita mantener un registro de todos los monitores que imparten cada actividad extraescolar. De cada uno de ellos quiere guardar su nombre, apellidos, email y teléfono de contacto. Además, quiere poder almacenar digitalmente su certificado de delitos de carácter sexual, para cumplir con la normativa vigente.

El AMPA necesita poder obtener un listado en el que aparezcan todas las actividades extraescolares que oferta, tanto de EEDD como de empresa. Por cada una de ellas necesita poder mostrar el nombre de la actividad, su horario, su cuota, su monitor, así como el número máximo y mínimo de participantes. También debe aparecer el tipo de instalación en la que se realiza. Además, necesita obtener un listado de alumnos inscritos por cada actividad, indicando su nombre, apellidos y curso.

Para agilizar la notificación a los padres y madres tras el periodo inicial de matrícula, el AMPA desea que el sistema notifique automáticamente a las familias por correo electrónico si la actividad en la que ha matriculado a su hijo o hija finalmente se realizará atendiendo al número de plazas disponible. Si no se alcanza el número mínimo, la actividad se elimina. Si se excede el número máximo, el sistema indica en su mensaje que se realizará un sorteo para decidir qué alumnos/as entran en la actividad.

A partir de la descripción anterior, se pide:

a) Construir el diagrama de casos de uso en UML.

b) Construir el diagrama de clases en UML, incluyendo los atributos de las clases, así como los nombres de las relaciones que detectéis.

Diagrama de casos de uso

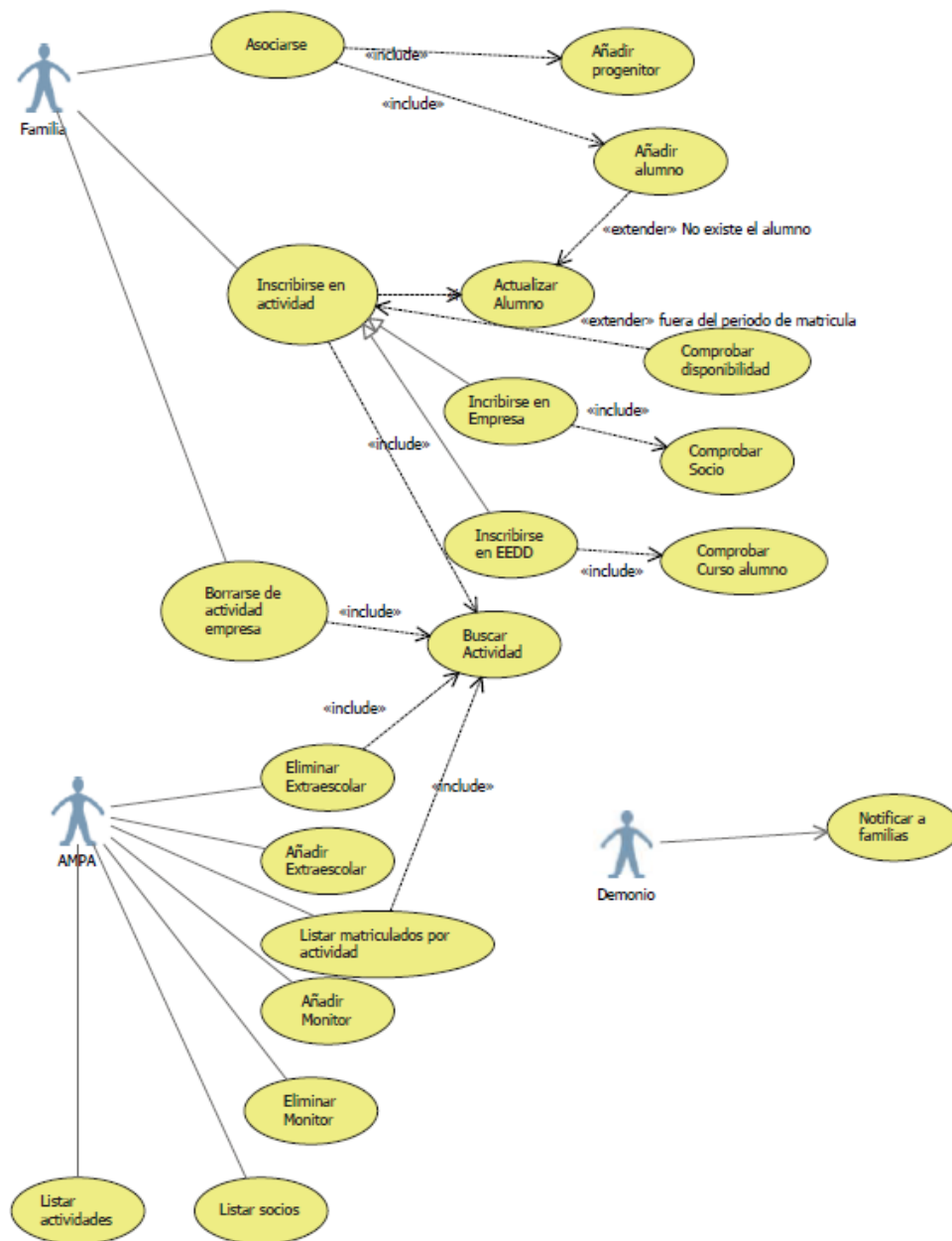
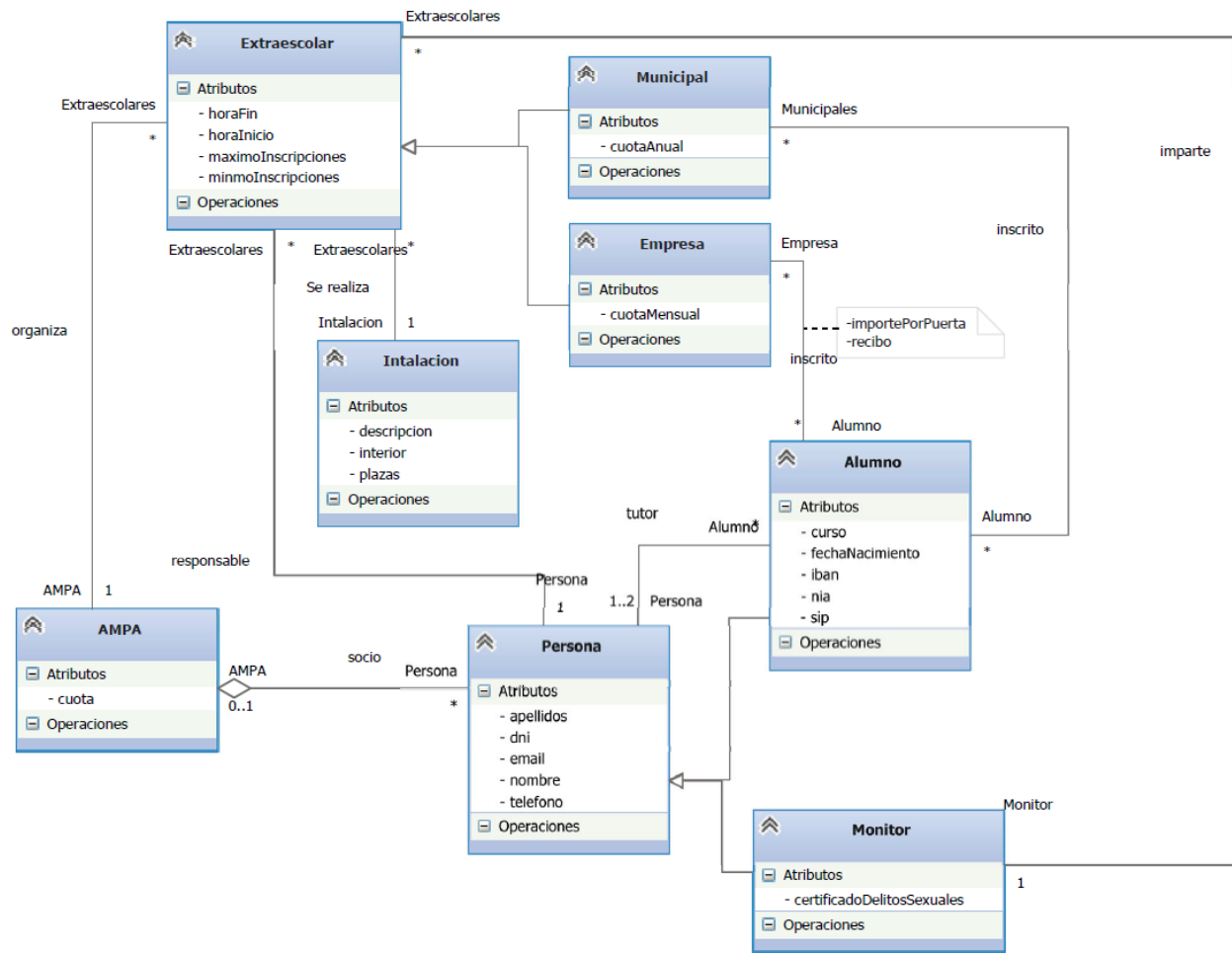


Diagrama de clases



15. Gestión del Servicio Técnico de un Concesionario

El concesionario *TuCoche, S.A* comenzará a prestar un nuevo servicio post-venta a sus clientes que consiste en un taller mecánico destinado la revisión y reparación de los vehículos vendidos por éste. Para ello, pretende obtener un nuevo sistema de gestión que le permita controlar las reparaciones pendientes de realizar y facturar los trabajos realizados a sus clientes.

Mediante este sistema, a la llegada de un nuevo cliente, los empleados de recepción crearán una nueva orden de trabajo en el que se especificará la matrícula del vehículo, en qué plaza ha sido aparcado, los datos del cliente y una descripción de los trabajos a desarrollar, quedando en el estado "pendiente de realización". Además, se genera una etiqueta con la matrícula del vehículo y el número de orden de trabajo que es añadida a las llaves entregadas por el cliente. El empleado pegará la etiqueta a las llaves y las colgará en el tablón "en reparación".

Los datos del cliente se recuperan desde su ficha, a partir de su NIF o del CIF de la empresa, en caso de que el vehículo pertenezca a una empresa. Si el cliente no estuviera en el sistema, el empleado deberá crearle una ficha aportando sus datos (nombre, dirección, código postal, provincia, NIF, teléfono fijo, teléfono móvil y correo electrónico), indicando si quiere recibir notificación del estado de la reparación por mensaje al móvil o por correo electrónico. En el caso de las empresas, en lugar del NIF se registrará el CIF y además del nombre de la empresa, deberá aportarse el nombre de una persona de contacto.

El jefe del taller consultará desde su sistema qué vehículos han entrado para reparar, siendo el encargado de asignar a cada trabajo el mecánico o mecánicos necesarios. Cada nueva asignación le llegará a cada mecánico a su PDA. Así, cuando estén disponibles para empezar un nuevo trabajo, aceptarán el primero de los que tengan pendientes, marcando la orden de trabajo correspondiente con el estado "en proceso" y se dirigirán al panel de llaves situado en la recepción para coger las llaves del vehículo e introducir el coche del parking de clientes al taller. La PDA registrará la hora y fecha en la que el mecánico ha marcado el inicio del proceso de reparación. Esta marca temporal será asignada a la orden de trabajo, a fin de calcular las horas de trabajo realizadas. Cuando el mecánico realice una pausa en el trabajo de reparación, deberá indicarlo al sistema mediante su PDA. Además, el mecánico añade a la orden de trabajo las piezas o productos que haya utilizado. Cuando finalice una reparación, la marcará como "realizada", pudiendo añadir cualquier comentario que considere oportuno.

Cuando una orden de trabajo sea marcada como realizada, el sistema se encargará de avisar al usuario mediante un mensaje SMS o un correo electrónico de que el coche ya está disponible.

En administración, los empleados realizarán las facturas conforme llegan los clientes, generando para cada uno de ellos una factura de la orden de trabajo con los datos fiscales del concesionario. El importe de la factura de calculará como la suma del importe de las piezas, productos y horas de trabajo de los mecánicos reflejada en la orden de trabajo. En la factura, además de su número, fecha, y datos de los clientes, se añadirá la descripción de las piezas/productos utilizados, así como los comentarios del mecánico, si los hubiera. Cada orden de trabajo facturada pasa de forma automática al estado "facturada". El importe total de la factura podrá ser abonado en metálico o mediante tarjeta de crédito. Una vez pagada, desde administración marcan la factura como "abonada", dando permiso a recepción para que entreguen las llaves al cliente.

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de casos de uso en UML.
- b) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

Diagrama de casos de uso:

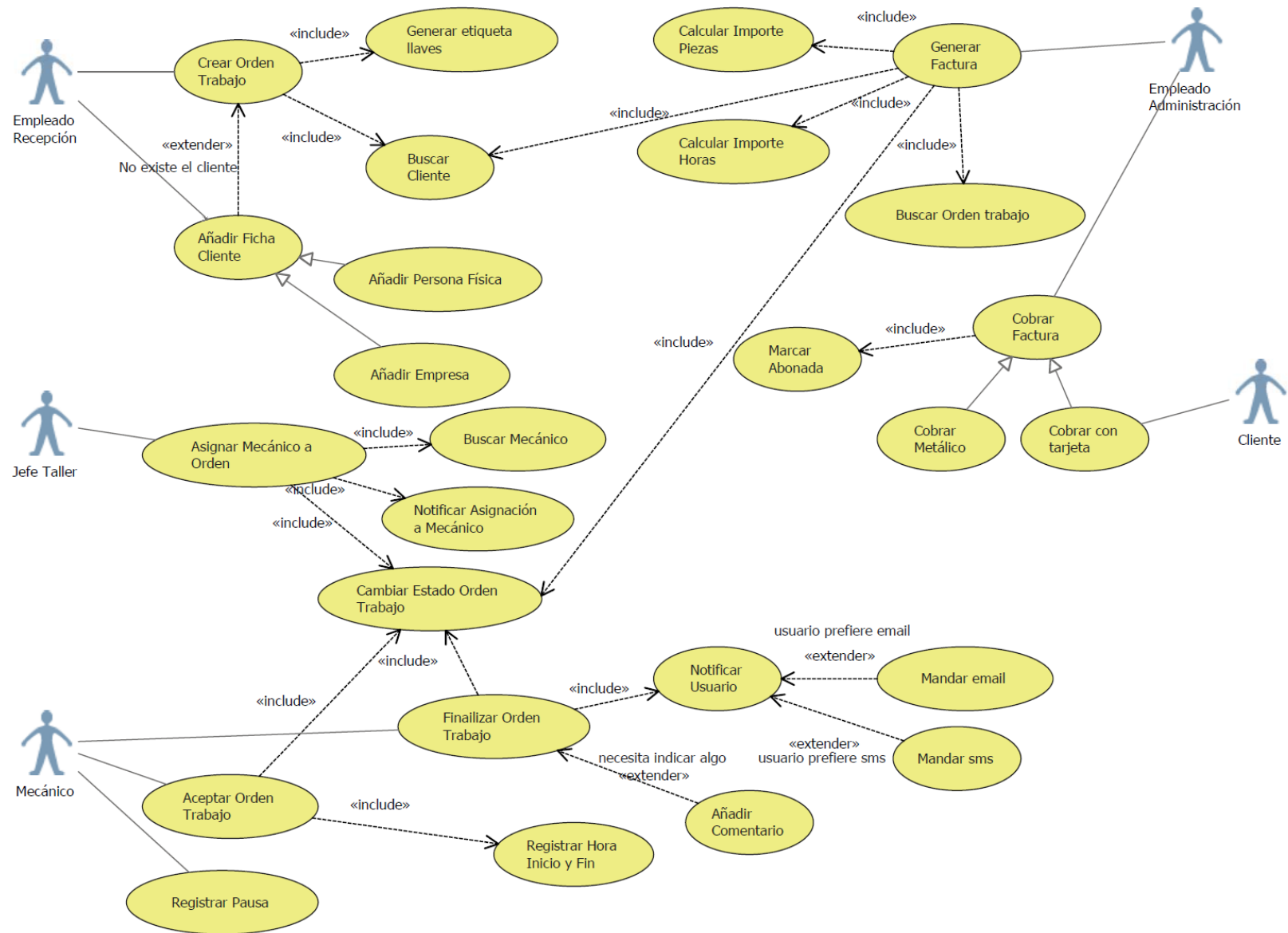
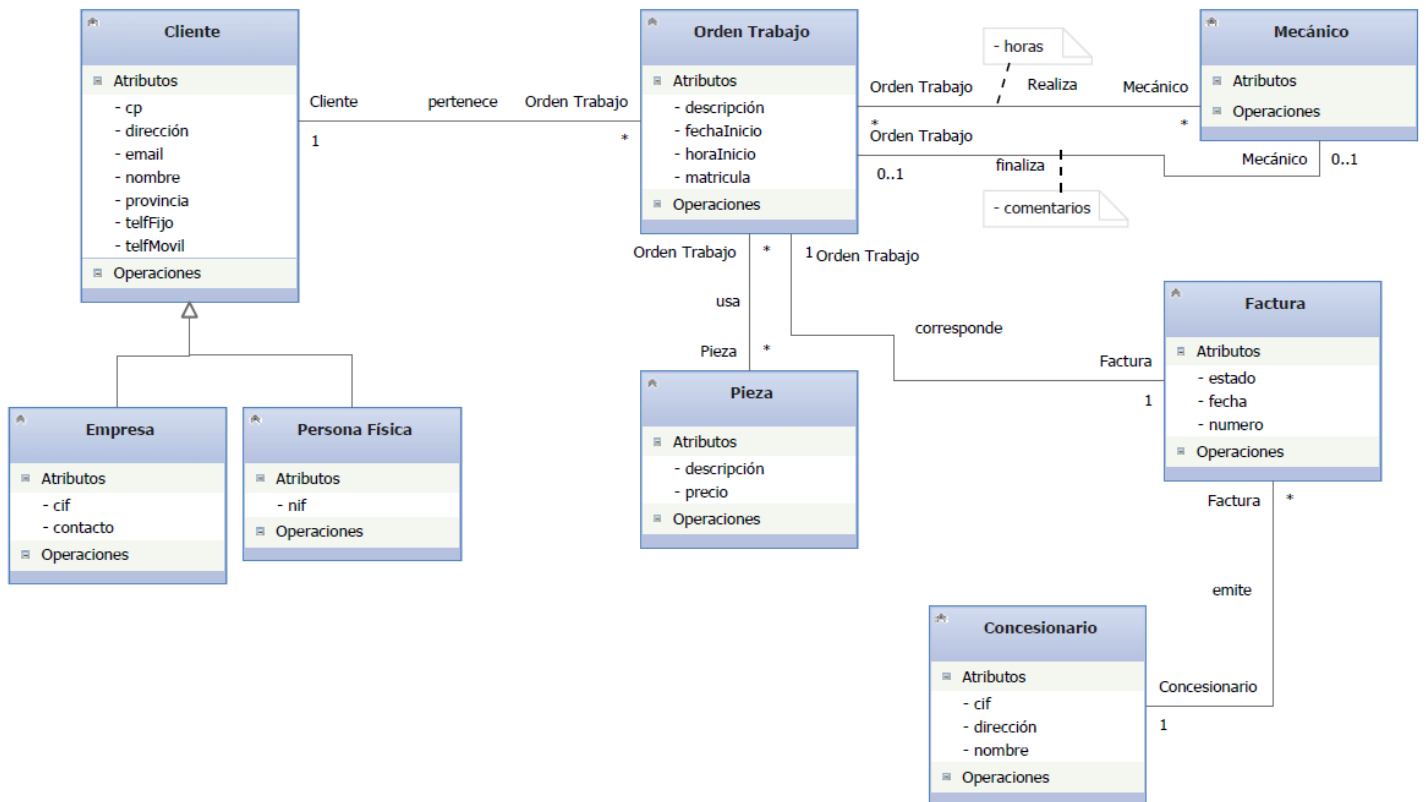


Diagrama de clases:



16. La Fábrica ACERUPV

La fábrica *ACERUPV* ha contactado con los profesores de ISW para que le desarrollemos un sistema informático que le asista a soportar los procesos productivos de la fábrica. La fábrica produce varillas redondas de acero como resultado de la mecanización de bloques de acero. *ACERUPV* es capaz de fabricar varios tipos de varillas que pueden ser de diferente calibre (diámetro de la varilla), longitud y peso. Cada una de estas varillas tiene un código, un nombre y un coste de fabricación (por unidad fabricada).

La fábrica posee varias líneas de producción que transforman los bloques de acero en varillas. Estas líneas están ubicadas en diferentes naves de la fábrica. Cada línea de la fábrica tiene un código de identificación y una ubicación.

Los clientes (de los que se sabe su razón social, CIF y dirección) realizan encargos indicando la varilla que desean (especificando su código o sus características) y las unidades que desean. La fábrica genera una orden de fabricación por cada encargo.

La fabricación de una cantidad de varillas se especifica mediante una orden de fabricación. Una orden de fabricación puede llevarse a cabo en una o varias sesiones de trabajo. En cada sesión de trabajo se anota la hora de inicio y la de fin de la sesión, así como cuantas unidades se han fabricado en dicha sesión al finalizar la sesión.

Una sesión de trabajo se lleva a cabo en una de las líneas de producción, e involucra a varios trabajadores: a uno que actúa como 'jefe' (controla el proceso) y a uno o varios trabajadores que actúan de 'operarios' (asisten a la línea cargando los bloques de acero y retirando las varillas ya fabricadas). Es posible que el mismo trabajador participe como jefe. De cada trabajador el sistema conoce su nombre y teléfono de contacto.

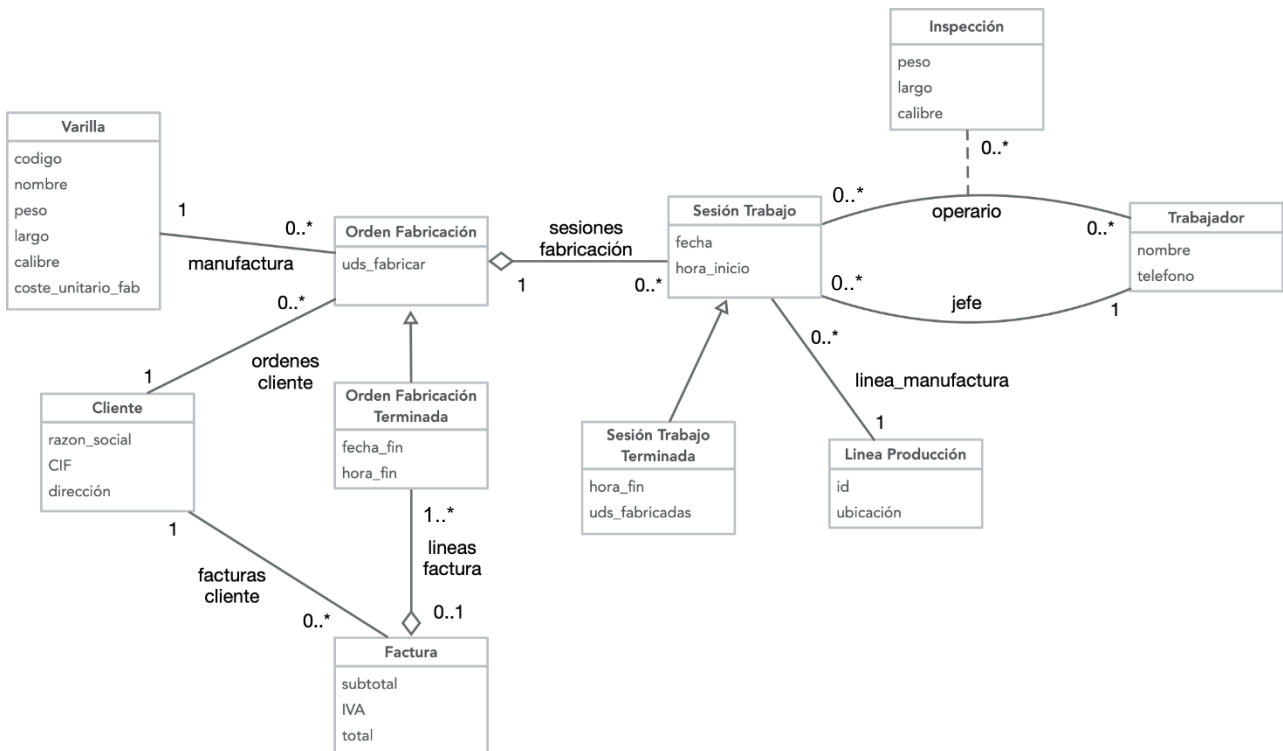
Los operarios deben realizar inspecciones durante la sesión de trabajo. Para ello toman una varilla fabricada y miden su longitud, calibre y peso, e introducen esta inspección en el sistema.

Cuando ya se han fabricado todas las varillas de una orden de fabricación (en las sesiones de trabajo), la orden pasaría a completada (especificando la fecha y hora de finalización de la producción) y se facturaría al cliente. La factura está constituida por las órdenes de fabricación completadas (una o varias), así como información del subtotal, IVA y total de la factura (unidades fabricadas x coste de fabricación por unidad).

A partir de la descripción anterior, se pide:

- a) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

SOLUCIÓN



17. Servicio de Alertas Medioambientales

Una empresa que fabrica sensores ambientales ha empezado desplegar dispositivos (sensores) por diferentes regiones con el objetivo que se puedan usar para que sus usuarios puedan configurarse (de manera personalizada) servicios de alertas medioambientales. De esta manera, un agricultor, por ejemplo, puede configurar un servicio de alerta medioambiental para que se le envíe una alerta si algún sensor de temperatura o de humedad relativa, de entre los que tiene la empresa desplegados cerca de su campo, lanzan una alerta.

El modo de funcionamiento del servicio será el siguiente. Los usuarios se registran a la aplicación con su id, y su nombre. Cada usuario podrá configurar sus propios servicios de alertas medioambientales (podrá tener varios). Al configurar un nuevo servicio de alertas, se le asignará un código y el usuario deberá indicar una descripción. Podrá asociar a este servicio al menos un sensor, pero podrá seleccionar varios (de entre los que se encuentren disponibles).

La empresa cuenta con un conjunto de sensores ya desplegados, y planea seguir desplegando nuevos. Cada sensor dispone de un nombre, una unidad de medida, y una ubicación GPS, compuesta por una latitud, longitud y altura.

Los sensores reportan periódicamente lecturas sobre las mediciones realizadas. El sistema almacena estas mediciones guardando el valor recibido y un *timestamp* del momento que se recibe, para poder generar gráficas del histórico de mediciones. Cada vez que un sensor envía una nueva lectura, esta se anota como la 'última medición'.

En la actualidad, la empresa sólo dispone de tres tipos de sensores: de temperatura, pluviómetros y de humedad relativa. Cada tipo de sensor tiene definidos umbrales que puede configurar la empresa para que se genere una alerta al rebasar dicho umbral. Al rebasar una

Quando un sensor genera una lectura que rebasa un umbral, el sistema crea una alerta para cada uno de los servicios de alertas medioambientales que están configurados con este sensor. Esta alerta incluye un motivo (que genera automáticamente el sistema), junto a una fecha y una hora. Además, a cada alerta se le asocia un tipo de alerta de entre un conjunto de tipos de alertas predefinidas.

a) Obtener el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis. No es necesario indicar los métodos.

```

classDiagram
    class Localización {
        - latitud: double
        - longitud: double
        - altura: double
    }
    class Sensor {
        - nombre: string
        - unidad de medida: string
        - umbral Superior: double
    }
    class AlertaMedioambiental {
        - codigo: integer
        - descripción: string
    }
    class Alerta {
        - motivo: string
        - fecha: date
        - hora: localTime
    }
    class Usuario {
        - id: integer
        - nombre: string
    }
    class TipoAlerta {
        - nombre: string
    }
    class Lectura {
        - valor: double
        - timeStamp: localdate
    }
    class Pluviometro
    class Temperatura {
        - umbral inferior: double
    }
    class HumedadRelativa {
        - umbral inferior: double
    }

    Localización "1" -- "1..*" Sensor : Se sitúa
    Sensor "1..*" -- "0..*" AlertaMedioambiental : Monitoriza
    AlertaMedioambiental "0..*" -- "0..*" Alerta : 
    Alerta "0..*" -- "1" Usuario : Pertenece
    Alerta "0..*" -- "1" TipoAlerta : Pertenece
    Alerta "1" -- "0..*" Sensor : Reporta
    Sensor "1" -- "0..*" Lectura : Última medición
    Lectura "0..1" -- "0..*" Sensor : 
    Pluviometro --> Sensor
    Temperatura --> Sensor
    HumedadRelativa --> Sensor
    
```