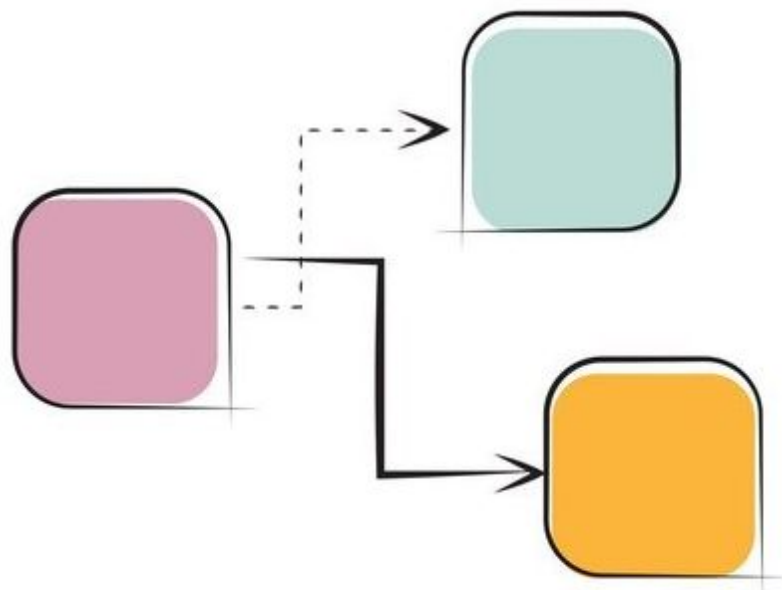


Ingeniería del Software

Problemas de modelización UML

Diagramas de Clases



Problema 1: Agencia de medio ambiente

Queremos desarrollar un Sistema de Información para una agencia de medio ambiente que quiere documentar casos prácticos de éxito empresarial en innovación.

Un caso práctico, que siempre se identifica por un nombre, **hace referencia** a un subsector, el cual, al mismo tiempo, siempre hace referencia a un sector que lo engloba. Cada sector y subsector se identifican por un nombre. Al mismo tiempo, un caso práctico siempre corresponde a una empresa, de la que sabemos el nombre, el número de trabajadores, el país de origen y una descripción breve.

Por ejemplo:

El caso práctico de "Los supermercados Supermarca", se encuentra en el subsector de la Distribución, que se encuentra en el sector de la Alimentación. Hace referencia a la empresa Supermarca, que tiene 74.000 empleados, es de España y es una cadena de supermercados. El caso práctico "Los cafés soleados", se encuentra en el subsector del Café, que se encuentra en el sector de la Alimentación. Hace referencia a la empresa Café Insoluble, que tiene 1.000 empleados, es de EEUU y es una empresa elaboradora de café instantáneo.

Las empresas ofrecen productos, y en nuestro sistema queremos tener documentados un máximo de 10 de estos productos. De cada producto sabemos el nombre y una breve descripción. A algunos de estos productos, la empresa aplica estrategias innovadoras.

Toda estrategia tiene una descripción donde consta una explicación de la misma. Pero además, toda estrategia es de un solo tipo: o bien es una estrategia basada en costes (es decir, en intentar reducir los costes) o bien en diferenciación (es decir, al intentar diferenciarse de la competencia). En el caso de los costes, nos interesa saber en qué área han reducido los costes (compra de mercancías, elaboración o logística).

Así, la empresa Café Insoluble tiene dos productos, unas cápsulas de café instantáneo para poder elaborar cafés expreso, y envases de café soluble.

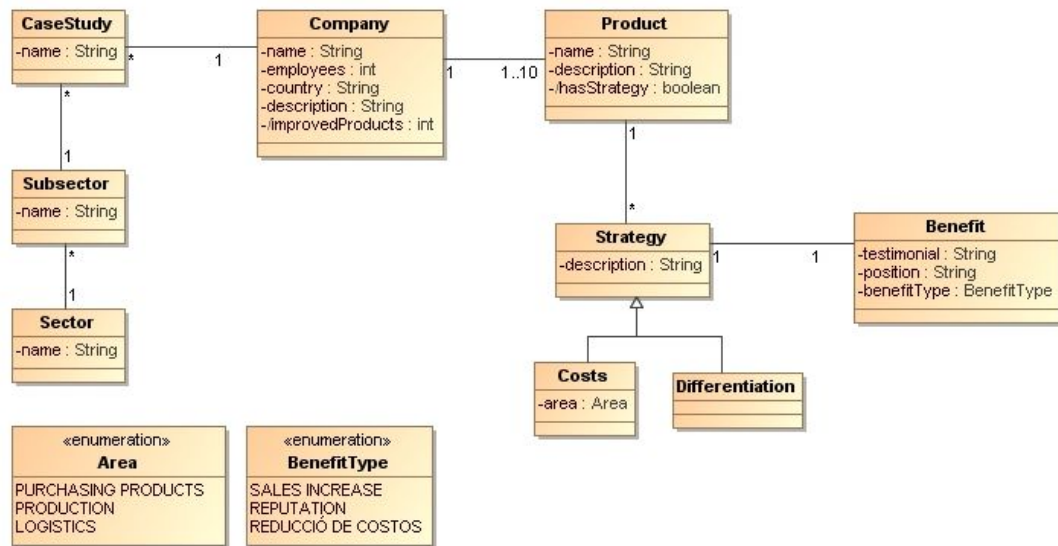
Para este segundo producto, los envases de café soluble, la empresa ha adoptado una estrategia de crear envases rellenables de manera que el ciclo de vida del producto sea más largo. Es una estrategia basada en costes que impacta en la logística.

Finalmente, cuando una estrategia se pone en práctica con un producto, hay que hacer notar los beneficios que se han obtenido. Por este motivo es necesario documentar el tipo de beneficio que ha habido (incremento en ventas, reputación, reducción de costes) y una frase de testimonio de alguna persona de la empresa para la que hay que indicar el cargo de la misma.

Así, para el caso de la empresa Café Insoluble y su producto de Café soluble, la estrategia aplicada se tradujo en un incremento en ventas, y en palabras de su CEO: "Estamos orgullosos de lo que hemos hecho y de cómo nuestros clientes nos están ayudando".

Para el sistema es importante que cada caso práctico sepamos cuántos productos se tienen documentados y cuántos de ellos tienen una estrategia aplicada.

Diagrama de classes:



Restricciones de clave:

- CaseStudy: name
- Sector: name
- Subsector: name
- Company: name
- Product: name

Otras restricciones de integridad:

- (implícita) Una Strategy puede ser de tipo Costs o de tipo Differentiation, pero no de ambos tipos a la vez.

Información derivada:

- /hasStrategy: se calcula en función de si el producto tiene una estrategia asociada o no.
- /improvedProducts: se calcula a partir del número de productos que tienen asociada una estrategia.

Problema 2: Agencia inmobiliaria

Queremos desarrollar un Sistema de Información para una agencia inmobiliaria que quiere documentar sus ofertas de inmuebles.

La agencia inmobiliaria está estructurada en una sede y varias sucursales. De la sede sabemos el nombre (de hecho el nombre de la empresa) y de las sucursales sabemos también el nombre y la población donde operan. La dirección de la sede es muy estricta en referencia a la publicación de las ofertas inmobiliarias y desean dar un paso adelante tecnológico que les permita remontar el vuelo en un sector muy criticado y en crisis estos últimos tiempos.

Los inmuebles se clasifican según una categoría de la que tenemos el nombre. Un inmueble es responsabilidad de una sucursal, que es quien lo da de alta y hace el mantenimiento. De los inmuebles, básicamente, tenemos esta información: una referencia (código), la dirección completa, los metros cuadrados, el número de dormitorios y el estado general del mismo (nuevo, bueno, a reformar).

Por ejemplo, la sucursal de Vic es responsable de estos inmuebles:

En la categoría de "piso", un inmueble en buen estado, en la Calle Mayor, 4 (Vic), de 80 m², con 3 dormitorios.

En la categoría de "dúplex", un inmueble a reformar, en la Calle Caracola, 7, 3º (Vic), de 130 m² con 3 dormitorios.

Los inmuebles pueden estar en venta, en alquiler o en venta y alquiler a la vez. Así, a un inmueble se le puede asignar al mismo tiempo un precio de venta y un precio de alquiler. Los dos precios tienen un importe en euros, pero el precio de venta, además, tiene un descuento, en tanto por ciento (que la agencia decide aplicar o no en función de cada caso) y el precio de alquiler tiene un importe de fianza, también en euros, y una periodicidad (mensual, trimestral o anual). El sistema necesita saber si un inmueble es o no de alquiler o venta, y lo sabe en función del precio asignado en cada caso.

El inmueble de la calle Caracola tiene un precio de venta de 120.000 euros, con un posible descuento del 7,5%; pero al mismo tiempo también está disponible como alquiler, 600 euros mensuales con una fianza de 500 euros.

En cambio, el inmueble de la calle Mayor está disponible para ser alquilado por 750 euros al mes con una fianza de también 750 euros.

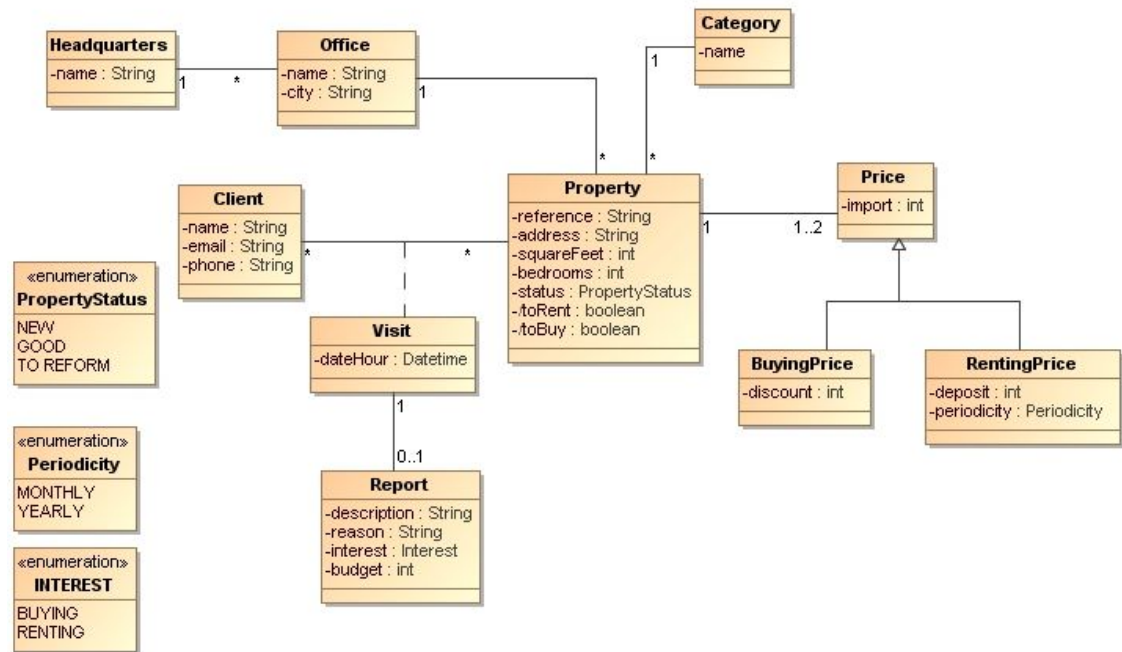
Por último, la sede está muy interesada en conocer el funcionamiento de las visitas a los inmuebles por parte de posibles arrendatarios o compradores y cuál es su principal interés. De cada posible cliente (arrendatario o comprador) tenemos el nombre, el teléfono y el email. Cada visita a un inmueble acuerda con un cliente por un día y una hora determinados. Para simplificar, supondremos que un cliente sólo puede visitar un inmueble concreto en una sola ocasión.

Cada vez que se produce una visita a uno de estos inmuebles, la sucursal encargada de realizar la visita debe, una vez finalizada la misma, completar un informe en el que se expone una breve descripción de la visita realizada y donde se indica si la persona está más interesada en compra o alquiler y el presupuesto disponible (total en caso de compra, mensual en caso de alquiler).

Para el inmueble de la calle Caracola, hay una visita prevista el próximo martes día 4 de julio de 2014 a las 10.00 con Ramon Pla (rpla@rpla.net y 991112233, email y teléfono respectivamente). Una vez realizada la visita, el agente indica que la visita no ha ido demasiado bien, que no era lo

que estaban buscando, buscaban una terraza más grande y un aparcamiento. Estaban interesados en alquilar con un presupuesto de 650 euros al mes.

Diagrama de clases:



Restricciones de clave:

- Headquarters: name
- Office: name
- Category: name
- Property: reference
- Client: name

Otras restricciones de integridad:

- (implícita) No se pueden producir dos visitas al mismo tiempo en una misma fecha y hora.
- (implícita) En caso de que un inmueble tenga dos precios, uno debe ser de compra y el otro de alquiler.

Informació derivada:

- /toRent: se calcula en función de si el inmueble tiene asociada una instancia de RentingPrice.
- /toBuy: se calcula en función de si el inmueble tiene asociada una instancia de BuyingPrice.

Problema 3: Restaurante italiano

Queremos desarrollar un Sistema de Información para un restaurante italiano que quiere documentar las salsas que utiliza para elaborar sus platos.

El restaurante dispone de un catálogo de todas las salsas que elabora clasificándolas en categorías jerarquizadas. De cada categoría conocemos el nombre, que la identifica, y una breve descripción. De cada salsa también conocemos el nombre, una descripción y los platos que puede acompañar (de los cuales, de momento, sólo queremos guardar el nombre que los identifica).

Por ejemplo, el restaurante elabora, entre muchas otras, estas salsas:

En la categoría de "Salsas para postres", dentro de la sub-categoría de "Salsas frías" encontramos el "Coulis de fresa", que acompaña el "Panna cotta".

En la categoría de "Salsas para primeros", encontramos la salsa "boloñesa", que puede acompañar a los "spaghetti", los "fettuccini", los "fusilli" o los "penne"....

Para elaborar las salsas hay que recurrir a la receta, donde consta el tiempo de preparación (en minutos), el número de personas para las que es la receta, los ingredientes necesarios y una descripción del procedimiento para elaborarla. Una receta incluye varios ingredientes, de cada uno de los cuales hay que especificar la cantidad y las unidades (que pueden ser gramos, litros, ml, piezas, etc.). De los ingredientes, que pueden ser de origen animal o de origen vegetal, conocemos el nombre y la zona geográfica de la que proceden. De los ingredientes de origen animal, además, queremos saber si han sido criados en libertad.

Además de la información anterior, de las salsas queremos saber si son vegetarianas (es decir, si únicamente contienen ingredientes de origen vegetal) o bien si son ecológicas (si todos los ingredientes de origen animal que contienen han sido criados en libertad).

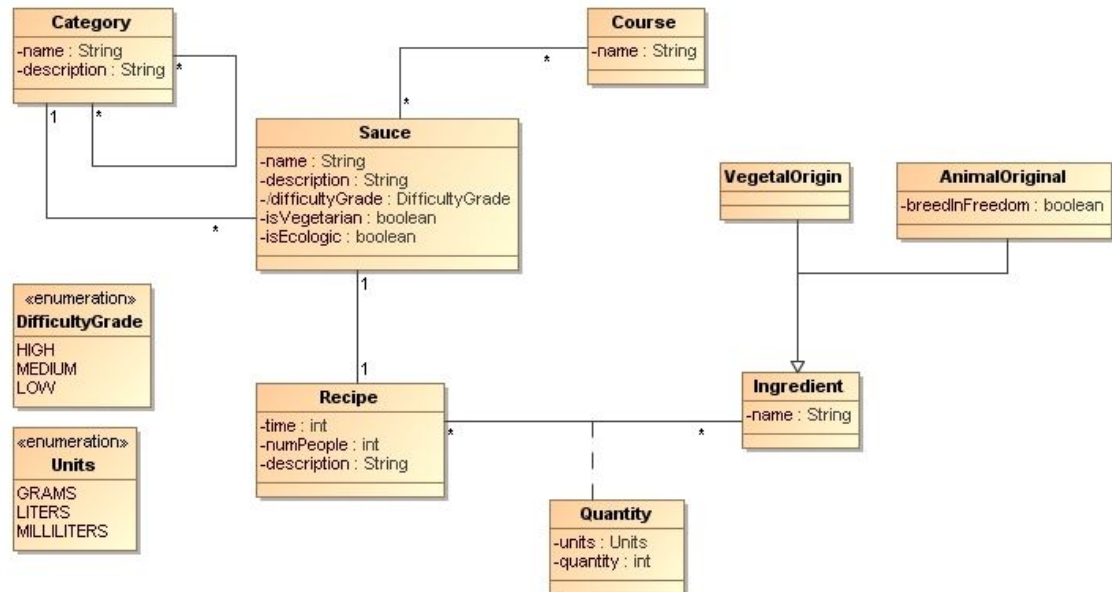
Por ejemplo, la descripción de una receta puede tener un aspecto como el siguiente:

Para hacer la salsa "boloñesa", que es ecológica, en una receta para 4 personas y un tiempo total de 75 minutos se necesitan 500 gramos de carne de ternera de Girona criada en libertad y 100 gramos de cebolla de Figueres, entre otros ingredientes. Para elaborarla hay que empezar picando la cebolla...

El restaurante tiene muchos estudiantes de cocina como becarios, y por eso quiere etiquetar cada salsa con un nivel de dificultad (bajo, medio o alto) que se calcula según el tiempo de preparación y de la cantidad de ingredientes que son usados en la receta.

Por ejemplo, la salsa "boloñesa", por su tiempo de preparación e ingredientes implicados, tiene un nivel de dificultad "media"; en cambio, el "Coulis de fresa" tiene un nivel de dificultad "bajo".

Diagrama de clases:



Restricciones de clave:

- Category: name
- Sauce: name
- Ingredient: name
- Course: name

Otras restricciones de integridad:

- No n'hi ha

Informació derivada:

- /difficultyGrade: se calcula en función del tiempo y la cantidad de ingredientes utilizados en la receta
- /isVegetarian: se calcula en función de los ingredientes utilizados en la receta
- /isEcologic: se calcula en función de los ingredientes utilizados en la receta

Problema 4: Lista de tareas

Queremos desarrollar un sistema de información que gestione una lista compartida de tareas.

En concreto, queremos tener una serie de listas de tareas (cada una con su nombre) cada una de las cuales tendrá una serie de tareas (también cada una con su nombre). Una tarea puede ser asignada a un usuario (los identificamos por email).

Por ejemplo, la lista de tareas "Enunciado EP" tiene asociadas tres tareas: "Pregunta 1", "Pregunta 2" y "Pregunta 3".

La tarea "Pregunta 1" está asignada al usuario "abc@example.com" mientras que el resto de tareas aún no están asignadas a nadie.

Cuando un usuario completa una tarea, queremos saber quién ha sido y cuando lo ha hecho. El usuario que completa una tarea no es, necesariamente, el que la tenía asignada.

La tarea "Pregunta 1" la terminó completando el usuario "def@example.com" el 14/10/2014.

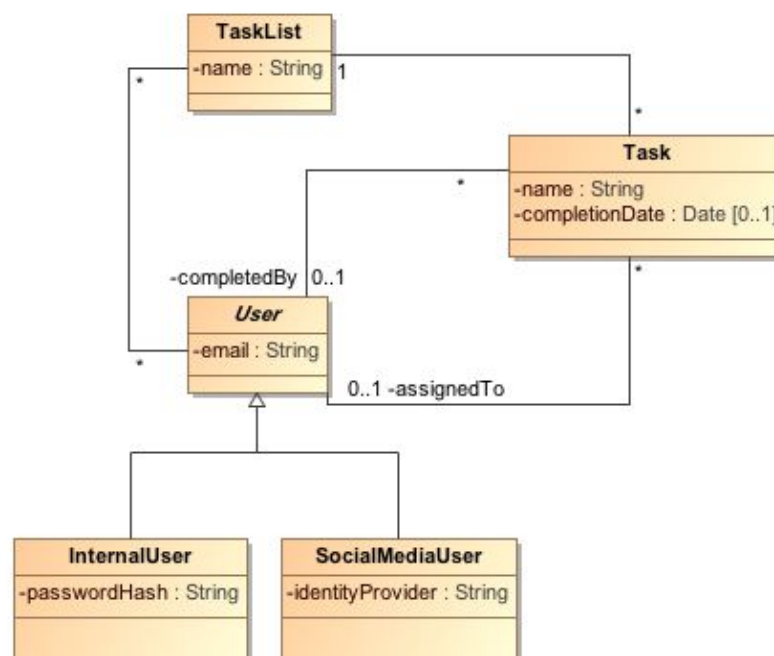
La tarea "Pregunta 2" la completó el mismo usuario el 16/10/2014.

La tarea "Pregunta 3" aún no está completada.

Finalmente, queremos que nuestros usuarios puedan acceder al sistema mediante las redes sociales, por este motivo, algunos usuarios (los llamados "usuarios internos") tendrán un password (de hecho, un hash) mientras que otras tendrán un proveedor de identidad ("facebook", "google", etc.).

El usuario "abc@example.com" tiene hash de password "dfaschkvcxiyuhdf" mientras que "def@example.com" ha utilizado "facebook" para identificarse en el sistema.

Diagrama de clases:



Restricciones de clave:

- Los usuarios se identifican por email
- Las listas de tareas se identifican por email

Otras restricciones de integridad:

- Una tarea sólo puede tener valor en el atributo "completionDate" si tiene un usuario asociado con el rol "completedBy".

Información derivada:

- No hay

Problema 5: Sistema de alertas

Queremos desarrollar un sistema de información que implemente un sistema de alertas sobre situaciones que se dan en una cadena de tiendas de alimentación.

En concreto, queremos saber, por cada alerta, cuando se generó y cuál es el motivo de la alerta, así como el lugar donde se produjo la incidencia. Este lugar puede ser una tienda (y entonces queremos saber la dirección) o un almacén (y en este caso queremos saber el nombre).

El 15/05/2014 a las 10:00:12 se generó una alerta con motivo "stock de patatas demasiado bajo" en la tienda de Avenida Roma, 4.

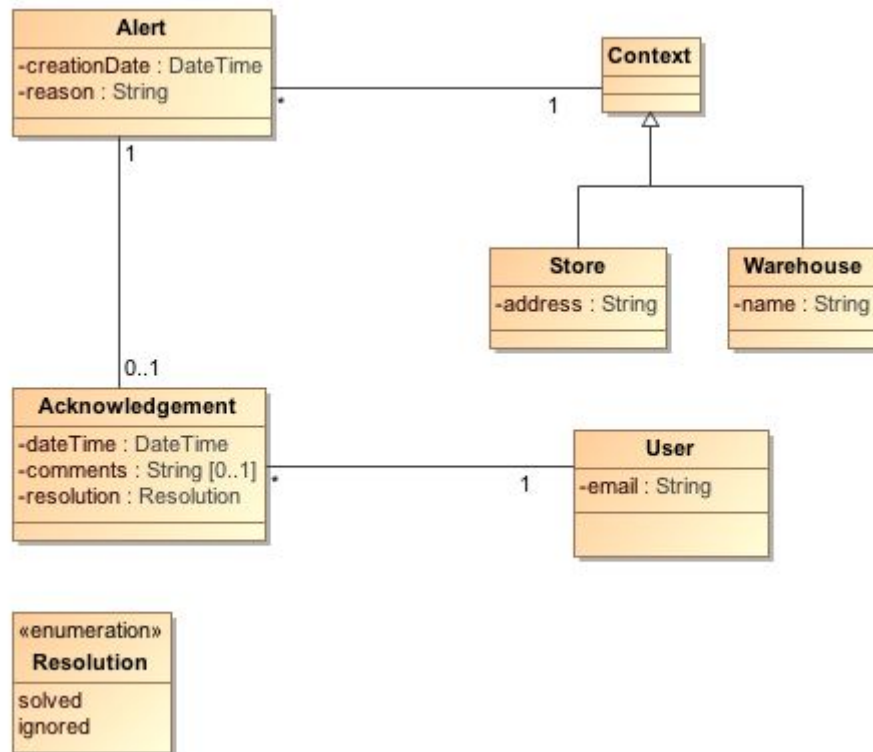
El 20/06/2014 a las 19:50:11 se generó una alerta con motivo "carne a punto de caducar" el almacén "Barcelona-1".

Por cada alerta queremos saber qué usuario (conoceremos el email) lo ha recibido (si es que lo ha recibido alguien), en qué momento (fecha y hora) lo hizo, como se resolvió (o bien se resuelve la incidencia o bien se ignora) así como si ha hecho algún comentario.

La alerta anterior sobre "stock de patatas demasiado bajo" fue recibida por el usuario julia@example.com el 15/05/2014 a las 10:05:11. La marcó como solucionada y añadió un comentario "Enviamos cargamento de patatas de emergencia".

La alerta sobre la carne a punto de caducar aún no la ha recibido nadie.

Diagrama de clases:



Restricciones de clave:

- Los usuarios se identifican por email

Otras restricciones de integridad:

- No hay

Información derivada:

- No hay

Problema 6: Niveles de un videojuego

Queremos desarrollar un sistema que nos permita saber cómo avanzan los usuarios de un conjunto de videojuegos por los diferentes niveles.

En concreto, todos los videojuegos de nuestro sistema tienen un nombre y una serie de niveles que tienen un nombre y están ordenados. Algunos de estos niveles tienen un límite de tiempo para superarlos.

El juego "Laundry Brush" tiene un primer nivel llamado "Level 1" que no tiene límite de tiempo y un nivel llamado "Level 2" que tiene un límite de tiempo de 30 segundos.

El juego "Happy Bird: Flying around the world" tiene un primer nivel llamado "Barcelona" (sin límite de tiempo), un segundo nivel llamado "London" (sin límite de tiempo) y un tercer nivel llamado "Moscow in a Hurry" con un límite de tiempo de 20 segundos.

Los usuarios (que identificamos por email) tienen un número de vidas que nos permiten saber si pueden jugar o no (sólo se puede jugar si el número de vidas es mayor que 0). Las vidas de un usuario no dependen del juego, sino que son del usuario, para utilizar en cualquier juego. Queremos saber, por cada usuario, en qué fecha superó cada nivel por primera vez. También nos interesa, para cada usuario, la puntuación global que es el número de niveles superados en todos los juegos.

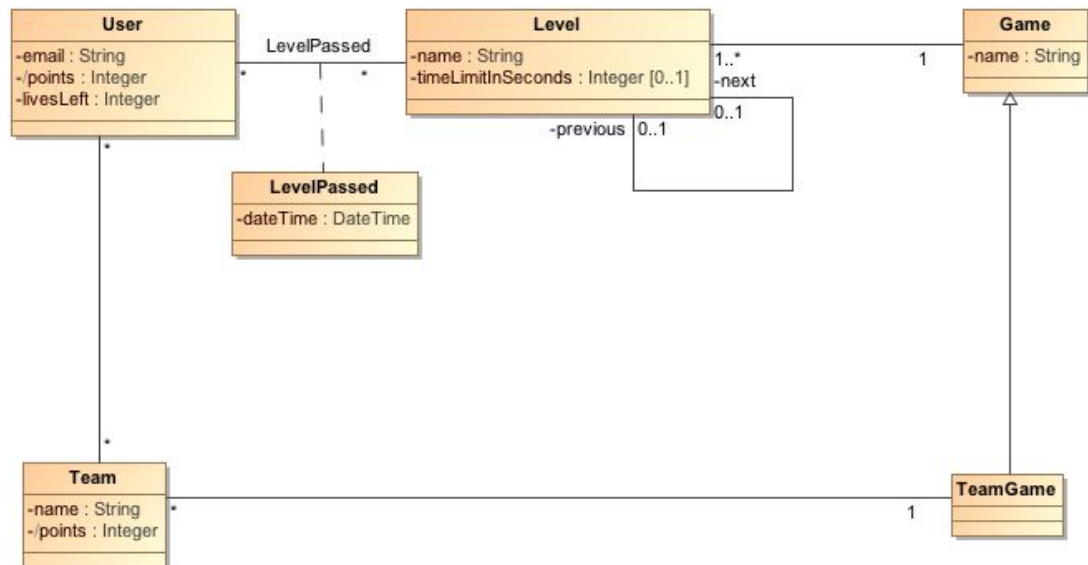
Al usuario "laia@example.com" le quedan 6 vidas, que puede utilizar en cualquiera de los juegos. Superó el nivel "Level 1" del "Laundry Brush" el 04/15/2014 a las 15:23:30 y el "Barcelona" del "Happy Bird" el 04/18/2014 a las 11:20:19. Por lo tanto, su puntuación global es 2.

Hay algunos juegos que se juegan en equipos. Un equipo sólo participará en un juego. Cada equipo tiene un nombre y una puntuación que es el número de niveles de aquel juego superados por los miembros del equipo.

El juego "Zombie Squad" es un juego en equipo con tres niveles: "ZS1", "ZS2" y "ZS3" donde sólo "ZS3" tiene límite de tiempo (30 segundos).

Los usuarios "joan@example.com" y "miquel@example.com" forman el equipo "Zombie killers". En "joan@example.com" superó el nivel "ZS1" el 06/15/2014 10:00:23 y el "ZS2" el 09/09/2014 09:00:14. En "miquel@example.com" sólo ha superado el "ZS1" el 07/10/2014. Por tanto, la puntuación global del equipo es de 3 puntos.

Diagrama de clases:



Restricciones de clave:

- Los usuarios se identifican por email
- No puede haber dos niveles con el mismo nombre en un mismo juego
- Los juegos se identifican por nombre
- No puede haber dos equipos con el mismo nombre en un mismo juego de equipo

Otras restricciones de integridad:

- El número de vidas de un usuario no puede ser negativo

Información derivada:

- Los puntos de un usuario es igual al número de niveles que tiene asociados
- Los puntos de un equipo es igual al número de niveles de su juego que tienen asociados los miembros del equipo

Problema 7: Compañía aérea

Nos han pedido desarrollar un Sistema de Información para una compañía aérea que gestione la información de sus vuelos, de los que queremos saber el número de vuelo, fecha y hora en la que sale, y la duración estimada del vuelo. Queremos determinar también los aeropuertos de los que salen y llegan los vuelos, sabiendo que el aeropuerto de destino será diferente del aeropuerto de salida. De los aeropuertos queremos saber la población y su nombre.

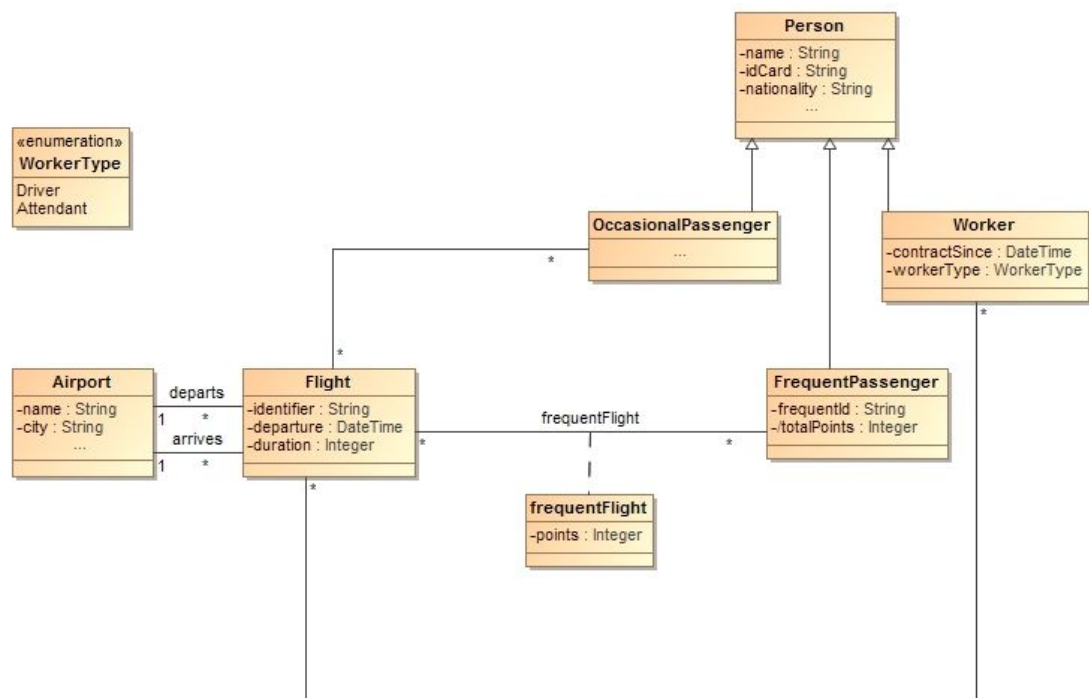
El 8 de julio de 2015 a las 18.00 saldrá el vuelo VLG1020 desde el aeropuerto del Prat de Barcelona hasta el aeropuerto Charles de Gaulle de París. La duración estimada del vuelo es de 60 minutos.

El vuelo estará ocupado por pasajeros de los que queremos saber el nombre, DNI y nacionalidad. Algunos de estos pasajeros serán socios del programa de puntos de la compañía. De estos socios, queremos saber el identificador del carné de puntos, el número de puntos acumulados, así como el número de puntos conseguidos en este vuelo.

Finalmente el vuelo estará formado también por 2 comandantes de vuelo que tripularán el avión y 4 auxiliares de vuelo. De estos trabajadores de la compañía queremos saber cuál es su antigüedad en la compañía, aparte del nombre, DNI y nacionalidad.

En el vuelo VLG1020 viajarán 120 personas, entre ellas estará Juan González, español con DNI 12345678-D. Juan González tiene el número 1234 del carné de puntos y tiene acumulados a día de hoy 250 puntos, 50 de los cuales han sido conseguidos en este vuelo. Este vuelo lo tripulará José Gómez, español con DNI 23456789-E que lleva en la compañía desde el 12 de enero de 2005, junto con otro comandante y 4 azafatas más.

Diagrama de clases:



Restricciones de clave:

- Airport : name
- Flight : identifier
- Person : idCard

Otras restricciones de integridad:

- El avión no puede despegar y aterrizar en el mismo aeropuerto.

Información derivada:

- La antigüedad del trabajador se obtiene de restar la fecha actual del día que empezó a trabajar.

Problema 8: Mundial de futbol

Nos han pedido desarrollar un Sistema de Información para el próximo mundial de fútbol que se realizará en Rusia en 2018. Cada uno de los partidos del mundial se hará en una de las sedes del Mundial, de las que sabemos el lugar, el nombre del estadio y la capacidad del estadio.

De cada partido del mundial queremos saber la sede donde se disputará, el día y la hora del partido, y qué dos equipos disputarán el partido así como cuáles serán sus árbitros. En un partido habrá 4 árbitros, repartidos en 1 árbitro principal, 2 jueces de línea y 1 árbitro suplente.

El 8 de julio de 2018 a las 18.00 se jugará el primer partido del grupo A en el estadio Zenit Arena de San Petersburgo. El partido enfrentará a España contra Argentina y será arbitrado por 1 árbitro brasileño como principal, 2 holandeses como jueces de línea y 1 inglés como árbitro suplente.

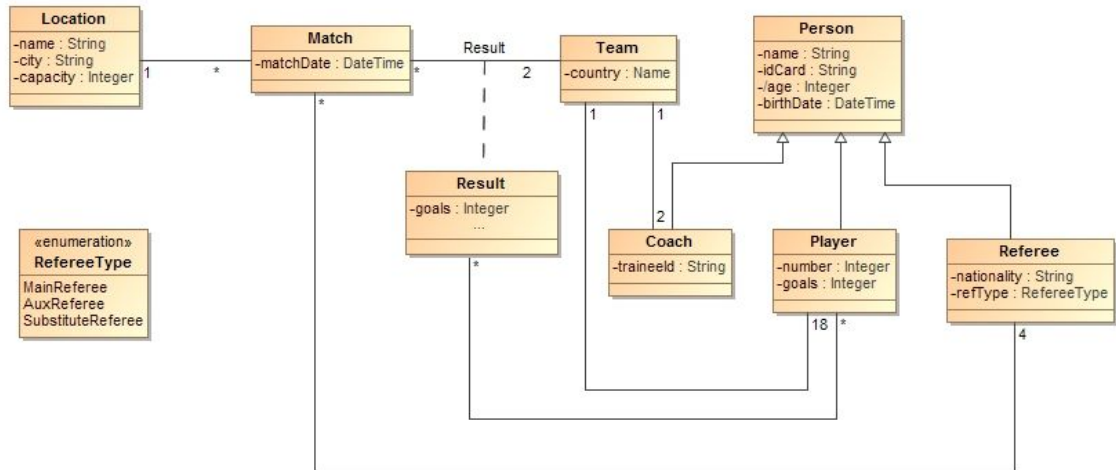
Cada equipo está formado por 18 jugadores y 2 entrenadores. De cada uno de los miembros de los equipos queremos saber su nombre y apellidos, su DNI y su edad. De los jugadores, además, queremos saber su dorsal, mientras que de los entrenadores queremos saber su identificador internacional de carné de entrenador.

Por otra parte, de los árbitros también queremos saber su nombre y apellidos, DNI y edad, su nacionalidad y si es árbitro principal, juez de línea o bien árbitro suplente.

Una vez jugado el partido queremos saber su resultado y cuáles fueron los goleadores.

El partido Inglaterra - Italia terminó con un resultado de 2 a 0 con goles marcados por Jones y Smith. Jones terminó marcando 3 goles en todo el torneo, pero a Smith, que marcó 9 goles, le dieron el trofeo de máximo goleador.

Diagrama de clases:



Restricciones de clave:

- Location : name
- Team : name
- Person : dni

Otras restricciones de integridad:

- Una selección no puede jugar 2 partidos a la vez.
- Un árbitro no puede arbitrar 2 partidos a la vez.
- Los goleadores de un resultado deben ser de los equipos que participan en el partido.

Información derivada:

- La edad de los jugadores se obtiene de la fecha actual menos su fecha de nacimiento

Problema 9: Hipotecas

La oficina de Hacienda del Estado español nos ha pedido desarrollar un Sistema de Información para gestionar las hipotecas concedidas por los diferentes bancos. Así, para cada banco queremos saber su nombre y su código de banco. Cada banco dispondrá de una serie de oficinas distribuidas a lo largo del país de las que queremos saber su código de oficina, y su localización.

Queremos gestionar el total de hipotecas concedidas por cada oficina bancaria, cuánto se pidió y cuánto queda por amortizar en el momento actual, sabiendo que una oficina sólo puede conceder una hipoteca a un cliente, pero un cliente puede pedir varias hipotecas en varias oficinas diferentes.

El señor Juan González contrató una hipoteca de 200.000 € en la oficina 0121 del Banco de Sabadell situada en el Paseo de Gracia número 1 de Barcelona. De esta hipoteca, el señor González ha amortizado ya € 80.000.

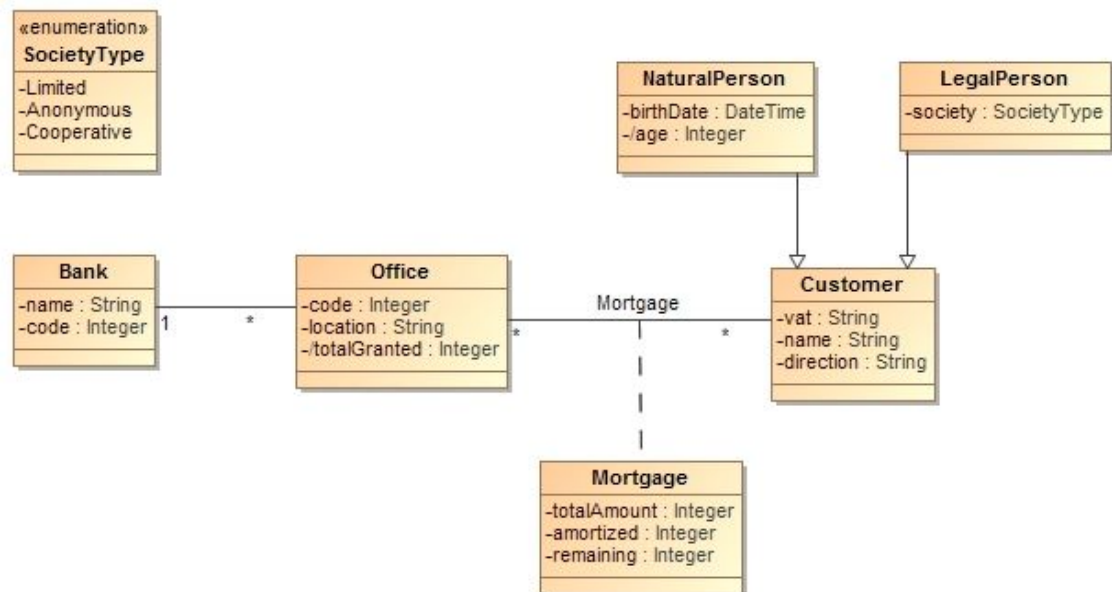
Las oficinas bancarias pueden conceder préstamos hipotecarios tanto a personas físicas como sociedades empresariales. En cualquier caso queremos saber el NIF, el nombre y su domiciliación. En el

caso de personas físicas queremos saber su edad, mientras que de las personas jurídicas queremos saber si es del tipo Sociedad Anónima, Sociedad Limitada o Sociedad Cooperativa.

Queremos saber, para cada oficina, cuál es el total acumulado de dinero prestado en este momento.

La oficina 0121 del Banco de Sabadell también concedió un préstamo de 60.000 € a la Sociedad Cooperativa librerías; y así, en estos momentos la oficina tiene un total de préstamos concedidos de 10.2 millones de euros.

Diagrama de classes:



Restricciones de clave:

- Bank : code
- Office : code
- Customer : vat

Otras restricciones de integridad:

- El total amortizado en una hipoteca no puede ser superior al total pedido.

Informació derivada:

- La cantidad pendiente de pagar en una hipoteca es igual al total pedido menos el total amortizado.
- La cantidad total prestada por una oficina es la suma de todas las cantidades pendientes de pagar por sus clientes.
- La edad de una persona es la fecha actual menos la fecha de nacimiento.

Problema 10: Películas en streaming

Un grupo de emprendedores nos ha pedido desarrollar un Sistema de Información para un sistema de visualización de películas en streaming. De cada película que se ofrezca queremos saber el título (que la identifica), la duración, el director, el género (acción, comedia, drama o fantasía) y, en caso de que sea de pago, el precio. Las películas pueden pertenecer a "sagas", de las que nos interesa el nombre (que la identifica) y cuál es la película que se ha visto más.

Actualmente se está ofreciendo la visualización de las seis películas de la saga Star Wars. La primera película (Episodio I - The Phantom Menace) es, desde hace mucho tiempo, la más vista por los usuarios.

También se está ofreciendo Memento, un Thriller del director Christopher Nolan que dura 113 minutos. A pesar de ser de pago (5 euros).

De los usuarios queremos saber una dirección de correo electrónico (que los identificará), el nombre, los apellidos en caso de que los proporcione y una contraseña. De cada usuario también queremos saber qué películas ha visualizado.

Mireia Pons (pons@gmail.com) ha visualizado 3 de las 6 películas que forman la saga Star Wars y Jordi Fermín (jf@gmail.com) ha visualizado Memento.

De cada visualización queremos saber qué usuario la ha hecho, qué película y la fecha. Los usuarios podrán puntuar las películas que han visto con una puntuación numérica entre el 0 y el 5. No tiene sentido que un mismo usuario dé más de una puntuación a una misma película aunque la haya visto más de una vez. De todas las películas queremos saber la puntuación media.

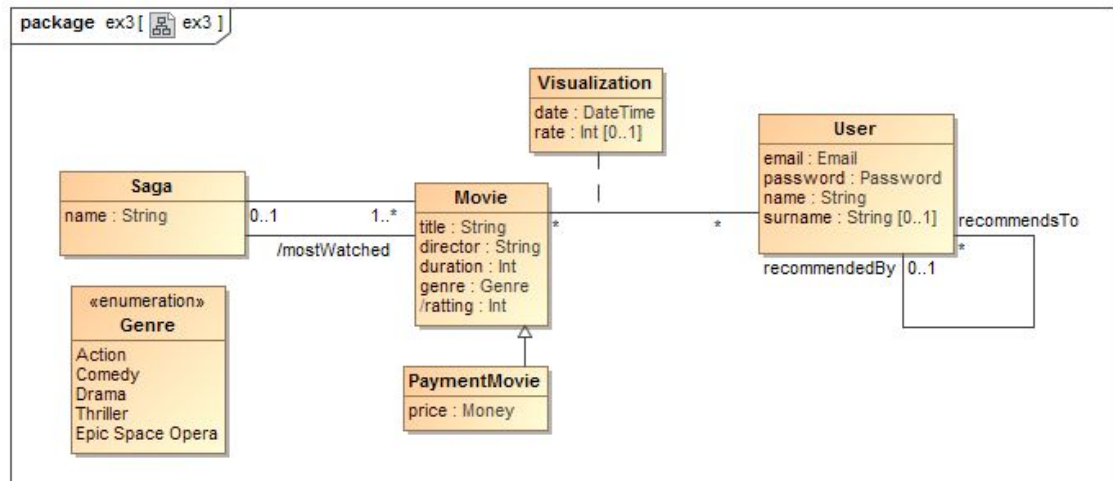
Mireia vio Episodio I - The Phantom Menace el 17 de Marzo y no la puntuó porque aún no sabía cómo funcionaba el sistema. Luego el 21 de Marzo vio Episode II - Attack of the Clones y la puntuó con un 4. Por último, el 25 de Marzo vio Episodio III - Revenge of the Sith y la puntuó con un 3.

Jordi vio Memento ayer y la puntuó con un 5. Memento es una de las películas mejor valoradas por los usuarios con una puntuación media de 4,9.

El sistema prevé un sistema de recomendaciones para premiar a los usuarios que traen más nuevos usuarios. Así, cuando un usuario se da de alta, puede indicar que ha sido recomendado por otro (y sólo por otro), de tal modo que de cada usuario sabremos qué usuarios fueron recomendados por él (es decir, qué usuarios se dieron de alta recomendados por él)..

Mireia se hizo usuaria del sistema después de ver un anuncio en su facultad. Como le gustó mucho el funcionamiento, recomendó a Jordi (cinéfilo empedernido) que se diera de alta y él lo hizo recomendado por Mireia.

Diagrama de clases:



Restricciones de clave:

- User: email
- Movie: title
- Saga: name

Otras restricciones de integridad:

- (implícita) La asociación recommendedBy no puede formar ciclos, de manera que un cliente no puede recomendarse a sí mismo ni recomendar a alguien que lo haya recomendado a él mismo directa o indirectamente.
- (explícita) La puntuación de una película debe ser un entero entre 0 y 5.
- (explícita) Los usuarios solo pueden puntuar películas que hayan visto.
- (explícita) Un mismo usuario no puede puntuar una película más de una vez.

Información derivada:

- /rating: La puntuación media de una película es la media de puntuaciones que han hecho los usuarios que la han visto.
- /mostWatched: La película más vista de una saga es la que ha tenido más visualizaciones.

Problema 11: Almacenamiento en la nube

Un grupo de emprendedores nos ha pedido desarrollar un Sistema de Información para permitir a sus usuarios almacenar en la nube información y compartirla. Los usuarios del servicio deben poder estructurar la información en carpetas y deben poder guardar archivos. De las carpetas sólo queremos saber el nombre (que las identifica), de los archivos nos interesa el nombre (que los identifica), la fecha de creación, el tipo (de momento binario o texto) y un enlace en caso de que el archivo haya sido compartido. Tanto para las carpetas como para los archivos también en queremos saber el propietario y un enlace y con quién ha sido compartido si han sido compartidos.

Noemí Raja ha decidido utilizar el servicio para compartir con Rosario Aymerich una carpeta con un vídeo de su despedida de soltera. La carpeta se llama "Desmadre!" Y el vídeo, "Las Vegas 2015", es un archivo binario creado el 27 de Abril de 2015. El sistema ha asignado el enlace <http://videocompartido> para que Roser lo pueda ver.

Ricard Jou y Jordi Guardia son profesores de un posgrado en la UOC y tienen un espacio donde guardan los enunciados y soluciones de las actividades del curso. Este espacio está organizado en carpetas por actividad y, dentro de cada una de ellas, hay documentos de texto con el correspondiente enunciado y solución de la actividad.

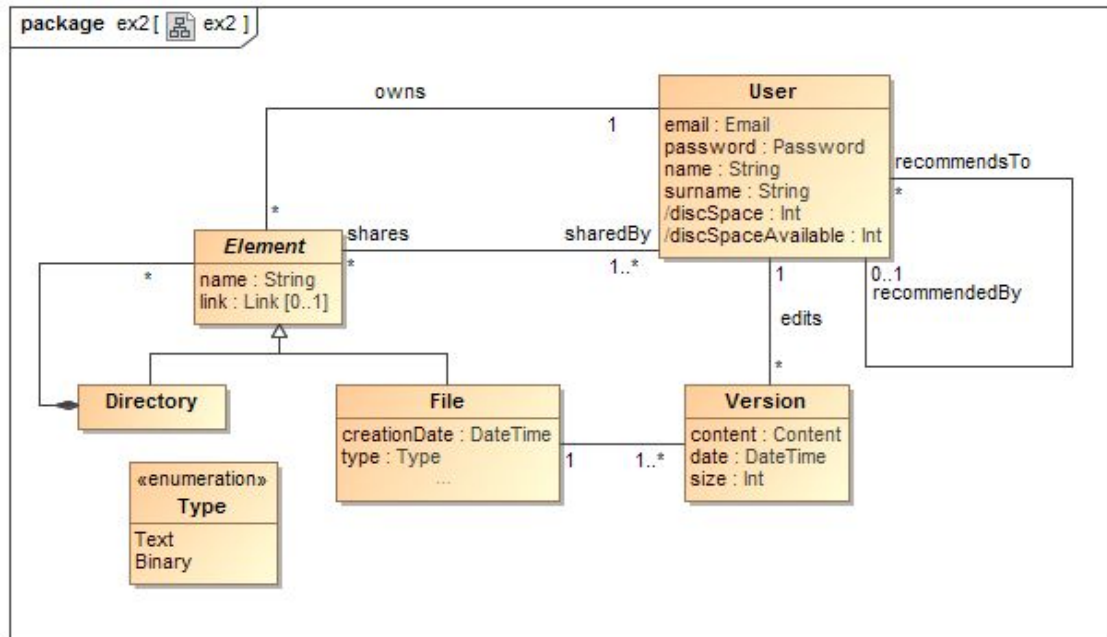
Una de las particularidades del servicio es que los diferentes usuarios que comparten un archivo lo pueden editar y que el sistema guarda todas las versiones que se vayan haciendo por si se quiere volver a una versión anterior. En un primer estadio del proyecto, para cada versión de un archivo queremos poder recuperar su contenido íntegro, saber la fecha de la versión, el tamaño y también qué usuario lo ha editado. Un archivo puede ser visualizado y editado por su propietario y por cualquiera de los usuarios con los que ha sido compartido.

Ricard y Jordi han estado trabajando en la solución de la primera práctica del postgrado. Ricard creó la primera versión el 21 de Octubre a las 17:15 y Jordi lo ha actualizado con su revisión el 23 de Octubre a las 21:00. El tamaño del fichero ha pasado de 3Kbytes a 7Kbytes con esta revisión.

Cuando un usuario se da de alta se le asignará un espacio de disco gratuito y, como interesa que el servicio crezca muy rápidamente, cuando un usuario se da de alta, puede indicar que ha sido recomendado por otro (y sólo por otro), de tal modo que de cada usuario sabremos qué usuarios fueron recomendados por él (es decir, qué usuarios se dieron de alta recomendados por él). Por cada recomendación se incrementará en un Giga este espacio. De los usuarios queremos saber el nombre, los apellidos, una dirección de correo electrónico (que los identificará), una contraseña, los "amigos", el espacio de disco máximo que tienen y el espacio de disco que les queda.

Noemí Raja (noemi@gmail.com) sólo ha conseguido un Giga de espacio de disco extra por invitar a unirse al servicio a Rosario. En cambio Ricard ya ha conseguido tres porque invitó a Jordi y a dos estudiantes en prácticas que colaborarán en el posgrado.

Diagrama de clases:



Notas:

- Para modelar los directorios y los ficheros se debe utilizar el patrón “Composite” con una clase abstracta Element.

Restricciones de clave:

- User: email
- Element: name
- Version: date

Otras restricciones de integridad:

- (implícita) La asociación recommendedBy no puede formar ciclos, de manera que un cliente no puede recomendarse a sí mismo, ni recomendar a alguien que ha recomendado él mismo directa o indirectamente.
- (explícita) Un fichero sólo puede ser visualizado y editado por su propietario y por cualquier usuario con el que ha estado compartido.
- (implícita) La fecha de creación de un fichero debe ser siempre menor o igual a la fecha de última modificación.
- (implícita) En un fichero con una única versión, la fecha de creación del fichero y la fecha de modificación de la versión coinciden.

Información derivada:

- /discSpace: El espacio de disco de un usuario será el espacio gratuito inicial más un Giga por cada “amigo” que se haya dado de alta.
- /discSpaceAvailable: Se calcula restando del espacio máximo el tamaño de todas las versiones de ficheros suyos o compartidos.

Problema 12: Alquiler de vehículos

Un grupo de emprendedores nos ha pedido desarrollar un Sistema de Información para gestionar un servicio de alquiler de vehículos por internet. El servicio tiene que trabajar con diferentes compañías de alquiler. De cada compañía queremos saber su nombre y el número total de vehículos de su flota. Cada compañía tendrá una serie de oficinas distribuidas a lo largo del país de las que queremos saber su localización.

TopCars sólo tiene dos oficinas, ambas en la ciudad de Barcelona, una en el aeropuerto y la otra en la terminal marítima y sólo dispone de 50 vehículos para alquilar. GreenCars, en cambio, tiene más de 250 oficinas distribuidas por todo el territorio y más de 30.000 vehículos para alquilar.

Cada oficina dispone de una flota de vehículos de diferentes tipos. De los vehículos queremos conocer su tipo (coche, motocicleta o furgoneta), el color, la matrícula, la marca, el modelo, la capacidad y su precio. Hay algunos vehículos que están en oferta y hay que indicar un porcentaje de descuento en el precio.

La oficina del aeropuerto de Barcelona de Top Cars tiene una furgoneta blanca, con matrícula "3567DKY", de la marca "Nissan", modelo "Vanette", con capacidad para seis personas. Esta furgoneta se puede alquilar con un 5% de descuento del precio total.

La oficina de la plaza Castilla de GreenCars tiene una motocicleta negra, con matrícula "7697BBC", de la marca "BMW", modelo "F650GS", con capacidad para dos personas. Esta motocicleta no tiene descuento.

La oficina del aeropuerto de Tenerife de GreenCars tiene un turismo rojo, con matrícula "B3456XT", de la marca "Volvo", modelo "V50", con capacidad para cinco personas. Este coche se puede alquilar con un 10% de descuento del precio total.

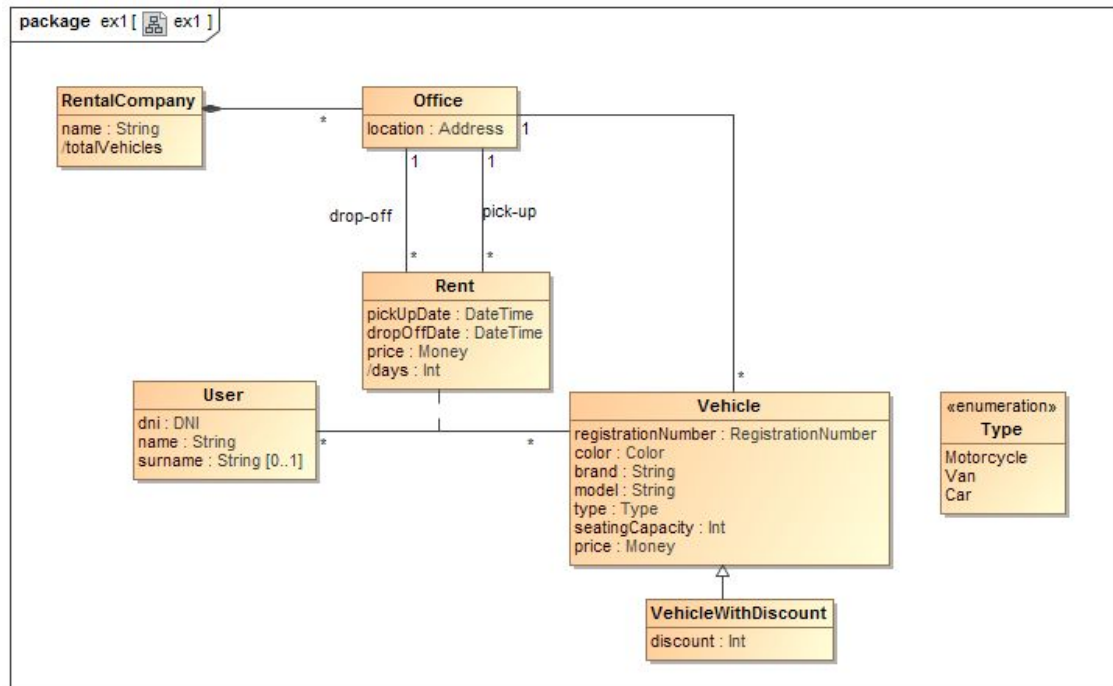
De los usuarios que utilicen el servicio de alquiler queremos conocer su DNI, su nombre y, si el usuario los ha introducido durante el proceso de registro, sus apellidos.

De los alquileres nos interesa saber la compañía, la fecha de recogida, la oficina de recogida, la fecha de devolución, la oficina de devolución, el número de días que se alquila el vehículo y el precio total a pagar (a este precio se le aplicará automáticamente el descuento en caso de que aplique).

Juan (34567891L) alquiló la furgoneta con matrícula "3567DKY" en el aeropuerto el 23 de Febrero y la volvió a la terminal marítima el 29 de Febrero. Juan pagó 120 euros por el alquiler.

Luisa Messeguer Pons (77789523K) alquiló un Volvo V50 rojo en el aeropuerto de Tenerife el 7 de Marzo y el regresó al cabo de 7 días en el mismo aeropuerto. El alquiler total le costó 200 euros.

Diagrama de clases:



Notas:

- Se puede modelar los diferentes tipos de Vehículo con una relación de herencia pero no es necesario ya que el enunciado no expone ni atributos ni asociaciones particulares para los diferentes tipos.
- Se puede modelar la marca y el modelo de los vehículos como clases con una asociación entre ellas.
- Si consideráis que un mismo vehículo puede estar a veces con descuento y a veces sin, entonces su relación es una herencia dinámica y debería modelarse con la modelización del rol de objeto y personas que tiene el módulo 4 de los materiales.

Restricciones de clave:

- Vehicle: registrationNumber
- RentalCompany: name
- Office: location
- User: dni

Otras restricciones de integridad:

- (implícita) La fecha de devolución debe ser posterior a la fecha de recogida.
- (implícita) La oficina de recogida y la de devolución deben pertenecer a la misma compañía de alquiler

Información derivada:

- /days: El número de días de alquiler son los días que hay entre la fecha de recogida y la fecha de devolución
- /totalVehicles: El número total de vehículos de una compañía es la suma de los vehículos que tiene cada oficina de la compañía.

Problema 13: Museo de zoología

Nos han pedido modelar un sistema de información que gestiona información sobre especímenes de una colección de un museo de zoología contemporánea. Esta colección contiene especímenes (los elementos de la colección, típicamente un animal disecado o un esqueleto) que clasificaremos en especies (el tipo de animal como "gato", "perro" o "ballena").

De cada espécimen tenemos un identificador interno y el año en que se encontró, además de la especie a la cual pertenece. Como se trata de un museo de zoología contemporánea, el año debe ser igual o posterior a 1970. De las especies, además del nombre, sabemos las áreas donde se puede encontrar este animal. Estas áreas geográficas, además de tener un nombre, sabemos que forman una jerarquía donde un área puede tener (o no) un área "madre" y varias áreas "hijas". Finalmente, también queremos saber si una especie, a un área concreta, es muy común o no.

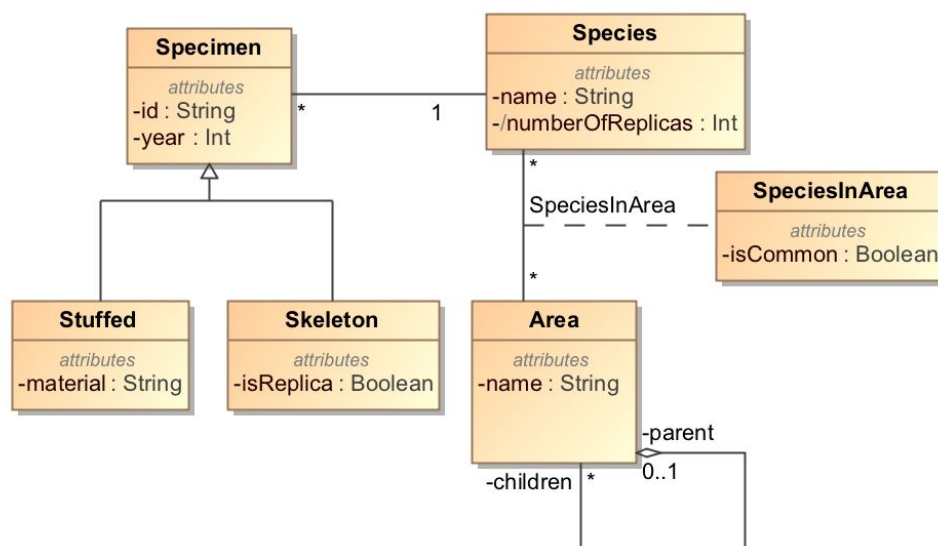
Por ejemplo, tenemos un espécimen de la especie "Escarabajo picudo rojo" encontrado en 1998 con identificador "E9921". Esta especie sabemos que es muy común en el área "Granada" (que es hija del área "España") pero poco común en el área "Badajoz" (que también es hija del área "España").

Los especímenes de nuestra colección pueden ser, tal y como hemos dicho anteriormente, de dos tipos: los animales disecados (de los que queremos saber qué material se ha utilizado para rellenar la piel del animal) y los esqueletos (de los que queremos saber si es una réplica o un esqueleto de verdad).

Nuestra colección tiene un esqueleto de "Mamut" (en realidad, una réplica) con identificador "E3342" que se hizo en 1982. Los mamuts son muy comunes en el área de "Siberia".

Finalmente, también queremos saber, para una especie en concreto, el número de réplicas que tenemos en la colección (es decir, el número de especímenes de esa especie, que son de tipo esqueleto y son réplicas).

Diagrama de clases:



Restricciones de clave:

- Specimen: id
- Species: name
- Area: name

Restricciones de integridad:

- El año de un espécimen tiene que ser igual o posterior a 1970 (Specimen::year >= 1970).

Informació derivada:

- Species::numberOfReplicas = número de especímenes de aquella especie, que son de tipo esqueleto i son replicas

Problema 14: Instrumentos y piezas musicales

Nos han pedido modelar un sistema de información que gestiona información sobre instrumentos musicales y piezas de música compuestas por estos instrumentos.

De cada pieza conocemos su nombre, el autor y el año en que se compuso. También sabemos qué instrumentos son necesarios para tocar esa pieza y, por cada instrumento, si tiene un solo o no a la pieza. De los instrumentos sabemos el nombre.

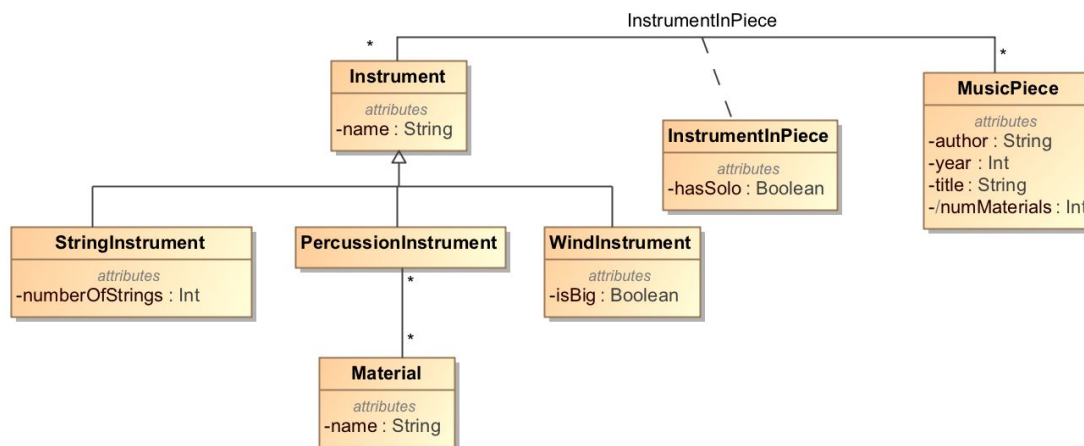
Por ejemplo, tenemos la pieza "Baile de la patata" que necesita un tambor y una guitarra para tocarse y que tiene un solo de guitarra (pero no de tambor).

En cuanto a los instrumentos, los hay de cuerda (y entonces queremos saber cuántas cuerdas tiene, aunque como mínimo debe tener una para ser considerado un instrumento de cuerda), de percusión (y entonces queremos saber de qué materiales está hecho el instrumento) y de viento (en este caso sólo queremos saber si es grande o no).

Por ejemplo, un ukelele es un instrumento de cuerda que tiene cuatro cuerdas; un triángulo es un instrumento de percusión que es de metal y un trombón es un instrumento de viento que es grande.

Finalmente, también queremos saber, para una pieza de música, cuántos materiales diferentes se utilizan (es decir, cuántos materiales diferentes se utilizan entre todos los instrumentos de percusión que son necesarios para tocar la pieza).

Diagrama de clases:



Restricciones de clave:

- Instrument: name
- MusicPiece: title

Restricciones de integridad:

- Un instrumento de cuerda tiene que tener como mínimo una cuerda (`StringInstrument::numberOfStrings >= 1`)

Información derivada:

- `MusicPiece::numMaterials` = Número de valores diferentes del conjunto de materiales usados en los instrumentos de percusión asociados con la pieza

Problema 15: Variedades de cerveza

Nos han pedido modelar un sistema de información que gestiona citas de variedades de cerveza.

El sistema debe permitir dar de alta varias variedades de cerveza. De cada variedad, queremos saber el nombre y los documentos donde se ha citado. Para cada documento donde se ha citado la variedad, queremos saber cuántas veces se cita la variedad de cerveza en el documento (obviamente, como mínimo deberá ser citada una vez).

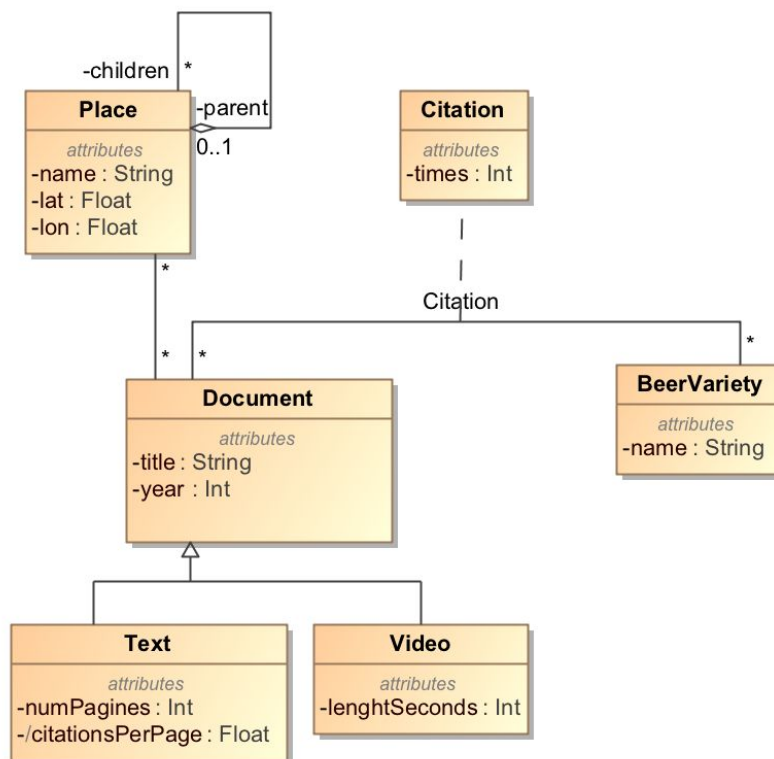
Por ejemplo, tenemos un documento con título "Estudio sobre las variedades de cerveza consumidas en el Barcelonés" que menciona la variedad "Lager" hasta 24 veces mientras que el mismo documento menciona la variedad "Pilsener" 3 veces.

Los documentos pueden ser de texto (y en este caso queremos saber el número de páginas del documento) o de video (y entonces nos interesa saber la duración). Tanto en un caso como en el otro, queremos saber, además del título, en qué año se publicó el documento así como a qué lugares geográficos hace referencia en el documento. Los lugares pueden formar una jerarquía donde un lugar puede tener un padre y varios hijos. De cada lugar queremos saber el nombre así como una latitud y

longitud de referencia por si queremos representarlo a un mapa. Para los documentos de texto, también queremos saber la ratio entre citaciones y páginas (suma de las veces en que se cita alguna variedad de cerveza dividido por el número de páginas del documento).

Volviendo al caso anterior, el documento "Estudio sobre las variedades de cerveza consumidas en el Barcelonés" se publicó en 2005 y es un texto de 28 páginas. El documento hace referencia a Barcelona (longitud N 41°23'19" y latitud E 2°9'32") y a Badalona (longitud N 41°27'0" y latitud E 2°14'50"), ambos lugares tienen por padre Cataluña (longitud N 41°49'13" y latitud E 1°52'3"). La ratio de citaciones por página es (24+3)/27.

Diagrama de clases:



Notas:

- Se podría haber modelado lat y lon como un tipo de dato ("dataType") con atributos hemisferio, grados, minutos y segundos.

Restricciones de clave:

- Variety: name
- Document: title
- Place: name

Otras restricciones de integridad:

- Citation::times té valor mínim 1.

Información derivada:

- Text::citationsPerPage = suma de les vegades en què es cita alguna varietat de cervesa dividit pel nombre de pàgines del document

Problema 16: Cadena de restaurantes

Queremos desarrollar el software para la gestión de una cadena de restaurantes. De cada restaurante queremos saber su nombre, su identificador fiscal (CIF), la dirección donde está ubicado y su teléfono. Cada restaurante está gestionado por un responsable del cual queremos saber su nombre completo, su dni, su teléfono y su dirección de correo electrónico.

El restaurante La Colmena, con CIF 12.345.678-A, está situado en la calle Passeig de Sant Joan, 12 de Barcelona, y tiene el teléfono 93 123 45 67. Este restaurante está gestionado por Roser Badia, con DNI 34567890-B, teléfono 610987654 y dirección de correo electrónico roser.badia@lacolmena.

Además de los responsables, cada restaurante tiene una serie de trabajadores los cuales se clasifican en la categoría de cocineros o de camareros. Cada trabajador desarrolla un único rol (responsable, cocinero o camarero) dentro del restaurante. De los trabajadores queremos saber su nombre completo, dni, teléfono y su dirección de correo electrónico.

El restaurante La Colmena dispone de tres cocineros y cinco camareros, de los que tenemos todos sus datos.

Cada restaurante dispone de una lista de productos disponibles los cuales se pueden clasificar en primer plato, segundo plato, postre o bebida. De los productos queremos saber su nombre, el precio y el tipo en el que están clasificados.

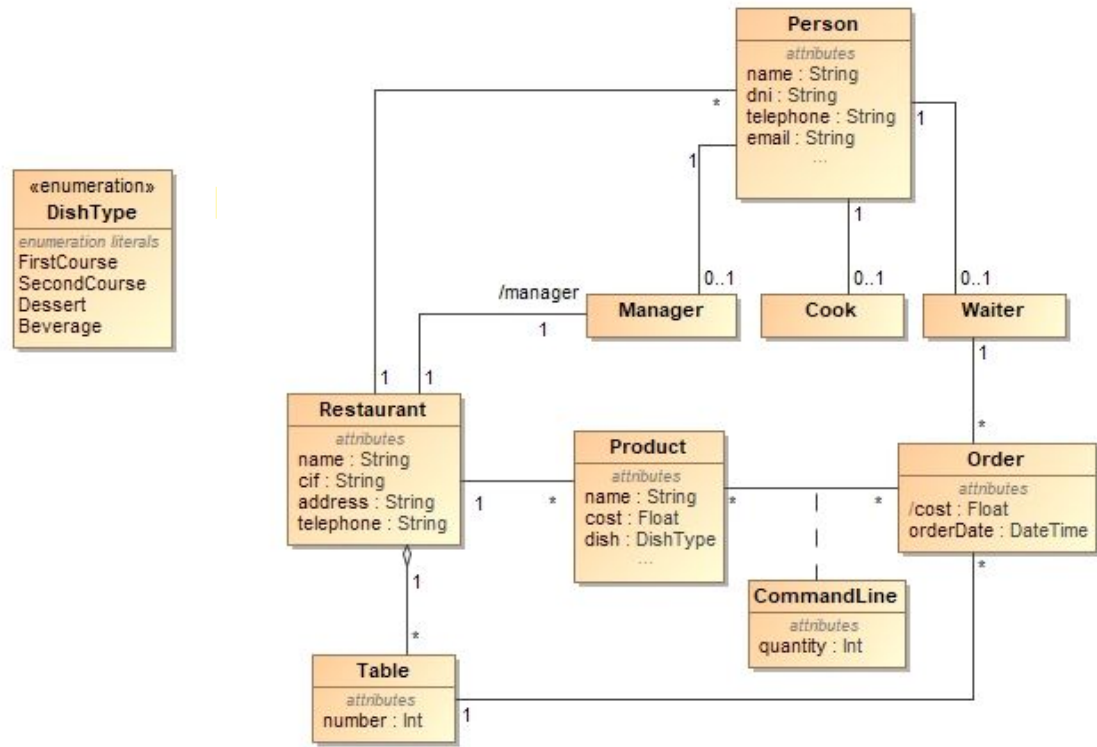
El restaurante dispone de una carta que contiene un listado de primeros: ensalada de la casa, crema de verduras y lasaña vegetal, todos ellos por 10€. El restaurante también dispone de un listado de segundos platos, un listado de postres y un listado de bebidas.

Cada restaurante tiene identificadas por números las diferentes mesas de las que dispone. Cuando una persona o personas se sientan en una mesa, piden un pedido. Para cada pedido, queremos saber qué camarero se encarga de procesarlo y en qué fecha y hora se ha pedido. Los pedidos están formados por los productos que ofrece el restaurante.

El día 3 de diciembre a las 14:30, el camarero Joel Riera se encarga de procesar un nuevo pedido para la mesa número 5, el cual se compone de 2 ensaladas y una crema de primero; unos canelones, un pollo y un lenguado de segundo; y 2 tiramisú y una pieza de fruta de postre. Además, la mesa ha pedido un agua grande para beber.

Cada pedido tiene un precio total, el cual está formado por la suma del precio de todos los productos que la mesa ha pedido.

Diagrama de clases:



Comentarios:

- Se puede cambiar la asociación de restaurante en persona, por 2 asociaciones de restaurante con cocinero y camarero respectivamente con multiplicidades 1 ..*.
- El precio, tanto del producto como del pedido, se puede representar utilizando el tipo Moneda.
- Sería posible modelar los roles de los trabajadores como una herencia de persona. En este caso, habría la restricción de que una persona con un rol no podría pasar a tener otro rol. Es decir, un camarero no podría pasar a cocinero.

Restricciones de clave:

- Person: dni
- Restaurante: cif
- Manager: person
- Waiter: person
- Cook: person
- Table: number + restaurante
- Product: name + restaurante
- Order: table + orderDate

Restricciones de integridad:

- No hay

Información derivada:

- El precio total del Pedido se calcula como la suma de los precios de los productos solicitados.

Problema 17: Venta de juguetes

Queremos desarrollar un Sistema de Información para una plataforma de venta de juguetes por internet.

La plataforma dispone de un catálogo de juguetes de los que conocemos el nombre, una descripción, la cantidad disponible y el precio de cada unidad. Para facilitar las búsquedas de los clientes, los juguetes pueden tener asociadas múltiples etiquetas. De cada etiqueta conocemos el nombre que la identifica, una descripción y el número de juguetes disponibles que tienen asociada la etiqueta.

Por ejemplo, el catálogo tiene 10 unidades de "Parchís" que se venden a 10 € la unidad. El juego del "Parchís" lo podemos encontrar tanto si buscamos la etiqueta "Juegos de mesa" como si buscamos "Juguetes educativos".

De los clientes queremos saber un número de documento, el tipo (ya sea un NIF, un CIF o un NIE) y el nombre. En este primer estadio del proyecto no permitiremos hacer pedidos a clientes que no se hayan dado de alta, por lo tanto, todos los clientes tendrán un nombre de usuario y una contraseña que se les asignará en el momento que se registren al sistema.

Todos los clientes deben poder consultar el estado de sus pedidos. De cada pedido queremos saber el cliente que lo ha hecho, el estado (en proceso, entregado o devuelto), los juguetes comprados, el precio total, la dirección de envío, los datos de pago (que se detallarán más adelante) y la fecha prevista de entrega. Los pedidos se identifican con un número entero mayor que 0 que generará y asignará automáticamente el sistema.

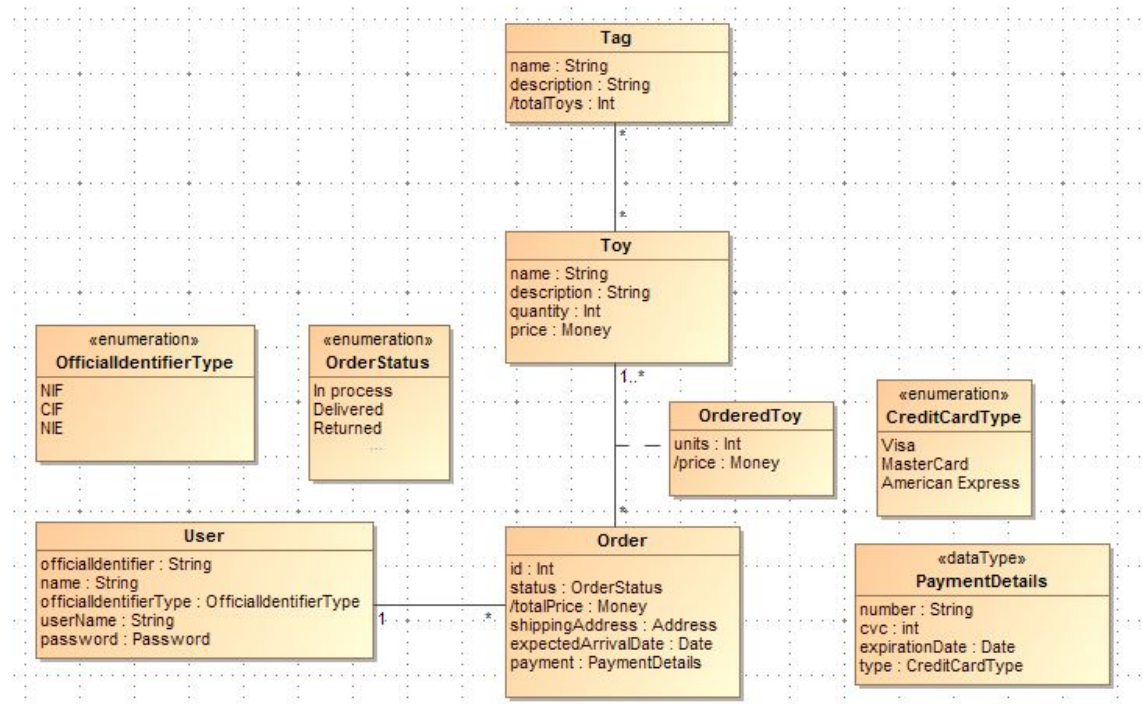
Cada pedido puede contener varios juguetes y también es necesario que se contemple la posibilidad de introducir más de un juguete del mismo tipo en el pedido. Para cada juguete del pedido sabremos el nombre, la cantidad demandada y el precio total.

En cuanto a los datos de pago sólo se va a permitir el pago con tarjeta de crédito y habrá que conocer el número de tarjeta que lo identificará, la fecha de caducidad, el tipo (Visa, MasterCard o American Express) y el código cvc.

Por ejemplo, Jordi Ciruela con DNI 393541987T ha hecho un pedido para que se lo llevamos a la calle Tilos, número 29, 08256, Barcelona, y ha hecho el pago con una tarjeta Visa. Jordi sabe que su pedido está actualmente en proceso de ser servido y que, previsiblemente, le llegará el pedido el próximo miércoles. Jordi, en su pedido, pidió 2 juegos de "la Oca" con un precio total de 15 euros y 1 "Parchís" con un precio total de 10 euros, por lo tanto, todo el pedido le ha costado 25 euros.

Nota: Podéis considerar que se ha definido un tipo de datos Address para modelar los datos típicos de las direcciones sin preocuparos de cuáles son exactamente. También se ha definido un tipo de datos Password para reflejar que suele haber restricciones sobre qué contraseñas son válidas y que, por tanto, no suele ser un String cualquiera, y un tipo de datos Money para modelar los importes.

Diagrama de clases:



Restricciones de clave:

- User: officialIdentifier
- Tag: name
- Toy: name
- Order: id
- PaymentDetails: number

Otras restricciones de integridad:

- (explícita) El identificador del pedido debe estar generado por el sistema y debe ser un valor numérico mayor que 0.

Información derivada:

- El precio total del pedido es la suma de los precios totales del conjunto de líneas de pedido que lo forman.
- El precio total de una línea de pedido es el producto entre las unidades de producto solicitado y el precio del producto. Tened en cuenta que esto es una simplificación de la realidad ya que si se cambia el precio de un elemento NO se desea que se cambie el precio de los pedidos ya hechos.
- El número total de juguetes de una etiqueta es la suma de todos los juguetes que tienen esa etiqueta asociada.

Problema 18: Gestión de gastos

Queremos desarrollar un sistema para gestionar gastos entre los integrantes de los grupos de amigos. Los integrantes del grupo se podrán registrar en el sistema con una dirección de correo electrónico que los identificará, una contraseña y un nombre. De los grupos queremos tener sólo un nombre que los identifique y una descripción. También queremos saber cuáles son los integrantes de un grupo para saber quién forma parte o ha formado parte en algún momento; de cada uno de ellos queremos saber la fecha en que entró a formar parte del grupo y, si ya no forma parte, la fecha en que se dió de baja. Ten en cuenta que un usuario sólo puede pertenecer a un grupo en un momento determinado.

Los "Rompesueños" son un grupo de amigos "Acostumbrados a disfrutar de los pequeños placeres de la vida" que está formado actualmente por 12 personas. Luisa Armin, que podemos identificar como `larmin@gmail.com`, forma parte del grupo desde Enero de 2010. En 2010 también se integró Juan Pérez pero en agosto de 2012 se marchó y ahora pertenece al grupo los "Madrugadores".

El sistema permitirá a los integrantes del grupo dar de alta gastos. De cada gasto queremos saber qué usuario lo pagó, en qué fecha, una descripción, un importe y como se ha hecho el pago, que puede ser efectivo, tarjeta de crédito o transferencia bancaria. Los gastos se identifican con un número entero mayor que 0 que generará y asignará automáticamente el sistema. También habrá que saber con qué grupo está relacionado cada gasto, que tiene que ser, por fuerza, el grupo del que sea miembro la persona que lo introduce.

El día 09/11/2014, Luisa hizo un gasto de 600,64€ al que puso descripción "Alquiler de la mesa de sonido para la fiesta del Marco" y que pagó por transferencia bancaria. Este gasto es del grupo de los "Rompesueños".

También queremos que los gastos se puedan clasificar en varias categorías, de tal manera que a cada gasto se le puedan asignar cualquier número de categorías. Las categorías tienen un nombre que las identifica y deben formar una jerarquía, de tal manera que una categoría puede ser subcategoría de una categoría madre.

El gasto antes mencionado estará en las categorías de nombre "Cumpleaños" y "Alquileres de equipos". La categoría "Cumpleaños" es subcategoría de "Celebraciones" que tiene otras subcategorías como "Fiesta mayor" o "Semana Santa".

También habrá que saber el importe total mensual de cada categoría de gastos que se puede calcular como la media mensual de la suma de los importes de todos los gastos de esa categoría.

En ambas categorías hay otros gastos, y si lo calculásemos, la categoría "Cumpleaños" tiene un gasto medio mensual de 55,68€, y la categoría "Alquiler de equipos" tiene una media mensual de 6,29€.

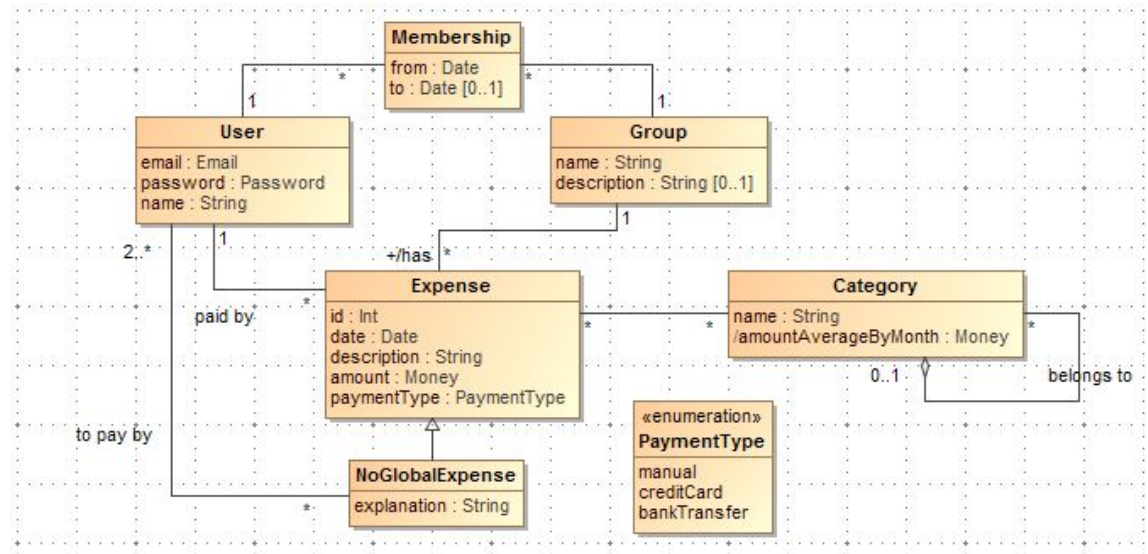
Algunos gastos no se repartirán entre todos los integrantes del grupo, sino sólo entre algunos. De estos gastos queremos saber, además, quién (al menos 2 personas) los tiene que pagar y una explicación de por qué sólo los tienen que pagar ellos.

Otro gasto de los "Rompesueños" es el que pagó Luisa el día 22/12/2015 con tarjeta de crédito, al que dio la descripción "Salida a Port de la Selva" y que fue de 145,58€. Este gasto sólo lo

tienen que pagar Luisa, Juan y Pedro y tiene por explicación "sólo lo deben pagar los que vamos a hacer la excursión en kayak en Port de la Selva".

Nota: Se puede considerar que se ha definido un tipo de datos para las contraseñas para reflejar que suele haber restricciones sobre qué contraseñas son válidas y que, por tanto, no suele ser un String cualquiera. Considera el mismo para la dirección de correo y por los importes.

Diagrama de clases:



Restricciones de clave:

- User: email
- Group: name
- Expense: id
- Category: name

Otras restricciones de integridad

- (explícita) El identificador del gasto debe estar generado por el sistema y debe ser un valor numérico mayor que 0.
- (implícita) Un usuario no puede pertenecer a dos grupos a la vez.
- (implícita) Un gasto que no tengan que pagar todos los miembros del grupo sólo puede tener asociados usuarios que pertenezcan al grupo en la fecha del gasto.

Información derivada:

- Un gasto está asociado al grupo donde pertenecía al usuario que lo pagó.
- El importe total mensual de una categoría es la media mensual de los importes de los gastos de esa categoría.

Problema 19: Trayectos compartidos

Queremos desarrollar un Sistema de Información que permita a sus usuarios compartir trayectos en coche. La aplicación pondrá en contacto pasajeros y conductores y las principales funcionalidades que el sistema les ofrecerá son la publicación de un trayecto con plazas libres, la búsqueda de trayectos con plazas libres, la comunicación entre conductor y pasajero y la reserva de plazas.

Para publicar un trayecto un usuario deberá estar registrado en el sistema como conductor. La información que debe proporcionar un conductor para registrarse incluye su NIF, que lo identificará, el nombre y apellidos, teléfono, una contraseña y su dirección de correo electrónico. Cada conductor tendrá uno o más coches asociados que también deberá dar de alta en el sistema. Para cada coche guardaremos la matrícula, la marca, el modelo y el color. No hace falta modelar ningún tipo de relación entre la marca y el modelo de cada coche.

Por ejemplo, Juan Romeo, con DNI 7776543T y teléfono 665 789 001, se ha registrado al sistema con el correo electrónico jromeo@gmail.com (la contraseña no os la diremos ;-)) y tiene un Volvo V50 rojo con matrícula 4567DKV y un Seat León negro con matrícula 4678HJK.

Los trayectos que se publiquen deben especificar la fecha y hora de salida, la fecha y hora de llegada, el lugar de salida, el lugar de llegada, el coche con el que se hará el trayecto, el número de plazas ofertadas, el número de plazas disponibles, el precio de cada una de las plazas y una descripción opcional. Los trayectos se identifican con un número entero mayor que 0 que generará y asignará automáticamente el sistema.

Juan tiene, actualmente, un trayecto publicado de Barcelona a Madrid que sale el jueves 21 de noviembre a las 9:00 de la mañana y tiene previsto llegar a Madrid el mismo jueves a las dos de la tarde. Este trayecto lo hará con el Volvo y ofrece 3 plazas a 20 euros cada plaza.

Para apuntarse a un trayecto, sólo será necesario que los usuarios se registren al sistema y se apunten al trayecto. La información que debe proporcionar un usuario para registrarse como pasajero incluye su NIF, nombre y apellidos, teléfono, una contraseña y su dirección de correo electrónico. Podéis considerar que un conductor no se convierte nunca un pasajero y que un pasajero no se convierte nunca un conductor.

Es importante que los pasajeros puedan preguntar todo lo que quieran a los conductores antes de hacer una reserva, por eso hay que ofrecer un servicio de mensajes públicos asociados al trayecto donde se podrá hacer preguntas al conductor. Por ejemplo, te puede interesar el trayecto pero sólo si el conductor te puede acercar a la estación de autobuses en llegar al destino. Este sistema de mensajes será muy sencillo y de los mensajes queremos saber el identificador (una cadena generada automáticamente por el sistema), el remitente, el texto y si responde a otro mensaje.

Luís Molina con DNI 678543K y teléfono 669 834 534, se ha registrado en el sistema de correo electrónico lmolina@gmail.com (la palabra de paso tampoco os la diremos ;-)) y se ha apuntado al trayecto Barcelona-Madrid que ha publicado en Juan y le interesa saber si puede pasar a recoger por la Estación de Sants. Juan le ha contestado que sí que la puede pasar a recoger.

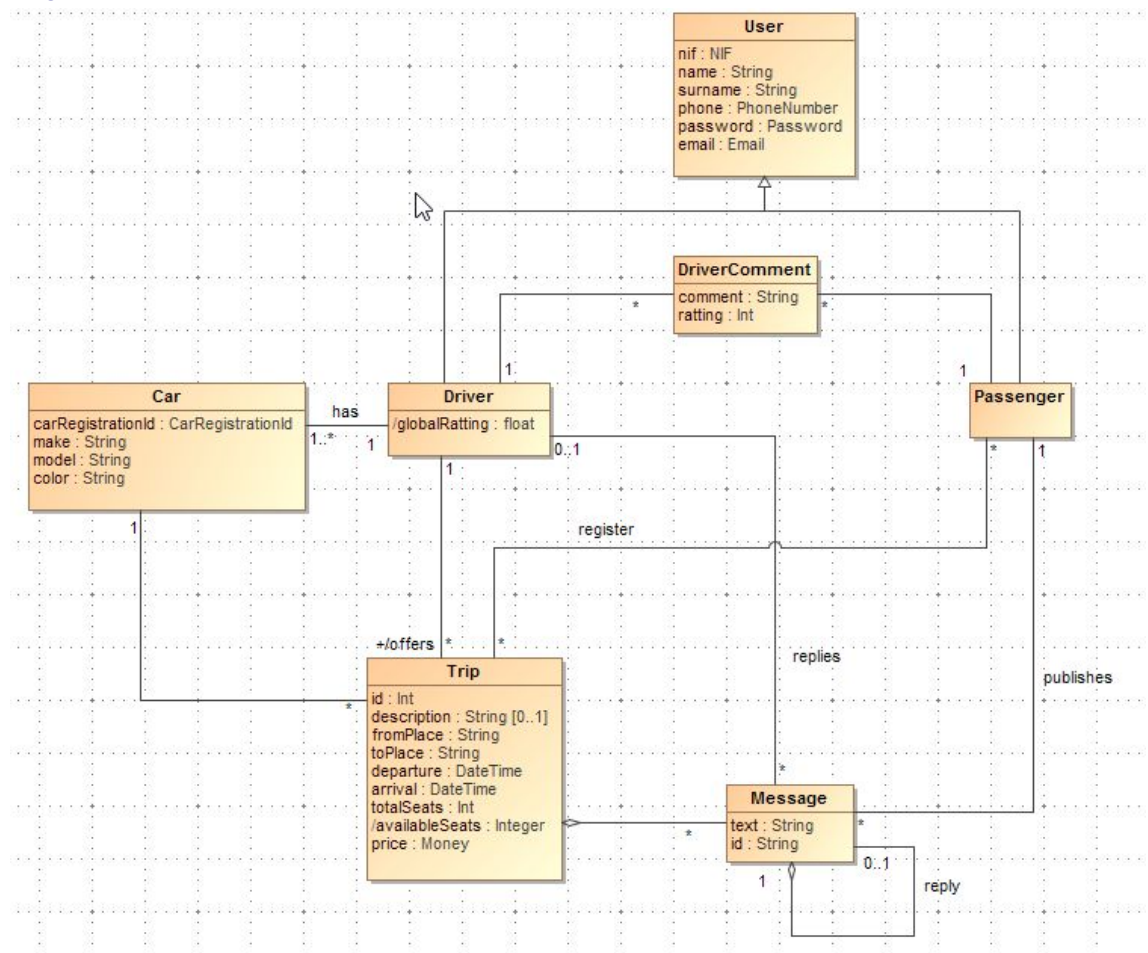
El éxito del sistema se basa en gran medida en la confianza y fiabilidad de la comunidad de usuarios que se forme, por eso es muy importante que los pasajeros puedan dejar una opinión breve sobre el conductor valorando aspectos como la puntualidad, el ambiente, la conducción, etc., y lo puedan puntuar con una puntuación de 0 a 10. La opinión y la puntuación serán siempre sobre el conductor y no sobre el trayecto.

Todos los trayectos deberán poder mostrar la puntuación media del conductor y los comentarios que hayan hecho sobre él los pasajeros.

Juan ya ha hecho 5 trayectos con este sistema y está muy contento, todas las opiniones sobre él han sido favorables, con descripciones como "Juan es el mejor compañero para un viaje relajado" o "Vehículo impecable y compañía agradable". Juan tiene una puntuación media en el sistema de 9 sobre 10.

Nota: Se puede considerar que se ha definido un tipo de datos para las contraseñas para reflejar que suele haber restricciones sobre qué contraseñas son válidas y que, por tanto, no suele ser un String cualquiera. Se ha hecho lo mismo para la dirección de correo, los importes, los números de teléfono, los NIF y las matrículas de los coches.

Diagrama de clases:



Restricciones de clave:

- User: nif
- Car: carRegistrationId
- Trip: id
- Message: id

Otras restricciones de integridad:

- (explícita) El identificador del trayecto debe estar generado por el sistema y debe ser valor numérico mayor que 0.
- (explícita) Las puntuaciones de un conductor van de 0 a 10.
- (implícita) Un conductor no puede ofrecer dos trayectos que coincidan en el tiempo.
- (implícita) Un pasajero no se puede apuntar a dos trayectos que coincidan en el tiempo.

Información derivada:

- La puntuación del conductor es la media de todas las puntuaciones que ha recibido.
- El número de plazas disponibles es el total de plazas menos el número de pasajeros apuntados.

Problema 20: Alquiler de coches

Queremos desarrollar el software para la gestión de una empresa de coches de alquiler. La empresa tiene una red de oficinas distribuidas por toda Europa. De las oficinas, queremos saber la dirección. Los clientes, de los que nos interesa saber el nombre y apellidos, dni, fecha en la que obtuvieron la licencia de conducción y los años de experiencia como conductores, pueden hacer reservas para alquilar los vehículos de la compañía. De cada reserva queremos saber la fecha y hora de la recogida del vehículo, fecha y hora de regreso del vehículo, la oficina desde la que se recogerá el vehículo y la oficina en la que se dejará el vehículo, que pueden ser diferentes.

El señor Carlos Loro, con DNI 12345678A, que tiene el carné de conducir desde el 1 de agosto de 2003, quiere alquilar un vehículo en la oficina de Barcelona-centro, desde el día 13 de marzo a las 09:00h hasta el día 20 de marzo a las 20:00h. El señor Loro ha elegido dejar el coche en la misma oficina.

Para cada reserva, el cliente debe indicar qué tipo de vehículo quiere alquilar, el cual puede ser del tipo económico, sedán, deportivo, familiar o furgoneta. Para poder alquilar un coche deportivo, el conductor debe tener, como mínimo, 5 años de experiencia como conductor.

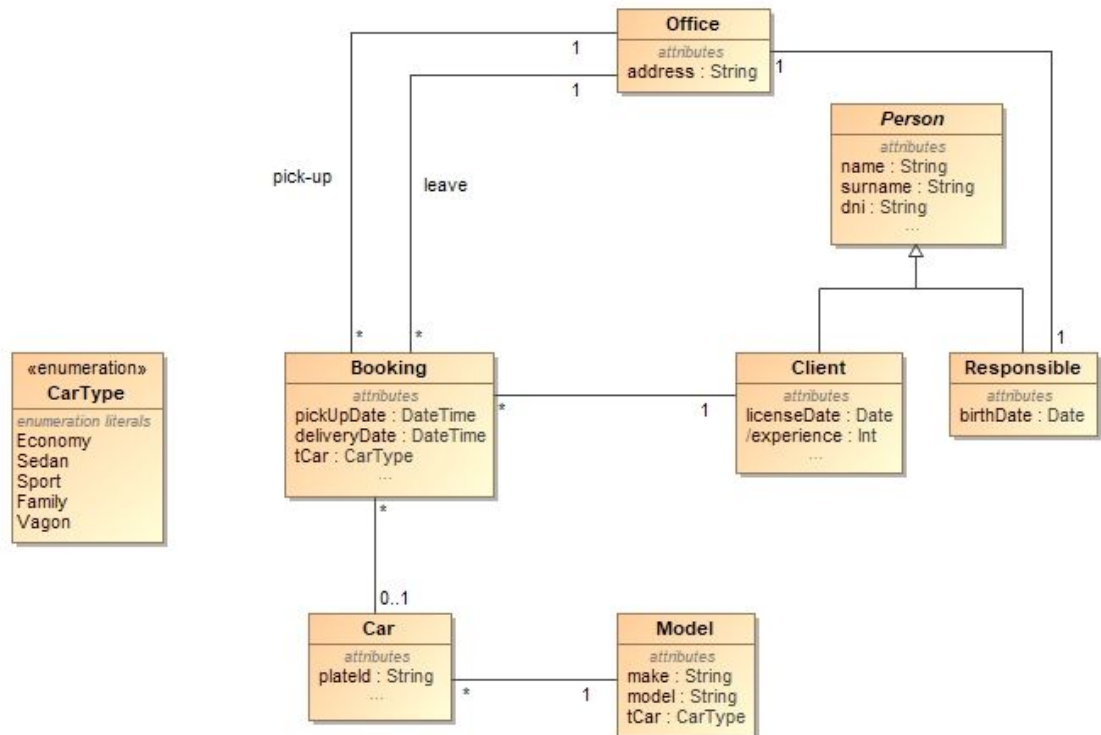
El señor Carlos Loro ha elegido alquilar un coche deportivo, y puede hacerlo ya que tiene más de 5 años de experiencia.

Cada oficina tiene un encargado del cual queremos saber su nombre, apellidos, dni y fecha de nacimiento. Un encargado sólo puede ser responsable de una única oficina. Para cada recogida que se realiza en una oficina, el encargado asigna a cada cliente un vehículo. Del vehículo queremos saber la marca, modelo y matrícula así como el tipo en el que está clasificado.

Carmen Sansa, con DNI 456789012B y que nació el 22 de noviembre de 1982, es la encargada de la oficina de Barcelona-centro.

Carmen Sansa asigna a la reserva del señor Carlos Loro el vehículo deportivo de la marca BMW, modelo Serie 1 y matrícula 1234 BCD.

Diagrama de clases:



Restricciones de clave:

- Person: dni
- Car: plateId
- Modelo: make + modelo
- Office: address
- Booking: cliente + pickUpDate

Restricciones de integridad

- Toda reserva con tipo de vehículo deportivo debe estar asociada con un conductor que tenga más de 5 años de experiencia.

Información derivada:

- La experiencia del Conductor se obtiene de la diferencia entre la fecha actual y la fecha en la que se sacó el carné de conducir.

Problema 21: Grupos musicales

Queremos desarrollar un Sistema de Información para una asociación cultural que quiere gestionar los grupos musicales asociados. La asociación cultural "CuatroEsquinas" deja que los alumnos aprendices de intérpretes de instrumentos formen grupos musicales. De los alumnos sabemos el nombre y el apellido y todos los grupos musicales tienen un nombre. Así, por ejemplo:

María Palau es una alumna que está aprendiendo a tocar el contrabajo y Guillem Biosca un alumno que toca el clarinete. Ambos forman parte del grupo "Somos dos pringados"

Un grupo musical lo pueden formar tantos alumnos como quieran, y cada alumno puede formar parte de tanto grupos musicales como desee.

En la asociación saben que no todos los alumnos pueden comprar los instrumentos que están estudiando y es por eso que han habilitado un mecanismo para alquilarlos. El alquiler de un instrumento por parte de un alumno sólo puede ser del instrumento que está aprendiendo a tocar. Y la asociación emite un contrato de alquiler: tiene una duración que comienza a contar a partir de una fecha y un precio (fuerza módico). De cada tipo de instrumento, del que sabemos el nombre, la asociación dispone de unos cuantos ejemplares, cada uno con un número de referencia.

María, por ejemplo, tuvo que hacer lo siguiente: "Tuve que alquilar el contrabajo, son muy caros para comprar uno. Y firmé un contrato con la asociación conforme me dejaban uno durante un trimestre, desde el 1 de octubre de 2016. Me lo dejaron por 50 euros. Yo tengo el contrabajo con la referencia CTB004"

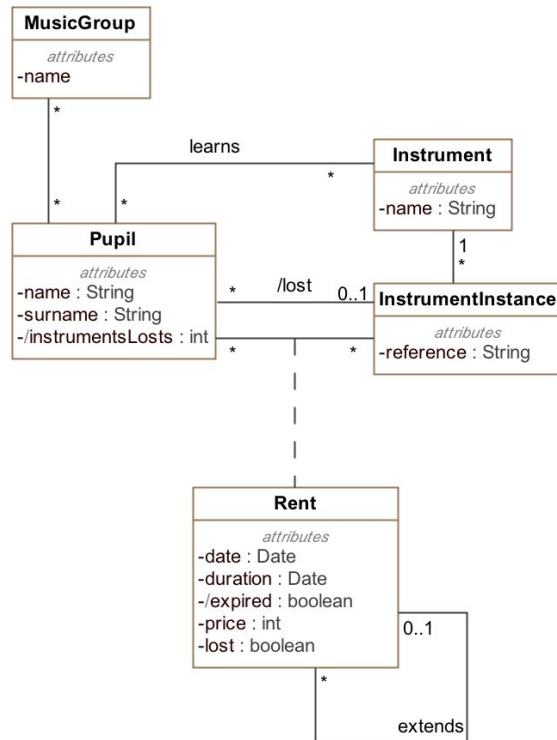
Los contratos de alquiler se pueden extender cuando el primero acaba. En tal caso, la asociación extiende el contrato anterior indicando las nuevas condiciones. Para estas operaciones, necesitamos saber de manera ágil si un contrato ha expirado o no.

"María ha extendido su contrato de alquiler del contrabajo. Le hemos hecho, pues, un nuevo contrato con las nuevas fechas, y en este caso pagará 50,5 €"

Un alumno no puede alquilar dos instrumentos del mismo tipo si ya tiene un contrato en curso de un instrumento (de este tipo). Para ello es necesario saber si un contrato ha expirado o no de manera sencilla.

En ocasiones los alumnos pierden el instrumento que han alquilado. Necesitamos saber en qué contrato se ha perdido para poder gestionar los trámites necesarios con nuestro servicio de seguro, que ya tramitamos a través de otro servicio que no interviene en este que estamos diseñando ahora. Hay que saber el número de instrumentos que un alumno ha perdido para gestionar posibles sanciones a estos.

Diagrama de clases:



Restricciones de clave:

- MusigGroup: name
- Pupil: name + surname
- Instrumento: name
- InstrumentInstance: reference

Restricciones de integridad:

- Un alumno sólo puede alquilar un instrumento que está aprendiendo.
- No pueden haber dos contratos de un mismo instrumento que tengan fechas que se solapen.
- Un contrato sólo puede extenderse a otro contrato que sea del mismo alumno.
- Un alumno no puede alquilar dos instrumentos del mismo tipo a la vez.

Información derivada:

- Hemos decidido añadir el atributo derivado */expired* a *Rent* para tener información rápida sobre si el contrato de alquiler ha expirado o no.
- Hemos decidido añadir la asociación derivada */ost* que asocia los instrumentos perdidos por parte de un alumno y el atributo derivado */instrumentsLosts* a *Pupil* para saber el total de instrumentos perdidos por parte de un alumno.

Problema 22: Grupos corales

Queremos desarrollar un Sistema de Información para una asociación cultural que quiere gestionar los grupos corales asociados. La asociación cultural "CuatroEsquinas" deja que los socios (que los llamaremos cantantes y de los que sabemos el nombre y el apellido) formen grupos corales. Así, por ejemplo:

María Palau es una cantante. Y forma parte de dos grupos corales: el coro "De ópera" y el coro "La Oscura".

Un grupo coral, que tiene un nombre, puede tener muchos cantantes y cada cantante puede estar en varios grupos, y claro, cada grupo tiene 2 o más cantantes.

Cada grupo coral usa partituras del fondo documental de la asociación. De cada partitura sabemos el número de registro en el fondo documental, el nombre y una descripción de la formación coral que la canta (coro de voces blancas, coro mixto a cuatro voces, con acompañamiento orquestal, etc ...). Y existen muchas copias, que son copias de la partitura original. Cada copia tiene un número que permite identificarla.

Por ejemplo: La partitura del "Brindis de la Traviata" a cuatro voces, con el número de registro OP_4567 la canta el grupo coral "De ópera", y existen 23 copias, cada una con un código que va del 1 al 23.

Cuando un cantante entra en un grupo coral, se le proporciona una copia de las partituras que este grupo coral interpreta. Cuando el cantante deja el grupo debe devolver todas y cada una de las copias que tiene; por lo tanto, un cantante no puede tener una copia de una partitura que cante un grupo coral al que no pertenezca.

María tiene la copia 23 de la partitura del "Brindis de la Traviata".

Un cantante puede perder una partitura. En tal caso, al cantante se le proporciona una nueva copia de la partitura y la copia anterior es necesario que quede marcada como perdida.

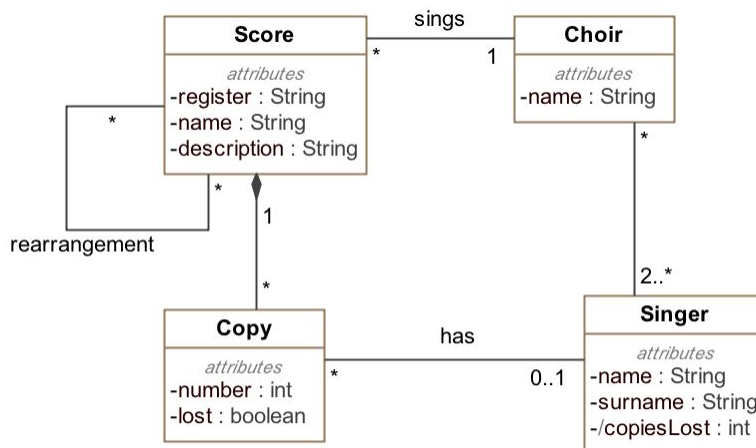
Revisando el historial de María, hemos visto que la copia 22 de la partitura del "Brindis de la Traviata" la había perdido.

Hay que saber el número de copias que un cantante ha perdido para gestionar posibles sanciones a estos.

Si una partitura es un arreglo, hay que saber de qué otra partitura lo es. Es habitual que en el mundo coral haya arreglos diferentes de una misma pieza y hay que saber qué arreglos están relacionados entre ellos.

Hay el arreglo del "Brindis de la Traviata" para coro de cuatro voces y otra versión de la misma pieza para coro de dos voces femeninas y dos masculinas.

Diagrama de clases:



Restricciones de clave:

- Choir: name
- Score: register
- Copy: Score (name) + number
- Singer: name + surname

Restricciones de integridad:

- Un cantante sólo puede tener copias de partituras del coro donde canta

Información derivada:

- Hemos decidido añadir el atributo derivado /copiesLosts a Singer para saber el total de copias perdidas por parte de un cantante.

Problema 23: Organización de conceritos

Queremos desarrollar un Sistema de Información para una asociación cultural que quiere gestionar los conciertos que organiza. La asociación cultural "CuatroEsquinas" organiza conciertos de muestra de intérpretes solistas. Un concierto tiene un nombre y se celebra en una fecha, y en una fecha concreta no puede haber dos conciertos.

La participación en los conciertos está limitada a un máximo de 15 intérpretes. De los intérpretes sólo guardamos el nombre, apellido y el mail de contacto.

El día 10/7/2017 tendrá lugar el concierto "Para todos un clamor", donde actuarán dos intérpretes: María Palau y Guillem Biosca.

Hay intérpretes que no pueden coincidir en un mismo concierto. Hay que saber qué intérpretes son, pues, incompatibles entre ellos y el motivo por el que lo son.

María Palau y Mario Roma no pueden coincidir en un mismo concierto porque tienen el mismo estilo musical.

Cada concierto tiene que funcionar como un espectáculo, por lo que la asociación define el orden de los intérpretes y por cada intérprete hay un texto introductorio que hay que leer.

En el concierto del día 10/7/2017, se ha decidido lo siguiente: Abrirá el espectáculo María Palau y luego Guillermo Biosca. Antes de la María se leerá una presentación del concierto y antes de

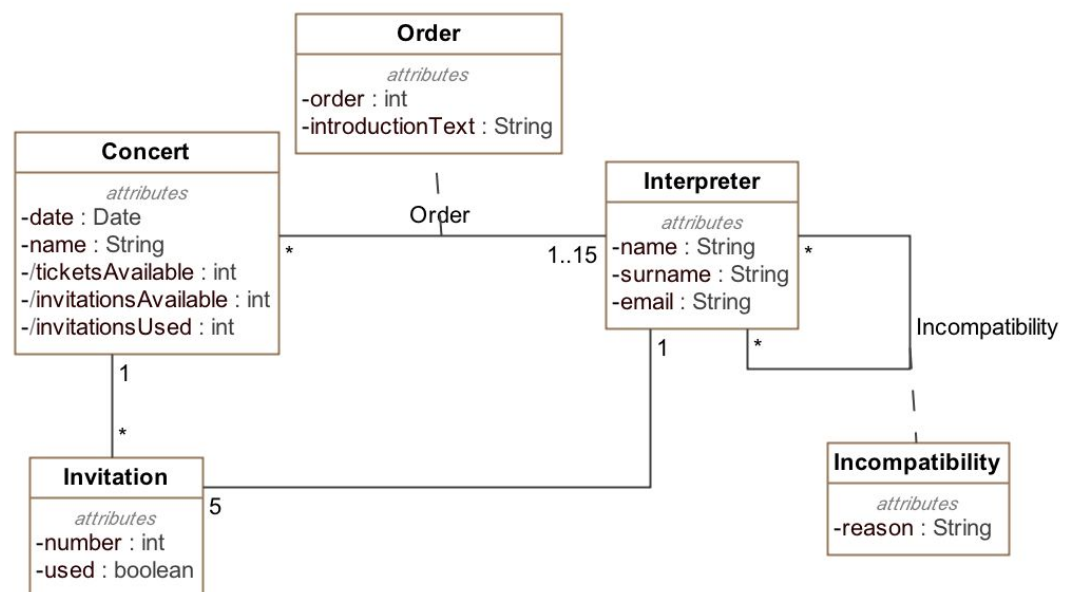
Para asistir a un concierto hay 100 entradas disponibles, que es el aforo de la sala de conciertos. Pero hay que tener en cuenta que a cada intérprete se le da un total de cinco entradas que podemos denominar invitaciones. Evidentemente las invitaciones deben ser del concierto donde participan los intérpretes. Hay que saber, pues, el número de entradas máximo de que se dispone, que será la diferencia entre 100 y las invitaciones proporcionadas.

De las 100 entradas disponibles para el concierto del día 07/10/2017, realmente sólo se dispone de 90, ya que las 10 restantes son los intérpretes; 5 por María y 5 por Guillermo.

Por cada invitación, hay que tener constancia de cuáles se han usado y cuáles no. De este modo, con el tiempo, la asociación sabrá si puede reducir las invitaciones y aumentar las entradas disponibles. Hay que saber el número de invitaciones disponibles y el número de invitaciones usadas por cada uno de los conciertos.

En el concierto del día 10/7/2017 por parte de María han ido tres personas y por parte de Guillermo sólo 1. Por tanto, sólo cuatro de las diez invitaciones se han usado.

Diagrama de clases:



Restricciones de clave:

- Concierto: date
- Interpreter: name
- Invitation: Concierto (date) + number

Restricciones de integridad:

- Un intérprete sólo puede disponer de invitaciones de conciertos donde actúe

Información derivada:

- Hemos decidido añadir el atributo derivado /ticketsAvailable, /invitationsAvailable, /invitationsUsed a Concert para tener constancia de las entradas disponibles, de las invitaciones entregadas y de las invitaciones usadas.

Problema 24: Cadena hotelera

Queremos desarrollar el software para la gestión de una cadena hotelera y sus diferentes hoteles. De cada hotel, queremos saber su nombre, dirección postal, su dirección web y su categoría en función del número de estrellas, que puede ser de una a cinco. Cada hotel tiene un gerente del cual queremos saber su nombre, documento de identidad, la fecha que comenzó a trabajar en la cadena hotelera y los años de experiencia que lleva en el cargo en el momento actual. El gerente sólo puede gestionar un único hotel a la vez.

El hotel Aiguaclara es un hotel de 4 estrellas situado en la calle Montserrat número 34 de Begur (Girona). Su dirección web es www.aiguaclara.com y está gestionado por Clara Farré, con documento de identidad 12345678A, que dirige el hotel desde el 1 de julio de 2013.

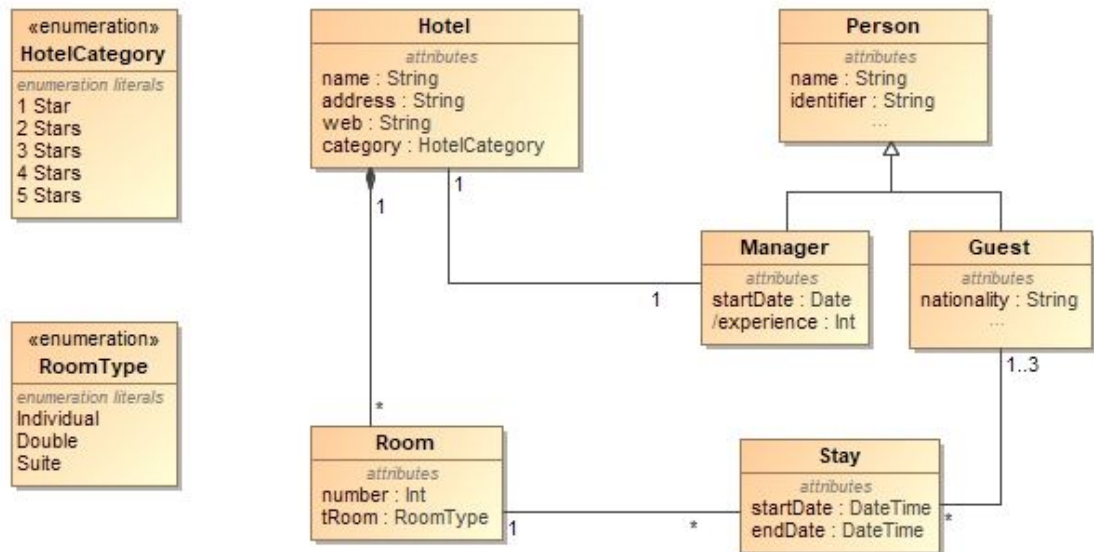
Queremos gestionar también las estancias que se producen en el hotel. Para cada estancia que se realiza en el hotel queremos saber la fecha de inicio y final de la estancia y las personas que se alojan. Para cada una de las personas alojadas queremos saber su nombre, su documento de identidad, y su nacionalidad.

La pareja francesa formada por Sophie Dómine, con documento de identidad 43210987B, y Logan Rosseau, con documento de identidad 98237645C, se alojaron desde el día 13 hasta el día 20 de julio.

A cada persona alojada se le asigna una única habitación durante toda su estancia. De la habitación queremos saber su número y el tipo de habitación, que puede ser: individual, doble o suite. Una habitación puede ser compartida entre hasta un máximo de 3 personas, teniendo en cuenta la limitación que las habitaciones individuales no pueden ser compartidas.

La pareja inglesa se alojó en la habitación doble 110.

Diagrama de clases:



Notas:

La categoría del hotel puede ser indicada como un enum, o como una restricción de integridad donde la categoría puede ir de 1 a 5.

Restricciones de clave:

- Hotel: Name + Address
- Person: identifier
- Room: hotel + number

Restricciones de integridad:

- Las instancias de Stay que estén asociadas con una Room de tipo Individual sólo pueden tener un Guest asociado.
- La fecha de inicio de cualquier instancia de Stay debe ser menor que su fecha final.
- En una fecha concreta, una instancia de Room sólo puede asignarse a una instancia de Stay.

Información derivada:

- La experiencia del Manager se obtiene de la diferencia entre la fecha actual y la fecha en la que comenzó a trabajar.