

Ingeniería del Software

PEC 2: Análisis UML

Presentación

Esta PEC se centra en el uso de UML para modelar diversos aspectos de un sistema de software.

La actividad cubre los contenidos estudiados en el módulo 4 de la asignatura. Se pide que la entreguéis en formato PDF a través del Campus Virtual. **Cualquier entrega que sea entregada en un formato incorrecto puede ver penalizada su nota.**

Nota previa

Recordad que, tal como se explica en el Plan docente de la asignatura, esta actividad debe resolverse de forma **individual**. En caso de detectar copias se penalizará la actividad con una D como nota.

Para garantizar que has resuelto esta actividad de forma individual, te pedimos que en tu solución entregada añadas el siguiente texto:

“Certifico que he realizado la PEC2 de forma completamente individual y solo con la ayuda que el profesorado de esta asignatura considera oportuna según las FAQs sobre plagio.”

Compromiso de autoresponsabilidad

Escribir aquí el texto con el compromiso de autoresponsabilidad que consta entrecomillado en la **nota previa**. **No escribirlo implica la no corrección de la PEC y una calificación de D.**

Enunciado

Pregunta 1 (10% puntuación)

Dibujad el elemento UML que se ajusta a la descripción de cada apartado siguiendo la notación gráfica descrita en el módulo 4. Para hacerlo, **no tenéis que usar ningún software de soporte al diseño UML; los tenéis que dibujar obligatoriamente a mano (papel, tablet...)**. En caso que lo hagáis en papel, usad preferentemente un escáner para su digitalización o una app tipo CamScanner (y en caso que lo hagáis con un móvil aseguráros que la imagen obtenida sea nítida).

Notas previas:

- Los elementos descritos en cada apartado son fragmentos de varios diagramas UML, inconexos entre sí, por tanto, no busquéis ninguna semántica en particular. Los apartados a), b) y c), corresponden a fragmentos de diagramas de clase, y el apartado d) corresponde a un fragmento de un diagrama de actividad.

El ejercicio tiene por objetivo que seáis capaces de aplicar correctamente la notación gráfica del lenguaje UML..

a) La clase abstracta `Car` tiene cuatro atributos :

- `model` de tipo `String`
- `power` de tipo `Int`
- `purchaseDate` de tipo `Date`
- `years` de tipo `Int`, derivado de `purchaseDate` (no es preciso indicar en el diagrama cómo se obtiene `years`)

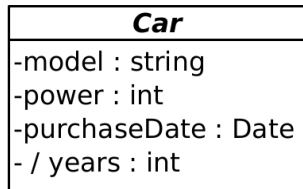
b) La clase `Student` tiene una asociación llamada `enrolled` con la clase `Subject` que se puede describir como:

- Un estudiante (`Student`) está matriculado cómo mínimo de una asignatura (`Subject`) y cómo máximo de diez.

- Una asignatura (`Subject`) está asociada a un número indeterminado de estudiantes (`Student`) (incluyendo la posibilidad de no tener ningún `Student` asociado).
- c) La clase `Feeling` es una clase asociativa que nace de la asociación `parenthood` entre dos personas (una asociación recursiva) representando la relación actual entre descendientes/ascendientes biológicos directos:
- La clase `Person` tiene dos atributos:
 - `name` de tipo `String`
 - `birthDate` de tipo `Date` (opcional)
 - La clase `Feeling` tiene dos atributos:
 - `love` de tipo `Float`
 - `hate` de tipo `Float`, derivada de `love` (no es preciso indicar en el diagrama cómo se obtiene `love`)
 - En esta asociación recursiva `parenthood`, una instancia de `Person` actúa con el rol de `child` y otra instancia actúa con el rol `parent`.
 - Una instancia de `Person` puede ser `child` de un máximo de 2 `Person` (puede darse el caso de conocer solo a uno de los padres biológicos o a ninguno).
 - Una instancia de `Person` puede actuar como `parent` de múltiples instancias de `Person` (incluyendo la posibilidad de no estar asociada a ninguna).
 - En este apartado se pide realizar dos diagramas para representar `Feeling`: uno como una clase asociativa y otro como una clase normal. ¿Cuál creéis que expresa mejor la situación expuesta?
- d) Tenemos un proceso que se puede describir mediante el siguiente flujo:
- el proceso inicia
 - una primera actividad que obtiene la edad de un espectador
 - si la edad es inferior a 16, se pide el permiso paterno y se autoriza la entrada (para simplificar asumimos que siempre se autoriza)
 - si la edad es igual o superior a 16, se autoriza la entrada
 - el proceso finaliza

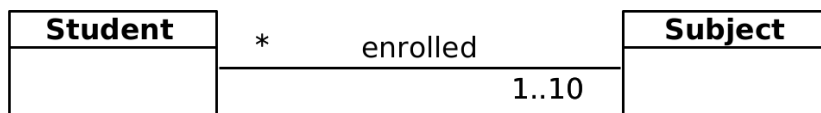
Solución

a)



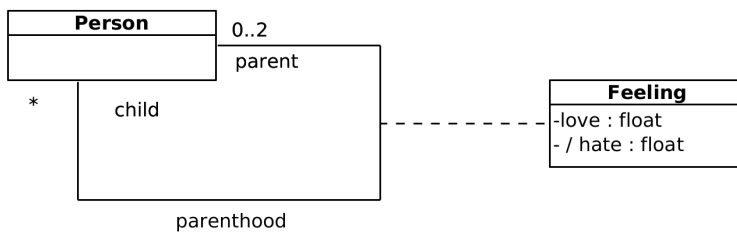
Powered ByVisual Paradigm Community Edition

b)

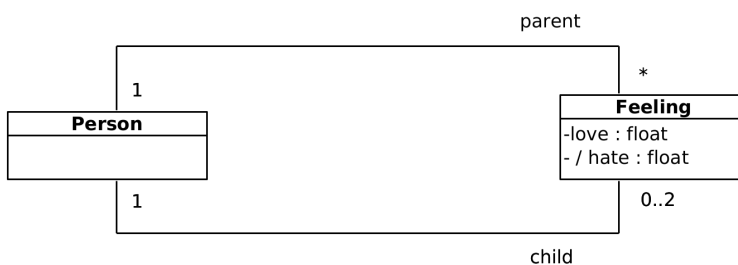


Powered ByVisual Paradigm Community Edition

c)



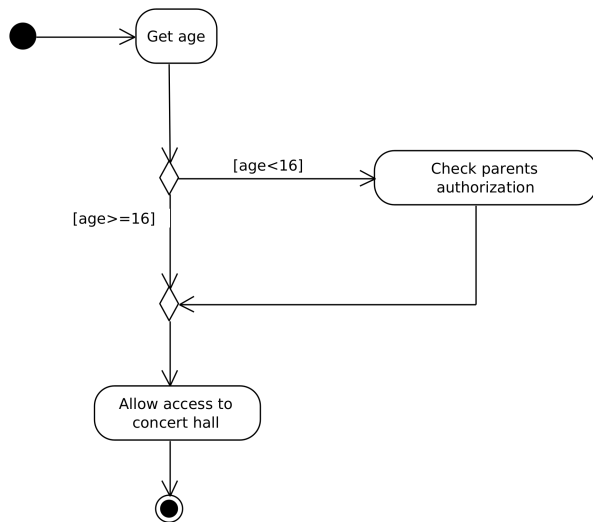
Powered ByVisual Paradigm Community Edition



Powered ByVisual Paradigm Community Edition

Con la clase asociativa, cada pareja *child/parent* tendrá un único *Feeling* asociado. Sin la clase asociativa, una pareja *child/parent* podría estar asociada dos veces, por lo tanto, tener asociados varios *Feeling*. En este caso, la clase asociativa expresa de forma más fiel la situación descrita en el enunciado: la clase *Feeling* describe la relación entre un ascendiente y un descendiente asumiendo - para simplificar - que habrá un único sentimiento (instancia de *Feeling*) entre ambos.

d)



Pregunta 2 (15% puntuación)

En la PEC1, en el ejercicio 6, esta fue la solución oficial:

Clases:

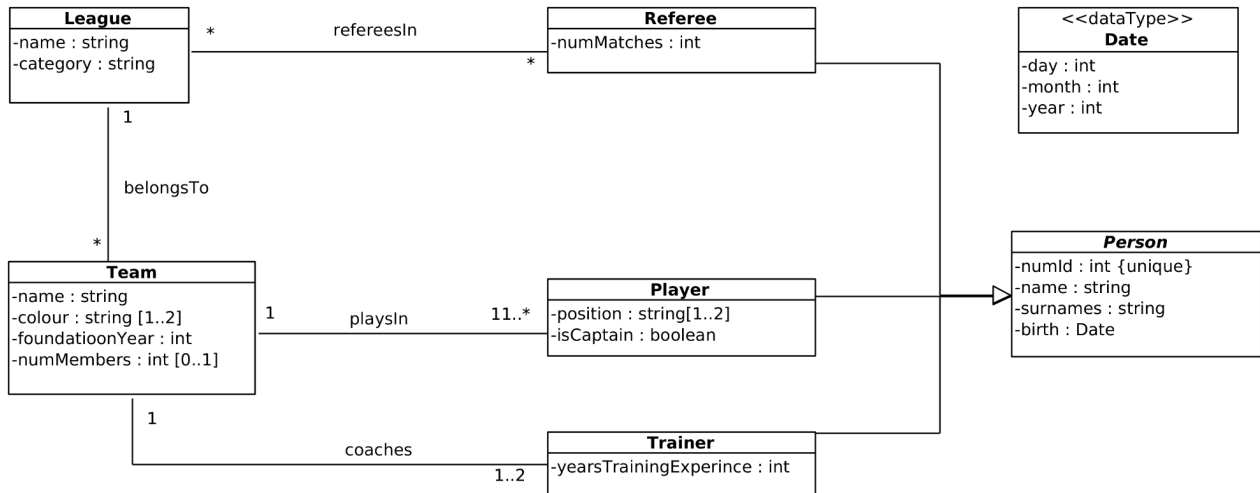
- *Liga (Concreta): nombre, categoría*
- *Equipo (Concreta): nombre, colores (multivaluado), añoFundación, numSocios (es válido considerarlo opcional para el caso en que no tenga socios)*
- *Persona (Abstracta): nombre, apellidos, añoNacimiento, numIdentificación*
- *Jugador (Concreta, subclase de Persona): posición (multivaluado), esCapitán*
- *Entrenador (Concreta, subclase de Persona): añosExperiencia*
- *Árbitro (Concreta, subclase de Persona): numPartidosArbitrados*

Asociaciones:

- *JuegaEn: Asocia cada instancia de Equipo con 11 o más instancias de Jugador y cada instancia de Jugador con una única instancia de Equipo.*
- *EntrenaA: Asocia cada instancia de Equipo con 1 o 2 instancias de Entrenador y cada instancia de Entrenador con una única instancia de Equipo.*
- *AsignadoA o ArbitraEn: Asocia cada instancia de Árbitro con cualquier número de instancias de Liga y cada instancia de Liga con cualquier número de instancias de árbitro.*
- *PerteneceA: Asocia cada instancia de Equipo con una única instancia de Liga y cada instancia de Liga con cualquier número de instancia de Equipo.*

Se pide: Realizar el **diagrama de clases UML** correspondiente al modelo descrito.

Solució



Powered by UML-Toolbox.com

Pregunta 3 (25% puntuación)

Una empresa de análisis energético nos pide una aplicación que permita hacer un seguimiento del estado de la distribución de gas por todo el mundo con el fin de poder elaborar sus informes periódicos.

La aplicación deberá trabajar con infraestructuras para obtener y distribuir gas. Cada infraestructura tendrá un nombre (que será único) y una fecha de puesta en marcha y se contemplarán tres tipos de infraestructuras:

- *Plantas regasificadoras*: de ellas conoceremos el nombre del puerto dónde se ubican (con el nombre es suficiente), la cantidad de gas licuado que pueden almacenar, la cantidad máxima de gas que pueden producir por hora y la cantidad que está produciendo en el momento actual.
- *Plantas de extracción*: debemos conocer el número de pozos de extracción con los que cuenta, la cantidad de gas máxima que pueden producir por hora y la cantidad producida en el momento actual.
- *Gasoductos*: tendremos información de los países por los que transita (con los nombres es suficiente), su capacidad máxima y el máximo de gas que puede suministrar (que será el máximo que pueden generar las plantas a él conectadas).

A cada planta regasificadora y a cada planta extractora llegará un único gasoducto pero un gasoducto puede estar alimentado por distintas plantas regasificadoras y extractoras. Existen nodos de conexión (identificados por un código distinto para cada uno de ellos y de los que conoceremos sus coordenadas en forma de latitud y longitud). Estos nodos son lugares dónde se conectan dos o más gasoductos. Un gasoducto podrá estar conectado a diversos nodos y, para cada nodo al que está conectado un gasoducto, se guardará el flujo máximo de gas que puede aportar (un número real).

Cada infraestructura estará gestionada por una corporación. De cada corporación conocemos su nombre (no habrá dos corporaciones con el mismo nombre), la dirección de la sede y su capital

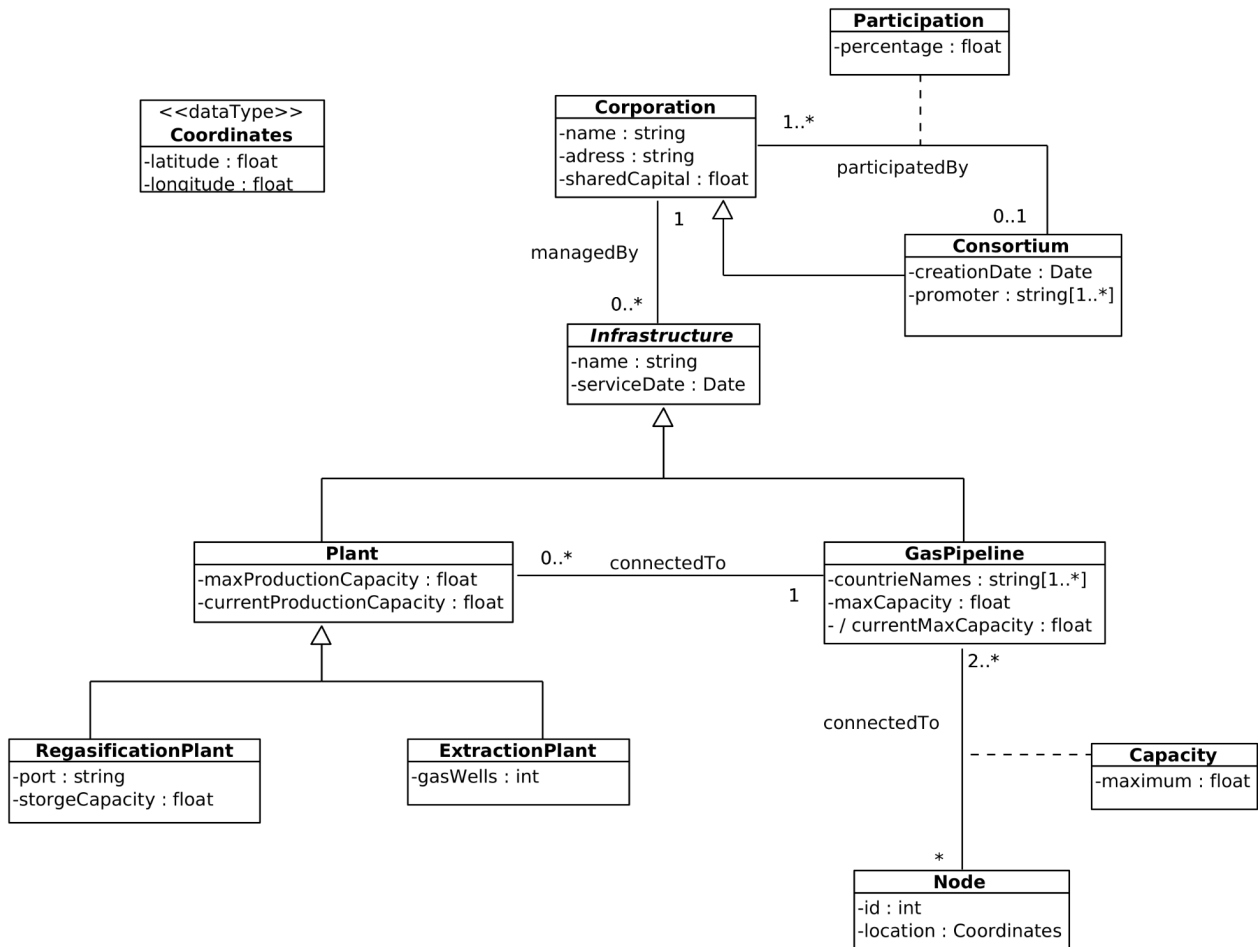
social. Contemplamos la posibilidad que alguna de estas corporaciones sea un consorcio formado por diversas corporaciones (para cada corporación miembro del consorcio debemos saber que porcentaje de participación representa en el total). Estos consorcios, a efectos de la aplicación, se podrán ver cómo una corporación más y de ellos nos interesa conocer su fecha de creación y el nombre de sus promotores (siempre habrá un promotor pero pueden tener diversos).

Se pide:

- a) Realizar el **diagrama de clases UML**.
- a) Documentar, de manera textual, las **restricciones de clave**, otras **restricciones de integridad** y describir la **información derivada** que pueda haber.

Solució

a) Diagrama de clases



En esta solución proponemos el nombre del país cómo una simple cadena de caracteres. También sería correcto plantear `Country` cómo un Data Type, o incluso cómo un clase junto a una asociación muchos a muchos entre `GasPipeline` y `Country`, en función de la información que quisiéramos incluir además del nombre.

La latitud y la longitud se han expresado cómo dos números pero también sería correcto representarlas en forma de grados, minutos y segundos.

b) Restricciones

Restricciones de clave:

- Infrastructure: name
- Node: id
- Corporation: name

Restricciones de integridad:

- maxCapacity de GasPipeline no puede ser inferior a la suma de las producciones actuales de las plantas a ella conectadas.
- la suma de los percentatge de Participation para todas las corporaciones asociadas a un consorcio debe sumar 100.
- currentProductionCapacity de Plant no puede ser superior a maxProductionCapacity.

Información derivada:

- GasPipeline: currentMaxCapacity se obtiene a partir de la suma de las capacidades de producción de las plantas a ella conectadas.

Pregunta 4 (25% puntuación)

La misma empresa de análisis energético nos pide ampliar el diseño de la aplicación anterior incorporando los elementos adicionales que se describen a continuación.

Las corporaciones que gestionan las infraestructuras pueden comprar lotes de gas licuado para alimentar sus plantas regasificadoras. Cada uno de estos lotes será transportado por unos cuantos barcos gaseros y será entregado en una o más plantas.

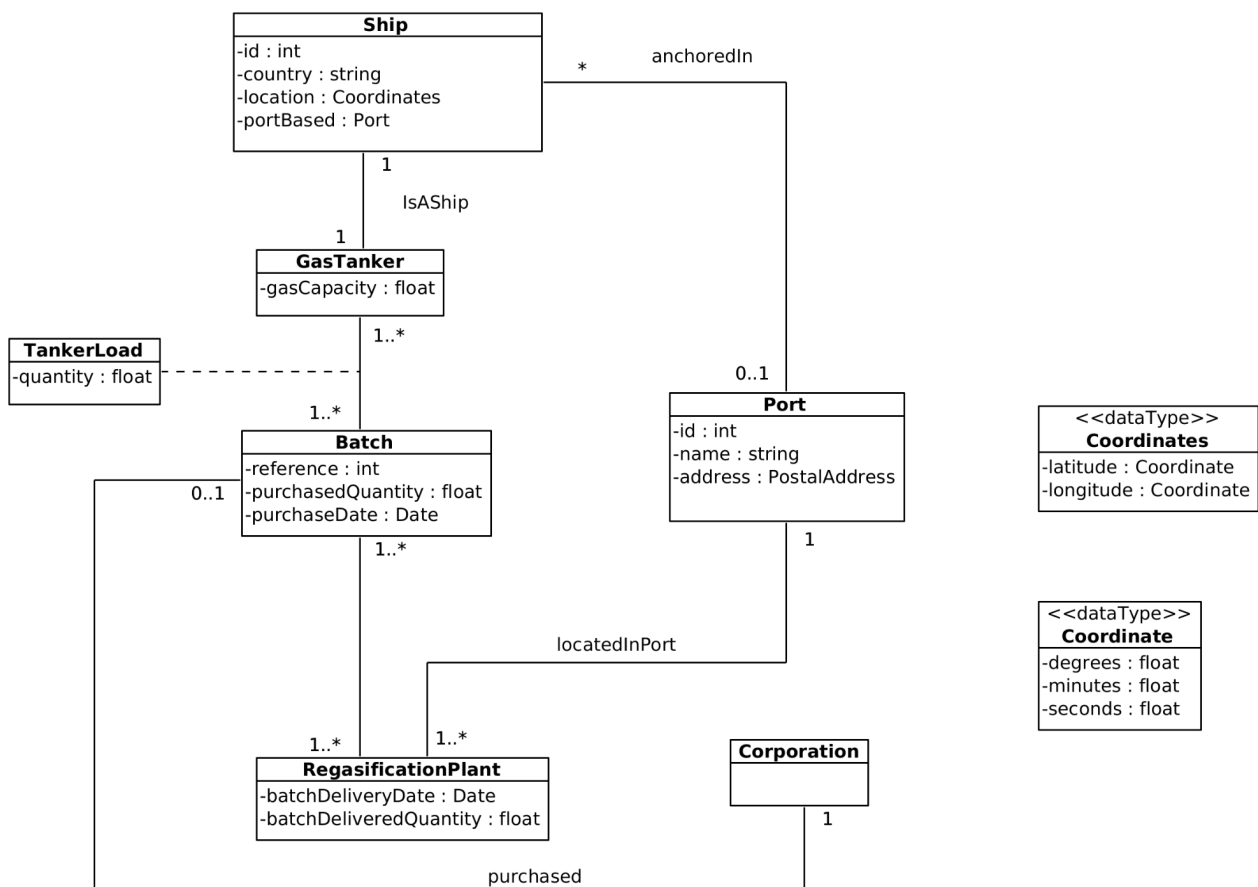
Un barco gasero es un tipo especial de barco del que conocemos su capacidad de carga de gas. Para todos los barcos tendremos un código y debemos conocer el país dónde está registrado, su posición actual (latitud y longitud expresadas en forma de grados, minutos y segundos), el recorrido realizado (un conjunto de coordenadas), su puerto base (siempre tendrá uno) y el puerto dónde está anclado (puede no estar en ningún puerto). Cada uno de estos barcos puede transportar gas licuado correspondiente a distintos lotes (deberemos conocer la cantidad de gas que transporta para cada lote) y cada lote podrá ser transportado por distintos barcos.

Cada uno de los puertos tendrá un código, un nombre y una dirección postal (calle, código postal, ciudad y país) correspondiente a sus oficinas centrales.

Los lotes tendrán un número de referencia (distinto para cada lote) y de ellos sabremos qué corporación lo ha comprado, el total de gas comprado, la fecha de la compra, los barcos que lo transportan y, para cada planta regasificadora la cantidad de gas a ser entregada y la fecha prevista de llegada de los barcos (para simplificar el problema asumimos que en caso que la entrega se realice por parte de más de un barco todos llegan el mismo día).

Las plantas regasificadoras estarán ubicadas en un puerto y en cada puerto puede haber un número indeterminado de plantas (incluyendo que no haya ninguna).

Para esta descripción de los requisitos, hemos realizado un primer análisis y hemos generado el siguiente diagrama de clases. **¡Pero hemos detectado algunos errores que no modelan correctamente la descripción anterior!**



Se pide

- Analizar el **diagrama de clases UML** e indicar los errores detectados. Para cada error, indicad una explicación del mismo y cuál es su solución.
- Realizar el **diagrama de clases UML** aplicando las soluciones que habéis escogido.

Solució

Los errores detectados son:

Error:

- Descripción: Falta el atributo “journeyTrack” en la clase Ship.
- Solución: Añadir el atributo “journeyTrack” de tipo Coordinates[0..*] en la clase Ship.

Error:

- Descripción: El atributo “portBased” en la clase Ship es un uso de clave foránea.
- Solución: Tendría que expresarse en forma de asociación entre Ship y Port con cardinalidad 0..* en el extremo de Ship y 1 en el extremo de Port.

Error:

- Descripción: La asociación “locatedInPort” tiene la cardinalidad en el extremo RegasificationPlant errónea.
- Solución: Tendría que ser 0..*, ya que un Port puede no tener ninguna RegasificationPlant.

Error:

- Descripción: Los atributos “batchDeliveryDate” y “batchDeliveredQuantity” de la clase RegasificationPlant son en realidad atributos de la asociación entre Batch y RegasificationPlant.
- Solución: Crear una clase asociativa entre Batch y RegasificationPlant con cardinalidad 1..* en los dos extremos con los dos atributos.

Error:

- Descripción: La asociación 1:1 isAShip entre Ship y GasTanker no debería ser una asociación: gasTanker es un barco y, por tanto, gasTanker es una especialización de Ship.
- Solución: Tenemos que cambiar la asociación por una especialización: GasTanker es una subclase de Ship.

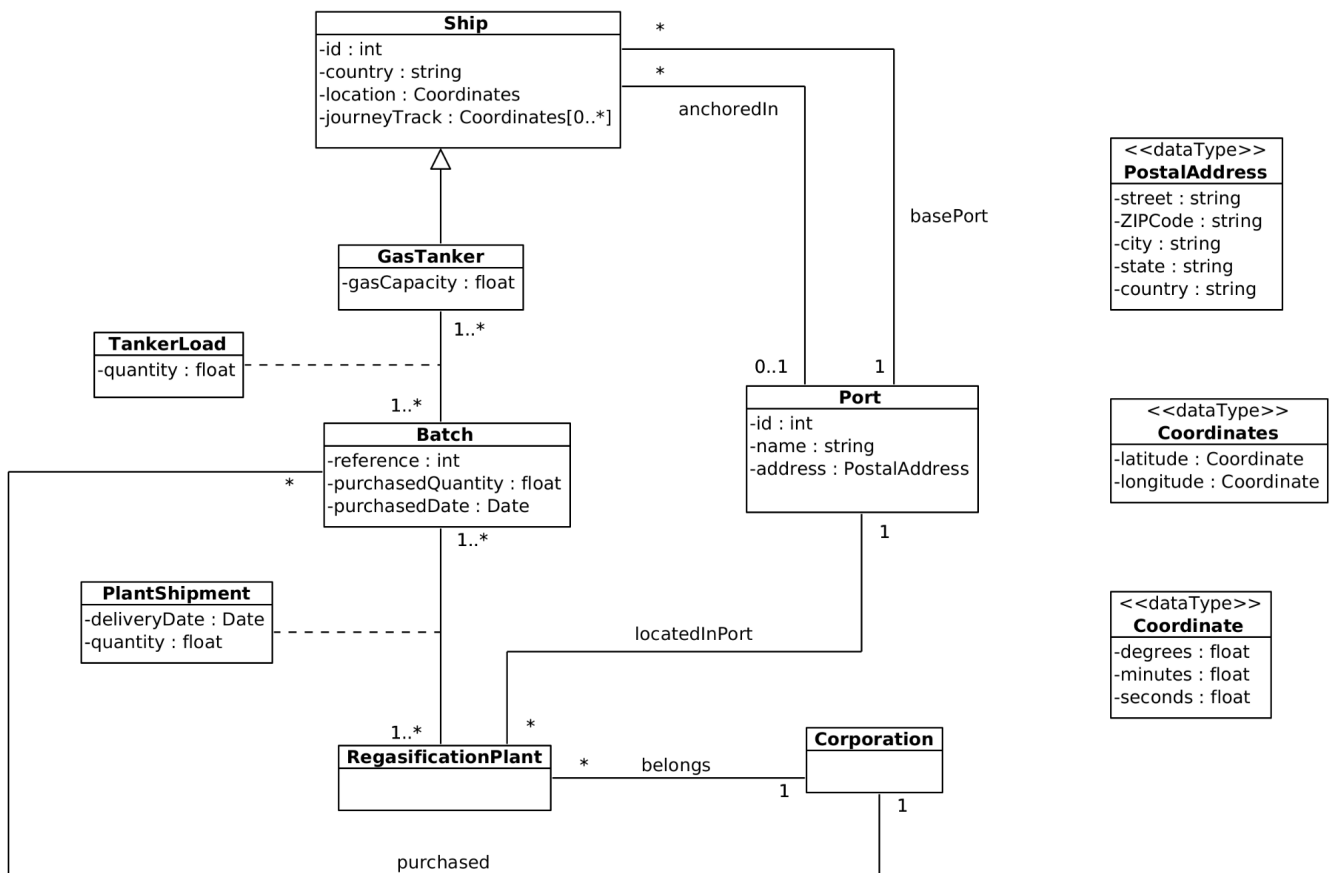
Error:

- Descripción: PostalAdress no es un tipo conocido
- Solución: Tenemos que crear un DataType PostalAddress con los atributos necesarios (calle, ...).

Error:

- Descripción: Falta la información del propietario de las RegasificationPlant
- Solución: Añadir una asociación entre RegasificationPlant y Corporation con cardinalidad 1 en el extremo Corporation y 1..* en el de RegasificationPlant.

El diagrama aplicando las anteriores soluciones:



Powered by Virtual Paradigm Community Edition

Pregunta 5 (10% puntuación)

Continuamos con la aplicación anterior pero ahora empezaremos a centrarnos en sus funcionalidades.

En el sistema tendremos dos roles de usuarios: los auxiliares y los responsables. Todas las acciones disponibles para los auxiliares lo están también para los responsables pero éstos podrán realizar acciones disponibles únicamente para ellos.

Los auxiliares, una vez identificados en el sistema, podrán crear lotes, consultar los barcos gaseros, las infraestructuras y los nodos.

Los responsables podrán dar de alta barcos, infraestructuras (indicando qué corporación las gestiona), nodos y corporaciones y podrán consultar información sobre la gestión de las infraestructuras de una corporación.

Se pide: Identificar los casos de uso y representarlos en un **diagrama de casos de uso**.

Solución



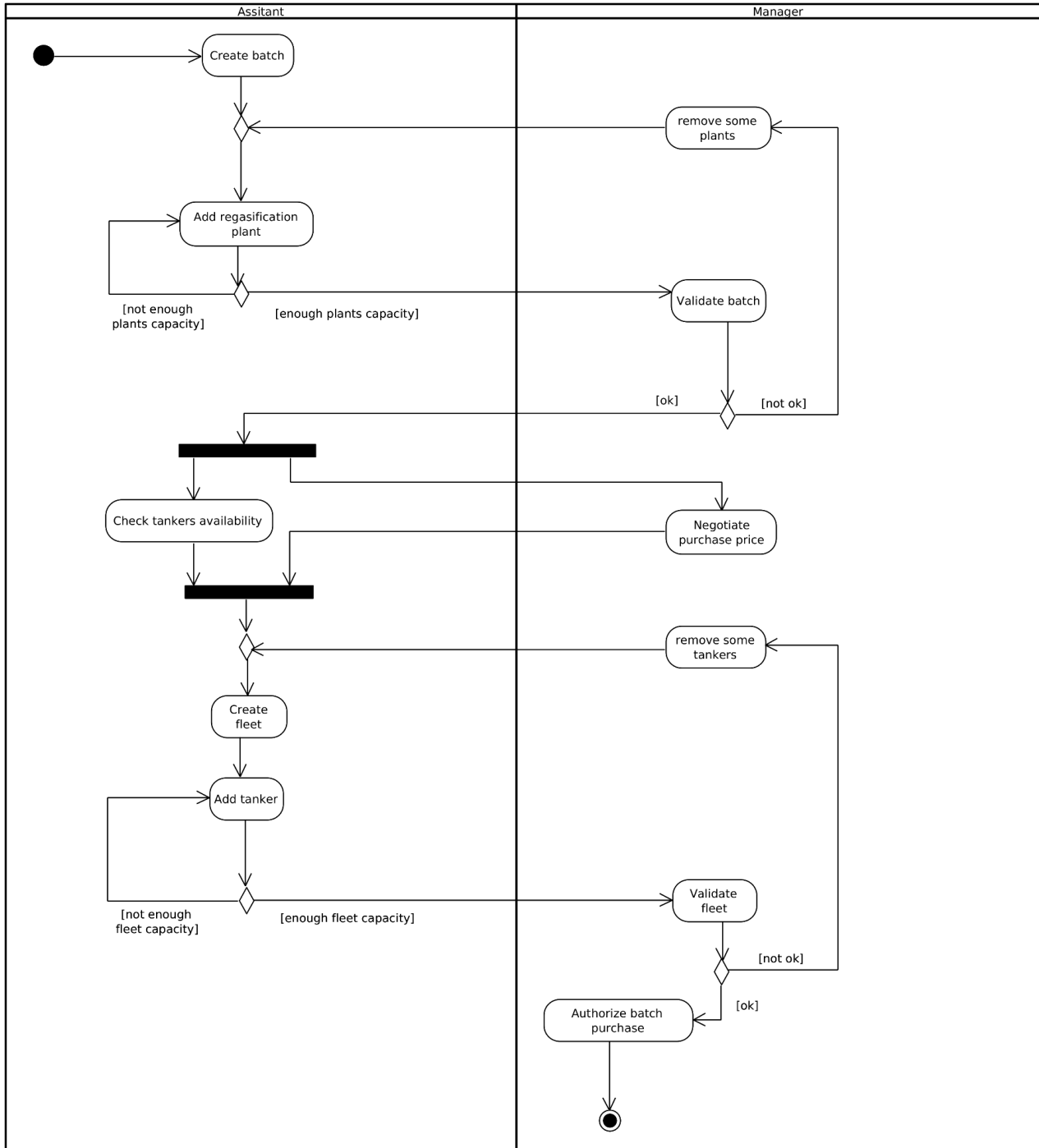
Pregunta 6 (15%)

Queremos modelar la creación de un lote para la adquisición de cierta cantidad de gas licuado. Para ello, el personal auxiliar de la empresa crea un nuevo lote y va añadiendo a éste plantas regasificadoras hasta conseguir que la capacidad de almacenamiento de ellas sea suficiente para almacenar la cantidad de gas que se quiere comprar.

Una vez añadidas las plantas necesarias, una persona responsable de la empresa validará el estado actual del lote. En caso que no lo valide, eliminará algunas de las plantas del lote y el personal auxiliar retomará la tarea de añadir plantas hasta conseguir la capacidad necesaria. Cuando la persona responsable haya validado el lote, empezará a negociar el precio de adquisición del gas mientras el personal auxiliar comprueba la disponibilidad de barcos gaseros. Una vez el precio se ha negociado y se dispone de una lista de barcos, el personal auxiliar empezará a crear la flota para el transporte de un modo parecido al de las plantas: va añadiendo barcos a la flota hasta alcanzar la capacidad que debe ser transportada. Una vez conseguida esta capacidad, la persona responsable validará la flota y podrá quitar algunos barcos en caso que no sean de su agrado o la dará por buena. En caso que quite algunos barcos, el personal auxiliar deberá retomar la tarea de añadir nuevos barcos para completar la flota y, cuando la persona responsable valide la flota, ésta finalmente autorizará la compra del lote y acabará el proceso.

Se pide: Realizar el **diagrama de actividades UML** correspondiente al proceso descrito.

Solución



UML Activity Diagram - Version 2.5.1