

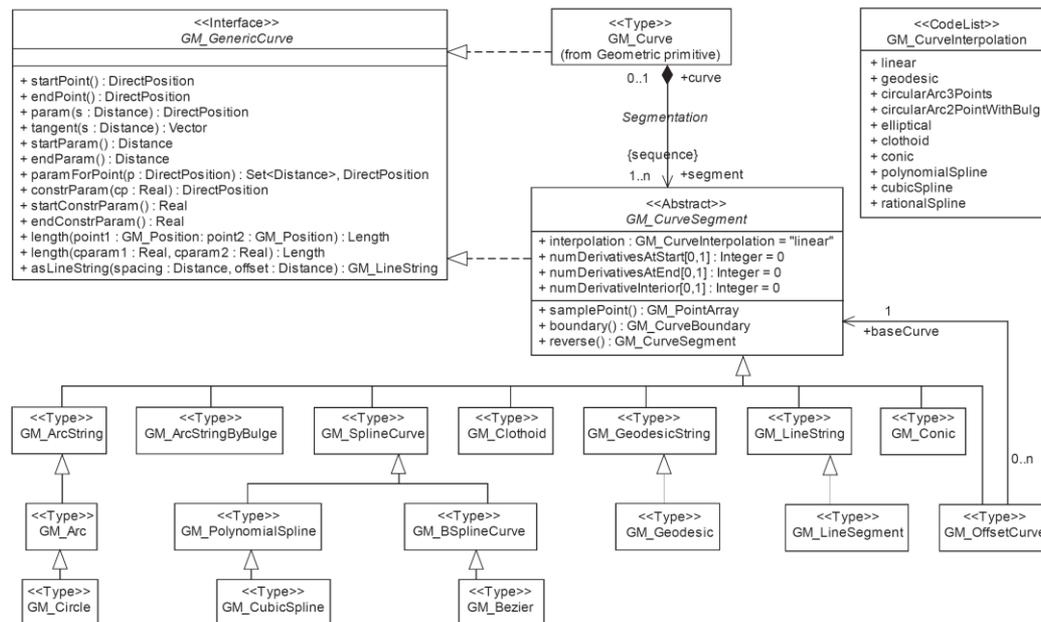


Modelos de datos: Lenguaje UML y prácticas

Paloma Abad, Alejandra Sánchez (CNIG)

- Los modelos de datos:
 - Es la búsqueda de la estructura de datos óptima
 - Cuántos ficheros, registros, campos, claves,...
 - Formatos, longitudes de campos...
 - Para explotar un SI
 - Para intercambiar datos
 - Para armonizar un campo de aplicación
 - Son independiente de plataformas
 - Facilita: migraciones, ampliaciones, y mejoras

- UML (Unified Modeling Language):
 - Lenguaje Unificado de Modelado
 - define una notación a través de diagramas que sirven para representar modelos o subsistemas
 - Es un lenguaje gráfico
 - detrás de cada símbolo hay una semántica bien definida
 - como todo lenguaje sigue unas reglas



- Es un lenguaje de modelado estándar Orientado a Objetos (OO)
- Sirve para visualizar, comunicar, construir y documentar el sistema
- El modelo describe los elementos de un sistema y sus relaciones, pero NO indica como implantarlo
- Las normas ISO 19100 utilizan los modelos UML
- La Norma ISO 19103 – Lenguaje de Esquemas Conceptuales
 - Establece un perfil de UML para su uso en el campo de la IG
- Las especificaciones OGC utilizan los modelos UML
- Las especificaciones de datos Inspire utilizan los modelos UML

- ISO/TS 19103
 - Es normativo usar UML
 - Con un estilo determinado
- Conformidad con
 - ISO 19109 Reglas para Modelos de Aplicación
 - ISO 19107/ ISO 19137 Esquema espacial
- Especificaciones Inspire



Teoría UML

17 de junio de 2015

- Si observamos a nuestro alrededor, estamos rodeados de cosas que poseen ciertas propiedades (**atributos**) y que realizan ciertas acciones (**operaciones**)
 - Ejemplo: proyector a color que se enciende, proyecta y se apaga
- Esas cosas las podemos agrupar en categorías, es a lo que llamamos: **clases**
- Una clase es un grupo de cosas que tienen atributos y acciones similares
 - Ejemplo de clase: Lavadora
 - Atributos: marca, modelo, número de serie y capacidad
 - Operaciones: agregar ropa, poner detergente, activarse, sacar ropa

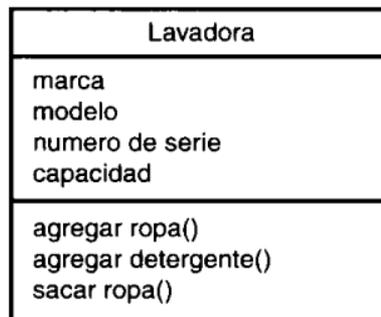
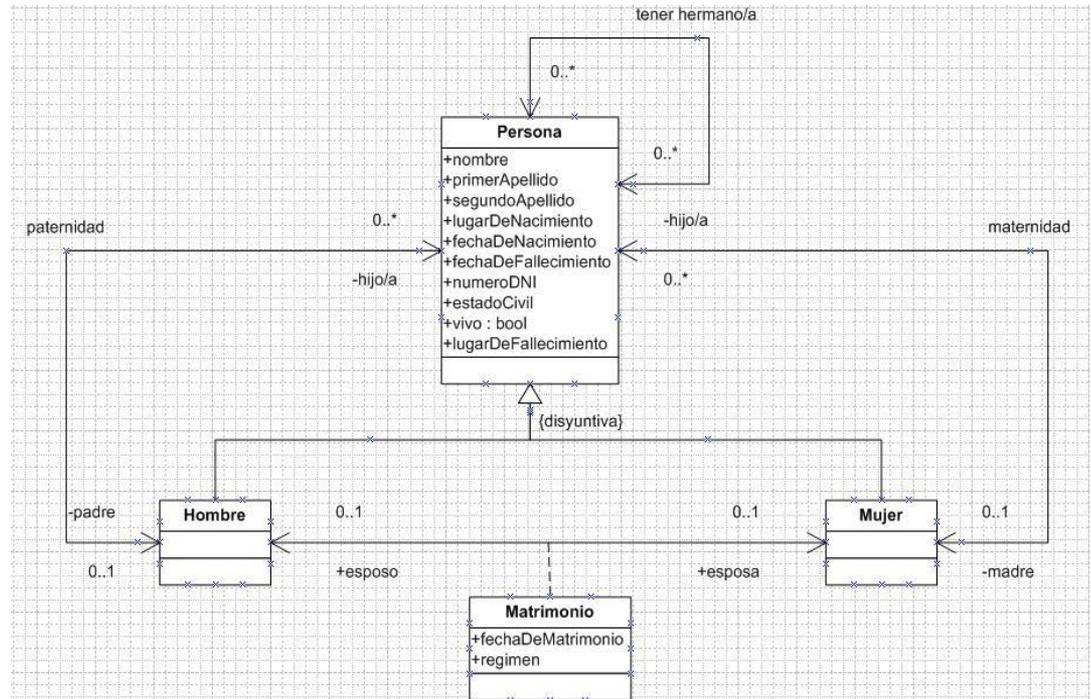


Diagrama de clases

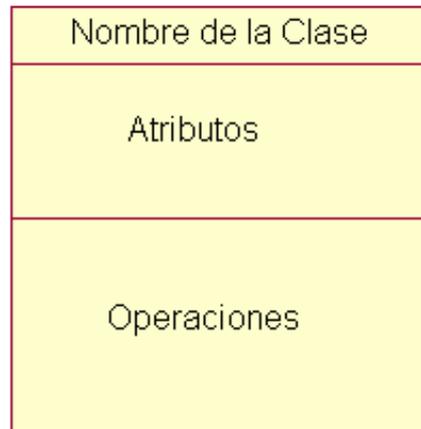
- El diagrama de clases es el diagrama principal para el análisis y diseño de un sistema
- Muestra las clases del sistema, sus interrelaciones y las operaciones y atributos de las mismas



- Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase).
- Definición: Conjunto de objetos con propiedades (atributos) similares, comportamiento (operaciones) común, relaciones comunes con otros objetos y semántica común

Propiedades, atributos o características

Valores que corresponden a un objeto, como color, material, cantidad, ubicación. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso



Operaciones o métodos,

Aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.

Ejemplos de CLASES

NombreClase
atributo1 atributo2
operación1() operación2()

Automovil
marca modelo peso
arrancar() correr() Parar()

Automovil
Nº de puertas Airbag GPS D. Asistida Bluetooch

Automovil
Propietario Precio Año

Ejemplos de CLASES

NombreClase
atributo1
atributo2
<<voidable>>
atributo3
atributo4

«featureType» <i>HydroObject</i>
«voidable»
+ geographicalName: GeographicalName [0..*]
+ hydroId: HydroIdentifier [0..*]

«featureType» <i>SurfaceWater</i>
+ geometry: GM_Primitive
+ inspireId: Identifier
+ levelOfDetail: MD_Resolution [0..1]
«voidable, lifeCycleInfo»
+ beginLifespanVersion: DateTime
+ endLifespanVersion: DateTime [0..1]
«voidable»
+ localType: LocalisedCharacterString [0..1]
+ origin: OriginValue
+ persistence: HydrologicalPersistenceValue
+ tidal: Boolean

Propiedades o Atributos

Los atributos pueden representarse solo mostrando su nombre, mostrando su nombre y su tipo, e incluso su valor por defecto.

Public (+): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todas las clases.

Private (-): Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus operaciones pueden acceder). Sólo la clase original puede utilizar ese atributo u operación

Protected (#): Indica que el atributo no será accesible desde fuera de la clase, pero a las subclases que se deriven. La funcionalidad se otorga sólo a las clases que heredan de la clase original

Automóvil
+ marca
- modelo
peso

Propiedades o Atributos

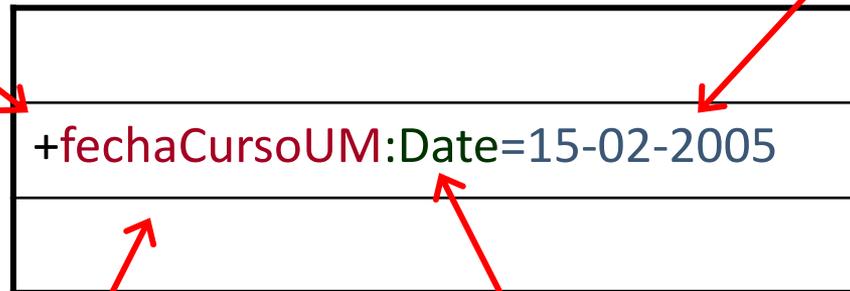
nombreAtributo

nombreAtributo : tipoDato

nombreAtributo : tipoDato = valorPorDefecto

publico +
privado -
protegido #

Valor inicial

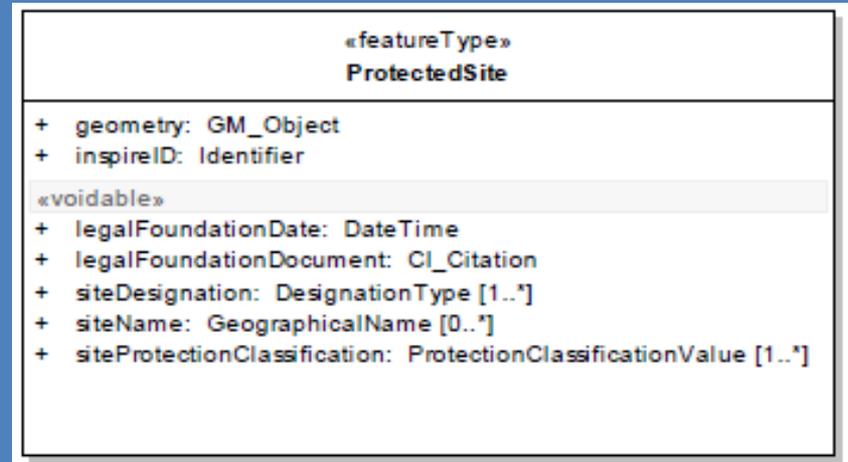
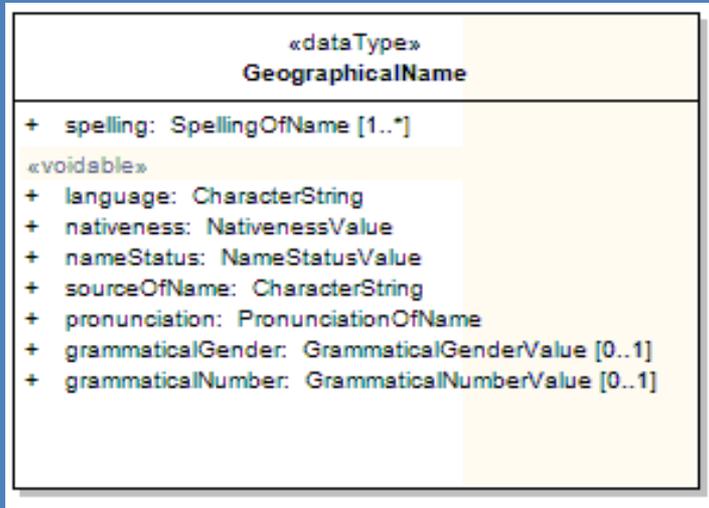


Nombre del atributo

Tipo

17 de junio de 2015

- **«vaciable»** (voidable, que puede quedar vacío), un atributo o rol de asociación al que se le puede asignar un valor de «vacío» (void) si los conjuntos de datos espaciales mantenidos por los Estados miembros no contienen un valor que corresponda o no puede obtenerse un valor que corresponda a partir de los valores existentes a un coste razonable. Si un atributo o rol de asociación no es vaciable (voidable), se deja en blanco la celda del cuadro que especifica su vaciabilidad (voidability, posibilidad de quedar vacío).



- [1] es obligatorio
- [0..1] es opcional
- «voidable» significa: si se conoce el valor, es obligatorio

«dataType» GeographicalName
+ spelling: SpellingOfName [1..*]
«voidable»
+ language: CharacterString
+ nativeness: NativenessValue
+ nameStatus: NameStatusValue
+ sourceOfName: CharacterString
+ pronunciation: PronunciationOfName
+ grammaticalGender: GrammaticalGenderValue [0..1]
+ grammaticalNumber: GrammaticalNumberValue [0..1]

- **<<Voidable>>**

INSPIRE distingue los siguientes casos:

- «unpopulated»: el atributo no se puede completar debido a que el propietario de la información no consideró necesario almacenar el dato.
- «unknown»: indica que el propietario de la información desconoce el valor del atributo y por tanto no se puede completar el atributo.
- «withheld» se dispone del dato pero es confidencial.

No se debe equiparar el concepto de voidable con la cardinalidad mínima 0. Cuando un atributo tiene cardinalidad 0 quiere decir que ese atributo puede no estar presente en el mundo real para algunas ocurrencias de un mismo fenómeno. Por ejemplo la fecha final del ciclo de vida de un objeto espacial tiene como cardinalidad mínima 0, ya que el objeto puede no haberse dado de baja.

Operaciones o Métodos

Las operaciones son funciones o transformaciones que se aplican a todos los objetos de una clase particular. La operación puede ser una acción ejecutada por el objeto o sobre el objeto.

Nombre de la Clase
Lista de Atributos
Lista de Operaciones

Persona
Nombre Edad
Trabajar() Votar()

Universidad
Nombre Dirección
Enseñar() Graduar()

Rectángulo
Ancho Largo
Dibujar() Borrar()

Operaciones o Métodos

«dataType» MarineExtent
<ul style="list-style-type: none"> + geometry :GM_MultiSurface + waterLevel :WaterLevelValue [0..1]

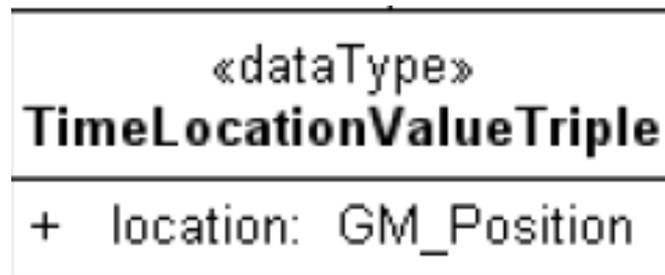
publico +
privado –
protegido #



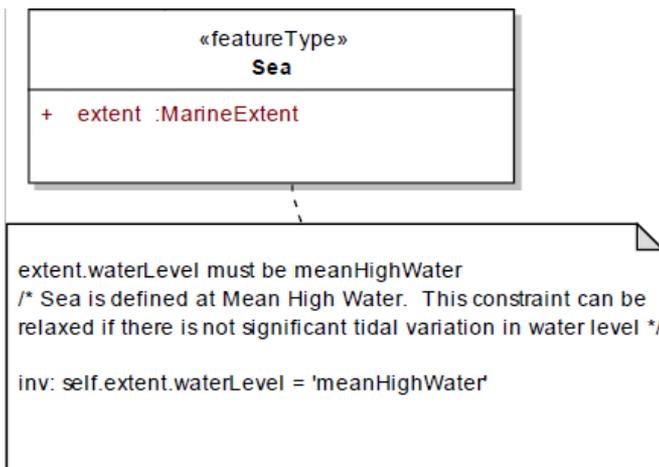
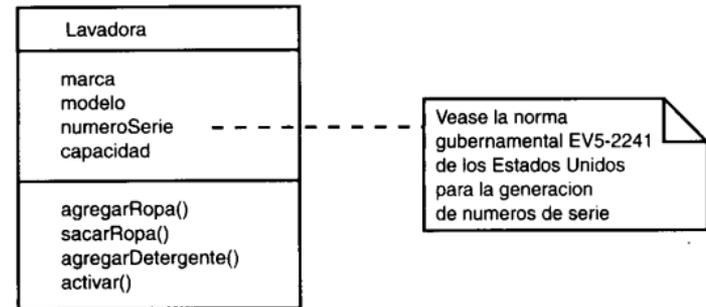
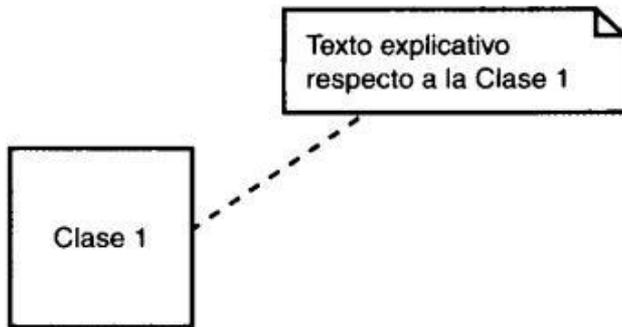
Nombre de la
operación y el
argumento

Tipo de retorno

- La Especificación Técnica ISO/TS 19103 especifica que el nombre de una clase no debe contener espacios en blanco y que las palabras individuales contenidas en el nombre deben empezar con letra mayúscula. Además
 - No debe llevar acentos
 - No utilizar la “ñ”
 - No utilizar “-” , ni “_”
 - Tan corto como sea posible
 - Clases, paquetes y asociaciones: Empiezan con mayúscula
 - Atributos, operaciones y roles: Empiezan con minúsculas
 - En caso de contener dos palabras, la segunda empieza con mayúsculas



- ❑ **Notas:** Para agregar información a la clase o comentar, también para especificar restricciones o condiciones
 - Cuadrado con esquina doblada



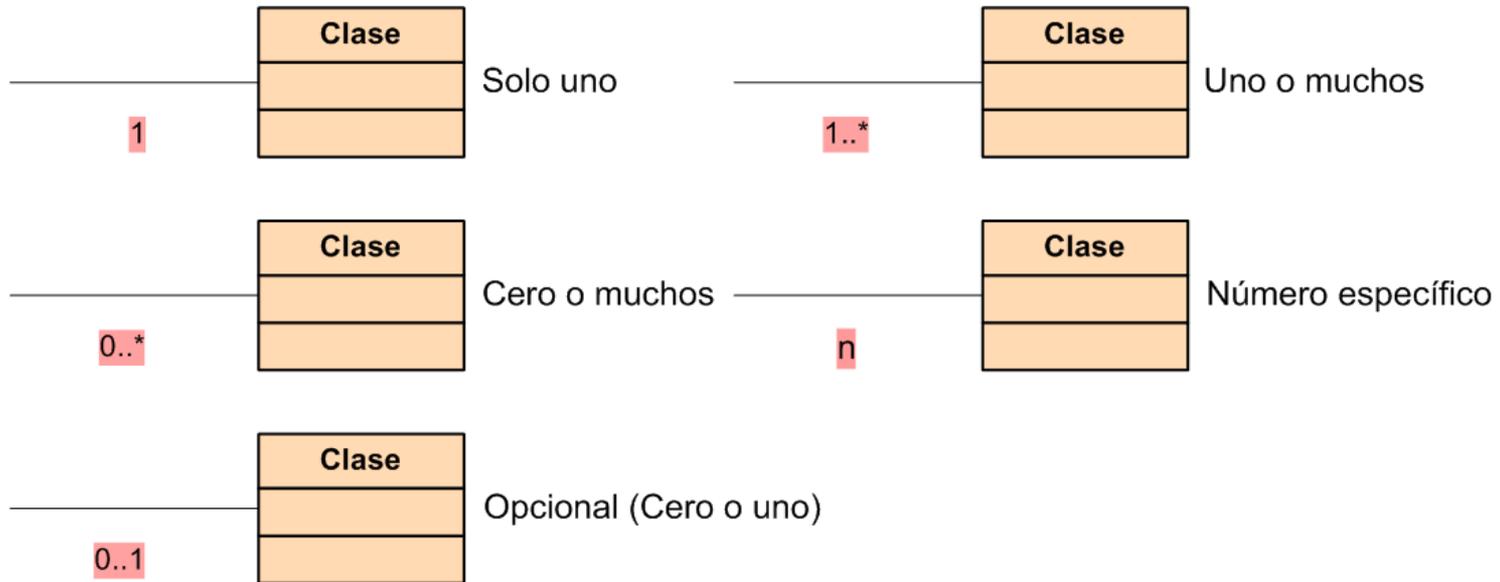
Las restricciones: pueden explicarse utilizando lenguaje natural o OCL (Object Constraint Language), que es el lenguaje de restricciones de UML

“Una restricción es una limitación sobre uno o más valores de (parte de) un modelo o un sistema orientado a objetos”

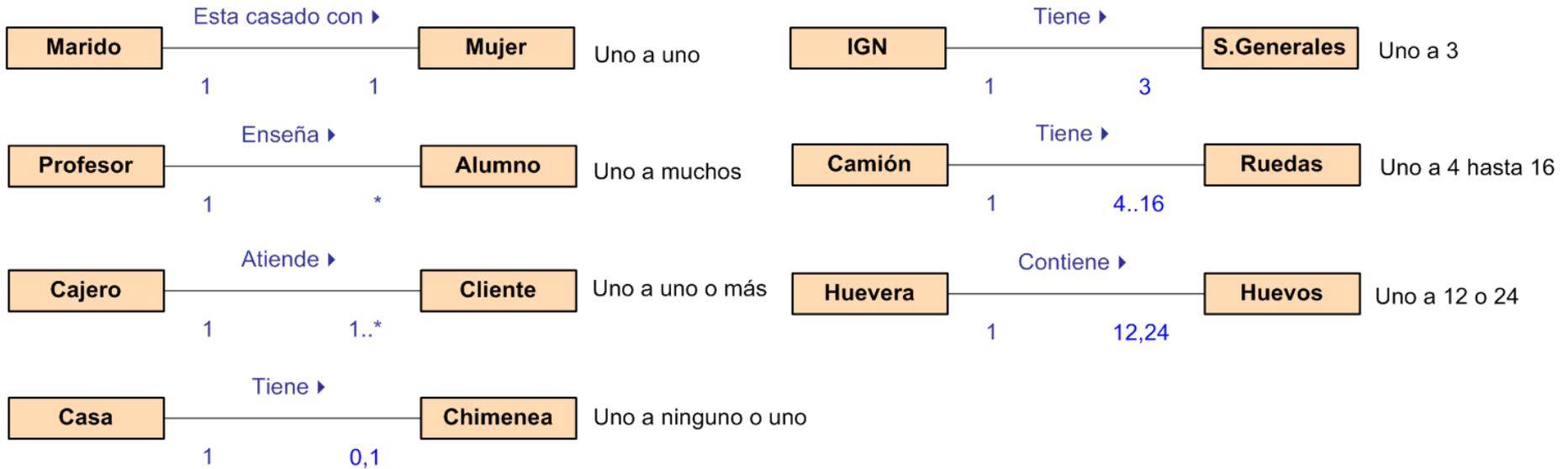
• Relaciones entre las clases

uno-uno
 uno-muchos
 muchos-muchos

Cardinalidad de relaciones
 especifica cuantas instancias de una clase se pueden relacionar a una sola instancia de otra clase.

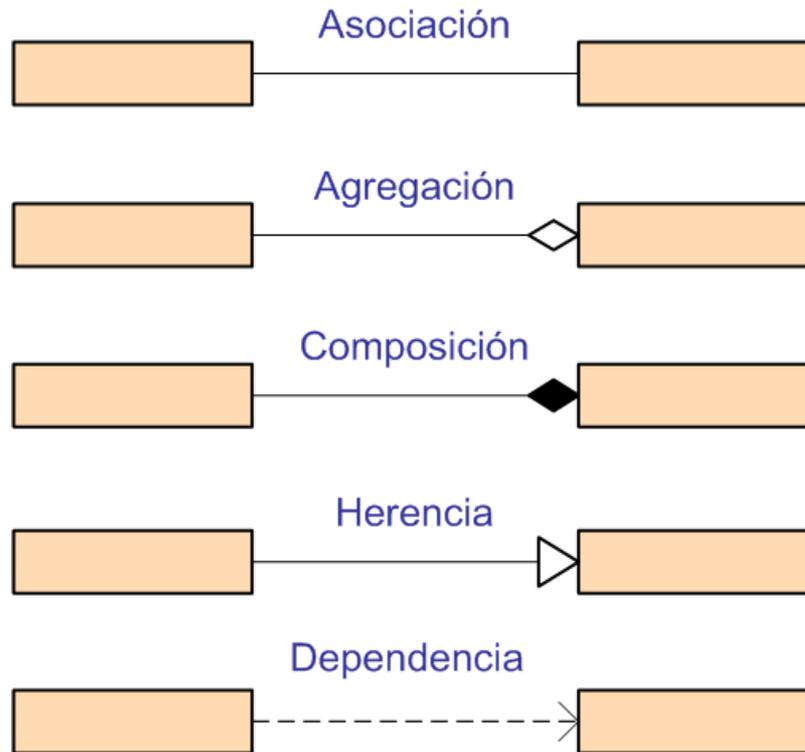


• Relaciones entre las clases



- Los objetos de un sistema se relacionan entre sí y esto se modela a nivel de clase.

Tipos de relaciones

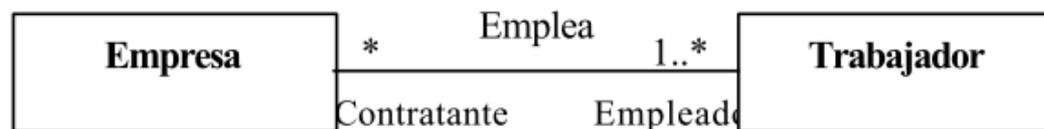


• Asociación

– Una asociación es una abstracción de la relación bidireccional existente entre los objetos



- Cada asociación tiene dos roles:
- indican el papel que asume una clase respecto de la otra
 - puede nombrarse explícitamente:
 - Cada rol, al lado de la clase



- **Asociación:** Multiplicidad o cardinalidad

La multiplicidad indica cuantos objetos de una clase pueden relacionarse con una única instancia de una clase asociada.

- Un número único
 - 1 uno
 - * muchos (cero o muchos)
- Un rango (se separa con dos puntos seguidos)
 - 0..1 cero o uno
 - 1..* uno a muchos (al menos uno)
 - 10..20 diez a veinte
- Una combinación discreta de valores
 - 1,3,4 uno, tres o cuatro

- **Asociación:** Multiplicidad o cardinalidad

- Ejemplos

- En una universidad estudian muchos estudiantes
- Un estudiante estudia en una sola universidad (al menos 1)



- En una universidad estudian entre 100 y 200 estudiantes
- Un estudiante estudia en una o dos universidades

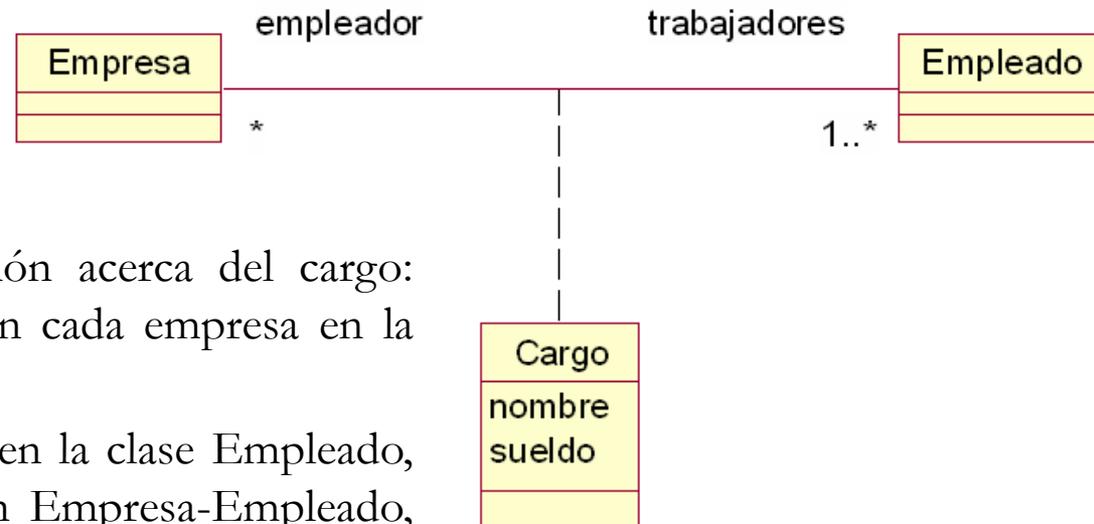


- En una universidad estudian 100, 200 o 300 alumnos
- Un estudiante estudia en cero o muchas universidades



• Clase Asociación

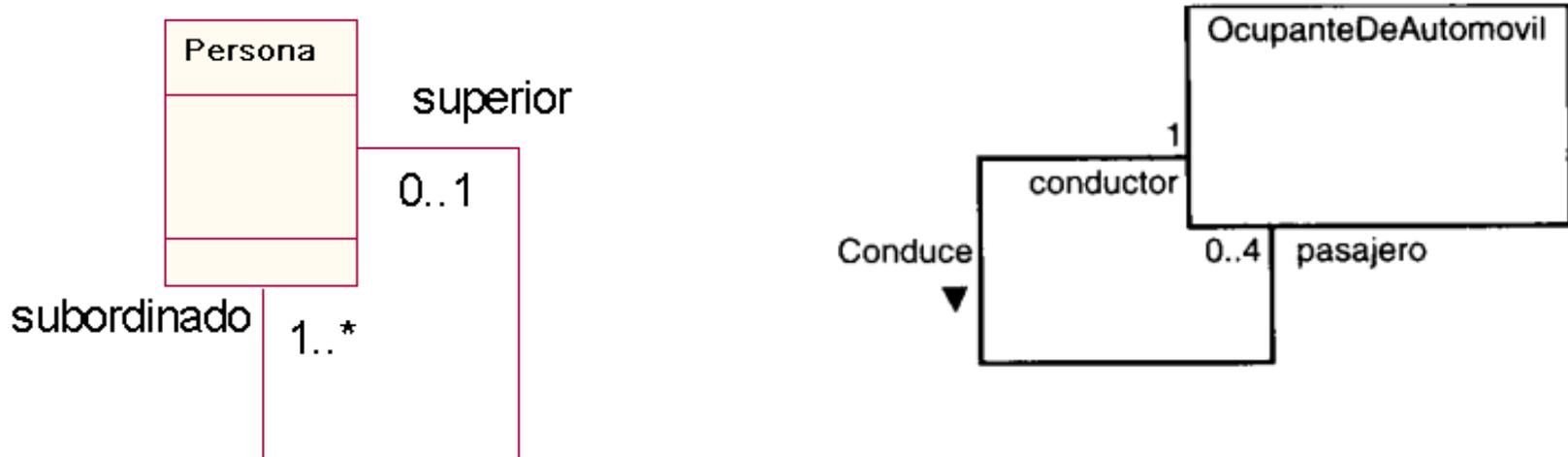
- Es una asociación que tiene también propiedades:
 - atributos, operaciones y otras características de las asociaciones
- Notación: símbolo de clase (rectángulo) unido por una línea de trazos a una línea de asociación
- El nombre de la clase asociación y de la asociación han de ser el mismo y es redundante incluirlo en ambos elementos



- Necesitamos mantener información acerca del cargo: nombre y sueldo del empleado en cada empresa en la que ha trabajado o trabaja
- Podríamos añadir estos atributos en la clase Empleado, pero es un atributo de la relación Empresa-Empleado, puesto que cada empleado tendrá un cargo y sueldo distinto en cada empresa que trabaje

• Tipo de Asociación: Reflexiva

- Relación de una clase consigo misma
- Se presenta cuando una clase tiene objetos que pueden jugar diversos papeles

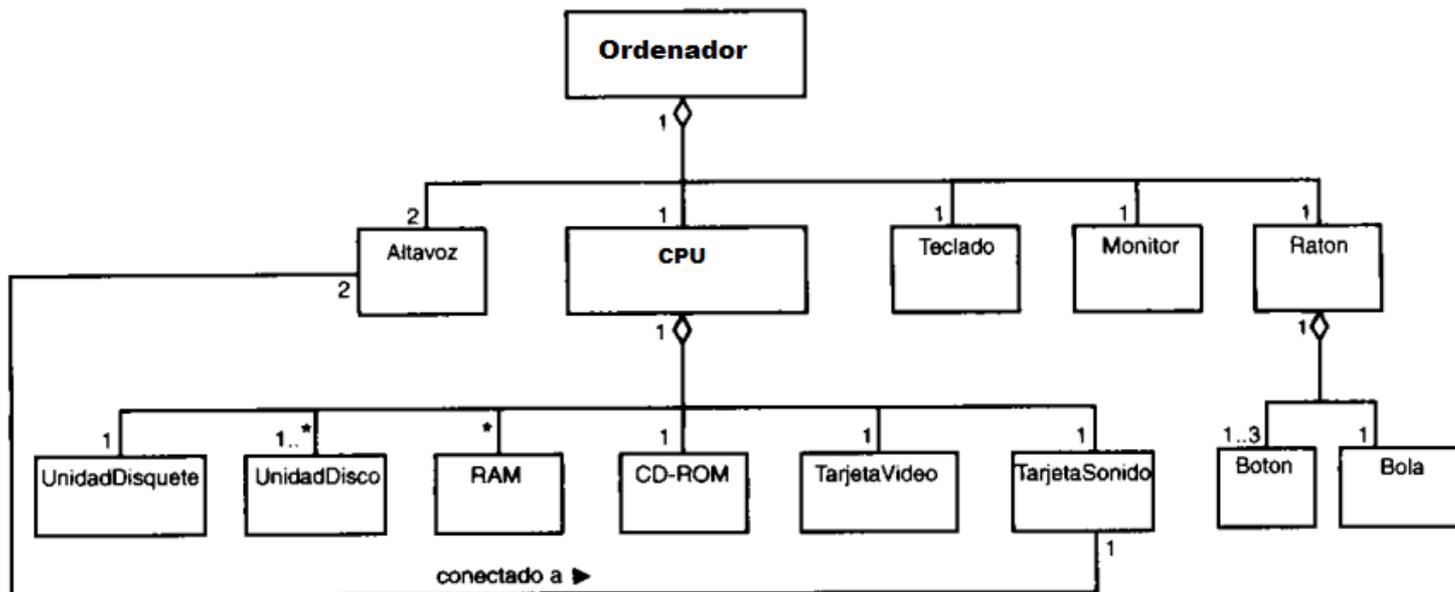


- Una subordinado tendrá cero o un superior
- Un superior tendrá uno o más subordinados
- Superiores y subordinados son personas

• Agregación



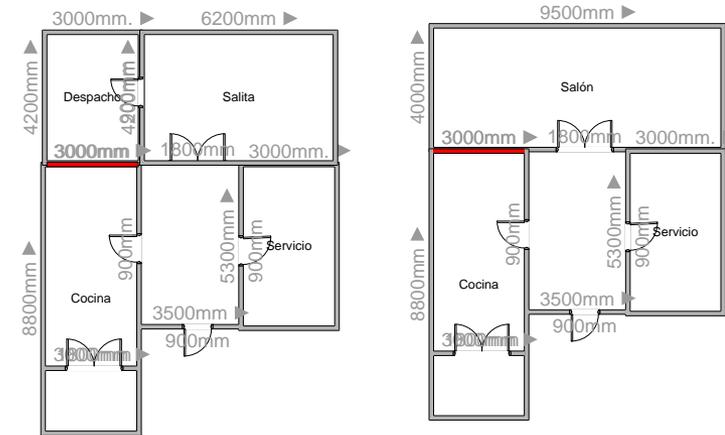
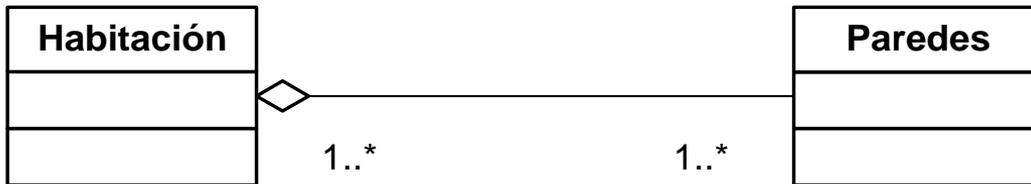
- Una agregación representa la relación “es parte de” entre objetos
- **Notación:** la agregación se indica con un rombo hueco en el extremo de la línea de asociación correspondiente al conjunto o agregado
- Ejemplo: Ordenador – ratón, monitor, CPU, altavoces, auriculares, etc.



• Agregación



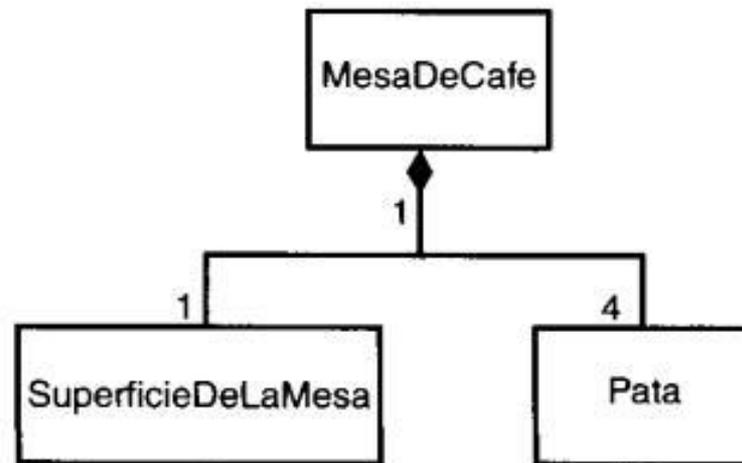
- Una habitación se compone (es un agregado) de una o más paredes.
- Una pared es parte de una o más habitaciones.
- Si desaparece la habitación, las paredes siguen teniendo sentido.



• Composición



- Una composición es un tipo de agregación que trae consigo una estrecha relación entre el agregado y sus componentes
- Los componentes sólo tienen sentido como parte del compuesto
- Cada componente solo puede pertenecer a un todo
 - la multiplicidad del componente es 1
- **Notación:** la agregación se indica con un rombo relleno en el extremo de la línea de asociación correspondiente al conjunto
- Ejemplo: Mesa – tablero, patas

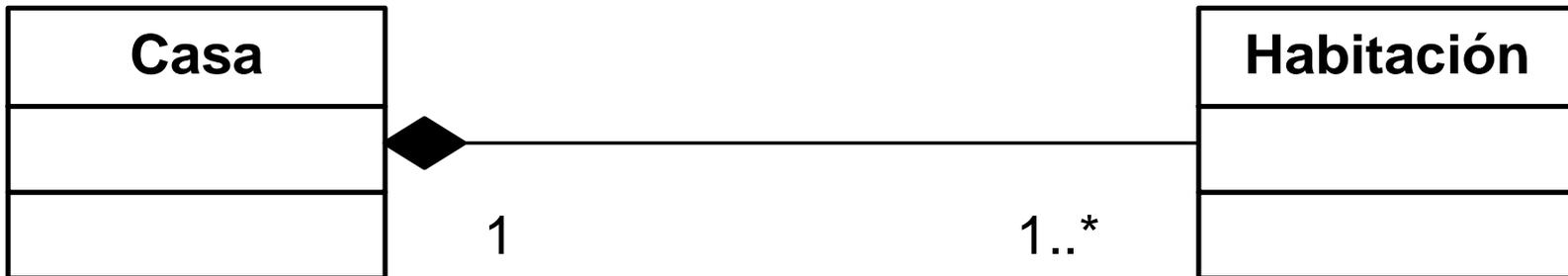


17 de junio de 2015

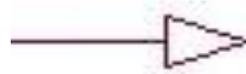
• Composición



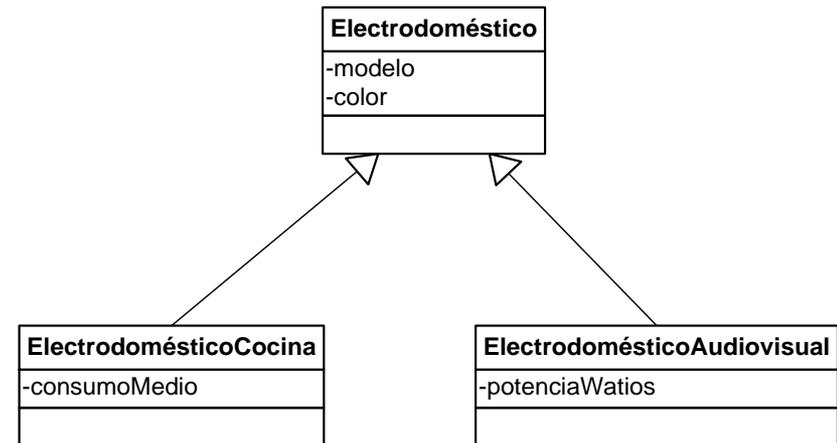
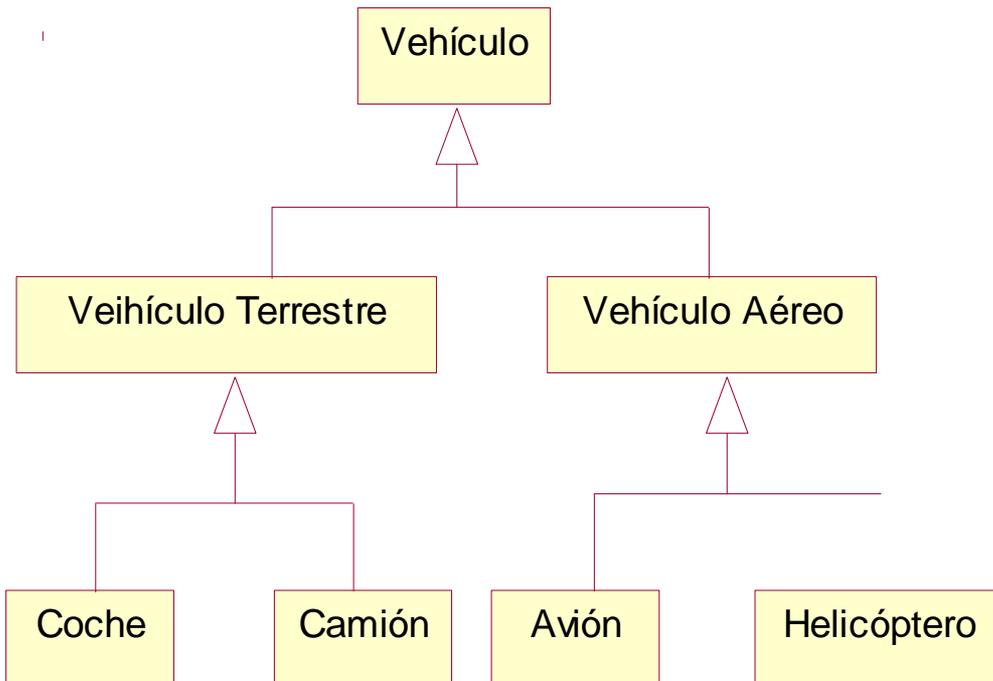
- Una casa se compone de una o más habitaciones.
- Una habitación es parte de una sola casa
- Si desaparece la casa, desaparecen las habitaciones.



• Generalización/Herencia

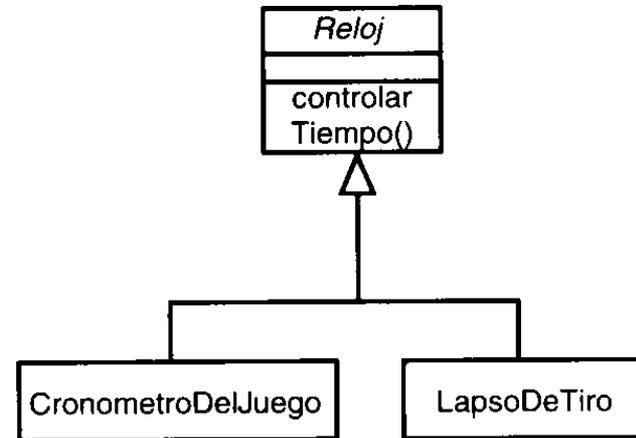
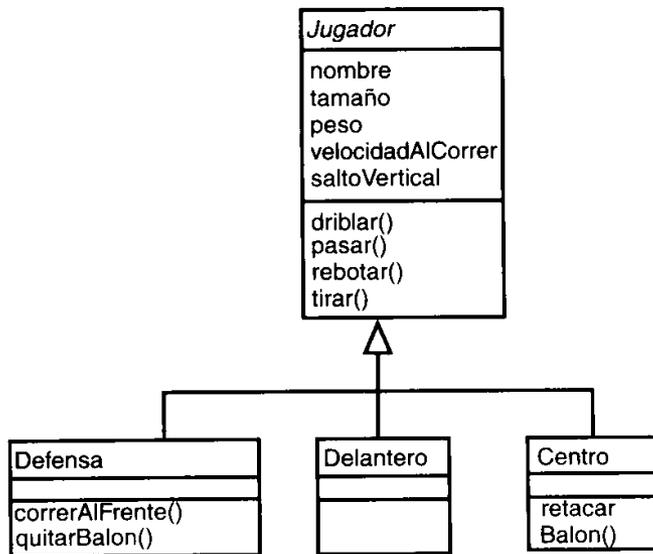


- La herencia es la capacidad de una clase (clase hija) de “heredar” la funcionalidad (atributos y operaciones) de otra clase (clase padre), además de tener las suyas propias
- Otros nombres: superclase - subclase, clase base - clase derivada
- **Notación:** la agregación se indica con una flecha cerrada en el extremo de la línea de asociación correspondiente a la clase padre.



• Clases abstractas

- Es una clase que no puede instanciarse, es decir, no tiene objetos
 - Se implementará a partir de sus clases hijas.
- **Notación:** nombre de la clase en cursiva o <<abstract>>



- Un objeto de la clase *jugador* o *reloj* no sirven para ningún propósito en un modelo determinado.
 - Se desarrollará a partir de las clases hijas: delantero, defensa, centro, etc.

- Clases abstractas

«tipo abstracto» (*abstract type*), un tipo que no se puede instanciar, pero que puede tener atributos y roles de asociación.



- Los estereotipos extienden la semántica, pero no la estructura, de los tipos y clases pre-existentes
- **Notación:** los estereotipos se representan incluyendo el texto correspondiente a su tipo entre comillas.
- Estereotipos de UML habituales:
 - **<<dataType>>**: es un descriptor de un conjunto de valores que carecen de identidad.
 - **<<enumeration>>**: tipo de dato cuyas instancias forman una lista de valores de nombres literales. (ejemplo: Boolean – TRUE, FALSE)
 - **<<codeList>>**: una enumeración extendible
 - **<<union>>**

Estereotipos INSPIRE:

- **<<FeatureType>>**: para clases de objeto espacial.
- **<<Voidable>>**: se emplea en aquellos casos en los que el valor de una propiedad (atributo) no está presente en el conjunto de datos espaciales, pero sí existe en el mundo real.
- **<<Placeholder>>**: clase que se definirá en un futuro

<<enumeration>>

- tipo de datos cuyas instancias son una lista de valores literales
- es una **lista cerrada**
- si se desea ampliar usar <<codelist>>
- ejemplo: enumeración de los días de la semana

«enumeración» (enumeration), un tipo de datos cuyas instancias forman una lista fija de valores literales denominados. Los atributos de un tipo enumerado solo pueden tomar valores de esa lista.

«enumeration»
LegalStatusValue

agreed
notAgreed

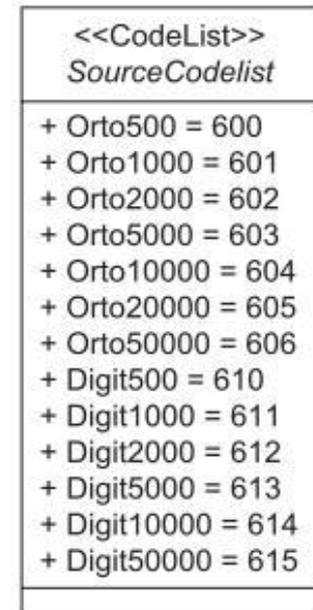
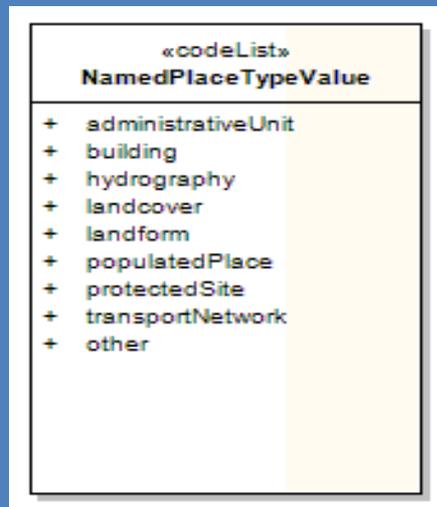
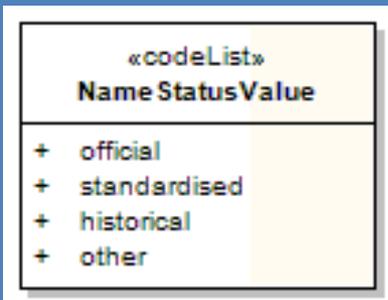
«enumeratio...»
OriginValue

natural
manMade

<<codeList>>

- puede utilizarse para describir una enumeración **más abierta**.
- es útil para expresar una larga lista de valores potenciales.
- Si se conocen completamente los elementos de la lista, debe utilizarse una enumeración, si sólo se conocen los probables elementos de la lista, debe utilizarse una lista de códigos.

«lista controlada» (*code list*), una enumeración abierta que puede ampliarse



<<codeList>> <<enumeration>>

- (a) listas controladas que se gestionan en un registro común de listas controladas y no deben ser ampliadas por los Estados miembros;
- (a) listas controladas que pueden ser ampliadas por los Estados miembros. Deben estar disponibles en un registro.



<http://inspire.ec.europa.eu/registry/>

- **Parámetro de sustitución y tipos candidato**
 - Un esquema de aplicación puede incluir tipos que se especificarán como partes de otros temas, pero que se usan como tipos de atributos o roles de asociación. Se distinguen dos tipos:
 - Placeholder o Parámetros de sustitución
 - para un tipo de objeto espacial para el que sólo se especifica una definición (basada en los requisitos de los temas del Anexo I). Recibe el estereotipo <<placeholder>>

Type	Package	Stereotypes	Section
Crossing	Hydro - Physical Waters	«featureType»	5.3.2.1.1
CrossingTypeValue	Hydro - Physical Waters	«codeList»	5.3.2.3.2

El tipo GlacierSnowfield se detallara en la DS Land Cover

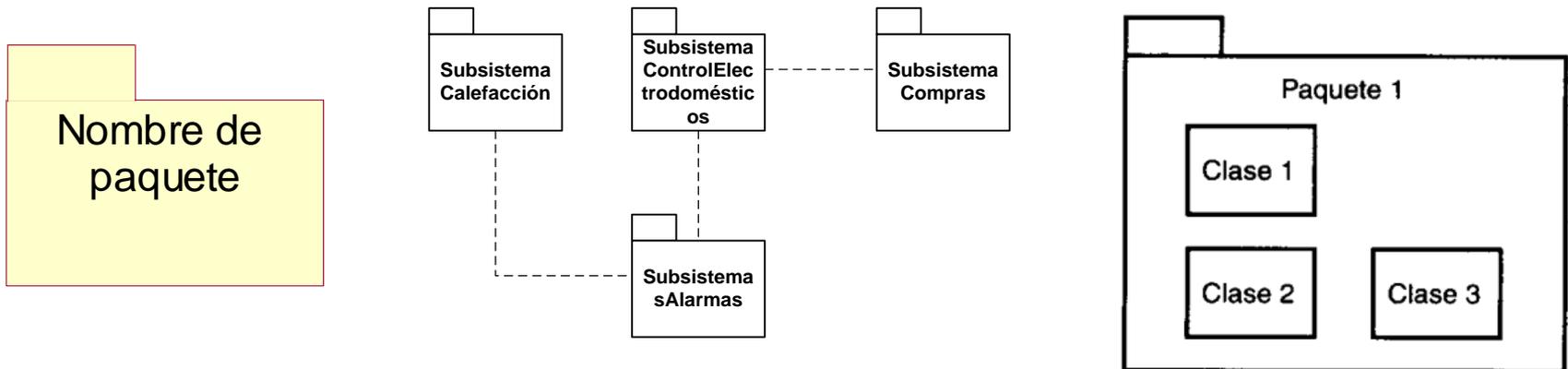
Embankment	Natural Risk Zones	«featureType»	5.3.2.4.1
Falls	Hydro - Physical Waters	«featureType»	5.3.2.1.4
FluvialPoint	Hydro - Physical Waters	«featureType»	5.3.2.1.5
Ford	Hydro - Physical Waters	«featureType»	5.3.2.1.6

GlacierSnowfield	Land Cover	«placeholder,featureType»
HydrologicalPersistenceValue	Hydro - Physical Waters	«codeList»
HydroOrderCode	Hydro - Physical Waters	«dataType»
HydroPointOfInterest	Hydro - Physical Waters	«featureType»
HydroPowerPlant	Energy Resources	«featureType»
InundatedLand	Natural Risk Zones	«featureType»
InundationValue	Natural Risk Zones	«codeList»
LandWaterBoundary	Hydro - Physical Waters	«featureType»
Lock	Hydro - Physical Waters	«featureType»

<i>Hydro Object</i>
«placeholder,featureType» Land Cover::GlacierSnowfield
+ geometry: GM_Surface + inspireId: Identifier
«voidable, lifeCycleInfo»
+ beginLifespanVersion: DateTime + endLifespanVersion: DateTime [0..1]

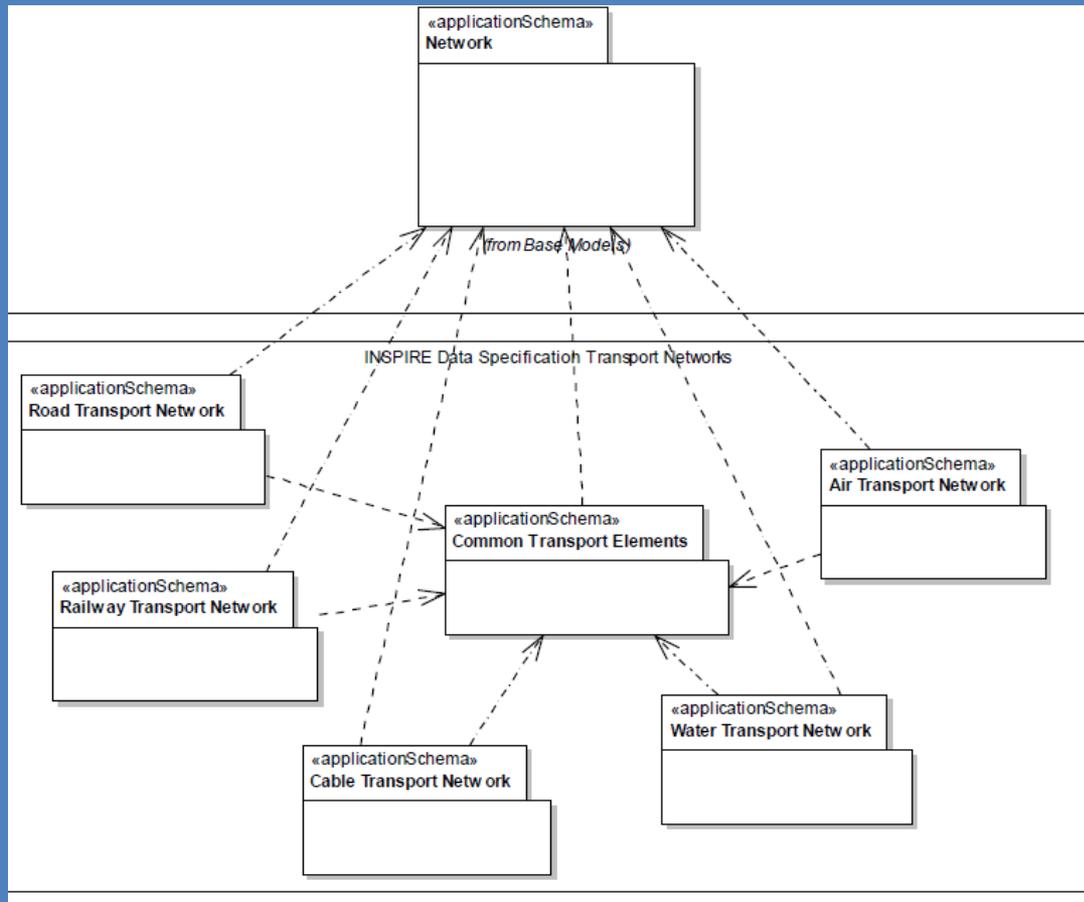
Paquete

- Un **paquete** es un elemento del diagrama que sirve para agrupar otros elementos de un subsistema particular
- **Notación:** se representa con una carpeta
- Los paquetes se relacionan entre sí
- Todas las clases no son necesariamente visibles desde el exterior del paquete, es decir, un paquete encapsula a la vez que agrupa



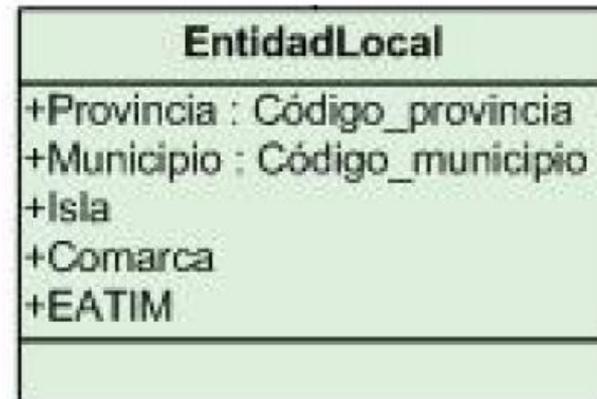
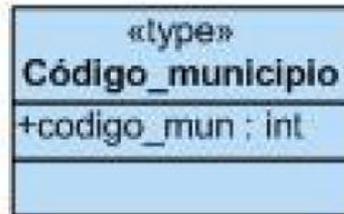
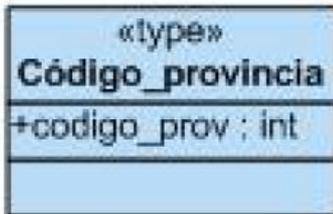
Paquete

«paquete» (*package*), un mecanismo de propósito general para organizar elementos en grupos,



Tipo de datos

- Un lenguaje nos da unos tipos de datos básicos
 - Char, string, boolean, integer, real....
- Se pueden definir tipos de datos de usuario



Tipos de Datos Comunes

- *Artículo 3* . Tipos comunes:
 - tipos que sean comunes a varios de los temas enumerados en los anexos I, II y III se ajustarán a las definiciones y restricciones e incluirán los atributos y los roles de asociación establecidos en el anexo I.

1. Para los tipos

- Area, Boolean, CharacterString, DateTime, Distance, Integer, Length, Measure, Number, Sign y Velocity : **ISO 19103.**

2. Para los tipos

- GM_Curve, GM_MultiSurface, GM_Object, GM_Point, GM_Primitive y GM_Surface : **EN ISO 19107.**

3. Para el tipo

- TM_Period: **EN ISO 19108.**

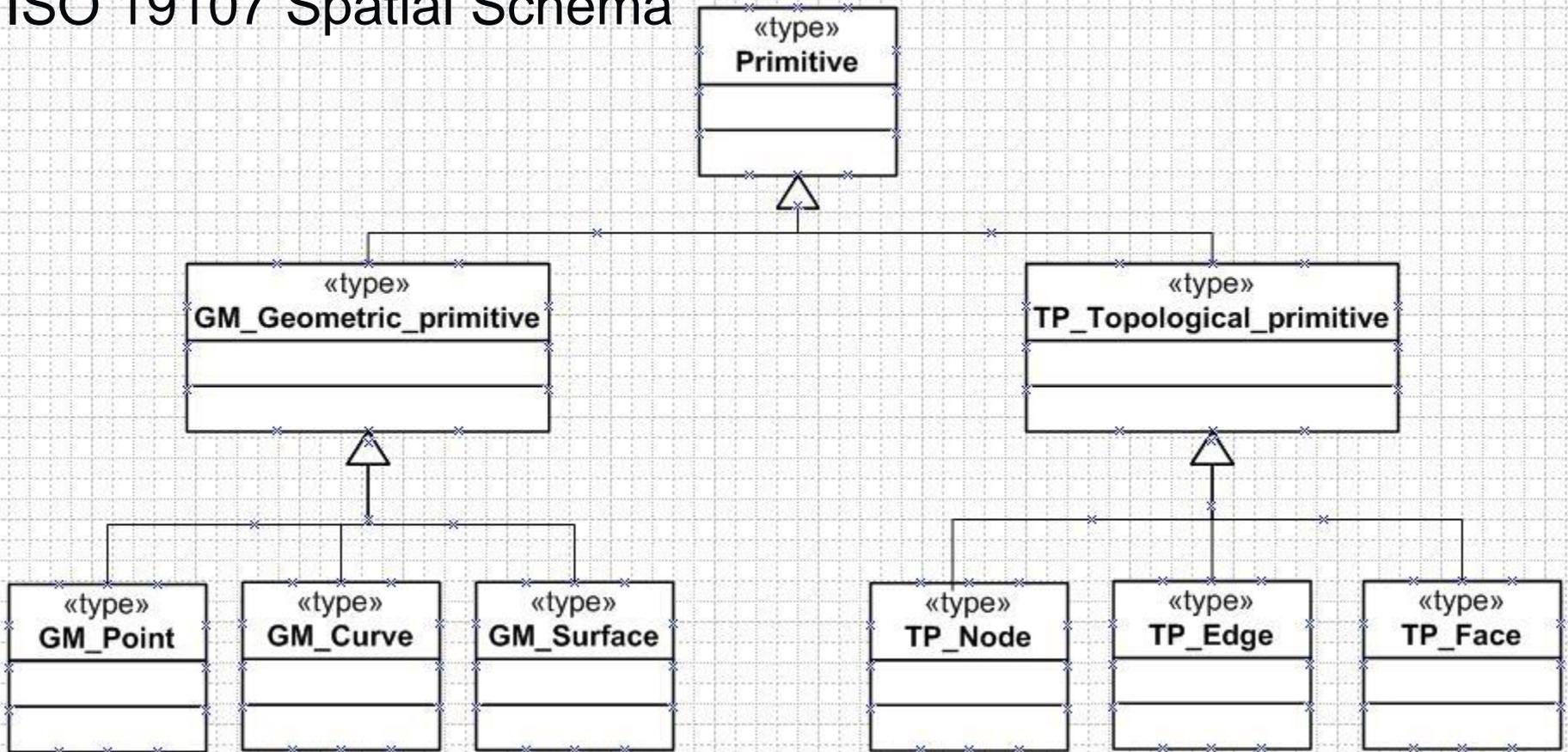
4. Para los tipos

- CI_Citation y MD_Resolution: **EN ISO 19115.**

5. Para los tipos

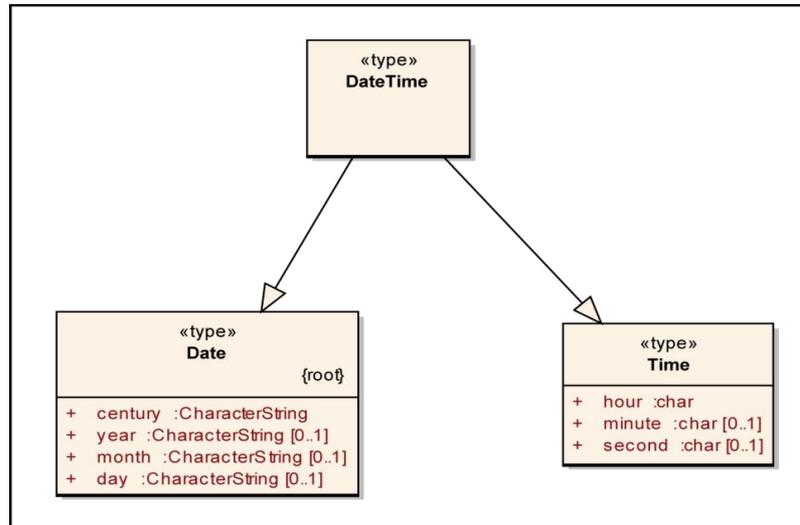
- LocalisedCharacterString y URI: **ISO 19139.**

ISO 19107 Spatial Schema



ISO 19103:2005 Conceptual schema language

- Atributo para fijar la fecha y la hora de un evento.



DateTime hereda los atributos de Date y de Time, que se pueden utilizar también de forma independiente. Sin embargo, en la implementación de la fecha y la hora se seguirán las especificaciones de la norma ISO 8601. Date y Time se representan mediante sendas cadenas de caracteres con diferentes formatos de los que se han seleccionado los siguientes:

Date: AAAA-MM-DD (formato completo extendido)

Time: hh:mm:ss (formato completo extendido)

17 de junio de 2015

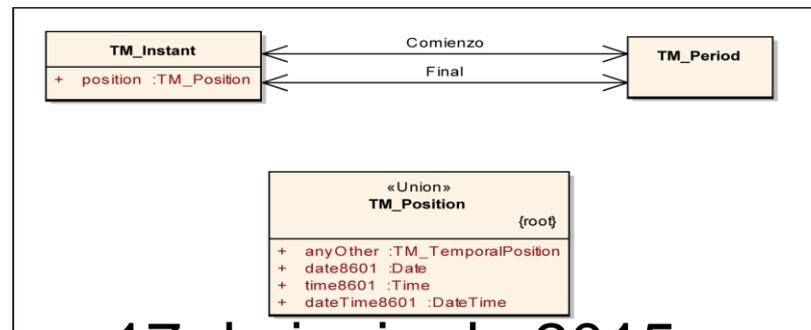
ISO 19108:2002 Geographic Information – Temporal Schema

- Atributos de tiempo para consignar periodos e instantes.

TM_Instant es un elemento temporal primario adimensional que representa una posición en una escala de tiempo. Es equivalente a un punto en el espacio cartesiano.

En la práctica un instante es un intervalo cuya duración es inferior a la resolución de la escala de tiempo.

TM_Period es un elemento temporal primario con el que se representa una extensión de tiempo. Es equivalente a una curva en el espacio cartesiano. Es un intervalo que está limitado por dos extremos a modo de instantes que describen además la posición del periodo es una escala de tiempo. La duración de un periodo es la distancia temporal entre sus dos instantes extremos.



ISO 19108:2002 Geographic Information – Temporal Schema

- Atributos de tiempo para consignar periodos e instantes.

Atributo	Descripción	Observaciones
position	Posición en una escala temporal del instante (TM_Instant). Su expresión es uno y solo uno de los elementos del estereotipo <<Union>> denominado TM_Position.	

Valores de TM_Position: **TM_TemporalPosition** (anyOther): utilizado cuando se emplea una escala temporal diferente del calendario greogoriano y del tiempo medio de Greenwich. Se compone de cuatro subtipos (TM_Coordinate, TM_OrdinalPosition, TM_CalDate y TM_ClockTime). En principio no se hará uso de este elemento. (apartado 5.4.3 de la Norma ISO 19108:2002)

Tipos de Datos Comunes

```

«featureType»
NamedPlace

+ inspireId: A
+ name: GeographicalName [1..*]
+ geometry: B

«voidable»
+ type: C
+ localType: LocalisedCharacterString [1..*]
+ relatedSpatialObject: Identifier [0..*]
+ leastDetailedViewingScale:
+ mostDetailedViewingScale: D

«voidable, lifeCycleInfo»
+ beginLifespanVersion: E
+ endLifespanVersion: DateTime [0..1]
    
```

```

«codeList»
NamedPlaceTypeValue

+ administrativeUnit
+ building
+ hydrography
+ landcover
+ landform
+ populatedPlace
+ protectedSite
+ transportNetwork
+ other
    
```

GM_Object Identifier Int [0..*]

DateTime NamedPlaceTypeValue

Tipos de Datos Comunes

«featureType»
NamedPlace

- + inspireId: Identifier
- + name: GeographicalName [1..*]
- + geometry: GM_Object

«voidable»

- + type: NamedPlaceTypeValue [1..*]
- + localType: LocalisedCharacterString [1..*]
- + relatedSpatialObject: Identifier [0..*]
- + leastDetailedViewingScale: int [0..1]
- + mostDetailedViewingScale: int [0..1]

«voidable, lifeCycleInfo»

- + beginLifespanVersion: DateTime
- + endLifespanVersion: DateTime [0..1]

Tipos de Datos Comunes

- **Identificador (Identifier):** Identificador único externo de objeto publicado por el organismo responsable, que pueden utilizar las aplicaciones externas para referenciar al objeto espacial: localId, namespace, versionId

Atributos del tipo de dato Identifier

Atributo	Definición	Tipo	Voidability
localId	Identificador local, asignado por el proveedor de datos. El identificador local es único dentro del espacio de nombres, es decir que ningún otro objeto espacial tiene el mismo identificador único.	CharacterString	
namespace	Espacio de nombres que identifica de manera única la fuente de datos del objeto espacial.	CharacterString	
versionId	Identificador de la versión particular del objeto espacial, con una longitud máxima de 25 caracteres. Si la especificación de un tipo de objeto espacial con un identificador externo de objeto incluye información sobre el ciclo de vida, el identificador de la versión se utiliza para distinguir entre las diferentes versiones de un objeto espacial. Dentro del conjunto de todas las versiones de un objeto espacial, el identificador de la versión es único.	CharacterString	voidable



Ejercicios UML

17 de junio de 2015

- Lo primero: Definir el Universo de discurso
- El subconjunto abstraído del mundo real que tiene todo lo que resulta de interés
- El contexto
 - Ayuda a fijar los límites del modelo y a no echar a volar la imaginación (¿no quedaría mejor si añadimos...?)

1) Definir el contexto

Universo de discurso: parte y visión del mundo que interesa

2) Resumirlo

3) Definir los requerimientos de usuario

Las preguntas que va a responder el sistema

4) Sustantivos objetos

5) Verbos relaciones

6) Adjetivos atributos

7) Simplificar

8) La semántica es importante

Ejemplo 1

17 de junio de 2015

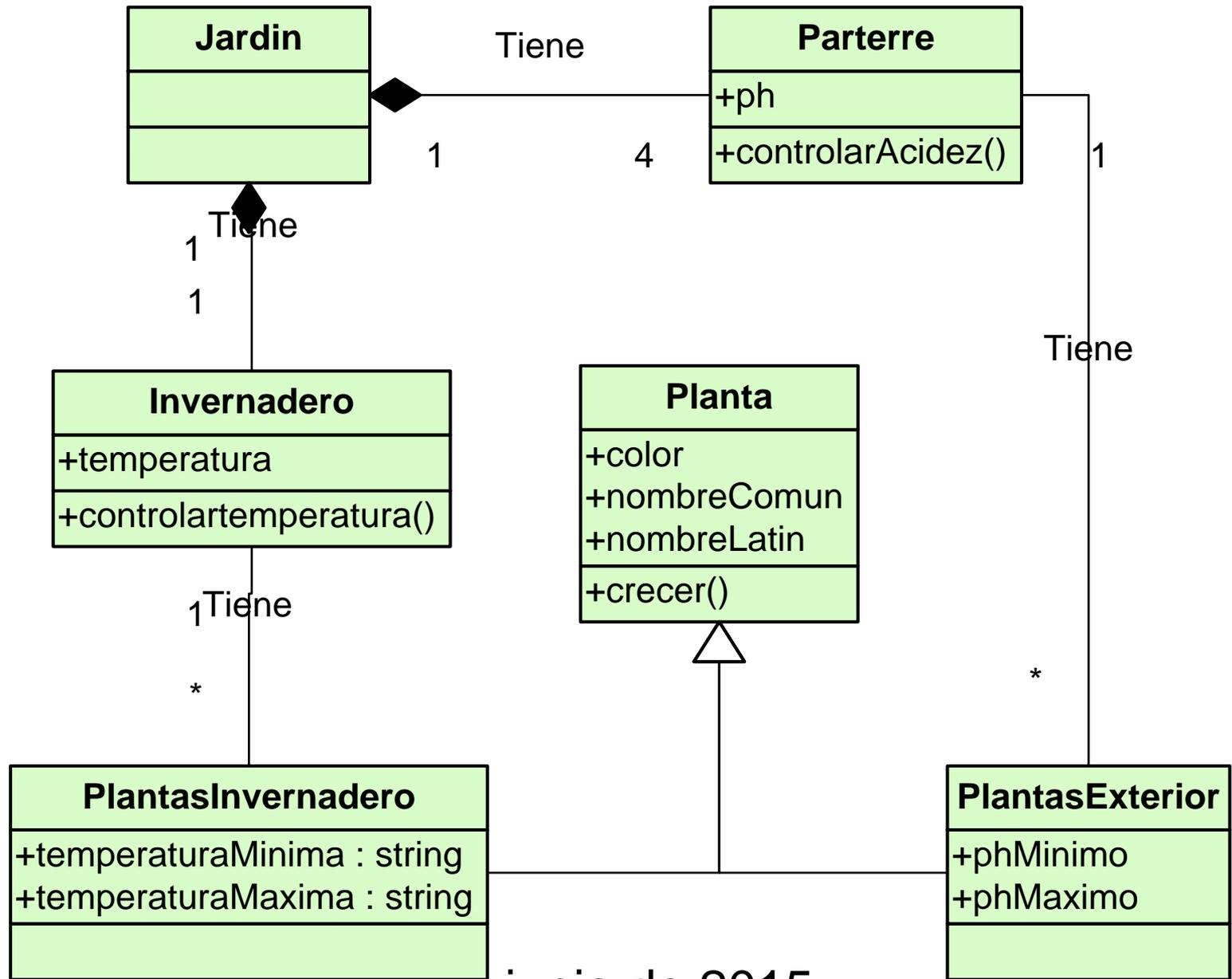
- Juan es un jardinero muy cuidadoso que tiene un jardín con un invernadero y cuatro parterres exteriores, dónde cultiva plantas con flores de colores
- Mantiene la temperatura del invernadero casi sin fluctuaciones, para que las plantas de interior que cultiva allí sobrevivan sin problemas
- También mantiene una acidez diferente en el suelo de cada parterre para poder cultivar plantas de exterior de todo varios tipos

	Jardin	
Parterre		Invernadero
PlantadeExterior		PlantadelInterior
	Planta	

17 de junio de 2015

- Tiene etiquetas para nombrar a cada planta con el nombre común y el nombre científico.
- El nombre latino de cada planta es único
- El ph de cada parterre está entre el ph mínimo y máximo de las plantas que contiene
- La temperatura del invernadero está entre las temperaturas mínima y máxima de cada planta que contiene
- Toda planta o bien es de invernadero o bien es de exterior

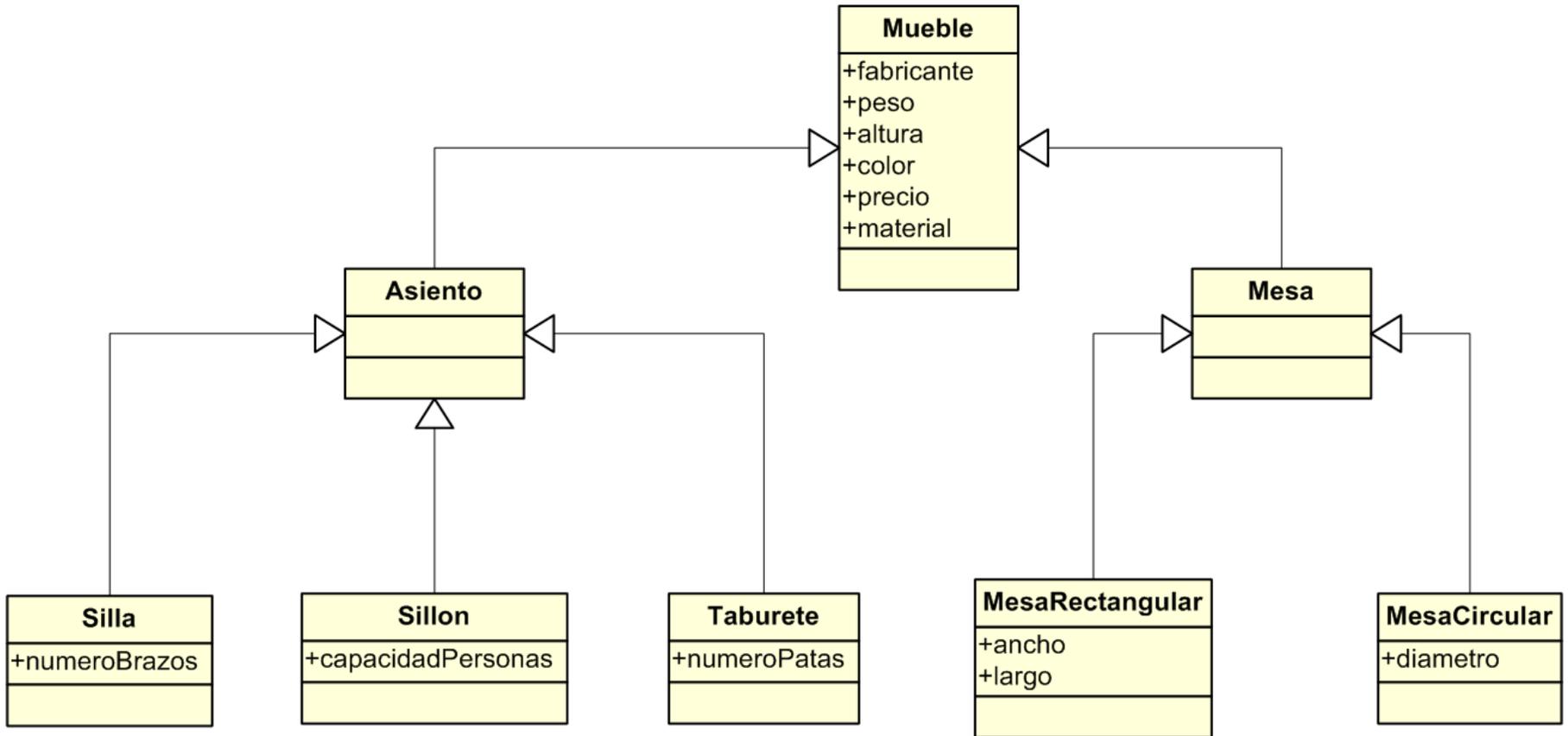
17 de junio de 2015



Ejemplo 2

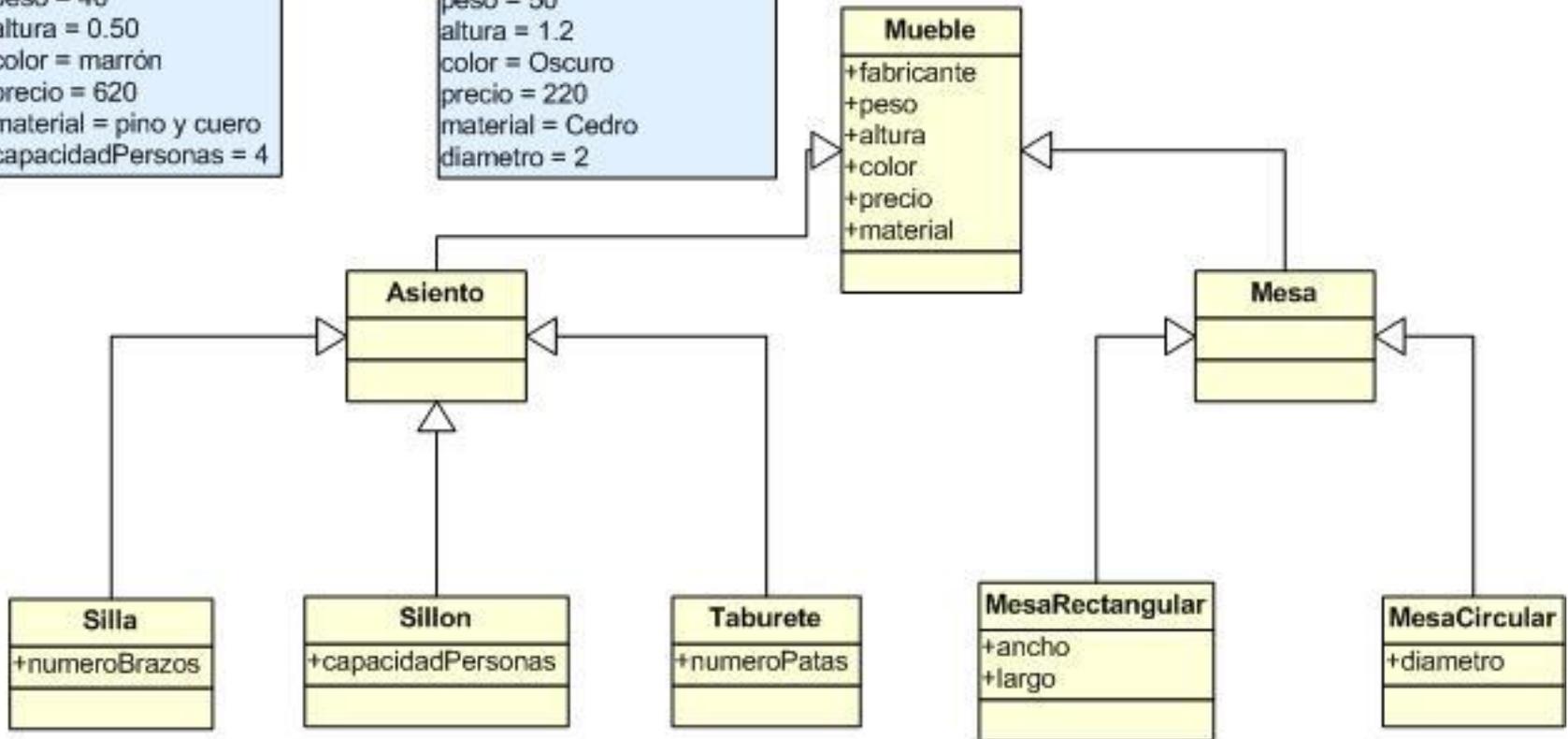
17 de junio de 2015

- En una oficina se ha realizado una compra de mobiliario clasificado en asientos y mesas de trabajo, concretamente: sillas, sillones, taburetes, mesas rectangulares y mesas circulares.
- Cada uno de los muebles tendrá propiedades particulares como el número de brazos, la capacidad de personas, el diámetro, ancho y largo de las mesas, nº de patas del taburete y propiedades heredadas como el fabricante,, peso, altura, color, material y precio.



Living blue : Sillon
 fabricante = Mundogar
 peso = 40
 altura = 0.50
 color = marrón
 precio = 620
 material = pino y cuero
 capacidadPersonas = 4

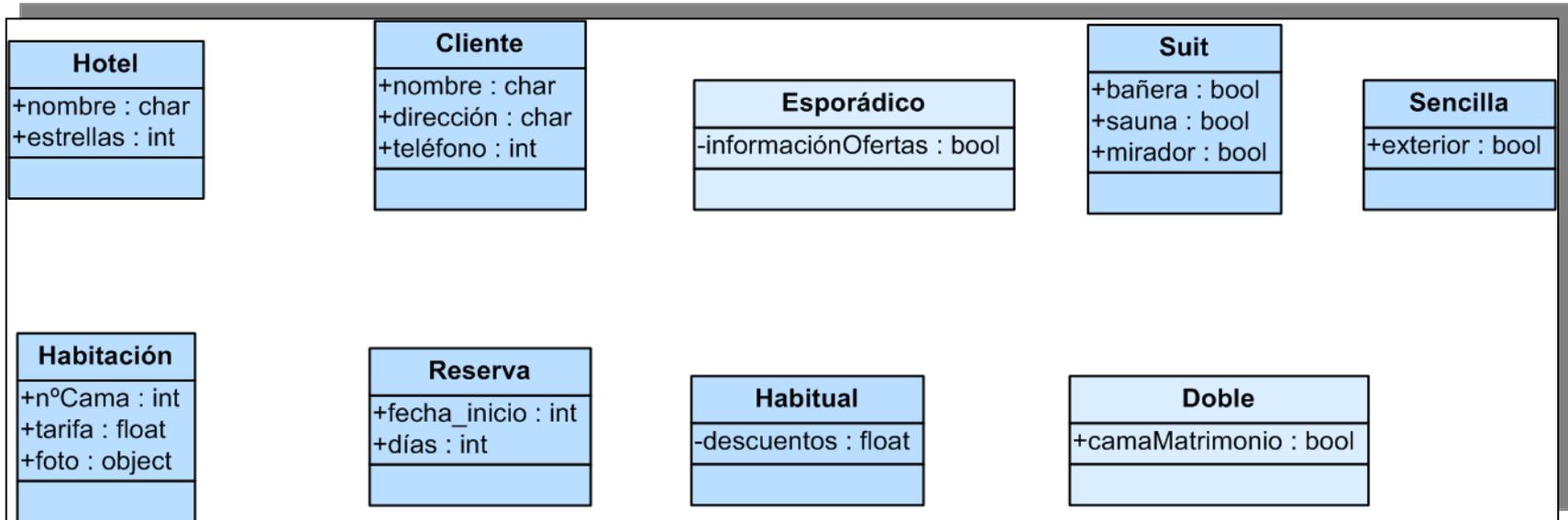
Room moon : MesaCircular
 fabricante = IKEA
 peso = 50
 altura = 1.2
 color = Oscuro
 precio = 220
 material = Cedro
 diametro = 2

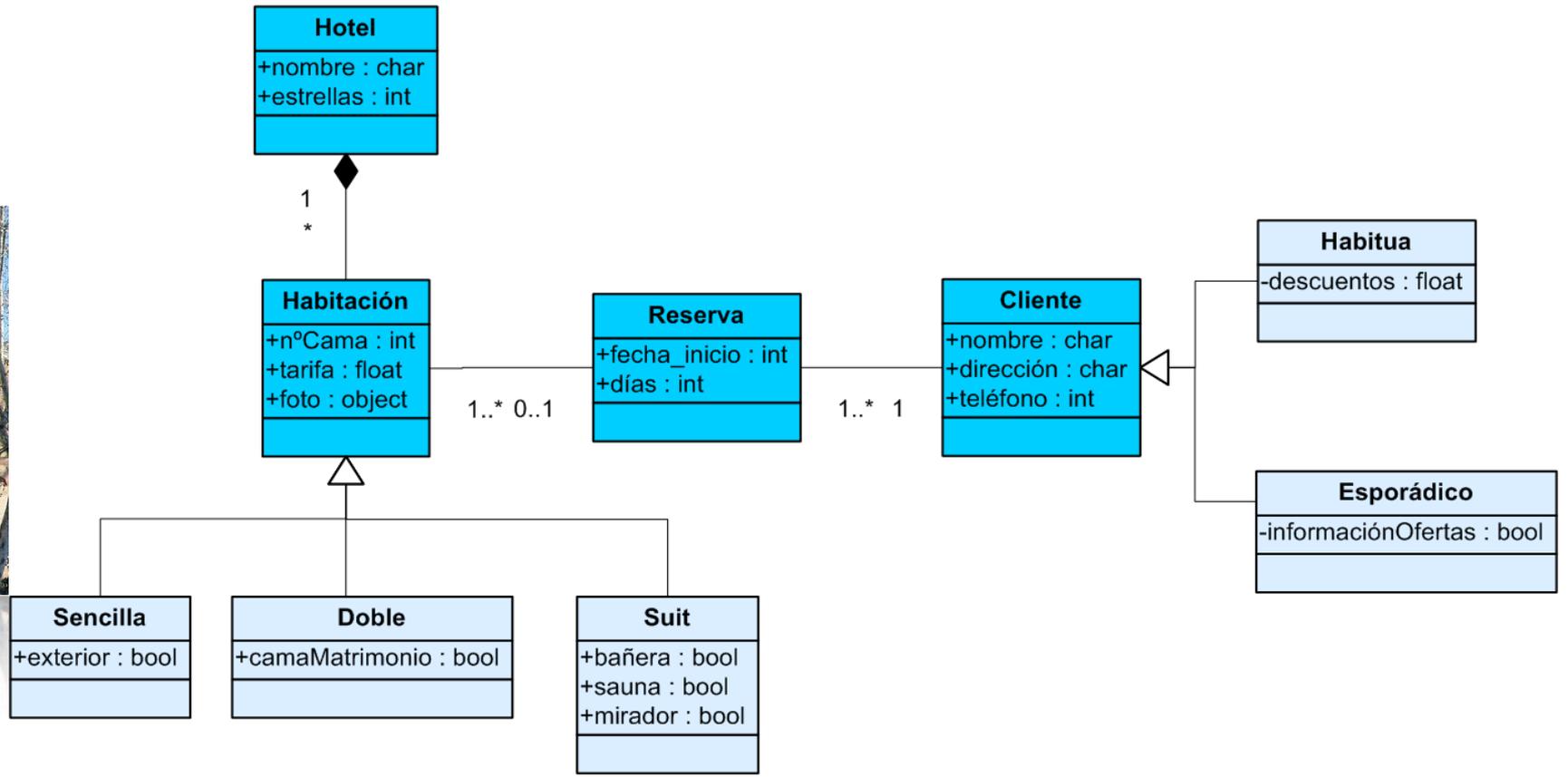


Ejemplo 3

17 de junio de 2015

- El hotel del que vamos a realizar el diagrama tiene un nombre y un número de estrellas. Posee tres tipos de habitaciones: sencilla, doble y suit y dos tipos de clientes: habituales y esporádicos. Cada reserva almacena: la fecha de entrada y el número de días que será ocupada la habitación.
- Al cliente se le toman los siguientes datos: el nombre, la dirección y su teléfono.
- Las habitaciones tienen distinto precio y distinto número, y además es posible mostrar una foto de cada una de ellas al cliente.
- En la habitación sencilla el cliente puede elegir si es exterior o no, en la habitación doble si la cama es de matrimonio o sencilla y en la suit la posibilidad de bañera, sauna y mirador.
- Los clientes que son habituales reciben un trato especial, es decir, una reducción en el precio de la habitación con respecto al cliente esporádico.
- A los esporádicos el hotel les ofrece la posibilidad de darles información sobre las ofertas.





Gracias por vuestra colaboración

Paloma Abad, Alejandra Sánchez

pabad@fomento.es,

asmaganto@fomento.es

17 de junio de 2015