

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/341679692>

Modelos y Metamodelos – Lecturas para Arquitectos de Negocio

Book · February 2021

CITATIONS

0

READS

20,666

1 author:



[Jorge Villalobos](#)

Los Andes University (Colombia)

89 PUBLICATIONS 555 CITATIONS

SEE PROFILE

The background of the slide is a detailed architectural floor plan of a building. It features various rooms, corridors, and circular spaces. Some circular areas are filled with a radial pattern, suggesting a specific design or structural element. The drawing is a technical line drawing in black and white, typical of architectural blueprints.

Lecturas para Arquitectos de Negocio (No. 4)

Modelos y Metamodelos

Jorge Villalobos, PhD

jvillalo@uniandes.edu.co

2021

Modelos y Metamodelos

Lecturas para Arquitectos de Negocio

ISBN: 978-958-49-2394-3

No. 4 – Versión 1.1 – febrero/2021

Contenido

1. ¿Qué es un modelo?
2. ¿Qué es un metamodelo?
3. Construcción de metamodelos
4. Diseño de lenguajes de dominio
5. Puntos de vista y vistas
6. Modelos en un contexto empresarial
7. Ejemplos completos de conceptualización
8. Hojas de trabajo

Autor

Jorge Villalobos. PhD en Informática (Université Joseph Fourier de Grenoble), Máster en Informática (Institut National Polytechnique de Grenoble – INPG), Profesor titular del Departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes, Investigador Senior, Consultor, Autor de múltiples artículos científicos y libros y Miembro Independiente de varias Juntas Directivas.

<https://www.linkedin.com/in/jorge-a-villalobos/>



Introducción

El tema de los modelos es algo que ha llamado la atención de la humanidad desde hace muchos siglos. Se ha abordado desde muchas perspectivas, porque está relacionado con la capacidad humana de construir abstracciones de la realidad y razonar sobre ellas, una habilidad fundamental en nuestra vida diaria.

En esta lectura vamos a partir desde una visión general de los modelos, para rápidamente concentrarnos en los modelos empresariales. Vamos a dedicar un buen espacio a los modelos conceptuales que allí encontramos, que son de particular interés para los arquitectos de negocio, ya que conceptualizar un dominio (crear el modelo conceptual de un dominio) es una de nuestras herramientas fundamentales de trabajo.

No olvidemos que los arquitectos nos movemos sobre abstracciones de los distintos elementos que encontramos en una empresa y que esas abstracciones las expresamos como modelos (modelo de negocio, modelo estratégico, modelo de aplicaciones, entre otros). Y, usando estos modelos, analizamos la empresa, diagnosticamos una situación problemática, proponemos y validamos alternativas, y finalmente diseñamos una solución o una aproximación a un problema. Todo esto a alto nivel, todo esto de manera abstracta, y todo claramente relacionado con esa realidad que queremos transformar.

Como la construcción de modelos es algo que requiere práctica, esta lectura tiene muchos ejemplos, guías metodológicas y ejercicios para trabajar, de manera que a la par con la teoría se puedan ir desarrollando las habilidades que se necesitan.

1. ¿Qué es un modelo?

Con el fin de introducir el vocabulario que necesitamos, vamos a partir del proceso básico que se sigue para crear un modelo (ver figura 1), el cual iremos extendiendo y profundizando a lo largo de esta primera sección.

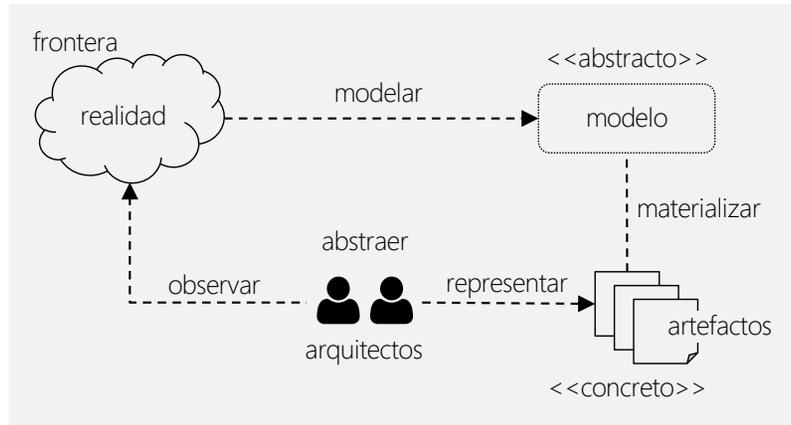


Fig. 1 – Visión global del proceso de crear un modelo

- > Aquello que se quiere modelar (llamado “realidad” en la figura 1) debe tener una frontera bien definida (debe ser claro qué está por dentro y qué está por fuera de esa frontera). En nuestro caso, dentro de esa realidad estará la empresa y posiblemente algunos elementos de su entorno (clientes, proveedores o competidores).
- > Un modelo es algo abstracto, inmaterial, que solo existe en la cabeza de las personas. Un modelo se representa (expresa) a través de un conjunto de artefactos, que corresponden a elementos físicos (diagramas, dibujos o textos) que describen un aspecto o una parte de dicho modelo.
- > El trabajo de los arquitectos es observar y entender esa realidad, abstraer lo que allí encuentran para crear un modelo (o parte de él) en sus cabezas y expresar dicho modelo usando un conjunto de artefactos.

Al ver el proceso anterior, se hacen evidentes tres riesgos: (1) que la información que extraigamos de la realidad sea inadecuada o falsa, (2) que el proceso de abstracción que realicemos sea incorrecto o (3) que elaboremos mal los artefactos (que sean ambiguos, por ejemplo). Nada fácil va a ser, entonces, asegurar que un modelo sea correcto. Veamos en detalle cada una de esas etapas.

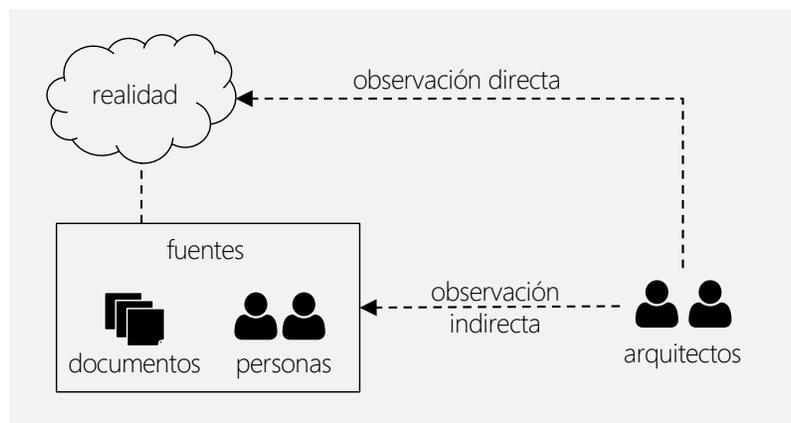


Fig. 2 – Etapa de recolección de información

- > Para obtener información de la realidad, podemos observar directamente su estructura y funcionamiento (figura 2), aunque eso en el caso de una empresa tiene una aplicación muy limitada, por cuanto existen muchos elementos que no se pueden observar directamente. ¿Cómo observar, por ejemplo, la estrategia?
- > El medio más común es obtener la información de fuentes secundarias, como documentos (manuales, actas, procesos, presentaciones, etc.) o la información que tienen algunas personas que conocen esa realidad. En el caso de la empresa, esas personas son los dueños, los empleados o los clientes de la misma. A través de entrevistas y talleres, esas personas nos van a contar cómo es esa realidad, sus partes, su funcionamiento y sus problemas.

El proceso de crear un modelo

Esta primera etapa de levantamiento y estructuración de información es una actividad que se desarrolla al inicio de la mayoría de los proyectos de arquitectura y puede tomar una cantidad considerable de tiempo dada su dificultad. Es usual que la información esté incompleta o que sea inconsistente (cada fuente nos dice algo distinto) o, en algunos casos, puede venir contaminada de sentimientos, expectativas e intereses de las personas que consultamos. Es información que no se puede considerar perfecta.

Una vez obtenida la información comienza el proceso de abstracción. Allí debemos ejecutar actividades como delimitar (decidir qué está por dentro de la frontera de la realidad y qué no), descomponer (dividir la realidad en partes), conceptualizar (construir el mapa de conceptos), simplificar (desechar información que no sea útil), estructurar (identificar formas, niveles y relaciones entre los conceptos), clasificar (asociar elementos de la realidad con estructuras y conceptos), mejorar el vocabulario (identificar términos ambiguos o conceptos sin un nombre claro y proponer una forma de nombrarlos) y caracterizar (asociar características con los conceptos identificados), entre otras. Esta actividad de abstracción la retomaremos más adelante en esta misma sección.

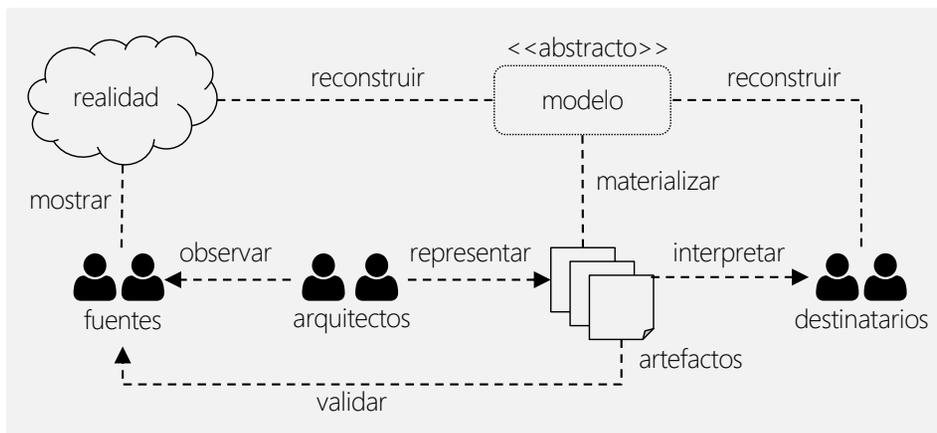


Fig. 3 – Etapa de creación y validación de los artefactos

- > Después del proceso de abstracción, la tarea de los arquitectos es crear los artefactos necesarios para representar el modelo y comunicarlo de manera adecuada a un conjunto de personas que vamos a llamar los “destinatarios” (figura 3). Estas personas pueden ser los mismos arquitectos en una etapa posterior del proyecto o cualquier otro participante del mismo.
- > El objetivo es que los destinatarios puedan reconstruir en su cabeza el modelo a partir de los artefactos y con eso logren una comprensión adecuada de la realidad.
- > Para validar los artefactos se debe recurrir de nuevo a las fuentes originales de la información, porque son los únicos que pueden hacer ese trabajo de revisión.

Lo que debemos mantener presente todo el tiempo es que el objetivo de construir el modelo es permitir que los destinatarios puedan usar el modelo en lugar de la realidad para hacer una tarea. Que para ellos sea indistinto. Esta tarea puede ser analizar la situación actual de la empresa o tomar una decisión o diseñar una transformación. Por esa razón el modelo debe ser una buena síntesis de la realidad, debe incluir todos los aspectos esenciales y suficiente información para que el modelo pueda reemplazar la realidad sin que eso afecte el resultado de la tarea. Pero de nuevo, como el modelo es algo abstracto, el reto está en que los artefactos logren comunicar a los destinatarios de manera correcta el modelo que representan y esa es nuestra responsabilidad como arquitectos. Por eso, debemos permanentemente tener en mente quiénes son los destinatarios y cuál es la tarea que tienen que hacer, para que los artefactos cumplan su propósito.

La abstracción como una agrupación de actividades

La palabra "abstraer" viene del latín *abstrahere* que significa "separar, apartar" y hace referencia a ese proceso de ignorar todas las características del objeto que no se necesitan para hacer el proceso de clasificación, separando lo esencial del objeto de lo que no lo es. La abstracción, en general, es un proceso mental que nos permite asociar objetos (elementos) con conceptos, ignorando las características particulares del mismo que no son indispensables para hacer dicha asociación. Cuando observamos un perro, por ejemplo, nuestro cerebro lo clasifica en la categoría de perros, haciendo abstracción (ignorando) detalles como la raza, el color, el tamaño, entre otros.

En nuestro caso como arquitectos, nuestro trabajo consiste en partir de un conjunto de objetos (elementos físicos o no) que observamos en la empresa (o su entorno) e identificar y hacer la asociación con los conceptos a los que corresponden. Allí vamos a observar y clasificar elementos como productos, maquinarias, clientes, contratos, procesos, bases de datos, etc. Para eso desarrollamos algunas de las actividades que se ilustran a continuación, que inician cuando ya hemos terminado la etapa de observación de la realidad.

"The purpose of abstraction is **not** to be vague, but to create a **new** semantic level in which one can be absolutely precise"

Edsger W. Dijkstra
"The Humble Programmer"

Por alguna razón, algunos arquitectos piensan que aumentar el nivel de abstracción es entrar en el terreno de la vaguedad y la imprecisión, y eso no es cierto. El esfuerzo que debemos hacer lo plantea con claridad Dijkstra y es crear unas estructuras abstractas, con un significado bien definido, que nos permitan expresar realidades complejas de manera simple (perdiendo algunos detalles, por supuesto), pero manteniendo la posibilidad de ser precisos, no solo en la manera de expresarlo (buenos lenguajes) sino también en lo que significan, de manera que no haya ambigüedad en la interpretación de dichas estructuras.

1. Delimitar: definir la frontera (en alcance y profundidad) teniendo en cuenta el destinatario del modelo y la tarea que este va a realizar (el uso que le va a dar al modelo)

Definir el alcance horizontal (frontera):

En esta actividad debe quedar clara la frontera de la realidad sobre la cual se va a construir el modelo (qué está por dentro y qué está por fuera). Podría incluir la empresa entera o solo una parte (como por ejemplo una unidad de negocio o una unidad organizacional o un proceso de negocio) y podría incluir o no algunos elementos externos a la empresa como socios, competidores, proveedores o clientes, e incluso entes de vigilancia y control.

Definir el alcance vertical (altura):

Ahora debemos decidir el nivel de detalle al que vamos a llegar. Tenga en cuenta que los elementos que estamos observando tienen partes, las cuales están compuestas a su vez de otras partes "más pequeñas". La pregunta es hasta qué punto vamos a detallar el modelo y cuánta información vamos a incluir. En el caso de una aplicación, por ejemplo, podríamos tener una ficha técnica de alto nivel o podríamos descender en detalles que nos pueden llevar hasta el código mismo de la aplicación. La pregunta es ¿para qué se va a usar este modelo? Dependiendo de la respuesta debemos definir qué tanto detalle incluimos (cuánto bajamos).

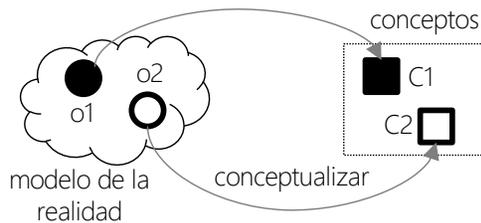
Conceptualizar es una actividad crítica

2. Descomponer: los modelos empresariales suelen ser grandes por lo que su construcción debe abordarse por partes. Para esto se debe identificar y documentar una buena descomposición de la realidad que quedó dentro de la frontera. Se puede hacer una descomposición por unidad de negocio, por eslabón de la cadena de valor, por área dentro de la empresa o por cualquier otro criterio que facilite el trabajo. Es usual, por ejemplo, que los modelos de tecnología de información los dividamos en 3 partes: (1) aplicaciones, (2) información e (3) infraestructura o plataforma tecnológica y que cada parte la abordemos con un equipo distinto de expertos.

Esta descomposición es una decisión que se debe tomar en el inicio de cualquier proyecto de arquitectura, porque va a tener impacto sobre el equipo de trabajo que se va a asignar y sobre la metodología que se va a utilizar.

3. Simplificar, filtrar y limpiar la información: como vimos antes, el proceso de levantamiento de información nos va a generar una alta cantidad de documentación no estructurada (entrevistas, talleres, presentaciones, etc.), la cual muy posiblemente viene llena de ruido, que puede dificultar el trabajo que vamos a desarrollar. Por eso, en esta actividad intentamos depurar la información recibida, eliminar la información que está por fuera de la frontera, eliminando cosas que en este momento no nos interesan (por ejemplo, reportes de problemas, o planes de largo plazo, o anécdotas de los empleados o la historia de la empresa), filtrar información que no está dentro del alcance vertical del modelo (ignorar por ejemplo documentos con detalles técnicos que están por fuera del alcance objetivo), eliminar opiniones (que en este momento tampoco son relevantes, porque estamos interesados solo en los hechos) e información que no es confiable (debemos analizar las fuentes para determinar la veracidad de la información).

4. Conceptualizar: esta es la actividad central del proceso, porque un requisito para poder hacer la clasificación (asociación) de los elementos observados en la realidad es tener diseñado el marco de conceptos que vamos a utilizar, tal como se sugiere en la figura. Allí aparecen dos



elementos o1 y o2 presentes en la realidad y los respectivos conceptos C1 y C2 con los cuales están asociados. Ese marco de conceptos es lo que se denomina el modelo conceptual o modelo ontológico y define el universo de conceptos existentes en la realidad. Cada concepto tiene un nombre y una descripción. Esto sirve para definir el vocabulario (glosario de términos) que vamos a utilizar en el proyecto. Pero las dos tareas más retadoras de esta actividad de conceptualización son identificar la estructura del mapa de conceptos (los conceptos tienen relaciones entre sí que debemos entender y describir) y caracterizar cada uno de los conceptos del mapa.

Definir el vocabulario

En muchos casos los términos que se utilizan en la empresa son ambiguos (se utiliza el mismo término para referirse a dos conceptos distintos). El modelo conceptual debe definir el vocabulario que se debe utilizar en la construcción del modelo.

Estructurar el mapa de conceptos

Los conceptos tienen relaciones diversas entre ellos. En esta tarea, que se aborda en detalle en la segunda sección de esta lectura, debemos identificar y describir las relaciones existentes entre los conceptos. Esa va a ser la estructura del modelo.

Caracterizar cada concepto

Un concepto tiene un conjunto de características (expresadas como atributos) que contienen información importante de la realidad. El concepto de mesa, por ejemplo, tiene como atributos sus medidas, su color y el material de fabricación, entre otros.

Ahora sí, ¿qué es un modelo?



Un **modelo** es una abstracción de una realidad, la cual se expresa a través de un conjunto de artefactos. Es muy común que el término "modelo" se asigne directamente a los artefactos, sin hacer diferencia entre lo que se quiere expresar y el mecanismo de expresión. En adelante usaremos indistintamente el término para referirnos a la abstracción o al conjunto de artefactos.

Propiedades de un modelo:

- > Como es posible construir muchos modelos distintos para representar una misma realidad, es responsabilidad de un arquitecto definir el modelo que se debe construir para abordar una tarea específica. En este proceso se deben considerar dos variables importantes: tiempo y costo. Vamos a utilizar el término **adecuado** (*suitable*) para calificar un modelo que es suficiente para desarrollar una tarea. Cualquier información innecesaria incluida en el modelo se considera un desperdicio. Siempre debemos buscar el modelo más simple que sea adecuado.
 - > Otra propiedad importante de un modelo es su corrección. Un modelo es **correcto** si representa de manera veraz la realidad. Como los modelos se construyen para reemplazar la realidad como base en el desarrollo de tareas, sería grave si el modelo no fuera correcto. Imagine lo que pasaría con un modelo financiero de una empresa que no refleja bien la realidad...
 - > Otra cosa distinta es si el modelo es o no **preciso** (*accurate*). La precisión se refiere a si representa de manera exacta la realidad. Es posible tener modelos correctos, poco precisos, que sean adecuados para una tarea. Piense por ejemplo, que en vez de valores numéricos para alguna característica de la realidad usamos rangos (0-10, 10-20, ...) o una escala (alto, medio, bajo).
 - > Un modelo es **ambiguo** si existe más de una interpretación posible para su contenido. Crear modelos ambiguos es un error en arquitectura, porque no estamos seguros de la interpretación que le vayan a dar los destinatarios. La ambigüedad la introducen artefactos demasiado informales.
- > Otra característica de un modelo es si está **bien construido**. Esto hace referencia a si respeta el modelo conceptual que lo acompaña, a si está bien estructurado. Esto debe quedar un poco más claro en la sección de metamodelos, en donde diremos que todo modelo debe ser **conforme** con su metamodelo. Pero no nos adelantemos.
 - > La realidad que queremos modelar tiene tres dimensiones (aspectos, facetas) complementarias que muchas veces modelamos de manera independiente. Por un lado están los **aspectos estructurales** que representan la forma que tiene la realidad. El modelo conceptual es un ejemplo de un modelo que representa un aspecto estructural (estructura de conceptos). Son modelos estáticos, en el sentido de que no transmiten cómo cambian de estado. Están por otro lado los aspectos **dinámicos** o de **comportamiento**. Los modelos que representan estos aspectos nos muestran la forma en que la realidad se comporta: cómo y cuándo suceden las cosas. Los conceptos como tiempo, acción y reacción son su fundamento. Complementan por supuesto los aspectos estructurales. También encontramos los aspectos **funcionales**. En ese caso lo que hacemos es tomar la realidad como una caja negra y observar desde afuera para qué sirve, qué es capaz de hacer y cómo se relaciona con lo que está por fuera de la frontera. Considere un vehículo. El modelo estructural habla de su chasis, sus llantas, sus puertas, etc. y las relaciones que hay entre ellos. El modelo de comportamiento explica cómo trabaja por dentro el motor y la transmisión. El modelo funcional hace referencia a la manera de conducir el vehículo.

Tipos de artefactos

Un artefacto es un instrumento (medio) para representar una parte de un modelo (que puede corresponder a un aspecto). Muchas veces un artefacto se nombra como si fuera un modelo. Hablamos por ejemplo del modelo de negocio, del modelo estratégico o del modelo financiero, aunque sabemos que son solo artefactos que representan una parte del modelo de una empresa. No está mal y es muy común que eso suceda. Lo que sí debemos tener claro todo el tiempo es que esos "modelos" no son independientes y que tienen elementos en común o relaciones entre sus elementos. Tampoco debemos olvidar que algunos de esos modelos requieren más de un artefacto para ser descritos. Por ejemplo, el modelo de negocio puede usar un artefacto para representar su estructura (componentes, canales, actividades, etc.) y otro artefacto distinto para representar su comportamiento (cómo y cuándo suceden las cosas en el negocio). El reto para un arquitecto es definir el conjunto de artefactos que va a utilizar (cuántos y cuáles) para representar un modelo.

Formato

Contiene información estructurada de un solo elemento de la realidad. Su objetivo es describirlo presentando los valores de sus atributos.

Nombre atributo	Valor atributo
ID	
Nombre	
Apellido	
Fecha nacimiento	

Tipos de artefacto

Incluye información básica de una lista de elementos. Su objetivo es mostrar los elementos de la realidad de un cierto concepto.

ID	Nombre	Apellido	Fecha nacimiento	Salario	Cargo

Catálogo

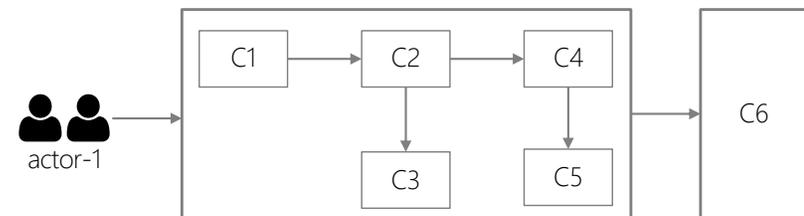
Matriz

Su objetivo es cruzar la información de dos tipos de elementos para estudiar sus relaciones. En el ejemplo hay una "X" si existe una relación entre un elemento A-i (del concepto Aplicación) con un elemento P-j (del concepto Proceso de negocio).

	A1	A2	A3	A4	A5	A6	A7
P1	X	X		X		X	
P2			X			X	
P3				X			X

Diagrama

Su objetivo es mostrar la estructura y las relaciones de un conjunto de elementos. Para esto se utilizan elementos gráficos (formas o íconos) que representan los elementos y líneas para mostrar las relaciones.



Artefactos y lenguajes

El tipo de artefacto solo define la estructura del mismo, así que necesitamos definir además el lenguaje (textual, matemático o gráfico) que se va a utilizar en cada parte del mismo. Cuando se trata de diagramas, el lenguaje es algo muy importante porque determina las reglas que se deben seguir y su significado. Lo más común es que se utilice lenguaje natural como parte (o complemento) de todo tipo de artefacto, con el consecuente riesgo de tener artefactos ambiguos. Veamos ahora un ejemplo completo de lo que hemos visto hasta ahora, y después continuamos profundizando en el tema de los lenguajes, el cual será abordado a fondo en la sección 4.

Ejemplo simple

Piense por un instante en su vehículo. Si no tiene uno, piense en el vehículo de un familiar o un amigo. Uno concreto. Visualícelo lo más claro posible en su mente. Eso que tiene en su cabeza en este instante es un modelo. Abstracto e intangible, pero lleno de información útil. Ahora imaginemos algunos usos distintos de ese modelo y determinemos los artefactos adecuados para comunicarlo. Recuerde que el objetivo del modelo (expresado a través de unos artefactos) es remplazar el vehículo real para ejecutar una tarea específica. El artefacto más simple es un texto en lenguaje natural. Si Ud. escribe en una hoja la descripción de su vehículo está construyendo un modelo (usando un artefacto). Eso es suficiente si va a poner un aviso en un periódico para venderlo. Un artefacto un poco más elaborado es la tarjeta de propiedad del vehículo, que corresponde al tipo de artefacto formato. Allí aparece más información, estructurada en parejas atributo-valor. Con este artefacto es posible hacer algunos trámites. Si consulta el manual del vehículo, va a encontrar allí una guía de uso, que corresponde a un artefacto con el aspecto funcional del modelo. En ese mismo manual es posible que aparezcan algunos diagramas con la estructura del vehículo, que se van a necesitar cuando se deba llevar a un taller. Encontrará diagramas generales del vehículo y diagramas en donde se detalla, por ejemplo, el sistema de amortiguación o el sistema eléctrico (más detalle implica menor altura del modelo). Un artefacto de tipo catálogo tendría la lista de todas las piezas del vehículo, con un código y un nombre. Ese artefacto se usa cuando se debe buscar un repuesto y se necesita tener el código para poderlo pedir. Una foto del vehículo sería otro artefacto, que utilizan las aseguradoras para verificar el estado del vehículo antes de asignarle una póliza. Es posible incluso tener un modelo financiero del vehículo, en el que se tiene un registro de todos los gastos en los que se incurre o de las cuotas que se han pagado, si fue adquirido a crédito.

- > Hay lenguajes más formales que otros, en el sentido de que su semántica (significado) está mejor definida, evitando así diversas interpretaciones.
- > Un lenguaje muy informal lo encontramos en las presentaciones que se suelen hacer usando herramientas como *PowerPoint*. Allí entre figuras, flechas, iconos, frases en lenguaje natural y una persona presentando, encontramos todas las interpretaciones posibles.

Tampoco debemos caer en la trampa de que solo las sintaxis matemáticas son formales. Es posible tener lenguajes gráficos formales, con una semántica bien definida.

"Many people believe that if a language is formal looking it must be formal, as if they were measuring formality by how many Greek letters and mathematical symbols the language contains. This is a myth" - David Harel

Modelos descriptivos vs. modelos prescriptivos

Un modelo se puede utilizar para representar la realidad en su estado actual (se denomina modelo descriptivo) o se puede utilizar para representar una realidad que todavía no existe pero que se quiere construir (se denomina modelo prescriptivo). Estas dos clases de modelos se ilustran en la figura 4.

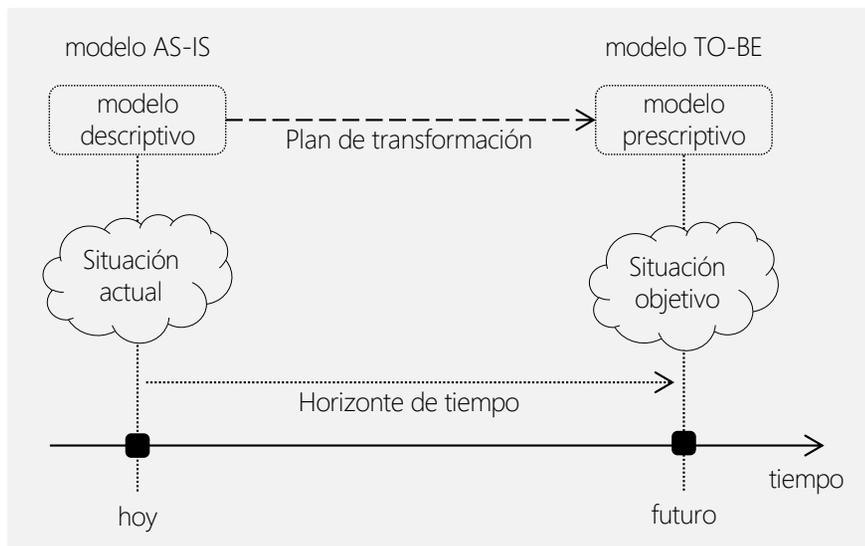


Fig. 4 – Modelos descriptivos vs. prescriptivos

- > Así como el modelo AS-IS se recoge a través de observación (directa e indirecta), el modelo TO-BE es el producto de un proceso de diseño y validación con los responsables de la empresa. Se utiliza una aproximación metodológica distinta.
- > Mientras el modelo AS-IS recoge hechos, estructurados en un grupo de artefactos, el modelo TO-BE recoge decisiones de lo que se debe tener (lograr) en un cierto instante de tiempo en el futuro.
- > El modelo TO-BE no se debe expresar necesariamente a través de los mismos artefactos que el modelo AS-IS. Se deben diseñar de manera que la situación objetivo sea construible.

- > En todos los proyectos de arquitectura, en donde hay un proceso de transformación, encontramos el modelo AS-IS (que representa la situación actual) y el modelo TO-BE (que representa la situación objetivo).
- > El primero es un modelo descriptivo con la situación actual de la empresa, posiblemente expresado a través de varios modelos como pueden ser el de negocio, el de aplicaciones, el de información y el de infraestructura. Estos modelos están representados por múltiples artefactos, escritos en distintos lenguajes. Su propósito es representar la situación de hoy de la empresa y sirven para hacer distintos análisis y diagnósticos, pero sobre todo para tomar decisiones.
- > El modelo TO-BE, por su parte, refleja las decisiones de la empresa de transformarse, indicando cómo debe ser su situación una vez transcurrido el horizonte de tiempo planteado.

- > El objetivo al desarrollar el modelo TO-BE (en lugar de lanzarse de una vez a construirlo) es permitir proponer escenarios para analizarlos, evaluarlos, estimar el esfuerzo y ajustarlos, como una forma de disminuir los riesgos de la transformación y como una manera de hacer explícito para todos los involucrados en la empresa cuál es el futuro que se quiere lograr.
- > Por eso es tan importante definir y construir el conjunto de artefactos necesario, y a la altura adecuada, para que sea útil. Que de verdad sirva. Si el modelo TO-BE se queda en generalidades con recomendaciones de alto nivel, se vuelve ambiguo y su utilidad disminuye (en realidad desaparece).

2. ¿Qué es un metamodelo?

Como vimos en la sección anterior, dentro de la actividad de modelar, los arquitectos desarrollamos dos tareas en paralelo: la primera es crear el modelo conceptual del dominio (de la realidad) y la segunda es crear el modelo, expresándolo a través de un conjunto de artefactos, en el que se identifican los elementos presentes en la realidad y se asocian con los conceptos. En la figura 5 se ilustra este proceso y se muestra que un modelo conceptual se puede representar utilizando un metamodelo.

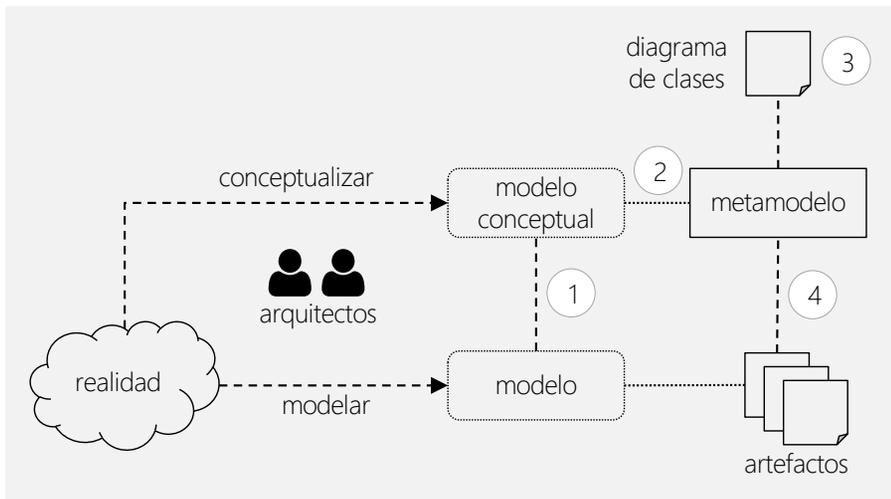


Fig. 5 – Modelo conceptual y metamodelos

- 1 En el modelo hay elementos, cada uno de los cuales está asociado con un concepto del modelo conceptual.
- 2 El modelo conceptual se puede representar por medio de un metamodelo, que es una herramienta que permite especificar estructuras de entidades.
- 3 Un metamodelo se puede representar usando un subconjunto de elementos de un diagrama de clases de UML.
- 4 Los artefactos deben respetar (cumplir con) el metamodelo del dominio, desde el punto de vista de contenido (solo puede haber en el modelo elementos cuyo concepto asociado esté en el metamodelo) y estructura (los elementos solo pueden tener las relaciones que tienen en el metamodelo los conceptos a los que pertenecen).



Un **metamodelo** es una estructura de entidades (conceptos o clases), asociaciones y restricciones que permite representar modelos conceptuales. Cada entidad tiene un nombre que es único, un conjunto de atributos, un conjunto de asociaciones con otras entidades y un conjunto de restricciones. Un metamodelo se puede representar utilizando un subconjunto de los diagramas de clases de UML, con el entendido de que el objetivo en este caso no es hacer ningún tipo de implementación.

Los metamodelos son herramientas muy poderosas y útiles, que se pueden usar para definir formalmente todo tipo de estructura. Adicional al modelo conceptual del dominio de una empresa (en donde aparecen los conceptos de ese negocio, como pueden ser los conceptos de cliente, cuenta, tarjeta de crédito, préstamo, para el caso de un banco), los vamos a usar también para definir las estructuras de los artefactos (en donde aparecen conceptos como formato o matriz), las estructuras de los lenguajes de dominio (en donde veremos iconos, formas y líneas), así como las estructuras de los dominios de trabajo de los arquitectos. Vamos a ver la manera de crear el metamodelo de un proceso de negocio, el metamodelo de un diagrama de casos de uso, el metamodelo de una estrategia o el metamodelo de una arquitectura de infraestructura tecnológica. Los metamodelos son el sueño de los arquitectos, primero por su sencillez, y segundo por su alta capacidad de expresión cuando se trata de especificar una estructura. Y es que un metamodelo no es otra cosa que el modelo de los modelos.

Elementos de un metamodelo

Un metamodelo es un grafo que contiene 4 tipos de elementos que se describen a continuación:

<div style="text-align: center; background-color: #d3d3d3; padding: 5px; margin-bottom: 10px;">Entidad</div> <p>En nuestro caso las entidades serán los conceptos del dominio. Estos conceptos pueden hacer parte de un negocio o de cualquier otro dominio de conocimiento. Un concepto se identifica con un nombre que debe ser único dentro del metamodelo, debe ser un nombre que identifique claramente el concepto que va a representar, corresponder a un sustantivo, comenzar con una letra mayúscula y estar en singular. Se utiliza como sintaxis una caja, siguiendo lo establecido en un diagrama de clases de UML. Es conveniente que cada concepto tenga una corta descripción, que debe ir en un artefacto complementario al diagrama de clases:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>Concepto-1</td></tr> <tr><td> </td></tr> </table> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #d3d3d3;"><tr><th>Concepto</th><th>Descripción</th></tr></tr></thead> <tbody> <tr><td>Concepto-1</td><td>Descripción Concepto-1</td></tr> </tbody> </table> </div>	Concepto-1		Concepto	Descripción	Concepto-1	Descripción Concepto-1	<div style="text-align: center; background-color: #d3d3d3; padding: 5px; margin-bottom: 10px;">Atributo</div> <p>Un atributo de un concepto es una característica que este tiene y que resulta importante dentro del dominio. Cada atributo tiene un nombre que lo identifica, debe corresponder a un sustantivo, comenzar con una letra minúscula, reflejar claramente la característica que representa y puede (o no) tener asociado un tipo simple de dato, el cual indica los valores que este atributo puede tomar. Los atributos se sitúan dentro de la caja del concepto (como aparece en la figura) y es conveniente que tenga una corta descripción en un artefacto complementario:</p> <div style="display: flex; justify-content: space-around; align-items: flex-start; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>Concepto-1</td></tr> <tr><td>atributo-1: tipo-1 atributo-2: tipo-2</td></tr> </table> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #d3d3d3;"><tr><th>Atributos</th><th>Descripción</th></tr></tr></thead> <tbody> <tr><td>atributo-1</td><td>Descripción atributo-1</td></tr> <tr><td>Atributo-2</td><td>Descripción atributo-2</td></tr> </tbody> </table> </div>	Concepto-1	atributo-1: tipo-1 atributo-2: tipo-2	Atributos	Descripción	atributo-1	Descripción atributo-1	Atributo-2	Descripción atributo-2
Concepto-1															
Concepto	Descripción														
Concepto	Descripción														
Concepto-1	Descripción Concepto-1														
Concepto-1															
atributo-1: tipo-1 atributo-2: tipo-2															
Atributos	Descripción														
Atributos	Descripción														
atributo-1	Descripción atributo-1														
Atributo-2	Descripción atributo-2														
<div style="text-align: center; background-color: #d3d3d3; padding: 5px; margin-bottom: 10px;">Asociación</div> <p>Las relaciones entre conceptos las vamos a representar a través de asociaciones. Solo vamos a utilizar dos tipos de relaciones entre conceptos de las múltiples que tiene UML: (a) las relaciones estructurales y (b) las relaciones de especialización. Este tema será abordado en la siguiente página. Por ahora veamos la sintaxis de cada una de esas relaciones:</p> <div style="margin-top: 10px;"> <table style="border-collapse: collapse; margin-bottom: 10px;"> <tr> <td style="border: 1px solid black; padding: 5px; text-align: center;">Concepto-1</td> <td style="text-align: center; padding: 0 10px;">→</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">Concepto-2</td> <td style="padding-left: 10px;">Concepto-1 tiene una relación con Concepto-2</td> </tr> </table> <table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 5px; text-align: center;">Concepto-1</td> <td style="text-align: center; padding: 0 10px;">▷</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">Concepto-2</td> <td style="padding-left: 10px;">Concepto-1 es una especialización de Concepto-2</td> </tr> </table> </div>	Concepto-1	→	Concepto-2	Concepto-1 tiene una relación con Concepto-2	Concepto-1	▷	Concepto-2	Concepto-1 es una especialización de Concepto-2	<div style="text-align: center; background-color: #d3d3d3; padding: 5px; margin-bottom: 10px;">Restricción</div> <p>Existen dos tipos de restricciones en un metamodelo: el primer tipo corresponde a restricciones sobre los valores que pueden tomar los atributos de un concepto. Esta información se debe incluir en la descripción del atributo. No necesitamos para eso ningún elemento adicional. El segundo tipo de restricciones tienen que ver con la estructura del modelo, y esos sí requieren un artefacto adicional para expresarlas. Estas restricciones se pueden expresar en lenguajes formales, pero en la mayoría de los casos es suficiente con expresarlas en lenguaje natural.</p> <div style="margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #d3d3d3;"><tr><th>ID</th><th>Descripción de la restricción</th></tr></tr></thead> <tbody> <tr><td>ID-1</td><td>Descripción Restricción-1</td></tr> <tr><td>ID-2</td><td>Descripción Restricción-2</td></tr> </tbody> </table> </div>	ID	Descripción de la restricción	ID-1	Descripción Restricción-1	ID-2	Descripción Restricción-2
Concepto-1	→	Concepto-2	Concepto-1 tiene una relación con Concepto-2												
Concepto-1	▷	Concepto-2	Concepto-1 es una especialización de Concepto-2												
ID	Descripción de la restricción														
ID	Descripción de la restricción														
ID-1	Descripción Restricción-1														
ID-2	Descripción Restricción-2														

Relaciones estructurales entre conceptos

Dentro del proceso de construcción de un modelo, adicional a los conceptos que identificamos, observamos que los conceptos tienen relaciones entre ellos. Un vehículo (concepto Vehículo) tiene un motor (concepto Motor), un vehículo tiene 4 llantas (concepto Llanta), una empresa (concepto Empresa) está dividida en departamentos (concepto Departamento), en los cuales trabajan empleados (concepto Empleado). Esas relaciones las vamos a denominar estructurales, porque definen la forma que tiene la realidad, la cual debe quedar plasmada en el metamodelo. Veamos el caso más simple de relación, que aparece ilustrada en la figura 6.

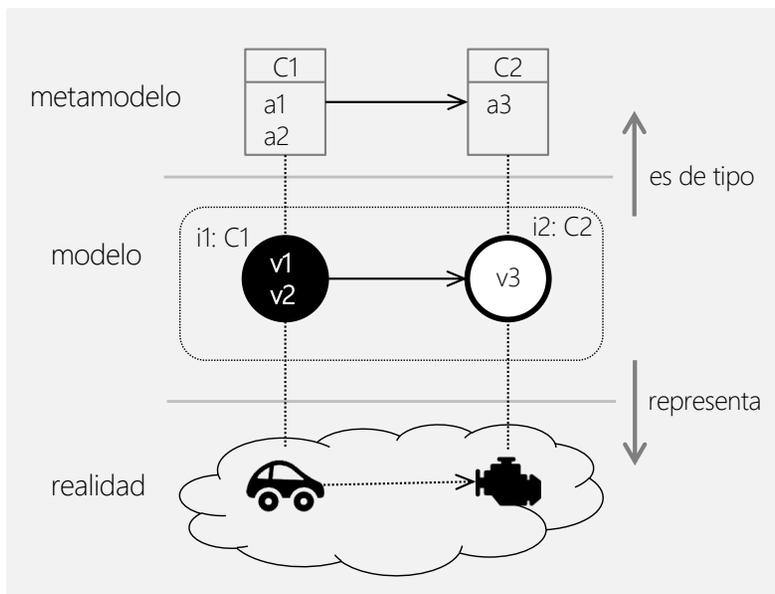


Fig.6 – Una relación entre dos conceptos

- > En la figura aparecen dos conceptos C1 y C2 en el metamodelo, el primero con dos atributos a1 y a2, y el segundo con un atributo a3. Hay además una relación desde C1 hacia C2.
- > En el modelo aparecen dos elementos i1 e i2. El primero está asociado con el concepto C1 (se dice que es de tipo C1) y el segundo está asociado con el concepto C2 (es de tipo C2). Se dice también que i1 es una instancia de C1 y que i2 es una instancia de C2. i1 tiene dos valores v1 y v2, el primero es el valor del atributo a1 y el segundo es el valor del atributo a2. La instancia i2 tiene a su vez un valor v3 asociado con el atributo a3. El modelo tiene que cumplir con la estructura definida desde el metamodelo. La relación que existe en el metamodelo, que dice que toda instancia de C1 debe estar relacionada con una instancia de C2, aparece dibujada en el modelo con una flecha. El metamodelo dice que esa relación es obligatoria y que va desde C1 hacia C2 (las relaciones tienen dirección).
- > La instancia i1 representa un elemento que aparece en la realidad, la instancia i2 representa otro elemento, y entre ellos existe una relación (así es el mundo que estamos modelando).



Cuando un modelo M cumple con todas las reglas que define un metamodelo MM se dice que M es **conforme** con MM. Dentro de esas reglas está que toda instancia tiene un valor para cada uno de los atributos definidos en el metamodelo y que este valor respeta el tipo de dato del atributo. Si el metamodelo dice que el tipo de un atributo debe ser entero, el valor que tiene la instancia para ese atributo debe ser entero.



Un metamodelo siempre tiene un concepto principal, llamado la **raíz** del metamodelo, cuyo objetivo es representar el dominio completo. Partiendo de ese punto se leen los metamodelos, recorriendo las relaciones que conectan la raíz con todos los demás conceptos. No pueden existir conceptos que no sean alcanzables desde la raíz, navegando por las relaciones.

Relaciones estructurales entre conceptos

En las figuras 7 y 8 aparecen ilustrados otros escenarios de metamodelos y modelos, en donde se agregan características a las relaciones y se muestra el significado a nivel de conceptos y de instancias.

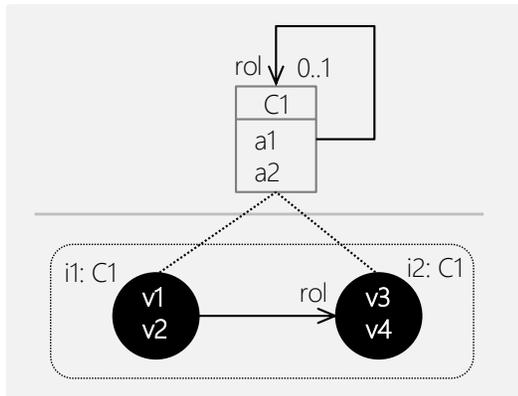


Fig.7 – Relaciones hacia sí mismo

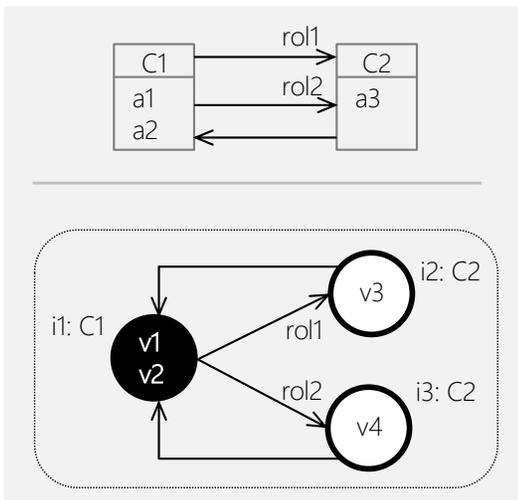


Fig.8 – Múltiples relaciones

- > En el metamodelo solo hay un concepto C1, que tiene una relación hacia sí mismo. Eso quiere decir que toda instancia de C1 tiene una relación hacia otra instancia de ese mismo tipo.
- > Aparecen en esa relación dos nuevas características: la primera es que le ponemos un **nombre al rol** que juega la instancia de destino (i2) con respecto a la instancia desde la que sale la relación (i1). En el ejemplo le pusimos el nombre "rol", que se puede observar también en el modelo. Allí se lee que i2 es el "rol" de i1. Si el concepto C1 fuera por ejemplo el de Persona, y el nombre del rol fuera "padre", se leería como i2 es el padre de i1.
- > La segunda característica es la **cardinalidad** de la relación. En este caso aparece una cardinalidad "0..1" que quiere decir que cada instancia de C1 tiene una o ninguna instancia de C1 asociada. En el caso en el que la cardinalidad no sea explícita, se entiende como que tiene el valor 1 ("0..1" representa una relación opcional, mientras "1" representa una relación obligatoria).
- > En la figura 8 encontramos de nuevo dos conceptos C1 y C2, que tienen varias relaciones entre ellos. Lo primero es que hay dos relaciones desde C1 hacia C2, por lo que el nombre del rol es obligatorio para poderlas distinguir. Fíjese que el nombre no se asigna a la relación como un todo, sino a la punta de la flecha que llega al concepto de destino. En el ejemplo aparecen los nombres "rol1" y "rol2".
- > En el modelo aparece ahora una tercera instancia i3, de tipo C2, con un valor v4 para su atributo a3. Observe que desde i1 sale una relación hacia i2 y una relación hacia i3. Se lee "i2 es el rol1 de i1", mientras "i3 es el rol2 de i1". Si C1 fuera el concepto Curso y C2 fuera el concepto Persona, rol1 podría ser "profesor" y rol2 podría ser "monitor". En ese caso la lectura del modelo diría que i1 es un curso que tiene un profesor que es i2 y un monitor que es i3, en donde i2 e i3 son instancias de Persona.
- > En la figura tenemos además una relación que va desde C2 hacia C1 (va en la otra dirección). La cardinalidad es 1, luego es obligatoria. Eso quiere decir que cada instancia de C2 (en este caso i2 e i3) debe tener una relación hacia una instancia de C1 (en este caso i1, pero podría ser cualquier otra instancia de C1).
- > Si la relación va en las dos direcciones y queremos hacer explícito que es entre las mismas instancias, usamos una línea sin flechas entre los dos conceptos (para dejar claro que no se trata de dos relaciones distintas, sino de una misma relación en las dos direcciones).

Cardinalidad de una relación

Adicional a las relaciones opcionales (0..1) y a las relaciones obligatorias (1), existen las relaciones con una cardinalidad múltiple, en las cuales existen varias instancias asociadas. La sintaxis y la explicación las encontramos en la figura 9.

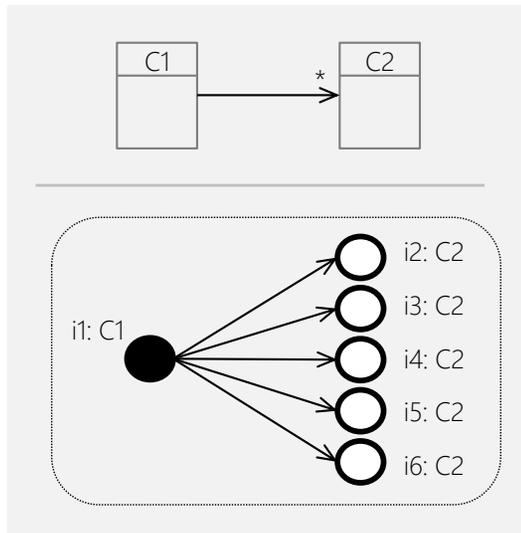
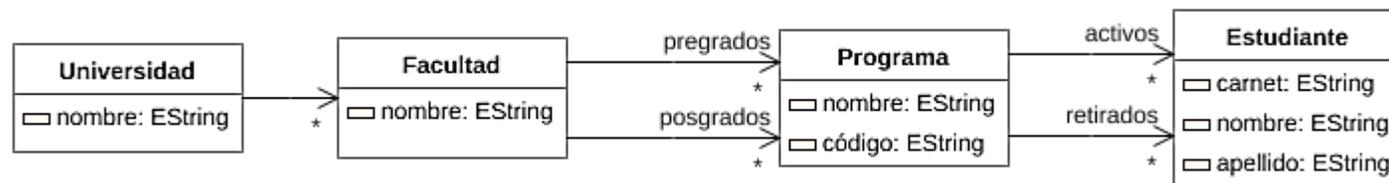


Fig. 9 – Cardinalidad múltiple

- > Ahora en el metamodelo tenemos dos conceptos C1 y C2 conectados por una relación con cardinalidad múltiple (*). Esto quiere decir que cada instancia de C1 (en el ejemplo i1) va a tener relaciones hacia varias instancias de C2 (en el ejemplo i2, i3, i4, i5 e i6).
- > Por simplicidad en el dibujo, eliminamos los atributos de los conceptos C1 y C2, para no tener que incluir los valores de los atributos en las instancias.
- > Es posible asignar un nombre al rol (para darle un nombre al conjunto de instancias), pero solo es obligatorio si existe alguna ambigüedad. De resto basta con leer el concepto C2 en plural. Por ejemplo, si C1 corresponde al concepto de Curso y C2 al concepto de Estudiante, la lectura más simple dice que un curso tiene múltiples estudiantes. Si queremos ser más explícitos, al rol le podemos asignar el nombre de "matriculados", por ejemplo, y ese será el nombre del conjunto de estudiantes del curso.
- > Al igual que con las relaciones simples, puede haber varias relaciones múltiples entre dos conceptos. En ese caso el nombre del rol es obligatorio. Piense para el mismo ejemplo anterior, que ahora queremos tener dos conjuntos separados de estudiantes en el curso. Uno con los estudiantes de pregrado y otro con los estudiantes de posgrado. Esos serían los nombres de los roles de las dos relaciones con cardinalidad múltiple.

Ejemplo



La raíz del metamodelo es el concepto de Universidad. Representa el dominio completo que se quiere modelar. Todo se debe leer a partir de ese punto del metamodelo. Una Universidad tiene varias Facultades (cada una tiene un nombre). Una Facultad por su lado tiene programas académicos, unos de pregrado y otros de posgrado. Cada programa tiene un nombre y un código asignado por el Ministerio de Educación con el permiso de funcionar. Por su parte cada programa tiene un grupo de estudiantes, algunos se encuentran en este momento activos (o sea matriculados) y otros se han retirado del programa. Cada estudiante tiene un carnet, un nombre y un apellido.

La relación de especialización (herencia)

Antes de iniciar esta parte debemos aclarar que todo metamodelo, por complejo y grande que sea, se puede expresar usando la relación de estructura que acabamos de ver y, que esta nueva relación de especialización, no es de ninguna manera indispensable.

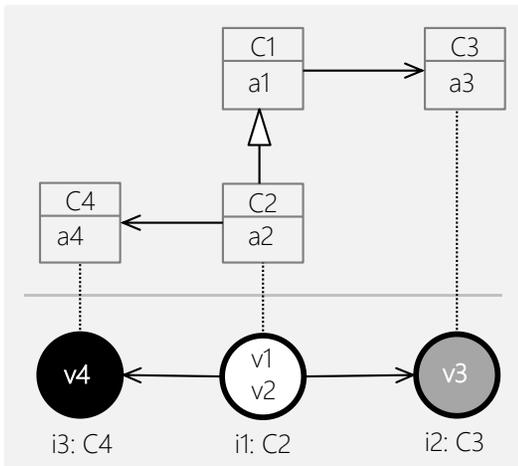
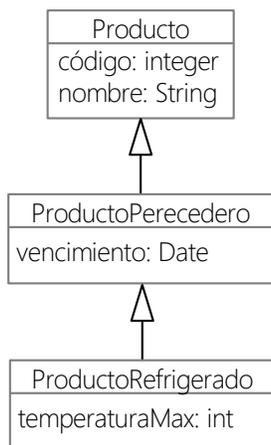


Fig.10 – Relación de especialización

- > En el metamodelo tenemos un concepto (o tipo o clase) C1, que tiene un atributo a1 y una relación hacia el concepto C3. Al mismo tiempo tenemos un concepto C2, que está definido como una especialización del concepto C1, que además tiene un atributo a2 y una relación hacia el concepto C4.
- > A nivel de modelo, tenemos una instancia i1 de tipo C2, la cual tiene dos valores v1 (asociado con a1) y v2 (asociado con a2). Al ser C2 una especialización de C1 hereda todos sus atributos y relaciones. Además, la instancia i1 tiene una relación hacia una instancia i2 de C3 (relación heredada del tipo C1) y una relación hacia una instancia i3 de C4 (propia del tipo C2).
- > Fíjese que en el ejemplo no tenemos instancias del tipo C1 (aunque sería posible tenerlas) y que solo usamos a C1 como base para definir el tipo C2. Esto quiere decir que las relaciones de especialización del metamodelo no se materializan en el modelo y que solo se usan para definir la estructura de las instancias de los tipos especializados.
- > El término "herencia" se utiliza también para hacer referencia a la relación de especialización. La sintaxis que se usa es la que aparece en la figura 15, en donde C2 hereda de C1.

Ejemplo



En este ejemplo tenemos 3 conceptos: (1) un producto, que tiene un código y un nombre, (2) un producto perecedero, que además de ser un producto (o sea que hereda de la clase Producto) tiene un atributo que indica su fecha de vencimiento y (3) un producto refrigerado, que es un producto perecedero, que incluye además información sobre la temperatura máxima a la que se debe almacenar.



i1: Producto

Así es una instancia de la clase Producto



i2: ProductoPerecedero

Así es una instancia de la clase ProductoPerecedero

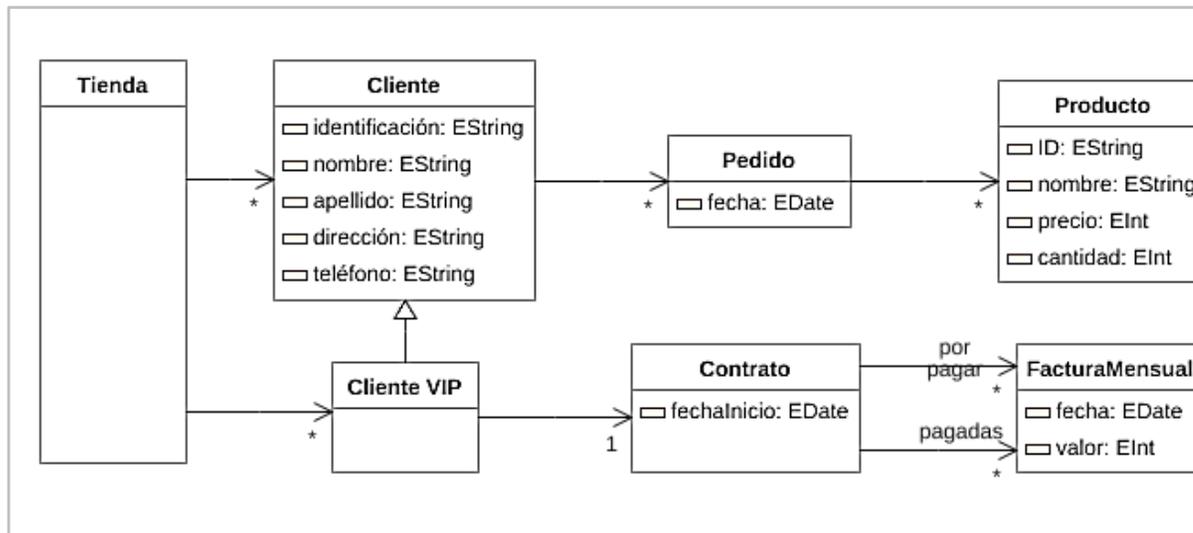


i3: ProductoRefrigerado

Así es una instancia de la clase ProductoRefrigerado

La relación de especialización (herencia)

Ejemplo



La tienda tiene dos tipos de clientes: unos normales y otros VIP (que también son clientes) los cuales adicionalmente tienen un contrato mensual que les permite recibir los pedidos a domicilio sin ningún pago adicional. Ese contrato se paga mensualmente y tiene un cierto valor. Todos los clientes tienen pedidos que están compuestos de productos, cada uno en una cierta cantidad y un precio.

¿Cuándo usar herencia?

- > La herencia surgió de la programación orientada por objetos y hace bastantes años, como un mecanismo de reutilización de código.
- > En nuestro caso el objetivo de la herencia es ayudarnos a estructurar de la manera más simple posible un modelo conceptual. Consideremos el ejemplo anterior: si no usamos la herencia, el diagrama saldría mucho más grande y estaríamos repitiendo información. Por eso, podemos aprovechar el hecho de que un Cliente VIP también es un Cliente, solo que es más especializado, en el sentido de que necesita más información para poder ser modelado.
- > Si encontramos conceptos en el dominio que tengan esa característica y que además la relación de herencia ayude a que éste resulte más simple, debemos usar esa relación.

¿Cuándo no usar herencia?

- > Cada vez que vayamos a usar la relación de herencia deberíamos preguntarnos (y cuestionarnos realmente) si de verdad hay una ganancia. Piense en el caso de la Universidad. Introducir la clase Persona (con atributos cédula, nombre y apellido) de la cual heredan las clases Estudiante, Profesor y Empleado es un desperdicio. Es más fácil repetir en las 3 clases esos 3 atributos que intentar dibujar un diagrama con esas relaciones de herencia. Además, ¿qué aporta?
- > En el caso de los metamodelos, como no existen métodos ni se van a implementar las clases, muchas veces las relaciones de herencia se pueden reemplazar por relaciones estructurales. Piense en el ejemplo anterior: ¿qué pasa si reemplazamos la relación de herencia por una relación estructural desde el Cliente VIP hacia el Cliente?

Expresión de restricciones

La estructura de un metamodelo se expresa sin problemas usando las relaciones entre conceptos que ya vimos. Pero hay una limitación y es que a nivel de metamodelo no podemos poner condiciones sobre las instancias y a veces eso es necesario. Veamos un par de ejemplos.

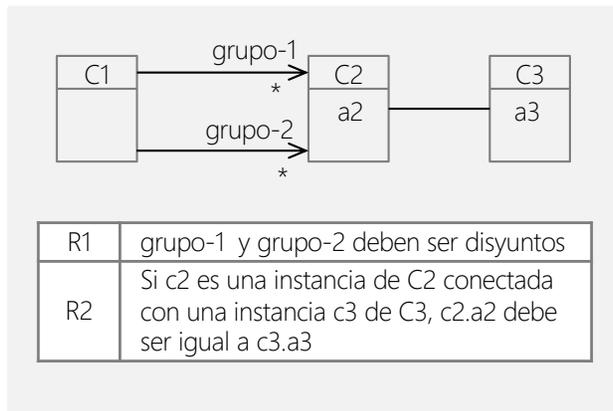


Fig.11 – Expresión de restricciones

- > En el metamodelo tenemos 3 conceptos C1, C2 y C3, con algunas relaciones entre ellos que explican su estructura sin ninguna ambigüedad. Pero imagine que queremos definir que grupo-1 y grupo-2 son disyuntos (las instancias de C2 que están en el grupo-1 no pueden estar también en el grupo-2). No hay manera de hacerlo a este nivel y por eso lo debemos expresar como una restricción.
- > Lo mismo sucede si quisiéramos definir alguna relación entre el valor del atributo a2 (de C2) y el valor del atributo a3 (de C3) cuando hay dos instancias de C2 y de C3 conectadas con una relación en las dos direcciones. Podríamos pedir por ejemplo que tuvieran el mismo valor, o que uno fuera mayor que el otro. Lo que sea. Eso también lo tenemos que definir como una restricción.
- > Aunque hay lenguajes formales para definir restricciones, para nuestro caso es suficiente con expresarlas en lenguaje natural.

Marcos de referencia y estándares de dominio

Algunos dominios ya tienen un modelo conceptual definido, público y ampliamente difundido, y eso tiene una gran cantidad de ventajas. Lo primero es que nos ahorra un trabajo grande de conceptualización y lo segundo es que facilita la creación de herramientas y formatos para intercambiar modelos. Si llegamos a un banco, por ejemplo, a construir su modelo en la situación actual, al contar con un modelo conceptual de banca ya desarrollado (con los conceptos que son comunes a todos los bancos), estaremos ahorrando mucho trabajo.

Considere ahora el dominio de los procesos de negocio. Lo que hizo BPMN (que es prácticamente un estándar) fue definir un modelo conceptual (con conceptos como *pool*, *lane*, *gateway*, *task*), de manera que cuando llegamos a una empresa a hacer el levantamiento y modelado de procesos lo podemos usar como base. Al hacerlo, encontramos herramientas de edición de modelos y formatos de intercambio entre herramientas. Lo mismo sucede con *ArchiMate*, un modelo conceptual para representar empresas y su componente de tecnología. Un metamodelo se convierte en un estándar cuando existe un acuerdo sobre los conceptos de un dominio, los nombres que se les asigna y sobre su estructura. Tenemos ejemplos como *soaML* (*Service oriented architecture Modeling Language*), *sysML* (*Systems Modeling Language*), *UML* (*Unified Modeling Language*), *PPDM data model* (para el sector petrolero) y *HL7* (para el sector de salud).

3. Construcción de metamodelos

En esta sección vamos a tratar algunos temas metodológicos y a presentar algunas técnicas de metamodelado, lo mismo que algunas recomendaciones que pueden ser de mucha utilidad. Vamos a utilizar un caso de estudio para ilustrar las etapas de construcción de un metamodelo, más que solo revisar el metamodelo resultante.

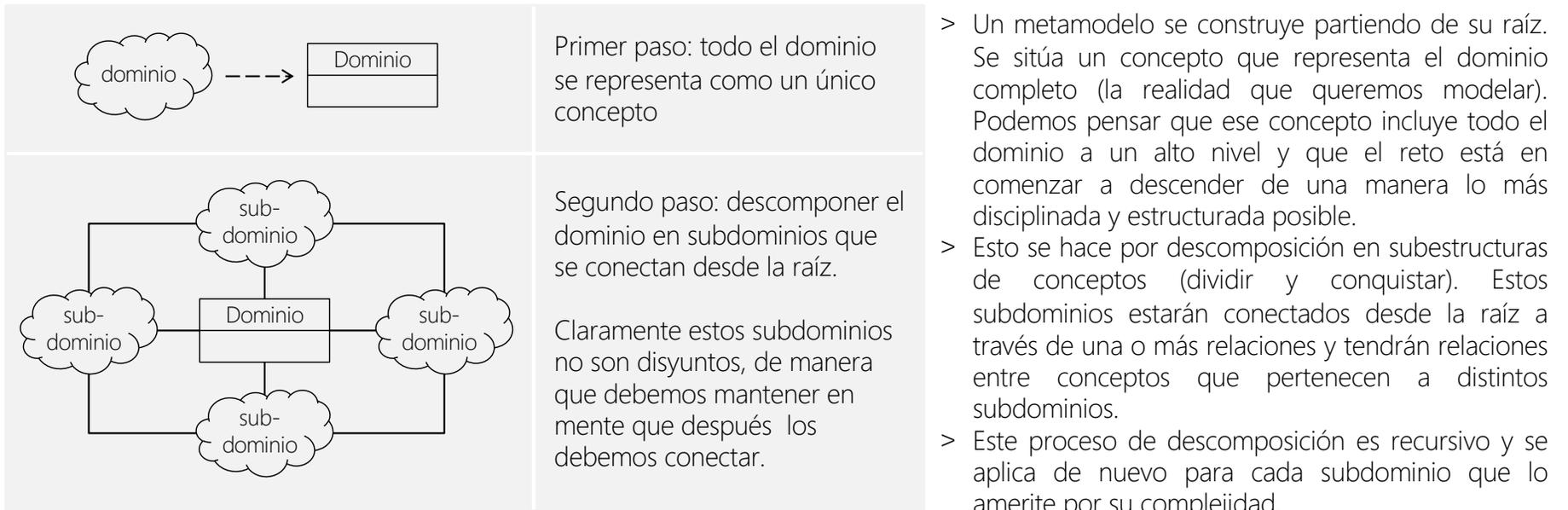


Fig.12 – Pasos iniciales en la estructuración de un metamodelo

Para ilustrar el proceso de construcción de un metamodelo, vamos a utilizar el caso de una empresa (RappiAlpes) que lleva a domicilio los productos que ofrece un conjunto de tiendas. Para esto utiliza un grupo de repartidores que se inscriben usando una aplicación móvil y a quienes la empresa les paga por cada pedido entregado. Los clientes por su parte descargan la aplicación en su dispositivo móvil, se inscriben en la plataforma y dan los datos de una tarjeta de crédito, a través de la cual se hacen todos los cobros. Por cada pedido la empresa le cobra al cliente un valor fijo, que se suma al valor que se carga a la tarjeta de crédito.

Cuando el cliente hace un pedido, la empresa contacta la tienda y le pasa la información de los productos que desea el cliente, de manera que los pueda preparar para que los recoja un repartidor. La empresa carga a la tarjeta de crédito del cliente el monto del pedido y hace inmediatamente una transferencia electrónica a la tienda por el valor del pedido menos la comisión que cobra. La empresa localiza un repartidor que se encuentre cerca del domicilio del cliente y le asigna la tarea. El repartidor debe informar el momento en el que recoge el pedido en la tienda y el momento en el que lo entrega. El repartidor va atendiendo una a una las entregas que la empresa le asigna.

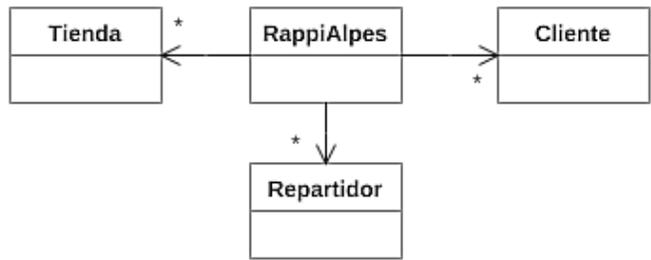
Estructuración de un metamodelo

1



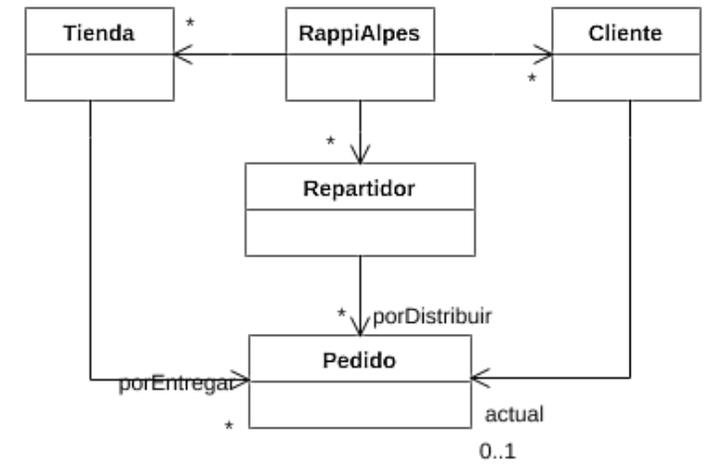
- > El metamodelo comienza con un concepto que representa la empresa entera, el cual puede o no tener atributos (en este caso no necesita).
- > De este concepto solo tendremos una instancia en el modelo.

2



- > Luego aparecen tres entidades (o conceptos o clases o tipos, todos son sinónimos) que representan la base de tres subdominios: todo lo que tiene que ver con los clientes, todo lo que tiene que ver con las tiendas y todo lo que tiene que ver con los repartidores.
- > Agregamos las relaciones con cardinalidad múltiple hacia cada uno de los conceptos que consideramos la base de un subdominio. Por ahora no nos vamos a preocupar de los nombres de los roles o de las relaciones que pueden existir entre los tres subdominios.
- > Recuerde que iremos refinando y completando el metamodelo a medida que vamos descendiendo en detalles.
- > Es importante que los nombres que usemos para los conceptos correspondan al vocabulario que se usa en la empresa.

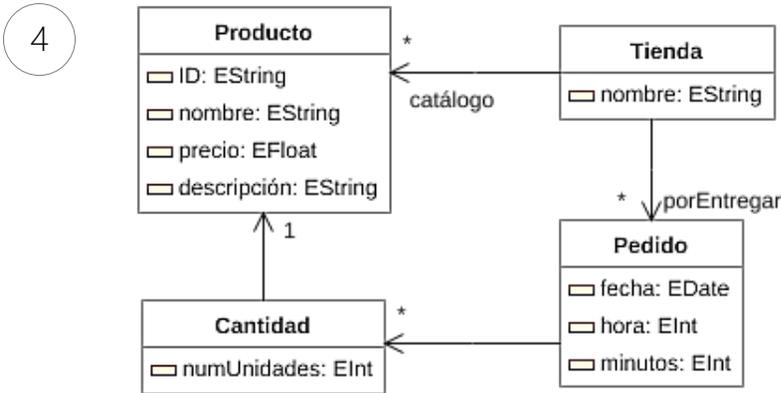
3



- > Aparece el concepto de "Pedido" (los productos que quiere comprar un cliente a una tienda), que conecta los subdominios identificados y que se encuentra en el siguiente nivel, entendiendo nivel como la distancia que existe desde la raíz hasta el concepto.
- > Por supuesto que todavía nos faltan otras relaciones, pero vamos a comenzar con las que son evidentes: un cliente tiene 0 o 1 pedido actual, un repartidor tiene múltiples pedidos por distribuir y una tienda tiene varios pedidos preparados, listos para ser entregado al repartidor.
- > Estas son las 5 principales clases del metamodelo y lo que debemos hacer ahora es refinar los 3 subdominios, sabiendo que la estructura del centro ya está completa. Esta primera etapa de estructuración es crítica. Ya el resto es más fácil.
- > Los atributos también aparecerán más adelante junto con otros conceptos que incluyen mayor detalle.
- > No olvide que el metamodelo se debe leer desde la raíz y que las relaciones son estructurales y no funcionales.

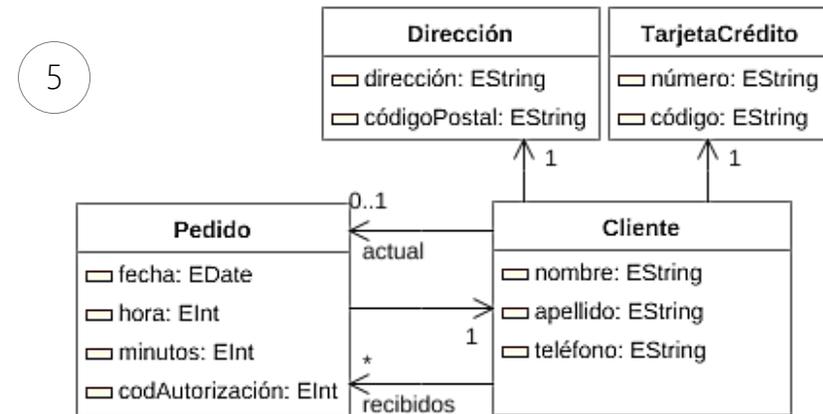
Estructuración de un metamodelo

- > Vamos ahora a detallar el concepto de Tienda y a revisar la relación que existe con la clase Pedido.



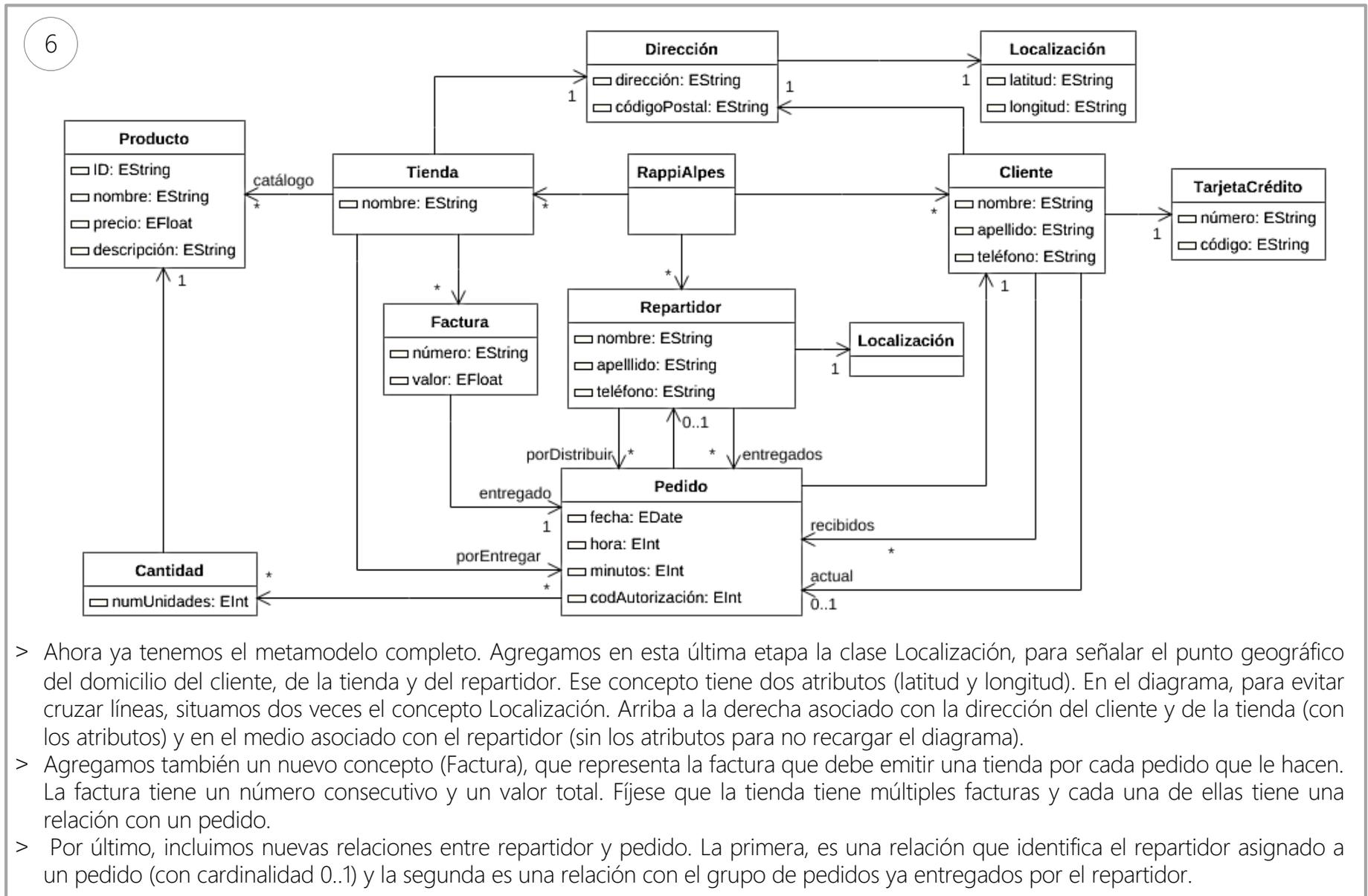
- > Lo primero es que agregamos algunos atributos, con un nombre y un tipo (EString, EInt y EFloat). La "E" al comienzo del tipo la agrega el editor de metamodelos que utilizamos, pero no es necesaria.
- > Una tienda tiene un grupo de productos a la venta (ese rol lo llamamos "catálogo").
- > La clase "Cantidad" corresponde a un patrón que vamos a encontrar en casi todos los metamodelos y es la solución que se utiliza cuando un atributo (numUnidades) no puede pertenecer a ninguna de las dos clases que están conectadas por una relación (Pedido y Producto). El número de unidades de un producto en un pedido es un atributo de la relación y se debe situar en una clase intermedia ("Cantidad").
- > Piense por ejemplo en la relación entre las clases Pared y Puerta. La localización de la puerta en la pared no puede ser un atributo de ninguno de los dos conceptos, por lo que se debe crear la clase "Localización" en la mitad de las dos para conectarlos.

- > Revisemos ahora la relación entre Cliente y Pedido:



- > Incluimos los atributos de la clase Pedido. Adicional a la fecha del pedido y a la hora (con minutos), agregamos un atributo con el código de autorización del banco para cargar el valor de esta compra a la tarjeta de crédito del cliente.
- > Definimos un nuevo concepto para la tarjeta de crédito (con sus respectivos atributos) y un nuevo concepto con la dirección del cliente. Ambos conectados con relaciones de cardinalidad 1. Por supuesto esto es una simplificación, y en un caso real el cliente podría tener múltiples tarjetas y direcciones.
- > Entre Cliente y Pedido agregamos dos nuevas relaciones: la primera, es una relación que permite ir desde el Pedido hacia el Cliente. Esa navegación la necesitamos, porque todo pedido debe saber cuál es el cliente que lo hizo. La segunda es una relación que contiene todos los pedidos hechos por un cliente y que ya le fueron entregados. El nombre del rol de esa relación es "recibidos".

Estructuración de un metamodelo



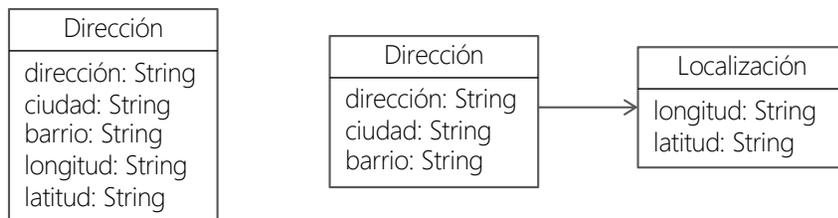
Técnicas, recomendaciones y preguntas frecuentes

En esta parte vamos a presentar algunos aspectos prácticos que apoyan el proceso de construcción de un metamodelo. Le vamos a asignar un identificador y un nombre corto a cada punto, para que sea fácil referirse a ellos. Eso va a facilitar la discusión entre arquitectos.

P1 - ¿Atributo o concepto?

Un atributo debe ser visto como una característica de un concepto que se puede representar a través de un tipo simple de dato.

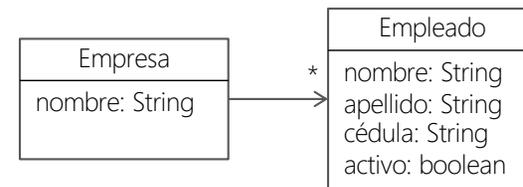
- > Si el valor de un atributo se puede calcular a partir de otro atributo, se debe eliminar de la clase. Por ejemplo, en la clase Persona, el atributo "edad" se puede calcular a partir del atributo "fecha de nacimiento", luego el primero no se debe incluir en la clase.
- > Si para modelar una característica se necesita un grupo de valores, se debe identificar el concepto detrás de cada valor individual e incluirlo en el metamodelo como un nuevo concepto, con una relación de cardinalidad múltiple.
- > Si un conjunto de atributos corresponde a información que se puede considerar un concepto en sí mismo, se debe extraer de allí, como se ilustra en el siguiente ejemplo:



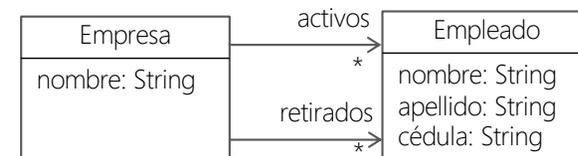
- > Si un atributo (barrio, por ejemplo) corresponde a un concepto que resulta importante para el dominio (cuyo término se utiliza con frecuencia en la descripción de los elementos de la realidad) se debe analizar la posibilidad de convertirlo en una clase y conectarlo con una relación.

P2 - ¿Atributo o agrupación?

Si un atributo representa una clasificación (o estado), se debe analizar si dicha clasificación se debe reflejar en la estructura del metamodelo y reemplazar así el atributo. Esconder estructura detrás de atributos es un error. Tomemos como ejemplo una empresa que tiene empleados. El hecho de que el empleado se encuentre activo o se haya retirado de la empresa se puede modelar con un atributo de la siguiente manera.



Eso por supuesto es un error, porque está escondiendo que hay dos grupos de empleados (o incluso dos conceptos distintos), como se sugiere a continuación:



También se debe considerar la opción de incluir dos conceptos distintos ("EmpleadoActivo" y "EmpleadoRetirado") con relaciones de cardinalidad múltiple hacia ellos, sobre todo si tienen atributos diferentes (por ejemplo, si para un empleado retirado se debe incluir un atributo con la fecha de retiro). En cualquier caso, esos atributos que esconden estructura se deben suprimir.

Técnicas, recomendaciones y preguntas frecuentes

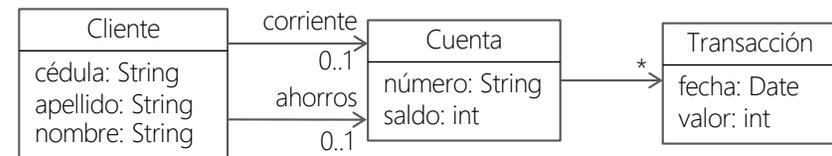
P3 - ¿Conceptos o actores del dominio?

Una confusión que se presenta muy a menudo es el hecho de que un concepto del dominio se llame igual a un actor del dominio (alguien que en la realidad tiene la responsabilidad de hacer algo). Esto nos lleva a incluir erróneamente relaciones que no son estructurales sino funcionales. Consideremos el caso de la Universidad: un profesor es un concepto (un elemento de la realidad con relaciones hacia otros conceptos) y como tal debe hacer parte del metamodelo. Pero en la realidad existe también el actor "profesor", que dicta clases y pone notas a los estudiantes. El actor desarrolla actividades, mientras el concepto no hace nada. El concepto solo es una estructura de información, que en ningún caso representa una acción o una actividad. Con las relaciones sucede lo mismo. En ningún caso se debe incluir una relación entre la clase Profesor y la clase Nota, con el argumento de que "el profesor pone notas". En el metamodelo es muy posible que no exista una relación directa entre las clases Profesor y Nota, porque la nota es un concepto relacionado con un Estudiante, aunque en la realidad sea el actor profesor el que la asigne.

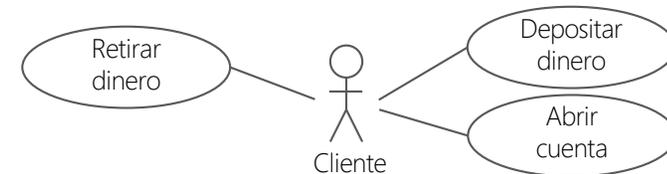
Algunos síntomas de que un concepto es en realidad un actor:

- > En la explicación del concepto aparecen acciones (verbos) o responsabilidades. Siempre se debe tener en cuenta que un concepto "no hace nada".
- > Un concepto sin atributos. No es que esto sea en sí mismo un error, pero esos conceptos son sospechosos de que en realidad sean actores.
- > Una relación que se explica como una acción, como parte de una secuencia de sucesos (como si fuera un proceso) o como una reacción ("entonces el Profesor envía...").

Veamos otro ejemplo. Pensemos ahora en el cliente de un banco. Es claro que el cliente es un concepto con atributos y con relaciones hacia las cuentas y sus movimientos. Algo como lo que se ilustra a continuación.



Pero para el banco el cliente también es un actor, que realiza actividades como hacer un depósito, abrir una cuenta o retirar dinero. Si utilizamos la sintaxis de los diagramas de casos de uso de UML se vería de la siguiente manera:



El primero es un modelo conceptual, mientras el segundo es un modelo funcional. Lo que no podemos hacer es confundirlos o mezclarlos. El primer error sería tratar de incluir en el metamodelo un concepto como "RetirarDinero". Recuerden que los conceptos nunca corresponden a verbos. El segundo error sería intentar incluir relaciones para representar las actividades que ejecuta el actor. Siempre deben tener en mente que se puede representar el resultado de la acción, pero no la acción misma. En el modelo funcional aparece la acción "Depositar dinero", mientras que en el metamodelo aparecen los conceptos involucrados en esa acción ("Cuenta" y "Transacción").

Técnicas, recomendaciones y preguntas frecuentes

P4 - ¿Cómo modelar el ciclo de vida de los elementos?

Cuando tenemos elementos que evolucionan en el tiempo, el metamodelo debe poder representar las etapas de dicha evolución. Un pedido, por ejemplo, puede estar hecho por parte del cliente, luego, empacado para entrega, después, en proceso de distribución y, finalmente, entregado. Ya vimos que no es conveniente agregar una variable de estado dentro del concepto, como lo haríamos si estuviéramos diseñando la base de datos para manejar esa información. Aquí hay varias posibilidades:

- > Definir un concepto para cada estado del pedido y colocar cada concepto con los atributos y relaciones correspondientes. Tendríamos algo como Pedido, PedidoPreparado, PedidoEnDistribución y PedidoEntregado. Para eso podemos utilizar relaciones de herencia desde un concepto básico. La dificultad que se presenta es que el metamodelo se vuelve más complejo por la cantidad de relaciones y de conceptos.
- > Definir un único concepto Pedido y agregar relaciones de cardinalidad múltiple hacia esa clase, para crear grupos con los pedidos que se encuentran en cada estado particular. Eso fue lo que hicimos en el caso de RappiAlpes. Tal vez vale la pena regresar y ver la solución planteada. Nos aparecen muchas relaciones alrededor del concepto, algunas con cardinalidad 0..1 para indicar que en algunos estados del pedido existen relaciones que no se han establecido (antes de que el pedido esté listo para distribución, no tiene asignado un repartidor).
- > Definir una mezcla de las dos aproximaciones anteriores. Esto se debe hacer cuando las estructuras de los conceptos varían fuertemente con el cambio de estado. Aparecen, por ejemplo, nuevos atributos o nuevas relaciones, que no se logran manejar de manera natural con un único concepto.

P5 - ¿Cómo definir la dirección de una relación?

Por definición, existe una relación desde una clase C1 hacia una clase C2 si para definir el estado de una instancia de C1 se necesita una instancia de C2. En ese caso la relación tiene una dirección (de C1 a C2), y podemos decir que C1 conoce a C2, o que desde C1 se puede navegar en el metamodelo hacia C2. Pero hay casos en los cuales necesitamos también que C2 tenga una relación hacia C1. Esa segunda relación (hacia atrás) solo la debemos agregar si de verdad es indispensable. Vamos entonces a tratar de caracterizar las relaciones:

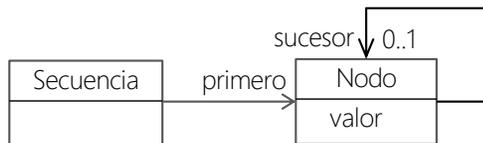
- > Primero vamos a tener el caso en el que hay una relación de C1 hacia C2, si C2 es una parte de C1 (C2 es un siguiente nivel de detalle de C1). Es la más común y se denomina relación *isPartOf*. En ese caso la dirección de la relación es clara y es excepcional que la parte deba conocer el todo. En el caso de la Universidad, un estudiante tiene notas, y las notas son parte del estudiante. En un banco, un cliente tiene cuentas, y estas cuentas son parte del cliente.
- > Después aparecen las relaciones entre conceptos que se pueden considerar del mismo nivel (ninguno es parte del otro). En el caso de la Universidad tenemos los conceptos de Estudiante y de Curso, donde ninguno es parte del otro. En ese caso es más usual que encontremos relaciones en las dos direcciones (un curso tiene estudiantes inscritos y un estudiante está tomando algunos cursos).
- > En general, el criterio es garantizar que todo concepto C1 que deba conocer otro concepto C2 del metamodelo, encuentre la manera de navegar por las relaciones existentes para llegar del uno al otro. En el caso de RappiAlpes, si no agregamos la relación inversa desde pedido hacia repartidor, el cliente no podría conocer a qué repartidor le asignaron a su pedido.

Técnicas, recomendaciones y preguntas frecuentes

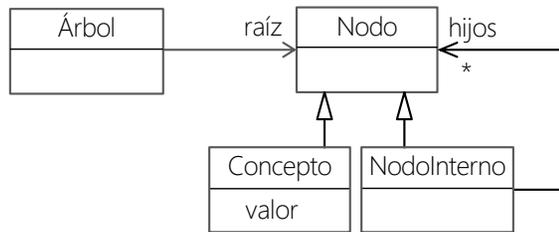
P6 - ¿Se pueden identificar patrones en los metamodelos?

En los metamodelos vamos a encontrar varias estructuras típicas que mostramos a continuación:

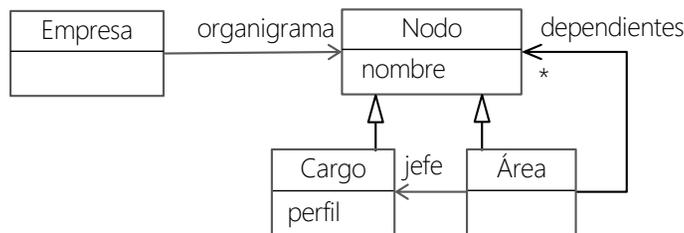
- > El primer patrón es el de una secuencia, en el cual el concepto tiene una relación hacia sí mismo con cardinalidad 0..1 para indicar que puede o no tener un sucesor:



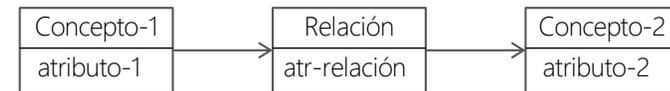
- > El segundo patrón (llamado *Composite*) permite representar una jerarquía de conceptos. En ese caso introducimos una clase llamada *Nodo*, que tiene dos subclases relacionadas a través de herencia:



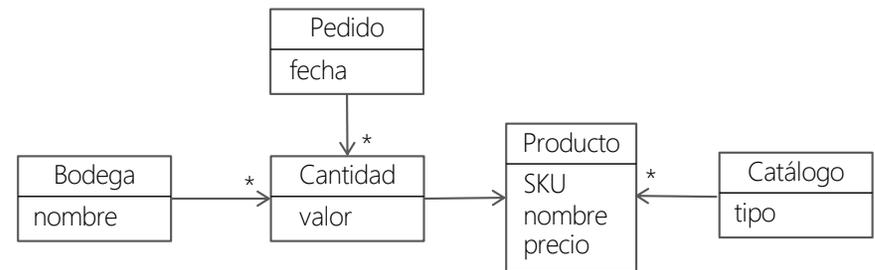
Como se muestra en el siguiente ejemplo:



- > El tercer patrón nos permite representar relaciones que tienen atributos (la información no puede estar en ninguna de las dos clases relacionadas), por lo que deben materializar la relación como otro concepto.



Para ilustrar este último patrón considere el siguiente fragmento de metamodelo:



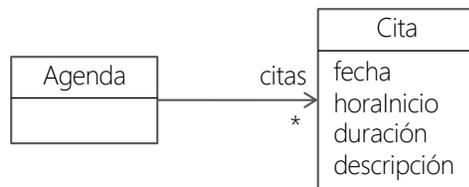
Por un lado tenemos una bodega y un pedido, cada uno de los cuales contiene una cierta cantidad de productos. De cada producto el metamodelo tiene un identificador único (SKU), un nombre y un precio. Por el otro lado tenemos el catálogo de productos de la tienda. La pregunta es ¿dónde situar la cantidad de productos que hay en la bodega o en el pedido? Es claro que no puede estar en la clase *Producto* (no es un atributo propio de ese concepto que representa un tipo de producto de la tienda). En ese caso, aplicando lo que se sugiere en este patrón debemos crear un concepto intermedio (*Cantidad*) que contiene los atributos que se necesitan (*valor*).

Técnicas, recomendaciones y preguntas frecuentes

P7 - ¿Hay metamodelos mejores que otros?

Lo primero que debemos aclarar es que hay más de una manera de conceptualizar una misma realidad. Eso quiere decir que, para un mismo objetivo, podemos construir varios metamodelos distintos. Veamos algunas características que pueden hacer mejor un metamodelo que otro:

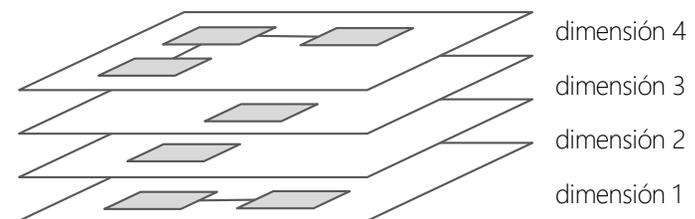
- > Los nombres de los conceptos y los nombres de los roles. Un metamodelo puede ser confuso si se utilizan los nombres inadecuados.
- > Lo simple es mejor que lo complejo. Un metamodelo debe ser tan simple como sea posible, sin perder por supuesto información esencial. Cada concepto innecesario o cada relación que no sea indispensable contamina el resultado.
- > La forma en que se dibuja es fundamental. Un metamodelo expresado con líneas cruzadas, o con líneas oblicuas, o donde no se logre apreciar simetría y armonía es inapropiado. Debe ser fácil localizar la raíz del metamodelo y debe ser fácil identificar los niveles. Esto no es sencillo, pero es importante que el cerebro del arquitecto lo pueda interpretar con el menor esfuerzo posible. Recuerden que una de las grandes habilidades de un arquitecto es hacer parecer simple (y ordenado) algo que es complejo. Dura tarea.
- > Copiar fielmente la estructura de la realidad. Esto quiere decir que en el proceso de conceptualización debemos incluir bien la estructura de la realidad. ¿Qué opinan de este metamodelo?



P8 - ¿Cómo manejar la complejidad de un metamodelo?

Se dice que un metamodelo es grande si tiene muchos conceptos y que es complejo si sus conceptos están muy relacionados entre sí. Si debemos manejar un metamodelo con esas características, podemos usar las siguientes aproximaciones:

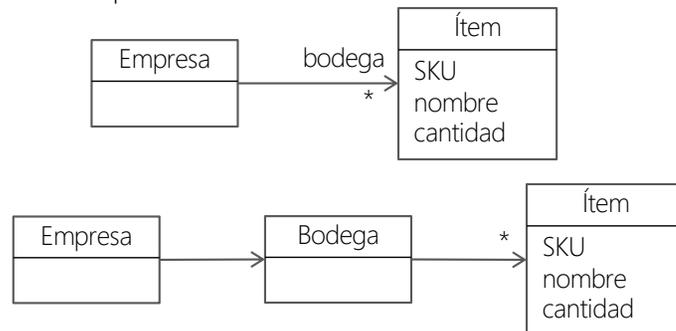
- > Si es solo un problema de dibujo porque no hay manera de evitar que se crucen las relaciones, podemos usar la técnica ilustrada en el ejemplo de RappiAlpes, en donde repetimos el concepto "Localización" en el metamodelo. Es mejor tener un concepto dibujado dos veces que tener una línea cruzada.
- > Si de todas maneras se van a cruzar dos líneas en el metamodelo, se debe buscar la manera en que no se introduzca una ambigüedad en la lectura. Que sea claro cuál línea pasa por encima de la otra.
- > Si el problema es de verdad difícil, debemos descomponer el metamodelo en dimensiones, usando para eso puntos de vista, tema tratado en la sección 5 de esta lectura. Cada dimensión contiene una parte del metamodelo (siguiendo algún criterio definido), siempre con la misma raíz (para que se puedan componer), y con conceptos en común que se pueden extender en otras dimensiones. Para esto se define una dimensión de base y luego se van sumando capas como se sugiere en la siguiente figura:



Técnicas, recomendaciones y preguntas frecuentes

P9 - ¿Es un concepto o solo una agrupación?

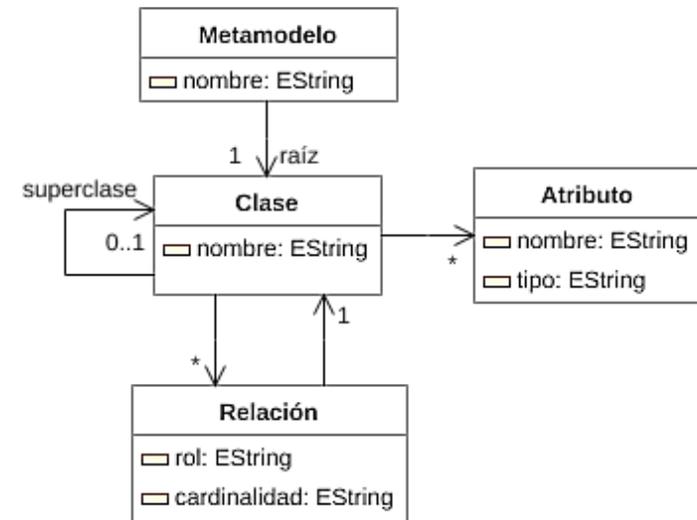
Hay algunos términos que aparecen mencionados en los negocios, pero que no es claro si se deben materializar como un concepto o si son solo una agrupación de otros conceptos que lo constituyen. Ese es el caso de términos como bodega, catálogo (de productos), contabilidad, inventario, entre otros. Tomemos como ejemplo el primero de esos términos y revisemos dos posibles conceptualizaciones:



- > En el primer caso "bodega" es el nombre que se asigna al grupo de ítems que se encuentran almacenados (SKU es el identificador único de un elemento: *Stock Keeping Unit*).
- > En el segundo caso "Bodega" se materializa como un concepto que conecta la empresa con el grupo de ítems.
- > Cualquiera de los dos metamodelos es válido, pero a no ser que la bodega requiera atributos (como por ejemplo una dirección o una capacidad o algo que se necesite representar) es preferible la primera conceptualización porque es más simple. Siempre debemos privilegiar la sencillez.
- > Piense cómo sería el caso de una contabilidad (un grupo de movimientos contables), o un catálogo (un grupo de productos) o un garaje (un grupo de vehículos).

P10 - ¿Es posible hacer el metamodelo de un metamodelo?

Es posible construir el metamodelo de cualquier dominio. El metamodelo de un metamodelo sería el siguiente:



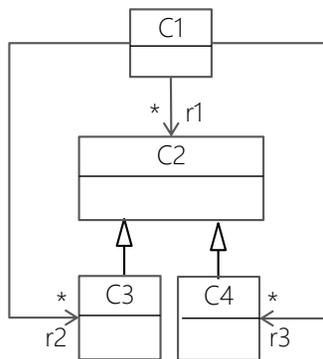
- > En la raíz tenemos la clase "Metamodelo", que tiene un único atributo que es su nombre. De allí parte una relación hacia la clase raíz.
- > Una clase tiene un conjunto de atributos, cada uno con un nombre y un tipo.
- > De una clase sale un conjunto de relaciones que llevan a otra clase. Cada relación tiene un nombre de rol y una cardinalidad.
- > Por último, una clase puede o no tener una super clase, a la cual está conectada por una relación de especialización (herencia).

Técnicas, recomendaciones y preguntas frecuentes

P11 - ¿Cómo se interpretan la herencia y los agrupamientos?

En los metamodelos se incorpora la terminología de los diagramas de clases de UML, de manera que decimos que C2 es la superclase de C3, si existe una relación de especialización (herencia) desde C3 hacia C2. En ese caso también decimos que C3 es una subclase de C2.

Considere ahora el siguiente ejemplo:

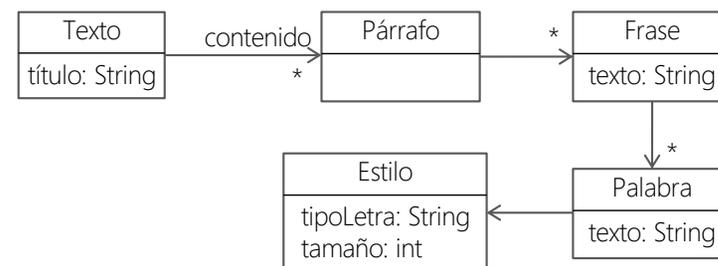
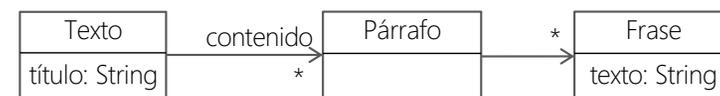


- > C2 es la superclase de C3 y C4
- > C3 y C4 son subclases de C2

- > El metamodelo afirma que el grupo r1 está compuesto por instancias de C2, de C3 y de C4 sin diferenciarlas. Se dice que es un agrupamiento polimorfo, porque hay instancias de distintos tipos a la vez.
- > Por su parte el grupo r2 contiene únicamente instancias de C3, mientras el grupo r3 contiene únicamente instancias de C4.
- > Si queremos tener un grupo con solo instancias de C2, debemos expresarlo utilizando una restricción, porque no hay manera de hacerlo usando solo el diagrama.
- > Recuerde que la relación de herencia solo se justifica cuando hay una ganancia clara en la simplicidad del metamodelo (es más fácil de interpretar).

P12 - ¿Cómo se ve la altura en un metamodelo?

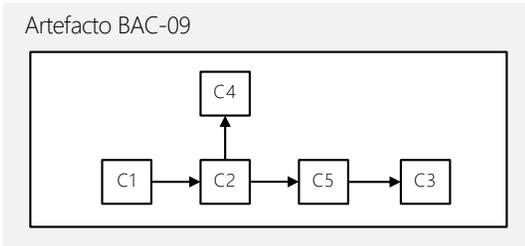
Al comienzo de esta lectura introdujimos el concepto de alcance vertical (o altura) de un modelo, que corresponde al nivel de detalle de la realidad que debemos recoger en el modelo. Veamos 4 ejemplos del mismo metamodelo a distintas alturas.



Como se puede apreciar, cada metamodelo da un poco más de detalle sobre lo que es un texto. Dependiendo del uso que se le vaya a dar a los modelos que se creen con este metamodelo se necesita bajar más (incluir más información) o menos. Los cuatro metamodelos son correctos, pero posiblemente están enfocados a usos distintos.

Documentación de un metamodelo

Metamodelo:



- > El primer artefacto es el diagrama de clases que representa el modelo conceptual.
- > Es fundamental que el diagrama sea claro, lo más simple posible y que no tenga líneas torcidas o cruzadas entre ellas.
- > Si el metamodelo es muy grande, es conveniente partirlo en dimensiones, cada una de ellas contenida en su propio artefacto (BAC-09).
- > Existen múltiples herramientas en el mercado para construir diagramas de clases. En lo posible se debe evitar el color de relleno de las clases. No hay como el fondo blanco.

Catálogo de conceptos:

Artefacto BAC-10

Clase	Descripción	Sinónimos

- > En este catálogo se describe cada una de las clases del metamodelo.
- > En algunos casos es conveniente registrar en el catálogo los sinónimos asociados con el concepto que representa la clase. Esto es común en algunas empresas, en donde las distintas unidades organizacionales utilizan un vocabulario diferente para referirse a conceptos comunes.
- > En el metamodelo solo aparece el nombre seleccionado para representar el concepto.

Catálogo de atributos y relaciones:

Artefacto BAC-11

Clase	Atributo / Relación	Descripción

- > Este artefacto se utiliza para describir los atributos de una clase, lo mismo que las relaciones que salen de ella. En el catálogo se registra el nombre de la clase, el nombre del atributo o de la relación (si el rol tiene nombre) y una breve descripción que indica su contenido y su interpretación (si no es evidente).
- > Si el rol no tiene nombre se debe incluir el nombre de la clase hacia la cual se dirige la relación. Es conveniente también registrar en este catálogo la cardinalidad de la relación, si esta no es 1.

Catálogo de restricciones:

Artefacto BAC-12

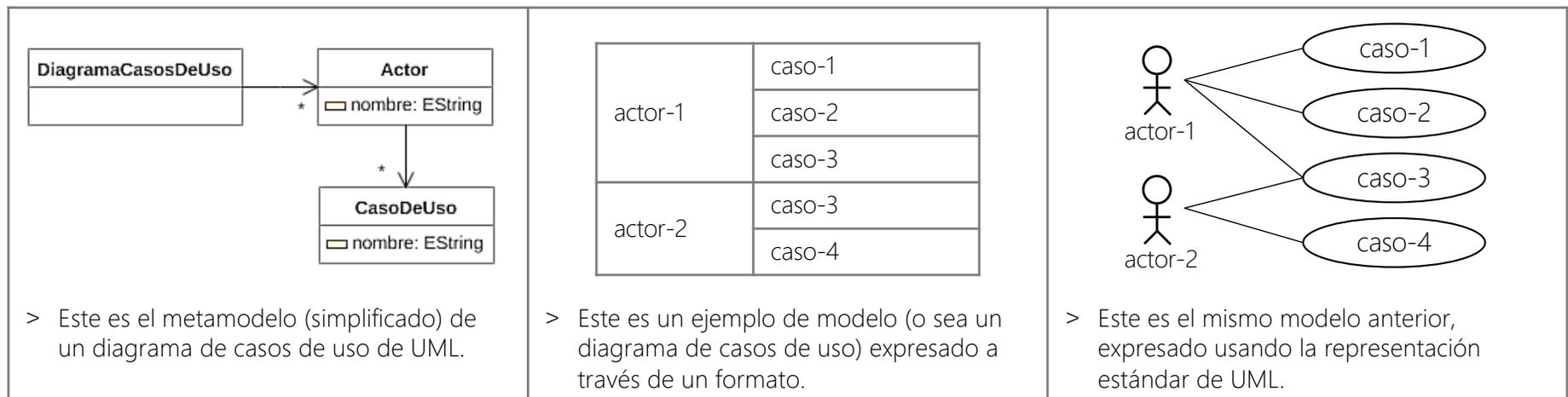
ID	Descripción

- > El catálogo de restricciones contiene una descripción en lenguaje natural de cada una de las restricciones que deben cumplir los modelos, para que se consideren conformes con el metamodelo.
- > Las restricciones no se colocan en un orden en particular, pero sería buena idea agruparlas por concepto o grupo de conceptos involucrados.
- > Cada restricción tiene un identificador único asociado

4. Diseño de lenguajes de dominio

Cerrado el tema de los metamodelos, ya tenemos una manera de estructurar los conceptos presentes en la realidad y podemos empezar a pensar en la forma de expresar un modelo. Por ahora sabemos que debemos hacerlo usando un conjunto de artefactos (que se deben diseñar a la medida del modelo) y que la manera de comunicarlo es crítica para que el modelo cumpla con su función.

Básicamente, podemos clasificar los lenguajes para expresar un modelo en dos grandes grupos: los primeros corresponden a lenguajes basados en formatos (formas estructuradas que contienen lenguaje natural) y los segundos, a lenguajes basados en representaciones gráficas. Veamos para comenzar un ejemplo que ilustre esta clasificación:

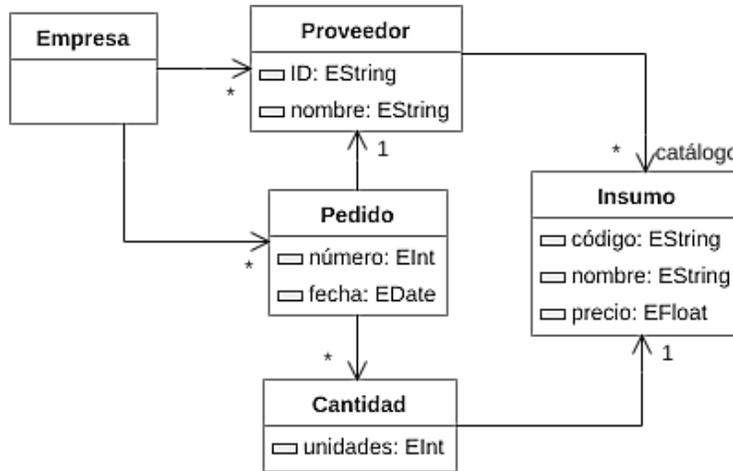


- > Los formatos son un lenguaje muy simple, que se deriva directamente del metamodelo. Por cada relación de cardinalidad múltiple, aparece un grupo de instancias (en el ejemplo, aparece una fila por cada actor). Por cada atributo de la clase (en el ejemplo sólo está el atributo nombre), aparece una columna para almacenar el respectivo valor.
- > Luego, el metamodelo dice que cada actor tiene un grupo de casos de uso. Eso hace que por cada fila de actor, aparezcan filas internas para almacenar los casos de uso. De nuevo, como solo hay un atributo (nombre), solo se tiene una columna. En la siguiente página aparece un ejemplo más completo.

- > Los formatos se representan muy bien en hojas de cálculo, por lo que no hay necesidad de desarrollar herramientas especiales, y esa es una gran ventaja.
- > Es muy posible que se deban utilizar varias hojas dentro del mismo archivo, las cuales se deben encadenar entre sí como se haría en una base de datos, utilizando algún tipo de llave o identificador único.
- > Los formatos tienen otra ventaja muy grande y es que sirven para recoger la información directamente de las fuentes. Una vez hecho el metamodelo, se les pueden construir y pasar los formatos en una hoja de cálculo y pedir que lo llenen.

Formatos como lenguaje de los modelos

Ejemplo



- > El metamodelo incluye conceptos referentes a los insumos que utiliza una empresa y que venden unos proveedores.
- > El formato para representar los modelos conformes con el metamodelo consta de dos tablas: en la primera aparecen los proveedores con el catálogo de insumos que venden. Fíjese la manera como se utilizan los nombres de las clases y los nombres de los atributos como parte del formato.
- > En la segunda tabla aparecen los pedidos. Un pedido tiene el ID de un proveedor (es una llave que cruza hacia la otra tabla) y una cantidad de un insumo, que se identifica con otra llave. Las llaves que cruzan tablas permiten representar las relaciones.

Proveedor		Insumo (catálogo)		
ID	nombre	código	nombre	precio
P1	Tienda-1	C1	Producto-1	\$1
		C2	Producto-2	\$2
		C3	Producto-3	\$3
P2	Tienda-2	C1	Producto-1	\$4
		C4	Producto-4	\$5
P3	Tienda-3	C1	Producto-1	\$6
		C5	Producto-5	\$7

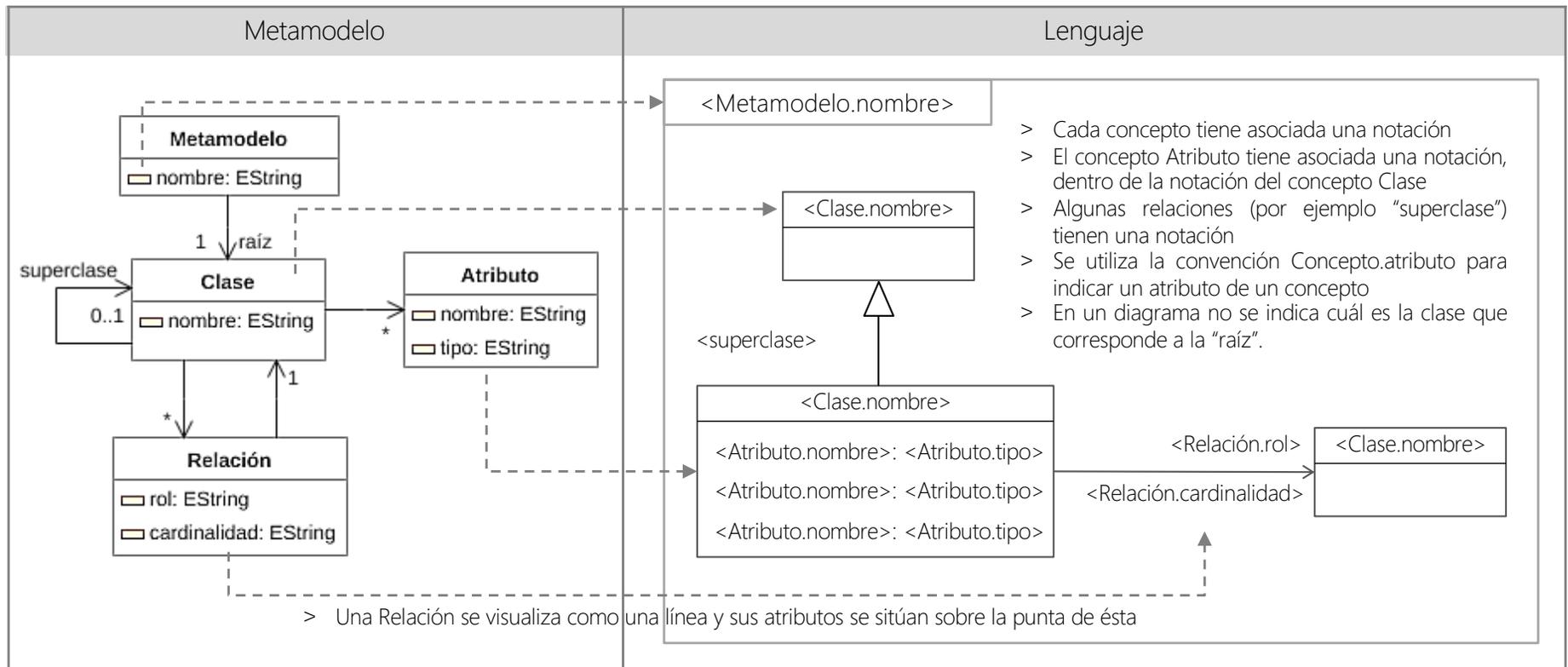
Pedido		Proveedor	Cantidad	Insumo
número	fecha	ID	unidades	código
orden-1	fecha-1	P1	5	C1
			7	C2
			2	C3
orden-2	fecha-2	P2	3	C1
			4	C4
orden-3	fecha-3	P3	2	C1
			10	C5

Una vez representados los modelos a través de formatos, es posible convertirlos de manera automática a otros lenguajes, almacenarlos en una base de datos o hacer todo tipo de análisis y cruces de información. En el mercado hay herramientas que crean los esquemas de la base de datos a partir de un metamodelo. La pregunta que surge, viendo el ejemplo, es hasta dónde vale la pena construir los modelos, sobre todo si contienen información que es muy volátil o implica grandes volúmenes de información. La respuesta es la de siempre: depende de para qué se va a utilizar el modelo. En algunos casos, incluso, puede ser más importante el metamodelo que el modelo, porque lo importante es la comprensión del dominio más que el estado exacto de la realidad.

Lenguajes gráficos estándar para representar modelos

Los lenguajes gráficos son más intuitivos y son capaces de comunicar información estructural de una manera mucho más efectiva que los formatos. Por supuesto a condición de que el lenguaje esté bien diseñado, tema que trataremos un poco más adelante. Hoy en día ya hay muchos dominios que no solo tienen un modelo conceptual bien definido, sino que cuentan además con un lenguaje gráfico usado de manera masiva. Estamos tan acostumbrados a usar lenguajes como BPMN, UML y ArchiMate, por ejemplo, que se nos olvida que por detrás de ellos hay un metamodelo que representa la conceptualización de un dominio. Las herramientas (editores) de esos lenguajes verifican que los modelos que creamos sean conformes con el metamodelo y nos señalan como un error cualquier cosa que no lo cumpla.

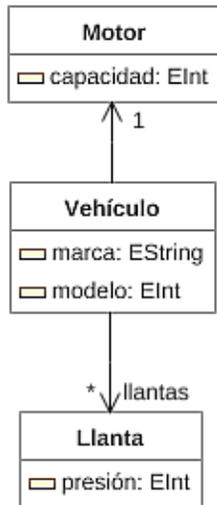
Formalmente, se dice que un metamodelo define la sintaxis abstracta de un dominio (el modelo conceptual), mientras que el lenguaje define la sintaxis concreta (la manera de expresar modelos conformes con el metamodelo). Para esto, un lenguaje debe asociar una notación con los conceptos, atributos y relaciones del metamodelo, tal como se muestra en el siguiente ejemplo para el metamodelo de un metamodelo que vimos en una sección anterior:



Diseño de lenguajes gráficos

También hay dominios que tienen una conceptualización, pero no tienen asociado un lenguaje estándar (por ejemplo BMM) o dominios a los cuales se les debe construir el modelo conceptual y el lenguaje. Veamos un ejemplo muy sencillo de este último caso, en el cual presentaremos un modelo conceptual y cuatro posibles lenguajes asociados, todos mostrando el mismo modelo.

Ejemplo

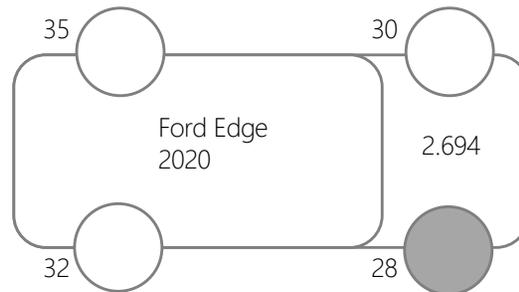


Un vehículo tiene una marca y un modelo (año de fabricación), lo mismo que un motor y un cierto número de llantas. Cada llanta tiene un único atributo que corresponde a su presión de aire. Por su parte el motor tiene una capacidad (cilindrada) en centímetros cúbicos.

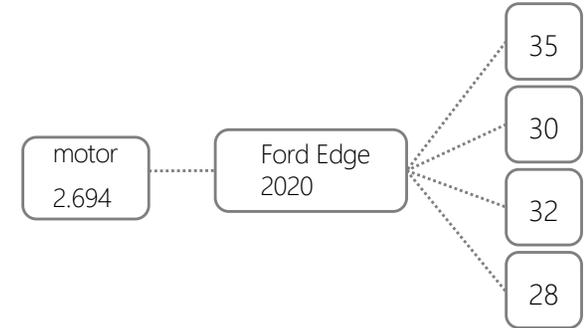
Un modelo conforme con ese metamodelo representa un vehículo con esa información asociada. Debe tener un valor para cada atributo.

marca	Ford Edge
modelo	2020
capacidad motor	2.694
presión llanta 1	35
presión llanta 2	30
presión llanta 3	32
presión llanta 4	28

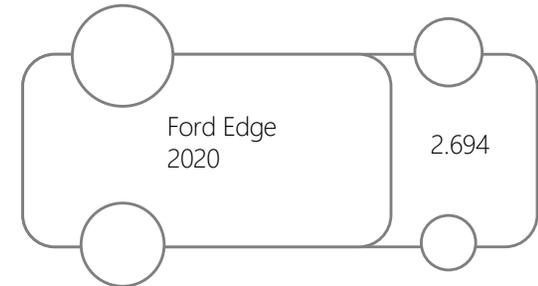
El primer lenguaje planteado corresponde a un formato, que representa un vehículo para el cual se tiene un valor en texto y valores numéricos para cada uno de los atributos. Es un lenguaje muy simple, con muchas ventajas, como vimos al comienzo de esta sección.



El tercer lenguaje intenta reflejar mejor el modelo visual que tiene una persona de lo que es un vehículo y aprovecha el color para mostrar las llantas que tienen una presión baja (por debajo de 30). Sin ser todavía un buen lenguaje, logra comunicar visualmente más información.



El segundo lenguaje planteado sigue la misma idea del diagrama de clases de la página anterior, en el cual cada concepto tiene una representación gráfica y las relaciones entre conceptos se visualizan como líneas entre las instancias. No muy bonito, lo sé.



El cuarto lenguaje es una variante del anterior, en el cual en vez de poner en texto la presión de cada llanta, vamos a utilizar el radio del círculo para indicar el valor del atributo. Es para ilustrar el uso de otros atributos gráficos para comunicar información. Que conste que nunca dije que fuera una buena idea.

Lenguajes gráficos y comunicación visual

- > Aunque la comunicación gráfica es algo de particular importancia en la vida de los arquitectos, solo la trataremos de manera tangencial en esta lectura. La razón es que es un tema muy amplio y es mejor no desenfocarnos del objetivo que nos planteamos al comienzo. Nos vamos a concentrar solamente en las variables visuales, en la manera de especificar un lenguaje y en presentar algunas recomendaciones.
- > Lo primero que debemos señalar es que nuestros lenguajes gráficos están orientados a comunicar información a personas (los destinatarios), y que éstos los van a procesar usando la poderosa capacidad visual con la que cuentan. No olvidemos que casi un cuarto de nuestro cerebro está dedicado al procesamiento visual, algo que logra hacer de una manera muy eficiente. Nuestro objetivo es hablar con este poderoso procesador de la manera más efectiva posible.

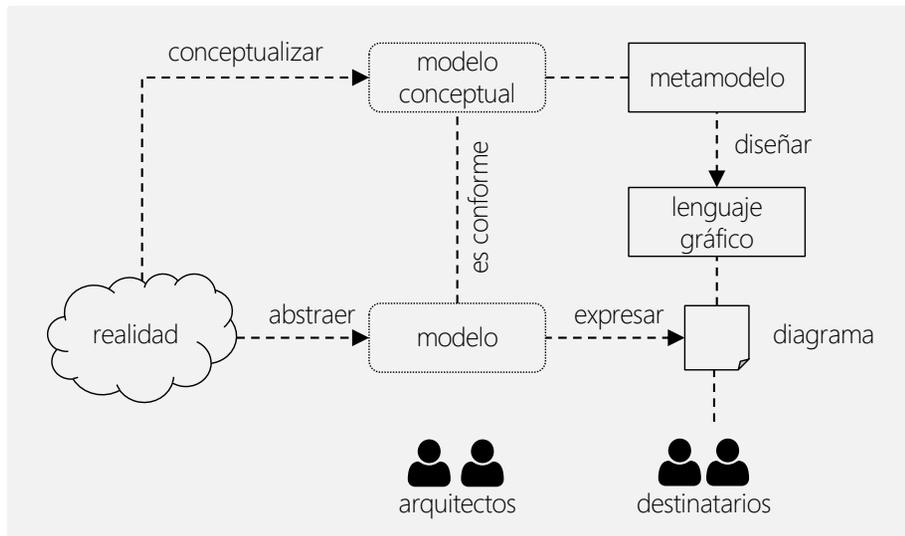


Fig.13 – Papel del lenguaje gráfico en el proceso global

- > Para nuestro cerebro es natural recibir imágenes y convertirlas en modelos. Eso lo hacemos todo el tiempo y es mucho más eficiente haciendo esto, que convirtiendo texto en modelos: lo primero se hace de manera altamente paralela, mientras lo segundo lo hacemos de forma secuencial, interpretando cada carácter que leemos. Es muy conocida (y cierta) la frase que dice que una imagen vale más que mil palabras.
- > Nuestro reto, entonces, es conocer la manera cómo funciona dicho procesador visual y diseñar los lenguajes gráficos de manera que el cerebro reconstruya de manera inmediata los elementos estructurales del modelo y luego sí pase a recoger y completar los detalles. Debemos saber que nuestro cerebro detecta inmediatamente formas, figuras y relaciones, sin hacer ningún esfuerzo. Lo que necesitamos es aprovechar esto para pasar información importante de los modelos que elaboremos como arquitectos.
- > En la figura 13 se ilustran las etapas que existen dentro del proceso de expresión de un modelo, incluyendo ahora el diseño del lenguaje gráfico, el cual se encuentra directamente relacionado con el metamodelo.
- > Se define la **efectividad** de un diagrama (expresado en un lenguaje gráfico) como la velocidad, facilidad y precisión con la cual dicho artefacto es procesado (y entendido) por el cerebro de los destinatarios. Nuestro objetivo es diseñar lenguajes gráficos que permitan construir diagramas efectivos.
- > Hay tres razones por las cuales se puede perder esa efectividad: (1) la conceptualización está mal hecha y se perdieron estructuras del dominio, (2) el lenguaje está mal diseñado y los elementos no logran comunicar bien el modelo o (3) el lenguaje está siendo mal usado y el diagrama no es de buena calidad. Esto último es mucho más frecuente de lo que pensamos, a veces por falta de esfuerzo y a veces por falta de habilidad.

El proceso cognitivo y la efectividad de un diagrama

Aunque son claras las ventajas de los lenguajes gráficos sobre los lenguajes textuales, casi siempre debemos diseñar lenguajes que tengan una mezcla y un equilibrio de los dos. El problema está en que la información gráfica satura rápidamente nuestro procesador visual y es muy posible que necesitemos expresar los detalles del modelo en texto. Veamos cómo es el proceso mental de interpretación de un diagrama, para poder entender esto un poco mejor.

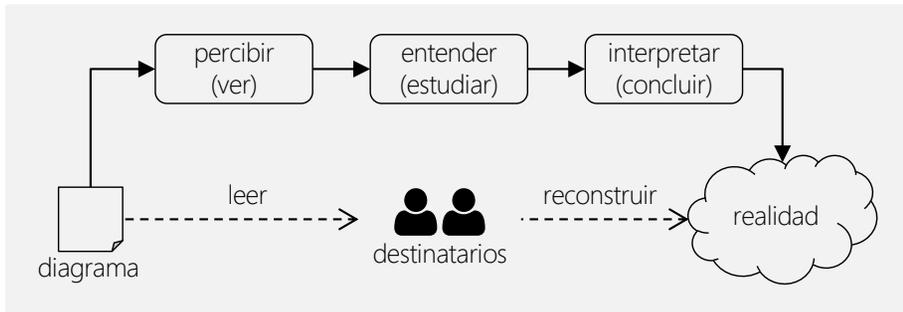


Fig.14 – Proceso mental de interpretación de un diagrama

Cognitive miser - Los seres humanos tenemos una capacidad de procesamiento limitada y debemos aprovecharla de la mejor manera posible para poder enfrentar problemas y entornos complejos. Cualquier gasto innecesario de esta capacidad de procesamiento se debe ver como un desperdicio, que nosotros como arquitectos debemos evitar. Un diagrama mal construido, una presentación mal hecha, una opinión confusa o una entrevista mal estructurada pueden botar a la basura horas de trabajo de un equipo.

Irony of simplicity - La simplicidad es una de las características más difíciles de conseguir en un diagrama. Se requiere mucho esfuerzo y tiempo para lograr que un diagrama, que refleja una realidad compleja, parezca simple y sea fácilmente entendible. Casi siempre un diagrama demasiado complejo refleja falta de trabajo de parte del equipo de arquitectos.

- > La primera etapa la hace automáticamente el cerebro de una manera muy eficiente. Allí detecta, entre otros aspectos, las formas, los colores y las relaciones. Si la cantidad de información es excesiva, el procesador se satura, y pierde por completo su eficacia. Es fácil encontrar diagramas que confunden el procesador visual, ya sea por su tamaño, por su complejidad o por su falta de uniformidad. Cualquier distractor gráfico que aparezca implica un esfuerzo adicional para la persona y el diagrama pierde la anhelada efectividad.
- > En la segunda etapa el cerebro se concentra en entender y analizar la información. Este es un proceso mucho más lento y lo hace la persona de manera consciente, focalizando y buscando más información y detalle, con el objetivo de entender el modelo. Es aquí donde pueden aparecer nuevos niveles de detalle (cosas que en el primer nivel no se podían apreciar) y que pueden estar expresados en formatos o texto. Ya el cerebro tiene un contexto y este descenso en profundidad lo puede hacer de manera más controlada. Lo importante en el descenso es que se respete la frontera del elemento por el que se bajó a este nuevo nivel. Aquí el riesgo más grande es que la persona se pierda en el recorrido del modelo o que se canse y abandone.
- > En la tercera etapa, el cerebro genera conclusiones que reflejan la comprensión que ha logrado de la realidad representada en el modelo. Es una etapa todavía más lenta, porque implica mezclar la información recibida con la que antes tenía la persona de la realidad. Es aquí donde hay realmente una comprensión del diagrama como un todo y en donde el cerebro debe pegar todas las partes que estudió en busca de detalles.

Buenos diagramas y el alfabeto visual del lenguaje

- > Se conoce como **ruido** cualquier elemento incluido en un diagrama que comprometa la efectividad del artefacto. Esto incluye elementos gráficos mal utilizados (líneas torcidas o cruzadas, por ejemplo), información innecesaria, cualquier tipo de distractor visual, información en un nivel de detalle inadecuado, inconsistencia en el uso de algún símbolo, entre otros.
- > Un diagrama debe ser uniforme en altura (todos los elementos deben ser del mismo nivel de abstracción), tan simple como sea posible, estéticamente agradable (lo que se ve desordenado comunica complejidad y lo que se ve feo comunica baja calidad), intuitivo (el cerebro debe poder hacer una parte del trabajo de manera automática) y limpio (sin elementos que generen ruido de cualquier tipo).
- > El reto al diseñar un lenguaje gráfico es permitir que un arquitecto cree buenos diagramas, pero claramente esto no se puede garantizar en todos los casos, porque no hay forma de controlar su uso (con un buen lenguaje de programación, por ejemplo, se pueden crear malos programas).
- > El procesador visual de un arquitecto se puede (y debe) entrenar. Es una de sus herramientas fundamentales de trabajo: debe ver formas y convertirlas en estructuras mentales de manera natural.

“The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering”

Daniel L. Moody

IEEE Transactions on Software Engineering (2009)

Este es un artículo que trata el tema de notaciones visuales, orientado al área de ingeniería de software. Un excelente documento para aquellos que quieran profundizar en este tema.

Alfabeto visual del lenguaje

Comencemos ahora sí a hablar del proceso de diseño de un lenguaje gráfico. El primer paso es tomar cada concepto del metamodelo y decidir cómo se va a representar. Puede ser algo gráfico, algo textual o una mezcla de los dos. Puede ser todo en un solo nivel o podemos tener una parte en un nivel y el resto en otro. En el primer nivel, por ejemplo, puede haber un ícono para representar la instancia del concepto y en el siguiente nivel (si alguien lo pide) aparece un formato con los valores de sus atributos. El conjunto de las representaciones gráficas de todos los conceptos se denomina el **alfabeto visual** del lenguaje.

Existen 6 variables (llamadas **variables visuales**) que nos permiten definir las características de una representación gráfica. Estas son (1) **forma** (cuadrado, círculo, ícono), (2) **color** (gris, negro, verde), (3) **tamaño** (radio, escala), (4) **textura** (relleno, transparencia, puntos), (5) **orientación** (normal, 90°, 180°) y (6) **brillo** (bajo, medio, alto). Podemos usar estas variables de manera estática (todas las instancias de un concepto se ven iguales) o de manera dinámica, en la cual asociamos los valores de los atributos de la instancia con alguna de las variables visuales (por ejemplo, si el valor del atributo “alerta” es verdadero, el color es rojo y, en caso contrario, el color es verde).

Aunque parece lógico, no sobra decir que no debería haber dos conceptos del metamodelo que usen la misma representación gráfica (principio de **claridad semiótica**). Imaginen la confusión del pobre cerebro si el concepto de “estudiante” se representa con una caja y que esa misma caja se utiliza en el modelo para dibujar el concepto de “curso”.

Alfabeto visual y reglas de composición

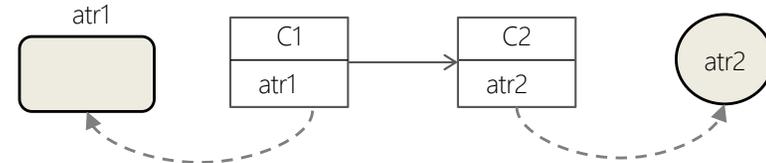
Alfabeto visual del lenguaje

- > Una recomendación es diseñar representaciones gráficas que se puedan distinguir fácilmente entre ellas (principio de **perceptual discriminability**). Si la única diferencia entre dos representaciones es el grosor de una línea, va a ser muy difícil para el cerebro diferenciarlas de manera eficiente. Eso nos va a pasar con mucha frecuencia con las relaciones entre conceptos, porque solo podemos jugar con el tipo de línea, el grosor de la línea y el tipo y tamaño de la flecha.
- > La siguiente recomendación es usar representaciones gráficas que el cerebro pueda asociar de manera natural con el concepto (principio de **transparencia semántica**). Si en un metamodelo asociamos con el concepto de "persona" un ícono que el cerebro pueda identificar fácilmente, le estaremos ayudando en el procesamiento de la información. En esto vamos a encontrar tres casos:

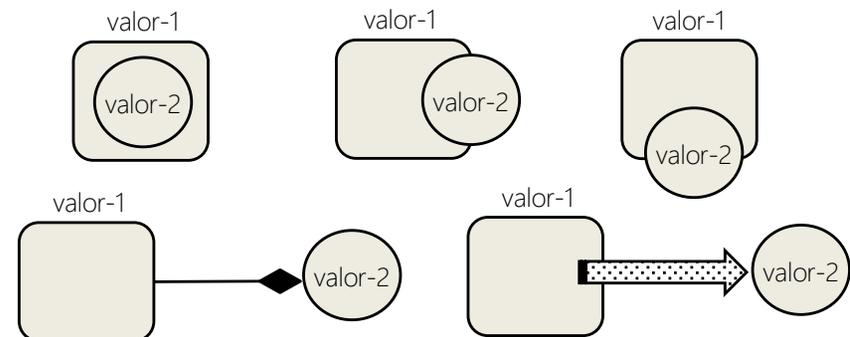
	Semántica directa: un arquitecto sin ningún entrenamiento hace inmediatamente la relación entre la representación y el concepto
	Semántica aprendida: un arquitecto puede aprender a relacionar el concepto con la forma, sin grandes dificultades (persona = caja)
	Semántica perversa: el cerebro del arquitecto va a estar confundido y va a rechazar la notación gráfica seleccionada (persona no es igual a avión).

- > Algo en lo que se debe tener cuidado es en garantizar que el número de elementos gráficos diferentes sea manejable por el cerebro (principio de **economía gráfica**). Después de un cierto número, el cerebro se satura y pierde efectividad.

Ahora que ya tenemos definido el alfabeto visual, debemos establecer la manera de representar las relaciones. Supongamos que tenemos el siguiente metamodelo, en el cual asociamos una visualización con cada concepto (las líneas curvas no hacen parte del metamodelo y solo las usamos para asociar un concepto con su notación gráfica):



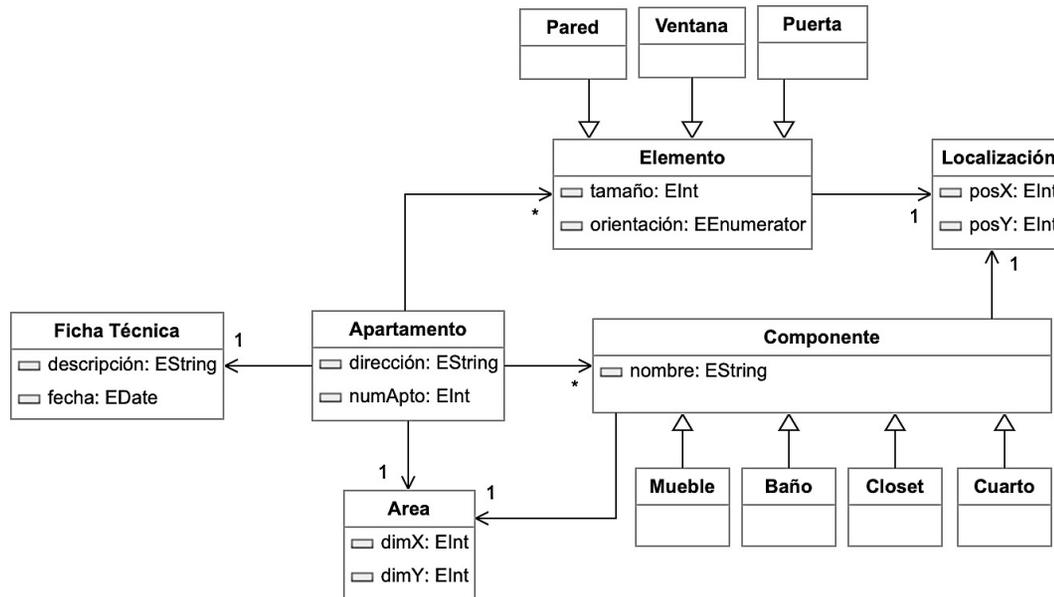
Tenemos tres opciones: (1) la relación se expresa situando la visualización de C2 dentro de la visualización de C1 (en un borde, en el centro, o algo así), (2) asignamos una visualización a la relación, o (3) ignoramos C2 y solo presentamos su visualización en el siguiente nivel de C1 (solo al descender de nivel aparece C2).



Si la relación entre C1 y C2 es de cardinalidad múltiple, podemos indicar la forma de situar la visualización de cada una de las instancias de C2 sobre la visualización de C1. Todas sobre el borde, por ejemplo. Hay múltiples lenguajes y herramientas para expresar el lenguaje gráfico de un metamodelo, que permiten generar de manera automática un editor para construir modelos.

Ejemplo: el plano de un apartamento (1)

Veamos ahora un ejemplo completo de diseño de un lenguaje gráfico para un dominio y la forma de utilizar el artefacto BAC-13 para especificarlo. Vamos a trabajar sobre una simplificación del dominio del plano de un apartamento. Todo está construido para que se pueda extender fácilmente.



- > El metamodelo tiene dos conceptos centrales: Elemento y Componente
- > El primero tiene un tamaño y una orientación (VERT=vertical, HORIZ=horizontal) y lo vamos a utilizar como base para representar paredes, ventanas y puertas.
- > El segundo tiene un tamaño en cada dimensión (clase Área) y lo vamos a utilizar como base para representar cuartos, clósets, baños y muebles.
- > Un apartamento es un espacio bidimensional (clase Área), que tiene asociada una ficha técnica y algunos atributos.
- > Tanto los elementos como los componentes tienen una Localización (posX, posY), con respecto al apartamento como un todo.

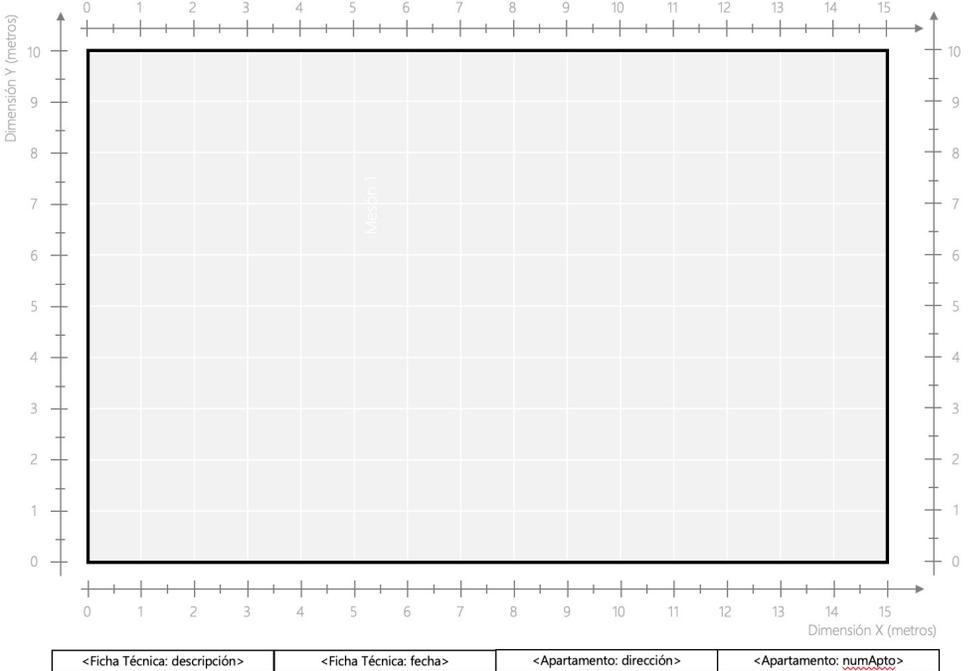
Un lenguaje gráfico se especifica definiendo la manera como se representan los conceptos del metamodelo y cada una de las relaciones que se van a incluir en el lenguaje. Recuerden que algunos conceptos o relaciones pueden tener una representación de tipo formato y no aparecer en el lenguaje gráfico. Vamos a utilizar el artefacto BAC-13 que se describe a continuación:

Artefacto BAC-13

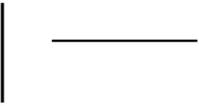
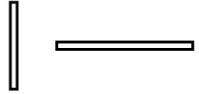
Concepto / Relación	Representación gráfica	Descripción

- > En la primera columna se incluye el nombre del concepto o de la relación del metamodelo que se va a representar de manera gráfica.
- > En la segunda columna se define la representación gráfica que se va a usar.
- > En la tercera columna se hace una descripción de la forma como se calculan las variables gráficas y visuales, a partir de los atributos y relaciones que aparecen en el metamodelo. Muchas veces es suficiente con una descripción informal de la forma en que se representa cada concepto. Nosotros vamos a utilizar la notación que se ilustra en la siguiente página. No es formal, pero intenta ser lo más precisa posible.

Ejemplo: el plano de un apartamento (2)

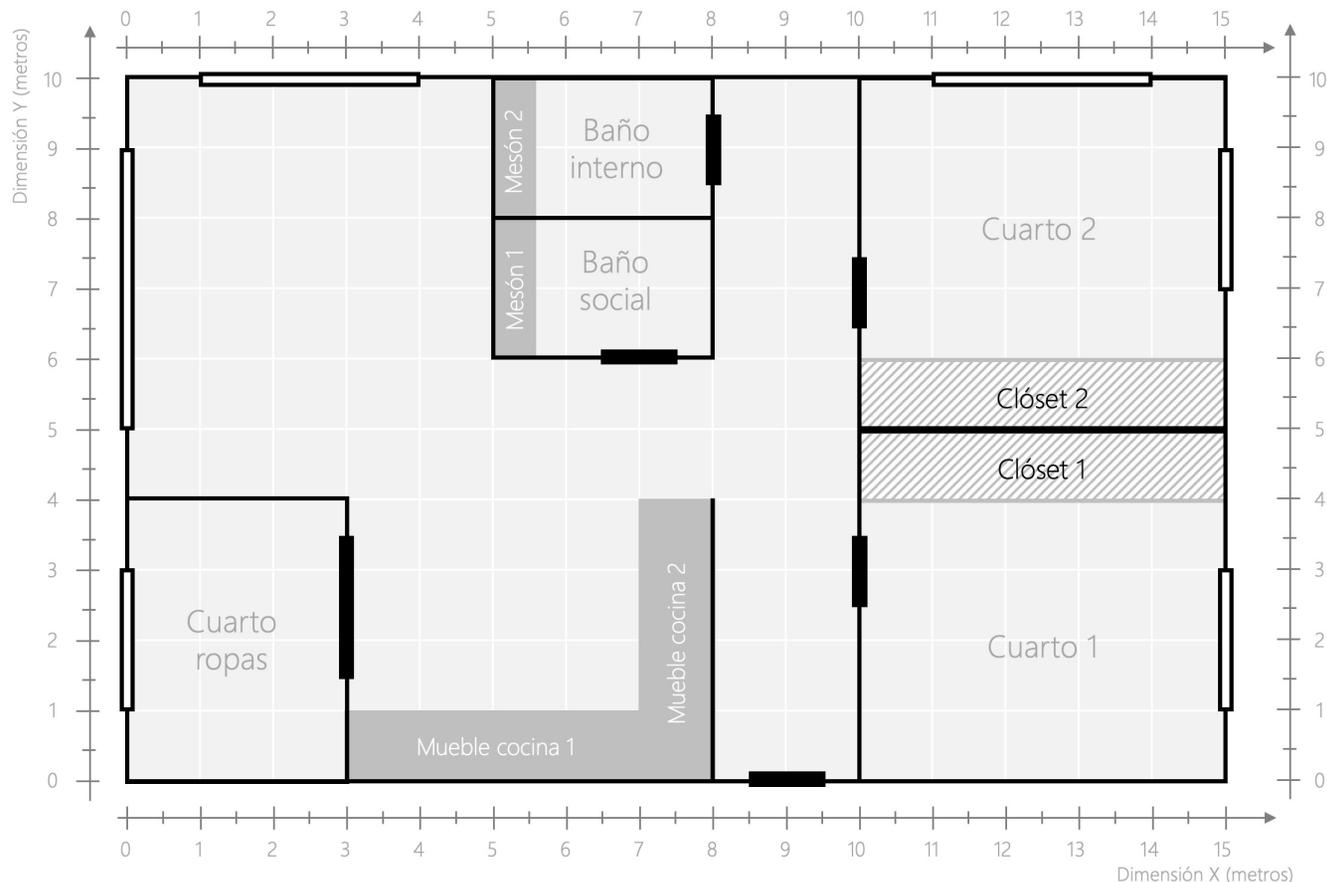
Concepto	Representación gráfica	Descripción
Apartamento		<ul style="list-style-type: none"> > Un apartamento se representa con una tabla con información general del plano (parte inferior). Allí aparece el atributo del metamodelo que se utiliza para llenarlo (sintaxis <Clase:atributo>). > Un apartamento se representa con un espacio vacío como el que aparece en la figura, que tiene en el eje X <Area:dimX> metros y en el eje Y <Area:dimY> metros. > La visualización del apartamento define el eje de coordenadas con respecto al cual están localizados todos los elementos y componentes.
Cuarto		<ul style="list-style-type: none"> > Un cuadrado como el de la figura, de dimensiones <Area:dimX> y <Area:dimY> > El nombre del cuarto (heredado de Componente) va en el centro del cuadrado, en color gris. > El cuadrado está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento
Clóset		<ul style="list-style-type: none"> > Un cuadrado como el de la figura, de dimensiones <Area:dimX> y <Area:dimY> > El nombre del clóset (heredado de Componente) va en el centro del cuadrado, en color negro. > El cuadrado está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento

Ejemplo: el plano de un apartamento (3)

Concepto	Representación gráfica	Descripción
Baño		> La misma representación gráfica de un cuarto.
Mueble		<ul style="list-style-type: none"> > Un cuadrado como el de la figura, de dimensiones <Area:dimX> y <Area:dimY> > El nombre del mueble (heredado de Componente) va en el centro del cuadrado, en color blanco. > El cuadrado está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento.
Pared		<ul style="list-style-type: none"> > Una línea vertical u horizontal según el valor del atributo <Elemento:orientación> > El extremo inferior izquierdo de la línea está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento.
Ventana		<ul style="list-style-type: none"> > Una delgada caja vertical u horizontal según el valor del atributo <Elemento:orientación>. > El extremo inferior izquierdo de la caja está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento.
Puerta		<ul style="list-style-type: none"> > Una delgada caja negra vertical u horizontal según el valor del atributo <Elemento:orientación>. > El extremo inferior izquierdo de la caja está situado en las coordenadas <Localización:posX>, <Localización:posY> usando el sistema de coordenadas del apartamento.

- > En este caso las relaciones las usamos para poder establecer un sistema común de coordenadas, que es definido por la raíz del metamodelo y seguido por todos los demás conceptos.
- > Noten que las clases Elemento y Componente no tienen una representación gráfica, solo las subclases de ellas dos. No siempre es así, pero en este caso esos dos conceptos son abstractos y solo se usaron para ayudar a estructurar de manera simple el metamodelo.
- > Las clases Localización y Área tampoco tienen representación gráfica. Solo se utilizan para almacenar información gráfica de los demás conceptos.
- > En la siguiente página aparece un ejemplo de un modelo (un plano de un apartamento) expresado usando el lenguaje que acabamos de especificar.
- > Hay herramientas en el mercado a las cuales se les puede dar el metamodelo y la descripción del lenguaje gráfico, y son capaces de generar un editor de modelos (en nuestro caso, lo podríamos utilizar para crear de manera automática un editor de planos de apartamentos). Dicho editor ofrece al usuario la posibilidad de incluir y manejar los elementos gráficos especificados y de verificar que los modelos respeten todas las reglas establecidas por el metamodelo: que sean conformes.

Ejemplo: el plano de un apartamento (4)



Plano de un apartamento	11-feb-21	Calle 100 # 7-25	Apartamento: 702
-------------------------	-----------	------------------	------------------

El modelo es de un apartamento de 150 metros cuadrados (10*15), con dos cuartos y dos baños. Se puede observar también la puerta principal (abajo en las coordenadas (0, 8.5) y otras 5 puertas internas. Tiene 6 ventanas hacia el exterior y algunos muebles fijos en la cocina y los baños. En total el modelo tiene 26 elementos. En la parte inferior tenemos la ficha técnica del plano, incluyendo su fecha de elaboración.

En la siguiente página aparece el mismo modelo, pero representado usando un lenguaje de formatos. Lo incluimos para que se pueda comparar la capacidad de expresión de cada uno. Aunque ambos tienen exactamente la misma información (representan la misma realidad), cuando usamos el lenguaje gráfico le comunicamos a quien lo observa unas características del modelo que no logramos expresar con el lenguaje de formatos.

Ejemplo: el plano de un apartamento (5)

Plano apartamento:			
Descripción:	Plano de un apartamento	Número apto:	702
Fecha:	11-feb-21	Dimensión X:	15
Dirección:	Calle 100 # 7-25	Dimensión Y:	10

Cuartos:

Nombre	Área		Localización	
	Tamaño X	Tamaño Y	X	Y
Cuarto 1	5	5	10	0
Cuarto 2	5	5	10	5
Cuarto ropas	3	4	0	0

Baños:

Nombre	Área		Localización	
	Tamaño X	Tamaño Y	X	Y
Baño interno	3	2	5	8
Baño social	3	2	5	6

Closets:

Nombre	Área		Localización	
	Tamaño X	Tamaño Y	X	Y
Clóset 1	5	1	10	4
Clóset 2	5	1	10	5

Muebles

Nombre	Área		Localización	
	Tamaño X	Tamaño Y	X	Y
Mueble cocina 1	1	4	7	0
Mueble cocina 2	4	1	3	0
Mesón 1	0,5	2	5	6
Mesón 2	0,5	2	5	8

Puertas

Dimensión	Orientación	Localización	
		X	Y
1	HORIZ	8,5	0
1	VERT	10	2,5
1	VERT	10	6,5
1	VERT	8	8,5
1	HORIZ	6,5	6
2	VERT	3	1,5

Ventanas

Dimensión	Orientación	Localización	
		X	Y
2	VERT	15	1
2	VERT	15	7
3	HORIZ	11	10
3	HORIZ	1	10
4	VERT	0	5
2	VERT	0	1

Pared

Dimensión	Orientación	Localización	
		X	Y
4	VERT	8	0

- > Lo primero que debemos señalar es que se trata del mismo modelo expresado con dos artefactos distintos, cada uno con un lenguaje diferente.
- > Un lenguaje de formatos se puede describir de varias maneras. La más sencilla (artefacto BAC-14) es construir el formato usando una herramienta como Excel (o cualquier otra hoja de cálculo) y luego sobre ese formato explicar la manera en que se debe colocar la información, recorriendo los conceptos y las relaciones del metamodelo. Para cada hoja del archivo, cada tabla y cada campo, debe quedar claro como se llena.
- > Hay casos en los cuales se deben incorporar llaves internas para poder separar la información en hojas distintas. Se usan las mismas técnicas que se usan para diseñar bases de datos relacionales a partir de un metamodelo.
- > La construcción del lenguaje de formatos para este ejemplo fue relativamente sencilla, porque no hubo necesidad de representar las relaciones entre los elementos. Dado que todo está relacionado a través del concepto de localización, bastó con incluir un conjunto de elementos por cada tipo que tiene el metamodelo.

5. Puntos de vista y vistas

La última herramienta que introduciremos en esta lectura nos va a permitir descomponer los modelos y los metamodelos (partirlos) para poder manejar así su complejidad y sacarles el máximo provecho. Se trata de los puntos de vista y de las vistas, que nos serán muy útiles tanto para la construcción de los metamodelos y los modelos, como para el uso por parte de los destinatarios.

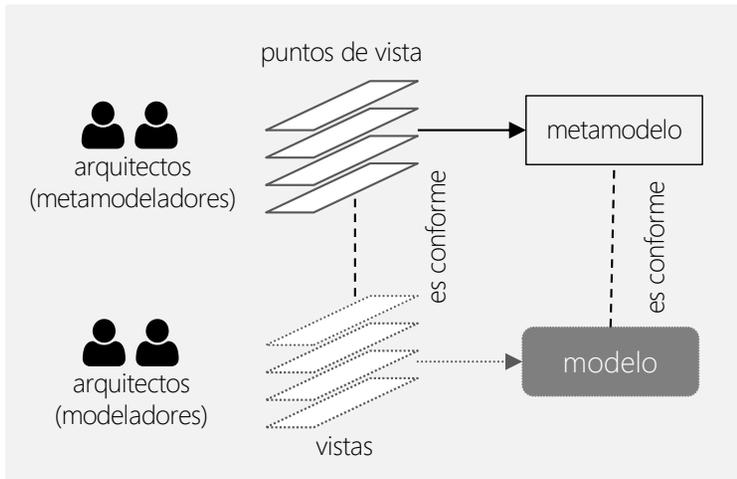


Fig.15 – Puntos de vista y vistas en la construcción

- > En la figura 15 aparecen los arquitectos en sus dos roles en el proceso de construcción del metamodelo (modelo conceptual) y del modelo.
- > En el primero de los casos, pueden expresar el metamodelo como una suma de dimensiones, cada una con un metamodelo parcial (una parte), que se denomina un punto de vista. Todos los puntos de vista tienen la misma raíz de manera que se pueden componer sin problema para obtener el metamodelo completo. Los puntos de vista no son disyuntos (pueden tener partes en común), pero debe haber una coherencia global entre ellos que permita que se puedan sumar sin problema.
- > Y en lugar de construir un modelo completo (monolítico), los arquitectos pueden crear el modelo por partes, elaborando el modelo asociado con cada punto de vista. Cada uno de esos modelos se denomina una vista. Una vista debe ser conforme con el punto de vista respectivo.

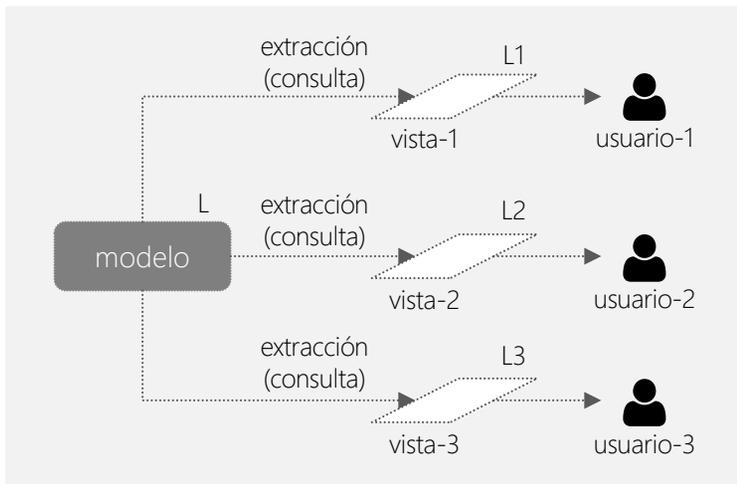
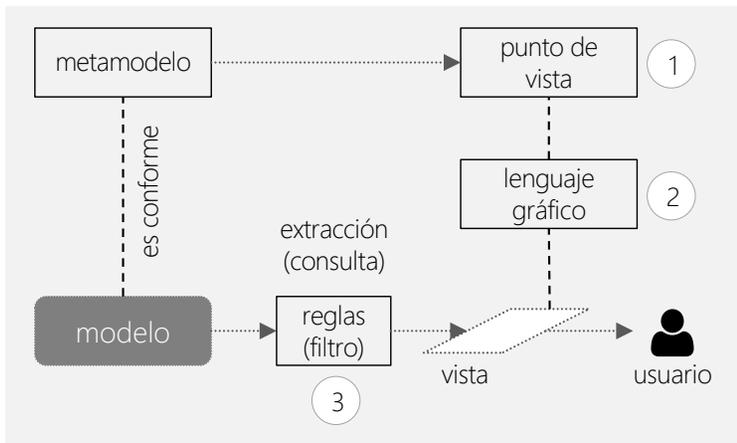


Fig.16 – Puntos de vista y vistas en la consulta

- > En la figura 16 aparece la dimensión de uso del modelo construido, independiente de la forma en la que fue armado. Aquí los usuarios (destinatarios) deben poder extraer las partes del modelo que les resulten de interés, para la tarea que deben realizar.
- > Cada usuario puede tener una necesidad distinta en cuanto a información y a tipo de visualización. Cada una de esas extracciones (consultas) se denomina también una vista. Debe ser claro que no existe necesariamente una relación entre las vistas con las que se construyó el modelo y las vistas correspondientes a las consultas.
- > Cada vista se puede expresar usando un lenguaje distinto (no es indispensable utilizar el mismo lenguaje asociado al modelo completo). En la figura, el modelo está expresado en el lenguaje L, mientras las vistas utilizan los lenguajes L1, L2 y L3. Recuerde que una vista sigue siendo un modelo, solo que representa únicamente una parte de la realidad.

¿Cómo expresar una vista de consulta?

Como vimos antes, una vista es una parte de un modelo y se puede utilizar tanto para construir dicho modelo (la llamaremos **vista de edición**) como para consultarlo y usarlo (la llamaremos **vista de consulta**). Para las vistas de edición ya establecimos que se definen usando un punto de vista del metamodelo completo. Veamos ahora cómo se define una vista de consulta.

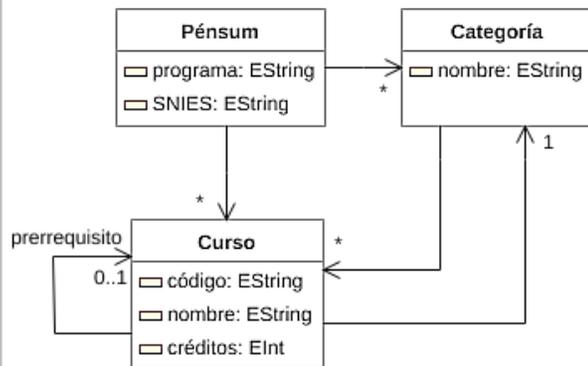


- > Tal como se muestra en la figura 17, una vista se especifica utilizando 3 elementos: (1) un punto de vista, (2) un lenguaje para ese punto de vista (usando lo que estudiamos en la sección anterior) y (3) un conjunto de reglas que especifican las instancias del modelo que deben quedar en la vista.
- > Un punto de vista es un metamodelo, que corresponde a una parte del metamodelo global y que comparte con éste algunos aspectos estructurales (comparte la raíz, por ejemplo).
- > El lenguaje puede ser textual, gráfico o una mezcla de los dos, y debe estar diseñado a la medida del usuario que va a utilizar la información.
- > Las reglas son un filtro o consulta sobre el modelo. Si la vista se va a calcular automáticamente (que es lo ideal), se debe expresar en un lenguaje formal.

Fig.17 – Especificación de una vista de consulta

Comencemos con un ejemplo muy simple en el que podamos observar un metamodelo (izquierda) y un modelo (derecha) expresado en un lenguaje textual (formato). En la siguiente página lo usaremos para ver algunos puntos de vista y las vistas respectivas:

Ejemplo



Maestría en Arquitecturas de TI SNIES: 101531	CATEGORÍA	CÓD.	NOMBRE	CR.	REQ.
	Fundamentación	MATI-01	Arquitectura de Negocio	4	
		MATI-02	Arquitectura Empresarial	4	
		MATI-03	Fundamentos de Arquitectura	4	
	Profundización	MATI-04	Arquitectura de Integración	4	MATI-03
		MATI-05	Arquitectura de Información	4	MATI-01
		MATI-06	Arquitectura de Infraestructura	4	MATI-03
		MATI-07	Arquitectura de Seguridad	4	MATI-06
	Complemento	MBIT-01	Comunicación Efectiva	2	
		MBIT-02	Finanzas para Arquitectos	2	MBIT-01
Integración	MATI-10	Proyecto Final	8	MBIT-02	

Vistas de consulta

Ejemplo (continuación...)

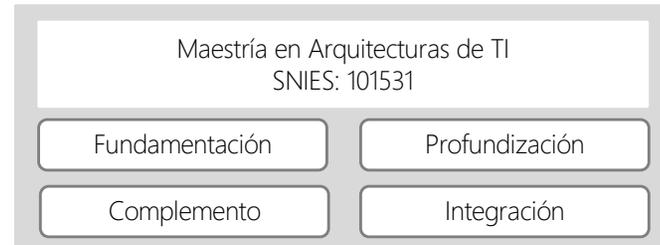
En el metamodelo anterior tenemos un p nsum (plan de estudios) de un programa, estructurado por categor as y cursos. En el modelo (formato) aparece un programa particular, con 4 categor as y 10 cursos en total. Algunos cursos tienen un prerrequisito.

PV1:



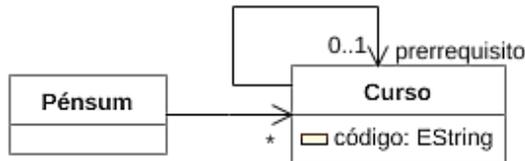
El primer punto de vista (PV1) incluye solo las categor as del p nsum. F jese que tiene la misma ra z del metamodelo completo y que es realmente una parte del mismo. Seleccionamos ahora un lenguaje gr fico para expresar la vista y no incluimos ninguna regla de consulta, de manera que en la vista aparecen todas las instancias del tipo Categor a que hay en el modelo.

V1.1:



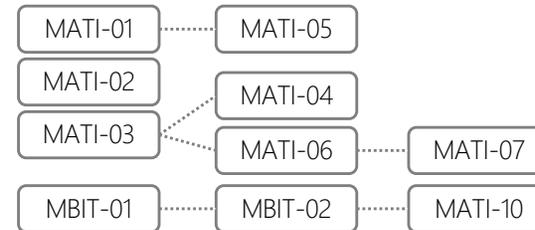
La vista resultante (V1.1) trae la informaci n del modelo que nos indica el punto de vista, luego aplica las reglas de extracci n (filtro o consulta) y finalmente lo muestra usando el lenguaje seleccionado.

PV2:



El segundo punto de vista (PV2) incluye solo los cursos del plan de estudios, ignora los atributos del concepto P nsum y solo deja un atributo del concepto Curso. Seleccionamos ahora un lenguaje gr fico distinto, en el que podamos ver el grafo de dependencia entre los cursos. Si no ponemos ninguna regla, obtenemos la primera vista de la derecha (V2.1). Si ponemos como regla que solo nos interesan los cursos que no tienen ning n requisito, aparece la segunda (V2.2). Si la regla dice que solo queremos tener los cursos de 4 cr ditos, tenemos la tercera vista (V2.3). F jese que la regla se hace sobre el metamodelo completo.

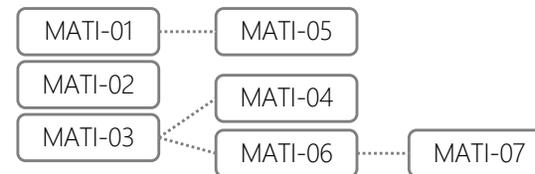
V2.1:



V2.2:

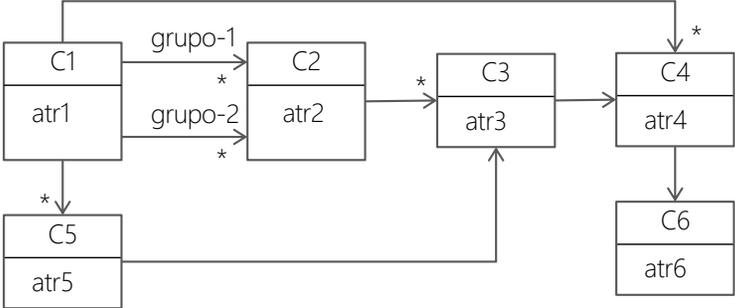
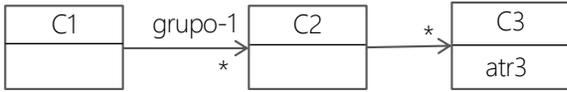
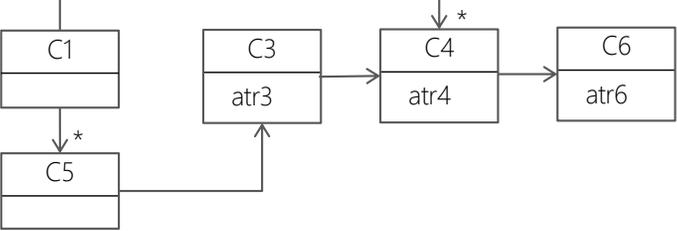
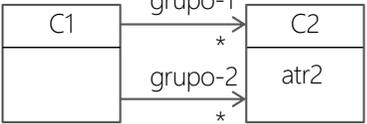


V2.3:



Puntos de vista de consulta

Hoy en día existen múltiples herramientas en el mercado para expresar un punto de vista, las reglas asociadas y el lenguaje, cada una de ellas basada en un formalismo distinto. Ese no es un tema que nos interese abordar en esta lectura, pero que sí debe ser estudiado por los arquitectos si queremos poder contar con herramientas y mecanismos poderosos de manejo de modelos. Por ahora vamos a cerrar esta sección con un ejemplo un poco más grande, en donde podamos ilustrar más formas de describir puntos de vista.

Metamodelo	Punto de vista 1 (PV1)
 <p>Partimos de un metamodelo con 6 clases (C1-C6) y múltiples relaciones entre ellas. La raíz del metamodelo es la clase C1 y, desde allí, se deben expresar todos los puntos de vista.</p>	 <p>En este primer punto de vista sólo incluimos 3 de las clases, y nos quedamos únicamente con el atributo "atr3" de la clase C3. A menos que la regla de consulta diga algo adicional, en este punto de vista estamos interesados en todas las instancias de C3 que son alcanzables por el camino señalado. Quedan por fuera de la vista, por ejemplo, las instancias de C3 que son alcanzables a través del camino que pasa por C5 o por el "grupo-2" de instancias de C2. Adicional, podemos agregar una regla para filtrar las instancias del conjunto definido (por ejemplo, que el atributo "atr3" tenga un valor dado).</p>
Punto de vista 2 (PV2)	Punto de vista 3 (PV3)
 <p>Ahora nos interesan todas las instancias de C4 y C6, alcanzables por los dos caminos marcados. Ese conjunto podemos filtrarlo por cualquier regla que incluya los atributos de las clases que hacen parte de los dos caminos. En el lenguaje de expresión, sólo podemos incluir los atributos que quedaron en el punto de vista (atr3, atr4 y atr6 en el ejemplo).</p>	 <p>En este punto de vista tenemos las instancias de C2 alcanzables por los dos caminos marcados. Como regla podemos definir que solo nos interesan las instancias que pertenecen simultáneamente a los dos grupos. Esto es, las instancias que a la vez están en "grupo-1" y en "grupo-2". La regla también podría hacer referencia a un valor específico del atributo "atr2". En resumen, el punto de vista localiza estructuras de interés dentro del metamodelo y luego incluye reglas de selección de las instancias específicas.</p>

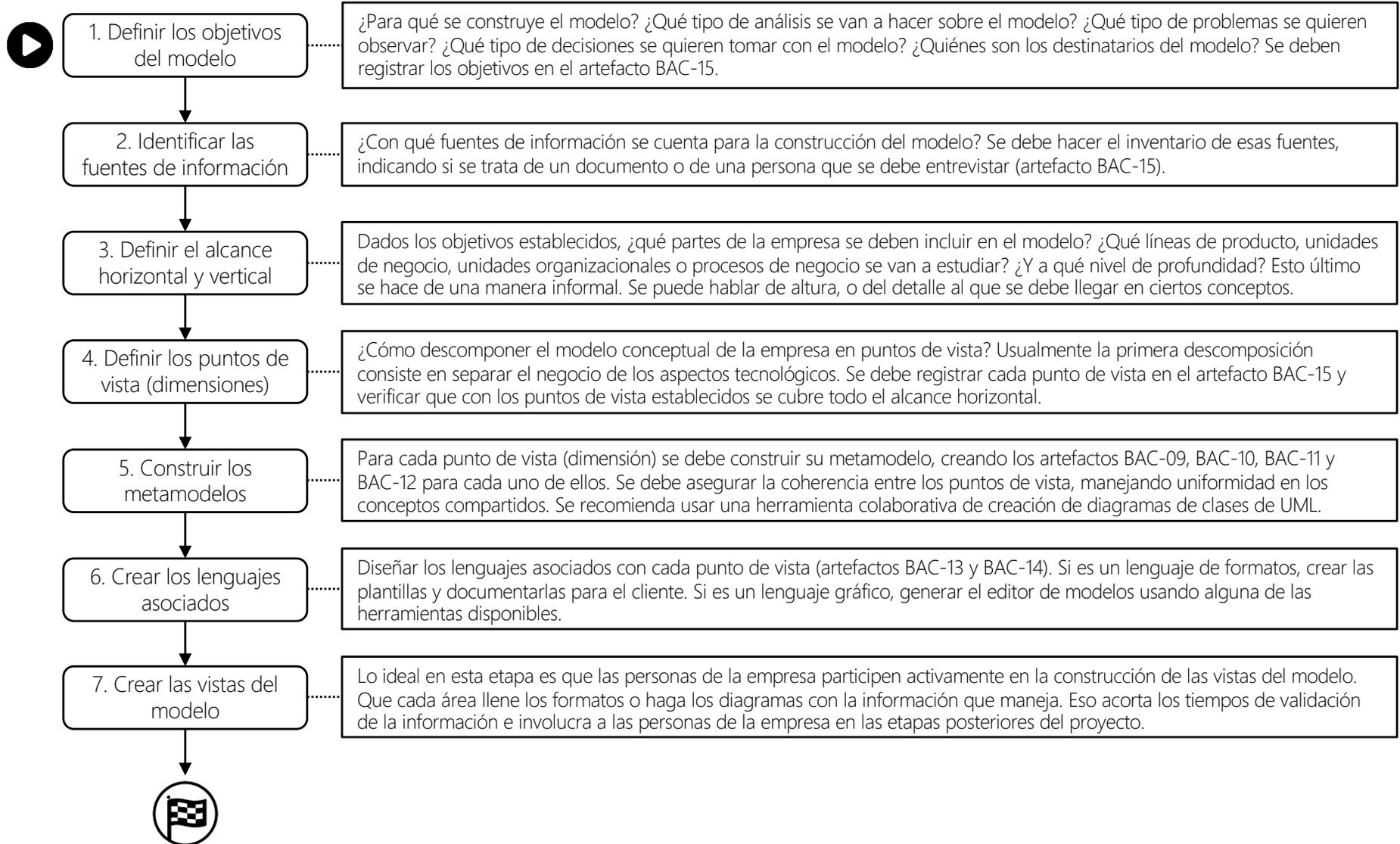
Ejemplo: puntos de vista de un apartamento

Veamos ahora cómo serían algunos puntos de vista aplicados al ejemplo del plano del apartamento que usamos en la sección anterior.

ID	Metamodelo	Regla	Lenguaje	Vista																
PV1	<p>Un catálogo de los cuartos que tiene el apartamento, incluyendo su área y el total</p>	Ninguna	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Espacio m2</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td>TOTAL</td> <td>0</td> </tr> </tbody> </table> <ul style="list-style-type: none"> > La primera columna contiene el nombre del cuarto (<Componente:nombre>) > La segunda columna contiene el tamaño en metros cuadrados, calculado de la siguiente manera: <Area.dimX> * <Area.dimY> > La última fila contiene la suma del espacio de todos los cuartos 	Nombre	Espacio m2			TOTAL	0	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Espacio m2</th> </tr> </thead> <tbody> <tr> <td>Cuarto 1</td> <td>25</td> </tr> <tr> <td>Cuarto 2</td> <td>25</td> </tr> <tr> <td>Cuarto ropas</td> <td>12</td> </tr> <tr> <td>TOTAL</td> <td>62</td> </tr> </tbody> </table>	Nombre	Espacio m2	Cuarto 1	25	Cuarto 2	25	Cuarto ropas	12	TOTAL	62
Nombre	Espacio m2																			
TOTAL	0																			
Nombre	Espacio m2																			
Cuarto 1	25																			
Cuarto 2	25																			
Cuarto ropas	12																			
TOTAL	62																			
PV2	<p>El cálculo de cuántos baños tiene el apartamento</p>	Ninguna	<table border="1"> <thead> <tr> <th>#baños</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> > El número de baños se calcula con la cardinalidad de la contenedora de componentes de tipo Baño. 	#baños				<table border="1"> <thead> <tr> <th>#baños</th> <th></th> </tr> </thead> <tbody> <tr> <td>2</td> <td></td> </tr> </tbody> </table>	#baños		2									
#baños																				
#baños																				
2																				
PV3	<p>Plano con las ventanas que tiene el apartamento, que son de 2 o más metros</p>	Todas las ventanas que tienen 2 o más metros de tamaño	<ul style="list-style-type: none"> > El mismo lenguaje del modelo, eliminando la ficha técnica y la escala. 																	

6. Modelos en un contexto empresarial

Al momento de abordar el problema de crear el modelo de una empresa, se recomienda seguir la secuencia de pasos que se plantea a continuación:



Artefacto para el modelo conceptual de una empresa

Artefacto BAC-15

Empresa:			
Objetivo:			
Alcance horizontal y vertical:			
Fuentes de información			
ID	Nombre	Tipo	Descripción
Puntos de vista			
ID	Nombre	Contenido	

- > En el artefacto BAC-15 se registra la información global del modelo que se va a construir de la empresa. Esta ficha técnica se puede extender con información como la fecha de elaboración del modelo, el proyecto del cual hace parte el modelo, la empresa que construye el modelo, entre otros.
- > Para las fuentes de información se debe asignar un ID, un nombre, un tipo (persona o documento con el tipo del archivo) y una descripción que depende del tipo de fuente. Si es una persona, se incluye el cargo, el nombre de la persona, la información que nos puede suministrar, etc. Si es un documento, se incluye el nombre del archivo, su localización física, la fecha de la última actualización y su contenido.
- > Por último aparece la información de los puntos de vista (dimensiones) en los que vamos a descomponer el modelo de la empresa. Cada dimensión tiene un identificador, un nombre y una descripción de su contenido.

Alcance y completitud de los modelos

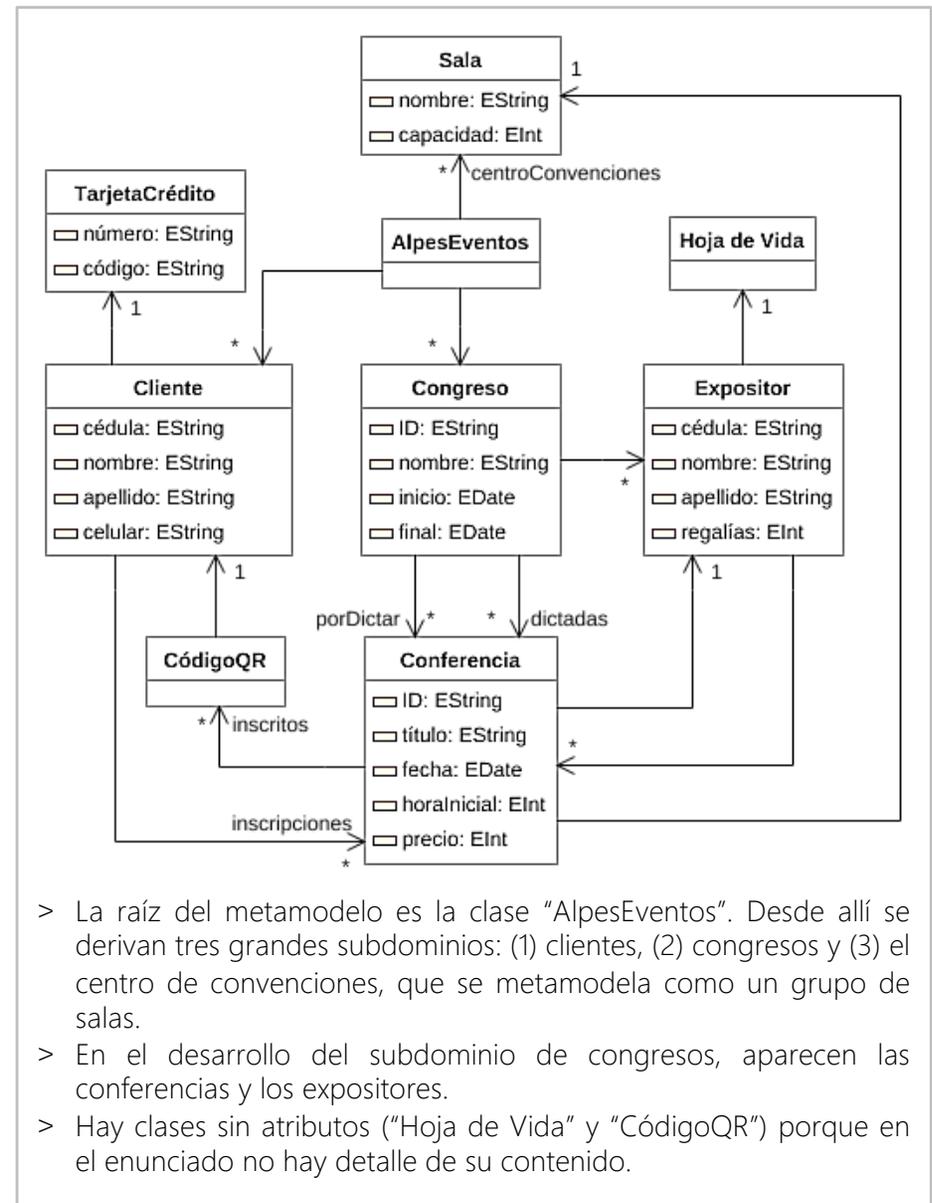
> Una pregunta recurrente en la construcción del metamodelo de una empresa es ¿cuál debe ser su alcance vertical? Y la respuesta es sencilla: debe incluir todos los conceptos y atributos que necesite para que el modelo sea adecuado para resolver la tarea para la que va a ser utilizado. Si vamos a diagnosticar un problema de calidad de información de una empresa, posiblemente nos toque bajar bastante en detalles de las bases de datos y detalles de integración de aplicaciones, para poder observar el problema y generar un concepto. Pero si el modelo solo va a servir para definir el conjunto de indicadores estratégicos que necesita la empresa, posiblemente los detalles de almacenamiento de la información no van a ser necesarios. Si recogemos demasiada información estamos perdiendo el tiempo. Si recogemos poca información, corremos el riesgo de hacer mal la tarea encomendada.

> Otro tema es ¿hasta donde construir el modelo de la empresa? A lo mejor en el metamodelo nos interesa tener el detalle completo de los clientes o del portafolio de productos de la empresa, pero ¿realmente vale la pena llenar los formatos del modelo con toda esa información? ¿Vamos a incluir todos los clientes, con todas sus transacciones? Eso implicaría copiar en nuestros formatos todas las bases de datos de la empresa, lo cual suena innecesario. En ese caso es mejor señalar que esa parte del modelo se encuentra en un repositorio de la compañía (archivo o base de datos) y que si se requiere, se debe consultar allí directamente la información. Eso quiere decir que en nuestros formatos lo que hacemos es incluir "un apuntador" hacia el repositorio de la empresa que contiene la información de esa parte del modelo. ¡Hay que ser muy prácticos!

7. Ejemplos completos de conceptualización

La empresa AlpesEventos tiene como negocio el manejo de congresos, que corresponden a un grupo coordinado de conferencias en un tema o área particular de conocimiento. Para esto, la empresa cuenta con un centro de convenciones que tiene múltiples salas, de distintas capacidades (número de sillas). Cada congreso tiene un identificador, un nombre (por ejemplo, "Congreso Internacional de Inteligencia Artificial") y se lleva a cabo entre una fecha inicial y una fecha final. Nunca hay dos congresos en paralelo. Cada sala, por su parte, tiene un nombre que la identifica, adicional a la ya mencionada capacidad. Durante los días que dura el congreso se dictan múltiples conferencias simultáneas de dos horas, en el tema del congreso. Cada conferencia tiene un identificador, un título (por ejemplo, "Aprendizaje Automático"), una fecha, un expositor, una hora inicial y un precio. Cada congreso hace con sus conferencistas un contrato por la duración del mismo. Cada conferencista tiene una cédula, un nombre, un apellido, una hoja de vida y el porcentaje de las entradas de sus conferencias con el cual es remunerado (lo usual es el 25% de regalías). Al final del congreso la empresa hace el cálculo global y le genera el respectivo pago.

Los clientes de la empresa deben descargar la aplicación móvil, registrarse con su número de teléfono celular, su cédula, su nombre, su apellido y la información de una tarjeta de crédito, a la cual se le carga el precio de las conferencias a las que se inscriba. A través de esta aplicación el cliente puede consultar las conferencias disponibles y puede inscribirse a ellas. Por cada inscripción que haga, la empresa le envía un código QR que debe presentar a la entrada de la sala y le carga a la tarjeta el monto de la inscripción. El cliente puede consultar todas las conferencias a las que está inscrito. Por política de la empresa no se hace ningún tipo de reembolso por la no asistencia a una conferencia inscrita. A la entrada de cada sala hay una persona encargada de controlar el acceso. Para esto, pide el código QR a cada persona que quiere entrar, verifica que sí esté inscrito en esa conferencia y le permite el paso a la sala.



Bolsa de valores: ejemplo de conceptualización

La bolsa de valores quiere desarrollar un sistema informático que le permita generar información actualizada de las acciones e índices bursátiles que allí maneja, así como soportar el intercambio de acciones entre los clientes de las distintas empresas corredoras de bolsa allí inscritos. Su trabajo se basa en un depósito virtual de títulos valores (acciones), en donde aparece para cada cliente (identificado por su cédula) la lista de acciones que posee (cantidad y código de la acción) y el código de la empresa corredora de bolsa autorizada para hacer transacciones a su nombre. Cada acción tiene un código, un nombre y un valor actual en el mercado. Un índice tiene una lista de acciones, cada una de las cuales tiene un peso (un porcentaje) y se reconoce por un nombre. Las operaciones básicas que debe soportar el sistema son: (1) El administrador del sistema debe poder registrar una nueva empresa corredora de bolsa. Para esto se debe suministrar un nombre y el código de la empresa asignado por la superintendencia, (2) La empresa corredora debe poder registrar un nuevo cliente en el depósito virtual, dando su nombre, apellido, cédula, teléfono, dirección y el código de la empresa corredora de bolsa. Un cliente sólo puede ser manejado por una empresa, (3) Cualquier persona debe poder consultar por el portal el valor actual de una acción, dado su código, (4) Cualquier persona debe poder consultar por el portal el valor actual de un índice, calculado como la suma de los valores de las acciones que hacen parte del índice por su respectivo peso, (5) Un corredor de bolsa (empleado de la empresa corredora de bolsa) debe poder hacer una oferta de venta. Para esto, un corredor de bolsa debe indicar la cédula del cliente que vende, el código de la acción que quiere vender, el número de estas y el precio pedido por cada acción. Inmediatamente el sistema despliega esta información en las pantallas de todos los corredores. Si alguien acepta la oferta y hace la compra, el precio de la acción toma como nuevo valor el precio pedido en la transacción (así suben o bajan los precios de las acciones). A cada oferta el sistema le asigna de manera automática un código único. Una oferta de venta tiene también una vigencia, definida como una fecha y una hora.

Si después de ese plazo nadie ha hecho una oferta de compra, ésta se cancela. Por reglamentación, si una acción sube o baja más del 10% en un día, esta acción se congela hasta el día siguiente y todas las ofertas de venta/compra se cancelan inmediatamente. (6) Un corredor de bolsa debe poder aceptar una oferta de venta (comprar). Para esto, un corredor de bolsa debe indicar el código de la oferta de venta y la cédula del cliente que la acepta. En ese caso las acciones se trasladan en el depósito virtual al cliente que acaba de comprar. Como pueden llegar muchas órdenes de compra para la misma oferta de venta, a cada una de ellas se le debe asignar un código, y deben mantener un estado que puede ser: aceptada (se hizo la venta pedida) o rechazada (ya la venta se había hecho cuando esta oferta de compra fue atendida). Las ofertas de compra se deben procesar en estricto orden de llegada. Por razones de auditoría, se debe guardar la información de cada una de ellas, con la hora exacta en la que fue recibida la orden. Es importante informar al corredor de bolsa sobre cada movimiento hecho y el resultado. Una vez la venta se ha concretado, la oferta de venta desaparece de las pantallas de los corredores. (7) Consultar el portafolio de acciones de un cliente. Para esto, una empresa corredora de bolsa debe dar la cédula del cliente. Al consultar el portafolio debe aparecer información de cada acción que tiene (código y número de estas), el valor en ese instante de cada una y el valor total por acción. También debe incluir el valor total de dinero que tiene el cliente en ese momento en el depósito virtual. Esta consulta también debe mostrar sus ofertas de venta que se encuentran vigentes y las órdenes de compra hechas que fueron aceptadas. (8) Cualquier persona debe poder consultar por el portal el movimiento de una acción durante el día. El sistema despliega una gráfica en la que muestra el valor de la acción cada minuto. La bolsa de valores abre a las 9 am y cierra operaciones (no recibe más órdenes) a la 1 pm. (9) El administrador del sistema debe poder registrar las acciones de las empresas que comienzan a cotizar en la bolsa de valores.

tuVivienda.com: ejemplo de conceptualización

La agencia inmobiliaria tuVivienda está dedicada al arrendamiento y venta de apartamentos, casas, locales, bodegas y oficinas. La atención de los clientes corre por cuenta de cuatro grupos de operación que se describen a continuación:

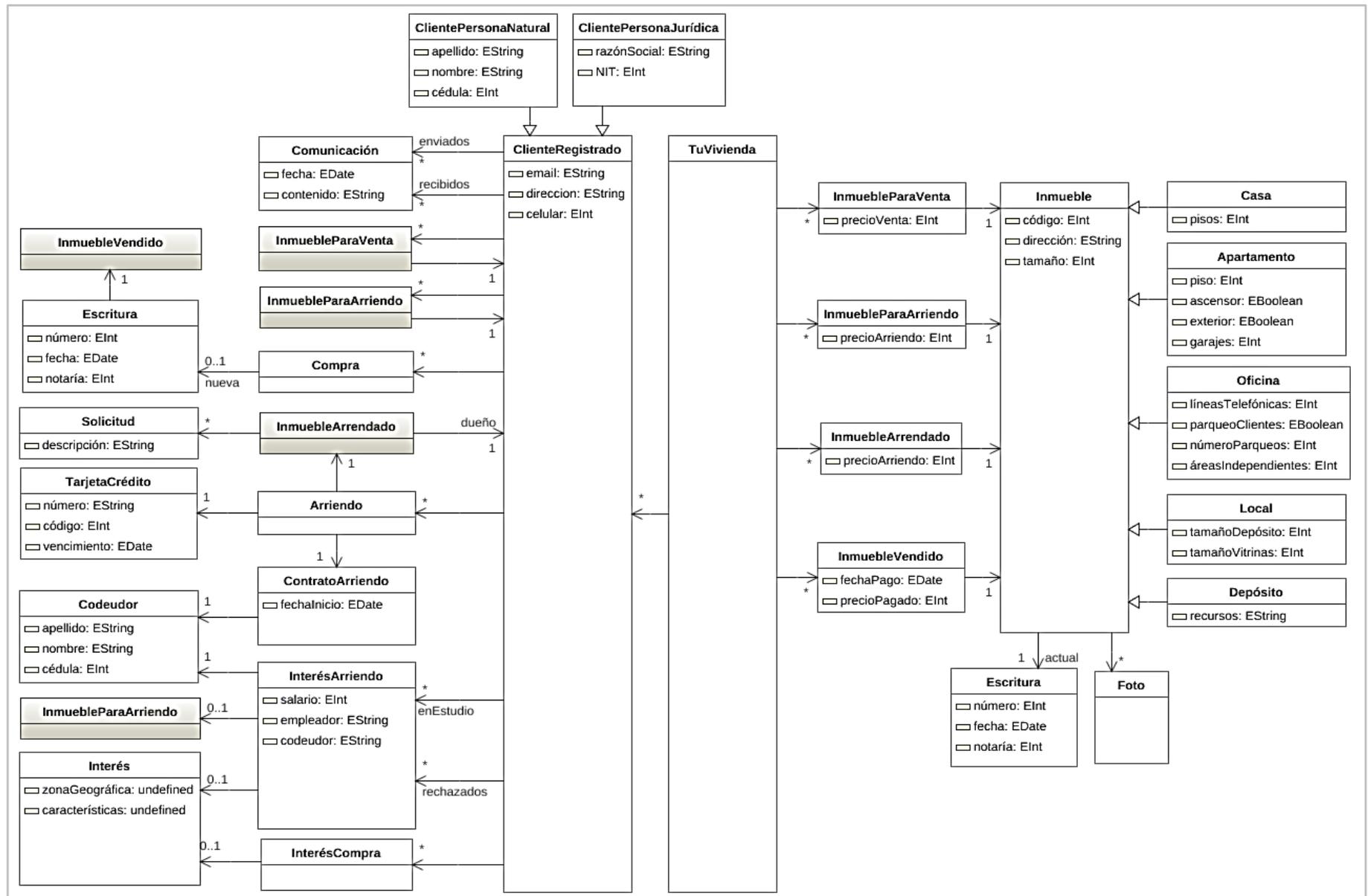
Grupo-1: Recepción de inmuebles. Cuando un cliente entrega en consignación su inmueble, la persona que lo atiende debe solicitar la siguiente información: datos personales del cliente (si es una persona natural, apellido, nombre y cédula; si es una empresa, NIT y razón social), correo electrónico, dirección del inmueble, dirección para correspondencia del cliente, teléfono celular del cliente, datos de la escritura (número, fecha y notaría), tamaño del inmueble en metros cuadrados, tipo de inmueble, tipo de consignación (venta o arrendamiento) y valor (total si es venta o mensual si es arrendamiento). Dependiendo del tipo de inmueble debe solicitar información adicional, de la siguiente manera: si es una casa, el número de pisos. Si es un apartamento, debe informar el piso, si tiene ascensor, si es exterior y el número de garajes. Si es una oficina, debe informar el número de líneas telefónicas, si hay parqueo para clientes, número de parqueos para los empleados y el número de áreas independientes. Si es un local, debe informar el tamaño del depósito asignado y el tamaño de las vitrinas. Si es un depósito, debe hacer una descripción del tipo de recursos con los que cuenta (v.g. refrigeradores, cuartos fríos, estantería, etc.). En cualquier caso, se le asigna un código consecutivo, mediante el cual se referencia cada inmueble. Hay clientes que dejan en consignación varias propiedades. Al cliente se le informa en ese momento los precios que cobra la agencia sobre los trabajos que hace: si es arrendamiento, cobra el 8% de comisión y el 2% de seguro. Si es venta, cobra el 3% del precio total. En todos los casos es posible agregar fotos del inmueble. La información de cada inmueble se publica en el portal (<http://www.tuVivienda.com.co>) y en la aplicación móvil, para que los potenciales clientes puedan consultar los inmuebles disponibles. Todo cliente que deja en consignación un inmueble debe registrarse por la app, medio por el cual se hace la comunicación.

Grupo-2: Selección de clientes. Este grupo se encarga de atender a los compradores o arrendatarios potenciales. En el caso de los arrendatarios, éstos deben dar todos sus datos personales (por el portal y luego deben descargar la app de la empresa y registrarse), su salario, empleador, un codeudor con finca raíz y el inmueble en el cual está interesado (si ya ha seleccionado uno). Una persona del grupo lo entrevista, lo lleva a visitar el inmueble y verifica que todos los datos sean correctos. Cuando se hace el negocio, la misma persona hace todo el trámite de firma de contrato y le entrega las llaves. En ese momento termina su labor. En el caso de los compradores, debe pedir todos los datos personales para poder elaborar la minuta de escritura y llevarla a la notaría. Una vez cancelada la totalidad de la deuda, se firma la escritura y se hace entrega del inmueble. En ese momento el empleado genera la orden para que se haga la transferencia bancaria al dueño, teniendo en cuenta los descuentos y termina su labor. Los compradores y arrendatarios potenciales pueden definir la zona geográfica en la cual tienen interés y las características del inmueble buscado. Tan pronto aparece uno que cumpla con sus requisitos es notificado. La empresa solo arrienda inmuebles con la modalidad de pago automático, razón por la cual el arrendatario debe suministrar los datos de una tarjeta de crédito.

Grupo-3: Administración de arrendamientos: Este grupo de empleados se encarga de hacer la administración de los inmuebles arrendados. Esto es, se encarga de atender las inquietudes y solicitudes de los arrendatarios (reportes y arreglo de daños, por ejemplo). El pago al arrendador se hace el 5 de cada mes, si el cobro al arrendatario ha sido efectivo. En caso de no poder cobrar, la empresa comienza a contar intereses de mora.

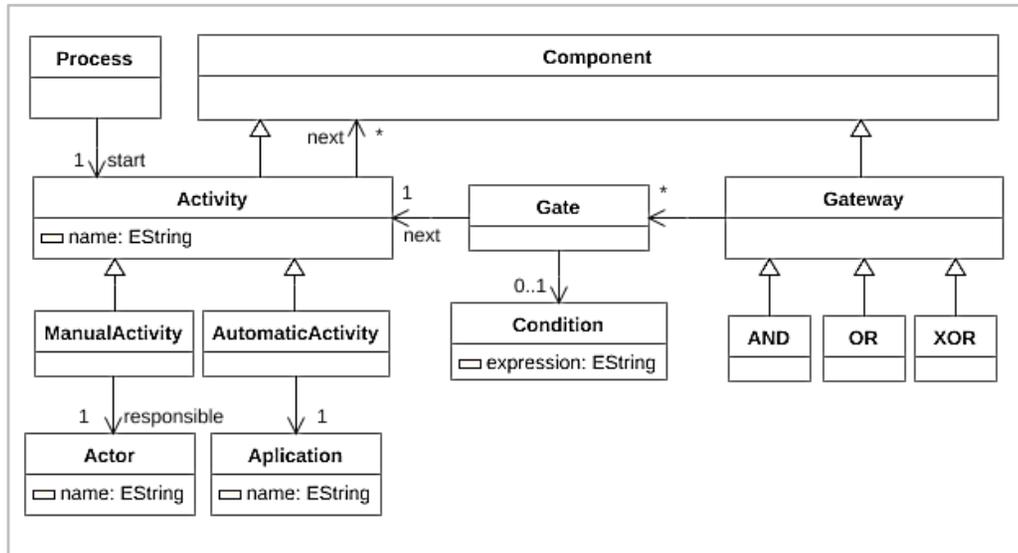
Grupo-4: Cobros judiciales. Este grupo se encarga de hacer los cobros por vía judicial cuando el arrendatario se ha atrasado más de 2 meses en su pago. Cuando un arrendatario debe pagar por este medio, se le cobra un 20% adicional por gastos de abogados.

tuVivienda.com: ejemplo de conceptualización

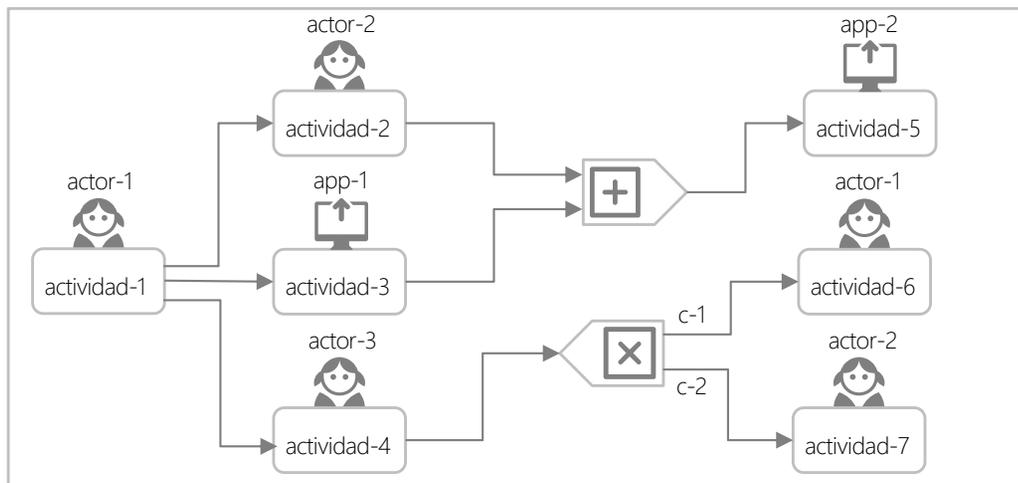


miniBPMN: diseño de un lenguaje

Ahora vamos a diseñar un lenguaje para expresar un proceso de negocio simple (miniBPMN) y lo vamos a especificar usando un metamodelo y las herramientas que hemos visto en esta lectura. Con estos artefactos construidos, podemos generar automáticamente el editor del lenguaje y comenzar a modelar procesos con miniBPMN.

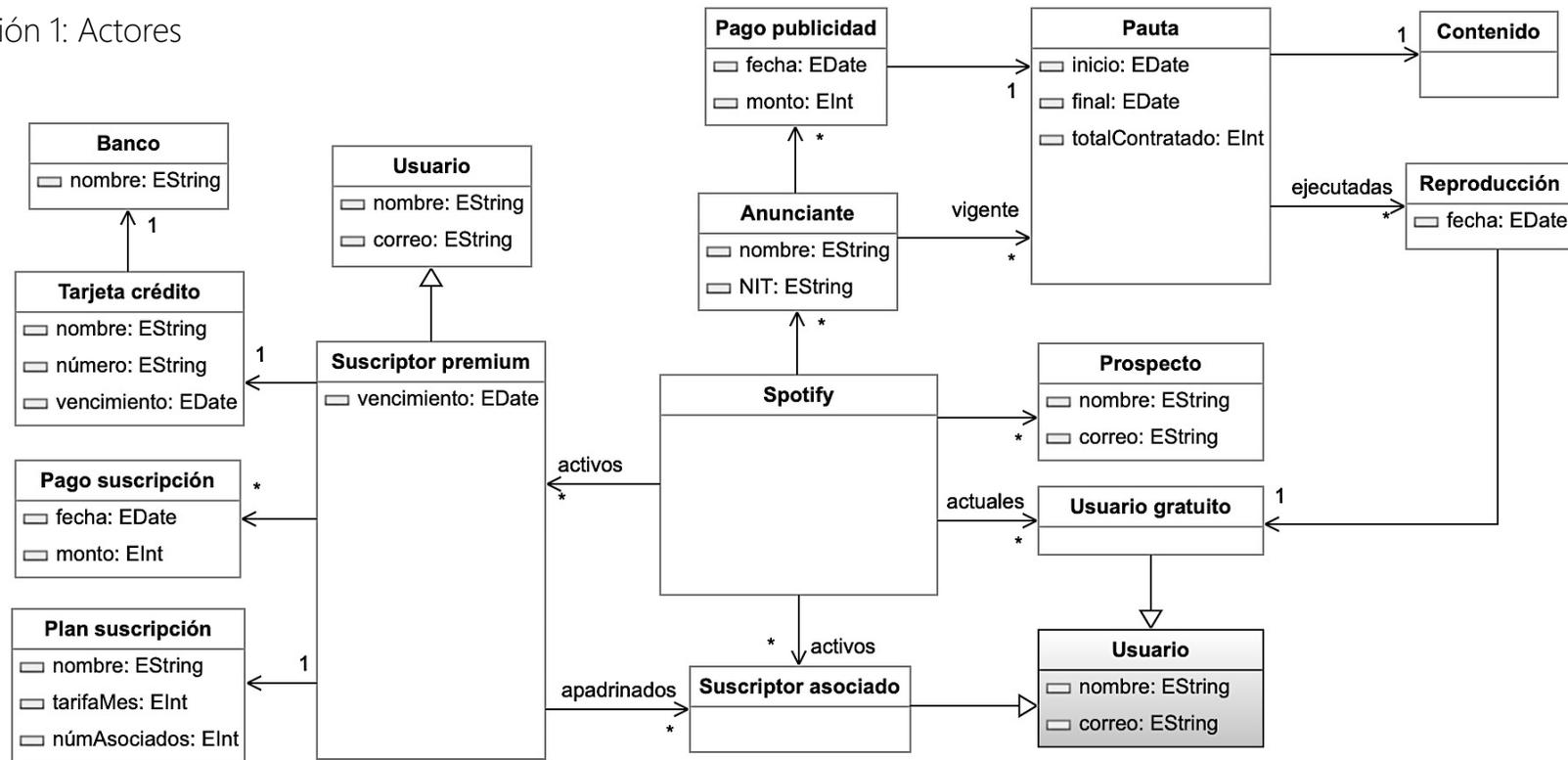


- > Los nombres de los conceptos están en inglés para que sea fácil relacionarlos con los respectivos conceptos de BPMN.
- > El metamodelo afirma que un proceso se describe con una actividad inicial (*start*). Las actividades pueden ser manuales o automáticas. La actividad inicial puede estar conectada con múltiples componentes (*next*), los cuales se especializan en actividades o en compuertas (*Gateway*). Cada compuerta tiene un conjunto de salidas (*Gate*), cada una con una condición opcional y debe tener una actividad con la que se conecta (*next*).
- > Hay muchos aspectos del lenguaje que se deben expresar como restricciones. Por ejemplo, que una compuerta puede tener múltiples entradas y una sola salida (*join*) o una sola entrada y múltiples salidas (*split*). Eso es difícil de expresar en el metamodelo (se complicaría bastante), de manera que lo más recomendable es incluir una tabla con todas las restricciones que queremos imponer.
- > Fíjese que a este nivel no hemos definido el comportamiento de cada concepto (cómo funciona). Esa es la semántica del lenguaje, que se expresa con otros mecanismos distintos a los metamodelos.
- > En el ejemplo hay una propuesta parcial de lenguaje. Escogimos una sintaxis distinta para las compuertas si están en modo bifurcar o en modo unir. Eso se puede hacer poniendo condiciones sobre el estado de cada instancia de la clase. Es muy posible que se pueda diseñar un lenguaje más bonito, ¡no lo dudo!



Spotify®: metamodelo (1)

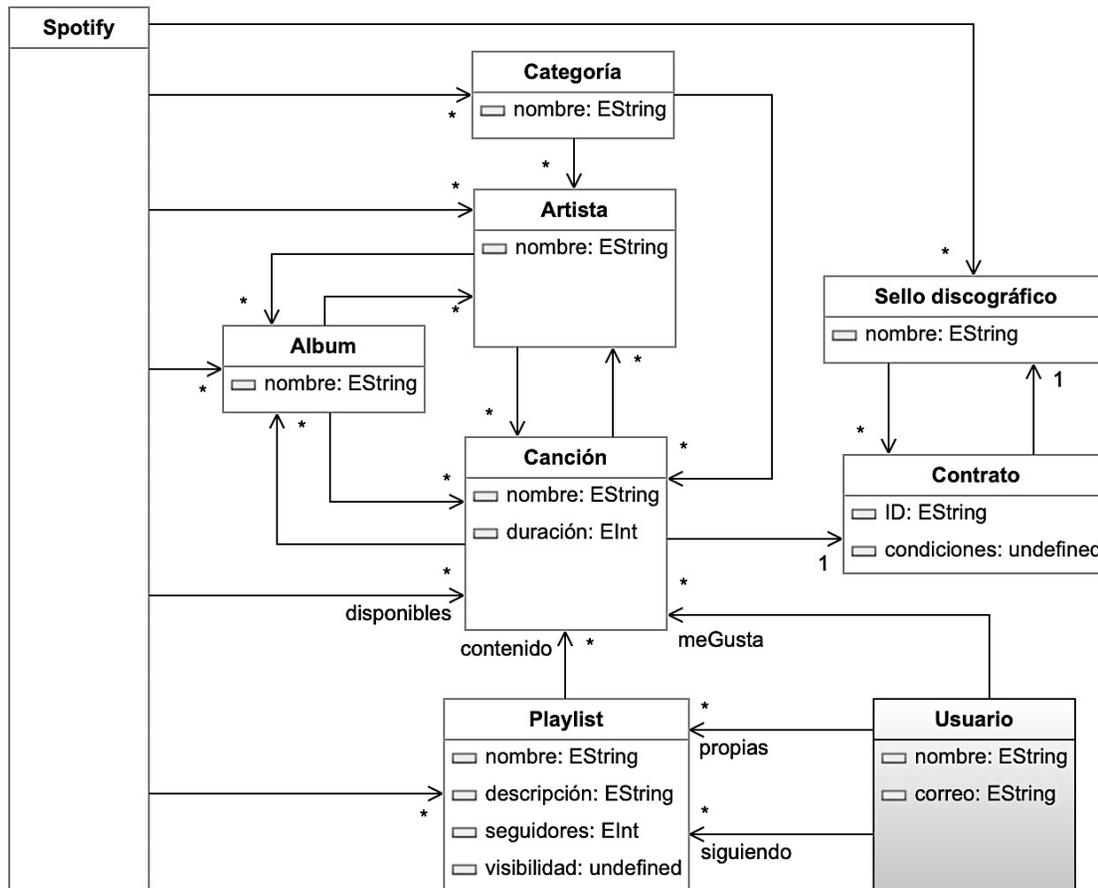
Dimensión 1: Actores



- > En este ejemplo presentamos el metamodelo de la empresa Spotify dividido en dos dimensiones (una en cada página), para ilustrar el proceso de partir un metamodelo complejo.
- > Spotify es un servicio de música en streaming que permite el acceso a millones de canciones y otros contenidos de artistas de todo el mundo. Los usuarios pueden acceder al contenido de forma gratuita o pueden mejorar su cuenta a premium para disfrutar de beneficios tales como la descarga del contenido en sus dispositivos (para poder disfrutarlo off-line), mejor calidad de audio y no escuchar los anuncios que pautan en la plataforma, para lo cual deben pagar una suscripción.
- > En esta primera dimensión se incluye toda la información relacionada con los actores (Anunciante, Suscriptor premium, Suscriptor asociado y Usuario gratuito). Allí encontramos para cada uno de ellos la información que la empresa debe manejar.
- > La raíz del metamodelo (clase Spotify) se encuentra en el centro del diagrama, y debe estar presente en todas las dimensiones del metamodelo.
- > La clase Usuario la usamos como base para derivar de allí los suscriptores y los usuarios gratuitos. Por claridad en el diagrama se repite dos veces. En la otra dimensión aparecen las relaciones que salen de esa clase y que justifican su existencia.

Spotify®: metamodelo (2)

Dimensión 2: Contenido



- > En esta segunda dimensión del metamodelo incluimos lo referente al contenido disponible en la plataforma. De nuevo partimos de la clase Spotify como raíz (a la izquierda).
- > La clase Usuario aparece en otro color (gris) para indicar que es un concepto común a las dos dimensiones. Aquí incluimos los atributos ya definidos en la dimensión 1 (eso es opcional) y las relaciones hacia los conceptos de esta segunda dimensión.
- > La clase Usuario se debe interpretar como la unión de todos los atributos y relaciones que tiene esa clase en todas las dimensiones del metamodelo.
- > En esta dimensión aparecen las canciones disponibles, clasificadas por los distintos conceptos que maneja ese negocio (Artista, Categoría, Álbum, Playlist).
- > Este ejemplo nos sirve para ilustrar la idea que se mencionó en la sección anterior de esta Lectura, de no crear los modelos completos de la empresa, cuando eso es innecesario. En el metamodelo tenemos todos los detalles de los actores y de las canciones (los necesitamos para entender el negocio), pero cuando vayamos a hacer el modelo, no vamos a copiar las bases de datos completas de Spotify en unos diagramas o formatos, porque no tendría sentido. Además de que esos modelos cambian cada segundo, y nunca tendríamos una versión del modelo que se pueda considerar estática.

8. Hojas de trabajo

Hoja de trabajo #1	
Objetivo:	Conceptualizar el dominio de negocio de algunas empresas conocidas y practicar creando distintos puntos de vista.
Tareas:	<ul style="list-style-type: none"> • Construya el metamodelo de cada una de las empresas que se plantean más adelante. Intente incluir al menos 10 conceptos en cada caso. • Especifique varios puntos de vista que tengan al menos 3 conceptos, una regla y un lenguaje (gráfico y/o textual). Para cada punto de vista defina el perfil de los posibles interesados (a quién estaría dirigido).
Empresas	
<ol style="list-style-type: none"> 1. Netflix 2. Amazon (Marketplace) 3. WhatsApp 4. Rappi 5. Spotify 6. YouTube 7. Facebook 8. Merqueo 9. Tinder 10. Legis 11. El Tiempo 12. Google (Buscador) 	<ol style="list-style-type: none"> 13. Tostao 14. Cosechas 15. Ecopetrol 16. Banco de Colombia 17. Avianca 18. Claro Móvil 19. Nueva EPS 20. Quala 21. Uber 22. McDonald's 23. FedEx 24. SNCF
<p>Toda la información que se necesita de las empresas se consigue sin problema por Internet.</p>	

Hoja de trabajo #2	
Objetivo:	Conceptualizar algunos dominios conceptuales importantes en arquitectura y diseñar los lenguajes gráficos adecuados para crear los modelos.
Tareas:	<ul style="list-style-type: none"> • Construya el metamodelo de cada uno de los dominios que se plantean más adelante. Intente incluir al menos 10 conceptos en cada uno. • Diseñe un lenguaje gráfico para cada metamodelo, pensando en su usabilidad. • Construya un modelo usando el lenguaje diseñado. Intente utilizar todos los elementos del lenguaje.
Dominios	
<ol style="list-style-type: none"> 1. Un diagrama de secuencia (UML) 2. Un diagrama de colaboración (UML) 3. Un diagrama entidad-relación 4. Un archivo en XML 5. Una base de datos no relacional 6. Un modelo estratégico (BMM) 7. Un modelo de negocio (<i>Canvas</i> de Osterwalder) 8. Una arquitectura de infraestructura 9. Una arquitectura de información 10. Un mapa de ruta (TOGAF) 11. Un bus de servicios (ESB) 12. Una bodega de datos (data warehouse) 13. Un <i>data lake</i> 14. Una arquitectura de integración de aplicaciones 15. Un motor de procesos 16. La plataforma <i>Kubernetes</i> 17. Un <i>chatbot</i> 	

Food Trucks de los Alpes: hoja de trabajo #3

Con 5 años en el mercado, Food Trucks de los Alpes (FTA) se ha ganado el reconocimiento de todos sus clientes por su amplia oferta gastronómica y su innovadora propuesta de ambientación. Ubicado en uno de los sectores más reconocidos de la ciudad, FTA cuenta con más de 20 Food Trucks que en conjunto, generan ventas por más de \$650 millones de pesos al mes. Para garantizar a sus clientes la mejor experiencia gastronómica y social, FTA está organizado alrededor de 4 ambientes: Clásico, Bienestar, Sugar Rush y Jubileo. Cada uno de estos espacios se compone de un grupo de food trucks y un área de socialización. El área de socialización cuenta con sillas, mesas, cojines y barras para que los clientes puedan disfrutar el ambiente. Así mismo, cada ambiente cuenta con una decoración propia de su temática y una zona VIP para clientes fieles. Aunque en sus inicios FTA solo se concebía como un espacio donde los clientes podían disfrutar alimentos, a la fecha se ha convertido en un espacio integral. Además de ir a comer, un cliente puede ir a FTA a trabajar, a compartir con sus amigos e incluso, a celebrar ocasiones especiales. Teniendo en cuenta las distintas necesidades que puede tener un cliente de FTA, la compañía ofrece a sus clientes la posibilidad de fidelizarse bajo 2 modalidades: Favorito y Completo. Un cliente favorito tiene derecho a descuentos especiales, acceso a la zona VIP y premios de los food trucks de su ambiente favorito. Un cliente completo tiene los mismos derechos solo que se aplican a los 4 ambientes de FTA. La fidelización favorito tiene un costo de \$10 mil pesos mensuales, mientras que la fidelización completa tiene un costo de \$15 mil pesos mensuales.

Todos los food trucks dentro de FTA son propiedad de la compañía, y son operados por empleados contratados directamente por FTA. Cada food truck cuenta con dos cocineros, un cajero y un anfitrión. Así mismo, cada food truck tiene una cocina equipada con una parrilla, un horno, una nevera, una caja y una pequeña bodega. Los productos de los food trucks tienen precios que oscilan entre los \$4.000 y los \$20.000 pesos. Un cliente puede ordenar el producto del food truck directamente en el food truck, o por medio de los puntos de autoservicio ubicados en el área de socialización de cada ambiente.

El proceso de atención de una orden por autoservicio sigue los siguientes pasos. Primero, un cliente se acerca al punto de autoservicio y digita su número de identificación y su nombre. Seguidamente selecciona el food truck de su preferencia, selecciona el producto del menú y confirma su orden. Cuando el cliente confirma la orden, esta es recibida por el cajero del truck mientras que el cliente queda a la espera de su llamada. Cuando el cajero recibe la orden, imprime el comprobante y lo pone en la cola de ordenes. Pasados 20 minutos, el cajero recibe el producto, y verifica si el cliente está fidelizado. Si lo está, el cajero verifica si el producto tiene descuento y si lo tiene, aplica el descuento, recalcula el precio de la orden y llama al cliente. En caso de que el producto no tenga descuento, o si el cliente no está fidelizado, el cajero llama al cliente. Cuando el cliente recibe el llamado, se acerca al food truck y confirma el método de pago. Si es en efectivo, el cliente entrega el dinero al cajero quien lo guarda en la caja y entrega el producto al cliente finalizando el proceso. De lo contrario, el cliente entrega la tarjeta, el cajero la pasa por el datáfono, y luego entrega el producto para finalizar el proceso.

Para asegurar la operación de los distintos food trucks, FTA ha constituido una red de proveedores nacionales e internacionales. Estos proveedores llevan los insumos directamente a cada uno de los food trucks para ser almacenados posteriormente por el anfitrión del food truck. Entre estos insumos se encuentran los ingredientes de los alimentos y los desechables en los que se sirven. Así mismo, la compañía cuenta con un área de TI encargada de la administración de los puntos de servicio, el portal web y los sistemas que apoyan la operación de la compañía. Para asegurar el mantenimiento de los food trucks y los ambientes, FTA tiene subcontratados servicios de limpieza y mantenimiento. Finalmente, FTA cuenta con un área administrativa en la que se encuentran los empleados encargados del área de finanzas, recursos humanos y marketing y ventas. El salario promedio de un empleado del food truck se encuentra alrededor de los \$900.000 pesos mientras que el de un empleado del área administrativa se encuentran en \$1.500.000

Mess-Away: hoja de trabajo #4

Bajo su premisa “no te preocupes por el desorden, nosotros lo limpiamos”, Mess-Away ofrece limpieza del hogar a domicilio. La empresa ofrece su servicio de limpieza a domicilio bajo 3 modalidades: S.O.S, Spring Clean y Helper. En la modalidad S.O.S el cliente contrata un servicio de limpieza de urgencia. Este servicio es 24/7 y suele contratarse después de fiestas o siniestros en el hogar (como inundaciones). Una vez el cliente solicita el servicio, la compañía se compromete a una franja de respuesta inferior a 3 horas (tiempo comprendido desde el momento en que el cliente reserva el servicio hasta que alguien llega al lugar donde debe limpiar). El precio del servicio varía de acuerdo a la hora en la cual el servicio es reservado, la cantidad de insumos utilizados y el tiempo que tome limpiar el lugar. En la modalidad Spring Clean un cliente contrata el servicio para realizar una limpieza de fin de año. En este caso el cliente contrata el servicio por un número de días (mínimo 1) y la empresa se compromete a limpiar y ordenar todo el lugar. En dado caso que la limpieza tome más días que los reservados, el cliente puede contratar el servicio hasta por 3 días adicionales. Finalmente, en la modalidad Helper un cliente paga una suscripción anual para que un equipo de limpieza de Mess-Away realice la limpieza del hogar periódicamente. El periodo de tiempo varía de acuerdo a la suscripción y además de elegir los días en los que se desea tener el servicio, un cliente puede configurar sus preferencias para personalizar el servicio. Este tipo de preferencias incluyen el tipo de productos a utilizar, el tiempo que debería tardar la limpieza y la descripción de los espacios a limpiar (cuartos, baños, cocina). Para contratar los servicios de Mess-Away un cliente debe ingresar a la aplicación o al portal web. Si no se encuentra registrado, debe ingresar sus datos personales (nombre, tipo y número de identificación, teléfono de contacto y dirección), el método de pago, y una contraseña para poder iniciar sesión. Seguidamente, el cliente debe elegir el servicio a contratar. En el caso de un servicio S.O.S o Spring Clean el cliente debe reservar el día y hora en el que espera recibir el servicio, y en el caso del servicio Helper el cliente debe elegir el tipo suscripción (semanal, quincenal o mensual), sus preferencias, los días en los que quiere recibir el servicio (por ejemplo el segundo fin de semana de cada mes), y realizar el pago de la suscripción.

Para los pagos, Mess-Away le permite a sus clientes pagar con PSE, tarjeta de crédito o efectivo. Este último se puede realizar en puntos Baloto o directamente con personal de la compañía. Es importante aclarar que cuando un cliente solicita un servicio S.O.S o Spring Clean debe cancelar el 50% de la tarifa base antes de recibir el servicio. Dependiendo de la complejidad del servicio de limpieza, una vez termine el servicio, el cliente debe pagar el 50% restante de la tarifa base y la tarifa adicional si aplica. Para brindar sus servicios de limpieza Mess-Away tiene un proveedor de productos de limpieza que cada 30 días lleva a la bodega central el pedido de productos. Una vez se recibe el pedido en bodega, un administrador se encarga de revisar el pedido y acomodar los productos. Para realizar el pedido Mess Away realiza un pronóstico mensual que contempla los servicios contratados el mes anterior y la orden de productos por parte de los clientes pues, además de sus servicios, Mess-Away también vende productos de limpieza a través de su tienda virtual. Un cliente puede adquirir estos productos por medio de la aplicación o el portal y estos últimos son llevados a domicilio al hogar del cliente. Además de su proveedor, Mess-Away cuenta con 50 empleados que conforman los equipos de limpieza. Diariamente, el líder del equipo debe verificar los servicios asignados para el día y acercarse a la bodega central para recoger los insumos y la maquinaria de limpieza si aplica (aspiradora y/o lavadora de tapetes). Luego, el líder del equipo debe recoger a los miembros del equipo de limpieza en la oficina central (para esto cuenta con una van de Mess-Away que es entregada por un encargado en la bodega) y dirigirse a los puntos de servicio asignados. En el caso de los servicios S.O.S el personal se encuentra asignado dos veces al mes en turno de noche y una vez reciben la notificación del servicio, el líder del equipo debe recoger los insumos en la bodega y a los miembros del equipo en un punto acordado. En la oficina central de Mess-Away trabajan 20 empleados administrativos de base, así como un equipo de 5 desarrolladores que se encargan de mantener la aplicación y el portal web. En la oficina también se encuentra un equipo de marketing encargado de gestionar la publicidad de la compañía, las redes sociales y las campañas promocionales dirigidas a sus distintos clientes.

Smoothies de los Alpes: hoja de trabajo #5

Smoothies de los Alpes es una compañía del grupo empresarial de Los Alpes que desde el 2016, se dedica a la venta de smoothies a domicilio en la ciudad de Bogotá.

La compañía fue fundada como respuesta a la creciente demanda de alimentos saludables y nutritivos, y, a pesar de encontrarse en un entorno con una amplia oferta de productos similares, logró captar el 60% del mercado (valorado en \$10.000 millones). Dicha respuesta se debe al innovador modelo de negocio que viene detrás de la venta de smoothies.

Smoothies de los Alpes opera bajo un modelo de suscripción, en el cual los clientes se suscriben para recibir los smoothies de su preferencia en la puerta de su casa. Los clientes arman una canasta con los smoothies de su preferencia y cada día reciben una caja con los ingredientes necesarios para preparar su smoothie. Lo único con lo que deben contar es con una licuadora y una base para el smoothie (agua, leche o yogurt). Para suscribirse al servicio, un cliente accede al portal web o a la aplicación de Smoothies de los Alpes y se inscribe con sus datos personales.

Seguidamente, escoge la suscripción de su preferencia. En este momento se cuentan con 3 tipos de suscripciones mensuales:

- EntreSemana (\$7.000 pesos / smoothie): El cliente recibe un smoothie diario de lunes a viernes por un mes (20 días).
- LongWeek (\$ 6.500 pesos / smoothie): El cliente recibe un smoothie diario de lunes a sábado por un mes (24 días).
- 24/7 (\$6.000 pesos / smoothie): El cliente recibe un smoothie diario durante el mes (30 días).

Después de escoger el tipo de suscripción, el cliente procede a seleccionar los smoothies de su preferencia. Para esto, el cliente cuenta con una canasta virtual que puede llenar con los smoothies que desea recibir. Dependiendo de la suscripción el cliente puede escoger hasta un máximo de 8 smoothies.

Existen 4 categorías de smoothies de las cuales es posible escoger:

- Fruity&Fresh: los cuales se realizan a base de frutas.
- Greens: los cuales se realizan mezclando frutas y verduras.
- Fit: los cuales se realizan con mezclas de frutas, verduras y alimentos ricos en proteínas (nueces, mantequilla de maní, semillas de chía, etc).
- DIY: El cliente arma su propio smoothie mezclando hasta 5 ingredientes.

Luego de armar su canasta el cliente selecciona los días en los que desea recibir su smoothie y finalmente realiza el pago de su suscripción por PSE o tarjeta de crédito. Al realizar el pago, el cliente puede seleccionar si desea renovar su suscripción automáticamente una vez caduque, o si desea recibir una notificación cuando esté próxima a vencerse.

La operación de Smoothies de los Alpes se concentra en su sede central en Bogotá. Los ingredientes de los smoothies se obtienen de agricultores de los municipios aledaños o de plazas de mercado. Algunos ingredientes son importados de proveedores en Estados Unidos.

Para asegurar el despacho de los más de 2000 batidos diarios, Smoothies de los Alpes cuenta con una red de distribución conformada por 80 distribuidores que reparten los batidos en 40 camiones por toda la ciudad. En la sede central, se cuenta con 30 empleados encargados de preparar las cajas para despachar, 20 empleados encargados de la toda la parte administrativa de la compañía y un equipo de 15 desarrolladores encargados de mantener la página web y la aplicación. Estas últimas se encuentran soportadas en la nube.

Así mismo, Smoothies de los Alpes cuenta con un equipo de marketing de 5 personas encargadas de gestionar las relaciones con los clientes. En particular, el equipo se encarga del manejo de redes sociales, publicidad en medios de comunicación y el diseño de campañas para incentivar la suscripción de nuevos clientes y las renovaciones de los antiguos.

Marky: hoja de trabajo #6

Marky es una empresa de mercadeo digital, especializada en traer nuevos clientes a cualquier tipo de empresa que cuenta con una tienda virtual. A cambio de llevar clientes nuevos, Marky se queda con el 5% de todas las compras (presenciales o virtuales) que hagan estos clientes durante dos años. La empresa cliente no paga ningún valor por el servicio. Al firmar el contrato, la empresa cliente entrega toda la información de sus productos y de sus clientes actuales. A partir de esto un grupo de expertos de Marky hace un análisis de la empresa cliente, de sus productos, de su imagen y de su portal. Con esta información hace un conjunto de recomendaciones a la empresa cliente (que ésta debe implementar) y define una estrategia, que puede consistir en publicidad en redes sociales, en correos directos, en mensajes de texto o llamadas telefónicas a clientes potenciales. También se pueden crear las páginas de la empresa cliente en redes sociales (como Facebook) y manejar desde ahí la comunicación con los clientes potenciales y nuevos. La empresa cliente no hace ninguna actividad de mercadeo por su cuenta y entrega a Marky información en tiempo real de todas las ventas que hace de manera virtual o presencial, detallando qué cliente compra qué, por qué canal y si fue contactado por Marky. Con esto Marky hace seguimiento de los resultados obtenidos y ajusta su estrategia. La especialidad de Marky es enviar a clientes potenciales bonos de compra de la empresa cliente, que se pueden redimir en su tienda virtual. Pueden ser descuentos, o productos adicionales que recibe el cliente por su compra. Al redimirlos, se le pide al cliente información personal y el permiso para usar esa información en las actividades de mercadeo de Marky. En la actualidad Marky tiene alrededor de 100 empresas cliente por todo el país y más de cuatro millones de clientes perfilados. Todo el tiempo Marky está comprando bases de datos de personas o empresas nuevas, sobre las cuales trabaja. Cuando la empresa cliente es pequeña (una tienda, por ejemplo), se imprimen y reparten volantes puerta a puerta en un radio de un par de kilómetros a la redonda de la sede física de la empresa cliente. En estos volantes se incluye información y eventualmente bonos de compra. Marky está estructurada en cinco grandes departamentos: clientes (analítica, campañas, estrategia), comunicaciones (call-center, redes sociales, mensajes), tecnología

(desarrollo y soporte), comercial (nuevas empresas cliente) y administración (oficina jurídica, recursos humanos, financiera y tesorería, servicios generales). Tiene un edificio propio desde donde opera, aunque la mayoría de sus empleados trabajan de manera remota. Para el 2021 su estrategia está focalizada en empezar a operar en otros países (internacionalización).

Marky cuenta con un CRM en el que se registran todas las interacciones que tiene con los clientes potenciales. Sobre ese componente se ejecutan múltiples algoritmos de machine learning, de donde se extraen recomendaciones y se sacan conclusiones sobre las estrategias de mercadeo utilizadas. Como todos los costos asociados con las promociones y bonos corren por cuenta de Marky, se deben seleccionar con mucho cuidado para garantizar su impacto. En el contrato con las empresas cliente se tiene que al final de cada mes, se hace el pago por el 5% de las compras de los nuevos clientes. Para esto Marky les envía una factura con todos los soportes y a los 30 días máximo recibe el pago. Los algoritmos de análisis de información han sido desarrollados por Marky con su equipo de tecnología y se consideran uno de los activos más importantes de la compañía. Estos algoritmos se ejecutan en la nube, cuyos recursos tecnológicos provee Amazon. Al terminar el contrato con una empresa cliente, Marky le entrega toda la información que tiene de los clientes y le sigue pasando facturas hasta que el último cliente nuevo incorporado por Marky completa 2 años. La oficina jurídica de Marky, que hace parte del departamento de administración, funciona con un conjunto de abogados contratados a la empresa LawyerTech, quienes revisan todos los contratos y atienden los pleitos que puedan surgir. Los servicios de vigilancia y aseo se encuentran tercerizados con la empresa ServiciosIntegrados. El área de recursos humanos de Marky busca permanentemente talento, focalizado sobre todo en expertos en marketing digital, en desarrollo de software y en todos los temas relacionados con inteligencia artificial. Marky es una empresa del grupo Empresarial de los Alpes y es dirigida por un gerente general y un subgerente por cada uno de los departamentos que tiene.



Jorge Villalobos, PhD
Profesor titular
Bogotá, Colombia

jvillalo@uniandes.edu.co
<https://profesores.virtual.uniandes.edu.co/jvillalo>

