

# Desarrollo de Ontologías

---

Javier Béjar

ECSDI - 2021/2022 2Q

CS-GEI-FIB 



# Elementos de un ontología

---

- ⊙ Una ontología es una descripción formal explícita de los **conceptos** (o **clases**) de un dominio
- ⊙ Los conceptos se describirán a partir de **propiedades** que representarán los **atributos**, **características**, **relaciones** y **correspondencias** entre los conceptos

- ⊙ **Relaciones:** Composición de conceptos  $R : C_1 \times C_2 \times \cdots \times C_n$

$$Padre\_de : Persona \times Persona$$

- ⊙ **Funciones:** Correspondencia entre conceptos  $F : C_1 \times C_2 \times \cdots \times C_n \rightarrow C_i$

$$preciobillete : Destino \times Fecha \times Fecha \rightarrow Numero$$

- ⊙ Adicionalmente, estas propiedades tendrán **restricciones** que delimitan diferentes características que las definen (dominio, rango, cardinalidades...)
- ⊙ Las **instancias** serán elementos identificables que constituirán los individuos concretos que representa la ontología
- ⊙ Los **axiomas** definen reglas de inferencia que consideramos ciertas para el dominio

- ⊙ Para poder **razonar** sobre los elementos de las ontologías necesitamos **definir una lógica** adecuada
- ⊙ La lógica de primer orden es demasiado expresiva para este propósito
- ⊙ **Description Logics** permite razonar sobre clases, subclases, instancias y definiciones
- ⊙ Es una restricción de lógica de primer orden con una visión conjuntista
- ⊙ Relacionada con la semántica de la orientación a objetos

# Metodologías de desarrollo

---

- ⊙ El proceso del desarrollo de una ontología requiere:
  - Definir las **clases** que forman el dominio
  - Organizar las clases en una **jerarquía taxonómica**
  - Definir las **propiedades** de cada clase e indicar las **restricciones** de sus valores
  - Asignar valores a las propiedades para crear **instancias**



- ⊙ Como cualquier pieza de software, se pueden aplicar metodologías de desarrollo de software a la construcción de ontologías
- ⊙ Estas metodologías deben ser adaptadas a las particularidades especiales que tienen las ontologías
- ⊙ **No existe una metodología estándar**
- ⊙ Diferentes proyectos reales de desarrollo de ontologías han dado lugar a diversas aproximaciones

- ⊙ “*Ontology Development 101: A Guide to Creating Your First Ontology*”, Noy & McGuinness, (2000)
- ⊙ Debemos tener en cuenta:
  1. **No existe un modo correcto** de modelar un dominio. La mejor solución dependerá de la aplicación/problema concreto
  2. El desarrollo de una ontología es un **proceso iterativo**
  3. Los elementos de la ontología deberían ser **cercanos a los conceptos y relaciones que se usan para describir el dominio** (generalmente se corresponden a nombres y verbos que aparecen en frases que describen el dominio)

- ⊙ **Fase 1:** Determinar el **dominio** y la **cobertura** de la ontología
  - ¿Qué dominio cubrirá la ontología?
  - ¿Para qué usaremos la ontología?
  - ¿A que tipos de preguntas ha de poder responder la ontología?
  - ¿Quién usará y mantendrá la ontología?

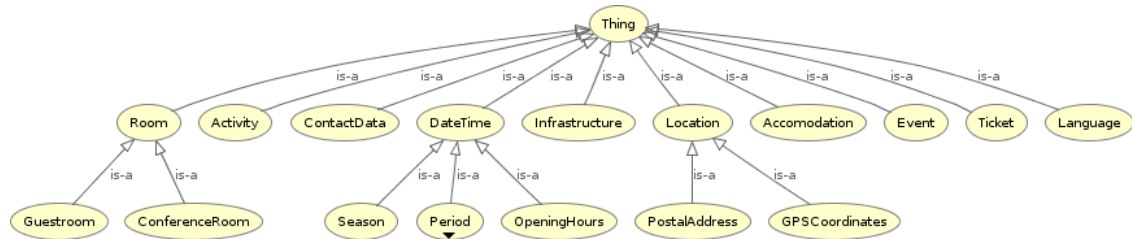
- ⊙ Desarrollar una aplicación capaz de **recomendar un plan para turistas** que quiere pasar unos días en una ciudad
- ⊙ La ontología debería incluir los diferentes **lugares que puede visitar** el turista incluyendo **actividades** culturales y de diversión
- ⊙ Deberíamos poder conocer los **detalles que describen** los diferentes lugares como por ejemplo horarios, compra de entradas, precio, abonos, ... y como llegar a ellos
- ⊙ La ontología **podría ser reusada** en otras aplicaciones relacionadas

### ⊙ **Fase 2:** Considerar la **reutilización** de ontologías existentes

Las ontologías se construyen para **comunicar conocimiento** en dominios, por lo que se desarrollan con la idea de compartición.

No es necesario rehacer un trabajo que ya está hecho, **si existe** una ontología sobre el dominio en el que trabajamos, **podemos incorporarla**

- ⊙ **Buscando** en Swoogle podemos encontrar 5 ontologías que incluyen el término turismo
- ⊙ Dos son **demasiado generales**: Viajes (TravelOntology.owl), elementos de una ciudad (OTN.owl)
- ⊙ Una **demasiado específica** e incompleta (tourism.owl)
- ⊙ De las dos más adecuadas, una es **muy compleja** (ETP-tourism.owl) y otra tiene **elementos que no** nos interesan (e-tourism.owl)
- ⊙ Podemos **aprovechar algunos conceptos** de estas dos últimas



⊙ **Fase 3:** Enumerar los **términos importantes** en la ontología

Escribir una lista de términos que podemos usar para referirnos a nuestro dominio, **elaborando frases** que podríamos utilizar para **preguntarnos cosas** sobre él o para **explicar a alguien** información sobre él.

- ¿Que **propiedades** tiene esos términos?
- ¿Que nos **gustaría decir** sobre ellos?

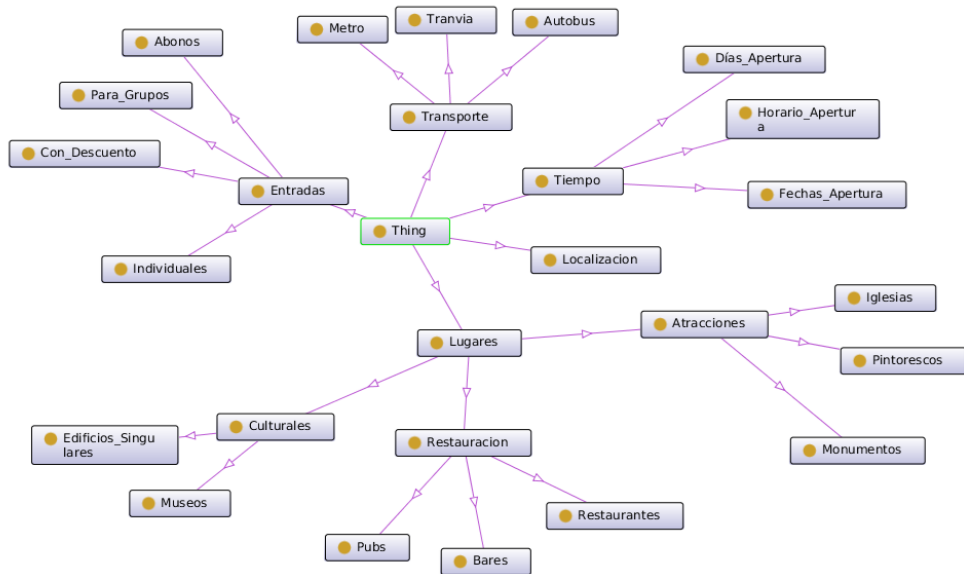


- ⊙ En nuestra aplicación **hablaremos de:**
  - Atracciones, servicios, lugares, eventos...
  - Entradas, pagos, descuentos...
  - Horarios, localizaciones, transporte...
- ⊙ Nos **gustaría decir/saber**
  - Que horario de visita tiene un lugar, una atracción, un servicio, ...
  - Donde esta un lugar, atracción...
  - Es necesario pagar (¿como?) para ir/entrar en un lugar, atracción
  - Cual es el tipo de un servicio
  - Como se va a un lugar, atraccion...
  - ...

- ⊙ **Fase 4:** Definir las **clases** y su **jerarquía**, aproximaciones:
  - **De arriba a abajo:** Definimos los conceptos más generales y vamos especializándolos
  - **De abajo a arriba:** Definimos las clases más específicas y la agrupamos según propiedades comunes generalizando
  - **Combinación de ambas:** Definimos los conceptos más importantes y especializamos y generalizamos para completar la ontología

**Ninguno** de estos métodos **es esencialmente mejor** y depende del dominio

- ⊙ Escogemos generar la jerarquía de arriba a abajo
- ⊙ Decidimos los conceptos más generales y los especializamos:
  - Lugares (culturales, entretenimiento, atracciones, eventos...)
  - Localizaciones
  - Tiempo (horarios, fechas, días)
  - Transporte (autobús, metro, tranvía, taxi)
  - Entradas (abonos, individuales, con descuento...)



⊙ **Fase 5:** Definir las **propiedades** de las clases

- Debemos describir la **estructura interna** de las clases
- Determinaremos una lista de **características** y en que clases debemos tenerlas
- Podemos tener muchos **tipos de propiedades**
  - Propiedades descriptivas, cualidades
  - Propiedades identificadoras, nombres
  - Partes
  - Relaciones con instancias de otras clases

- ⊙ **Propiedades identificadoras:** Nombre de lugar, línea de transporte...
- ⊙ **Propiedades descriptivas:** hora, calle, descripción de lugar...
- ⊙ **Relaciones:**
  - Un transporte y un lugar tienen un horario de apertura
  - Un lugar tiene una localización
  - Museos, edificios singulares e iglesias tienen unos días/fechas de apertura
  - Los lugares culturales necesitan una entrada
  - Los lugares tienen asociados medios de transporte
  - ...

⊙ **Fase 6:** Definir las **características de las propiedades**

- Cardinalidad (número de valores permitidos)
- Tipo, valores
- Valores por defecto
- Obligatoriedad
- Si es una relación definir la cardinalidad y el rango

- ⊙ Un lugar tiene solo una localización y un horario de apertura
- ⊙ Un monumento puede o no necesitar una entrada
- ⊙ Una entrada tiene un precio con valor real (podría ser 0)
- ⊙ Una entrada abono tiene un tiempo de validez expresado con un número (días)



### ⊙ Fase 7: Crear **instancias**

Si es necesario (en algún momento lo será) crear las instancias que formarán parte de la ontología

Esto incluye las instancias estáticas del dominio (constantes y objetos)

# Principios de desarrollo

---

## ⊙ Claridad y objetividad

- Cada concepto de la ontología debe ser definido objetivamente y con una semántica lo más cercana posible a su representación en el lenguaje

## ⊙ Completitud

- Los términos se deben definir a partir de condiciones suficientes y necesarias

## ⊙ Consistencia

- Se ha de asegurar que las deducciones obtenidas a partir de la ontología son consistentes con las definiciones de los conceptos

## ⊙ Extensibilidad monótona

- Incluir nuevas especializaciones y generalizaciones en la ontología no implica revisar conceptos ya definidos

## ⊙ Principio de distinción ontológica

- Las clases de la ontología son disjuntas

## ⊙ Diversificación

- Diversificación de las jerarquías incluidas para aprovechar la potencia de la herencia múltiple

## ⊙ Modularidad

- Reducir la interacción entre módulos semánticamente disjuntos

## ⊙ Estandarización de nombres

- Definir y seguir un estándar para nomenclatura de los conceptos

## ⊙ Minimización de la distancia semántica

- Minimizar la distancia semántica entre conceptos fuertemente relacionados.  
Conceptos similares deben estar agrupados y representados usando los mismos elementos primitivos

- ⊙ Definir una **forma común para denominar** las clases (por ejemplo, no usar nombres en singular y plural)
  - e.g. Sería confuso tener una clase *personas* y otra *animal*, el plural podría tomarse como un concepto para grupos y el singular para individuos
- ⊙ **Los nombres no son las clases**, debemos distinguir la clase del nombre que le damos. Podemos tener sinónimos, pero todos representan a la misma clase
  - e.g. podemos tener una clase llamada *personas* y otra *humanos*, pero sabiendo que se refieren a las mismas entidades

- ⊙ Observar las relaciones de **transitividad** y comprobar si son correctas (evitarlas si podemos)
  - e.g.: podemos tener una relación *ancestro* que sea transitiva, pero una consulta sobre todos los ancestros de una persona puede llevar a una explosión combinatoria
- ⊙ Asegurarnos de que la **jerarquía está correctamente** construida (por ejemplo, sin ciclos)

- ⊙ Todas las **subclases** de una clase deben estar **al mismo nivel de generalidad**
  - e.g. podemos tener una clase *persona* que se especialice en *niños*, *jóvenes*, *adultos* y *ancianos*, pero no tendría sentido añadir una especialización que fuera *jóvenes europeos*
- ⊙ No hay un criterio respecto al **número de clases**, la experiencia dice que un número entre **dos y doce** es habitual, más clases indicaría que debemos estructurarlas añadiendo más niveles



## ⊙ ¿Cuando introducir nuevas clases?

Suele ser incómodo navegar por jerarquías o muy planas o muy profundas, se debería elegir un punto intermedio:

- Las nuevas clases tienen **propiedades adicionales** que no tiene la superclase
- Tienen **restricciones diferentes**
- Participan en **relaciones diferentes**

e.g. en un dominio médico puede no tener sentido especializar por el estado civil de una persona, pero en uno sobre impuestos sí

- ⊙ Decidir si hemos de usar una **propiedad** o crear una **clase**

A veces un **atributo** es **suficientemente importante** como para considerar que sus valores diferentes corresponden a objetos diferentes

e.g. en un dominio médico ser un niño o un adulto lleva a decisiones diferentes, por lo que se les puede considerar entidades distintas

- ⊙ Decidir donde está el **nivel de las instancias**

Pensar cual es nivel mínimo de granularidad que necesitamos en nuestro dominio

⊙ Limitar el ámbito de la ontología

- La ontología **no necesita incluir todas las clases** posibles del dominio, solo las necesarias para la aplicación que se desarrolla
- **Tampoco** necesitamos incluir **todos los atributos, restricciones, relaciones** posibles