

# Introducción

En el contexto de la iniciativa OpenCityData [?](<https://github.com/opencitydata/>) y del proyecto Ciudades Abiertas (<https://ciudadesabiertas.es/>) se han desarrollado un buen número de ontologías que permitan representar datos abiertos de ciudades, en muy diversas áreas (transporte, gestión económica, equipamientos, servicios, etc.). Dentro de esta iniciativa se encuentran muchas áreas todavía por desarrollar, y es una de ellas en las que se enmarca el desarrollo de nuestro trabajo.

El objetivo es diseñar e implementar una ontología que represente de manera adecuada instalaciones deportivas, sus características y acciones relacionadas. Las ontologías y otras fuentes de conocimiento utilizadas durante el desarrollo de este trabajo serán citadas, y se puede acceder a ellas a través de los hipervínculos de la bibliografía.

La ontología por desarrollar constituirá un modelo para representar los datos asociados a una instalación deportiva, donde se entiende como instalación deportiva un recinto o construcción provista de los medios necesarios para la práctica, aprendizaje y competición de uno o más deportes, y que está asociada, organizada y mantenida por una determinada organización. Como iremos describiendo en los siguientes apartados de la memoria, se especificarán todos los requisitos necesarios para la creación de la ontología, y se detallarán todas las características de esta.

A la hora de la creación de la ontología se han seguido las pautas que marca la metodología NeOn, haciendo uso tanto de recursos ontológicos como no ontológicos para su reutilización y reingeniería. Cada una de las actividades de la metodología han sido realizadas en una secuencia especificada y con un planteamiento ordenado, y tras ello se ha usado la herramienta Protégé para la formalización en OWL de la ontología y OOPS! para su evaluación.

# Metodología NeOn

La Metodología NeOn para la construcción de redes de ontologías es una metodología basada en escenarios que se apoya en los aspectos de colaboración de desarrollo de ontologías y la reutilización, así como en la evolución dinámica de las redes de ontologías en entornos distribuidos.

[?](<https://oeg.fi.upm.es/index.php/es/methodologies/59-neon-methodology/index.html#:~:text=La%20Metodolog%C3%ADa%20NeOn%20para%20la,de%20ontolog%C3%ADas%20en%20entornos%20distribuidos.>)

Como define esta metodología, hay un conjunto de nueve escenarios para la construcción de ontologías, haciendo hincapié en la reutilización de recursos ontológicos, la reingeniería y la fusión. Para este trabajo, en el cual se ha seguido la metodología NeOn se han seguido los siguientes escenarios:

- **Escenario 2: La reutilización y reingeniería de los recursos no ontológicos (NOR).** Se llevan a cabo procesos de reutilización NOR para decidir en base a los requisitos definidos de la ontología que NORs se utilizarán para construir la red de la ontología. Después de ello estos NORs se readaptarán mediante re-ingeniería.
- **Escenario 4: La reutilización y re-ingeniería de los recursos ontológicos.** Se llevan a cabo procesos de reutilización de recursos y se reorganizan los recursos ontológicos.
- **Escenario 7: Reutilización de los patrones de diseño de ontologías (ODPs).** Los desarrolladores de ontologías acceden a repositorios de reutilización ODPs.
- **Escenario 9: Localización de recursos ontológicos.** Los desarrolladores de ontologías adaptan una ontología a otras lenguas y la cultura las comunidades, obteniendo así una ontología multilingüe

Otro punto importante dentro de la metodología NeOn es el glosario de actividades [? <https://oeg.fi.upm.es/files/pdf/NeOnGlossary.pdf>]. Durante todo el proceso de construcción de nuestra ontología se han realizado actividades incluidas en este glosario. A continuación se listan algunas de las más importantes:

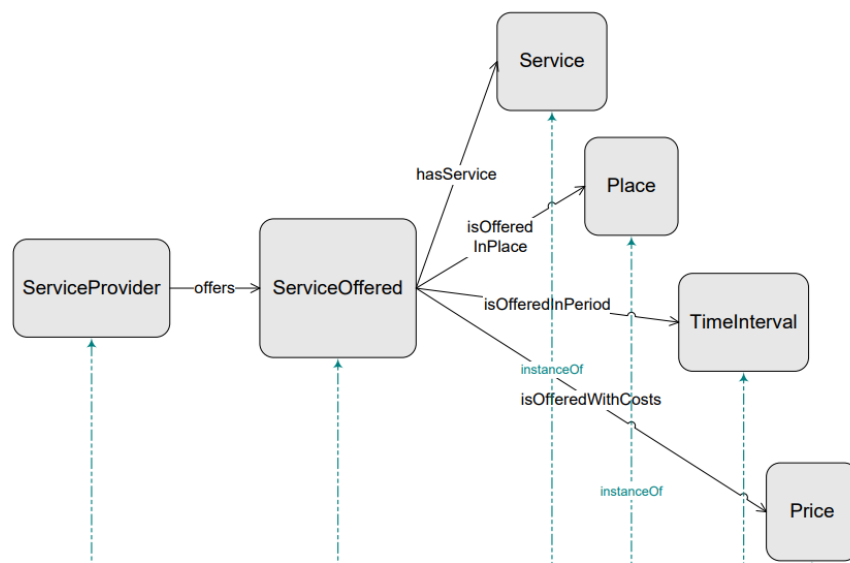
- **Ontology Requirements Specification:** Planteamiento de preguntas y la resolución de estas para detectar las necesidades y requisitos de la ontología.
- **Ontology Aligning:** Buscamos entre los diferentes recursos ontológicos los elementos comunes para evaluar cual es más adecuada para nuestra ontología.
- **Ontology Annotation:** A los elementos de nuestra ontología les hemos asignado etiquetas (labels) y comentarios para enriquecerla.
- **Ontology Comparison:** A la hora de buscar ontologías ya creadas sobre el mismo dominio tuvimos que comparar y evaluar las diferencias entre ellas para elegir las mas adecuadas.
- **Ontology Conceptualization:** Durante el proceso de adquisición tuvimos que organizar toda la información recopilada para crear un modelo conceptual que cubriese los requisitos definidos.
- **Control:** Se siguió la planificación temporal prevista para completar las tareas de desarrollo programadas.

- **Ontology Design Pattern Reuse:** Se han reusado patrones de diseño para nuestra ontología, concretamente el patrón de Normas y el de Servicios.
- **Ontology Diagnosis:** Gracias al uso de herramientas como OOPS! pudimos evaluar los fallos de la ontología y corregirlos durante el proceso de desarrollo.
- **Ontology Elicitation:** Se realizó una adquisición de estructuras conceptuales como el T-Box de documentos de expertos del dominio de las instalaciones deportivas.
- **Ontology Enrichment:** Hemos extendido con más relaciones y clases algunas de las ontologías que hemos reusado para completarlas para cubrir ciertos requisitos.
- **Ontology Implementation:** Una vez diseñado el modelo conceptual se transformó la ontología a un modelo formal (código OWL) gracias a Protégé.
- **Ontology Localization:** Las entidades y relaciones creadas se etiquetaron tanto en español como inglés.
- **Ontology Merging:** Se fusionaron clases y funciones de distintas ontologías para crear una nueva.
- **Ontology Search:** Se realizó una búsqueda de ontologías o módulos de ontologías candidatas para ser reutilizadas.
- **Ontology Module Extraction:** Se han extraído módulos concretos de ciertas ontologías.
- **Ontology Module Reuse:** Se han reutilizado los módulos extraídos de otras ontologías.
- **Non-Ontological Resource Reengineering:** Se han reestructurado contenidos no ontológicos para adaptarlos a nuestra ontología.
- **Non-Ontological Resource Reuse:** Se han usado recursos no ontológicos para transformarlos en partes de ontologías.
- **Scheduling:** Se ha realizado una planificación de las actividades anteriores para completarlas en el orden correcto y así llevar un control de las mismas.

# Patrones en la ontología

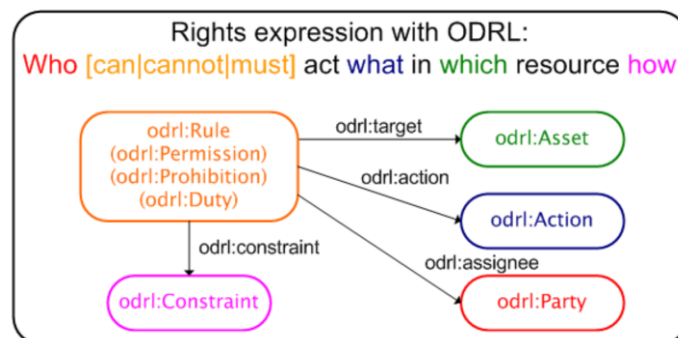
A la hora de diseñar nuestra ontología, de entre los patrones de diseño ya existentes vimos que algunos de ellos eran aplicables a nuestro caso de instalaciones deportivas. Concretamente, los patrones que hemos implementado han sido el de “Servicios ofrecidos” y el de “Normas”.

Para el diseño de los servicios ofrecidos por la organización en una instalación deportiva, se reutilizó el patrón de una relación N-aria según lo visto en clase. Este patrón se utiliza para representar una relación N-aria en el que todos los elementos tienen la misma importancia. Para representar esta relación N-aria se crea una clase, en nuestra ontología la clase *ServicioOfrecido*, a la que asociar todos los atributos de la relación. Este patrón se utilizará para representar los servicios que una organización ofrece que se realizarán en una instalación deportiva en un horario determinado y por un precio indicado.



El patrón de normas está destinado a la definición de una serie de reglas y normas que aplican a la organización. El patrón de reglas se expresa mediante ODRL (Open Digital Rights Language), y se definen mediante la estructura:

“Una **acción** es **permitida/obligatoria/prohibida** de ser realizada por una **parte** sobre el **activo**, siempre que se mantengan las **restricciones**”



En la sección de modelado conceptual de la ontología se entrará más en detalle sobre las clases y relaciones de estos patrones.

## Modelo conceptual

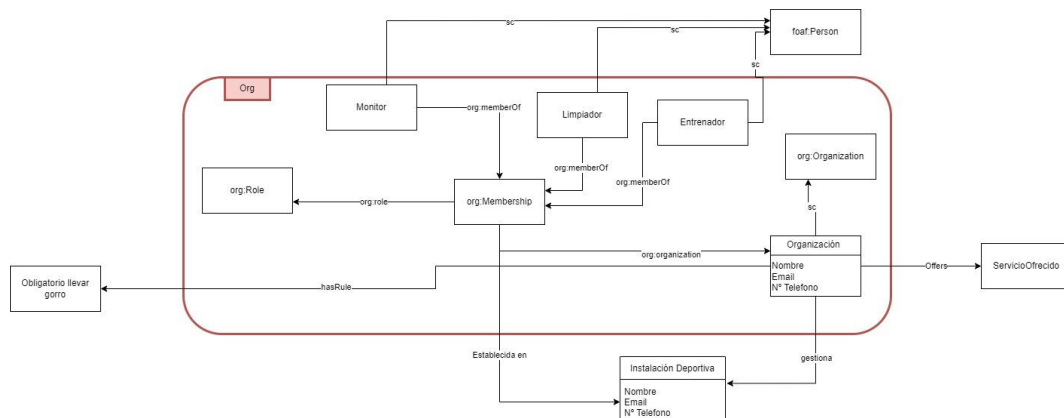
Tras todo el proceso de búsqueda de recursos ontológicos y no ontológicos para su reutilización como se ha explicado en los apartados anteriores así como la selección de patrones de diseño de ontologías, se desarrolló el modelo conceptual de la ontología. Esta es una de las fases más importantes en la construcción de ontologías, en este apartado se hará una descripción de las clases y relaciones que se han implementado y que en los siguientes apartados se formalizarán a código OWL. Debido a la amplitud del modelo y para un mejor entendimiento se procederá a explicar los distintos módulos del modelo por separado.

Primeramente, encontramos cuatro clases principales que serán el núcleo de nuestra ontología. Estas clases son **Organización** e **Instalación deportiva**, **ServicioOfrecido** y **Regla**.

### Organización

En lo que a la organización respecta, reutilizando clases y relaciones de la ontología *org* hemos creado el modelo que se puede observar en la Figura X1. La forma de interpretar esta parte del modelo es: “Una organización encargada de gestionar instalaciones deportivas establece acuerdos de membresía con sus trabajadores para que estos tengan un determinado rol dentro de la organización.” Analizando el modelo con sus clases y relaciones tenemos:

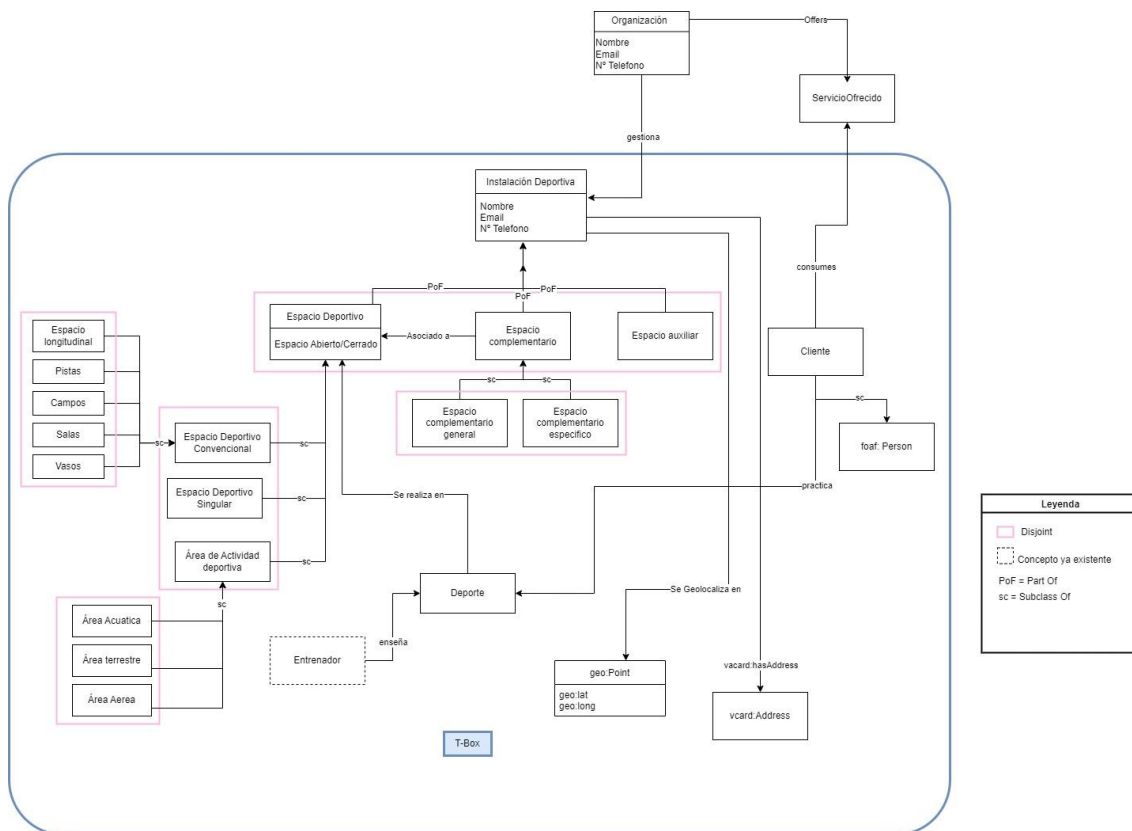
- Una organización es *SubClassOf* de **org:Organization**. Organización tiene como propiedades el Nombre, Email y Número de teléfono, que son literales.
- Una organización es la que gestiona una o más instalaciones deportivas, por lo que la clase Instalación Deportiva se relaciona con Organización mediante la relación *managedBy*.
- Una organización establece unos acuerdos de membresía con sus trabajadores para un determinado rol (tal y como está definido en la ontología *org*) por lo que tenemos la clase **org:Membership** que se relaciona con organización mediante la relación *org:organization*, se relaciona con las clases de los trabajadores mediante la relación *org:memberOf* y se relaciona con **org:Role** mediante la relación *org:role*. Al diseñar esta sección hemos querido contemplar la posibilidad de que un trabajador de la organización tenga asignada la instalación o las instalaciones en las que trabaja por lo que hemos unido **org:Membership** con **Instalación Deportiva** mediante la relación *establecidaEn*.
- Para determinar los tipos de trabajadores que hay en una organización hemos decidido crear una clase para cada uno de los tipos. Debido a la gran cantidad de clases de trabajadores que existen para su simplificación en nuestro modelo hemos ejemplificado las clases **Monitor**, **Limpiador**, y **Entrenador**, todas ellas *SubClassOf* de **foaf:Person**.



## Instalación Deportiva

Para la construcción de la taxonomía de las instalaciones deportivas se ha hecho uso del recurso no ontológico explicado en apartados anteriores, y mediante un T-Box se ha modelado a nuestra ontología.

- Una **Instalación Deportiva** puede estar compuesta por diferentes partes (*partOf*): **Espacio Deportivo** (que puede ser al aire libre o cerrado), **Espacio Complementario**, **Espacio Auxiliar**. Estas partes son disjuntas unas de otras. Un **Espacio Complementario** va *asociado a* un **Espacio Deportivo**
- Cada una de estas partes tiene su propia taxonomía tal y como se define en el recurso no ontológico y se muestra las diferentes subclases y las clases que son disjuntas (recuadros rosas) en el esquema de la Figura X2.
- La clase Instalación Deportiva cuenta con propiedades como N° de teléfono e Email que serán literales y no clases.
- Una instalación deportiva se geolocaliza en un **geo:point** mediante la relación *hasGeopoint*. A su vez expresamos su localización mediante una dirección, para ello relacionamos **Instalación Deportiva** con la clase **vcard:Address** mediante la relación *vcard:hasAddress*.
- La clase **Deporte** se relaciona con un **Espacio Deportivo** mediante la relación *esPracticadoEn*.
- La clase **Cliente** es *subClassOf* **foaf:Person**, y **Cliente** *practica* uno o varios **Deporte**. En el siguiente modulo también se verá que **Cliente** *consume* **ServicioOfrecido**
- **Entrenador** y **Monitor** (definidos en el apartado de organización) tienen una relación con **Deporte** que es *enseña*.



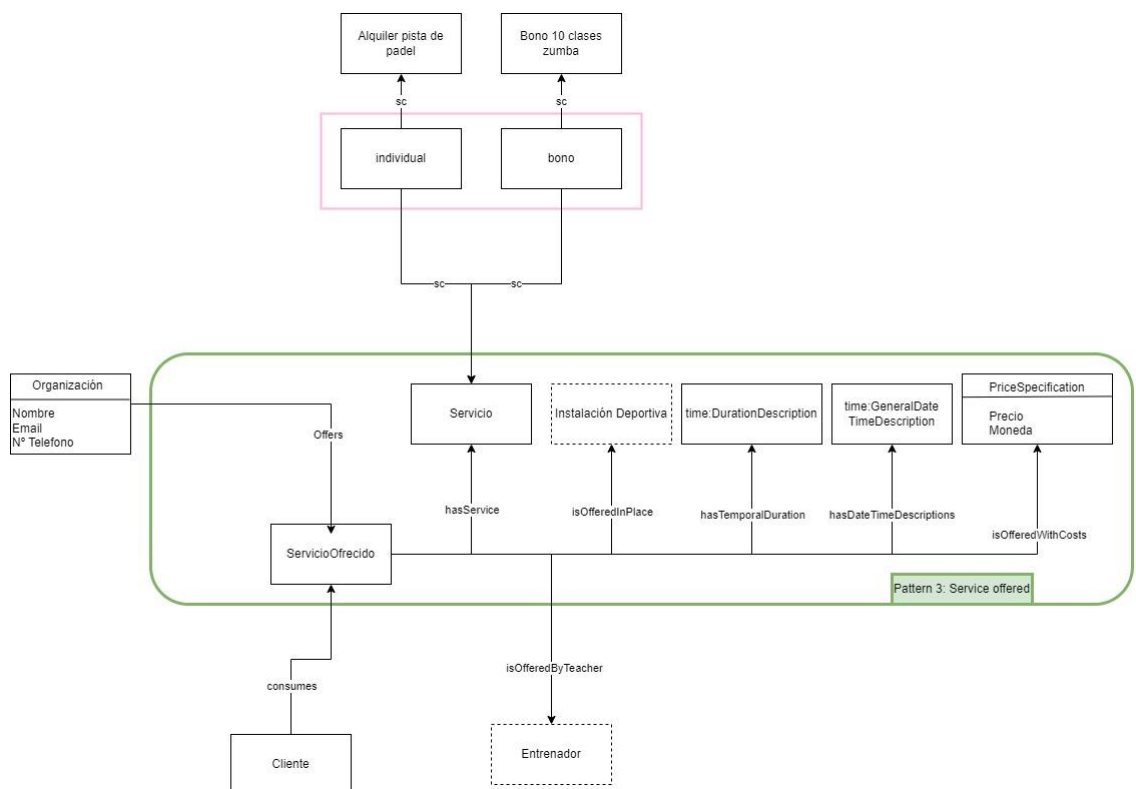
## Servicios ofrecidos

Esta clase **ServicioOfrecido** tendrá asociadas varias clases mediante diferente relaciones:

- **Servicio**, que tiene dos subclases disjuntas **Individual** y **Bono de servicios**. De estas dos clases habría más subclases, para ejemplificar hemos incluido el alquiler de una pista de pádel o un bono de 10 clases de zumba
- **Instalación Deportiva** en la que se ofrece dicho servicio. Por ejemplo, una pista de pádel, o la sala de musculación.
- **time:DurationDescription** y **time:GeneralDateTimeDescription**, que indicaran la duración y/o la fecha en la que se ofrece el servicio. Por ejemplo, jueves 3 de noviembre del 2022 de 18:00 a 19:00, o Todos los lunes y miércoles de febrero de 15:00 a 16:00.
- **PriceSpecification**, teniendo asociada como propiedades una cantidad y una moneda de pago. Por ejemplo 59,99 euros.

Estos serían los componentes básicos del patrón, pero adicionalmente para adaptarlo a nuestro caso de trabajo de una manera más precisa, hemos decidido añadir dos relaciones con dos clases más a ServicioOfrecido que son:

- **Trabajador**. El servicio puede ser llevado a cabo por un trabajador de la organización, por ejemplo un entrenador de tenis que ofrece clases de tenis.
- **Cliente**. Será el que consume el servicio ofrecido, por ejemplo un niño que se apunta a clases de natación.



## Normas

Al incluir este patrón de normas, hemos creado una única regla para ejemplificar las normas de una organización (en la realidad habría muchas de ellas, pero para simplificar el modelo para este trabajo se ha trabajado solo con una). Las reglas constan de la clase Rule que define si se trata de una prohibición, una obligación o permiso, y de ella salen varias relaciones a una serie de clases que serán *subclass of* de las siguientes entidades de la ontología ODRL:

- **odrl:Constraint.** Para cada norma hay una subclass de odrl:Constraint y puede ser un lugar físico (piscina, pista de tenis, gimnasio) o un marco temporal (los sábados, el día 1 de cada mes, todos los días de 9:00 a 10:00) en el que se aplica la norma.
- **odrl:Asset.** Para cada norma tendremos una clase subclass of odrl:Asset que establece el activo o recurso que es sujeto de la norma. Este activo puede ser cualquier forma de recurso identificable, como datos/información, servicios o elementos físicos. En nuestra ontología hemos puesto el ejemplo de gorro de baño.
- **odrl:Action.** Para cada norma tendremos una clase subclass of odrl:Action que indicará la acción realizada sobre el activo. En nuestro modelo encontramos la clase Llevar puesto para el caso de gorro de baño.
- **odrl:Party** Tendremos una subclass de esta clase de ODRL para cada regla, y definirá a que parte de las personas se le aplica dicha regla. Puede ser por ejemplo una regla que aplique a todos los usuarios, o una regla que aplique solo a trabajadores.

