

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

BBDD RDF y SHACL

Álvaro Palacios

RIAM Intelearning Lab – GNOSS

alvaropalacios@gnooss.com



HĒRCULES



Introducción

- ☐ BBDD RDF
- ☐ Restricciones de la ontología
- ☐ SHACL
- ☐ SHACL en ASIO

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. BBDD RDF



BBDD RDF

Una base de datos RDF, también llamada triplestore o almacén de RDF es una base de datos diseñada para el almacenamiento y recuperación de triples a través de consultas semánticas.

El RDF Store es el componente de la arquitectura que **almacena el grafo de conocimiento de ASIO en formato RDF**. Dentro de ASIO actúa como un servidor de datos que responde a consultas SPARQL, por lo que también puede ser nombrado como **Servidor RDF o Servidor SPARQL**.

El RDF Store proporciona el servicio de datos y consultas al Linked Data Server y al SPARQL Endpoint.

BBDD RDF

ASIO proporciona dos interfaces de reutilización de los datos del grafo de conocimiento modelados por la Red de Ontologías Hércules ROH:

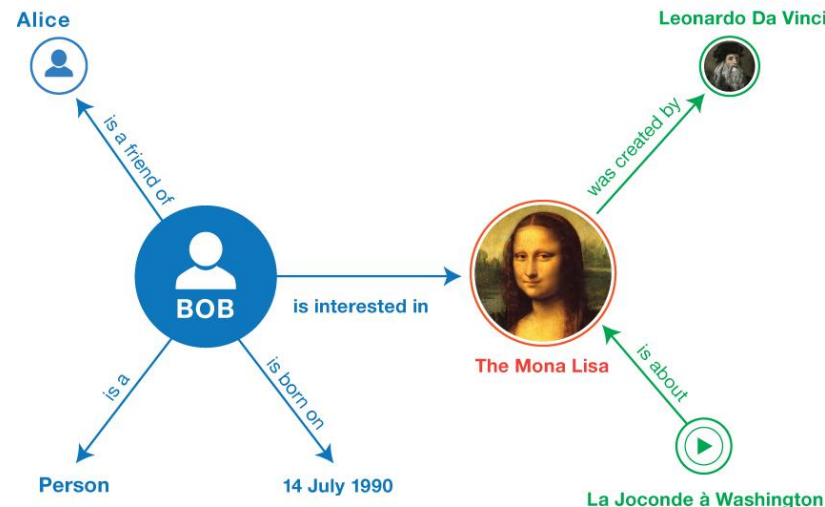
- **Linked Data Server**. Es el componente que permite la publicación de los datos cargados en el RDF Store como datos abiertos y enlazados (linked open data).
- **SPARQL Endpoint**. Permite a usuarios y administradores consultar los datos del grafo de conocimiento almacenados en el RDF Store, mediante el lenguaje y protocolo de consultas SPARQL.

BBDD RDF. RDF

❑ RDF identifica conceptos usando identificadores Web (URIs), y describe recursos con propiedades y valores de las mismas

❑ Definiciones:

- Un **Sujeto** es cualquier cosa que puede tener una URI, como por ejemplo "http://www.w3schools.com/RDF"
- Un **Predicado** es una propiedad que tiene un sujeto, como "autor" o "páginaweb", identificado por URI
- Un **Objeto** de propiedad es el valor de una Propiedad, tal como "Diego Ipiña" (literal) o "http://www.w3schools.com" (un valor de propiedad puede corresponder a un recurso o URI)



BBDD RDF. Terminología

- ☐ Grafo RDF: Un grafo RDF es un conjunto de triples RDF
- ☐ Triples RDF: Un triple consiste en tres componentes: sujeto, predicado y objeto.
- ☐ Sujeto: Puede ser un IRI o un Blank node.
- ☐ Predicado: Tiene que ser un IRI
- ☐ Objeto: Puede ser un IRI, un literal o un Blank node.
- ☐ IRI: Internationalized Resource Identifier
- ☐ Literal: Un literal consta de dos o tres elementos, la forma léxica, el IRI del tipo de datos y si el IRI es <http://www.w3.org/1999/02/22-rdf-syntax-ns#langString> una etiqueta de idioma
- ☐ Nodo en blanco: También conocidos como blank nodes son **identificadores** locales que se utilizan en algunas sintaxis RDF concretas o implementaciones de RDF Stores. Siempre tienen un alcance local para el archivo o el almacén RDF y *no* son identificadores persistentes o portables
- ☐ Node: El conjunto de **nodos** de un grafo RDF es el conjunto de sujetos y objetos de triples en el grafo.

BBDD RDF. Formatos de serialización RDF

Existen diferentes formatos a la hora de realizar serializaciones de datos RDF

❑ Formato RDF/XML:

Serialización de XML de RDF, definida en **RDF 1.1 XML Syntax**. Tipo de medios: **application/rdf+xml**. Extensión de archivo típica: **.rdf**.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos/"
  xmlns:edu="http://www.example.org/">
  <rdf:Description rdf:about="http://www.deusto.es">
    <geo:lat>43.270737</geo:lat>
    <geo:long>-2.939637</geo:long>
    <edu:hasFaculty>
      <rdf:Bag>
        <rdf:li rdf:resource="http://www.eside.deusto.es" dc:title="Facultad de Ingeniería"/>
        <rdf:li rdf:resource="http://www.lacomercial.deusto.es" dc:title="Facultad de Empresariales"/>
      </rdf:Bag>
    </edu:hasFaculty>
  </rdf:Description>
</rdf:RDF>
```


BBDD RDF. Formatos de serialización RDF

Existen diferentes formatos a la hora de realizar serializaciones de datos RDF

❑ Formato RDF/XML:

Serialización de XML de RDF, definida en **RDF 1.1 XML Syntax**. Tipo de medios: **application/rdf+xml**. Extensión de archivo típica: **.rdf**.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos/"
  xmlns:edu="http://www.example.org/" >
  <rdf:Description rdf:about="http://www.deusto.es/">
    <edu:hasFaculty rdf:nodeID="N101a0be3f2db4c55a5d8d270c5551f15" />
    <geo:lat>43.270737</geo:lat>
    <geo:long>-2.939637</geo:long>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.eside.deusto.es/">
    <dc:title>Facultad de Ingeniería</dc:title>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.lacomercial.deusto.es/">
    <dc:title>Facultad de Empresariales</dc:title>
  </rdf:Description>
  <rdf:Bag rdf:nodeID="N101a0be3f2db4c55a5d8d270c5551f15">
    <rdf:_1 rdf:resource="http://www.eside.deusto.es/" />
    <rdf:_2 rdf:resource="http://www.lacomercial.deusto.es/" />
  </rdf:Bag>
</rdf:RDF>
```

BBDD RDF. Formatos de serialización RDF

❑ Turtle:

Una sintaxis textual para RDF definida en RDF 1.1 Turtle que permite que un gráfico RDF se escriba por completo en un formato de texto compacto y natural, con abreviaturas para los patrones de uso y tipos de datos comunes. Turtle proporciona niveles de compatibilidad con el formato N-Triples además de con la sintaxis del patrón triple de SPARQL. Tipo de medios: Extensión de archivo típica de text/turtle: .ttl: .

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
```

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
```

```
@prefix xml: <http://www.w3.org/XML/1998/namespace>.
```

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
```

```
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos/>.
```

```
@prefix edu: <http://www.example.org/>.
```

```
<http://www.deusto.es/> edu:hasFaculty [a rdf:Bag ;  
                                     rdf:_1 <http://www.eside.deusto.es/> ;  
                                     rdf:_2 <http://www.lacomercial.deusto.es/>];  
      geo:lat "43.270737";  
      geo:long "-2.939637".
```

```
<http://www.eside.deusto.es/> dc:title "Facultad de Ingeniería".
```

```
<http://www.lacomercial.deusto.es/> dc:title "Facultad de Empresariales".
```

BBDD RDF. Formatos de serialización RDF

❑ N-triples:

Un formato basado en líneas no cifrado para codificar un gráfico de RDF, definido en RDF 1.1 N-Triples. Tipo de medios: application/n-triples: text/turtle, text/plain o .
Extensión de archivo típica: .nt: .

```
<http://www.deusto.es/> <http://www.w3.org/2003/01/geo/wgs84_pos/lat> "43.270737" .  
<http://www.deusto.es/> <http://www.w3.org/2003/01/geo/wgs84_pos/long> "-2.939637" .  
<http://www.deusto.es/> <http://www.example.org/hasFaculty> _:N101a0be3f2db4c55a5d8d270c5551f15 .  
_:N101a0be3f2db4c55a5d8d270c5551f15 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag> .  
_:N101a0be3f2db4c55a5d8d270c5551f15 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> <http://www.eside.deusto.es/> .  
_:N101a0be3f2db4c55a5d8d270c5551f15 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_2> <http://www.lacomercial.deusto.es/> .  
<http://www.eside.deusto.es/> <http://purl.org/dc/elements/1.1/title> "Facultad de Ingenier\u00EDA" .  
<http://www.lacomercial.deusto.es/> <http://purl.org/dc/elements/1.1/title> "Facultad de Empresariales" .
```

Hércules ASIO. Restricciones de la ontología



Restricciones de la ontología

La ontología utilizada en ASIO se encuentra disponible aquí:

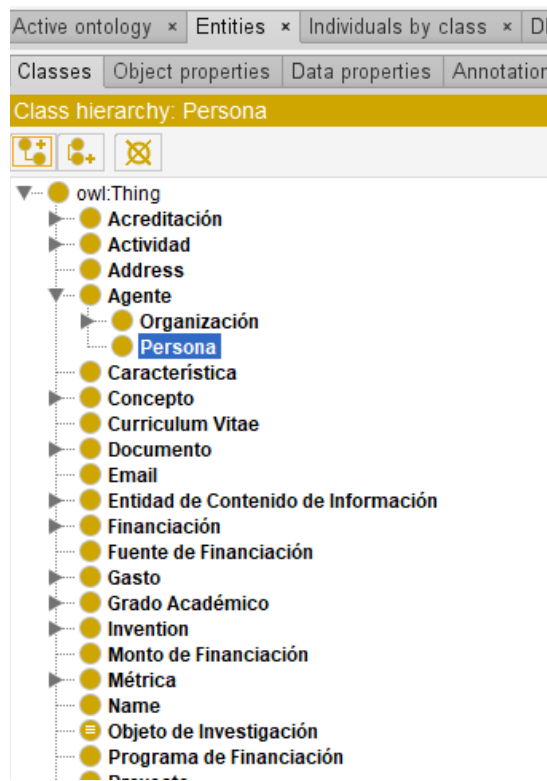
<https://github.com/HerculesCRUE/GnossDeustoOnto/blob/master/roh-v2.owl>

Puede verse usando Protégé, disponible aquí: <https://protege.stanford.edu>

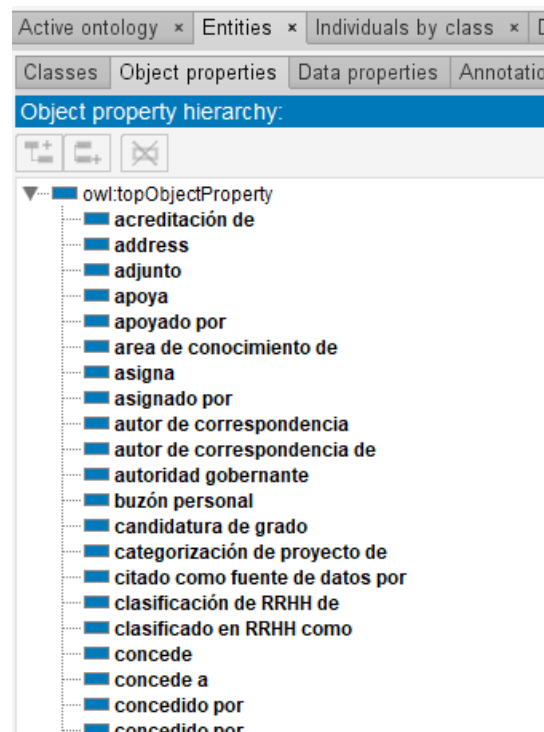
Restricciones de la ontología

Esta ontología cuenta con una serie de clases y propiedades

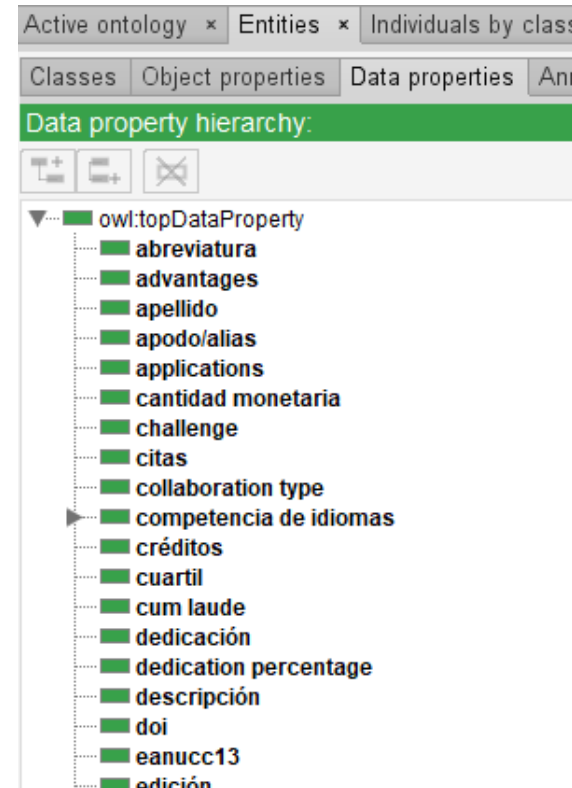
Classes:



Object properties:



Data properties:



Restricciones de la ontología

Dentro de la ontología puede haber una serie de restricciones que indican como deberían ser los datos para ajustarse a la ontología.

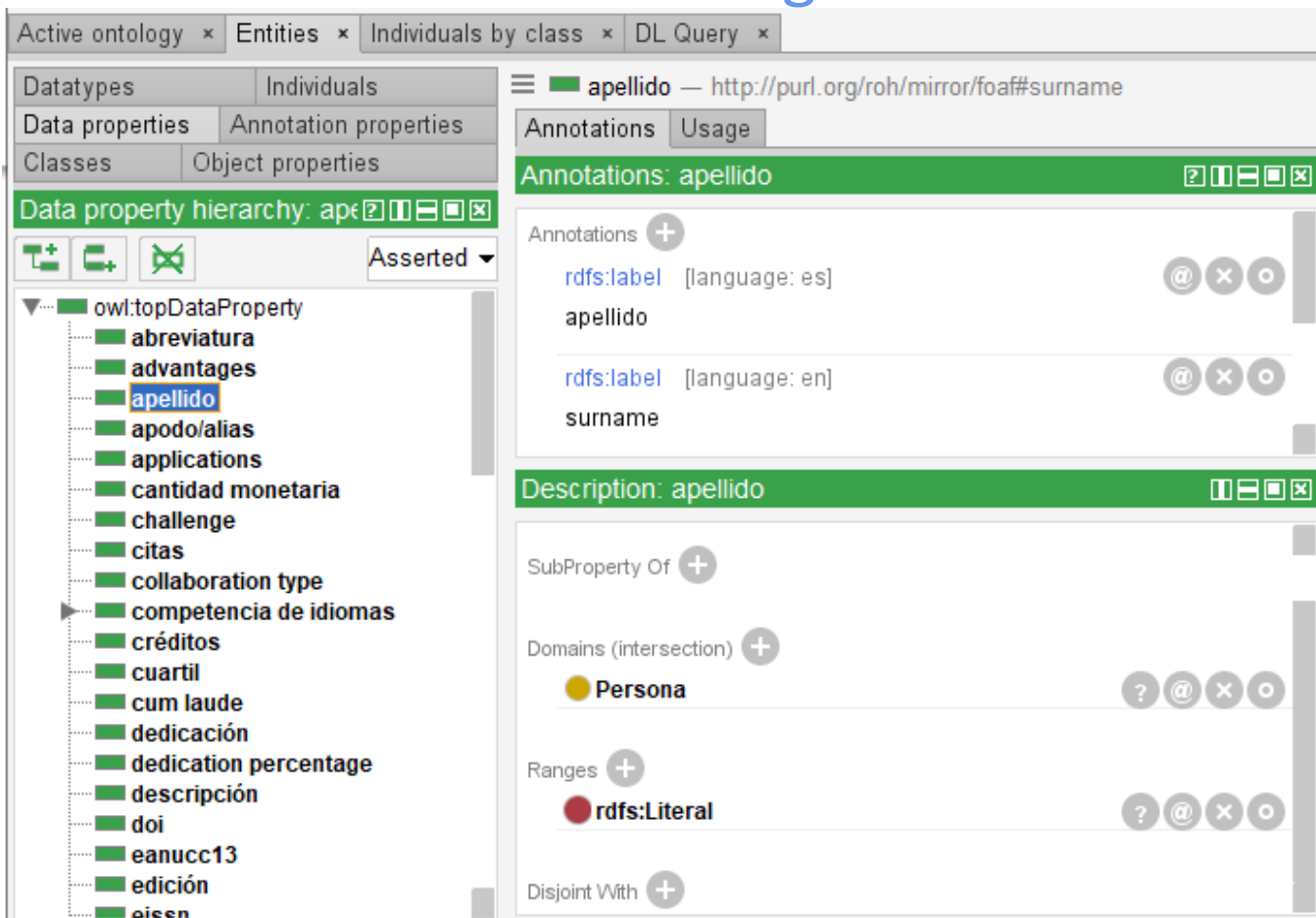
Estas restricciones pueden estar definidas tanto en las propiedades como en las clases.

Restricciones de la ontología. Restricciones en las propiedades

Restricciones en las propiedades:

1. Restricciones de dominio: Especifica la clase que tiene que figurar como sujeto cuando se usa la propiedad.
2. Restricciones de rango: En las object properties, especifica la clase que tiene que figurar como objeto cuando se usa la propiedad. En las data properties, especifica el tipo de dato que tiene que figurar como objeto.

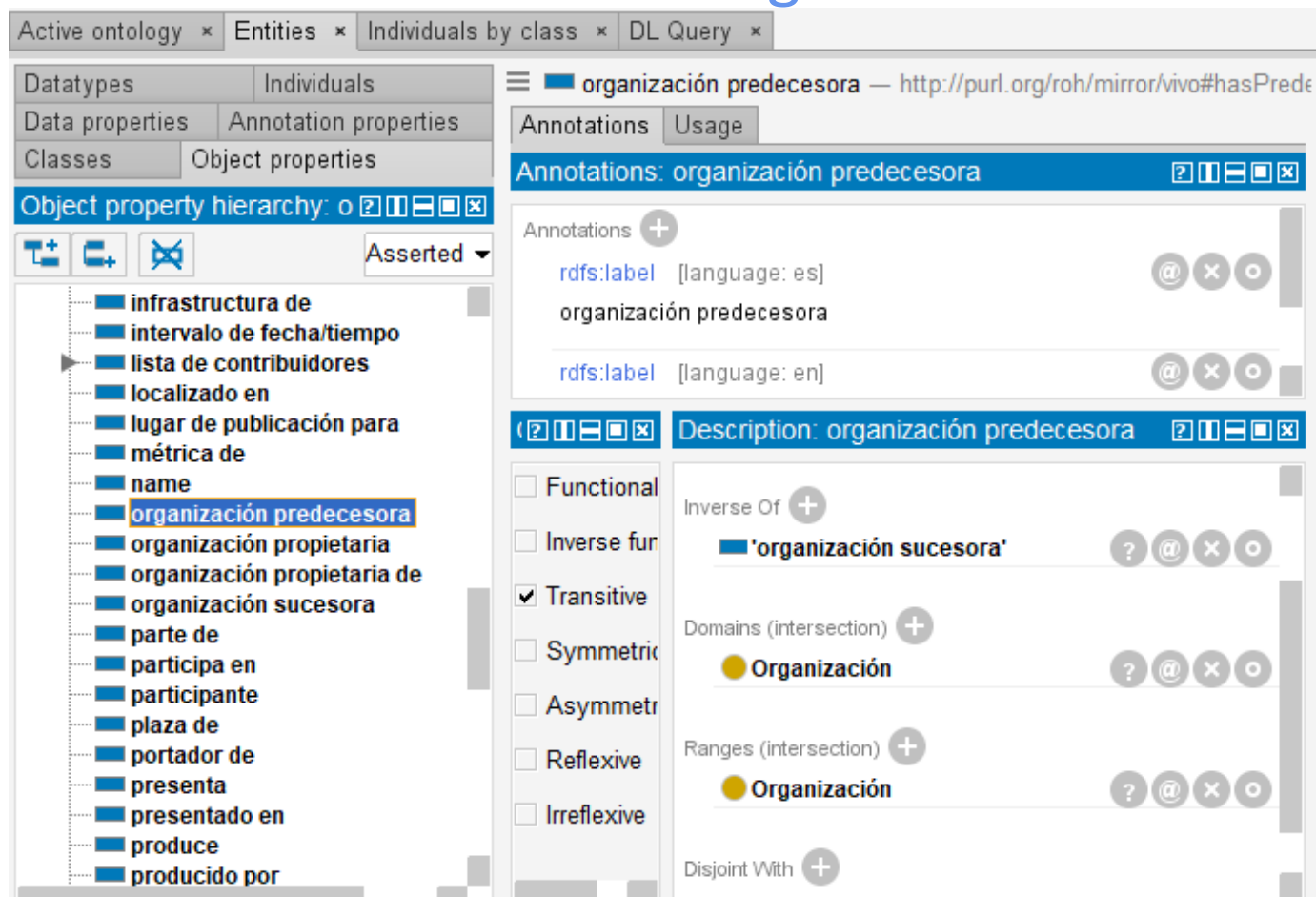
Restricciones de la ontología. Restricciones en las propiedades



The screenshot shows the Protégé ontology editor interface. The left sidebar displays a list of properties under the 'owl:topDataProperty' class, with 'apellido' highlighted. The main panel shows the configuration for the 'apellido' property, which is defined by the URI `http://purl.org/roh/mirror/foaf#surname`. The 'Annotations' tab is active, showing two labels: 'apellido' in Spanish (language: es) and 'surname' in English (language: en). The 'Description' tab is also visible, showing the property's domain and range. The domain is set to 'Persona' (indicated by a yellow circle) and the range is set to 'rdfs:Literal' (indicated by a red circle). The 'SubProperty Of' and 'Disjoint With' sections are currently empty.

En este caso, la propiedad ‘apellido’ tiene que tener como dominio entidades del tipo ‘Persona’ y como rango datos ‘rdfs:Literal’

Restricciones de la ontología. Restricciones en las propiedades



The screenshot shows a web-based ontology editor interface. On the left, a tree view lists various properties, with 'organización predecesora' selected. The main panel displays the configuration for this property. It includes a tab for 'Annotations' showing 'rdfs:label' in Spanish and English. Below that, the 'Description' tab is active, showing a list of restrictions: 'Inverse Of' (set to 'organización sucesora'), 'Domains (intersection)' (set to 'Organización'), and 'Ranges (intersection)' (set to 'Organización'). The 'Transitive' checkbox is checked, while others like 'Functional', 'Symmetric', and 'Reflexive' are unchecked.

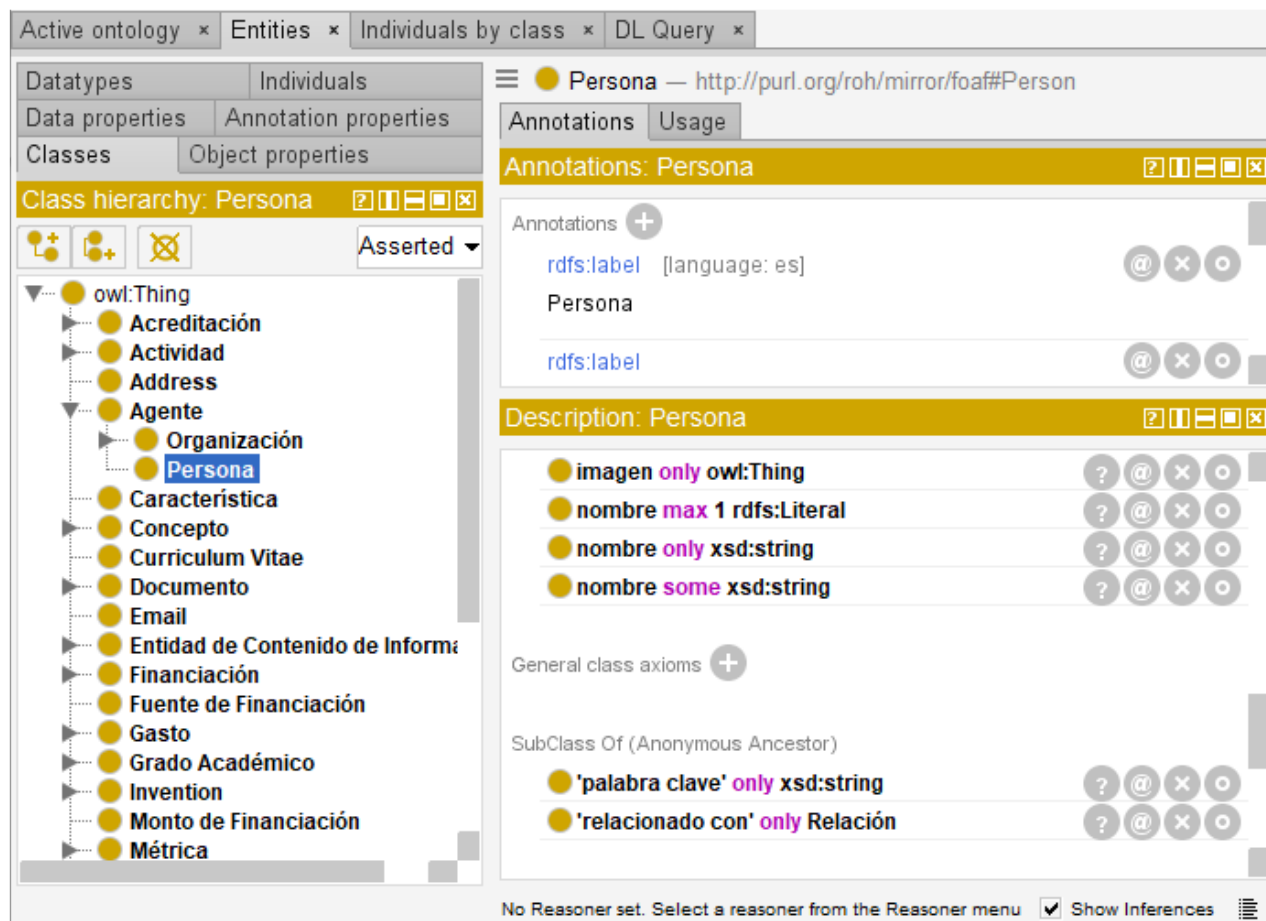
En este caso, la propiedad ‘organización predecesora’ tiene que tener como dominio y como rango entidades del tipo ‘Organización’

Restricciones de la ontología. Restricciones en las clases

En las clases se especifican las propiedades que pueden contener y para cada una de ellas se puede especificar su multiplicidad y obligatoriedad:

1. **only:** Especifica que en esta clase, la propiedad indicada sólo puede tener valores del tipo/clase definido.
2. **some:** Especifica que en esta clase, la propiedad indicada debe tener obligatoriamente algún valor del tipo/clase definido.
3. **min:** Especifica el número mínimo de valores que tiene que tener para la clase los datos del tipo/clase definido.
4. **max:** Especifica el número máximo de valores que puede tener para la clase los datos del tipo/clase definido.

Restricciones de la ontología. Restricciones en las clases



The screenshot shows the Protégé ontology editor interface. The left pane displays the 'Class hierarchy: Persona' with a tree structure including classes like 'owl:Thing', 'Acreditación', 'Actividad', 'Address', 'Agente', 'Organización', 'Persona', 'Característica', 'Concepto', 'Curriculum Vitae', 'Documento', 'Email', 'Entidad de Contenido de Informa', 'Financiación', 'Fuente de Financiación', 'Gasto', 'Grado Académico', 'Invention', 'Monto de Financiación', and 'Métrica'. The 'Persona' class is selected. The right pane shows the 'Annotations: Persona' and 'Description: Persona' sections. The 'Annotations' section lists 'rdfs:label' with the value 'Persona' and '[language: es]'. The 'Description' section lists several restrictions: 'imagen only owl:Thing', 'nombre max 1 rdfs:Literal', 'nombre only xsd:string', and 'nombre some xsd:string'. The 'General class axioms' section shows 'SubClass Of (Anonymous Ancestor)' with restrictions like 'palabra clave only xsd:string' and 'relacionado con only Relación'.

En este caso, entre otras restricciones, la clase 'Persona' tiene que tener obligatoriamente una propiedad nombre cuyo valor sea del tipo 'xsd:string'

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. SHACL



SHACL

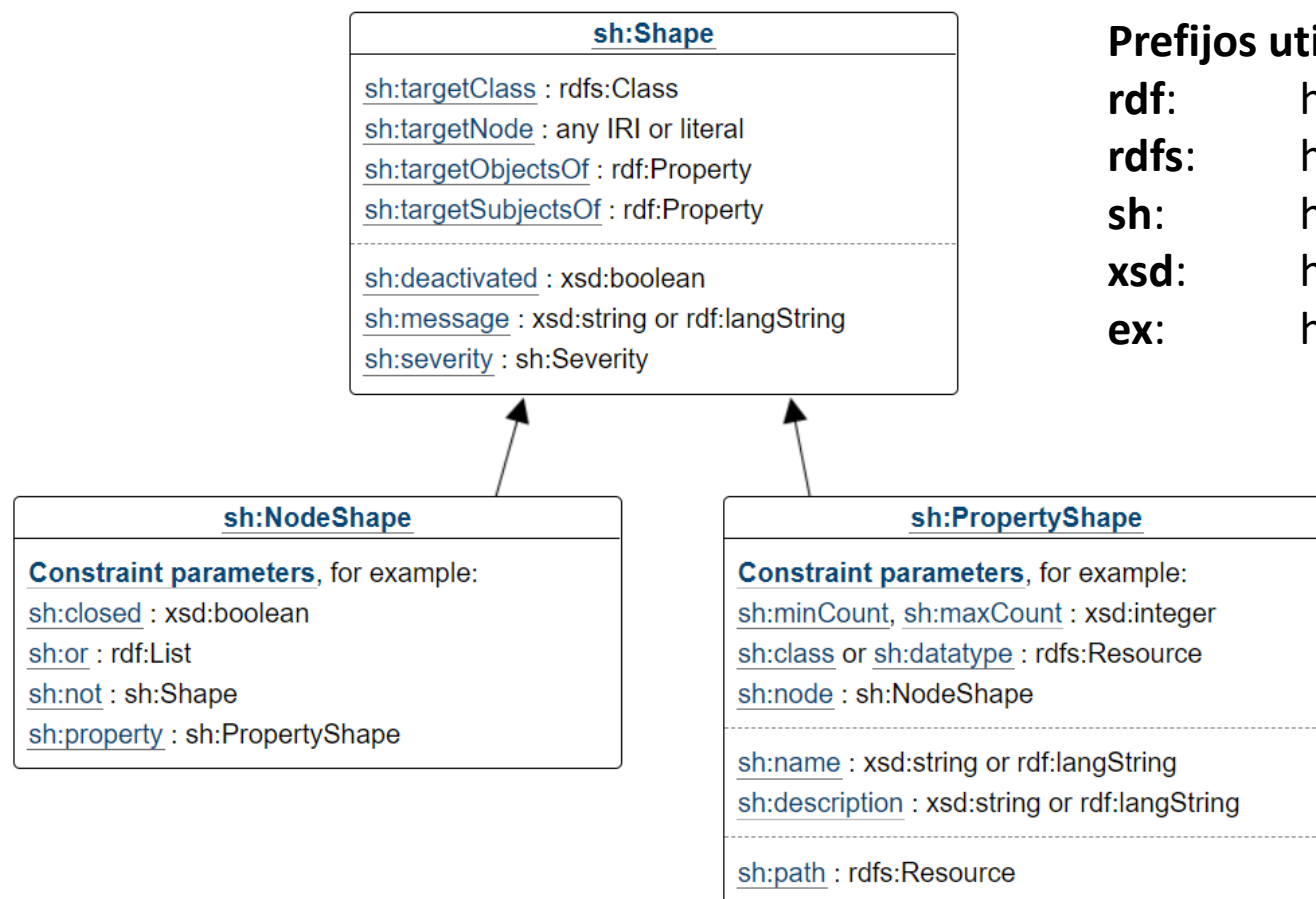
Shapes Constraint Language (SHACL) es una especificación del World Wide Web Consortium (W3C) para validar grafos de datos contra un conjunto de condiciones.

Estas condiciones se proporcionan como shapes en forma de un grafo RDF.

Los grafos RDF que se utilizan de esta manera se denominan “shapes graphs” en SHACL y los grafos RDF que se validan con un shape graph se denominan “data graphs”.

Documentación: <https://www.w3.org/TR/shacl/>

SHACL. Diagrama



Prefijos utilizados en esta documentación:

rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs: <http://www.w3.org/2000/01/rdf-schema#>
sh: <http://www.w3.org/ns/shacl#>
xsd: <http://www.w3.org/2001/XMLSchema#>
ex: <http://example.com/ns#>

SHACL. sh:Shape

sh:Shape es la superclase SHACL de esos dos tipos de shapes en el vocabulario SHACL. Sus subclases sh:NodeShape y sh:PropertyShape se pueden utilizar como tipo SHACL de formas de nodo y propiedad, respectivamente.

Los shapes pueden declarar restricciones usando los parámetros de restricciones.

SHACL. `sh:Shape`

Propiedades:

- [`sh:targetClass`](#) : `rdfs:Class`

En este caso el objetivo del Shape son las entidades de la clase especificada.

Shape graph:

```
ex:PersonShape  
  a sh:NodeShape ;  
  sh:targetClass ex:Person .
```

Data graph:

```
ex:Alice a ex:Person .  
ex:Bob a ex:Person .  
ex:NewYork a ex:Place .
```

SHACL. sh:Shape

Propiedades:

- [sh:targetNode](#) : any IRI or literal

En este caso el objetivo del Shape son las entidades especificadas.

Shape graph:

```
ex:PersonShape  
  a sh:NodeShape ;  
  sh:targetNode ex:Alice .
```

Data graph:

```
ex:Alice a ex:Person .  
ex:Bob a ex:Person .
```

SHACL. sh:Shape

Propiedades:

- [sh:targetObjectsOf](#) : rdf:Property

En este caso el objetivo del Shape son los objetos en los que el predicado sea la propiedad especificada.

Shape graph:

```
ex:TargetObjectsOfExampleShape  
  a sh:NodeShape ;  
  sh:targetObjectsOf ex:knows .
```

Data graph:

```
ex:Alice ex:knows ex:Bob .  
ex:Bob ex:livesIn ex:NewYork .
```

SHACL. sh:Shape

Propiedades:

- [sh:targetSubjectsOf](#) : rdf:Property

En este caso el objetivo del Shape son los sujetos en los que el predicado sea la propiedad especificada.

Shape graph:

```
ex:TargetSubjectsOfExampleShape  
  a sh:NodeShape ;  
  sh:targetSubjectsOf ex:knows .
```

Data graph:

```
ex:Alice ex:knows ex:Bob .  
ex:Bob ex:livesIn ex:NewYork .
```

SHACL. sh:Shape

Propiedades:

- [sh:deactivated](#) : xsd:boolean

En caso de que tenga como valor 'true' indica que el shape está desactivado.

Shape graph:

ex:PersonShape

a sh:NodeShape ;
sh:targetClass ex:Person ;
sh:property ex:PersonShape-name .

ex:PersonShape-name

a sh:PropertyShape ;
sh:path ex:name ;
sh:minCount 1 ;
sh:deactivated true .

Data graph:

ex:JohnDoe a ex:Person .

En este caso no se produciría una violación de la restricción porque la restricción estaría desactivada

SHACL. sh:Shape

Propiedades:

- [sh:message](#) : xsd:string or rdf:langString
Indica el mensaje que se producirá si se viola la restricción.

Shape graph:

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:property ex:PersonShape-name .
```

```
ex:PersonShape-name
  a sh:PropertyShape ;
  sh:path ex:name ;
  sh:minCount 1 ;
  sh:message "Nombre obligatorio".
```

Data graph:

```
ex:JohnDoe a ex:Person .
```

En este caso al producirse la violación se produciría este mensaje como resultado de la validación

SHACL. sh:Shape

Propiedades:

- [sh:severity](#) : sh:Severity

Indica el nivel de severidad de la violación de la restricción

Shape graph:

ex:PersonShape

a sh:NodeShape ;
sh:targetClass ex:Person ;
sh:property ex:PersonShape-name .

ex:PersonShape-name

a sh:PropertyShape ;
sh:path ex:name ;
sh:minCount 1 ;
sh:severity sh:Violation.

Data graph:

ex:JohnDoe a ex:Person .

Existen 3 niveles de severidad:

- sh:Info → Violación no crítica que muestra un mensaje
- sh:Warning → Violación no crítica que indica una advertencia
- sh:Violation → Violación crítica

El valor por defecto, si no se especifica, es sh:Violation

En este caso el nivel de severidad de la violación de la restricción sería sh:Violation

SHACL. sh:NodeShape

Un node shape es un shape que no es el sujeto de un triple con sh:path como predicado, es recomendado pero no requerido que un node shape esté declarado como una instancia de sh:NodeShape, las instancias de sh:NodeShape no pueden tener la propiedad sh:path.

De manera informal, los node shape especifican las restricciones que deben cumplirse con respecto a los nodos de enfoque . A diferencia de los sh:PropertyShape, se aplican principalmente al nodo de enfoque en sí, no a sus valores de propiedad.

SHACL. sh:NodeShape

Propiedades:

- [sh:property](#) : sh:PropertyShape

Especifica los sh:PropertyShape que debe cumplir un nodo

Shape graph:

ex:PersonShape

a sh:NodeShape ;
sh:targetClass ex:Person ;
sh:property ex:PersonShape-name .

ex:PersonShape-name

a sh:PropertyShape ;
sh:path ex:name ;
sh:minCount 1 .

Data graph:

ex:JohnDoe

a ex:Person ;
ex:name "Álvaro" ;
ex:surname "Palacios" .

En este caso no se produciría una violación porque el nodo 'ex:JohnDoe' tiene al menos un valor para la propiedad 'ex:name'

SHACL. sh:NodeShape

Propiedades:

- sh:closed : xsd:boolean

Especifica que los nodos los que hace referencia solo pueden tener valor para aquellas propiedades que tiene definidas explícitamente.

Shape graph:

ex:PersonShape

```
a sh:NodeShape ;  
sh:targetClass ex:Person ;  
sh:closed true ;  
sh:property ex:PersonShape-name .
```

ex:PersonShape-name

```
a sh:PropertyShape ;  
sh:path ex:name ;  
sh:minCount 1 .
```

Data graph:

ex:JohnDoe

```
a ex:Person ;  
ex:name "Álvaro" ;  
ex:surname "Palacios" .
```

En este caso se produciría una violación porque la persona sólo puede tener la propiedad 'ex:name'

SHACL. sh:NodeShape

Propiedades:

- [sh:or](#) : rdf:List

Especifica que debe cumplirse al menos una de las restricciones.

Shape graph:

ex:OrConstraintExampleShape

a sh:NodeShape ;

sh:targetNode ex:Bob ;

sh:or (

[

sh:path ex:firstName ;

sh:minCount 1 ;

]

[

sh:path ex:givenName ;

sh:minCount 1 ;

]

).

Data graph:

ex:JohnDoe

a ex:Person ;

ex:firstName "Álvaro" .

En este caso no se produciría una violación porque la persona tiene al menos la propiedad 'ex:firstName'

SHACL. sh:NodeShape

Propiedades:

- sh:not : sh:Shape

Especifica que un nodo no debe cumplir la restricción especificada

Shape graph:

```
ex:NotExampleShape
  a sh:NodeShape ;
  sh:targetNode ex:InvalidInstance1 ;
  sh:not [
    a sh:PropertyShape ;
    sh:path ex:property ;
    sh:minCount 1 ;
  ] .
```

Data graph:

```
ex:InvalidInstance1 ex:property "Some value" .
```

En este caso se produciría una violación porque el nodo 'ex:InvalidInstance1' no puede tener la propiedad 'ex:property'

SHACL. `sh:PropertyShape`

Un property shape es un shape que es sujeto de un triple que tiene `sh:path` como predicado.

Un property shape tiene como máximo un valor para `sh:path`.

Se recomienda, pero no es obligatorio, que un property shape se declare como una instancia SHACL de `sh:PropertyShape`.

SHACL. `sh:PropertyShape`

Propiedades:

- [`sh:path`](#) : `rdfs:Resource`
Especifica el predicado del data graph

Shape graph:

```
ex:ExamplePropertyShape
  a sh:PropertyShape ;
  sh:targetNode ex:Bob ;
  sh:path ex:email ;
  sh:minCount 1 .
```

Data graph:

```
ex:Bob a ex:Person ;
       ex:name "Bob".
```

En este caso se produciría una violación

SHACL. sh:PropertyShape

Propiedades:

- [sh:minCount](#), [sh:maxCount](#) : xsd:integer
Especifica el número mínimo o máximo de valores para la propiedad que deben cumplirse

Shape graph:

ex:MinCountExampleShape

```
a sh:PropertyShape ;  
sh:targetNode ex:Alice, ex:Bob ;  
sh:path ex:name ;  
sh:minCount 1 .  
sh:maxCount 2 .
```

Data graph:

```
ex:Alice ex:name "Alice" .  
ex:Bob ex:givenName "Bob"@en .  
ex:Alvaro ex:name "Álvaro 1"@en .  
ex:Alvaro ex:name "Álvaro 2"@en .  
ex:Alvaro ex:name "Álvaro 3"@en .
```

En este caso se produciría una violación para la entidad 'ex:Bob' porque no tiene ningún valor para la propiedad 'ex:name' y también para la entidad 'ex:Alvaro' porque tiene más de 2 propiedades ex:name

SHACL. `sh:PropertyShape`

Propiedades:

- [`sh:class`](#) or [`sh:datatype`](#) : `rdfs:Resource`
Especifica que el valor de la propiedad tiene que ser de la clase o el tipo especificado

Shape graph:

```
ex:ClassExampleShape
  a sh:NodeShape ;
  sh:targetNode ex:Bob, ex:Alice, ex:Carol ;
  sh:property [
    sh:path ex:address ;
    sh:class ex:PostalAddress ;
  ] ;
  sh:property [
    sh:path ex:age ;
    sh:datatype xsd:integer ;
  ] .
```

Data graph:

```
ex:Alice a ex:Person .
ex:Bob ex:address [ a ex:PostalAddress ; ex:city ex:Berlin ] .
ex:Carol ex:address [ ex:city ex:Cairo ] .
ex:Alice ex:age "23"^^xsd:integer .
ex:Bob ex:age "twenty two" .
```

En este caso se produciría una violación para la entidad 'ex:Carol' porque su propiedad 'ex:address' no es de la clase 'ex:PostalAddress' y también para la entidad 'ex:Bob' porque su propiedad 'ex:age' no es del 'tipo xsd:integer'

SHACL. sh:PropertyShape

Propiedades:

- [sh:node](#) : sh:NodeShape

Especifica la condición de que cada valor del nodo se ajuste al sh:NodeShape

Shape graph:

ex:AddressShape

```
a sh:NodeShape ;  
sh:property [  
    sh:path ex:postalCode ;  
    sh:datatype xsd:string ;  
    sh:maxCount 1 ;  
] .
```

ex:PersonShape

```
a sh:NodeShape ;  
sh:targetClass ex:Person ;  
sh:property [ # _:b1  
    sh:path ex:address ;  
    sh:minCount 1 ;  
    sh:node ex:AddressShape ;  
] .
```

Data graph:

```
ex:Bob a ex:Person ;  
        ex:address ex:BobsAddress .  
ex:BobsAddress  
        ex:postalCode "1234" .  
ex:Reto a ex:Person ;  
        ex:address ex:RetosAddress .  
ex:RetosAddress  
        ex:postalCode 5678 .
```

En este caso se produciría una violación para el nodo ex:Reto porque el nodo 'ex:RetoAdress' de su propiedad 'ex:address' incumple el NodeShape porque 'ex:postalCode' debería ser del tipo xsd:string

SHACL. sh:PropertyShape

Propiedades:

- [sh:name](#) : xsd:string or rdf:langString
- [sh:description](#) : xsd:string or rdf:langString

Especifica el nombre y la descripción para el property shape

Shape graph:

```
ex:ExamplePropertyShape
  a sh:PropertyShape ;
  sh:targetNode ex:Bob ;
  sh:path ex:email ;
  sh:name "Email" ;
  sh:description "Es necesario algún email" ;
  sh:minCount 1 .
```

Data graph:

```
ex:Bob a ex:Person ;
      ex:name "Bob".
```

En este caso se produciría una violación y se devolvería el nombre y la descripción del shape violado.

SHACL. sh:PropertyShape

Propiedades:

- [sh:qualifiedValueShape](#) : sh:Shape
- [sh:qualifiedMinCount](#) : xsd:integer
- [sh:qualifiedMaxCount](#) : xsd:integer

Especifica una expresión regular

Shape graph:

```
ex:QualifiedValueShapeExample
  a sh:NodeShape ;
  sh:targetNode ex:Alvaro;
  sh:property [
    sh:path ex:parent ;
    sh:qualifiedValueShape [
      sh:class ex:Mujer ;
    ] ;
    sh:qualifiedMinCount 1 ;
    sh:qualifiedMaxCount 1 ;
```

].

Data graph:

```
ex:Alvaro
  ex:parent ex:Petri ;
  ex:parent ex:Fonso .

ex:Petri a ex:Mujer
  ex:gender ex:female .

ex:Fonso a ex:Hombre
  ex:gender ex:male .
```

En este caso no se produciría una violación porque en la propiedad ex:parent sólo tiene una madre

SHACL. sh:PropertyShape

Propiedades:

- [sh:pattern](#) : xsd:string
Especifica una expresión regular

Shape graph:

```
ex:ExamplePropertyShape
  a sh:PropertyShape ;
  sh:targetNode ex:Bob ;
  sh:path ex:name ;
  sh:pattern "^B".
```

Data graph:

```
ex:Bob a ex:Person ;
      ex:name "Bob".
```

En este caso no se produciría una violación porque el nombre empieza por B

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. SHACL en ASIO



SHACL en ASIO

En este proyecto todas las interacciones SHACL se realizan en el API de Carga. Este API dispone de los métodos necesarios para añadir/modificar/eliminar restricciones SHACL al sistema.

También dispone de los métodos necesarios para ejecutar estas validaciones y para probar las validaciones antes de darlas de alta en el sistema.

<http://herc-as-front-des.a.atica.um.es/carga/swagger/index.html>

La interacción con la gestión de los Shapes configurado se puede realizar desde el FrontEndCarga: <https://herc-as-front-des.a.atica.um.es/carga-web/ShapeConfig>

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. Casos de Práctica.



Preparación

Para la realización de los ejercicios de SHACL utilizaremos el método `/etl/data-validate-personalized` del servicio API CARGA.

Para ello hay que ir a la siguiente url <http://herc-as-front-des.a.tica.um.es/carga/swagger/index.html> e introducir en la sección authorize lo siguiente:

Bearer

yJhbGciOiJSUzI1NiIsImtpZCI6Ijlk2c1BwYnhjZEZRkpjRG5iQnJVWciLCJOeXAiOiJhdCtqd3QifQ.eyJ0bmYiOiJlM0ODAsImV4cCI6MTYxNDM0OTg4MCwiaXNzIjoiaHR0cDovL2hlcmMtYXMtZnJvbncTqZGVzYS5hdGljYS51bS5lczo1MTA4IiwiaXYvkljoiYXBpQ2FyZ2EiLCJBjGllbnRfaWQiOiJJZWwiLCJzY29wZSI6WyJhcGlDYXNjYnJsdfQ.IxmwrIdIOba4VR45HnRCh6qEPvjxbve3wZB5_OTuovmK2w3Wq6r3BXNmkvIV4S34pbXjPbywMMznMcFnImZVIEeSRENT4PkXjWtB44yyJdhgndERUR1L7utZJW9cUDt_QSVBLwVnoiahBxxtg3qdOMgjXWNNGp02o1yUrb1adtnrBoO7Gulzh5ceZQ0BFtW6VHz0Ke6fwdIGrMrdlN9yd-wpvQG3LSCHHSBBirNW5drjxYrrGPrJhyaUloS9JgYwUXYbh1erCMzPGGO-Za1-l6fecHOGFwim4kAv3jh2fa002iqel10gaDtPsizIED42kpRx6lcNRl_5seFIDbA

Para los ejercicios vamos a partir de los siguientes RDF:

- datagraph.rdf: RDF con los datos para validar.
- shapegraph.ttl. RDF con los datos de validación.

Casos de uso. SHACL. Ejercicio 1

Crea una validación para que las personas tengan que tener obligatoriamente un nombre, para ello has de modificar el fichero 'shapegraph.ttl' para que el sh:NodeShape afecte a la clase adecuada.

Pista: hay que utilizar la propiedad 'sh:targetClass'

Casos de uso. SHACL. Ejercicio 2

Crea una validación para que el nombre de las personas sea de tipo texto, para ello has de modificar el fichero generado en el ejercicio anterior para especificar el tipo de dato.

Pista: hay que utilizar la propiedad 'sh:datatype'

Casos de uso. SHACL. Ejercicio 3

Modifica la validación anterior para que sólo afecte a la persona <http://graph.um.es/res/person/1491480>.

Pista: hay que utilizar la propiedad 'sh:targetNode'

Casos de uso. SHACL. Ejercicio 4

Añade una validación para validar que todos los objetos de una propiedad 'vivo:relatedBy' tienen 2 propiedades 'vivo:relates', una que punta a una persona y otra que apunta a una organización.

Pista: hay que utilizar la propiedades 'sh:targetObjectsOf', 'sh:qualifiedMinCount', 'sh:qualifiedMaxCount', 'sh:qualifiedValueShape' y 'sh:class'

Casos de uso. SHACL. Ejercicio 5

Haz que cuando se produzca la violación que se produce tras el ejercicio anterior se muestre el mensaje “Una entidad que es apuntada con vivo:relatedBy debe apuntar con vivo:relates a una Persona”.

Pista: hay que utilizar la propiedad ‘sh:message’

Casos de uso. SHACL. Ejercicio 6

Marca la violación anterior como sh:Info, en lugar de sh:Violation

Pista: hay que utilizar la propiedad 'sh:severity'

Casos de uso. SHACL. Ejercicio 7

Desactiva la validación que se está incumpliendo tras el ejemplo anterior.

Pista: hay que utilizar la propiedad 'sh:deactivated'



Casos de uso. SHACL. Ejercicio 8

Marcamos el node shape 'roh:Person_Shape' como sh:closed true y le modificamos el target de 'roh:Person_Shape' para que apunte a la clase foaf:Person

Observamos que se producen múltiples violaciones.

Casos de uso. SHACL. Ejercicio 9

Añade dentro del Shape 'roh:Person_Shape' todas las restricciones necesarias para que no se produzcan violaciones:

Es necesario añadir las propiedades `rdf:type`, `vivo:relatedBy` y `roh:ORCID` y eliminar de la restricción 'roh:Name_Person_Shape' el `datatype` y el `minCount`

Casos de uso. SHACL. Ejercicio 10

Añade una restricción con una expresión regular para que la propiedad 'vivo:identifier' de las personas tenga que ser obligatoriamente una URL.
La expresión regular a utilizar sería "^http:|https:)"

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

GRACIAS