

# SHACL

Mikel Egaña Aranguren

[mikel-egana-aranguren.github.io](https://mikel-egana-aranguren.github.io)

[mikel.egana@ehu.eus](mailto:mikel.egana@ehu.eus)



# SHACL

<https://github.com/mikel-egana-aranguren/ABD>



# SHACL

SHApes Constraint Language

Define la forma que un grafo RDF debería tener: que relaciones, a que nodos, con que valores, etc.

Se validan datos RDF contra esa forma, y se genera un informe que dice si los datos cumplen la forma o no (ValidationReport, en RDF)

# SHACL

SHACL es un recomendación (Estándar) del W3C: ([W3C](#))

El [vocabulario SHACL](#), definido en RDF, contiene todo el lenguaje SHACL

El mismo lenguaje RDF para definir datos (RDF) esquema (SHACL) e informes de resultados (ValidationReport) (NoSQL!RDF!)

# SHACL vs OWL

Confusion muy comun: OWL como lenguaje de esquema

OWL es un lenguaje de inferencia, no de esquema (\*): por ejemplo, no dice cuantas relaciones tiene que tener una instancia, si no que si nos encontramos una instancia con esas relaciones, se infiere que pertenece a cierta clase

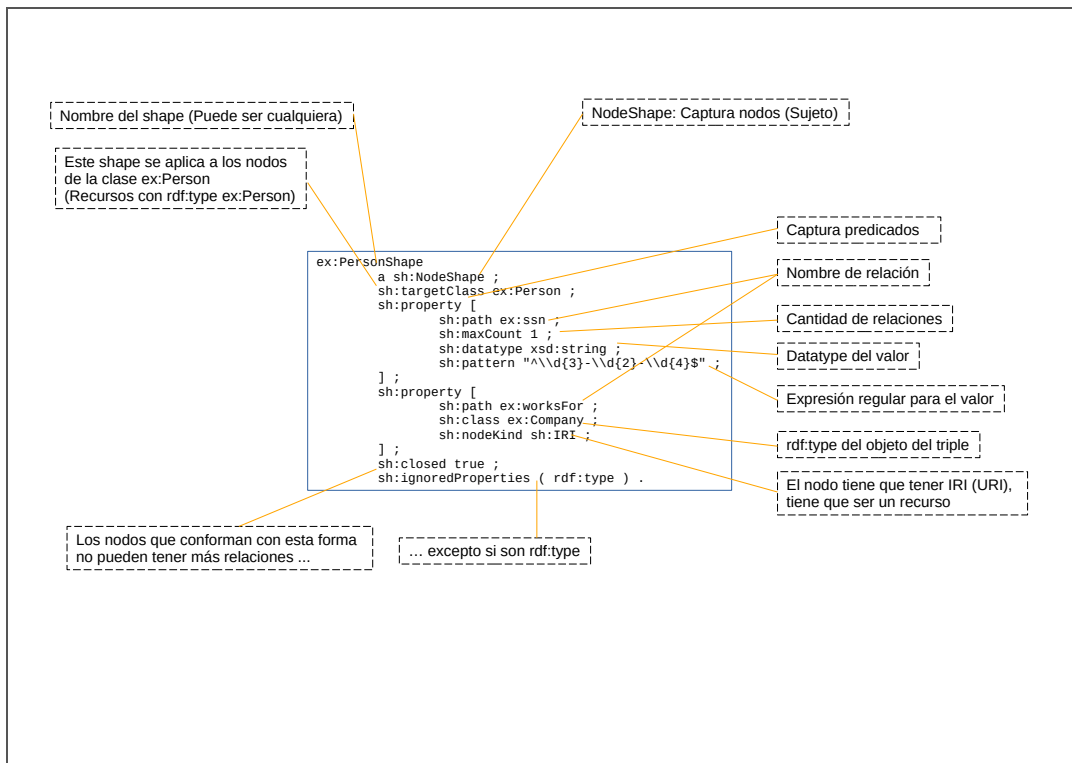
(\*) se puede conseguir "cerrar" el mundo en OWL, pero con muchos axiomas y no por defecto

# SHACL vs OWL

Confusion muy comun: OWL como lenguaje de esquema

SHACL si es un lenguaje de esquema: por ejemplo, nos dice cuantas relaciones tiene que tener una instancia, y si esa instancia no las tiene, es invalida

# Shape SHACL



# SHACL en GraphDB

1. Cargar Shapes (RDF)
2. Cargar datos
3. Si son validos, los datos se cargan
4. Si son invalidos, no se cargan y el por que del fallo aparece en el Validation Report



# SHACL en GraphDB

Crear repositorio activando opción SHACL

Los shapes residen en un grafo específico:

- Por defecto <http://rdf4j.org/schema/rdf4j#SHACLShapeGraph>
- Se pueden definir shapes en cualquier grafo y enlazarlo con los datos:

`ex:dataGraph sh:shapesGraph ex:GraphConShapes`

[Ejecutar ejemplos ex1, rdf\_type]