

# Introducción al entorno de desarrollo GNOSS Semantic Framework

IES Comercio

Logroño, 13 de Marzo de 2018

# Quién soy



Juan Valer

- Estudié Ingeniería Técnica en Informática de gestión en la UR (2003–2008)
- Trabajo en gnooss.com desde Verano de 2006 (¡casi 12 años!)
- Soy programador senior .NET, Scrum Master y el responsable de GNOSS Semantic Framework for Developers

[juanvaler@gnooss.com](mailto:juanvaler@gnooss.com)

@juanvalert

# Objetivos

- Darnos a conocer
- Conoceros a vosotros
- Poner a prueba GNOSS Semantic Framework
- Crear vuestra primera Web Semántica

# ¿Qué es GNOSS?



GNOSS es una plataforma Web que permite crear web semánticas de manera sencilla.

# ¿Y qué demonios es una Web semántica?

# Web semántica

Santo Domingo de la Calzada, La Rioja Alta

## PARADOR DE SANTO DOMINGO



DIRECCIÓN:

Plaza del Santo, 3 26250

TELÉFONO:

941 340 300

FAX:

941 340 325

E-MAIL:

[sto.domingo@parador.es](mailto:sto.domingo@parador.es)

PÁGINA WEB:

[www.parador.es](http://www.parador.es)

TIPO DE ESTABLECIMIENTO:

Restaurante

TIPO DE COMIDA:

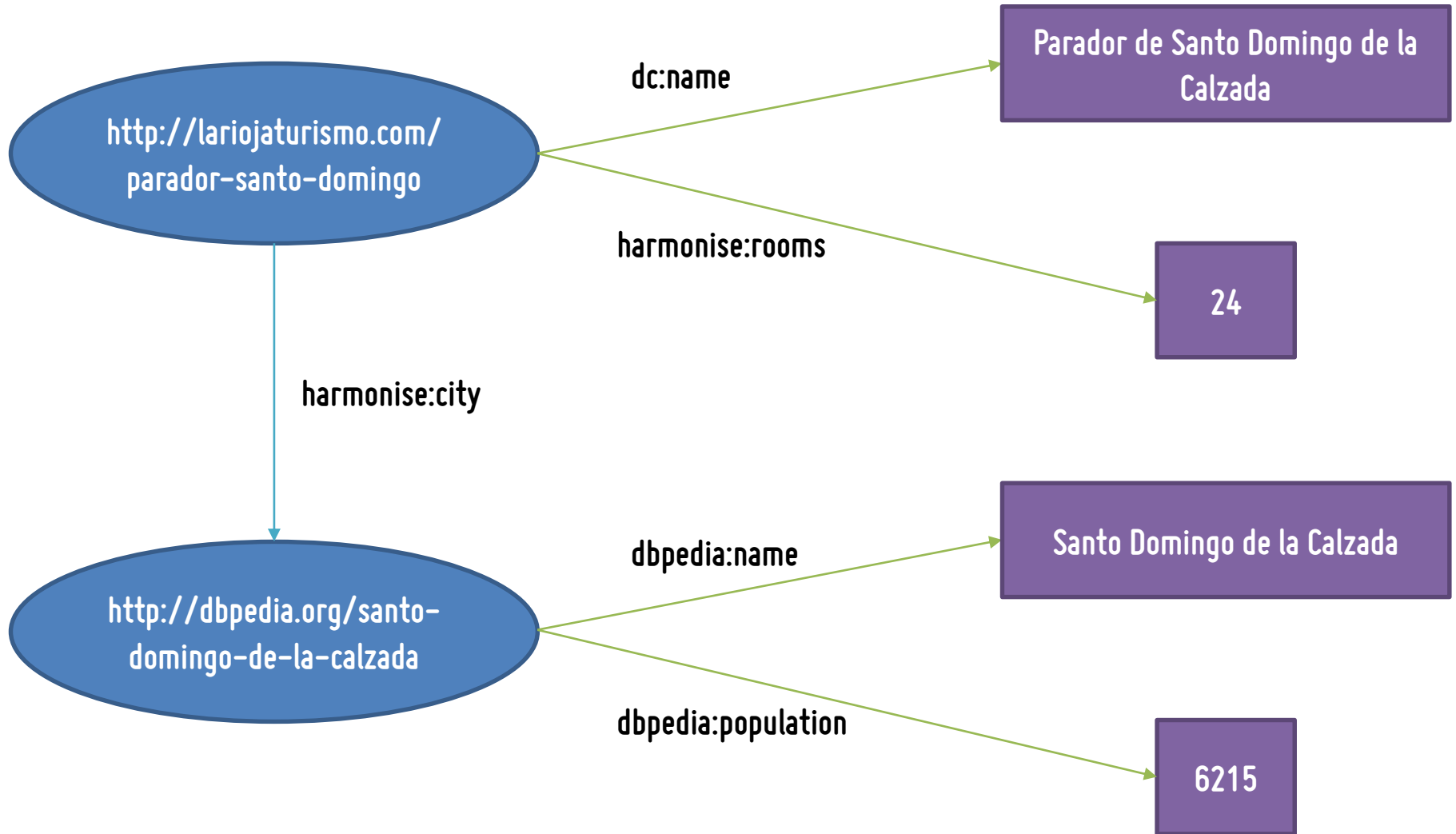
```
dégustation, menu végétarien, menu cœliaques, carte pour enfants et bébés.]]></harmonise:comment>
<harmonise:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string" xml:lang="de">
<![CDATA[Verkostungsmenü, vegetarisches Menü, Menü bei Zöliakie, Speisekarte für Kinder und für Babys. ]]>
</harmonise:comment>
<harmonise:streetAddress rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><![CDATA[Plaza del Santo, 3
26250]]></harmonise:streetAddress>
<harmonise:url rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><![CDATA[www.parador.es]]></harmonise:url>
<onTour:hasEndDate rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><![CDATA[Ninguno]]></onTour:hasEndDate>
<onTour:hasEndDate rdf:datatype="http://www.w3.org/2001/XMLSchema#string" xml:lang="en"><![CDATA[None]]>
</onTour:hasEndDate>
<onTour:hasEndDate rdf:datatype="http://www.w3.org/2001/XMLSchema#string" xml:lang="es"><![CDATA[Ninguno]]>
</onTour:hasEndDate>
<onTour:hasEndDate rdf:datatype="http://www.w3.org/2001/XMLSchema#string" xml:lang="de"><![CDATA[An keinem
Tag]]></onTour:hasEndDate>
<onTour:hasEndDate rdf:datatype="http://www.w3.org/2001/XMLSchema#string" xml:lang="fr"><![CDATA[Ouvert toute
l'année]]></onTour:hasEndDate>
<eharmonise:places rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><![CDATA[270]]></eharmonise:places>
<harmonise:email rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><![CDATA[sto.domingo@parador.es]]>
</harmonise:email>
```

# Web semántica





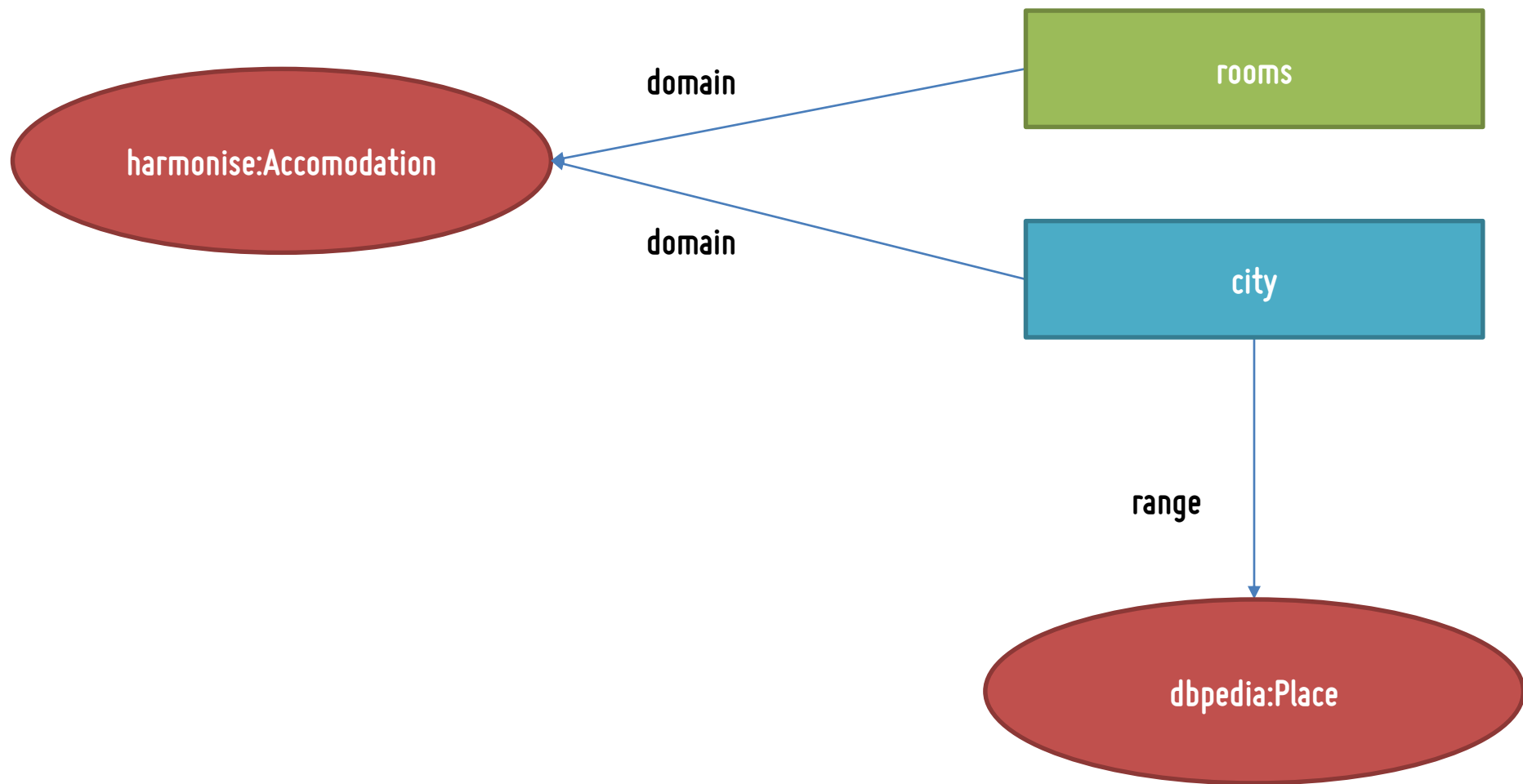
# RDF



# RDF

```
<http://lariojaturismo.com/parador-santo-domingo> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> "Accomodation"  
<http://lariojaturismo.com/parador-santo-domingo> <http://purl.org/dc/elements/1.1/title>  
"Parador de Santo Domingo de la Calzada"  
<http://lariojaturismo.com/parador-santo-domingo> <http://harmonet.org/harmonise.owl#rooms> 24  
<http://lariojaturismo.com/parador-santo-domingo> <http://harmonet.org/harmonise.owl#city>  
<http://dbpedia.org/santo-domingo-de-la-calzada>  
  
...  
  
<http://dbpedia.org/santo-domingo-de-la-calzada> <http://dbpedia.org/name> "Santo Domingo de la Calzada"  
<http://dbpedia.org/santo-domingo-de-la-calzada> <http://dbpedia.org/population> 6215
```

# OWL



# OWL

```
<owl:Class rdf:about="http://harmonet.org/harmonise.owl#Accommodation">
</owl:Class>

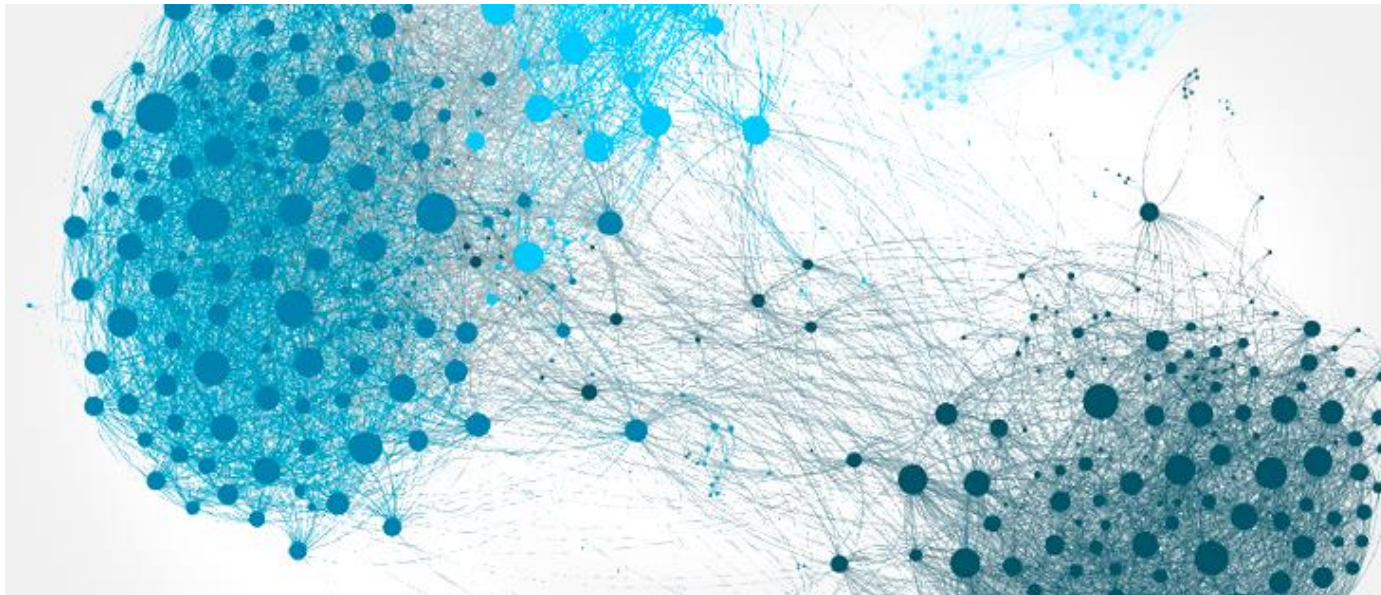
<owl:DatatypeProperty rdf:about="http://harmonet.org/harmonise.owl#rooms">
  <rdfs:label xml:lang="es">Número de habitaciones</rdfs:label>
  <rdfs:label xml:lang="en">Rooms number</rdfs:label>
  <rdfs:comment xml:lang="es">Número total de habitaciones que ofrece el alojamiento</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="http://harmonet.org/harmonise.owl#Accommodation"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:about="http://harmonet.org/harmonise.owl#city">
  <rdfs:range rdf:resource="http://dbpedia.org/Place"/>
  <rdfs:domain rdf:resource="http://harmonet.org/harmonise.owl#Accommodation"/>
  <rdfs:comment xml:lang="es">Localidad</rdfs:comment>
  <rdfs:comment xml:lang="en">City</rdfs:comment>
</owl:ObjectProperty>
```

# ¿Por qué (para qué) usar Web semántica?

# Ventajas de la Web semántica

- Información estructurada
- Datos abiertos
- Contenido multi-idioma
- Uso de bases de datos de tipo grafo



# Ejemplo de consulta SPARQL

```
SELECT ?sujeto ?predicado ?objeto
FROM <http://lariojaturismo.com>
WHERE
{
    ?sujeto rdf:type 'Accomodation'.

    ?sujeto <http://harmonet.org/harmonise.owl#city> ?city.

    ?city dbpedia:name ?cityName.

    FILTER(?cityName = "Santo Domingo de la Calzada").

    ?sujeto ?predicado ?objeto.
}
```

# Ejemplo de consulta SPARQL

```
SELECT ?name ?rooms
FROM <http://lariojaturismo.com>
WHERE
{
    ?sujeto rdf:type 'Accomodation'.

    ?sujeto <http://harmonet.org/harmonise.owl#city> ?city.

    ?city dbpedia:name ?cityName.

    FILTER(?cityName = "Santo Domingo de la Calzada") .

    ?sujeto haromnise:rooms ?rooms.

    ?sujeto dc:name ?name.
}
```



# Ejemplo de consulta SPARQL

```
SELECT SUM(?rooms)
FROM <http://lariojaturismo.com>
WHERE
{
    ?sujeto rdf:type 'Accomodation'.

    ?sujeto <http://harmonet.org/harmonise.owl#city> ?city.

    ?city dbpedia:name ?cityName.

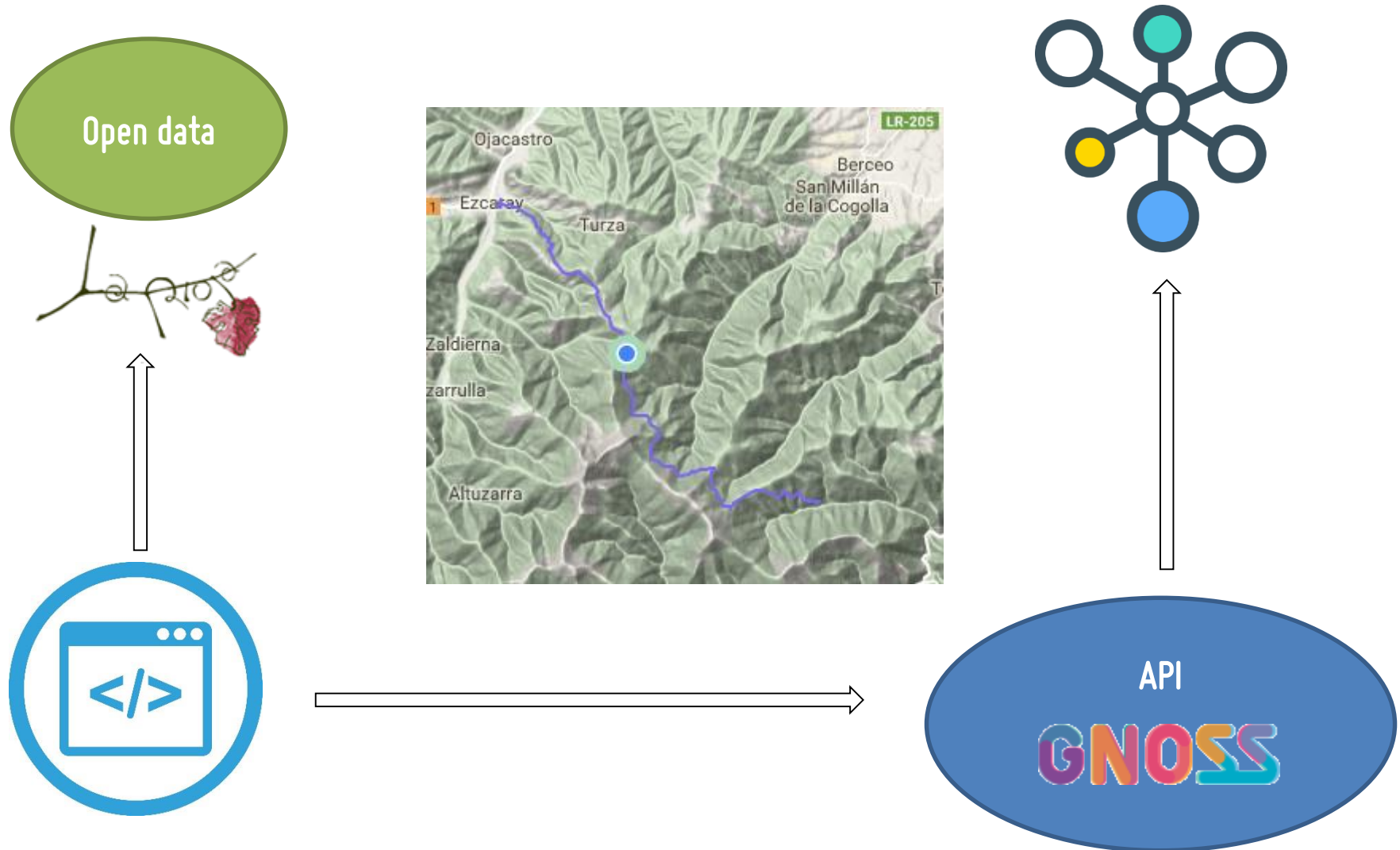
    FILTER(?cityName = "Santo Domingo de la Calzada") .

    ?sujeto haromnise:rooms ?rooms.
}
group by (?sujeto)
```

# Repasamos...

- Qué es GNOSS
- Qué es la Web semántica
- Qué es RDF y OWL
- Qué es una base de datos de tipo Grafo

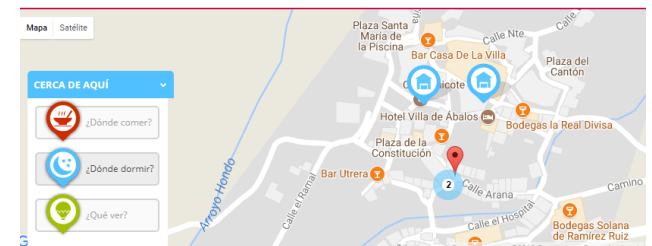
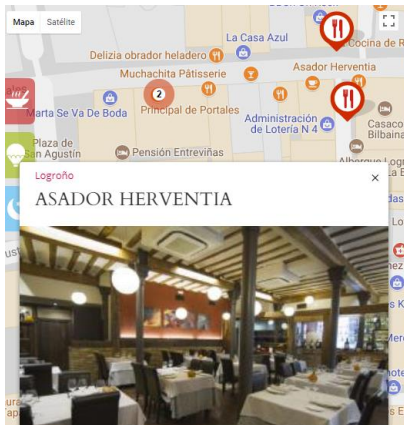
# Qué os hemos preparado...



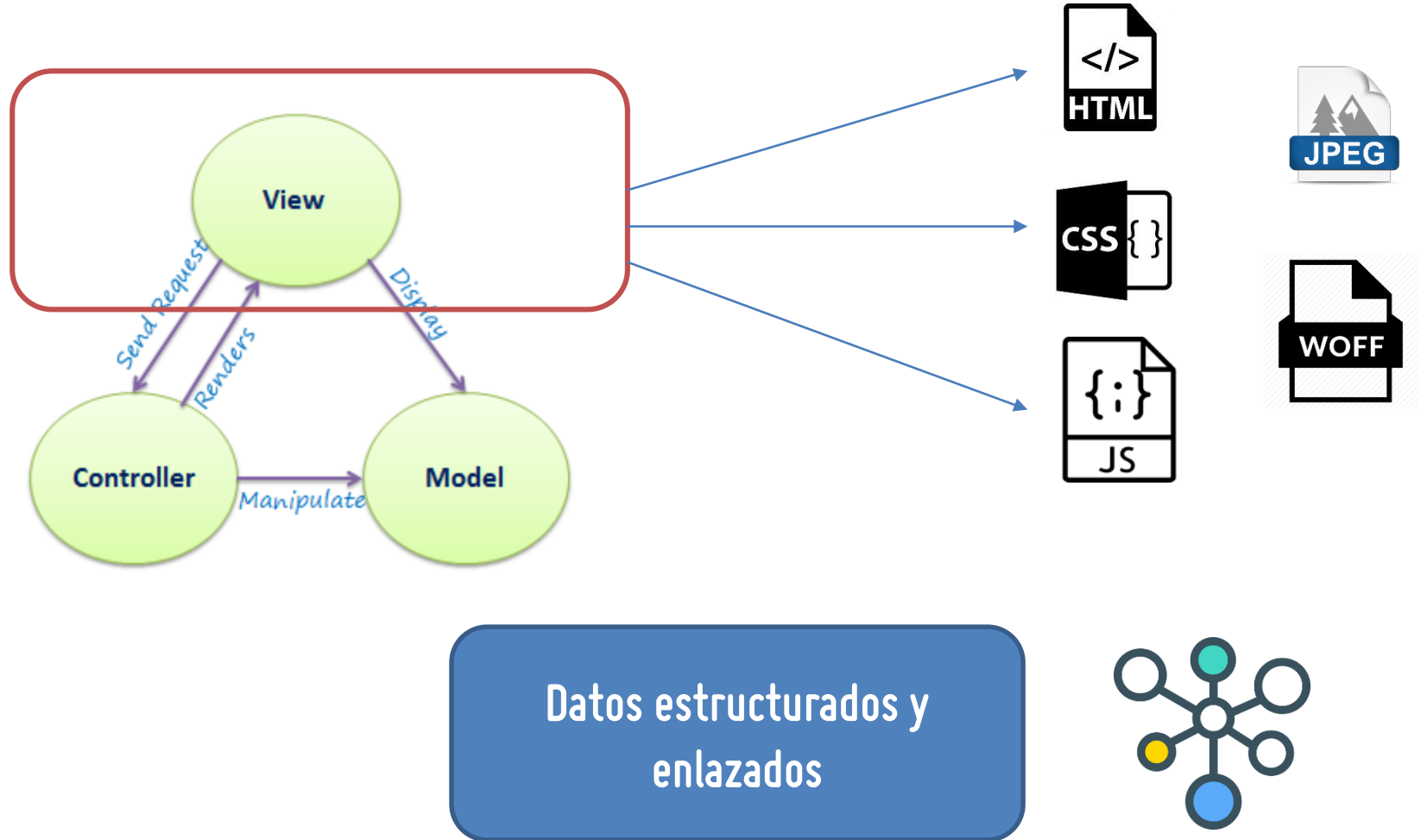
# Qué queremos que hagáis



# Algunos ejemplos...



# Y todo lo que se os ocurra...



# DataSets de La Rioja Turismo

- Punto interés (rtlandmark.owl)
- Servicio turístico (touristservice.owl)
- Alojamiento (accommodation.owl)
- Lugar de interés (attraction.owl)
- Ruta (rtpath.owl)
- Etapa (rtroute.owl)
- Destino (destination.owl)
- Transporte (transport.owl)
- Restaurante (gastro.owl)
- Bodega (winery.owl)

# ¡Habrá premio para el mejor equipo!





# Talk is cheap, show me the code

@Linus Torvalds

# Lectura de datos

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private static Dictionary<string, Entity> LeerRutas()
{
    Dictionary<string, Entity> entityList = new Dictionary<string, Entity>();
    string routesCsvFile = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "data", "rtroute.csv");

    using (CsvReader csvReader = new CsvReader(new StreamReader(routesCsvFile), true))
    {
        while (csvReader.ReadNextRecord())
        {
            string subject = csvReader[0];
            string predicate = csvReader[1];
            string obj = csvReader[2];
            string lang = null;

            if (csvReader.FieldCount > 3)
            {
                lang = csvReader[3];
            }

            if (!entityList.ContainsKey(subject))
            {
                entityList.Add(subject, new Entity(subject));
            }
            Entity entity = entityList[subject];

            if (!entity.Properties.ContainsKey(predicate))
            {
                entity.Properties.Add(predicate, new Property(predicate));
            }
            Property property = entity.Properties[predicate];

            property.Objects.Add(new ObjectWithLanguage() { Value = obj, Language = lang });
        }
    }
    return entityList;
}
```

# Pasar datos a triples

```
16 referencias | - cambios | -autores, - cambios
public class Entity : IEquatable<Entity>
{
    1 referencia | - cambios | -autores, - cambios
    public Entity(string subject)
    {
        Subject = subject;
        Properties = new Dictionary<string, Property>();
    }

    3 referencias | - cambios | -autores, - cambios
    public string Subject { get; set; }

    86 referencias | - cambios | -autores, - cambios
    public Dictionary<string, Property> Properties { get; }

    63 referencias | - cambios | -autores, - cambios
    public bool Equals(Entity other)
    {
        return Subject.Equals(other.Subject);
    }
}

5 referencias | - cambios | -autores, - cambios
public class Property
{
    1 referencia | - cambios | -autores, - cambios
    public Property(string predicate)
    {
        Predicate = predicate;
        Objects = new List<ObjectWithLanguage>();
    }

    2 referencias | - cambios | -autores, - cambios
    public string Predicate { get; set; }

    55 referencias | - cambios | -autores, - cambios
    public List<ObjectWithLanguage> Objects { get; }
}

28 referencias | - cambios | -autores, - cambios
public class ObjectWithLanguage
{
    54 referencias | - cambios | -autores, - cambios
    public string Value { get; set; }
    24 referencias | - cambios | -autores, - cambios
    public string Language { get; set; }
}
```

# Pasar datos a triples

```
if (route.Properties.ContainsKey(Constants.RouteProperties.Name))
{
    recurso.Title = route.Properties[Constants.RouteProperties.Name].Objects[0].Value;

    foreach (ObjectWithLanguage obj in route.Properties[Constants.RouteProperties.Name].Objects)
    {
        listPropiedades.Add(new StringOntologyProperty(Constants.RouteProperties.NameWithNamespace, obj.Value, obj.Language));
    }
}

if (route.Properties.ContainsKey(Constants.RouteProperties.Description))
{
    recurso.Description = route.Properties[Constants.RouteProperties.Description].Objects[0].Value;

    foreach (ObjectWithLanguage obj in route.Properties[Constants.RouteProperties.Description].Objects)
    {
        listPropiedades.Add(new StringOntologyProperty(Constants.RouteProperties.DescriptionWithNamespace, obj.Value, obj.Language));
    }
}

if (route.Properties.ContainsKey(Constants.RouteProperties.Ads))
{
    recurso.Description = route.Properties[Constants.RouteProperties.Ads].Objects[0].Value;

    foreach (ObjectWithLanguage obj in route.Properties[Constants.RouteProperties.Ads].Objects)
    {
        listPropiedades.Add(new StringOntologyProperty(Constants.RouteProperties.AdsWithNamespace, obj.Value, obj.Language));
    }
}

if (route.Properties.ContainsKey(Constants.RouteProperties.Color))
{
    recurso.Description = route.Properties[Constants.RouteProperties.Color].Objects[0].Value;
```

# Cargar recursos en GNOSS

1 referencia | 0 cambios | 0 autores, 0 cambios

```
private static void SubirRutas(Dictionary<string, Entity> entityList)
{
    List<Entity> rutas = entityList.Values.Where(ent => ent.Properties.Values.Any(prop => prop.Predicate.Equals($"{Constants.Ontologies.Rdf}type") &&
        prop.Objects.Exists(obj => obj.Value.Equals($"{Constants.Ontologies.Route}Route")))).ToList();

    ResourceApi resourceAPI = new ResourceApi(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "config", "config.xml"));

    foreach (Entity ruta in rutas)
    {
        ComplexOntologyResource resource = CreateComplexOntologyResourceRoute(entityList, ruta, resourceAPI);

        try
        {
            resourceAPI.LoadComplexSemanticResource(resource);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}
```

# Acceso a las comunidades

- <http://try.gnoss.com>
- Desde el menú comunidades podéis acceder a vuestra comunidad
- [Entrad en developers.gnoss.com y daros de alta](http://developers.gnoss.com)

# Primeros pasos...

Subir ontología y configuración XML (Administración semántica -> Plantillas semánticas)

## Plantillas semánticas

### Agregar plantilla

- ☒ Crear Ontología
- ☐ Crear Ontología secundaria

Puedes crear una ontología desde cero usando alguno de los siguientes programas:

[Protégé](#) – Universidad de Stanford

[TopBraid Composer](#) – TopQuadrant

O elegir algún otro del siguiente [listado de Wikipedia](#)

O puedes [solicitar](#) que se la desarrollemos en GNOSS

Cancelar

Siguiente

# Primeros pasos...

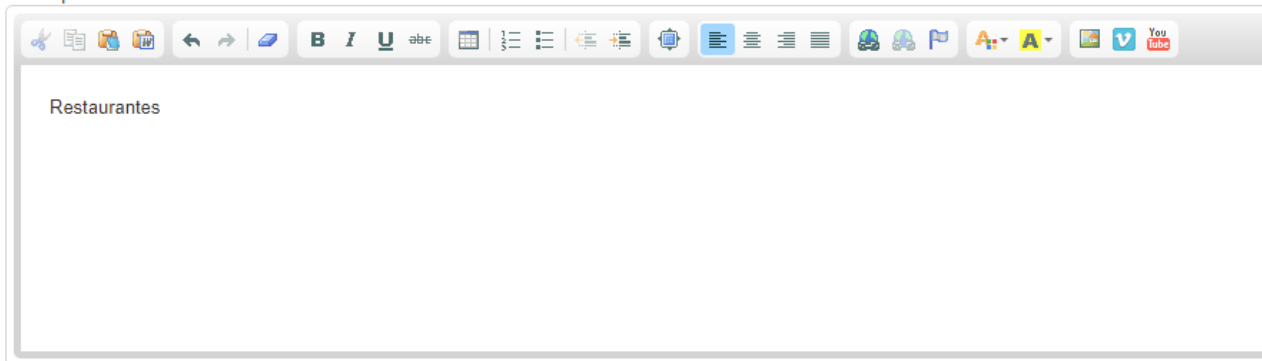
Subir ontología y configuración XML (Administración semántica -> Plantillas semánticas)

## Plantillas semánticas

### Agregar plantilla

Título

Descripción

A rich text editor interface with a toolbar at the top containing icons for undo, redo, bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, insert table, insert image, and YouTube. The main text area contains the word "Restaurantes".

Restaurantes

Selecciona la ontología de la plantilla, si ya la tienes disponible

accommodation.owl

Selecciona el archivo de configuración para el Html de la plantilla

accommodation.xml

☐ Utilizar archivo de configuración genérico



# Primeros pasos...

Crear objeto de conocimiento (Administración semántica -> objetos de conocimiento)

## Configurar Objetos de Conocimiento

Configura los objetos de conocimiento de tu comunidad

Añade un objeto de conocimiento para que sea buscable

Restaurantes ▼

Añadir Objeto de Conocimiento

Guardar Todo

Ruta – rtroute.owl  

Nombre de la ontología ?	<input type="text" value="Ruta"/>
Nombre corto de la ontología ?	<input type="text" value="ruta"/>
Namespace ?	<input type="text" value="rtpath"/>
Namespaces extra ?	<input type="text" value="eroute:http://www.eroute/2014-04-28# route:http://www.obertadecatalunya/route# wgs84_pos:http://www.w3.org/2003/01/geo/wgs84_pos#"/>
Nombre de la categoría de tesauro exclusivo ?	<input type="text"/>
Cachear datos semánticos ?	<input checked="" type="checkbox"/>
Es buscable ?	<input checked="" type="checkbox"/>

# Primeros pasos...

## Dar permisos al usuario administrador (Administración básica -> Utilidades)

### Configuración de las opciones de los recursos

En una comunidad **privada**, el administrador decide si permite que sus recursos puedan ser compartidos por sus miembros en las diferentes comunidades en las que participa, siempre y cuando estos tengan permiso para hacerlo. Esta disponibilidad depende del tipo de recurso y del rol de cada miembro. Aquí se muestra el check donde decide el administrador si permite compartir.

#### Tipos de recursos disponibles para cada usuario

Los roles de los miembros de la comunidad, por niveles de permisos en la misma ordenados de mayor a menor, pueden ser: administrador, supervisor y usuario. Cada rol incorpora los privilegios del inmediatamente inferior. Así, un recurso disponible para un supervisor también lo estará para un administrador, pero no para un usuario.

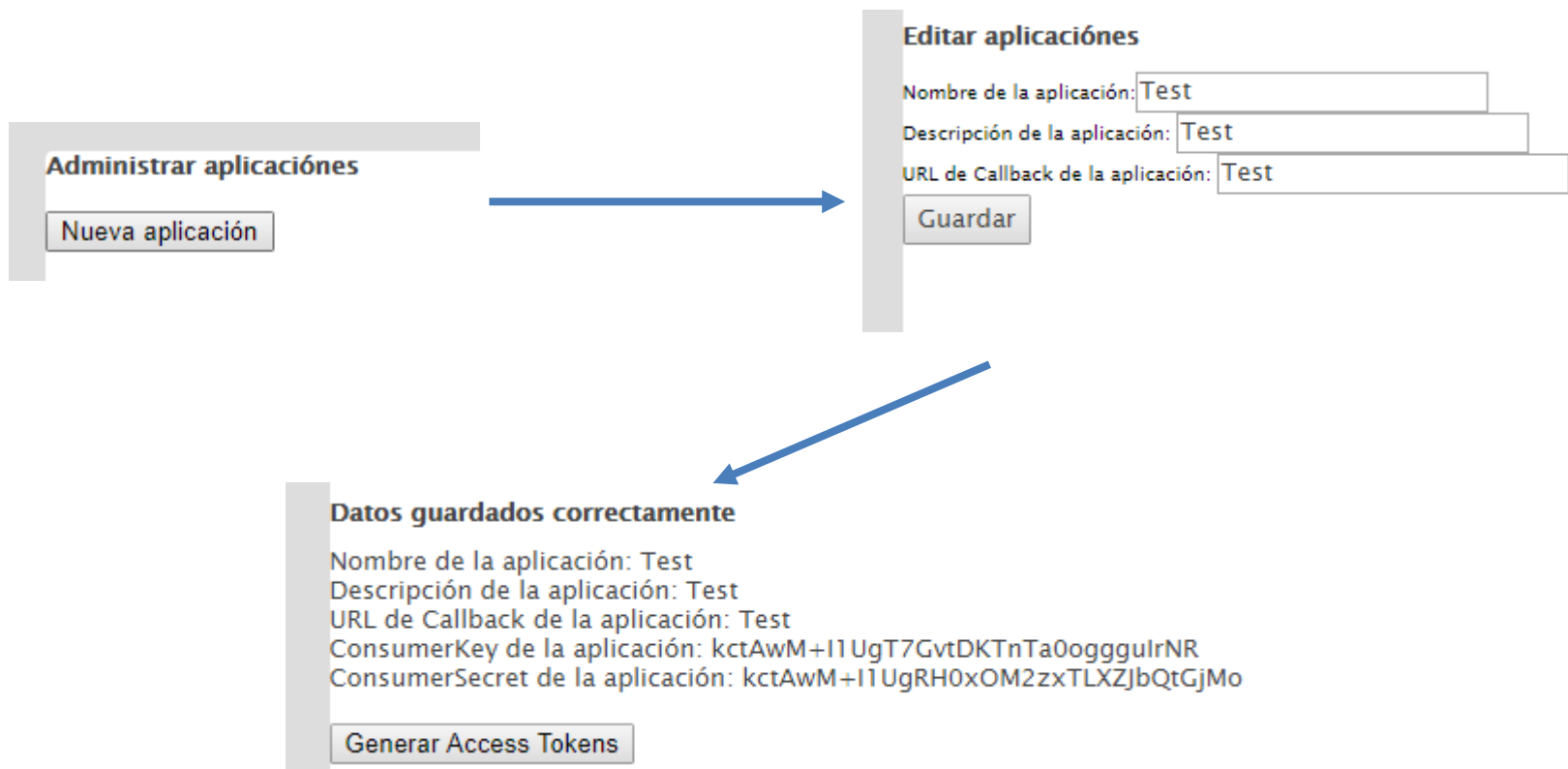
Tipo de recurso	Disponible para		
Archivo digital	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
<input checked="" type="radio"/> Cualquier tipo			
<input type="radio"/> Sólo videos			
<input type="radio"/> Sólo imágenes			
Referencia a documento físico	<input type="checkbox"/> Usuario	<input type="checkbox"/> Supervisor	<input type="checkbox"/> Administrador
Dirección de internet	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Nota	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Artículo wiki	<input type="checkbox"/> Usuario	<input type="checkbox"/> Supervisor	<input type="checkbox"/> Administrador
Pregunta	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Encuesta	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Debate	<input checked="" type="checkbox"/> Usuario	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Documento a partir de una plantilla	<input type="checkbox"/> Usuario	<input type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
<i>Seleccionar todos</i>	<input type="checkbox"/> Usuario	<input type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador
Ruta	<input type="checkbox"/> Usuario	<input type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> Administrador

[Añadir grupos](#)

Descarga de ficheros

# Primeros pasos...

Crear tokens OAuth (Administración desarrolladores -> Administrar aplicaciones)



# Primeros pasos...

## Configurar tokens Oauth (en el archivo config/config.xml)

```
<?xml version="1.0" standalone="yes"?>
<config>
  <apiEndpoint>http://serviciostry.gnoss.com/api/v3</apiEndpoint>
  <communityShortName></communityShortName>
  <developerEmail>test@gnos.com</developerEmail>
  <ontologyName>rtroute.owl</ontologyName>
  <consumer>
    <consumerKey></consumerKey>
    <consumerSecret></consumerSecret>
  </consumer>
  <token>
    <tokenKey></tokenKey>
    <tokenSecret></tokenSecret>
  </token>
</log>
  <logPath>c:\logs</logPath>
  <logFileName>gnossapi.log</logFileName>
  <logLevel>TRACE</logLevel>
</log>
</config>
```

# Primeros pasos...

- Descargar el código fuente:  
<https://github.com/equipognoss/DemoRutasLRT>
- Si no vais a user .NET, descargaros el API Wrapper:  
<https://github.com/equipognoss/GnossApiWrapper>
- Si vais a editar vistas:  
<https://github.com/equipognoss/GnossTestView>

# Si vais a crear una ontología nueva...

- Crear plantilla con XML por defecto
- Descargar archivo de configuración
- Añadir estas opciones:

```
<idiomasOnto>
  <idiomaOnto>es</idiomaOnto>
</idiomasOnto>
<ocultarTituloDescpImgDoc></ocultarTituloDescpImgDoc>
```

```
<TituloDoc EntidadID="http://protege.stanford.edu/rdf/HTOv4002#Accommodation">http://protege.stanford.edu/rdf/HTOv4002#name</TituloDoc>
<DescripcionDoc EntidadID="http://protege.stanford.edu/rdf/HTOv4002#Accommodation">http://protege.stanford.edu/rdf/HTOv4002#longDescription</DescripcionDoc>
<CategorizacionTesauroGnossObligatoria>false</CategorizacionTesauroGnossObligatoria>
```

```
<HtmlNuevo>true</HtmlNuevo>
<IncluirIconoGnoss></IncluirIconoGnoss>
<ImagenDoc EntidadID="http://www.eharmonise/2014-03-12#Image">http://www.eharmonise/2014-03-12#image</ImagenDoc>

<MultiIdioma> </MultiIdioma>
<OcultarEditoresDoc></OcultarEditoresDoc>
<OcultarPublicadorDoc></OcultarPublicadorDoc>
```

# Gracias

[juanvaler@gnoss.com](mailto:juanvaler@gnoss.com)  
[@juanvalert](#)