Logroño, diciembre-2019

Hércules backend asio. ESPECIFICACIÓN DE ESQUEMA DE URIS

Proyecto Hércules

[1 INTRODUCCIÓN 3](#_Toc27349011)

[2 Características del Esquema de URIs 4](#_Toc27349012)

[3 Estructura del Esquema de URIs 5](#_Toc27349013)

[3.1 Base 5](#_Toc27349014)

[3.2 Carácter de la información 5](#_Toc27349015)

[3.3 Sector o ámbito 6](#_Toc27349016)

[3.4 dominio o temática 6](#_Toc27349017)

[3.5 conceptos específicos 6](#_Toc27349018)

[4 Tipos de URIs 7](#_Toc27349019)

[4.1 URI para identificar vocabularios 7](#_Toc27349020)

[4.2 URI para identificar esquemas de conceptos 7](#_Toc27349021)

[4.3 URI para identificar a cualquier instancia física o conceptual 7](#_Toc27349022)

[5 Normalización de los componentes de los URI 8](#_Toc27349023)

[6 Prácticas relativas a la gestión de recursos semánticos a través de URI 9](#_Toc27349024)

[7 Definición del Esquema de URIs para la Factoría de URIs 10](#_Toc27349025)

# INTRODUCCIÓN

El presente documento describe la Especificación del Esquema de URIs de Hércules. En su elaboración tenemos en cuenta las recomendaciones de la [Norma Técnica de Interoperabilidad de Reutilización de recursos de la Información](https://www.boe.es/boe/dias/2013/03/04/pdfs/BOE-A-2013-2380.pdf) (NTI), de la Secretaría de Estado de Administraciones Públicas; que se basan, a su vez, en las iniciativas de datos abiertos y las recomendaciones procedentes del mundo de la Web Semántica.

Como veremos, estas recomendaciones serán adaptadas al ámbito de un sistema de investigación universitaria; y tendrán presente la resolución y conexión de las entidades identificadas por la URIs.

# Características del Esquema de URIs

Este documento de diseño del Esquema de URIs Hércules tendrá unos requisitos genéricos similares a los utilizados en la citada NTI. Estos requisitos serán:

1. Utilizar el protocolo HTTP, de forma que se garantice la resolución de cualquier URI en la web.
2. Usar una estructura de composición de URI consistente, extensible y persistente. Las normas de construcción de los URI seguirán unos patrones determinados que ofrezcan coherencia en la uniformidad, los cuales podrán ser ampliados o adaptados en caso de necesidad.
3. Seguir una estructura de composición de URIs comprensible y relevante. Esto significa que el propio identificador debe ofrecer información semántica autocontenida, lo que permitirá a cualquier agente reutilizador disponer de información sobre el propio recurso, así como su procedencia.
4. No exponer información sobre la implementación técnica de los recursos que representan los URIs. En la medida de lo posible se omitirá información específica sobre la tecnología subyacente del recurso representado; por ejemplo, no se incluirán las extensiones correspondientes a tecnologías con las que se generan los recursos web como .php, .jsp, etc.
5. Cumplir el principio de persistencia de los URIs, lo que significa que los que ya han sido creados previamente nunca deberían variar, y que el contenido al que hacen referencia debería ser accesible. En el caso de que sea necesario cambiar o eliminar el recurso al que apunta un identificador, se deberá establecer un mecanismo de información sobre el estado del recurso usando los códigos de estado de HTTP. En el caso de poder ofrecer una redirección a la nueva ubicación del recurso, se utilizarán los códigos de estado HTTP 3XX, mientras que para indicar que un recurso ha desaparecido permanentemente se utilizará el código de estado HTTP 410.

# Estructura del Esquema de URIs

Los URIs generados tendrán una estructura uniforme que proporcionará coherencia al esquema de URIs de Hércules ASIO como sistema de representación de los recursos, de acuerdo con los requisitos genéricos previamente descritos y proporcionará información intuitiva sobre la procedencia y el tipo de información identificada.

Además de la base, los elementos que compondrán la ruta del URI serán: carácter de la información y, opcionalmente, dominio, concepto y extensión/formato. El orden de los elementos es el siguiente (se señalan entre corchetes los elementos opcionales):

http://{base}/{carácter}[/{dominio}][/{concepto}][.{ext}]

Definimos a continuación cada uno de los elementos de la ruta del URI.

## Base

Es un componente obligatorio que define el espacio y dominio dedicado por cada universidad al albergue de la plataforma de datos abiertos y reutilizables. Recomendamos el uso de un subdominio, por ejemplo:

http://datos.um.es

http://datos.crue.org

http//sgi-data.deusto.es

## Carácter de la información

Es un componente obligatorio, que puede tomar una de las siguientes formas:

* catalogo o cat. En principio, el proyecto HERCULES ASIO no contempla el alojamiento de datasets como medio de publicación de datos abiertos, sino que todo el portal será un sistema Linked Data interrogable mediante un API y un punto SPARQL. No parece necesario que existan declaraciones de datasets.
* def. Indica que el recurso identificado es un vocabulario u ontología definido por OWL.
* kos. Indica que se trate de un sistema de organización del conocimiento (Knowledge Organization System) en el dominio de SGI: tesauros, taxonomías, etc.
* res. Indica que se trata de una entidad del dominio.

## Sector o ámbito

Es un componente opcional de posible aplicación en URIs de organización de conocimiento.

## dominio o temática

Es un componente opcional de posible aplicación en URIs de organización de conocimiento o en entidades que puedan tener subclases. Por ejemplo, podría servir para distinguir el tema de una publicación.

## conceptos específicos

Es un componente opcional del URI, pero funcionalmente obligatorio si se trata de declarar entidades del ámbito de SGI, como: investigador, publicación, proyecto, etc.

# Tipos de URIs

A continuación, se especifican los tipos de URI específicos para recursos semánticos de una iniciativa basada en *Linked Data*.

## URI para identificar vocabularios

Cualquier vocabulario u ontología seguirá el esquema: http://{base}/def/{sector}/{dominio}

## URI para identificar esquemas de conceptos

Cualquier sistema de organización del conocimiento –taxonomías, diccionarios, tesauros, etc.– sobre un dominio concreto será identificado mediante un esquema de URI basado en la estructura: http://{base}/kos/{sector}/{dominio}

## URI para identificar a cualquier instancia física o conceptual

Estos recursos son las representaciones atómicas de los documentos y recursos de información. A su vez suelen ser instancias de las clases que se definen en los vocabularios. Estos recursos se identifican mediante el esquema: http://{base}/res/{sector}[/{dominio}]/{clase}/{ID}

Por ejemplo: http://data.um.es/res/investigador/{id-investigador}

Las instancias físicas o conceptuales que se incluirán como fragmentos en las URIs se corresponderán con las entidades identificadas en la Red de Ontologías Hércules (ROH), como: researcher/investigador, project/proyecto, publication/publicación, etc.

# Normalización de los componentes de los URI

Para garantizar la coherencia y el mantenimiento posterior del esquema de URI se aplicarán las siguientes reglas para normalizar las distintas partes que componen los URI:

1. Seleccionar identificadores alfanuméricos cortos únicos, que sean representativos, intuitivos y semánticos.
2. Usar siempre minúsculas, salvo en los casos en los que se utilice el nombre de la clase o concepto. Habitualmente, los nombres de las clases se representan con el primer carácter de cada palabra en mayúsculas.
3. Eliminar todos los acentos, diéresis y símbolos de puntuación. Como excepción puede usarse el guión (–).
4. Eliminar conjunciones y artículos en los casos de que el concepto a representar contenga más de una palabra.
5. Usar el guión (–) como separador entre palabras.
6. Evitar en la medida de lo posible la abreviatura de palabras, salvo que la abreviatura sea intuitiva.
7. Los términos que componen los URI deberán ser legibles e interpretables por el mayor número de personas posible, por lo que se utilizará el castellano, cualquiera de las lenguas oficiales de España o el inglés como lengua franca de la investigación.

# Prácticas relativas a la gestión de recursos semánticos a través de URI

Las siguientes prácticas se desarrollarán como requisitos del servidor Linked Data de Hércules ASIO y se aplicarán para la gestión de recursos semánticos descritos en RDF:

1. Siempre que sea posible, y existan versiones del recurso en formato legible para personas HTML o similar y RDF, el servidor que gestiona los URI realizará negociación del contenido en función de la cabecera del agente que realiza la petición. En el caso de que el cliente acepte un formato de representación RDF en cualquiera de sus notaciones (p.e., especificando en su cabecera que acepta el tipo MIME application/rdf+xml) se servirá el documento RDF a través del mecanismo de redirecciones alternativas mediante los códigos de estado HTTP 3XX.
2. En el caso de que no se realice una negociación del contenido desde el servidor y, para favorecer el descubrimiento de contenido RDF desde los documentos HTML relacionados con las descripciones de los recursos, se incluirán enlaces a la representación alternativa en cualquiera de las representaciones en RDF desde los propios documentos HTML de la forma <link rel=«alternate» type=«application/rdf+xml» href=«documento.rdf»> o similar. En esa sentencia se incluye el tipo de formato MIME del documento (application/rdf+xml, text/n3, etc.).
3. Cuando se establezcan enlaces entre distintos recursos de información, se procurará la generación de enlaces que conecten los recursos bidireccionales para facilitar la navegación sobre los recursos de información en ambos sentidos.

# Definición del Esquema de URIs

El Esquema de URIs tiene que declararse en un formato informático que pueda ser interpretado por la Factoría de URIs para devolver el identificador único que precisa cada entidad cargada en ASIO. La propuesta se declara como un JSON y tiene la siguiente forma:

[

{

"base": {url},

"characters": [

{

"character": {nombre del carácter},

"labelCharacter": {etiqueta del carácter}

}

]

"uriResourceStructure": [

{

"uriComponent": {identificador del componente},

"uriComponentValue": {origen del valor del componente},

"uriComponentOrder": {orden del componente en el URI},

"mandatory": {true or false},

"finalCharacter": {si lo tiene, será una barra “/”}

}

]

"resourcesClasses": [

{

"resourceClass": {se corresponde con una entidad de ROH},

"labelResourceClass": {opcional, etiqueta de la entidad},

"resourceURI": {nombre de la estructura de URI que aplica}

}

]

}

]

Las partes del formato anterior son:

* base. Contendrá el dominio en el que se encuentra el servidor que resolverá el URI.
* characters. Podrá tener más de uno, pero al menos necesitará el que indica las entidades (res). Por cada carácter se indica:
  + character. Identifica al tipo de carácter del URI.
  + labelCharacter. Devuelve el fragmento del URI para el tipo de carácter.
* uriResourceStructure. Podrá tener más de uno, pero al menos necesitará uno que indique como se compone el URI de las entidades de la ROH. La composición del URI se declara mediante elementos que tienen los siguientes atributos:
  + uriComponent. Identifica el componente.
  + uriComponentValue. Identifica como se obtiene el valor del componente. Las opciones pueden ser:
    - {porción del JSON@IDENTIFICADOR}, cuando el valor está definido en el propio esquema.
    - {@IDENTIFICADOR}, cuando el valor se suministra desde la aplicación que invoca a la factoría de URIs.
  + uriComponentOrder. Define el orden del componente en el URI devuelto.
  + mandatory. Indica si el componente del URI es obligatorio.
  + finalCharacter. Indica si el componente lleva una barra para la composición de un URI correcto.
* resourceClasses. Tendrá tantos elementos como entidades de ROH necesiten disponer de un URI a través de la Factoría de URIs. Los atributos por los que se declara como se compone el URI para cada entidad son:
  + resourceClass. Identificador del tipo de entidad.
  + labelResourceClass. Opcional, se declara si se desea que la URL tenga otro texto, habitualmente por requisitos de idioma.
  + resourceURI. Identifica el elemento uriResourceStructure que se usará para componer el URI de la entidad.

Se indica a continuación un ejemplo:

[

{

"base": "http://datos.um.es",

"characters": [

{

"character": "resource",

"labelCharacter": "res"

}

{

"character": "kos"

"labelCharacter": "kos"

}

]

"uriResourceStructure": [

{

"uriComponent": "base",

"uriComponentValue": "base",

"uriComponentOrder": 1,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "character",

"uriComponentValue": "character@RESOURCE",

"uriComponentOrder": 2,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "resourceClass",

"uriComponentValue": "resourceClass@RESOURCECLASS",

"uriComponentOrder": 3,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "identifier",

"uriComponentValue": "@ID",

"uriComponentOrder": 4,

"mandatory": true,

"finalCharacter": ""

}

]

"uriPublicationStructure": [

{

"uriComponent": "base",

"uriComponentValue": "base",

"uriComponentOrder": 1,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "character",

"uriComponentValue": "character@RESOURCE",

"uriComponentOrder": 2,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "sector",

"uriComponentValue": "@SECTOR",

"uriComponentOrder": 3,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "resourceClass",

"uriComponentValue": "resourceClass@RESOURCECLASS",

"uriComponentOrder": 4,

"mandatory": true,

"finalCharacter": "/"

}

{

"uriComponent": "identifier",

"uriComponentValue": "@ID",

"uriComponentOrder": 5,

"mandatory": true,

"finalCharacter": ""

}

]

"resourcesClasses": [

{

"resourceClass": "researcher",

"labelResourceClass": "investigador",

"resourceURI": "uriResourceStructure"

},

{

"resourceClass": "publication",

"labelResourceClass": "publicacion",

"resourceURI": "uriPublicationStructure"

}

]

}

]