GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
Unión Europea
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa

Descubrimiento

Álvaro Palacios
RIAM Intelearning Lab – GNOSS
alvaropalacios@gnoss.com

HĒRCULES

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

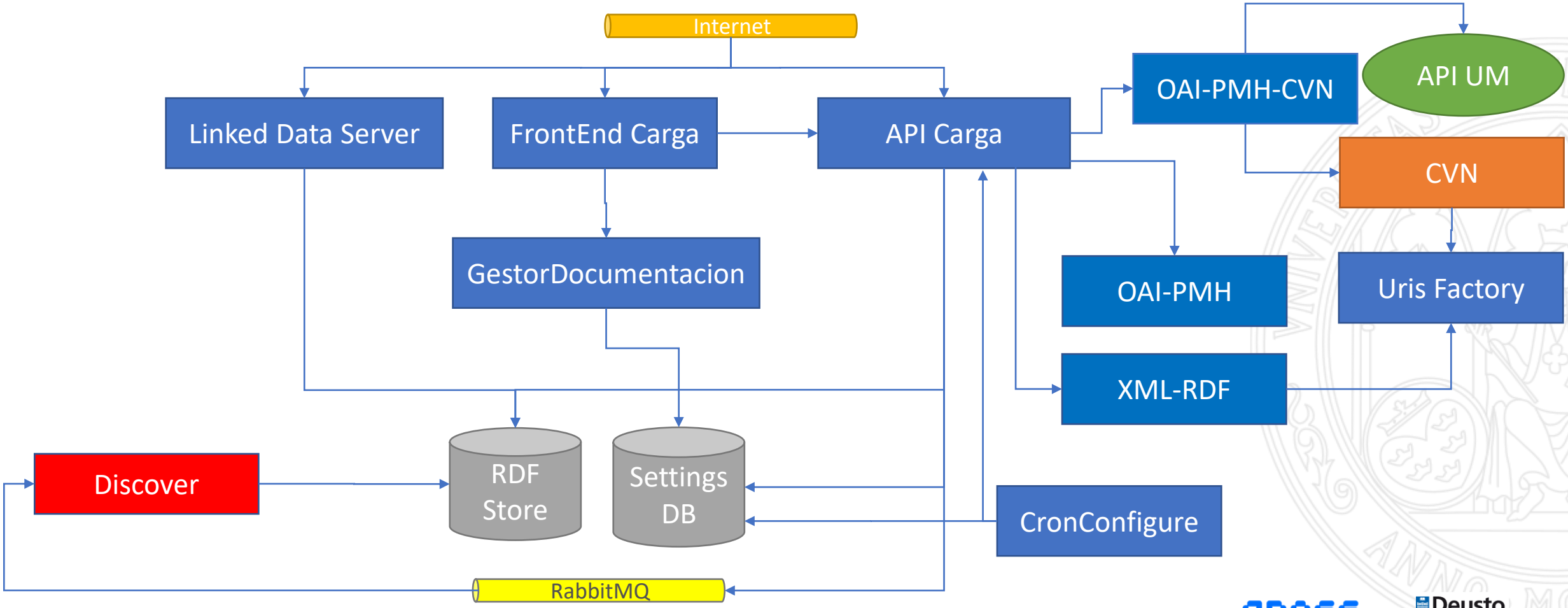
Introducción.

- ❑ Arquitectura ASIO.
- ❑ Descubrimiento en ASIO.
- ❑ Reconciliación de entidades.
- ❑ Descubrimiento de enlaces.
- ❑ Descubrimiento de equivalencias.
- ❑ Casos prácticos.

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Arquitectura ASIO.

Arquitectura ASIO



Arquitectura ASIO

- API Carga: Contiene los procesos ETL (Extract, Transform and Load) necesarios para la carga de datos. Junto con los procesos de gestión de repositorios, validaciones y sincronizaciones.
- CronConfigure: Es un api para la gestión y configuración del programado de tareas, tanto de ejecución recurrente como ejecución única sobre los repositorios configurados.
- OAI-PMH: Servicio recolector de metadatos.
- XML-RDF: Tras generar los XML con OAI-PMH, los transforma en RDF.
- FrontEndCarga: Constituye el interfaz Web de administración de las cargas de datos en la plataforma Hércules ASIO.

Arquitectura ASIO

- Gestor Documentación: Permite publicar páginas webs que informen del nodo ASIO de la universidad, con contenido estático y dinámico, obtenido éste último del API de consulta o del SPARQL Endpoint
- Identity Server: Encargado de la securización mediante tokens para los APIs que forman el proyecto.
- URIs Factory: Es el encargado de generar las URIs de todas las entidades existentes en ASIO.
- LinkedDataServer: Proporciona el servicio de datos enlazados de Hércules ASIO, cumpliendo la recomendación Linked Data Platform.
- Discover: Servicio encargado del descubrimiento: reconciliación de entidades, descubrimiento de enlaces y detección de equivalencias.

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Descubrimiento en ASIO.

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Descubrimiento en ASIO

Una vez cargados los datos desde el SGI, se ejecutan los siguientes procesos de descubrimiento:

- ❑ Reconciliación.
- ❑ Descubrimiento de enlaces.
- ❑ Detección de equivalencias.

Descubrimiento en ASIO

Una vez cargados los datos desde el SGI, se ejecutan los siguientes procesos de descubrimiento:

- Reconciliación.
  - Evita la duplicación de entidades mediante un conjunto de reglas.
  - Toma decisiones autónomas si la evaluación de las reglas supera un umbral.
  - Solicita la validación del usuario si la evaluación queda en un rango de duda.
  - Utiliza datos obtenidos en el descubrimiento de enlaces, desde fuentes externas (ORCID, DOI, etc.) y desde Unidata.
- Descubrimiento de enlaces.
- Detección de equivalencias.

Descubrimiento en ASIO

Una vez cargados los datos desde el SGI, se ejecutan los siguientes procesos de descubrimiento:

- Reconciliación.
- Descubrimiento de enlaces.
  - Obtención de identificadores.
  - Enriquecimiento con enlaces a fuentes externas y/o Unidata.
  - Información para la reconciliación.
  - Proceso con ejecución continua que no se ejecuta sólo en el proceso de carga.
- Detección de equivalencias.

Descubrimiento en ASIO

Una vez cargados los datos desde el SGI, se ejecutan los siguientes procesos de descubrimiento:

- ❑ Reconciliación.
- ❑ Descubrimiento de enlaces.
- ❑ Detección de equivalencias.
  - ❑ Obtención de enlaces a entidades de otros nodos ASIO.
  - ❑ Uso del nodo Unidata.
  - ❑ Información para la reconciliación.

Descubrimiento en ASIO

En resumen, las funciones de el API Descubrimiento, que son parte del proceso de carga, se dividen en 3 grupos:

- Reconciliación de entidades. Evita la duplicación de entidades, detecta las entidades ya cargadas en el nodo ASIO y evita que se carguen entidades que ya están cargadas con otra URI.
- Descubrimiento de enlaces. Genera enlaces hacia datasets externos (incluidos los de otros datasets ASIO a través del nodo Unidata), puede incorporar datos en ASIO y ofrece información de ayuda en la reconciliación de entidades.
- Detección de equivalencias. Detecta equivalencias con los datos cargados en el nodo Unidata.

Los 3 grupos de funciones actúan en el proceso de descubrimiento para todos los datos a cargar en ASIO.

Descubrimiento en ASIO

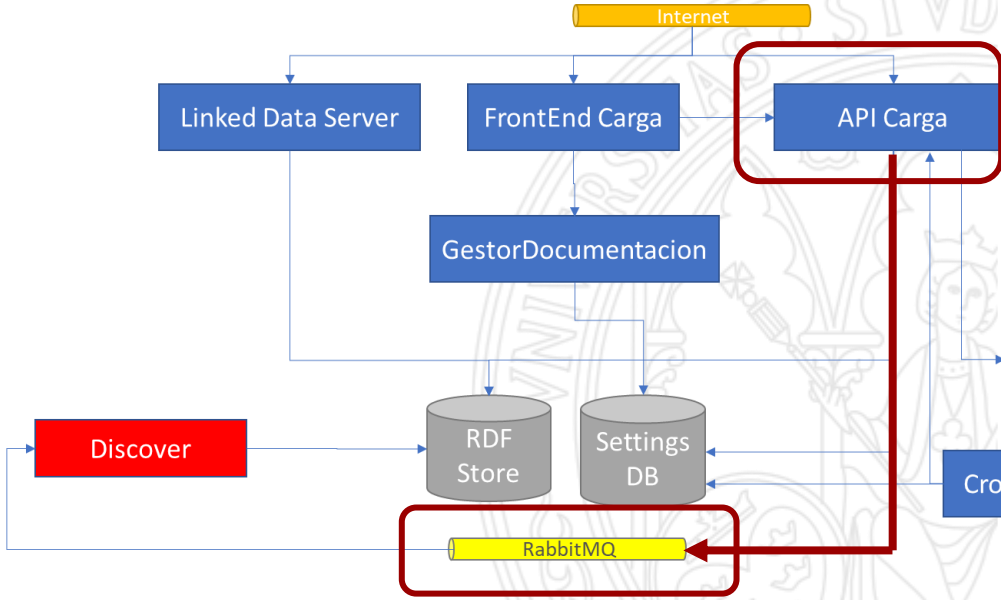
El servicio de descubrimiento tiene dos procesos diferenciados:

- Proceso general de descubrimiento. Este proceso se ejecuta cada vez que se carga un RDF en el sistema.
- Enriquecimiento continuo. Este proceso está ejecutándose continuamente aplicando el descubrimiento de enlaces a los datos que ya están cargados en el sistema.

Descubrimiento en ASIO. Proceso general de descubrimiento.

Este proceso se ejecuta cada vez que se carga un RDF a través del API de Carga con los datos provenientes de un repositorio OAI-PMH transformados a RDF con el servicio XML-RDF.

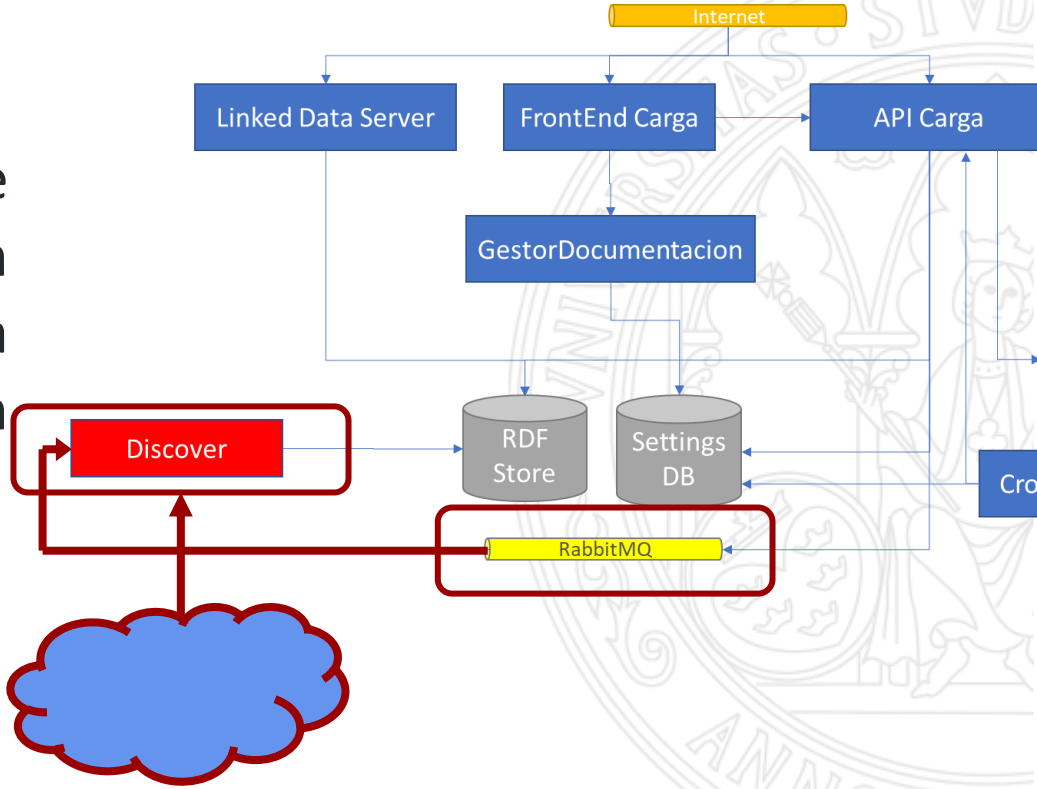
El API de Carga añade los RDFs a una cola de RabbitMQ.



Descubrimiento en ASIO. Proceso general de descubrimiento.

El Servicio de Descubrimiento procesa cada RDF transformado, leído desde la cola de Rabbit.

El Servicio trabaja con cada RDF en memoria y le aplica los tres procesos de forma iterativa, que van enriqueciendo/modificando el RDF, hasta que llega una iteración en la que no se detecta ningún enriquecimiento/modificación adicional.



Descubrimiento en ASIO. Proceso general de descubrimiento.

Como resultado de la aplicación del proceso de descubrimiento pueden suceder dos cosas:

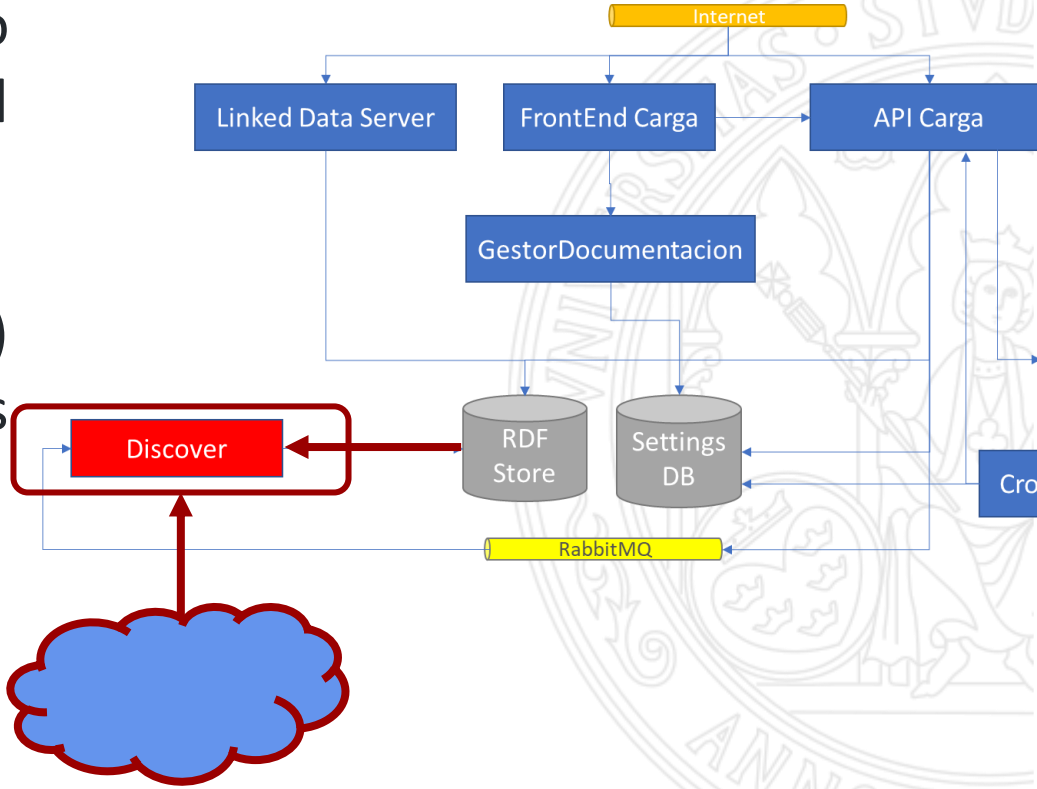
- ❑ Si no hay ningún problema de desambiguación con el proceso de reconciliación se envía el RDF al RDF Store.
- ❑ Si hay algún problema de desambiguación el RDF no se publica y se queda a la espera de que se resuelva el problema de desambiguación de forma manual.

Ejemplo de problema de desambiguación:
https://herc-as-front-desa.atica.um.es/carga-web/Job/274?repository_id=14e66ce5-7bbf-44f4-8ad4-e4019f34edad

Descubrimiento en ASIO. Enriquecimiento continuo.

El enriquecimiento continuo se ejecuta con una periodicidad establecida, aplicando el descubrimiento de enlaces a los datos que ya están cargados en el grafo.

El objetivo es detectar y añadir enlaces (descubrir) que hayan sido introducidos en los sistemas externos y en Unidata posteriormente a su alta en el grafo.



GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Reconciliación de entidades.

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Reconciliación de entidades.

El proceso de reconciliación tiene como entrada un RDF generado a partir de un XML proveniente de la sincronización con un repositorio OAI-PMH.

El proceso de reconciliación de entidades estará apoyado por el proceso de descubrimiento de enlaces y de equivalencias.

Para llevara cabo la reconciliación de entidades se utilizan las configuraciones establecidas en el fichero config/reconciliationConfig.json. En el que se especifican las reglas para llevar a cabo la desambiguación entre entidades.

Reconciliación de entidades. Configuraciones para la reconciliación

En el fichero config/reconciliationConfig.json se configuran las reglas por tipo de entidad (rdf:type) y se tiene en cuenta la herencia de clases.

Cada configuración tiene 3 propiedades:

1. rdfType: Clase a la que afecta la configuración.
2. identifiers: Propiedades utilizadas como identificador.
3. properties: Propiedades para detectar el grado de igualdad de las entidades.

```
{
        "rdfType": "http://purl.org/roh/mirror/foaf#Person",
        "identifiers": [
                "http://purl.org/roh/mirror/vivo#identifier",
                …
        ],
        "properties": [
        {
                …
        },
        {
                …
        }
}
```

Reconciliación de entidades. Configuraciones para la reconciliación

Dentro de cada uno de los elementos de la propiedad ‘properties’ se establecen varios parámetros para detectar el grado de igualdad de las entidades:

1. property: Propiedad a tener en cuanta en al reconciliación ('@@@' implica un 'salto' y '?' implica que puede ser cualquier propiedad)
2. mandatory: Indica si el cumplimineto de esa propiedad es condición necesaria para considerar a dos entidades la misma
3. inverse: Si vale 'false' se buscan los valores de las propiedades utilizando la entidad como sujeto, si vale 'true' se buscan los valores de las propiedades utilizando la entidad como objeto

Reconciliación de entidades. Configuraciones para la reconciliación

4. type: Es el tipo de igualdad que se debe cumplir
   - 0 (equals): Misma entidad o mismo valor de la propiedad
   - 1 (ignoreCaseSensitive): Mismo valor de la propiedad (ignorando mayúsculas y minúsculas)
   - 2(name): Uso del algoritmo para nombres (para nombres de personas)
   - 3(title): Uso del algoritmo para títulos (para títulos de documentos por ejemplo)

5. maxNumWordsTitle: En el caso de que type tenga como valor '3' implica el número de palabras que debe tener el título para considerar el máximo valor de igualdad
6. scorePositive: Score positivo que se da a la relación cuando se da una coincidencia.
7. scoreNegative: Score negativo que se da a la relación cuando no se da una coincidencia y ambas entidad tienen algún valor para esa propiedad.

Reconciliación de entidades. Configuraciones para la reconciliación

A continuación se muestra un fragmento para la configuración para las entidades de tipo 'http://purl.org/roh/mirror/foaf#Person'

```
{
    "rdfType": "http://purl.org/roh/mirror/foaf#Person", //Entidad a la que afecta
    "identifiers": [//Listado de propiedades que actúan como identificadores
      "http://purl.org/roh/mirror/vivo#identifier",
      "http://purl.org/roh/mirror/vivo#eRACommonsId",
      "http://purl.org/roh/mirror/vivo#researcherId",
      "http://purl.org/roh#ORCID"
    ],
    "properties": [
      {
        "property": "http://purl.org/roh/mirror/foaf#name", //Propiedad
        "mandatory": true,            //Obligatoria
        "inverse": false,             //Propiedad directa
        "type": 2,                    //Algoritmo de nombres
        "scorePositive": 0.89,        //Score positivo
        "scoreNegative": null         //Score negativo
      },
```

Reconciliación de entidades. Configuraciones para la reconciliación

```
    {
      "property": "http://purl.org/roh#participates", //Propiedad
      "mandatory": false,      //No obligatoria
      "inverse": false,                  //Propiedad directa
      "type": 0,                         //Igualdad por valor
      "scorePositive": 0.5,    //Score positivo
      "scoreNegative": null    //Score negativo
    },
    {
      "property": "http://purl.org/roh/mirror/foaf#mbox", //Propiedad
      "mandatory": false,                //No obligatoria
      "inverse": false,                  //Propiedad directa
      "type": 0,                         //Igualdad por valor
      "scorePositive": 0.9,    //Score positivo
      "scoreNegative": 0.025   //Score negativo
    },
    {
      "property": "http://purl.org/roh/mirror/bibo#authorList@@@?", //Propiedad con salto y con variable
      "mandatory": false,                //No obligatoria
      "inverse": true,                   //Propiedad inversa
      "type": 0,                         //Igualdad por valor
      "scorePositive": 0.9,    //Score positivo
      "scoreNegative": null    //Score negativo
    }
  ]
}
```

Reconciliación de entidades. Flujo

1. Se lee el RDF y se obtienen todas las entidades para realizar la reconciliación.

2. Para cada una de las entidades se hace una consulta al grafo RDF del RDF Store para obtener posibles candidatos para la reconciliación con las propiedades que estén configuradas para llevar a cabo la desambiguación.
    1. Se busca si existe alguna entidad con la misma URI.
    2. Si no se ha encontrado la entidad y la entidad cuenta con algún identificador, se intenta reconciliar la entidad a través de alguno de sus identificadores. Si existe alguna entidad cargada en el nodo ASIO que comparta identificador se considera la misma entidad.
    3. Si no se ha encontrado la entidad, se buscan similitudes con entidades ya cargadas. Para cada tipo de entidad utilizan las reglas establecidas en el fichero reconciliationConfig.json.

Reconciliación de entidades. Flujo

3. Una vez obtenidos todos los candidatos se aplican las reglas de cálculo de reconciliación para obtener las entidades finales ya cargadas. Este punto además es apoyado por el descubrimiento de enlaces

4. En función del resultado obtenido se realiza una de las siguientes acciones:
   - Si para alguna entidad hay más de un candidato que supere el umbral máximo o el umbral mínimo, se agregará el RDF a una BBDD junto con todos los datos necesarios para una revisión manual.
   - Si para alguna entidad sólo se obtiene un candidato que supere el umbral máximo, se modificará la URL de la entidad en el RDF a cargar por la URL de la entidad encontrada.
   - Se obtienen las entidades principales del RDF y se eliminan todos los triples que haya en el grafo RDF en los que aparezcan como sujeto u objeto.
   - Se eliminan todos los triples de las entidades cuyo sujeto y predicado estén en el RDF a cargar y estén marcados como monovaluados según la especificación de la ontología.
   - Se vuelcan los triples al grafo RDF

Reconciliación de entidades. Algoritmo de nombres

• Se ha optado por una medida basada en conjuntos de caracteres, usando n-gramas y obteniendo el coeficiente de Jaccard. Los aspectos a considerar son:

- Reordenar la cadena de nombre + apellidos si aparece una coma. Por ejemplo, “Pérez Lara, Ángel” a “Ángel Pérez Lara”.
- Dividir el nombre y apellidos en sus palabras, retirando stop words (de, del, la) y guiones.
- Considerar la puntuación de las palabras con un coeficiente de Jaccard por encima de 0,5. Si no se supera, el índice resultante sería 0.
- Otorgar un peso fijo de 0,5 al reconocimiento de una inicial (“Eduardo” y “E.”).
- La puntuación de una palabra será 0 si no aparece en el orden adecuado.

Reconciliación de entidades. Algoritmo de nombres

Para "Ángel Pérez Lara" podríamos obtener los siguientes candidatos:

Superan el corte de 0,5:

- Ángela Pérez Lara: 0,90
- A. Pérez Lara: 0,83
- Miguel Pérez Lara: 0,67
- Ángel Pérez Rodríguez: 0,67
- Miguel Ángel Pérez Lara: 0,625
- Ángel Pedro Pérez Laras: 0,58

NO superan el corte de 0,5:

- Ángel Pedro Pérez Talavera: 0,46
- Ángel Pedro Pérez Calatayud: 0,46
- Ángel Yoset Lara Pérez: 0,42

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Descubrimiento de enlaces.

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Descubrimiento de enlaces

Se buscan en los diferente APIs las coincidencias de publicaciones +autores.

El nombre del autor o de la publicación, por sí solos, no son suficientes para considerarlos la misma entidad. Se utilizan en el caso de que exista una relación entre los nombres de ambas entidades.

Hay diferencias en lo que se puede hacer con cada una de las APIs externas, por lo que no todas se usan del mismo modo.

Descubrimiento de enlaces. Procedencia

Los datos de fuentes externas se cargan junto con unos triples que indican su procedencia. El triple con el dato obtenido desde una fuente externa se carga en un grafo que indica su procedencia. Por ejemplo, un código ORCID recuperado desde DBLP:

Grafo: http://graph.um.es/graph/sgi

```
roh:res/researcher/id1 roh:ORCID "00000".

roh:res/agent/idAgente1
    a prov:SoftwareAgent;
    foaf:name "Algoritmo Hércules ASIO";
```

Grafo: http://graph.um.es/graph/dblp

```
roh:res/researcher/id1 prov:wasUsedBy _:bnode1.
_:bnode1
    a prov:Activity;
    rdf:predicate roh:ORCID;
    rdf:object "00000";
    prov:startedAtTime "2020-04-25T01:30:00Z"^^xsd:dateTime;
    prov:endedAtTime "2012-04-25T03:40:00Z"^^xsd:dateTime;
    prov:wasAssociatedWith roh:res/agent/idAgente1;
    prov:wasAssociatedWith roh:res/organization/dblp;
```

Descubrimiento de enlaces. Fuentes externas

Crossref (https://www.crossref.org/)

Crossref makes research outputs easy to find, cite, link, assess, and reuse. A not-for-profit membership organization that exists to make scholarly communications better.

Dentro de discover se hacen llamadas al método del API 'https://api.crossref.org/works?query.author={nombre_autor}&rows=200' y se obtienen las publicaciones que tienen como autor el parámetro pasado.

Descubrimiento de enlaces. Fuentes externas

DBLP Computer Science Bibliography (https://dblp.org/)

The dblp computer science bibliography is the on-line reference for bibliographic information on major computer science publications. It has evolved from an early small experimental web server to a popular open-data service for the whole computer science community.

Dentro de discover se hacen llamadas al método del API 'https://dblp.org/search/author/api?q={nombre_autor}&h=5' y se obtienen los autores junto con sus publicaciones, a continuación, se llama al método 'https://dblp.org/pid/{id_autor}' con los identificadores de DBLP de los autores para obtener más metadatos.

Descubrimiento de enlaces. Fuentes externas

DOAJ (https://doaj.org/)

The DOAJ (Directory of Open Access Journals) was launched in 2003 with 300 open access journals. Today, this independent database contains over 15 000 peer-reviewed open access journals covering all areas of science, technology, medicine, social sciences, arts and humanities. Open access journals from all countries and in all languages are welcome to apply for inclusion.

Dentro de discover se hacen llamadas a los métodos del API 'https://doaj.org/api/v2/search/articles/title:{nombre_documento}' y 'https://doaj.org/api/v2/search/journals/title:{nombre_documento}' y se obtienen las publicaciones junto con sus autores.

Descubrimiento de enlaces. Fuentes externas

ORCID (https://orcid.org/)

ORCID is a nonprofit organization helping create a world in which all who participate in research, scholarship and innovation are uniquely identified and connected to their contributions and affiliations, across disciplines, borders, and time.

Dentro de discover se hacen llamadas al método del API 'https://pub.orcid.org/v3.0/expanded-search?q={nombre_autor}&rows=5' para obtener los identificadores de los autores y posteriormente se hacen llamadas a los métodos del API 'https://pub.orcid.org/v3.0/{id_autor}/person' y 'https://pub.orcid.org/v3.0/{id_autor}/works para obtener metadatos de los autores y sus obras respectivamente.

Descubrimiento de enlaces. Fuentes externas

PubMed (https://pubmed.ncbi.nlm.nih.gov/)

PubMed® comprises more than 30 million citations for biomedical literature from MEDLINE, life science journals, and online books.

Dentro de discover se hacen llamadas al método del API 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?term={nombre_documento}&field=title&sort=relevance&retmax=10' con el que obtenemo los IDs de los documentos y posteriormente se hacen llamadas al método del API 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed&id={id_documento}&retmode=xml' con los identificadores de los documentos y obtenemos los documetnos junto con sus autores.

Descubrimiento de enlaces. Fuentes externas

Recolecta (https://recolecta.fecyt.es/)

RECOLECTA, o Recolector de Ciencia Abierta, es el agregador nacional de repositorios de acceso abierto. En esta plataforma se agrupan a todas las infraestructuras digitales españolas en las que se publican y/o depositan resultados de investigación en acceso abierto.

Dentro de discover se hacen llamadas al método del API 'https://buscador.recolecta.fecyt.es/buscador-recolecta?search_api_fulltext={nombre_documento}' y se obtienen las publicaciones junto con sus autores.

Descubrimiento de enlaces. Fuentes externas

Scopus (https://www.scopus.com/)

Scopus is the largest abstract and citation database of peer-reviewed literature: scientific journals, books and conference proceedings. Delivering a comprehensive overview of the world's research output in the fields of science, technology, medicine, social sciences, and arts and humanities, Scopus features smart tools to track, analyze and visualize research.

Dentro de discover se hacen llamadas al método del API 'https://api.elsevier.com/content/search/scopus?query=TITLE({nombre_documento})&view=COMPLETE&apiKey={ScopusApiKey}&httpAccept=application/xml' y se obtienen las publicaciones junto con sus autores, posteriormente se llama al método del API 'https://api.elsevier.com/content/author/author_id/{id_autor}?apiKey={ScopusApiKey} para obtener metadatos de los autores.

Descubrimiento de enlaces. Fuentes externas

Web of Science (http://wos.fecyt.es/)

FECYT provides access to Web of Science, the world's largest publisher-neutral citation index and research intelligence platform.

Dentro de discover se hacen llamadas al método del API 'http://search.webofknowledge.com/esti/wokmws/ws/WokSearch' con los nombres de las publicaciones y se obtienen las publicaciones junto con sus autores.

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Descubrimiento de equivalencias.

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Detección de equivalencias

Este proceso utiliza la información del nodo Unidata para detectar equivalencias de entidades entre el nodo del SGI y Unidata.

Detectara las equivalencias de entidades y añadirá los triples ‘SameAs’ a las entidades para que estén relacionadas entre los nodos SGI y Unidata.

Enriquece el nodo del SGI pudiendo obtener identificadores de Unidata

Apoya el proceso de reconciliación tras enriquecer los datos de las entidades

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Una manera de hacer Europa
Unión Europea

Hércules ASIO. Casos prácticos.

gnoss
Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Casos de uso.

- ❑ **Discover. Ejercicio 0. Manipulación de grafos en memoria**
- ❑ **Discover. Ejercicio 1. PrepareData**
- ❑ **Discover. Ejercicio 2. Reconciliación**
- ❑ **Discover. Ejercicio 3. Descubrimiento de enlaces**
- ❑ **Discover. Ejercicio 4. Implementación del API de ORCID**

Casos de uso. Introducción

Para estos ejercicios usaremos el servicio API Discover, pero en lugar de conectar con los otros servicios y la BBDD usaremos MOCKs para simular la interacción con los otros sistemas.

En lugar de ejecutar el programa de forma ‘normal’, creamos ejercicios dentro de los test unitarios.

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
UNIVERSIDAD DE MURCIA
FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)
Unión Europea
Una manera de hacer Europa

Casos de uso. Ejercicio 0. Manipulación de grafos en memoria

Antes de comenzar con los ejercicios del servicio de descubrimiento veremos como manipular grafos en memoria, que es el tipo de objeto con el que trabajaremos en el servicio.

```
public void Ejercicio_0_GraphInMemory()
```

Casos de uso. Ejercicio 0.1. Cargar datos de usuario

Añadir a dataGraph los triples necesarios para cargar una entidad de tipo persona con tu nombre y 2 documentos en los que figures en su lista de autores

Casos de uso. Ejercicio 1. PrepareData

En este ejercicio veremos el funcionamiento del método 'PrepareData', utilizado de forma frecuente dentro del servicio de descubrimiento.

```
public void Ejercicio_1_TestPrepareData()
```

Casos de uso. Ejercicio 2. Reconciliación

En este ejercicio simulamos el proceso de reconciliación, para ello intentamos reconciliar el fichero rdfFile.rdf con los datos cargado en la BBDD, en este caso utilizaremos un MOCK como BBDD con el fichero rdfFileDatabase.rdf

```
public void Ejercicio_2_Reconciliation()
```

Casos de uso. Ejercicio 3. Descubrimiento de enlaces

En este ejercicio simulamos el proceso de descubrimiento de enlaces. En el código hay varios APIs externos comentados porque requieren de autorización.

```
public void Ejercicio_3_TestDiscoverLinks()
```

Casos de uso. Ejercicio 4. Implementación del API de ORCID

En este ejercicio implementaremos la interacción con el API de ORCID.

Para ello dentro del fichero DiscoverUtility.cs hay que descomentar las líneas en la región ***Descomentar en el ejercicio 'Ejercicio_3_TestDiscoverLinksORCID'*** y comentar las de la región ***Comentar en el ejercicio 'Ejercicio_3_TestDiscoverLinksORCID'***

```
public void Ejercicio_4_TestDiscoverLinksORCID()
```

Casos de uso. Ejercicio 4.1. Consulta para cargar personas y documentos

Dentro de DiscoverUtility.cs localizar la región ‘Ejercicio 4.1’ y modificar la query para obtener todos los documentos de cada persona

Casos de uso. Ejercicio 4.2. Añadir triples de persona

Dentro de DiscoverUtility.cs localizar la región ‘Ejercicio 4.2’ y añadir los triples de la persona en ‘orcidGraph’

Casos de uso. Ejercicio 4.2. Añadir triples de documentos

Dentro de DiscoverUtility.cs localizar la región ‘Ejercicio 4.2’ y añadir los triples de los documentos en ‘orcidGraph’

¡GRACIAS!