Proyecto LILACS con datos enlazados

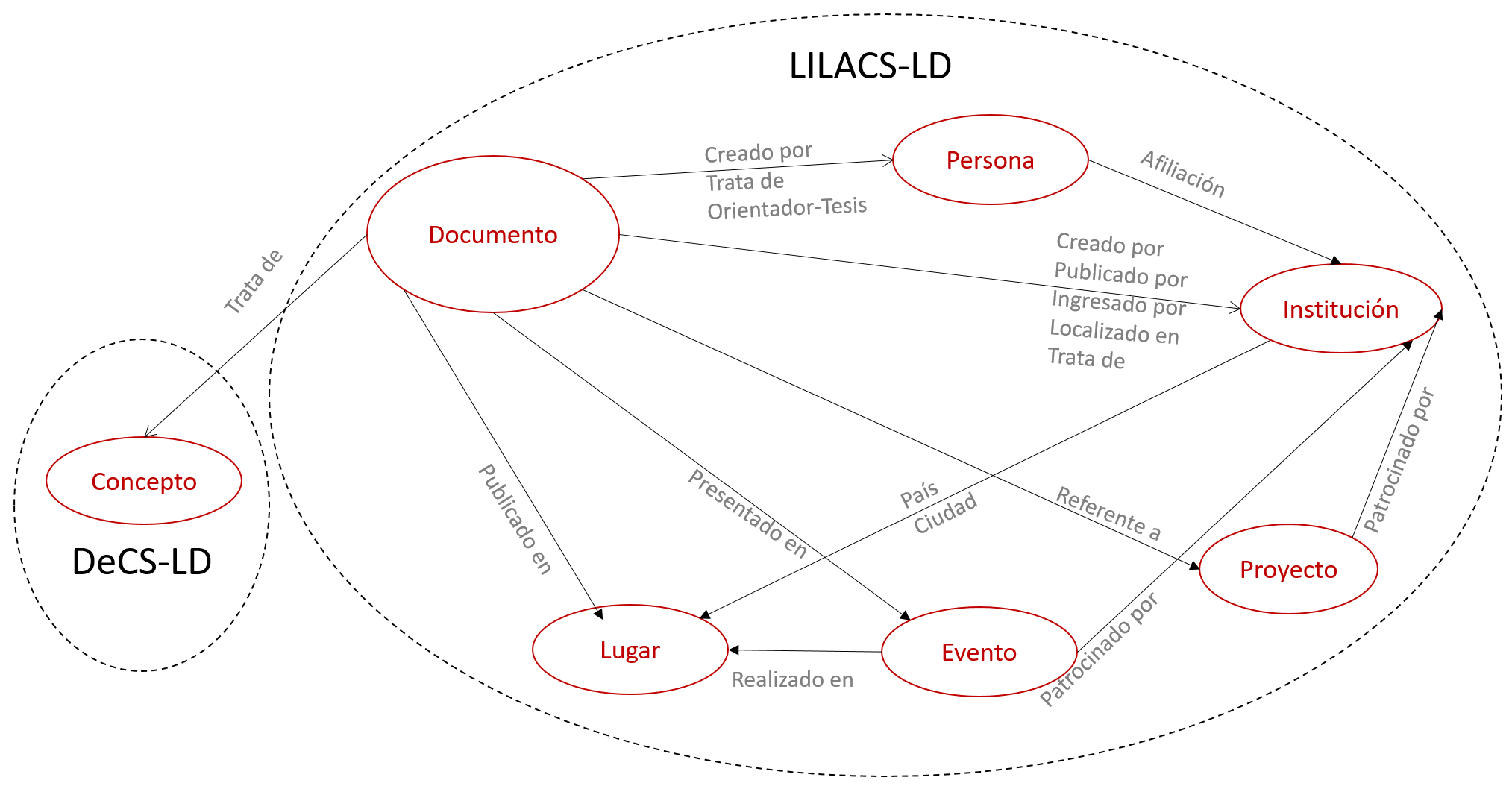
Objetivos:

* Crear el nuevo modelo de datos de LILACS basado en los *estándares de la Web Semántica*; para publicar sus datos como *datos enlazados*, de modo que puedan ser recolectados y procesados por aplicaciones, fomentando su conexión con otros conjuntos de datos, aumentando así su presencia y relevancia.

Implementacion:

1. Definicion del nuevo modelo de datos con estandares de la web semántica (LILACS-LD)

* El nuevo modelo de datos LILACS se representa usando **RDF**. Con **schema.org** como espacio de nombres base, que se extiende con espacio de nombres **lilacs**, donde se definen las propiedades específicas. Los datos RDF se serializan con **JSON-LD**.
* Entidades y relaciones en LILACS-LD



1. Plataforma de trabajo
   * **MySQL 5.7.8+:** incluye el tipo de datos JSON, garantiza la validación y el acceso eficiente a los elementos del JSON. Cuenta con funciones para: crear, modificar, buscar y obtener atributos de documentos JSON
   * **PHP 5.6.19**+ soporta el tipo de dato JSON
2. Migración LILACS

* Se utiliza el servicio **isis-oai-provider** para obtener los registros originales.
* Los campos de registros originales se traducen a propiedades del nuevo modelo. Durante la migración de los documentos se detectan otras entidades: Personas, Organizaciones, Eventos, Proyectos y Conceptos (DeCS).
* A los documentos y al resto de las entidades (después de validadas) se les asigna su URI a partir de la URL de la aplicación y su identificador unico. Los Conceptos son entidades del DeCS, que obtienen su URI al migrar el DeCS al modelo de datos enlazados.
* Se detectan y asignan los datos enlazados. Por ejemplo en un documento a la propiedad author se le asigna su URI (*"@id":"http://lilacs.sld.cu/entity/persons/1"*).
* Ejemplo de documento migrado (document.json)

1. DeCS como datos enlazados

* Definicion del nuevo modelo de datos
  + Teniendo en cuenta que el DeCS es una extension del MeSH y que existe **MeSH-RDF**, se utiliza este mismo modelo de datos para DeCS-LD. Se serializan los datos con **JSON-LD**.
* Plataforma de trabajo
  + **MySQL 5.7.8+**
  + **PHP 5.6.19**+
* Migracion:

1. Categorias comunes de MeSH y DeCS
   * Se descargan los archivos de MeSH RDF
   * Se guardan en la BD serializados con JSON-LD. Se le asignan URIs a partir de la URL de la aplicación y su identificador único en MeSH, la URI de MeSH se asigna a sameAs.

Ej: *"@id":"http://decs.sld.cu/Descriptors/D000001*

*"sameAs":"http://id.nlm.nih.gov/mesh/D000001"*

* + Se agregan las traducciones al espanol y portugués de Descriptores y Calificadores consultando al servicio del DeCS (<http://decs.bvsalud.org/cgi-bin/mx/cgi=@vmx/decs/>").
  + Ejemplo de descriptor migrado de MeSH con textos traducidos de decs: descriptor.json

1. Categorias propias de DeCS (SH, HP, SP y VS)
   * Se recorre el árbol correspondiente con el servicio del DeCS, para obtener los descriptores y calificadores en los 3 idiomas.
   * Se representan en el nuevo modelo los Descriptores y Calificadores, asi como entidades relacionadas: Terminos, Conceptos, Numeros de árbol. Se les asignan sus URIs a partir de la URL de la aplicación y un identificador único, similar al de MeSH (con ‘\_’ delante)

Ej: *"@id":"http://decs.sld.cu/Descriptors/\_D000001*

* + Ejemplo de descriptor migrado de DeCS: descriptorDeCS.json

1. WebService RESTful para consultar LILACS-LD
   * Permite realizar consultas de Creacion, Selección, Actualizacion y Eliminacion (CRUD) a cualquier tipo de entidad de LILACS-LD (Documento, Persona, Organización, Evento, Proyecto).
   * Peticiones a través de url con los verbos GET, POST, PUT, UPDATE y DELETE:
     + GET <http://lilas.sld.cu/entity/documents/23>
     + GET <http://lilas.sld.cu/entity/organizations>
     + POST http://lilas.sld.cu/entity/documents?data=
     + UPDATE http://lilas.sld.cu/entity/persons/123?data=
     + DELETE http://lilas.sld.cu/entity/persons/123
   * Respuestas en formato JSON
   * Plataforma: **MySQL 5.7.8+** y **PHP 5.6.19**+
   * Estado: En progreso (creación de API para document)
2. WebService RESTful para consultar DECS-LD