

## 1.1. Almacén RDF

El almacén RDF se obtiene a partir de los datos almacenados en la base de datos geoespacial. Para ello se genera un almacén de tipo TDB que contiene toda la información almacenada en la base de datos en formato RDF y posteriormente se carga en el servicio de SPARQL, para su uso en la parte de web semántica del sistema. TDB es un componente de Jena para el almacenamiento y consulta de datos RDF. Soporta todas las operaciones contenidas en la API de Jena. TDB se utiliza como un almacén de RDF de alto rendimiento. TDB está protegido contra la corrupción, las terminaciones de procesos inesperados y los fallos del sistema. Para que TDB soporte concurrencia se debe acceder mediante el uso de un servicio de SPARQL. Los recursos almacenados son los mismos que se encuentran en la base de datos geoespacial del sistema, ya que es a partir de estos de los que se genera la información que contiene este fichero.

## 1.2. Creación del almacén RDF

El almacén RDF contiene la información de los recursos almacenados en la base de datos geoespacial. Por este motivo se ha creado una aplicación Java que permite obtener los recursos de la base de datos y generar a partir de estos un fichero en formato RDF, que posteriormente permite la creación de un TDB<sup>1</sup>, que es el almacén de datos RDF del sistema. TDB es un componente de Jena para el almacenamiento y consulta de datos RDF. Soporta todas las operaciones contenidas en la API de Jena. TDB se utiliza como un almacén de RDF de alto rendimiento. TDB está protegido contra la corrupción, las terminaciones de procesos inesperados y los fallos del sistema. Para que TDB soporte concurrencia se debe acceder mediante el uso de un servicio de SPARQL. Para el desarrollo de esta aplicación Java se ha hecho uso del framework Jena, que proporciona librerías que facilitan la adaptación de la información de la base de datos al formato RDF.

### 1.2.1. Estructura del almacén RDF

La creación de la estructura que almacena los datos en formato RDF, se ha hecho siguiendo los estándares oficiales, y haciendo uso de los vocabularios ya existentes, que se adaptan a la naturaleza de los datos con los que trabaja la aplicación.

El modelo resultante en el que se estructuran los datos del sistema, hace uso de tres vocabularios: GeoSPARQL, SKOS-XL (Simple Knowledge Organization System) y Dublin Core.

**GeoSPARQL** es un estándar para la representación y consulta de Linked Data geoespaciales para la Web Semántica del Open Geospatial Consortium (OGC). Tiene por objeto proporcionar una base de intercambio estandarizado para datos geoespaciales RDF que puede apoyar tanto el razonamiento espacial cualitativo y cuantitativo y consultar con el lenguaje de consulta de base de datos SPARQL.

**SKOS-XL** proporciona un modelo para representar la estructura básica y el contenido de esquemas conceptuales como listas encabezamientos de materia, taxonomías, esquemas de clasificación, tesauros y cualquier tipo de vocabulario controlado.

**Dublin Core** es un sistema de 15 definiciones semánticas descriptivas que pretenden transmitir un significado semántico a las mismas, proporcionando un vocabulario de características "base", capaces de proporcionar la información descriptiva básica sobre cualquier recurso, sin que importe el formato de origen, el área de especialización o el origen cultural.

La estructura genérica del modelo es la siguiente:

```
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix wkt:    <http://www.opengis.net/ont/geosparql#> .
@prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .

uri recurso
  geo:geometry
  [
    wkt:asWKT "POINT(lon,lat)"
  ]
  skosxl:labelrelation
  [
    skosxl:preferLabel
    [
      skosxl:literalForm "nombre"
      dc:source "fuente"
      dc:language "idioma"
    ]
    skosxl:altLabel
    [
      skosxl:literalForm "nombre"
      dc:source "fuente"
      dc:language "idioma"
    ]
    skosxl:hiddelabel
    [
```

```
        skosxl:literalForm "nombre"  
        dc:source "fuente"  
        dc:language "idioma"  
    ]  
]
```

En este modelo se puede observar al principio la definición de los prefijos con los que se va a declarar cada vocabulario, en el caso de Dublin Core no es necesario, ya que reconoce implícitamente su prefijo como DC.

Después de la definición de los prefijos se encuentra, la URI del recurso, y a continuación la declaración de las propiedades que puede tener asociado un recurso.

- *geo:geometry* especifica el punto de la localización del recurso en el mapa.
- *skosxl:preferLabel* especifica el nombre, fuente e idioma oficiales del recurso.
- *skosxl:altLabel* especifica el nombre, fuente e idioma alternativos del recurso.
- *skosxl:hiddelabel* especifica el nombre, fuente e idioma variantes del recurso.

En el caso de las propiedades *geo:geometry* y *skosxl:preferLabel* deben tener valor obligatoriamente, ya que son las propiedades mínimas que se pueden tener sobre un recurso.

Las propiedades *skosxl:altLabel* y *skosxl:hiddelabel* son multivaluadas, es decir, pueden repetirse, dado que un recurso puede tener varios nombres alternativos/variantes, y de esta forma se permite almacenar todos los valores, así como en caso de que el recurso no tenga valores asociados para estas propiedades, el recurso almacenado no tendrá esas propiedades asociadas.

Se puede encontrar la documentación completa del proyecto en <https://github.com/IAAA-Lab/LinkedMapService/blob/master/Memoria.pdf>