Implementación de un buscador semántico de documentos en una plataforma de gestión documental con soporte para firma digital

María Laura Caliusco[1], Agustín Martínez[2] and Graciela Brusa[2]

[1] CIDISI-UTN – Facultad Regional Santa Fe, 3000, Santa Fe, Argentina
[2] Lyris IT SAS, 3000, Santa Fe, Argentina
mcaliusco@frsf.utn.edu.ar, mrtnz.agustin@gmail.com, gbrusa@lyris.com.ar

Resumen. La búsqueda semántica de documentos es un tema ampliamente estudiado en el ámbito académico. Sin embargo, la implementación de estas tecnologías en la industria del software todavía sigue siendo un desafío. Mucho de ello se debe a que las tecnologías semánticas recién ahora están lo suficientemente maduras para lograr implementaciones escalables. El objetivo del presente trabajo es mostrar la incorporación de tecnologías semánticas para realizar búsquedas de documentos en la plataforma Signar de la empresa Lyris IT S.A.S. Este trabajo muestra cómo investigaciones que se llevan a cabo en una Universidad aportan valor agregado a empresas para potenciar sus productos y hacerlos más competitivos en el mercado internacional, incorporando tecnologías innovadoras.

Palabras claves: gestión documental, tecnologías semánticas

Implementation of a semantic document search engine on a document management platform with support for digital signature

Abstract. The semantic search of documents is a subject widely studied in the academic field. However, the implementation of these technologies in the software industry still remains a challenge. Much of this is because semantic technologies are only now mature enough for reliable implementations. The objective of this work is to show the incorporation of semantic technologies to search for documents in the Signar platform of the company Lyris IT S.A.S. This work shows how research carried out at a University adds value to companies to enhance their products and make them more competitive in the international market, incorporating innovative technologies.

Palabras claves: document management, semantic technologies

Received April 2023 Accepted July 2023; Published July 2023
https://doi.org/10.24215/15146774e026
ISSN 1514-6774

1 Introducción

La pandemia del COVID-19 ha empujado a muchas empresas en el mundo a tener que llevar adelante procesos de digitalización que no tenían pensado (Almeida, 2020). El proceso de digitalización refiere a convertir información en papel a información digital con su correspondiente clasificación y que además esta resulte en búsquedas sencillas y ágiles para la recuperación y manejo de la misma (Vrana). En consecuencia, las empresas se encuentran actualmente con un importante cúmulo de documentos digitales que deben gestionar.

En este sentido, los servicios de software con la posibilidad de gestionar documentos digitales con firma digital se volvieron de suma utilidad. La firma digital es una tecnología que permite firmar documentos electrónicos con las mismas propiedades que tiene un documento firmado en papel. Sus principales ventajas son la no presencialidad, la reducción de tiempos y costos vinculados al transporte de la documentación y la despapelización efectiva que surge al ser equiparada totalmente a la firma manuscrita por la legislación de diferentes países en la materia.

Lyris IT SAS es una PyME de base tecnológica de la ciudad de Santa Fe especializada en tecnologías de la información y la comunicación (TIC), específicamente, en procesos de gestión documental con valor jurídico. Lyris IT desarrolló una plataforma de software, denominada SIGNAR, que permite la gestión de documentos digitales que fluyen en los diferentes procesos administrativos y de gestión, tanto en organizaciones públicas como privadas, mediante la aplicación de firma digital y firma electrónica.

La plataforma SIGNAR contempla la gestión de certificados de firma electrónica, recibos digitales, procesos digitales y documentos electrónicos, proceso de aplicación y verificación de firmas digitales/electrónicas. Dentro de las funcionalidades provistas por la plataforma SIGNAR se detectó la necesidad de mejorar la funcionalidad de las búsquedas de documentos permitiendo contar con una gestión documental inteligente.

Para cumplir con el objetivo planteado anteriormente, se formuló un proyecto cuyo aspecto diferencial es que cuenta con la capacidad de realizar búsquedas inteligentes de documentos a partir de la anotación semántica de su contenido. En el proceso de anotación semántica se identifican formalmente las relaciones entre conceptos y documentos para enriquecer el contexto de la información (Viltres Sala). Para realizar la anotación semántica se utilizan diferentes técnicas y herramientas (Aguado de Cea). El propósito del presente trabajo es presentar una implementación de un módulo que permite realizar búsquedas semánticas, basadas en ontologías, de documentos en una plataforma de gestión documental con soporte para firma digital.

El trabajo se organiza de la siguiente manera. En la Sección 2 se introducen los conceptos básicos que se necesitan para comprender el trabajo. En la Sección 3 se describe el Análisis Preliminar llevado a cabo. En la Sección 4 se presenta la construcción del Modelo Semántico construido para la plataforma. En la Sección 5 se discute la arquitectura que se diseñó e implementó junto con las herramientas utilizadas. En la Sección 6 se presenta una implementacion resultado de la extensión del proyecto y finalmente en la Sección 7 se discuten las conclusiones y lecciones aprendidas.

2 Background

El objetivo de esta sección es introducir los conceptos necesarios para poder comprender el trabajo en general.

2.1. Plataforma Signar

Signar es una plataforma de software que contempla **módulos** que se distribuyen en modalidad SaaS (Software as a Service) y otros productos **específicos** de firma complementarios, integrables en aplicaciones web de los clientes, incluso en la misma Plataforma Signar.

Los módulos en modalidad SaaS son los siguientes:

- Gestión **de certificados de firma** electrónica. Abarca las funcionalidades que permiten emitir, renovar y revocar certificados digitales de firma electrónica mediante la

tecnología de clave pública. En la práctica se traduce en la constitución de una PKI (Public Key Infrastructure) en la empresa Lyris IT para ofrecer estos servicios a sus clientes, en el marco de la Ley 25.506/01 de firma digital en el República Argentina.

- Gestión de documentos electrónicos. Este módulo permite gestionar en carpetas documentos simples o compuestos que además, están sujetos a acciones tales como subir documentos a la plataforma, firmar digitalmente, mover, borrar, agregar metadatos, etc.

  En el caso particular de los documentos compuestos (conjunto de documentos simples relacionados funcionalmente, tal como un trámite, gestión o expediente), el software permite modelar el proceso que lo controla dando acceso a usuarios o áreas específicas, con determinados permisos y acciones a realizar.

  Este módulo tiene una gran potencialidad porque es totalmente configurable por el usuario de acuerdo a sus necesidades de gestión de los documentos y permite a su vez, mantener un archivo digital con valor legal, ya que aquellos documentos que tengan el requerimiento de escritura y archivado, se pueden mantener con el cumplimiento de esos requisitos en un esquema de trabajo ordenado y con facilidad para las búsquedas.

  El agregado de metadatos actualmente lo realiza el usuario de forma manual y las búsquedas se realizan a través de un buscador que filtra por los metadatos definidos. Se pretende trabajar sobre este aspecto, incorporando nuevas herramientas que fortalezcan y automaticen el agregado de metadatos y dichas búsquedas.
- Gestión **de recibos de sueldo digitales. Este** módulo **se** considera un submódulo de la gestión documental, ya que está orientado a un tipo específico de documentos electrónicos, que son los recibos de pagos a empleados. **Contempla dos bandejas de acceso a los recibos: una para el *empleador* que sube y firma todos los recibos en una** única operación y otra para el *empleado*, que automáticamente dispone de los recibos firmados por el empleador en su bandeja y puede consultar la liquidación previamente, firmar en conformidad o

disconformidad y descargar el recibo firmado digitalmente por su empleador.

Si bien la plataforma cuenta con estos tres módulos en su modalidad de distribución SaaS, nuestra mejora sustancial se encuentra en la gestión de los metadatos y búsquedas más inteligentes que faciliten la experiencia de usuario (UX) a través de una propuesta de software más intuitiva, atractiva y eficiente.

2.2. Anotaciones semánticas

Las anotaciones semánticas son un tipo específico de metadatos cuyo objetivo es permitir nuevos métodos de acceso a la información y extender los que ya existen. En ese sentido, las anotaciones semánticas proporcionan referencias entre las entidades existentes en los recursos y conceptos de un dominio previamente modelado en una ontología.

Las anotaciones textuales están dirigidas a la inserción de palabras clave para su uso por el creador del documento. En cambio, las anotaciones semánticas permiten individualizar los conceptos, y las relaciones que existen entre ellos, en el contenido de un documento con el objetivo de que sean entendibles por humanos, pero sobre todo por las máquinas, de manera que otorgue valor y sea consistente con los esquemas y ontologías que se adopten. Diferentes ontologías representan diferentes conceptualizaciones de conocimiento; por lo tanto, pueden ser utilizadas para recuperar información del mismo documento en función del conocimiento especificado en la ontología.

Para obtener un corpus anotado semánticamente la anotación puede realizarse en forma asistida o automáticamente dejando la actividad en manos de una computadora:

- En la anotación automática propiamente dicha, el objetivo final es asignar a cada elemento del texto una etiqueta correcta y única, eliminando todo rastro de ambigüedad, así como la necesidad de la participación humana en el proceso.

- En la anotación asistida, un anotador o revisor humano es el encargado de determinar la etiqueta final de aquellos elementos que el sistema informático no haya sido capaz de anotar bien o unívocamente, aún después de ejecutar las rutinas de desambiguación adecuadas.

El enfoque más utilizado para realizar anotaciones semánticas automáticas consiste en emplear una ontología previamente creada; de esta forma, la creación de la ontología se trabaja separadamente y la tarea principal se dirige hacia la anotación semántica solamente.

3. Análisis Preliminar

Durante la etapa de análisis preliminar, que antecede a la definición de los requerimientos, se llevaron a cabo actividades tendientes a realizar un análisis del dominio de trabajo. El objetivo de esta etapa fue la de analizar la Plataforma Signar y sus fuentes de información para comprender el dominio de aplicación del módulo a desarrollar. La segunda actividad que se realizó tuvo como propósito analizar el dominio del anotado semántico y algunas técnicas que se utilizan en dicho dominio y que son importantes para definir el alcance y usos de la aplicación que se va a desarrollar.

3.1. Análisis de la Plataforma Signar

Refiere al análisis tanto de tecnologías utilizadas como de funcionalidades presentes en la plataforma Signar. Se detallan a continuación los objetivos de esta actividad.

- **Relevamiento de tecnologías:** Relevar cuáles son las tecnologías utilizadas en la plataforma para coordinar luego la implementación de estándares, modelos y tecnologías que sean compatibles con las mismas a la hora de especificar requerimientos e implementar soluciones.

- **Relevamiento de funcionalidades:** Detectar los flujos de trabajo presentes en el sistema, cómo interviene el modelado de la información en el almacenamiento de metadatos, cómo se realizan las búsquedas de documentos actualmente en la plataforma, entre otras.

Durante esta etapa se mantuvieron reuniones periódicas, tanto con el equipo de tecnologías de información y análisis funcional como con el área directiva de Lyris IT para diferentes tareas: 1) Relevamiento de requerimientos generales y específicos para la consultoría, 2) Análisis y corrección de posibles desvíos que se produzcan en la consultoría, 3) Relevamiento de información asociada a la plataforma y al esquema de datos y 4) Solicitud de cambios para adaptabilidad de la plataforma a fines de implementar requerimientos.

Debido a las restricciones impuestas por la pandemia de Covid 19, todas las reuniones se llevaron a cabo en forma virtual. La virtualidad en este caso nos dio otra forma de trabajar sin ser un impedimento para llevar a cabo las tareas programadas. Más aún, generó una comunicación más fluída con la empresa dado que las reuniones se podían organizar en franjas horarias más amplias y con más duración.

Para realizar el análisis de la Plataforma Signar, se solicitó a Lyris IT un dump SQL con datos y esquema asociado a la base de datos del sistema. A partir de la misma se realizó un análisis de:

- Donde están ubicados cada uno de los datos/informaciones relevantes para incorporar al modelo semántico y a la especificación de requerimientos
- Cómo están relacionadas las diferentes entidades y tablas que representan la información del sistema
- Qué restricciones poseen la relaciones
- Qué tipos de datos están asociados a cada una de las informaciones asociadas a documentos (cadenas de texto, enteros, flotantes, valores de verdad, etc).

- Qué datos de los indicados se actualizan, bajo qué condiciones y cuándo.

3.2. Estándares, Modelos y Tecnologías Semánticas

Refiere al relevamiento y análisis tanto de estándares como de modelos y tecnologías semánticas asociadas y relevantes para el manejo de documentos y de metadatos asociados a los mismos.

Dentro de algunas posibilidades se enuncia una lista no taxativa y meramente descriptiva de estándares, modelos y tecnologías que fueron materia de análisis para actuales y futuras implementaciones: DublinCore, CIDOC CRM [1], ElasticSearch, Apache Lucene, Apache Tika, Spacy, GraphDB y BPMN.

4. Construcción del Modelo Semántico para la plataforma

El objetivo de esta etapa es el desarrollo de un modelo semántico, basado en ontologías, confeccionado a medida donde se representen las terminologías y modelos que son relevantes para la plataforma Signar.

Para el desarrollo del modelo semántico se siguió la metodología NeOn (Gómez-Pérez) basada en escenarios, definida para el diseño e implementación de redes de ontologías. En particular, se implementó el escenario que propone la reutilización de ontologías. A partir del relevamiento de los requerimientos se identificaron los términos principales del dominio.

Una vez identificado los términos principales, sus propiedades y sus relaciones se procedió a modelar los mismos en la herramienta de edición de ontologías Protégé (https://protege.stanford.edu/). Dicha herramienta ofrece la posibilidad de guardar la ontología modelada en diferentes lenguajes, como RDF, OWL y Turtle.

Una vez definida la estructura principal de la ontología, se decidió enriquecerla con las ontologías estándares que se habían

analizado. De esta forma, tenemos un modelo interoperable. Ese enriquecimiento se muestra a continuación (Figura 1).

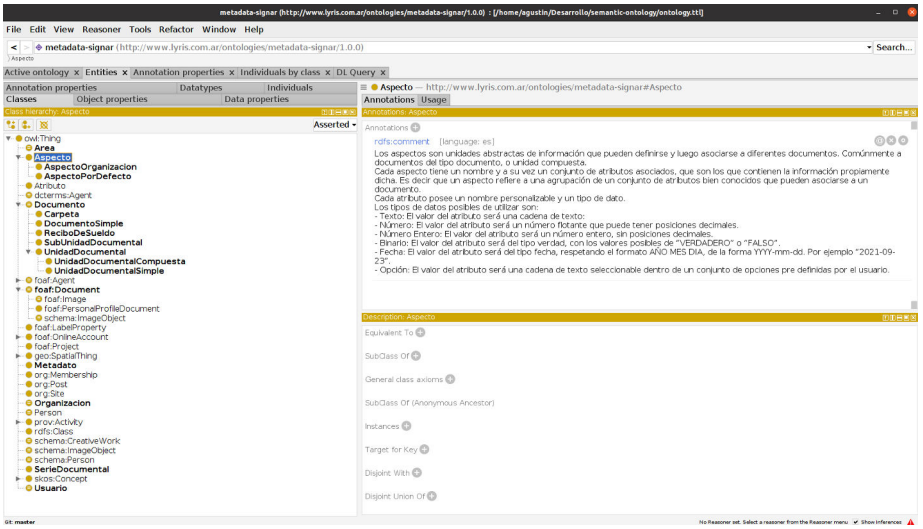


Figura 1. Modelo ontológico del dominio.

Con el objetivo de enriquecer la ontología base desarrollada se procedió a reutilizar las ontologías estándares recomendadas por la W3C (https://www.w3.org/): FOAF, Organization Ontology y Dublin Core Elements. Para realizar el enriquecimiento, las mismas se importaron en el modelo anterior utilizando la herramienta Protégé. Protégé es un editor de ontologías de código abierto y un sistema de adquisición de conocimiento. Esta herramienta cuenta con el respaldo de una sólida comunidad de desarrolladores y usuarios académicos, gubernamentales y corporativos.

5. Diseño e Implementación de la Arquitectura

Tomando como base la arquitectura relevada con respecto a tecnologías y servicios de la plataforma Signar, se detalla en esta sección la arquitectura de la solución de la API desarrollada y a su vez cómo se integra en la arquitectura de Signar.

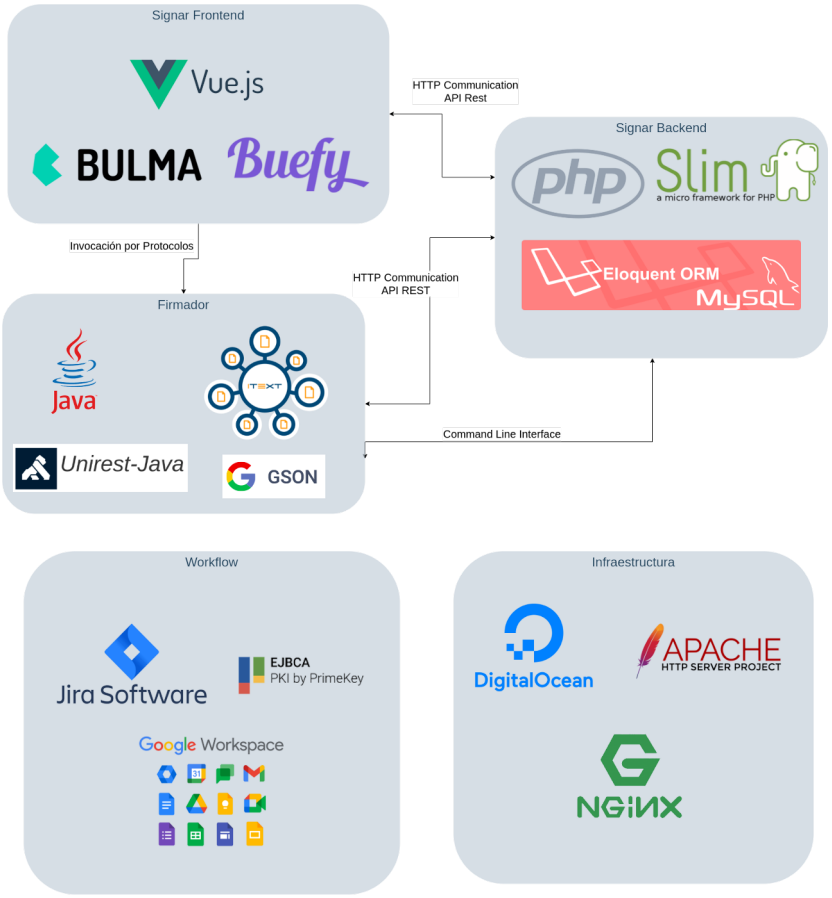


Figura 2. Arquitectura de Signar

La arquitectura con la que consta la solución desarrollada se integra exclusivamente en la sección de servicios Backend de la presente arquitectura de Signar.

En detalle, la interacción con el API semántico desarrollado en Python, se dará exclusivamente desde el servicio de backend de Signar en una red privada y con seguridad mínima con autenticación básica http (incluída en el desarrollo y habilitada según configuración de entorno).

Por otro lado, si bien el componente API semántico desarrollado será el que se comunique en forma exclusiva con el servicio de ElasticSearch, no se descarta la posibilidad de una interacción directa entre los servicios backend de Signar y el servidor de ElasticSearch ya

que pueden realizarse interacciones personalizadas, de una forma similar a la que hoy se interactúa con los servicios de MySQL.
Se deja un detalle a continuación de la arquitectura de la solución.

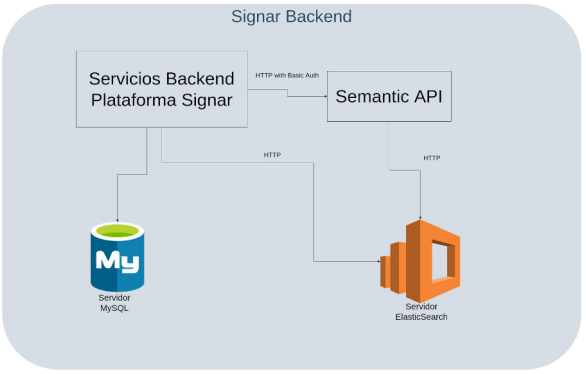


Figura 3. Arquitectura de Signar con Semantic API

Para la implementación de la solución se utilizaron las siguientes tecnologías.

5.1. Elasticsearch

Elasticsearch es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multitenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache.

Casos de uso

1. Búsqueda de información en una app o sitio web
2. Motor de almacenamiento para automatizar flujos de negocio
3. Machine learning, obtener información sobre grandes set de datos.
4. Manejar información geoespacial usando elasticsearch como un GIS.
5. Entre otros.

5.2. Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Las principales librerías que se utilizaron para la construcción de la API son:

- flask[1]: Web microframework para la creación de endpoints y manejo de requests/responses http.
- elasticserach-dsl[2]: Manipulación alto nivel de elastic para Python. Permite acceder a un cluster elasticsearch y manipularlo.
- marshmallow[3]: Validaciones de objetos a partir de la creación de esquemas. En este caso utilizado para validar la estructura de los documentos.
- flask-swagger-ui[4]: Swagger ui para flask. Con Swagger se pueden visualizar de una forma gráfica la documentación OpenAPI de un conjunto de servicios web.

5.3. Especificación OpenAPI

La especificación OpenAPI[5], originalmente conocida como la especificación Swagger, es una especificación para archivos de interfaz legibles por máquina para describir, producir, consumir y visualizar servicios web RESTful.

Todos los servicios web implementados en la solución fueron documentados con esta especificación.

---

[1] https://flask.palletsprojects.com/en/2.0.x/
[2] https://elasticsearch-dsl.readthedocs.io/en/latest/
[3] https://marshmallow.readthedocs.io/en/stable/
[4] https://github.com/sveint/flask-swagger-ui
[5] https://www.openapis.org/

5.4. Mapeo Índice Elasticsearch

Para la elaboración de los servicios de búsqueda y manipulación de documentos, se realizó un relevamiento de las fuentes de información y una propuesta de mapeo simple.

En la siguiente imagen se puede observar el mapeo definido con los diferentes tipos de datos elasticsearch, para cada uno de los atributos definidos en el mapeo simple.

```
class Usuario(InnerDoc):
  acceso = Keyword()
  nombres = Text()

class Organizacion(InnerDoc):
  slug = Keyword()
  razon_social = Text()

class Permiso(InnerDoc):
  id = Keyword()
  tipo = Keyword()

# Clase documento asociada a los documentos de signar
class Documento(Document):
    id = Keyword()
    estado = Text()
    mime_type = Keyword()
    confidencial = Boolean()
    ruta = Text()
    nombre = Text()
    tipo = Keyword()
    organizacion = Object(Organizacion)
    propietario = Object(Usuario)
    permisos = Nested(Permiso)
```

Figura 4. Mapeo de indice ElasticSearch

Más detalles de definición de cada uno de los tipos de datos existentes en elasticsearch puede encontrarse en la documentación de ElasticSearch[6].

6 https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html

Este mapeo es el que habilita luego la inicialización del índice relacionado a los documentos en elasticsearch y a su vez es la clase Python que permitirá luego hacer uso de las funciones que nos brinda la librería de implementación para la creación, eliminación, modificación y búsqueda de documentos.

5.5. Servicios Web

Se detallarán en el presente apartado los diferentes servicios web implementados para cumplimentar las actividades:

1. Instanciación de ElasticSeach con dicho mapeo
2. Prueba y ajuste de mapeo Elasticsearch para anotado simple
3. Servicio web para ABM de documentos en índices
4. Desarrollo y prueba de servicio web para búsqueda simple
5. Desarrollo y prueba de servicio web para búsqueda avanzada

5.5.1 Inicialización de Índice de Documentos en Elasticsearch

Se utiliza este servicio web para la inicialización/instanciación del índice de documentos a partir de mapeo definido anteriormente.
Esto es necesario realizarlo cuando se instala e inicia elasticsearch por primera vez y el mapeo aún no existe en el servidor. A su vez, es necesario su utilización si por alguna razón se elimina el índice, por ejemplo para su reconstrucción

5.5.2 Alta, baja y modificación de Documentos

Todos los servicios para alta, baja y modificación de documentos respetan una arquitectura API RESTful (Fielding Roy, 2000).

Figura 5. Servicios de abm de documentos

- Alta de un documento en forma individual.
- Alta de más de un documento en forma masiva, haciendo uso de la operación bulk facilitada por ElasticSearch.
- Eliminación de un documento individual.
- Modificación de un elemento. Es importante aclarar que esta operación reemplaza el documento original por el que se envía dentro del cuerpo de la solicitud, a diferencia de otras implementaciones que hacen uso del verbo PATCH.
- Obtención de un documento individual a partir de su identificador.

5.5.3 Búsquedas

A nivel general, todos los servicios de búsqueda permiten los siguientes parámetros generales:

- **start**: Posición a partir de la cual se desean obtener resultados. Esto sirve para el paginado. Si por ejemplo listamos resultados en páginas de a 10, para ir a la página 2 deberíamos colocar el valor de "11".
- **end**: Posición hasta la cual se desean obtener resultados. Esto sirve para el paginado. Si por ejemplo listamos resultados en páginas de a 10, para ir a la página 2 deberíamos colocar el valor de "20".
- **permisos**: Vector/Arreglo donde se pueden indicar un listado de permisos, para los cuales al menos uno de los enviados debe existir en los documentos que se están retornando de la búsqueda.

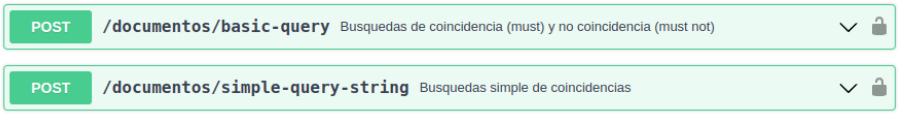


Figura 6. Servicios de busqueda

Basic Query

El servicio web de búsqueda basic query, permite hacer búsquedas de coincidencia (must) y no coincidencia (must not) puntuales de campos con ciertos valores. Se pueden enviar las coincidencias a incluirse y las a excluirse. Tanto para coincidencias como para no coincidencias se

utiliza la búsqueda full text “match query” de elastic. Los parámetros included_fields y excluded_fields son ambos objetos json donde dentro cada propiedad representa un campo del documento y el valor asociado indica la coincidencia que debe o no cumplirse.

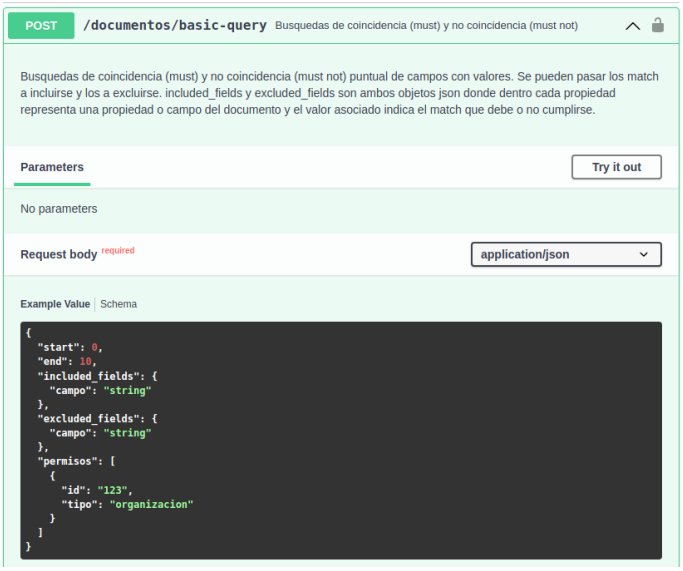


Figura 7. Servicio web para búsqueda basic query

Simple Query String

El servicio web de búsqueda de simple query string, permite la utilización de la búsqueda full text “simple query string” de elastic (Ver detalle en documentación elastic), asociada a las búsquedas simples y avanzadas sobre anotado simple. Se indica el texto a buscar, y si se desea se da un array de string con los campos en los que se desea buscar, pudiendo definir una prioridad. Si no se indica ningún campo específico, la búsqueda se realiza sobre todos los campos/metadatos del documento.

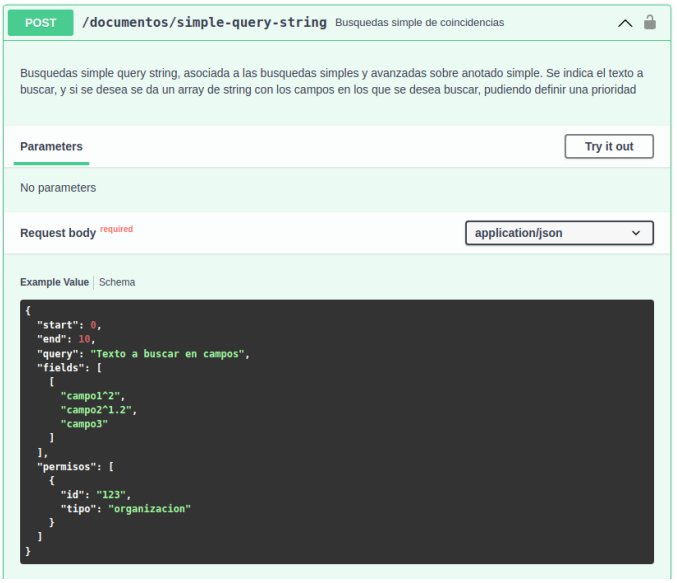


Figura 8. Servicio web para simple query string

6 Extensión del proyecto

Como extensión del proyecto, se realizó a su vez un trabajo en conjunto a partir de una propuesta generada a la finalización del trabajo principal.

Se propuso el desarrollo de un sub-módulo, integrado al sistema Signar, que permita la exportación del total de documentos anotados semánticamente existentes en una organización puntual, utilizando lenguajes y estándares para tal fin.

El propósito de tener este sub-módulo en la herramienta Signar es que los clientes puedan obtener una representación estandarizada y semántica del total de información que poseen dentro de la plataforma, a partir de lenguajes y estándares propuestos y difundidos por la W3C. Como es de público conocimiento, estos lenguajes están fuertemente relacionados y/o nacen a partir de la necesidad de integrar información en sistemas con diferentes modelos de base, por lo que contar con una

representación de este estilo permitirá al cliente poder trasladar y manipular la información según sean sus necesidades.
Para dar contexto, el Consorcio WWW, en inglés: World Wide Web Consortium (W3C), es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.
Este consorcio fue creado en octubre de 1994, y está dirigido por Tim Berners-Lee, el creador original del URL (Uniform Resource Locator, Localizador Uniforme de Recursos), del HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y del HTML (Hyper Text Markup Language, Lenguaje de Marcado de HiperTexto), que son las principales tecnologías sobre las que se basa la Web.
Dentro de los lenguajes y estándares utilizados en el modulo de exportación, se destacan:

- XML: https://www.w3.org/XML/
- RDF: https://www.w3.org/RDF/
- RDFS: https://www.w3.org/TR/rdf-schema/
- OWL: https://www.w3.org/OWL/
- OWL2: https://www.w3.org/TR/owl2-overview/

Es importante destacar que si bien los enunciados anteriormente son los principales, se destaca que cada estándar propone un conjunto de predicados para modelar los dominios a nivel de meta-modelos, a su vez se pueden utilizar diferentes lenguajes para la sintaxis, como ser RDF/OWL/Turtle.
Por ser el más leible por humanos, se llevo a cabo la implementación con la utilización de la sintaxis Turtle (http://www.w3.org/TR/turtle/).

6.1 Implementacion

En lo que respecta al desarrollo de la solución, al estar Signar desarrollado a nivel de servicios backend íntegramente en lenguaje PHP, se llevo a cabo, para una mejor integración, la utilización de la libreria https://www.easyrdf.org/
Se llevó a cabo el desarrollo de un módulo PHP que íntegramente toma la información existente en las bases de datos de Signar y realizará lo

que se conoce ampliamente como una serialización de dicha información al formato Turtle, utilizando los lenguajes/estándares anteriormente indicados y respetando el modelo semántico definido para la plataforma.

En la imagen que se muestra a continuación se puede ver una porcion del código asociado al módulo de exportación donde se crea el grafo representativo de los documentos y se realiza el mapeo parcial del usuario y la organización a la que está asociada:

```
// Inicializa graph para exportacion, primeramente con ontologia
$ontology = file_get_contents("../app/ontology.ttl");
$graph = new Graph();
$parser = new Turtle();
$totalTripletas = $parser->parse($graph, $ontology, "turtle", "");

// Creacion de recurso usuario
$uriResourceUsuario = "{$signarUri}Usuario-{$usuario->id}";
$resourceUsuario = $graph->resource($uriResourceUsuario, [ "{$signarUri}Usuario" ]);
$resourceUsuario->add("{$signarUri}email", $usuario->email);
$resourceUsuario->add("{$signarUri}cuil", $usuario->cuil);
$resourceUsuario->add("{$signarUri}acceso", $usuario->acceso);
$resourceUsuario->add("{$signarUri}nombrePersona", $usuario->nombre);
$resourceUsuario->add("{$signarUri}apellidoPersona", $usuario->apellido);

// Creacion de recurso organizacion
$uriResourceOrganizacion = "{$signarUri}Organizacion-{$organizacion->id}";
$resourceOrganizacion = $graph->resource($uriResourceOrganizacion, [ "{$signarUri}Organizacion" ]);
```

Figura 9. Parte del codigo fuente del modulo de exportacion

El modulo a su vez mapea informacion proveniente de cada uno de los documentos.

Este desarrollo se realizo integrando el codigo a la misma aplicación nucleo Signar, donde se trabajo en conjunto con desarrolladores de la empresa, utilizando enrutamiento propio de Slim php[7] y acceso a datos a partir del ORM Eloquent[8].

6.2 Visualizacion

Ademas de los beneficios de exportacion de datos en un formato apto para la integracion de sistemas de representacion de conocimiento, se comparte a continuacion ejemplos del resultado de importación de la exportación en la plataforma de Ontotext GraphDB[9].

[7] https://www.slimframework.com/
[8] https://laravel.com/docs/5.0/eloquent
[9] https://graphdb.ontotext.com/

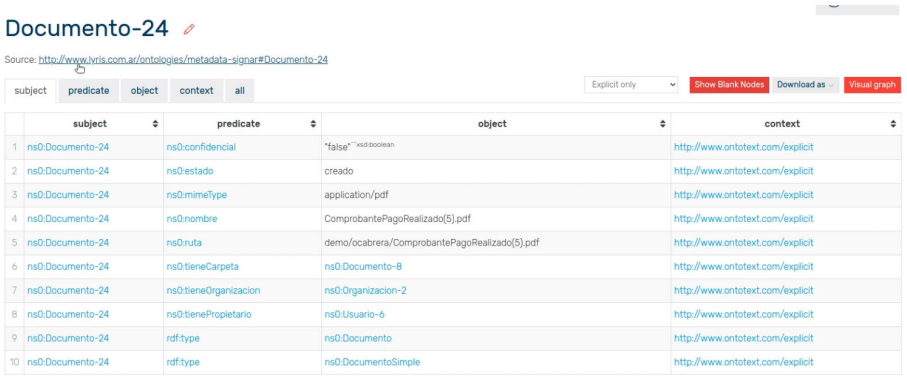
GraphDB es una base de datos de grafos y una herramienta de descubrimiento de conocimiento compatible con RDF y SPARQL y disponible como un clúster de alta disponibilidad.

A partir de esta integracion, se pueden consultar no solo almacenar los datos de los documentos, usuarios y organizaciones sino que puede generarse una instancia de explotación de datos a partir del uso de consultas SPARQL y a su vez la visualizacion y navegacion de los nodos existenes representando entidades y sus propiedades a traves de las relaciones existentes entre los mismos

Aqui se puede observar una representacion visual de una porcion de datos exportados desde la plataforma, luego de ser importados en graphdb.

Figura 10. Representacion visual de sub-grafo RDF

Y finalmente en la siguiente imagen se pueden observar la visualizacion tabular de los sujetos, predicados y objetos donde el sujeto en cuestión es uno de los documentos exportados.



Documento-24

Source: http://www.lyris.com.ar/ontologies/metadata-signar#Documento-24

subject | predicate | object | context | all

Explicit only | Show Blank Nodes | Download as | Visual graph

	subject	predicate	object	context
1	ns0:Documento-24	ns0:confidencial	"false"^^xsd:boolean	http://www.ontotext.com/explicit
2	ns0:Documento-24	ns0:estado	creado	http://www.ontotext.com/explicit
3	ns0:Documento-24	ns0:mimeType	application/pdf	http://www.ontotext.com/explicit
4	ns0:Documento-24	ns0:nombre	ComprobantePagoRealizado(5).pdf	http://www.ontotext.com/explicit
5	ns0:Documento-24	ns0:ruta	demo/ocabrera/ComprobantePagoRealizado(5).pdf	http://www.ontotext.com/explicit
6	ns0:Documento-24	ns0:tieneCarpeta	ns0:Documento-8	http://www.ontotext.com/explicit
7	ns0:Documento-24	ns0:tieneOrganizacion	ns0:Organizacion-2	http://www.ontotext.com/explicit
8	ns0:Documento-24	ns0:tienePropietario	ns0:Usuario-6	http://www.ontotext.com/explicit
9	ns0:Documento-24	rdf:type	ns0:Documento	http://www.ontotext.com/explicit
10	ns0:Documento-24	rdf:type	ns0:DocumentoSimple	http://www.ontotext.com/explicit

Figura 11. Representacion de tripletas rdf en graphdb

7 Discusión de los resultados y lecciones aprendidas

Se destaca el trabajo en equipo logrado a través de una interacción del sector privado con la Universidad, como primer punto relevante de lo aprendido. Este trabajo es posible y potencia las capacidades de las empresas de un modo efectivo. Y desde la óptica de la Universidad, permite plasmar en el medio en el que se inserta, todos aquellos conocimientos que desarrolla y expande de forma continua internamente, motivando a quienes día a día estudian e investigan pensando en sus posibles aplicaciones.

Por otra parte, desde el punto de vista técnico, se han sentado las bases para lograr una mejora inmediata en las búsquedas de documentos a través de Signar Gestor Documental pero también para continuar con otros proyectos derivados, tales como la extracción de información de dichos documentos para realizar un análisis posterior de la misma y generar conocimiento para la toma de decisiones. Queda ahora por parte de la empresa aplicar todo lo aprendido durante el desarrollo de este proyecto para seguir mejorando su producto.

References

1. Aguado de Cea, G.; Álvarez de Mon y Rego, I, Pareja Lora, A. (2009) Un visión interdisciplinar de la anotación semántica. Terminología y sociedad del conocimiento / coord. por María Amparo Alcina Caudet, Esperanza Valero Doménech; Elena Rambla (aut.), ISBN 978-3-03911-593-8, págs. 219-254.
2. Almeida, F.; Duarte Santos, J. and Augusto Monteiro, J. (2020) "The Challenges and Opportunities in the Digitalization of Companies in a Post-COVID-19 World", in *IEEE Engineering Management Review*, vol. 48, no. 3, pp. 97-103, 1 third quarter, Sept. 2020.
3. Fielding, R. T. (2000) Architectural Styles and the Design of Network-Based Software Architecture. . Ph.D. Dissertation. University of California, Irvine.
4. Gómez-Pérez, A.; Suárez de Figueroa Baonza, M. C,; Villazón, B. (2008). Neon methodology for building ontology networks: Ontology specification .Methodology, 1-18.
5. Le Boeuf, P., Doerr, M., Emil Ore, C., Stead, S (2018). CIDOC Conceptual Reference Model. Produced by the ICOM/CIDOC. Accesible en: http://www.cidoc-crm.org/Version/version-6.2.3
6. Viltres Sala, H.; Rodríguez Leyva, P. Componente para la anotación semántica de información. Avances, ISSN-e 1562-3297, Vol. 21, Nº. 1, 2019, págs. 32-44.
7. Vrana, J., Singh, R. (2021). Digitization, Digitalization, and Digital Transformation. In: Meyendorf, N., Ida, N., Singh, R., Vrana, J. (eds) Handbook of Nondestructive Evaluation 4.0. Springer, Cham.
8. Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). University of California, Irvine.