



Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado Universitario en Ingeniería Informática

Trabajo de Fin de Grado

Transformación de los datos de licitaciones de la Plataforma de Contratación Pública al estándar OCDS y su publicación como datos enlazados

Autor: **Pablo Palomo López**

Tutora: **María Poveda Villalón**

Supervisor: **Óscar Corcho García**

Madrid, mayo de 2021

Abstract

...

Resumen

...

Agradecimientos

...

Índice

Abstract	i
Resumen	ii
Agradecimientos	iii
1 Introducción	1
2 Estado del arte	2
2.1 Datos abiertos y datos enlazados	2
2.1.1 Modelo de datos RDF	4
2.1.2 Consultas sobre datos enlazados	5
2.2 Licitaciones públicas	6
2.2.1 Estándar CODICE	6
2.2.2 Estándar OCDS	7
2.3 <i>TheyBuyForYou</i>	9
3 Software desarrollado	11
3.1 Consideraciones de diseño	11
3.1.1 Lenguaje de programación	11
3.1.2 Arquitectura de componentes orientada a servicios	11
3.1.3 Patrón de productor-consumidor	11
3.1.4 Mecanismo de consultas	12
3.1.5 Librerías utilizadas	12
3.2 Estructura de la aplicación	13
3.3 Diseño de la arquitectura	14
3.3.1 Componentes software	14
3.3.2 Especificación de requisitos	16
3.4 Diseño detallado del sistema	19
3.4.1 Casos de uso del componente de provisión de datos	19
3.4.2 Casos de uso del componente de parseo de datos	21
3.4.3 Casos de uso del componente de mapeado de datos	22
3.4.4 Casos de uso del componente de empaquetado de datos	23
3.5 Trazabilidad de requisitos	23
3.6 Flujo de ejecución	25
3.7 Cobertura de código	28
4 Correspondencias entre CODICE y OCDS	29
4.1 Datos generales del expediente	29
4.1.1 Estado	29
4.1.2 Número de expediente	29
4.1.3 Objeto del contrato	30
4.1.4 Valor estimado e importe de licitación	30
4.1.5 Duración del contrato	30
4.1.6 Tipo de contrato	30
4.2 Lotes	31
4.2.1 Número de lote	31
4.2.2 Objeto del lote	31

4.2.3	Importe del lote	31
4.2.4	Clasificación CPV	32
4.3	Procesos de licitación	32
4.3.1	Tipo de procedimiento	32
4.3.2	Sistema de contratación	33
4.3.3	Presentación de la oferta	33
4.3.4	Idioma de presentación de la oferta	33
4.3.5	Licitación son subasta electrónica	33
4.4	Entidades adjudicadoras	34
4.4.1	Órgano de contratación	34
4.4.2	Ubicación orgánica	34
4.4.3	Otros campos	34
4.5	Resultado del procedimiento	35
4.5.1	Identificador	35
4.5.2	Resultado	35
4.5.3	Identidad del adjudicatario	35
4.5.4	Importe de adjudicación	36
4.5.5	Número de licitadores participantes	36
4.5.6	Fecha de la adjudicación	36
4.5.7	Descripción de la adjudicación	36
4.6	Información sobre el contrato	37
4.6.1	Identificador	37
4.6.2	Fecha de entrada en vigor	37
5	Generación de datos enlazados	38
5.1	Validación de los datos	38
5.2	Reglas RML	39
5.3	Consultas sobre los datos generados	40
5.4	Despliegue del sistema	40
6	Análisis del impacto del trabajo	42
7	Conclusiones	43
8	Referencias	44
9	Anexos	53
9.1	Anexo I: Interfaces de los componentes del <i>OCDS_Mapper</i>	53
9.1.1	Componente <i>Provider</i>	53
9.1.2	Componente <i>Parser</i>	54
9.1.3	Componente <i>Mapper</i>	56
9.1.4	Componente <i>Packager</i>	57
9.2	Anexo II: Ejemplos de los mapeados entre CODICE y OCDS	58
9.2.1	Datos generales del expediente	58
9.2.2	Lotes y artículos	62
9.2.3	Procesos de licitación	65
9.2.4	Entidades adjudicadoras	67
9.2.5	Resultado del procedimiento	70
9.2.6	Información sobre el contrato	74
9.3	Anexo III: Consultas SPARQL sobre los datos enlazados	76

9.3.1	Prefijos de los espacios de nombres utilizados	76
9.3.2	Consulta para la búsqueda de <i>releases</i>	76
9.3.3	Consulta para la búsqueda de procesos de contratación	76
9.3.4	Consulta para la búsqueda de adjudicaciones	77
9.3.5	Consulta para la búsqueda de contratos	78
9.3.6	Consulta para la búsqueda de planificaciones	78
9.3.7	Consulta para la búsqueda de licitaciones	79
9.3.8	Consulta para la búsqueda de artículos y lotes de licitaciones	80
9.3.9	Consulta para la búsqueda de organizaciones involucradas	81

Índice de tablas

Tabla 1	Tabla de descripción de componentes	15
Tabla 2	Tabla de requisitos funcionales	19
Tabla 3	Tabla de trazabilidad de requisitos	25

Índice de figuras

Figura 1	Zoom de la <i>Linked Open Data Cloud</i>	2
Figura 2	Ejemplo de una consulta de datos enlazados	3
Figura 3	Ejemplo de tripletas del modelo RDF	4
Figura 4	Ejemplo de consultas en lenguajes SQL y SPARQL	5
Figura 5	Logo del estándar CODICE (versión 2)	6
Figura 6	Logo del estándar OCDS	7
Figura 7	Modelo del proceso de licitaciones en OCDS	8
Figura 8	Logo del proyecto TBFY	9
Figura 9	Dimensiones del grafo de conocimiento de TBFY (mayo de 2021)	10
Figura 10	Ejemplo de una sentencia de LINQ	12
Figura 11	Diagrama de estructura de la aplicación	14
Figura 12	Diagrama de los componentes del sistema	15
Figura 13	Diagrama UML del componente <i>Provider</i>	20
Figura 14	Diagrama UML del componente <i>Parser</i>	21
Figura 15	Diagrama UML del componente <i>Mapper</i>	22
Figura 16	Diagrama UML del componente <i>Packager</i>	23
Figura 17	Flujo de ejecución general de la aplicación	25
Figura 18	Flujo de ejecución de la provisión de documentos	26
Figura 19	Flujo de ejecución del procesado de documentos	27
Figura 20	Métricas registradas por <i>SonarQube</i>	28
Figura 21	<i>Pipeline</i> y formato de los datos en las distintas etapas	38
Figura 22	Reporte general de la herramienta de revisión de OCDS	38
Figura 23	Reporte específico de la herramienta de revisión de OCDS	39
Figura 24	Comparativa de tiempos entre <i>RMLMapper</i> y <i>RocketRML</i>	39
Figura 25	Captura de la web https://publiccontracting.linkeddata.es	41
Figura 26	Captura de parte del directorio json/ de la web	41
Figura 27	Captura de parte del directorio rdf/ de la web	41
Figura 28	Logo del Objetivo de Desarrollo Sostenible nº 16	42

1 Introducción

...

2 Estado del arte

2.1 Datos abiertos y datos enlazados

Para poder valorar este proyecto, es fundamental comprender las bases sobre las que se sustenta el mismo: los datos abiertos y los datos enlazados.

Los datos abiertos (*open data*, en inglés) son datos que pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona, y que se encuentran sujetos, a lo sumo, al requerimiento de atribución y de compartirse de la misma manera que aparecen [1].

Sin embargo, más allá de las propiedades que los datos abiertos deben satisfacer (disponibilidad, accesibilidad, reutilización, redistribución y participación universal), el valor de estos datos es la interoperabilidad. La interoperabilidad denota la habilidad de diversos sistemas y organizaciones para trabajar en conjunto. En este escenario, nos referimos a la cualidad de integrar diferentes conjuntos de datos.

Gracias a la interoperabilidad de los datos abiertos, se pueden construir [sistemas](#) como los que se describirán en este trabajo posteriormente, capaces de reutilizar los datos publicados por una fuente y transportarlos a otro dominio, con el objetivo de explotar la información a través de nuevas vías.

Por otro lado, los datos enlazados (*linked data*, en inglés) son la base de la denominada *Web Semántica*. La peculiaridad de esta información reside en la referenciación: mediante dicho mecanismo, los datos pueden vincularse entre ellos de la misma manera que lo hacen los enlaces de las páginas web [2].

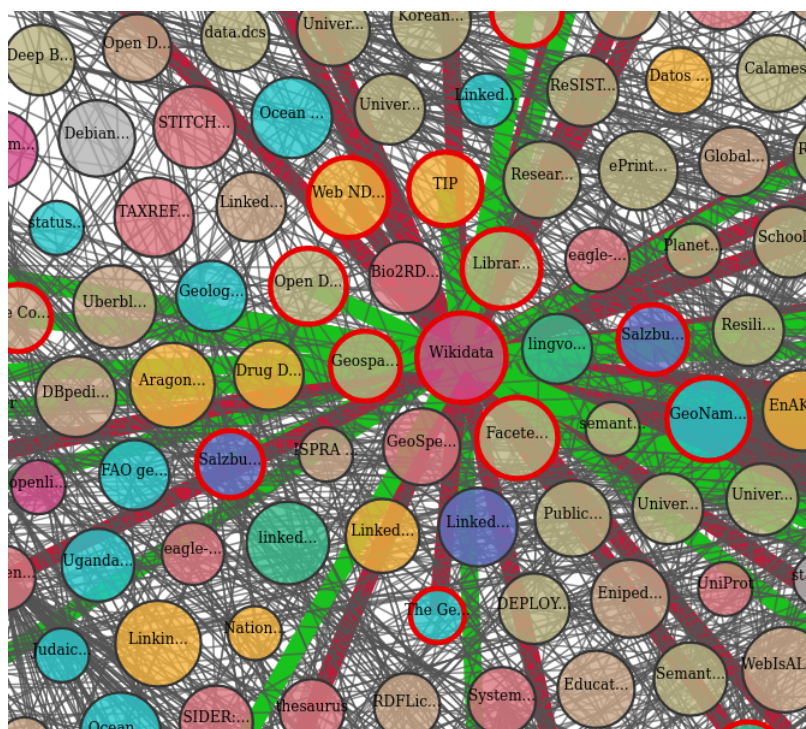


Figura 1: Zoom de la *Linked Open Data Cloud*

En la [figura 1](#) se puede apreciar una pequeña parte de la *Linked Open Data Cloud* [3], que refleja todos los conjuntos de datos abiertos y enlazados publicados, así como las relaciones entre los mismos. Esta inmensa “nube” evidencia el poder y el valor de esta forma de publicación de datos, capaz de conformar una red de conocimiento muy amplia.

A diferencia de otros modelos de almacenamiento y gestión de la información, como las bases de datos convencionales, en los que se necesitan mecanismos auxiliares para reflejar esta referenciación e interconexión de datos, los datos enlazados poseen de manera intrínseca estas relaciones. Esta diferencia radica en la forma en la que se describen los datos: la web del hipertexto, la que estamos acostumbrados a explorar, suele estar descrita mediante el lenguaje HTML (por sus siglas en inglés, *HyperText Markup Language* [4]), mientras que la *Web Semántica* se describe mediante el modelo RDF (*Resource Description Framework*, en inglés [5]).

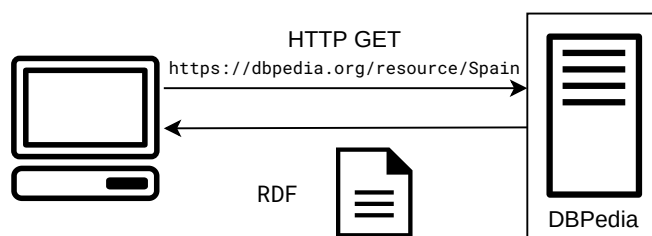


Figura 2: Ejemplo de una consulta de datos enlazados

A modo de ejemplo, y con el objetivo de reflejar algunos de los principios básicos de los datos enlazados, se muestra en la [figura 2](#) un caso de uso de una consulta sobre datos enlazados. La consulta es simple: un equipo solicita el recurso que representa la entidad de España al servidor de *DBpedia* [6], uno de los proyectos más ambiciosos de la “Web Semántica”, que busca extraer la información de la más amplia enciclopedia *online*, *Wikipedia*, para generar datos enlazados.

En el ejemplo se pueden ver algunas de las peculiaridades de los datos enlazados:

- Uso de URIs como identificadores: con el objetivo de evitar ambigüedades y poder ofrecer una forma estandarizada y unívoca de identificación de entidades, todos los recursos se identifican mediante una URI (por su traducción del inglés, Identificador de recursos uniforme). Por lo tanto, una URI no es sólo una dirección, sino también un identificador: en el ejemplo, <https://dbpedia.org/resource/Spain> es la dirección en la que se aloja el recurso, pero también es el identificador que permite a cualquier otro sistema reutilizar dicho recurso.
- Protocolo HTTP: para evitar la multiplicidad de esquemas de URIs, se utiliza HTTP para asegurar que cualquier recurso pueda ser buscado y accedido en la Web.
- Obtención de recursos mediante RDF: Una vez se ha realizado una búsqueda y se pretende acceder a un recurso identificado por su URI, se obtiene una respuesta en forma de documento en formato RDF, que será explicado con más precisión en la [posterior sección](#). Además, la forma de realizar consultas estructuradas sobre datos enlazados más allá de su obtención mediante URIs será descrita en [otra sección](#).

2.1.1 Modelo de datos RDF

Si bien debido a la comparación previa entre la Web estándar y la denominada “Web Semántica” podría parecer que RDF es un lenguaje de marcado de texto como lo es HTML, esto no es así: RDF es un *framework* de representación de información: un modelo que permite codificar la información, sus propiedades, y las relaciones entre sí.

La sintaxis de este modelo es simple: se basa en conjuntos de tripletas de la forma sujeto-predicado-objeto, donde éstos pueden ser URIs, literales o elementos vacíos. Mediante estos conjuntos de definiciones se pueden construir *datasets*, que a su vez se pueden organizar en grafos de conocimiento; estructuras con una alta riqueza en términos de información.

```
<http://dbpedia.org/resource/Spain>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://schema.org/Country> .
<http://dbpedia.org/resource/Spain>
  <http://dbpedia.org/ontology/currencyCode>
  "EUR" .
<http://dbpedia.org/resource/Spain>
  <http://www.w3.org/2002/07/owl#sameAs>
  <http://openei.org/resources/Spain> .
```

Figura 3: Ejemplo de tripletas del modelo RDF

En la [figura 3](#) se pueden apreciar algunas de las tripletas que se otorgarían como respuesta a una petición como la de la [figura 2](#). En [azul](#) aparecen los sujetos de las tripletas, mientras que en [verde](#) y en [rojo](#) aparecen los predicados y los objetos, respectivamente.

La información de las tripletas, como se explicó previamente, está codificada mediante URIs, nodos vacíos o literales que representan entidades, propiedades y relaciones. En este caso, la primera tripleta nos dice que la entidad que representa a España es de tipo ([type](#)) país ([Country](#)). La segunda, que en España el tipo de moneda ([currencyCode](#)) tiene un código ([EUR](#)), que es un literal. La última de ellas establece una relación de identidad ([sameAs](#)) entre el recurso [http://dbpedia.org/resource/Spain](#) y el recurso [http://openei.org/resources/Spain](#), que se encuentran en distintos dominios.

En este ejemplo también pueden ser destacados otros elementos característicos de RDF: los vocabularios. Para expresar las propiedades (como [type](#) o [sameAs](#)), que potencialmente pueden aparecer en múltiples *datasets*, es de interés poder contar con un índice de términos que poder reutilizar, siguiendo uno de los principios fundamentales de los datos abiertos y enlazados. Ésa es la función de proyectos como *Linked Open Vocabularies* [7], que recoge a modo de código todos los términos publicados como datos abiertos.

Un último apunte sobre RDF y el ejemplo provisto; como previamente se explicó sobre RML, éste no es un lenguaje de marcado sino un *framework* que refleja cómo deben de ser los datos enlazados para que sigan los principios que los rigen. El ejemplo de la [figura 3](#) está codificado mediante el formato N-Triples [8] (un subconjunto del formato Turtle [9]). Sin embargo, existen otros formatos, como JSON-LD [10], RDF/XML [11], RDFa [12] o N-Quads [13].

2.1.2 Consultas sobre datos enlazados

De la misma manera que para la realización de consultas sobre bases de datos convencionales se tiene el lenguaje SQL [14], para construir búsquedas estructuradas sobre grafos de conocimiento de datos enlazados se utiliza SPARQL [15].

SPARQL trabaja con los datos en formato RDF y contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones. La sintaxis de SPARQL es similar a la de su análogo de bases de datos relacionales; en la figura 4 se puede apreciar una consulta similar mediante los dos lenguajes:

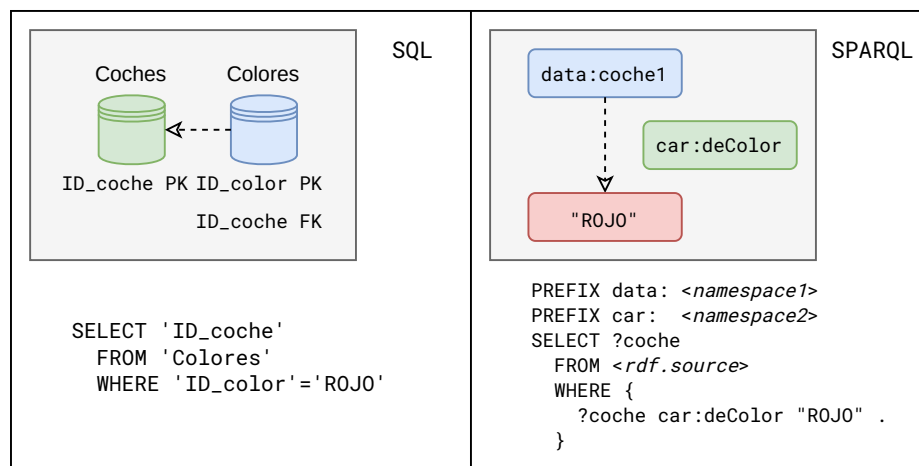


Figura 4: Ejemplo de consultas en lenguajes SQL y SPARQL

Las dos consultas muestran similitudes en distintos aspectos, como las cláusulas (SELECT, FROM, WHERE...) o la estructura de la sentencia, pero también evidencian las diferencias entre los lenguajes y los modelos tanto de bases de datos relacionales, como el modelo RDF: las consultas SQL se realizan sobre unos datos estructurados y sus atributos, teniendo que conocer por anticipado dicha estructura para poder construir la sentencia. En cambio, mediante SPARQL se pueden consultar directamente las tripletas del conjunto de datos. Además, este protocolo permite el uso de prefijos para ejecutar las consultas sobre elementos en dominios diversos.

Existen diversos sistemas que implementan, mediante *APIs* o *triplestores* (un tipo de base de datos *ad-hoc* para datos enlazados), sistemas con mecanismos de construcción de consultas SPARQL. Algunas de dichas implementaciones son:

- Apache Jena [16]: *framework* de código abierto escrito en el lenguaje Java para el desarrollo de aplicaciones sobre datos RDF.
- RDFLib [17]: biblioteca en Python para el procesamiento de datos en formato RDF/XML y sus grafos de conocimiento asociados. Utilizada [posteriormente](#) en el trabajo.
- RDF4J [18]: previamente conocido como *OpenRDF Sesame*, es otro *framework* de código abierto escrito en el lenguaje Java.
- Virtuoso [19]: *middleware* que combina las funcionalidades de un gestor de bases de datos relacionales convencionales con el procesamiento de datos enlazados.

2.2 Licitaciones públicas

2.2.1 Estándar CODICE

CODICE es una librería de componentes y documentos electrónicos XML estándar para el desarrollo de aplicaciones de contratación pública electrónica de conformidad con los procedimientos y prescripciones de las directivas 2014/23/UE [20]; 2014/24/UE [21]; 2014/25/UE [22]; 2009/81/CE [23] y de la normativa española en materia de contratación pública [24], así como con los estándares y recomendaciones internacionales aplicables a la identificación, denominación, definición y construcción de dichos componentes.

El proyecto CODICE persigue proporcionar a todo el Sector Público una arquitectura de componentes, documentos y mensajes XML estandarizados, conforme con las normas y estándares internacionales aplicables, que pueda ser usada por todos los sistemas, aplicaciones y componentes informáticos necesarios para la construcción de soluciones interoperables de contratación electrónica.

La finalidad del proyecto es asegurar la interoperabilidad, tanto de los subsistemas de contratación electrónica entre sí (Plataforma de Contratación Pública, registros electrónicos de empresas, catálogos electrónicos, sistemas de subastas electrónicas, etc.), como con los sistemas de información de los agentes económicos participantes en los procesos de contratación y con los de los propios órganos de las Administraciones Públicas.



Figura 5: Logo del estándar CODICE (versión 2)

Para acomodar de forma armónica y coordinada las variadas necesidades y condicionantes de los distintos usuarios potenciales de los sistemas de contratación electrónica a desplegar, así como para facilitar la interoperabilidad de la plataforma de contratación pública y sus distintos subsistemas con las infraestructuras técnicas, organizativas y de información de todos los participantes en los procesos de contratación pública, se ha definido una arquitectura [25] que proporciona tanto los componentes comunes esenciales del sistema (definiciones y denominaciones normalizadas, componentes o módulos elementales reutilizables, documentos electrónicos de uso común, etc.), como la estructura y reglas que permite su extensión o adaptación a las necesidades especiales de diferentes contextos específicos de contratación.

Dicha arquitectura proporciona a los sistemas a desarrollar posteriormente los “bloques” o componentes básicos de información, así como las reglas de composición necesarias para la construcción de los documentos y mensajes electrónicos que licitadores y órganos de contratación han de intercambiarse a lo largo de los procesos de contratación, garantizando al mismo tiempo la interoperabilidad de todos los elementos cuya interacción proporcionará las funcionalidades necesarias para hacer efectiva la contratación electrónica [26].

Las normas y estándares internacionales existentes, tales como las normas ISO 11179 [27] y 15000 [28], los estándares ebXML de OASIS [29] y UN/CEFACT [30], y las Recomendaciones del W3C sobre lenguaje y esquema XML [31], junto con las iniciativas promovidas por el programa IDA de la Unión Europea en relación con la contratación electrónica [32], han proporcionado la base para la definición y construcción de una arquitectura interoperable de información [33].

En el portal institucional del Ministerio de Hacienda, concretamente en la sección de licitaciones publicadas en la Plataforma de Contratación del Sector Público [34], se pueden encontrar los documentos de licitaciones elaborados por la Dirección General del Patrimonio del Estado a partir de los datos que introducen los órganos de contratación como responsables de sus perfiles de contratantes. Estos documentos siguen el esquema CODICE, si bien no en todos los documentos van a aparecer los mismos elementos, dado que se proporcionan tres distintos tipos de documentos: licitaciones publicadas en la Plataforma de Contratación del Sector Público (excluyendo contratos menores), licitaciones publicadas mediante mecanismos de agregación y licitaciones correspondientes a contratos menores [35]. Este trabajo se ha centrado en el primero de los tipos previamente descritos, que incluye como superconjunto al tercero, el de los contratos menores.

2.2.2 Estándar OCDS

El Estándar de Datos para las Contrataciones Abiertas (OCDS, por sus siglas en inglés) permite la divulgación de datos y documentos de todas las etapas del proceso de contratación mediante la definición de un modelo de datos común. Se creó para apoyar a las organizaciones a aumentar la transparencia de la contratación y permitir un análisis más profundo de los datos de contrataciones por una amplia gama de usuarios [36].



Figura 6: Logo del estándar OCDS

OCDS es un estándar de datos abierto, gratuito y no protegido por derechos de propiedad intelectual para la contratación pública, implementado por más de 30 gobiernos en todo el mundo. Es el único estándar abierto internacional para la publicación de información relacionada con la planeación, licitación e implementación de contratos públicos y ha sido avalado por el G20, el G7 e importantes organizaciones internacionales.

OCDS describe cómo publicar datos y documentos en todas las etapas del proceso de contratación. Fue creado para brindar apoyo a las organizaciones con el objetivo de aumentar la transparencia en las contrataciones y permitir un análisis más exhaustivo de los datos sobre contrataciones por parte de una amplia variedad de usuarios. OCDS brinda:

- Una serie de campos de datos y documentos para publicación recomendados
- Un modelo de datos estructurado común
- Orientación y herramientas para la implementación y el uso de datos
- Perfiles para asociaciones público-privadas, proyectos de infraestructura, la Unión Europea y el Acuerdo sobre Contratación Pública de la Organización Mundial del Comercio
- Un mecanismo de extensión para incorporar información adicional clave a sus datos de OCDS
- Un servicio de asistencia global gratuito

Los datos sobre contrataciones publicados en un formato de OCDS resultan más fáciles de compartir, comparar y analizar. Permiten que quienes los publican adapten y reutilicen las herramientas existentes de visualización y análisis como éstas [37], lo cual reduce los costes y favorece la innovación. OCDS ofrece orientación sobre qué publicar y cómo publicar los detalles más importantes sobre contratación pública identificados por profesionales, investigadores y otras partes interesadas.

OCDS fue concebido para atender cuatro necesidades específicas de los usuarios, en función de un análisis campo por campo de cómo pueden utilizarse los datos para dar respuesta a tales necesidades. Estas son:

- Lograr mayores beneficios a cambio del dinero que se paga, con lo cual se permite que el gobierno ahorre tiempo y dinero.
- Generar un entorno de negocios más justo y condiciones equitativas para los proveedores.
- Mejorar la integridad pública al desalentar el fraude y la corrupción.
- Hacer un seguimiento de la prestación de los servicios y mejorarlos.

Si se incorporan a reformas más amplias, los gobiernos pueden utilizar las contrataciones abiertas para obtener mayores beneficios a cambio del dinero que se paga por los bienes, las obras y los servicios y para ganarse la confianza del sector privado, la sociedad civil y la ciudadanía [38].

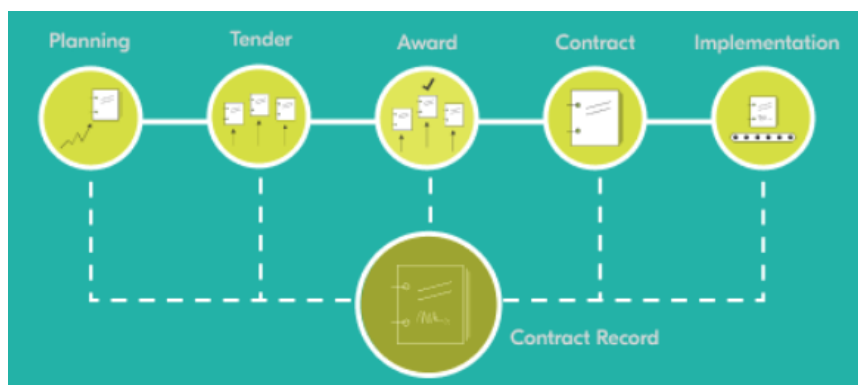


Figura 7: Modelo del proceso de licitaciones en OCDS

El proceso de licitaciones (véase [figura 7](#)), si bien sigue un modelo lineal, no siempre ha de cumplirse estrictamente de esa manera, dado que no todas las contrataciones pasan por todas las etapas, o en algunos casos es necesario proveer nueva información sobre una etapa previa.

Un documento OCDS está compuesto por bloques, íntegramente relacionados con el modelo del proceso de licitaciones [\[39\]](#):

- **parties**: información sobre las organizaciones y otros participante involucrados en el proceso de contratación.
- **planning**: información sobre los objetivos, presupuestos y proyectos a los que se refiere un proceso de contratación.
- **tender**: información sobre la forma en que tendrá lugar la licitación o su realización.
- **awards**: información sobre las adjudicaciones otorgadas como parte de un proceso de contratación.
- **contracts**: información sobre contratos firmados como parte de un proceso de contratación.
 - **implementation**: información sobre el progreso de cada contrato hasta su finalización.

2.3 *TheyBuyForYou*

TheyBuyForYou (en adelante, TBFY) es un consorcio de empresas, universidades, centros de investigación, departamentos gubernamentales y autoridades locales de diversos países europeos, concretamente Reino Unido, Noruega, Italia, España y Eslovenia.



Figura 8: Logo del proyecto TBFY

El proyecto de TBFY comenzó en enero de 2018 con el objetivo de mejorar el estado del mercado de adquisiciones europeo mediante el desarrollo de herramientas y funcionalidades que permitan explorar los datos abiertos en el ámbito de las licitaciones públicas. Para situarlo en contexto, ya hay diferentes organizaciones aprovechando el uso de las herramientas desarrolladas por el equipo de TBFY; el gobierno de Eslovenia las utiliza para identificar patrones que pudieran ser indicadores de fraude y autoridades españolas las usan para tomar decisiones de contrataciones más transparentes, entre otros casos.

El mercado de licitaciones europeo es tan grande que su navegación por parte de administraciones públicas, licitadores y empresas es complejas; es por ello que habitualmente grandes compañías consiguen contratos por encima de pequeños suministradores. TBFY quiere cambiar esto mediante la construcción de un gran grafo de conocimiento que recoja millones

de datos sobre contrataciones públicas en el ámbito europeo. Siguiendo los principios que rigen los datos abiertos, los recursos de TBFY son públicos y de acceso abierto [40].



Figura 9: Dimensiones del grafo de conocimiento de TBFY (mayo de 2021)

Los datos recogidos en el grafo de conocimiento de TBFY son accesibles mediante *endpoints* SPARQL, nodos sobre los cuales se pueden realizar peticiones mediante el protocolo HTTP y obtener respuestas en distintos formatos. Además, TBFY proporciona otras herramientas; el repositorio central [41], una aplicación de búsquedas construida mediante *OptiqueVQS* (*Visual Query System over Ontologies*) [42], una herramienta capaz de encontrar patrones anómalos en datos de contrataciones [43], etc.

Mediante el uso de la aplicación desarrollada en la [posterior sección](#), se generarán datos abiertos y enlazados capaces de ser explotados mediante algunas de las herramientas de TBFY.

3 Software desarrollado

La presente sección describe el sistema software desarrollado, *OCDS_Mapper*, mediante el formato de un documento de diseño software (SDD, por sus siglas en inglés). En esta sección se proporcionará una visión general del sistema, desde su descripción hasta sus funcionalidades, de las que se derivan las características concretas que han sido implementadas.

La documentación del software que se describirá a continuación estará guiada por la norma J-STD-016-1995 [44], un estándar para el ciclo de vida de los procesos de desarrollo de software promulgado por dos de los organismos más relevantes en el área de las tecnologías de la información; la EIA (*Electronic Industries Association*) y el IEEE (*The Institute of Electrical and Electronics Engineers*).

En esta sección que especifica el diseño de la arquitectura del sistema previamente citado, se incluyen los siguientes apartados:

- Consideraciones de diseño
- Estructura de la aplicación
- Diseño de la arquitectura
- Diseño detallado del sistema
- Trazabilidad de requisitos
- Flujo de ejecución
- Cobertura de código

3.1 Consideraciones de diseño

3.1.1 Lenguaje de programación

La implementación del sistema se ha realizado mediante el *framework* .NET Core 5.0.202, usando el lenguaje C# bajo el estándar que describe su sintaxis e interpretación [45].

3.1.2 Arquitectura de componentes orientada a servicios

Con el objetivo de alcanzar la máxima abstracción de tareas posible, el sistema se ha diseñado de tal forma que cada componente realice un servicio concreto, con una funcionalidad bien definida, capaz de integrarse con el resto de componentes del sistema.

El *OCDS_Mapper*, por tanto, estará compuesto de componentes individuales, concretamente 4, descritos en la subsección [Componentes software](#).

3.1.3 Patrón de productor-consumidor

El sistema implementará el patrón de diseño software de productor-consumidor de la siguiente manera:

Las funcionalidades que implementen la recogida de datos, sea de manera local o remota, serán los productores, y las funcionalidades que implementen el procesamiento de los datos, esto es, el mapeado entre estándares, serán los consumidores.

Para asegurar un correcto flujo de recursos entre los componentes que sigan dichos roles se implementan mecanismos de sincronización y exclusión mutua entre ellos.

3.1.4 Mecanismo de consultas

Para realizar las búsquedas de elementos sobre los documentos de licitaciones de la Plataforma de Contratación Pública del Estado se ha utilizado la tecnología *Language Integrated Query* (LINQ), propia del framework .NET.

A través de LINQ se pueden realizar consultas sobre estructuras de datos mediante una sintaxis parecida a la del lenguaje *Structured Query Language* (SQL). En concreto, para el tipo de datos que la aplicación obtiene (extensión .atom, formato XML), la rama de LINQ utilizada es LINQ to XML [46].

Para ejemplificar el funcionamiento de LINQ, se utilizará una sentencia utilizada en esta aplicación. Concretamente, el ejemplo provisto en la figura 10 representa una consulta que busca el elemento raíz (ContractFolderStatus) de cada entrada de un documento de licitaciones.

```
IEnumerable<XElement> query =  
    from node in entry.Elements()  
    where node.Name.LocalName.Equals("ContractFolderStatus")  
    select node;
```

Figura 10: Ejemplo de una sentencia de LINQ

3.1.5 Librerías utilizadas

El sistema utiliza distintas librerías para realizar la implementación de las funcionalidades requeridas. En concreto, estas librerías son:

- AsyncEx [47]: Librería utilizada para el uso de colecciones aptas para la exclusión mutua. Licencia MIT [48].
- FluentAssertions [49]: Librería de ayuda en la codificación de tests. Licencia Apache [50].
- Microsoft.Extensions.Configuration [51]: Librería utilizada para pasar información a la aplicación mediante ficheros de configuración. Licencia MIT [48].
- NLog [52]: Librería para llevar el registro del funcionamiento del sistema. Licencia BSD 3-Clause [53].
- Json.NET [54]: Librería utilizada para la construcción de documentos en formato JSON. Licencia MIT [48].
- xUnit [55]: Librería para realizar el testing del sistema. Licencia Apache [50].

3.2 Estructura de la aplicación

La aplicación *OCDS_Mapper* (una *solución*, en el ámbito de estructuras de proyectos en *Visual Studio* [56]) está estructurada en dos *proyectos*; *src* y *test*, con el código fuente del programa y el módulo de testeo de la aplicación, respectivamente. En la [figura 11](#) se puede visualizar la estructura de directorios de la aplicación.

Además del fichero *Program.cs*, que incluye el punto de entrada de la aplicación, y el fichero de configuración *appsettings.json*, dentro del proyecto del código fuente de la aplicación se han estructurado los ficheros en los siguientes directorios:

- **Examples:** ejemplos de casos de uso de la aplicación.
 - *json*: ejemplos de documentos mapeados por la aplicación.
 - *xml*: ejemplos de documentos de la Plataforma de Contratación Pública.
- **Exceptions:** excepciones propias del funcionamiento de la aplicación.
 - *EmptyMappingRuleException.cs*: se lanza si se provee una regla de mapeado vacía.
 - *InvalidOperationCodeException.cs*: se lanza si no se provee un modo de operación válido.
 - *InvalidParsedElementException.cs*: se lanza si no se encuentra el elemento raíz en una entrada de un documento de licitaciones.
 - *InvalidPathLengthException.cs*: se lanza si se provee una regla de mapeado de longitud inválida.
 - *WrongMappingException.cs*: se lanza si se encuentra un elemento inesperado en el mapeado de los datos.
- **Interfaces:** interfaces de los componentes de la arquitectura.
 - *IMapper.cs*: interfaz del componente de mapeado de datos.
 - *IPackager.cs*: interfaz del componente de empaquetado de datos.
 - *IParser.cs*: interfaz del componente de parseo de datos.
 - *IProvider.cs*: interfaz del componente de provisión de datos.
- **Model:** implementación de los componentes de la arquitectura.
 - *Mapper.cs*: implementación del componente de mapeado de datos.
 - *Packager.cs*: implementación del componente de empaquetado de datos.
 - *Parser.cs*: implementación del componente de parseo de datos.
 - *Provider.cs*: implementación del componente de provisión de datos.
- **Utils:** ficheros con funcionalidades extra.
 - *Document.cs*: tipo de datos que abstrae una instancia de un documento de licitaciones.
 - *EnumCodes.cs*: fichero con los tipos enumerados del programa.
 - *Mappings.cs*: clase estática con las reglas de mapeado.

En cuanto al proyecto de *testing*, en los ficheros *UnitTests.cs* y *IntegrationTests.cs* se pueden encontrar las pruebas unitarias y de integración, respectivamente.

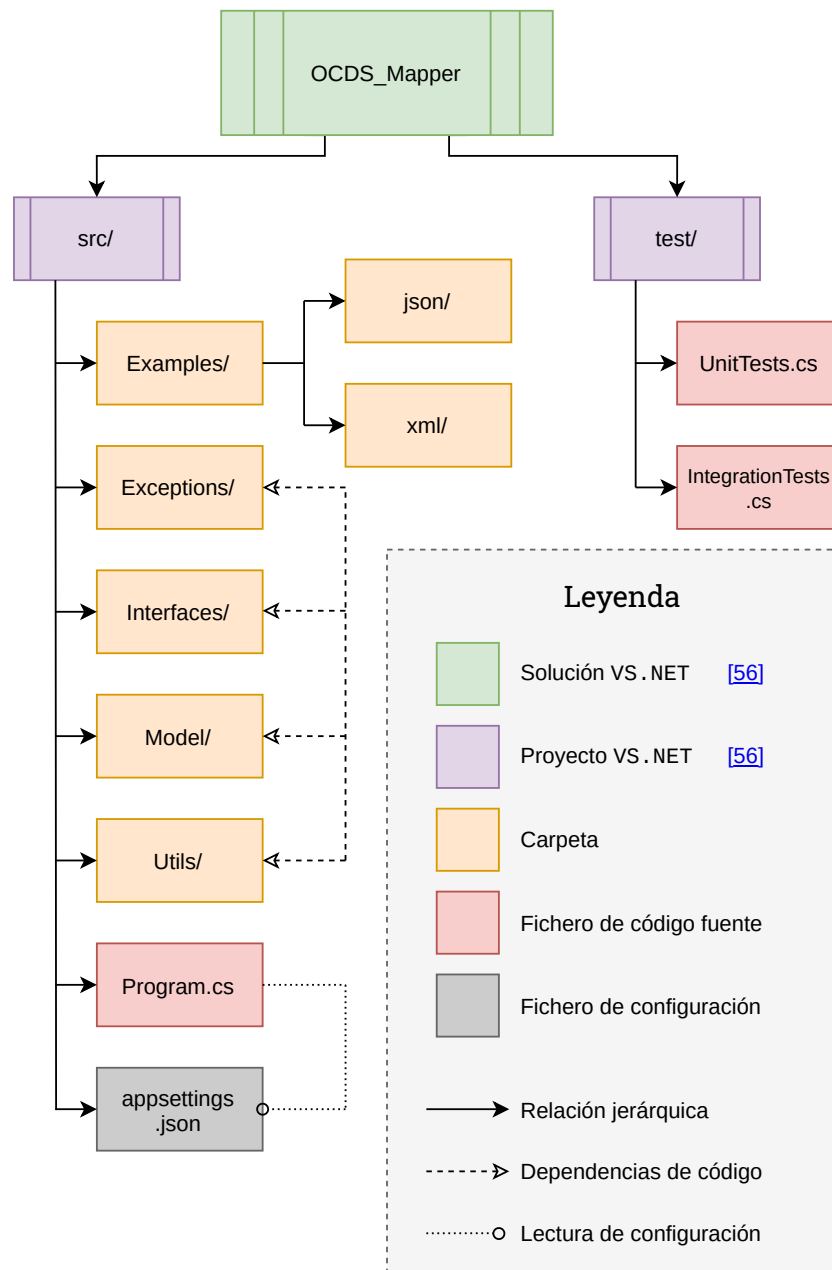


Figura 11: Diagrama de estructura de la aplicación

3.3 Diseño de la arquitectura

3.3.1 Componentes software

Dado que el sistema está diseñado mediante una arquitectura de componentes orientada a servicios, el diseño de la arquitectura se limita a la propia descripción de los componentes que lo integran.

Posteriormente, en el [diseño detallado del sistema](#), se incluirán los diagramas UML individuales de cada componente, acompañados en la sección de [flujo de ejecución](#) por una serie de diagramas de flujo con la representación de la ejecución del sistema.

Nombre del componente	Descripción del componente
<i>Provider</i>	Componente encargado de la provisión de los datos. Admite cuatro modos de operación: proveer el documento más reciente disponible, proveer un documento específico (ya sea local o en forma de URL), proveer los documentos correspondientes únicamente al día anterior, o proveer un flujo continuo de documentos desde el más reciente hasta no encontrar un fichero enlazado más o abortar la aplicación
<i>Parser</i>	Componente encargado del parseo de los datos de los documentos de la Plataforma de Contratación Pública. Provee servicios para extraer los espacios de nombres, los elementos dadas sus rutas, etcétera
<i>Mapper</i>	Componente encargado del mapeado de los datos al estándar OCDS. A través de los elementos XML provistos por el <i>Parser</i> y las reglas especificadas en <i>Mappings</i> , construye el fichero JSON correspondiente al mapeado de cada entrada XML
<i>Packager</i>	Componente encargado del empaquetado de los datos ya mapeados. Publica los paquetes de datos de manera local para su posterior carga y procesado a RDF

Tabla 1: Tabla de descripción de componentes

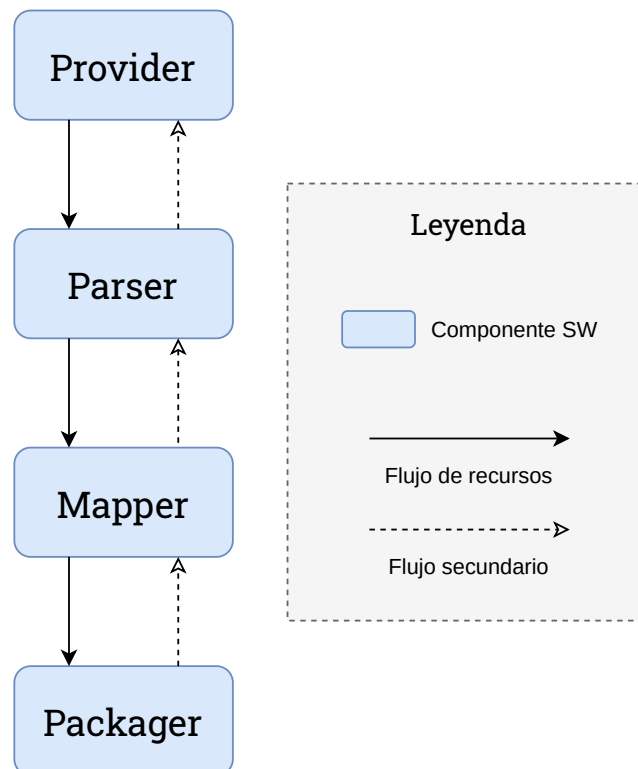


Figura 12: Diagrama de los componentes del sistema

En el diagrama de interrelación de componentes (véase [figura 12](#)) se pueden apreciar dos tipos de flujo distintos.

El principal, el flujo de recursos, representa el funcionamiento básico del sistema: el componente de provisión de datos (*Provider*) suministra los documentos con los datos de contrataciones públicas bajo el esquema CODICE al componente de extracción de datos (*Parser*). Una vez los datos del documento han sido extraídos, por cada entrada del documento de contrataciones se genera una nueva instancia del componente de mapeado (*Mapper*), que implementa las reglas de mapeado descritas en el componente estático *Mappings*. Cuando el documento ha sido totalmente procesado, el componente de empaquetado de datos (*Packager*) finaliza el pipeline del sistema, publicando los datos de manera local.

El flujo secundario representa el uso de métodos entre los distintos componentes, ya sea de manera estática o mediante llamadas a instancias de los componentes.

3.3.2 Especificación de requisitos

Este apartado recoge todos los requisitos, funcionales y de calidad del sistema, que han sido diseñados para la arquitectura del sistema.

Código del requisito	Descripción del requisito
RF-1	El sistema debe ser capaz de descargar documentos de la Plataforma de Contratación Pública
RF-2	El sistema debe ser capaz de recuperar el documento de licitaciones más reciente
RF-3	El sistema debe ser capaz de descargar un documento provisto con una ruta, sea de manera local o remota
RF-4	El sistema debe ser capaz de detectar que una ruta a un documento provisto, local o remota, sea incorrecta
RF-5	El sistema debe ser capaz de otorgar los documentos provistos de manera asíncrona y bloqueante
RF-6	El sistema debe ser capaz de eliminar aquellos documentos que ya hayan sido procesados
RF-7	El sistema debe ser capaz de recuperar los espacios de nombres del XML del documento de licitaciones
RF-8	El sistema debe ser capaz de proveer los documentos enlazados a los que se están procesando
RF-9	El sistema debe ser capaz de recopilar el conjunto de entradas que aparecen en cada documento de licitaciones
RF-10	El sistema debe ser capaz de establecer el elemento raíz de cada documento de licitaciones
RF-11	El sistema debe ser capaz de detectar que a un documento de licitaciones le falta el elemento raíz
RF-12	El sistema debe ser capaz de encontrar un elemento específico en el documento de licitaciones dada su ruta
RF-13	El sistema debe ser capaz de encontrar un conjunto de elementos específicos en el documento de licitaciones dada su ruta

Código del requisito	Descripción del requisito
RF-14	El sistema debe ser capaz de devolver el documento de licitaciones enlazado al que está siendo procesado actualmente
RF-15	El sistema debe ser capaz de encontrar un elemento específico a partir de otro en el documento de licitaciones
RF-16	El sistema debe ser capaz de devolver el valor de un código descrito en un documento de códigos
RF-17	El sistema debe ser capaz de construir un documento en formato JSON con el documento de licitaciones procesado
RF-18	El sistema debe ser capaz de detectar una regla de mapeado vacía
RF-19	El sistema debe ser capaz de detectar una regla de mapeado de longitud inválida
RF-20	El sistema debe ser capaz de detectar una regla de mapeado sin definir
RF-21	El sistema debe ser capaz de construir las colecciones de objetos JSON en el documento siendo mapeado
RF-22	El sistema debe ser capaz de mapear el elemento <code>ContractFolderStatusCode</code> en el elemento <code>tag</code>
RF-23	El sistema debe ser capaz de mapear el elemento <code>ContractFolderID</code> en el elemento <code>OCID</code>
RF-24	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/Name</code> en el elemento <code>tender.title</code>
RF-25	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/BudgetAmount/EstimatedOverallContractAmount</code> en el elemento <code>tender.value</code>
RF-26	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/BudgetAmount/TotalAmount</code> en el elemento <code>budget.amount</code>
RF-27	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/PlannedPeriod/StartDate</code> en el elemento <code>tender.period.startDate</code>
RF-28	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/PlannedPeriod/EndDate</code> en el elemento <code>tender.period.endDate</code>
RF-29	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/PlannedPeriod/DurationMeasure</code> en el elemento <code>tender.period.durationInDays</code>
RF-30	El sistema debe ser capaz de mapear el elemento <code>ProcurementProject/TypeCode</code> en el elemento <code>tender.mainProcurementCategory</code>

Código del requisito	Descripción del requisito
RF-31	El sistema debe ser capaz de mapear el elemento ProcurementProjectLot/ID en los elementos tender.lots.id, tender.items{id, relatedLot}
RF-32	El sistema debe ser capaz de mapear el elemento ProcurementProjectLot/ProcurementProject/BudgetAmount/TotalAmount en el elemento tender.lots.value
RF-33	El sistema debe ser capaz de mapear el elemento ProcurementProjectLot/ProcurementProject/RequiredCommodityClassification/ItemClassificationCode en el elemento tender.items.classification
RF-34	El sistema debe ser capaz de mapear el elemento TenderingProcess/ProcedureCode en el elemento tender.procurementMethod
RF-35	El sistema debe ser capaz de mapear el elemento TenderingProcess/ContractingSystemCode en el elemento tender.procurementMethodDetails_es
RF-36	El sistema debe ser capaz de mapear el elemento TenderingProcess/SubmissionMethodCode en el elemento tender.submissionMethod
RF-37	El sistema debe ser capaz de mapear el elemento TenderingTerms/Language/ID en el elemento tender.submissionMethodDetails
RF-38	El sistema debe ser capaz de mapear el elemento TenderingProcess/AuctionTerms/AuctionConstraintIndicator en el elemento tender.submissionMethod
RF-39	El sistema debe ser capaz de mapear el elemento LocatedContractingParty/Party/PartyName/Name en el elemento parties[i].name
RF-40	El sistema debe ser capaz de mapear el elemento LocatedContractingParty/Party/PartyIdentification/ID en los elementos parties[i].{additionalIdentifiers, id, identifier.{id, schema}, roles}
RF-41	El sistema debe ser capaz de mapear el elemento LocatedContractingParty/Party/PostalAddress/Contact en el elemento parties[i].{address, countryName, contactPoint}
RF-42	El sistema debe ser capaz de mapear el elemento TenderResult/AwardedTenderedProject/ProcurementProjectID en el elemento awards[i].id
RF-43	El sistema debe ser capaz de mapear el elemento TenderResult/ResultCode en el elemento awards[i].status

Código del requisito	Descripción del requisito
RF-44	El sistema debe ser capaz de mapear el elemento <code>TenderResult/WinningParty</code> en los elementos <code>{awards[i].suppliers, parties[j]}</code>
RF-45	El sistema debe ser capaz de mapear el elemento <code>TenderResult/AwardedTenderedProject/LegalMonetaryTotal/PayableAmount</code> en el elemento <code>awards[i].value</code>
RF-46	El sistema debe ser capaz de mapear el elemento <code>TenderResult/ReceivedTenderQuantity</code> en el elemento <code>tender.numberOfTenderers</code>
RF-47	El sistema debe ser capaz de mapear el elemento <code>TenderResult/AwardDate</code> en el elemento <code>awards[i].date</code>
RF-48	El sistema debe ser capaz de mapear el elemento <code>TenderResult/Description</code> en el elemento <code>awards[i].description_es</code>
RF-49	El sistema debe ser capaz de mapear el elemento <code>TenderResult/Contract/ID</code> en el elemento <code>contracts[i].id</code>
RF-50	El sistema debe ser capaz de mapear el elemento <code>TenderResult/StartDate</code> en el elemento <code>contracts[i].period.startDate</code>
RF-51	El sistema debe ser capaz de detectar las ocurrencias de los identificadores para prevenir los duplicados
RF-52	El sistema debe ser capaz de crear un paquete de entrega siguiendo el formato OCDS
RF-53	El sistema debe ser capaz de publicar un paquete de entrega de manera local
RF-54	El sistema debe ser capaz de detectar argumentos incorrectos
RF-55	El sistema debe ser capaz de detectar la falta de argumentos requeridos como el directorio de salida

Tabla 2: Tabla de requisitos funcionales

3.4 Diseño detallado del sistema

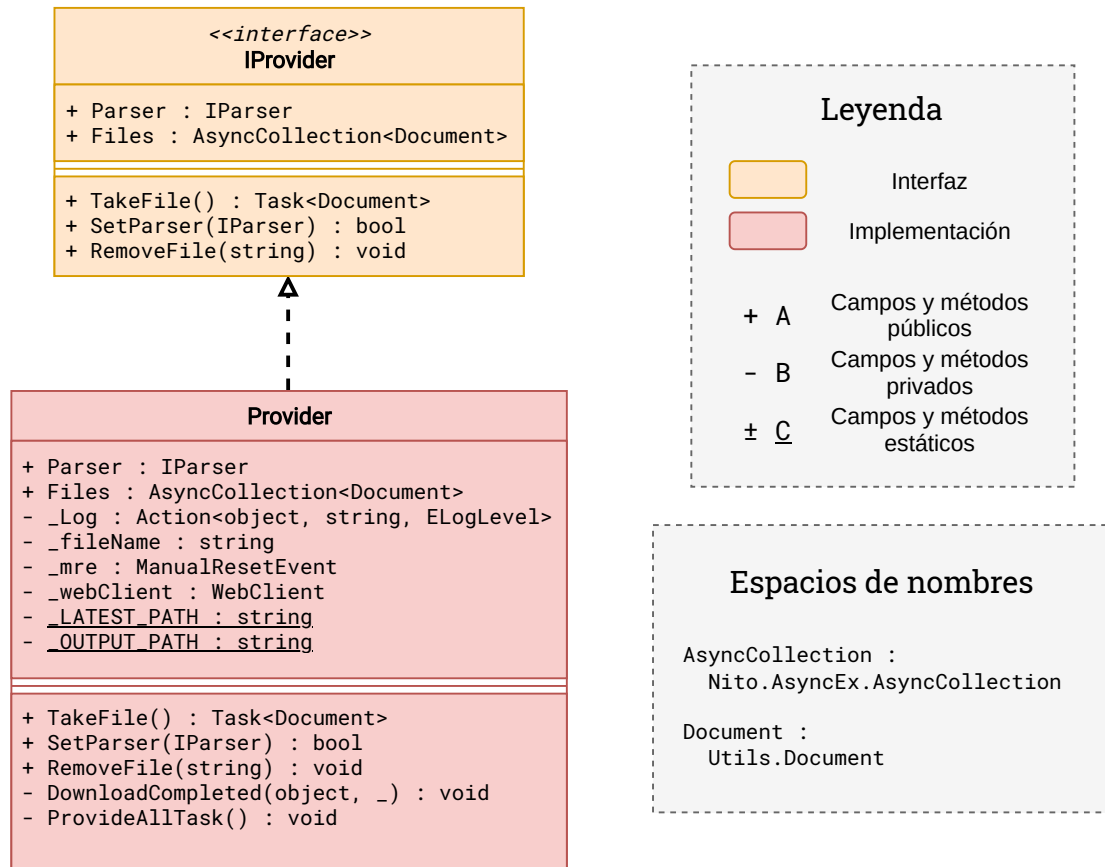
3.4.1 Casos de uso del componente de provisión de datos

3.4.1.1 Caso de uso I: modo de carga local

El componente de provisión de datos debe ser capaz de cargar en memoria para su posterior procesamiento un fichero localizado en el sistema de ficheros local de la máquina ejecutando el proceso. También debe notificar si la ruta al fichero local especificado no es correcta o si en ella no existe dicho fichero.

3.4.1.2 Caso de uso II: modo de descarga remota

El componente de provisión de datos debe ser capaz de descargar para su posterior procesamiento

Figura 13: Diagrama UML del componente *Provider*

un documento de licitaciones desde la Plataforma de Contratación Pública. También debe notificar si la URL no es correcta.

3.4.1.3 Caso de uso III: provisión simple

El componente de provisión de datos debe ser, si así se especifica, capaz de proveer un único documento de licitaciones, el cual podrá ser cargado o descargado de manera local o remota, respectivamente. En el caso de una provisión simple de manera remota, el componente tratará de descargar aquel documento más reciente, cuya URL [57] queda descrita en el portal de datos abiertos del Ministerio de Hacienda [34].

3.4.1.4 Caso de uso IV: provisión continua

El componente de provisión de datos debe ser capaz, si así se especifica, y sólo en el modo de carga remota, de encontrar en el documento de licitaciones siendo descargado el enlace al siguiente documento, con el fin de proveer un flujo continuo de recursos para el procesado en el sistema. El primer documento del flujo será el más reciente, descrito en el apartado anterior.

3.4.1.5 Caso de uso V: provisión diaria

El componente de provisión de datos debe ser capaz, si así se especifica, y sólo en el modo de carga remota, de igual manera que con el caso de uso de la provisión continua, de proveer un flujo de recursos para ser procesados, con la salvedad que dicho flujo se limite exclusivamente al conjunto de documentos publicados en el día anterior.

3.4.2 Casos de uso del componente de parseo de datos

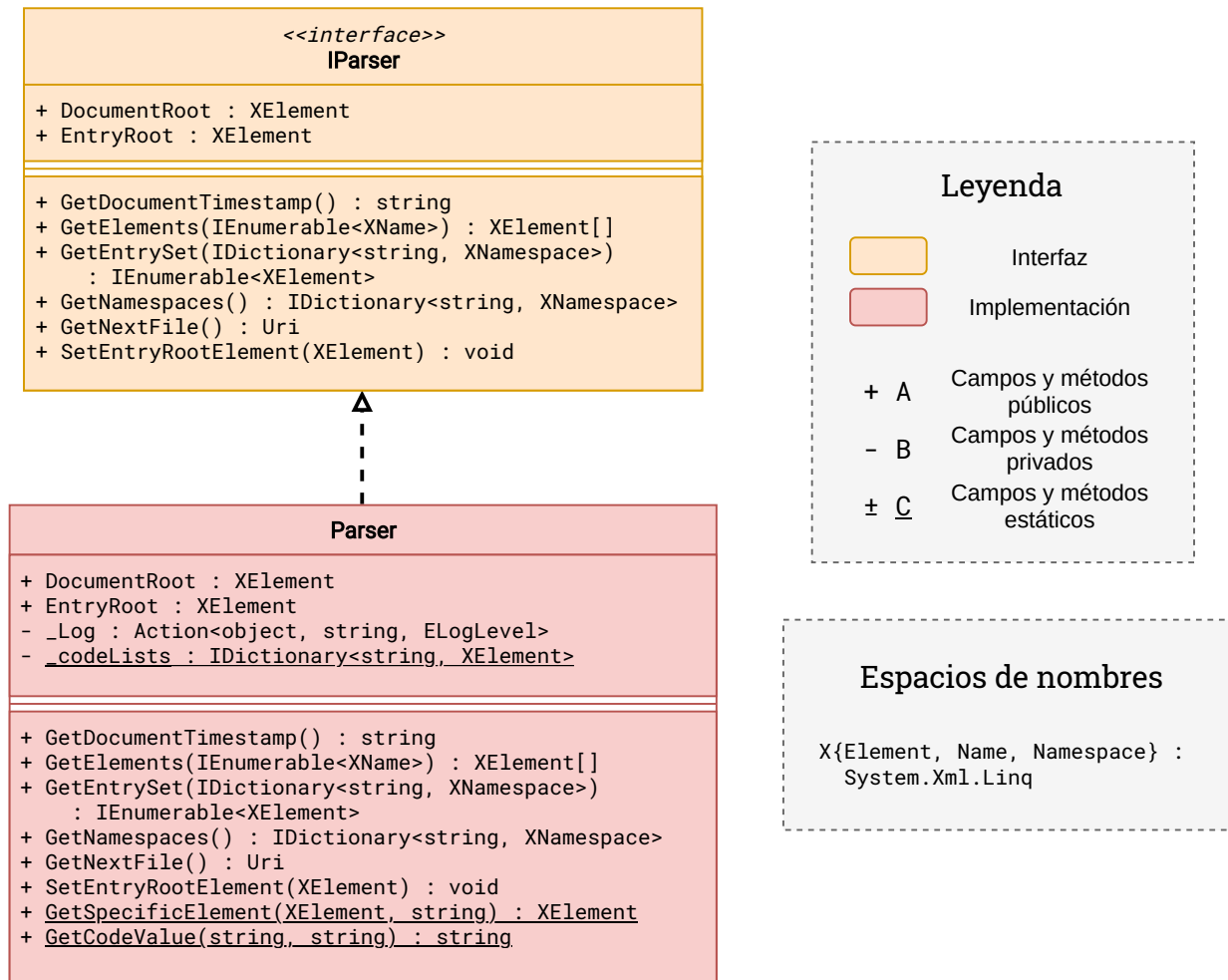


Figura 14: Diagrama UML del componente *Parser*

3.4.2.1 Caso de uso I: extracción de elementos

El componente de parseo de datos debe ser capaz de extraer todos los elementos requeridos dado un documento de licitaciones. Los elementos de interés para ser extraídos son, entre otros, el *timestamp* de la publicación, el conjunto de entradas, el conjunto de espacios de nombres, o la ruta al documento enlazado.

3.4.2.2 Caso de uso II: extracción específica de elementos

El componente de parseo de datos debe ser capaz de, dada una ruta que iterar o un punto del que partir, bajar por los niveles del documento y extraer un elemento específico, con el objetivo de realizar posteriores mapeados lo más independientes posible.

3.4.2.3 Caso de uso III: obtención de códigos

El componente de parseo de datos debe ser capaz de descargar en memoria aquellos documentos de códigos usados por el sistema y poder extraer de ellos los valores necesitados en cada momento.

3.4.3 Casos de uso del componente de mapeado de datos

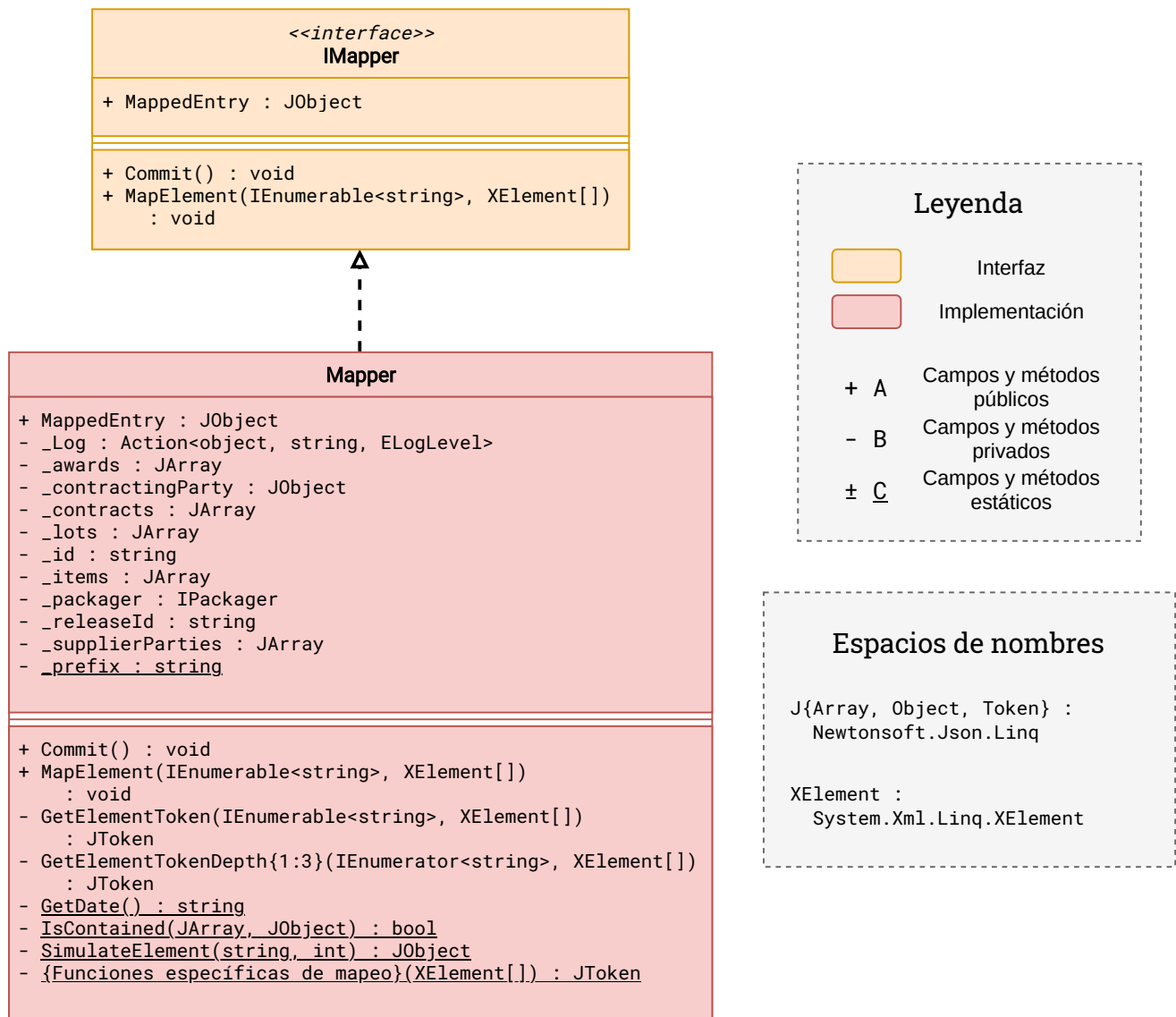


Figura 15: Diagrama UML del componente *Mapper*

3.4.3.1 Caso de uso I: mapeado unitario de elementos

El componente de mapeado de datos debe ser capaz de realizar mapeados unitarios de elementos extraídos de los documentos de licitaciones (esquema CODICE) al esquema OCDS. Para ello, al componente se le indicará tanto el elemento a procesar como la ruta en la que debe almacenar el resultado.

3.4.3.2 Caso de uso II: mapeado múltiple de elementos

El componente de mapeado de datos debe ser capaz de realizar mapeados múltiples de elementos siempre y cuando no pueda realizar mapeados unitarios de dichos elementos. Para ello, al componente se le provea un conjunto de elementos emparentados de la forma más próxima posible. Cuando finalice la ejecución del mapeado, el componente deberá persistir las estructuras de datos en el objeto final para su posterior empaquetado.

3.4.4 Casos de uso del componente de empaquetado de datos

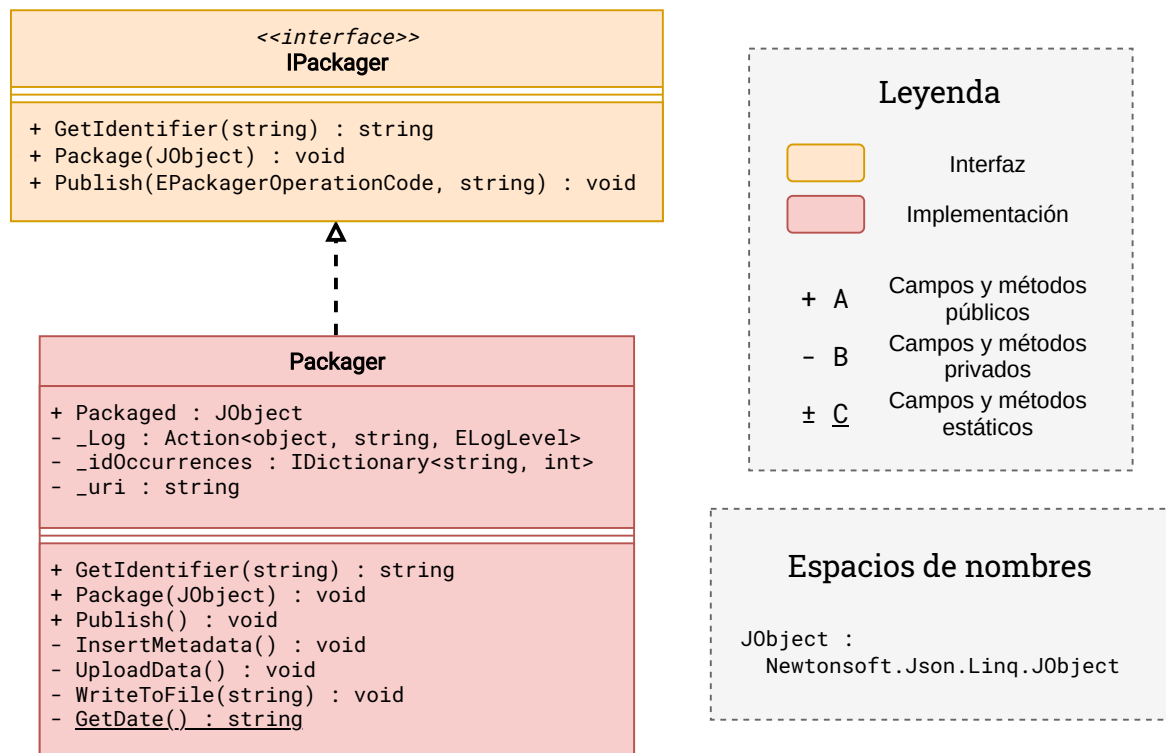


Figura 16: Diagrama UML del componente *Packager*

3.4.4.1 Caso de uso I: empaquetado

El componente de empaquetado de datos debe ser capaz de recopilar las entradas de los documentos procesados que le lleguen por parte de otros componentes del sistema para persistirlos en un fichero de manera local.

3.4.4.2 Caso de uso II: evitar colisiones

El componente de empaquetado de datos debe ser capaz de almacenar temporalmente los identificadores tanto de los propios documentos de licitaciones como de elementos concretos de cada entrada para así evitar colisiones de identificadores no únicos.

3.5 Trazabilidad de requisitos

Código del requisito	Ruta del elemento de prueba
RF-1	UnitTests.ProviderTests.TestConstructor{1:4}
RF-2	UnitTests.ProviderTests.TestConstructor1
RF-3	UnitTests.ProviderTests.TestConstructor{2:3}
RF-4	UnitTests.ProviderTests.TestConstructor4

Código del requisito	Ruta del elemento de prueba
RF-5	UnitTests.ProviderTests.TestTakeFile{1:2}
RF-6	UnitTests.ProviderTests.TestRemoveFile{1:2}
RF-7	UnitTests.ParserTests.TestGetNamespaces
RF-8	IntegrationTests.TestProviderParser
RF-9	IntegrationTests.ParserTests.TestGetEntrySet
RF-10	IntegrationTests.ParserTests.TestSetEntryRootElement1
RF-11	IntegrationTests.ParserTests.TestSetEntryRootElement2
RF-12	IntegrationTests.ParserTests.TestGetElements1
RF-13	IntegrationTests.ParserTests.TestGetElements2
RF-14	UnitTests.ParserTests.TestGetNextFile{1:2}
RF-15	IntegrationTests.ParserTests.TestGetSpecificElement
RF-16	UnitTests.ParserTests.TestGetCodeValue
RF-17	UnitTests.MapperTests.TestConstructor
RF-18	UnitTests.MapperTests.TestMapElement1
RF-19	UnitTests.MapperTests.TestMapElement2
RF-20	UnitTests.MapperTests.TestMapElement3
RF-21	IntegrationTests.MapperTests.TestCommit
RF-22	UnitTests.MapperTests.TagTests
RF-23	UnitTests.MapperTests.OCIDTests
RF-24	UnitTests.MapperTests.TenderTitleTests
RF-25	UnitTests.MapperTests.TenderValueTests
RF-26	UnitTests.MapperTests.BudgetAmountTests
RF-27	UnitTests.MapperTests.TenderPeriodStartDateTests
RF-28	UnitTests.MapperTests.TenderPeriodEndDateTests
RF-29	UnitTests.MapperTests.TenderPeriodDurationInDaysTests
RF-30	UnitTests.MapperTests.TenderMainProcurementCategoryTests
RF-31	UnitTests.MapperTests.TenderLotsIdTests
RF-32	UnitTests.MapperTests.TenderLotsValueTests
RF-33	UnitTests.MapperTests.TenderItemsClassificationTests
RF-34	UnitTests.MapperTests.TenderProcurementMethodTests
RF-35	UnitTests.MapperTests.TenderProcurementMethodDetailsTests
RF-36	UnitTests.MapperTests.TenderSubmissionMethodTests
RF-37	UnitTests.MapperTests.TenderSubmissionMethodDetailsTests
RF-38	UnitTests.MapperTests.TenderSubmissionMethodTests
RF-39	UnitTests.MapperTests.PartiesNameTests

Código del requisito	Ruta del elemento de prueba
RF-40	UnitTests.MapperTests.PartiesIdentifierTests
RF-41	UnitTests.MapperTests.PartiesFieldsTests
RF-42	UnitTests.MapperTests.AwardIdTests
RF-43	UnitTests.MapperTests.AwardStatusTests
RF-44	UnitTests.MapperTests.AwardSupplierTests
RF-45	UnitTests.MapperTests.AwardValueTests
RF-46	UnitTests.MapperTests.TenderNumberOfTenderersTests
RF-47	UnitTests.MapperTests.AwardDateTests
RF-48	UnitTests.MapperTests.AwardDescriptionTests
RF-49	UnitTests.MapperTests.ContractIdTests
RF-50	UnitTests.MapperTests.ContractPeriodStartDateTests
RF-51	UnitTests.PackagerTests.TestGetIdentifier
RF-52	UnitTests.PackagerTests.TestPackage
RF-53	UnitTests.MapperTests.TestPublish
RF-54	IntegrationTests.ProgramTests.TestProgram{1:3}
RF-55	IntegrationTests.ProgramTests.TestProgram4

Tabla 3: Tabla de trazabilidad de requisitos

3.6 Flujo de ejecución

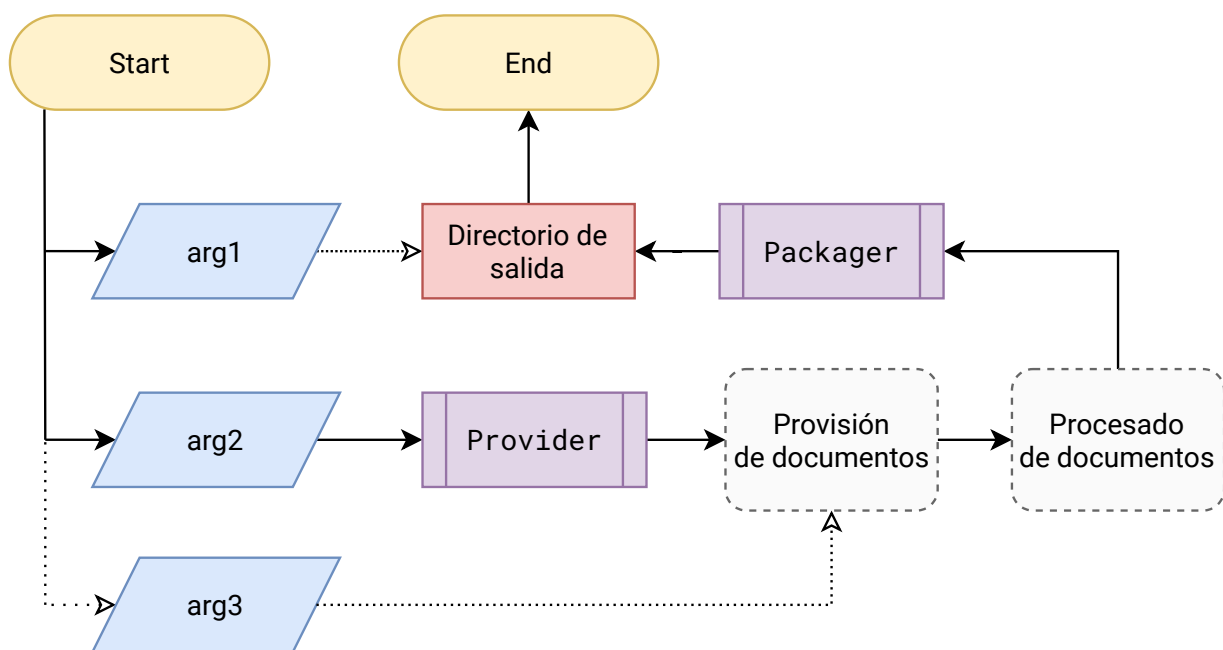


Figura 17: Flujo de ejecución general de la aplicación

En la [figura 17](#) se puede apreciar la representación del flujo principal de la aplicación *OCDS_Mapper*. En primer lugar, cabe destacar la presencia de dos argumentos obligatorios, siendo el primero la ruta del directorio de salida de los ficheros que se generen, y el segundo el descriptor del modo de operación que se pretende ejecutar.

En las dos siguientes figuras se detallan los flujos de ejecución correspondientes a la provisión y al procesamiento de documentos, con el objetivo de simplificar y hacer más clara la representación del funcionamiento del programa.

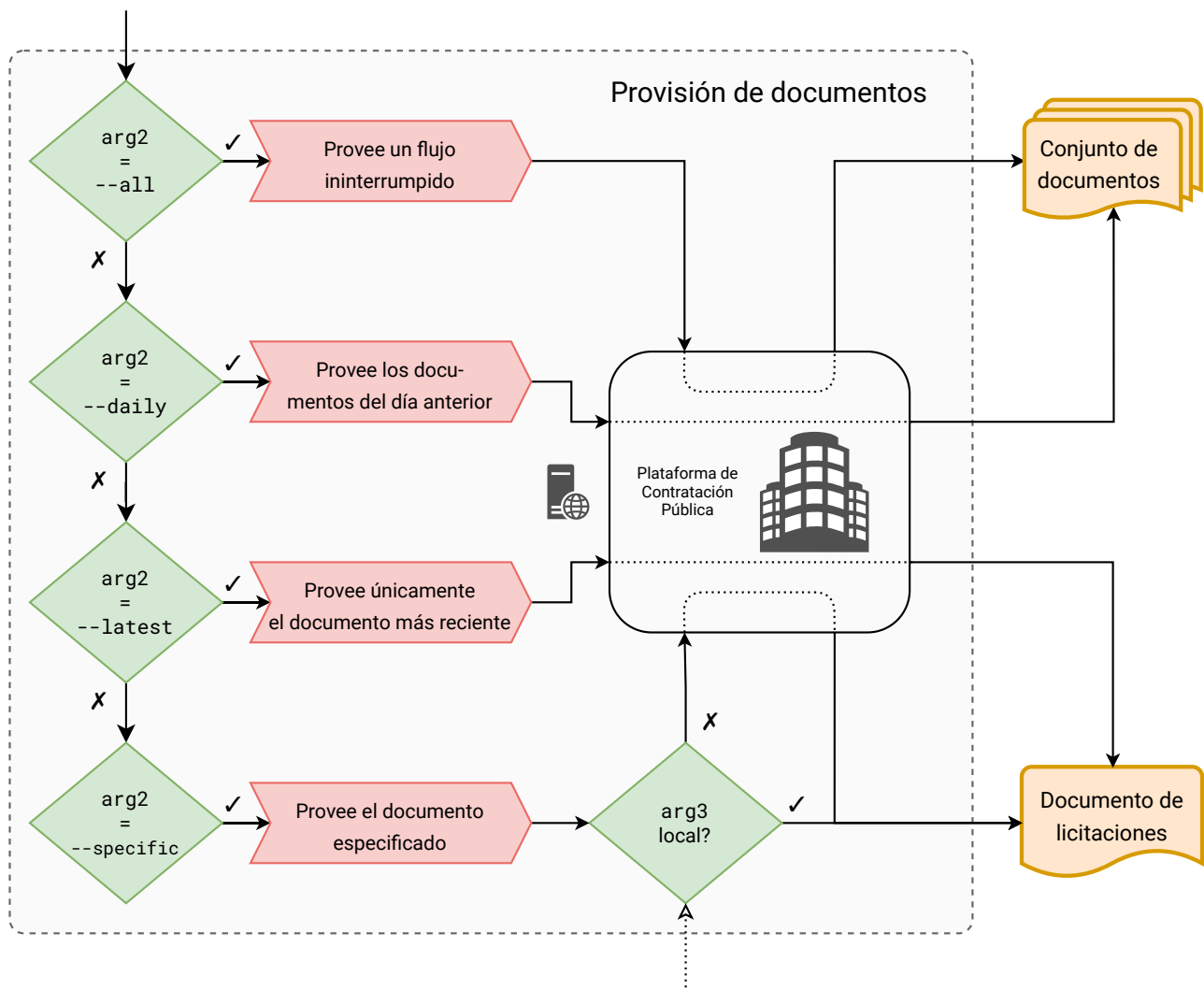


Figura 18: Flujo de ejecución de la provisión de documentos

Mediante la representación del flujo de provisión de documentos se se pueden apreciar los distintos modos de operación del sistema, detallados en secciones previas. Los modos `--all` y `--daily` proveen un conjunto de documentos, mientras que los otros dos modos, `--latest` y `--specific` proveen un único documento.

Este último modo, además, es el único que no accede a la Plataforma de Contratación Pública para descargar documentos si la ruta provista coincide con un archivo local.

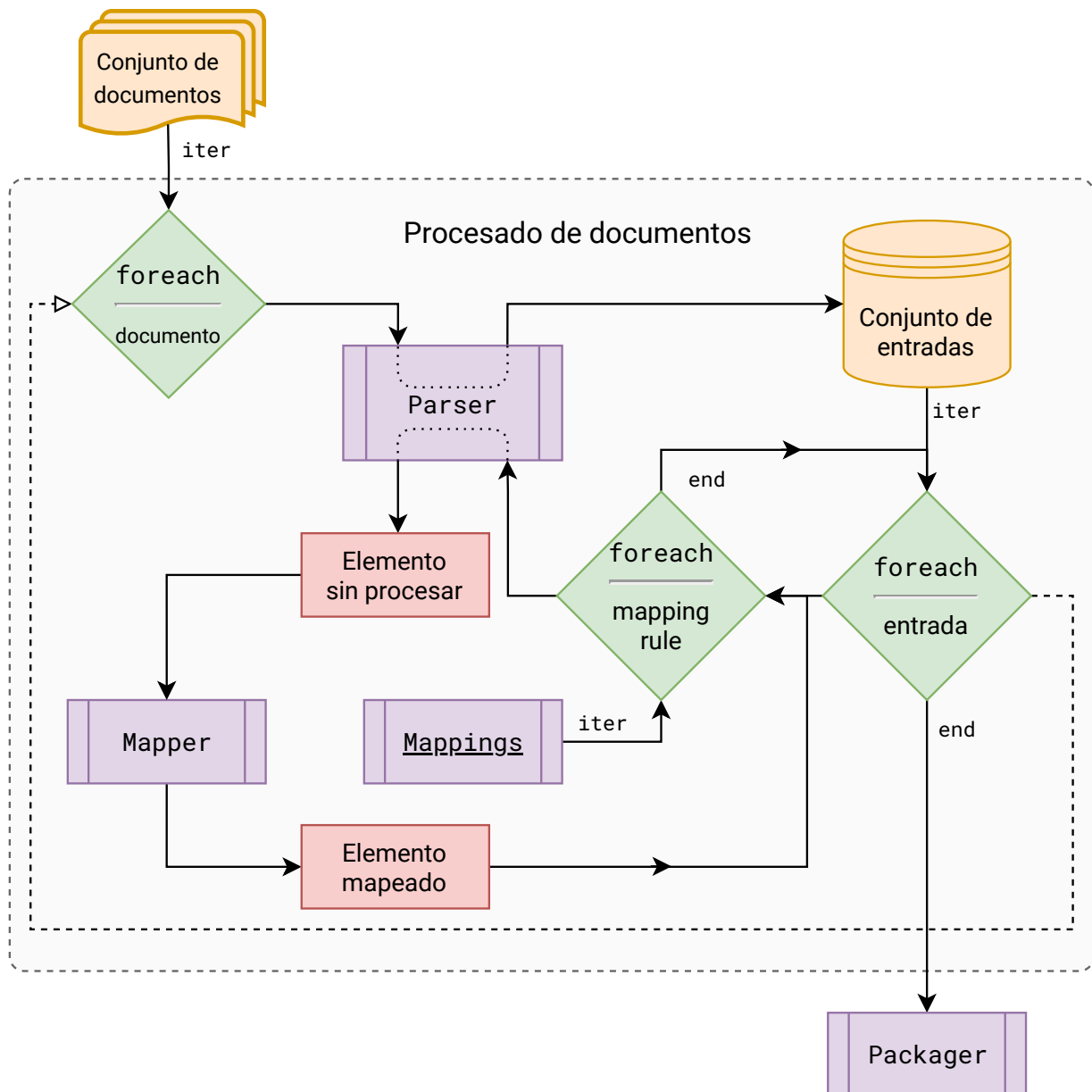


Figura 19: Flujo de ejecución del procesamiento de documentos

Para la representación del flujo de procesamiento de documentos se ha utilizado como *input* un conjunto de documentos en vez de uno único, dado que éste es el caso general y los modos que proveen un solo documento son un caso particular, siendo el conjunto de longitud 1.

Por cada documento de licitaciones, se inicializa un componente de parseo que extrae el conjunto de entradas. El procesamiento se realiza entrada a entrada, y en cada una de ellas se evalúan las reglas de mapeado descritas en la clase estática *Mappings*. Se solicita al componente *Parser* que devuelva el elemento concreto de la entrada descrito por la regla de mapeado siendo evaluada, y es el componente *Mapper* el que realiza la transformación de la información de CODICE a OCDS.

Una vez cada documento ha sido completamente procesado al finalizar de iterar el conjunto de entradas del mismo, se llama al componente de empaquetado para persistir el documento mapeado.

3.7 Cobertura de código

Con el objetivo de proveer un software de calidad, el sistema desarrollado ha sido analizado mediante la plataforma *SonarQube* [58], capaz de realizar una evaluación del código fuente de manera estática para obtener métricas que sirvan para mejorar la calidad y la seguridad del sistema siendo desarrollado.



Figura 20: Métricas registradas por *SonarQube*

Mediante todos los *tests* descritos en la sección de [trazabilidad de requisitos](#) se ha conseguido una cobertura del código del 86.8%, sobre las más de 2600 líneas de código de la aplicación. De acuerdo a la *quality gate* por defecto de *SonarQube* para el lenguaje C#, dicha métrica de cobertura de código es superior al límite (80%).

Entre otras condiciones que se han superado para superar dichas condiciones mínimas, se encuentran la de la duplicidad de líneas (0%, el límite es 3%), y los *ratings* de mantenibilidad, fiabilidad y seguridad en calificación A.

Se han realizado algunas excepciones al análisis estático del código, cuyas justificaciones se presentan a continuación:

- Regla S125: *"Sections of code should not be commented out"* [59]. Ignorada debido a que se generaba la alerta en comentarios informativos del funcionamiento de los métodos.
- Regla S1168: *"Empty arrays and collections should be returned instead of null"* [60]. Teniendo en cuenta la implementación del componente *Mapper*, los métodos que no devuelven un objeto *JsonObject* devuelven *null* debido a que el objeto *JsonObject* vacío representa un elemento JSON vacío, lo cual no es equivalente a *null*.
- Regla S3776: *"Cognitive Complexity of functions should not be too high"* [61]. Por el motivo descrito en el apartado anterior, debido a que cada regla de mapeado está implementada en un sólo método por motivos del modelado del sistema, se ha ignorado esta regla en dichos métodos.

4 Correspondencias entre CODICE y OCDS

En esta sección se describirán las correspondencias que se han establecido entre los elementos del estándar CODICE y el estándar OCDS. Dichas correspondencias han sido las implementadas en el [software desarrollado](#).

No todos los elementos de CODICE han sido mapeados, ni todos los campos de OCDS han sido cubiertos, pero sí se han considerado aquellos elementos de mayor interés.

4.1 Datos generales del expediente

Elementos correspondientes a información genérica del expediente de licitaciones.

4.1.1 Estado

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cbc-place-ext:ContractFolderStatusCode`
- **Elemento en el esquema OCDS:** `tag`
- **Comentarios:** La lista de posibles códigos del elemento en CODICE se encuentra en el siguiente enlace [62]. La lista análoga de códigos en el esquema OCDS se puede consultar en el siguiente enlace [63]. La correspondencia entre ambas listas de códigos se ha establecido de la siguiente manera:
 - PRE (Anuncio previo) → `planning`
 - PUB (En plazo) → `tender`
 - EV (Pendiente de adjudicación) → `tender`
 - ADJ (Adjudicada) → `award`
 - RES (Resuelta) → `contract`
 - ANUL (Anulada) → `awardCancellation`

4.1.2 Número de expediente

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cbc:ContractFolderID`
- **Elemento en el esquema OCDS:** `ocid`
- **Comentarios:** En el esquema de OCDS, los procesos de contratación deben identificarse mediante un código unívoco que será idéntico en cualquier posterior entrega del mismo proceso de contratación. Para asegurar que dichos códigos no puedan colisionar, OCDS provee a los publicadores prefijos que concatenar a los identificadores internos para así asegurar la creación de identificadores globales únicos. Debido a que OCDS no provee prefijos válidos a particulares, se ha utilizado el prefijo otorgado al Ayuntamiento de Zaragoza (`ocds-1xraxc`) como *placeholder* para superar la herramienta de validación de datos de OCDS [64]. Para cambiar dicho parámetro, sólo hace falta modificar el campo `ID_prefix` del fichero de configuración del *OCDS_Mapper*, ubicado en la ruta `OCDS_Mapper/src/appsettings.json` (véase [figura 11](#)).

4.1.3 Objeto del contrato

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProject/
cbc:Name`
- **Elemento en el esquema OCDS:** `tender.title`

4.1.4 Valor estimado e importe de licitación

- **Elementos en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProject/
cac:BudgetAmount/
{cbc:EstimatedOverallContractAmount,
cbc:TotalAmount}`
- **Elementos en el esquema OCDS:** `EstimatedOverallContractAmount
→ tender.value
TotalAmount
→ planning.budget.amount`

4.1.5 Duración del contrato

- **Elementos en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProject/
cac:PlannedPeriod/
{cbc:StartDate,
{cbc:EndDate,
cbc:DurationMeasure}}`
- **Elementos en el esquema OCDS:** `StartDate
→ tender.tenderPeriod.startDate
EndDate
→ tender.tenderPeriod.endDate
DurationMeasure
→ tender.tenderPeriod.durationInDays`
- **Comentarios:** La duración prevista del contrato puede expresarse mediante una fecha de inicio y una de final, o bien mediante una fecha de inicio y una duración prevista, cuyos valores pueden ser expresados mediante días (DAY), meses (MON) o años (ANN).

4.1.6 Tipo de contrato

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProject/
cbc:TypeCode`
- **Elemento en el esquema OCDS:** `tender.mainProcurementCategory`
- **Comentarios:** La lista de posibles valores que pueden tomar los códigos que describen los tipos de contrato en CODICE pueden encontrarse en el siguiente enlace [65]. La lista análoga de códigos en el esquema OCDS se puede consultar en el siguiente enlace [66]. La correspondencia entre ambas listas de códigos se ha establecido de la siguiente manera:

- 1 (Suministros) → `goods`
- 2 (Servicios) → `services`
- 3 (Obras) → `works`
- 21 (Gestión de servicios públicos) → `services`
- 22 (Gestión de servicios) → `services`
- 31 (Concesión de obras públicas) → `works`
- 32 (Concesión de obras) → `works`

4.2 Lotes

El uso de lotes en el esquema OCDS viene otorgado por la extensión de mismo nombre [68].

4.2.1 Número de lote

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProjectLot/
cbc:ID`
- **Elementos en el esquema OCDS:** `tender.lots[i].id
tender.items[i].id
tender.items[i].relatedLot`
- **Comentarios:** En todos los apartados de esta sección se utilizará la notación `lots[i]` e `items[i]` para referirse a los *i*-ésimos objetos representando el lote y el artículo, respectivamente, dentro de las colecciones de éstos. El campo `relatedLot` se utilizará para enlazar el artículo con su lote.

4.2.2 Objeto del lote

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProjectLot/
cac:ProcurementProject/
cbc:Name`
- **Elemento en el esquema OCDS:** `tender.lots[i].name`

4.2.3 Importe del lote

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProjectLot/
cac:ProcurementProject/
cac:BudgetAmount/
cbc:TotalAmount`
- **Elemento en el esquema OCDS:** `tender.lots[i].value`

4.2.4 Clasificación CPV

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:ProcurementProjectLot/
cac:ProcurementProject/
cac:RequiredCommodityClassification/
cbc:ItemClassificationCode`
- **Elemento en el esquema OCDS:** `tender.lots[i].classification`
- **Comentarios:** El sistema de clasificación de bienes y servicios utilizado en CODICE, CPV [69], es un estándar válido en OCDS, indicándolo en el campo `classification.schema`.

4.3 Procesos de licitación

Elementos correspondientes a los procedimientos de contratación.

4.3.1 Tipo de procedimiento

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderingProcess/
cbc:ProcedureCode`
- **Elemento en el esquema OCDS:** `tender.procurementMethod`
- **Comentarios:** La lista de posibles códigos del elemento en CODICE se encuentra en el siguiente enlace [70]. La lista análoga de códigos en el esquema OCDS se puede consultar en el siguiente enlace [71]. La correspondencia entre ambas listas de códigos se ha establecido de la siguiente manera:
 - 1 (Abierto) → `open`
 - 2 (Restringido) → `selective`
 - 3 (Negociado sin publicidad) → `limited`
 - 4 (Negociado con publicidad) → `limited`
 - 5 (Diálogo competitivo) → `open`
 - 6 (Contrato menor) → `open`
 - 7 (Derivado de acuerdo macro) → `selective`
 - 8 (Concurso de proyectos) → `open`
 - 9 (Abierto simplificado) → `open`
 - 10 (Asociación para la innovación) → `limited`
 - 11 (Derivado de asociación para la innovación) → `limited`
 - 12 (Basado en un sistema dinámico de adquisición) → `open`
 - 13 (Licitación con negociación) → `open`
 - 100 (Normas internas) → `limited`

4.3.2 Sistema de contratación

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderingProcess/
cbc:ContractingSystemCode`
- **Elemento en el esquema OCDS:** `tender.procurementMethodDetails_es`
- **Comentarios:** Este elemento describe si el sistema de contratación se trata de un contrato, de un acuerdo marco, o de un sistema dinámico de adquisición. Con el sufijo `es` se especifica que el campo muestra su información en castellano, tal y como aparece en el documento de códigos [72] de CODICE.

4.3.3 Presentación de la oferta

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderingProcess/
cbc:SubmissionMethodCode`
- **Elemento en el esquema OCDS:** `tender.submissionMethod`
- **Comentarios:** La lista de posibles códigos del elemento en CODICE se encuentra en el siguiente enlace [73]. La lista análoga de códigos en el esquema OCDS se puede consultar en el siguiente enlace [74]. La correspondencia entre ambas listas de códigos se ha establecido de la siguiente manera:
 - 1 (Electrónica) → `[electronicSubmission]`
 - 2 (Manual) → `[written]`
 - 3 (Manual y/o Electrónica) → `[electronicSubmission, written]`

4.3.4 Idioma de presentación de la oferta

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderingTerms/
cac:Language/
cbc:ID`
- **Elemento en el esquema OCDS:** `tender.submissionMethodDetails`

4.3.5 Licitación son subasta electrónica

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderingProcess/
cac:AuctionTerms/
cbc:AuctionConstraintIndicator`
- **Elemento en el esquema OCDS:** `tender.submissionMethod`
- **Comentarios:** Si en el elemento `AuctionConstraintIndicator` está indicado el valor `true`, significa que la licitación estará sujeta a subasta electrónica, por lo que el valor `electronicAuction` de la lista de códigos `submissionMethod` [73] se sobrescribe en el campo `tender.submissionMethod`, que ya había sido inicializado por el elemento [presentación de la oferta](#).

4.4 Entidades adjudicadoras

Elementos correspondientes a información acerca de los adjudicadores.

4.4.1 Órgano de contratación

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac-place-ext:LocatedContractingParty/
cac:Party/
cac:PartyName/
cbc:Name`
- **Elemento en el esquema OCDS:** `parties[i].name`
- **Comentarios:** En todos los apartados de esta sección se utilizará la notación `parties[i]` para referirse al mismo objeto representando a la entidad adjudicadora, debido a que `parties` es una colección de objetos representando a todas las partes involucradas.

4.4.2 Ubicación orgánica

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac-place-ext:LocatedContractingParty/
cac:Party/
cac:PartyIdentification/
cbc:ID`
- **Elementos en el esquema OCDS:** `parties[i].id
parties[i].identifier.schema
parties[i].identifier.id
parties[i].additionalIdentifiers
parties[i].roles`
- **Comentarios:** Siguiendo la directriz del estándar OCDS sobre los esquemas de identificación de organizaciones [75], estos elementos se procesan dependiendo de su esquema: DIR3 [76] o NIF [77] (RMC). Adicionalmente, si `PartyIdentification` contiene ambos identificadores, se utilizará el campo `additionalIdentifiers` para mapear la máxima cantidad de información. Por último, en este elemento de mapeado se ha añadido el rol de la parte adjudicadora (`procuringEntity`).

4.4.3 Otros campos

- **Elementos en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac-place-ext:LocatedContractingParty/
cac:Party/
cac:PostalAddress
cac:Contact`
- **Elementos en el esquema OCDS:** `parties[i].address
parties[i].countryName
parties[i].contactPoint`

4.5 Resultado del procedimiento

Elementos referidos al resultado de la licitación.

4.5.1 Identificador

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderResult/
cac:AwardedTenderedProject/
cbc:ProcurementProjectLotID`
- **Elemento en el esquema OCDS:** `awards[i].id`
- **Comentarios:** En todos los apartados de esta sección se utilizará la notación `awards[i]` para referirse al mismo objeto representando el resultado de un procedimiento de adjudicación dentro de un mismo proceso de contratación.

4.5.2 Resultado

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderResult/
cbc:ResultCode`
- **Elemento en el esquema OCDS:** `awards[i].status`
- **Comentarios:** La lista de posibles códigos del elemento en CODICE se encuentra en el siguiente enlace [78]. La lista análoga de códigos en el esquema OCDS se puede consultar en el siguiente enlace [79]. La correspondencia entre ambas listas de códigos se ha establecido de la siguiente manera:
 - 1 (Adjudicado provisionalmente) → `pending`
 - 2 (Adjudicado definitivamente) → `active`
 - 3 (Desierto) → `cancelled`
 - 4 (Desistimiento) → `unsuccessful`
 - 5 (Renuncia) → `unsuccessful`
 - 6 (Desierto provisionalmente) → `pending`
 - 7 (Desierto definitivamente) → `cancelled`
 - 8 (Adjudicado) → `active`
 - 9 (Formalizado) → `active`
 - 10 (Licitador mejor valorado: requerimiento de documentación) → `active`

4.5.3 Identidad del adjudicatario

- **Elementos en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderResult/
cac:WinningParty/
{cac:PartyIdentification,
cac:PartyName}`

- **Elementos en el esquema OCDS:** `awards[i].suppliers`
`parties[j]`
- **Comentarios:** Debido a que este mapeado introduce en el documento la identidad del adjudicatario, también debe introducir en el campo `parties` la referencia a dicha parte involucrada. De manera análoga al mapeado que realiza [ubicación orgánica](#), el campo `identifier` tanto en el campo `suppliers` como en la colección `parties` detecta si la identificación de la entidad es del tipo DIR3 o NIF, indicándolo en el campo `schema`. Por último, este mapeado también introduce en la referencia al adjudicatario en `parties` en el campo `roles` el valor `supplier`, tal y como se indica en la lista de códigos de OCDS sobre roles de las partes involucradas [80].

4.5.4 Importe de adjudicación

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/`
`cac:TenderResult/`
`cac:AwardedTenderedProject/`
`cac:LegalMonetaryTotal/`
`cbc:PayableAmount`
- **Elemento en el esquema OCDS:** `awards[i].value`

4.5.5 Número de licitadores participantes

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/`
`cac:TenderResult/`
`cbc:ReceivedTenderQuantity`
- **Elemento en el esquema OCDS:** `tender.numberOfTenderers`
- **Comentarios:** Dado que OCDS no provee el campo `numberOfTenderers` dentro de la colección de `awards`, este mapeado sólo se realizará en el caso de que el proceso de adjudicación sólo contenga un lote, es decir, la longitud de la colección `awards` sea 1 y, por tanto, `numberOfTenderers` refleje efectivamente el número de participantes.

4.5.6 Fecha de la adjudicación

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/`
`cac:TenderResult/`
`cbc:AwardDate`
- **Elementos en el esquema OCDS:** `awards[i].date`

4.5.7 Descripción de la adjudicación

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/`
`cac:TenderResult/`
`cbc:Description`
- **Elemento en el esquema OCDS:** `awards[i].description_es`
- **Comentarios:** Como sucedía en el elemento [sistema de contratación](#), se hace uso del sufijo `_es` para indicar que el campo contendrá su información en castellano.

4.6 Información sobre el contrato

Elementos relativos a los contratos adjudicados.

4.6.1 Identificador

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderResult/
cac:Contract/
cbc:ID`
- **Elemento en el esquema OCDS:** `contracts[i].id`
- **Comentarios:** En todos los apartados de esta sección se utilizará la notación `contracts[i]` para referirse al mismo objeto representando un contrato formalizado dentro de un mismo proceso de contratación.

4.6.2 Fecha de entrada en vigor

- **Elemento en el esquema CODICE:** `cac-place-ext:ContractFolderStatus/
cac:TenderResult/
cbc:StartDate`
- **Elemento en el esquema OCDS:** `contracts[i].period.startDate`

5 Generación de datos enlazados

Una vez se han generado los datos de contrataciones mediante el [software desarrollado](#) de acuerdo a las Correspondencias establecidas entre CODICE y OCDS, sólo queda el último paso para completar la línea de ejecución: la publicación como *linked data*.

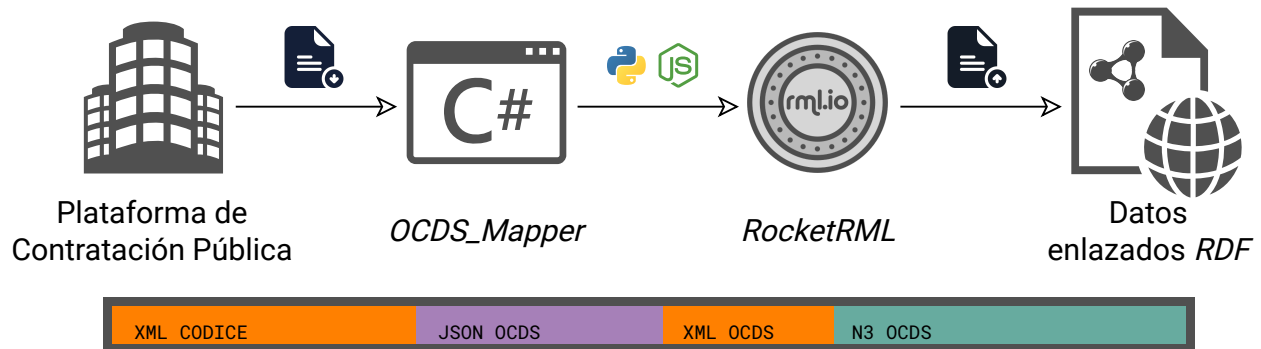


Figura 21: *Pipeline* y formato de los datos en las distintas etapas

En la [figura 21](#) se representa el *pipeline* completo, además del formato de los datos a lo largo de las diferentes fases del procesamiento. Los datos de licitaciones originales que provienen de la Plataforma de Contratación Pública están en el formato XML y siguiendo el esquema CODICE. Una vez son procesados por el sistema *OCDS_Mapper*, éstos pasan al esquema OCDS y al formato JSON. Tras ser mapeados, se transforman mediante un sencillo script en el lenguaje Python (accesible en el repositorio del trabajo [81]) al formato XML para su procesamiento de reglas RML. Éste se realiza haciendo uso de la librería *RocketRML* [82] (licencia CC-BY-SA 4.0 [83]), ejecutada en el entorno Node.js, para obtener finalmente los datos enlazados RDF.

Podría parecer que la transformación a JSON en medio del *pipeline* para posteriormente volver al formato XML inicial es innecesaria, pero esto se hace debido a que OCDS requiere JSON como formato de sus datos, mientras que las reglas de mapeado de RML utilizadas están declaradas haciendo uso de XPath, necesitando XML como formato de entrada.

5.1 Validación de los datos

Previamente a poder realizar la transformación a datos enlazados RDF, es necesario validarlos para poder aseverar la correcta adecuación al estándar OCDS de los mismos. Para ello se ha hecho uso de la herramienta de revisión de OCDS [64] con una batería de múltiples documentos, pero sólo se mostrará el reporte de uno cualquiera en las siguientes figuras.

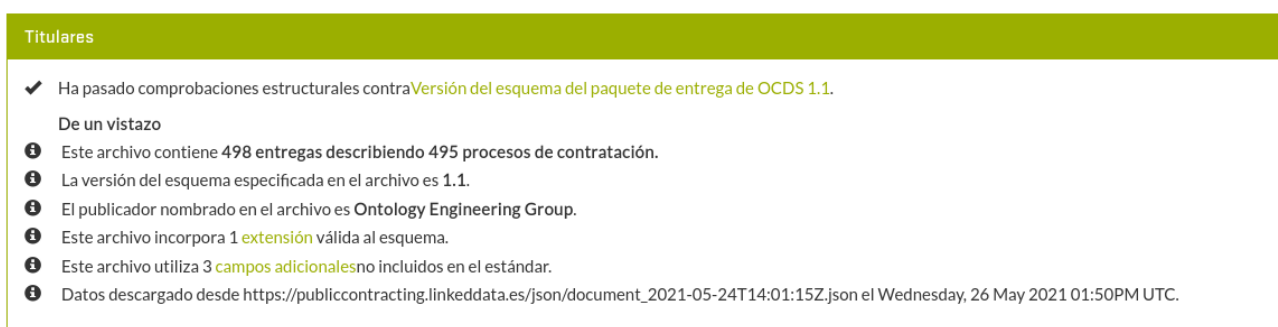


Figura 22: Reporte general de la herramienta de revisión de OCDS

Información de Campos Clave		
Tipos de iniciación: tender Etiquetas: contract: 128 award: 142 tender: 223 awardCancellation: 1 planning: 4 Planificación: 495 Licitaciones: 495 Adjudicaciones: Total: 408 OCIDs únicas: 268 Contratos: Total: 181 OCIDs únicas: 126 Contratos sin adjudicaciones: 14 Implementación: Total: 0 OCIDs únicas: 0	Apertura de entrega: 24 May 2021, 09:38 (UTC) Última entrega: 24 May 2021, 12:32 (UTC) Apertura de inicio de licitación: 1 Dic 2018, 00:00 (UTC) Último inicio de licitación: 1 Abr 2022, 00:00 (UTC) Apertura de adjudicación: 20 Abr 2018, 00:00 (UTC) Última adjudicación: 24 May 2021, 00:00 (UTC) Apertura de inicio del contrato: 19 May 2021, 00:00 (UTC) Último inicio del contrato: 1 Jul 2021, 00:00 (UTC)	Número Total de Organizaciones Distintas: 313 con detalles de dirección: 0 con puntos de contacto: 0 que tienen identificadores: 0 utilizando esquemas: Roles de la Organización: Compradores: 0 Proveedores: 313 Entidades Contratantes: 0 Licitadores: 0 Número Total de Artículos: 163 IDs de artículos únicos: 29 Esquemas de ID de artículos: CPV Tipos de Artículos: Licitación: 163 Adjudicación: 0 Contrato: 0 Idiomas: es Divisas: EUR

Figura 23: Reporte específico de la herramienta de revisión de OCDS

El reporte mostrado como ejemplo es accesible en el siguiente enlace [84] hasta el día 24 de agosto de 2021, gracias al acceso ofrecido durante 90 días de los reportes ejecutados por la herramienta de revisión.

5.2 Reglas RML

Para poder realizar la transformación a *linked data*, se ha utilizado el lenguaje RML (RDF Mapping Language) [85], cuyo propósito es mapear información de muy diversas fuentes de información al sistema RDF.

Con respecto al procesamiento de las reglas para producir datos RDF, se ha valorado la utilización de dos distintos métodos de ejecución, *RMLMapper* [86] y *RocketRML* [82]. Sin embargo, debido a la grandísima diferencia de velocidad entre ambas que se puede apreciar en la figura 24, favorable a *RocketRML*, se ha decidido utilizar este software.

Comparativa de tiempos de ejecución

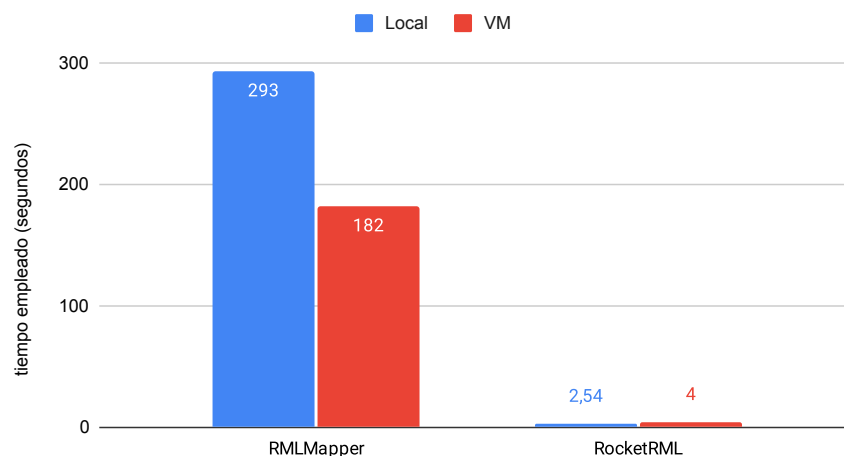


Figura 24: Comparativa de tiempos entre *RMLMapper* y *RocketRML*

Si bien es cierto que la construcción de las reglas se ha ajustado a las necesidades específicas de este proyecto, la mayor parte de éstas son una reutilización de una batería de reglas del proyecto TBFY, accesibles en el siguiente enlace [87]. Sobre ellas se han realizado algunas modificaciones *ad-hoc*, como el cambio de las URIs base (de <http://data.tbfy.eu> a <https://publiccontracting.linkeddata.es>), o cambios en algunos *TriplesMap*. Además, se han eliminado aquellas reglas que quedan fuera de las correspondencias por motivos de legibilidad de las mismas y agilidad en el procesamiento. El conjunto final de reglas utilizado en el sistema desplegado es accesible en el siguiente enlace del repositorio del trabajo [88].

5.3 Consultas sobre los datos generados

Para comprobar que la integridad de los datos enlazados generados se ha preservado tras su procesamiento mediante las reglas RML, se ha confeccionado una batería de consultas SPARQL que asevera que toda la información codificada en dichas reglas se haya preservado.

La implementación de las consultas se ha realizado mediante un sencillo script en Python (accesible en el repositorio [89]) haciendo uso de la librería *RDFLib* (licencia BSD 3-Clause [53]). El conjunto de *queries* SPARQL se encuentra en el Anexo III.

En el repositorio del trabajo se localizan ficheros de ejemplo de la ejecución completa del *pipeline* del sistema, desde los documentos originales [90] hasta los ficheros de texto resultantes de la ejecución de las consultas descritas en esta sección [91], pasando por su primer procesado a JSON [92] y su versión RDF [93].

5.4 Despliegue del sistema



Finalmente, el *pipeline* completo, desde la adquisición de documentos de la Plataforma de Contratación Pública hasta la publicación de los datos enlazados ha sido desplegado en una máquina virtual provista para este propósito gracias al *Ontology Engineering Group* [94] de la UPM.

Dicha máquina virtual está compuesta por 2 procesadores Intel Xeon CPU E5-2650 v3 @ 2.30GHz, con 40 *cores* por procesador. Cuenta asimismo con 4096 MBs de RAM de tecnología RDIMM y velocidad 2133 MT/s, y su almacenamiento es de 19 GBs. El software de virtualización que soporta la infraestructura es QEMU-KVM.

En la máquina virtual, mediante el servicio Unix *cron*, se ha configurado un *script* de ejecución diaria que realiza todas las etapas del procesamiento: en primer lugar, ejecuta el software *OCDS_Mapper* con el modo de operación *--daily*, dado que el servicio se ejecuta diariamente, de medianoche, y se pretenden realizar procesamientos de los datos publicados en el día anterior. Una vez se completa esa fase y se realiza el procesado a RML descrito en esta sección, el *script* lleva al directorio `/var/www/html` dos conjuntos de ficheros: los JSON que devuelve el *OCDS_Mapper*, de interés por ser documentos válidos con respecto al estándar OCDS, y los correspondientes N-Triples (RDF) que salen de la ejecución del *RocketRML*.

Una vez se han colocado los ficheros de interés en los directorios descritos, éstos quedan completamente accesibles a través de la página web <https://publiccontracting.linkeddata.es>, para su acceso y consulta.

Index of /

Name	Last modified	Size	Description
 json/	2021-05-26 02:05	-	
 rdf/	2021-05-26 02:05	-	

Apache/2.4.41 (Ubuntu) Server at publiccontracting.linkeddata.es Port 80

Figura 25: Captura de la web <https://publiccontracting.linkeddata.es>

Index of /json

Name	Last modified	Size	Description
 Parent Directory		-	
 document_2021-05-03T15:00:18Z.json	2021-05-25 18:57	1.3M	
 document_2021-05-03T15:00:18Z_1.json	2021-05-25 18:57	1.2M	
 document_2021-05-04T13:00:21Z.json	2021-05-25 18:47	1.7M	
 document_2021-05-04T13:00:21Z_1.json	2021-05-25 18:47	1.4M	
 document_2021-05-04T14:00:20Z.json	2021-05-25 18:47	1.3M	
 document_2021-05-05T14:10:17Z.json	2021-05-25 18:47	1.4M	
 document_2021-05-05T14:10:17Z_1.json	2021-05-25 18:47	1.6M	
 document_2021-05-05T20:11:27Z.json	2021-05-25 18:47	1.5M	
 document_2021-05-06T13:53:02Z.json	2021-05-25 18:47	1.4M	
 document_2021-05-06T13:53:02Z_1.json	2021-05-25 18:47	1.4M	
 document_2021-05-09T10:22:06Z.json	2021-05-25 18:47	1.2M	

Figura 26: Captura de parte del directorio json/ de la web

Index of /rdf













Name	Last modified	Size	Description
 Parent Directory		-	
 document_2021-05-03T15:00:18Z.n3	2021-05-25 19:01	6.9M	
 document_2021-05-03T15:00:18Z_1.n3	2021-05-25 19:01	6.6M	
 document_2021-05-04T13:00:21Z.n3	2021-05-25 19:03	8.9M	
 document_2021-05-04T13:00:21Z_1.n3	2021-05-25 19:04	7.4M	
 document_2021-05-04T14:00:20Z.n3	2021-05-25 19:01	7.2M	
 document_2021-05-05T14:10:17Z.n3	2021-05-25 19:02	7.6M	
 document_2021-05-05T14:10:17Z_1.n3	2021-05-25 19:04	8.2M	
 document_2021-05-05T20:11:27Z.n3	2021-05-25 19:01	7.8M	
 document_2021-05-06T13:53:02Z.n3	2021-05-25 19:01	7.7M	
 document_2021-05-06T13:53:02Z_1.n3	2021-05-25 19:00	7.4M	
 document_2021-05-09T10:22:06Z.n3	2021-05-25 19:01	6.4M	

Figura 27: Captura de parte del directorio rdf/ de la web

6 Análisis del impacto del trabajo

Según lo previamente descrito y los resultados provistos por este trabajo, se pretende que éste tenga una fuerte componente práctica en el ámbito de la información de las licitaciones públicas. Por ello, se espera que el impacto de este trabajo sea significativo en este campo, el de los datos abiertos, que además tiene como algunas de sus bases fundamentales la cooperación e interoperabilidad de sistemas con el fin de proveer servicios que puedan ser accedidos y aprovechados por todos.

Si bien es cierto que el funcionamiento del sistema desarrollado y desplegado en este trabajo se basa fundamentalmente en la transformación de datos, se espera que gracias a esta transformación y a su publicación como datos enlazados se pueda construir y mejorar la base de conocimiento existente acerca del ámbito de las licitaciones de tal manera que sirva tanto de apoyo en la toma de decisiones de las entidades públicas, como de disuasión de algunos delitos fiscales que puedan ser encontrados mediante la explotación de los nuevos datos.

En relación con los Objetivos de Desarrollo Sostenible de la Agenda 2030 de la Organización de las Naciones Unidas [95], este trabajo se alinea con el objetivo decimosexto, ODS 16 - *Promover sociedades pacíficas e inclusivas para el desarrollo sostenible, facilitar el acceso a la justicia para todos y crear instituciones eficaces, responsables e inclusivas a todos los niveles* [96]. Según lo expuesto en los dos anteriores párrafos, se espera que este trabajo sirva para mejorar la eficacia de las instituciones públicas, en este caso de aquellas relacionadas con las ofertas de contrataciones.



Figura 28: Logo del Objetivo de Desarrollo Sostenible nº 16

7 Conclusiones

...

8 Referencias

- [1] Open Knowledge Foundation,
Open Data Handbook: ¿Qué son los datos abiertos?.
<https://opendatahandbook.org/guide/es/what-is-open-data/>
- [2] World Wide Web Consortium,
Guía breve de Linked Data.
<https://www.w3c.es/Divulgacion/GuiasBreves/LinkedData>
- [3] The Linked Open Data Cloud.
<https://lod-cloud.net/>
- [4] Web Hypertext Application Technology Working Group,
HTML Living Standard.
<https://html.spec.whatwg.org/multipage/>
- [5] World Wide Web Consortium,
RDF 1.1 Concepts and Abstract Syntax.
<https://www.w3.org/TR/rdf11-concepts/>
- [6] DBpedia.
<https://www.dbpedia.org/>
- [7] Linked Open Vocabularies.
<https://lov.linkeddata.es/dataset/lov>
- [8] World Wide Web Consortium,
RDF 1.1 N-Triples, A line-based syntax for an RDF graph.
<http://www.w3.org/TR/n-triples/>
- [9] World Wide Web Consortium,
RDF 1.1 Turtle, Terse RDF Triple Language.
<https://www.w3.org/TR/turtle/>
- [10] World Wide Web Consortium,
JSON-LD 1.1, A JSON-based Serialization for Linked Data.
<http://www.w3.org/TR/json-ld/>
- [11] World Wide Web Consortium,
RDF 1.1 XML Syntax.
<https://www.w3.org/TR/rdf-syntax-grammar/>
- [12] World Wide Web Consortium,
RDFa 1.1 Primer - Third Edition, Rich Structured Data Markup for Web Documents.
<https://www.w3.org/TR/xhtml-rdfa-primer/>
- [13] World Wide Web Consortium,
RDF 1.1 N-Quads, A line-based syntax for RDF datasets.
<https://www.w3.org/TR/n-quads/>
- [14] International Organization for Standardization,
Information processing systems — Database language — SQL.
ISO 9075:1987
<https://www.iso.org/standard/16661.html>

- [15] World Wide Web Consortium,
SPARQL Query Language for RDF.
<https://www.w3.org/TR/rdf-sparql-query/>
- [16] The Apache Software Foundation,
A free and open source Java framework for building Semantic Web and Linked Data applications.
<https://jena.apache.org/>
- [17] RDFLib Team,
Python library for working with RDF, a simple yet powerful language for representing information.
<https://github.com/RDFLib/rdfLib>
- [18] Eclipse Foundation, Inc.,
Scalable RDF for Java.
<https://rdf4j.org/>
- [19] OpenLink Software,
SQL-ORDBMS and Web Server hybrid with RDF data management.
<https://virtuoso.openlinksw.com/>
- [20] Diario Oficial de la Unión Europea,
Directiva 2014/23/UE del Parlamento Europeo y del Consejo de 26 de febrero de 2014 relativa a la adjudicación de contratos de concesión.
<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32014L0023&from=ES>
- [21] Diario Oficial de la Unión Europea,
Directiva 2014/24/UE del Parlamento Europeo y del Consejo de 26 de febrero de 2014 sobre contratación pública.
<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32014L0024&from=ES>
- [22] Diario Oficial de la Unión Europea,
Directiva 2014/25/UE del Parlamento Europeo y del Consejo de 26 de febrero de 2014 relativa a la contratación por entidades que operan en los sectores del agua, la energía, los transportes y los servicios postales.
<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32014L0025&from=ES>
- [23] Diario Oficial de la Unión Europea,
Directiva 2009/81/CE del Parlamento Europeo y del Consejo de 13 de julio de 2009 sobre coordinación de los procedimientos de adjudicación de determinados contratos de obras, de suministro y de servicios por las entidades o poderes adjudicadores en los ámbitos de la defensa y la seguridad.
<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32009L0081&from=ES>
- [24] Boletín Oficial del Estado,
Ley 9/2017, de 8 de noviembre, de Contratos del Sector Público, por la que se transponen al ordenamiento jurídico español las Directivas del Parlamento Europeo y del Consejo

- 2014/23/UE y 2014/24/UE, de 26 de febrero de 2014.
<https://www.boe.es/buscar/pdf/2017/BOE-A-2017-12902-consolidado.pdf>
- [25] Ministerio de Hacienda,
Guía de implementación de documentos CODICE 2.0.
https://contrataciondelestado.es/codice/2.0/doc/CODICE_2_GuiaImplementacion_v1.3.pdf
- [26] Ministerio de Hacienda,
Formato de sindicación y reutilización de datos sobre licitaciones publicadas en la Plataforma de Contratación del Sector Público.
https://www.hacienda.gob.es/Documentacion/Publico/D.G.PATRIMONIO/Plataforma_Contratacion/especificacion-sindicacion-1-3.pdf
- [27] International Organization for Standardization,
Information technology — Metadata registries (MDR) — Part 7: Metamodel for data set registration.
ISO/IEC 11179-7:2019
<https://www.iso.org/standard/68766.html>
- [28] International Organization for Standardization,
Electronic Business Extensible Markup Language (ebXML) — Part 5: Core Components Specification (CCS).
ISO 15000-5:2014
<https://www.iso.org/standard/61433.html>
- [29] OASIS,
Electronic Business using eXtensible Markup Language.
<http://www.ebxml.org/l>
- [30] Comisión Económica de las Naciones Unidas para Europa,
Trade Facilitation and E-business.
<https://unece.org/trade/uncefact>
- [31] World Wide Web Consortium,
Extensible Markup Language (XML).
<https://www.w3.org/XML/>
- [32] Diario Oficial de las Comunidades Europeas,
Intercambio electrónico de datos entre administraciones: Programa IDA.
<https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=LEGISSUM:l24147a&from=ES>
- [33] Portal de Administración Electrónica, Ministerio de Asuntos Económicos y Transformación Digital,
Componentes y Documentos Interoperables para la Contratación Electrónica.
<https://administracionelectronica.gob.es/cise/137>
- [34] Ministerio de Hacienda,
Licitaciones publicadas en la Plataforma de Contratación del Sector Público.
https://www.hacienda.gob.es/es-ES/GobiernoAbierto/DatosAbiertos/Paginas/licitaciones_plataforma_contratacion.aspx

- [35] Ministerio de Hacienda,
Resumen de contenido en conjuntos de datos abiertos.
https://www.hacienda.gob.es/Documentacion/Publico/D.G.PATRIMONIO/Plataforma_Contratacion/Resumen-Datos-Abiertos.pdf
- [36] Open Contracting Partnership,
Estándar de Datos de Contrataciones Abiertas: Documentación.
<https://standard.open-contracting.org/latest/es/#>
- [37] Open Contracting Partnership,
Open Contracting Tools Directory.
<https://www.open-contracting.org/es/resources/open-contracting-tools-directory/>
- [38] Open Contracting Partnership,
El Estándar de Datos para las Contrataciones Abiertas.
<https://www.open-contracting.org/es/data-standard/>
- [39] Open Contracting Partnership,
Estándar de Datos de Contrataciones Abiertas: Bloques.
https://standard.open-contracting.org/latest/es/getting_started/building_blocks/
- [40] TheyBuyForYou,
TheyBuyForYou.
<https://theybuyforyou.eu/>
- [41] TheyBuyForYou,
TheyBuyForYou GitHub Repositories.
<https://github.com/TBFY>
- [42] TheyBuyForYou,
TBFY - OptiqueVQS.
<https://www.youtube.com/watch?v=l4mqMvqo3qQ>
- [43] TheyBuyForYou,
TBFY - Anomaly detection in public spending data.
https://www.youtube.com/watch?v=bxFtEeT_KDU
- [44] Institute of Electrical and Electronics Engineers,
Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement.
<https://ieeexplore.ieee.org/document/6569022>
- [45] International Organization for Standardization,
Information technology — Programming languages — C#.
ISO/IEC 23270:2018
<https://www.iso.org/standard/75178.html>
- [46] Microsoft,
LINQ to XML overview.
<https://docs.microsoft.com/en-us/dotnet/standard/linq/linq-xml-overview>

- [47] Stephen Cleary,
A helper library for async/await.
<https://github.com/StephenCleary/AsyncEx>
- [48] Open Source Initiative,
The MIT License.
<https://opensource.org/licenses/MIT>
- [49] Dennis Doomen,
Fluent API for asserting the results of unit tests that targets .NET Framework.
<https://fluentassertions.com/>
- [50] Open Source Initiative,
Apache License, Version 2.0.
<https://opensource.org/licenses/Apache-2.0>
- [51] Microsoft,
Implementation of key-value pair based configuration.
<https://github.com/dotnet/runtime/tree/main/src/libraries/Microsoft.Extensions.Configuration>
- [52] NLog Project,
Advanced and Structured Logging for Various .NET Platforms .
<https://nlog-project.org/>
- [53] Open Source Initiative,
The 3-Clause BSD License.
<https://opensource.org/licenses/BSD-3-Clause>
- [54] Newtonsoft,
Popular high-performance JSON framework for .NET Framework.
<https://www.newtonsoft.com/json>
- [55] .NET Foundation,
Free, open source, community-focused unit testing tool for the .NET Framework.
<https://xunit.net/>
- [56] Microsoft,
Introducción a proyectos y soluciones.
<https://docs.microsoft.com/es-es/visualstudio/get-started/tutorial-projects-solutions?view=vs-2019>
- [57] Ministerio de Hacienda,
Documento de licitaciones de actualización diaria.
https://contrataciondelestado.es/sindicacion/sindicacion_643/licitacionesPerfilesContratanteCompleto3.atom
- [58] SonarSource,
SonarQube.
<https://www.sonarqube.org/>
- [59] SonarSource,
Rules Repository / RSPEC-125.
<https://jira.sonarsource.com/browse/RSPEC-125>

- [60] SonarSource,
Rules Repository / RSPEC-1168.
<https://jira.sonarsource.com/browse/RSPEC-1168>
- [61] SonarSource,
Rules Repository / RSPEC-3776.
<https://jira.sonarsource.com/browse/RSPEC-3776>
- [62] Ministerio de Hacienda,
Codigos de estados de licitación definidos por la DGPE para la sindicacion entre plataformas.
<https://contrataciondelestado.es/codice/cl/2.04/SyndicationContractFolderStatusCode-2.04.gc>
- [63] Open Contracting Partnership,
Listas de códigos - Etiqueta de entrada.
<https://standard.open-contracting.org/latest/es/schema/codelists/#release-tag>
- [64] Open Data Services,
Herramienta de revisión de datos.
<https://standard.open-contracting.org/review/>
- [65] Ministerio de Hacienda,
Codigos de tipos de contrato definidos por la DGPE para la sindicacion entre plataformas.
<https://contrataciondelestado.es/codice/cl/2.08/ContractCode-2.08.gc>
- [66] Open Contracting Partnership,
Listas de códigos - Categoría de compra.
<https://standard.open-contracting.org/latest/es/schema/codelists/#procurement-category>
- [67] Open Contracting Partnership,
Listas de códigos - Tipo de documento.
<https://standard.open-contracting.org/latest/es/schema/codelists/#document-type>
- [68] Open Contracting Partnership,
OCDS Extension Explorer - Lots.
<https://extensions.open-contracting.org/en/extensions/lots/v1.1.5/>
- [69] PublicTendering,
List of the CPV codes.
<https://www.publictendering.com/cpv-codes/list-of-the-cpv-codes/>
- [70] Ministerio de Hacienda,
Codigos de tipos de procedimientos de contratación definidos por la DGPE para la sindicacion entre plataformas.
<https://contrataciondelestado.es/codice/cl/2.07/SyndicationTenderingProcessCode-2.07.gc>
- [71] Open Contracting Partnership,
Listas de códigos - Método.
<https://standard.open-contracting.org/latest/es/schema/codelists/#method>

- [72] Ministerio de Hacienda,
Codigos de tipos de sistemas de contratación definidos por la DGPE para la sindicacion entre plataformas.
<https://contrataciondelestado.es/codice/cl/2.08/ContractingSystemTypeCode-2.08.gc>
- [73] Ministerio de Hacienda,
Codigos de tipos de presentación de oferta definidos por la DGPE para la sindicacion entre plataformas.
<https://contrataciondelestado.es/codice/cl/1.04/TenderDeliveryCode-1.04.gcenlace>
- [74] Open Contracting Partnership,
Listas de códigos - Método de presentación.
<https://standard.open-contracting.org/latest/es/schema/codelists/#submission-method>
- [75] Open Contracting Partnership,
Listas de códigos - Esquema de Identificación de Organizaciones.
<https://standard.open-contracting.org/latest/es/schema/codelists/#organization-identifier-scheme>
- [76] Open Data Services,
Common Directory of Organizational Units and Offices - DIR3.
<http://org-id.guide/list/ES-DIR3>
- [77] Open Data Services,
Central Commercial Register of the Kingdom of Spain - RMC.
<http://org-id.guide/list/ES-RMC>
- [78] Ministerio de Hacienda,
Codigos de resultados de licitaciones definidos por la DGPE para la sindicacion entre plataformas.
<http://contrataciondelestado.es/codice/cl/2.02/TenderResultCode-2.02.gc>
- [79] Open Contracting Partnership,
Listas de códigos - Estado de la adjudicación.
<https://standard.open-contracting.org/latest/es/schema/codelists/#award-status>
- [80] Open Contracting Partnership,
Listas de códigos - Rol de la parte.
<https://standard.open-contracting.org/latest/es/schema/codelists/#party-role>
- [81] ¿?,
Script para la transformación JSON a XML.
https://github.com/hxshfx/TFG_PabloPalomoLopez/blob/main/Linked_Data/generate_rdf.py
- [82] Semantify,
RocketRML - A NodeJS implementation of a use-case specific RML mapper.
<https://github.com/semantifyit/RocketRML>

- [83] Creative Commons,
Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).
<https://creativecommons.org/licenses/by-sa/4.0/>
- [84] Open Data Services,
Reporte de la herramienta de revisión de datos.
<https://standard.open-contracting.org/review/data/825aff61-2d11-4f3d-b377-060d3b35d914>
- [85] IDLab - imec - Ghent University,
RDF Mapping Language (RML).
<https://rml.io/specs/rml/>
- [86] IDLab - imec - Ghent University,
RDF Mapping Language (RML).
<https://github.com/RMLio/rmlmapper-java>
- [87] TheyBuyForYou,
RML mapping rules for OCDS 1.0.
https://github.com/TBFY/knowledge-graph/blob/master/rml-mappings/openopps_mapping.ttl
- [88] ¿?,
Reglas RML utilizadas en el trabajo.
https://github.com/hxshfx/TFG_PabloPalomoLopez/blob/main/Linked_Data/rml_mappings.ttl
- [89] ¿?,
Script con las queries SPARQL.
https://github.com/hxshfx/TFG_PabloPalomoLopez/blob/main/Linked_Data/sparql.py
- [90] ¿?,
Ejemplos del pipeline: documentos de licitaciones.
https://github.com/hxshfx/TFG_PabloPalomoLopez/tree/main/OCDS_Mapper/src/Examples/xml
- [91] ¿?,
Ejemplos del pipeline: consultas SPARQL.
https://github.com/hxshfx/TFG_PabloPalomoLopez/tree/main/Linked_Data/queries1
- [92] ¿?,
Ejemplos del pipeline: procesado JSON.
https://github.com/hxshfx/TFG_PabloPalomoLopez/tree/main/OCDS_Mapper/src/Examples/json
- [93] ¿?,
Ejemplos del pipeline: procesado RDF.
https://github.com/hxshfx/TFG_PabloPalomoLopez/tree/main/Linked_Data/rdf
- [94] Ontology Engineering Group.
<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/index.html>
- [95] Organización de las Naciones Unidas,
Objetivos de Desarrollo Sostenible.
<https://www.un.org/sustainabledevelopment/es/>

- [96] Organización de las Naciones Unidas,
Objetivo 16: Promover sociedades justas, pacíficas e inclusivas.
<https://www.un.org/sustainabledevelopment/es/peace-justice/>

9 Anexos

9.1 Anexo I: Interfaces de los componentes del *OCDS_Mapper*

9.1.1 Componente *Provider*

```
namespace OCDS_Mapper.src.Interfaces
{
    /* Interfaz del componente de provisión de datos */

    public interface IProvider
    {
        /* Propiedades */

        /* propiedad Parser => IParser
         *      Instancia del Parser para obtener los enlaces de los documentos
         */
        IParser Parser { get; set; }

        /* propiedad Files => AsyncCollection<Document>
         *      Colección thread-safe que permite una extracción
         *      ordenada de los documentos provistos
         */
        AsyncCollection<Document> Files { get; set; }

        /* Funciones */

        /* función asíncrona TakeFile() => Document
         *      Función bloqueante que devuelve un documento provisto cuando éste
         *      está disponible
         *      @return : stream de texto del documento provisto, o null si no quedan
         *      más documentos que proveer
         */
        Task<Document> TakeFile();

        /* función SetParser(IParser) => bool
         *      (utilizada solo en modo PROVIDE_ALL y PROVIDE_DAILY)
         *      Desbloquea el thread en background que integra al Provider y al Parser
         *      @param parser : Instancia del Parser para cada documento
         *      @return : true si el documento del Parser es día anterior al actual,
         *      false e.o.c
         */
        bool SetParser(IParser parser);

        /* función RemoveFile(string) => void
         *      Función que elimina un archivo provisto ya mapeado
         *      @param filePath : path del documento a eliminar
         */
        void RemoveFile(string filePath);
    }
}
```

9.1.2 Componente *Parser*

```

namespace OCDS_Mapper.src.Interfaces
{
    /* Interfaz del componente de parseo */

    public interface IParser
    {
        /* Propiedades */

        /* propiedad DocumentRoot => XElement
         * Representa el elemento XML raíz del documento atom
         */
        XElement DocumentRoot { get; set; }

        /* propiedad EntryRoot => XElement
         * Representa el elemento XML raíz de la entrada siendo parseada:
         * @ej : XElement(<cac-place-ext:ContractFolderStatus>)
         */
        XElement EntryRoot { get; set; }

        /* Funciones */

        /* función GetNamespaces() => IDictionary<string, XNamespace>
         * Representa mediante pares de nombres y espacios de nombres los
         * namespaces del documento
         * Recupera dicha información de los atributos del elemento feed en los
         * ficheros atom
         * @return : Diccionario con los pares de espacios de nombres
         * @ej : { {"cac" : XNamespace("...:CommonAggregateComponents-2")}, ... }
         */
        IDictionary<string, XNamespace> GetNamespaces();

        /* función GetEntrySet(IDictionary<string, XNamespace>) => IEnumerable<XElement>
         * Devuelve el conjunto de entradas en el fichero atom
         * Las entradas vienen definidas por el elemento entry
         * @param namespaces : Diccionario de espacios de nombres
         * @return : Conjunto de elementos XML
         * @ej : [ XElement(<entry xmlns="..."> <id>...</id> ... </entry>), ... ]
         */
        IEnumerable<XElement> GetEntrySet(IDictionary<string, XNamespace> namespaces);

        /* función SetEntryRootElement(XElement) => void
         * Actualiza la propiedad del objeto EntryRoot
         * Extrae el elemento XML correspondiente a "ContractFolderStatus"
         * @throws InvalidParsedElementException : si el elemento no contiene
         * "ContractFolderStatus"
         */
        void SetEntryRootElement(XElement entry);
    }
}

```

```
/* función GetElements(IEnumerable<XName>) => XElement[]
*     Devuelve el (los) elemento(s) XML descrito por la ruta pasada
*     como parámetro
* @param pathToElement : lista enlazada con la ruta del elemento deseado:
*     @ej : [ XName(XNamespace("cac") + "ProcurementProject"),
*           XName(XNamespace("cbc") + "Name") ]
* @return : elemento(s) buscado(s), o null si no se puede encontrar
*     @ej : XElement(<cbc:Name>..."</cbc:Name>)
*/
XElement[] GetElements(IEnumerable<XName> pathToElement);

/* función GetNextFile() => Uri
*     Devuelve la URI correspondiente al siguiente fichero, descrito por el
*     elemento "link" y el atributo "rel=next"
* @return : La URI que describe el fichero enlazado al siendo parseado en
*     esta instancia
*/
Uri GetNextFile();

/* función GetDocumentTimestamp(bool) => string
*     Devuelve el timestamp del documento o de una entrada específica
* @param document : True si se quiere recuperar el timestamp del documento,
*     False si se quiere el de la entrada
* @return : timestamp (o campo "updated") del documento de licitaciones
*     o de la entrada
*/
string GetDocumentTimestamp(bool document);
}
}
```


9.1.3 Componente *Mapper*

```
namespace OCDS_Mapper.src.Interfaces
{
    /* Interfaz del componente de mapeo */

    public interface IMapper
    {
        /* Propiedades */

        /* propiedad MappedEntry => JObject
        * Representa el elemento JSON con la información mapeada
        */
        JObject MappedEntry { get; set; }

        /* Funciones */

        /* función MapElement(IEnumerable<string>, XElement[]) => void
        * Realiza el mapeo de elemento(s) CODICE a elemento(s) OCDS
        * @param pathMap : ruta del elemento cuando sea mapeado
        * @param parsedElement : elemento(s) a mapear
        * @ej : MapElement( [ "tender", "title" ], [ "ABCD" ] )
        * => MappedEntry = { "tender": { "title": "ABCD" } }
        * @ej : MapElement( [ "tag" ], [ "PRE" ] )
        * => MappedEntry = { "tag": [ "planning" ] }
        */
        void MapElement(IEnumerable<string> pathMap, XElement[] parsedElement);

        /* función Commit(string) => void
        * Introduce los cambios al JSON que no se pueden introducir
        * mediante los mapeos unitarios de elementos (metadatos, colecciones, etc.)
        * @param publishDate : fecha de publicación de la entrega
        */
        void Commit(string publishDate);
    }
}
```

9.1.4 Componente *Packager*

```
namespace OCDS_Mapper.src.Interfaces
{
    /* Interfaz del componente de empaquetado de datos */

    public interface IPackager
    {
        /* Funciones */

        /* función GetIdentifier(string) => string
         *     Obtiene el número de ocurrencias del identificador hasta el momento
         *     con el objetivo de evitar colisiones de ids no únicos
         * @param entryID : identificador de la entrada
         * @return : número de ocurrencias del identificador
         */
        string GetIdentifier(string entryID);

        /* función Package(JObject) => void
         *     Introduce una entrada al paquete
         * @param entry : objeto JSON que introducir
         */
        void Package(JObject entry);

        /* función Publish(string) => void
         *     Publica los datos escribiéndolos en el directorio de salida
         * @param dirPath : path del directorio de salida
         */
        void Publish(string dirPath);
    }
}
```

9.2 Anexo II: Ejemplos de los mapeados entre CODICE y OCDS

9.2.1 Datos generales del expediente

9.2.1.1 Estado

```
<cac-place-ext:ContractFolderStatus>
  <cbc-place-ext:ContractFolderStatusCode languageID="es"
    listURI="https://contrataciondelestado.es/codice/cl/2.04/
    SyndicationContractFolderStatusCode-2.04.gc">
    ADJ
  </cbc-place-ext:ContractFolderStatusCode>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tag": [
    "award"
  ]
}
```

9.2.1.2 Número de expediente

```
<cac-place-ext:ContractFolderStatus>
  <cbc:ContractFolderID>
    002/2021-CONTR
  </cbc:ContractFolderID>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "ocid": "ES-002/2021-CONTR"
}
```

9.2.1.3 Objeto del contrato

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProject>
    <cbc:Name>
      Suministro de energía eléctrica para las instalaciones de RTVE
    </cbc:Name>
  </cac:ProcurementProject>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "title": "Suministro de energía eléctrica para las instalaciones de RTVE"
  }
}
```

9.2.1.4 Valor estimado e importe de licitación

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProject>
    <cac:BudgetAmount>
      <cbc:EstimatedOverallContractAmount currencyID="EUR">
        185000
      </cbc:EstimatedOverallContractAmount>
      <cbc:TotalAmount currencyID="EUR">
        120000
      </cbc:TotalAmount>
    </cac:BudgetAmount>
  </cac:ProcurementProject>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "value": {
      "amount": 185000,
      "currency": "EUR"
    }
  },
  "planning": {
    "budget": {
      "amount": {
        "amount": 120000,
        "currency": "EUR"
      }
    }
  }
}
```

9.2.1.5 Duración del contrato

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProject>
    <cac:PlannedPeriod>
      <cbc:StartDate>
        2021-09-01
      </cbc:StartDate>
      <cbc:EndDate>
        2021-12-01
      </cbc:EndDate>
    </cac:PlannedPeriod>
  </cac:ProcurementProject>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "tenderPeriod": {
      "startDate": "2021-09-01T00:00:00Z",
      "endDate": "2021-12-01T00:00:00Z"
    }
  }
}
```

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProject>
    <cac:PlannedPeriod>
      <cbc:DurationMeasure unitCode="MON">
        3
      </cbc:DurationMeasure>
    </cac:PlannedPeriod>
  </cac:ProcurementProject>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "tenderPeriod": {
      "durationInDays": 90
    }
  }
}
```

9.2.1.6 Tipo de contrato

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProject>
    <cbc:TypeCode
      listURI="https://contrataciondelestado.es/codice/cl/2.08/
      ContractCode-2.08.gc">
      2
    </cbc:TypeCode>
  </cac:ProcurementProject>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "mainProcurementCategory": "services"
  }
}
```

9.2.1.7 Enlaces de descarga de pliegos

```
<cac-place-ext:ContractFolderStatus>
  <cac:LegalDocumentReference>
    <cbc:ID>
      PPT_2020_00120.pdf
    </cbc:ID>
    <cac:Attachment>
      <cac:ExternalReference>
        <cbc:URI>
          https://contrataciondelestado.es/wps/wcm/...
        </cbc:URI>
      </cac:ExternalReference>
    </cac:Attachment>
  </cac:LegalDocumentReference>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "documents": [
      {
        "documentType": "technicalSpecifications",
        "id": "PPT_2020_00120.pdf",
        "url": "https://contrataciondelestado.es/wps/wcm/..."
      }
    ]
  }
}
```

9.2.2 Lotes y artículos

9.2.2.1 Número de lote

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProjectLot>
    <cbc:ID>
      5
    </cbc:ID>
  </cac:ProcurementProjectLot>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "lots": [
      {
        "id": "lot-5"
      }
    ],
    "items": [
      {
        "id": "5",
        "relatedLot": "lot-5"
      }
    ]
  }
}
```

9.2.2.2 Objeto del lote

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProjectLot>
    <cac:ProcurementProject>
      <cbc:Name>
        Equipos respiratorios autónomos
      </cbc:Name>
    </cac:ProcurementProject>
  </cac:ProcurementProjectLot>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "lots": [
      {
        "name": "Equipos respiratorios autónomos"
      }
    ]
  }
}
```

9.2.2.3 Importe del lote

```
<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProjectLot>
    <cac:ProcurementProject>
      <cac:BudgetAmount>
        <cbc:TotalAmount>
          2250
        </cbc:TotalAmount>
      </cac:BudgetAmount>
    </cac:ProcurementProject>
  </cac:ProcurementProjectLot>
</cac-place-ext:ContractFolderStatus>
```




```

{
  "tender": {
    "lots": [
      {
        "value": {
          "value": 2250.0,
          "currency": "EUR"
        }
      }
    ]
  }
}

```

9.2.2.4 Clasificación CPV

```

<cac-place-ext:ContractFolderStatus>
  <cac:ProcurementProjectLot>
    <cac:ProcurementProject>
      <cbc:RequiredCommodityClassification>
        <cbc:ItemClassificationCode
          listURI="http://contrataciondelestado.es/codice/cl/1.04/
          CPV2007-1.04.gc">
          35111100
        </cbc:ItemClassificationCode>
      </cbc:RequiredCommodityClassification>
    </cac:ProcurementProject>
  </cac:ProcurementProjectLot>
</cac-place-ext:ContractFolderStatus>

```



```

{
  "tender": {
    "items": [
      {
        "classification": {
          "id": "35111100",
          "schema": "CPV",
          "description_es": "Aparatos respiratorios para extinción de
            incendios."
        }
      }
    ]
  }
}

```

9.2.3 Procesos de licitación

9.2.3.1 Tipo de procedimiento

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderingProcess>
    <cbc:ProcedureCode
      listURI="https://contrataciondelestado.es/codice/cl/2.07/
      SyndicationTenderingProcessCode-2.07.gc">
      9
    </cbc:ProcedureCode>
  </cac:TenderingProcess>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "procurementMethod": "open"
  }
}
```

9.2.3.2 Sistema de contratación

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderingProcess>
    <cbc:ContractingSystemCode
      listURI="https://contrataciondelestado.es/codice/cl/2.08/
      ContractingSystemTypeCode-2.08.gc">
      3
    </cbc:ContractingSystemCode>
  </cac:TenderingProcess>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "procurementMethodDetails_es": "Contrato basado en un Acuerdo Marco"
  }
}
```

9.2.3.3 Presentación de la oferta

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderingProcess>
    <cbc:SubmissionMethodCode
      listURI="https://contrataciondelestado.es/codice/cl/1.04/
      TenderDeliveryCode-1.04.gc">
      1
    </cbc:SubmissionMethodCode>
  </cac:TenderingProcess>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "submissionMethod": [
      "electronicSubmission"
    ]
  }
}
```

9.2.3.4 Idioma de presentación de la oferta

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderingTerms>
    <cac:Language>
      <cbc:ID>
        es
      </cbc:ID>
    </cac:Language>
    <cac:Language>
      <cbc:ID>
        ca
      </cbc:ID>
    </cac:Language>
  </cac:TenderingTerms>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "submissionMethodDetails": "Languages: es, ca"
  }
}
```

9.2.3.5 Licitación son subasta electrónica

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderingProcess>
    <cac:AuctionTerms>
      <cbc:AuctionConstraintIndicator>
        true
      </cbc:AuctionConstraintIndicator>
    </cac:AuctionTerms>
  </cac:TenderingProcess>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "submissionMethod": [
      "electronicAuction"
    ]
  }
}
```

9.2.4 Entidades adjudicadoras

9.2.4.1 Órgano de contratación

```
<cac-place-ext:ContractFolderStatus>
  <cac-place-ext:LocatedContractingParty>
    <cac:Party>
      <cac:PartyName>
        <cbc:Name>
          Junta de Gobierno del Ayuntamiento de Orihuela
        </cbc:Name>
      </cac:PartyName>
    </cac:Party>
  </cac-place-ext:LocatedContractingParty>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "parties": [
    {
      "name": "Junta de Gobierno del Ayuntamiento de Orihuela"
    }
  ]
}
```

9.2.4.2 Ubicación orgánica

```

<cac-place-ext:ContractFolderStatus>
  <cac-place-ext:LocatedContractingParty>
    <cac:Party>
      <cac:PartyIdentification>
        <cbc:ID schemeName="DIR3">
          L01030993
        </cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyIdentification>
        <cbc:ID schemeName="NIF">
          P0309900I
        </cbc:ID>
      </cac:PartyIdentification>
    </cac:Party>
  </cac-place-ext:LocatedContractingParty>
</cac-place-ext:ContractFolderStatus>

```



```

{
  "parties": [
    {
      "identifier": {
        "scheme": "ES-DIR3",
        "id": "L01030993",
        "additionalIdentifiers": [
          {
            "scheme": "ES-RMC",
            "id": "P0309900I"
          }
        ]
      },
      "id": "L01030993",
      "roles": [
        "procuringEntity"
      ]
    }
  ]
}

```

9.2.4.3 Otros campos

```

<cac-place-ext:ContractFolderStatus>
  <cac-place-ext:LocatedContractingParty>
    <cac:Party>
      <cbc:WebsiteURI>
        http://www.oriuela.es/
      </cbc:WebsiteURI>
      <cac:PostalAddress>
        <cbc:CityName>
          Oriuela
        </cbc:CityName>
        <cbc:PostalZone>
          03300
        </cbc:PostalZone>
        <cac:AddressLine>
          <cbc:Line>
            Marqués de Arneva, 1
          </cbc:Line>
        </cac:AddressLine>
        <cac:Country>
          <cbc:IdentificationCode
            listURI="http://docs.oasis-open.org/ubl/os-ubl-2.0/
            cl/gc/default/CountryIdentificationCode-2.0.gc">
            ES
          </cbc:IdentificationCode>
        </cac:Country>
      </cac:PostalAddress>
      <cac:Contact>
        <cbc:Name>
          Junta de Gobierno del Ayuntamiento de Oriuela
        </cbc:Name>
        <cbc:Telephone>
          966076100
        </cbc:Telephone>
        <cbc:Telefax>
          966076101
        </cbc:Telefax>
        <cbc:ElectronicMail>
          contratacion@oriuela.es
        </cbc:ElectronicMail>
      </cac:Contact>
    </cac:Party>
  </cac-place-ext:LocatedContractingParty>
</cac-place-ext:ContractFolderStatus>

```



```
{
  "parties": [
    {
      "countryName": "ES",
      "address": {
        "streetAddress": "Marqués de Arneva, 1",
        "locality": "Orihuela",
        "postalCode": "03300"
      },
      "contactPoint": {
        "email": "contratacion@orihuela.es",
        "faxNumber": "966076101",
        "name": "Junta de Gobierno del Ayuntamiento de Orihuela",
        "telephone": "966076100",
        "url": "http://www.orihuela.es/"
      }
    }
  ]
}
```

9.2.5 Resultado del procedimiento

9.2.5.1 Identificador

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cac:AwardedTenderedProject>
      <cbc:ProcurementProjectLotID>
        1
      </cbc:ProcurementProjectLotID>
    </cac:AwardedTenderedProject>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "awards": [
    {
      "id": "1"
    }
  ]
}
```

9.2.5.2 Resultado

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cac:ResultCode
      listURI="http://contrataciondelestado.es/codice/cl/2.02/
      TenderResultCode-2.02.gc">
      8
    </cac:ResultCode>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "awards": [
    {
      "status": "active"
    }
  ]
}
```

9.2.5.3 Identidad del adjudicatario

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cac:WinningParty>
      <cac:PartyIdentification>
        <cbc:ID schemeName="NIF">
          B37012515
        </cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyName>
        <cbc:Name>
          AGRUPESCA, S.L.
        </cbc:Name>
      </cac:PartyName>
    </cac:WinningParty>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```




```

{
  "awards": [
    {
      "suppliers": [
        {
          "id": "B37012515",
          "name": "AGRUPESCA, S.L."
        }
      ]
    }
  ],
  "parties": [
    {
      "identifier": {
        "scheme": "ES-RMC",
        "id": "B37012515"
      },
      "id": "B37012515",
      "name": "AGRUPESCA, S.L.",
      "roles": [
        "supplier"
      ]
    }
  ]
}

```

9.2.5.4 Importe de adjudicación

```

<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cac:AwardedTenderedProject>
      <cac:LegalMonetaryTotal>
        <cbc:PayableAmount currencyID="EUR">
          18260.5
        </cbc:PayableAmount>
      </cac:LegalMonetaryTotal>
    </cac:AwardedTenderedProject>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>

```



```
{
  "awards": [
    {
      "value": {
        "amount": 18260.5,
        "currency": "EUR"
      }
    }
  ]
}
```

9.2.5.5 Número de licitadores participantes

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cbc:ReceivedTenderQuantity>
      5
    </cbc:ReceivedTenderQuantity>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "tender": {
    "numberOfTenderers": 5
  }
}
```

9.2.5.6 Fecha de la adjudicación

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cbc:AwardDate>
      2021-04-15
    </cbc:AwardDate>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "awards": [
    {
      "date": "2021-04-15T00:00:00Z"
    }
  ]
}
```

9.2.5.7 Descripción de la adjudicación

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cbc:Description>
      Mejor oferta
    </cbc:Description>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "awards": [
    {
      "description_es": "Mejor oferta"
    }
  ]
}
```

9.2.6 Información sobre el contrato

9.2.6.1 Identificador

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cac:Contract>
      <cbc:ID>
        1
      </cbc:ID>
    </cac:Contract>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "contracts": [
    {
      "id": "1"
    }
  ]
}
```

9.2.6.2 Fecha de entrada en vigor

```
<cac-place-ext:ContractFolderStatus>
  <cac:TenderResult>
    <cbc:StartDate>
      2021-04-01
    </cbc:StartDate>
  </cac:TenderResult>
</cac-place-ext:ContractFolderStatus>
```



```
{
  "contracts": [
    {
      "period": {
        "startDate": "2021-04-01T00:00:00Z"
      }
    }
  ]
}
```

9.3 Anexo III: Consultas SPARQL sobre los datos enlazados

9.3.1 Prefijos de los espacios de nombres utilizados

```
@prefix dc:      <http://purl.org/dc/terms/> .
@prefix ocids:   <http://data.tbify.eu/ontology/ocids#> .
@prefix ocidsext: <http://data.tbify.eu/ontology/ocids_extension#> .
@prefix org:     <http://www.w3.org/ns/org#> .
@prefix rdf:     <http://www.w3.org/TR/rdf-schema/> .
@prefix schema:  <http://schema.org/> .
@prefix skos:    <http://www.w3.org/2004/02/skos/core#> .
@prefix tbify:   <http://data.tbify.eu/ontology/tbify#> .
```

9.3.2 Consulta para la búsqueda de *releases*

```
SELECT ?date ?id ?tag
WHERE {
    ?release a tbify:Release .
    ?release tbify:releaseDate ?date .
    ?release tbify:releaseId ?id .
    ?release tbify:releaseTag ?tag .
}
```

9.3.3 Consulta para la búsqueda de procesos de contratación

```
SELECT ?ocid ?award ?contract ?plan ?release ?tender
WHERE {
    ?contractingProcess a ocids:ContractingProcess .
    ?contractingProcess ocids:ocid ?ocid .
    OPTIONAL {
        ?contractingProcess ocids:hasAward ?award
    } .
    OPTIONAL {
        ?contractingProcess ocids:hasContract ?contract
    } .
    OPTIONAL {
        ?contractingProcess ocids:hasPlan ?plan
    } .
    OPTIONAL {
        ?contractingProcess ocids:hasRelease ?release
    } .
    OPTIONAL {
        ?contractingProcess ocids:hasTender ?tender
    } .
}
```

9.3.4 Consulta para la búsqueda de adjudicaciones

```

SELECT ?awardId ?tender ?awardDate ?awardStatus ?awardDescription ?awardAmount
      ?awardCurrency ?awardSupplier ?supplierName ?supplierId
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?contractingProcess ocds:hasAward ?award .
  ?award ocds:awardId ?awardId .
  ?award ocds:isIssuedForTender ?tender .
  ?award ocds:awardDate ?awardDate .
  ?award ocds:awardStatus ?awardStatus .
  OPTIONAL {
    ?award dc:description ?awardDescription
  } .
  OPTIONAL {
    ?award ocds:hasAwardValue ?awardValue
    OPTIONAL {
      ?awardValue ocds:valueAmount ?awardAmount .
      ?awardValue ocds:valueCurrency ?awardCurrency .
    } .
  } .
  OPTIONAL {
    ?awardSupplier ocds:isSupplierFor ?award
    OPTIONAL {
      OPTIONAL {
        ?awardSupplier ocds:legalName ?supplierName
      } .
      OPTIONAL {
        ?awardSupplier org:identifier ?supplierId
      } .
    } .
  } .
}

```

9.3.5 Consulta para la búsqueda de contratos

```
SELECT ?contractId ?award ?contractStartDate ?contractEndDate
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?contractingProcess ocds:hasContract ?contract .
  ?contract ocds:contractId ?contractId .
  ?contract ocds:isIssuedForAward ?award .
  OPTIONAL {
    ?contract ocds:hasContractPeriod ?contractPeriod
    OPTIONAL {
      OPTIONAL {
        ?contractPeriod ocds:periodStartDate ?contractStartDate
      } .
      OPTIONAL {
        ?contractPeriod ocds:periodEndDate ?contractEndDate
      } .
    } .
  } .
}
```

9.3.6 Consulta para la búsqueda de planificaciones

```
SELECT ?plan ?planBudgetAmount ?planBudgetCurrency
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?contractingProcess ocds:hasPlan ?plan .
  ?plan ocds:hasBudget ?planBudget .
  ?planBudget ocds:budgetId ?planBudgetId .
  ?planBudget ocds:hasReservedValue ?planBudgetValue .
  ?planBudgetValue ocds:valueAmount ?planBudgetAmount .
  ?planBudgetValue ocds:valueCurrency ?planBudgetCurrency .
}
```

9.3.7 Consulta para la búsqueda de licitaciones

```

SELECT ?tenderId ?tenderTitle ?tenderProcCategory ?tenderProcMethod
      ?tenderProcMethodDetails ?tenderSubMethod ?tenderSubMethodDetails
      ?tenderTenderers ?tenderValue ?tenderAmount ?tenderCurrency
      ?tenderPeriod ?tenderPeriodStartDate
      ?tenderPeriodEndDate ?tenderPeriodDurationInDays
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?contractingProcess ocds:hasTender ?tender .
  ?tender ocds:tenderId ?tenderId .
  ?tender dc:title ?tenderTitle .
  OPTIONAL {
    ?tender ocds:mainProcurementCategory ?tenderProcCategory
  } .
  OPTIONAL {
    ?tender ocds:procurementMethod ?tenderProcMethod
  } .
  OPTIONAL {
    ?tender ocds:procurementMethodDetails ?tenderProcMethodDetails
  } .
  OPTIONAL {
    ?tender ocds:submissionMethod ?tenderSubMethod
  } .
  OPTIONAL {
    ?tender ocds:submissionMethodDetails ?tenderSubMethodDetails
  } .
  OPTIONAL {
    ?tender ocds:numberOfTenderers ?tenderTenderers
  } .
  OPTIONAL {
    ?tender ocds:hasMaxEstimatedValue ?tenderValue .
    ?tenderValue ocds:valueAmount ?tenderAmount .
    ?tenderValue ocds:valueCurrency ?tenderCurrency .
  } .
  OPTIONAL {
    ?tender ocds:hasTenderPeriod ?tenderPeriod
    OPTIONAL {
      OPTIONAL {
        ?tenderPeriod ocds:periodStartDate ?tenderPeriodStartDate
      } .
      OPTIONAL {
        ?tenderPeriod ocds:periodEndDate ?tenderPeriodEndDate
      } .
      OPTIONAL {
        ?tenderPeriod ocds:periodDurationInDays
          ?tenderPeriodDurationInDays
      } .
    } .
  } .
} .
}

```


9.3.8 Consulta para la búsqueda de artículos y lotes de licitaciones

```
SELECT ?tenderId ?tenderItem ?tenderItemClassification ?tenderItemScheme
      ?tenderItemDescription ?tenderLot ?tenderLotTitle ?tenderLotValue
      ?tenderLotAmount ?tenderLotCurrency
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?contractingProcess ocds:hasTender ?tender .
  ?tender ocds:tenderId ?tenderId .
  ?tender ocds:hasItemToBeProcured ?tenderItem .
  ?tenderItem ocds:hasClassification ?tenderItemClassification .
  ?tenderItem ocds:classificationScheme ?tenderItemScheme .
  ?tenderItem dc:description ?tenderItemDescription .
  ?tenderItem ocdsext:relatedLot ?tenderLot .
  ?tenderLot ocdsext:lotId ?tenderLotId .
  ?tenderLot dc:title ?tenderLotTitle .
  OPTIONAL {
    ?tenderLot ocdsext:hasLotValue ?tenderLotValue .
    ?tenderLotValue ocds:valueAmount ?tenderLotAmount .
    ?tenderLotValue ocds:valueCurrency ?tenderLotCurrency .
  } .
}
```

9.3.9 Consulta para la búsqueda de organizaciones involucradas

```

SELECT ?party ?partyName ?partyId ?partyIdScheme ?partyAddress
      ?partyAddressCountry ?partyAddressLocality ?partyAddressPostalCode
      ?partyAddressStreet ?partyContact ?partyContactEmail
      ?partyContactName ?partyContactFax ?partyContactTelephone
      ?partyContactUrl
WHERE {
  ?contractingProcess a ocds:ContractingProcess .
  ?party ocds:playsRoleIn ?contractingProcess .
  ?party ocds:legalName ?partyName .
  ?party org:identifier ?partyId .
  OPTIONAL {
    ?party skos:notation ?partyIdScheme .
  } .
  OPTIONAL {
    ?party ocds:hasAddress ?partyAddress .
    OPTIONAL {
      ?partyAddress schema:addressCountry ?partyAddressCountry
    } .
    OPTIONAL {
      ?partyAddress schema:addressLocality ?partyAddressLocality
    } .
    OPTIONAL {
      ?partyAddress schema:postalCode ?partyAddressPostalCode
    } .
    OPTIONAL {
      ?partyAddress schema:streetAddress ?partyAddressStreet
    } .
  } .
  OPTIONAL {
    ?party ocds:hasContactPoint ?partyContact .
    OPTIONAL {
      ?partyContact schema:email ?partyContactEmail
    } .
    OPTIONAL {
      ?partyContact schema:name ?partyContactName
    } .
    OPTIONAL {
      ?partyContact schema:faxNumber ?partyContactFax
    } .
    OPTIONAL {
      ?partyContact schema:telephone ?partyContactTelephone
    } .
    OPTIONAL {
      ?partyContact schema:URL ?partyContactUrl
    } .
  } .
}

```