

# E2.1 SERVICIOS Y ESQUEMA DEL CATÁLOGO

ESPACIO DE DATOS LINGÜÍSTICO (SER 15/23 OTT)

## Resumen

Este documento actualiza los servicios del conector Inesdata reportados en la versión previa incluyendo servicios de borrado, actualización, y recuperación de datos. Estos servicios se invocan desde la interfaz de usuario del conector del espacio de datos lingüísticos y permiten la gestión de activos, políticas y contratos, así como la consulta del catálogo del espacio de datos y el inicio y finalización de negociaciones. Finalmente, se reporta el esquema del catálogo interno de cada conector.

Andrés García-Silva

19/06/2024

Expert.ai Expert.ai Language Technology Research Lab

## Historia

revisión	Fecha	Descripción	Autor
0.1	19/06/2024	Primera versión completa	Andrés García-Silva
0.2	25/06/2024	Revisión del documento	José Manuel Gómez

## Tabla de Contenido

1	Introducción.....	4
2	Visualización del plano de datos .....	4
3	Gestión y búsqueda de activos .....	5
3.1	Recuperar todos los activos .....	5
3.2	Recuperación de un activo por su id .....	6
3.3	Actualización de un activo:.....	7
3.4	Borrado de un activo .....	7
4	Gestión y búsqueda de políticas .....	7
4.1	Recuperación de todas las políticas.....	8
4.2	Recuperación de una política por identificador .....	9
4.3	Actualización de una política.....	9
4.4	Borrado de una política .....	10
5	Gestión y búsqueda de contratos.....	10
5.1	Recuperación de todos los contratos .....	10
5.2	Recuperación de contrato por identificador.....	11
5.3	Actualización de un contrato .....	12
5.4	Borrado de un contrato.....	12
6	Esquema del catálogo.....	12
7	Conclusiones.....	13

## 1 Introducción

La interfaz de usuario del espacio de datos lingüístico es la puerta de entrada de los usuarios al espacio de datos. Esta interfaz se conecta al conector del usuario en el espacio de datos que es el componente de software encargado de realizar las gestiones propias de los activos del usuario y la negociación y transferencia de activos gestionados por otros conectores.

En la primera versión de este entregable se describió el flujo de trabajo normal del conector dependiendo de si el usuario actuaba como un proveedor de activos o un consumidor, y los servicios del conector que soportaban dichos flujos de trabajo. Sin embargo, además de los servicios de inicio de sesión solo se incluyó los servicios de i) creación de plano de datos, activos, políticas y contratos, ii) de consulta del catálogo de activos, iii) de inicio de negociaciones y transferencia de activos. En esta versión del documento se describen servicios de recuperación y gestión de recursos tales como actualización y borrado.

Además, se incluye el esquema de datos del catálogo interno de cada conector. El catálogo interno se basa en la especificación DCAT<sup>1</sup>. Cada conector expone un API que permite consultar este catálogo. Así, el nodo del catálogo federado que cada conector tiene puede consultar los catálogos internos de cada conector del espacio de datos, y generar una visualización unificada de todos los activos disponibles en el espacio de datos.

## 2 Visualización del plano de datos

El plano de datos es donde los datos residen y se transfieren. El conector de Inesdata soporta transferencia de datos usando el protocolo Http y servicios de almacenamiento en la nube como AmazonS3 que ofrecen una interfaz de servicio web.

En la configuración del conector se define en que puerto se exponen los servicios de gestión que soporta. En este caso el puerto definido para estos servicios, incluyendo el registro del plano de datos, se exponen en el puerto 19193. El servicio de listado del plano de datos es accesible en la ruta /management/v2/dataplanes usando una llamada get.:

```
GET /management/v2/dataplanes HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access token
```

Salida del servicio:

```
[
  {
    "@id": "inesdata-dataplane",
    "@type": "DataPlaneInstance",
    "url": "http://connector-c1:19192/control/transfer",
    "lastActive": 1718729222583,
    "turnCount": 0,
    "properties": {
      "publicApiUrl": "http://connector-c1:19291/public/"
    },
    "allowedSourceTypes": "HttpData",
    "allowedDestTypes": [
```

---

<sup>1</sup> <https://www.w3.org/TR/vocab-dcat-2/>

```
        "HttpProxy",
        "HttpData"
    ],
    "allowedTransferTypes": [],
    "@context": {
        "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
        "edc": "https://w3id.org/edc/v0.0.1/ns/",
        "odrl": "http://www.w3.org/ns/odrl/2/"
    }
}
]
```

### 3 Gestión y búsqueda de activos

El conector expone un servicio para la gestión de activos de activos en la ruta `/management/v3/assets`.

#### 3.1 Recuperar todos los activos

Para recuperar todos los activos se hace una llamada POST al servicio pasando un documento json con criterios de búsqueda y opciones de ordenación:

```
POST /management/v3/assets/request HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access Token
Content-Length: 179

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "offset": 0,
  "limit": 5,
  "sortOrder": "ASC",
  "sortField": "id",
  "filterExpression": []
}
```

El servicio devuelve un json con todos los activos del usuario:

```
[
  {
    "@id": "inesdata-asset",
    "@type": "Asset",
    "properties": {
      "name": "inesdata-asset",
      "id": "inesdata-asset",
      "contenttype": "text/html"
    },
    "dataAddress": {
```

```
        "@type": "DataAddress",
        "proxyPath": "true",
        "type": "HttpData",
        "name": "inesdata-home",
        "baseUrl": "https://inesdata-
project.eu/content/en/index.html"
    },
    "@context": {
        "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
        "edc": "https://w3id.org/edc/v0.0.1/ns/",
        "odrl": "http://www.w3.org/ns/odrl/2/"
    }
}
]
```

### 3.2 Recuperación de un activo por su id

Se usa una llamada get del servicio pasando el id del activo en la ruta como se expone a continuación:

```
GET /management/v3/assets/{id-Activo}/ HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access token
```

El servicio devuelve un json con la información del activo consultado, en este caso inesdata-asset:

```
{
  "@id": "inesdata-asset",
  "@type": "Asset",
  "properties": {
    "name": "inesdata-asset",
    "id": "inesdata-asset",
    "contentType": "text/html"
  },
  "dataAddress": {
    "@type": "DataAddress",
    "proxyPath": "true",
    "type": "HttpData",
    "name": "inesdata-home",
    "baseUrl": "https://inesdata-project.eu/content/en/index.html"
  },
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
    "edc": "https://w3id.org/edc/v0.0.1/ns/",
    "odrl": "http://www.w3.org/ns/odrl/2/"
  }
}
```

### 3.3 Actualización de un activo:

Para actualizar un activo es necesario hacer una llamada PUT al servicio enviando un documento json indicando el id del activo a actualizar y la información actualizada del activo: nombre, tipo del activo, y la dirección donde se encuentra el activo. Esta dirección incluye el tipo de dirección que define si es un recurso disponible vía http o si está en un repositorio s3. Para los recursos web se especifica la url donde se encuentra y para los recursos en S3 se especifica el bucket y el nombre del fichero entre otros.

Para actualizar un activo se usa la siguiente llamada.

```
PUT /management/v3/assets HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access token

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@id": "inesdata-asset",
  "properties": {
    "name": "inesdata-asset mod",
    "contentType": "text/html"
  },
  "dataAddress": {
    "type": "HttpData",
    "name": "inesdata-home",
    "baseUrl": "https://inesdata-project.eu/content/en/index.html",
    "proxyPath": "true"
  }
}
```

### 3.4 Borrado de un activo

Para borrado de un activo se hace una llamada Delete al servicio indicando en la ruta el id del activo a borrar como se expone a continuación:

```
DELETE /management/v3/assets/{ID-ACTIVO}/ HTTP/1.1
Host: localhost:19193
Authorization: Bearer
```

Si el borrado ha sido exitoso se devuelve el código 204.

## 4 Gestión y búsqueda de políticas

El servicio de gestión de políticas está disponible en la ruta /management/v2/policydefintions.

#### 4.1 Recuperación de todas las políticas

Para recuperar todas las políticas del usuario se hace una llamada POST al servicio pasando un documento json con criterios de búsqueda y opciones de ordenación:

```
POST /management/v2/policydefinitions/request HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access Token

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "offset": 0,
  "limit": 5,
  "sortOrder": "ASC",
  "sortField": "id",
  "filterExpression": []
}
```

El servicio retorna un listado de políticas:

```
[
  {
    "@id": "always-true",
    "@type": "PolicyDefinition",
    "createdAt": 1718729079726,
    "policy": {
      "@id": "81f33de9-d14b-443c-bdbf-df4a5baea3a7",
      "@type": "odrl:Set",
      "odrl:permission": {
        "odrl:action": {
          "odrl:type": "USE"
        },
        "odrl:constraint": {
          "odrl:leftOperand": "ALWAYS_TRUE",
          "odrl:operator": {
            "@id": "odrl:eq"
          },
          "odrl:rightOperand": "true"
        }
      }
    },
    "odrl:prohibition": [],
    "odrl:obligation": []
  },
]
```

```
    "@context": {
      "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
      "edc": "https://w3id.org/edc/v0.0.1/ns/",
      "odrl": "http://www.w3.org/ns/odrl/2/"
    }
  },
  {
    "@id": "inesdata-policy",
    "@type": "PolicyDefinition",
    "createdAt": 1718729727949,
    "policy": {
      "@id": "dd549970-fa3c-4f65-9a6e-9ebc47bc1cae",
      "@type": "odrl:Set",
      "odrl:permission": [],
      "odrl:prohibition": [],
      "odrl:obligation": []
    },
    "@context": {
      "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
      "edc": "https://w3id.org/edc/v0.0.1/ns/",
      "odrl": "http://www.w3.org/ns/odrl/2/"
    }
  }
]
```

## 4.2 Recuperación de una política por identificador

Para recuperar una política determinada se hace una llamada Get al servicio y se pasa el identificador de la política en la ruta:

```
GET /management/v2/policydefinitions/{Identificador-Política} HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access Token
```

## 4.3 Actualización de una política

Para actualizar una política se hace una llamada PUT al servicio y se envía un documento json con la información de la política a actualizar.

```
PUT /management/v2/policydefinitions HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access token

{
  "@context": {
```

```
"@vocab": "https://w3id.org/edc/v0.0.1/ns/",
"odrl": "http://www.w3.org/ns/odrl/2/"
},
"@id": "inesdata-policy",
"policy": {
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Set",
  "permission": [],
  "prohibition": [],
  "obligation": []
}
}
```

En la versión actual el servicio tiene un bug y retorna un error 405.

#### 4.4 Borrado de una política

Para borrar una política se hace una llamada Delete al servicio y se pasa el identificador de la política en la ruta como se expone a continuación:

```
DELETE /management/v2/policydefinitions/{identificador-política}
HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access Token
```

En caso de éxito el servicio retorna un código 204.

## 5 Gestión y búsqueda de contratos

El servicio de creación de un contrato está disponible en la ruta `/management/v2/contractdefinitions`

### 5.1 Recuperación de todos los contratos

Para recuperar todos los contratos del usuario se hace una llamada POST al servicio pasando un documento json con criterios de búsqueda y opciones de ordenación:

```
POST /management/v2/contractdefinitions/request HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access Token

{
  "@context": {
```

```
"@vocab": "https://w3id.org/edc/v0.0.1/ns/"
},
"offset": 0,
"limit": 5,
"sortOrder": "ASC",
"sortField": "id",
"filterExpression": []
}
```

El servicio retorna un documento json con la lista de contratos definidos:

```
[
  {
    "@id": "in-2024-contract",
    "@type": "ContractDefinition",
    "accessPolicyId": "in-2024-policy",
    "contractPolicyId": "in-2024-policy",
    "assetsSelector": [],
    "@context": {
      "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
      "edc": "https://w3id.org/edc/v0.0.1/ns/",
      "odrl": "http://www.w3.org/ns/odrl/2/"
    }
  }
]
```

## 5.2 Recuperación de contrato por identificador

Para recuperar un contrato determinado se hace una llamada Get al servicio y se pasa el identificador de la política en la ruta:

```
GET /management/v2/contractdefinitions/{identificador-contrato}
HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access Token
```

El servicio retorna la información del contrato:

```
{
  "@id": "identificador-contrato",
  "@type": "ContractDefinition",
  "accessPolicyId": "in-2024-policy",
  "contractPolicyId": "in-2024-policy",
  "assetsSelector": [],
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
    "edc": "https://w3id.org/edc/v0.0.1/ns/",

```

```
"odrl": "http://www.w3.org/ns/odrl/2/"
  }
}
```

### 5.3 Actualización de un contrato

Para actualizar un contrato se llama el servicio usando PUT y pasando un documento json con el identificador del contrato y la información que se quiera actualizar.

```
PUT /management/v2/contractdefinitions HTTP/1.1
Host: localhost:19193
Content-Type: application/json
Authorization: Bearer Access Token

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@id": "inesdata-contract",
  "accessPolicyId": "inesdata-policy",
  "contractPolicyId": "inesdata-policy",
  "assetsSelector": []
}
```

El servicio retorna Código 204 si la actualización fue exitosa

### 5.4 Borrado de un contrato

Para borrar un contrato se llama el servicio usando DELETE y se pasa el identificador del contrato en la ruta:

```
DELETE /management/v2/contractdefinitions/{identificador-contrato}
HTTP/1.1
Host: localhost:19193
Authorization: Bearer Access token
```

## 6 Esquema del catálogo

El catálogo del espacio de datos lingüísticos sigue las directrices definidas para el conector de INESDATA donde cada conector tiene un nodo del catálogo federado. Este nodo del catálogo se carga por medio de programas de ejecución programada que consultan los catálogos internos de cada conector en el espacio de datos. El nodo del catálogo federado no se persiste y se tiene que actualizar continuamente con la información que provea cada participante del espacio de datos.

El catálogo interno de cada conector contiene los activos que el usuario pone a disposición de los otros usuarios del espacio de datos. El catálogo interno se modela por medio de clases y

atributos en el código del conector. Sin embargo, se prevé que esta información se persista en una base de datos con el esquema que se presenta en la Figura 1.

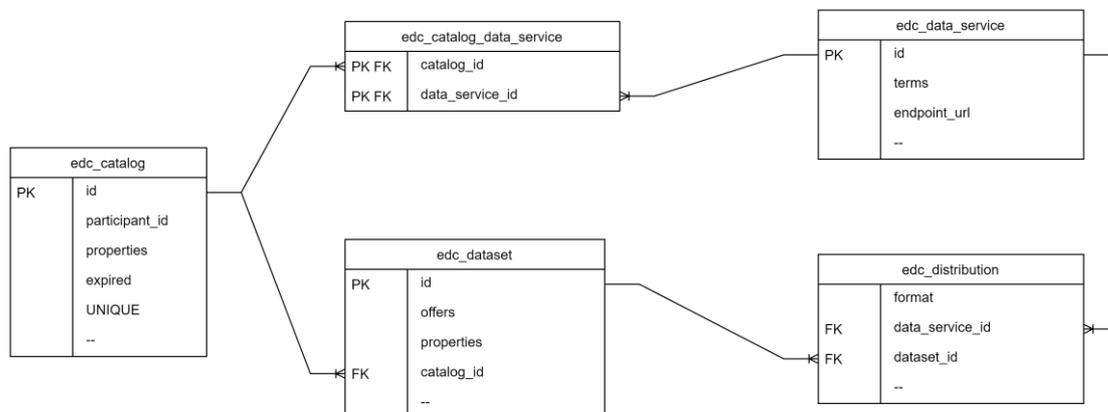


Figura 1. Diagrama entidad relación del catálogo interno de cada conector en el espacio de datos lingüístico.

El catálogo del participante sigue la especificación DCAT<sup>2</sup>. En DCAT un catálogo tiene asociado uno o varios DataService<sup>3</sup>, que son una colección de operaciones que proveen acceso a los conjuntos de datos. En otras palabras, un DataService define el punto de conexión (url) donde se puede iniciar la negociación de un contrato o la transferencia de un activo.

Además, el catálogo tiene asociados uno o más Datasets<sup>4</sup>. Un conjunto de datos es una colección de datos disponibles para acceso o descarga. Los datos incluidos pueden ofrecerse con políticas distintas y puede estar disponibles en distribuciones diferentes.

Finalmente, la distribución<sup>5</sup> se una representación particular de un Dataset. La distribución está enlazada al DataService por el que la distribución está disponible.

## 7 Conclusiones

Se actualizado el documento con los servicios de gestión de activos, políticas y contratos mas allá de los servicios de creación reportados en la primera versión. Los servicios de gestión incluyen servicios de actualización y borrado, así como servicios de recuperación por identificados o criterios de búsqueda. Además, se presenta el esquema del catálogo interno de cada conector. Este catálogo interno de cada participante se consulta regularmente para poblar el nodo del catálogo federado que cada conector incluye.

<sup>2</sup> <https://www.w3.org/TR/vocab-dcat-2/>

<sup>3</sup> [https://www.w3.org/TR/vocab-dcat-3/#Class:Data\\_Service](https://www.w3.org/TR/vocab-dcat-3/#Class:Data_Service)

<sup>4</sup> <https://www.w3.org/TR/vocab-dcat-3/#Class:Dataset>

<sup>5</sup> <https://www.w3.org/TR/vocab-dcat-3/#Class:Distribution>