E6.1V2 DEMOSTRADORES Y REPORTES

ESPACIO DE DATOS LINGÜÍSTICO (SER 15/23 OTT)

Resumen

Este documento describe los demostradores del espacio de datos que implementan casos de uso definidos en el proyecto. Estos casos de uso se apoyan en el espacio de datos para obtener los datos necesarios para el análisis de afirmaciones científicas, así como para el uso de modelos cuya factualidad ha sido mejorada. En esta segunda versión del documento se presenta una versión actualizada del análisis de afirmaciones científicas, así como un nuevo demostrador para la descarga de recursos en el espacio de datos lingüístico de forma programática a través del API del conector. Además, se realiza una actualización del reporte de estadísticas del catálogo del espacio de datos que incluye el número de activos por tipo y lenguaje.

Cristian Berrío,
Raúl Ortega,
José Manuel Gómez Pérez,

30 de junio 2025

Expert.ai Expert.ai Language Technology Research Lab

Revision History

Revision	Date	Description	Author (Organisation)
1.0	26/06/2025	Versión inicial del documento	Cristian Berrío

Contenido

1 Introducción

Este documento describe los resultados finales del paquete de trabajo 6 Demostración y reportes. Los objetivos del paquete de trabajo son desarrollar demostradores que usen los servicios y datos del catálogo del espacio de datos lingüístico EDL[1], y generar reportes periódicos con el estado de los recursos almacenados en catálogo del espacio de datos.

Los demostradores permiten la interacción de los usuarios con el catálogo del EDL por ejemplo, consultando activos disponibles, contratar y negociar activos o descargarlos para utilizarlos en una tarea determinada. Estos demostradores se apoyan en los resultados del proyecto KG4LLM[2] en el marco del proyecto en INESData, donde se llevó a cabo una investigación en la evaluación y mejora de la factualidad de los modelos del lenguaje masivos LLM en dominios como el de los seguros, o en el análisis de afirmaciones científicas.

En este entregable se presenta una actualización del demostrador de análisis de afirmaciones científicas que incluye el uso del espacio de datos lingüístico para buscar, negociar y descargar los datos y su posterior procesamiento en un notebook de Jupyter con la ayuda de un LLM. Se ha desarrollado además un demostrador para la búsqueda y descarga de activos de forma programática, mediante el uso del API del conector.

Los reportes están disponibles en la interfaz del conector de EDL permitiendo que todos los miembros del espacio de datos cuenten con la información actualizada. En este documento se realiza una actualización de los recursos presentes en el catálogo a la fecha de presentación de este entregable.

2 Demostradores

Tras revisar los posibles demostradores presentados en el E6.1v1, se ha seleccionado el demostrador de análisis de afirmaciones científicas para su desarrollo por su posible impacto a medio plazo en diversas áreas de investigación de industrias tales como la industria farmacéutica, y en concreto en el análisis automatizado de literatura científica. Por otro lado, más allá de aplicaciones específicas que se pueden construir con los activos contenidos en el espacio de datos, el principal demostrador sigue siendo el conector personalizado desarrollado en el proyecto, que facilita la gestión de estos activos.

De forma complementaria a la descarga de recursos a través de la interfaz, se ha desarrollado un demostrador para hacer esto de forma programática, haciendo llamadas directamente al API del conector.

2.1 Análisis de afirmaciones científicas

En la primera versión del entregable E6.1 se presentó y desarrolló este caso de uso. En esta versión, se hace una revisión para la parte de búsqueda y descarga del recurso lingüístico en el EDL, con capturas de la última versión del interfaz, teniendo en cuenta además que en este caso el conector se encuentra finalmente desplegado en la infraestructura de INESData.

2.1.1 Desarrollo del caso de uso

Búsqueda y descarga del recurso lingüístico en el EDL.

Este caso de uso inicia buscando y descargando el recurso lingüístico “Pubmed Corpus for verification (lite)” usando el conector “conn-consumer”[3] en el EDL. El usuario debe iniciar sesión en el espacio de datos usando la URL y las credenciales que el promotor del espacio de datos le ha enviado al registrarse en la plataforma. Una vez ha ingresado al sistema, debe buscar el

[1] https://labdemos.expertcustomers.ai/edl

[2] https://labdemos.expertcustomers.ai/kg4llm

[3] https://conn-consumer-language.ds.inesdata-project.eu/dataspace

Field Code Changed

activo correspondiente al recurso lingüístico en el catálogo del EDL. Para esto primero debe acceder a la opción “Datasets” en el menú izquierdo y realizar una búsqueda por una palabra clave (ver Figura 1).

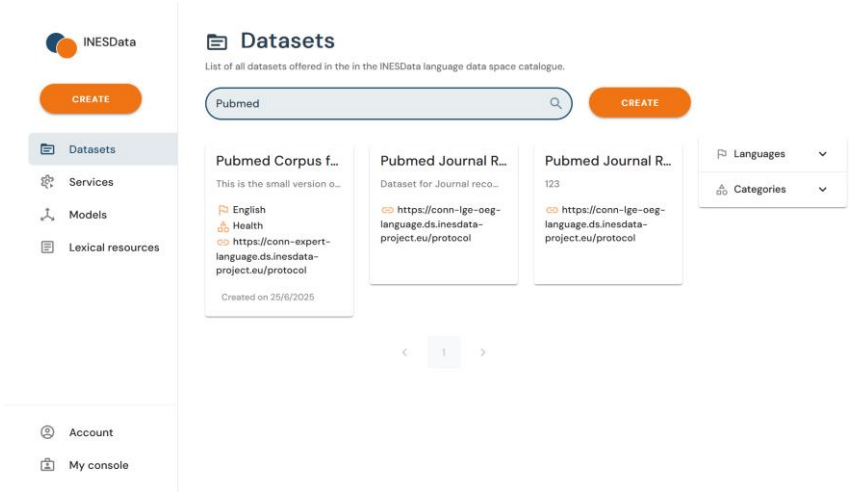

Figura 1. Búsqueda y selección de activo en el catálogo del espacio de datos lingüístico.

Una vez se ha identificado el recurso, el usuario debe negociar el contrato que ha definido el proveedor del activo. En la vista del catálogo del EDL se debe clicar sobre el activo a negociar para que se despliegue la información detallada del activo y las opciones de contratación (ver Figura 2). El usuario puede revisar las condiciones de los diferentes contratos y negociar el que más le interese. La negociación se inicia al clicar sobre el botón negociar del contrato correspondiente.

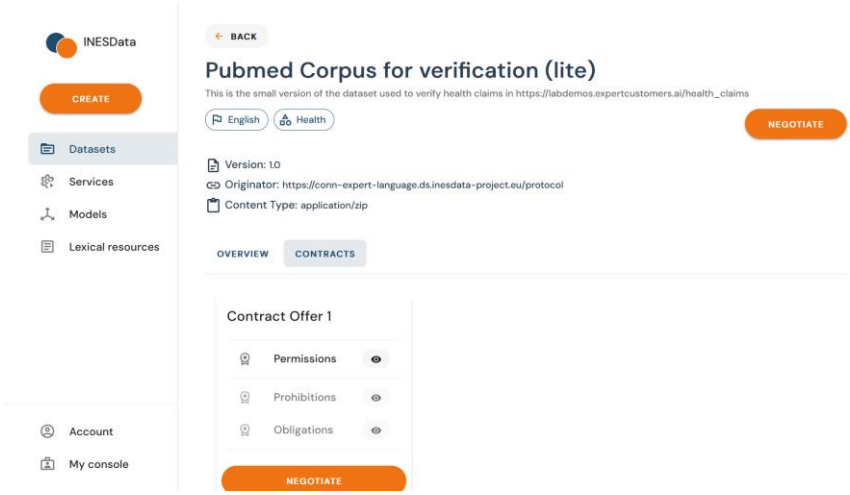

Figura 2. Negociación del contrato asociado al recurso lingüístico

Inmediatamente el sistema lleva al usuario a la vista de las negociaciones que ha iniciado en la opción “Mi consola” (Figura 3). Inicialmente la negociación estará en estado pendiente, pero en cuestión de menos de 1 minuto debería cambiar a estado completado, ya que las políticas definidas en el EDL son automáticamente evaluadas como satisfechas por definición. Para actualizar la vista de negociaciones es necesario clicar en otra opción y regresar a la vista negociaciones.

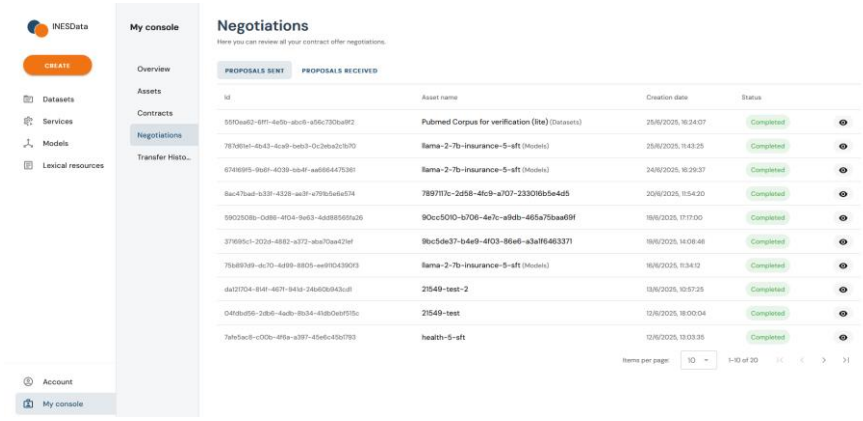

Figura 3. Estado de la negociación

Cuando la negociación se ha completado, se puede iniciar el proceso de transferencia del activo. Este proceso tiene por objetivo generar un endpoint para acceder al recurso lingüístico. Para solicitar la transferencia del activo hay que clicar en la opción ver (icono de ojo en la última columna de la tabla de negociaciones) que corresponde a la negociación. En la pantalla que se despliega se clica en el botón Transfer.

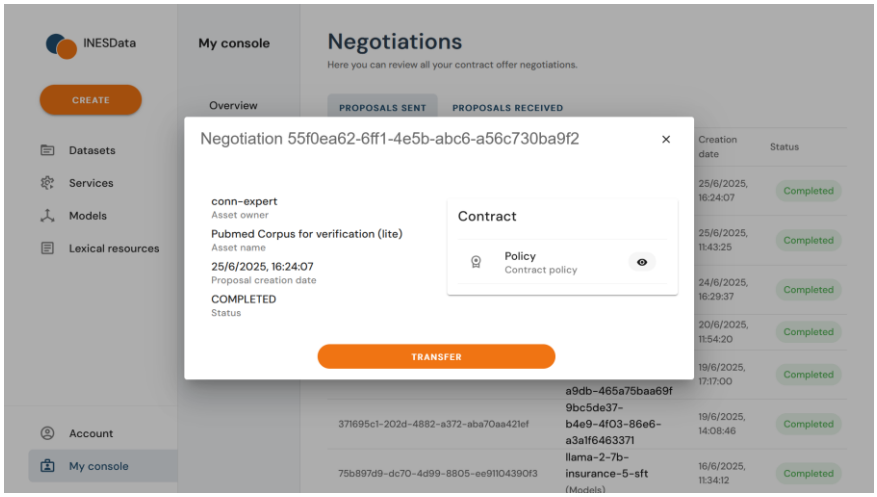

Figura 4. Iniciar transferencia del activo

Finalmente, sí la transferencia ha sido exitosa se habilita el botón download (ver Figura 5) que permite descargar el recurso lingüístico al entorno local del usuario.

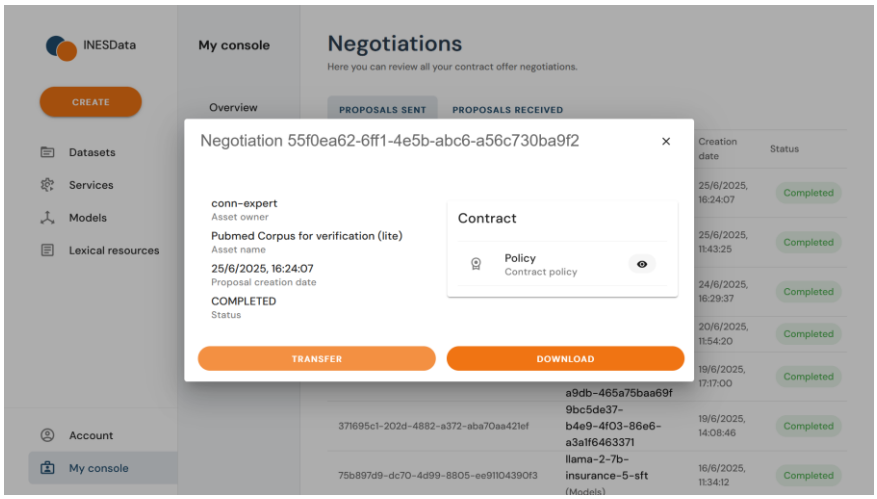

Figura 5. Descarga del recurso lingüístico.

Notebook para el análisis de afirmaciones científicas.

El notebook se mantiene igual que en la primera versión del entregable E6.1, y sigue disponible en el repositorio de demostradores del proyecto, en la siguiente url:

https://github.com/oeg-upm/inesdata-espacio-linguistico-demostradores/blob/main/health_claims.ipynb

Field Code Changed

2.2 Descarga de un modelo en el espacio de datos mediante el API del conector

Con el fin de mostrar la invocación de servicios del catálogo, se ha desarrollado un notebook, el cual ilustra la descarga de un modelo de lenguaje masivo (LLM) disponible en el espacio de datos proveniente del proyecto KG4LLM, todo esto de forma programática a través del API del conector. El notebook está disponible en el repositorio de demostradores del proyecto, en la siguiente url:

https://github.com/INESData/inesdata-espacio-linguistico-demostradores/blob/main/INESData_EDL_model_download.ipynb

Field Code Changed

La primera parte del notebook consiste en la autenticación en el espacio de datos, para ello se debe disponer de las correspondientes credenciales. Este proceso devuelve un token de acceso que será el que se use para la posterior autenticación de cara al conector.

Here we input our credentials, including the One-Time Password (OTP).

```
USER = ""
PASSWORD = ""
OTP = ""
```

We log into the language data space using Keycloak, and this will return an access token.

```
from keycloak import KeycloakOpenID

keycloak_openid = KeycloakOpenID(
    server_url="https://auth.ds.inesdata-project.eu/",
    client_id="dataspace-users",
    realm_name="language"
)

token = keycloak_openid.token(USER, PASSWORD, totp=OTP)
```

Acess token has to be constantly renewed, so we will use this function to check if it has expired, and in if so, refresh it.

Figura 6. Autenticación en el espacio de datos.

En el notebook se muestra una consulta al catálogo federado, para ello se hace la llamada directamente al connector “conn-consumer” utilizando el token de acceso obtenido en el paso anterior.

```
import requests
import json

url = "https://conn-consumer-language.ds.inesdata-project.eu/federatedcatalog/v1alpha/catalog/query"

payload = json.dumps({
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "operandLeft": "",
  "operandRight": "",
  "operator": "",
  "Criterion": ""
})

headers = {
  'Authorization': f'Bearer {get_valid_token()}',
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(json.dumps(response.json(),indent=2))
```

```
[
  {
    "@id": "73aff7d5-9c50-4251-83d6-897732aba8ca",
    "@type": "dcat:Catalog",
    "dcat:dataset": [],
    "dcat:catalog": [],
    "dcat:distribution": [],
    "dcat:service": {
```

Figura 7. Exploración del catálogo federado.

El catálogo federado permite hacer la búsqueda por un texto en particular. En nuestro caso aplicamos un filtro para buscar el modelo que nos interesa descargar.

```
url = "https://conn-consumer-language.ds.inesdata-project.eu/management/federatedcatalog/request"

payload = json.dumps({
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "offset": 0,
  "limit": 100,
  "filterExpression": [{"operandLeft":"genericSearch","operator":"=","operandRight":"llama-2-7b-insurance-5-sft"}]

})

headers = {
  'Authorization': f'Bearer {get_valid_token()}',
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)
filtered_catalog = response.json()
print(json.dumps(filtered_catalog,indent=2))
```

```
    },
    "https://w3id.org/dspace/v0.8/participantId": "conn-expert",
    "originator": "https://conn-expert-language.ds.inesdata-project.eu/protocol",
    "http://www.w3.org/ns/dcat#dataset": {
      "@id": "llama-2-7b-insurance-5-sft",
      "@type": "http://www.w3.org/ns/dcat#Dataset",
      "http://www.w3.org/ns/dcat#distribution": [
        {
          "@type": "http://www.w3.org/ns/dcat#Distribution",
          "http://purl.org/dc/terms/format": {
```

Figura 8. Búsqueda de recurso específico en el catálogo federado.

El siguiente paso consiste en la negociación del contrato, teniendo en cuenta el *policy* del activo en cuestión.

```
url = "https://conn-consumer-language.ds.inesdata-project.eu/management/v3/contractnegotiations"

payload = json.dumps({
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/",
    "odrl": "http://www.w3.org/ns/odrl.jsonld"
  },
  "@type": "ContractRequest",
  "counterPartyAddress": filtered_catalog[0]["http://www.w3.org/ns/dcat#dataset"]['http://www.w3.org/ns/dcat#distribution'][0]['http://www.w3.org/ns/dcat
  "protocol": "dataspace-protocol-http",
  "policy": {
      "@context": "http://www.w3.org/ns/odrl.jsonld",
    **filtered_catalog[0]["http://www.w3.org/ns/dcat#dataset"]['odrl:hasPolicy']['offer'],
    "assigner": filtered_catalog[0]["http://www.w3.org/ns/dcat#dataset"]['participantId'],
    "target": filtered_catalog[0]["http://www.w3.org/ns/dcat#dataset"]['@id']
  }
})
headers = {
  'Authorization': f'Bearer {get_valid_token()}',
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```


```
{"@type":"IdResponse","@id":"787d61e1-4b43-4ca9-beb3-0c2eba2c1b70","createdAt":1750844605140,"@context":{"@vocab":"https://w3id.org/edc/v0.0.1/ns/","ed
c":"https://w3id.org/edc/v0.0.1/ns/","odrl":"http://www.w3.org/ns/odrl/2/"}}
```


Figura 9. Negociación del contrato.

Una vez se ha iniciado la negociación, se puede consultar el estado de esta, y obtener el identificador del acuerdo de contrato.

```
url = f"https://conn-consumer-language.ds.inesdata-project.eu/management/v3/contractnegotiations/{response.json()['@id']}"

payload = ""
headers = {
  'Authorization':  f'Bearer {get_valid_token()}'
}

response = requests.request("GET", url, headers=headers, data=payload)

print(response.text)
```


```
{"@type":"ContractNegotiation","@id":"787d61e1-4b43-4ca9-beb3-0c2eba2c1b70","type":"CONSUMER","protocol":"dataspace-protocol-http","state":"FINALIZE
D","counterPartyId":"conn-expert","counterPartyAddress":"https://conn-expert-language.ds.inesdata-project.eu/protocol","callbackAddresses":[],"createdA
t":1750844605140,"contractAgreementId":"248a2293-34ea-4f6a-8d29-98397c8aa367","@context":{"@vocab":"https://w3id.org/edc/v0.0.1/ns/","edc":"https://w3i
d.org/edc/v0.0.1/ns/","odrl":"http://www.w3.org/ns/odrl/2/"}}
```


Figura 10. Consulta del estado de la negociación del contrato.

Es el momento de hacer la petición de transferencia del activo, en esta petición se indica cómo se quiere hacer la transferencia, en este caso es de tipo “HttpData-PULL”, es decir que se quiere hacer la descarga directamente, sin necesidad de transmitirlo o subirlo a otro sitio.

```
url = "https://conn-consumer-language.ds.inesdata-project.eu/management/v3/transferprocesses"

payload = json.dumps({
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "TransferRequestDto",
  "connectorId": "conn-consumer-language",
  "counterPartyAddress": response.json()["counterPartyAddress"],
  "contractId":response.json()["contractAgreementId"],
  "assetId": filtered_catalog[0]["http://www.w3.org/ns/dcat#dataset"]['@id'],
  "protocol": "dataspace-protocol-http",
  "transferType": "HttpData-PULL"
})
headers = {
  'Authorization': f'Bearer {get_valid_token()}',
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

transferId = response.json()['@id']
```


```
{"@type":"IdResponse","@id":"5b32fa23-cad7-47ce-ba5d-5c9268ffc4e5","createdAt":1750844829782,"@context":{"@vocab":"https://w3id.org/edc/v0.0.1/ns/","ed
c":"https://w3id.org/edc/v0.0.1/ns/","odrl":"http://www.w3.org/ns/odrl/2/"}}
```


Figura 11. Descarga del recurso lingüístico.

Si todo ha ido correctamente, con la siguiente llamada se puede consultar el *endpoint* para la descarga del activo. Esto devolverá a su vez un token de autenticación que pasarle a la hora de llamar al *endpoint*.

```
url = f"https://conn-consumer-language.ds.inesdata-project.eu/management/v3/edrs/{transferId}/dataaddress"

payload = ""
headers = {
  'Authorization': f'Bearer {get_valid_token()}'
}

response = requests.request("GET", url, headers=headers)

print(response.text)
```

Figura 12. Obtención del endpoint para la recarga del recurso.

Finalmente haciendo la llamada al *endpoint* nos podemos descargar el activo en cuestión.

```
url = response.json()["endpoint"]

payload = ""
headers = {
  'Authorization': response.json()["authorization"]
}

r = requests.request("GET", url, headers=headers, data=payload, stream=True)

with open("download.zip", 'wb') as fd:
    for chunk in r.iter_content(chunk_size=1024):
        fd.write(chunk)
```

Once downloaded, the file can be extracted and the model can be used.

```
!unzip download.zip
```

```
Archive:  download.zip
   creating: llama-2-7b-insurance-5-sft/
  inflating: llama-2-7b-insurance-5-sft/adapter_model.safetensors
  inflating: llama-2-7b-insurance-5-sft/tokenizer.json
  inflating: llama-2-7b-insurance-5-sft/adapter_config.json
  inflating: llama-2-7b-insurance-5-sft/training_args.bin
  inflating: llama-2-7b-insurance-5-sft/README.md
  inflating: llama-2-7b-insurance-5-sft/tokenizer_config.json
  inflating: llama-2-7b-insurance-5-sft/special_tokens_map.json
```

Figura 13. Descarga del recurso lingüístico.

Al tratarse de un archivo ZIP en este caso, se puede extraer el contenido, y finalmente se dispondrá del modelo listo para su uso posterior.

3 Reportes

En esta sección se hace una actualización del reporte de recursos disponibles en el espacio de datos lingüístico, con respecto a la primera versión del entregable E6.1.

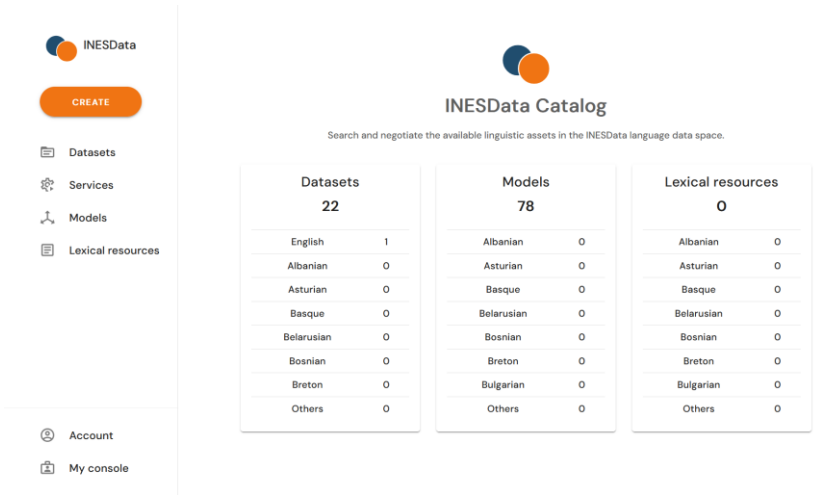

Figura 14. Versión actualizada del reporte de estadísticas del catálogo del EDL. Este reporte se despliega al acceder a la interfaz del conector y cuando se da click sobre el logo de INESData en el menú de la izquierda.

Este reporte tiene previsto ser actualizado, a falta de añadir recursos pendientes provenientes del European Language Grid, así como de algunos datasets provenientes del proyecto KG4LLM.