

# E3.1V2 INTERFAZ INTERACTIVA

ESPACIO DE DATOS LINGÜÍSTICO (SER 15/23 OTT)

## Resumen

Para facilitar la interacción entre los usuarios del espacio de datos lingüísticos se ha desarrollado una versión inicial de la interfaz de usuario. Esta interfaz media entre los usuarios y los artefactos subyacentes del espacio de datos entre los que se incluye el conector de INESData y el catálogo. La interfaz de usuario facilita la visualización, navegación, y búsqueda de activos disponibles en el espacio de datos y los procesos de negociación. Además, ofrece soporte a los procesos de gestión, es decir creación, visualización, actualización y borrado, de sus propios activos, políticas, y contratos, entre otras funcionalidades.

Andrés García-Silva  
Matteo Perissinotto  
Eduard Capdevila

27/08/2024  
Expert.ai Research Lab

## Revision History

Revision	Date	Description	Author (Organisation)
0.1	23/02/2024	Versión inicial del documento	Matteo Perissinotto
0.2	26/02/2024	Adaptación a la plantilla del proyecto	Andrés García Silva
1.0	29/02/2024	Versión final después de la revisión interna	Andrés García Silva
2.0	23/08/2024	Añadida implementación de los métodos de la API del conector INESData	Eduard Capdevila

## Contents

1	Introducción.....	4
2	Interfaz interactiva.....	4
2.1	Catálogo.....	4
2.2	Consola.....	5
2.3	Flujos de navegación y opciones de visualización .....	6
3	Servicios API implementados.....	7
3.1	Catálogo.....	7
3.1.1	Listado de activos del catálogo.....	7
3.2	Gestión de Activos .....	8
3.2.1	Listado de activos .....	8
3.2.2	Detalle de un activo .....	9
3.2.3	Creación de un activo .....	9
3.2.4	Actualización de un activo .....	9
3.2.5	Borrado de un activo.....	10
3.3	Gestión de Pólizas .....	10
3.3.1	Listado de definiciones de pólizas .....	10
3.3.2	Detalle de una definición de póliza .....	11
3.3.3	Creación de una definición de póliza .....	11
3.3.4	Actualización de una definición de póliza .....	12
3.3.5	Borrado de una definición de póliza.....	12
3.4	Gestión de Contratos .....	13
3.4.1	Listado de definiciones de contratos.....	13
3.4.2	Detalle de una definición de contrato.....	13
3.4.3	Creación de una definición de contrato.....	13
3.4.4	Actualización de una definición de contrato.....	14
3.4.5	Borrado de una definición de contrato .....	14
3.5	Negociaciones.....	14
3.5.1	Listado de negociaciones de contratos.....	14
3.5.2	Detalle de una negociación de contrato.....	15
3.5.3	Detalle de un acuerdo de contrato .....	15
3.5.4	Creación de una negociación de contrato.....	16
4	Herramientas y tecnologías usadas .....	17
5	Repositorio.....	17
6	Conclusiones.....	17

## 1 Introducción

Este documento presenta la segunda versión de la interfaz interactiva del catálogo del espacio de datos lingüísticos. La interfaz se diseñó teniendo en cuenta los principios que rigen los espacios de datos definidos por la asociación internacional de espacios de datos. Además, se ha tenido en cuenta en su diseño el vocabulario ELG schema usado para describir recursos en el European Language Grid. Este vocabulario es la base sobre la que se construye el modelo de datos asociado a la descripción de los activos en el espacio de datos. ELG schema incluye una taxonomía para describir conjuntos de datos, servicios, y modelos, así como metadatos descriptivos de cada activo y su función esperada.

## 2 Interfaz interactiva

La interfaz del espacio de datos se compone de dos componentes principales: El catálogo y la consola del usuario. El catálogo le permite a un usuario que haya iniciado sesión ver, navegar y buscar los activos con ofertas de contrato que estén disponibles en el espacio de datos. La consola del usuario es la vista donde el usuario puede gestionar todos sus recursos incluyendo activos, políticas, contratos, así como ver la historia de sus transacciones en el espacio de datos.

### 2.1 Catálogo

El catálogo sirve como puerta de entrada principal a la plataforma, ofreciendo a los usuarios un punto de partida conveniente para navegar por sus contenidos (ver Figura 1). Ubicados en el menú de la izquierda, los usuarios pueden localizar y seleccionar fácilmente el tipo específico de activo que desean explorar o buscar. Al realizar una selección, el área central de la interfaz se actualiza dinámicamente para mostrar una sección dedicada a mostrar los resultados de la búsqueda y las opciones de filtrado correspondientes. Dentro de esta sección, a los usuarios se les presenta una variedad de activos organizados visualmente, representados a través de tarjetas individuales. Estas tarjetas brindan una visión sucinta pero informativa de los diversos aspectos y características de cada activo, lo que facilita la navegación eficiente y los procesos de toma de decisiones.

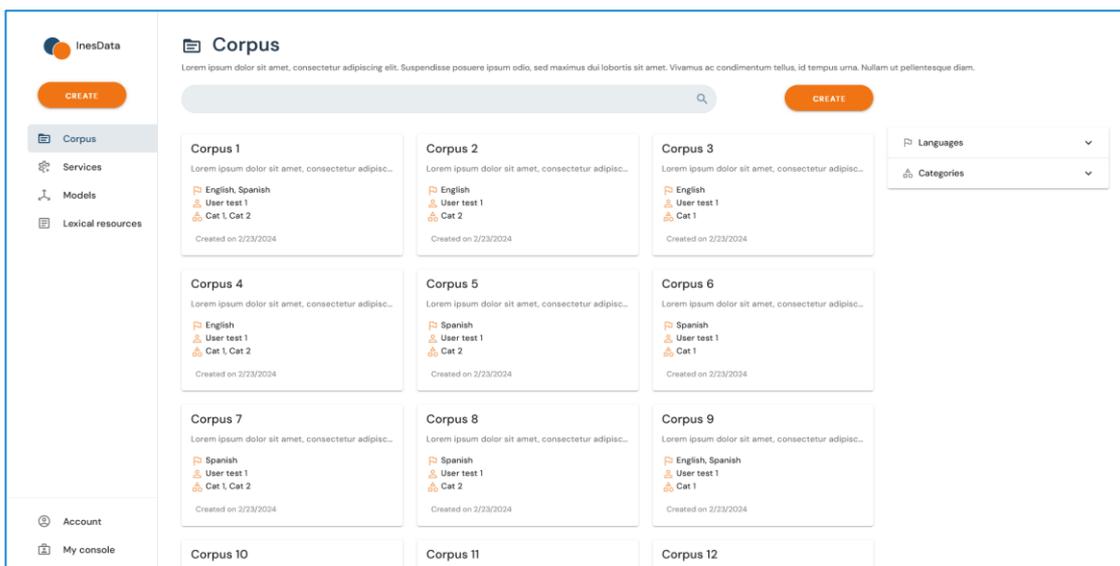


Figura 1. Vista del catálogo del espacio de datos.

Al hacer clic en la tarjeta del activo, se abrirá una página dedicada a los detalles del activo seleccionado que muestra información como el nombre, una breve descripción, algunas

etiquetas como categorías e idiomas, una descripción textual completa, archivos adjuntos y el contrato relacionado (ver Figura 2).

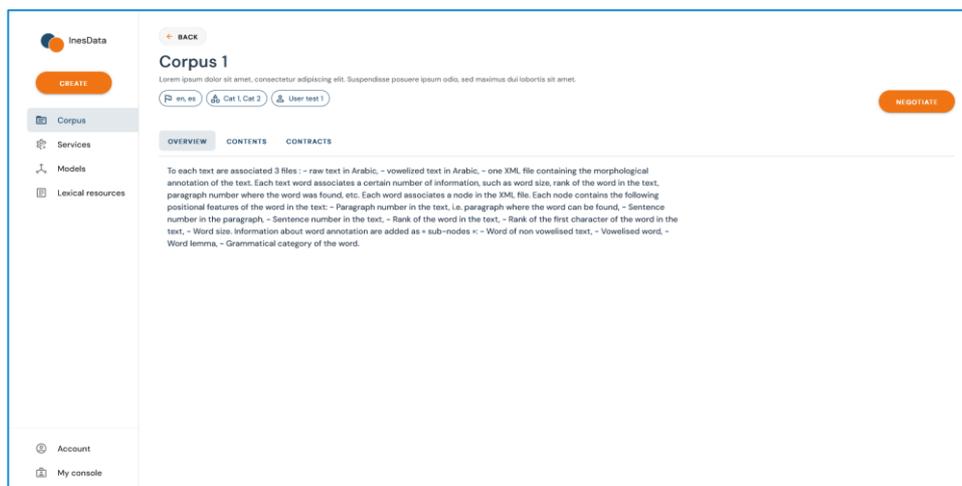


Figura 2. Visualización del detalle de un activo en el catálogo

## 2.2 Consola

La consola es una parte fundamental de la interfaz de usuario (ver Figura 3), ofreciendo a los usuarios un conjunto completo de herramientas y funcionalidades para administrar eficazmente sus activos. En este entorno, los usuarios pueden ver sus activos existentes y emprender diversas acciones para gestionar su colección de activos. Esto incluye tareas como editar detalles de activos, actualizar información de propiedad y monitorear las estadísticas de los activos.

Además, la consola proporciona a los usuarios un punto centralizado para supervisar y administrar contratos y políticas vinculados a sus activos. A través de esta interfaz rica en funciones, los usuarios pueden navegar sin problemas a través de los términos del contrato, revisar los detalles de la política y ejecutar las modificaciones necesarias.

La consola también permite a los usuarios visualizar las negociaciones de sus propios activos o los que ha solicitado a otros participantes del espacio de datos. Aquí, los usuarios pueden acceder a una sección dedicada donde pueden visualizar la historia de negociaciones relacionadas con sus activos.

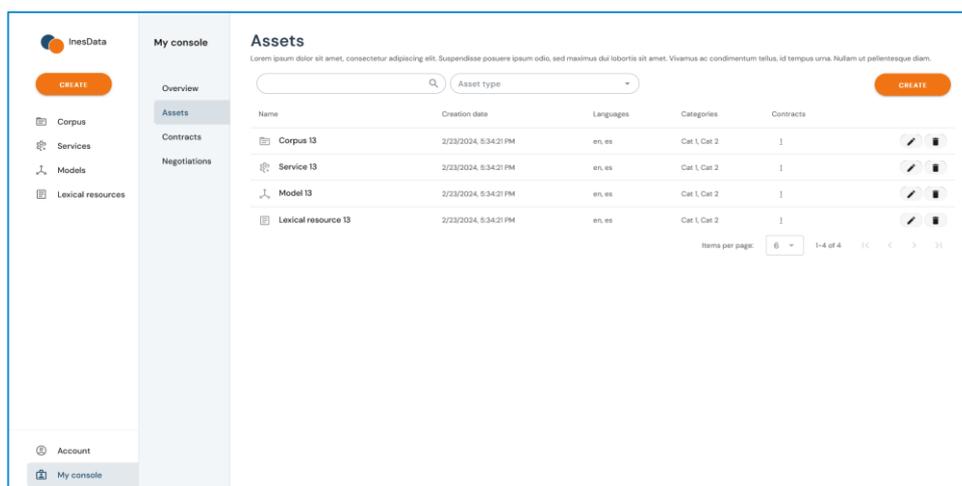


Figura 3. Visualización de la consola de usuario

### 2.3 Flujos de navegación y opciones de visualización

La Figura 4 presenta el flujo de navegación de las páginas del catálogo. En este flujo se puede ver como el catálogo permite visualizar los activos de acuerdo con su tipo (corpus, servicio, modelo, recurso léxico) y realizar una búsqueda usando palabras claves. El usuario tiene la opción de seleccionar uno de los activos que resulten de los filtros aplicados o la búsqueda realizada para visualizar los detalles.

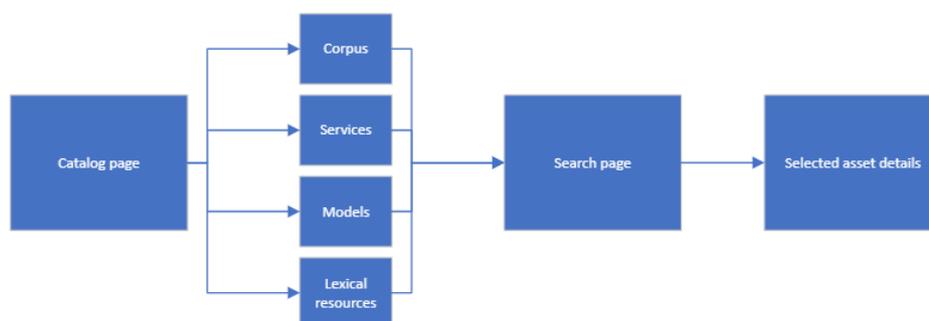


Figura 4. Flujo de navegación entre páginas del catálogo

En la Figura 5 se presenta el flujo de navegación de la consola de usuario. Se puede ver que la consola despliega una página de visión general y ofrece opciones para gestionar activos, contratos, y visualizar negociaciones. Cada una de estas opciones lleva a las páginas correspondientes que dan soporte a los procesos de gestión.

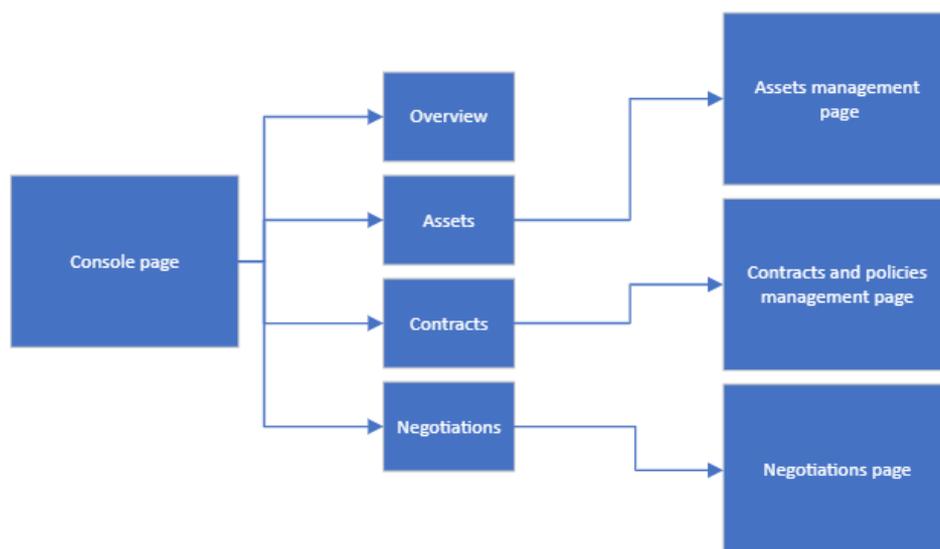


Figura 5. Flujo de navegación entre páginas de la consola del usuario

### 3 Servicios API implementados

En el siguiente apartado se describen los diferentes servicios de la API del conector INESData que se han usado en la implementación de esta versión de la Interfaz web.

#### 3.1 Catálogo

##### 3.1.1 Listado de activos del catálogo

API URL	<b>/management/federatedcatalog</b>
Método	POST
Descripción	<p>Devuelve todos los activos del catálogo que tengan una oferta de contrato asociada. En la interfaz web se muestran en tres listados separados, uno para cada tipo de activos:</p> <ul style="list-style-type: none"> <li>• Corpus</li> <li>• Modelos</li> <li>• Recursos lingüísticos.</li> </ul> <p>* La versión de la API usada en esta entrega no permite el filtrado de los activos del catálogo mediante una consulta, por lo que la búsqueda de activos por nombre y/o metadatos se ha implementado mediante JavaScript en la propia interfaz web</p>

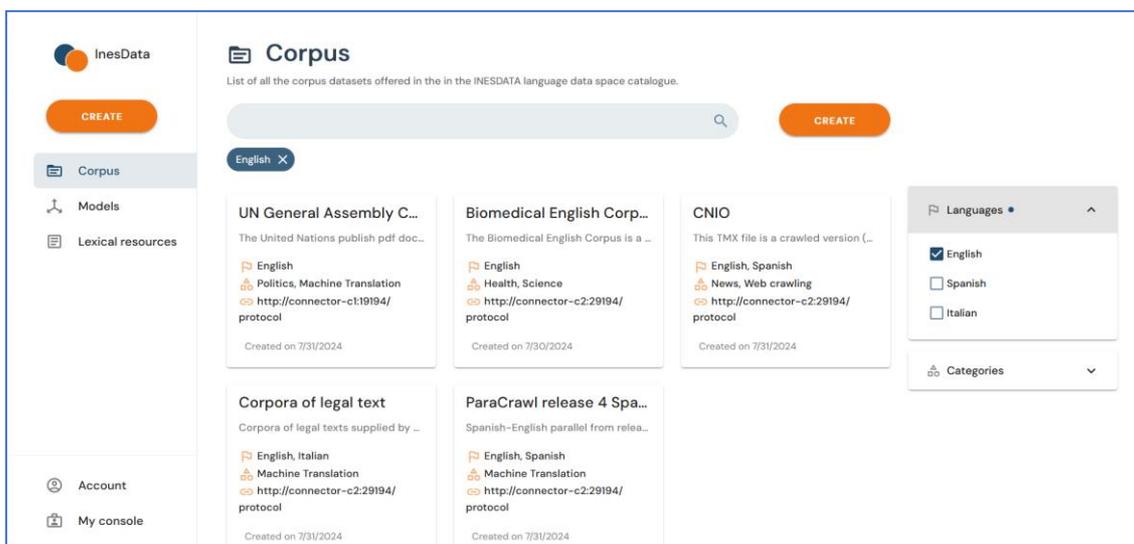


Figura 6. Búsqueda del catálogo del espacio de datos

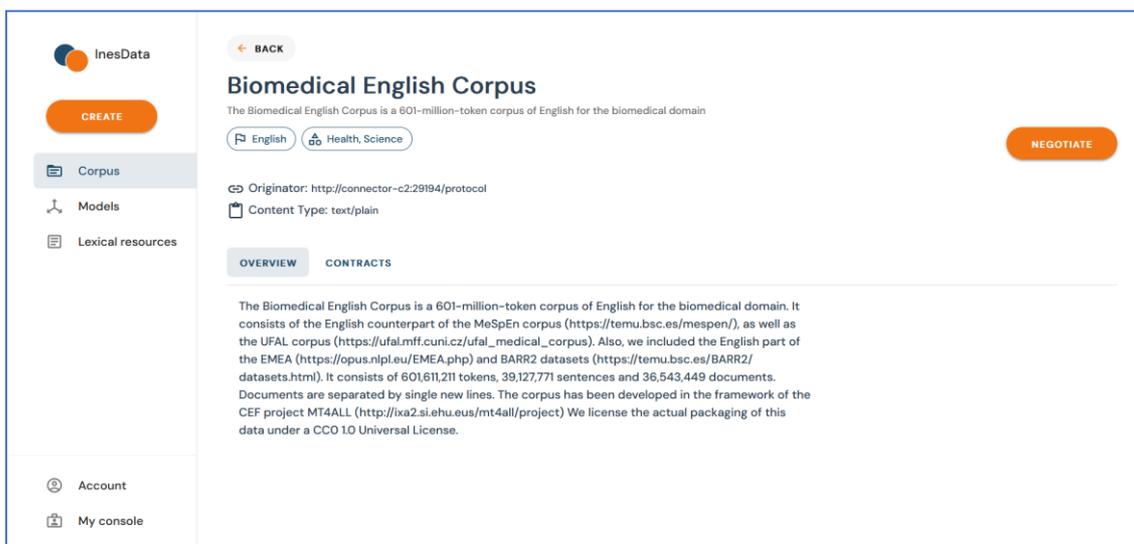


Figura 7. Visualización del detalle de un activo en el catálogo

## 3.2 Gestión de Activos

### 3.2.1 Listado de activos

API URL	<b>/management/v3/assets/request</b>
Método	POST
Descripción	Devuelve todos los activos acordes a la consulta solicitada. Se permite la búsqueda por nombre y/o tipo de activo (corpus, modelos y recursos lingüísticos)

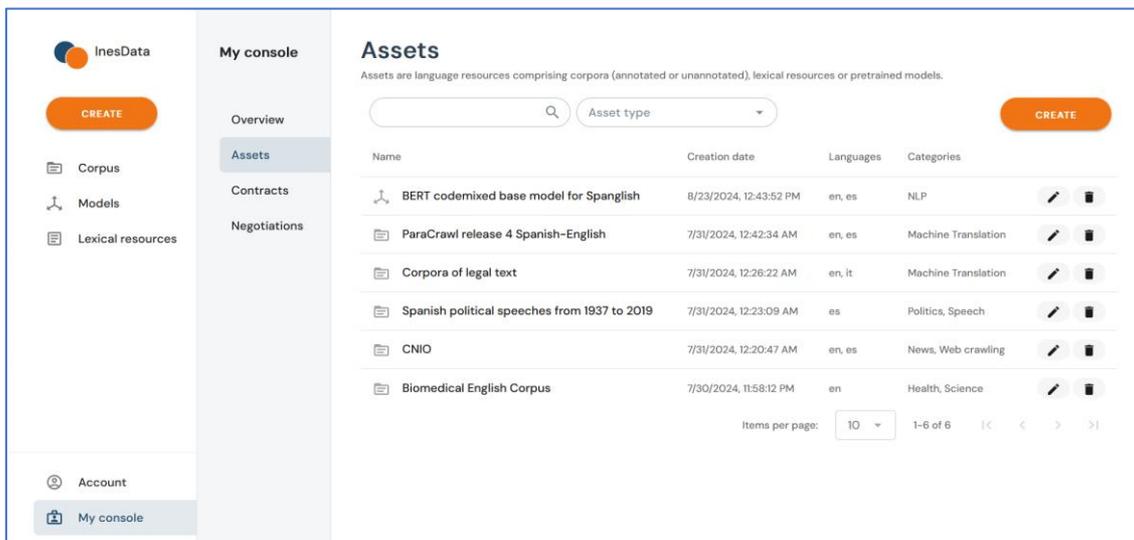


Figura 8. Listado de activos del conector del usuario

### 3.2.2 Detalle de un activo

API URL	<b>/management/v3/assets/{id}</b>
Método	GET
Descripción	Devuelve un activo correspondiente al ID especificado

### 3.2.3 Creación de un activo

API URL	<b>/management/v3/assets</b>
Método	POST
Descripción	Crea un nuevo activo junto con una dirección de datos

### 3.2.4 Actualización de un activo

API URL	<b>/management/v3/assets</b>
Método	PUT
Descripción	Actualiza un activo correspondiente al ID especificado

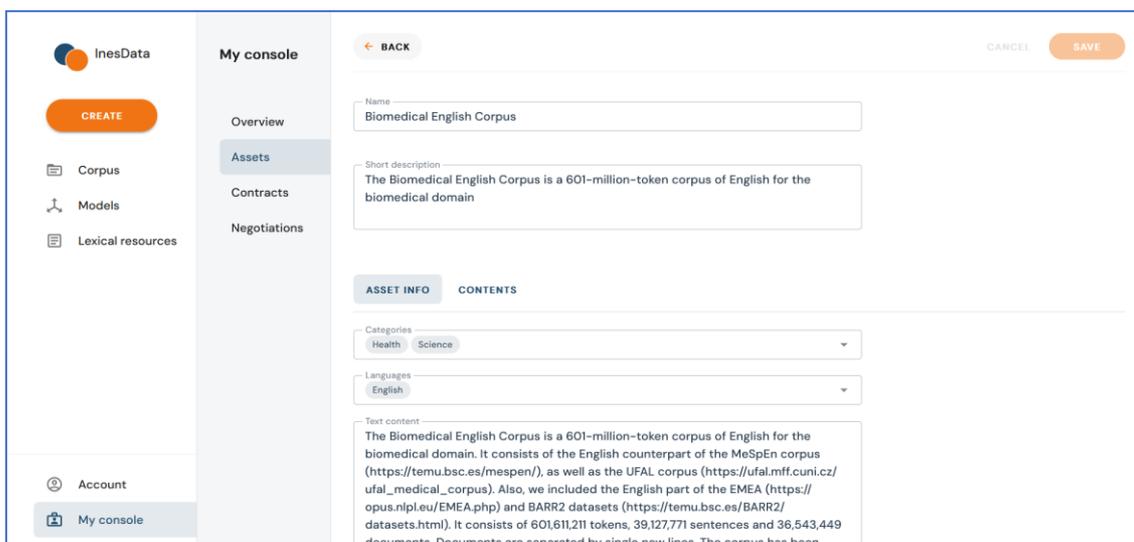


Figura 9. Visualización del detalle de un activo del conector del usuario

### 3.2.5 Borrado de un activo

API URL	<b>/management/v3/assets/{id}</b>
Método	DELETE
Descripción	Elimina un activo con el ID indicado. Solo es posible eliminar un activo si ese activo aún no está referenciado en un acuerdo de contrato, en cuyo caso se devuelve un error

## 3.3 Gestión de Pólizas

### 3.3.1 Listado de definiciones de pólizas

API URL	<b>/management/v2/policydefinitions/request</b>
Método	POST
Descripción	Devuelve todas las definiciones de pólizas acordes a la consulta solicitada

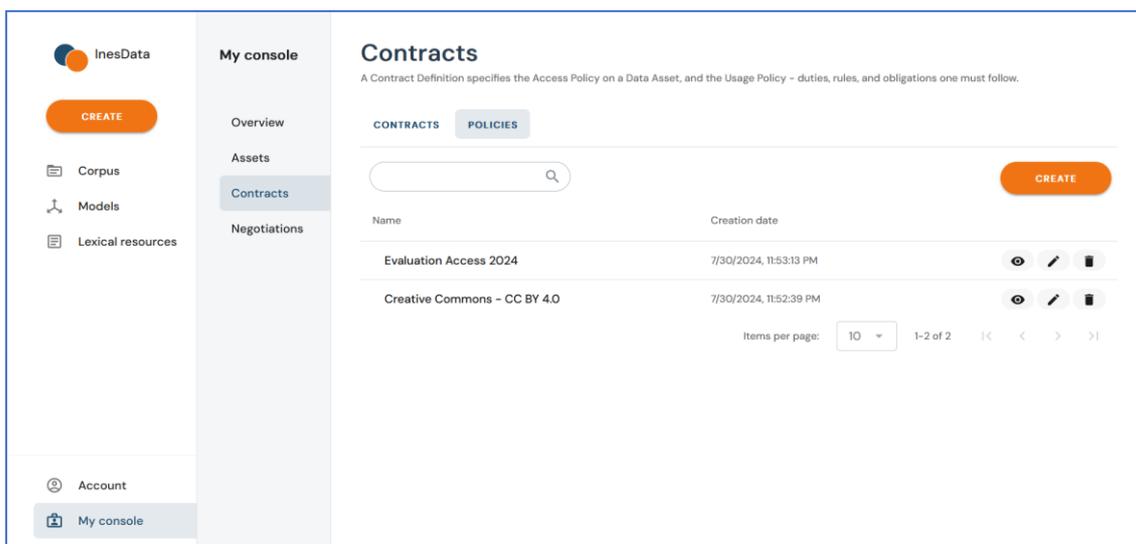


Figura 10. Listado de definiciones de pólizas del conector del usuario

### 3.3.2 Detalle de una definición de póliza

API URL	<b>/management/v2/policydefinitions/{id}</b>
Método	GET
Descripción	Devuelve una definición de póliza correspondiente al ID especificado

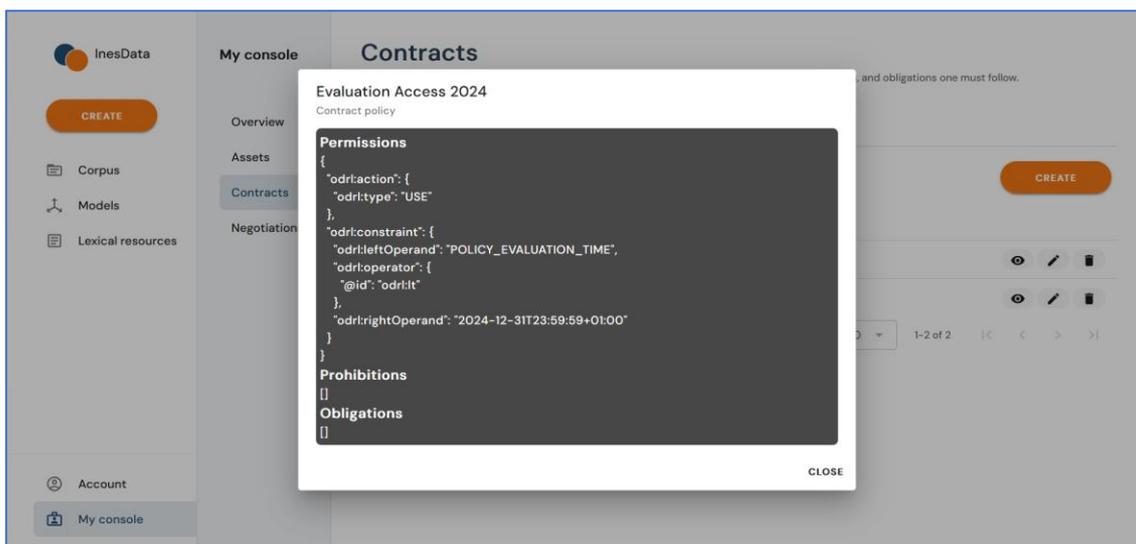


Figura 11. Visualización del detalle de una definición de póliza del conector del usuario

### 3.3.3 Creación de una definición de póliza

API URL	<b>/management/v2/policydefinitions</b>
---------	---

Método	POST
Descripción	Crea una nueva definición de póliza. Los permisos, prohibiciones y obligaciones de la póliza se han de definir mediante un JSON-LD válido

### 3.3.4 Actualización de una definición de póliza

API URL	<b>/management/v2/policydefinitions/{id}</b>
Método	PUT
Descripción	Actualiza una definición de póliza correspondiente al ID especificado. Los permisos, prohibiciones y obligaciones de la póliza se han de definir mediante un JSON-LD válido

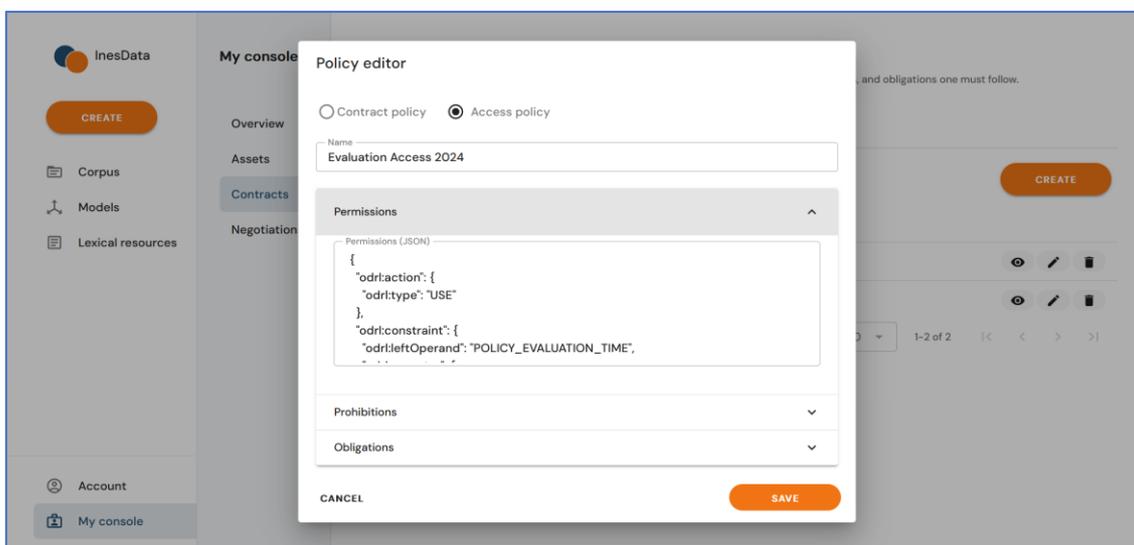


Figura 12. Pantalla de creación/edición de una definición de póliza

### 3.3.5 Borrado de una definición de póliza

API URL	<b>/management/v2/policydefinitions/{id}</b>
Método	DELETE
Descripción	Elimina una definición de póliza correspondiente al ID indicado. Solo es posible eliminar la definición de póliza si no está referenciada por un contrato, en cuyo caso se devuelve un error

### 3.4 Gestión de Contratos

#### 3.4.1 Listado de definiciones de contratos

API URL	<b>/management/v2/contractdefinitions/request</b>
Método	POST
Descripción	Devuelve todas las definiciones de contratos acordes a la consulta solicitada

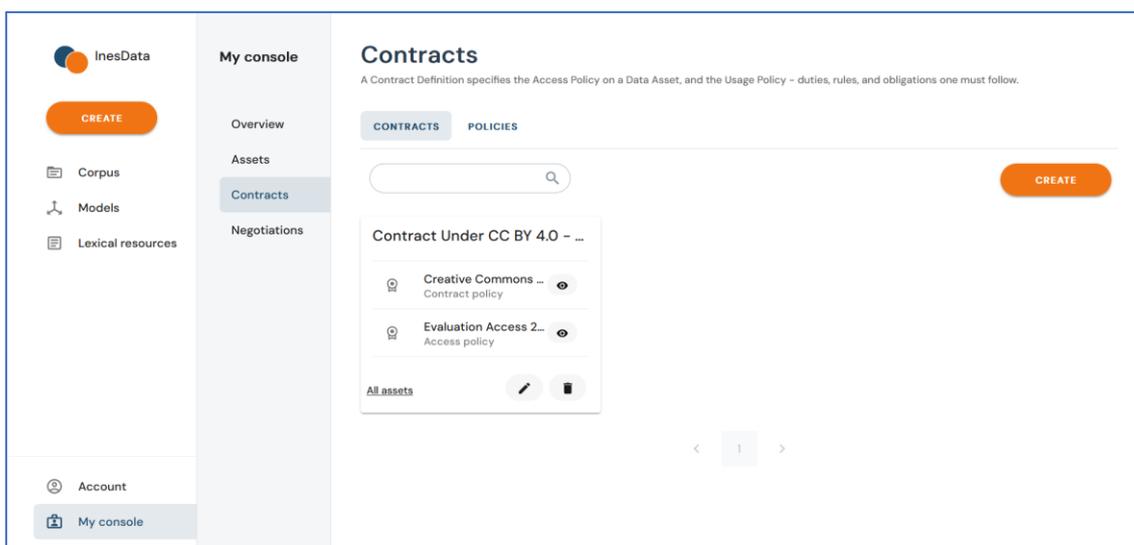


Figura 13. Listado de definiciones de contratos del conector del usuario

#### 3.4.2 Detalle de una definición de contrato

API URL	<b>/management/v2/contractdefinitions/{id}</b>
Método	GET
Descripción	Devuelve una definición de contrato correspondiente al ID especificado

#### 3.4.3 Creación de una definición de contrato

API URL	<b>/management/v2/contractdefinitions</b>
Método	POST
Descripción	Crea una nueva definición de contrato, el cual puede tener activos asociados. Si no se le asocia ningún activo, el contrato aplicará a todos los activos del conector por defecto

### 3.4.4 Actualización de una definición de contrato

API URL	<b>/management/v2/contractdefinitions/{id}</b>
Método	PUT
Descripción	Crea una nueva definición de contrato, el cual puede tener activos asociados. Si no se le asocia ningún activo, el contrato aplicará a todos los activos del conector por defecto

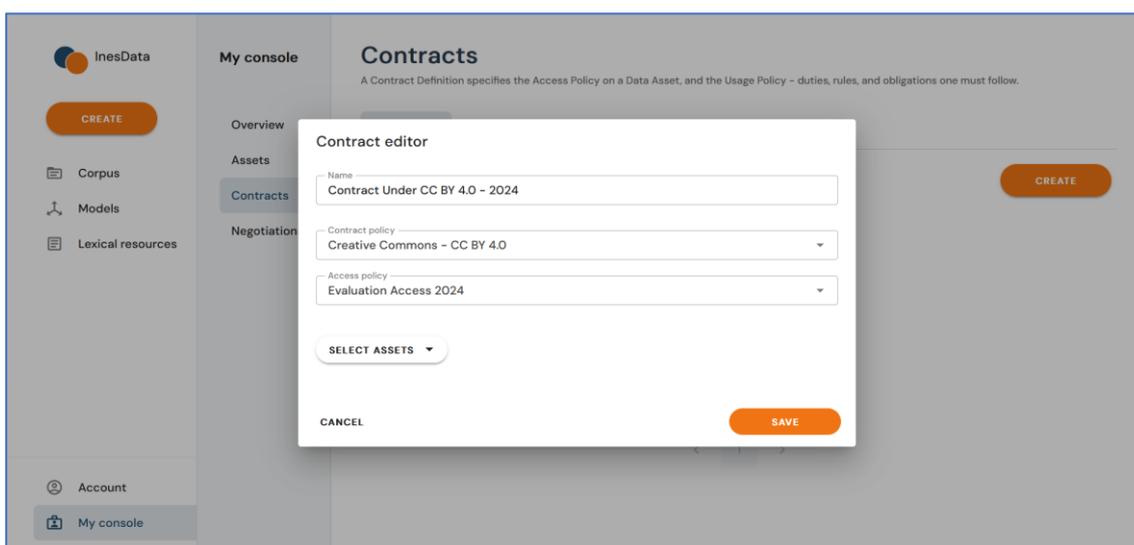


Figura 14. Pantalla de creación/edición de una definición de contrato

### 3.4.5 Borrado de una definición de contrato

API URL	<b>/management/v2/contractdefinitions/{id}</b>
Método	DELETE
Descripción	Elimina una definición de contrato correspondiente al ID indicado

## 3.5 Negociaciones

### 3.5.1 Listado de negociaciones de contratos

API URL	<b>/management/v2/contractnegotiations/request</b>
Método	POST
Descripción	Devuelve todas las negociaciones de contratos acordes a la consulta solicitada. En la interfaz web se muestran dos listados separados para las negociaciones enviadas y las recibidas

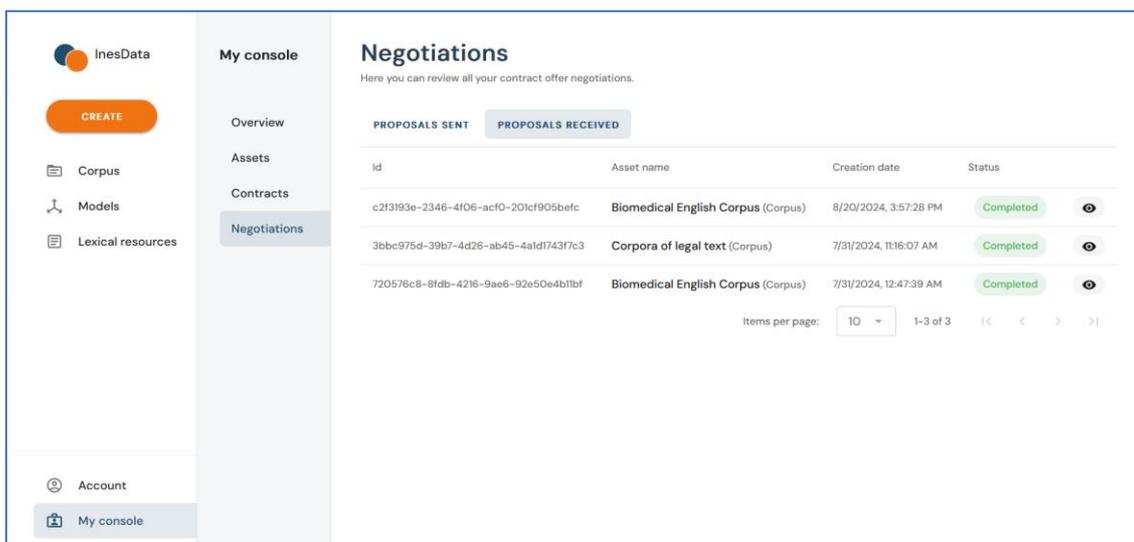


Figura 15. Listado de negociaciones de contrato recibidas

### 3.5.2 Detalle de una negociación de contrato

API URL	<b>/management/v2/contractnegotiations/{id}</b>
Método	GET
Descripción	Devuelve una negociación de contrato correspondiente al ID especificado

### 3.5.3 Detalle de un acuerdo de contrato

API URL	<b>/management/v2/contractagreements/{id}</b>
Método	GET
Descripción	Devuelve un acuerdo de contrato correspondiente al ID especificado. El detalle del acuerdo solamente se mostrará en caso de que se cumplan las condiciones y la negociación haya concluido con éxito

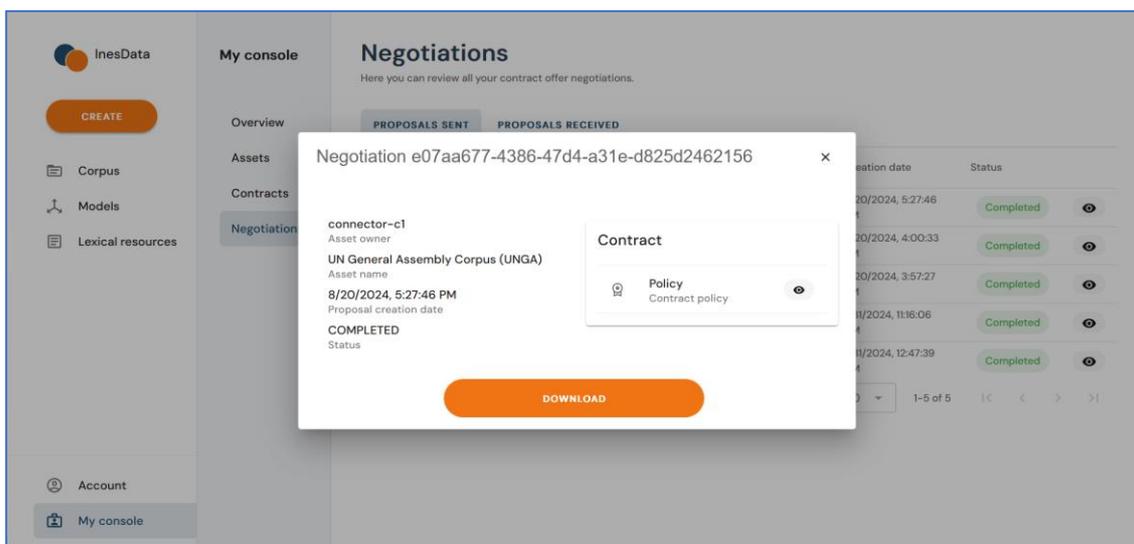


Figura 16. Detalle de una negociación de contrato completada

### 3.5.4 Creación de una negociación de contrato

API URL	<b>/management/v2/contractnegotiations</b>
Método	POST
Descripción	Inicia una negociación de contrato para un activo y una oferta de contrato determinada. La negociación se iniciará desde el detalle del activo en el catálogo, con la contraparte indicada (el conector propietario de ese activo)

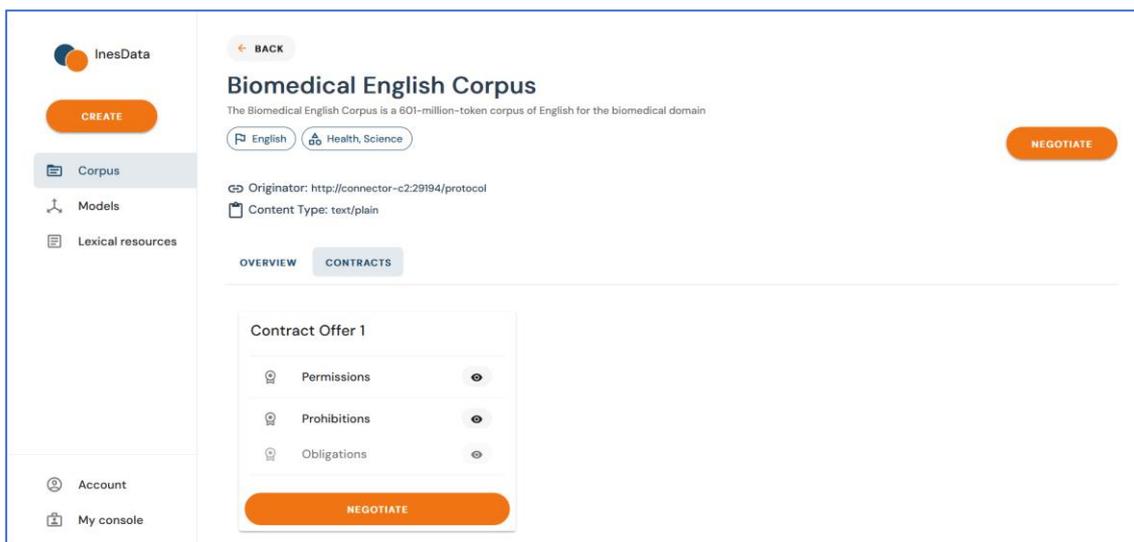


Figura 17. Visualización del detalle de un activo en el catálogo del cual se quiere iniciar una negociación

## 4 Herramientas y tecnologías usadas

Para diseñar la interfaz web del catálogo se utilizó Figma<sup>1</sup>. Figma es una herramienta de diseño de interfaz colaborativa que permite crear, compartir y colaborar en diseños en tiempo real. Permite a los diseñadores crear interfaces de usuario, prototipos y sistemas de diseño de manera eficiente.

Para la implementación se utilizaron las siguientes librerías y herramientas:

- Vue 3: Vue.js es un marco de JavaScript progresivo para crear interfaces de usuario. La versión 3 ofrece un rendimiento mejorado, tamaños de paquetes más pequeños y una integración mejorada de TypeScript, lo que la convierte en una opción ideal para desarrollar aplicaciones web dinámicas e interactivas.
- Vite: Vite es una herramienta de compilación moderna para Vue.js y otros marcos de JavaScript. Proporciona un inicio del servidor en frío increíblemente rápido, reemplazo instantáneo de módulos en caliente (HMR) y flujos de trabajo de desarrollo eficientes, lo que permite una iteración rápida durante el desarrollo.
- TypeScript: TypeScript es un superconjunto de JavaScript que agrega escritura estática y otras características al lenguaje. Mejora la calidad del código, proporciona un mejor soporte de herramientas y ayuda a detectar errores durante el desarrollo, lo que genera bases de código más sólidas y fáciles de mantener.
- Vuetify: Vuetify es una biblioteca de interfaz de usuario popular para Vue.js que proporciona un conjunto de componentes personalizables y prediseñados siguiendo las pautas de Material Design de Google. Simplifica el desarrollo de interfaces de usuario responsivas y estéticamente agradables, ofreciendo una amplia gama de componentes y utilidades.
- Cypress: Cypress es un marco de prueba de un extremo a otro para aplicaciones web. Permite a los desarrolladores escribir pruebas automatizadas que simulan las interacciones del usuario y verifican el comportamiento de sus aplicaciones web en diferentes navegadores y entornos, garantizando calidad y confiabilidad.

Esta pila de tecnología combina potentes herramientas y librerías para facilitar el diseño, desarrollo, prueba y mantenimiento del sitio web y, en última instancia, ofrece una aplicación web moderna, eficiente y fácil de usar.

## 5 Repositorio

El código fuente de la interfaz web del espacio de datos está publicado en el siguiente enlace de Github:

- <https://github.com/oeg-upm/inesdata-espacio-linguistico>

## 6 Conclusiones

Se ha liberado una nueva versión de la interfaz de usuario del espacio de datos lingüístico. La interfaz cumple con los requisitos establecidos para los espacios de datos y usa vocabularios estándar para la descripción de recursos de tecnologías del lenguaje. La interfaz de usuario es funcional y de un diseño que facilita su uso por los usuarios del espacio de datos.

En la siguiente entrega se incluirán el resto de las funcionalidades de la API del conector de INESData que no se han implementado en esta versión, como, por ejemplo, la transferencia y descarga de activos.

---

<sup>1</sup> <https://www.figma.com/>