

# Anjana Data – v23.1

Fecha de la release: 30/11/2023

## Artefactos:

- *admin v4.5*
- *customer-lib v4.5*
- *drittesta v4.5*
- *edusa v4.5*
- *hecate v4.5*
- *hermes v4.5*
- *horus v4.5*
- *inventory-interceptor-example v4.5*
- *kerno v4.5*
- *minerva v4.5*
- *portuno v4.5*
- *portal v4.5*
- *tot v4.5*
- *tot-plugin-lib v4.5*
- *tot-plugin-aws-s3 v4.5*
- *tot-plugin-azure-ad v4.5*
- *tot-plugin-ldap v4.5*
- *tot-plugin-jdbc v4.5*
- *tot-plugin-jdbc-denodo v4.5*
- *tot-plugin-jdbc-redshift v4.5*
- *tot-plugin-jdbc-sqlserver v4.5*
- *viator v4.5*
- *zeus v4.5*

**Compatibilidad única con v4.5.0 de los módulos de Anjana.**

## Scripts integrations:

- *zRecursos/v23.1/update\_aris.sql*

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	22/11/2023	Anjana Producto	Anjana Producto	Creación del documento. Compatibilidad con la v4.5 de todos los módulos de Anjana



# Actuaciones manuales requeridas

## ANTES DEL UPGRADE

- Haber ejecutado todas las RN de la versión 4.4.x previamente.
- Parar los microservicios backend de Anjana.
  - Durante el proceso de upgrade puede darse el caso de que convivan microservicios 4.4.x que todavía no se han actualizado y microservicios 4.5.0. Si no se realiza esta actuación de parar los microservicios, en caso de que algún usuario o batch inicie un proceso, podrían generarse datos corruptos.
  - Tras la parada de microservicios se recomienda realizar un backup completo antes del upgrade.
- Crear el schema **tot** en la base de datos para gestionar correctamente las migraciones existentes en ese servicio
- Hacer un backup de toda la configuración YAML de los micro servicios, será usado más adelante.
- Utilizar los YAML proporcionados por Anjana, que ahora solo contienen conexiones y credenciales, y rellenar con los valores correspondientes.
- Revisar los YAML de ejemplo, que viene informado con valores por defecto, y si se tenía alguna configuración específica añadirla al YAML mencionado anteriormente.
- El resto de propiedades YAML, serán revisadas en el apartado [DESPUÉS DEL UPGRADE](#)
- Se ha añadido que la configuración de servicios se pueda obtener de un servicio externo y así está configurado por defecto. Pero para Hermes es necesario ser incluida en el descriptor de servicio la siguiente propiedad -D de java  
(-Danjana.properties.url[0]=http://<portuno\_server>:8998/internal/v2/appconf?prefix=hermes). Ojo que esta propiedad va antes de la especificación de la aplicación jar.

```
ExecStart=/usr/lib/jvm/java-8-openjdk-amd64/bin/java -XX:+UseG1GC  
-XX:+UseStringDeduplication -Xmx4G  
-javaagent:/opt/common/xjar-agent-hibernate.jar  
-Danjana.properties.url[0]=http://portunoserver:8998/internal/v2/appconf?  
prefix=hermes -jar /opt/hermes/hermes.jar  
--spring.cloud.config.uri=http://edusaserver:8888  
--spring.profiles.active=default --spring.cloud.config.failFast=true
```

## DESPUÉS DEL UPGRADE Y ARRANQUE

- Revisar la configuración creada con valores por defecto en la tabla app\_configuration de Portuno y editarla si se tuviera algún valor diferente en los YAML backup. Para más detalle revisar el documento de 'Anjana Data - 23.1 - Configuración Funcional'. En caso de haber propiedades 'desaparecidas' es porque se han eliminado.
- Ejecutar el script de actualización de ARIs en base datos, se encuentra en el artefacto Integrations en la carpeta recursos de esta versión.
- Para solventar posibles problemas anteriores con los objetos indexados en Solr, es necesario llevar a cabo estos pasos en la actualización:
  1. Borrar colección Kerno de Solr con este comando (necesario token con permiso API ADMIN)

```
curl --location --request DELETE  
'http://<anjana_url>/gateway/api/admin/minerva/collection/kerno'  
--header 'Authorization: Bearer XXXXX'
```

2. Parar y arrancar de nuevo Minerva
  3. Indexar todo desde Portuno ( si se tiene la propiedad solr.indexing con valor por defecto o 'only\_index' o 'delete\_and\_index' no es necesario este subpaso)
- Desde Portuno:
    - Actualizar los ficheros de traducciones de Minio
    - Calcular el linaje completo
  - Las colecciones de Solr de auditoría y snapshots van a ser migradas a BD.
    - Se recomienda generar un backup de las colecciones de SolR de snapshot y audit\_log, además con ello se puede calcular el tamaño de disco que incrementará la bbdd por estas colecciones que servirá para interpretar si es necesario alguna actuación para base de datos.
    - Para ello, ejecutar el siguiente endpoint (necesario token con permiso API ADMIN)

```
curl --location --request 'POST'  
'http://<anjana_url>/gateway/api/admin/minerva/migrateAuditLogsAndSnapshots'  
s' --header 'Authorization: Bearer XXXXX'
```

- Modificación de iconos. Se han renombrado las carpetas de iconos alojadas en el CDN (normalmente MinIO). Si se tenían iconos de subtipos de objetos deberán de ser movidos a subtype\_icons.
  - Sólo si no se usa Ansible: subir el contenido del artefacto cdn y mover los iconos anteriormente nombrados.  
<https://releases.anjanadata.org/repository/releasesraw/com/anjana/cdn/4.5.0/cdn-4.5.0.tgz>
- Hay que tener en cuenta en sistemas de monitorización y entornos basados en contenedores que las nuevas versiones de los microservicios en su primera ejecución pueden tardar hasta 6 minutos en arrancar y contestar los Health Check dado que se ejecutarán las operaciones automáticas de actualización de base de datos. Se recomienda por tanto que al menos para este primer arranque se revise la configuración de los entornos basados en uso de contenedores subiendo el número de segundos de delay inicial. Los microservicios con este comportamiento son Portuno, Kernio, Hermes, Minerva y Zeus.

## Actuaciones manuales opcionales

- Después de ejecutar la migración de auditoría y snapshots, si se desea, se pueden eliminar las colecciones audit\_logs y snapshots de SolR, además de una colección residual mediante los siguientes comandos de consola:

```
/opt/solr/bin/solr delete -c audit_logs  
/opt/solr/bin/solr delete -c snapshots  
/opt/solr/bin/solr delete -c business_tags
```

Si SolR está instalado en otra ubicación, sustituya la ruta del comando por esa ruta.

- Durante el upgrade Anjana crea un usuario, una unidad organizativa, un rol y una asignación a esa OU y rol al usuario para que, en entornos sin configuración cargada éste pueda entrar en Portuno a configurar. Esta configuración es opcional y puede ser borrada con este comando:

```
delete from zeus.user_ou_role where user_id in (select user_id from
zeus.users where user_name = 'portuno.default');
delete from zeus."permission" where id_role in (select id_role from
zeus."role" where role_name = 'portunoDefault');
delete from zeus."role" where role_name = 'portunoDefault';
delete from zeus.organizational_unit where alias = 'PortunoDefault';
delete from zeus.users where user_name = 'portuno.default';
```

## Cambios estructurales

- Encriptación de todas las llamadas internas entre los microservicios
- Securización de endpoints internos mediante el uso de tokens de servicio
- Inclusión de sistema de gestión de cambios en base de datos para evitar actuaciones manuales en la misma

# Actualizaciones

## Tot y plugins

- Adaptación de Tot para implementar la detección de cambios en las estructuras de los sistemas con los que Anjana se integra. De esta forma, Tot queda preparado para cuando los plugins capturen los eventos y desencadenen acciones en Anjana de creación o actualización de forma desatendida.
- Al producirse la modificación de un objeto en Anjana se llama a Tot para que éste invoque al plugin correspondiente y se efectúe la acción implementada.
- Posibilidad de hacer descubrimiento de metadato de objetos de tipo relación.
- Añadido sistema de reintentos en la recepción de importaciones de metadato de los plugins.
- Registro de los plugins en Hecate a través de Tot.
- Tot permite registrar múltiples instancias de un mismo plugin.
- Control de excepciones incluido para que Anjana muestre correctamente los mensajes de error que llegan desde los plugins en la pantalla de resumen de Importación.

## Plantillas de objetos

- Nueva validación para los atributos Entidad y Lista de entidades que permite limitar los subtipos de entidad posibles con los que rellenarlos.
- Nuevo atributo `entity_container` para crear relaciones nativas entre entidades de Anjana pudiendo utilizar la validación anteriormente mencionada para filtrar por subtipo en caso de ser necesario
  - En el DSA, por medio de un atributo de este tipo, se pueden elegir las entidades que el DSA contendrá para que, al ser guardado, se creen relaciones internas entre el DSA y las entidades elegidas.
  - En una instancia se pueden añadir los datasets input y output con dos atributos de este tipo para gestionar las relaciones entre ellos automáticamente
  - En una solución se pueden añadir las instancias relacionadas con un atributo de este tipo
- Nuevos atributos para selección única o selección múltiple de taxonomías. Estos atributos presentan al usuario un árbol jerárquico en el que éste puede seleccionar cualquiera de los nodos, sea cual sea su nivel en el árbol.
- Nuevos atributos para listados de valores con icono o con icono y texto, tanto de selección única como múltiple.

## Configuración de entidades extremo en las relaciones

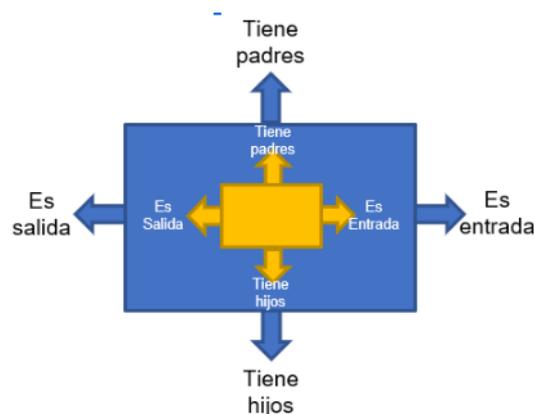
- Se incorpora la posibilidad de configurar los subtipos de entidades origen y destino de las relaciones. Esta configuración indica si cada extremo de una relación permite uno, varios o todos los subtipos de entidad por medio de validaciones para cada uno de ellos. Con esto, en la creación de relaciones manual o por excel y en el clonado se prohibirán relaciones que no cumplan con la restricción impuesta.

## DSAs flexibles

- La incorporación del nuevo tipo de atributo `entity_container` permite que los DSAs no solo incluyan datasets si no que puedan contener otras entidades no nativas y que éstas puedan ser gobernadas y, por tanto, se automatice la concesión de permisos a cualquiera de ellas.
- Por ello, se pueden añadir al carrito DSAs, datasets y entidades no nativas y quedar marcadas todas ellas como incluidas en el carrito, pendientes de adherencia o adherido en caso de haber finalizado el workflow de validación.
- Además, la herencia de atributos booleanos se puede aplicar a DSAs con respecto a cualquier otra entidad que lo componga.
- La desactivación de una entidad no nativa afecta de la misma forma que la expiración de un dataset, revocando los permisos a la estructura correspondiente.

## Linaje

- Se presenta un linaje exploratorio en 2D donde el usuario decide qué ramas del grafo explotar para ver su contenido expandiendo los nodos buscando relaciones en las que es padre o hijo, input o output siguiendo este esquema:



- Se incorporan capas de explotación configurables que permiten filtrar la información que se muestra en el grafo, tanto de entidades como de relaciones, facilitando su uso a usuarios de distintos ámbitos: de negocio, funcionales o técnicos.
- En estas capas es posible configurar relaciones de agregación para mostrar al usuario entidades que contienen otras entidades presentando la información de una forma más intuitiva.
- En el caso del linaje completo, el usuario elige el Elemento eje con el que comenzar la exploración y la capa de visualización a aplicar en el grafo pudiendo cambiarlas en cualquier momento de la navegación.

## Importación vía excel

- Se incluye un módulo de seguimiento con el que, tras una importación, es posible descargar el excel utilizado con información que indica, para cada registro, si se ha podido crear con éxito, si se ha creado pero no se ha podido lanzar su workflow o no se ha podido crear para que puedan ser corregidos y vuelvan a ser cargados en la plataforma.

- En la plantilla excel es posible añadir las relaciones nativas entre los objetos de Anjana para agilizar el catalogado. Así, el DSA incluye en el excel las entidades que éste contiene, las instancias sus datasets input y output y las soluciones sus instancias relacionadas.
- Se reestructura la información mostrada en el excel descargado desde Anjana con la plantilla. Con esto, se incluye el menú y la sección a la que pertenece cada atributo. Además, los atributos que forman parte de la PK de los objetos se marcan para indicar su obligatoriedad.
- Se genera al vuelto la información contenida en la pestaña de Read.me en función de los atributos de la plantilla del objeto.
- Se limita la descarga y subida de ficheros excel .xlsx.

## Workflows de validación

- En los workflows de modificación o versionado de objetos, se incluye la posibilidad de hacer la comparativa de los snapshots del objeto a validar y el original. Esta opción presenta al usuario, de forma visual e intuitiva, las diferencias entre ambos objetos en formato tabla y en formato json.
- Se incluye la opción de Envío masivo de workflows de creación o modificación. Para ello, el usuario elige un conjunto de objetos en el Portal, habiendo utilizado los filtros y el buscador, de un mismo subtipo en estado borrador y elige el rol con el que quiere lanzarlos a validar.
- Se incorpora la posibilidad de Validar masivamente una lista de workflows en los que el usuario tenga pendiente de respuesta el siguiente paso del workflow. La respuesta para todos esos workflows será la misma tanto la aprobación o rechazo como el motivo.
- Se añaden botones para la validación de workflows desde la pantalla con el listado completo sin necesidad de entrar al detalle.

## UX & UI

- Se actualiza la vista de un objeto con todo su contenido y sus diferentes opciones incluyendo filtros y ordenación en todas ellas:
  - Estructura.
    - Se ha modificado la forma en la que se crean los dataset\_fields pudiendo crear varios seguidos sin necesidad de informar todos sus atributos. La edición de ellos y la inclusión de atributos personalizados se hace solo desde su vista específica
  - Relaciones, que incluye:
    - Directas: relaciones no nativas de la entidad con otras entidades
    - Indirectas:
      - para un dataset, relaciones de sus dataset\_fields con otras entidades
      - para un dataset\_field, relaciones de su dataset con otras entidades
      - para un proceso, relaciones de sus instancias con otras entidades
    - De composición: relaciones nativas de la entidad creadas internamente debido al uso de atributos entity\_container
  - Intervinientes, que incluye:
    - Usuarios principales: creador y propietarios del objeto
    - Usuarios secundarios: usuarios asignados en algún atributo de la plantilla o atributo personalizado del objeto

- Adheridos: usuarios adheridos al objeto
  - Versiones. En esta pestaña se muestran todas las versiones activas (aprobadas y deprecadas) o no activas (importadas, borradores, pendientes, rechazadas y expiradas) del objeto permitiendo la navegación a cualquiera de ellas, la descarga del snapshot correspondiente o al workflow de validación en caso de la versión pendiente
  - Muestra de datos. Solo aparece para datasets aprobados con los flags `is_governed` y `sample_data` a true.
  - Se incluyen accesos directos a las versiones con estado borrador, pendiente o rechazado y a la versión actual del objeto desde la cabecera
- Se disponibiliza al usuario la ordenación alfabética de los valores de los atributos de tipo lista tanto en las plantillas de los objetos como en los filtros
- Se facilita la creación de entidades y relaciones consecutivas manteniendo los valores de la anterior.
- Se incluye la posibilidad de crear relaciones desde una entidad para considerarla origen o destino de ella.
- Se simplifica la selección de los proveedores de identidades en el login permitiendo definir uno por defecto que vendrá marcado de forma predeterminada.
- Se unifican los títulos de las pantallas de Anjana y se personaliza la pestaña del navegador correspondiente.
- Se permiten todas las acciones sobre un objeto desde el Portal sin necesidad de entrar a su pantalla.
- Nuevo botón para limpiar el carrito de la compra.
- Login automático de Anjana si el navegador ya tiene un token de usuario de Anjana.
- Link añadido en las pantallas desde las que se crean entidades y relaciones para poder acceder al nuevo objeto creado.

### Desadherencia de otro usuario

- Si un usuario es owner de un DSA puede elegir uno o más usuarios para desadherirlos del DSA automáticamente.

### Ciclo de vida de los objetos

- Al editar una entidad deprecada o expirada y generar, con esto, un borrador del objeto, éste no lleva la fecha de expiración del objeto original para obligar al usuario a informarla de nuevo.
- Actualización de la fecha de modificación del objeto cuando éste expira.

### Borrado de objetos

- Se disponibiliza el borrado de objetos por API o por la interfaz de Anjana en función de dos permisos:
  - el permiso `DELETE_MY_OBJ` de un subtipo de objeto permite al usuario eliminar objetos de ese subtipo creados por él mismo sea cual sea su estado

- DELETE\_ALL de un subtipo de objeto permite eliminar cualquier objeto de ese subtipo sea cual sea su estado y su usuario creador
- El borrado no es posible para entidades relacionadas con otras entidades y se avisará al usuario de ello. Para poder eliminar la entidad es necesario eliminar antes sus relaciones previamente.

## API admin

- Se ha creado un endpoint que permite el cambio de nombre de un objeto.
- Se ha actualizado el endpoint de creación y edición de objetos para controlar los estados permitidos y que se correspondan con el ciclo de vida del objeto
  - Creación y actualización de entidad o de relacion no nativa no permite los estados Expirado y Deprecado
  - Creación y actualización de entidades nativas no permite el estado Deshabilitado

## Configuración

- Añadida por defecto la carga de propiedades de configuración desde un microservicio que tendrán preferencia sobre otros ficheros de configuración.

## Seguridad y arquitectura

- Encriptación y securización de comunicaciones internas con nuevo algoritmo y ahora de forma configurable.
- Posibilidad de reinicio de microservicios a través de API.
- Posibilidad de configurar aislamiento de instancias de microservicio para no ser accesibles desde el punto de entrada de la API.
- Gestión de migración automatizada dejando de ser necesario actuación manual en la persistencia casi totalmente.
- Inclusión de proveedores cloud Vault para gestión de configuración. Están disponibles:
  - AWS Secret Manager
  - Azure Key
  - GCP Secret Manager
  - HashiCorp
- Activación de caché L2 de Hibernate para optimizar peticiones a base de datos. Es posible activarla o desactivarla a través de la siguiente configuración:

```
spring:
  jpa:
    properties:
      hibernate:
        cache:
          use_second_level_cache: false
          use_query_cache: false
```

## Licencias

- Nuevos tipos de licencia y nueva mecánica implementada:
  - Se han añadido nuevos estados, activados de forma manual:
    - TEMPORARY: licencia provisional, usada para entornos de pruebas
    - READ\_ONLY: licencia de sólo lectura
  - El estado EXPIRED de la licencia no permite el acceso a la aplicación: ni al Portal ni a Portuno.
  - Cuando la licencia deja de ser válida (debido a la fecha de expiración o a exceder el número de objetos gobernados) continúa 3 días más con estado VALID y después pasa a estado EXPIRING durante 4 días.

## Portuno

- Nuevo botón para validar la configuración de los atributos:
  - comprueba que los atributos de la PK de los objetos tienen las validaciones de NOT\_EDITABLE y REQUIRED
  - comprueba si los valores configurados para atributos de tipo taxonomía (de selección única o múltiple) forman árboles de valores sin ciclos y con nodos raíz
- Se incluye un nuevo usuario por defecto (portuno.default) en la base de datos para que pueda entrar a Portuno a configurar en un entorno vacío.
- Añadida la opción de filtro por campo null o distinto de null.
- Optimización en el proceso de indexación de todos los objetos y en la indexación individual de cada objeto al ser creado o editado.
- Distinción entre campos con valor null y cadena de texto vacía.

## ANEXOS

# Cambios en la API

A continuación se detallan los cambios en los endpoints, para más detalle consultar el manual de *API Interacciones básicas*.

### Endpoint modificados (su entrada o su salida o su método http)

- `/api/v1/auth/roles/byActionAndOrganizationalUnit` (obtener roles de usuario basado en permiso y unidad organizativa)
- `/api/v2/auth/permission/organizational-unit` (obtener unidades organizativas basado en permiso y usuario)
- `/api/v2/auth/users/organizational-unit` (obtener usuarios basado en permiso y unidades organizativa)
- `[POST] /api/v2/auth/permissions` → `[GET] /api/v2/auth/user/permissions` (obtener permisos de usuario)
- `/api/v2/auth/authorization` (comprobar si un usuario tiene un permiso en una unidad organizativa)
- `/api/v2/auth/subtype_ou` (obtener los subtipo y en qué unidades organizativas el usuario tiene un permiso en particular)
- `/entity/change-state/{idObject}/{state}`, `/ENTITY/change-state/{idObject}/{state}` (Cambiar el estado de una entidad). En endpoint anterior era `/change-state/{idObject}/{state}`
- `[GET] /api/v2/objectsubtypes` (Deprecado) → `[POST] /api/v2/objectsubtypes` (Ahora recibe una lista de `ObjectType` en vez de solo uno)
- `[POST] /api/v2/auth/organizational-unit/role` → `[GET] /api/v2/auth/organizational-unit/role` (Ahora ya no recibe el dto por parámetro)
- `[POST] /api/v1/language/obtain` → `[GET] /api/v1/language/obtain` (Ahora ya no recibe dto por parámetro)
- `[DELETE] /api/admin/minerva/collection/delete-field/{field}/{targetType}` → `[DELETE] /api/admin/minerva/collection/delete-field/{field}` (ya no recibe la colección por parámetro, la anterior se ha deprecado y dejado solo con KERNO)
- `[PUT] /api/admin/kerno/{workflowExecutionId}` → `[PUT] /api/admin/kerno/workflow/finish/{workflowExecutionId}`
- `[POST] /api/v1/dsa/adhere/{idObject}` → `[POST] /api/v1/dsa/adhere` (Se elimina el `idObject` como parámetro de la URL y se le envía en el body).
- `[POST] /api/admin/kerno/attribute/update/{objectType}/{objectSubType}` (cambiado el nombre del DTO de salida de `EditionMassiveResultDTO` a `MassiveOperationResult` y dentro del mismo la lista de errores pasa de ser `EditionMassiveFailDTO` a `ErrorDetailDTO`, con exactamente la misma información)
- `[GET] /api/v1/dataset/{idObject}/versions/{idObject}` (cambio de DTO) → `/api/v4/dataset/{idObject}/versions/{idObject}`
- `[GET] /api/v1/dsa/{idObject}/versions/{idObject}` (cambio de DTO) → `/api/v4/dsa/{idObject}/versions/{idObject}`
- `[GET] /api/v1/solution/{idObject}/versions/{idObject}` (cambio de DTO) → `/api/v4/solution/{idObject}/versions/{idObject}`
- `[GET] /api/v1/instance/{idObject}/versions/{idObject}` (cambio de DTO) → `/api/v4/instance/{idObject}/versions/{idObject}`

- [GET] /api/v1/process/{idObject}/versions/{idObject} (cambio de DTO)→  
/api/v4/process/{idObject}/versions/{idObject}
- [DELETE] /api/admin/kerano/delete/entity/{id} (añadidas restricciones de uso)
- [DELETE] /api/admin/kerano/delete/relationship/{id} (añadidas restricciones de uso)
- [POST] /api/admin/kerano/create-update/entity/dsa (cambio en DTO de entrada y salida en todas sus variantes)
  - /api/admin/kerano/create-update/entity/dsa/{idObject}
  - /api/admin/kerano/create-update/entity/DSA
  - /api/admin/kerano/create-update/entity/DSA/{idObject}
- [POST] /api/admin/kerano/create-update/entity/instance (cambio en DTO de entrada y salida en todas sus variantes)
  - /api/admin/kerano/create-update/entity/instance/{idObject}
  - /api/admin/kerano/create-update/entity/INSTANCE
  - /api/admin/kerano/create-update/entity/INSTANCE/{idObject}
- [POST] /api/admin/kerano/create-update/entity/solution (cambio en DTO de entrada y salida en todas sus variantes)
  - /api/admin/kerano/create-update/entity/solution/{idObject}
  - /api/admin/kerano/create-update/entity/SOLUTION
  - /api/admin/kerano/create-update/entity/SOLUTION/{idObject}

## Endpoints añadidos

- /api/admin/hermes/delete/workflow/{id\_workflow} (eliminar un workflow por API Admin)
- /relationship/change-state/{idObject}/{state},  
/RELATIONSHIP/change-state/{idObject}/{state} (Cambia el estado de una relación)
- [POST] /api/v4/audit/search (búsqueda con filtros de auditoría/histórico)
- [POST] /api/admin/minerva/migrateAuditLogsAndSnapshots (migración de auditoría y snapshots de SolR a BBDD)
- [DELETE] /api/admin/minerva/collection (borrado de la colección de Kerano en SolR)
- [POST] /create-update/entity/process (creación y edición de procesos, con varias variantes)
  - /create-update/entity/process/{idObject}
  - /create-update/entity/PROCESS
  - /create-update/entity/PROCESS/{idObject}
- /body-create-update/{objectSubType}/{idObject}
- [GET] /api/admin/kerano/relationship/{objectSubType}/{idObject} (Devuelve la relación en forma formulario dinámico)
- [PUT] /api/admin/kerano/name/{objectType} (Cambia el nombre de una entidad o una relación)
- [PUT] /api/admin/kerano/relationship/exchange (Intercambiar extremos relación)
- [PUT] /api/admin/kerano/relationship/change (Modificar extremos relación)
- [PUT] /api/admin/kerano/relationship/change-subtype/{id} (Cambiar subtipo relación y añadir atributos)
- [POST] /api/v2/common/checkSubmit (De una lista de objetos del mismo subtipo devuelve aquellos que están preparados para el submit)
- [POST] /api/v2/common/massiveSubmit/{objectSubType} (submit masivo de una lista de objetos)
- [DELETE] /api/v4/entity/delete/{objectSubType}/{id} (borrado de entidades)
- [DELETE] /api/v4/relationships/delete/{objectSubType}/{id} (borrado de relaciones)
- [GET] /api/v4/entity/{idEntity}/relationships/{typeTab} (obtener relaciones)
- [GET] /api/v4/excel/{type}/{subtype} (descarga de plantilla)
- [POST] /api/v4/excel/{type}/{subtype} (importación de plantilla)

- [GET] /api/v4/excel/results/{type}/{requestImportId} (descarga de resultado de la importación)
- [POST] /api/v4/lineage (obtiene el nodo eje del linaje)
- [POST] /api/v4/lineage/parents (obtiene los padres del nodo que recibe)
- [POST] /api/v4/lineage/inputs (obtiene las relaciones de entrada del nodo que recibe)
- [POST] /api/v4/lineage/children (obtiene los hijos del nodo que recibe)
- [POST] /api/v4/lineage/outputs (obtiene las relaciones de salida del nodo que recibe)
- [GET] /api/v4/lineage/layers (obtiene todas las capas configuradas para el linaje)
- [GET] /api/v4/lineage/layers/{layerId} (obtiene la capa con el id recibido)

### Endpoints deprecados

- [POST] /api/v1/auth/roles/byActionAndOrganizationalUnit
- [POST] /api/v2/entity/submit/{objectSubType}/{idObject}
- [POST] /api/v2/relationship/submit/{objectSubType}/{idObject}
- [GET] /api/v1/dataset/{idObject}/versions/{idObject} (cambio de DTO)
- [GET] /api/v1/dsa/{idObject}/versions/{idObject} (cambio de DTO)
- [GET] /api/v1/solution/{idObject}/versions/{idObject} (cambio de DTO)
- [GET] /api/v1/instance/{idObject}/versions/{idObject} (cambio de DTO)
- [GET] /api/v1/process/{idObject}/versions/{idObject} (cambio de DTO)
- [GET] /api/v2/entity/relationships/{objectSubType}/{idEntity} (obtener relaciones)
- [GET] /api/v2/excel/{type}/{subtype} (descarga de plantilla)
- [POST] /api/v2/excel/{type}/{subtype} (importación de plantilla)
- [GET] /api/v2/graphics/getAll (obtiene el linaje entero)
- [GET] /api/v2/graphics/listGraphics/{objectType}/{idObject} (obtiene el linaje por objeto)
- [GET] /api/v2/graphics/myLineage (obtiene el linaje de usuario)

### Endpoints eliminados

- [POST] /api/admin/minerva/migrateSnapshotsCollection