

# Anjana Data – v4.4

Release date: 28 oct 2022

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Anjana Producto	Anjana Producto	Creación del documento

## Actuaciones manuales requeridas

- Durante la intervención para realizar el upgrade es necesario parar los microservicios backend de Anjana.
  - Durante el proceso de upgrade puede darse el caso que convivan microservicios 4.3 que todavía no se han actualizado y microservicios 4.4. Si no se realiza esta actuación de parar los microservicios, en el caso de que algún usuario o batch inicie un proceso, podrían generarse datos corruptos.
- Ejecutar el script de upgrade de BD y Solr (artifactory, releasesraw tar de integrations)
- Eliminar la carpeta /cdn/images/profile y su contenido de Minio
- Obtener los iconos del artefacto "cdn-4.4.0.tar" disponible en el repositorio de Anjana releasesraw e Incluir los iconos plus-calendar.svg y check-calendar.svg en cdn/images/icons/\*
  - [Link repositorio](#)
- En el proyecto Drittesta añadir la propiedad **anjana.privateKey** que representa la clave privada del cliente. Anjana proporcionará esta clave o bien el cliente puede generarla (ver [Actuaciones manuales opcionales](#)).
- Para lanzar los servicios contra Edusa hay que hacer la siguiente modificación en el comando de arranque de todos los microservicios **salvo en Hermes**; normalmente ubicados en /etc/systemd/system/.:
  - antes: --spring.cloud.config.uri=http://<edusa\_server>:<port>
  - ahora: --spring.config.import=configserver:http://<edusa\_server>:<port>
  - Ejecutar el siguiente comando tras los cambios:

```
systemctl daemon-reload
```

- Las propiedades YAML spring.minio del módulo Portuno ya no son necesarias, se usan las configuradas en la tabla app\_configuration de base de datos.
- Es necesario sustituir las variables de Zookeeper para el control de los batches por la nueva configuración de base de datos en Kernu, Zeus y Drittesta:
  - antes:

```
anjana:
  scheduling:
    zookeeper: <Borrar>
    <Borrar cualquier propiedad aquí contenida>
```

- ahora:

```
anjana:
  scheduling:
    datasource:
      url: jdbc:postgresql://<hostnameDB>:<DBport>/anjana?currentSchema=portuno
      username: <usuario de conexión a base de datos>
      password: <contraseña de conexión a base de datos>
```

## Actuaciones manuales opcionales

- El NIST recomienda usar claves de como mínimo 2048 bits, que son las que proporciona Anjana. Aún así, si el cliente lo desea, podrá generar un nuevo par de claves RSA para incrementar la seguridad de la encriptación hasta 4096 bits creando las claves de forma manual; de este modo el cliente será el creador y validador de la seguridad y unicidad de las claves. Una vez generadas, la pública la deberá proporcionar a Anjana para configurar su instalación. La privada se establecerá en el proyecto de **Dritttesta** con la nueva propiedad del yaml **anjana.privateKey** teniendo en cuenta que:
  - Representa la clave privada del cliente para la comunicación entre los servicios de licencia.
  - Será una propiedad obligatoria.
  - Podrá ser generada por el cliente de la siguiente manera:
    - Con el endpoint habilitado en **Dritttesta** para ello (clave de 2048 bits):

```
curl --location --request GET 'http://{{host}}:{{port}}/api/license/keys'
```

Donde `{{host}}` es la dirección de la máquina y `{{port}}` el puerto de **Dritttesta**. La respuesta es un json con el siguiente formato:

```
{
  "publicKey": clave pública,
  "privateKey": clave privada
}
```

- O bien,

En una consola UNIX ejecutar las instrucciones:

- **ssh-keygen -m PKCS8 -t rsa -b 4096**
- **openssl rsa -in nombreClave -pubout**

La primera genera las claves en un directorio local. Descripción de los parámetros:

- **-m** indica el tipo de algoritmo que es obligatorio que sea PKCS8
- **-t** indica el tipo de clave que es RSA obligatoriamente.
- **-b** indica el tamaño de la clave. El cliente puede configurar el tamaño que desee. Los tamaños más comunes son 1024, 2048, 3072 o 4096 bits, siendo 4096 el tamaño máximo.

La segunda instrucción devuelve la clave pública en el formato correcto para usarse en Anjana. El parámetro “nombreClave” será el nombre de la clave que hemos dado en el paso anterior.

- En el proyecto **Dritttesta** sustituir la propiedad **anjana.anjanaPublicKey** por la nueva clave pública proporcionada por Anjana.
- Una vez modificada la configuración de **Dritttesta**, llamar al siguiente endpoint para renovar el código de licencia en la tabla **zeus.license**:
  - `curl --location --request PUT 'http://<host dritttesta>:<port dritttesta>/api/license/validate'`
- Si se quieren configurar claves (o propiedades de cualquier tipo) de manera que no estén en texto plano si no encriptadas, se puede hacer colocando la propiedad encriptada en Base64 entre comillas simples y concatenando delante {cipher}.

Ejemplo en el usuario y la contraseña para la conexión a base de datos de Kerno:

```
spring:
  datasource:
    username: '{cipher}$2a$10$w6y3docQwV08hgF8n9fL6uwwp808aSK8NEMapAWTU7h/13/zj1Kj.'
    password: '{cipher}$2a$10$w6y3docQwV08hgF8n9fL6uwwp808aSK8NEMapAWTU7h/13/zj1Kj.'
```

- Si al borrar una colección en Minerva y arrancar Solr se observa que existe un schema asociado a la colección que se ha recreado es debido a que Zookeeper guarda la configuración de esas colecciones y no se ha borrado correctamente. Se podría conseguir mediante la siguiente url

```
http://<SOLR_URL>:<SOLR_PORT>/solr/admin/configs?action=DELETE&omitHeader=true&name=<COLLECTION_NAME>.AUTOCREATED
```

## Cambios estructurales

- Ampliación de las columnas `ari`, `id_source` y `id_destination` de la tabla `anjana.relationship`
- Ampliación de la columna `ari` de la tabla `anjana.entity`
- Ampliación de la columna `value` de la tabla `portuno.translations`
- Cambio a no nullables las columnas `id_section` y `id_attribute_definition` de la tabla `anjana.template_attribute`
- Cambio a nullable la columna `organizational_unit` de la tabla `anjana.imported_metadata`
- Añadida restricción de unicidad en `anjana.attribute_value`
- Cambio de la PK en `anjana.edge`

## Cambios en la API

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

- Cambio en la cabecera de los endpoints para usar x-language en vez de Accept-language
  
- Endpoint modificados (su entrada o su salida o su método http)
  - /api/v1/process/versions/{idObject} (Obtener versiones de un proceso)
  - /api/v1/dataset/versions/{idObject} (Obtener versiones de un dataset)
  - /api/v1/solution/versions/{idObject} (Obtener versiones de una solución)
  - /api/v1/instance/versions/{idObject} (Obtener versiones de una instancia)
  - /api/v1/dsa/versions/{idObject} (Obtener versiones de un dsa)
  - /api/v2/entity/create/{objectSubType} (Crear una entidad)
  - /api/v2/entity/{objectSubType}/{idEntity} (Obtener una entidad)
  - /api/v1/adherence/combinations (Obtener combinaciones para adherencia)
  - /api/v2/entity/dynamic-catalog/{objectSubType} (Obtener el formulario dinamico de un subtipo)
  - /api/v2/entity/dynamic-catalog/{objectSubTypeParam}/{idObject} (Obtener el formulario dinamico de un objeto con sus valores)
  - /api/v1/solution/instances/related/{idObject} (Obtener las instancias relacionadas de una solucion)
  - /api/v1/dsa/disadhere/{idObject} (Desadherir de un dsa)
  - /api/admin/kerno/disadhere/{idObject} (Desadherir de un dsa)
  - /api/v1/task/validate (Validación de workflow)
  - /api/v2/auth/login/ldap → /public/v4/auth/login/ldap (Login en ldap)
  - /api/v2/auth/login/local → /public/v4/auth/login/local (Login por base de datos)
  - /api/v2/auth/providers (Recuperar los providers)
  - /api/v2/auth/login/oidc/password/{provider} →  
/public/v4/auth/login/oidc/password/{provider} (Login en un proveedor externo)
  - /api/admin/zeus/list/{objectSubtype} (Obtener lista de OUs donde se puede crear/editar)
  - /api/v2/workflow (Obtener lista de workflow filtrada)
  - /api/v1/workflow/execution/{id} (Obtener el detalle de un workflow)
  - /api/admin/kerno/save/entity/{objectSubType} →  
/api/admin/kerno/create-update/entity/{objectSubType}
  - /api/admin/kerno/save/entity/{objectSubType}/{idObject} →  
/api/admin/kerno/create-update/entity/{objectSubType}/{idObject}
  - /api/admin/kerno/create-update/dataset →  
/api/admin/kerno/create-update/entity/dataset
  - /api/admin/kerno/create-update/dataset/{idObject} →  
/api/admin/kerno/create-update/entity/dataset/{idObject}
  - /api/admin/kerno/create-update/dsa →  
/api/admin/kerno/create-update/entity/dsa
  - /api/admin/kerno/create-update/dsa/{idObject} →  
/api/admin/kerno/create-update/entity/dsa/{idObject}
  - /api/admin/kerno/create-update/solution/{idObject} →  
/api/admin/kerno/create-update/entity/solution/{idObject}
  - /api/admin/kerno/create-update/instance/{idObject} →  
/api/admin/kerno/create-update/entity/instance/{idObject}
  - /api/admin/kerno/create-update/relationships/{objectSubType} (Crear relaciones)
  - /api/admin/kerno/create-update/relationships/{objectSubType}/{idObject} (Modificar relaciones)

- /api/v2/notification/create (Envío de notificación)
- /api/admin/hermes/{module}/execution/{id} → /api/admin/hermes/execution/{id} (Obtener los detalles de un workflow execution)
- /api/admin/minerva/{target}/index/{object\_type}/{object\_subType} → /api/admin/minerva/index (Indexar una entidad)
- Endpoints añadidos:
  - /api/admin/kerano/{workflowExecutionId} (Finalización de un workflow)
  - /api/admin/kerano/attribute/update/{objectType}/{objectSubType} (Actualizar una lista de objetos)
  - /api/admin/kerano/lineage/load (Generación completa del linaje)
  - /api/admin/kerano/lineage/entity/load/{id} (Generación de linaje para una entidad)
  - /api/admin/kerano/lineage/relationship/load/{id} (Generación de linaje para una relación)
- Endpoints eliminados:
  - /api/admin/kerano/save/relationship/{objectSubType} (Creación de relación)
  - /api/admin/kerano/save/relationship/{objectSubType}/{idObject} (Edición de relación)
  - /api/admin/kerano/batch/lineage/loadAll (Generación completa del linaje)

## Actualizaciones

### Kerno

- Feature
  - Inclusión del atributo physicalName, opcional en la plantilla de los DSAs. En caso de informarlo en un DSA, la concesión de permisos al aprobar la adherencia de un usuario será contra el grupo correspondiente a ese mismo nombre físico. Si este atributo no se informa, se usará el name seguido de la versión como hasta ahora se hacía.
  - Revisión de la deprecación y la expiración de las entidades nativas
  - Importación por excel
    - Control de errores para mostrar en la pantalla de detalle el motivo (por ejemplo: cuando el workflow que debe lanzarse para validación no existe)
    - Mejora de rendimiento en el procesado de la importación
  - API
    - Nuevo endpoint de API administrativa que permite la finalización de un workflow en caso de error en la validación normal
    - Revisión del endpoint para recuperar una entidad para completar con todo su metadato
    - Modificación del endpoint de edición masiva para que admita una lista de objetos
    - Nuevo endpoint de API administrativa que permite la edición de una lista de objetos

- Homogeneización de las API de creación y edición de entidades y relaciones (API administrativa).
- Envío de notificaciones
  - Aviso al usuario afectado de una desadherencia con API administrativa
  - Mejora en el envío del aviso anticipando la expiración de las entidades
- Listado de versiones de un objeto ordenada
- Eliminación de Zookeeper como gestor de bloqueos para batches
- Inclusión de un flag para deshabilitar las validacion NOT\_EDITABLE (api admin en las llamadas create-update de entidades y relaciones)
- Nombre del snapshot descargado modificado. El patrón empleado es: <object\_subtype>\_<name>\_<id>\_<timestamp> (por ej. DATASET\_clientes\_10616\_20220903103638)
- Añadida fecha de descarga en el nombre de la plantilla generada por Anjana para la importación de metadatos por excel
- Cálculo del linaje completo modificado, por lo que no será necesario programar el batch. El linaje ahora incluirá los objetos de Anjana sin necesidad de que se recalculen el grafo completo. No obstante, el batch para el recálculo sigue disponible (y ejecutable desde Portuno)
- PKs no editables por edición masiva para evitar duplicados de objetos
- Llamada a Tot al expirar un dataset sin importar si éste está o no incluido en un DSA y hay adherencias a él

## Portal

- Feature
  - Nueva funcionalidad para el guardado de filtros personalizables. El usuario puede guardar una búsqueda realizada sobre el portal de Anjana para volver a reutilizarla o modificarla posteriormente. Esta búsqueda comprende los filtros, las opciones de Ver todas las versiones y Ver todas las relaciones y el cuadro del buscador de la cabecera de la aplicación.
  - Visualización de los valores de los atributos de los objetos que van a ser editados masivamente para conocer los cambios que se van a llevar a cabo sobre ellos
  - Permiso LINEAGE\_ALL aplicado para permitir la visualización del linaje por objeto
  - Mejora del buscador en el árbol de estructuras disponibles para la extracción de metadatos
  - Añadida la posibilidad de tener un css personalizado desde Minio
  - Añadida configuración para limitar la visualización del grafo de linaje en función del número de nodos a mostrar. Cuando el número de nodos supera el límite configurado, no se muestra el linaje para evitar que el usuario vea un grafo difícil de entender. Esta variable se llama MAX\_GRAPH\_NODE y se puede editar desde portuno para establecer el límite, por defecto es 2000.
- Bugfix
  - Depuración de los filtros en pantallas de histórico y auditoría
  - Arreglo para el solape de información con textos largos en algunas pantallas
  - Revisión de los filtros de tipo input\_text para mantener el texto escrito una vez usado en una búsqueda

- Depuración de dependencias entre atributos de selección única y múltiple

## Hermes

- Feature
  - API
    - Endpoint de validación de workflow modificado para que acepte una lista de workflows. Todos ellos se validarán con la misma respuesta y mismo comentario
    - Nuevo endpoint de API administrativa que permite validar una lista de workflows a pesar de no ser el usuario que lo lanza un usuario validador. Todos los workflows se validarán con la misma respuesta y mismo comentario
  - Revisión del envío de notificaciones a roles y usuarios
- Bugfix
  - Solventado error en el receptor del envío de notificaciones por mail
- Deprecado:
  - En la definición de un workflow, configurar un evento de notificación previo a un evento de usuario para avisar de una acción necesaria por parte de un usuario para continuar el workflow. En v4.3 se simplificó y desde entonces se puede configurar la notificación en el mismo evento de usuario. (Consultar la Guía de configuración de workflows para más detalle)

## Minerva

- Feature
  - API
    - Nuevo endpoint para indexar auditoría en la colección de Solr de Anjana
  - Se ha añadido la posibilidad de establecer las urls de solr de diferentes maneras:

```
solr:  
  url: http://indexservice:8983/solr,....
```

ó

```
solr:  
  urls:  
    - http://indexservice:8983/solr  
    - ...
```

- Bugfix
  - Depuración de la indexación completa para evitar que un error en un objeto bloquee el resto de indexaciones

## Plugins Tot

- Feature
  - Cifrado en la comunicación entre Tot y los plugins
  - Unificada la gestión de rutas y ficheros en plugins dedicados a ficheros
  - Uso del physicalName de los DSAs para la generación de grupos y concesión de permisos (leer actualización correspondiente en Kernio)
  - Tratamiento centralizado de los distintos tipos de ficheros para los plugins de Azure Storage, S3 y GCP Storage: Hadoop, Delta Lake...
  - Denodo: nuevo batch de actualización de tags de los dataset\_fields en Denodo
  - Powerbi: cuando un usuario se incorpora al sistema añadiendo un registro a la tabla de Users, se añade automáticamente la adherencia de ese usuario al DSA por defecto que bloquea los accesos de éste a cualquier reporte de PBI

## Portuno

- Feature
  - Mejoras en la edición y guardado de workflows
  - Mejoras en la edición y guardado de registros de las tablas de configuración
  - Scroll horizontal incluido para permitir leer completamente los valores de tablas con muchas columnas
  - Valores predeterminados incluidos en el campo module de attribute\_definition y object\_type en object\_sub\_type
  - Edición no permitida en la tabla user\_ou\_role debido a que sus tres campos componen la PK del registro. Para editar un registro será necesario eliminarlo y crear uno nuevo.
  - Nuevo botón para el lanzamiento del cálculo del linaje completo
  - Nueva propiedad en la tabla appConfigurations para limitar el máximo de nodos a mostrar en los linajes: MAX\_GRAPH\_NODE por defecto es 2000.

## Drittista

- Feature
  - Eliminación de Zookeeper como gestor de bloqueos para batches

## Zeus

- Feature
  - Eliminación de Zookeeper como gestor de bloqueos para batches
  - Añadida la posibilidad de que los nombres de usuario contengan '\_'
  - Añadida la posibilidad de que las unidades organizativas contengan ',' en sus alias



## Swagger

- Bugfix
  - Para Hermes, Kerno, Minerva, Portuno y Zeus se puede configurar una lista de url que puede utilizar Swagger para hacer llamadas REST a la API en su yml. En caso de no necesitar Swagger, es una configuración opcional y, por tanto, se puede obviar. Estas urls deberán apuntar al proxy de viator para el correcto funcionamiento de Swagger, si no se incluye, swagger genera una url válida por sí mismo.

```
swagger:  
  server:  
    url:  
      - https://viatorserver/gateway  
      - https://<anjana_host>/gateway  
      - https://hostalias/gateway
```