



# Guía de configuración funcional

<b>Control de versiones</b>	<b>5</b>
<b>Introducción</b>	<b>5</b>
Conceptos básicos de Anjana Data	6
Visiones para la configuración	6
<b>Configuración en BD</b>	<b>11</b>
Esquema Anjana de BD	11
1. Configurar subtipos de objetos: tipos de entidades y relaciones	12
Estructura de la tabla	14
Visión de Administrador	14
Visión de Desarrollador	16
2. Configurar el Menú	17
Estructura de la tabla	17
Visión de Administrador	18
Visión de Desarrollador	19
3. Configurar las secciones (sections)	20
Estructura de la tabla	20
Visión de Administrador	20
Visión de Desarrollador	22
4. Definir y configurar los atributos de metadatos	22
Estructura de la tabla	24
Visión de Administrador	26
Visión de Desarrollador	28
Atributos de clave primaria	29
Atributos Obligatorios	29
5. Definir y configurar los posibles valores de los atributos	33
Estructura de la tabla	33
Visión de Administrador	34
Visión de Desarrollador	35
6. Definir qué atributos aparecerán en cada una de las secciones de los formularios según el tipo de objeto	35
Estructura de la tabla	36
Visión de Administrador	36
Visión de Desarrollador	38
7. Definir validaciones para los valores de entrada de cada atributo	38
Estructura de la tabla	42
Visión de Administrador	43
Visión de Desarrollador	44
8. Definir relaciones entre valores de atributos para pre-filtrar los reference metadata o definir las taxonomías	45

Estructura de la tabla	46
Visión de Administrador	46
Visión de Desarrollador	48
9. Reglas de Versionado	48
Estructura de la tabla	48
Visión de Administrador	49
Visión de Desarrollador	50
10. Reglas de lanzamiento de workflow en ediciones y versionado.	50
Estructura de la tabla	50
Visión de Administrador	51
Visión de Desarrollador	52
11. Agregaciones para el linaje	52
Estructura de la tabla	53
Visión de Administrador	53
Visión de Desarrollador	54
12. Capas del linaje	55
Estructura de la tabla	55
Visión de Administrador	55
Visión de Desarrollador	56
13. Subtipos de las capas del linaje	56
Estructura de la tabla	57
Visión de Administrador	57
Visión de Desarrollador	59
14. Agregaciones en las capas del linaje	59
Estructura de la tabla	60
Visión de Administrador	60
Visión de Desarrollador	61
Esquema Hermes de BD	61
1. Notificaciones del sistema	62
Estructura de la tabla	62
Visión de Administrador	64
Visión de Desarrollador	65
Esquema Minerva de BD	66
1. Filtros del sistema	66
Estructura de la tabla	68
Visión de Administrador	69
Visión de Desarrollador	70
Esquema Portuno de BD	71
1. Variables del sistema	71
Estructura de la tabla	71
Visión de Administrador	77
Visión de Desarrollador	79

2. Traducciones	79
Estructura de la tabla	79
Visión de Administrador	80
Visión del Desarrollador	81
Traducciones internas de Anjana	82
Esquema Zeus de BD	84
1. Definir la estructura organizativa y las unidades organizacionales	85
Estructura de la tabla	86
Visión de Administrador	87
Visión de Desarrollador	88
2. Definir los roles del modelo de gobierno	89
Estructura de la tabla	90
Visión de Administrador	90
Visión de Desarrollador	92
3. Definir los permisos asociados a cada rol	93
Estructura de la tabla	94
Visión de Administrador	95
Visión de Desarrollador	96
4. Idiomas de la aplicación	97
Estructura de la tabla	97
Visión de Administrador	97
Visión de Desarrollador	100
5. Autenticación y autorización por base de datos	100
a. Usuarios	100
Estructura de la tabla	100
Visión de Administrador	101
Visión de Desarrollador	102
b. Rol y OU asignados a los usuarios	103
Estructura de la tabla	103
Visión de Administrador	103
Visión de Desarrollador	104
<b>Configuración de propiedades</b>	<b>105</b>
Todos los módulos core	105
Tot Plugin	105
<b>Ejemplos de configuración de proveedores de identidades</b>	<b>105</b>
Configuración de base de datos	105
Configuración de LDAP	105
Proveedores externos	109
<b>Configuración de Batches</b>	<b>110</b>
1. Comprobación de roles transversales (cross)	110
2. Licencia a punto de expirar	111
3. Expiración	111

4. Preaviso por expiración	111
5. Indexación	111
<b>Preguntas frecuentes</b>	<b>112</b>
1. Permisos	112
2. Atributos en el Portal de Anjana	112
3. Configuración de workflows	112
4. Despliegue de workflows	112
5. Asignación de roles en los pasos de los workflows	113
6. Error en la solicitud de adherencia	113
7. En el formulario de mi entidad o relación aparecen atributos que no deben estar	114
8. Necesito eliminar un atributo de una plantilla para la que ya tengo objetos creados	114
9. ¿Cómo configurar una nueva plantilla en Anjana?	114
10. Borrado de colección de Solr	115
11. Necesito cambiar el tipo de un atributo de plantilla	115
12. Error subiendo un fichero a una plantilla	115
13. ¿Cómo configurar una capa para el linaje?	115

## Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	22/09/2023	Anjana Producto	Anjana Producto	Creación del documento.
1.1	11/12/2023	Anjana Producto	Anjana Producto	Corregidos permisos necesarios para el funcionamiento de Portuno.
1.2	01/02/2024	Anjana Producto	Anjana Producto	Nota acerca de la actualización de traducciones desde Portuno. Configuración de autorización y autenticación en base de datos.
1.3	28/02/2024	Anjana Producto	Anjana Producto	Revisadas las variables del sistema de app_configuration. Incluido más detalle acerca de la configuración de los batches.
1.4	15/03/2024	Anjana Producto	Anjana Producto	Añadido más detalle en el flag only_on_edition de las validaciones de atributo.

# Introducción

El presente documento es una guía para configurar Anjana Data. A lo largo del documento se explicará el metamodelo de Anjana y se proporcionarán ejemplos para que tanto desarrolladores como administradores puedan configurar las plantillas, roles y permisos de Anjana Data para implementar así el modelo de gobierno de su organización en la solución.

## Conceptos básicos de Anjana Data

A continuación se describen una serie de conceptos fundamentales que tanto el desarrollador como el administrador deben asimilar para el correcto entendimiento de esta guía.

El modelo de gobierno de Anjana Data se organiza en **Unidades Organizativas** (*organizational units*). Las unidades organizativas que se definan en Anjana Data deben corresponderse con las áreas de negocio de su organización, los dominios de datos, los contenedores o cualquier otra estructura jerárquica que su organización considere oportuna.

Dentro de Anjana Data deben definirse una serie de **roles**, para el **Catálogo de Datos** y para el **Glosario de Negocio**, que deben corresponderse con los definidos dentro de su organización. Los roles pueden estar asociados solo a algunas unidades organizativas específicas o a todas ellas, en cuyo caso se trata de roles transversales (roles *cross*). Cada uno de estos roles dispondrá de un paquete de permisos a bajo nivel.

Los **Permisos** son los que habilitan a cada rol a realizar acciones a bajo nivel en Anjana Data. Existen distintos permisos para los objetos debido a las particularidades de cada uno de ellos.

Por último, los **Usuarios** son los usuarios nominales que se asignan a las personas para otorgarles acceso a Anjana Data. A cada usuario se le debe asignar al menos un rol asociado con una o más unidades organizativas.

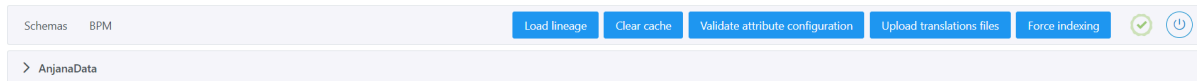
El Catálogo de Datos y el Glosario de Negocio se componen de Entidades y Relaciones. Cada una de éstas se corresponde con un **Subtipo de objeto** y para cada uno se configura en Anjana una plantilla con el conjunto de **atributos** que define su metadato.

## Visiones para la configuración

La configuración de Anjana debe ser llevada a cabo por usuarios cuyos roles les otorguen permisos de acceso al modelo de datos de la aplicación (permiso ADMIN ANJANA y API\_ADMIN ALL).

Hay dos vías o visiones posibles en Anjana para acceder a este modelo.

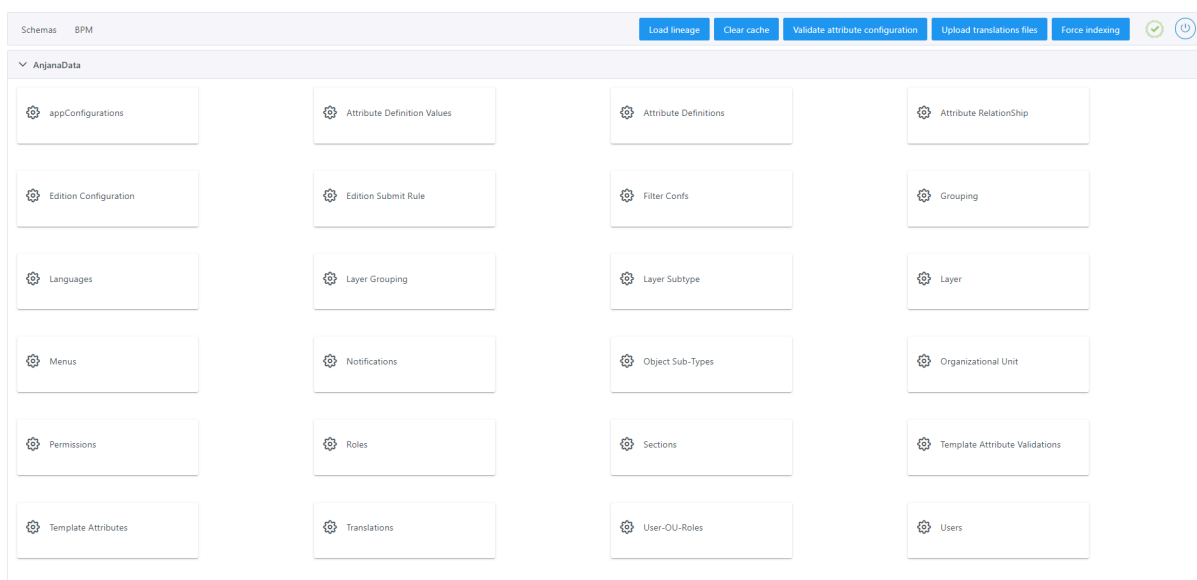
**Visión de Administración:** desde el Portal de Administración (o Portuno) el usuario podrá acceder al contenido de las tablas de configuración para poder hacer actualizaciones sobre ellas. Además, se necesita un permiso especial para configurar ciertas tablas de usuarios y roles como se verá más adelante (permiso CREDENTIAL\_ADMIN ANJANA).



### Opciones disponibles de este Portal de Administración:

- Schemas: acceso al contenido de las tablas de configuración de Anjana
- BPM: acceso al editor de workflows (más información en la Guía de configuración de Workflows)
- Load lineage: acción que genera el linaje total de entidades y relaciones para poder visualizarlo posteriormente en el Portal de Anjana
- Clear Cache: acción que permite limpiar cachés de la aplicación, recomendable cuando la configuración cambia en el modelo interno de Anjana o en cualquiera de los proveedores de identidades.
- Validate attribute configuration: acción que permite comprobar si los atributos que forman parte de la PK de los objetos tienen configuradas las validaciones de atributo requerido y no editable y que los valores configurados para los atributos de tipo taxonomía generan árboles correctos (sin ciclos, por ejemplo). Antes de lanzar esta validación es necesario limpiar la caché.
- Upload translations files: acción que actualiza los ficheros de traducciones de la aplicación, ubicados en Minio, con todo el contenido de la tabla portuno.translations de BD (Translations en Portuno). Los usuarios no tendrán disponibles los cambios en su Portal hasta el siguiente login que hagan
- Force indexing: acción que fuerza una indexación completa de todas las entidades y relaciones para evitar que BD y Solr estén desincronizadas y la información de ambas no sea coherente.

Desde Schema, se accede al catálogo completo de tablas de configuración de Anjana.



Para cada una de las tablas disponibles en este panel de administración se disponibilizan filtros y ordenaciones de las columnas.



Los filtros permiten buscar valores iguales, que comienzan, finalizan o contienen el texto que introduce el usuario o se muestran como listado de opciones posibles. También es posible filtrar por valores vacíos (nulos) o no vacíos.

Schemas BPM Load lineage Clear cache Validate attribute configuration Upload translations files Force indexing + New

Attribute Definitions

Id	Name	Type	Description	Label	Label Translation Key	Place Holder	Place Holder Translation Key	Short Description Translation Key	startDate	LM D
1	name	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.name	Name	ATTRIBUTE_NOMBRE.name	Name here	ATTRIBUTE_PLACEHOLDER.name	ATTRIBUTE_DESCRIPTION.name	01/03/2021	01,
2	termType	INPUT_DATE	ATTRIBUTE_DESCRIPTION.termType	Term type	ATTRIBUTE_NOMBRE.termType	Term type here	ATTRIBUTE_PLACEHOLDER.termType	ATTRIBUTE_DESCRIPTION.termType	01/03/2021	01,
3	domain	INPUT_RANGE	ATTRIBUTE_DESCRIPTION.domain	Domain	ATTRIBUTE_NOMBRE.domain	Domain here	ATTRIBUTE_PLACEHOLDER.domain	ATTRIBUTE_DESCRIPTION.domain	01/03/2021	01,
4	subdomain	SELECT	ATTRIBUTE_DESCRIPTION.subdomain	Subdomain	ATTRIBUTE_NOMBRE.subdomain	Subdomain here	ATTRIBUTE_PLACEHOLDER.subdomain	ATTRIBUTE_DESCRIPTION.subdomain	01/03/2021	01,
5	geography	MULTI_SELECT_IMG_TXT	ATTRIBUTE_DESCRIPTION.geography	Geography	ATTRIBUTE_NOMBRE.geography	Geography here	ATTRIBUTE_PLACEHOLDER.geography	ATTRIBUTE_DESCRIPTION.geography	01/03/2021	11,
6	des	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.des	Description	ATTRIBUTE_NOMBRE.des	Description here	ATTRIBUTE_PLACEHOLDER.des	ATTRIBUTE_DESCRIPTION.des	07/10/2020	01,
7	sdescription	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.sdescription	Short description	ATTRIBUTE_NOMBRE.sdescription	Short description here	ATTRIBUTE_PLACEHOLDER.sdescription	ATTRIBUTE_DESCRIPTION.sdescription	01/03/2021	01,
8	ontology	SELECT	ATTRIBUTE_DESCRIPTION.ontology	Ontology	ATTRIBUTE_NOMBRE.ontology	Ontology here	ATTRIBUTE_PLACEHOLDER.ontology	ATTRIBUTE_DESCRIPTION.ontology	01/03/2021	01,

Schemas BPM Load lineage Clear cache Validate attribute configuration Upload translations files Force indexing + New

Organizational Unit

Id	Description	Module	alias	parent	organizationalUnitType
2	Finance	ALL	Europe/SPA/Finance	1-Europe/SPA	2
3	Marketing	ALL	Europe/SPA/Marketing	1-Europe/SPA	2
4	Commercial	ALL	Europe/SPA/Commercial	1-Europe/SPA	2
5	Risk	ALL	Europe/SPA/Risk	1-Europe/SPA	2
6	Market	ALL	Europe/SPA/Market	1-Europe/SPA	2
7	Accounting	ALL	Europe/SPA/Accounting	1-Europe/SPA	2
41	HealthCare	ALL	Europe/SPA/HealthCare	1-Europe/SPA	2
1002	pruebaAna	ALL	Europe/SPA/pruebaAna	1-Europe/SPA	2

<< < 1 > >> 20

Para deshacer cualquiera de estos filtros, es necesario dejar en uno de los campos del filtro (operador o valor) el carácter '-' y volver a aplicarlo:

Schemas BPM Load lineage Clear cache Validate attribute configuration Upload translations files Force indexing + New

**Organizational Unit**

Id	Description	Module	alias	parent	organizationalUnitType		
1	SPA	ALL	Europe/SPA	100-Europe	1		
2	Finance	ALL	Europe/SPA/Finance	1-Europe/SPA	2		
3	Marketing	ALL	Europe/SPA/Marketing	1-Europe/SPA	2		
4	Commercial	ALL	Europe/SPA/Commercial	1-Europe/SPA	2		
5	Risk	ALL	Europe/SPA/Risk	1-Europe/SPA	2		
6	Market	ALL	Europe/SPA/Market	1-Europe/SPA	2		
7	Accounting	ALL	Europe/SPA/Accounting	1-Europe/SPA	2		
8	SCIB_HQ	ALL	General/SCIB_HQ	102-General	1		
9	Finance	ALL	General/SCIB_HQ/Finance	8-General/SCIB_HQ	2		
10	Accounting	ALL	General/SCIB_HQ/Accounting	8-General/SCIB_HQ	2		

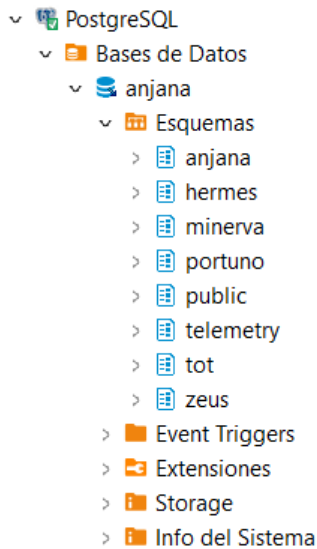
Para visualizar los registros de las tablas ordenados es necesario clicar sobre el nombre de la columna hasta conseguir la ordenación ascendente o descendente deseada. Es posible ordenar por varias columnas pulsando la tecla Ctrl a la vez que se clicca sobre el nombre de la columna a ordenar.

Schemas BPM Load lineage Clear cache Validate attribute configuration Upload translations files Force indexing + New

**Notifications**

Id	moduleType	notificationCode	notificationReceiverType	translationKey	notificationType	receiverRole	severity	subject		
17	DC	ADHERENCE_FAIL	ROLE	NOTIFICATION.ADHERENCE_FAIL	ADMIN_ALERT	administrator	HIGH	NOTIFICATION.SUBJECT.ADHERENCE.FA ILDSA		
57	DC	CHECK_CONFIGURATION_IN_PROVIDER	ROLE	NOTIFICATION.CHECK_CONFIGURATIO N_IN_PROVIDER	ADMIN_ALERT	administrator	HIGH	NOTIFICATION.SUBJECT.CHECK.CONFIG URATION.IN.PROVIDER		
15	DC	DATASET_FAIL	ROLE	NOTIFICATION.DATASET_FAIL	ADMIN_ALERT	administrator	HIGH	NOTIFICATION.SUBJECT.CREATION.FAIL. DATASET		
78	DC	DELETE_ENTITY	USER	NOTIFICATION.DELETE_ENTITY	NOTICE	[NULL]	LOW	NOTIFICATION.SUBJECT.DELETE_ENTITY		
75	DC	DELETE_ENTITY	USER	NOTIFICATION.DELETE_ENTITY	NOTICE	[NULL]	LOW	NOTIFICATION.SUBJECT.DELETE_ENTITY		
79	DC	DELETE_RELATIONSHIP	USER	NOTIFICATION.DELETE_RELATIONSHIP	NOTICE	[NULL]	LOW	NOTIFICATION.SUBJECT.DELETE_RELATI ONSHIP		
76	DC	DELETE_RELATIONSHIP	USER	NOTIFICATION.DELETE_RELATIONSHIP	NOTICE	[NULL]	LOW	NOTIFICATION.SUBJECT.DELETE_RELATI ONSHIP		
21	DC	DEPRECATION	ROLE	NOTIFICATION.DEPRECACTION	NOTICE	data_owner	LOW	NOTIFICATION.SUBJECT.DEPRECACTION		
19	DC	DEPRECATION_ADHERED	USER	NOTIFICATION.DSA_DEPRECATED	NOTICE	[NULL]	LOW	NOTIFICATION.SUBJECT.CREATION.FAIL. DSA		

**Visión de Desarrollador:** acceso del usuario administrador directamente a las tablas de BD.



Es recomendable que los usuarios que configuren Anjana con esta opción (Visión desarrollador) ejecuten al finalizar la actualización de las secuencias de las tablas de BD con el siguiente sql:

```
SELECT setval('anjana.sc_pk_attribute_definition', (SELECT
COALESCE(MAX(id_attribute_definition), 0) FROM anjana.attribute_definition) + 1);
SELECT setval('anjana.sc_pk_attribute_definition_value', (SELECT
COALESCE(MAX(id_attribute_definition_value), 0) FROM anjana.attribute_definition_value) + 1);
SELECT setval('anjana.sc_pk_attribute_relationships', (SELECT
COALESCE(MAX(id_attribute_relationships), 0) FROM anjana.attribute_relationships) + 1);
SELECT setval('anjana.sc_pk_edition_configuration', (SELECT COALESCE(MAX(id_edition_conf), 0)
FROM anjana.edition_configuration) + 1);
SELECT setval('anjana.sc_pk_edition_submit_rule', (SELECT
COALESCE(MAX(id_edition_submit_rule), 0) FROM anjana.edition_submit_rule) + 1);
SELECT setval('anjana.sc_pk_grouping', (SELECT COALESCE(MAX(id), 0) FROM
anjana.grouping_lineage) + 1);
SELECT setval('anjana.sc_pk_layer', (SELECT COALESCE(MAX(id), 0) FROM anjana.layer) + 1);
SELECT setval('anjana.sc_pk_menu', (SELECT COALESCE(MAX(id_menu), 0) FROM anjana.menu) +
1);
SELECT setval('anjana.sc_pk_object_subtype', (SELECT COALESCE(MAX(id_object_subtype), 0)
FROM anjana.object_subtype) + 1);
SELECT setval('anjana.sc_pk_sections', (SELECT COALESCE(MAX(id_section), 0) FROM
anjana.sections) + 1);
SELECT setval('anjana.sc_pk_template_attribute', (SELECT COALESCE(MAX(id_template_attribute),
0) FROM anjana.template_attribute) + 1);
SELECT setval('hermes.sc_pk_notification', (SELECT COALESCE(MAX(id_notification), 0) FROM
hermes.notification) + 1);
SELECT setval('minerva.sc_pk_filter_conf', (SELECT COALESCE(MAX(id), 0) FROM
minerva.filter_conf) + 1);
SELECT setval('portuno.sc_pk_app_configuration', (SELECT COALESCE(MAX(id), 0) FROM
portuno.app_configuration) + 1);
SELECT setval('portuno.sc_pk_translations', (SELECT COALESCE(MAX(id_translations), 0) FROM
portuno.translations) + 1);
SELECT setval('zeus.sc_pk_languages', (SELECT COALESCE(MAX(id), 0) FROM zeus.languages) + 1);
SELECT setval('zeus.sc_pk_organizational_unit', (SELECT COALESCE(MAX(id_organizational_unit),
```

```
0) FROM zeus.organizational_unit )+ 1);  
SELECT setval('zeus.sc_pk_permission', (SELECT COALESCE(MAX(id_permission), 0) FROM  
zeus.permission) + 1);  
SELECT setval('zeus.sc_pk_role', (SELECT COALESCE(MAX(id_role), 0) FROM zeus.role) + 1);  
SELECT setval('zeus.sc_pk_user', (SELECT COALESCE(MAX(id_user), 0) FROM zeus.users) + 1);
```

## Configuración en BD

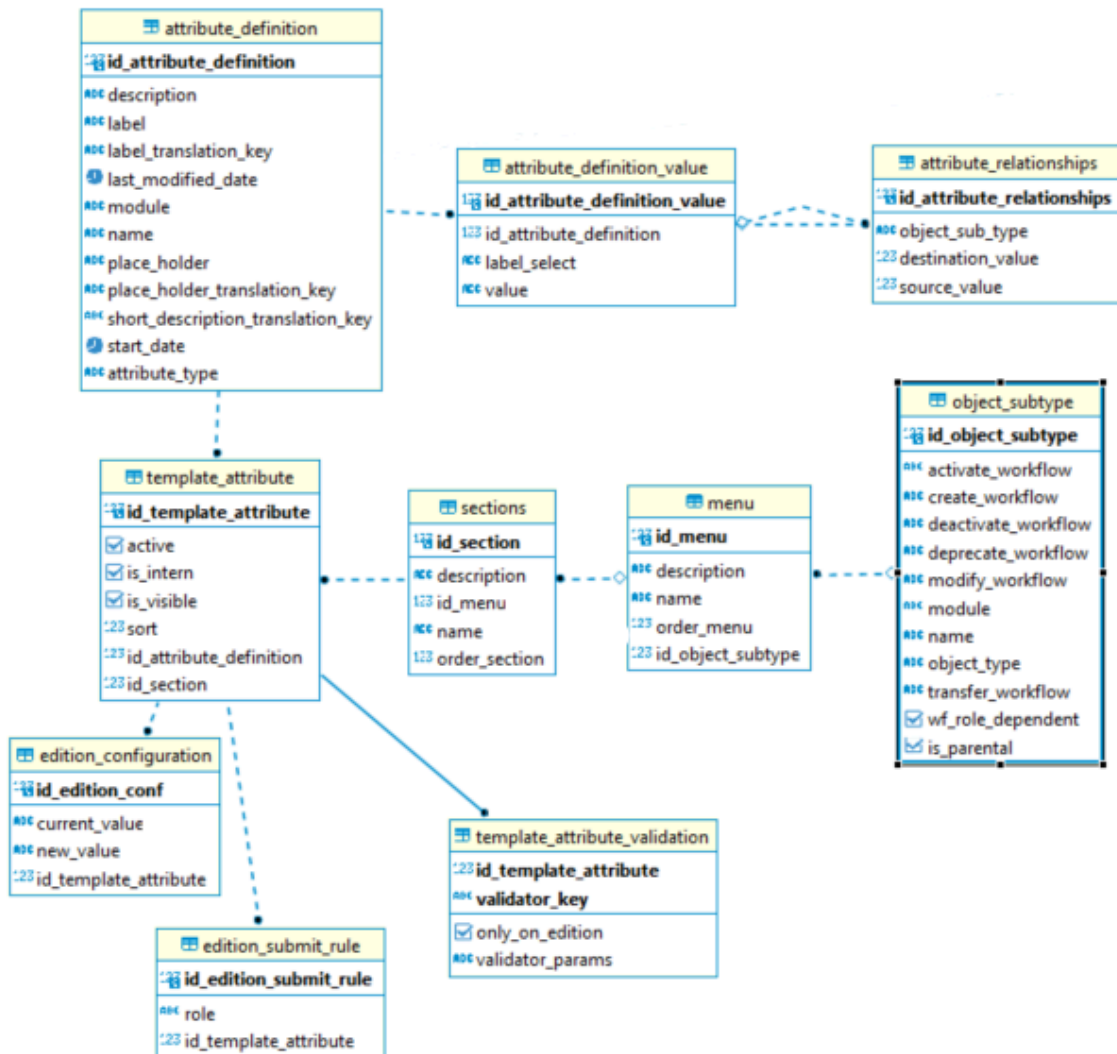
Las tablas de configuración de Anjana Data se agrupan en cuatro esquemas:

- **Anjana:** Esquema que permite configurar los formularios dinámicos de Anjana Data junto con las entidades y relaciones.
- **Hermes:** Esquema que contiene las tablas de Activiti y de Anjana para el despliegue y ejecución de los workflows y las notificaciones que se envían en Anjana Data.
- **Minerva:** Esquema que incluye la configuración para el filtrado de las colecciones de SolR.
- **Portuno:** Esquema que incluye configuraciones de la aplicación y traducciones para el multidioma.
- **Zeus:** Esquema que contiene las tablas relacionadas con los permisos, roles, unidades organizativas, idioma, etcétera.

## Esquema Anjana de BD

La configuración de las tablas del esquema Anjana permite definir los distintos tipos de entidades y relaciones que se podrán declarar en Anjana Data y sus metadatos.

El modelo de base de datos se muestra en la siguiente imagen:



## 1. Configurar subtipos de objetos: tipos de entidades y relaciones

Anjana Data permite gobernar diferentes tipos de entidades así como las relaciones entre ellos. Las entidades nativas del Catálogo de Datos son:

- Dataset
- Dataset field
- DSA
- Process
- Instance
- Solution

Y, entre estas entidades, se crean en Anjana relaciones nativas:

- **Structure:** entre un dataset (origen de la relación) y sus dataset fields (destino)
- **DSA\_content:** entre un DSA (origen de la relación) y las entidades que contenga (destino)
- **Adherence:** entre un DSA (origen de la relación) y un User (destino)
- **Instance\_process:** pertenencia de una instancia (destino de la relación) a su proceso (origen)

- **Instance\_dataset\_in:** entre una instancia (destino de la relación) y los datasets que lee de los que lee (origen)
- **instance\_dataset\_out:** entre una instancia (origen de la relación) y los datasets en los que escribe (destino)
- **Solution\_related\_instance:** entre solución (origen) y sus instancias relacionadas (destino)
- **Solution\_owned\_instance:** pertenencia de una instancia (destino) a su solución propietaria (origen)

No obstante, el metamodelo de Anjana es flexible y permite añadir al catálogo citado entidades y relaciones no nativas.

De esta forma se puede definir, sin restricciones, cualquier activo funcional o técnico siempre indicando si pertenece al Glosario de Negocio o al Catálogo de Datos para garantizar el buen funcionamiento de los filtros, por ejemplo:

- **Term:** Términos de negocio (Glosario de Negocio)
- **Report:** Informes, reportes y Cuadros de Mandos (Glosario de Negocio o Catálogo de datos, según perspectiva de cada organización)
- **KPI:** Indicadores y métricas (Glosario de Negocio)
- **Dimension:** Dimensiones (Glosario de Negocio)
- **DQ Rule:** Reglas de calidad de datos (Glosario de Negocio)
- **Policy:** Políticas (Glosario de Negocio)
- **Data base:** Bases de datos (Catálogo de datos)
- **View:** Vistas (Catálogo de datos)

Las entidades se asocian entre sí mediante relaciones. Es posible dar de alta cualquier relación que resulte de utilidad, de esta forma, tendríamos relaciones tipadas como por ejemplo:

- Término - dataset
- Policy - DSA
- Report - KPI
- Report - dimension
- Dimension - dataset
- Report - DQ Rule
- DQ Rule - dataset
- ...

Las entidades o relaciones se configuran en Anjana en la tabla de `object_subtype` y se visualizan en los wizard de creación en función de los permisos del usuario:

Panel de creación Buscar en ▼ Buscar... 🔍 + 📄 📁 📊 📋 🔔 👤 ana 👍 🗑️

1 Seleccione el tipo de objeto 2 Seleccione la forma de creación 3 Información completa

**Entidad:** Objeto con existencia distinta e independiente Relación: Una conexión entre entidades

Buscar una entidad... 🔍 Buscar una relación... 🔍

DATASET	DATA_BASE	DSA	AQIVA_DQ_CONTROL	ASSOCIATED_BUSINESS_PROCE...	ASSOCIATED_TERMS
INSTANCE	PROCESS	QA_BG	CONTROL_DQ	DIMENSION_DATASET	DQ_RULE_DATASET
SOLUTION	TEST_SUBTYPE	AQIVA_RULE	KPI_DATASET	POLICY_DSA	RELATED_DATASET
BUSINESS_PROCESS	DATA_QUALITY_RULE	DIMENSION	RELATED_DQ_RULES	RELATED_MODEL	RELATED_REPORTS
DOCUMENT	KPI	POLICY	RELATION_PARENT_TERM	REPORT_ATTRIBUTES	REPORT_DATASET
REPORT	TERM		TERM_DATASET		

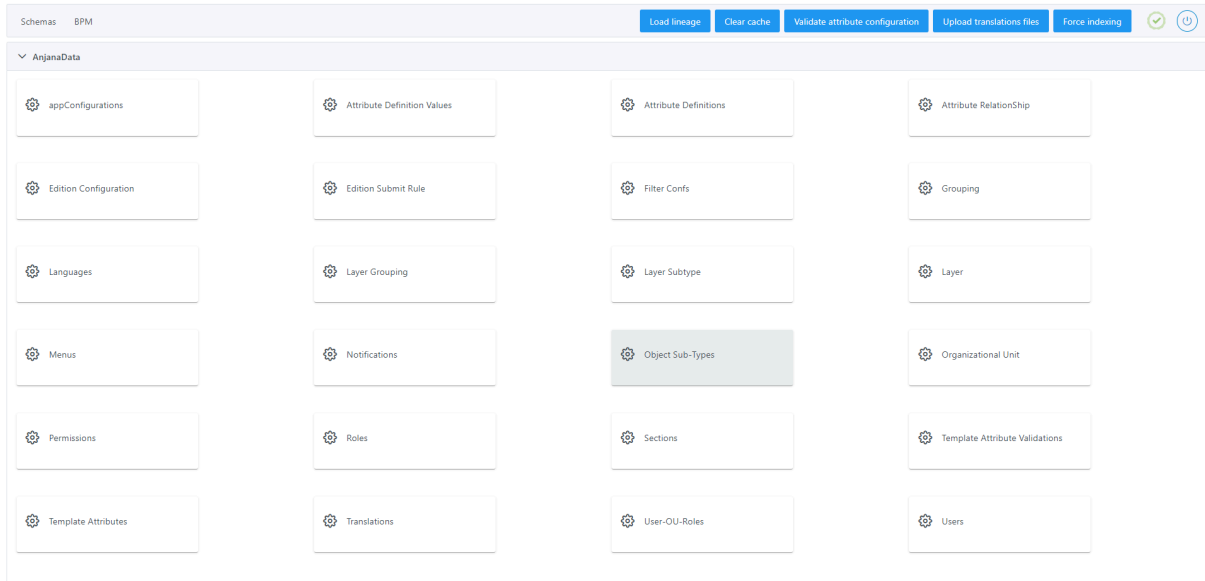
## Estructura de la tabla

Cada subtipo de objeto registrado se caracteriza por los siguientes elementos:

- **id\_object\_subtype:** identificador único de la tabla.
- **name:** nombre del subtipo de objeto.
- **module:** módulo al que pertenece, indicar “BG” para *Business Glossary* (Glosario de Negocio), “DC” para *Data Catalog* (Catálogo de Datos).
- **object\_type:** indica si se trata de una entidad “ENTITY” o una relación “RELATIONSHIP”.
- **activate\_workflow:** nombre del workflow de activación de entidades no nativas y relaciones
- **create\_workflow:** nombre del workflow de creación.
- **deactivate\_workflow:** nombre del workflow de desactivación de entidades no nativas y relaciones.
- **modify\_workflow:** nombre del workflow de edición.
- **transfer\_workflow:** nombre del workflow de transferencia (cambio de unidad organizativa) de entidades.
- **deprecate\_workflow:** nombre del workflow de deprecación de entidades nativas del catálogo de datos.
- **wf\_role\_dependent:** indica si hay configurados distintos workflows en función del usuario que lance el workflow, es decir, si existe un workflow configurado para el caso de que sea un data\_owner quien envíe a validar un objeto distinto a si quien envía el mismo objeto a validar es un data\_steward.
- **is\_parental:** indica, en caso de ser una relación, si ésta es de parentesco y, por tanto, en el linaje se mostrará como una relación de padre-hijo (arriba y abajo). En caso contrario, la relación será de input-output (izquierda y derecha).

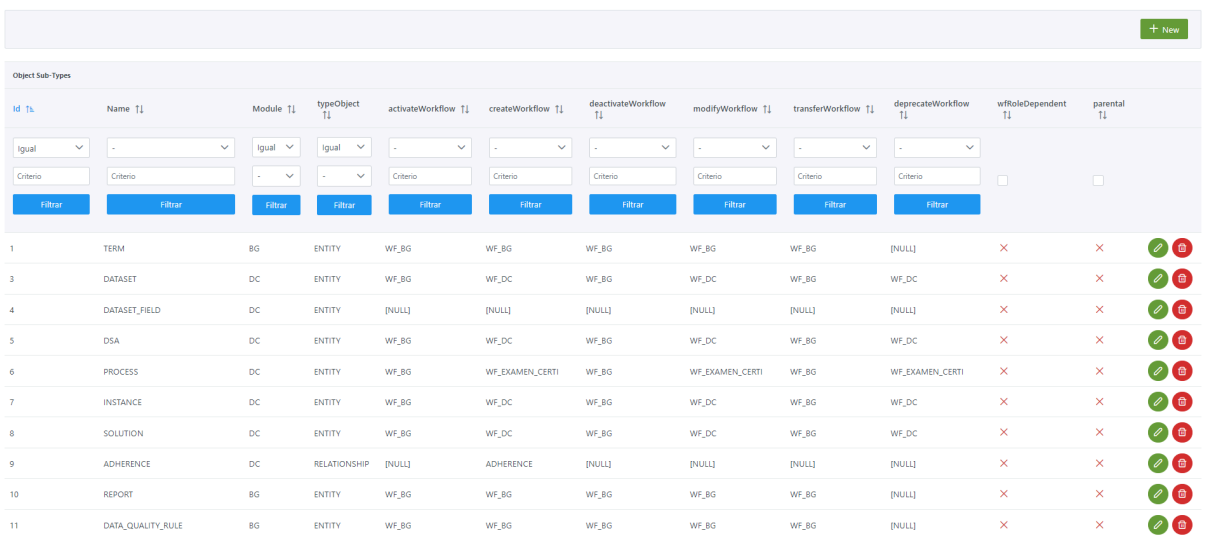
## Visión de Administrador

El alta de entidades o relaciones mediante el Panel de Administrador de Anjana Data se realiza en la tabla **Object Sub-types**:



Al acceder, se muestra una tabla que contiene todas las entidades y relaciones existentes en la configuración actual.

La creación de un nuevo tipo de entidad/relación se realiza mediante el botón **New**:



												+ New	
Object Sub-Types													
Id	Name	Module	typeObject	activateWorkflow	createWorkflow	deactivateWorkflow	modifyWorkflow	transferWorkflow	deprecateWorkflow	wRoleDependent	parental		
1	TERM	BG	ENTITY	WF_BG	WF_BG	WF_BG	WF_BG	WF_BG	[NULL]	×	×		
3	DATASET	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	WF_BG	WF_DC	×	×		
4	DATASET_FIELD	DC	ENTITY	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	×	×		
5	DSA	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	WF_BG	WF_DC	×	×		
6	PROCESS	DC	ENTITY	WF_BG	WF_EXAMEN_CERTI	WF_BG	WF_EXAMEN_CERTI	WF_BG	WF_EXAMEN_CERTI	×	×		
7	INSTANCE	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	WF_BG	WF_DC	×	×		
8	SOLUTION	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	WF_BG	WF_DC	×	×		
9	ADHERENCE	DC	RELATIONSHIP	[NULL]	ADHERENCE	[NULL]	[NULL]	[NULL]	[NULL]	×	×		
10	REPORT	BG	ENTITY	WF_BG	WF_BG	WF_BG	WF_BG	WF_BG	[NULL]	×	×		
11	DATA_QUALITY_RULE	BG	ENTITY	WF_BG	WF_BG	WF_BG	WF_BG	WF_BG	[NULL]	×	×		

Mediante el wizard de creación se elige el módulo al que pertenece el objeto, el tipo y el nombre que se desea crear. A continuación, se muestra cómo crear una nueva entidad en el Data Catalog ('DC') de nombre DOCUMENT:



### Object Sub-Types Edit ✕

Name DOCUMENT	Module BG	typeObject ENTITY
activateWorkflow	createWorkflow WF_DC	deactivateWorkflow
modifyWorkflow WF_DC	transferWorkflow	deprecateWorkflow
<input type="checkbox"/> wfRoleDependent	<input type="checkbox"/> parental	

✕ Cancel
✓ Save

Junto con la definición de la nueva entidad o relación, se definen los nombres de los workflows que se lanzarán cuando un usuario solicite la aprobación o rechazo de algunas de las diferentes acciones que se pueden llevar a cabo dentro de Anjana Data sobre esa nueva entidad o relación.

### Visión de Desarrollador

Para catalogar los distintos activos de Anjana y el nombre de sus workflows hay que configurar la tabla **object\_subtype** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```

INSERT INTO anjana.object_subtype
(id_object_subtype, activate_workflow, create_workflow, deactivate_workflow,
deprecate_workflow, modify_workflow, "module", "name", object_type, transfer_workflow,
wf_role_dependent, is_parental) VALUES
(1, 'WF_BG', 'WF_BG', 'WF_BG', NULL, 'WF_BG', 'BG', 'TERM', 'ENTITY', 'WF_BG', false, false),
(3, 'WF_BG', 'WF_DC', 'WF_BG', 'WF_DC', 'WF_DC', 'DC', 'DATASET', 'ENTITY', 'WF_BG', false, true),
(4, NULL, NULL, NULL, NULL, "", 'DC', 'DATASET_FIELD', 'ENTITY', NULL, false, false),
(5, 'WF_BG', 'WF_DC', 'WF_BG', 'WF_DC', 'WF_DC', 'DC', 'DSA', 'ENTITY', 'WF_BG', false, true),
(6, 'WF_BG', 'WF_EXAMEN_CERTI', 'WF_BG', 'WF_EXAMEN_CERTI', 'WF_EXAMEN_CERTI', 'DC',
'PROCESS', 'ENTITY', 'WF_BG', false, false),
(7, 'WF_BG', 'WF_DC', 'WF_BG', 'WF_DC', 'WF_DC', 'DC', 'INSTANCE', 'ENTITY', 'WF_BG', false,
false),
(8, 'WF_BG', 'WF_DC', 'WF_BG', 'WF_DC', 'WF_DC', 'DC', 'SOLUTION', 'ENTITY', 'WF_BG', false,
false),
(9, NULL, 'ADHERENCE', NULL, NULL, NULL, 'DC', 'ADHERENCE', 'RELATIONSHIP', NULL, false, false),
(13, 'WF_BG', 'WF_BG', 'WF_BG', NULL, 'WF_BG', 'BG', 'KPI', 'ENTITY', 'WF_BG', false, false),
(20, 'WF_BG', 'WF_BG', 'WF_BG', NULL, 'WF_BG', 'BG', 'RELATED_REPORTS', 'RELATIONSHIP', NULL,
false, false),
(21, 'WF_BG', 'WF_BG', 'WF_BG', NULL, 'WF_BG', 'BG', 'RELATED_DATASET', 'RELATIONSHIP', NULL,
false, false);

```

#### NOTAS:

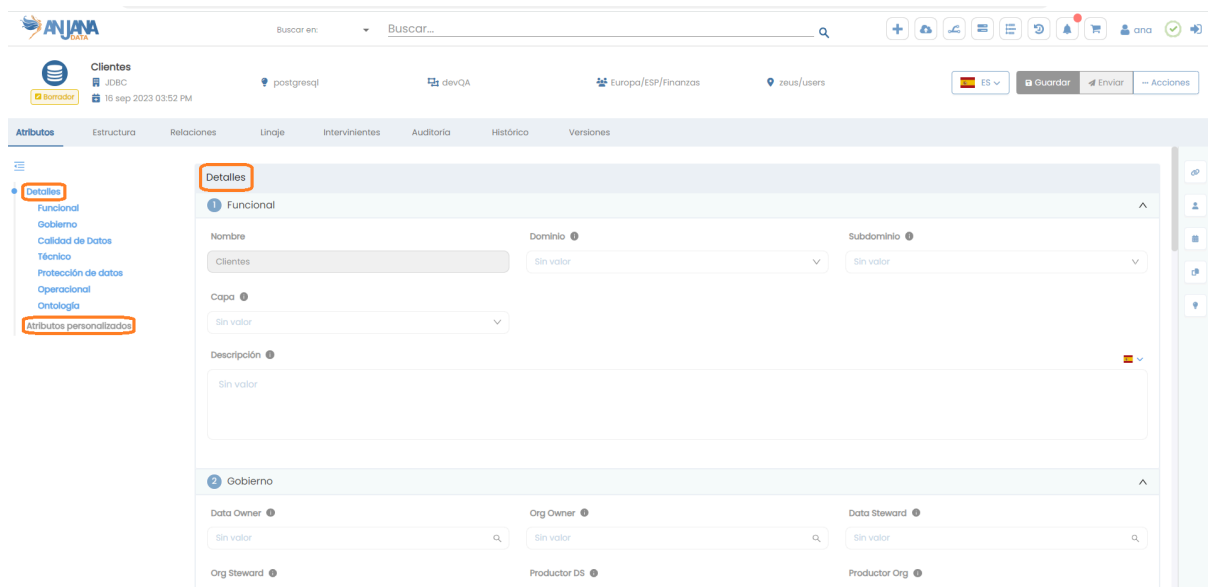
- El nombre de los subtipos debe no contener ':', '#', '(', ')' o espacios para no interferir con los identificadores internos de Anjana y deben estar escritos en mayúsculas.

- El nombre de los subtipos no tiene traducción.
- El nombre de los subtipos no debe cambiar si ya existen objetos para ese subtipo en Anjana.
- La relación ADHERENCE, nativa, debe ser configurada en object\_subtype para poder informar del workflow a lanzar cuando un usuario solicite adherencia a un DSA. En esta relación sólo es necesario configurar el workflow de creación “create\_workflow” quedando los demás vacíos.
- Cuando se añade un subtipo de objeto nuevo es necesario configurar las capas donde se desea visualizar en la tabla [layer\\_subtype](#).

## 2. Configurar el Menú

Los menús son los bloques de mayor nivel de los formularios dinámicos de Anjana Data. Estos se registran en la tabla **menu**, son totalmente configurables y es posible añadir cuantos menús se desee en cada plantilla de los diferentes tipos de objeto (entidad o relación).

Los menús son los contenedores de las distintas secciones y estas, a su vez, son las que contienen los atributos de metadatos de cada objeto.



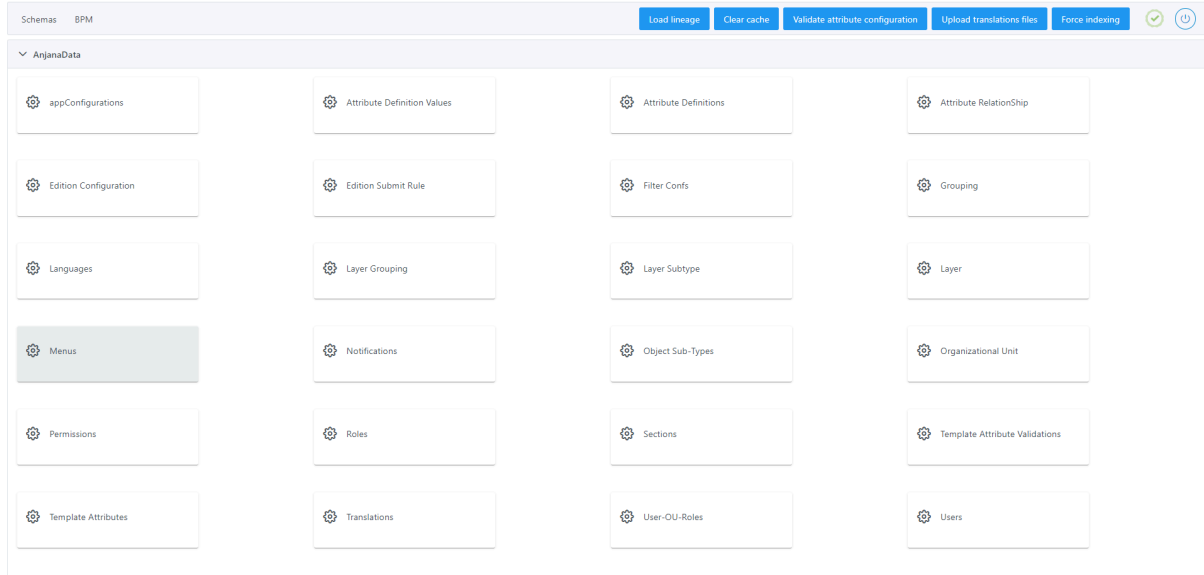
### Estructura de la tabla

Cada menú se caracteriza por los siguientes elementos:

- **id\_menu**: identificador único de la tabla.
- **name**: nombre del menú.  
En caso de querer visualizarlo en los distintos idiomas de la aplicación debe usarse el name como clave de traducción en la tabla de portuno de translations.
- **description**: descripción del menú. Es una buena práctica indicar a qué objeto corresponde.
- **order\_menu**: indica el orden en que se muestran los distintos menús para un mismo objeto.
- **id\_object\_subtype**: indica el subtipo de objeto para el cual se define el menú.

## Visión de Administrador

El alta de un nuevo menú en el panel de administración de Anjana Data se realiza en **Menus**:



Al acceder se muestra una tabla que contiene todos los menús existentes en la configuración actual.

La creación de un nuevo objeto se realiza mediante el botón **New**:

+ New

Menus				
id	Name	Description	order	objectSubType
-	-	-	-	Igual
1	TERMMENU	General information of Business Term (Functional,Technical,operational)	1	1-TERM
2	REPORTSMENU	General information of Reports (Functional,Technical,operational)	1	10-REPORT
3	DQRULESMENU	General information of Data Quality Rules (Functional,Technical,operational)	1	11-DATA_QUALITY_RULE
4	BUSSINESPROCESSMENU	General information of Business Processes (Functional,Technical,operational)	1	12-BUSINESS_PROCESS
5	KPISMENU	General information of KPIs (Functional,Technical,operational)	1	13-KPI
6	DIMENSIONSMENU	General information of Dimensions (Functional,Technical,operational)	1	14-DIMENSION
7	POLICIESMENU	General information of Policies (Functional,Technical,operational)	1	15-POLICY

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos y se selecciona mediante el combo el subtipo de objeto (template) para el cual se define el menú.

A continuación, se muestra cómo crear un menú para el objeto DOCUMENT creado con anterioridad:

Menus Edit
✕

Name

order

objectSubType

Description  

Details of a document

✕ Cancel
✓ Save

### Visión de Desarrollador

Para definir los distintos menús hay que configurar la tabla **menu** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```

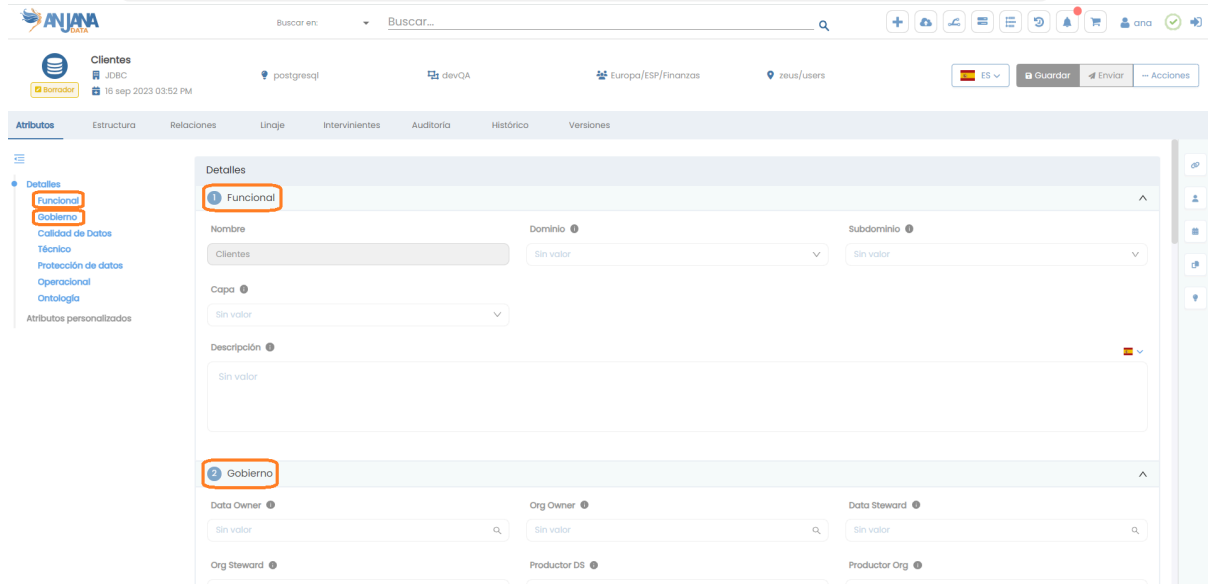
INSERT INTO anjana.menu (id_menu, description, "name", order_menu, id_object_subtype)
VALUES
(1, 'General information of Business Term (Functional,Technical,operational)', 'DETAILS', 1, 1),
(2, 'General information of Reports (Functional,Technical,operational)', 'DETAILS', 1, 10),
(8, 'General information of Relationships (Functional,Technical,operational)', 'DETAILS', 1, 16),
(16, 'General information of DSA (Functional,Technical,operational)', 'DETAILS', 1, 5),
(27, 'Licensing Terms of DSA', 'LICENSING TERMS', 2, 5);
  
```

#### NOTAS:

- No se puede definir un menú para un subtipo de objeto que no ha sido creado aún.
- Una plantilla de subtipo de objeto debe contener, al menos, un menú.
- El nombre de los menús puede ser traducible a los idiomas de la aplicación en caso de que se utilice ese nombre como clave en la tabla de portuno.translations.
- A continuación de los menús configurados para los distintos subtipos de objetos siempre se añade el menú “Atributos personalizados”. Para ello, la ordenación de los menús configurados se respeta añadiendo, posteriormente, este nuevo menú a cada plantilla. En este menú se podrán dar de alta atributos específicos de cada objeto en particular.
- Aunque ADHERENCE no tiene vista propia, es necesario definir un menú para esta relación para poder añadir los atributos necesarios que debe contener.

### 3. Configurar las secciones (sections)

Los atributos de metadatos se agrupan bajo secciones que permiten clasificar el conjunto de metadatos que se va a visualizar. Las secciones son totalmente configurables para cada tipo de objeto y deben estar contenidas dentro de un menú. Estas se registran en la tabla **sections**.



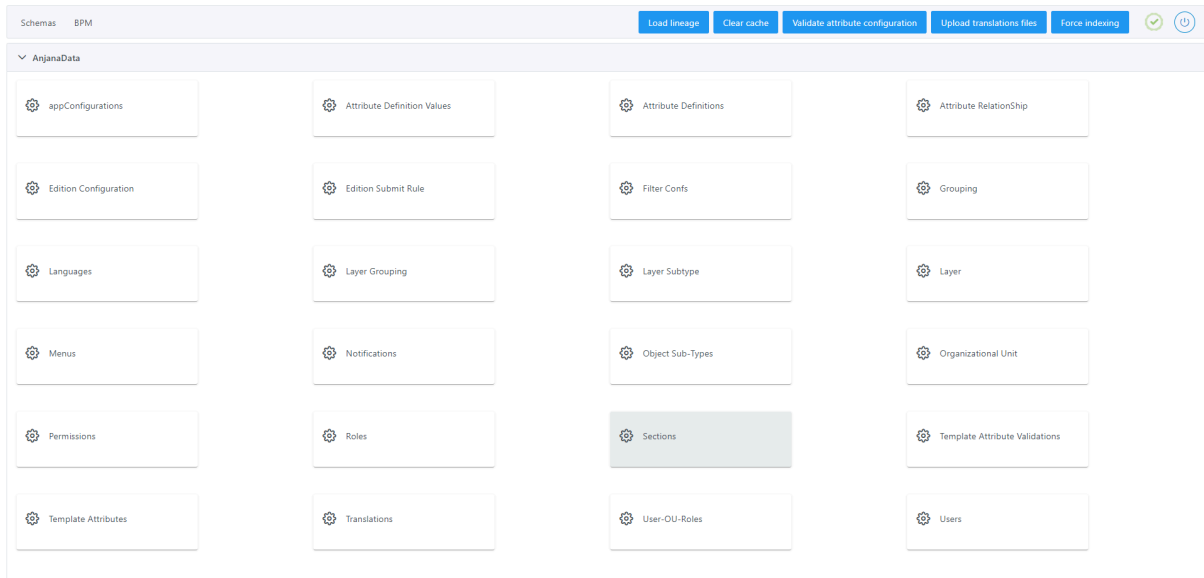
#### Estructura de la tabla

Cada sección se caracteriza por los siguientes elementos:

- **id\_section:** identificador único de la tabla.
- **name:** nombre de la sección en Anjana Data.  
En caso de querer visualizarlo en los distintos idiomas de la aplicación debe usarse el name como clave de traducción en la tabla de portuno de translations.
- **description:** descripción de la sección. Es una buena práctica indicar a qué objeto corresponde.
- **order\_section:** indica el orden en que se muestran las distintas secciones dentro de un menú
- **id\_menu:** indica el menú para el cual se define la sección.

#### Visión de Administrador

El alta de una nueva sección en el panel de administración de Anjana Data se realiza en **Sections**:



Al acceder se muestra una tabla que contiene todas las secciones existentes en la configuración actual.

La creación de una nueva sección se realiza mediante el botón **New**:

+ New

Sections				
id	Name	Description	order	menu
-	-	-	-	Igual
1	TERMPROPERTIESSECTION	Properties of Business Terms	1	1-TERMMENU-TERM
2	TERMOPERATIONALSECTION	Operational information of Business Terms	2	1-TERMMENU-TERM
3	TERMSECURITYSECTION	Security information of Business Terms	3	1-TERMMENU-TERM
4	TERMQUALITYSECTION	Quality information of Business Terms	4	1-TERMMENU-TERM
5	TERMSOCIALSECTION	Social information of Business Terms	5	1-TERMMENU-TERM
6	REPORTPROPERTIESSECTION	Properties of Reports	1	2-REPORTSMENU-REPORT
7	REPORTOPERATIONALSECTION	Operational information of Reports	2	2-REPORTSMENU-REPORT
8	REPORTSECURITYSECTION	Security information of Reports	3	2-REPORTSMENU-REPORT

Mediante el wizard de creación se asignan valores a los elementos name, order y description, y mediante el combo de menu se selecciona uno de los menús existentes para el cual se define la sección.

A continuación, se muestra cómo crear una sección llamada PROPERTIES para el menú de DETAILS del objeto DOCUMENT creado con anterioridad:

**Sections Edit** ✕

Name:     order:     menu:

Description:

Properties of Documents

### Visión de Desarrollador

Para definir las secciones hay que configurar la tabla **sections** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```

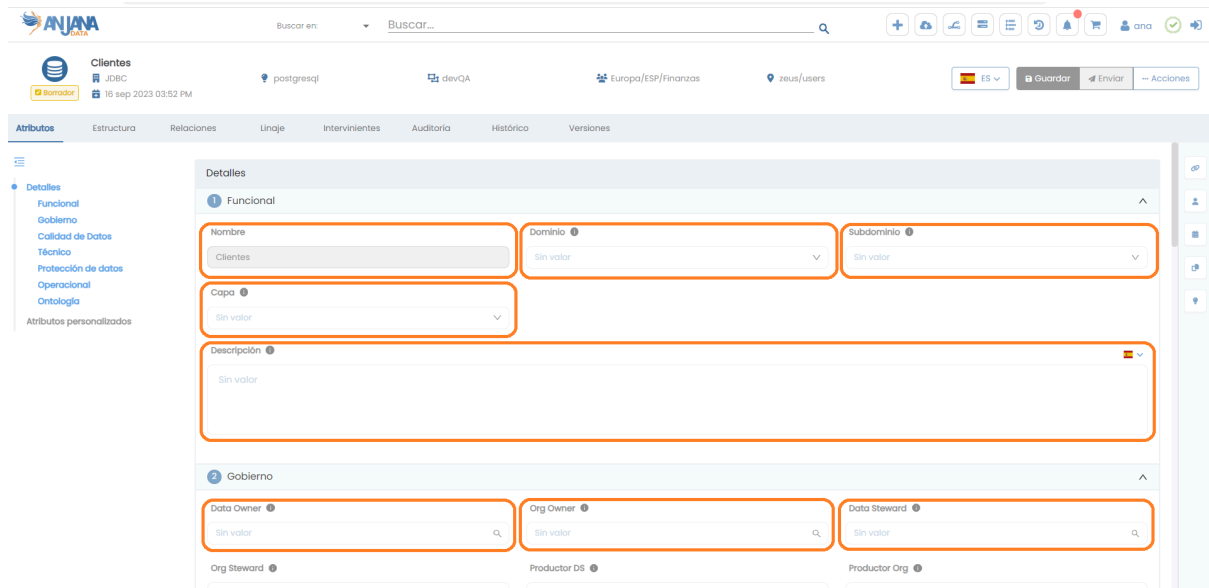
INSERT INTO anjana.sections (id_section, description, id_menu, "name", order_section)
VALUES
(1, 'Properties of Business Terms', 1, 'PROPERTIES', 1),
(2, 'Operational information of Business Terms', 1, 'OPERATIONAL', 2),
(3, 'Security information of Business Terms', 1, 'SECURITY', 3),
(4, 'Quality information of Business Terms', 1, 'QUALITY', 4),
(5, 'Social information of Business Terms', 1, 'SOCIAL', 5),
(55, 'Functional information of DSA', 16, 'FUNCTIONAL', 1),
(56, 'Governance information of DSA', 16, 'GOVERNANCE', 2),
(57, 'Operational information of DSA', 16, 'OPERATIONAL', 3),
(58, 'Taxonomies information of DSA', 16, 'TAXONOMIES', 4),
(59, 'Terms of use information of DSA', 27, 'TERMS OF USE', 1);
  
```

**NOTAS:**

- No se puede definir una sección para un menú que no ha sido creado aún.
- Aunque ADHERENCE no tiene vista propia, es necesario definir una sección para esta relación para poder añadir los atributos necesarios que debe contener.

## 4. Definir y configurar los atributos de metadatos

Los atributos de metadatos son datos que hablan de los objetos (entidades y relaciones) que representan los datos. Estos se configuran en la tabla **attribute\_definition**:



En Anjana Data los distintos tipos de atributos de metadatos soportados son los siguientes:

- **Array de boolean:** atributo para indicar uno o varios valores true o false
- **Array de date:** atributo para indicar uno o varias fechas
- **Array de decimal:** atributo para indicar uno o varios números decimales
- **Array de entities:** atributo para elegir una o varias entidades aprobadas en Anjana
- **Array de file:** atributo para adjuntar uno o varios ficheros
- **Array de number:** atributo para indicar uno o varios números enteros
- **Array de Organizational Unit:** atributo para seleccionar una o varias unidades organizativas del listado de todas ellas
- **Array de text:** atributo para introducir uno o varios textos cortos de hasta 255 caracteres
- **Array de URL:** atributo para introducir uno o varios links a URLs navegables
- **Array de users:** atributo para elegir uno o varios usuario de la lista completa de usuarios de la aplicación
- **Boolean:** true o false
- **Date:** fecha (año, mes, día, hora y minutos)
- **Decimal:** número con decimales
- **Enriched Text Area:** atributo para introducir un texto largo de hasta 300 mil caracteres enriquecido con formato (negrita, subrayados, cursivas etc)
- **Entity Container:** atributo para elegir una entidad aprobada en Anjana Data y generar relaciones nativas entre entidades. Sólo es posible utilizar este atributo en las entidades DSA, instancia y solución
- **Entity Search:** atributo para elegir una entidad aprobada en Anjana Data
- **File:** fichero que se almacena internamente en Anjana Data. También se permite descargar el fichero si se tiene permisos de lectura
- **International Text:** cuadro de texto normal disponible para los distintos tipos de idiomas disponibles de la aplicación
- **International Text Editor:** atributo para introducir un texto largo de hasta 300 mil caracteres enriquecido con formato (negrita, subrayados, cursivas etc) disponible para los distintos tipos de idiomas disponibles de la aplicación
- **International Textarea:** atributo para introducir un texto largo de hasta 300 mil caracteres en los distintos idiomas disponibles para la aplicación



- **MultiSelect**: atributo para seleccionar uno o varios valores de una lista preconfigurada en la pestaña Reference Metadata
- **MultiSelect con iconos**: atributo para seleccionar uno o varios iconos de una lista preconfigurada en la pestaña Reference Metadata
- **MultiSelect con iconos y texto**: atributo para seleccionar uno o varios valores (icono+texto) de una lista preconfigurada en la pestaña Reference Metadata
- **Number**: atributo para indicar un número entero
- **Number range**: selector de número entero entre un mínimo y un máximo definidos
- **Organizational Unit**: atributo para seleccionar una unidad organizativa del listado de todas ellas
- **Reference Metadata**: lista de valores posibles que se tiene que definir para el atributo
- **Selector con icono**: lista de iconos posibles que se tiene que definir para el atributo
- **Selector con icono y texto**: lista de valores (icono+texto) posibles que se tiene que definir para el atributo
- **Taxonomía única**: árbol de taxonomía
- **Taxonomía de selección múltiple**: árbol de taxonomías donde se pueden seleccionar uno o varios valores
- **Text**: cuadro de texto normal
- **Text Area**: atributo para introducir un texto largo de hasta 300 mil caracteres
- **URL**: texto considerado como una URL para que el usuario pueda clicar sobre el atributo y se abra una nueva pestaña con esa URL
- **User**: lista completa de usuarios de Anjana Data

### Estructura de la tabla

Cada atributo registrado se caracteriza por los siguientes elementos:

- **id\_attribute\_definition**: identificador único de la tabla.
- **name**: nombre interno del atributo.
- **attribute\_type**: indica el tipo de atributo. A continuación se muestran las tablas de equivalencias entre los tipos de atributos soportados y el valor en el campo Type.
- **description**: descripción detallada del atributo y su contenido. Este campo puede no completarse.
- **label**: etiqueta del atributo. Este campo puede no completarse.
- **label\_translation\_key**: clave para la traducción de la etiqueta del atributo en el idioma configurado para el usuario. Este texto debe coincidir con la clave de la traducción en portuno.translations.
- **place\_holder**: texto de relleno del atributo si no tiene valor informado. Este campo puede no completarse.
- **place\_holder\_translation\_key**: clave para la traducción del texto de relleno si no tiene valor. Este campo puede no completarse.
- **short\_description\_translation\_key**: clave para la traducción de la descripción. Este campo permite definir la descripción que saldrá en el formulario en el icono de ⓘ.
- **start\_date**: fecha de creación del atributo (sólo informativo).
- **last\_modified\_date**: fecha de última modificación del atributo (sólo informativo).
- **module**: módulo al que pertenece, a elegir entre “BG” para el Business Glossary o “DC” para el Data Catalog, (solo informativo, no limita las plantillas en las que se puede incluir).

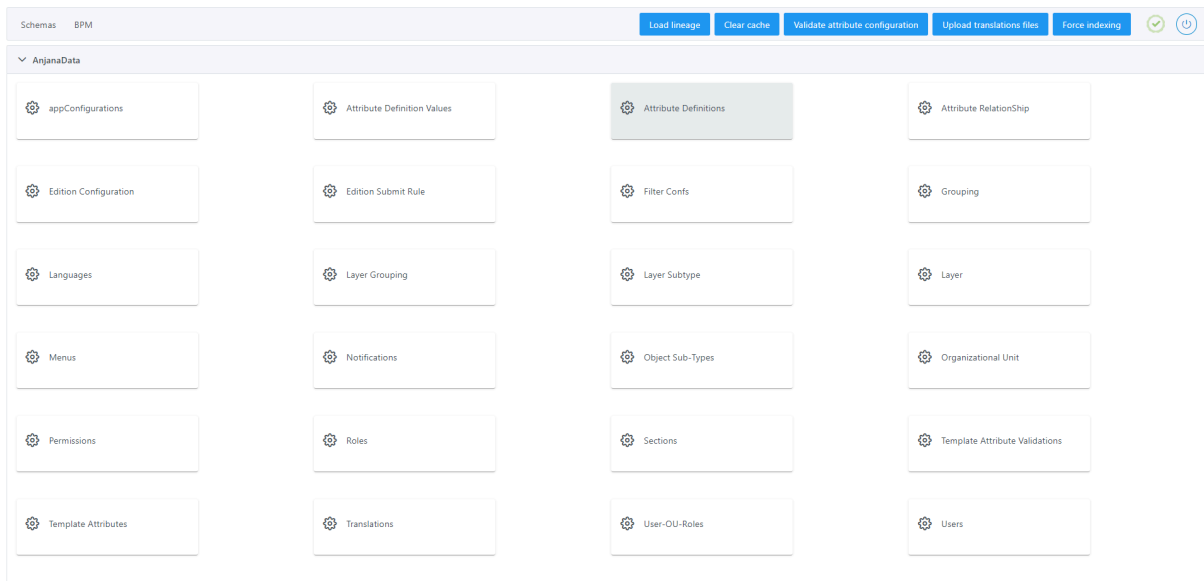
A continuación se presenta la equivalencia entre los tipos de atributos y los valores a rellenar en el campo attribute\_type de la tabla:

Tipo de campo	Valor campo attribute_type
Array de Boolean	ARRAY_BOOLEAN
Array de Date	ARRAY_DATE
Array de Decimal	ARRAY_DECIMAL
Array de Entity	ARRAY_ENTITY
Array de File	ARRAY_UPLOAD_FILE
Array de Number	ARRAY_NUMBER
Array de Organizational Unit	MULTI_ORGANIZATIONAL_UNIT
Array de Text	ARRAY_ALPHANUMERICAL
Array de URL	ARRAY_UPLOAD_URL
Array de Users	MULTI_USERS
Boolean	INPUT_CHECKBOX
Date	INPUT_DATE
Decimal	INPUT_DECIMAL
Enriched Text Area	ENRICHED_TEXT_AREA
Entity Container	ENTITY_CONTAINER
Entity Search	ENTITY_SEARCH
File	UPLOAD_FILE
International Text	INPUT_TEXT_INTERNATIONAL
International Text Editor	ENRICHED_TEXT_AREA_INTERNATIONAL
International Textarea	TEXT_AREA_INTERNATIONAL
MultiSelect	MULTI_SELECT
MultiSelect con iconos	MULTI_SELECT_IMG
MultiSelect con iconos y texto	MULTI_SELECT_IMG_TXT
Number	INPUT_NUMBER
Number range	INPUT_RANGE
Organizational Unit	SELECT_ORGANIZATIONAL_UNIT
Reference Metadata	SELECT

Selector con icono	SELECT_IMG
Selector con icono y texto	SELECT_IMG_TXT
Taxonomía unica	TREE_SELECT
Taxonomía de selección múltiple	TREE_MULTISELECT
Text	INPUT_TEXT
Text Area	TEXT_AREA
URL	UPLOAD_URL
User	SELECT_USERS

## Visión de Administrador

El alta de un nuevo atributo en el panel de administración de Anjana Data se realiza en la tabla **Attribute Definitions**.



Al acceder se muestra una tabla que contiene todos los atributos existentes en la configuración actual.

La creación de un nuevo atributo se realiza mediante el botón **New**:

[+ New](#)

Attribute Definitions

Id	Name	Type	Description	Label	Label Translation Key	Place Holder	Place Holder Translation Key	Short Description Translation Key	startDate	Last Modified Date	Module
-	-	Igual	-	-	-	-	-	-	-	-	Igual
1	name	INPUT_TEXT	Name of object	Name	ATTRIBUTE_NOMB	Name here	ATTRIBUTE_PLACE		16/03/2021		BG
2	termType	SELECT	Term type	Term type	ATTRIBUTE_NOMB	Term type here	ATTRIBUTE_PLACE		16/03/2021		BG
3	domain	SELECT	Domain	Domain	ATTRIBUTE_NOMB	Domain here	ATTRIBUTE_PLACE		16/03/2021		BG
4	subdomain	SELECT	Subdomain	Subdomain	ATTRIBUTE_NOMB	Subdomain here	ATTRIBUTE_PLACE		16/03/2021		BG
5	geography	SELECT	Geography	Geography	ATTRIBUTE_NOMB	Geography here	ATTRIBUTE_PLACE		16/03/2021		BG
6	des	INPUT_TEXT	Description of object	Description	ATTRIBUTE_NOMB	Description here	ATTRIBUTE_PLACE		07/10/2020		BG
-	-	-	Short	Short	-	Short	-	-	-	-	-

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos y mediante el combo de Module se selecciona el módulo para el cual se define el atributo.

A continuación, se muestra cómo crear un atributo llamado **DocMgmtTool** de tipo metadatos de referencia que se utilizará en la plantilla del objeto DOCUMENT creado con anterioridad:

### Attribute Definitions Edit

Name

Type

Label

Label Translation Key

Place Holder

Place Holder Translation Key

Short Description Translation Key

Module

Description

## Visión de Desarrollador

Para definir los atributos hay que configurar la tabla **attribute\_definition** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.attribute_definition (id_attribute_definition,description, label, label_translation_key,last_modified_date,module,name,place_holder,place_holder_translation_key,short_description_translation_key,start_date,attribute_type) VALUES
```

```
(1,'Name of object','Name','ATTRIBUTE_NOMBRE.name',NULL,'BG','name','Name here','ATTRIBUTE_PLACEHOLDER.name',NULL,'2021-03-16 16:36:30.875','INPUT_TEXT'),  
(2,'Term type','Term type','ATTRIBUTE_NOMBRE.termType',NULL,'BG','termType','Term type here','ATTRIBUTE_PLACEHOLDER.termType',NULL,'2021-03-16 16:36:30.875','SELECT'),  
(3,'Domain','Domain','ATTRIBUTE_NOMBRE.domain',NULL,'BG','domain','Domain here','ATTRIBUTE_PLACEHOLDER.domain',NULL,'2021-03-16 16:36:30.875','SELECT'),  
(4,'Subdomain','Subdomain','ATTRIBUTE_NOMBRE.subdomain',NULL,'BG','subdomain','Subdomain here','ATTRIBUTE_PLACEHOLDER.subdomain',NULL,'2021-03-16 16:36:30.875','SELECT'),  
(5,'Geography','Geography','ATTRIBUTE_NOMBRE.geography',NULL,'BG','geography','Geography here','ATTRIBUTE_PLACEHOLDER.geography',NULL,'2021-03-16 16:36:30.875','SELECT');
```

### NOTAS:

- El campo name name de attribute\_definition deben no contener espacios, '(' , ')' ni '/'.
- Los label\_translation\_key y los place\_holder\_translation\_key deben definirse en la tabla anjana.translations incluyendo un registro para cada idioma configurado en la aplicación.
- Es recomendable no cambiar el tipo de ningún atributo una vez que se hayan creado objetos que lo contengan en su metadato. Cambiar el tipo puede ocasionar errores en la indexación con Solr si los tipos original y nuevo son incompatibles (por ej: si un atributo se declara numérico y, posteriormente, se cambia el atributo a alfanumérico ya que Solr tendrá indexado el atributo de tipo numérico y será imposible automatizar el cambio de campo). En caso de ser necesario, leer el procedimiento a seguir en las [Preguntas Frecuentes](#).
- Es necesario crear el atributo organizationalUnit para que, en la importación por excel, se incluya como columna a editar para el objeto a dar de alta. Este atributo sólo sirve para conocer la definición de él, no es necesario asociarlo a ninguna plantilla de objeto.
- Para los atributos con tipo SELECT, SELECT\_IMG, SELECT\_IMG\_TXT, TREE\_SELECT, MULTI\_SELECT, MULTI\_SELECT\_IMG, MULTI\_SELECT\_IMG\_TXT o TREE\_MULTISELECT se deben definir los posibles valores del combo de selección en la tabla attribute\_definition\_value (ver apartado [Definir y configurar los posibles valores de los atributos](#)).
- Los atributos de tipo INPUT\_RANGE necesitan tener validación de mínimo y máximo para poder identificar el rango de valores a elegir. En caso de no configurar estas validaciones, por defecto el valor mínimo será MIN\_RANGE y el máximo MAX\_RANGE, configurados en la tabla de Portuno app\_configuration (ver apartado [Variables del sistema](#) de Esquema Portuno de BD)
- Los atributos de tipo ENRICHED\_TEXT\_AREA o ENRICHED\_TEXT\_AREA\_INTERNATIONAL admiten subida de ficheros con extensión gif, png o jpg.

### Atributos de clave primaria

A continuación se detalla una lista de atributos que forman parte de la clave primaria (*primary key*) de cada uno de los objetos del Catálogo de Datos o del Glosario de Negocio. La funcionalidad y el tipado de alguno de estos datos se especificará en el siguiente punto (atributos obligatorios):

- I. Para dataset: *name, infrastructure, path, technology* y *zone*.
- II. Para dataset field: *name, infrastructure, path, technology* y *zone*.
- III. Para DSA: *name*.
- IV. Para proceso: *name, infrastructure, path, technology* y *zone*.
- V. Para instancia de proceso: *name, processAri* y *solutionAri*.
- VI. Para solución: *name*.
- VII. Para otras entidades no nativas predefinidas por cliente: *name*.
- VIII. Para las relaciones: *name, source* y *destination*.

### Atributos Obligatorios

Debido a que algunos de los atributos se visualizan en el Portal de Datos o Anjana tiene cierta lógica interna sobre alguno de ellos, es obligatorio que, para esos atributos, el campo *name* de la tabla *attribute\_definition* tenga un nombre particular y así como el *attribute\_type* sea de un tipo particular. De esta forma, Anjana Data podrá identificar estos atributos y funcionar correctamente.

Atributos comunes a todas las entidades:

Valor en campo <i>name</i>	Valor en columna <i>attribute_type</i>	Finalidad del atributo
<i>description</i>	TEXT_AREA	Permite introducir una descripción extensa del objeto
<i>name</i>	INPUT_TEXT	Almacena el nombre lógico de la entidad

Atributos comunes a todas las entidades nativas:

Valor en campo <i>name</i>	Valor en columna <i>attribute_type</i>	Finalidad del atributo
<i>expirationDate</i>	INPUT_DATE	Almacena la fecha en la que el objeto expira

Atributos comunes a los datasets y a todas las entidades no nativas que vayan a ser gobernadas:

Valor en campo <i>name</i>	Valor en columna <i>attribute_type</i>	Finalidad del atributo
<i>infrastructure</i>	SELECT	Indica del entorno donde se encuentra localizado el objeto, usado con <i>technology</i> y <i>zone</i> para identificar un plugin en caso de gobierno activo.

isGoverned	INPUT_CHECKBOX	Indica si hay gobierno activo sobre el objeto
path	INPUT_TEXT	Indica la localización del objeto
technology	SELECT	Indica la tecnología en la que está depositado el objeto, usado con infrastructure y zone para identificar un plugin en caso de gobierno activo.
zone	SELECT	Indica la zona donde se encuentra el objeto, usado con infrastructure y technology para identificar un plugin en caso de gobierno activo.

Atributos para dataset:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
data_format	SELECT	Indica el formato de los datos del dataset
physicalName	INPUT_TEXT	Indica el nombre físico del dataset de cara al gobierno activo de éste
pi	INPUT_CHECKBOX	Indica si el dataset contiene información personal
sampleData	INPUT_CHECKBOX	Indica si se puede ver una muestra de los datos del dataset en Anjana
datasetFields	INPUT_TEXT	Indica si los cambios en los dataset_fields afectan al propio dataset. Este atributo debe configurarse oculto en la plantilla (template_attribute) ya que sólo es necesario para poder versionar datasets o no lanzar workflows en caso de modificación de los dataset_fields de un dataset.

Atributos para dataset\_field:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
fieldDataType	INPUT_TEXT	Permite seleccionar el tipo del dato que contiene el field
physicalName	INPUT_TEXT	Permite indicar el nombre físico del field de cara al gobierno activo de éste
pi	INPUT_CHECKBOX	Permite indicar si el field contiene

		información personal
position	INPUT_NUMBER	Permite indicar el orden del field en un dataset

Atributos para DSA:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
data_sensitivity	SELECT	Permite indicar la sensibilidad de dato
pi	INPUT_CHECKBOX	Permite indicar si el DSA contiene información personal
termCondFile	UPLOAD_FILE o UPLOAD_URL	Permite incluir el fichero o la ruta para acceder a los términos de licencia de un DSA
dsaContent	ENTITY_CONTAINER	Almacena el conjunto de entidades a las que se da acceso a los usuarios por medio del DSA

Atributos para proceso:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
isEngine	INPUT_CHECKBOX	Permite indicar si el proceso es motor y por tanto puede tener múltiples instancias o si no lo es, pudiendo crear sólo una
infrastructure	SELECT	Permite seleccionar el entorno donde se encuentra localizado el proceso
technology	SELECT	Permite seleccionar la tecnología que almacena el proceso
zone	SELECT	Permite seleccionar la zona donde se encuentra el proceso
path	INPUT_CHECKBOX	Permite indicar la localización del proceso

Atributos para instancia de proceso:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
---------------------	---------------------------------	------------------------



exec_user	INPUT_TEXT	Permite indicar el usuario de ejecución de la instancia
processAri	INPUT_TEXT	Permite indicar el proceso de la instancia. Este atributo no debe añadirse a la plantilla (template_attribute) ya que sólo es necesario para poder importar instancias por medio de fichero excel.
solutionAri	INPUT_TEXT	Permite indicar la solución propietaria de la instancia. Este atributo no debe añadirse a la plantilla (template_attribute) ya que sólo es necesario para poder importar instancias por medio de fichero excel.
instanceInDataset	ENTITY_CONTAINER	Almacena el conjunto de datasets que lee la instancia para su procesamiento
instanceOutDataset	ENTITY_CONTAINER	Almacena el conjunto de datasets que escribe la instancia tras su procesamiento

Atributos para solución:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
solutionRelatedInstance	ENTITY_CONTAINER	Almacena el conjunto de instancias relacionadas en la solución

Para relación:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
source	ENTITY_SEARCH	Permite indicar el origen de una relación o uno de sus extremos
destination	ENTITY_SEARCH	Permite indicar el destino de una relación o uno de sus extremos

Para poder completar la información de las **adherencias**, es necesario definir plantilla, menú, sección y atributos para ella, aunque no sea accesible esta información por medio de un formulario como ocurre con cualquier otra entidad en Anjana:

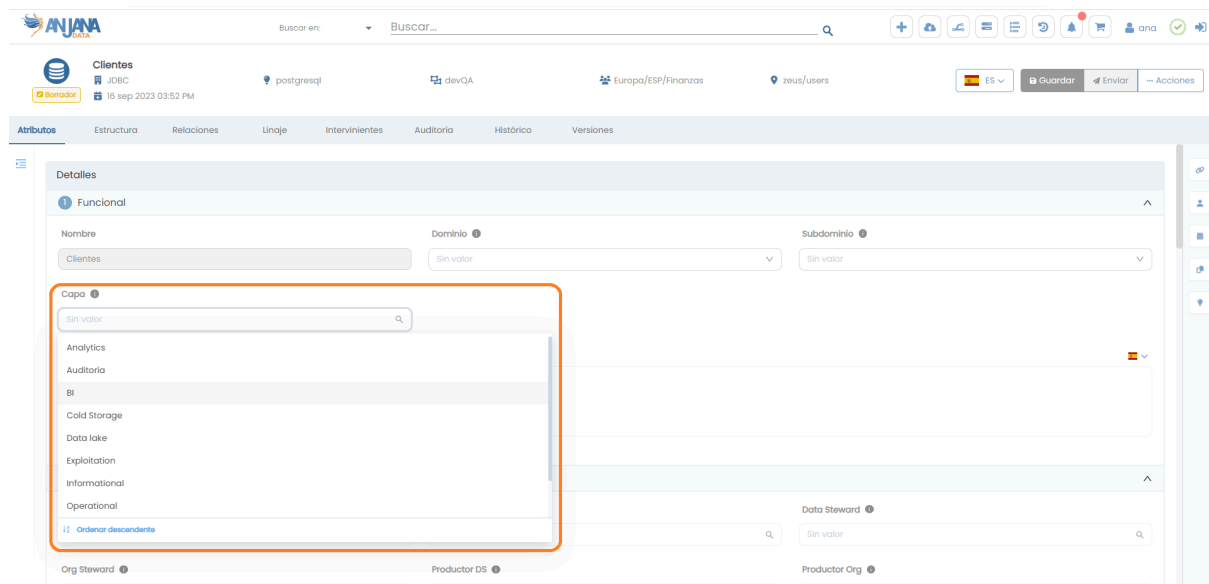
Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
expirationDate	INPUT_DATE	Permite almacenar la fecha en la que la adherencia finaliza

pae	INPUT_TEXT	Permite asociar internamente la adherencia con la solicitud enviada por el usuario
requestReason	INPUT_TEXT	Permite almacenar el motivo de la solicitud de adherencia

## 5. Definir y configurar los posibles valores de los atributos

Para aquellos atributos de tipo selección de valores (atributos con attribute\_type SELECT, SELECT\_IMG, SELECT\_IMG\_TXT, TREE\_SELECT, MULTI\_SELECT, MULTI\_SELECT\_IMG, MULTI\_SELECT\_IMG\_TXT o TREE\_MULTISELECT) hay que definir los metadatos de referencia de entre los cuales el usuario puede seleccionar el o los que convengan.

Esta configuración se lleva a cabo en la tabla **attribute\_definition\_value**.



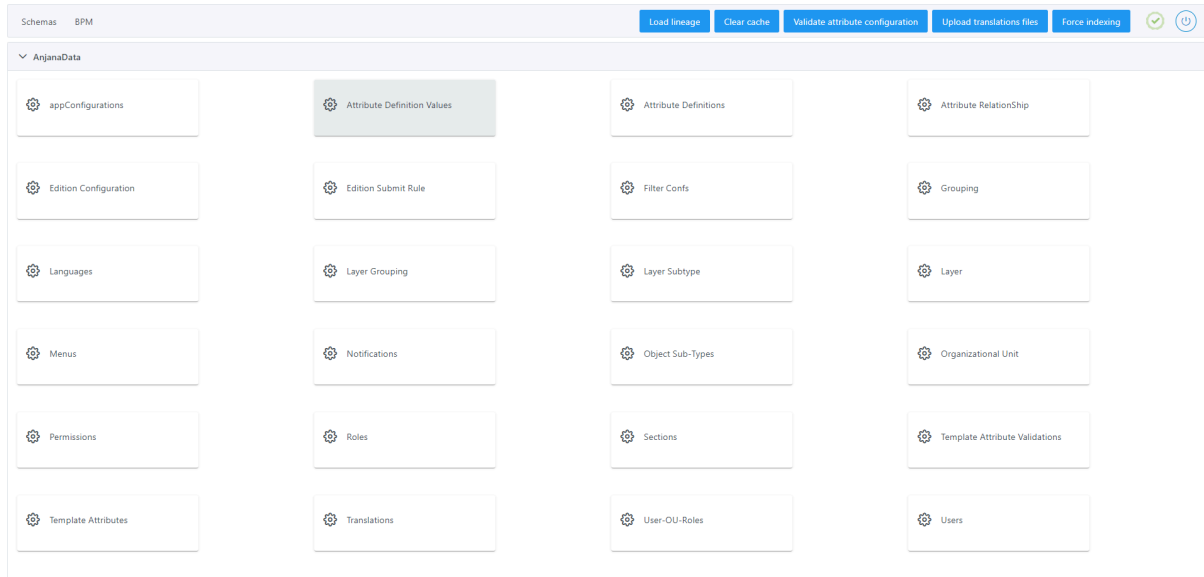
### Estructura de la tabla

Cada valor registrado se caracteriza por los siguientes elementos:

- **id\_attribute\_definition\_value:** identificador único de la tabla.
- **value:** valor del atributo.  
En caso de querer visualizarlo en los distintos idiomas de la aplicación debe usarse el value como clave de traducción en la tabla de portuno de translations.  
Si el atributo es de imagen (SELECT\_IMG, SELECT\_IMG\_TXT, MULTI\_SELECT\_IMG o MULTI\_SELECT\_IMG\_TXT) el icono a utilizar debe estar localizado en Minio y se llamará <nombreAtributo>\_<valor>.svg (siendo valor el value de attribute\_definition\_value). Más detalle en el documento 'Anjana Data - CONF - Iconografía y CSS'.
- **label\_select:** descripción del valor.
- **id\_attibute\_definition:** Atributo de metadatos para el cual se despliega el combo de selección.

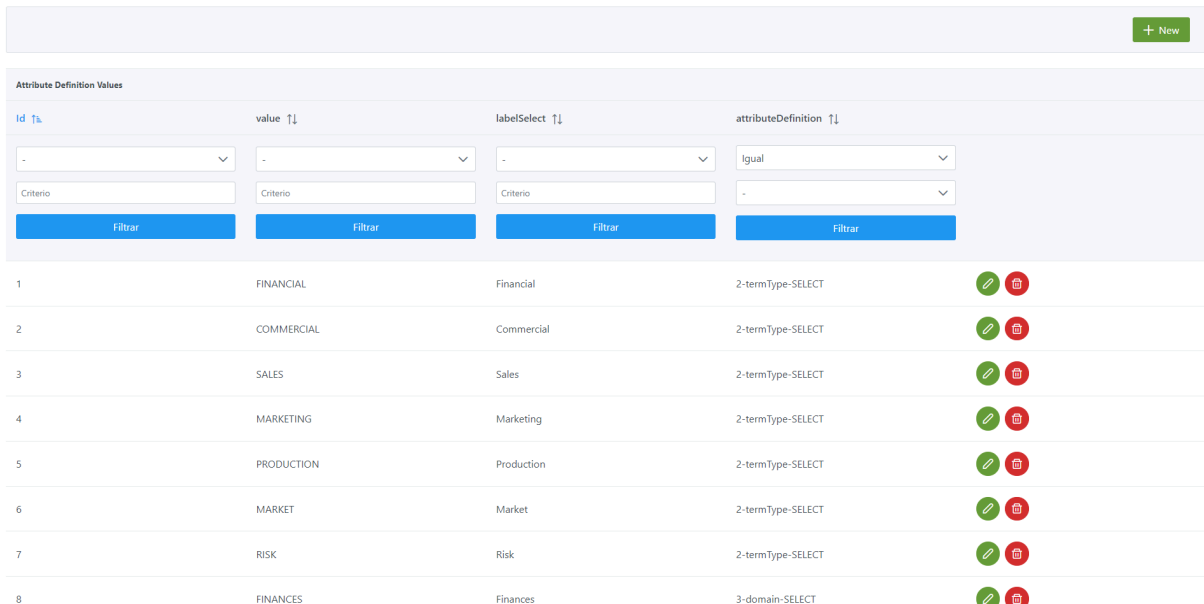
## Visión de Administrador

El alta de un nuevo atributo en el panel de administración de Anjana Data se realiza en **Attribute Definition Values**:



















Al acceder se muestra una tabla que contiene todos los atributos existentes en la configuración actual.

La creación de un nuevo metadato de referencia se realiza mediante el botón **New**:

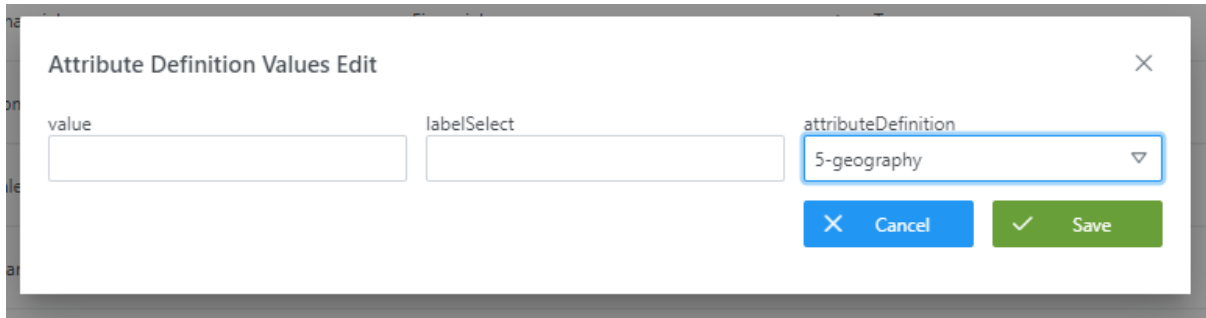


The screenshot shows the 'Attribute Definition Values' table. At the top right, there is a green '+ New' button. The table has four columns: 'id', 'value', 'labelSelect', and 'attributeDefinition'. Each column has a dropdown menu and a 'Filtrar' button below it. The table contains 8 rows of data, each with a green edit icon and a red delete icon at the end.

id	value	labelSelect	attributeDefinition	
1	FINANCIAL	Financial	2-termType-SELECT	 
2	COMMERCIAL	Commercial	2-termType-SELECT	 
3	SALES	Sales	2-termType-SELECT	 
4	MARKETING	Marketing	2-termType-SELECT	 
5	PRODUCTION	Production	2-termType-SELECT	 
6	MARKET	Market	2-termType-SELECT	 
7	RISK	Risk	2-termType-SELECT	 
8	FINANCES	Finances	3-domain-SELECT	 

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos.

A continuación, se muestra cómo crear atributos de referencia para un atributo:



### Visión de Desarrollador

Para definir los metadatos de referencia hay que configurar la tabla **attribute\_definition\_value** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

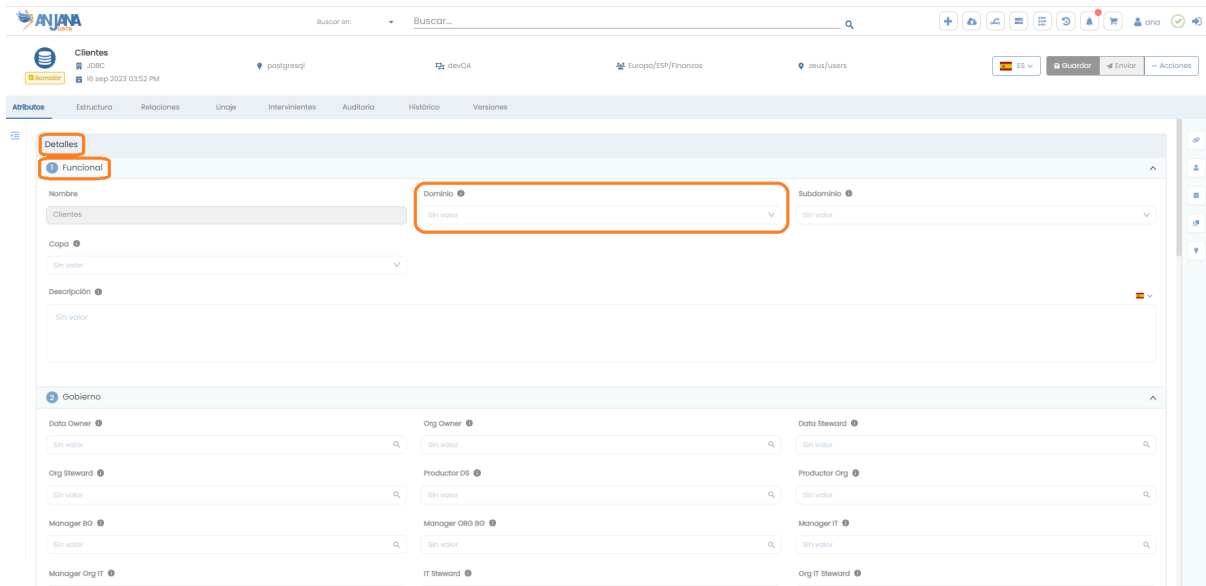
```
INSERT INTO anjana.attribute_definition_value
(id_attribute_definition_value, id_attribute_definition, label_select, value) VALUES
(1,2,'Financial','0')
,(2,2,'Commercial','1')
,(3,2,'Sales','2')
,(4,2,'Marketing','3')
,(5,2,'Production','4')
,(6,2,'Marketing','5')
,(7,2,'Risk','6')
,(8,3,'Finances','1')
,(9,3,'Marketing','2')
,(10,3,'Operations','3')
,(11,3,'Human Resources','4')
,(12,3,'Legal','5');
```

#### NOTAS:

- En el caso de los atributos infrastructure, technology y zone se debe usar como value (y, por tanto, como clave en portuno.translations) exactamente el mismo nombre que se usa para esos campos en las tripletas para los plugins de Tot.
- No deben existir dos valores iguales para el mismo atributo en la tabla attribute\_definition\_value.

## 6. Definir qué atributos aparecerán en cada una de las secciones de los formularios según el tipo de objeto

Configurar las plantillas de metadatos supone definir en qué menú y sección aparecerán los metadatos de entre los definidos en la tabla attribute\_definition. Esta asociación de atributos a plantillas se configura en la tabla **template\_attribute**.



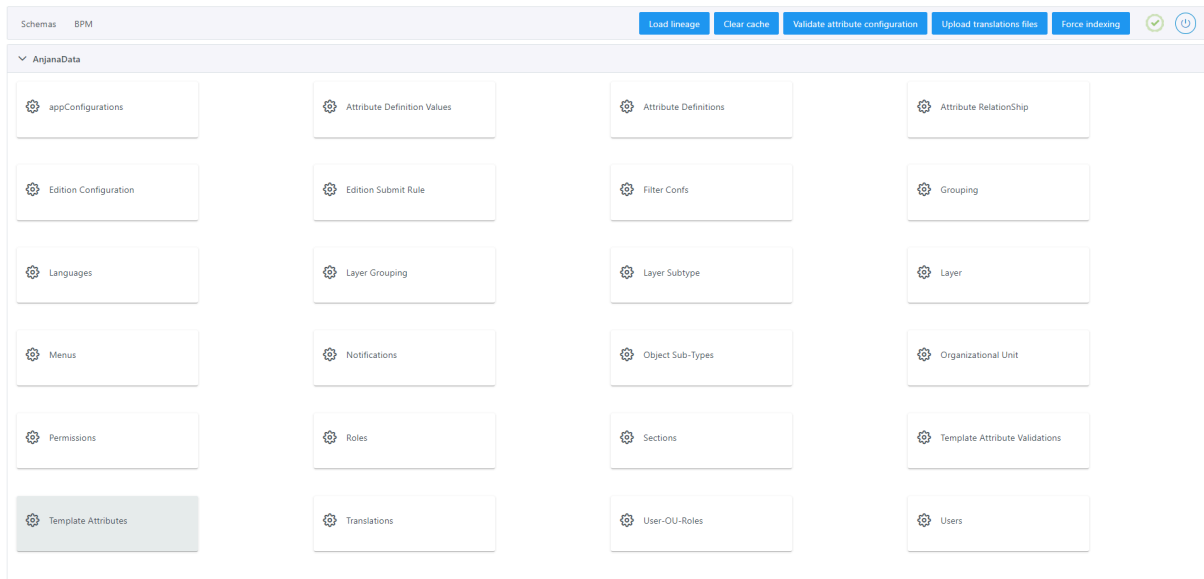
## Estructura de la tabla

Cada atributo de plantilla registrado se caracteriza por los siguientes elementos:

- **id\_template\_attribute:** identificador único de la tabla.
- **active:** flag que indica si el atributo está activo o no en el formulario.  
En caso de querer eliminar un atributo de una plantilla, éste debe tener active=false para no estar disponible en edición o visualización. Los objetos que tuvieran anteriormente este atributo informado siguen manteniéndolo pero no es visible en el formulario así que no puede modificarse. Si más adelante se desea volver a incluir el atributo en la plantilla, active=true permitirá que vuelva a estar disponible y se podrá ver y editar el valor del atributo en todos los objetos.  
Si el atributo pertenece a la plantilla de adherencia, se debe rellenar con “false” debido a que no existe una pantalla específica del objeto adherencia.
- **is\_intern:** boolean que por defecto adquiere el valor “true”. No se hace uso de este campo.
- **is\_visible:** flag que indica si un atributo de los dataset\_fields es visible en la tabla Structure de la pantalla de su dataset.  
En caso de tratarse de un atributo perteneciente a la plantilla de adherencia, se debe rellenar con “false” puesto que no existe una pantalla específica del objeto adherencia.  
Si el atributo pertenece a una plantilla distinta de la de dataset\_field y adherencia se debe rellenar con “true”.
- **sort:** orden en el que aparece en el atributo en la sección donde queda asignado.
- **id\_attribute\_definition:** atributo que se desea introducir en la plantilla.
- **id\_section:** sección en la que se desea introducir el atributo.

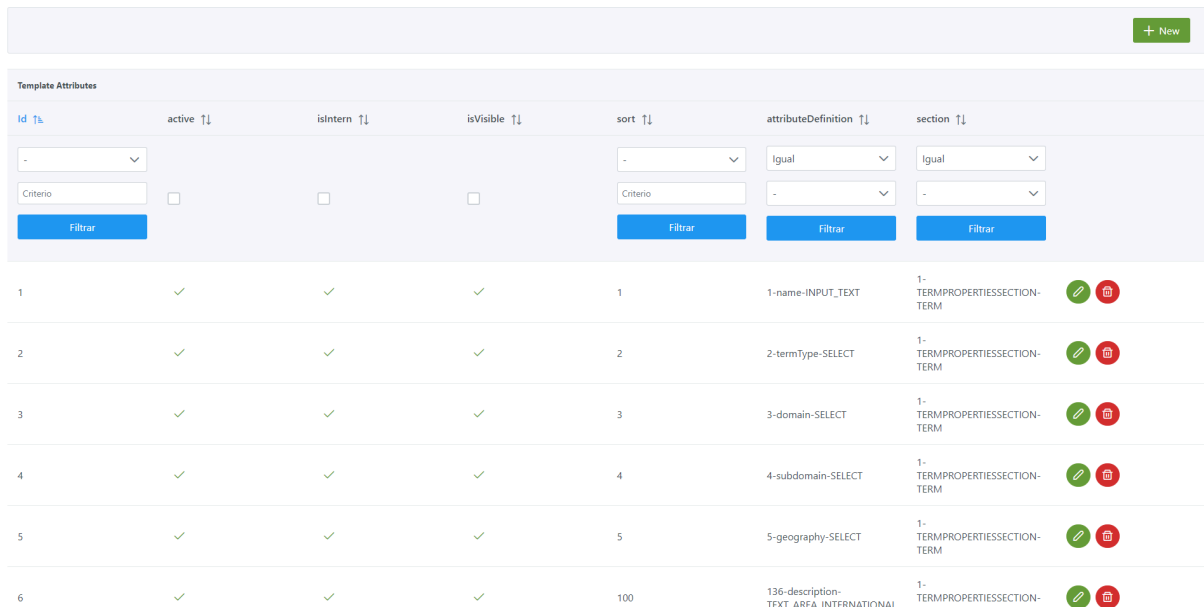
## Visión de Administrador

El alta de los atributos dentro de una plantilla se realiza en el panel de administración de Anjana Data en **Template Attributes**:



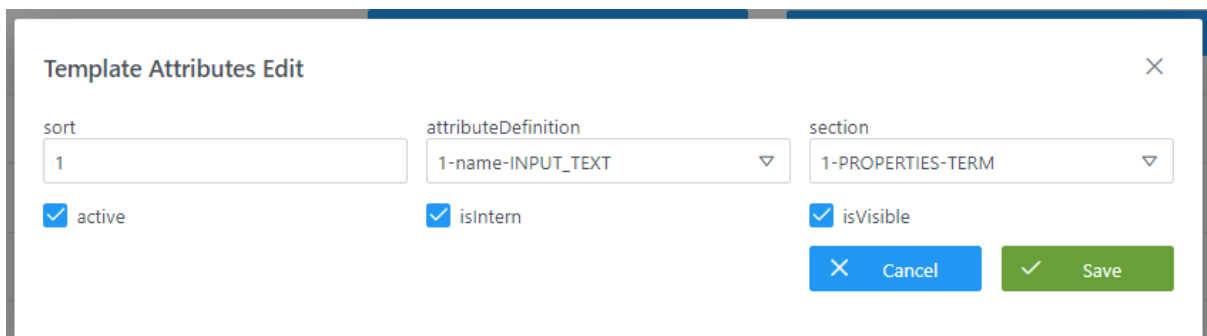
Al acceder se muestra una tabla que contiene todos los atributos de plantilla existentes en la configuración actual.

La creación de un nuevo registro en esta tabla se realiza mediante el botón **New**:



Id	active	isIntern	isVisible	sort	attributeDefinition	section
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1-name-INPUT_TEXT	1-TERMPROPERTIESSECTION-TERM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	2-termType-SELECT	1-TERMPROPERTIESSECTION-TERM
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	3-domain-SELECT	1-TERMPROPERTIESSECTION-TERM
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	4-subdomain-SELECT	1-TERMPROPERTIESSECTION-TERM
5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	5-geography-SELECT	1-TERMPROPERTIESSECTION-TERM
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100	136-description-TEXT AREA INTERNATIONAL	1-TERMPROPERTIESSECTION-TERM

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:



### Template Attributes Edit

active
  isIntern
  isVisible

## Visión de Desarrollador

Para definir los atributos dentro de un template hay que configurar la tabla **template\_attribute** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

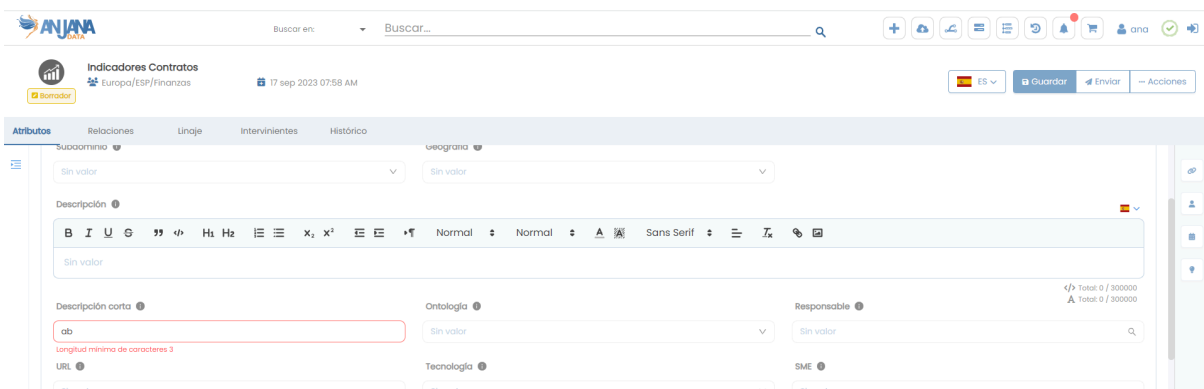
```
INSERT INTO anjana.template_attribute
(id_template_attribute, active, is_intern, is_visible, sort, id_attribute_definition, id_section)
VALUES
(1, true, true, true, 1, 1, 1),
(2, true, true, true, 2, 2, 1),
(3, true, true, true, 3, 3, 1),
(4, true, true, true, 4, 4, 1),
(5, true, true, true, 5, 5, 1),
(6, true, true, true, 6, 136, 1),
(7, true, true, true, 7, 7, 1);
```

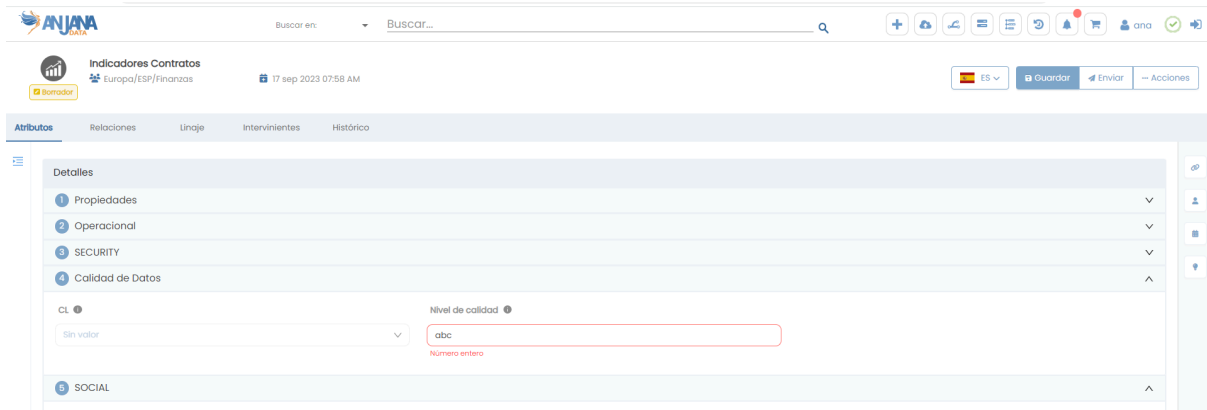
### NOTAS:

- Es importante que los atributos marcados como [obligatorios](#) se incluyan en las plantillas de los subtipos de objetos a los que aplican.
- Aunque ADHERENCE no tiene vista propia, es necesario añadir registros en template\_attribute para esta relación para poder añadir los atributos necesarios que debe contener. Como se especifica anteriormente, los campos active e is\_visible de estos atributos deben marcarse a “false”.

## 7. Definir validaciones para los valores de entrada de cada atributo

Existe la posibilidad de definir un conjunto de validaciones para asegurar la correcta entrada de los atributos de metadatos. Estas validaciones se configuran en la tabla **template\_attribute\_validation**.





Indicadores Contratos  
Europa/ESP/Finanzas  
17 sep 2023 07:58 AM

ES Guardar Enviar Acciones

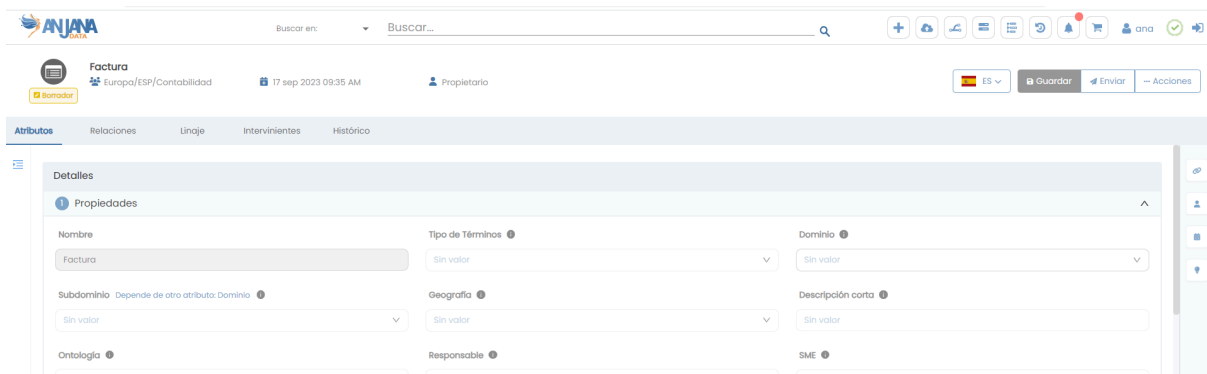
Atributos Relaciones Linaje Intervinientes Histórico

Detalles

- Propiedades
- Operacional
- SECURITY
- Calidad de Datos

cl  Nivel de calidad   
Numero entero

SOCIAL



Factura  
Europa/ESP/Contabilidad  
17 sep 2023 09:35 AM  
Propietario

ES Guardar Enviar Acciones

Atributos Relaciones Linaje Intervinientes Histórico

Detalles

- Propiedades

Nombre	Tipo de Términos	Dominio
Factura	<input type="text" value="Sin valor"/>	<input type="text" value="Sin valor"/>
Subdominio <small>Depende de otro atributo: Dominio</small>	Geografía	Descripción corta
<input type="text" value="Sin valor"/>	<input type="text" value="Sin valor"/>	<input type="text" value="Sin valor"/>
Ontología	Responsable	SME
<input type="text" value="Sin valor"/>	<input type="text" value="Sin valor"/>	<input type="text" value="Sin valor"/>

Estas validaciones pueden ser:

- **Atributos obligatorios (REQUIRED)**

Esta regla permite identificar los atributos que obligatoriamente deben ser informados en una plantilla.

- **Atributos no modificables (NOT EDITABLE)**

Esta regla permite identificar los atributos que no pueden ser editados.

- **Longitud máxima y mínima para valores alfanuméricos (MIN/MAX LENGTH)**

Esta regla permite establecer unas dimensiones mínimas y máximas de contenido en los campos de texto.

En el caso de los atributos de texto enriquecido las etiquetas html generadas computan para esta validación.

- **Mínimo y Máximo para los valores en los campos enteros, decimales, arrays de los mismos y rangos de valores (MIN/MAX)**

Esta regla permite definir un rango de valores, con mínimo y máximo respectivamente, para atributos numéricos.

- **Dependencia entre valores de referencia (DEPENDS ON)**



Esta regla permite hacer que el conjunto de valores válido para un atributo de tipo selector de valores (select, select con imágenes, select con imágenes y texto o los array de estos tipos) dependa del valor escogido en otro atributo de este mismo tipo

- **Heredable (HERITABLE)**

Esta regla permite identificar un atributo booleano no editable y cuyo valor se calcula en función de los valores que adquiriera ese mismo atributo en la plantilla de otros objetos.

La herencia sólo puede darse entre entidades que tengan un parentesco según el metamodelo nativo de Anjana, es decir, los siguientes casos::

- El dataset hereda de sus dataset\_fields
- El DSA hereda de las entidades incluidas en el atributo dsaContent
- La instancia hereda de sus datasets input o output
- **Longitud máxima y mínima de la parte decimal de campos y arrays de decimales (MAX/MIN DECIMAL PRECISION)**

Esta regla permite establecer un mínimo y máximo de dígitos para la parte decimal.

- **Longitud máxima y mínima de valores en el conjunto (MAX/MIN LENGTH ARRAY)**

Esta regla permite establecer el número mínimo y máximo de valores seleccionados en un conjunto.

- **Tipo de relación creada (RELATIONSHIP\_TYPE)**

Esta regla identifica la relación que se creará con los atributos de tipo ENTITY\_CONTAINER entre la entidad que tiene en su plantilla el atributo y aquellas entidades con las que se completa.

- Para el atributo dsaContent la relación debe ser DSA\_CONTENT.
- Para el atributo instanceInDataset la relación debe ser INSTANCE\_DATASET\_IN.
- Para el atributo instanceOutDataset la relación debe ser INSTANCE\_DATASET\_OUT.
- Para el atributo solutionRelatedInstance la relación debe ser SOLUTION\_RELATED\_INSTANCE.
- **Restricción de subtipos en atributos de tipo entidad o array de entidades (SUBTYPE\_FILTER)**

Esta regla permite limitar el subtipo de las entidades de los extremos de las relaciones (es decir, los subtipos que pueden incluirse en los atributos source y destination). Puede pasarse una lista separada por ‘\_’ (y sin espacios) para limitar a varios tipos. Si no se configura ninguna lista, los extremos podrán contener cualquier subtipo de Entidad.

### Validaciones para cada tipo de atributo:

Tipo de atributo numérico	Validaciones							
	MIN	MAX	MAX DECIMAL PRECISION	MIN DECIMAL PRECISION	MAX LENGTH ARRAY	MIN LENGTH ARRAY	REQUIRED	NOT EDITABLE
ARRAY DECIMAL	X	X	X	X	X	X	X	X
ARRAY NUMBER	X	X			X	X	X	X

INPUT DECIMAL	X	X	X	X			X	X
INPUT NUMBER	X	X					X	X
INPUT RANGE	X	X					X	X

Tipo de atributo alfanumérico	Validaciones							
	MAX LENGTH	MIN LENGTH	MAX ARRAY	LENGTH	MIN ARRAY	LENGTH	REQUIRED	NOT EDITABLE
ARRAY ALPHANUMERICAL	X	X	X		X		X	X
ARRAY UPLOAD URL			X		X		X	X
ENRICHED TEXT AREA							X	X
INPUT TEXT INTERNATIONAL	X	X					X	X
ENRICHED TEXT AREA INTERNATIONAL	X	X					X	X
TEXTAREA INTERNATIONAL	X	X					X	X
INPUT TEXT	X	X					X	X
TEXT AREA	X	X					X	X
UPLOAD URL							X	X

Tipo de atributo de selección de valores	Validaciones						
	DEPENDS ON	MAX LENGTH ARRAY	MIN LENGTH ARRAY	REQUIRED	NOT EDITABLE	SUBTYPE FILTER	RELATIONSHIP TYPE
ARRAY ENTITY		X	X	X	X	X	
MULTI ORGANIZATIONAL UNIT		X	X	X	X		
MULTI USERS		X	X	X	X		
ENTITY CONTAINER		X	X	X	X	X	X
ENTITY SEARCH				X	X	X	
MULTI SELECT	X	X	X	X	X		
MULTI SELECT IMG	X	X	X	X	X		
MULTI SELECT IMG TXT	X	X	X	X	X		
SELECT ORGANIZATIONAL UNIT				X	X		

SELECT	X			X	X		
SELECT IMG	X			X	X		
SELECT IMG TXT	X			X	X		
TREE SELECT				X	X		
TREE MULTISELECT		X	X	X	X		
SELECT USERS				X	X		

Otros tipos de atributos	Validaciones						
	MAX DECIMAL PRECISION	MIN DECIMAL PRECISION	HERITABLE	MAX LENGTH ARRAY	MIN LENGTH ARRAY	REQUIRED	NOT EDITABLE
ARRAY BOOLEAN				X	X	X	X
ARRAY DATE				X	X	X	X
ARRAY UPLOAD FILE				X	X	X	X
INPUT CHECKBOX			X			X	X
INPUT DATE						X	X
UPLOAD FILE						X	X

### Estructura de la tabla

Cada validación registrada se caracteriza por los siguientes elementos:

- **id\_template\_attribute:** atributo de plantilla para el que se define la validación.
- **validator\_key:** clave de la validación:
  - DEPENDS\_ON: dependencia con otro atributo del template
  - HERITABLE: en el caso de que su valor se herede de otro metadato
  - MAX: número máximo
  - MAX\_DECIMAL\_PRECISION: número máximo de precisión para valores decimales.
  - MAX\_LENGTH: longitud máxima
  - MAX\_LENGTH\_ARRAY: cantidad máxima de elementos de array
  - MIN: número mínimo
  - MIN\_DECIMAL\_PRECISION: número mínimo de precisión para valores decimales.
  - MIN\_LENGTH: longitud mínima
  - MIN\_LENGTH\_ARRAY: cantidad mínima de elementos de array
  - NOT\_EDITABLE: no modificable
  - RELATIONSHIP\_TYPE: subtipo de relación que se crea
  - REQUIRED: atributo obligatorio
  - SUBTYPE\_FILTER: subtipo de entidad permitida
- **only\_on\_edition:** flag que indica si la validación solo se aplica en modo edición.

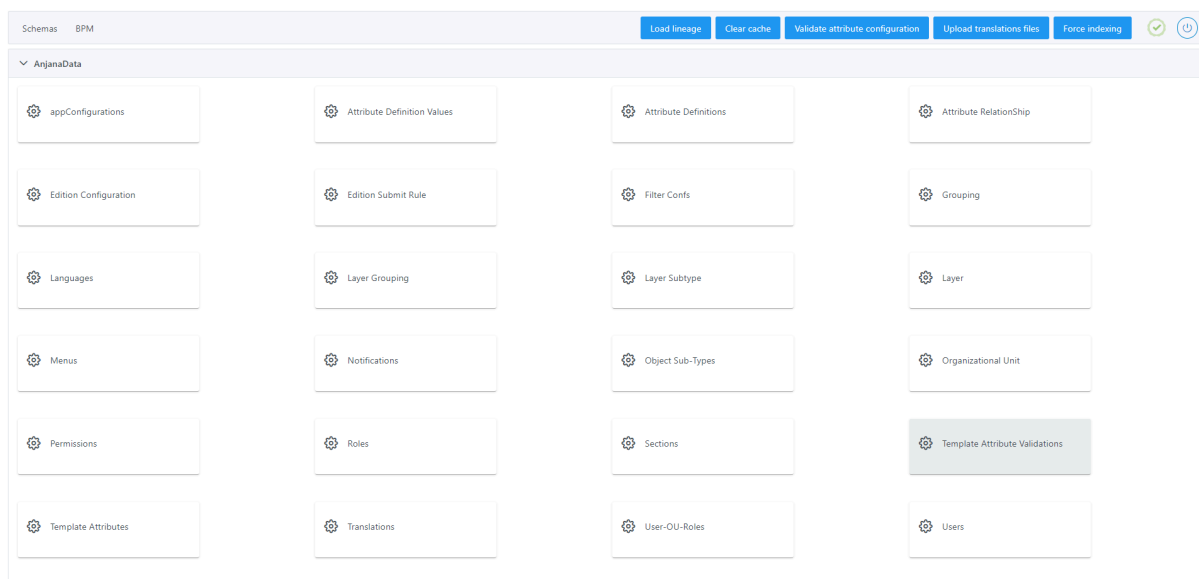
En el frontal de Anjana siempre se visualiza el formulario en modo edición por tanto aplica siempre.

Sin embargo, para importar objetos con excel es necesario que los atributos que vayan a ser editados y no formen parte de la PK del objeto, en caso de tener validaciones, tengan este flag a false. Los atributos que sí forman parte de la PK del objeto deben tener la validación de REQUIRED con este flag a false y la de NOT\_EDITABLE con el flag a true para que pueda ser informado en la creación desde excel.

- **validator\_params:** parámetros a completar en función de la clave:
  - En el caso de validaciones de tipo DEPENDS\_ON, indicar el nombre del atributo del que depende (name de la tabla attribute\_definition).
  - En el caso de valores máximos o mínimos, indicar el valor numérico.
  - En el caso de las validaciones REQUIRED y NOT\_EDITABLE indicar siempre *“true”*.
  - En el caso de HERITABLE el parámetro será:
    - *“HIGH”* en caso de que el valor heredado tenga que ser *“true”* si al menos uno de los elementos de los que se hereda tiene *“true”*
    - *“LOW”* en caso de que el valor heredado tenga que ser *“false”* si al menos uno de los elementos de los que se hereda tiene *“false”*
  - En el caso de validaciones de tipo RELATIONSHIP\_TYPE, indicar el nombre del subtipo de relación (name de la tabla object\_subtype)
  - En el caso de validaciones de tipo SUBTYPE\_FILTER, indicar los nombres de los subtipos de entidad (name de la tabla object\_subtype) separados por ‘\_’ (y sin espacios).

## Visión de Administrador

El alta de validaciones de este tipo dentro de un template se realiza en el panel de administración de Anjana Data en **Template Attribute Validations**:



Al acceder se muestra una tabla que contiene todas las validaciones existentes en la configuración actual.

La creación de una nueva validación se realiza mediante el botón **New**:

templateAttribute T1	key T1	onlyOnEdition T1	params T1
Igual	Igual	<input type="checkbox"/>	-
-	-		Criterio
<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>		<input type="button" value="Filtrar"/>

templateAttribute	key	onlyOnEdition	params	edit	delete
6-description-TERM	MAX_LENGTH	<input checked="" type="checkbox"/>	300000	<input type="checkbox"/>	<input type="checkbox"/>
6-description-TERM	MIN_LENGTH	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	<input type="checkbox"/>
7-sdescription-TERM	MAX_LENGTH	<input checked="" type="checkbox"/>	100	<input type="checkbox"/>	<input type="checkbox"/>
7-sdescription-TERM	MIN_LENGTH	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	<input type="checkbox"/>
34-sdescription-REPORT	MAX_LENGTH	<input checked="" type="checkbox"/>	100	<input type="checkbox"/>	<input type="checkbox"/>
34-sdescription-REPORT	MIN_LENGTH	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	<input type="checkbox"/>
60-sdescription-DATA_QUALITY_RULE	MAX_LENGTH	<input checked="" type="checkbox"/>	100	<input type="checkbox"/>	<input type="checkbox"/>
60-sdescription-DATA_QUALITY_RULE	MIN_LENGTH	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	<input type="checkbox"/>

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:

### Template Attribute Validations Edit

templateAttribute	key	params
6-description-TERM	MIN_LENGTH	3

onlyOnEdition

### Visión de Desarrollador

Para definir validaciones rápidas de atributos dentro de un template hay que configurar la tabla **template\_attribute\_validation** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.template_attribute_validation
(id_template_attribute, only_on_edition, validator_key, validator_params) VALUES
(1, false, 'REQUIRED', 'true'),
(1, false, 'MAX_LENGTH', '300000'),
(1, false, 'MIN_LENGTH', '1'),
(1, true, 'NOT_EDITABLE', 'true'),
(4, false, 'DEPENDS_ON', 'domain'),
(375, false, 'SUBTYPE_FILTER', 'TERM'),
(470, true, 'HERITABLE', 'HIGH');
```

### NOTAS:

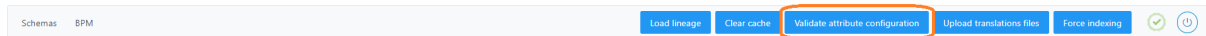
- Los atributos de tipo INPUT\_RANGE necesitan tener validación de mínimo y máximo para poder identificar el rango de valores a elegir. En caso de no configurarla, por defecto los

valores mínimo y máximo vendrán determinados por MIN\_INTEGER y MAX\_INTEGER de la tabla de app\_configurations.

- Los atributos que componen la PK de los objetos (name, source, destination, infrastructure, technology, zone...) deben tener la validación de obligatoriedad y no ser editables (REQUIRED y NOT\_EDITABLE) para que siempre estén completos.

La validación de obligatoriedad debe ser marcada con only\_on\_edition=false y, sin embargo, la validación de no edición debe tener only\_on\_edition=true.

En caso contrario, la validación de la configuración efectuada desde la acción específica de Portuno saldrá errónea o la descarga del excel con la plantilla del objeto fallará.



- La validación para indicar el tipo de relación (RELATIONSHIP\_TYPE) que se crea con cada atributo ENTITY\_CONTAINER de la aplicación es obligatoria y no debe ser modificada.

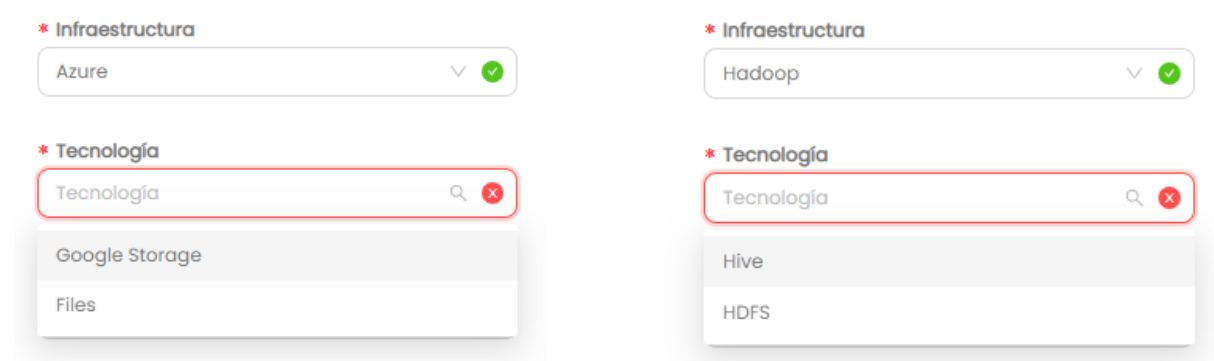
Anjana proporciona el script para la inserción automática con la actualización de versión. En caso de una instalación en vacío (sin configuración previa), tras la configuración de las plantillas de los objetos, se deben dar de alta las validaciones especificadas al inicio del [apartado](#).

- Para poder completar la dependencia entre los valores de los atributos de tipo SELECT es necesario que se configuren las relaciones entre ellos (ver en apartado [attribute relationships](#)).

## 8. Definir relaciones entre valores de atributos para pre-filtrar los reference metadata o definir las taxonomías

Es posible relacionar entre sí metadatos de referencia de forma que el valor que se seleccione en un combo filtre los resultados que se muestran en el siguiente combo como resultado de una lógica existente en el ecosistema de datos de su organización.

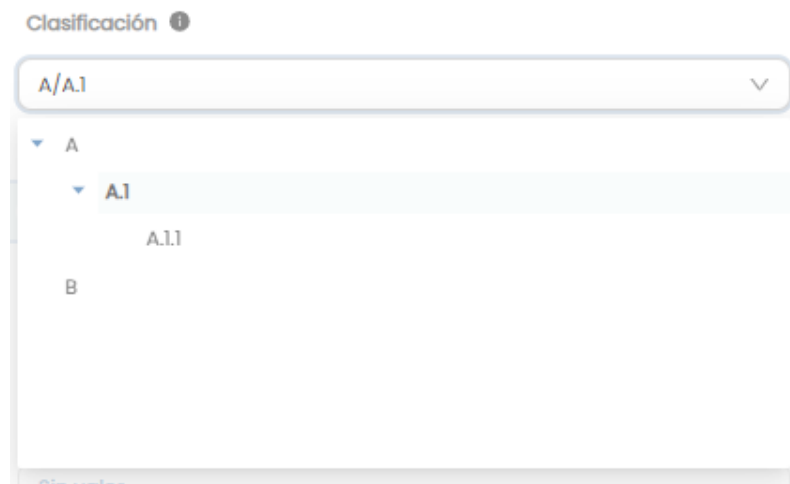
Esta configuración está disponible tanto para los atributos infraestructura, tecnología y zona elegidos en el wizard de creación de objetos como para el resto de atributos de las plantillas de tipo Metadatos de Referencia (o listados de valores predeterminados).



Por ejemplo, si se hablara de comunidades autónomas y ciudades, si se eligiera la CA de Madrid, en el combo de ciudades sólo se visualizarán aquellas ciudades propias de dicha comunidad y no las de toda España.

Para todos estos casos en los que los valores de un atributo dependen de la selección hecha en otro atributo, es necesario que se configure la validación de dependencia (“depends\_on” en template\_attribute\_validation) entre estos atributos identificando cuál de los dos depende del otro, siendo ambos de la misma plantilla.

Otra configuración con la que se relacionan valores de atributos es la que se establece para la taxonomía de selección única o múltiple. En estos casos, los valores forman un árbol de dependencias en el que debe detallarse qué nodo del árbol es padre de qué otro nodo.



Para este ejemplo, cada uno de los nodos raíz (A y B) deben tener configurada una relación cuyo padre es <null> para que Anjana interprete ese nodo como inicio de una rama. En cualquiera de los atributos de tipo taxonomía de selección única o múltiple es necesario tener, al menos, un nodo raíz.

Estas relaciones entre valores de atributos se definen en la tabla **attribute\_relationships**.

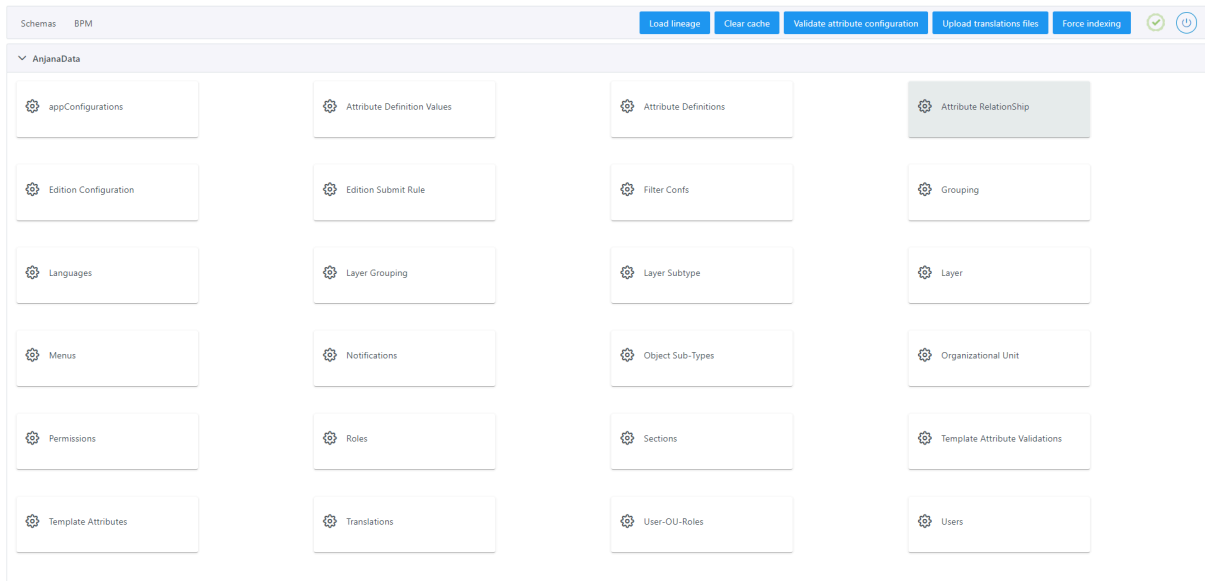
### Estructura de la tabla

Cada relación registrada se caracteriza por los siguientes elementos:

- **id\_attribute\_relationships**: identificador único de la tabla.
- **destination\_value**: valor del atributo que depende del otro.
- **source\_value**: valor del atributo original del que depende el destinationValue.
- **object\_sub\_type**: plantilla a la que se asocia el atributo (null si es a cualquier plantilla donde esté el atributo)

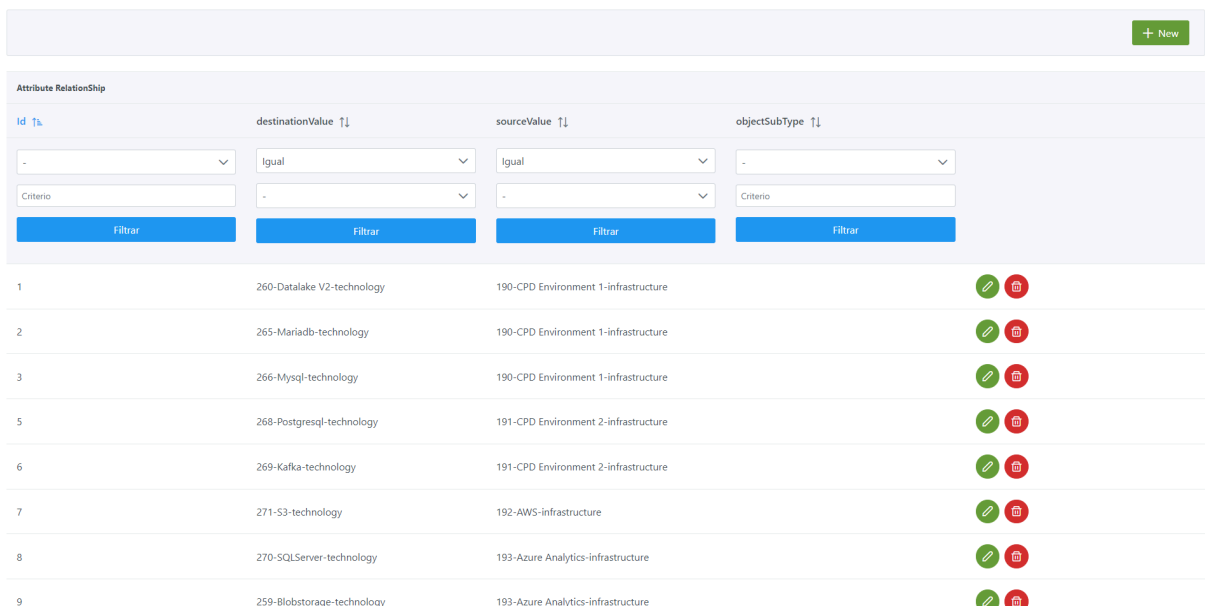
### Visión de Administrador

El alta de dependencias de este tipo se realiza en el panel de administración de Anjana Data en **Attribute Relationship**:

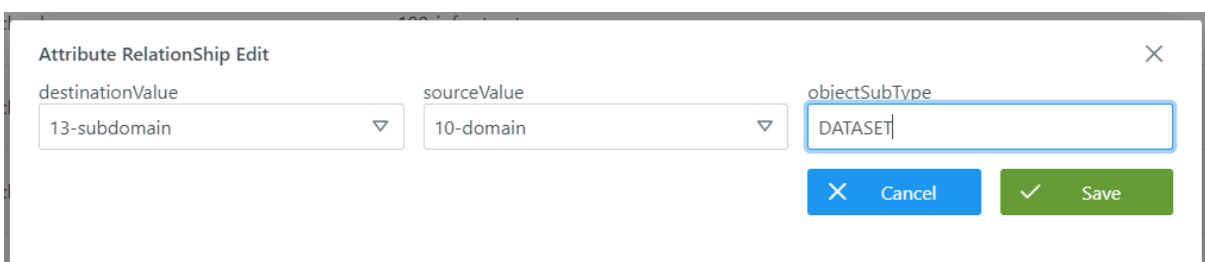


Al acceder se muestra una tabla que contiene todas las relaciones entre atributos y plantillas existentes en la configuración actual.

La creación de una nueva relación se realiza mediante el botón **New**:



Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:





## Visión de Desarrollador

Para definir las relaciones entre atributos hay que configurar la tabla **attribute\_relationships** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
insert into anjana.attribute_relationships
(id_attribute_relationships, source_value, destination_value, object_sub_type) values
(1,190,260, null)
,(2,190, 265, null)
,(3,190, 266, null)
,(4,191, 267, null)
,(5,191,268, null)
,(6,191,269, null);
```

NOTAS:

- Para poder identificar la dependencia entre los valores de los atributos infraestructura, tecnología y zona es necesario que se configuren las relaciones entre ellos.

## 9. Reglas de Versionado

En Anjana es posible configurar qué cambios generan un versionado en los objetos mediante las reglas de versionado. Estas reglas permiten identificar los atributos de los objetos que, al ser editados, son suficientemente relevantes como para versionar el objeto y deprecar la versión original.

Las reglas se definen en la tabla **edition\_configuration** y sólo pueden ser configuradas para las entidades nativas del Catálogo de Datos: dataset, dataset\_field, DSA, proceso, instancia de proceso y solución.

Es importante que los atributos identificados como [obligatorios](#) estén configurados para poder versionar por ellos si se desea.

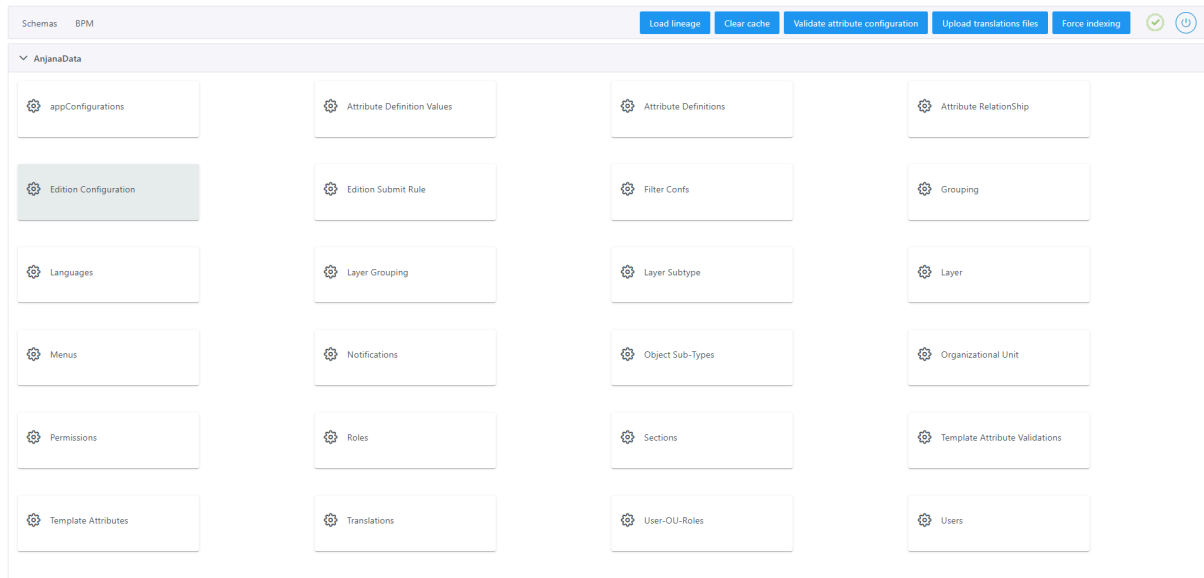
### Estructura de la tabla

Cada regla de versionado registrada se caracteriza por los siguientes elementos:

- **id\_edition\_configuration:** identificador único de la tabla.
- **current\_value:** valor original del atributo en caso de que sea booleano y sea de interés para la regla. Si se desea que haya versionado independientemente del valor inicial, introducir *null*. Valor técnico configurado en attribute\_definition\_value.
- **new\_value:** valor nuevo del atributo. Si se desea que haya versionado independientemente del valor nuevo, introducir *null*. Valor técnico configurado en attribute\_definition\_value.
- **id\_template\_attribute:** atributo de una plantilla concreta al que aplica la regla.

## Visión de Administrador

El alta de las reglas de versionado se realiza en el panel de administración de Anjana Data en **Edition Configuration**:



Al acceder se muestra una tabla que contiene todas las reglas de versionado existentes en la configuración actual.

La creación de una nueva regla se realiza mediante el botón **New**:

+ New

Edition Configuration				
Id ↑	currentValue ↑↓	newValue ↑↓	templateAttribute ↑↓	
-	-	-	Igual	
<input type="text" value="Criterio"/> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<input type="text" value="Criterio"/> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<input type="text" value="Criterio"/> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<input type="text" value="-"/> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	
1	false	true	258-pi-DATASET	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
2	false	true	455-pi-DATASET_FIELD	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
4	false	true	275-golden-DATASET	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
5			21-data_sensitivity-TERM	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
6			46-data_sensitivity-REPORT	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
7			96-data_sensitivity-BUSINESS_PROCESS	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
8			121-data_sensitivity-KPI	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
9			147-data_sensitivity-DIMENSION	<span style="color: green;">✎</span> <span style="color: red;">✖</span>

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:

Edition Configuration Edit ✕

currentValue	newValue	templateAttribute
false	true	pi ▾

✕ Cancel
✓ Save

### Visión de Desarrollador

Las reglas de versionado se configuran en la tabla **edition\_configuration** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.edition_configuration
(id_edition_conf, current_value, new_value, id_template_attribute) VALUES
(1, false, true, 258)
,(2, false, true, 455)
,(5, null, null, 21)
,(6, null, null, 46)
,(30, null, null, 472)
,(31, null, null, 328);
```

#### NOTAS:

- Si se desea versionar un dataset en caso de que se modifiquen sus dataset\_fields, es necesario añadir una regla de versionado para el atributo datasetFields del dataset. De esta forma, se comprobarán los cambios ocurridos en los dataset\_fields y se versionará el dataset si alguno de esos cambios coincide con alguna regla de versionado para los atributos de dataset\_fields.

## 10. Reglas de lanzamiento de workflow en ediciones y versionado.

Es posible configurar qué cambios no generan el lanzamiento de workflows de ediciones y versionado en los objetos de Anjana Data mediante las reglas de edición. Estas reglas permiten identificar los atributos de los objetos que, al ser editados por cierto rol que lanza la validación, no generen workflow de aprobación quedando el objeto validado automáticamente.

Esta configuración se lleva a cabo en la tabla **edition\_submit\_rule**.

### Estructura de la tabla

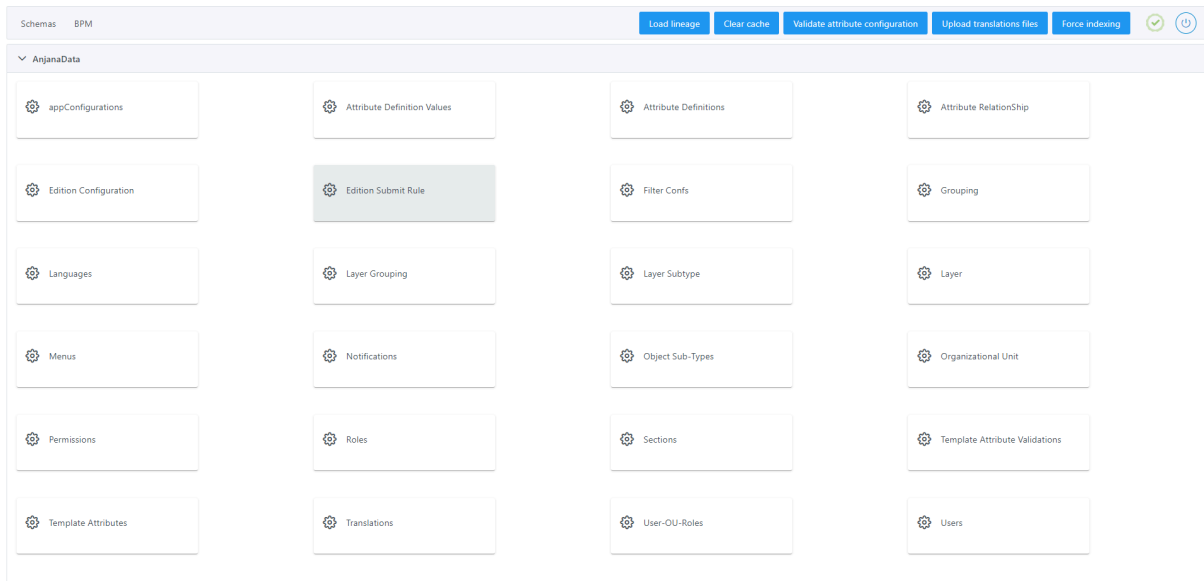
Cada regla de lanzamiento de workflow registrada se caracteriza por los siguientes elementos:

- **id\_edition\_submit\_rule**: identificador único de la tabla.
- **id\_template\_attribute**: atributo de una plantilla concreta al que aplica la regla.

- **role:** nombre del rol que no tendría que lanzar un workflow si se edita el atributo definido en templateAttribute.

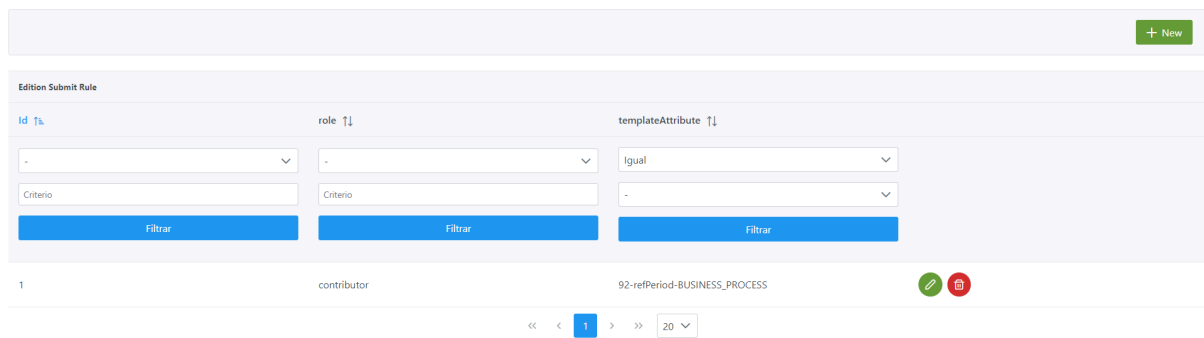
## Visión de Administrador

El alta de las reglas de lanzamiento de workflow en ediciones y versionado se realiza en el panel de administración de Anjana Data en **Edition Submit Rule**:



Al acceder se muestra una tabla que contiene todas las reglas que se usan para decidir si se lanza un workflow o no cuando se edita o versiona un objeto.

La creación de una nueva regla se realiza mediante el botón **New**:



Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:

Edition Submit Rule Edit ✕

role

templateAttribute

✕ Cancel ✓ Save

### Visión de Desarrollador

Las reglas de lanzamiento de workflow en ediciones y versionado se configuran en la tabla **edition\_submit\_rule** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.edition_submit_rule
(id_edition_submit_rule, "role", id_template_attribute) VALUES
(1,'contributor',92),
(3,'data_owner',46);
```

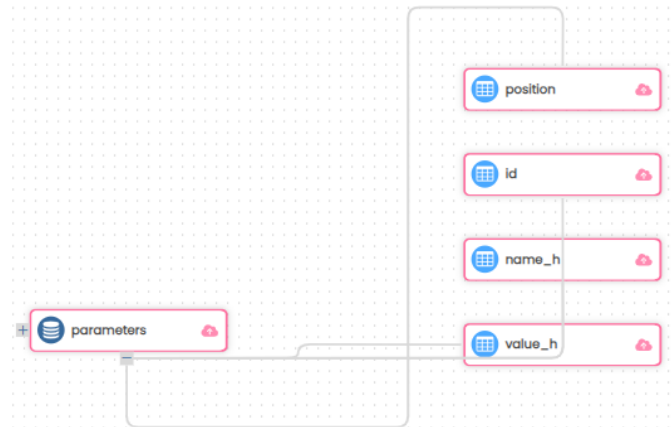
#### NOTAS:

- Si se desea no lanzar workflow en caso de que se modifiquen los dataset\_fields de un dataset, es necesario añadir una regla para el atributo datasetFields del dataset. De esta forma, se comprobarán los cambios ocurridos en los dataset\_fields y no se lanzará workflow de validación del dataset si los cambios coinciden con alguna regla de lanzamiento de workflow para los atributos de los dataset\_fields. En caso de que los dataset\_fields no haya sufrido modificaciones pero se haya añadido o eliminado alguno existente, con la regla para el atributo datasetFields no se lanzará workflow de validación.

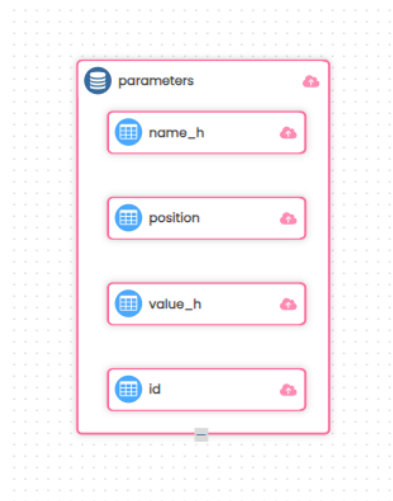
## 11. Agregaciones para el linaje

En el linaje de Anjana es posible representar determinadas relaciones entre entidades como una relación de agregación donde una entidad contiene a otras.

Por defecto una relación en el linaje se mostrará como una línea entre los dos nodos que representan sus entidades origen y destino.



Pero, si está configurada una agregación para esa relación, esa relación pasará a mostrarse como de contenido.



Estas agregaciones se definen en la tabla **grouping\_lineage**.

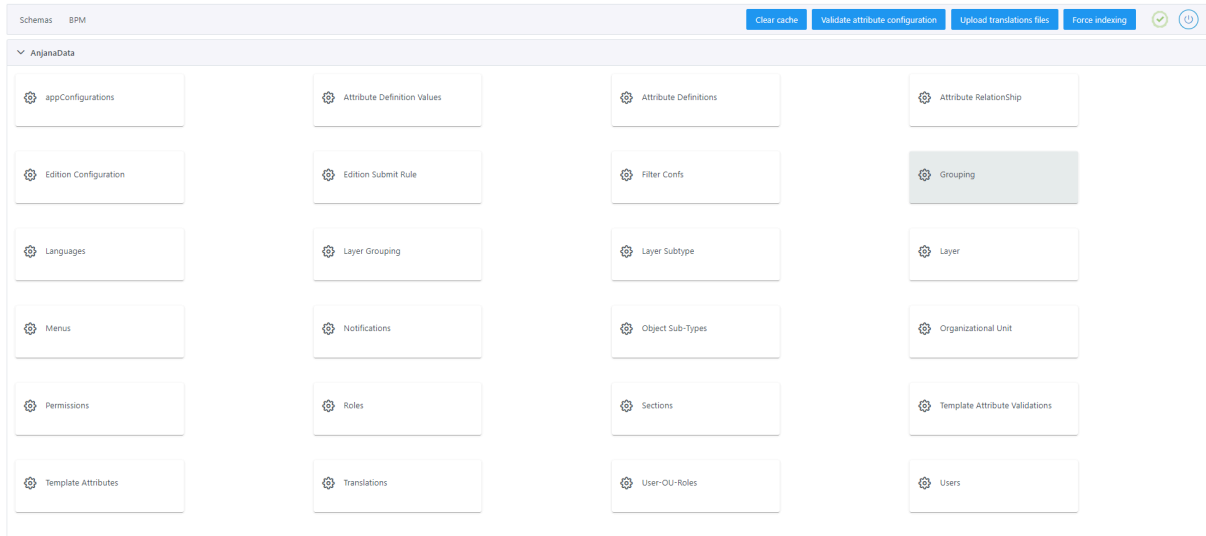
### Estructura de la tabla

Cada agregación registrada se caracteriza por los siguientes elementos:

- **id**: identificador único de la tabla.
- **source\_subtype**: subtipo de objeto que agrega.
- **destination\_subtype**: subtipo de objeto que aparece en el linaje contenido en entidades de tipo source\_subtype.
- **relationship\_subtype**: subtipo de relación que hay entre source\_subtype y destination\_subtype y que se representa como una relación de contenido.

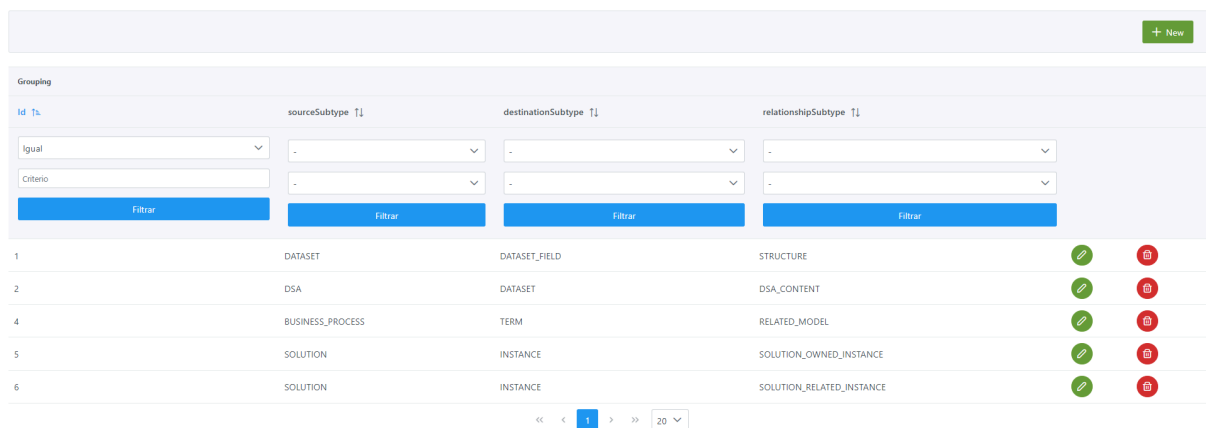
### Visión de Administrador

El alta de las agregaciones del linaje se realiza en el panel de administración de Anjana Data en **Grouping**:

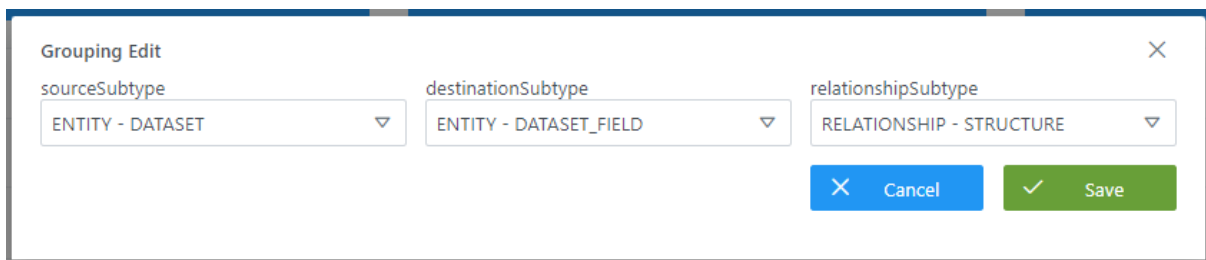


Al acceder se muestra una tabla que contiene todas las agregaciones definidas para las pantallas de linaje.

La creación de una nueva agregación se realiza mediante el botón **New**:



Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:



### Visión de Desarrollador

Las agregaciones de entidades para el linaje se configuran en la tabla **grouping\_lineage** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.grouping_lineage
```

```
(id, source_subtype, destination_subtype, relationship_subtype) VALUES
(1, 'DATASET', 'DATASET_FIELD', 'STRUCTURE'),
(2, 'DSA', 'DATASET', 'DSA_CONTENT');
```

## 12. Capas del linaje

Es posible visualizar el linaje con distintas perspectivas. Estas perspectivas se configuran como capas de visualización que facilitan al usuario la navegación filtrando en función de la configuración entidades o relaciones por capas.

Estas capas se definen en la tabla **layer**.

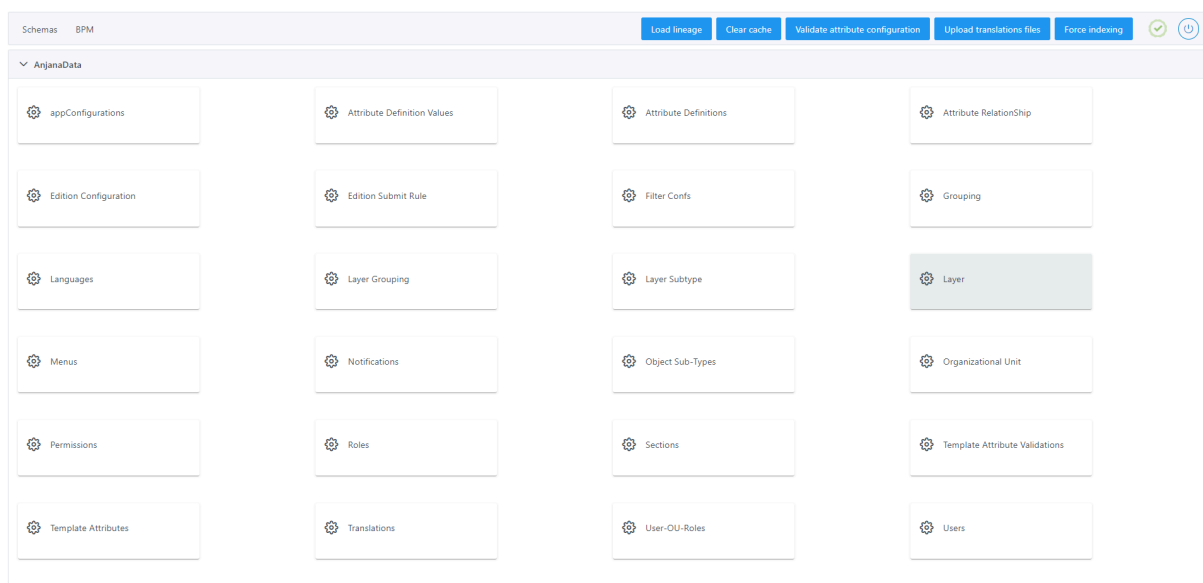
### Estructura de la tabla

Cada capa registrada se caracteriza por los siguientes elementos:

- **id**: identificador único de la tabla.
- **layer\_name**: nombre de la capa
- **is\_default**: flag que determina si la capa se muestra por defecto cuando los usuarios acceden al linaje. Se debe marcar sólo una capa con `is_default = "true"`

### Visión de Administrador

El alta de las capas del linaje se realiza en el panel de administración de Anjana Data en **Layer**:



Al acceder se muestra una tabla que contiene todas las capas definidas para las pantallas de linaje.

La creación de una nueva capa se realiza mediante el botón **New**:



Layer			
Id	Name	isDefault	
1	-		
2	Técnica	✓	✓
3	Negocio	✗	✓
4	Funcional	✗	✓

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:

**Layer Edit** ✕

Name   isDefault

### Visión de Desarrollador

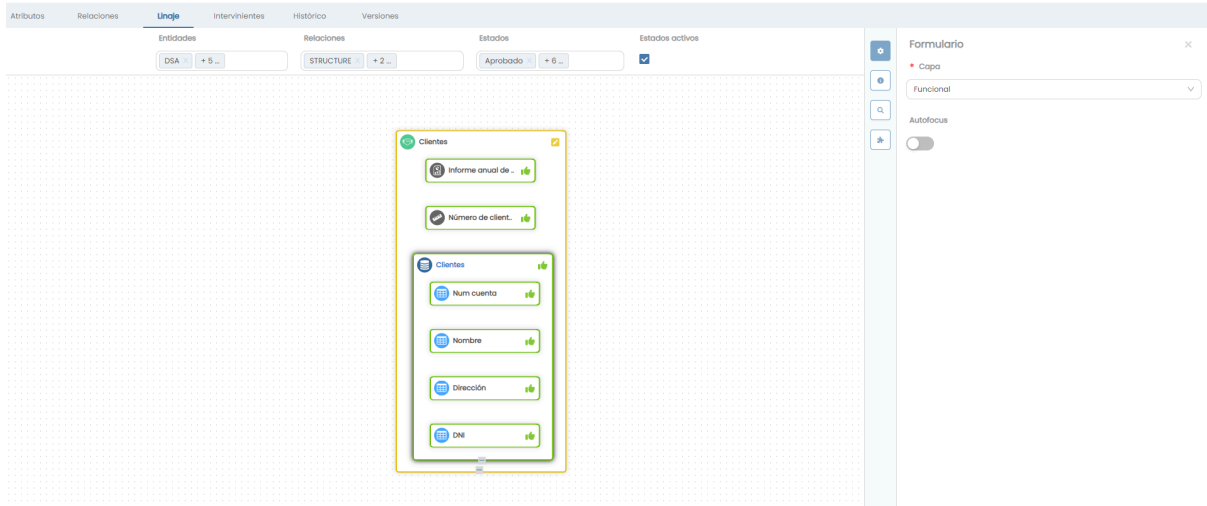
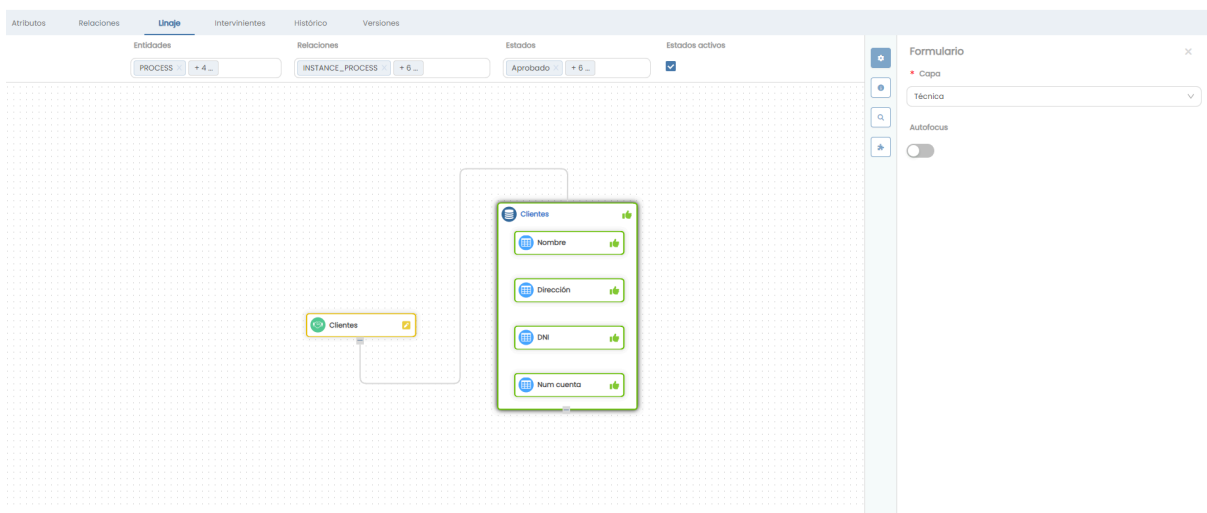
Las capas del linaje se configuran en la tabla **layer** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.layer (id, layer_name, is_default) VALUES
(2, 'Técnica', true),
(3, 'Negocio', false),
(4, 'Funcional', false);
```

## 13. Subtipos de las capas del linaje

Las capas del linaje filtran al usuario entidades o relaciones en función de la configuración. De esta forma, se configuran capas específicas para usuarios técnicos evitando entidades del Glosario de Negocio distintas de las capas necesarias para usuarios de negocio en las que se pueden obviar entidades del Catálogo de Datos.

En este caso, en la capa Funcional se visualizan un reporte y una dimensión que no se ven en la capa Técnica:

El conjunto de entidades o relaciones que se va a presentar en cada capa se configura en la tabla **layer\_subtype**.

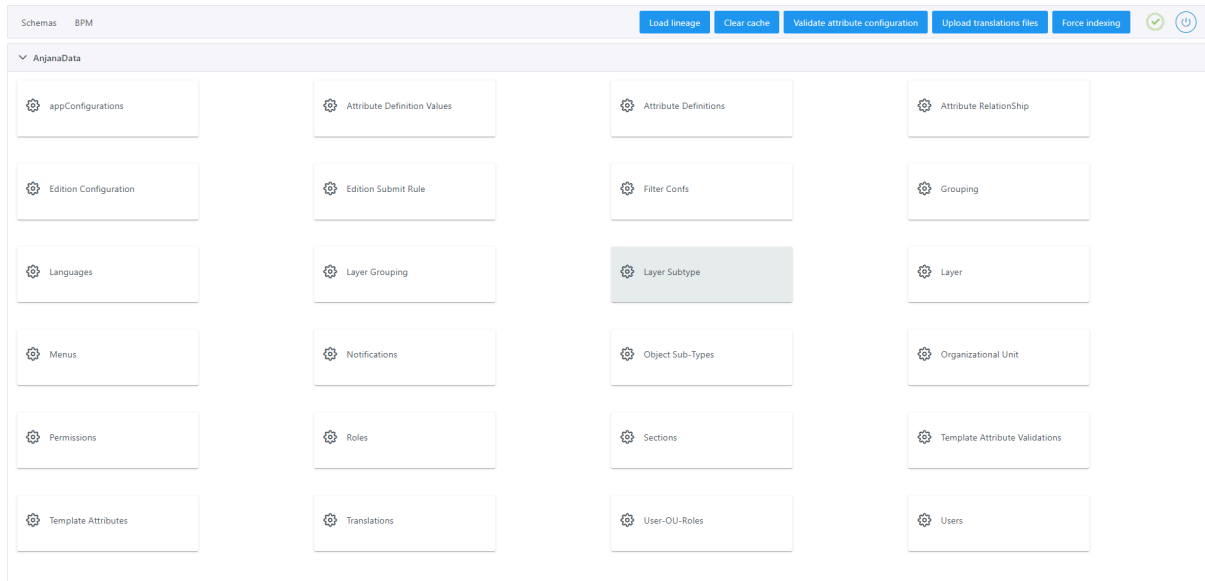
### Estructura de la tabla

Cada subtipo elegido por capa definido se caracteriza por los siguientes elementos:

- **layer\_id**: identificador de la capa en la tabla layer.
- **object\_subtype**: subtipo de objeto que aparece en la capa.

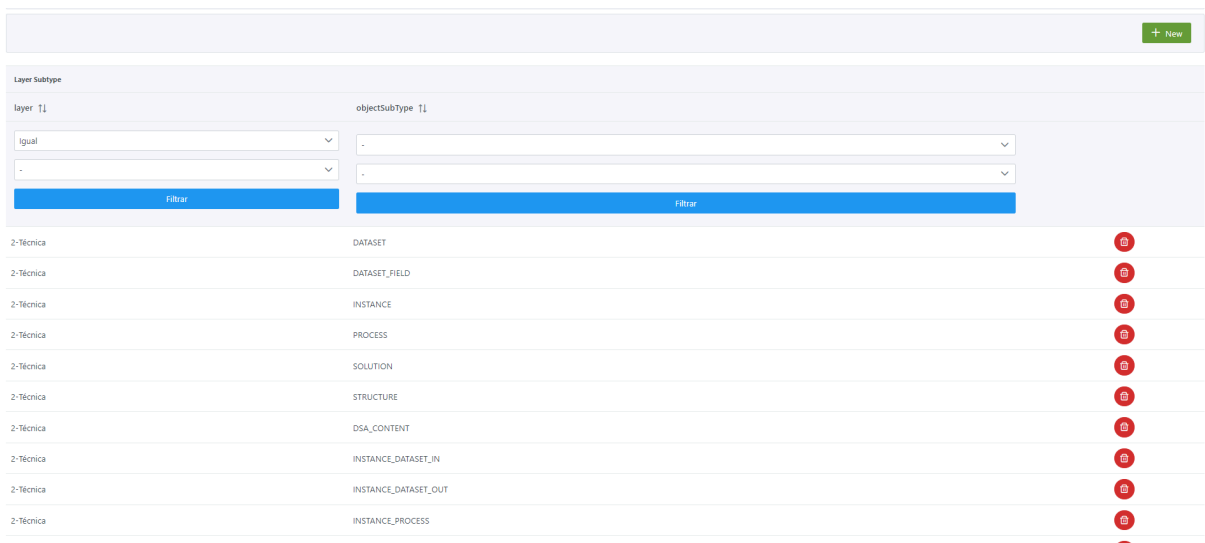
### Visión de Administrador

El alta del conjunto de subtipos de objeto que aparecen en cada capa se realiza en el panel de administración de Anjana Data en **Layer Subtype**:

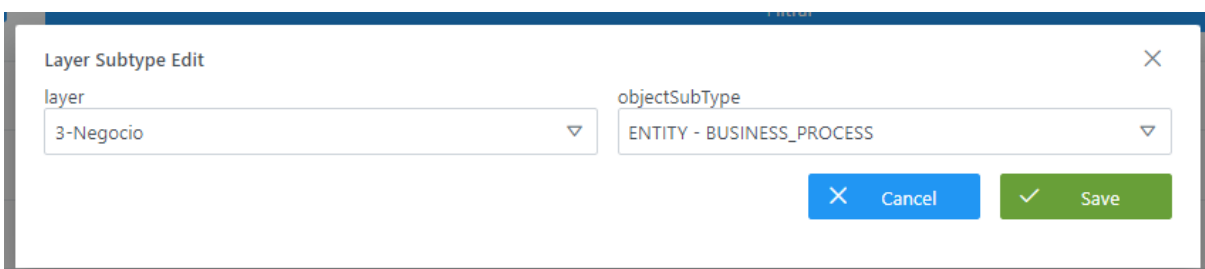


Al acceder se muestra una tabla que contiene los subtipos por capa definidos para las pantallas de linaje.

La creación de un nuevo subtipo para una capa se realiza mediante el botón **New**:



Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:



## Visión de Desarrollador

Los subtipos por capa del linaje se configuran en la tabla **layer\_subtype** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.layer_subtype (layer_id, object_subtype) VALUES
(2, 'DATASET'),
(2, 'DATASET_FIELD'),
(2, 'INSTANCE'),
(2, 'PROCESS'),
(2, 'SOLUTION'),
(2, 'STRUCTURE'),
(2, 'DSA_CONTENT'),
(2, 'INSTANCE_DATASET_IN'),
(2, 'INSTANCE_DATASET_OUT'),
(2, 'INSTANCE_PROCESS'),
(2, 'SOLUTION_OWNED_INSTANCE'),
(2, 'SOLUTION_RELATED_INSTANCE'),
(3, 'BUSINESS_PROCESS');
```

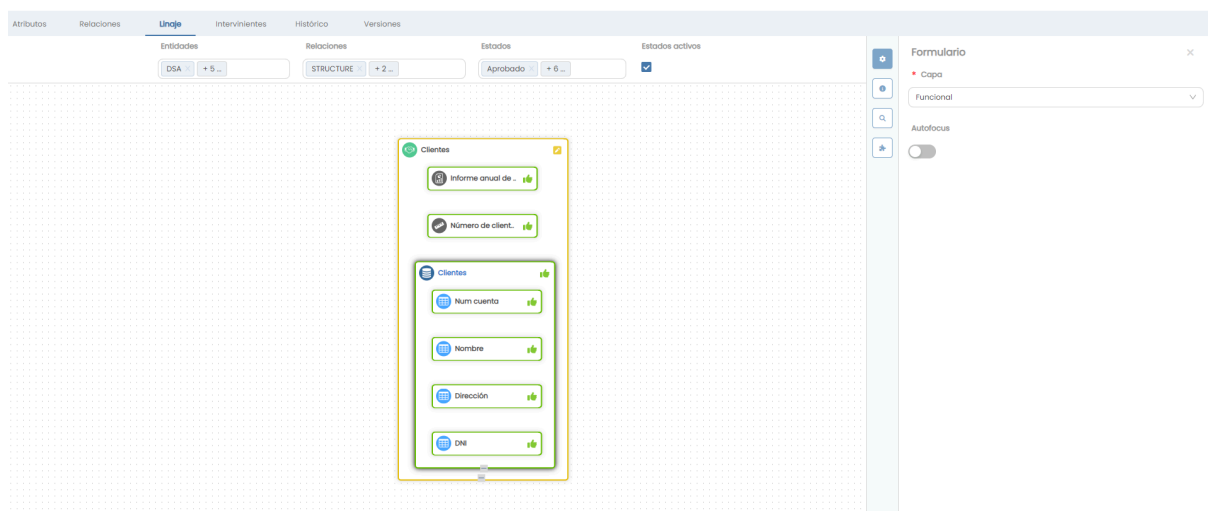
### NOTAS:

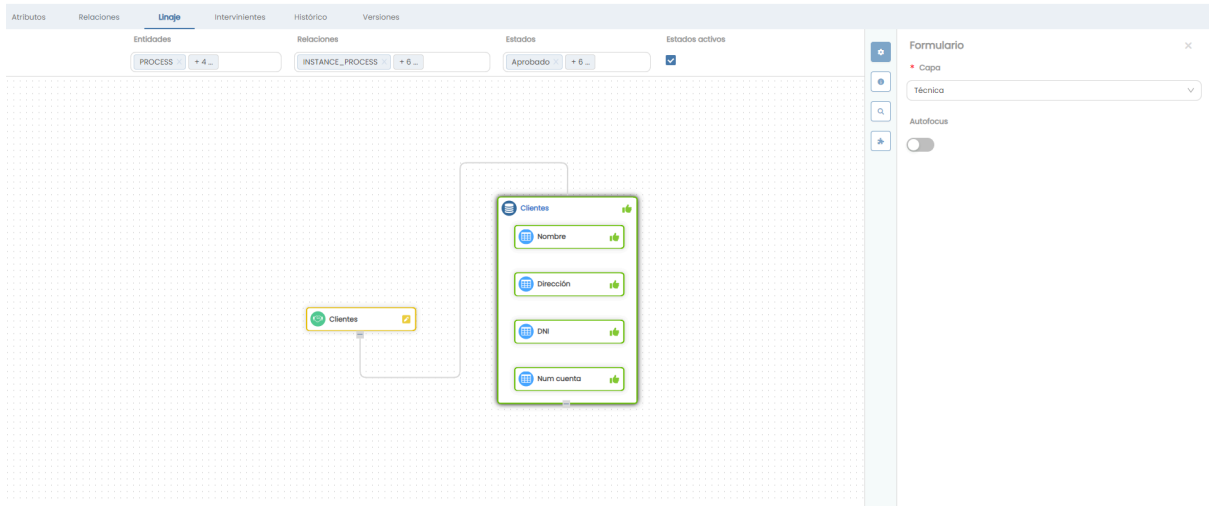
- Cuando se añade un subtipo de objeto nuevo en la tabla [object\\_subtype](#) es necesario configurar las capas donde se desea visualizar.

## 14. Agregaciones en las capas del linaje

En las capas del linaje, así como se definen las entidades o relaciones que se cargan en cada una, también se puede configurar qué relaciones de agregación se desea que apliquen. En caso de que no se configure una agregación para una capa, la relación se mostrará como una línea entre los dos nodos correspondientes a las entidades origen y destino de la relación.

En este caso, en la capa Funcional el DSA agrega a las entidades que contiene y, sin embargo, en la capa Técnica no:





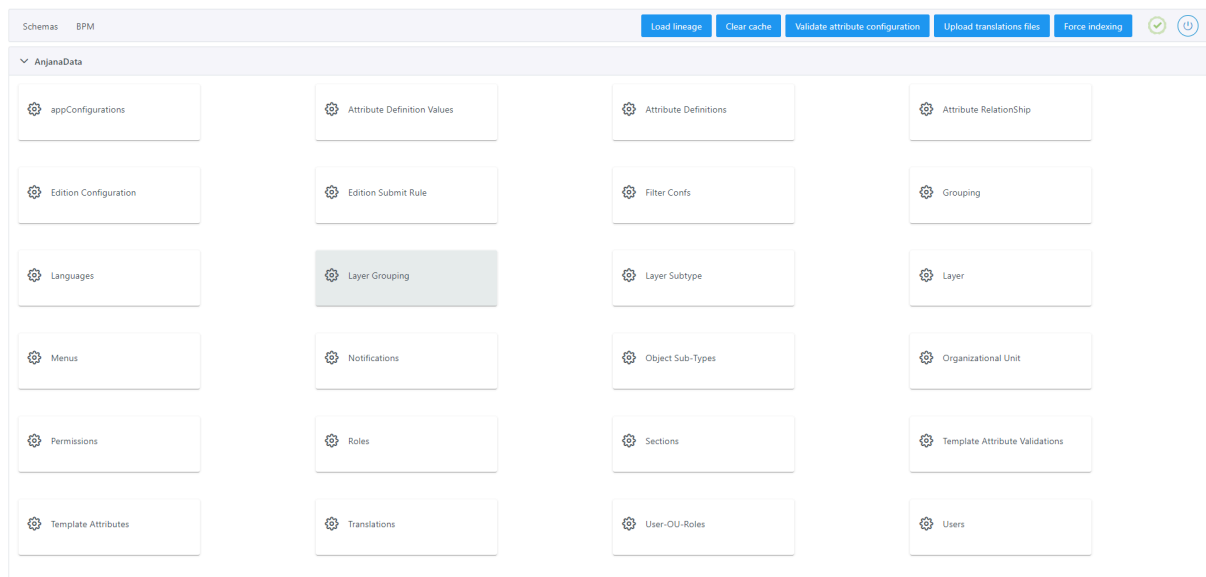
## Estructura de la tabla

Cada agregación por capa definida se caracteriza por los siguientes elementos:

- **layer\_id**: identificador de la capa en la tabla layer.
- **grouping\_id**: identificador de la agregación en la tabla grouping-lineage.

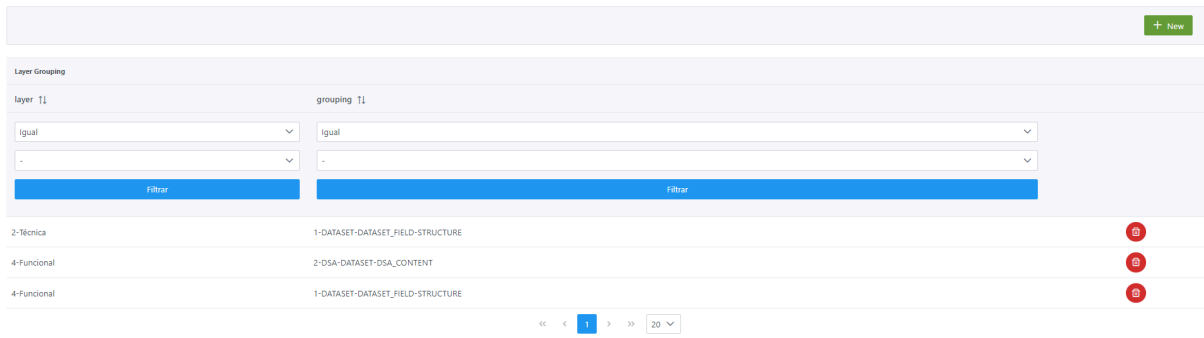
## Visión de Administrador

El alta de las agregaciones para cada capa se realiza en el panel de administración de Anjana Data en **Layer Grouping**:



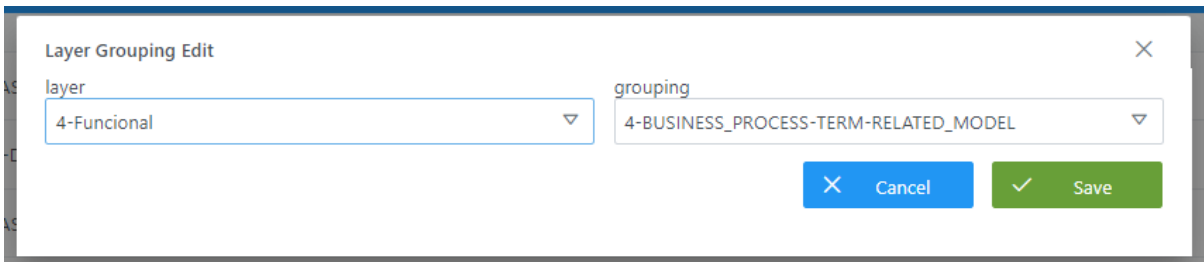
Al acceder se muestra una tabla que contiene las agregaciones por capa definidas para las pantallas de linaje.

La creación de un nuevo subtipo para una capa se realiza mediante el botón **New**:



layer	grouping
2-Técnica	1-DATASET-DATASET_FIELD-STRUCTURE
4-Funcional	2-DSA-DATASET-DSA_CONTENT
4-Funcional	1-DATASET-DATASET_FIELD-STRUCTURE

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:



### Visión de Desarrollador

Las agregaciones por capa del linaje se configuran en la tabla **layer\_grouping** del esquema Anjana. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO anjana.layer_grouping (layer_id, grouping_id) VALUES
(2, 1),
(4, 2),
(4, 1);
```

## Esquema Hermes de BD

notification	
123	<b>id_notification</b>
ABC	module_type
ABC	notification_code
ABC	notification_receiver_type
ABC	translation_key
ABC	notification_type
ABC	receiver_role
ABC	severity
ABC	subject

## 1. Notificaciones del sistema

La tabla **notification** es la tabla que contiene los cuerpos de las diferentes notificaciones que emite el sistema.

### Estructura de la tabla

Cada notificación se caracteriza por los siguientes elementos:

- **id\_notification**: identificador único de la tabla.
- **module\_type**: módulo al que aplica la notificación, indicar “BG” para *Business Glossary*, “DC” para *Data Catalog* o “ALL” si aplica a todo.
- **notification\_code**: código alfanumérico para poder enviar notificaciones desde las funcionalidades propias de Anjana.
- **notification\_receiver\_type**: indica si la notificación será recibida por un único usuario nominal (‘USER’) o por todos los usuarios que tengan asignado un rol (‘ROLE’).
- **translation\_key**: clave de traducción correspondiente a la tabla de `portuno.translations` con el mensaje de la notificación con variables que serán sustituidas en el momento de la creación de la notificación a enviar.
- **notification\_type**: indica si la notificación es alerta (‘ALERT’), aviso (‘NOTICE’) o alerta de administración (‘ADMIN\_ALERT’).
- **receiver\_role**: rol que recibe la notificación en caso de que no sea para un usuario nominal.
- **severity**: criticidad de la notificación.
- **subject**: clave de traducción con el asunto de la notificación.

A continuación se especifican los **notification\_code** para los que deben existir notificaciones en la tabla con el fin de conseguir un correcto funcionamiento de la aplicación:

- ADHERENCE\_FAIL → Indica el motivo por el que ha fallado la adherencia.
- CHECK\_CONFIGURATION\_IN\_PROVIDER → Error al recuperar la información de un usuario en el provider indicado.
- COMPLETED\_AUTOMATIC\_METADATA → Indica que ha terminado la importación de metadatos automática.
- DATASET\_EXPIRATION\_TOT\_FAIL → Fallo en el borrado de permisos al expirar un dataset.
- DATASET\_FAIL → Fallo en la creación de dataset en sistemas de terceros.
- DELETE\_ENTITY → Avisa del borrado de una entidad.
- DELETE\_RELATIONSHIP → Avisa del borrado de una relación.
- DEPRECATION → Avisa a todos los usuarios de un rol que el objeto va a ser deprecado.
- DEPRECATION\_ADHERED → Avisa al usuario que el objeto al que estaba adherido (dataset o DSA) va a ser deprecado.
- DISADHERENCE → Se indica que se ha realizado correctamente la desadherencia.
- DISADHERENCE\_FAIL → Fallo en la desadherencia indicando el motivo.
- DISADHERENCE\_LIST → Informa de los detalles del resultado del proceso de desadherencia.
- DISADHERENCE\_LIST\_OK → Se indica que la lista de objetos en los que ha tenido éxito la desadherencia.
- DSA\_FAIL → La creación de un DSA ha fallado indicando el motivo.
- DSA\_TRANSFER\_FAIL → Se le indica al administrador que no se ha podido hacer el cambio de unidad organizativa del DSA.

- ERROR\_AUTOMATIC\_METADATA → Se produce un fallo en la importación de metadatos.
- EXPIRATION → El objeto está expirando.
- EXPIRATION\_OBJECT\_FAIL → Avisa de un fallo en la expiración de un objeto y el motivo
- EXPIRATION\_ADHERED → El objeto al que se está adherido ha expirado.
- EXPIRATION\_WARNING → Aviso que se envía a los propietarios del objeto debido a que está próxima su expiración.
- EXPIRATION\_WARNING\_ADHERED → Aviso que se envía a los usuarios adheridos al objeto que indica que está próxima su expiración.
- FORM\_FAIL → Errores encontrados en el formulario del objeto.
- INDEX\_FAIL → Se ha producido un error en la indexación.
- INDEX\_RELATIONSHIP\_FAIL → Error de indexación en relaciones.
- LICENSE\_EXPIRED → La licencia ha expirado.
- LICENSE\_EXPIRING → Aviso de que la licencia, aunque aún es válida, está cerca de expirar.
- NEW\_DSA → Cuando el usuario solicita la creación de un nuevo DSA.
- PENDING\_WF\_FAIL\_ADHERENCE → Indica el fallo en la creación de un workflow
- POLICY\_MODIFICATION → Avisa de la modificación de políticas en un objeto.
- REQUIRED\_FIELDS → Avisa de errores en la configuración de las plantillas.
- TAXONOMY\_BAD\_CONFIGURED → Avisa de algún error en la configuración de la taxonomía de algún subtipo de objeto.
- TRANSLATION\_FILE\_UPLOAD\_FAIL → Avisa de que ha habido un error subiendo el fichero de traducciones a Minio.
- USER\_CROSS\_ROLE\_FAIL → Indica los usuarios que no han sido configurados como cross.
- WORKFLOW\_DELETE\_FAIL → No se ha podido borrar un workflow.
- WORKFLOW\_INFO\_FAIL → Error en el workflow debido a que no tenía un workflow info asociado, mediante la api administrativa se pueden ejecutar los últimos pasos del workflow para que no queden en mal estado si sucede este error.
- WORKFLOW\_OK → El workflow ha sido creado.
- WORKFLOW\_FAIL → Indica el motivo por el que ha fallado la creación de un workflow.

Adicionalmente, los textos correspondientes a las **translation\_keys** en la tabla de traducciones pueden incluir variables entre **##** con el fin de que éstas sean sustituidas por los valores correspondientes a la notificación enviada. Un ejemplo de un mensaje con variables sería:

*“Ocurrió un error en el borrado del workflow con id #OBJECT\_ID#”.*

Si alguna de estas variables se usa en una notificación de workflow en la que no le corresponde tener valor, se sustituye por el valor “N/A”.

Las distintas variables que permite Anjana son:

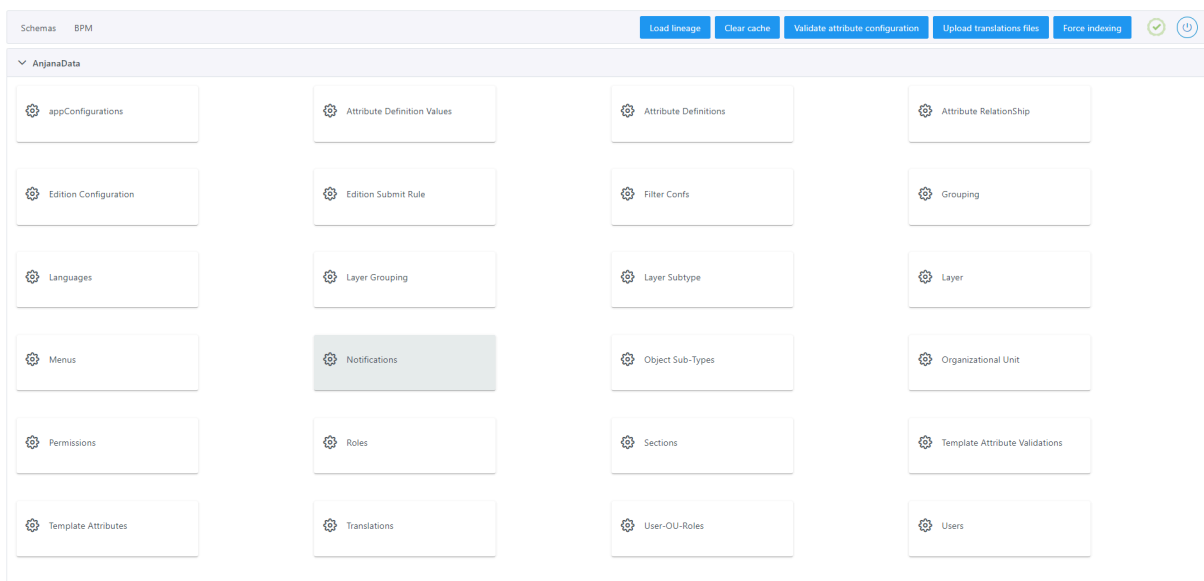
- ACTION → Acción realizada
- EXPIRATION\_DAYS -> Días hasta la expiración de un objeto (es un número negativo en caso de que se una fecha a pasado). Sólo se permite en las notificaciones de aviso de expiración cercana y deprecación
- OBJECT\_ID → Identificador del objeto
- OBJECT\_LIST → Lista de objetos, se usa en notificaciones como las de solicitud de creación de un nuevo DSA
- OBJECT\_NAME → Nombre del objeto



- OBJECT\_SUB\_TYPE → Subtipo del objeto
- ORGANIZATIONAL\_UNIT → Unidad organizativa del objeto
- ORGANIZATIONAL\_UNIT\_CHANGED → Unidad de destino del objeto en una transferencia de OU
- REASON → Motivo de aprobación/rechazo del workflow
- REQUEST\_REASON → Indica el motivo de la solicitud de la acción, se rellena en las notificaciones de solicitud de creación de dsa o de solicitud de adherencia, por ejemplo
- RESULT → Resultado de la acción
- ROLE → Rol del usuario que debe realizar la acción o recibe la notificación
- TYPE\_OBJECT → Tipo del objeto
- USER\_NAME → Usuario que ha realizado la acción
- VERSION → Número de versión del objeto
- WORKFLOW\_TYPE → Tipo del workflow lanzado

## Visión de Administrador

El alta de una nueva notificación panel de administración de Anjana Data se realiza en **Notifications**:



Al acceder se muestra una tabla que contiene el catálogo de notificaciones existentes en la configuración actual.

La creación de una nueva notificación se realiza mediante el botón **New**:

[+ New](#)

Notifications								
Id	moduleType	notificationCode	notificationReceiverType	translationKey	notificationType	receiverRole	severity	subject
-	Igual	-	Igual	-	Igual	-	Igual	-
Criterio	-	Criterio	-	Criterio	-	Criterio	-	Criterio
<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>	<a href="#">Filtrar</a>
1	BG	ROLE	NOTIFICATION.1.BG.ROL	ALERT			HIGH	Test
3	BG	ROLE	NOTIFICATION.3.BG.ROL	NOTICE			LOW	Test
5	BG	USER	NOTIFICATION.5.BG.USE	NOTICE			LOW	Test
6	BG	ROLE	NOTIFICATION.6.DC.ROL	ALERT			HIGH	Test
7	DC	ROLE	NOTIFICATION.7.DC.USE	ALERT			HIGH	Test
8	DC	USER	NOTIFICATION.8.DC.USE	NOTICE			LOW	Test
9	DC	USER	NOTIFICATION.9.DC.USE	NOTICE			LOW	Test

A continuación, se muestra cómo crear una notificación :

### Notifications Edit

moduleType

notificationCode

notificationReceiverType

notificationType

receiverRole

severity

subject

translationKey

NOTIFICATION.EXPIRATION

X Cancel
✓ Save

### Visión de Desarrollador

Para añadir las notificaciones hay que configurar la tabla **notifications** del esquema Hermes. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO hermes.notification
```

```
(id_notification, module_type, notification_code, notification_receiver_type, translation_key,
notification_type, receiver_role, severity, subject) VALUES
(1,'BG',NULL,'ROLE','NOTIFICATION.1.BG.ROLE','ALERT',NULL,'HIGH','Test'),
(3,'BG',NULL,'ROLE','NOTIFICATION.3.BG.ROLE','NOTICE',NULL,'LOW','Test'),
(12,'BG','NEW_DSA','ROLE','NOTIFICATION.NEW_DSA','NOTICE','architect','LOW','Test').
(13,'BG','EXPIRATION','ROLE','NOTIFICATION.EXPIRATION','NOTICE','data_owner','LOW','Expiration'
),
(14,'DC','ERROR_AUTOMATIC_METADATA','ROLE','NOTIFICATION.AUTOMATIC_METADATA','NOTICE
','architect','LOW','METADATA'),
(15,'DC','DATASET_FAIL','ROLE','NOTIFICATION.DATASET_FAIL','ADMIN_ALERT','admin','HIGH','Data
set creation failure'),
(16,'DC','DSA_FAIL','ROLE','NOTIFICATION.DSA_FAIL','ADMIN_ALERT','admin','HIGH','DSA creation
failure');
```

## Esquema Minerva de BD

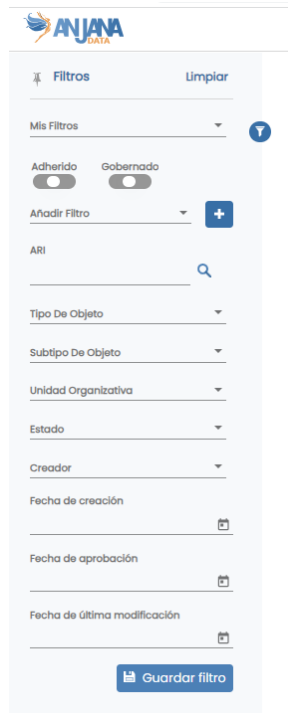
filter_conf	
123	id
ABC	collection
<input checked="" type="checkbox"/>	groupable
ABC	label
<input checked="" type="checkbox"/>	main
123	position
<input checked="" type="checkbox"/>	sortable
ABC	type
ABC	filter_name
ABC	page_name

### 1. Filtros del sistema

Los filtros disponibles en los buscadores del Portal de Datos y la cabecera de la aplicación o los atributos Anjana en los que se muestran entidades o relaciones indexados en Solr son configurables. Adicionalmente, se pueden configurar filtros para las pantallas de histórico y auditoría, persistidos en la base de datos.

A continuación se muestran algunos de los filtros configurados:

- Filtros del Portal



ANJANA DATA

Filtros Limpiar

Mis Filtros

Adherido  Gobernado

Añadir Filtro +

ARI

Tipo De Objeto

Subtipo De Objeto

Unidad Organizativa

Estado

Creador

Fecha de creación

Fecha de aprobación

Fecha de última modificación

Guardar filtro

- Filtros en atributo de tipo Entidad

Buscador de objetos

Filtros seleccionados

Buscar Fecha de última modificación Fecha de aprobación Creador Fecha de creación

Unidad Organizativa ARI

Europa(ESP)/Finanzas ARI

Añadir filtro Restablecer Aplicar

Objeto	Proveedor	Origen	Unidad Organizativa	Entidad
DS_TEST_VALIDATED VERSION 1 Aprobado	aws3	aws3	Europa(ESP)/Finanzas	ASD
df_2010 JOBIC VERSION 0 Aprobado	postgresql	aws3	Europa(ESP)/Finanzas	asaaa
df_DBII AWS VERSION 0 Aprobado	aws3glue	aws3	Europa(ESP)/Finanzas	azccc
depreca Google VERSION 1 Aprobado	gcpstorage	aws3	Europa(ESP)/Finanzas	gflgh

Cancelar Confirmar

- Filtros en histórico

Filtros seleccionados				
Usuario	Acción de búsqueda	Acción	Fecha	
<input type="text" value="Usuario"/>	<input type="checkbox"/>	<input type="text" value="Acción"/>	13 nov 2023 - 20 nov 2023	<input type="button" value="Reestablecer"/> <input type="button" value="Aplicar"/>
Acción	Usuario	Fecha	Descripción	
Guardado	architect	20 nov 2023 05:26 PM	El usuario architect ha guardado ENTITY Clientes con subtipo DGA	<input type="button" value="⬇"/>
Añadir entidad contenida	architect	20 nov 2023 05:26 PM	El usuario architect ha añadido la entidad REPORT Informe anual de clientes	<input type="button" value="⬇"/>
Añadir entidad contenida	architect	20 nov 2023 05:26 PM	El usuario architect ha añadido la entidad DATASET Clientes	<input type="button" value="⬇"/>
Añadir entidad contenida	architect	20 nov 2023 05:26 PM	El usuario architect ha añadido la entidad DIMENSION Número de clientes	<input type="button" value="⬇"/>
Creación	architect	20 nov 2023 05:25 PM	El usuario architect ha creado el ENTITY Clientes con subtipo DGA	<input type="button" value="⬇"/>

La tabla **filter\_conf** es la tabla que contiene los filtros preconfigurados disponibles en la plataforma.

Existen filtros cuyos valores no se traducen como el de tipo de objeto, subtipo del objeto y nombres de usuario o de tipo SELECT\_USERS.

## Estructura de la tabla

Cada filtro registrado se caracteriza por los siguientes elementos:

- **Id:** identificador único de la tabla.
- **Collection:** datos para los que se configura el filtro.
- **Groupable:** boolean que por defecto adquiere el valor *“false”*. No se hace uso de este campo.
- **Label:** clave (*key*) de traducción que permite que el usuario vea el nombre del filtro en su idioma. Para los atributos de metadatos que se añadan como filtros adicionales, en este campo se debe poner el mismo **label** que en la tabla `anjana.attribute_definition`.
- **Main:** boolean que por defecto adquiere el valor *“false”*. Se debe poner *“true”* si se desea que el filtro sea estático, es decir, que aparezca por defecto sin necesidad de añadirlo el usuario.
- **Position:** orden en el que se desea que se visualice el filtro dentro de Anjana.
- **Sortable:** poner a *“true”*. No se hace uso de este campo.
- **Type:** indica el tipo de filtro.
- **Filter\_Name:** campo de Apache Solr o de la tabla `audit_log` por el que va efectuar el filtro.
- **Page\_Name:** página a la que aplica el filtro configurado.

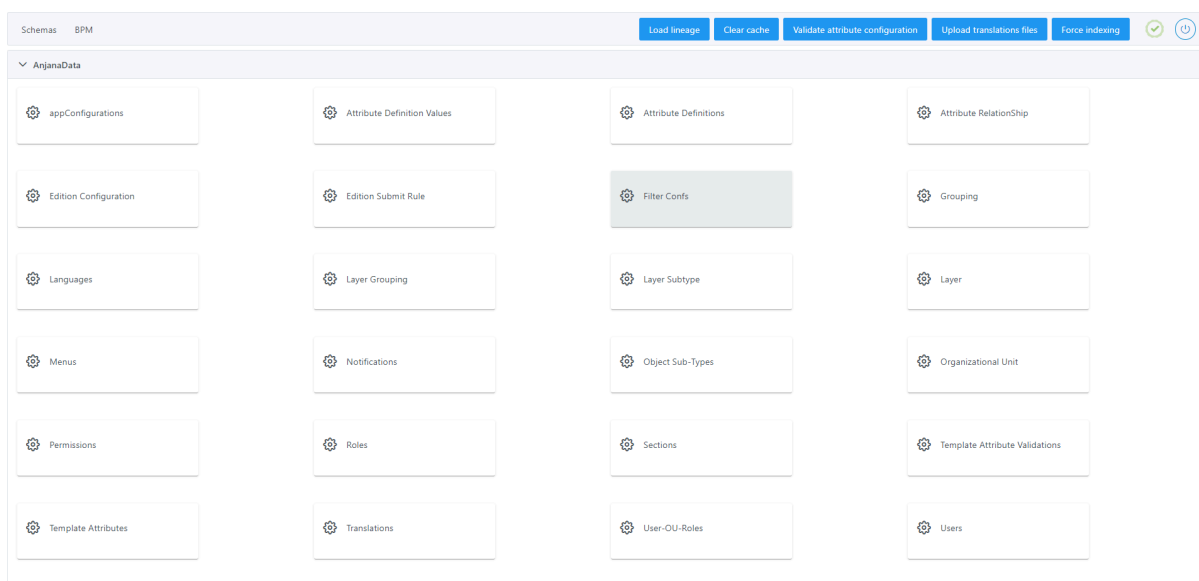
En función de las pantallas para las que se configuren, **Collection**, **Name**, **Type** y **Page\_Name** deben ser los siguientes:

- Filtros del Portal
  - **Collection** será *“kerno”*
  - El Nombre (**Name**) de los filtros debe tener el mismo **name** que tiene el atributo sobre el que se efectúa el filtro en la tabla `anjana.attribute_definition` pero todo en mayúsculas (EX: `technology` como filtro tendría que ser `TECHNOLOGY`, `isEngine` como `ISENGINE`, `data_sensitivity` como `DATA_SENSITIVITY`, etc.).
  - Los Tipos (**Type**) posibles son:
    - `INPUT_CHECKBOX` para multi-selección
    - `INPUT_DATE` para introducir fechas en calendario

- INPUT\_TEXT para la especificación de un texto
    - SELECT para selección única
    - SELECT\_USERS para indicar que es un selector de usuarios cuyos nombres no se deben de traducir.
  - **Page\_Name** queda vacío. No es un campo utilizado para los filtros de “kerno”
- En las pantallas de histórico y auditoría
  - **Collection** será “audit\_logs”
  - Los campos (**Name**) posibles son:
    - userName
    - action
    - objectType
    - objectSubtype
    - objectId
    - objectName
  - Los Tipos (**Type**) posibles son:
    - INPUT\_TEXT para objectId y objectName
    - SELECT para el resto
  - Page\_Name puede ser:
    - AUDIT\_ALL: auditoría completa
    - AUDIT\_OBJECT: auditoría por objeto
    - AUDIT\_USER: auditoría de usuario
    - HISTORIC\_ALL: histórico completo
    - HISTORIC\_OBJECT: histórico por objeto
    - HISTORIC\_USER: histórico de usuario

## Visión de Administrador

El alta de un nuevo filtro en el panel de administración de Anjana Data se realiza en **Filter Confs**:



Al acceder se muestra una tabla que contiene todos los filtros existentes en la configuración actual.

La creación de un nuevo filtro se realiza mediante el botón **New**:

Filter Confs										+ New
Id	Name	Label	Type	collection	pageName	groupable	main	position	sortable	
34	STATE	FILTERS.STATE	INPUT_CHECKBOX	kerno	[NULL]	✓	✓	4	✓	✓
35	OBJ_TYPE	FILTERS.OBJ_TYPE	INPUT_CHECKBOX	kerno	[NULL]	✓	✓	1	✓	✓
36	ORGANIZATIONAL_UNIT	FILTERS.CONTAINER	INPUT_CHECKBOX	kerno	[NULL]	✓	✓	3	✓	✓
37	OBJ_SUB_TYPE	FILTERS.OBJ_SUB_TYPE	INPUT_CHECKBOX	kerno	[NULL]	✓	✓	2	✓	✓
38	CREATED_USER	FILTERS.CREATED_USER	SELECT_USERS	kerno	[NULL]	✓	✓	5	✓	✓
40	ORGANIZATIONAL_UNIT	FILTERS.BUSINESS_NODE	INPUT_CHECKBOX	kerno	[NULL]	✓	✓	2	✓	✓
43	START_DATE	FILTERS.START_DATE	INPUT_DATE	kerno	[NULL]	✓	✓	5	✓	✓
44	VALIDITY_DATE	FILTERS.VALIDITY_DATE	INPUT_DATE	kerno	[NULL]	✓	✓	6	✓	✓
45	LAST_MODIFIED_DATE	FILTERS.LAST_MODIFIED_DATE	INPUT_DATE	kerno	[NULL]	✓	✓	7	✓	✓
49	TECHNOLOGY	FILTERS.TECHNOLOGY	INPUT_CHECKBOX	kerno	[NULL]	✓	✗	9	✓	✓
50	INFRASTRUCTURE	FILTERS.INFRASTRUCTURE	INPUT_CHECKBOX	kerno	[NULL]	✓	✗	8	✓	✓

A continuación, se muestra cómo crear un filtro:

### Filter Confs Edit

Name userName	Label FILTERS.USER	Type SELECT
collection Audit logs	pageName HISTORIC_ALL	position 1
<input checked="" type="checkbox"/> groupable	<input checked="" type="checkbox"/> main	<input checked="" type="checkbox"/> sortable

✗ Cancel
✓ Save

### Visión de Desarrollador

Para definir los filtros hay que configurar la tabla **filter\_conf** del esquema Minerva. Para ello, hay que rellenar un sql como el siguiente:

```

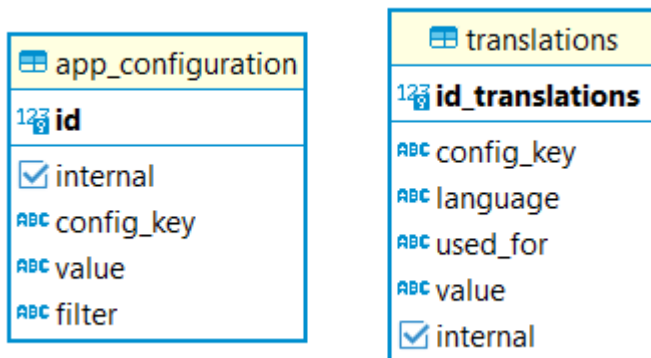
INSERT INTO minerva.filter_conf
(id, collection, groupable, "label", main, filter_name, "position", sortable, "type", page_name)
VALUES
(107, 'audit_logs', false, 'FILTERS.OBJ_TYPE', true, 'objectType', 4, false, 'SELECT', 'HISTORIC_ALL'),
(118, 'audit_logs', false, 'FILTERS.OBJ_SUB_TYPE', true, 'objectSubType', 4, false, 'SELECT',
'HISTORIC_USER'),
(115, 'audit_logs', false, 'FILTERS.OBJECT_ID', true, 'objectId', 6, false, 'INPUT_TEXT', 'AUDIT_ALL'),
(120, 'audit_logs', false, 'FILTERS.OBJECT_ID', true, 'objectId', 6, false, 'INPUT_TEXT',
'HISTORIC_USER'),

```

```
(38, 'kerno', true, 'FILTERS.CREATED_USER', true, 'CREATED_USER', 5, true, 'SELECT_USERS', NULL),
(35, 'kerno', true, 'FILTERS.OBJ_TYPE', true, 'OBJ_TYPE', 1, true, 'INPUT_CHECKBOX', NULL);
```

## Esquema Portuno de BD

El modelo de base de datos se muestra en la siguiente imagen:



### 1. Variables del sistema

Se pueden introducir en la tabla **app\_configuration** las variables del sistema, estas variables son cargadas por cada módulo (Kerno, Hermes, Minerva...) en el arranque.

Las variables del sistema permiten identificar, por ejemplo, caracteres como separadores de valores o ARIs, valores máximos o mínimos para las validaciones de atributos de plantilla, la periodicidad de ejecución de los procesos batch.

#### Estructura de la tabla

Cada variable registrada se caracteriza por los siguientes elementos:

- **id:** (PK) identificador único, sirve para gestión interna.
- **internal:** boolean utilizado para indicar si una propiedad se puede ver desde la API pública en claro (si internal es 'false' no debe verse en claro).
- **config\_key:** clave de la configuración.
- **value:** el valor que tiene la propiedad.
- **filter:** módulo al que aplica la propiedad (con valor nulo, aplicaría a todos los módulos).



Es necesario que dicha tabla tenga las siguientes propiedades, siendo la columna **config\_key** la misma que la tabla, **value** el valor por defecto que Anjana proporciona en la columna value, **filter** el mismo valor que filter en la tabla y **descripción** que no existe en la tabla y es una mera explicación de cada propiedad:

config_key	value	filter	descripción
ARI_OBJECT_SEPARATOR	:		Carácter usado para separar palabras claves en el nombre de la ARI de archivos subidos a minio
DEFAULT_LANGUAGE_APP	en-US		Idioma por defecto de la aplicación, valor del campo i18n_code de la tabla zeus.languages
LIMIT_NOTIFICATIONS	30		Número máximo de notificaciones mostradas por página
MAX_DECIMAL_PRECISION	16		Número máximo de dígitos en la parte decimal de un input_decimal (o array) permitido
MAX_GRAPH_NODE			Número máximo de nodos en el grafo de linaje
MAX_INTEGER	2100000000		Número máximo en atributo de tipo entero, array o rango
MAX_LENGTH_TEXT_BD	255		Número máximo de caracteres permitidos en base de datos para campos de tipo input_text, text_area, enriched_textarea y multi idiomas de los tres, como en la validación de un workflow o la solicitud de adherencia
MAX_LENGTH_TEXT_EDITOR	300000		Número máximo de caracteres para un atributo de tipo text_area, enriched_textarea y multi idiomas de ambos permitido
MIN_DECIMAL_PRECISION	0		Número mínimo de dígitos en la parte decimal de un input_decimal (o array) permitido
MIN_INTEGER	-2100000000		Número mínimo en atributo de tipo entero, array o rango
MIN_LENGTH_TEXT_EDITOR	1		Número mínimo de caracteres para un atributo de tipo input_text, text_area, enriched_textarea y multi idiomas de los tres permitido
SHOW_RESULTS_SOLR	20		Número de resultados mostrados en búsqueda de Solr
default_language_app	en-US		Idioma por defecto de anjana debe coincidir con DEFAULT_LANGUAGE_APP
attribute_restriction_inactive	external		
anjana.combinations.max	10		Máximo número de combinaciones posibles en las adherencias

config_key	value	filter	descripción
anjana.activiti.data	true	hermes	Flag para importar los WFs en el arrancado de Hermes
anjana.activiti.input	DC, BG	hermes	Carpetas a usar para poder importar WFs
anjana.activiti.output	/data/WF	hermes	Ruta temporal del directorio de ficheros para tratar los WFS
anjana.async.pool.corePoolSize	5	hermes	Tamaño del pool asíncrono de procesamiento de Hermes
anjana.async.pool.maxPoolSize	20	hermes	Tamaño máximo del pool asíncrono de procesamiento de Hermes
anjana.async.pool.queueCapacity	100	hermes	Tamaño de cola de procesamiento de Hermes
anjana.last-execution.maxRetry	0	hermes	Cantidad de reintentos en los pasos LastExecution de los WFS, siendo 0 intentos infinitos
anjana.last-execution.waitRetry	300	hermes	Tiempo de espera entre intentos en los pasos LastExecution de los WFs
wf.async.waiter	20000	hermes	Tiempo de espera en ms de tareas internas de Activiti
wf.validation.automatic	true	hermes	Flag para habilitar tareas de validación automática sin petición de acción humana, no confundir con
anjana.async.db.corePoolSize	5	kerno	Tamaño del pool de conexiones a base de datos
anjana.async.db.maxPoolSize	20	kerno	Tamaño máximo del pool de conexiones a base de datos
anjana.async.db.queueCapacity	100	kerno	Tamaño de cola de las ejecuciones a base de datos
anjana.async.hermes.corePoolSize	5	kerno	Tamaño del pool de hilos contra ejecuciones de Hermes
anjana.async.hermes.maxPoolSize	10	kerno	Tamaño máximo del pool de hilos contra ejecuciones de Hermes
anjana.async.hermes.queueCapacity	1000	kerno	Tamaño de la cola de las ejecuciones contra Hermes
anjana.async.minerva.corePoolSize	5	kerno	Tamaño del pool de hilos contra ejecuciones de Minerva
anjana.async.minerva.maxPoolSize	20	kerno	Tamaño máximo del pool de hilos contra ejecuciones de Minerva

config_key	value	filter	descripción
anjana.async.minerva.queueCapacity	100	kerno	Tamaño de la cola de las ejecuciones contra Minerva
anjana.combinations.max	10	kerno	Cantidad de combinaciones a mostrar como máximo para el carrito de adherencias
anjana.internationalization.separatorNameColumnsExcel	_	kerno	Carácter para utilizar como separador de valores en atributos multivaluados en Excel
anjana.notification.expiration	4	kerno	Número de días de preaviso para las notificaciones indicando la próxima expiración de un objeto
anjana.numberDatasetFieldsForWarning	100	kerno	Cantidad de dataset_fields para considerar un Dataset grande en el que las acciones efectuadas por los usuarios les redirigirán al Portal en vez de dejarles a la espera de la finalización
anjana.numberEntitiesToSubmit	50	kerno	Cantidad de entidades a enviar a validar a la vez en la importación por Excel
anjana.pagination.defaultPage	0	kerno	Página por defecto en la paginación de objetos en la vista de objeto
anjana.pagination.defaultSize	30	kerno	Tamaño por defecto de las páginas en la paginación de objetos en la vista de objeto
anjana.retries	5	kerno	Reintentos en caso de fallo de comunicación en la reindexación
anjana.scheduling.enabled	true	kerno	Flag para habilitar los batches de Kerno
anjana.scheduling.expiration	0 0 0 ? * *	kerno	Programación de la expiración de objetos (todos los días a las 00.00h)
anjana.scheduling.lineageObjectsPerPacket	1000	kerno	Cantidad de objetos por paquete para procesar el linaje
anjana.scheduling.notification	0 45 7 ? * *	kerno	Programación de la notificación de entidades a expirar (todos los días a las 7.45h)
anjana.scheduling.objectsPerPacket	500	kerno	Cantidad de objetos a procesar por paquete en procesamientos internos de Anjana
anjana.scheduling.threadScheduleSeconds	15	kerno	Retraso en segundos entre paquetes de indexación
anjana.scheduling.threads	2	kerno	Cantidad de hilos para batches

config_key	value	filter	descripción
anjana.separatorAttribute	_-	kerno	Carácter utilizado como separador de atributos multivaluados en base de datos
anjana.tot.createCustom	false	kerno	Flag para que, en caso de que la extracción de metadato con un plugin obtenga un atributo que no se encuentre en plantilla, se cree como atributo custom del objeto
anjana.waitRetry	10	kerno	Tiempo de espera entre cada intento en caso de fallo de comunicación en la reindexación
anjana.workflow.launchIfCustomChanges	true	kerno	Flag para incluir los custom al tratamiento de cambios de un objeto al lanzar el workflow
solr.module.audit.collection	audit_logs	minerva	Nombre de la colección de Solr para auditoría e histórico
solr.module.kerno.collection	kerno	minerva	Nombre de la colección de Solr para objetos de Anjana
solr.module.kerno.generate	true	minerva	Flag para crear la colección de objetos en Solr, si no existe, en el arrancado
solr.module.kerno.replication	1	minerva	Replicación SolR de la colección de objetos
solr.module.kerno.shards	1	minerva	Sharding SolR de la colección de objetos
solr.module.recommendations.collection	recommendations	minerva	Nombre de la colección de Solr para píldoras asistenciales de Anjana
solr.module.recommendations.generate	true	minerva	Flag para crear la colección de recomendaciones en Solr, si no existe, en el arrancado
solr.module.recommendations.replication	1	minerva	Replicación SolR de la colección de recomendaciones
solr.module.recommendations.shards	1	minerva	Sharding SolR de la colección de recomendaciones
solr.module.snapshots.collection	snapshots	minerva	nombre de la colección de Solr para snapshots de Anjana
solr.multivalued-separator	_-	minerva	Carácter que se utiliza para concatenar atributos multivaluados que debe coincidir con anjana.separatorAttribute de Kerno
anjana.group.crossName	Cross	zeus	Nombre del grupo que aglutina los roles cross en el perfil de usuario

config_key	value	filter	descripción
anjana.mail.aws[0]	email	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]' (revisar las propiedades anjana.mail.azure[x] o anjana.mail.google[x])
anjana.mail.azure[0]	mail	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.azure[1]	otherMails	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.azure[2]	userPrincipalName	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.db[0]	email	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.google[0]	primaryEmail	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.google[1]	emails	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.mail.ldap[0]	mail	zeus	Propiedad del proveedor de donde sacar el email, se pueden tener varios numerando las propiedades con '[numero]'
anjana.provider.primary	Database	zeus	Propiedad para marcar si se quiere tener un proveedor principal preseleccionado en las pantallas de login y cuál debe ser. Este valor debe coincidir con la propiedad security.authentication.<proveedor> y el nombre de la instancia del proveedor elegido. Para base de datos y Ldap los valores son 'Database' y 'Ldap' correspondientemente.
anjana.scheduling.licenseNotification	0 0 0 ? * *	zeus	Programación de la tarea programada de aviso de caducidad de la licencia (cada día a las 00.00)

config_key	value	filter	descripción
anjana.scheduling.notification	0 0 0 ? * *	zeus	Programación de la tarea programada de revisión de roles cross

Adicionalmente, hay casos en los que, debido a limitaciones de la base de datos o el propio funcionamiento del atributo, es necesario indicar un límite en algunos componentes y priorizar el valor configurado de la tabla appConfigurations ante la configurada en templateAtributtesValidations.

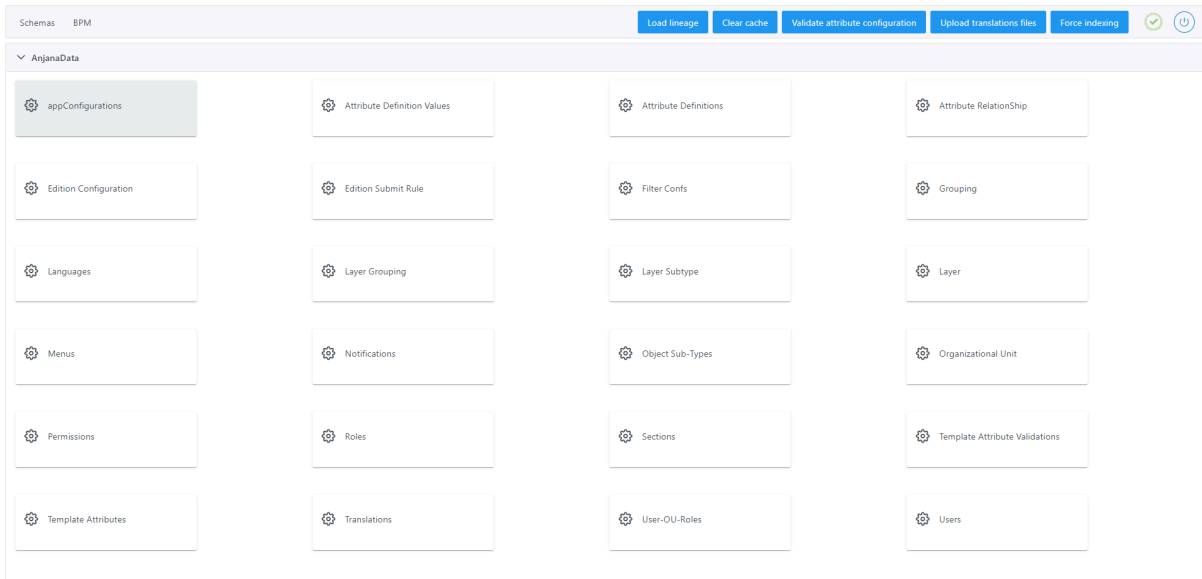
Si el valor de templateAtributtesValidations es mayor al valor que hay en appConfigurations, se priorizará el valor de appConfigurations.

Los casos son los siguientes:

- MAX -> Si el MAX configurado en template attribute validations es superior a MAX\_INTEGER de appConfigurations, se priorizará MAX\_INTEGER de appConfigurations.
- MIN -> Si el MIN configurado en template attribute validations es inferior a MIN\_INTEGER o MIN\_RANGE de appConfigurations, se priorizará MIN\_INTEGER de appConfigurations o MIN\_RANGE si es un input range.
- MAX\_DECIMAL\_PRECISION -> Si el MAX\_DECIMAL\_PRECISION configurado en template attribute validations es superior a MAX\_DECIMAL\_PRECISION de appConfigurations, se priorizará MAX\_DECIMAL\_PRECISION de appConfigurations.
- MIN\_DECIMAL\_PRECISION -> Si el MIN\_DECIMAL\_PRECISION configurado en template attribute validations es inferior a MIN\_DECIMAL\_PRECISION de appConfigurations, se priorizará MIN\_DECIMAL\_PRECISION de appConfigurations.
- MAX\_LENGTH-> Si el MAX\_LENGTH configurado en template attribute validations es superior a MAX\_LENGTH\_TEXT\_EDITOR o MAX\_LENGTH\_TEXT\_BD de appConfigurations, se priorizará MAX\_LENGTH\_TEXT\_EDITOR en el caso de los TEXTAREA o MAX\_LENGTH\_TEXT\_BD si es un input text o los textarea de los modales de workflows y adherencias.
- MIN\_LENGTH -> Si el MIN\_LENGTH configurado en template attribute validations es inferior a MIN\_LENGTH\_TEXT\_EDITOR de appConfigurations, se priorizará MIN\_LENGTH\_TEXT\_EDITOR de appConfigurations.

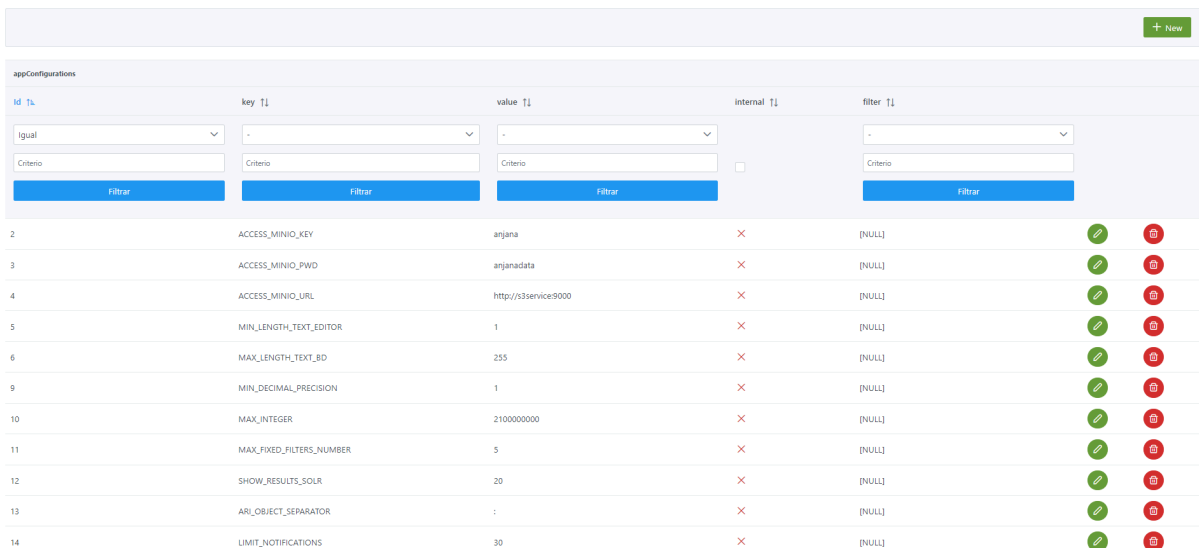
### Visión de Administrador

El alta de una nueva variable desde el panel de administración de Anjana Data se realiza en **appConfigurations**:



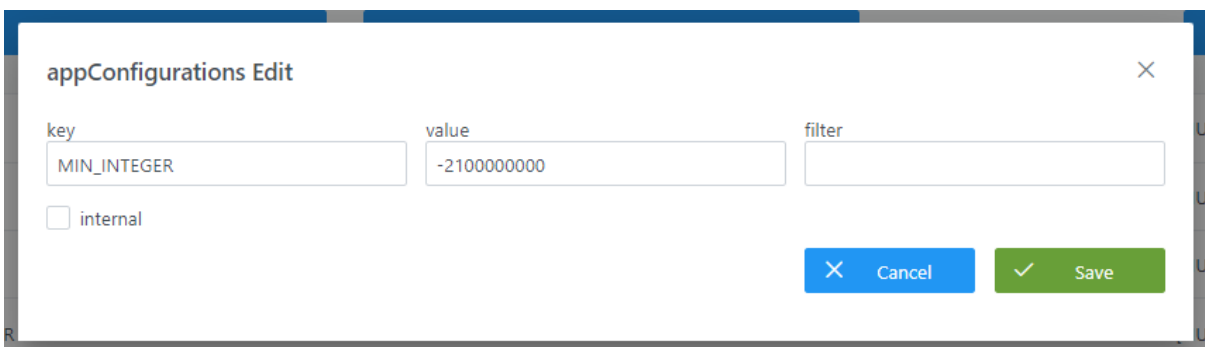
Al acceder se muestra una tabla que contiene todas las variables de la aplicación de la configuración actual.

La creación de una nueva variable se realiza mediante el botón **New**:



id	key	value	internal	filter		
2	ACCESS_MINIO_KEY	anjana	×	[NULL]	✓	🗑️
3	ACCESS_MINIO_PWD	anjanadata	×	[NULL]	✓	🗑️
4	ACCESS_MINIO_URL	http://s3service9000	×	[NULL]	✓	🗑️
5	MIN_LENGTH_TEXT_EDITOR	1	×	[NULL]	✓	🗑️
6	MAX_LENGTH_TEXT_BD	255	×	[NULL]	✓	🗑️
9	MIN_DECIMAL_PRECISION	1	×	[NULL]	✓	🗑️
10	MAX_INTEGER	2100000000	×	[NULL]	✓	🗑️
11	MAX_FIXED_FILTERS_NUMBER	5	×	[NULL]	✓	🗑️
12	SHOW_RESULTS_SIZE	20	×	[NULL]	✓	🗑️
13	ARI_OBJECT_SEPARATOR	:	×	[NULL]	✓	🗑️
14	LIMIT_NOTIFICATIONS	30	×	[NULL]	✓	🗑️

A continuación, se muestra cómo crear un parámetro de configuración:



**appConfigurations Edit** ✕

key:  value:  filter:

internal

## Visión de Desarrollador

Para añadir las variables de configuración hay que configurar la tabla **app\_configurations** del esquema Portuno. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO portuno.app_configuration (id, internal, config_key, value, "filter") VALUES
(4, false, 'ACCESS_MINIO_URL', 'http://s3service:9000', NULL),
(5, false, 'MIN_LENGTH_TEXT_EDITOR', '1', NULL),
(6, false, 'MAX_LENGTH_TEXT_BD', '255', NULL),
(9, false, 'MIN_DECIMAL_PRECISION', '1', NULL),
(10, false, 'MAX_INTEGER', '2100000000', NULL);
```

### NOTAS:

- Puesto que las variables se cargan en el arranque, Portuno debe estar arrancado antes que el resto de servicios para obtener correctamente las propiedades indicadas. Si algún servicio se iniciara antes que Portuno se quedará a la espera de que Portuno esté disponible para poder cargar esta configuración y terminar su inicialización.

## 2. Traducciones

Las traducciones disponibles para el portal en los distintos idiomas configurados en Anjana se encuentran almacenadas en la tabla **translations**. Cuando se añadan nuevos textos o se incorpore un nuevo idioma a la aplicación habrá que incorporar aquí esas nuevas traducciones.

Cualquier cambio en esta tabla requiere la actualización de los ficheros de traducciones desde Portuno para poder hacer uso de los nuevos textos en la aplicación.



### Estructura de la tabla

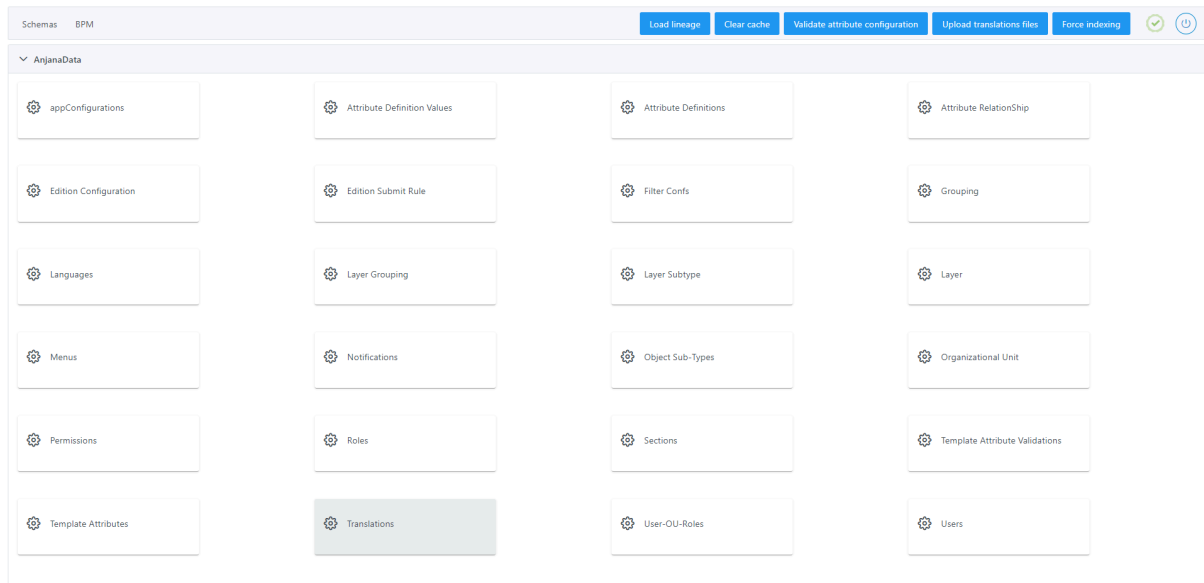
Cada traducción registrada se caracteriza por los siguientes elementos:

- **id\_translations**: identificador único, sirve para gestión interna.
- **config\_key**: clave de traducción. Cada clave debería aparecer en esta tabla tantas veces como idiomas haya configurados.
- **language**: idioma de traducción en código i18n.
- **value**: texto traducido al idioma indicado.
- **used\_for**: texto que permite segregar las traducciones por uso. Es informativo.
- **internal**: boolean que permite indicar si la traducción se corresponde con un texto propio de la aplicación (`internal = "true"`) o si, por el contrario, se corresponde con una traducción propia de la configuración utilizada como nombres de atributos, roles... (`internal = "false"`).



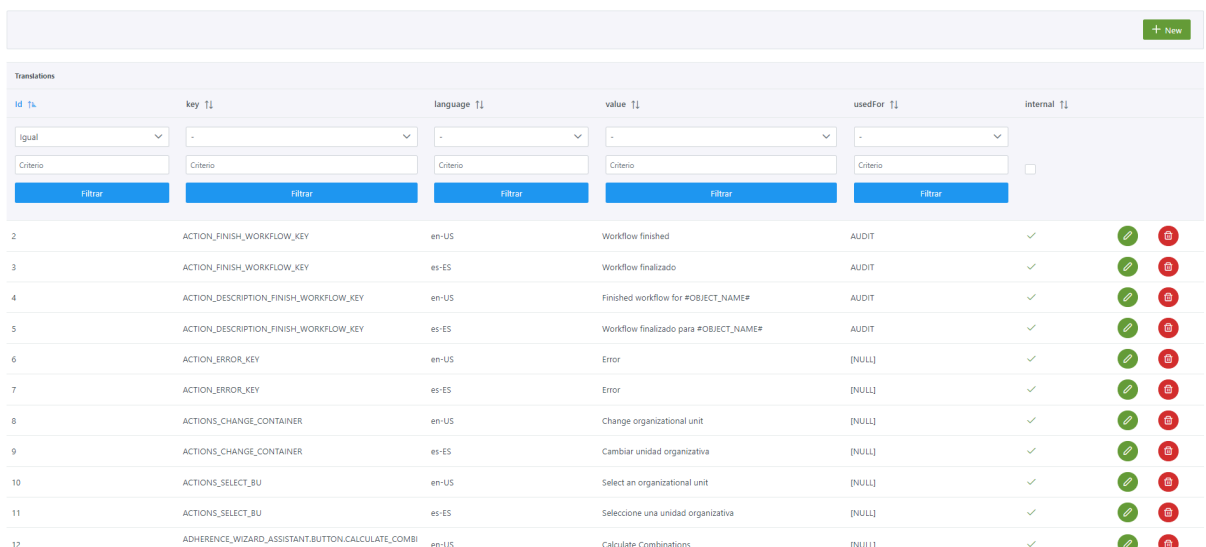
## Visión de Administrador

El alta de las traducciones se realiza en el panel de administración de Anjana Data en **Translations**:



Al acceder se muestra una tabla que contiene todas las traducciones usadas por el portal de Anjana.

La creación de una nueva traducción se realiza mediante el botón **New**:



Id	key	language	value	usedFor	internal
2	ACTION_FINISH_WORKFLOW_KEY	en-US	Workflow finished	AUDIT	✓
3	ACTION_FINISH_WORKFLOW_KEY	es-ES	Workflow finalizado	AUDIT	✓
4	ACTION_DESCRIPTION_FINISH_WORKFLOW_KEY	en-US	Finished workflow for #OBJECT_NAME#	AUDIT	✓
5	ACTION_DESCRIPTION_FINISH_WORKFLOW_KEY	es-ES	Workflow finalizado para #OBJECT_NAME#	AUDIT	✓
6	ACTION_ERROR_KEY	en-US	Error	[NULL]	✓
7	ACTION_ERROR_KEY	es-ES	Error	[NULL]	✓
8	ACTIONS_CHANGE_CONTAINER	en-US	Change organizational unit	[NULL]	✓
9	ACTIONS_CHANGE_CONTAINER	es-ES	Cambiar unidad organizativa	[NULL]	✓
10	ACTIONS_SELECT_BU	en-US	Select an organizational unit	[NULL]	✓
11	ACTIONS_SELECT_BU	es-ES	Seleccione una unidad organizativa	[NULL]	✓
12	ADHERENCE_WIZARD_ASSISTANT.BUTTON.CALCULATE_COMBI	en-US	Calcular Combinations	[NULL]	✓

Mediante el wizard de creación se asignan valores a los elementos anteriormente descritos:

### Translations Edit ✕

key ACTION_DESCRIPTION_FINISH_WORKFLC	language en-US	usedFor AUDIT
--	-------------------	------------------

internal

value

Finished workflow for #OBJECT\_NAME#

✕ Cancel
✓ Save

### Visión del Desarrollador

Para añadir las traducciones hay que configurar la tabla **translations** del esquema Portuno. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO portuno.translations (id_translations, config_key, "language", used_for, value,
internal) VALUES
(2, 'ACTION_FINISH_WORKFLOW_KEY', 'en-US', 'AUDIT', 'Workflow finished', true),
(3, 'ACTION_FINISH_WORKFLOW_KEY', 'es-ES', 'AUDIT', 'Workflow finalizado', true),
(4, 'ACTION_DESCRIPTION_FINISH_WORKFLOW_KEY', 'en-US', 'AUDIT', 'Finished workflow for
#OBJECT_NAME#', true),
(5, 'ACTION_DESCRIPTION_FINISH_WORKFLOW_KEY', 'es-ES', 'AUDIT', 'Workflow finalizado para
#OBJECT_NAME#', true),
(6, 'ACTION_ERROR_KEY', 'en-US', NULL, 'Error', true),
(7, 'ACTION_ERROR_KEY', 'es-ES', NULL, 'Error', true);
```

#### NOTAS:

- Es necesario que se configuren las traducciones siguiendo este criterio:
  - *internal="true"*: traducciones internas de Anjana, necesarias para la correcta visualización de la aplicación
  - *internal="false"*: traducciones de textos propios de la configuración, es decir, textos relacionados con atributos de las plantillas, menús, secciones, roles, OUs... que son configurables y varían en función del entorno y su parametría.
- No se debe añadir en la tabla de portuno.translations dos claves de traducción distintas con el mismo texto para traducir dos valores de un atributo.
- No es recomendable establecer como clave de traducción los nombres de los diferentes subtipos de objetos puesto que supondrá que estos aparezcan traducidos en el portal a pesar de que no tienen traducción.

- Para los atributos Infraestructura, Tecnología y Zona:
  - Todas las claves de traducción de los valores posibles de dichos campos deben ser iguales que las distintas partes que componen la tripleta con la que se configura la conexión con los plugins de Tot. EX: Si una tripleta en el plugin de tot de azure se configura como azure/azure analytics/production entonces las clave de traducción del valor azure en infraestructura debe ser “azure”, para el valor de azure analytics en technology debe ser “azure analytics” y para el valor production de zone debe ser “production”.
  - Los valores de estos atributos deben no contener ‘:’, ‘#’, ‘(’ o ‘)’ para no interferir con los identificadores internos de Anjana.
- Para las traducciones de los roles, la clave de traducción ha de ser el texto “ROLE.” concatenado del nombre del rol establecido en la tabla zeus.role en la columna role\_name. Por ejemplo: ROLE.developer
- Para las traducciones de los nombres de los idiomas, la clave de traducción ha de ser el código i18n del idioma en la tabla [languages](#).

### Traducciones internas de Anjana

Hay valores de traducciones internas que se repiten en diferentes claves, ya que estas se han generado según la funcionalidad o pantalla en la que se encuentran. En caso de querer cambiar una traducción interna de Anjana se tendrá que tener en cuenta que habrá que cambiar el valor de varias claves en la tabla de Translations de Portuno.

Por ejemplo, en caso de querer cambiar la traducción de un atributo obligatorio de una entidad como Infraestructura, habrá que cambiar tanto el valor de la clave de Infraestructura del wizard de creación como la de los filtros, el atributo de la plantilla o el del tooltip de la cabecera de un objeto.

A continuación se desglosan las claves de traducción de los atributos obligatorios de las plantillas de entidades/relaciones:

#### Nombre:

- COMMON.NAME
- NEW\_WIZARD\_ASSISTANT.CARD\_NAME
- labelTranslationKey y placeholderTranslationKey del atributo con name = ‘name’ en anjana.attribute\_definition

#### Infraestructura:

- NEW\_WIZARD\_ASSISTANT.INFRASTRUCTURE
- NEW\_WIZARD\_ASSISTANT.METADATA\_AUTO.INFRASTRUCTURE
- TOOLTIP.INFRASTRUCTURE
- FILTERS.INFRASTRUCTURE
- labelTranslationKey y placeholderTranslationKey del atributo con name = ‘infraestructura’ en anjana.attribute\_definition

**Tecnología:**

- NEW\_WIZARD\_ASSISTANT-METADATA\_AUTO.TECHNOLOGY
- NEW\_WIZARD\_ASSISTANT.TECHNOLOGY
- TOOLTIP.TECHNOLOGY
- FILTERS.TECHNOLOGY
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'technology' en anjana.attribute\_definition

**Zona:**

- NEW\_WIZARD\_ASSISTANT.ZONE
- TOOLTIP.ZONE
- FILTERS.ZONE
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'zone' en anjana.attribute\_definition

**Localización:**

- NEW\_WIZARD\_ASSISTANT.PATH
- FILTERS.LOCATION
- TOOLTIP.PATH
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'path' en anjana.attribute\_definition

**Unidad Organizativa:**

- CONTAINER
- GRAPH.NODE\_ORGANIZATIONAL\_UNIT
- MODAL.ORGANIZATIONAL\_UNIT
- TOOLTIP.ORGANIZATIONAL\_UNIT
- FILTERS.ORGANIZATIONAL\_UNIT
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'organizationalUnit' en anjana.attribute\_definition

**Posición:**

- COMMON.POSITION
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'position' en anjana.attribute\_definition

**Nombre físico:**

- COMMON.PHYSICAL\_NAME

- labelTranslationKey y placeholderTranslationKey del atributo con name = 'physicalName' en anjana.attribute\_definition

**Origen:**

- COMMON.SOURCE
- FORM\_RELATION.SOURCE
- RELATION.SOURCE
- FILTERS.SOURCE
- SOURCE
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'source' en anjana.attribute\_definition

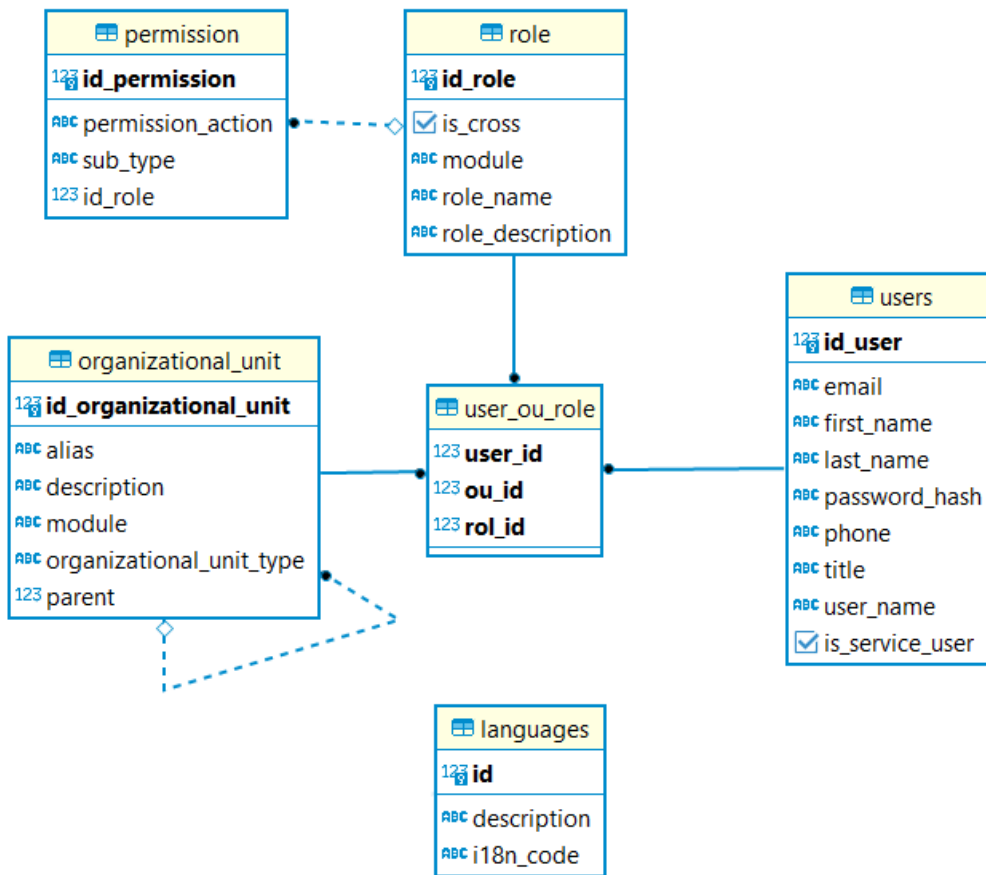
**Destino:**

- COMMON.DESTINATION
- FORM\_RELATION.DESTINATION
- RELATION.DESTINATION
- FILTERS.DESTINATION
- DESTINATION
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'destination' en anjana.attribute\_definition

## Esquema Zeus de BD

La configuración de las tablas del esquema Zeus permite definir el modelo de gobierno a través de la definición de la jerarquía de unidades organizativas así como los roles y los permisos que estos tendrán en la aplicación.

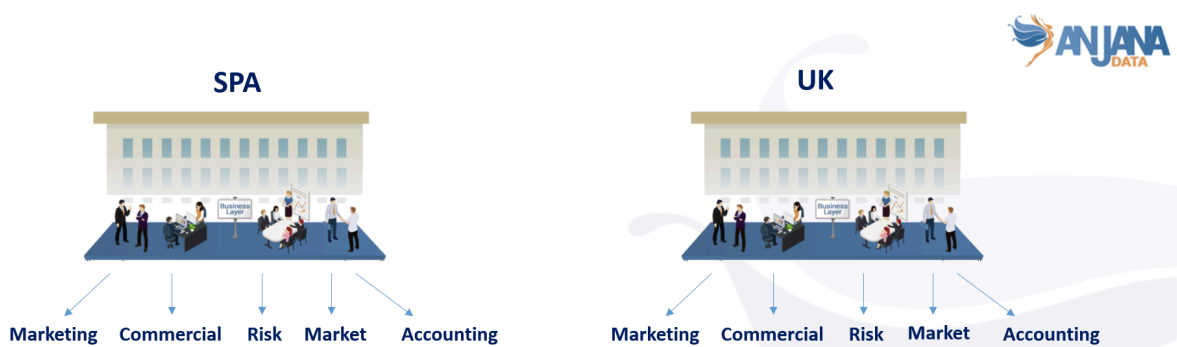
El modelo de base de datos se muestra en la siguiente imagen:



## 1. Definir la estructura organizativa y las unidades organizacionales

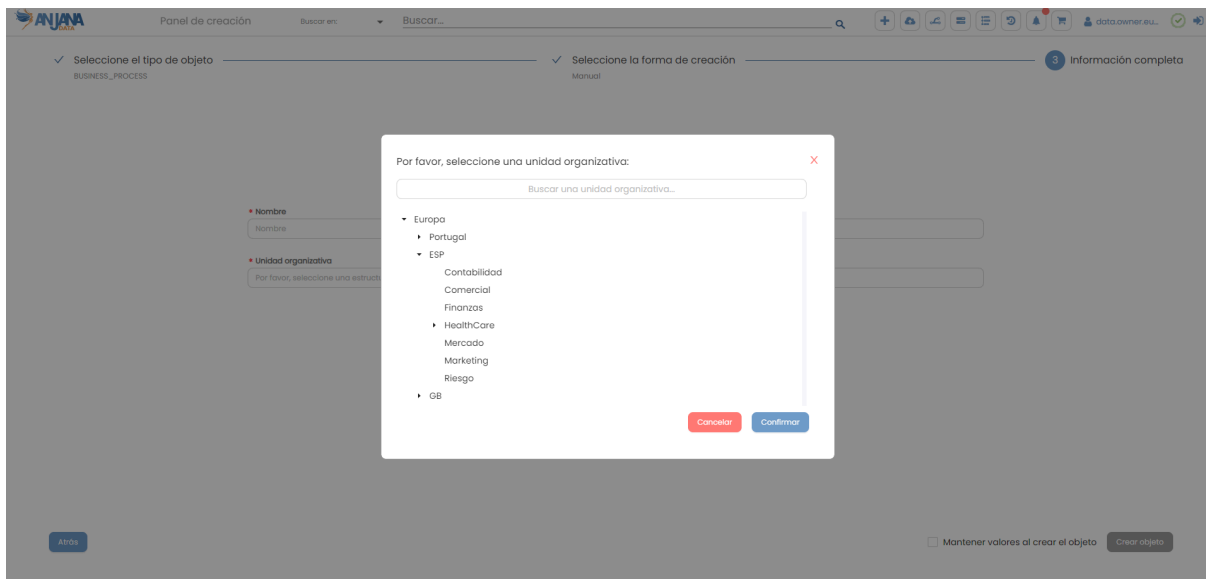
El primer paso para aterrizar el modelo de gobierno de la organización es establecer la jerarquía de unidades organizativas. Las Unidades Organizativas representan dominios de datos que permiten establecer custodia y responsabilidad sobre los mismos.

Puede tratarse de una estructura “plana” o de un único nivel o, por el contrario, de una jerarquía de varios niveles.



Una vez definidos los niveles de la jerarquía de unidades organizacionales, se deben asignar nombres a cada uno de los niveles conforme al modelo de gobierno de la organización.

Las unidades organizacionales se configuran en la tabla **organizational\_unit** y serán visibles dentro de Anjana Data en las pantallas de creación de objetos:



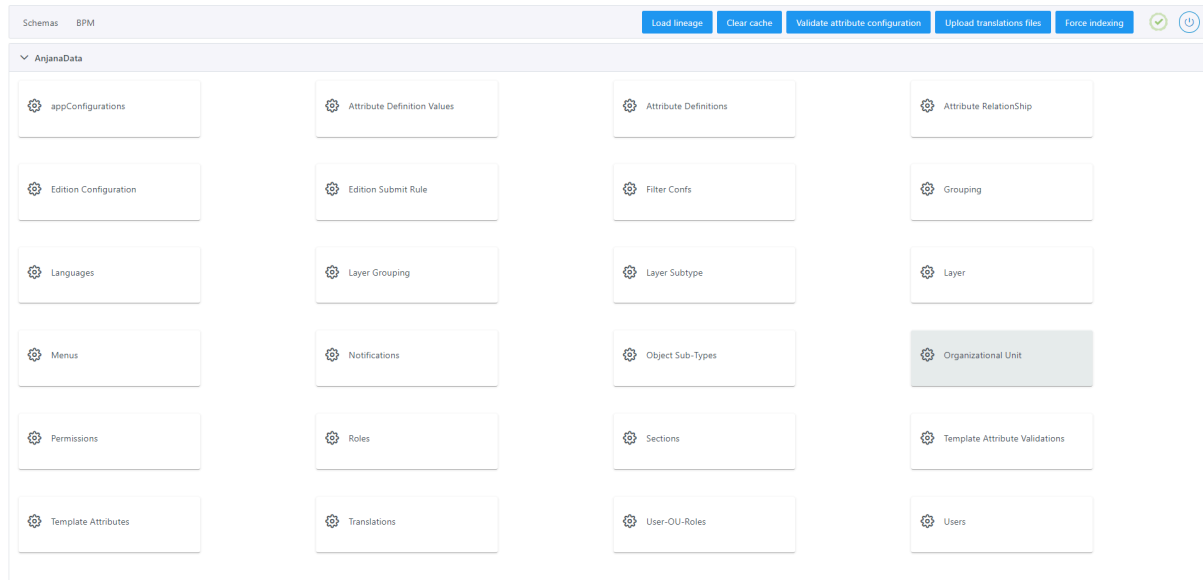
## Estructura de la tabla

Cada unidad organizativa registrada se caracteriza por los siguientes elementos:

- **id\_organizational\_unit:** identificador único de la unidad organizacional.
- **description:** descripción de la unidad organizacional.
- **module:** módulo al que pertenece, indicar “BG” para *Business Glossary*, “DC” para *Data Catalog* o “ALL” si aplica a todo. Actualmente este campo se encuentra sólo a nivel informativo.
- **alias:** alias de la unidad organizacional. Este alias debe coincidir exactamente con el nombre usado en los proveedores de identidades para la unidad.  
Dicho alias es la clave de traducción de la unidad organizativa, por lo que se puede añadir una traducción en la tabla translations de portuno. Para ello es importante considerar que el valor de la columna ‘key’ de la tabla translations de portuno debe coincidir con el valor de la columna ‘alias’ de la tabla organizational\_unit de zeus.  
NOTA: El separador de los distintos niveles que componen una unidad organizativa, en esta tabla, siempre será “/”. Por ello, en caso de jerarquía plana (un único nivel) se debe evitar usar “/” en el alias.  
En caso de utilizar otro separador de unidad organizativa en el proveedor de identidades utilizado para la autorización de los usuarios, es necesario detallarlo en la configuración como se indica en la documentación específica de las integraciones.
- **parent:** Indica la unidad organizacional padre. En caso de ser una jerarquía plana o de estar dando de alta a un nivel padre, este campo se dejará vacío.
- **organizationalUnitType:** Campo informativo que indica el nivel de jerarquía al que pertenece la unidad organizativa.

## Visión de Administrador

El alta de nuevas unidades organizacionales mediante el Panel de Administrador de Anjana Data se realiza en **Organizational Unit**:



Al acceder se muestra una tabla que contiene todas las unidades organizacionales existentes en la configuración actual.

La creación de una nueva unidad organizacional se realiza mediante el botón **New**:

+ New

Organizational Unit						
Id ↑↓	Description ↑↓	Module ↑↓	alias ↑↓	parent ↑↓	organizationalUnitType ↑↓	
-	-	Igual	-	Igual	-	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	
1	SPA	BG	SPA		1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Finance	BG	SPA/Finance	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Marketing	BG	SPA/Marketing	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Commercial	BG	SPA/Commercial	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Risk	BG	SPA/Risk	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	Market	BG	SPA/Market	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	Accounting	BG	SPA/Accounting	1-SPA	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
8	SCIB_HQ	BG	SCIB_HQ		1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Mediante el wizard de creación se elige el alias que se asigna a la unidad organizacional, el módulo (*Module*) al que pertenecerá y se seleccionará la unidad organizacional padre de la que cuelga mediante el campo parent. También se asignará el tipo de unidad dentro de nuestra jerarquía en el campo organizationalUnitType. A continuación, se muestra como crear la unidad organizacional “SPA/Market” que cuelga de la unidad padre “SPA”.



### Organizational Unit Edit ✕

Module

alias

parent

Description

Market

organizationalUnitType

2

### Visión de Desarrollador

Para definir unidades organizacionales hay que configurar la tabla **organizational\_unit**.



Para configurar las unidades organizacionales de la imagen hay que rellenar un sql como el siguiente:

```

INSERT INTO zeus.organizational_unit
(id_organizational_unit, alias, description, "module", organizational_unit_type, parent) VALUES
(100, 'Europe', 'SPA', 'ALL', '1', NULL),
(53, 'Europe/PT', 'Portugal', 'ALL', '1', 100),
(54, 'Europe/PT/Telco', 'Telco', 'ALL', '2', 53),
(1, 'Europe/SPA', 'SPA', 'ALL', '1', 100),
```

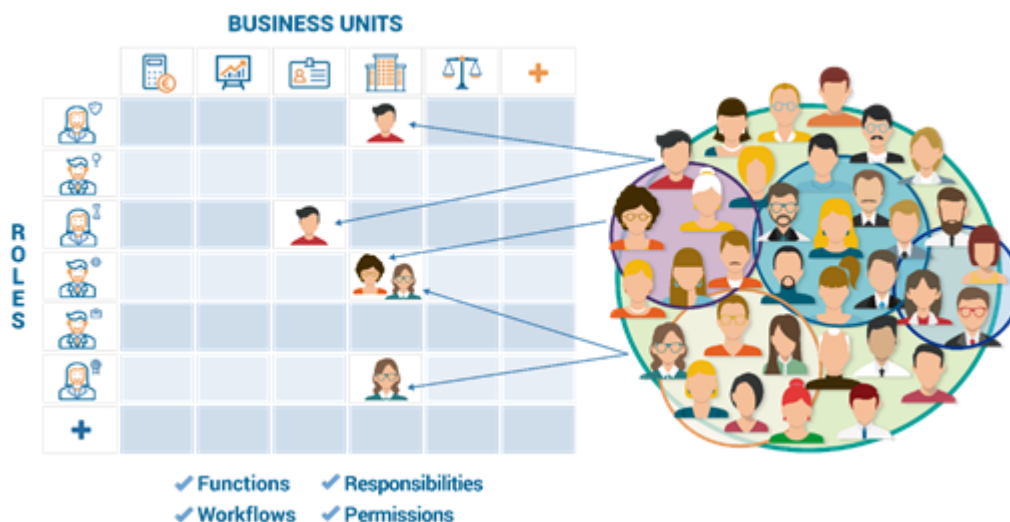
(7, 'Europe/SPA/Accounting', 'Accounting', 'ALL', '2', 1),  
 (4, 'Europe/SPA/Commercial', 'Commercial', 'ALL', '2', 1),  
 (2, 'Europe/SPA/Finance', 'Finance', 'ALL', '2', 1);

NOTAS:

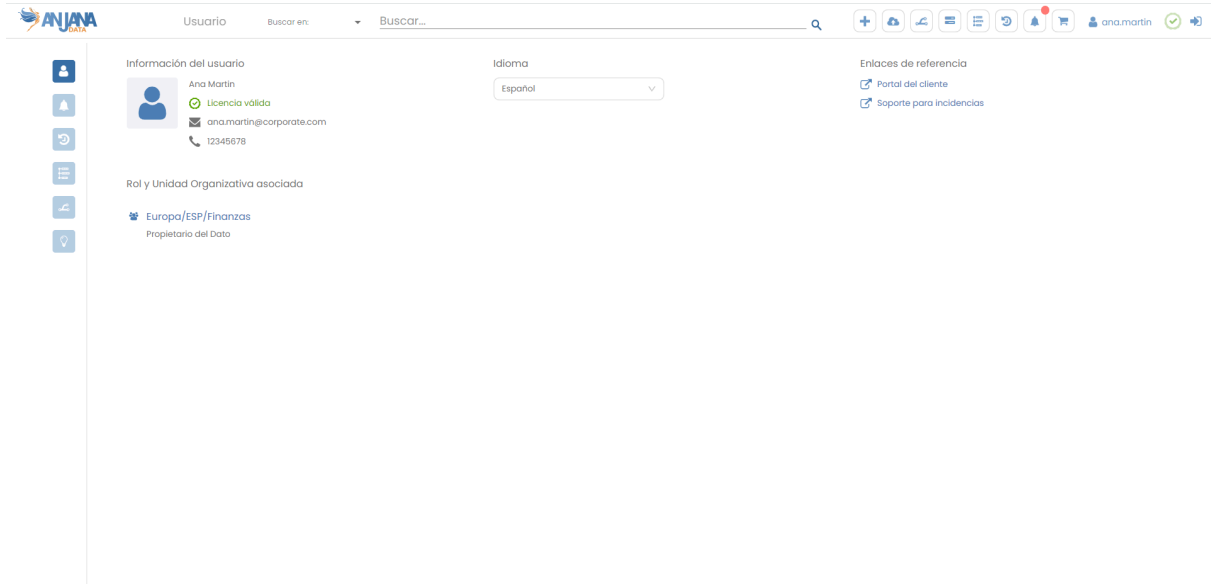
- El nombre y el alias de las unidades organizativas no deben contener los caracteres: “:” ni “#” para no interferir con los identificadores internos de Anjana.
- Cada una de las partes de las que se compone el alias de la unidad organizacional (concatenándolas con “/”) debe coincidir exactamente con el nombre usado en los proveedores de identidades para la unidad y será la clave de traducción en caso de querer traducir los nombres de las unidades a los idiomas de la aplicación. O, dicho de otro modo, el alias debe ser equivalente a la concatenación de las Unidades Organizativas padre desde el nodo raíz del árbol jerárquico hasta el nodo propio.
- Una vez creados objetos en Anjana, el cambio de las Unidades Organizativas o sus alias puede provocar que los objetos, permisos, workflows... asociados a las UO queden incoherentes.

## 2. Definir los roles del modelo de gobierno

Una vez definidas las unidades organizacionales, se deben definir los roles que intervienen en el gobierno del dato.



Dentro de Anjana Data, deben definirse roles para el Catálogo de Datos y para el Glosario de Negocio en la tabla **role**. Los roles pueden estar asociados a unidades organizativas específicas o tratarse de roles transversales.



The screenshot shows the user profile interface for 'Ana Martin'. At the top, there is a search bar and navigation icons. The profile information includes the user's name, a valid license status, email address (ana.martin@corporate.com), and phone number (12345678). The language is set to Spanish. Reference links for the client portal and incident support are provided. The user's role is identified as 'Europa/ESP/Finanzas' with the specific role of 'Propietario del Dato'.

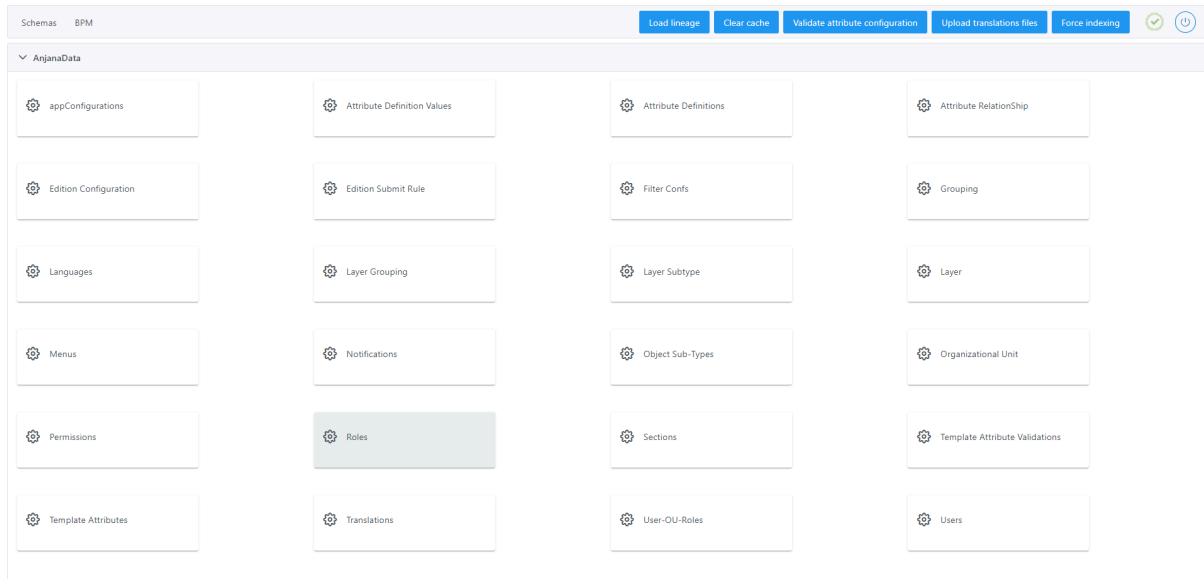
## Estructura de la tabla

Cada rol registrado se caracteriza por los siguientes elementos:

- **id:** Identificador único del rol.
- **module:** Módulo al que pertenece, indicar “BG” para *Business Glossary*, “DC” para *Data Catalog* o “ALL” si aplica a todo. Actualmente este campo se encuentra sólo a nivel informativo.
- **name:** Nombre asignado al rol. Este nombre debe ser idéntico al nombre usado para el rol en los proveedores de identidades. Además, puede ser traducido a los idiomas de la aplicación usando este mismo nombre como clave de traducción en `portuno.translations`.
- **description:** Descripción del rol.
- **isCross:** Flag para indicar si el rol es transversal (*cross*).

## Visión de Administrador

El alta de roles mediante el Panel de Administrador de Anjana Data se realiza en **Roles**:



Al acceder se muestra una tabla que contiene los roles existentes en la configuración actual.

La creación de un rol se realiza mediante el botón **New**:

+ New

Roles					
Id	Module	Name	Description	isCross	
-	Igual	-	-		
1	DC	developer	Developer	✓	
2	DC	architect	Architect	✓	
3	DC	administrator	Administrator	✓	
4	DC	data_steward	Data Steward	✓	
5	DC	data_owner	Data Owner	✗	
6	DC	compliance	Compliance	✓	
7	DC	process_owner	Process Owner	✗	
8	DC	business_user	Business User	✓	

Mediante el *wizard* de creación se selecciona el módulo (*Module*) en el combo de selección, se introduce el nombre (*Name*) del rol, se marca el selector de **isCross** en caso de tratarse de un rol transversal y se introduce una descripción del rol. A continuación, se muestra cómo crear el rol de CDO, el cual es transversal.

**Roles Edit** ✕

Module:  Name:   isCross

Description:

### Visión de Desarrollador

Para definir roles hay que configurar la tabla **role** del esquema Zeus. Para configurar los roles, rellenar un sql como el siguiente:

```

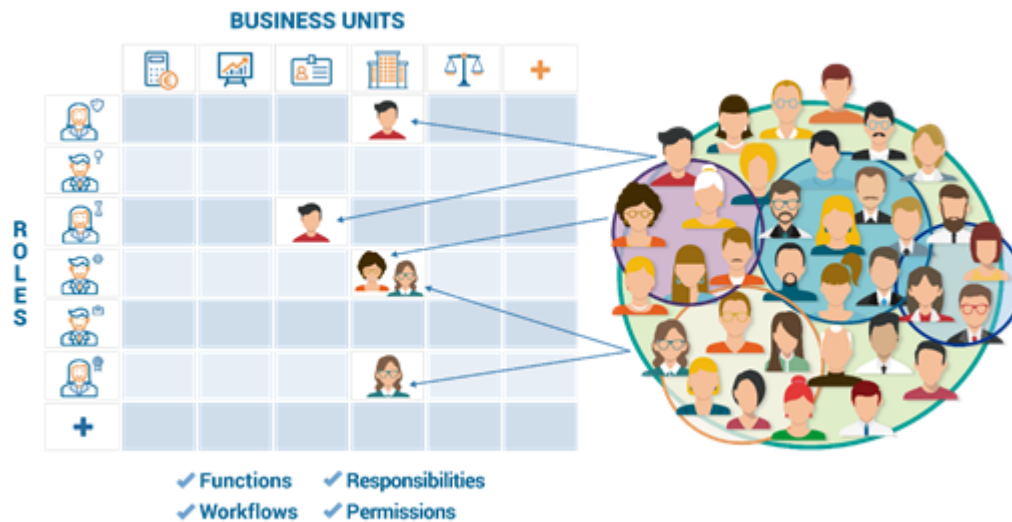
INSERT INTO zeus."role"
(id_role, is_cross, "module", role_name, role_description) VALUES
(1, false, 'ALL', 'developer', 'Developer'),
(2, true, 'ALL', 'architect', 'Architect'),
(3, true, 'ALL', 'administrator', 'Administrator'),
(4, false, 'ALL', 'data_steward', 'Data Steward'),
(5, false, 'ALL', 'data_owner', 'Data Owner'),
(6, false, 'ALL', 'compliance', 'Compliance'),
(7, false, 'ALL', 'process_owner', 'Process Owner'),
(8, false, 'ALL', 'contributor', 'Role for responsible of entity or relationship'),
(9, true, 'ALL', 'business_user', 'Business User'),
(10, true, 'ALL', 'apiAdmin', 'Admin by API'),
(11, true, 'ALL', 'default', 'default'),
(12, true, 'ALL', 'portunoAdmin', 'Portuno admin'),
(13, true, 'ALL', 'portunoCredential', 'Portuno Credential Admin');
    
```

**NOTAS:**

- Para que los roles sean traducibles a los idiomas de la aplicación, es necesario que el campo `role_name` se rellene idéntico a como se configura el rol en los proveedores de identidades. La clave de traducción en la tabla `portuno.translations` será "ROLE:" seguido del `role_name`.
- El nombre asignado a los roles no debe contener caracteres que los proveedores de identidades utilizados no permitan o que coincidan con el separador de roles y unidades organizativas o con el propio separador de unidades organizativas.

### 3. Definir los permisos asociados a cada rol

Una vez definidos los roles, se deben definir permisos asociados a cada rol conforme a las responsabilidades definidas en el gobierno del dato en la tabla **permission**.



Los permisos son los que habilitan a cada rol a realizar acciones a bajo nivel. Existen distintos permisos para el Catálogo de Datos y para el Glosario de Negocio debido a las particularidades de cada uno de los módulos y sus entidades. Además, estos permisos se configuran de forma individual para cada uno de los tipos de entidades definidos en el metamodelo.

Los distintos tipos de permisos a configurar por tipo de entidad y relación son los siguientes:

- **Automatic\_Metadata:** que habilita la creación de entidades o relaciones en Anjana Data de forma asistida mediante el descubrimiento y la importación automática de metadatos de los sistemas origen.
- **Change\_status:** que habilita a activar y desactivar entidades o relaciones no nativas.
- **Change\_organizational\_unit:** que habilita a modificar la unidad organizativa de una entidad.
- **Creation\_modif:** que habilita a crear, modificar y enviar a validar entidades o relaciones creados o modificados.
- **Delete\_all:** que habilita al usuario a eliminar entidades o relaciones sea quien sea el creador y el estado que tengan.
- **Delete\_my\_obj:** que habilita al usuario a eliminar entidades o relaciones creados por él mismo sea cual sea su estado.
- **Deprecation:** que habilita a deprecar manualmente entidades nativas.
- **Organizational\_unit\_owner:** que habilita como propietario de aquellas entidades que se encuentren dentro de la unidad organizativa.

Además, algunos permisos afectan a toda la aplicación:

- **Access:** que habilita el acceso a Anjana Data, la visualización de los objetos y el carrito para la solicitud de adherencias.

- **Admin:** que habilita el acceso al Portal de administración de Anjana.
- **Api\_admin:** que habilita el uso de la API administrativa. Usuarios con este permiso, además, pueden editar masivamente objetos por el Portal de Anjana. Además permite acciones especiales de procesamiento en Portuno.
- **Credential\_admin:** que habilita el acceso a las tablas de credenciales que desde el Portal de administración de Anjana se gestionan: appConfiguration, users y userOuRole.
- **Lineage\_access:** que habilita la visualización del linaje.
- **Wizard:** que habilita el acceso al *wizard* de creación de objetos de Anjana Data. Este permiso tiene sentido cuando el rol tiene algún permiso de Creation\_modif o Automatic\_metadata de alguna entidad o relación.
- **Workflow\_access:** que habilita la visualización (y, por tanto, también aprobación o rechazo) de los flujos (*workflows*) de validación.

### Estructura de la tabla

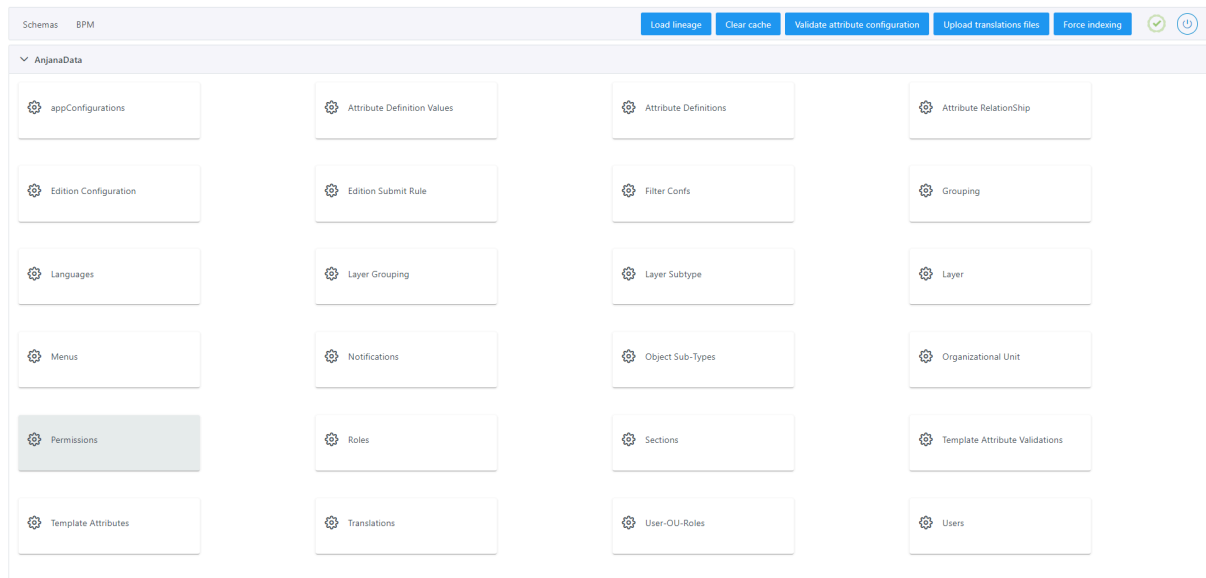
Cada permiso se caracteriza por los siguientes elementos:

- **id\_permission:** identificador único del permiso.
- **permission\_action:** indica la acción para la cual se está dando permiso de entre las de la siguiente lista:
  - ACCESS
  - ADMIN
  - API\_ADMIN
  - AUTOMATIC\_METADATA
  - CHANGE\_OU
  - CHANGE\_STATUS
  - CREATION\_MODIF
  - CREDENTIAL\_ADMIN
  - DELETE\_ALL
  - DELETE\_MY\_OBJ
  - DEPRECATION
  - LINEAGE\_ACCESS
  - ORGANIZATIONAL\_UNIT\_OWNER
  - WIZARD
  - WORKFLOW\_ACCESS
- **sub\_type:** indica el objeto sobre el que aplica el permiso
  - será el nombre de un subtipo de entidad o relación (name de la tabla object\_subtype del esquema de Anjana) para los *permission\_action* AUTOMATIC\_METADATA, CREATION\_MODIF, DELETE\_ALL, DELETE\_MY\_OBJ y ORGANIZATIONAL\_UNIT\_OWNER.
  - será el nombre de un subtipo de entidad nativa o no nativa menos INSTANCE (name de la tabla object\_subtype del esquema de Anjana), que no tiene OU propia para el *permission\_action* CHANGE\_OU.
  - será el nombre de un subtipo de entidad nativa (name de la tabla object\_subtype del esquema de Anjana) para el *permission\_action* DEPRECATION.

- será el nombre de un subtipo de entidad no nativa (name de la tabla object\_subtype del esquema de Anjana) para el *permission\_action* CHANGE\_STATUS.
  - será “ALL” para los *permission\_action* API\_ADMIN, LINEAGE\_ACCESS, WIZARD y WORKFLOW\_ACCESS.
  - será “ANJANA” para los *permission\_action* ADMIN y CREDENTIAL\_ADMIN.
  - El permiso ACCESS permite el acceso al Portal y a los objetos si el *subtype* es “ALL” y acceso al carrito para la solicitud de adherencias cuando *subtype* = “ADHERENCE”.
- **id\_role:** indica el rol al que se le están otorgando permisos.

## Visión de Administrador

El alta de permisos mediante el Panel de Administrador de Anjana Data se realiza en **Permissions**:



Al acceder se muestra una tabla que contiene los permisos existentes en la configuración actual.

La creación de un permiso nuevo se realiza mediante el botón **New**:



+ New

Permissions				
id	action	subType	role	
-	-	-	Igual	
Criterio	-	-	-	
<span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	<span style="background-color: #0070C0; color: white; padding: 2px 5px;">Filtrar</span>	
1	ACCESS	ALL	1-developer-Developer	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
2	ACCESS	ALL	2-architect-Architect	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
3	ACCESS	ALL	3-administrator-Administrator	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
4	ACCESS	ALL	4-data_steward-Data Steward	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
5	ACCESS	ALL	5-data_owner-Data Owner	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
6	ACCESS	ALL	6-compliance-Compliance	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
7	ACCESS	ALL	7-process_owner-Process Owner	<span style="color: green;">✎</span> <span style="color: red;">✖</span>
8	ACCESS	ALL	8-business_user-Business User	<span style="color: green;">✎</span> <span style="color: red;">✖</span>

Mediante el *wizard* de creación se selecciona la acción, el subtipo de objeto al que aplica y el rol que tendrá el permiso a crear.

### Permissions Edit ✕

action

subType

role

✕ Cancel
✓ Save

### Visión de Desarrollador

Para definir permisos hay que configurar la tabla **permission** del esquema Zeus. Para configurar los permisos, hay que rellenar un sql como el siguiente:

```
INSERT INTO zeus."permission" (id_permission, permission_action, sub_type, id_role) VALUES
(104, 'CREATION_MODIF', 'DATASET', 2),
(8, 'LINEAGE_ACCESS', 'ALL', 2),
(9, 'WORKFLOW_ACCESS', 'ALL', 2),
(10, 'WIZARD', 'ALL', 2),
(7, 'ACCESS', 'ALL', 2),
(10051, 'CHANGE_STATUS', 'DATA_BASE', 2),
(10031, 'AUTOMATIC_METADATA', 'DATASET', 2);
```

#### NOTAS:

- Para evitar inconsistencias y debido a que los permisos de los usuarios se cachean en la aplicación, es necesario limpiar las cachés (Clear Cache) desde Portuno tras hacer cambios

en la tabla de zeus.permission. Además, es conveniente que los usuarios cierren sesión para que empiecen a aplicar los cambios

- No se deben definir permisos para DATASET\_FIELD ya que, para él, aplican los permisos de DATASET.

#### 4. Idiomas de la aplicación

La tabla **languages** contiene los lenguajes disponibles del sistema.

Los textos de Anjana deben traducirse en cada uno de estos idiomas en la tabla de portuno translations para que los usuarios puedan utilizar la aplicación en todos ellos.

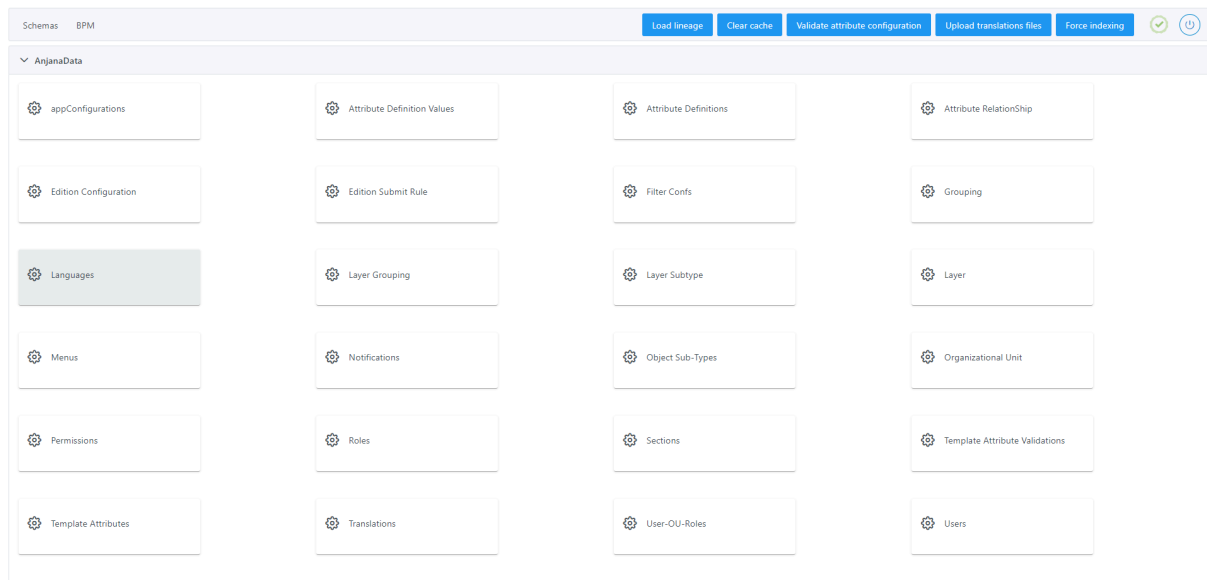
##### Estructura de la tabla

Cada idioma se caracteriza por los siguientes elementos:

- **id:** identificador único de la tabla.
- **i18nCode:** código i18n para el idioma.
- **description:** nombre del idioma.





##### Visión de Administrador

El alta de un nuevo idioma en el panel de administración de Anjana Data se realiza en **Languages**:



Al acceder se muestra una tabla que contiene todos los idiomas existentes en la configuración actual.

La creación de un nuevo idioma se realiza mediante el botón **New**:

Languages			
id	i18nCode	Description	
-	-	-	
Criterio	Criterio	Criterio	
Filtrar	Filtrar	Filtrar	
1	en-US	Inglés	 
2	es-ES	Español	 

Mediante el *wizard* de creación se identifica el **código** i18n del idioma y la **descripción**.

**Languages Edit** ✕

i18nCode

Description

✕ Cancel
✓ Save

Después de añadir el nuevo idioma hay que incluir las traducciones en la tabla translations del schema Portuno (en el apartado [Traducciones](#) se explica cómo se añade cada una de ellas).

La manera más sencilla de añadir todas las traducciones necesarias para el nuevo idioma es duplicar las que ya existen en uno de los idiomas que se encuentran precargados en Anjana como son el inglés o el español y después cambiar el valor de las columnas 'language' (por el código i18nCode creado) y la columna 'value' por la traducción en el idioma que hemos creado.

En Anjana hay traducciones que deben existir en esta tabla. Algunas de ellas se encuentran listadas en el apartado [Traducciones internas de Anjana](#)

A continuación se incluye un listado con los campos de base de datos que contienen claves de traducciones:

- anjana
  - attribute\_definition
    - label\_translation\_key
    - place\_holder\_translation\_key
    - short\_description\_translation\_key
  - entity
    - state

- imported\_metadata
  - imported\_status
- node
  - state
- relationship
  - state
- request\_import
  - state
- menu
  - name
- sections
  - name
- template\_attribute\_validation
  - validator\_key
- workflow\_info
  - workflow\_action
- hermes
  - notification
    - subject
    - translation\_key
  - sent\_notification
    - message\_key
  - task\_step
    - state
  - workflow\_execution
    - state
    - workflow\_type
  - workflow\_task
    - state
- minerva
  - audit\_log
    - action
    - action\_description
  - filter\_confs
    - label
- portuno
  - translations
    - config\_key

Una vez se han añadido todas las traducciones en la tabla translations de portuno es necesario actualizar los ficheros de traducción que se encuentran en Minio seleccionando la opción 'Upload Translations Files' en el portal de administrador para que estén disponibles en el portal de Anjana.



## Visión de Desarrollador

Para definir los idiomas hay que configurar la tabla **languages** del esquema Zeus. Para ello, hay que rellenar un sql como el siguiente:

```
INSERT INTO zeus.languages
(id,i18n_code,description) VALUES
(1,'en-US','Inglés'),
(2,'es-ES','Español');
```

NOTA:

- Es posible añadir un texto traducido para identificar cada idioma en los selectores de la aplicación. La clave de traducción en la tabla portuno.translations será el código i18n del idioma en la tabla languages.

## 5. Autenticación y autorización por base de datos

En Anjana la autenticación y autorización de los usuarios en base de datos es una opción obligatoria y no configurable por necesidades técnicas, pero el uso es elección del administrador. De esta manera se dispone, en el esquema Zeus de base de datos, toda la información de los usuarios, sus contraseñas y los roles que tienen cada uno en las unidades organizativas de Anjana en estas dos tablas:

- users → Donde se incluye la información de los usuarios (email, teléfono, nombre, apellido y contraseña).
- user\_ou\_role → Tabla donde se asigna a los usuarios los roles en ciertas unidades organizativas.

IMPORTANTE: Para proteger la privacidad de la contraseña se usa la función de cifrado lento Bcrypt, en el siguiente enlace se dispone de más información de este algoritmo de encriptación :

<https://www.devglan.com/online-tools/bcrypt-hash-generator>

De esta forma no se incluye en claro la contraseña en la tabla 'users' sino que es necesario calcular un hash e introducirlo en la columna 'password\_hash'.

Para configurar el algoritmo de encriptación es necesario incluir la configuración en el yml de zeus, toda la información está detallada en la sección correspondiente [Configuración en ficheros yml](#).

### a. Usuarios

La tabla **users** contiene la información de los usuarios de Anjana cuando se autentican o autorizan por medio de base de datos.

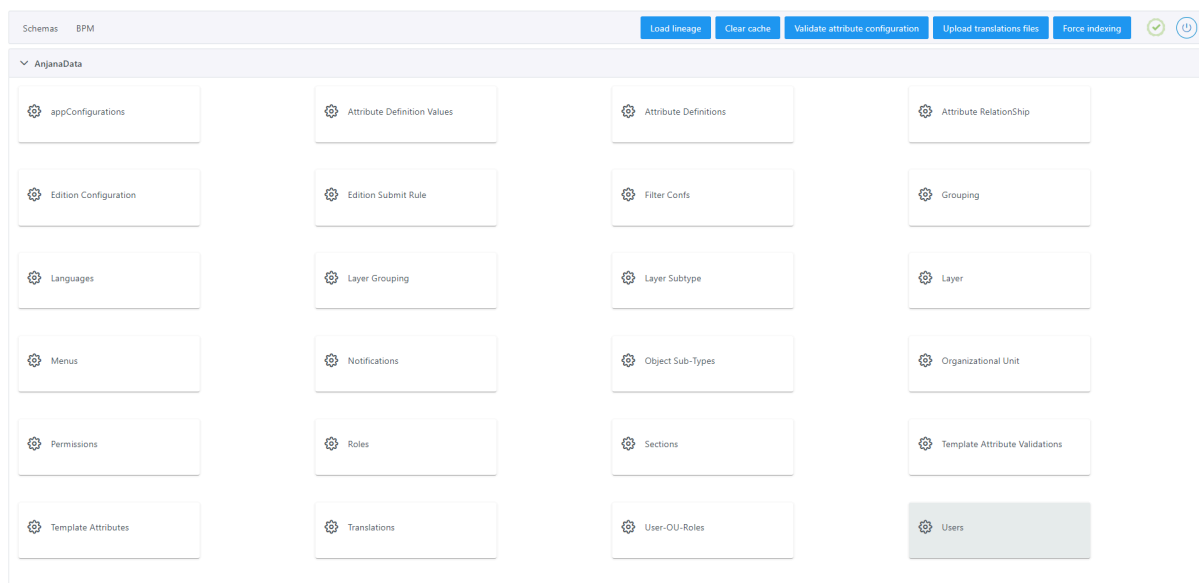
### Estructura de la tabla

Cada usuario se caracteriza por los siguientes elementos:

- **id\_user:** identificador único, sirve para gestión interna de Anjana.
- **user\_name:** Identificador único del usuario, es el valor que hay que introducir en el apartado 'Username' cuando hacemos login en Anjana.
- **first\_name:** Nombre del usuario.
- **last\_name:** Apellidos del usuario.
- **email:** Dirección de correo del usuario.
- **is\_service\_user:** Boolean para indicar si se trata de un usuario de servicio utilizado para la securización de los microservicios. Este campo es informativo.
- **phone:** Número de teléfono del usuario.
- **title:** Título del usuario, no es necesario rellenarlo
- **password\_hash:** Hash de la contraseña del usuario. Para obtenerlo se puede usar cualquier calculadora de Bcrypt de internet, el hash obtenido es único.





## Visión de Administrador

El alta de permisos mediante el Panel de Administrador de Anjana Data se realiza en **Users**:



Al acceder a la tabla de Users se muestra una tabla que contiene todos los usuarios existentes en la configuración actual.

La creación de un nuevo usuario se realiza mediante el botón **New**:

								+ New
Users								
id ↑	userName ↑	firstName ↑	lastName ↑	email ↑	phone ↑	title ↑	Password ↑	
-	-	-	-	-	-	-	-	
Criterio	Criterio	Criterio	Criterio	Criterio	Criterio	Criterio	Criterio	
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	
1	paco.perez	paco	perez	paco.perez@email.es			\$2a\$04\$KEnH62Lg0F5jAIBc	 
2	maria.abad	maria	abad	maria@mail.com	1234		\$2a\$12\$L66ZukacO/M6yRu	 

Mediante el *wizard* de creación se identifica el **username** identificativo del usuario, su **nombre** y **apellido**, **correo** electrónico, **teléfono**, **rol** en la compañía y la **contraseña** cifrada.

### Users Edit ✕

<b>userName</b> <input type="text" value="ana.martin"/>	<b>firstName</b> <input type="text" value="Ana"/>	<b>lastName</b> <input type="text" value="Martin"/>
<b>email</b> <input type="text" value="ana.martin@corporate.com"/>	<b>phone</b> <input type="text" value="12345678"/>	<b>title</b> <input type="text"/>
<b>Password</b> <input type="text" value="\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp"/>		
<input type="button" value="✕ Cancel"/>		<input type="button" value="✓ Save"/>

### Visión de Desarrollador

Se pueden incluir usuarios relleno un sql similar al siguiente:

```

INSERT INTO zeus.users
(id_user, email, first_name, last_name, password_hash, phone, title, user_name, is_service_user)
VALUES
(571, 'ana.martin@corporate.com', 'Ana', 'Martin',
'$2a$10$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/I3/zjIKj.', '12345678', NULL,
'ana.martin', false),
(3, 'pepe.lopez@corporate.com', 'Pepe', 'Lopez',
'$2a$10$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/I3/zjIKj.', '(+34)999 999 999',
'Empleado', 'pepe.lopez', false),
(2, 'maria.gonzalez@corporate.com', 'Maria', 'Gonzalez',
'$2a$10$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/I3/zjIKj.', '(+34)999 999 999',
'Empleado', 'maria.gonzalez', false);

```

#### NOTAS:

Para el valor del campo `user_name` se aplican las siguientes restricciones:

- No debe contener ':', '#', '(' o ')' porque interfiere con las ARI
- No puede estar vacío o ser nulo.
- Los caracteres '.' (punto), '\_' (guión bajo), '-' (guión) no pueden estar al inicio, al final, ni aparecer consecutivamente.
- Longitud mínima 1 y máxima 50 caracteres.

### b. Rol y OU asignados a los usuarios

La asignación de usuarios a roles en unidades organizativas se configura en la tabla **users\_ou\_role**.

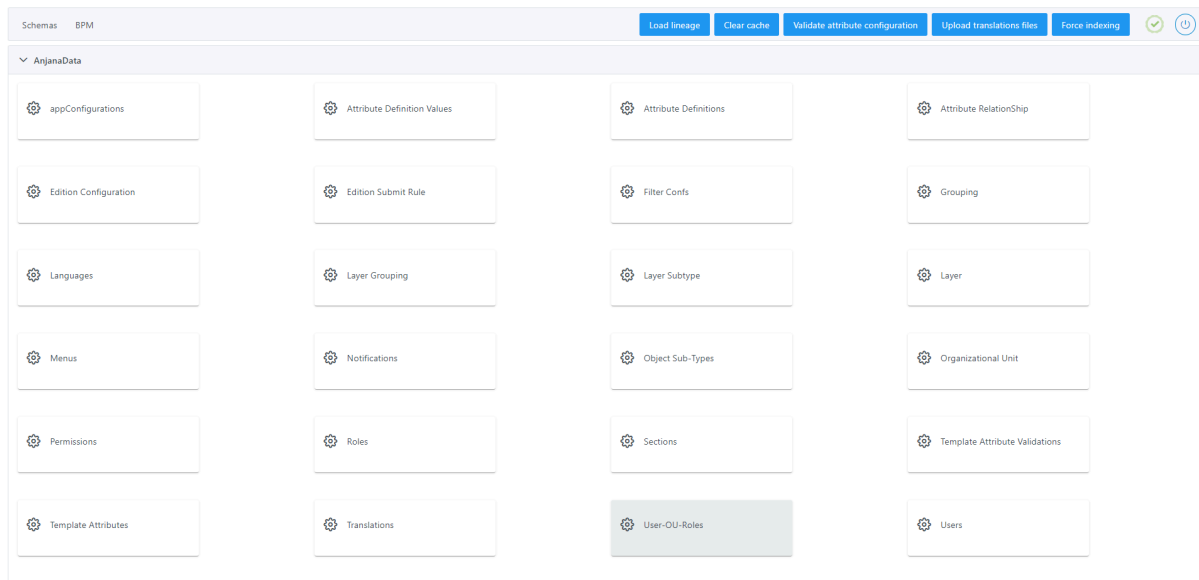
#### Estructura de la tabla

Cada asignación de usuario a rol se caracteriza por los siguientes elementos:

- **user\_id**: identificador del usuario, es el campo id\_user de la tabla de users.
- **ou\_id**: identificador de la unidad organizativa, es el campo id\_organizational\_unit de la tabla organizational\_unit.
- **role\_id**: identificador del rol, es el campo id\_role de la tabla role.

#### Visión de Administrador








El alta de permisos mediante el Panel de Administrador de Anjana Data se realiza en **User-OU-Roles**:



Al acceder a la tabla de User\_ou\_roles se muestra una tabla que contiene los usuarios de la aplicación con los roles y OUs que tienen asignados.

La asignación de un nuevo rol a un usuario se realiza mediante el botón **New**:



User-OU-Roles			+ New
user	ou	role	
Igual	Igual	Igual	
-	-	-	
Filtrar	Filtrar	Filtrar	
2-architect	2-Europe/SPA/Finance	2-architect-Architect	
2-architect	3-Europe/SPA/Marketing	2-architect-Architect	
2-architect	4-Europe/SPA/Commercial	2-architect-Architect	
2-architect	5-Europe/SPA/Risk	2-architect-Architect	
2-architect	6-Europe/SPA/Market	2-architect-Architect	
2-architect	7-Europe/SPA/Accounting	2-architect-Architect	
2-architect	9-General/SCIB_HQ/Finance	2-architect-Architect	

Mediante el *wizard* de creación se identifica el username identificativo del usuario, la OU y el rol que se le asigna y que le otorga permisos.

### User-OU-Roles Edit

user

ou

role

✕ Cancel
✓ Save

### Visión de Desarrollador

Para definir los roles de cada usuario en Anjana hay que configurar la tabla **users\_ou\_roles** del esquema Zeus. Para configurarlo, hay que rellenar un sql como el siguiente:

```
INSERT INTO zeus.user_ou_role (user_id,ou_id,rol_id) VALUES
(1,1,1),
(1,39,4),
(2,1,15),
(2,1,16),
(2,3,2);
```

#### NOTAS:

- Al ser todos los campos de esta tabla los que hacen único al registro, cuando se desee cambiar algún rol de algún usuario será necesario eliminar el registro original y crear uno nuevo.

## Configuración de propiedades

Esta configuración se explica como configurar en formato ficheros YAML, pero puede ser configurada en estos o en la tabla **app\_configuration**, respetando la nomenclatura definida tras los ejemplos.

### Todos los módulos core

La configuración de propiedades de los módulos core se ha desplazado a base de datos a la tabla **app\_configuration** como se menciona en el apartado "[Esquema Portuno de BD/Variables del Sistema](#)" y allí se explican sus características.

### Tot Plugin

Cada plugin tiene una configuración especial, revisar los documentos para cada uno de ellos.

- Tot plugin AWS IAM
- Tot plugin AWS Glue
- Tot plugin AWS S3
- Tot plugin Azure AD
- Tot plugin Azure Storage
- Tot plugin GCP IAM
- Tot plugin GCP BigQuery
- Tot plugin GCP Storage
- Tot plugin JDBC
- Tot plugin JDBC Denodo
- Tot plugin LDAP
- Tot plugin PowerBi
- Tot plugin Sql Server

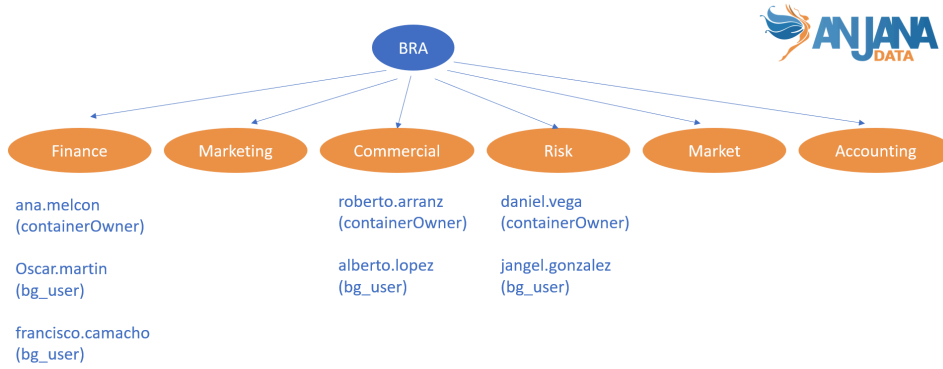
## Ejemplos de configuración de proveedores de identidades

### Configuración de base de datos

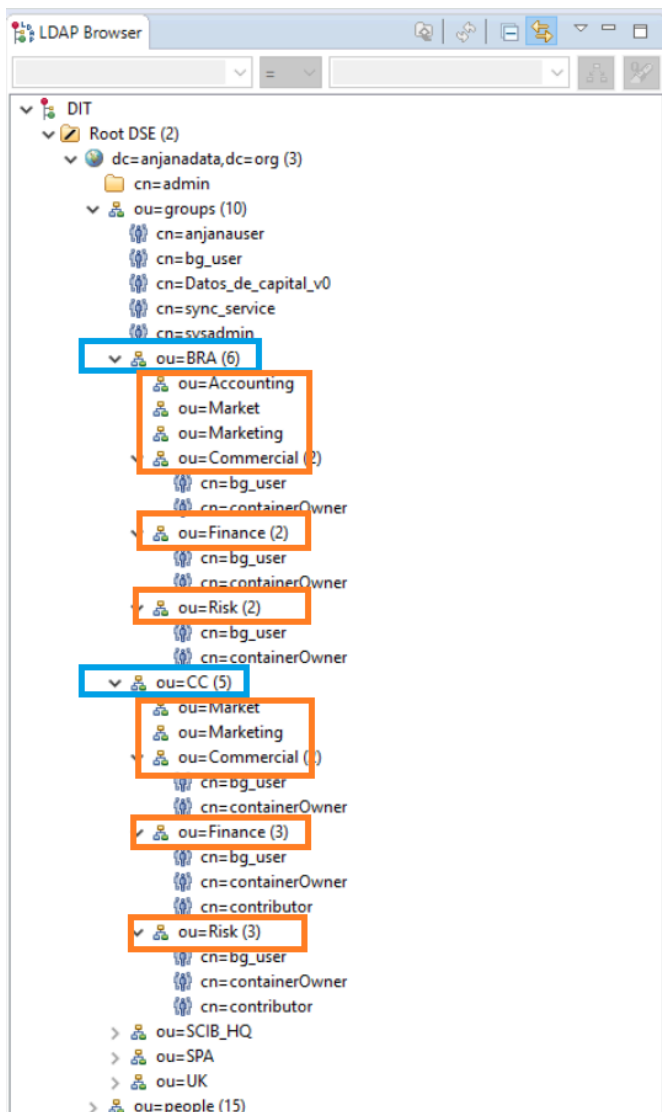
En la sección [Autenticación y autorización por base de datos](#) se encuentra la información acerca de las tablas implicadas del esquema de Zeus y cómo deben completarse.

### Configuración de LDAP

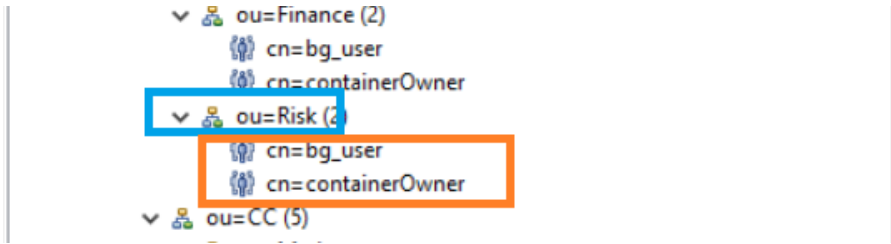
Dentro de **groups** se deben dar de alta las unidades organizativas con su correspondiente jerarquía de niveles padre-hijo.



Por ejemplo, para la Business Unit de BRA se deben crear las unidades de Finance, Marketing, Risk, Market y Accounting:



A su vez, dentro del nivel 2 de cada Business Unit, se deben incluir los **roles**. Por ejemplo, para la Business Unit de BRA>Risk se debe incluir el rol de **containerOwner** y **bg\_user**:



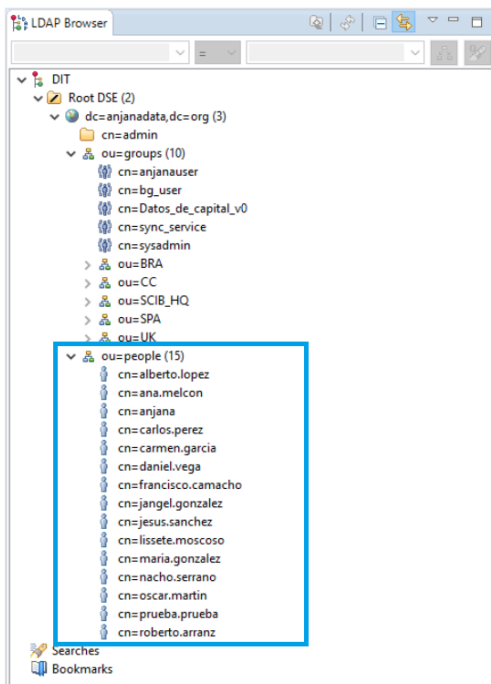
También se debe indicar el rol que tiene un determinado usuario dentro de la Unidad Organizativa:

```

#!RESULT OK
#!CONNECTION ldap://lab0.int.anjanadata.org:389
#!DATE 2020-09-11T15:39:37.620
dn: cn=containerOwner,ou=Risk,ou=BRA,ou=groups,dc=anjanadata,dc=org
changetype: add
objectClass: groupOfNames
objectClass: top
description: admin
cn: containerOwner
member: cn=daniel.vega,ou=people,dc=anjanadata,dc=org

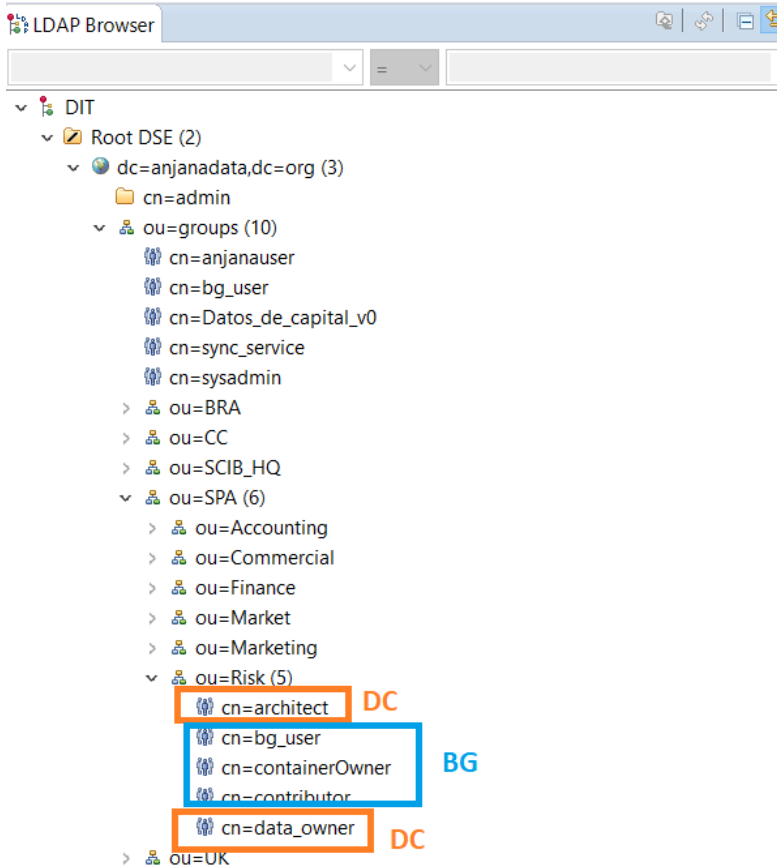
```

Al mismo nivel de group, en **people** se deben dar de alta los distintos usuarios:

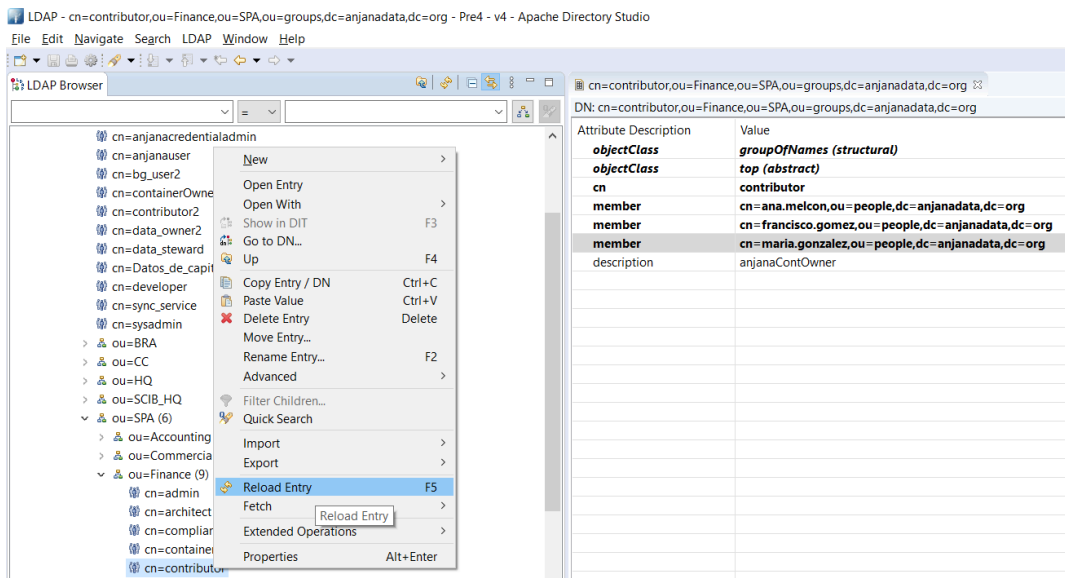



**NOTAS:**

- Si como proveedor de identidades sólo se utiliza LDAP, todos los roles de la aplicación deben estar configurados en él:









- Si en algún momento no coincide lo que se visualiza en LDAP y en el perfil de un usuario en Anjana, pulsar en Reload Entry, ya que a veces se quedan roles cacheados





User
Search in: Search...


---

- 
- 
- 
- 
- 
- 







### User information




maria.gonzalez

 License valid

### Role & Organizational Unit associated

 CC/Accounting Data Owner	 Bra/Market Data Owner
 SPA/Market containerOwner	 HQ/People Process Owner, Contributor, Data Owner
 UK/Market Data Owner	
 Cross Architect, Admin	

 Modification Logs
 Search Logs
 Error Log

```

#!RESULT OK
#!CONNECTION ldap://lab0.int.anjanadata.org:389
#!DATE 2020-09-11T15:39:37.620
dn: cn=containerOwner,ou=Risk,ou=BRA,ou=groups,dc=anjanadata,dc=org
changetype: add
objectClass: groupOfNames
objectClass: top
description: admin
cn: containerOwner
member: cn=daniel.vega,ou=people,dc=anjanadata,dc=org

```

- Los nombres de usuario no deben contener ':', '#', '(' o ') porque interfiere con las ARI

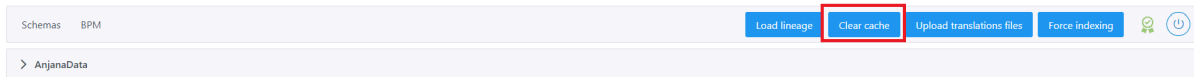
## Proveedores externos

Existen documentos específicos para el resto de proveedores externos soportados por Anjana Data

- Anjana Data 4.4 - DS - Integracion Genérico
- Anjana Data 4.4 - DS - Integracion Azure AD
- Anjana Data 4.4 - DS - Integracion GCP
- Anjana Data 4.4 - DS - Integracion AWS
- Anjana Data 4.4 - DS - Integracion OKTA

## NOTAS:

- Después de cualquier actualización de los proveedores de identidades de Unidades Organizativas, roles o usuarios de la aplicación, es recomendable que se limpie la caché de los microservicios desde Portuno:



## Configuración de Batches

Para todos los batches (procesos de Anjana desatendidos que se ejecutan automáticamente con cierta periodicidad) se puede configurar su programación en la tabla `app_configuration` de portuno, empleando expresiones cron de Spring.

Las expresiones cron de Spring se componen de 6 elementos:

- El primer elemento hace referencia a los segundos (De 0 a 59).
- El segundo elemento a los minutos (De 0 a 59).
- El tercer elemento a las horas (De 0 a 23).
- El cuarto elemento al día del mes (De 1 a 31).
- El quinto elemento al mes (De 1 a 12).
- El sexto elemento al día de la semana (De 0 a 7).

De este modo, si se configura `0 0 8 * * *` el batch se ejecutará todos los días a las 8 de la mañana. Otras convenciones que se siguen:

- Los campos que se rellenen con `*` representan del primero al último, es decir se ejecuta para todos los posibles valores. Para los campos de día del mes o día de la semana se usa el símbolo `?`.
- Las comas se pueden usar para separar los elementos de una lista (por ejemplo: `1,15`, el 1 y el 15).
- Se puede usar el `-` para expresar un rango de elementos (por ejemplo: `de 8-10`, de 8 a 10, ambos incluidos). El rango siempre incluye los extremos.

Para más ejemplos ver la documentación oficial: [Spring Cron Expressions](#).

### 1. Comprobación de roles transversales (cross)

El batch de comprobación de roles transversales o cross se configura en la tabla de `app_configuration` mediante la propiedad:

- `anjana.scheduling.notification` se usa para configurar la tarea programada para avisar de roles Cross mal configurados (por defecto: `0 0 0 ? * *`).

El módulo al que aplica y que debe rellenarse en el campo `filter` de `app_configuration` debe ser `'zeus'`.

## 2. Licencia a punto de expirar

También se puede configurar el aviso de licencia a punto de expirar en la tabla de `app_configuration` mediante la propiedad:

- `anjana.scheduling.licenseNotification` se usa para configurar la tarea programada para enviar notificaciones por licencia a punto de expirar o expirada (por defecto = `0 0 0 ? * *`).

El módulo al que aplica y que debe rellenarse en el campo `filter` de `app_configuration` debe ser `'zeus'`.

## 3. Expiración

El batch de expiración se configura en la tabla de `app_configuration` mediante la propiedad :

- `anjana.scheduling.expiration` se usa para configurar la expiración de objetos (por defecto: `0 45 7 ? * *`).

El módulo al que aplica y que debe rellenarse en el campo `filter` de `app_configuration` debe ser `'kerno'`.

## 4. Preaviso por expiración

El batch de preaviso por expiración se configura en la tabla de `app_configurations` mediante las propiedades:

- `anjana.scheduling.notification` se usa para configurar la notificación de objetos próximos a expirar (por defecto: `0 45 7 ? * *`).
- `anjana.notification.expiration` se usa para configurar, en número de días, los objetos que van a expirar dentro de ese rango (por defecto: 4).

El módulo al que aplican y que debe rellenarse en el campo `filter` de `app_configuration` debe ser `'kerno'`.

## 5. Indexación

El batch de indexación de todos los objetos (entidades y relaciones) se puede configurar en la tabla de `app_configuration` mediante la siguiente propiedad:


- `anjana.scheduling.indexAll` se usa si se quiere configurar la indexación periódica de todos los objetos de la aplicación (con las excepciones mencionadas en la Guía de Usuario). Esta configuración está, por defecto, desactivada.

El módulo al que aplican y que debe rellenarse en el campo `filter` de `app_configuration` debe ser `'minerva'`.



# Preguntas frecuentes

## 1. Permisos

Un rol que no puede crear objetos tiene disponible el icono  para abrir el wizard y no tiene subtipos de entidad o de relación disponibles, ¿a qué puede deberse?

Es necesario comprobar los permisos del rol. Es probable que en la tabla permission no tenga ningún permiso de CREATION\_MODIF o AUTOMATIC\_METADATA pero, sin embargo, sí tenga el de WIZARD. En este caso, eliminar el permiso de WIZARD para que desaparezca la opción en el front de Anjana de creación.

## 2. Atributos en el Portal de Anjana

He editado un atributo de un objeto y, sin embargo, en el Portal no me aparece el atributo relleno.

En este caso es necesario comprobar que el atributo tiene el mismo nombre de attribute\_definition que el indicado en Atributos obligatorios.

## 3. Configuración de workflows

Tengo configuradas las plantillas y los permisos por rol y he desplegado los workflows pero, cuando selecciono Submit en un formulario, no parece que haga nada.

En este caso hay que revisar si la tabla object\_subtypes tiene un registro para la plantilla con el nombre de los workflows que se han configurado.

## 4. Despliegue de workflows

He arrancado correctamente Hermes, habiendo subido algunos workflows, y no tengo certeza de que se hayan desplegado.

La tabla act\_re\_procdef del esquema **hermes** es propia de Activiti y en ella se registran los workflows desplegados.

Si la subida de los workflows se ha hecho por medio de subida de xml y no están desplegados los workflows, chequear que la extensión de los ficheros es bpmn20.xml

## 5. Asignación de roles en los pasos de los workflows

He lanzado un workflow de validación y todos los pasos del workflow se han quedado con estado *Waiting (Esperando)* y sin asignación a rol.

Spanish Banks						
Tipo de workflow		Modificación	Solicitante	maria.diez	Rol	Propietario del Dato
					Fecha de petición	10 de oct. de 2023 20:01
<b>Data Owner Validation</b>						
Usuario	Rol	Unidad organizativa	Respuesta	Fecha de recepción	Fecha de respuesta	Comentarios
-	-	Workshops/Certificación	⏸	No hay fechas	No hay fechas	-
<b>Admin Validation</b>						
Usuario	Rol	Unidad organizativa	Respuesta	Fecha de recepción	Fecha de respuesta	Comentarios
-	-	Workshops/Certificación	⏸	No hay fechas	No hay fechas	-
<b>Architect</b>						
Usuario	Rol	Unidad organizativa	Respuesta	Fecha de recepción	Fecha de respuesta	Comentarios
-	-	Workshops/Certificación	⏸	No hay fechas	No hay fechas	-
<b>Steward</b>						
Usuario	Rol	Unidad organizativa	Respuesta	Fecha de recepción	Fecha de respuesta	Comentarios
-	-	Workshops/Certificación	⏸	No hay fechas	No hay fechas	-

Esto se debe a un error en la configuración del workflow. Para resolverlo es necesario acceder a Portuno y, en el apartado de BPM, abrir el workflow correspondiente a la acción lanzada (revisar la configuración del subtipo del objeto en la tabla `object_subtypes`) y revisar los estados y las condiciones entre ellos. Para más detalle acceder a la Guía de configuración de workflows.

El error puede deberse a un error en el nombre del atributo utilizado, a la ausencia de unas comillas, una condición de salto mal definida...

En cualquier caso, el workflow lanzado que se haya quedado con este estado no es recuperable. La opción para poder aprobar o rechazar el objeto asociado es clonarlo y, una vez arreglado el workflow, guardado, validado y desplegado desde Portuno, lanzarlo para este nuevo objeto.

## 6. Error en la solicitud de adherencia

En caso de recibir un error al enviar la solicitud de una adherencia, es probable que el problema sea que faltan los atributos necesarios para esta relación.

La relación de ADHERENCE es obligatorio definirla en la tabla `object_subtype` y deben añadirse en `template_attribute` los registros correspondientes a los campos `expirationDate`, `requestReason` y `pae` para ella.

## 7. En el formulario de mi entidad o relación aparecen atributos que no deben estar

En caso de haber configurado las plantillas y ver en el formulario atributos equivocados puede deberse a un mapeo incorrecto de los atributos a la sección del objeto visualizado.

Es decir, conviene revisar la tabla `anjana.template_attribute` para chequear si los atributos están en la sección correcta.

## 8. Necesito eliminar un atributo de una plantilla para la que ya tengo objetos creados

En este caso, para evitar dejar objetos corruptos con valor para un atributo eliminado, en vez de eliminarlo, Anjana recomienda desactivar el atributo en la plantilla.

Para ello, en la tabla `template_attribute`, se cambiará en el registro correspondiente a ese atributo en esa plantilla el flag `active` a `false`.

Con esto, los objetos siguen manteniendo el valor que tenían para el atributo aunque no es visible en el formulario y, por tanto, no es editable ni consultable. Así, en caso de querer activarlo posteriormente, simplemente es necesario cambiar el flag `active` a `true` y, de nuevo, se podrá consultar su valor para los objetos que tuvieran valor.

## 9. ¿Cómo configurar una nueva plantilla en Anjana?

1. Definir el template a configurar (tabla `anjana.object_subtype`)
2. Definir los menús que van a completar su formulario dinámico y las secciones de estos (tablas `anjana.menu` y `anjana.sections`)
3. Definir o reutilizar los atributos (y sus valores, en caso de que sean valores de referencia) que compondrán la plantilla (tabla `anjana.attribute_definition` y `anjana.attribute_definition_value`)
4. Organizar los atributos en las secciones de la plantilla (tabla `anjana.template_attribute`)
5. Definir las validaciones de atributos para la plantilla (si el atributo es requerido, si depende de otro, si es editable...) (tabla `template_attribute_validation`)
6. Definir las reglas de versionado que identifican los atributos de la plantilla que generan versión (caso de las entidades de DC propias de Anjana: `dataset`, `dataset_field`, `process`, `instance` o `solution`)
7. Definir las reglas de lanzamiento de workflow en caso de que algún rol no tenga que enviar workflow cuando cambia algún atributo de la plantilla de los objetos

8. Completar `anjana.object_subtype` con la información de los workflows que serán lanzados con cada acción

## 10. Borrado de colección de Solr

Si al borrar una colección de Solr y arrancar Minerva 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
```

## 11. Necesito cambiar el tipo de un atributo de plantilla

En caso de ser necesario cambiar el tipo de un atributo de plantilla y tener ya creados objetos de ese subtipo previamente es conveniente:

1. [Eliminar](#) la colección de Kern de Solr
2. Reiniciar Minerva
3. Indexar todos los registros de nuevo desde Portuno

## 12. Error subiendo un fichero a una plantilla

Si al subir un fichero a un atributo de plantilla aparece la pantalla de “Error embarazoso” es posible que se deba al límite de tamaño de los ficheros por defecto configurado.

Para cambiar esta configuración y ampliar el límite es necesario editar el yml de Kern añadiendo lo siguiente:

```
spring.servlet.multipart.max-file-size: 10MB
```

Tras el cambio, es necesario reiniciar Kern para que la configuración se aplique.

## 13. ¿Cómo configurar una capa para el linaje?

Para configurar una capa del linaje es necesario completar las siguientes tablas en el siguiente orden:

1. Layer: se debe incluir el nombre de la capa que va a ser configurada (y que será visible por el usuario) y definir si será la capa visualizada por defecto en el linaje o no. Solamente puede haber configurada una capa por defecto.
2. Layer Subtype: se deben incluir todos los subtipos de entidades y relaciones (nativas y no nativas) que aparecerán en la capa del linaje. Los elementos configurados en esta capa

podrán ser ocultados por el usuario utilizando los filtros presentes en las pantallas de linaje del Portal de Datos.

3. Grouping: se deben configurar las relaciones entre entidades que pueden ver visualizadas en el linaje como agregación en cualquiera de las capas. Por medio de la relación identificada, la entidad origen puede contener a la entidad destino en la capa si así se define en Layer Grouping. Para que una relación del metamodelo (no nativa) pueda realmente funcionar como agregación, en Object Sub Type la relación tiene que tener activado el flag Parental.
4. Layer Grouping: se define qué relaciones de agrupación de las configuradas en Grouping actúan como relaciones de agregación en la capa que se está configurando. Las agregaciones (Grouping) no incluidas en la capa se visualizarán como una línea entre ambas entidades en el linaje.