



Guía de Configuración

Control de versiones	5
Introducción	5
Conceptos básicos de Anjana Data	6
Visiones para la configuración	6
Configuración en BD	11
Esquema Zeus de BD	12
1. Definir la estructura organizativa y las unidades organizacionales	12
Visión de Administrador	13
Visión de Desarrollador	16
2. Definir los roles del modelo de gobierno	17
Visión de Administrador	18
Visión de Desarrollador	20
3. Definir los permisos asociados a cada rol	20
Visión de Administrador	22
Visión de Desarrollador	23
4. Idiomas de la aplicación	25
Visión de Administrador	25
Visión de Desarrollador	26
5. Autorización y autenticación por base de datos	27
a. Usuarios	27
Visión de Administrador	27
Visión de Desarrollador	28
b. Rol y OU asignados a los usuarios	29
Visión de Administrador	29
Visión de Desarrollador	30
Esquema Anjana de BD	31
1. Configurar entidades y relaciones	32
Visión de Administrador	34
Visión de Desarrollador	36
2. Configurar el Menú	37
Visión de Administrador	37

Visión de Desarrollador	39
3. Configurar las secciones (sections)	39
Visión de Administrador	40
Visión de Desarrollador	42
4. Definir y configurar los atributos de metadatos	42
Visión de Administrador	44
Visión de Desarrollador	47
Atributos de clave primaria	50
Atributos Obligatorios	50
5. Definir y configurar los posibles valores de los atributos	54
Visión de Administrador	54
Visión de Desarrollador	56
6. Definir qué atributos aparecerán en cada una de las secciones de los formularios según el tipo de objeto	57
Visión de Administrador	57
Visión de Desarrollador	58
7. Definir validaciones para los valores de entrada de cada atributo	59
Visión de Administrador	62
Visión de Desarrollador	65
8. Definir relaciones entre valores de atributos para pre-filtrar los reference metadata	67
Visión de Administrador	67
Visión de Desarrollador	69
9. Reglas de Versionado	69
Visión de Administrador	70
Visión de Desarrollador	71
10. Reglas de lanzamiento de workflow en ediciones y versionado.	72
Visión de Administrador	72
Visión de Desarrollador	73
Esquema Hermes de BD	74
1. Notificaciones del sistema	74
Visión de Administrador	74
Visión de Desarrollador	76
Esquema Minerva de BD	78

1. Filtros del sistema	78
Visión de Administrador	79
Visión de Desarrollador	81
Esquema Portuno de BD	82
1. Variables del sistema	82
Visión de Administrador	82
Visión de Desarrollador	83
2. Traducciones	84
Visión de Administrador	85
Visión del desarrollador	86
Traducciones internas de Anjana	87
Configuración en ficheros yml	89
Swagger	89
Kerno	89
Dritttesta	91
Viator	91
Hecate	91
Hermes	91
Minerva	92
Portuno	93
Tretesta	93
Zeus	93
Tot Plugin	94
Ejemplo de configuración de proveedor de identidades	94
Configuración de base de datos	94
Configuración de LDAP	97
Proveedores externos	101
Configuración de Batches	102
1. Comprobación de roles transversales (cross)	102
2. Licencia a punto de expirar	103
3. Expiración	103
4. Preaviso por expiración	103

5. Indexación completa de elementos	103
Preguntas frecuentes	104
1. Permisos	104
2. Anjana Portal	104
3. Lanzamiento de workflows	104
4. ¿Cómo puedo saber si se han desplegado correctamente los workflows?	104
5. Error en la solicitud de adherencia	105
6. En el formulario de mi entidad o relación aparecen atributos que no deben estar	105
7. Necesito eliminar un atributo de una plantilla para la que ya tengo objetos creados	105
8. ¿Cómo configurar una nueva plantilla en Anjana?	105
9. ¿Cómo borrar completamente campos de Solr si al borrar una colección vemos que siguen existiendo campos que no deberían?	106

Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Anjana Producto	Anjana Producto	Creación del documento
2.0	10/02/2023	Equipo Desarrollo	CPO	Configuración batch indexación Eliminado configuración de workflows por atributo Añadida configuración para gestor de ficheros

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 los formularios, roles y permisos de Anjana Data para aterrizar 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 a unidades organizativas específicas o tratarse de roles transversales (roles *cross*). Cada uno de estos roles dispondrá de un paquete de permisos a bajo nivel y cada uno de los usuarios podrá tener asignado más de un rol.

Los **Permisos** son los que habilitan a cada rol a realizar acciones a bajo nivel en Anjana Data. 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 objetos.

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.

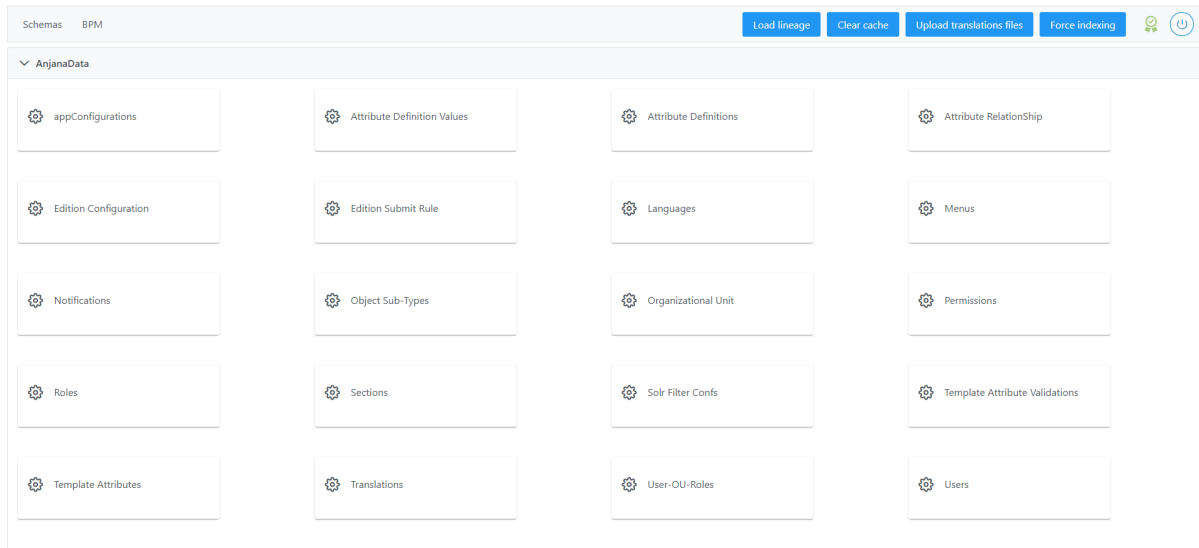
El Catálogo de Datos y el Glosario de Negocio se componen de Entidades y Relaciones. Cada una de éstas se corresponde con un **Subtipo de objeto** y para cada uno se configura en Anjana una plantilla con el conjunto de **atributos** que define su metadato.

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).

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 Anjana Data 4.4 - DOC - 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.
- 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). No se verán los cambios hasta el próximo login de los usuarios.
- Force indexing: acción que fuerza una reindexación de todas las entidades y relaciones para evitar que BD y Solr estén desincronizadas y la información de ambas no sea coherente.

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](#)
[Upload translations files](#)
[Force indexing](#)



[+ New](#)

Attribute Definitions

Id	Name	Type	Description	Label	Label Translation Key	Place Holder	Place Holder Translation Key	Short Description Translati
1	name	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.name	Name	ATTRIBUTE_NOMBRE.name	Name here	ATTRIBUTE_PLACEHOLDER.name	ATTRIBUTE_DESCRIPTION.na
2	termType	INPUT_DATE	ATTRIBUTE_DESCRIPTION.termType	Term type	ATTRIBUTE_NOMBRE.termType	Term type here	ATTRIBUTE_PLACEHOLDER.termType	ATTRIBUTE_DESCRIPTION.te
3	domain	INPUT_RANGE	ATTRIBUTE_DESCRIPTION.domain	Domain	ATTRIBUTE_NOMBRE.domain	Domain here	ATTRIBUTE_PLACEHOLDER.domain	ATTRIBUTE_DESCRIPTION.dc
4	subdomain	SELECT	ATTRIBUTE_DESCRIPTION.subdomain	Subdomain	ATTRIBUTE_NOMBRE.subdomain	Subdomain here	ATTRIBUTE_PLACEHOLDER.subdomain	ATTRIBUTE_DESCRIPTION.su
5	geography	SELECT	ATTRIBUTE_DESCRIPTION.geography	Geography	ATTRIBUTE_NOMBRE.geography	Geography here	ATTRIBUTE_PLACEHOLDER.geography	ATTRIBUTE_DESCRIPTION.ge
6	des	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.des	Description	ATTRIBUTE_NOMBRE.des	Description here	ATTRIBUTE_PLACEHOLDER.des	ATTRIBUTE_DESCRIPTION.de
7	sdescription	INPUT_TEXT	ATTRIBUTE_DESCRIPTION.sdescription	Short description	ATTRIBUTE_NOMBRE.sdescription	Short description here	ATTRIBUTE_PLACEHOLDER.sdescription	ATTRIBUTE_DESCRIPTION.sd

Schemas BPM
[Load lineage](#)
[Clear cache](#)
[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

















<< < 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 Upload translations files Force indexing  

[+ New](#)

Organizational Unit













Id ↑	Description ↑↓	Module ↑↓	alias ↑↓	parent ↑↓	organizationalUnitType ↑↓		
-	-	Igual	-	Igual	-		
Criterio	Criterio	-	Criterio	-	Criterio		
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar		
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		

Para visualizar los registros de las tablas visualizados 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.

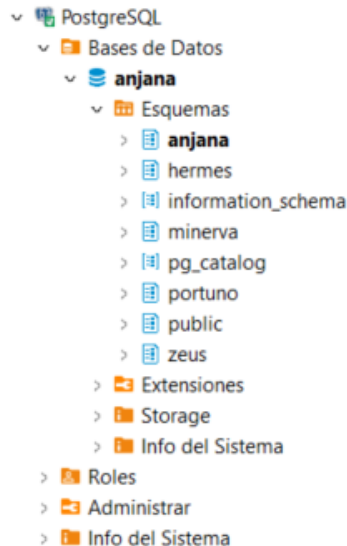
Schemas BPM Clear cache Upload translations files Force indexing  

[+ New](#)

Notifications

Id ↑	moduleType ↑↓	notificationCode ↑	notificationReceiverType ↑↓	translationKey ↑↓	notificationType ↑	receiverRole ↑↓	severity ↑↓	subject ↑↓		
-	Igual	-	Igual	-	Igual	-	Igual	-		
Criterio	-	Criterio	-	Criterio	-	Criterio	-	Criterio		
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar		
17	DC	ADHERENCE_FAIL	ROLE	NOTIFICATION.ADHEREN	ADMIN_ALERT	admin	HIGH	NOTIFICATION.SUBJECT.		
31	DC	CHECK_CONFIGURATION	ROLE	NOTIFICATION.CHECK_C	ADMIN_ALERT	admin	HIGH	Check configuration of provider		
38	DC	CHECK_CONFIGURATION	ROLE	NOTIFICATION.CHECK_C	ADMIN_ALERT	admin	HIGH	Check configuration of provider		
22	DC	DATASET_EXPIRATION_TI	ROLE	NOTIFICATION.DATASET.	ADMIN_ALERT	admin	HIGH	NOTIFICATION.SUBJECT.		
15	DC	DATASET_FAIL	ROLE	NOTIFICATION.DATASET.	ADMIN_ALERT	admin	HIGH	NOTIFICATION.SUBJECT.		
18	DC	DISADHERENCE_FAIL	ROLE	NOTIFICATION.DISADHE	ADMIN_ALERT	admin	HIGH	NOTIFICATION.SUBJECT.		

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



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

```

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_attribute_value', (SELECT COALESCE(MAX(id_attribute_value), 0)
FROM anjana.attribute_value) + 1);
SELECT setval('anjana.sc_pk_combination', (SELECT COALESCE(MAX(id_combination), 0) FROM
anjana.combination) + 1);
SELECT setval('anjana.sc_pk_dsa_ds_combination', (SELECT COALESCE(MAX(id_dsa_ds_comb), 0)
FROM anjana.dsa_ds_combination) + 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_entity', (SELECT COALESCE(MAX(id_entity), 0) FROM anjana.entity) +
1);
SELECT setval('anjana.sc_pk_entity_synonymous', (SELECT COALESCE(MAX(id_synonymous), 0)
FROM anjana.entity_synonymous) + 1);
SELECT          setval('anjana.sc_pk_imported_metadata',            (SELECT
COALESCE(MAX(id_imported_metadata), 0) FROM anjana.imported_metadata) + 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_pae', (SELECT COALESCE(MAX(id_pae), 0) FROM
anjana.pending_adherence_event) + 1);
SELECT setval('anjana.sc_pk_relationship', (SELECT COALESCE(MAX(id_relationship), 0) FROM
anjana.relationship) + 1);
SELECT setval('anjana.sc_pk_relationship_synonymous', (SELECT COALESCE(MAX(id_synonymous),

```

```

0) FROM anjana.relationship_synonymous) + 1);
SELECT setval('anjana.sc_pk_request_import', (SELECT COALESCE(MAX(id_request), 0) FROM
anjana.request_import) + 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('anjana.sc_pk_workflow_info', (SELECT COALESCE(MAX(id_workflow_info), 0) FROM
anjana.workflow_info) + 1);
SELECT setval('hermes.sc_pk_notification', (SELECT COALESCE(MAX(id_notification), 0) FROM
hermes.notification) + 1);
SELECT setval('hermes.sc_pk_sent_notification', (SELECT COALESCE(MAX(id_sent_notification), 0)
FROM hermes.sent_notification) + 1);
SELECT setval('hermes.sc_pk_task_step', (SELECT COALESCE(MAX(id_task_step), 0) FROM
hermes.task_step) + 1);
SELECT setval('hermes.sc_pk_we_ou', (SELECT COALESCE(MAX(id_we_ou), 0) FROM
hermes.workflow_ou) + 1);
SELECT setval('hermes.sc_pk_workflow_execution', (SELECT
COALESCE(MAX(id_workflow_execution), 0) FROM hermes.workflow_execution) + 1);
SELECT setval('minerva.sc_pk_solr_filter_conf', (SELECT COALESCE(MAX(id), 0) FROM
minerva.solr_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

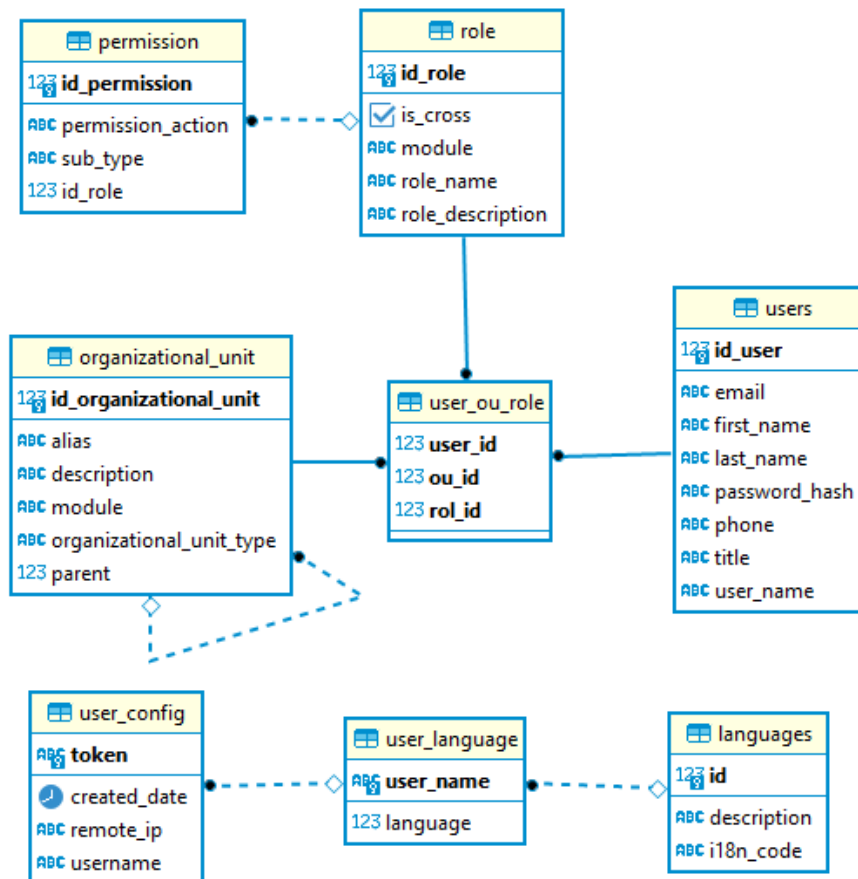
Las tablas de configuración de Anjana Data se agrupan en cuatro esquemas:

- **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 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:



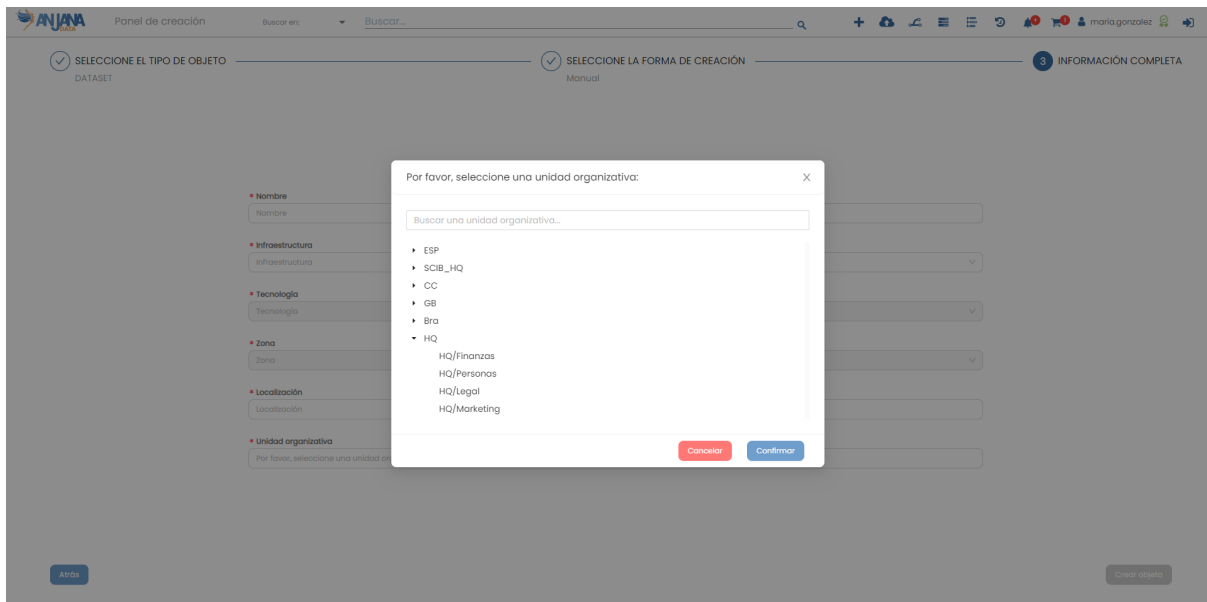
1. Definir la estructura organizativa y las unidades organizacionales

El primer paso a la hora de aterrizar el modelo de gobierno de su organización es establecer la jerarquía de unidades organizativas que su organización haya definido. Puede tratarse de una estructura “plana” o de un único nivel para el caso de, por ejemplo, una organización dividida en departamentos; por el contrario, una gran compañía multinacional dispondrá de una jerarquía de varios niveles, donde el primer nivel sean las sedes de países y los siguientes niveles sean las distintas áreas de negocio de la compañía.



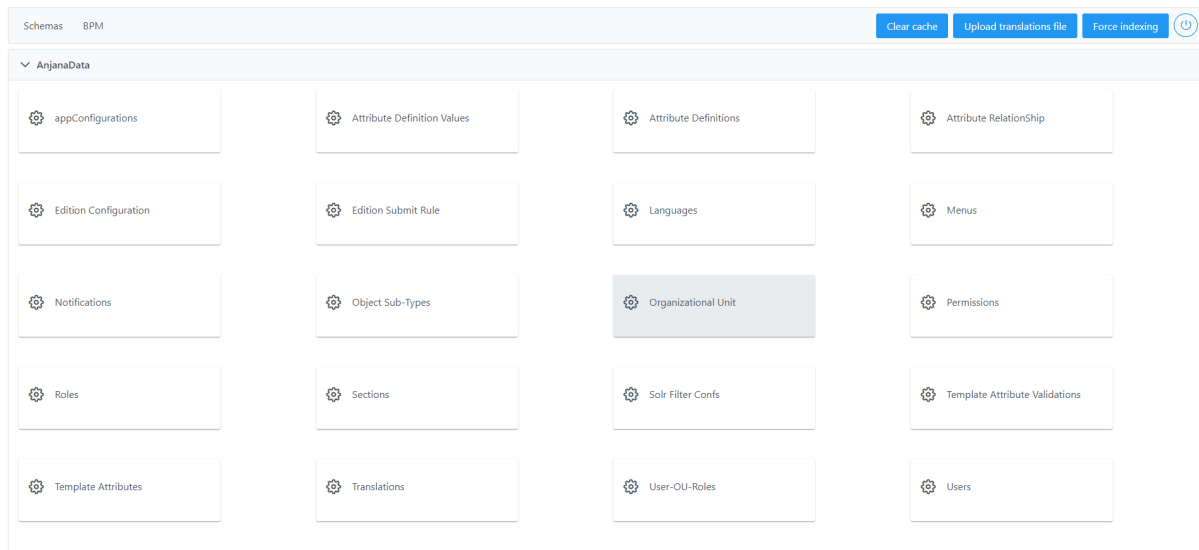
Una vez definidos los niveles de la jerarquía de unidades organizacionales, se deben asignar nombres a cada uno de los niveles conforme al modelo de gobierno de la organización.

Las unidades organizacionales que se establezcan serán visibles dentro de Anjana Data mediante los wizards de creación y en el linaje de datos:



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. Cada unidad organizacional existente se caracteriza por los siguientes elementos:

- **Id:** identificador único de la unidad organizacional.
- **Description:** descripción de la unidad organizacional.
- **Module:** módulo al que pertenece, indicar “BG” para *Business Glossary* o “DC” para *Data Catalog*. 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.
 IMPORTANTE: 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.
 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.
- **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.

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

Organizational Unit						+ New
Id ↑	Description ↑↓	Module ↑↓	alias ↑↓	parent ↑↓	organizationalUnitType ↑↓	
-	-	Igual	-	Igual	-	
Criterio	Criterio	-	Criterio	-	Criterio	
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	
1	SPA	BG	SPA		1	 
2	Finance	BG	SPA/Finance	1-SPA	2	 
3	Marketing	BG	SPA/Marketing	1-SPA	2	 
4	Commercial	BG	SPA/Commercial	1-SPA	2	 
5	Risk	BG	SPA/Risk	1-SPA	2	 
6	Market	BG	SPA/Market	1-SPA	2	 
7	Accounting	BG	SPA/Accounting	1-SPA	2	 
8	SCIB_HQ	BG	SCIB_HQ		1	 

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 ✕

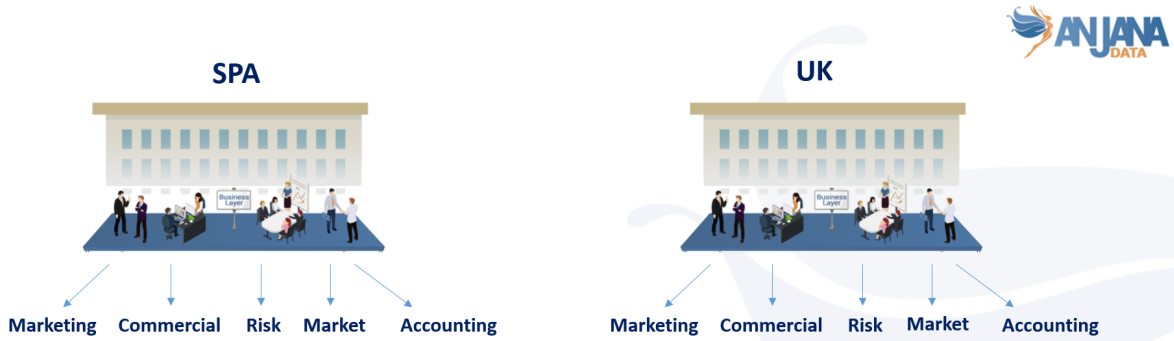
Module: alias: parent:

Description:

organizationalUnitType:

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 el siguiente SQL:

```

INSERT INTO zeus.organizational_unit
(id_organizational_unit,alias,description,module,parent,organizational_unit_type) VALUES
(2,'SPA/Finance','Finance','BG',1,'2'),
(1,'SPA','SPA','BG',NULL,'1'),
(3,'SPA/Marketing','Marketing','BG',1,'2'),
(4,'SPA/Commercial','Commercial','BG',1,'2'),
(5,'SPA/Risk','Risk','BG',1,'2'),
(6,'SPA/Market','Market','BG',1,'2'),
(7,'SPA/Accounting','Accounting','BG',1,'2'),
(8,'UK','UK','BG',NULL,'1'),
(9,'UK/Finance','Finance','BG',22,'2'),
(10,'UK/Marketing','Marketing','BG',22,'2'),
(11,'UK/Commercial','Commercial','BG',22,'2'),
(12,'UK/Risk','Risk','BG',22,'2'),
(13,'UK/Market','Market','BG',22,'2'),
(14,'UK/Accounting','Accounting','BG',22,'2');

```

Donde:

- **id_organizational_unit:** (*primary key* de la tabla) es el identificador único de la unidad organizacional.
- **alias:** alias de la unidad organizacional. Este alias debe coincidir exactamente con el nombre usado en los proveedores de identidades para la unidad.

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.

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.

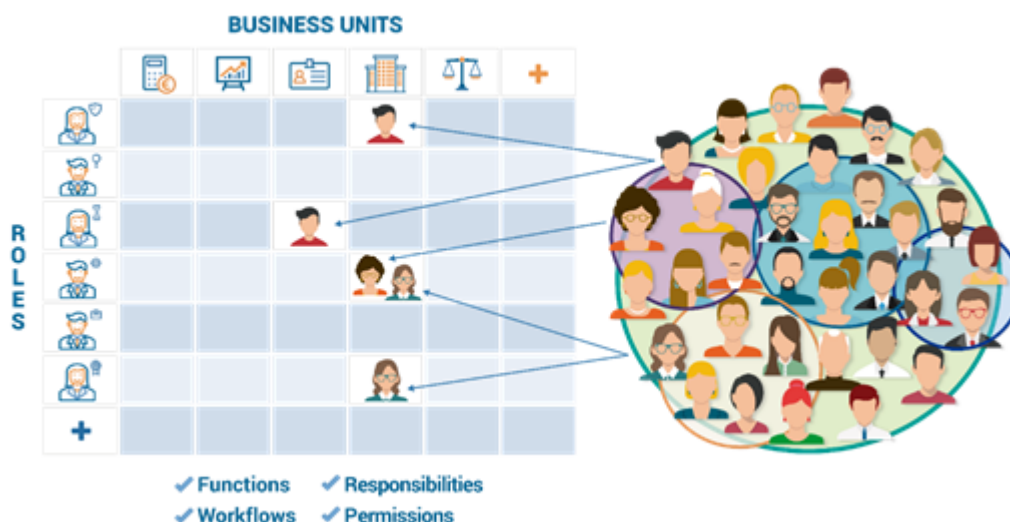
- **description:** descripción de la unidad organizacional.
- **module:** módulo al que pertenece la unidad. Si pertenece al glosario de negocio (“BG”) o al catálogo de datos (“DC”).
- **parent:** (*foreign key* contra el id de la misma tabla) indica quién es la unidad organizacional padre. Se corresponde con el *id_organizational_unit* de la unidad padre o nulo en caso de que sea la jerarquía más alta.
- **organizational_unit_type:** indica a qué nivel de la jerarquía pertenece.

NOTAS:

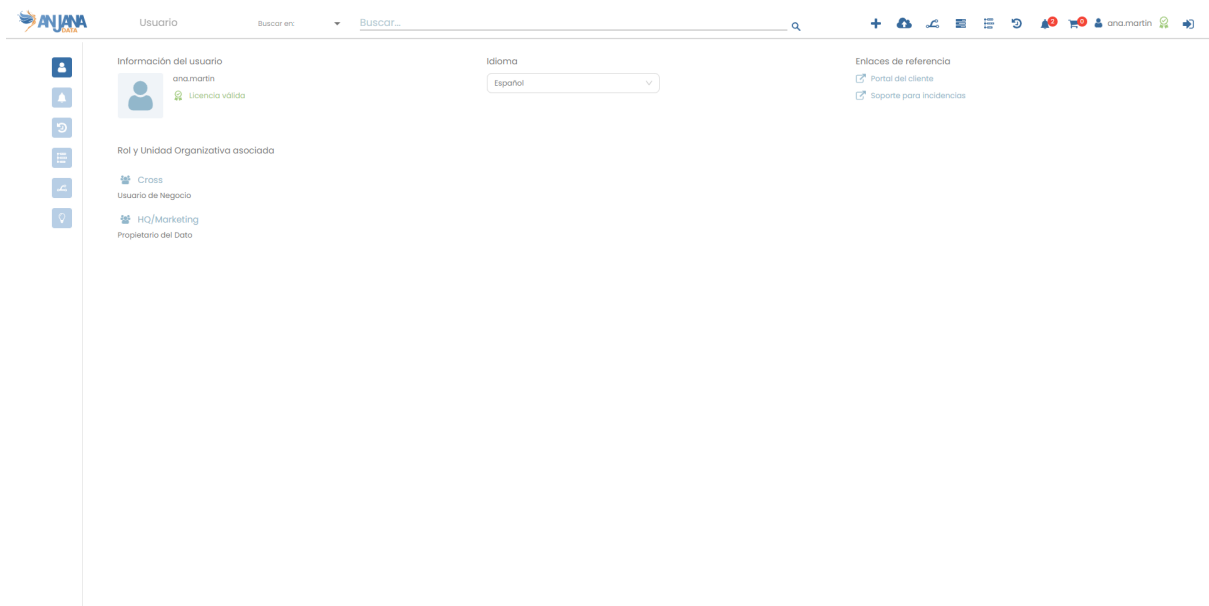
- El nombre y el alias de las unidades organizativas no deben contener los caracteres: “:” ni “#” para no interferir con los identificadores internos de Anjana.
- Cada una de las partes de las que se compone el alias de la unidad organizacional (concatenándolas con “/”) debe coincidir exactamente con 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.

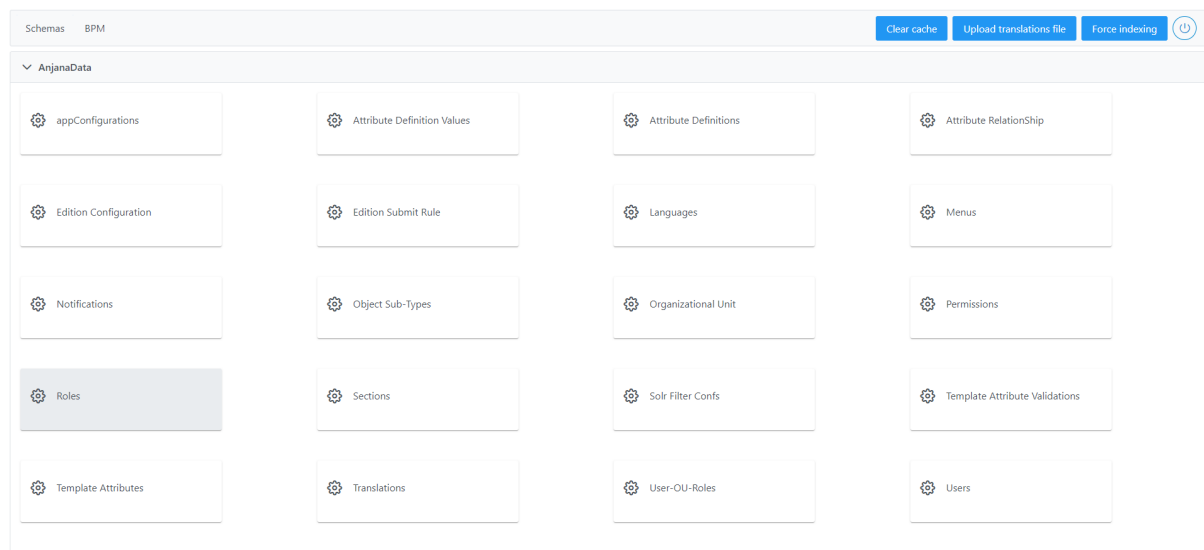


Dentro de Anjana Data, deben definirse roles para el **Catálogo de Datos** y para el **Glosario de Negocio**. Los roles pueden estar asociados a unidades organizativas específicas o tratarse de roles transversales.



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. Cada rol existente se caracteriza por los siguientes elementos:

- **Id:** Identificador único del rol.
- **Module:** Módulo al que pertenece, indicar "BG" para *Business Glossary* o "DC" para *Data Catalog*. 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*).

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

+ New

Roles					
Id	Module	Name	Description	isCross	
-	Igual	-	-		
<input type="text" value="Criterio"/> Filtrar	<input type="text" value="Criterio"/> Filtrar	<input type="text" value="Criterio"/> Filtrar	<input type="text" value="Criterio"/> Filtrar	<input type="checkbox"/>	
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

Chief Data Officer

✕ Cancel

✓ Save

Visión de Desarrollador

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

```
INSERT INTO zeus.role
(id_role, is_cross,module, role_name, role_description)
VALUES
(1, true, 'DC', 'developer', 'Developer')
,(2, true, 'DC', 'architect', 'Architect')
,(3, false, 'DC', 'administrator', 'Administrator')
,(4, true, 'DC', 'data_steward', 'Data Steward')
,(5, false, 'DC', 'data_owner', 'Data Owner')
,(6, true, 'DC', 'compliance', 'Compliance')
,(7, false, 'DC', 'process_owner', 'Process Owner')
,(8, true, 'DC', 'business_user', 'Business User')
,(9, false, 'BG', 'containerOwner', 'Role for container owner')
,(10, false, 'BG', 'contributor', 'Role for responsible of entity or relationship')
,(12, true, 'BG', 'bg_user', 'Role for read a entity or relationship')
,(13, true, 'BG', 'sync_service', 'Role for sync service')
,(14, true, 'BG', 'admin', 'Administrator')
,(15, true, 'ALL', 'anjanaadmin', 'anjana admin')
,(16, true, 'ALL', 'anjanacredentialadmin', 'anjana credential admin');
```

Donde:

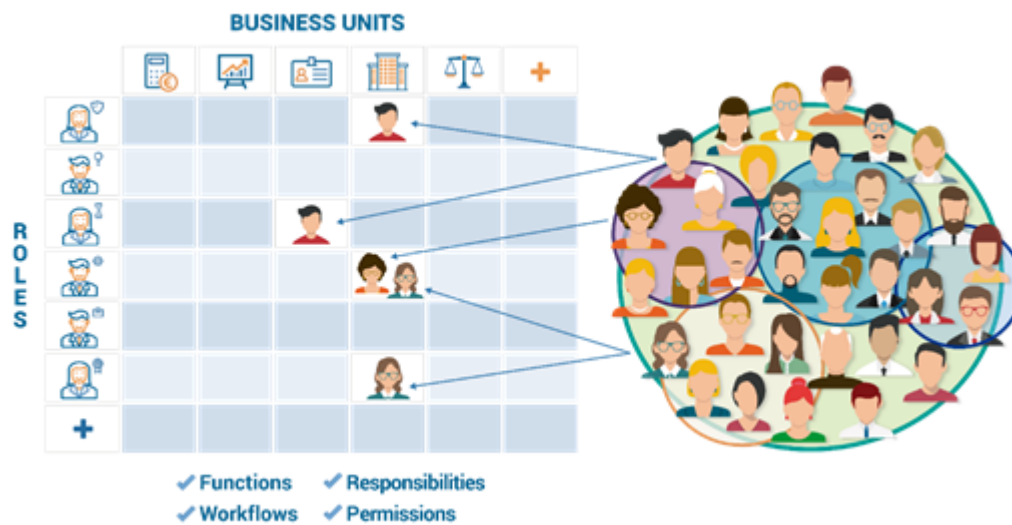
- **id_role**: (*primary key*) identificador único de la tabla.
- **is_cross**: *flag* que indica si se trata de un rol transversal (*cross*).
- **module**: campo que indica si pertenece al *Business Glossary* ("BG") o al *Data Catalog* ("DC").
- **role_name**: nombre del rol. Este nombre debe ser idéntico 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`.
- **rol_description**: descripción del rol.

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 y que, además, sea clave de traducción en la tabla `portuno.translations`.

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.



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 son los siguientes:

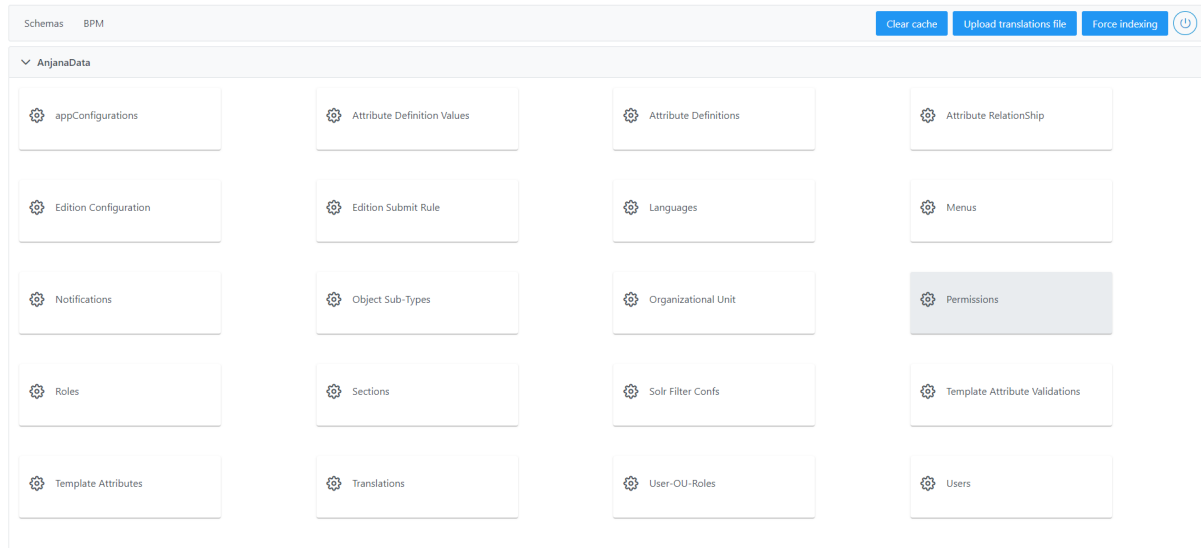
- **Creation & Modification:** que habilitan a crear y modificar objetos respectivamente.
- **Deprecation:** que habilita a deprecar manualmente objetos.
- **Change status:** que habilita a activar y desactivar objetos.
- **Organizational_unit_owner:** que habilita como propietario de aquellos objetos que se encuentren dentro de la unidad organizativa.
- **Change Organizational Unit:** que habilita a modificar la unidad organizativa de un objeto.

Además, algunos permisos afectan a toda la aplicación:

- **Access:** que habilita el acceso a Anjana Data y la visualización de los objetos.
- **Workflow_access:** que habilita la visualización (y, por tanto, también aprobación o rechazo) de los flujos(*workflows*) de validación.
- **Lineage_access:** que habilita la visualización del linaje.
- **Search:** que habilita la posibilidad de realizar búsquedas dentro de Anjana Data.
- **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 & Modification de alguna entidad o relación.
- **Automatic Metadata:** que habilita la creación de objetos en Anjana Data de forma asistida mediante el descubrimiento y la importación automática de metadatos de los sistemas origen.
- **Api admin:** que habilita el uso de la API administrativa. Usuarios con este permiso, además, pueden editar masivamente objetos por el Portal de Anjana.
- **Admin:** que habilita el acceso al portal de administración de Anjana.
- **Credential admin:** que habilita el acceso a la configuración de credenciales de TOT y de usuarios y asignación de rol para autenticación y autorización por BD.

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. Cada uno de los permisos se caracteriza por los siguientes elementos:

- **Id:** identificador único del permiso.
- **Action:** indica la acción para la cual se está dando permiso de entre las de la siguiente lista:
 - ORGANIZATIONAL_UNIT_OWNER
 - WIZARD
 - CHANGE_OU
 - SEARCH
 - AUTOMATIC_METADATA
 - WORKFLOW_ACCESS
 - ACCESS
 - CREATION_MODIF
 - CHANGE_STATUS
 - LINEAGE_ACCESS
 - DEPRECATION
 - ADHERENCE
 - ADMIN
 - API_ADMIN
 - CREDENTIAL_ADMIN
- **SubType:** indica el objeto del Catálogo de Datos o del Glosario de Negocio sobre el que se están otorgando permisos (TERM, DATASET, REPORT, DSA, ...).

Para los **Action** ACCESS, WIZARD, SEARCH, LINEAGE_ACCESS, WORKFLOW_ACCESS y API_ADMIN el **SubType** debe ser "ALL".

Además, para el permiso de adherencia se debe configurar un permiso con action = "ACCESS" y subtype = "ADHERENCE".

- **Role:** indica el rol al que se le están otorgando permisos.

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

+ New

Permissions				
id ↑	action ↓	subType ↓	role ↓	
-	-	-	Igual	
Criterio	-	-	-	
Filtrar	Filtrar	Filtrar	Filtrar	
1	ACCESS	ALL	1-developer-Developer	✎ ✖
2	ACCESS	ALL	2-architect-Architect	✎ ✖
3	ACCESS	ALL	3-administrator-Administrator	✎ ✖
4	ACCESS	ALL	4-data_steward-Data Steward	✎ ✖
5	ACCESS	ALL	5-data_owner-Data Owner	✎ ✖
6	ACCESS	ALL	6-compliance-Compliance	✎ ✖
7	ACCESS	ALL	7-process_owner-Process Owner	✎ ✖
8	ACCESS	ALL	8-business_user-Business User	✎ ✖

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, rellenar el siguiente sql:

```

INSERT INTO zeus.permission
(id_permission, permission_action, sub_type, id_role)
VALUES
(1, 'ACCESS', 'ALL', 1)
,(2, 'ACCESS', 'ALL', 2)
,(3, 'ACCESS', 'ALL', 3)
,(4, 'ACCESS', 'ALL', 4)
,(5, 'ACCESS', 'ALL', 5)

```



```

,(6, 'ACCESS', 'ALL', 6)
,(7, 'ACCESS', 'ALL', 7)
,(8, 'ACCESS', 'ALL', 8)
,(9, 'ACCESS', 'ALL', 9)
,(10, 'ACCESS', 'ALL', 10)
,(11, 'ACCESS', 'ALL', 12)
,(12, 'ACCESS', 'ALL', 13)
,(13, 'ACCESS', 'ALL', 14)
,(14, 'SEARCH', 'ALL', 1)
,(15, 'SEARCH', 'ALL', 2)
,(16, 'SEARCH', 'ALL', 3)
,(17, 'SEARCH', 'ALL', 4)
,(18, 'SEARCH', 'ALL', 5)
,(19, 'SEARCH', 'ALL', 6)
,(20, 'SEARCH', 'ALL', 7)
,(21, 'SEARCH', 'ALL', 8)
,(22, 'SEARCH', 'ALL', 9)
,(23, 'SEARCH', 'ALL', 10)
,(24, 'SEARCH', 'ALL', 12)
,(25, 'SEARCH', 'ALL', 13)
,(26, 'SEARCH', 'ALL', 14);

```

Donde:

- **id_permission:** (*primary key*) identificador único de la tabla.
- **permission_action:** indica la acción para la cual se está dando permiso de entre las de la siguiente lista:
 - ORGANIZATIONAL_UNIT_OWNER
 - WIZARD
 - CHANGE_OU
 - SEARCH
 - AUTOMATIC_METADATA
 - WORKFLOW_ACCESS
 - ACCESS
 - CREATION_MODIF
 - CHANGE_STATUS
 - LINEAGE_ACCESS
 - DEPRECATION
 - ADHERENCE
 - ADMIN
 - API_ADMIN
 - CREDENTIAL_ADMIN
- **sub_type:** indica el objeto del catálogo de datos o del glosario de negocio sobre el que se están otorgando permisos (TERM, DATASET, REPORT, DSA, ...).

Para los **Action** ACCESS, WIZARD, SEARCH, LINEAGE_ACCESS, WORKFLOW_ACCESS y API_ADMIN el **SubType** debe ser "ALL".

Además, para el permiso de adherencia se debe configurar un permiso con action = “ACCESS” y subtype = “ADHERENCE”.

- **id_role:** (*foreign key* con el identificador de la tabla *Role*) indica el rol al que se le están otorgando permisos.

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

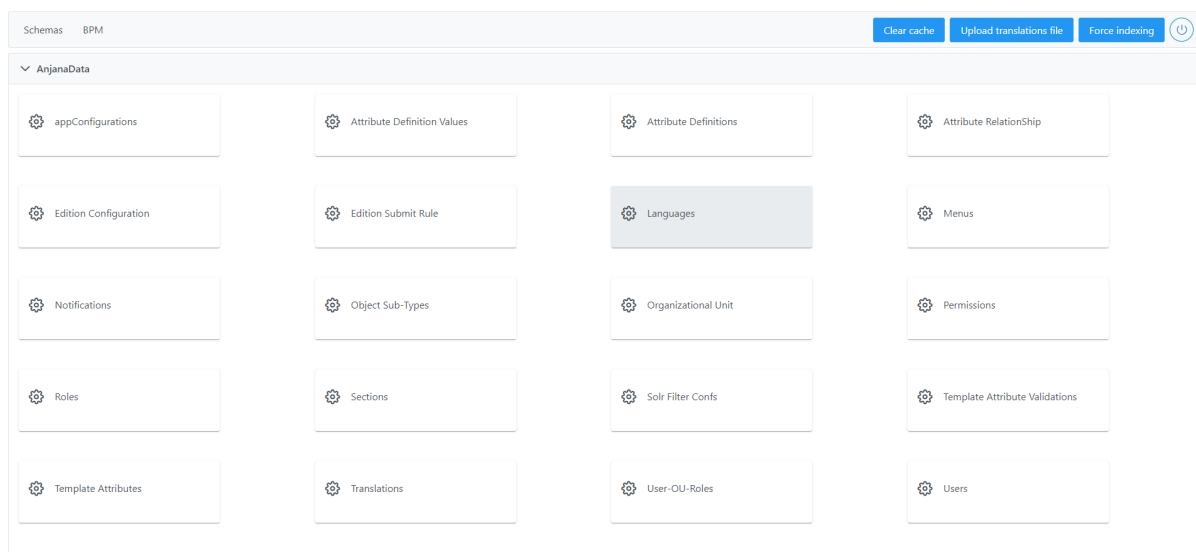
4. Idiomas de la aplicación

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 hacer uso de ellos.

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. Cada idioma registrado se caracteriza por los siguientes elementos:

- **Id:** identificador único de la tabla.
- **I18nCode:** código i18n para el idioma.
- **Description:** nombre del idioma.

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

+ New			
Languages			
id ↑	i18nCode ↑↓	Description ↑↓	
-	-	-	
Criterio	Criterio	Criterio	
Filtrar	Filtrar	Filtrar	
1	en-US	Inglés	 
2	es-ES	Español	 
<< < 1 > >> 20 ▾			

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

Languages Edit ✕

i18nCode

Description

English

✕ Cancel
✓ Save

Visión de Desarrollador

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

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

Consta de los siguientes campos:

- **id:** (*primary key*) identificador único, sirve para gestión interna de Anjana.
- **i18n_code:** código i18n para el idioma.
- **description:** descripción del idioma.

5. Autorización y autenticación por base de datos

Es posible configurar en Anjana la autenticación y autorización de los usuarios en base de datos. 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.

Para activar/desactivar la autenticación y autorización en base de datos es necesario usar los flags del yml de zeus:

- security.authentication.db.enable
- security.authorization.db.enable

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>

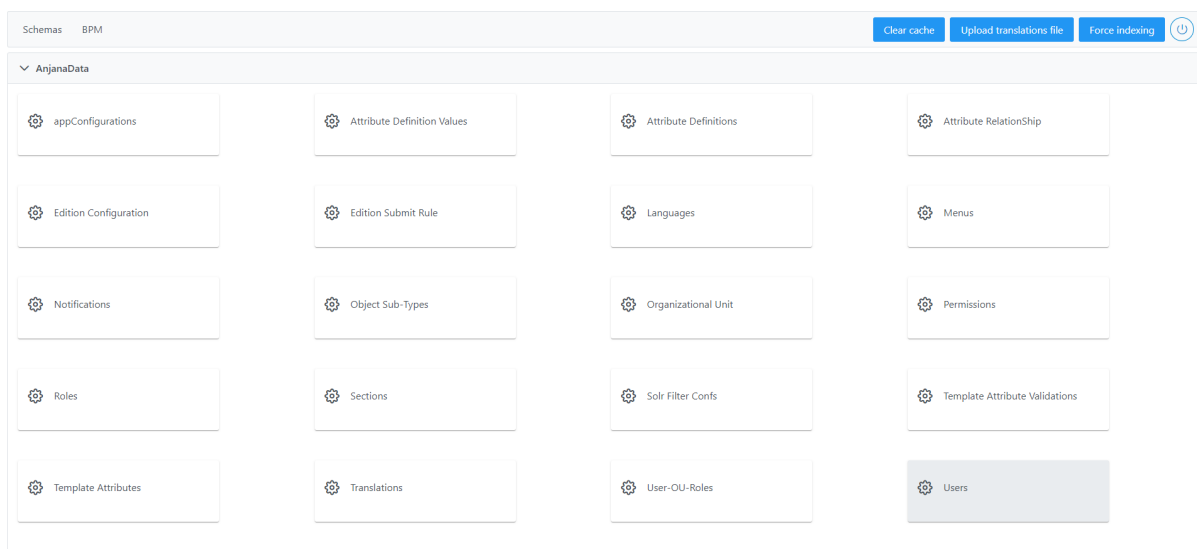
De esta forma no se incluye en claro la contraseña en la tabla 'users' sino que es necesario calcular un hash e introducirlo en la columna 'password_hash'.

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

Visión de Administrador

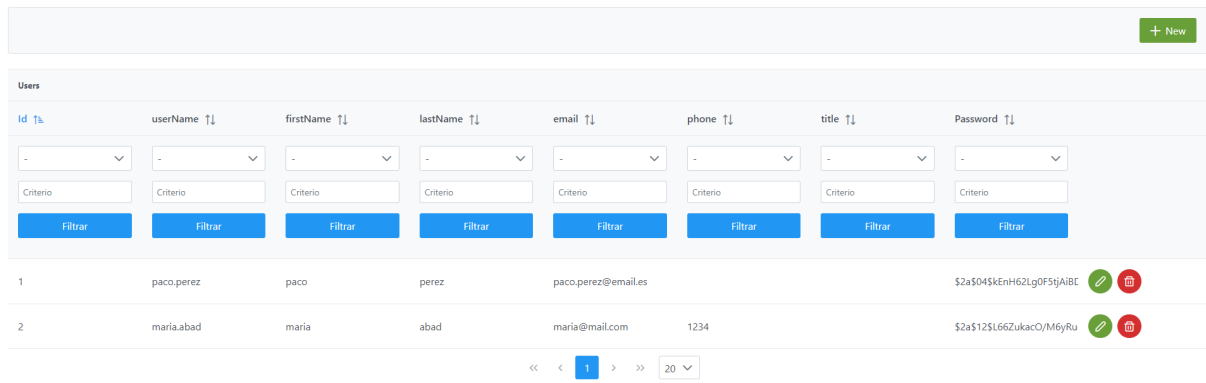
La tabla donde se configuran los usuarios aparece en el portal de administración de Anjana:



Al acceder a la tabla de Users se muestra una tabla que contiene todos los usuarios existentes en la configuración actual. Cada usuario registrado se caracteriza por los siguientes elementos:

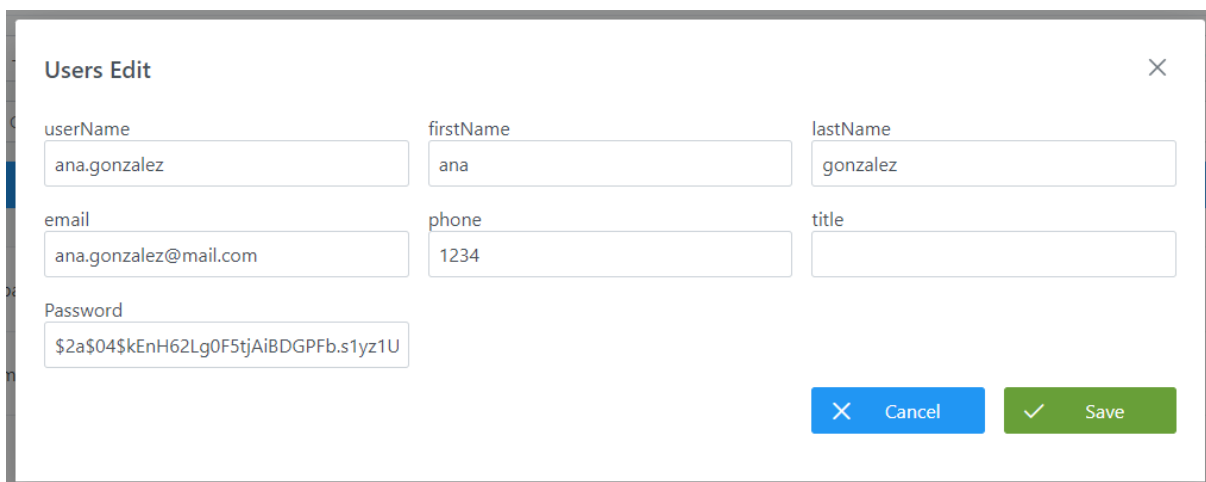
- **id:** (*primary key*) identificador único, sirve para gestión interna de Anjana.
- **userName:** Identificador único del usuario, es el valor que hay que introducir en el apartado 'Username' cuando hacemos login en Anjana.
- **firstName:** Nombre del usuario.
- **lastName:** Apellidos del usuario.
- **email:** Dirección de correo del usuario.
- **phone:** Número de teléfono del usuario.
- **title:** Título del usuario, no es necesario rellenarlo.
- **password:** Hash de la contraseña del usuario. Para obtenerlo se puede usar cualquier calculadora de Bcrypt de internet, el hash obtenido es único.

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



id	userName	firstName	lastName	email	phone	title	Password
1	paco.perez	paco	perez	paco.perez@email.es			\$2a\$04\$kenH62Lg0F5tjAiBC
2	maria.abad	maria	abad	maria@mail.com	1234		\$2a\$12\$166ZukacO/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 ✕

userName:

firstName:

lastName:

email:

phone:

title:

Password:

Visión de Desarrollador

Se pueden incluir usuarios con la siguiente secuencia sql:

```

INSERT INTO zeus.users
(id_user,email,first_name,last_name,password_hash,phone,title,user_name)
VALUES
(1,'paco.perez@email.es','paco','perez','$2a$04$KEnH62Lg0F5tjAiBDGPFb.s1yz1UU13CDXM5iFb5i
KZB8V0Gveyq.',5678,NULL,'paco.perez'),
(2,'maria@mail.com','maria','abad','$2a$12$L66ZukacO/M6yRuggtlvXufzN5KlpdNARhOtDazBYqUx
94zp4b3P2','1234',NULL,'maria.abad');

```

Donde:

- **id_user:** (primary key) identificador único, sirve para gestión interna de Anjana.
- **email:** Dirección de correo del usuario.
- **first_name:** Nombre del usuario.
- **last_name:** Apellidos del usuario.
- **password_hash:** Hash de la contraseña del usuario. Para obtenerlo se puede usar cualquier calculadora de Bcrypt de internet, el hash obtenido es único.
- **phone:** Número de teléfono del usuario.
- **title:** Título del usuario, no es necesario rellenarlo.
- **user_name:** Identificador único del usuario, es el valor que hay que introducir en el apartado 'Username' cuando el usuario hace login en Anjana.

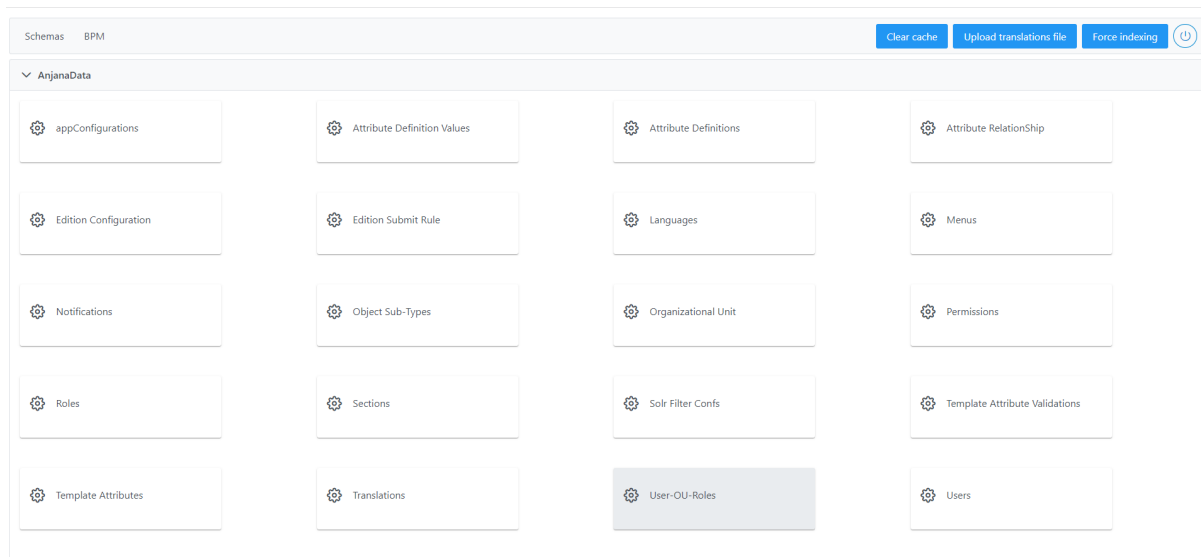
NOTAS:

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

b. Rol y OU asignados a los usuarios

Visión de Administrador

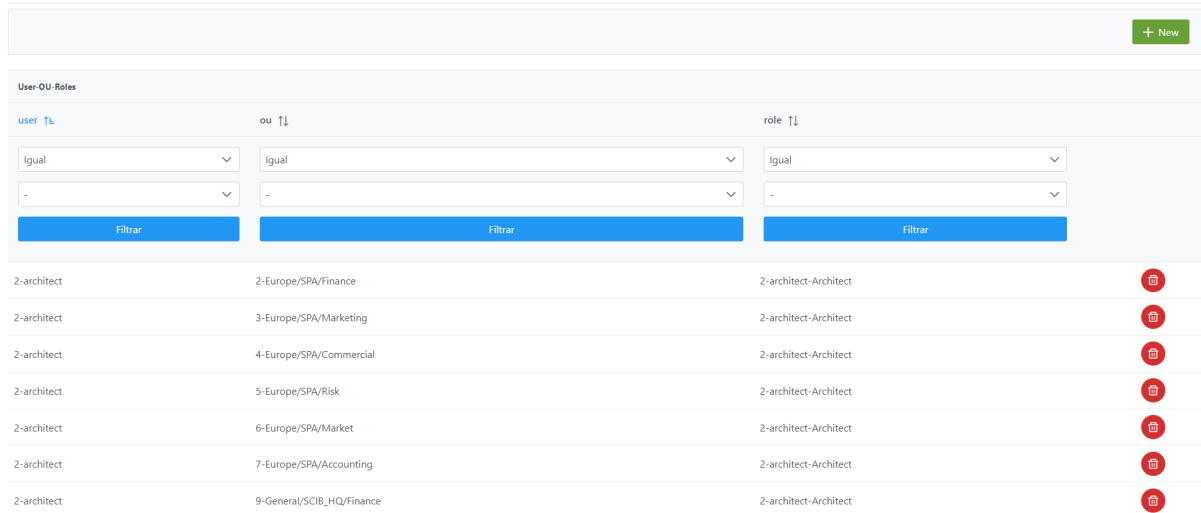
La tabla donde se asignan roles y OUs a los usuarios aparece en el portal de administración de Anjana:










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. Cada registro de esta tabla se caracteriza por los siguientes elementos:

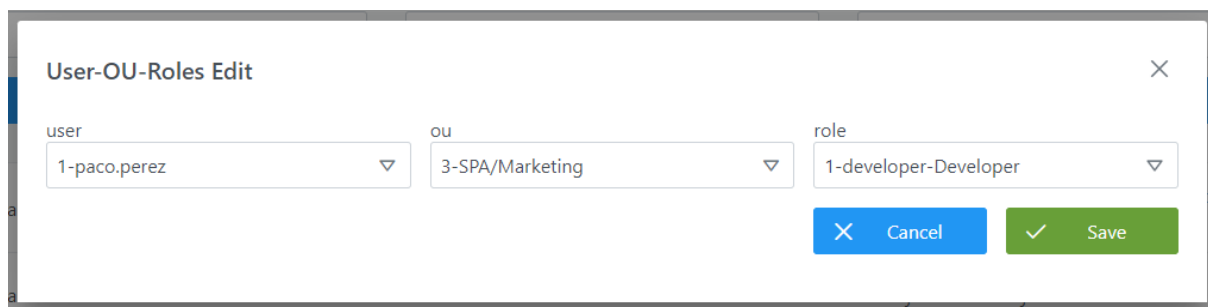
- **user:** Nombre del usuario, es el campo userName de la tabla de users.
- **ou:** Alias de la unidad organizativa, es el campo alias de la tabla organizational_unit.
- **role:** Nombre del rol, es el campo role_name de la tabla role.

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



user	ou	role	
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.



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.

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 el siguiente sql:

```
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);
```

Donde los campos son:

- **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.
- **rol_id:** Identificador del rol, es el campo id_role de la tabla role.

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.

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:

- **Solution:** objeto lógico que agrupa una o más instancias para poder conocer el flujo completo de los datos de una aplicación

En caso de que el usuario considere necesario añadir entidades o relaciones adicionales al metamodelo del catálogo, es posible gracias a la flexibilidad del metamodelo de Anjana Data, por ejemplo, se podría añadir la entidad Vista. En este caso, lo único que tiene que tener en cuenta el usuario es que las relaciones entre esta nueva entidad y el resto de entidades del metamodelo no serán intrínsecas como ocurre por ejemplo entre un dataset field y el dataset; deberán definirse manualmente.

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 relacionan 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 que se den de alta se visualizarán en los wizard de creación:

Panel de creación Buscar en: Buscar...

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...

DATASET	DSA	INSTANCE
METADATO	PROCESS	QA_BG
SOLUTION	BUSINESS_PROCESS	DATA_QUALITY_RULE
DIMENSION	API	POLICY
REPORT	TERM	

Buscar una relación...

ASSOCIATED_BUSINESS_PROCESSES	ASSOCIATED_TERMS	DEFAULT
RELATED_DATASET	RELATED_DQ_RULES	RELATED_REPORTS
REPORT_ATTRIBUTES		

Visión de Administrador

El alta de entidades o relaciones mediante el Panel de Administrador de Anjana Data se realiza desde el esquema de **Anjana**, en la tabla **Object Subtype**. Esta tabla también se utiliza para configurar los Workflows para cada tipo de entidad (esta parte se verá en detalle en el apartado “configurar la llamada a los Workflows”)

Schemas BPM Load lineage Clear cache Upload translations files Force indexing

▼ AnjanaData

appConfigurations	Attribute Definition Values	Attribute Definitions	Attribute Relationship
Edition Configuration	Edition Submit Rule	Languages	Menus
Notifications	Object Sub-Types	Organizational Unit	Permissions
Roles	Sections	Solr Filter Confs	Template Attribute Validations
Template Attributes	Translations	User-OU-Roles	Users

Al acceder se muestra una tabla que contiene todas las validaciones existentes en la configuración actual. Cada validación registrada se caracteriza por los siguientes elementos:

- **id:** Identificador único de la tabla.
- **name:** Nombre del subtipo de objeto
- **module:** Módulo al que pertenece, pudiendo escoger entre BG para Business Glossary, DC para Data Catalog
- **typeObject:** Indicar si se trata de una entidad (ENTITY) o una relación (RELATIONSHIP)

- **activateWorkflow:** Nombre del workflow de activación de entidades no nativas y relaciones.
- **createWorkflow:** Nombre del workflow de creación
- **deactivateWorkflow:** Nombre del workflow de desactivación de entidades no nativas y relaciones
- **modifyWorkflow:** Nombre del workflow de edición
- **transferWorkflow:** Nombre del workflow de transferencia (cambio de unidad organizativa) de entidades.
- **deprecateWorkflow:** Nombre del workflow de depreciación de entidades nativas del catálogo de datos
- **wfRoleDependent:** Flag que indica si el workflow tiene configuraciones diferentes según quien lance el workflow, es decir, si cambia dependiendo del rol (true) o siempre tiene los mismos validadores independientemente de quién lo mande a validar (false)

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	wfRoleDependent
1	TERM	BG	ENTITY	WF_BG	WF_BG	WF_BG	WF_BG	TRANSFER_ENTITY		✗
2	DEFAULT	BG	RELATIONSHIP	WF_BG	WF_BG	WF_BG	WF_BG			✗
3	DATASET	DC	ENTITY	WF_BG	prueba_bifurcacion	WF_BG	prueba_bifurcacion	TRANSFER_ENTITY	WF_DC	✗
4	DATASET_FIELD	DC	ENTITY							✗
5	DSA	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	TRANSFER_ENTITY	WF_DC	✗
6	PROCESS	DC	ENTITY	WF_BG	WF_PROCESS	WF_BG	WF_PROCESS	TRANSFER_ENTITY	WF_PROCESS	✓
7	INSTANCE	DC	ENTITY	WF_BG	WF_DC	WF_BG	WF_DC	TRANSFER_ENTITY	WF_DC	✗

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

typeObject
activateWorkflow
createWorkflow
deactivateWorkflow
modifyWorkflow
transferWorkflow

Object Sub-Types Edit ✕

Name
DOCUMENT

activateWorkflow
activateDocument

modifyWorkflow
createDocument

wfRoleDependent

Module
DC

createWorkflow
createDocument

transferWorkflow
transferDocument

typeObject
ENTITY

deactivateWorkflow
deactivateDocument

deprecateWorkflow

✕ Cancel
✓ Save

Junto con la definición de la nueva entidad, se definen los nombres de los workflows que se lanzarán para validar 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 el siguiente sql:

```
INSERT INTO anjana.object_subtype
(id_object_subtype, activate_workflow, create_workflow, deactivate_workflow, modify_workflow,
module, name, object_type, transfer_workflow, wf_role_dependent, deprecate_workflow)
values
(1, 'ACTIVATE_ENTITY', 'CREATE_ENTITY', 'DEACTIVATE_ENTITY', 'MODIFY_ENTITY', 'BG', 'TERM',
'ENTITY', 'TRANSFER_ENTITY', false, NULL)
,(3, 'ACTIVATE_ENTITY', 'CREATE_DATASET', 'DEACTIVATE_ENTITY', 'EDIT_DATASET', 'DC', 'DATASET',
'ENTITY', 'TRANSFER_ENTITY', true, 'DEPRECATE_DATASET')
,(4, NULL, NULL, NULL, NULL, 'DC', 'DATASET_FIELD', 'ENTITY', NULL, false, NULL)
,(5, 'ACTIVATE_ENTITY', 'CREATE_DSA', 'DEACTIVATE_ENTITY', 'EDIT_DSA', 'DC', 'DSA', 'ENTITY',
'TRANSFER_ENTITY', true, 'DEPRECATE_DSA')
,(10, 'ACTIVATE_ENTITY', 'CREATE_ENTITY', 'DEACTIVATE_ENTITY', 'MODIFY_ENTITY', 'BG',
'REPORT', 'ENTITY', 'TRANSFER_ENTITY', false, NULL)
,(16, 'ACTIVATE_RELATIONSHIP', 'CREATE_RELATIONSHIP', 'DEACTIVATE_RELATIONSHIP',
'MODIFY_RELATIONSHIP', 'BG', 'ASSOCIATED_TERMS', 'RELATIONSHIP', NULL, false, NULL)
,(17, 'ACTIVATE_RELATIONSHIP', 'CREATE_RELATIONSHIP', 'DEACTIVATE_RELATIONSHIP',
'MODIFY_RELATIONSHIP', 'BG', 'ASSOCIATED_BUSINESS_PROCESSES', 'RELATIONSHIP', NULL, false,
NULL);
```

Donde:

- **id_object_subtype:** Identificador único de la tabla.
- **name:** Nombre del subtipo de objeto
- **module:** Módulo al que pertenece, pudiendo escoger entre BG para Business Glossary, DC para Data Catalog
- **object_type:** Indicar si se trata de una entidad (ENTITY) o una relación (RELATIONSHIP)
- **activateWorkflow:** Nombre del workflow de activación de entidades no nativas y relaciones.
- **createWorkflow:** Nombre del workflow de creación
- **deactivateWorkflow:** Nombre del workflow de desactivación de entidades no nativas y relaciones.
- **modifyWorkflow:** Nombre del workflow de edición
- **transferWorkflow:** Nombre del workflow de transferencia (cambio de unidad organizativa) de entidades.
- **deprecateWorkflow:** Nombre del workflow de deprecación de entidades nativas del catálogo de datos (datasets, datasets_fields, instancias...)
- **wfRoleDependent:** Flag que indica si el workflow tiene configuraciones diferentes según quien lance el workflow, es decir, si cambia dependiendo del rol (true) o siempre tiene los mismos validadores independientemente de quién lo mande a validar (false)

NOTAS:

- El nombre de los subtipos debe no contener ‘:’, ‘#’, ‘(’ o ‘)’ para no interferir con los identificadores internos de Anjana y deben estar escritos en mayúsculas
- Los nombres de los subtipos no tienen traducción

2. Configurar el Menú

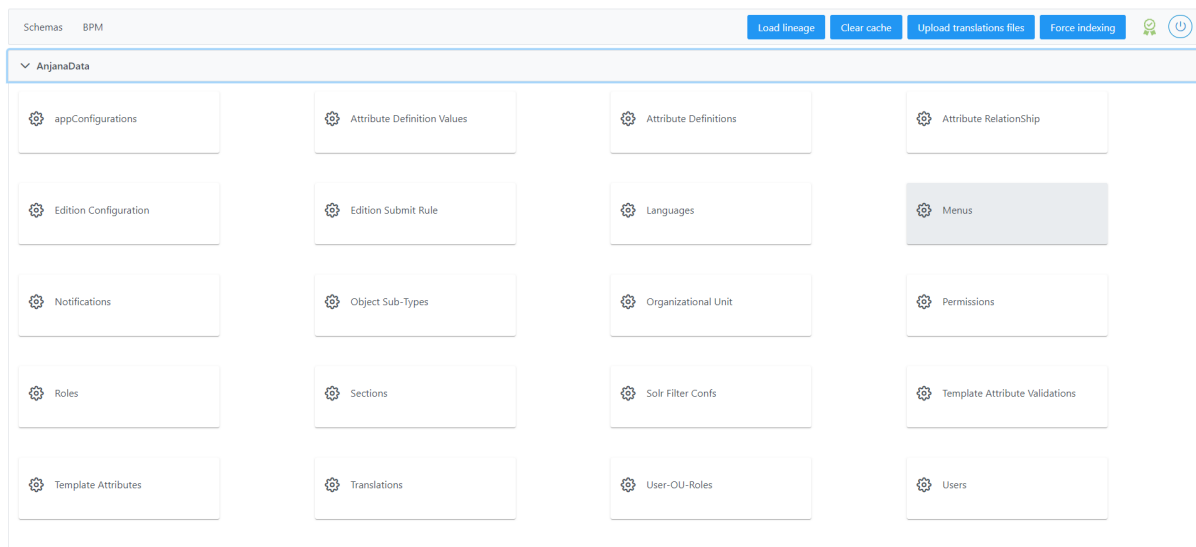
Los menús son las pestañas seleccionables que se sitúan en los formularios dinámicos de Anjana Data. Son totalmente configurables, pudiendo añadir cuantos menús se desee en cada plantilla de los diferentes tipos de objeto (entidad o relación).

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.



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. Cada menú registrado se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla
- **Name:** Nombre del menú desde 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 del menú. Es una buena práctica indicar a qué objeto corresponde.
- **Order:** Indica el orden en que se muestran los distintos menús para un mismo objeto
- **Object Subtype:** Indica el objeto (object_subtype) para el cual se define el menú.

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

+ New

Menus					
id	Name	Description	order	objectSubType	
-	-	-	-	Igual	
Criteria	Criteria	Criteria	Criteria	-	
Filtrar Filtrar Filtrar Filtrar Filtrar					
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 configurar los menús, rellenar el siguiente sql:

```
insert into anjana.menu
(id_menu,name,description,order_menu,id_object_subtype) values
(1,'DETAILS','General information of Business Term (Functional,Technical,operational)',1,1)
,(2,'DETAILS','General information of Reports (Functional,Technical,operational)',1,10)
,(14,'DETAILS','General information of DATASETS (Functional,Technical,operational)',1,3)
,(15,'DETAILS','General information of DATASET_FIELD (Functional,Technical,operational)',1,4)
,(16,'DETAILS','General information of DSA (Functional,Technical,operational)',1,5)
,(26,'DETAILS','General information of Documents',1,28)
,(27,'LICENSING TERMS','Licensing Terms of DSA',2,12));
```

Donde:

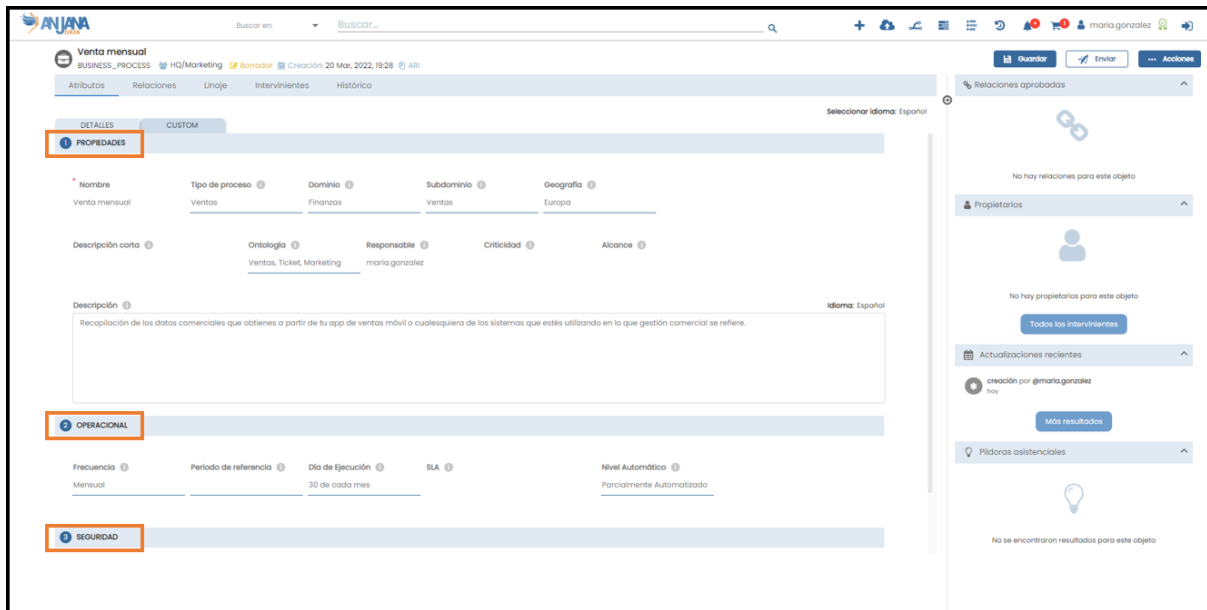
- **Id_menu:**(PK) Es el identificador único de la pestaña del menú
- **Name:** Es el nombre que se asigna a la pestaña del menú. En caso de querer visualizarlo en los distintos idiomas de la aplicación debe usarse el name como clave de traducción en la tabla de portuno de translations.
- **Description:** Descripción del contenido del menú
- **Order:** Indica el orden en que se muestran las pestañas de los menús
- **Object_subtype:** : (FK contra object_subtype) Indica el objeto que para el que se define el menú

NOTAS:

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

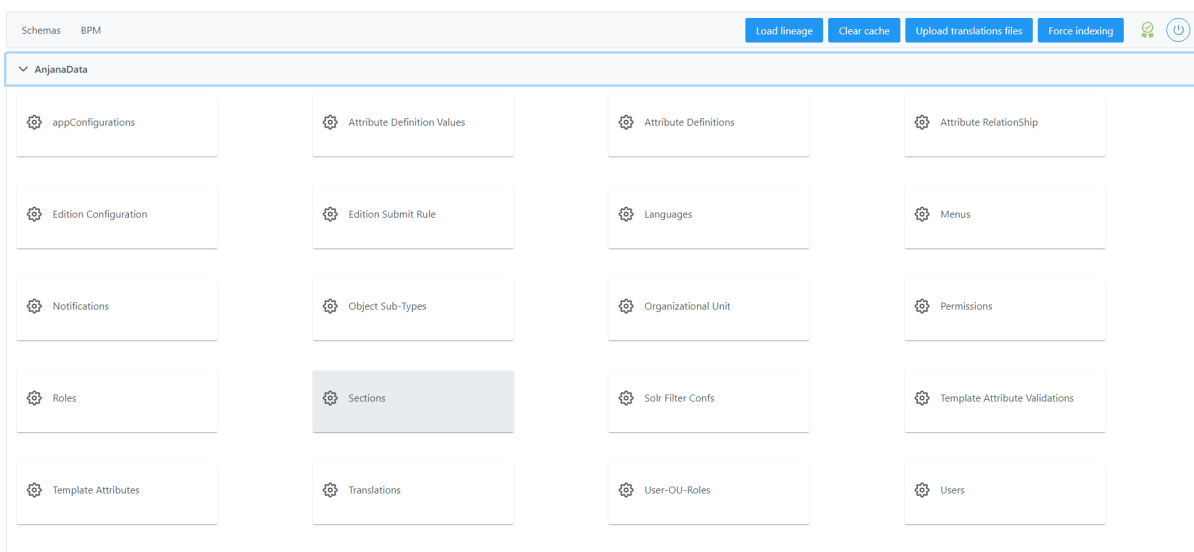
3. Configurar las secciones (sections)

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.



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. Cada sección registrada se caracteriza por los siguientes elementos:

- **Id:** 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:** Indica el orden en que se muestran las distintas secciones dentro de un menú.

- **Menu:** Indica el menú para el cual se define la sección.

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

+ New

Sections					
Id	Name	Description	order	menu	
-	-	-	-	Igual	
Criterio	Criterio	Criterio	Criterio	-	
<div style="display: flex; justify-content: space-between;"> Filtrar Filtrar Filtrar Filtrar Filtrar </div>					
1	TERMPROPERTIESSECTION	Properties of Business Terms	1	1-TERMMENU-TERM	
2	TERMOOPERATIONALSECTION	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-REPORTSMFNUJ-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:

Description
order
menu

Sections Edit
✕

Name

order

menu

Description

Properties of Documents

✕ Cancel
✓ Save

Visión de Desarrollador

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

```
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)
,(6,'Properties of Reports',2,'PROPERTIES', 1)
,(7,'Operational information of Reports',2,'OPERATIONAL', 2)
,(8,'Security information of Reports',2,'SECURITY', 3)
,(9,'Quality information of Reports',2,'QUALITY', 4)
,(10,'Social information of Reports',2,'SOCIAL', 5);
```

Donde:

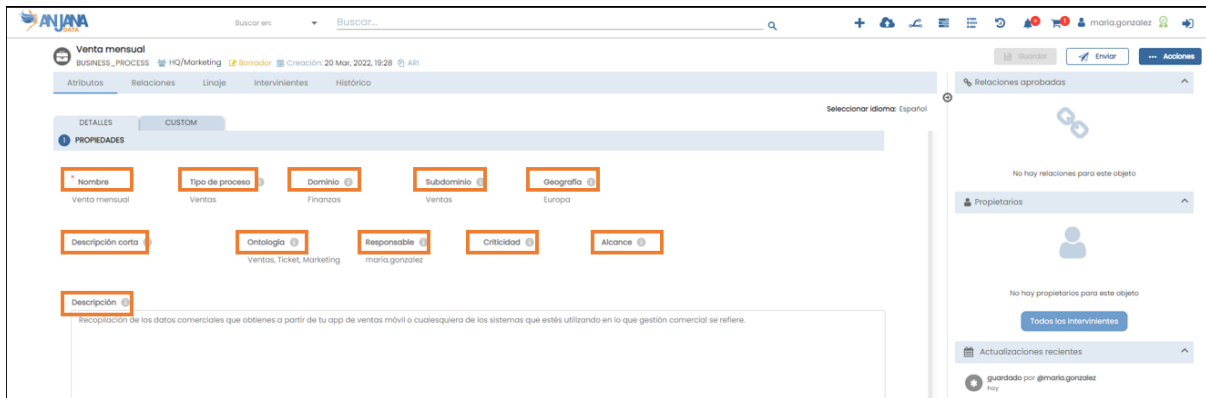
- **Id_section:** (PK) Es el identificador único de la sección
- **Description:** Descripción de la sección
- **Id_menu:** (FK contra *menu*) Identificador único de la pestaña en la que se aloja la sección
- **Name:** Nombre de la sección en Anjana. En caso de querer visualizarlo en los distintos idiomas de la aplicación debe usarse el name como clave de traducción en la tabla de portuno de translations.
- **Order_section:** Orden en el que aparece la sección

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:



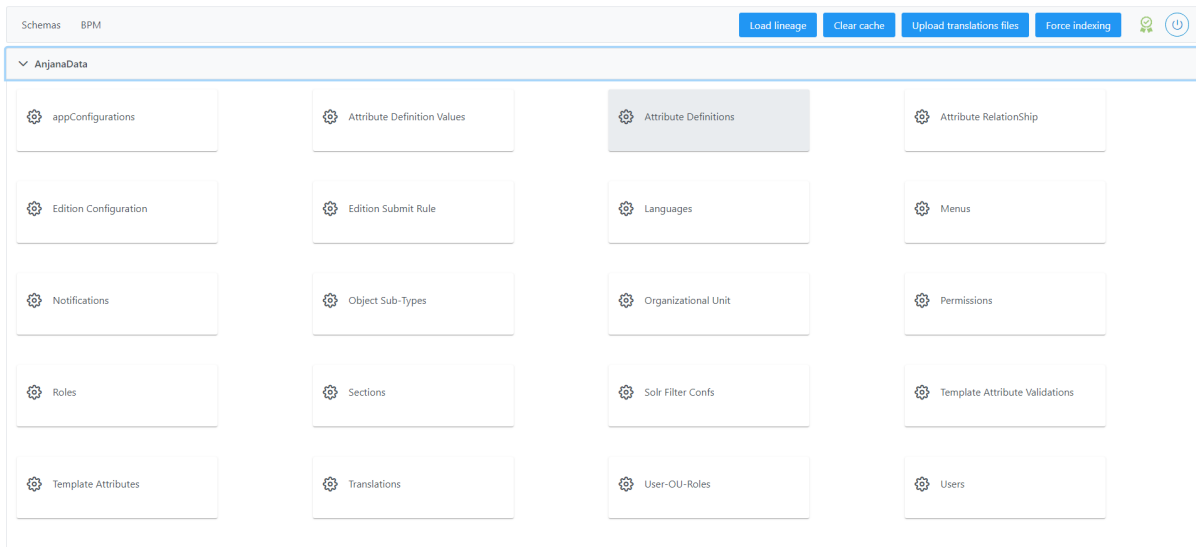
En Anjana Data los distintos tipos de atributos de metadatos soportados son los siguientes:

- **Boolean:** true o false
- **Date:** fecha (año, mes, día)
- **Number range:** selector de número entre un mínimo y un máximo definidos
- **Decimal:** número con decimales
- **File:** fichero que se almacena internamente en Anjana Data. También se permite descargar el fichero si se tiene permisos de lectura.
- **Number:** atributo para indicar un número entero
- **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
- **Text:** cuadro de texto normal
- **Text Area:** atributo para introducir un texto largo de hasta 300 mil caracteres
- **Enriched Text Area:** Atributo para introducir un texto largo de hasta 300 mil caracteres enriquecido con formato (negrita, subrayados, cursivas etc).
- **International Text:** Cuadro de texto normal 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.
- **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.
- **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 de usuarios definidos en Anjana Data
- **Entity Search:** atributo para elegir una entidad aprobada en Anjana Data
- **MultiSelect:** atributo para seleccionar uno o varios valores de una lista preconfigurada en la pestaña Reference Metadata
- **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 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

- **Array de entities:** atributo para elegir una o varias entidades aprobadas en Anjana

Visión de Administrador

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



Al acceder se muestra una tabla que contiene todos los atributos existentes en la configuración actual. Cada atributo registrado se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla.
- **Name:** Nombre interno del atributo
- **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:

Tipo de campo	Valor campo Type
Boolean	INPUT_CHECKBOX
Date	INPUT_DATE
Number range	INPUT_RANGE
Decimal	INPUT_DECIMAL
File	UPLOAD_FILE
Number	INPUT_NUMBER
Organizational Unit	SELECT_ORGANIZATIONAL_UNIT
Reference Metadata	SELECT
Text	INPUT_TEXT

Text Area	TEXT_AREA
Enriched Text Area	ENRICHED_TEXT_AREA
International Text	INPUT_TEXT_INTERNATIONAL
International Textarea	TEXT_AREA_INTERNATIONAL
International Text Editor	ENRICHED_TEXT_AREA_INTERNATIONAL
URL	UPLOAD_URL
Users	SELECT_USERS
Entity	ENTITY_SEARCH
MultiSelect	MULTI_SELECT
Array de Boolean	ARRAY_BOOLEAN
Array de Date	ARRAY_DATE
Array de Decimal	ARRAY_DECIMAL
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
Array de Entity	ARRAY_ENTITY

- **Description:** Descripción detallada del atributo
- **Label:** Nombre visible para el atributo en Anjana Data
- **Label Translation key:** Clave para la traducción del label en el idioma configurado para el usuario. Este texto debe coincidir con la clave de la traducción en portuno.translations
- **Place Holder:** Texto que el usuario verá en gris en el interior del cuadro que debe rellenar en modo edición
- **Place Holder Translation Key:** Clave para la traducción del Place Holder. Este texto debe coincidir con la clave de la traducción en portuno.translations
- **Short description translation key:** Clave para la traducción del short description. Este campo permite definir la descripción que saldrá en el formulario en el icono de ⓘ .
- **StartDate:** Fecha de creación del atributo (sólo informativo)
- **Last Modified Date:** Fecha de última modificación (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).

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	
<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	-	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	-	
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	
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	ATTRIBUTE_NOMB	Short	ATTRIBUTE_PLACE		-		BG	✎ 🗑

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 DocMgmtTool	Type SELECT	Label Document Management Tool
Label Translation Key	Place Holder Document Management Tool here	Place Holder Translation Key
Short Description Translation Key	startDate	Last Modified Date
Module BG		
Description Technology use for document management.		

Cancel Save


Visión de Desarrollador

Para definir los atributos hay que configurar la tabla **attribute_definition** del esquema **Anjana**. Para ello, hay que rellenar el siguiente sql:

```
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');
```

Donde:

- **Id_attribute_definition:** (PK) Es el identificador único del atributo de cada objeto
- **Description:** Es la descripción del atributo

- **Label:** Nombre del atributo que es visible desde Anjana Data
- **Label_translation_key:** Clave que permite buscar la traducción del nombre del atributo correspondiente al idioma que tiene configurado el usuario. Este texto debe coincidir con la clave de la traducción en portuno.translations.
- **Last Modified Date:** Fecha de última modificación (sólo informativo)
- **Module:** Indicar si pertenece al Data Catalog ('DC') o al Business Glossary ('BG'), (solo informativo, no limita las plantillas en las que se puede incluir).
- **Name:** Nombre del atributo, reconocido internamente en Anjana Data para la lógica de la herramienta
- **Place_holder:** Texto que el usuario verá en gris en el interior del cuadro que debe rellenar en modo edición
- **Place_holder_translation_key:** Clave que permite buscar la traducción del mensaje del atributo correspondiente al idioma que tiene configurado el usuario. Este texto debe coincidir con la clave de la traducción en portuno.translations.
- **Short description translation key:** Clave para la traducción del short description. Este campo permite definir una descripción que saldrá en el formulario en el icono de 
- **StartDate:** Fecha de creación del atributo (sólo informativo)
- **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 **attribute_type**:

Tipo de campo	Valor en columna attribute_type
Boolean	INPUT_CHECKBOX
Date	INPUT_DATE
Number range	INPUT_RANGE
Decimal	INPUT_DECIMAL
File	UPLOAD_FILE
Number	INPUT_NUMBER
Organizational Unit	SELECT_ORGANIZATIONAL_UNIT
Reference Metadata	SELECT
Text	INPUT_TEXT
Text Area	TEXT_AREA
Enriched Text Area	ENRICHED_TEXT_AREA
International Text	INPUT_TEXT_INTERNATIONAL
International Textarea	TEXT_AREA_INTERNATIONAL
International Text Editor	ENRICHED_TEXT_AREA_INTERNATIONAL
URL	UPLOAD_URL

Users	SELECT_USERS
Entity	ENTITY_SEARCH
MultiSelect	MULTI_SELECT
Array de Boolean	ARRAY_BOOLEAN
Array de Date	ARRAY_DATE
Array de Decimal	ARRAY_DECIMAL
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
Array de Entity	ARRAY_ENTITY

NOTAS:

- Para los registros con **attribute_type** de tipo SELECT, se deben definir los posibles valores del combo de selección en la tabla **attribute_definition_value**
- 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 que Solr, al indexar el objeto, falle si los tipos original y nuevo son incompatibles (por ej: si un atributo se declara alfanumérico y se indexa un objeto cuyo atributo tiene como valor “abc” y, posteriormente, se cambia el atributo para ser numérico, Solr tendrá indexado el atributo como alfanumérico y será imposible automatizar el cambio de campo).
- 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.
- 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 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)

Atributos de clave primaria

Las entidades de catálogo dinámico que proporciona Anjana de manera predefinida son las siguientes: *dataset*, *dataset field*, DSA, proceso, instancia y solución.

Para estas entidades es imposible especificar, a través de la base de datos, una clave primaria puesto que los datos de la entidad estarán especificados tanto en la tabla de entidades como en la tabla de atributos. El modelo relacional no permite definir una clave primaria que afecte a varias tablas.

Para solventar este problema Anjana incluye una gestión de atributos de clave primaria a través de la lógica del código para evitar relaciones duplicadas y poder identificar cada entidad inequívocamente.

A continuación se detalla una lista de atributos que forman parte de la clave primaria (*primary key*) para cada una de las entidades del catálogo dinámico. 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 *custom* 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 el campo **name** de la tabla **attribute_definition** tenga un código particular y así como el **attribute_type** sea de un tipo particular. De esta forma, Anjana Data podrá identificar estos atributos y funcionar correctamente:

```
insert into attribute_definition (id attribute_definition,acronym,alias,description,"label",label_translation_key,last_modified_date,"module".name,place_holder
(1,'name','name','Name of object','Name',NULL,NULL,'BG',name,'Name here',NULL,'Name of object',NULL,now(),'INPUT_TEXT')
```

Atributos comunes a todas las entidades:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
description	TEXT_AREA	Permite introducir una descripción extensa del objeto
finishDate	INPUT_DATE	Almacena la fecha en la que el objeto expira
name	INPUT_TEXT	Almacena el nombre lógico de la entidad

Atributos para datasets:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo
data_format	SELECT	Selección del formato del dataset
infrastructure	SELECT	Selección del entorno donde se encuentra localizado el dataset, usado con technology y zone para identificar un plugin en caso de gobierno activo.
isGoverned	INPUT_CHECKBOX	Indica si hay gobierno activo sobre el dataset
path	INPUT_TEXT	Indica la localización 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
technology	SELECT	Indica la tecnología en la que está depositado el dataset, usado con infrastructure y zone para identificar un plugin en caso de gobierno activo.
sampleData	INPUT_CHECKBOX	Indica si se puede ver una muestra de los datos del dataset en Anjana
zone	SELECT	Indica la zona donde se encuentra el dataset, usado con infrastructure y technology para identificar un plugin en caso de gobierno activo.

Atributos para dataset_fields:

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 indica el orden del field en un dataset

Para DSAs:

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

Para procesos:

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

Para instancias 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
-----------	------------	---

Para relaciones:

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
finishDate	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

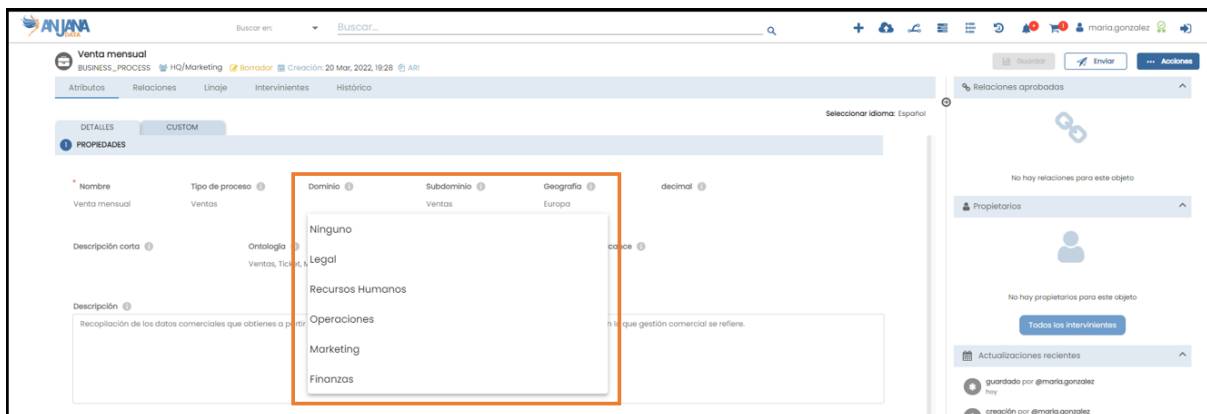
Para configurar el **versionado** o el lanzamiento de workflow de ciertos roles hay que definir los siguientes atributos que deberán aparecer ocultos en los templates de dataset, dataset field, instancia y DSA:

Valor en campo name	Valor en columna attribute_type	Finalidad del atributo	Entidad a la que aplica
datasetsInputInInstancia	INPUT_TEXT	Campo para versionado de una instancia cuando cambian los datasets input	Instancia
datasetsOutputInInstancia	INPUT_TEXT	Campo para versionado de una instancia cuando	Instancia

		cambian los datasets output	
datasetFields	INPUT_TEXT	Campo de versionado de datasets cuando versiona un dataset field	Dataset
datasetsInDsa	INPUT_TEXT	Campo de versionado de DSA cuando cambian los dataset asociados	DSA

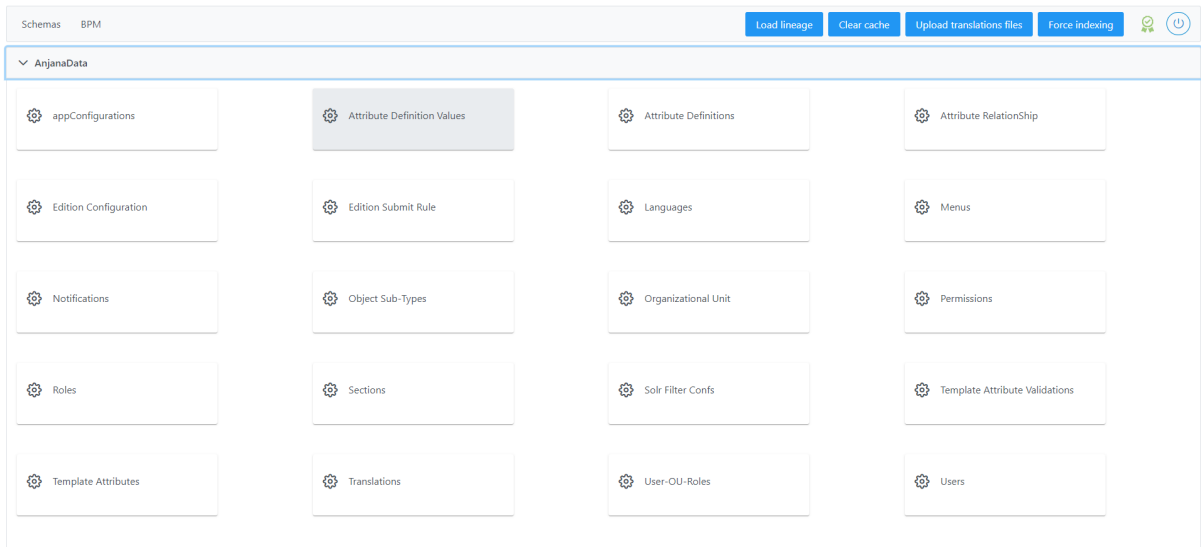
5. Definir y configurar los posibles valores de los atributos

Para aquellos atributos de tipo combo de selección (atributos con **attribute_type** SELECT o MULTI_SELECT) hay que definir los metadatos de referencia de entre los cuales, el usuario puede seleccionar el o los que convengan.



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. Cada atributo registrado se caracteriza por los siguientes elementos:

- **Id:** 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.
- **LabelSelect:** Descripción del valor
- **AttributeDefinition:** Atributo de metadatos para el cual se despliega el combo de selección.

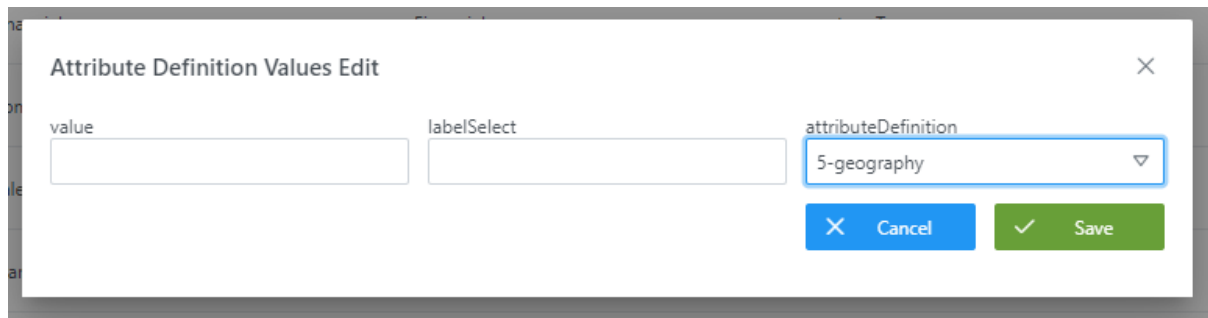
La creación de un nuevo metadato de referencia se realiza mediante el botón **New**:

+ New

Attribute Definition Values				
id <small>↑↓</small>	value <small>↑↓</small>	labelSelect <small>↑↓</small>	attributeDefinition <small>↑↓</small>	
-	-	-	Igual	
<input type="text" value="Criterio"/> Filtrar	<input type="text" value="Criterio"/> Filtrar	<input type="text" value="Criterio"/> Filtrar	<input type="text" value="-"/> Filtrar	
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 el siguiente sql:

```
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');
```

Donde:

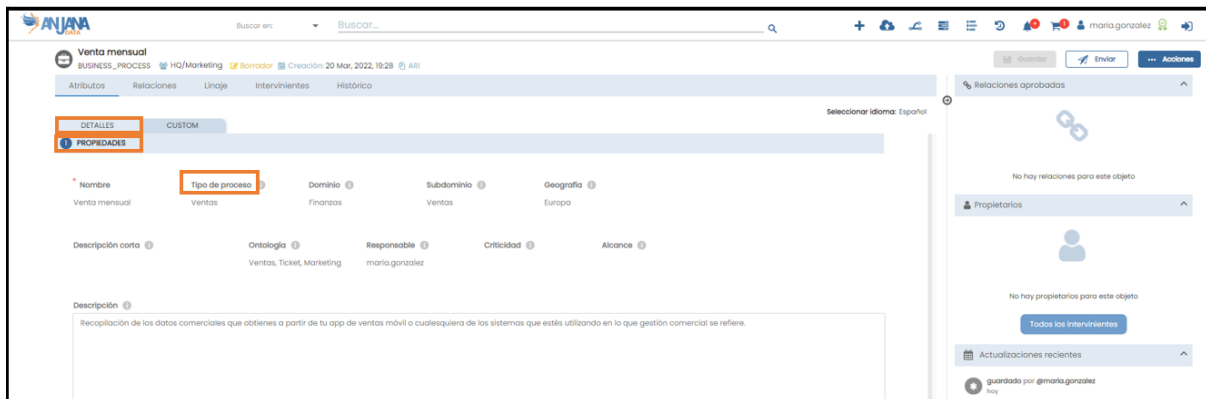
- **id_attribute_definition_value:** (PK) Es el identificador único de cada uno de los valores que pueden tomar los diferentes atributos
- **id_attribute_definition:** (FK contra *attribute_definition*) Identificador del atributo para el que se está definiendo un posible valor
- **labelSelect:** Descripción del valor
- **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.

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 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

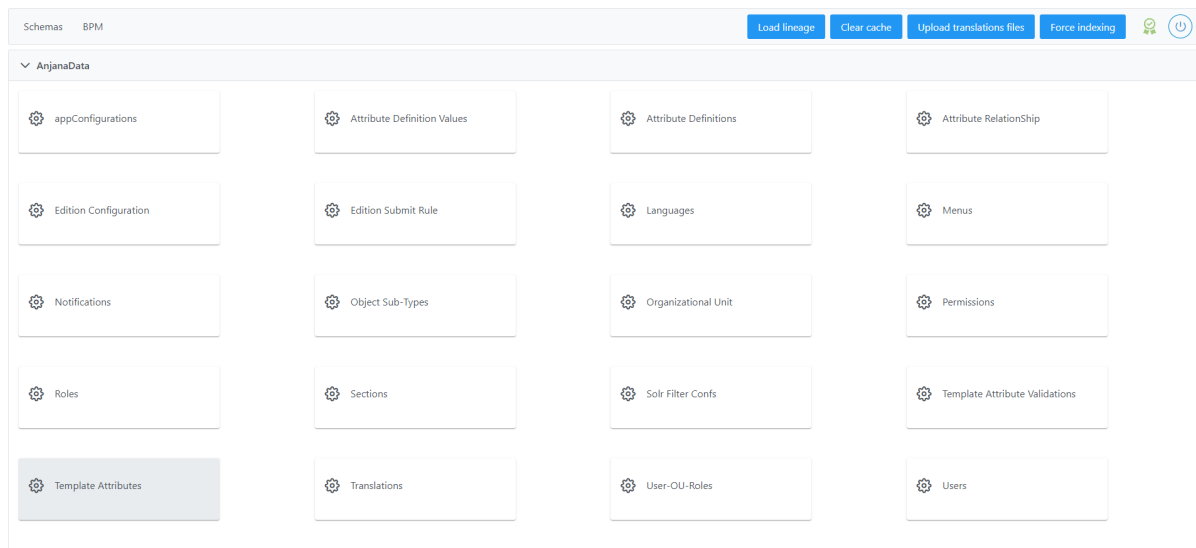
6. Definir qué atributos aparecerán en cada una de las secciones de los formularios según el tipo de objeto

Configurar las plantillas de metadatos supone definir en qué menú y sección aparecerán los metadatos de entre los definidos en la tabla `attribute_definition`.



Visión de Administrador

El alta de los atributos dentro de un template se realiza en el panel de administración de Anjana Data en **Template Attribute**:















Al acceder se muestra una tabla que contiene todos los atributos existentes en la configuración actual. Cada atributo registrado se caracteriza por los siguientes elementos:

- **Id:** 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.

- **IsIntern:** dejar a true (no se utiliza)
- **IsVisible:** flag que indica si un atributo es visible en el caso de los `dataset_fields`, en la tabla Structure de la pantalla de dataset
- **Sort:** Número de orden en el que aparece en la plantilla de metadatos
- **AttributeDefinition:** Atributo que se desea introducir en el template
- **Section:** Sección en la que se desea introducir el atributo

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

Template Attributes							+ 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

sort

attributeDefinition

section

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 el siguiente sql:

```
insert into anjana.template_attribute
```

```
(id_template_attribute, active, id_attribute_definition, is_intern, sort, id_section, is_visible)
VALUES
(1,true,1,true,1,1,true)
,(2,true,2,true,2,1,true)
,(3,true,3,true,3,1,true)
,(4,true,4,true,4,1,true)
,(5,true,5,true,5,1,true)
,(6,true,6,true,6,1,true)
,(7,true,7,true,7,1,true);
```

Donde:

- **Id_template_attribute:** (PK) Es el identificador único de la tabla. Identifica cada atributo dentro de una sección
- **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.
- **Id_attribute_definition:** (FK contra attribute_definition) Es el identificador del atributo que va a aparecer en el formulario, dentro de la sección que se indique en el parámetro id_section
- **Is_intern:** De momento, dejarlo a “true”
- **Sort:** Indica el orden que toma el atributo dentro de la sección que se indica en el siguiente parámetro
- **Id_section:** (FK contra sections) Identificador de la sección en la que se sitúa el atributo
- **Is_visible:** flag que indica si un atributo es visible en el caso de los dataset_fields, en la tabla Structure de la pantalla de dataset

NOTAS:

- Es importante que los atributos **datasetsInputInInstance**, **datasetsOutputInInstance**, **datasetFields** y **datasetsInDsa** se incluyan en los templates de INSTANCE, DATASET y DSA en modo oculto para que no interfieran en las actividades de los usuarios, por tanto, deben tener los flags de IsVisible y Active a false.
- Aunque ADHERENCE no tiene vista propia, es necesario añadir registros para esta relación para poder añadir los atributos necesarios que debe contener.

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

Existe la posibilidad de definir una serie de validaciones rápidas para asegurar la correcta entrada de los atributos de metadatos.

Indicadores Contratos
 KPI | HQ/Legal | Aprobado | Creación: 25 Mar, 2022, 12:16 | Aprobación: 25 Mar, 2022, 12:19 | ARI

Atributos | Relaciones | Linaje | Intervinientes | Histórico

Seleccionar idioma: Español

DETALLES | CUSTOM

1 PROPIEDADES

* Nombre	Tipo de KPI	Dominio	Subdominio	Geografía	decimal
Indicadores Contratos		Legal			2.

Solo se admiten números decimales

Descripción Idioma: Español

pr

Descripción corta	Idioma: Español	Ontología	Responsable	URL	Tecnología	SME
df				ggg		

Longitud mínima de caracteres: 3

El formato de URL no es correcto

Fórmula Unidad

Estas validaciones pueden ser:

- **Atributos obligatorios**

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

- **Atributos no modificables**

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

- **Longitud máxima y mínima para valores alfanuméricos**

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

- **Mínimo y Máximo para los valores en los campos enteros, decimales, arrays de los mismos y rangos de valores**

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

- **Dependencia entre valores de referencia**

Esta regla permite hacer dependiente un campo de otro y que el campo dependiente pueda arrojar posibles valores según las opciones elegidas en primera instancia del campo dependido.

- **Heredable**

Esta regla permite hacer identificar un atributo no editable y cuyo valor se calcula en función de los valores de ese mismo atributo en objetos relacionados.

Sólo los atributos booleanos pueden ser heredables (HERITABLE) y existen dos opciones de herencia de atributos entre las entidades nativas de Anjana:

- dataset_field -> dataset
- dataset -> DSA
- dataset -> instance
- **Longitud máxima y mínima de la parte decimal de campos y arrays de decimales**

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

- **Longitud máxima y mínima de valores en el conjunto**

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

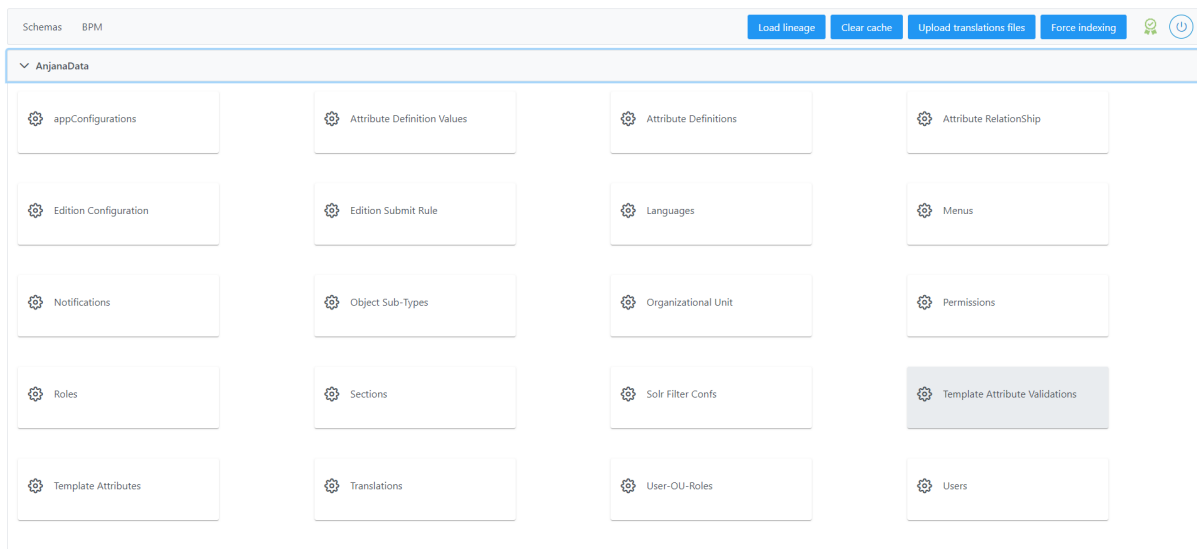
Validaciones para cada tipo de atributo:

TIPO ATRIBUTO	DE	VALIDACIONES									
		MAX LENGTH	MIN LENGTH	DEPENDS ON	MIN	MAX	MAX DECIMAL PRECISION	MIN DECIMAL PRECISION	HERITABLE	MAX LENGTH ARRAY	MIN LENGTH ARRAY
TEXTO		x	x								
CONJUNTO DE TEXTOS		x	x							x	x
TEXTO INTERNACIONAL		x	x								
CUADRO DE TEXTO		x	x								
CUADRO DE TEXTO INTERNACIONAL		x	x								
CUADRO DE TEXTO ENRIQUECIDO											
CUADRO DE TEXTO ENRIQUECIDO INTERNACIONAL											
RANGO ENTEROS					x	x					
ENTERO					x	x					
CONJUNTO DE ENTEROS					x	x				x	x
DECIMAL					x	x	x	x			
CONJUNTO DE DECIMALES					x	x	x	x		x	x
BOOLEANO									x		

CONJUNTO DE BOOLEANOS										x	x
FECHA											
CONJUNTO DE FECHAS										x	x
FICHERO											
CONJUNTO DE FICHEROS										x	x
URL											
CONJUNTO DE URLS										x	x
USUARIO											
CONJUNTO DE USUARIOS										x	x
ENTIDAD											
CONJUNTO DE ENTIDADES										x	x
SELECCIÓN ÚNICA DE METADATOS DE REFERENCIA			x								
SELECCIÓN MÚLTIPLE DE METADATOS DE REFERENCIA			x							x	x
UNIDAD ORGANIZATIVA											
CONJUNTO DE UNIDADES ORGANIZATIVAS										x	x

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. Cada validación registrada se caracteriza por los siguientes elementos:

- **TemplateAttribute:** Atributo del template para el que se define una validación rápida
- **Key:** Clave de la validación:
 - MAX_LENGTH_ARRAY: cantidad máxima de elementos de array
 - MIN_LENGTH_ARRAY: cantidad mínima de elementos de array
 - MAX_LENGTH: longitud máxima
 - MIN_LENGTH: longitud mínima
 - MAX: número máximo
 - MIN: numero mínimo
 - NOT_EDITABLE: no modificable
 - DEPENDS_ON: dependencia con otro atributo del template
 - REQUIRED: atributo obligatorio
 - HERITABLE: en el caso de que su valor se herede de otro metadato
 - MAX_DECIMAL_PRECISION: número máximo de precisión para valores decimales.
 - MIN_DECIMAL_PRECISION: número mínimo de precisión para valores decimales.
- **Name:** Nombre de la validación. Los nombres de las validaciones están establecidos y se deberán escoger en función de la validación que se desee realizar. Siguiendo la especificación de las claves especificadas en el punto anterior (*key*), los nombre serán los siguientes:
 - *maxLengthArray*
 - *minLengthArray*
 - *maxLength*
 - *minLength*
 - *max*
 - *min*
 - *not_editable*
 - *dependsOn*
 - *required*
 - *heritable*
 - *maxDecimalPrecision*

- *minDecimalPrecision*
- **OnlyOnEdition:** 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 pero, sin embargo, para importar objetos con excel es necesario que los atributos que vayan a ser editados, en caso de tener validaciones, tengan este flag a false.
- **applyOnStateDraftOrRejected:** Flag para indicar que la validación aplica a objetos en estado de borrador o rechazado. Se recomienda que este flag siempre esté a true para asegurar que ninguna validación queda sin aplicar
- **Params:** Parámetros:
 - En el caso de validaciones de tipo DEPENDS_ON, indicar el nombre del atributo del que depende (name de la tabla anjana.attribute_definition).
 - En el caso de valores máximos o mínimos, indicar el valor numérico.
 - En caso de validaciones de tipo booleano como REQUIRED ó NOT_EDITABLE indicar el valor en formato texto especificando true o false.
 - En caso de HERITABLE el parámetro será HIGH (si de los elementos que se heredan hay uno a true, el valor del campo heredado es true) o LOW (si de los elementos que se heredan hay uno a false, el valor del campo heredado es false)

El mensaje que aparecerá en el formulario dinámico es posible incluirlo en diferentes idiomas, para ello hay que incluir en la tabla 'translations' la key de nuestra tabla template_attribute_validation, así como el texto y el idioma introducido.

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

+ New

Template Attribute Validations						
templateAttribute ↑↓	key ↑↓	Name ↑↓	onlyOnEdition ↑↓	applyOnStateDraftOrRejected ↑↓	params ↑↓	
Igual	-	-	<input type="checkbox"/>	<input type="checkbox"/>	-	
-	Criterio	Criterio	<input type="checkbox"/>	<input type="checkbox"/>	Criterio	
Filtrar	Filtrar	Filtrar			Filtrar	
1-name-TERM	DISABLED	disabled	✓	✓	true	✎ ✖
1-name-TERM	REQUIRED	required	✓	✓	true	✎ ✖
6-description-TERM	MAX_LENGTH	maxLength	✗	✓	300000	✎ ✖
6-description-TERM	MIN_LENGTH	minLength	✗	✓	3	✎ ✖
7-sdescription-TERM	MAX_LENGTH	maxLength	✗	✓	100	✎ ✖
7-sdescription-TERM	MIN_LENGTH	minLength	✗	✓	3	✎ ✖
28-name-REPORT	DISABLED	disabled	✓	✓	true	✎ ✖

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

Template Attribute Validations Edit ✕

templateAttribute <input type="text" value="Select Item"/>	key <input type="text" value="Select Item"/>	Name <input type="text"/>
params <input type="text"/>	<input type="checkbox"/> onlyOnEdition	<input type="checkbox"/> applyOnStateDraftOrRejected

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 el siguiente sql:

```

insert into anjana.template_attribute_validation
(id_template_attribute,name, key, params, only_on_edition,apply_on_state_draft_or_rejected)
VALUES
(6, 'maxLength','MAX_LENGTH','300000', false, true)
,(6, 'minLength','MIN_LENGTH','3', false, true)
,(7, 'required','REQUIRED','true', false, true)
,(7, 'minLength','MIN_LENGTH','3', false, true)
,(34, 'disabled','NOT_EDITABLE','true', false, true)
,(60, 'max','MAX','100', false, true)
,(60, 'min','MIN','3', false, true)
,(83, 'dependsOn','DEPENDS_ON','sourceType', false, true)
,(105, 'heritable','HERITABLE','HIGH', false, true)
,(133, 'minDecimalPrecision','MAX_DECIMAL','4', false, true)
,(133, 'maxDecimalPrecision','MIN_DECIMAL','2', false, true)
,(159, 'minLengthArray','MIN_LENGTH_ARRAY','3', false, true)
,(159, 'maxLengthArray','MAX_LENGTH_ARRAY','10', false, true)
,(181, 'maxLength','MAX_LENGTH','100', false, true);
  
```

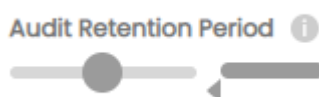
Donde:

- **Id_template_attribute:** (FK contra *template_attribute*) Identifica un atributo dentro de una sección que va a sufrir la validación que se indique
- **Name:** Nombre de la regla de validación que debe pasar un atributo de una sección
 - *maxLengthArray*
 - *minLengthArray*
 - *maxLength*
 - *minLength*
 - *max*
 - *min*
 - *notEditable*
 - *dependsOn*
 - *required*
 - *heritable*

- o *maxDecimalPrecision*
- o *minDecimalPrecision*
- **Key:** Clave de la validación:
 - o MAX_LENGTH_ARRAY: cantidad máxima de elementos de array
 - o MIN_LENGTH_ARRAY: cantidad mínima de elementos de array
 - o MAX_LENGTH: longitud máxima
 - o MIN_LENGTH: longitud mínima
 - o MAX: número máximo
 - o MIN: numero mínimo
 - o NOT_EDITABLE: no modificable
 - o DEPENDS_ON: dependencia con otro atributo del template
 - o REQUIRED: atributo obligatorio
 - o HERITABLE: en el caso de que su valor se herede de otro metadato
 - o MAX_DECIMAL_PRECISION: número máximo de precisión para valores decimales.
 - o MIN_DECIMAL_PRECISION: número mínimo de precisión para valores decimales.
- **Params:** Parámetros:
 - o En el caso de validaciones de tipo DEPENDS_ON, indicar el nombre del atributo del que depende (name de la tabla anjana.attribute_definition).
 - o En el caso de valores máximos o mínimos, indicar el valor numérico.
 - o En caso de validaciones de tipo booleano como REQUIRED ó NOT_EDITABLE indicar el valor en formato texto especificando true o false.
- **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 pero, sin embargo, para importar objetos con excel es necesario que los atributos que vayan a ser editados, en caso de tener validaciones, tengan este flag a false.
- **Apply_on_state_draft_or_rejected:** flag para indicar que la validación aplica a objetos en estado de borrador o rechazado. Se recomienda que este flag siempre esté a true para asegurar que ninguna validación queda sin aplicar

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 el valor mínimo será 1 y el máximo 365.
- Es recomendable que los atributos de tipo INPUT_RANGE incluídos en plantillas de objetos que van a ser importados por excel tengan la validación de obligatorios para evitar que, al importar el fichero, el usuario deje el atributo sin completar y, al no coincidir el valor vacío (null) con ningún valor permitido del rango, se muestre de forma errónea en el frontal de la aplicación:



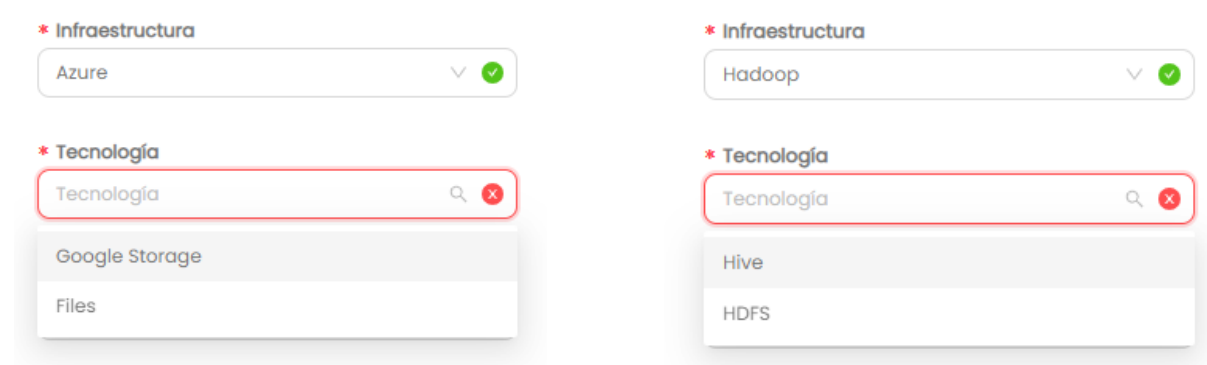
- Los atributos source y destination de las plantillas de las relaciones deben tener la validación de obligatoriedad para que siempre estén completos y no queden relaciones sin extremo definido.

- Los atributos infrastructure, technology y zone de las plantillas de dataset y proceso deben tener la validación de obligatoriedad para que siempre estén completos y no queden relaciones sin extremo definido.
- Los atributos que componen la PK de los objetos no serán editables a pesar de que no se le definan la validaciones correspondientes

8. Definir relaciones entre valores de atributos para pre-filtrar los reference metadata

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).

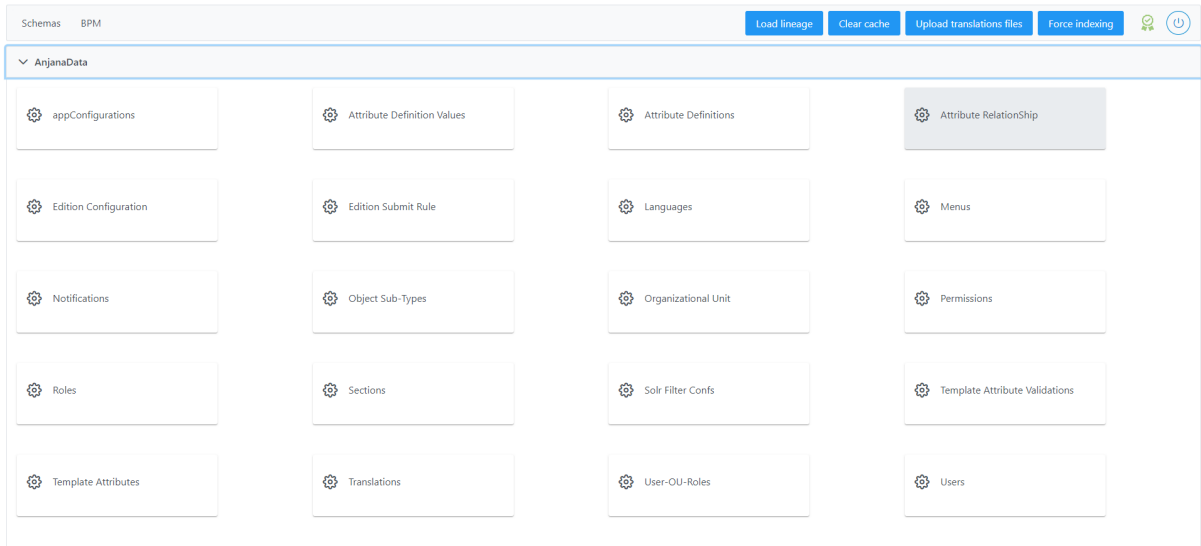


The image shows two side-by-side screenshots of a configuration interface. Each screenshot has two dropdown menus. The top dropdown in both is labeled '* Infraestructura' and has a green checkmark icon. The left one is set to 'Azure' and the right one to 'Hadoop'. The bottom dropdown in both is labeled '* Tecnología' and has a red border and a red 'X' icon. The left one is set to 'Tecnología' and shows a search bar with 'Tecnología' and a magnifying glass icon. Below it, a list of options is shown: 'Google Storage' and 'Files'. The right one is also set to 'Tecnología' and shows a list of options: 'Hive' and 'HDFS'.

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.

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. Cada relación se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla.
- **DestinationValue:** Valor del atributo que depende del otro
- **SourceValue:** Valor del atributo original del que depende el destinationValue
- **ObjectSubType:** Plantilla a la que se asocia el atributo (null si es a cualquier plantilla donde esté el atributo)

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

+ New

Attribute Relationship			
Id	destinationValue	sourceValue	objectSubType
-	Igual	Igual	-
<input type="text" value="Criterio"/> Filtrar	<input type="text" value="-"/> Filtrar	<input type="text" value="-"/> Filtrar	<input type="text" value="Criterio"/> Filtrar
1	260-Datalake V2-technology	190-CPD Environment 1-infraestructure	✎ ✖
2	265-Mariadb-technology	190-CPD Environment 1-infraestructure	✎ ✖
3	266-Mysql-technology	190-CPD Environment 1-infraestructure	✎ ✖
5	268-Postgresql-technology	191-CPD Environment 2-infraestructure	✎ ✖
6	269-Kafka-technology	191-CPD Environment 2-infraestructure	✎ ✖
7	271-S3-technology	192-AWS-infraestructure	✎ ✖
8	270-SQLServer-technology	193-Azure Analytics-infraestructure	✎ ✖
9	259-Blobstoragee-technoloav	193-Azure Analytics-infraestructure	✎ ✖

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

Attribute Relationship Edit ✕

destinationValue sourceValue objectSubType

Visión de Desarrollador

Para definir las relaciones entre atributos hay que configurar la tabla **attribute_relationships** del esquema **Anjana**. Para ello, hay que rellenar el siguiente sql:

```
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);
```

Donde:

- **id_attribute_relationships:** (PK) Identificador único de la tabla
- **source_value:** (FK contra *attribute_definition_value*) Identifica el reference metadata origen cuya selección condiciona que aparezca el reference metadata del destino
- **destination_value:** (FK contra *attribute_definition_value*) Identifica uno de los reference metadata que aparecerán en el combo de selección como consecuencia de la selección previa del reference metadata del source
- **objectSubType:** Plantilla a la que se asocia el atributo (null si es a cualquier plantilla donde esté el atributo)

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

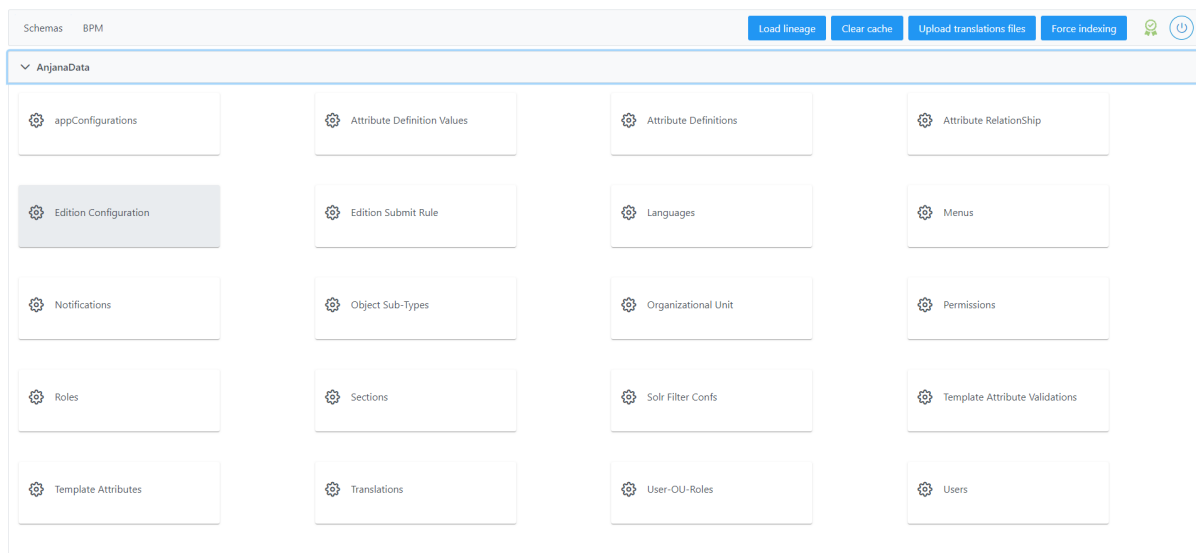
Es posible configurar qué cambios generan un versionado en los objetos de Anjana Data mediante las reglas de versionado. Estas reglas permiten identificar los atributos de los objetos que, al ser editados, son suficientemente importantes como para versionar el objeto y mantener la versión original deprecada.

Estas reglas de versionado sólo se pueden configurar para las entidades de Data Catalog: DATASET, DATASET_FIELD, DSA, PROCESS, INSTANCE y SOLUTION.

Es importante que los atributos **datasetsInputInInstance**, **datasetsOutputInInstance**, **datasetFields** y **datasetsInDsa** se hayan creado en los templates de INSTANCE, DATASET y DSA.

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 relaciones entre atributos y plantillas existentes en la configuración actual. Cada relación se caracteriza por los siguientes elementos:

- **Id**: Identificador único de la tabla.
- **CurrentValue**: 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**
- **NewValue**: Valor nuevo del atributo. Si se desea que haya versionado independientemente del valor nuevo, introducir **null**
- **TemplateAttribute**: Atributo de una plantilla concreta al que aplica la regla

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

Edition Configuration				
id	currentValue	newValue	templateAttribute	
-	-	-	Igual	
Criterio	Criterio	Criterio	-	
Filtrar	Filtrar	Filtrar	Filtrar	
1	false	true	258-pi-DATASET	 
2	false	true	455-pi-DATASET_FIELD	 
4	false	true	275-golden-DATASET	 
5			21-data_sensitivity-TERM	 
6			46-data_sensitivity-REPORT	 
7			96-data_sensitivity-BUSINESS_PROCESS	 
8			121-data_sensitivity-KPI	 
9			147-data_sensitivity-DIMENSION	 

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

Edition Configuration Edit ✕

currentValue

newValue

templateAttribute

Visión de Desarrollador

Las reglas de versionado se configuran en la tabla **edition_configuration** del esquema **Anjana**. Rellenar el siguiente sql:

```
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);
```

Donde:

- **id_edition_conf**: es el identificador único de la tabla
- **id_template_attribute**: (FK contra template_attribute) Es el identificador del template que al modificarse genera una nueva versión del objeto del template al que pertenece
- **current_value**: Valor inicial del atributo que al cambiarse al valor indicado en *new_value*

genera versionado. Si se desea que haya versionado independientemente del valor inicial, introducir **null**

- **new_value:** Valor que adquiere el atributo y que como consecuencia se genera una nueva versión. Si se desea que se genere una nueva versión independientemente del nuevo valor, introducir **null** en este campo

NOTAS:

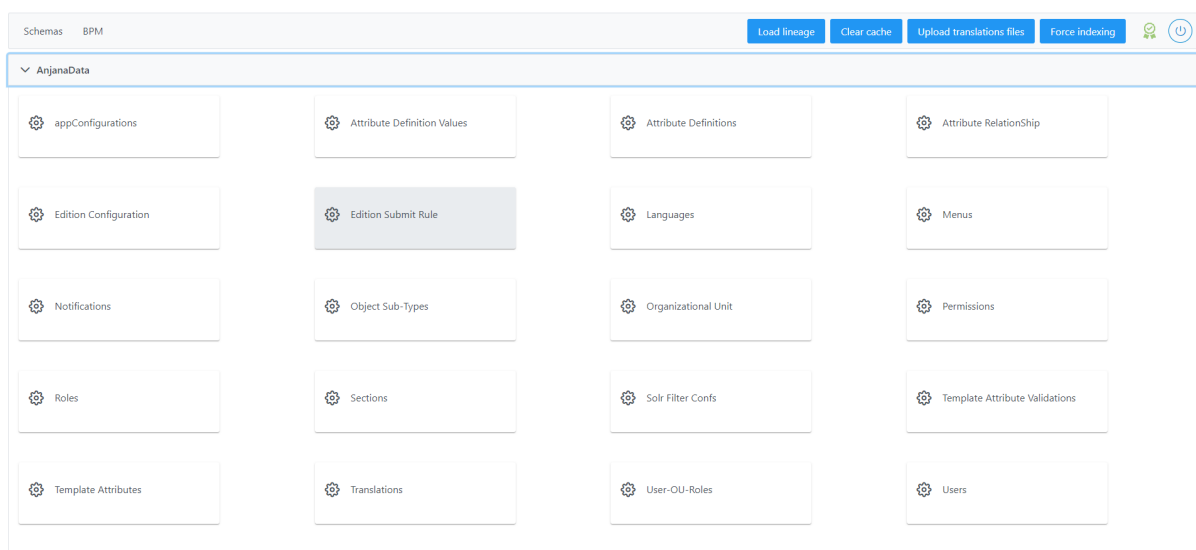
- Si se desea versionar un dataset en caso de que se modifiquen sus dataset_fields, es necesario añadir el atributo **datasetFields** a la tabla anjana.attribute y la relación con dataset en anjana.template_attribute. De esta forma, se comprobarán los cambios ocurridos en los dataset_fields y se versionará el dataset si alguno de esos cambios coincide con alguna regla de versionado para los atributos de dataset_fields.
- Si se desea versionar un DSA en caso de que varíe el conjunto de datasets asociado, es necesario añadir el atributo **datasetsInDsa** a la tabla anjana.attribute y la relación con dataset en anjana.template_attribute.
- Si se desea versionar una instancia en caso de que varíe el conjunto de datasets asociado, es necesario añadir los atributos **datasetsInputInInstance** y **datasetsOutputInInstance** a la tabla anjana.attribute y la relación con dataset en anjana.template_attribute.

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.

Visión de Administrador

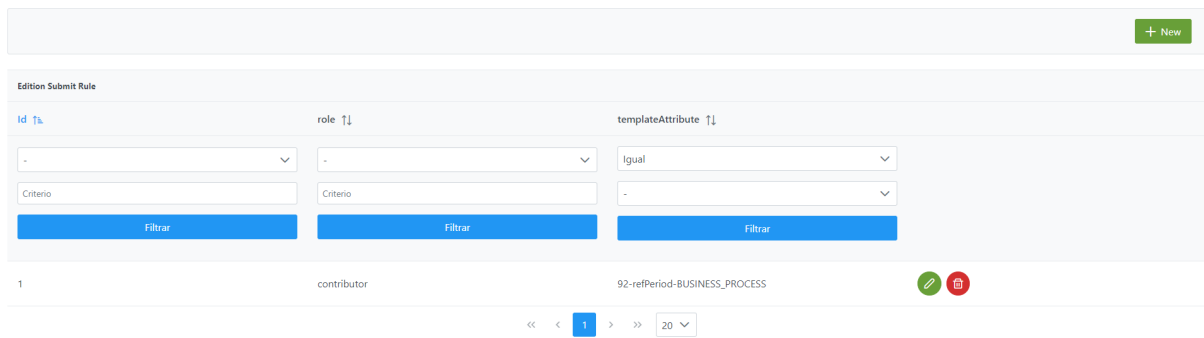
El alta de las reglas de lanzamiento de workflow en ediciones y versionado se realiza en el panel de administración de Anjana Data en **Edition Submit Rule**:



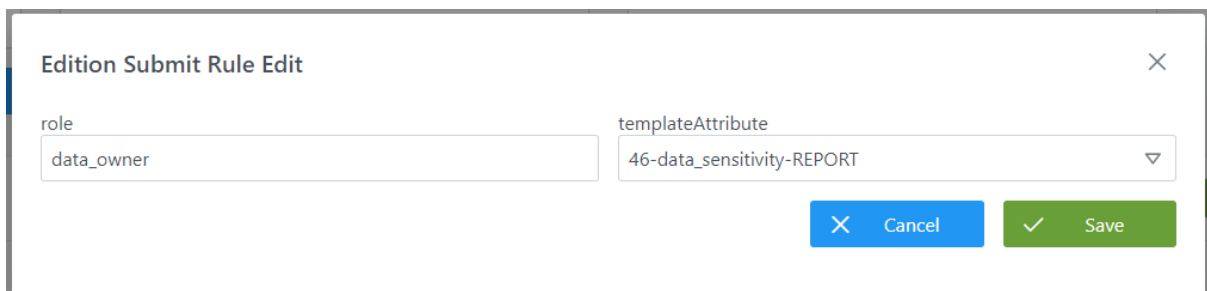
Al acceder se muestra una tabla que contiene todas las reglas que se usan para decidir si se lanza un workflow o no cuando se edita o versiona un objeto. Cada regla se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla.
- **TemplateAttribute:** 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.

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:



Visión de Desarrollador

Las reglas de lanzamiento de workflow en ediciones y versionado se configuran en la tabla **edition_submit_rule** del esquema **Anjana**. Rellenar el siguiente sql:

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

Donde:

- **id_edition_submit_rule:** es el identificador único de la tabla
- **role:** es el nombre del rol que aplica
- **id_template_attribute:** (FK contra template_attribute) Es el identificador del template que aplica en la regla

Esquema Hermes de BD

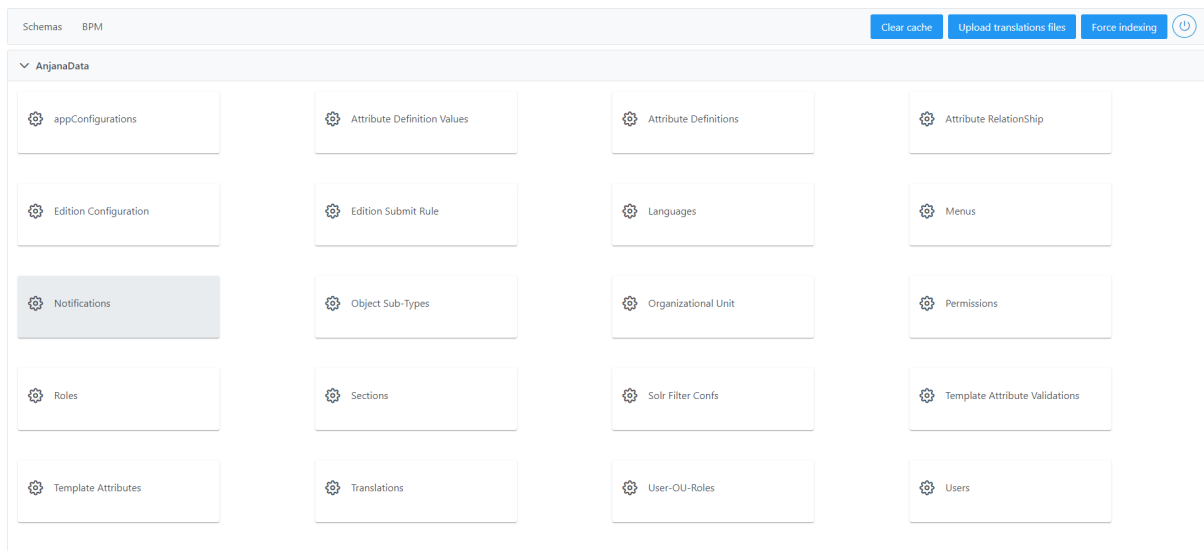
notification	
123	id_notification
ABC	module_type
ABC	notification_code
ABC	notification_receiver_type
ABC	translation_key
ABC	notification_type
ABC	receiver_role
ABC	severity
ABC	subject

1. Notificaciones del sistema

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

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 todas las notificaciones de la configuración actual. Cada notificación registrada se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla.
- **ModuleType:** valor string que solo puede adoptar un grupo de valores cerrado de entre los disponibles en la librería core ('BG', 'DC')

- **NotificationCode:** código para poder enviar notificaciones desde las funcionalidades propias de Anjana
- **NotificationReceiverType:** indica si la notificación será recibida por un único usuario nominal ('USER') o por todos los usuarios que tengan asignado un rol ('ROLE')
- **TranslationKey:** 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
- **NotificationType:** indica si la notificación es alerta ('ALERT') o aviso ('NOTICE')
- **ReceiverRole:** 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

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
Filtrar Filtrar Filtrar Filtrar Filtrar Filtrar Filtrar Filtrar								
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

✕ Cancel
✓ Save

Visión de Desarrollador

Para añadir las notificaciones hay que configurar la tabla **notifications** del esquema **Hermes**. Hay que rellenar este sql:

```

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');

```

La tabla consta de los siguientes campos:

- **id_notification:** (PK) Identificador único, sirve para gestión interna
- **module_type:** valor string que solo puede adoptar un grupo de valores cerrado de entre los disponibles en la librería core ('BG', 'DC')
- **notification_code:** código 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 de la notificación definida en la tabla "translations". El mensaje puede contener 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')
- **severity:** criticidad de la notificación
- **subject:** Clave de traducción con el asunto de la notificación.

En dicha tabla de notificaciones se pueden añadir nuevas pero para el correcto funcionamiento de Anjana deben de existir las siguientes:

- NEW_DSA → Cuando el usuario solicita la creación de un nuevo DSA.
- EXPIRATION → El objeto está expirando.
- ERROR_AUTOMATIC_METADATA → Se produce un fallo en la importación de metadatos.
- DATASET_FAIL → Fallo en la creación de dataset en sistemas de terceros.
- DSA_FAIL → La creación de un DSA ha fallado indicando el motivo.
- ADHERENCE_FAIL → Indica el motivo por el que ha fallado la adherencia.
- DISADHERENCE_FAIL → Fallo en la desadherencia indicando el motivo.
- DEPRECATION_ADHERED → Avisa al usuario que el objeto al que estaba adherido (dataset o DSA) va a ser deprecado.
- DEPRECATION → Avisa a todos los usuarios de un rol que el objeto va a ser deprecado.
- WORKFLOW_OK → El workflow ha sido creado.
- WORKFLOW_FAIL → Indica el motivo por el que ha fallado la creación de un workflow.
- EXCEL_IMPORT → Resumen de la importación por excel.
- USER_CROSS_ROLE_FAIL → Indica los usuarios que no han sido configurados como cross.
- INDEX_FAIL → Se ha producido un error en la indexación.
- FORM_FAIL → Errores encontrados en el formulario del objeto.
- LICENSE_EXPIRING → Aviso de que la licencia, aunque aún es válida, está cerca de expirar.
- LICENSE_EXPIRED → La licencia ha expirado.
- CHECK_CONFIGURATION_IN_PROVIDER → Error al recuperar la información de un usuario en el provider indicado.
- 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.
- DSA_TRANSFER_FAIL → Se le indica al administrador que no se ha podido hacer el cambio de unidad organizativa del dsa.
- EXPIRATION_WARNING → Aviso que se envía a los propietarios del objeto debido a que está próxima su expiración.
- INDEX_RELATIONSHIP_FAIL → Error de indexación en relaciones.
- WORKFLOW_DELETE_FAIL → No se ha podido borrar un workflow.
- EXPIRATION_WARNING_ADHERED → Aviso que se envía a los usuarios adheridos al objeto que indica que está próxima su expiración.
- EXPIRATION_ADHERED → El objeto al que se está adherido ha expirado.
- DISADHERENCE → Se indica que se ha realizado correctamente la desadherencia.

Y también se pueden añadir distintos mensajes que aparecerán en las notificaciones, estos mensajes son multi idioma y se pueden añadir variables entre #. Un ejemplo de un mensaje con variables sería:

“Ocurrió un error en el borrado del workflow con id #OBJECT_ID#”

Las distintas variables que permite Anjana son:

- OBJECT_ID → Identificador del objeto
- OBJECT_NAME → Nombre del objeto
- TYPE_OBJECT → Tipo del objeto
- OBJECT_SUB_TYPE → Subtipo del objeto
- VERSION → Número de versión del objeto
- USER_NAME → Usuario que ha realizado la acción
- RESULT → Resultado de la acción
- WORKFLOW_TYPE → Tipo del workflow lanzado
- REASON → Razón de aprobación/rechazo del workflow
- ORGANIZATIONAL_UNIT → Unidad organizativa del objeto
- ORGANIZATIONAL_UNIT_CHANGED → Unidad de destino del objeto en una transferencia de OU
- ACTION → Acción realizada
- REQUEST_REASON → Indica la razón de la solicitud de la acción, se rellena en las notificaciones de solicitud de creación de dsa por ejemplo
- OBJECT_LIST → Lista de objetos, se usa en notificaciones como las de la importación por excel para indicar el resultado de un listado de objetos
- ROLE → El rol involucrado
- NAME_PROVIDER → Nombre del proveedor de identidades donde ocurre la acción

Esquema Minerva de BD

solr_filter_conf	
123	id
ABC	collection
<input checked="" type="checkbox"/>	groupable
ABC	label
<input checked="" type="checkbox"/>	main
ABC	name
123	position
<input checked="" type="checkbox"/>	sortable
ABC	type

1. Filtros del sistema

Los filtros disponibles en las pantallas de Anjana en las que se muestran objetos de las colecciones de Solr son configurables. Estos objetos son: entidades, relaciones, histórico o auditoría.

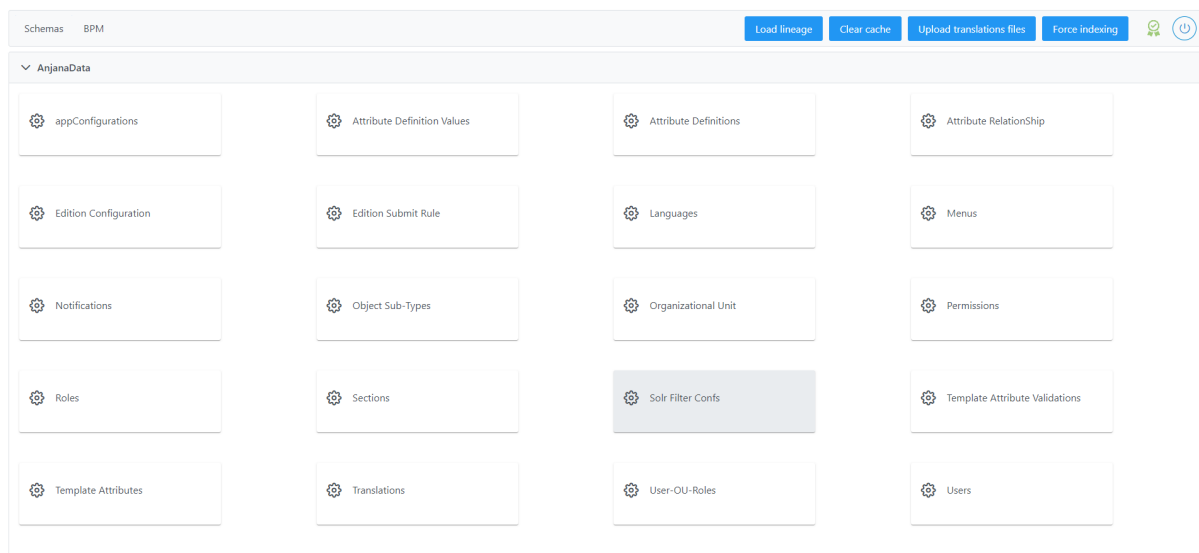
La tabla **solr_filter_conf** es la tabla que contiene los filtros preconfigurados disponibles en la plataforma.

A la hora de especificar los filtros que se quieren visualizar en el portal habrá que poner especial atención para evitar que salgan duplicados en el portal. Esto ocurrirá si, por ejemplo, para el histórico completo (búsquedas sobre la colección de *audit_logs*), en el que ya se está filtrando desde una fecha a otra (en un rango inicial de una semana), se especifica en la configuración de nuevo esas fechas para filtrar.

Asimismo esta misma casuística podrá ocurrir, por ejemplo, en la página principal (donde se muestran todos los objetos, entidades y relaciones, de nuestro dominio) y se intenta añadir en la configuración de Solr (tabla *solr_filter_conf*) filtros como los que especifican propiedad, adherencia o gobierno.

Visión de Administrador

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



















Al acceder se muestra una tabla que contiene todos los filtros existentes en la configuración actual. Cada filtro registrado se caracteriza por los siguientes elementos:

- **Id:** Identificador único de la tabla.
- **Name:** campo (*field*) de Apache Solr por el que va a efectuarse el filtro. Para los atributos de metadatos que se añadan como filtros adicionales para la colección de kernel, en este campo se debe poner el mismo **name** que en la tabla *anjana.attribute_definition* pero todo en mayúsculas (EX: *technology* como filtro tendría que ser *TECHNOLOGY*, *isEngine* como *IENGINE*, *data_sensitivity* como *DATA_SENSITIVITY*, etc)
- **Label:** clave (*key*) que permite que el usuario vea el nombre del filtro en el idioma que tiene configurado. 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*.
- **Type:** Indica el tipo de filtro. Los valores posibles son:
 - **INPUT_CHECKBOX** para multi-selección

- SELECT para selección única
- INPUT_DATE para introducir fechas en calendario
- INPUT_TEXT para la especificación de un texto
- **Collection:** Identifica la colección de Apache Solr a la que pertenece el filtro especificado: “kerno” si es para las entidades y relaciones y “audit_logs” si es para auditoría e histórico
- **Groupable:** boolean que por defecto adquiere el valor “false”. No se hace uso de este campo.
- **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.
- **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.

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

+ New									
Solr Filter Confs									
Id ↑	Name ↑	Label ↑	Type ↑	collection ↑	groupable ↑	main ↑	position ↑	sortable ↑	
-	-	-	-	-			-		
criterio	criterio	criterio	criterio	criterio	<input type="checkbox"/>	<input type="checkbox"/>	criterio	<input type="checkbox"/>	
	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar		Filtrar		
6	object_type	FILTERS.OBJ_TYPE	INPUT_CHECKBOX	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	 
7	user_name	FILTERS.USER	SELECT	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	 
8	start_time	COMMON.START_DATE	INPUT_DATE	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	 
9	end_time	COMMON.EXPIRATION_	INPUT_DATE	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	 
10	action	FILTERS.ACTION	INPUT_CHECKBOX	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	 
11	object_id	FILTERS.OBJECT_ID	INPUT_TEXT	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	 
12	object_name	FILTERS.OBJECT_NAME	INPUT_TEXT	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	<input checked="" type="checkbox"/>	 
22	object_sub_type	FILTERS.OBJ_SUB_TYPE	INPUT_CHECKBOX	audit_logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	<input checked="" type="checkbox"/>	 

A continuación, se muestra cómo crear un filtro:

Solr Filter Confs Edit ×

Name <input type="text" value="user_name"/>	Label <input type="text" value="FILTERS.USER"/>	Type <input type="text" value="SELECT"/>
collection <input type="text" value="audit_logs"/>	position <input type="text" value="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 **solr_filter_conf** del esquema **Minerva**. Para ello, hay que rellenar el siguiente sql:

```
INSERT INTO minerva.solr_filter_conf
(id,collection,groupable,"label",main,"name","position",sortable,"type") VALUES
(6,'audit_logs',true,'FILTERS.OBJ_TYPE',true,'object_type',0,true,'INPUT_CHECKBOX'),
(7,'audit_logs',true,'FILTERS.USER',true,'user_name',1,true,'SELECT'),
(11,'audit_logs',true,'FILTERS.OBJECT_ID',false,'object_id',5,true,'INPUT_TEXT'),
(12,'audit_logs',true,'FILTERS.OBJECT_NAME',false,'object_name',6,true,'INPUT_TEXT'),
(34,'kerno',true,'FILTERS.STATE',true,'STATE',4,true,'INPUT_CHECKBOX'),
(35,'kerno',true,'FILTERS.OBJ_TYPE',true,'OBJ_TYPE',1,true,'INPUT_CHECKBOX'),
(38,'kerno',true,'FILTERS.CREATED_USER',true,'CREATED_USER',5,true,'SELECT');
```

La tabla consta de los siguientes campos:

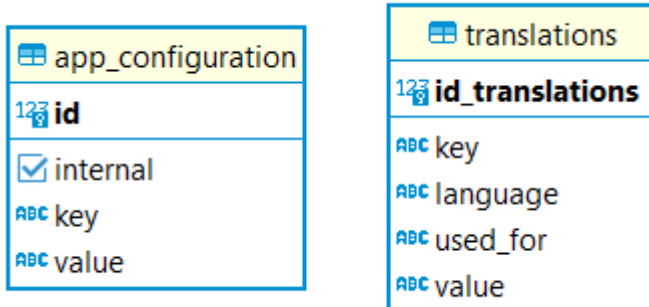
- **id:** (PK) Identificador único, sirve para gestión interna
- **collection:** Identifica la colección de SOLR a la que pertenece: “*kerno*” si es para las entidades y relaciones y “*audit_logs*” si es para auditoría e histórico
- **groupable:** boolean que por defecto adquiere el valor ‘false’. No se hace uso de este campo.
- **label:** KEY que permite que el usuario vea el nombre del filtro en el idioma que tiene configurado. 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 a TRUE si se desea que el filtro sea estático, es decir, que aparezca por defecto sin necesidad de añadirlo
- **name:** Objeto de SOLR por el que va a filtrar el filtro. Para los atributos de metadatos que se añadan como filtros adicionales, en este campo se debe poner el mismo **name** que en la tabla *anjana.attribute_definition*
- **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. Los valores posibles son:
 - INPUT_CHECKBOX para multi-selección
 - SELECT para selección única
 - INPUT_DATE para introducir fechas en calendario
 - INPUT_TEXT para la especificación de un texto

NOTAS:

- Es recomendable no configurar filtros sobre las fechas de la pantalla de Histórico y Auditoría (colección *audit_logs*) puesto que ya hay un selector sobre la tabla visualizada donde se puede elegir inicio y fin del rango de fechas a consultar.

Esquema Portuno de BD

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

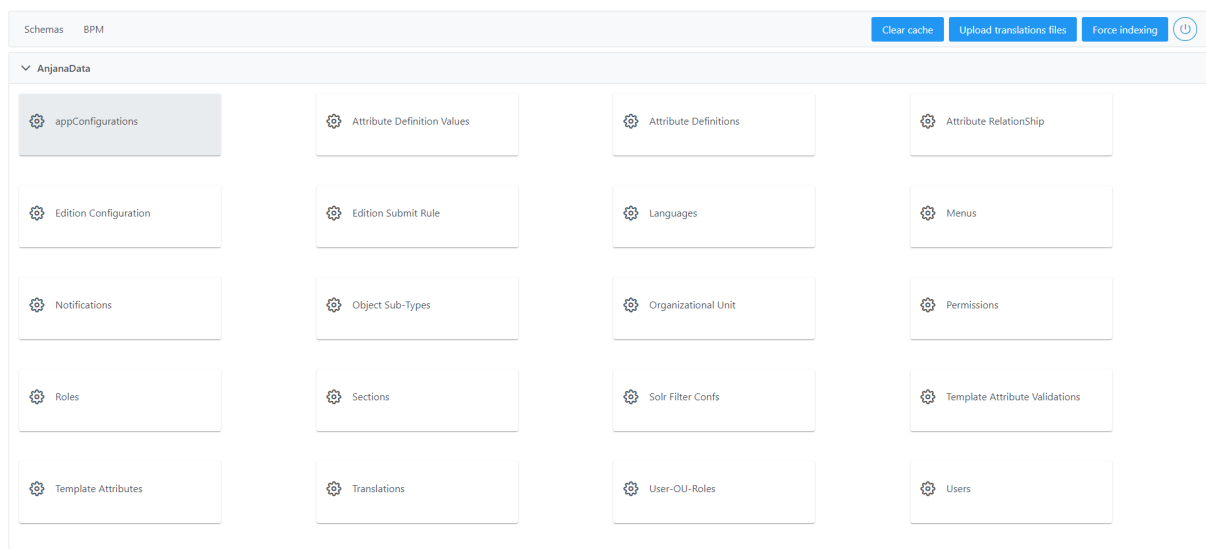


1. Variables del sistema

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.

Visión de Administrador

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



La tabla consta de los siguientes campos:

- **id:** (PK) Identificador único, sirve para gestión interna
- **key:** Clave de la configuración
- **value:** El valor que tiene la propiedad
- **internal:** boolean utilizado para indicar que una propiedad no se debe de ver desde la api pública en claro (internal = false)

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

+ New

appConfigurations				
id ↑↓	key ↑↓	value ↑↓	internal ↑↓	
-	-	-	<input type="checkbox"/>	
	Criterio	Criterio		
	Filtrar	Filtrar	Filtrar	
2	MIN_LENGTH	1	<input checked="" type="checkbox"/>	✎ ✖
3	MAX_LENGTH_INPUT_TEXT	255	<input checked="" type="checkbox"/>	✎ ✖
4	MIN_RANGE	1	<input checked="" type="checkbox"/>	✎ ✖
5	MAX_RANGE	365	<input checked="" type="checkbox"/>	✎ ✖
6	MIN_DECIMAL_PRECISION	1	<input checked="" type="checkbox"/>	✎ ✖
7	MAX_INTEGER	2100000000	<input checked="" type="checkbox"/>	✎ ✖
8	MAX_FIXED_FILTERS_NUMBER	5	<input checked="" type="checkbox"/>	✎ ✖
9	SHOW_RESULTS_SOLR	20	<input checked="" type="checkbox"/>	✎ ✖

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

appConfigurations Edit
✕

key

Select Item
▼

value

1

internal

✕ Cancel

✓ Save

Visión de Desarrollador

Para definir las propiedades de configuración podemos lanzar la secuencia sql:

```
INSERT INTO portuno.app_configuration (id,internal,"key",value) VALUES
(3,false,'MAX_LENGTH_INPUT_TEXT','255'),
(4,false,'MIN_RANGE','1'),
(5,false,'MAX_RANGE','365'),
(6,false,'MIN_DECIMAL_PRECISION','1'),
(7,false,'MAX_INTEGER','2100000000'),
(9,false,'SHOW_RESULTS_SOLR','20'),
(10,false,'ARI_OBJECT_SEPARATOR',':'),
(11,false,'LIMIT_NOTIFICATIONS','30'),
(2,false,'MIN_LENGTH','1'),
(13,false,'MAX_LENGTH','300000'),
(14,false,'ACCESS_MINIO_KEY','anjana'),
(15,false,'ACCESS_MINIO_PWD','anjanadata'),
(17,false,'default_language_app','en-US'),
```

```
(16,false,'MAX_DECIMAL','16'),  
(8,false,'MAX_FIXED_FILTERS_NUMBER','5');
```

La secuencia consta de los siguientes campos:

- **id:** (PK) Identificador único, sirve para gestión interna
- **key:** Clave de la configuración.
- **value:** El valor que tiene la propiedad.
- **internal:** boolean que sirve para indicar que una propiedad no se debe de ver desde la api (internal = false)

NOTAS:

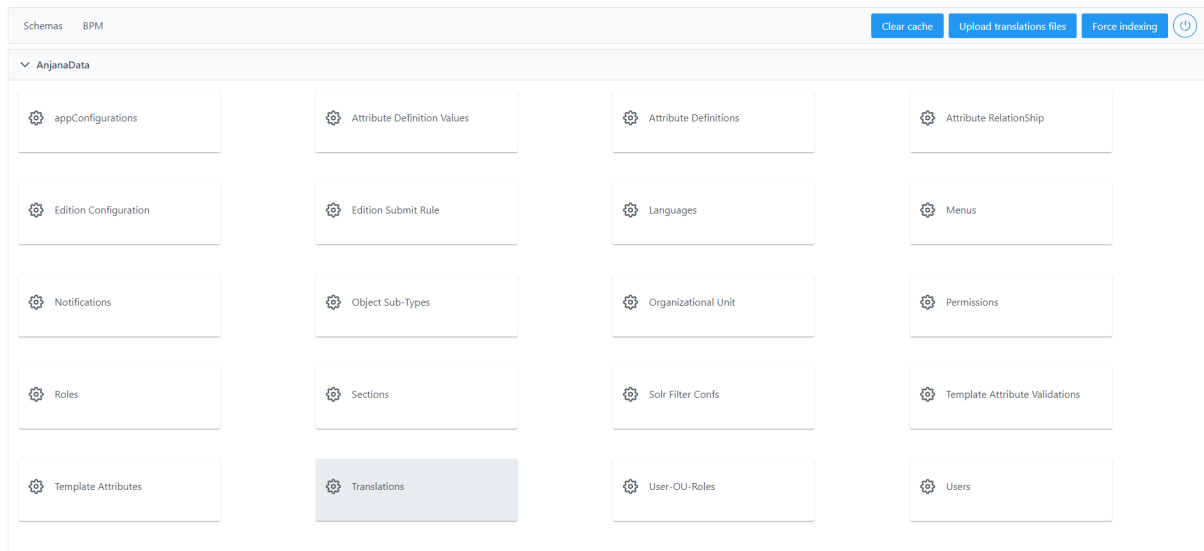
- Puesto que las variables se cargan en el arranque, Portuno debe estar arrancado antes que el resto de servicios para obtener correctamente las propiedades indicadas. Si algún servicio se iniciara antes que Portuno se quedará a la espera de que Portuno esté disponible para poder cargar esta configuración y terminar su inicialización.
- Es necesario que dicha tabla tenga las propiedades:
 - ACCESS_MINIO_KEY → Access key de minio
 - ACCESS_MINIO_PWD → Secret key de minio
 - DEFAULT_LANGUAGE_APP → Idioma por defecto de la aplicación, valor del campo i18n_code de la tabla zeus.languages
 - MIN_LENGTH → Número mínimo de caracteres para input de tipo textarea (default: 1)
 - MAX_LENGTH_INPUT_TEXT → Número máximo de caracteres permitidos en un input de tipo texto (default: 255)
 - MIN_RANGE → Número mínimo permitido en un input de tipo range (default: 1)
 - MAX_RANGE → Número máximo permitido en un input de tipo range (default: 365).
 - MAX_INTEGER → Número máximo de entero (default: 2100000000)
 - SHOW_RESULTS_SOLR → Número de resultados mostrados en búsqueda de solr (default: 20)
 - ARI_OBJECT_SEPARATOR → Carácter usado para separar palabras claves en el nombre de la ARI de archivos subidos a minio (default: “:”)
 - LIMIT_NOTIFICATIONS → Número máximo de notificaciones mostradas por página (default: 30).
 - MAX_LENGTH → Número máximo de caracteres para input de tipo textarea (default: 300000)
 - MAX_DECIMAL → Número máximo de input decimal (default: 10)

2. Traducciones

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

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. Cada traducción se caracteriza por los siguientes elementos:

- **Id:** Identificador único, sirve para gestión interna. Se utiliza id<100000 para las traducciones internas de Anjana, necesarias para la aplicación. Los id>100000 se utilizarán para las traducciones propias de la configuración: atributos, menús, secciones, unidades organizativas, roles...
- **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
- **UsedFor:** Texto que permite segregar las traducciones por uso. Es informativo

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

Translations					+ New
id	key	language	value	usedFor	
-	-	-	-	-	
Criterio	Criterio	Criterio	Criterio	Criterio	
<div style="display: flex; justify-content: space-around;"> Filtrar Filtrar Filtrar Filtrar Filtrar </div>					
2	ACTION_ERROR_KEY	en-US	Error		 
3	ACTION_ERROR_KEY	es-ES	Error		 
4	ACTIONS_CHANGE_CONTAINER	en-US	Change organizational unit		 
5	ACTIONS_CHANGE_CONTAINER	es-ES	Cambiar unidad organizativa		 
6	ACTIONS_SELECT_BU	en-US	Select an organizational unit		 
7	ACTIONS_SELECT_BU	es-ES	Seleccione una unidad organizativa		 
8	ADHERENCE_WIZARD_ASSISTANT.BUTTON	en-US	Calculate Combinations		 
9	ADHERENCE_WIZARD_ASSISTANT.BUTTON	es-ES	Calcular combinaciones		 

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

Translations Edit

key

language

value

usedFor

✕ Cancel
✓ Save

Visión del desarrollador

Existen ciertas traducciones que tienen que existir para que Anjana muestre correctamente toda la información (los identificadores son indicativos) ya que por ejemplo Hermes necesitará de determinadas traducciones:

```
INSERT INTO translations (id_translations,"key","language",value) values
(6557, 'WF.CANCELLATION', 'en-US', 'This task was cancelled because a previous step was rejected')
,(6558, 'WF.CANCELLATION', 'es-ES', 'Esta tarea fue cancelada porque un paso anterior del workflow fue rechazado')
,(6559, 'WF.AUTOVALIDATION', 'en-US', 'This task was automatically validated because the user %s who started the workflow or a previous validator is a candidate to approve this task step')
,(65510, 'WF.AUTOVALIDATION', 'es-ES', 'Esta tarea fue automáticamente validada porque el usuario %s que lanzó el workflow o un validador anterior son candidatos a aprobar esta tarea')
,(65511, 'WF.REJECTION', 'en-US', 'Cancelled by the Rejection of %s')
,(65512, 'WF.REJECTION', 'es-ES', 'Cancelada por el rechazo de %s');
```

NOTAS:

- Es necesario que se configuren las traducciones siguiendo este criterio:
 - id entre 1 y 100mil: traducciones internas de Anjana, necesarias para la correcta visualización de la aplicación
 - id mayor de 100mil: 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.
- Todas las claves de traducción de los valores posibles de los campos infrastructure, technology y zone deben ser iguales que las distintas partes que componen la tripleta con la que se configura la conexión con los plugin 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 infrastructure 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 los atributos Infraestructura, Tecnología y Zona deben no contener ‘:’, ‘#’, ‘(’ o ‘)’ para no interferir con los identificadores internos de Anjana.
- 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.

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 encuentren. 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:

- HEADER_TITLE.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 = 'infrastructure' 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:

- CARDS.ORGANIZATIONAL_UNIT
- CONTAINER
- GRAPH.NODE_ORGANIZATIONAL_UNIT
- HEADER_TITLE.CONTAINER
- MODAL.ORGANIZATIONAL_UNIT
- TOOLTIP.ORGANIZATIONAL_UNIT
- FILTERS.ORGANIZATIONAL_UNIT
- labelTranslationKey y placeHolderTranslationKey del atributo con name = 'organizationalUnit' en anjana.attribute_definition

Origen:

- COMMON.SOURCE
- FORM_RELATION.SOURCE

- HEADER_TITLE.SOURCE
- RELATION.SOURCE
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'source' en anjana.attribute_definition

Destino:

- COMMON.DESTINATION
- FORM_RELATION.DESTINATION
- HEADER_TITLE.DESTINATION
- RELATION.DESTINATION
- labelTranslationKey y placeholderTranslationKey del atributo con name = 'destination' en anjana.attribute_definition

Configuración en ficheros yml

Swagger

Para Hermes, Kerno, Minerva, Portuno y Zeus es necesario configurar una lista de urls que utilizará Swagger para hacer llamadas REST a la API en sus yml. Estas urls deberán apuntar al proxy de viator para el correcto funcionamiento de Swagger.

NOTA: Esta configuración es opcional.

Este es un caso de ejemplo con un servidor de viator levantado en <https://hostviator.org>

```
swagger:  
  server:  
    url:  
      - https://hostviator.org/gateway
```

Kerno

```
anjana:  
  file:  
    explorer:  
      clientType: MINIO  
    folder:  
      textArea: textarea  
      excelImport: imports
```

```

    attach: dsa
    region: eu-central-1
    clientId: anjana
    secretKey: anjanadata
    url: http://s3service:9000
graphics:
  maxLevel: 2
combinations:
  max: 10
tot:
  createCustom: false
scheduling:
  expiration: 0 45 7 ? * *
  notification: 0 45 7 ? * *
  indexAll: 0 45 7 ? * *
notification:
  expiration: 4
separatorAttribute: _-
numberDatasetFieldsForWarning:50

```

- `anjana.file.explorer.*`: define las propiedades de conexión contra el gestor de ficheros
- `anjana.file.explorer.clientType`: gestor de ficheros (MINIO ó AWS_S3)
- `anjana.file.explorer.folder.textArea`: bucket para archivos en atributos de texto enriquecido
- `anjana.file.explorer.folder.excellImport`: bucket para importaciones de excel
- `anjana.file.explorer.folder.attach`: bucket para archivos adjuntos de objetos
- `anjana.file.explorer.clientId`: id de conexión
- `anjana.file.explorer.secretKey`: clave de conexión
- `anjana.file.explorer.url`: url base (sólo aplicable a MINIO)
- `anjana.file.explorer.region`: región de AWS (solo aplicable a AWS_S3)
- `anjana.graphics.maxLevel`: profundidad del linaje por objeto partiendo de la entidad o relación seleccionada. (por defecto = 2)
- `anjana.combinations.max`: máximo número de combinaciones posibles en las adherencias. (por defecto = 10)
- `anjana.tot.createCustom`: indica si Tot puede crear atributos *custom*. (por defecto = false)
- `anjana.scheduling.expiration`: programación en sintaxis *crontab* para expiración de objetos. (por defecto = 0 0 0 ? * *)
- `anjana.scheduling.notification`: programación en sintaxis *crontab* para notificación de objetos próximos a expirar. (por defecto = 0 45 7 ? * *)
- `anjana.scheduling.indexAll`: programación en sintaxis *crontab* reindexación de objetos. (por defecto = -)
- `anjana.notification.expiration`: cantidad de días para analizar objetos que van a expirar. (por defecto = 4)
- `anjana.separatorAttribute`: separador de atributos array, en lugares como la importación por excel vendrán valores concatenados en una celda. (por defecto = _-)
- `anjana.internationalization.separatorNameColumnsExcel`: separador para las columnas del excel para concatenar el idioma en campos internacionales. (por defecto = _)
- `anjana.pagination.defaultSize`: tamaño por defecto de el listado de importaciones (por defecto = 30)

- `anjana.pagination.defaultPage`: página de comienzo en el listado de importaciones (por defecto = 0)
- `anjana.numberDatasetFieldsForWarning`: número de `dataset_fields` a partir del cual, cuando el usuario lleva a cabo alguna acción sobre el dataset, Anjana le avisa de que va a tardar unos minutos y le redirige al portal. Configuración opcional (por defecto = 100)
- `anjana.workflow.launchIfCustomChanges`: flag para tener en cuenta los atributos custom para poder enviar el objeto a validar en un workflow por defecto = true)
- `wf.custom`: posibilidad de lanzar un flujo (*workflow*) en función de un atributo. En el ejemplo se puede observar que se va a lanzar el flujo para el atributo *obfuscation* de las entidades de tipo DSA y para el atributo *obfuscation* de las relaciones de tipo adherence.

Drittista

N/A

Viator

N/A

Hecate

N/A

Hermes

```
wf:
  validation:
    automatic: true
  async:
    waiter: 20000
anjana:
  file:
    explorer:
      clientType: MINIO
    folder:
      workflows: workflows
      cdn: cdn
    region: eu-central-1
    clientId: anjana
    secretKey: anjanadata
    url: http://s3service:9000
```

- `wf.validation.automatic`: indica si los pasos de los flujos se validan automáticamente cuando el creador o un validador anterior puede validar un paso futuro.
- `wf.async.waiter`: Tiempo de espera del servicio entre los distintos pasos a ejecutar en la validación automática.

- `anjana.file.explorer.*`: define las propiedades de conexión contra el gestor de ficheros
- `anjana.file.explorer.clientType`: gestor de ficheros (MINIO ó AWS_S3)
- `anjana.file.explorer.folder.textArea`: bucket para archivos en atributos de texto enriquecido
- `anjana.file.explorer.folder.excellImport`: bucket para importaciones de excel
- `anjana.file.explorer.folder.attach`: bucket para archivos adjuntos de objetos
- `anjana.file.explorer.clientId`: id de conexión
- `anjana.file.explorer.secretKey`: clave de conexión
- `anjana.file.explorer.url`: url base (sólo aplicable a MINIO)
- `anjana.file.explorer.region`: región de AWS (solo aplicable a AWS_S3)

Minerva

```
solr:  
  multivalued-separator: _-  
  module:  
    kerno:  
      collection: kerno  
      shards: 1  
      replication: 1  
      generate: true
```

- `solr.multivalued-separator`: valor que indica con qué separador se están persistidos los valores de los campos con valores múltiples. Tiene que coincidir con la propiedad `anjana.separatorAttribute` de Kerno. (por defecto = `_-`)
- `solr.url`: el listado de urls de nodos con instancias de Apache SolR. En caso de más de una instancia separar las urls con “,”.
- `solr.client.type`: indica el tipo de Apache SolR utilizado. Acepta CLOUD para cluster y HTTP para una única instancia.
- `solr.user`: el usuario de conexión, en caso de que la seguridad esté habilitada.
- `solr.password`: la contraseña de conexión, en caso de que la seguridad esté habilitada.
- `solr.security`: flag que indica si la seguridad está habilitada en los nodos de Apache SolR.
- `solr.module`: especificación de las colecciones en Apache Solr.
- `solr.module.<name>.collection`: nombre de la colección.
- `solr.module.<name>.shards`: número de *shards* en la colección (usado cuando la colección es muy grande para un sólo nodo). Se recomienda usar un único *shard*.
- `solr.module.<name>.replication`: número de servidores en los que se tiene que replicar la información indexada. Se recomienda usar un factor de replicación de 1.
- `solr.module.<name>.generate`: flag que especifica si se debe crear la colección en el arranque de Minerva en caso de que no esté creada.

Portuno

```
anjana:
  file:
    explorer:
      clientType: MINIO
      folder:
        workflows: workflows
      region: eu-central-1
      clientId: anjana
      secretKey: anjanadata
      url: http://s3service:9000
```

- anjana.file.explorer.*: define las propiedades de conexión contra el gestor de ficheros
- anjana.file.explorer.clientType: gestor de ficheros (MINIO ó AWS_S3)
- anjana.file.explorer.folder.textArea: bucket para archivos en atributos de texto enriquecido
- anjana.file.explorer.folder.excellImport: bucket para importaciones de excel
- anjana.file.explorer.folder.attach: bucket para archivos adjuntos de objetos
- anjana.file.explorer.clientId: id de conexión
- anjana.file.explorer.secretKey: clave de conexión
- anjana.file.explorer.url: url base (sólo aplicable a MINIO)
- anjana.file.explorer.region: región de AWS (solo aplicable a AWS_S3)

Tretesta

N/A

Zeus

```
anjana:
  scheduling:
    notification: 0 0 0 ? * *
    licenseNotification: 0 0 0 ? * *
  security:
    authentication:
      db:
        enable: false
    authorization:
      db:
        enable: false
```

- anjana.scheduling.notification: configuración de tarea programada para avisar de roles Cross mal configurados. (por defecto = 0 0 0 ? * *)

- `anjana.scheduling.licenseNotification`: configuración de tarea programada para enviar notificaciones por licencia a punto de expirar o expirada. (por defecto = 0 0 0 ? * *)
- `security.authentication.db.enabled`: flag para habilitar el uso de autenticación por base de datos (por defecto = false)
- `security.authorization.db.enabled`: flag para habilitar el uso de autorización por base de datos (por defecto = false)

Las integraciones de autorización y autenticación se detallan en los documentos específicos para tal fin:

- Anjana Data 4.x - DS - Integración Azure
- Anjana Data 4.x - DS - Integración AWS
- Anjana Data 4.x - DS - Integración GCP
- Anjana Data 4.x - DS - Integración LDAP
- Anjana Data 4.x - DS - Integración OKTA

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

Ejemplo de configuración de proveedor de identidades

Configuración de base de datos

Para la configuración de permisos a través de la base de datos se va a necesitar interactuar con 3 tablas:



- Usuario (user)
- Unidad organizativa (organizational_unit)
- Rol (role)
- Relación entre usuario, rol y Unidad organizativa (user_ou_role)

Todos los datos se pueden dar de alta mediante la interfaz disponible desde Portuno. De esta forma en la tabla de Usuarios se dará de alta a los distintos usuarios que hay disponibles.

Schemas		BPM						Load lineage	Clear cache	Upload translations files	Force indexing		
												+ 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						Filtrar
2	architect	Architect	arc	architect@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
3	administrator	Admin	adm	admin@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
9	business.user	Business	User	business.user@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
10	api.admin	API	Admin	api.admin@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
11	default.user	Default	User	default.user@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
12	portuno.admin	Portuno	Admin	portuno.admin@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
13	portuno.credential	Portuno	Credential	portuno.credential@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
14	developer.europe	Developer	EU	developer.eu@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
15	developer.latam	Developer	LATAM	developer.latam@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						
16	developer.general	Developer	Gen	developer.general@corporate.com	(+34)999 999 999	Empleado	\$2a\$10\$w6y3docQwV08hgF8n9fL6uwwp8O8aSK8NEMapAWTU7h/13/zjIKj.						























La contraseña está almacenada encriptada (bcrypt) y hay que añadirla al sistema de esta manera cuando se quiera dar de alta un usuario.

Para las Unidades Organizativas hay que indicar los distintos niveles de jerarquía existentes mediante la columna parent, dejando vacío ese campo si se quiere dar de alta una OU de primer nivel e indicando el identificador de la OU padre según el nivel a crear.



Schemas BPM Load lineage Clear cache Upload translations files Force indexing  

+ New

Organizational Unit























Id ↑	Description ↑↓	Module ↑↓	alias ↑	parent ↑↓	organizationalUnitType ↑↓	
-	-	Igual	-	Igual	-	
<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="text" value="-"/>	<input type="text" value="Criterio"/>	<input type="text" value="-"/>	<input type="text" value="Criterio"/>	
Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	Filtrar	
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	 
11	Marketing	ALL	SCIB_HQ/Marketing	8-General/SCIB_HQ	2	 

Los roles se definen de igual manera en su sección dedicada

Schemas BPM Load lineage Clear cache Upload translations files Force indexing  

+ New

Roles

Id ↑	Module ↑↓	Name ↑↓	Description ↑↓	isCross ↑↓	
-	Igual	-	-		
<input type="text" value="Criterio"/>	<input type="text" value="-"/>	<input type="text" value="Criterio"/>	<input type="text" value="Criterio"/>	<input type="checkbox"/>	
Filtrar	Filtrar	Filtrar	Filtrar		
1	ALL	developer	Developer	<input checked="" type="checkbox"/>	 
2	ALL	architect	Architect	<input checked="" type="checkbox"/>	 
3	ALL	administrator	Administrator	<input checked="" type="checkbox"/>	 
4	ALL	data_steward	Data Steward	<input checked="" type="checkbox"/>	 
5	ALL	data_owner	Data Owner	<input checked="" type="checkbox"/>	 
6	ALL	compliance	Compliance	<input checked="" type="checkbox"/>	 
7	ALL	process_owner	Process Owner	<input checked="" type="checkbox"/>	 
8	ALL	contributor	Role for responsible of entity or relationship	<input checked="" type="checkbox"/>	 
9	ALL	business_user	Business User	<input checked="" type="checkbox"/>	 
10	ALL	apiAdmin	Admin by API	<input checked="" type="checkbox"/>	 
11	ALL	default	default	<input checked="" type="checkbox"/>	 

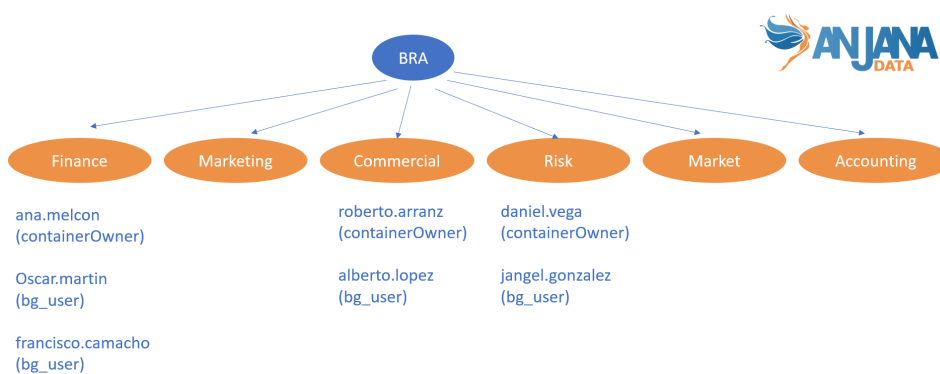
Y finalmente la asignación de los roles a cada OU y usuarios:

User-OU-Roles		
user ↑↓	ou ↑↓	role ↑↓
Igual	Igual	Igual
-	-	-
Filtrar	Filtrar	Filtrar
14-developer.europe	54-Europe/PT/Telco	1-developer-Developer
14-developer.europe	7-Europe/SPA/Accounting	1-developer-Developer
14-developer.europe	4-Europe/SPA/Commercial	1-developer-Developer
14-developer.europe	2-Europe/SPA/Finance	1-developer-Developer
14-developer.europe	41-Europe/SPA/HealthCare	1-developer-Developer
14-developer.europe	6-Europe/SPA/Market	1-developer-Developer
14-developer.europe	3-Europe/SPA/Marketing	1-developer-Developer
14-developer.europe	5-Europe/SPA/Risk	1-developer-Developer
14-developer.europe	28-Europe/UK/Accounting	1-developer-Developer
14-developer.europe	25-Europe/UK/Commercial	1-developer-Developer
14-developer.europe	23-Europe/UK/Finance	1-developer-Developer

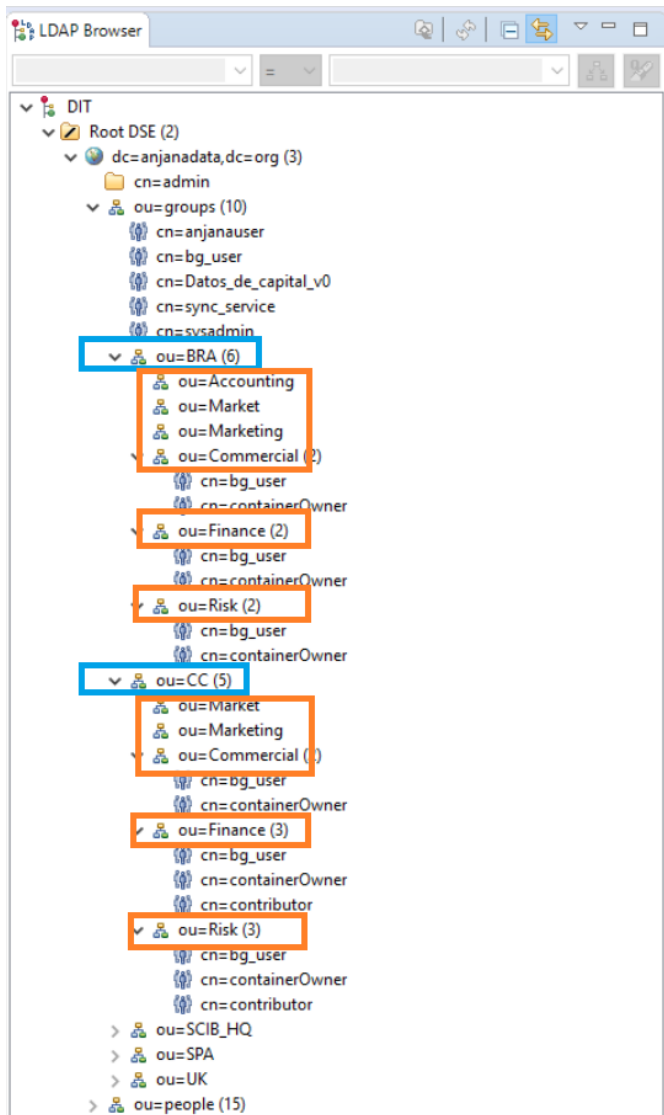
Donde se indica para cada usuario cuál es el rol asignado en una OU en particular.

Configuración de LDAP

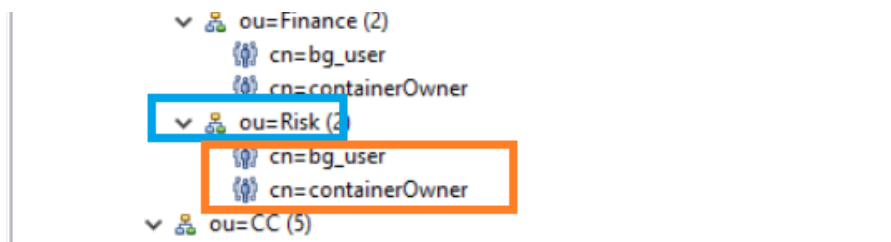
Dentro de **groups** se deben dar de alta las unidades organizativas con su correspondiente jerarquía de niveles padre-hijo.



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**:



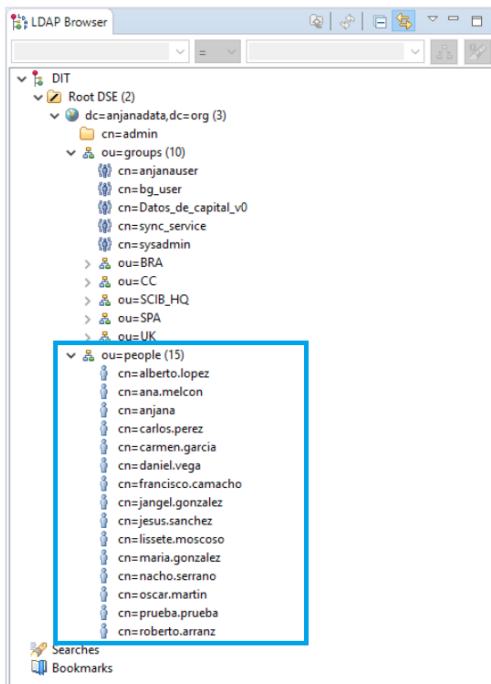
También se debe indicar el rol que tiene un determinado usuario dentro de la Unidad Organizativa:

```

#!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

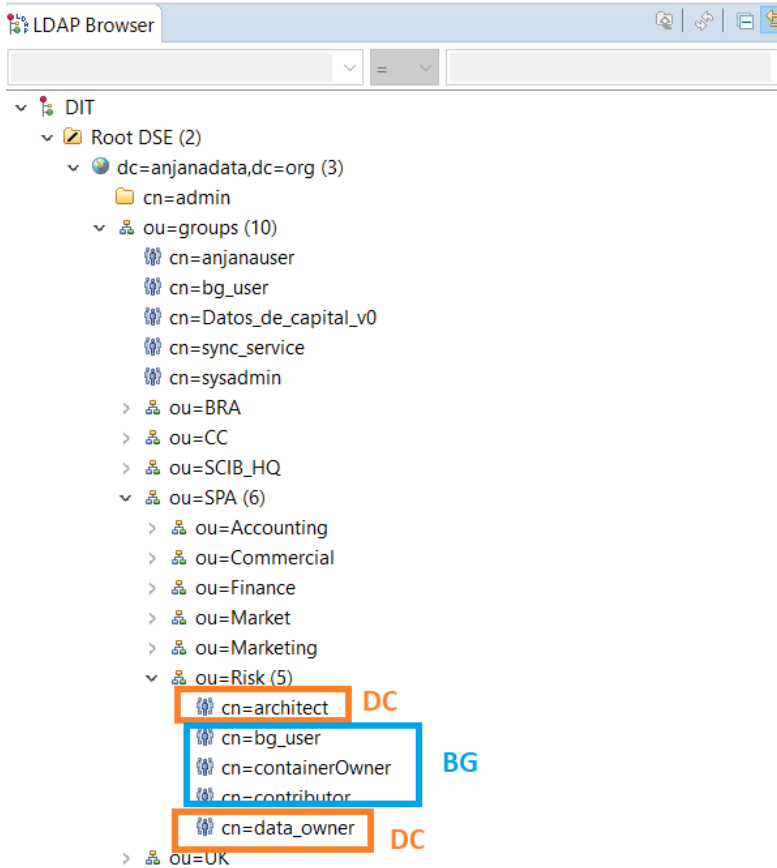
```

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

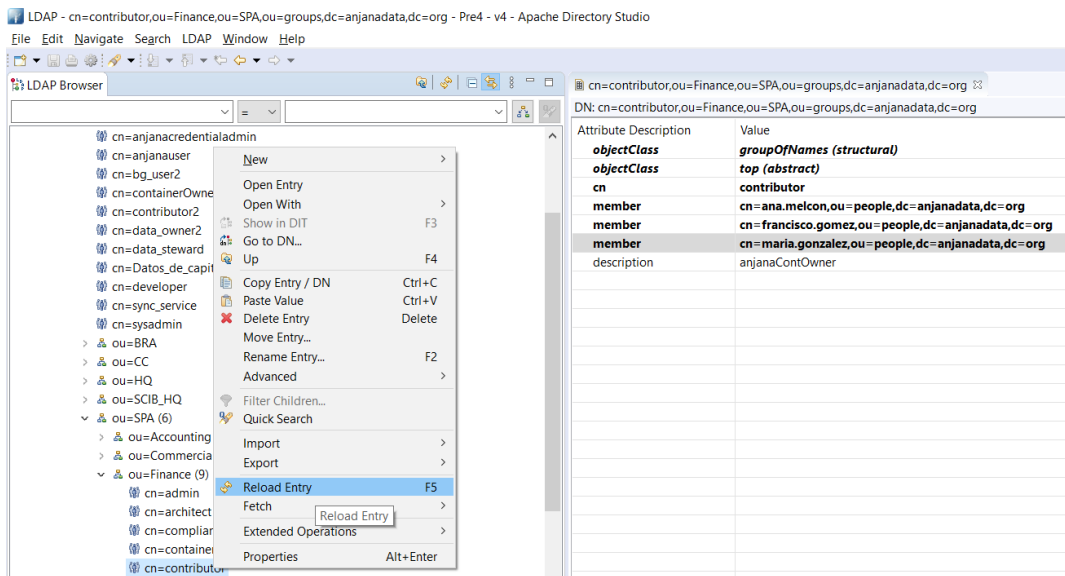



NOTAS:







- 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

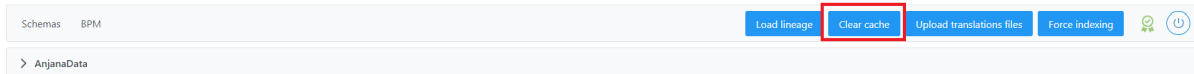
Proveedores externos

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 se puede configurar su programación en el yaml de su servicio correspondiente, 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 usa: `0 0 8 * * *` serviría para configurar un batch que 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 el fichero yaml de Zeus:

```
anjana:
  scheduling:
    notification: 0 0 0 ? * *
```

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

2. Licencia a punto de expirar

También se puede configurar el aviso de licencia a punto de expirar en el fichero yaml de Zeus:

```
anjana:
  scheduling:
    licenceNotification: 0 0 0 ? * *
```

- `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 ? * *`).

3. Expiración

El batch de expiración se configura en el fichero yaml de Kerno:

```
anjana:
  scheduling:
    expiration: 0 0 0 ? * *
```

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

4. Preaviso por expiración

El batch de preaviso por expiración se configura en el fichero yaml de Kerno:

```
anjana:
  scheduling:
    notification: 0 45 7 ? * *
  notification:
    expiration: 4
```

- `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. Para el preaviso aquí mencionado (por defecto: `4`).

5. Indexación completa de elementos

El batch de indexación completa de elementos se configura en el fichero yaml de Kerno:


```
anjana:  
scheduling:  
  indexAll: 0 0 2 ? * *
```

- anjana.scheduling.indexAll se usa para configurar la reindexación de objetos (por defecto: -).

Preguntas frecuentes

1. Permisos

Un rol que no puede crear objetos tiene disponible el + 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 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. Anjana Portal

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. Lanzamiento 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. ¿Cómo puedo saber si se han desplegado correctamente los 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. 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 las tablas `object_subtype` y `template` y deben añadirse en `template_attribute` los registros correspondientes a los campos `finishDate`, `requestReason` y `pae` para ella.

6. 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.

7. 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, se desactivará.

Para ello, en la tabla `template_attribute`, se cambiará, en el registro concreto para 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.

8. ¿Cómo configurar una nueva plantilla en Anjana?

- a. Definir el `template` a configurar (tabla `anjana.object_subtype`)
- b. Definir los menús que van a completar su formulario dinámico y las secciones de estos (tablas `anjana.menu` y `anjana.sections`)

- c. 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`)
- d. Organizar los atributos en las secciones de la plantilla (tabla `anjana.template_attribute`)
- e. Definir las validaciones de atributos para la plantilla (si el atributo es requerido, si depende de otro, si es editable...) (tabla `template_attribute_validation`)
- f. 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`).

9. ¿Cómo borrar completamente campos de Solr si al borrar una colección vemos que siguen existiendo campos que no deberían?

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

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