DOCUMENTACIÓN TÉCNICA PARA CREACIÓN, MODIFICACIÓN Y ACTUALIZACIÓN DE PIPELINE ETL DE CONSULTAS BIBLIOGRÁFICAS

Autor: Mgs. Robinson Barrazueta

Fecha: 09 de noviembre de 2025

PIPELINE ETL PARA CONSULTAS EN CROSSREF Y VISUALIZACIÓN EN APACHE SUPERSET

Resumen Ejecutivo:

Este proyecto ha desarrollado un sistema automático ETL y visualización de datos bibliográficos mediante el uso de scripts en Python, bases de datos SQLite y dashboards interactivos con Apache Superset, tomando como fuente de información a CrossRef, una organización sin fines de lucro cuyo trabajo es brindar infraestructura a la comunidad de investigación mundial. A través de este pipeline es posible analizar el estado de la generación científica universitaria de forma directa, gratuita, organizada y automática, además de ser una solución replicable y escalable.

Entregables:

- Código en Python (*barrazueta_pipeline_etl_crossref.py*) con el detalle de los logros del proceso ETL, creación de base de datos SQLite y creación de archivo institucional complementario y su integración a la base de datos.
- Archivo complementario en formato CSV (*ups_institucional.csv*) con el detalle de sedes y áreas del conocimiento para su integración con la base de datos y útil posteriormente para la visualización en Apache Superset.
- Base de datos generada a través del pipeline ETL (*barrazueta_db_ups_crossref.db*) que contiene toda la información requerida sobre las obras con afiliación “Universidad Politécnica Salesiana”.
- Archivo formato PDF con la exportación del panel de visualización de Superset (*dashboard-ups-robinson-barrazueta-2025-11-09T21-10-18.656Z.pdf*).
- El documento técnico en formato PDF.

Prerrequisitos:

1. Instalación de Python 3.12 (evitar versiones superiores por cuestiones de compatibilidad con Apache Superset, incluso si se puede instalar una versión anterior como 3.11 es mucho mejor).
2. Instalación de un IDE para la edición del código Python (recomiendo PyCharm en su versión Community, la cuál es gratuita).
3. En caso de no tenerlas, instalar las dependencias de Visual Studio C++ (visitar este sitio web: https://visualstudio.microsoft.com/es/visual-cpp-build-tools/) ya que son obligatorias como compiladores nativos de Superset.

1. Creación y Ejecución del pipeline ETL en Python

1.1. Flujo de Trabajo del script en Python:

Para la creación del código debemos tener claro qué es lo que necesitamos que realice el mismo, por lo cuál se genera el siguiente flujo de trabajo sobre el cuál se diseñará el código:

1. Extracción: El script debe conectarse a CrossRef para buscar y obtener información sobre trabajos publicados entre el 1-ene-2022 y el 30-nov-2025 donde aparezca la afiliación “Universidad Politécnica Salesiana”.
2. Transformación: Una vez con los trabajos encontrados, se debe recoger información puntual sobre ellos: título, año y fecha de publicación, revista, editorial, tipo de documento, citas, autores y sus afiliaciones. La transformación también incluye un proceso de limpieza, por lo que el código, luego de recoger la información requerida, debe:
    a. limpiar nombres y códigos: librarse de caracteres especiales, estandarizar DOIs, entre otros.
    b. detecta si la afiliación es de la UPS y qué sede menciona (Cuenca, Quito o Guayaquil),
    c. reconocer el país que aparece en la afiliación,
    d. guardar los temas declarados de los trabajos,
    e. filtrar y limpiar la información, eliminando duplicados.
3. Carga: una vez se tiene la información lista, se debe crear una base de datos relacional SQLite donde se encuentre todos los datos obtenidos correctamente enlazados. Esta base de datos debe ser capaz de ser traducida por Apache Superset para la creación de un dashboard.
4. Se debe generar un archivo complementario para etiquetado de sedes y áreas con fines de filtrado en Superset.

Con este flujo de trabajo se empieza la programación.

1.2. Programación en Python:

El código se encuentra adjunto a este informe, en donde se puede revisar detalladamente todos los comentarios e información particular para cada una de las secciones del código. Para este reporte se incluye la información más relevante y útil para su entendimiento y replicación.

- Librerías: se trabaja con las librerías requests (para peticiones a la API de CrossRef, unicodedata, re, time, html, sqlite3 (para creación de base de datos), json, os, pandas (para limpieza y tratamiento de datos).
- Funciones: se encuentran funciones para proteger el pipeline en caso de problemas de conexión, para extracción de datos específicos, entre otros. Destaco la función de creación de la base de datos ejecutando SQLite dentro de Python a través de su función. Además, se incluye una sección de creación de logs con propósitos de auditoría.
- Técnica de Paginación: para el script adjunto se realiza un análisis por "páginas de resultados" usando cursor para conocer si ya se analizó todos los resultados de una página y, por ende, continuar a la siguiente para repetir el proceso. Una vez que encuentra una o varias páginas que ya no arrojan resultados (se determinó 15 en este código, pero se puede modificar), el código se detiene y arroja los resultados encontrados hasta ese momento para evitar un bucle infinito.
- Variables de afiliación: debido a que es posible que algunas obras hayan sido almacenadas con errores o con afiliaciones a UPS en inglés o con otra redacción, se agrega una sección de variables para mejorar la captura de obras. Se encuentran desactivadas para este código.
- Buenas prácticas en request a CrossRef: se identifica claramente el cliente que está solicitando y su email, con el fin de contacto de parte de CrossRef si existe un problema que afecte su API (se identifica con User-Agent y mailto).

1.3. Resultados Finales:

Como resultado final se espera un texto como el mostrado en la figura 1, donde se comprueba que la ejecución fue exitosa.

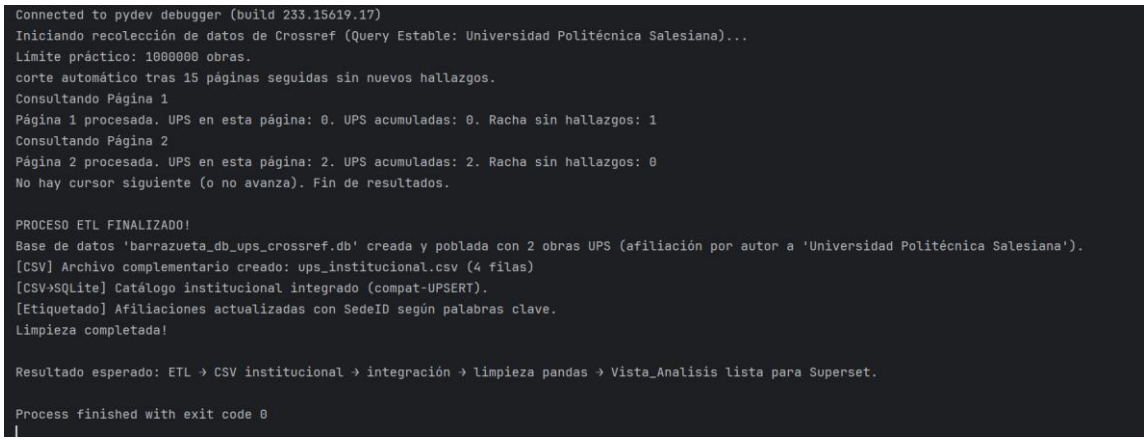

Fig. 1: resultado de código en consola mostrando que el script se ejecutó con éxito.

A tomar en cuenta:

- Se recomienda la ejecución del código varias veces (4-5 veces) para asegurar que se está capturando todos los artículos con afiliación UPS. No hay que preocuparse por la estructura del CSV, el código está preparado para sobrescribir el CSV con nueva información de ser el caso. Esto es recomendable antes de su carga a Superset.
- Hubo una situación complicada a la hora de obtener los temas (subjects) de las obras capturadas, ya que CrossRef no brindaba esa información, por lo que la sección de temas no pudo ser introducida. Sin embargo, en mira al futuro, esto se puede obtener al analizar el contenido/título/resumen del trabajo, mucho más sencillo si se incluyen APIs de IA como Gemini para esto.
- Para la base de datos entregada se muestra un número mayor al observado en las capturas de pantalla de Superset, esto se debe a que, en una última ejecución de prueba, el código devolvió nuevos resultados de obras afiliadas a UPS y la base de datos se actualizó. Es por eso que se recomienda hacer las ejecuciones deseadas antes de su carga a Superset.

Con esto claro y en mente, se procede a la conexión de esta base de datos con Apache Superset para su visualización.

2. Creación y Visualización del dashboard en Apache Superset

Para la creación del dashboard en Apache Superset es necesario primero instalar y configurar correctamente el entorno donde se va a ejecutar Superset. Recordar que todo cambio y modificación se debe hacer en el entorno virtual que se crea a continuación.

2.1. Creación de entorno virtual:

Para usar Apache Superset hay una serie de múltiples consideraciones a tomar en cuenta, comenzando por la creación de un entorno virtual (venv). Una buena práctica es utilizar Docker para usa Superset, evitando errores de dependencias, pero para este caso se usará un entorno virtual separado de la versión de Python principal del ordenador.

1. Crear una carpeta dedicada al entorno, considerando que va a contener la totalidad de una versión de Python y los archivos de Superset, que la pueden hacer más pesada de lo esperado.
2. Desde una terminal, usar el comando “py -3.11 -m venv .venv” para la instalación del entorno.
    a. Si bien en este reto se ha utilizado Python 3.12, como buena práctica se recomienda 3.11. No olvidar incluir también la carpeta donde será instalada este entorno, usando el comando cd y luego ubicando la ruta de la misma.
3. Usar el comando “.\.venv\Scripts\Activate.ps1” para levantar el venv. Esto hará que la ruta de ejecución del terminal tenga el prefijo (.venv), lo que indicará que lo hiciste correctamente.
4. Con el venv activo, actualizar las dependencias con el comando “python -m pip install --upgrade pip setuptools wheel”.
5. Superset no soporta una versión de Marshmallow 4.x, es necesario instalar una versión 3.x usando el comando “pip install marshmallow==3.26.1”.
    a. Esto mismo podría aplicar a SQLAlchemy, pero para este caso no nos preocuparemos de eso.
6. Instalar Apache Superset con el comando “pip install apache-superset”.
7. Ejecutar el comando $env:FLASK_APP="superset" para configurar la variable de entorno Flask antes de cualquier comando relacionado con Superset **(esto se debe realizar siempre que se vaya a levantar una sesión de Superset).**

Con esto se ha instalado Superset y se puede preparar el entorno para su ejecución por primera vez.

2.2. Configuración de variables y ejecución de Superset:

Antes de ejecutar Superset por primera vez, es necesario ejecutar algunos comandos previos, algunos de ellos será necesario ejecutarlos siempre que se levanta una nueva sesión de Superset, como en la consideración de la sección anterior. Antes de ello, se debe ejecutar los siguientes pasos:

- Se debe generar un SECRET_KEY para la configuración de Apache Superset. Recomiendo uso de función “openssl rand -base64 42” para una clave segura, en un script cualquier de Python.
- Configurar archivo “superset_config.py” (crear archivo en carpeta raíz del proyecto, importante la modificación de la última línea del código) con SECRET_KEY, quitando también la prohibición de bases de datos de SQLite, como se muestra en el código de la fig. 2. Ojo, Superset suele bloquear bases de datos SQLite, se podría migrar a PostgreSQL en el futuro.

```python
import os
from datetime import timedelta

# Genera tu SECRET_KEY con: openssl rand -base64 42
# O usa cualquier string largo y aleatorio, el que tu quieras
SECRET_KEY = 'Gwd3Lfn1LvABuyWS8l9XijblOkyujYxyccMzf5gQagUtqi2Zv-I8yuK1A7F5AZ74vX_5Edkx4Dm3TZ1dU0ZGyA'

# Permitir conexiones SQLite (solo para desarrollo local, no recomendable si se externa)
PREVENT_UNSAFE_DB_CONNECTIONS = False

# Base de datos interna de Superset (metastore)
# ESTO CAMBIA DEPENDIENDO DEL CASO
SQLALCHEMY_DATABASE_URI = 'sqlite:///C:/Users/Lenovo/Downloads/RETO_UPS_BARRAZUETA/pythonProject/superset.db'
```

Fig. 2: código de configuración de Superset en Python.

Para la ejecución es necesario realizar los siguientes pasos:

1. Ejecutar el comando “superset db upgrade” para actualizar la base de datos que hemos creado en el paso anterior.
2. Ejecutar el comando “superset fab create-admin” para crear nuestras credenciales. El proceso es interactivo en la terminal y se puede ir respondiendo las opciones que nos da, pero también, si se desea, se puede utilizar flags para introducir toda la información en un solo comando.
3. Ejecutar el comando “superset init” para iniciar roles y permisos.
4. Ejecutar el comando “superset run” para ejecutar Superset. Como alternativa, se propone el comando “superset run -p 8088 --with-threads --reload –debugger” para indicar todas las configuraciones por defecto.

Una vez realizadas estas configuraciones, dentro de nuestro navegador favorito se ingresa a la dirección: http://127.0.0.1:8088 o http://localhost:8088 en la cual la primera

pantalla es Login, donde debemos ingresar el usuario y contraseña que hemos definido anteriormente. Al ingresar veremos una pantalla como la mostrada en la figura 3 (al ser la primera vez, aparecerá en blanco). Con esto, estamos listos para el enlace de la base de datos y la creación de tablas y dashboards.

Como consideración final, se recomienda la ejecución de los comandos: $env:FLASK_APP="superset", superset init y superset run cada vez que se vaya a levantar el servicio.

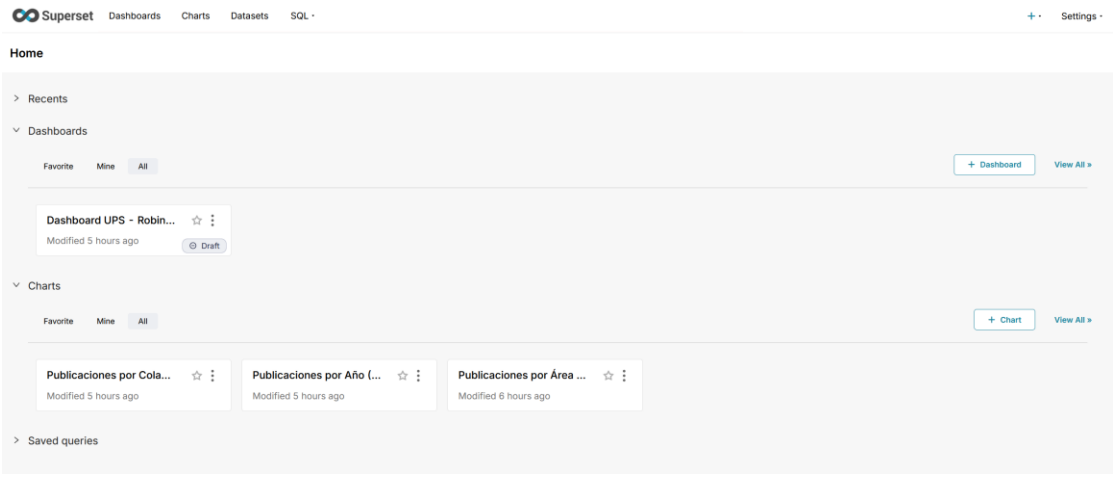

Fig. 3: pantalla inicial de Apache Superset

2.3. Enlace de la base de datos y creación de tablas y dashboard:

Una vez ya en la pantalla inicial, nuestro primer paso es la conexión de la base de datos con Superset, para su enlace con un dataset. Se realiza de la siguiente manera:

1. En la parte superior, hacer clic en “Settings” y hacer clic en “Database Connections”
2. En la nueva ventana mostrada, hacer clic en “+ Database”.
3. Seleccionar “SQLite” como base de datos. Se desplegará un cuadro de diálogo como el de la fig. 4.
4. Modificar el nombre de la base de datos, con ese se le conocerá a la base de datos en el entorno Superset.
5. Importante, se debe incluir la SQLAlchemy URI del archivo que se va a incluir como base de datos. El formato para generar esta dirección, para SQLite, es: sqlite:///C:/ruta/al/archivo.db (3 barras si la dirección es absoluta).
6. Como recomendación, hacer clic en “Test Connection” para saber si Superset ha encontrado a la base de datos. Si no hay errores, hacer clic en “Connect”.
7. Listo, la base de datos ha sido enlazada a Superset.

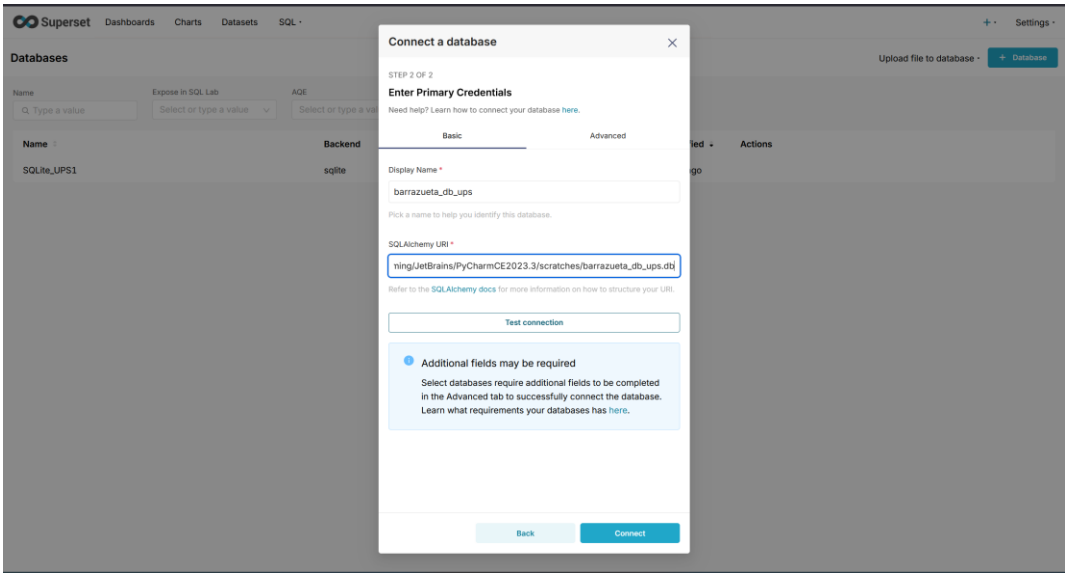

Fig. 4: Cuadro de diálogo para la conexión de una base de datos.

Una vez ya conectada la base de datos, el siguiente paso es la creación de un dataset y la conectaremos a nuestra base de datos para

1. En la parte superior seleccionar “SQL” y hacer clic en “SQLab”.
2. En la nueva ventana a la izquierda, hacer clic en Database y seleccionar la base de datos que hemos conectado.
3. En el espacio de la derecha, copiar los comandos mostrados en la fig. 5 y hacer clic en Run.
4. Hacer clic en la flecha junto a “Save” y seleccionar “Save Dataset”.

Fig. 5: código para creación de dataset en Superset.

Una vez creado el Dataset, se da paso a la creación de las gráficas:

1. En la parte superior, hacer clic en “Chart” y en la nueva ventana desplegada, hacer clic en “+ Chart”.
2. Se muestra una visualización como la de la fig. 6. Aquí se puede seleccionar el gráfico a desarrollar, sin olvidar primero seleccionar el dataset de trabajo. Se puede hacer uso de la herramienta de selección para buscar el gráfico deseado. Para este reto se usará el gráfico de barras. Hacer clic en “Create new Chart”.

3. Se despliega una imagen como la mostrada en la fig. 7, donde se hacen las configuraciones necesarias para los gráficos. Tomar en cuenta lo siguiente:
    a. Los valores de “x-axis” y “metrics” son obligatorios.
    b. Se puede añadir filtros, aunque es preferible hacerlo en el dashboard.
    c. Se puede configurar la orientación, colores, títulos, entre otros.
4. Una vez realizado el proceso de configuración a deseo del usuario, hacer clic en “Create Chart” en la parte inferior. Se mostrará el gráfico como el usuario lo ha deseado. Si el usuario está satisfecho con el resultado y es la primera vez que va a publicar en dashboard, hacer clic en “Save” de la parte superior derecha. Se mostrará un cuadro de diálogo con opción para escoger un dashboard o crear uno nuevo, que será lo que se escogerá por primera vez.
5. El proceso se repite las veces que sea necesario hasta obtener los gráficos deseados. Una vez terminado, se puede salir de la configuración y entrar a Dashboards en la parte superior.

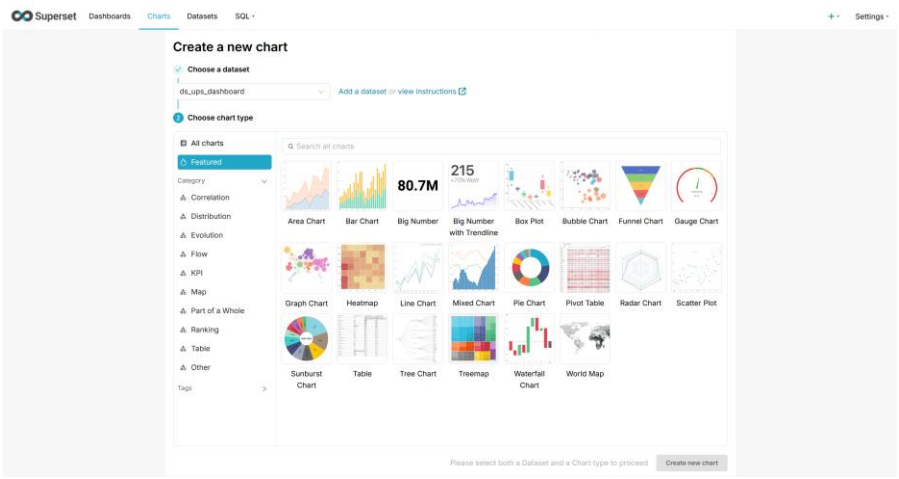

Fig. 6: selector de gráficos en Apache Superset.

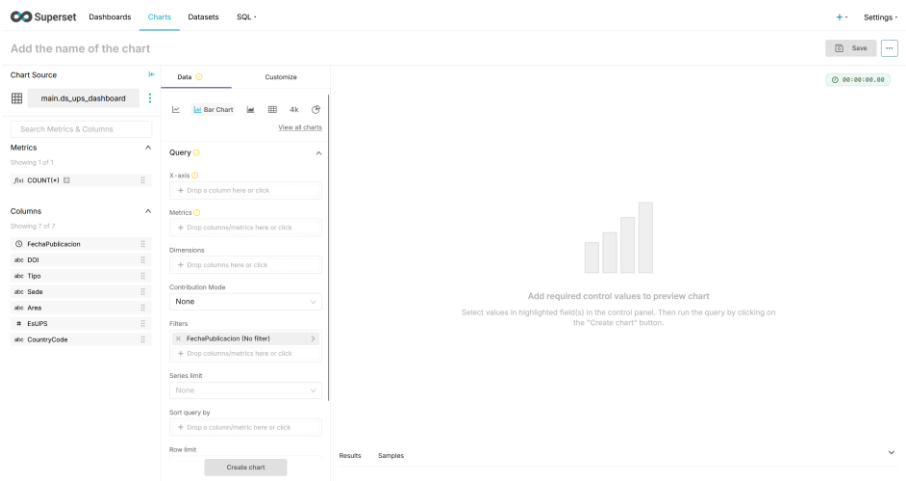

Fig. 7: configuración de gráficos en Superset.

Una vez que ya se tiene los gráficos generados y almacenados en el dashboard, es hora de configurar el dashboard y crear los filtros de visualización:

1. Una vez seleccionada la opción “Dashboard” de la parte superior de la ventana, se mostrará la lista de dashboards disponibles. Se escoge el que se desea personalizar.
2. Una vez abierto, se puede clic en “Edit Dashboard”. En esta sección se puede desde crear un nuevo gráfico hasta añadir columnas, filas, texto, modificar la ubicación de los gráficos ya creados, entre otros.
3. En la parte izquierda se encuentra un botón de una flecha y 3 rayas, hacer clic para abrir los filtros. Por default, no existirá ninguno.
4. Hacer clic en la rueda y seleccionar “Add or edit filters”, se mostrará una ventana como la de la fig. 8 (los filtros ya existentes son de mi autoría).
5. Se tiene opciones para filtro de tiempo, valor, rango numérico, para seleccionar varios valores de filtro a la vez, indicar a cuál columna afectará el filtro, el nombre del filtro, el dataset que se está filtrando, entre otros. Dependiendo del resultado deseado es que se elegirá una u otra opción.
6. Una vez terminada la configuración, hacer clic en “Save”. Con esto se puede seleccionar las diferentes opciones y visualizar datos específicos.

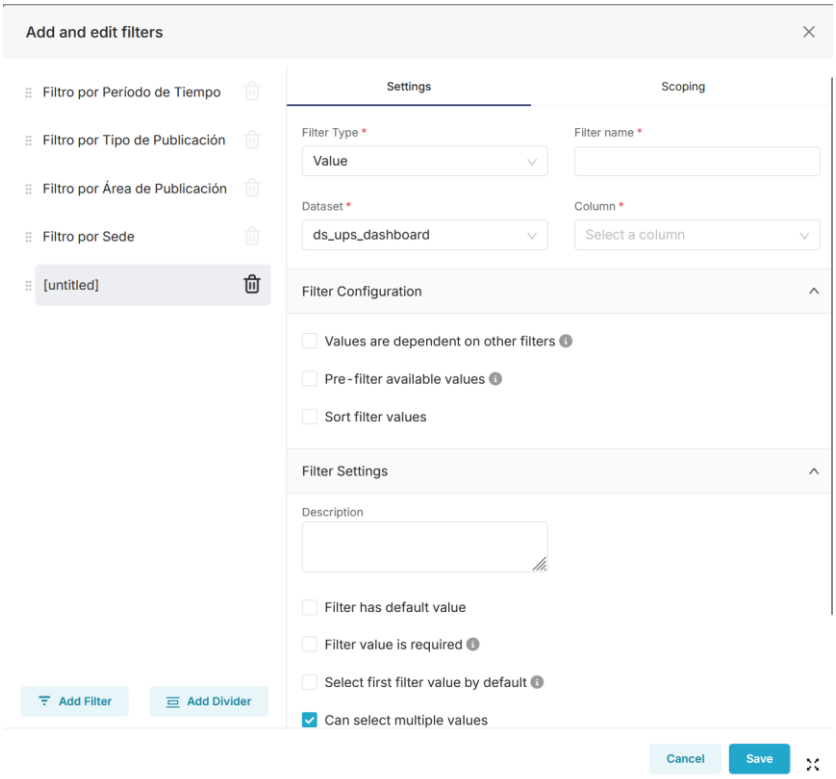

Fig. 8: cuadro de diálogo para la creación y edición de filtros.

2.4. Resultados Finales:

Al finalizar la configuración y diseño de las tablas y el dashboard, se muestra 3 gráficos que contienen los 3 parámetros principales solicitados por el reto, mostrado en la fig. 9:

- Análisis de Publicaciones por año.
- Análisis de Publicaciones por países asociados.
- Análisis de Publicaciones por Área de Conocimiento.

Se visualiza también los filtros disponibles para obtener información específica sobre un área, un año o tipo de publicación para los resultados obtenidos, mostrados en la fig. 10, considerando que es posible crear nuevos filtros.

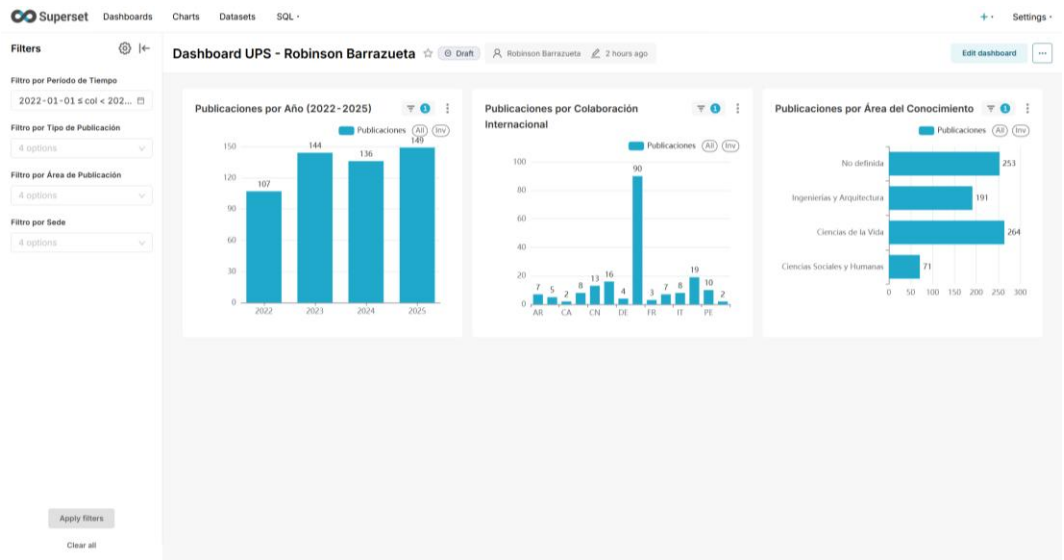

Fig. 9: Captura de pantalla mostrando el dashboard generado en Apache Superset.

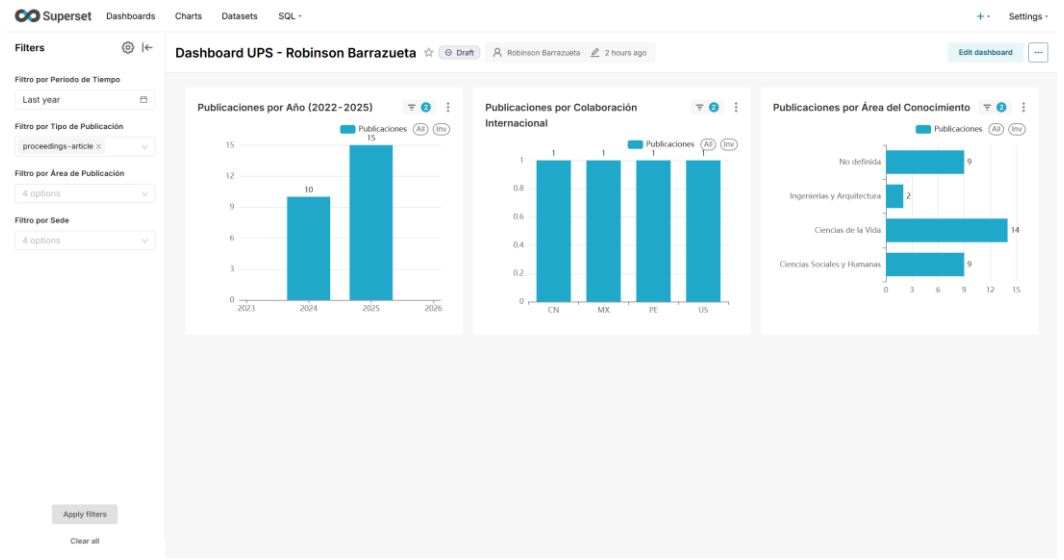

Fig. 10: Dashboard en Superset con filtros de tiempo, tipo/área de publicación y sede.

Con esto se ha cumplido el objetivo del reto, considerando que existe trabajo futuro por realizar y mejoras a implementar en el código mostrado y al dashboard en Superset, de así desearlo.