

Prueba Técnica: Data Governance Developer

Gladys Fernanda Guerrero Azuara

Parte 1: Configuración del Entorno en Dataplex

1. Se crea el Lake

ID del lake

stackoverflow-lake

Estado

✓

Lake activo

Zonas

Detalles

Permisos

Acciones

+ Agregar zona

⊖ Borrar zona

Filtro

Filtrar instancias

<input type="checkbox"/>	Nombre visible <div>↑</div>	Tipo	Estado	Elementos que requieren una acción	Recursos	Ubicaciones de los datos	Última modificación
<input type="checkbox"/>	so-curated-zone	Zona seleccionada	<div>✓</div> Activo	-	1	Regional (us-central1)	26 de septiembre de 2

Nota: antes de comenzar a trabajar se habilitan las APIs necesarias.

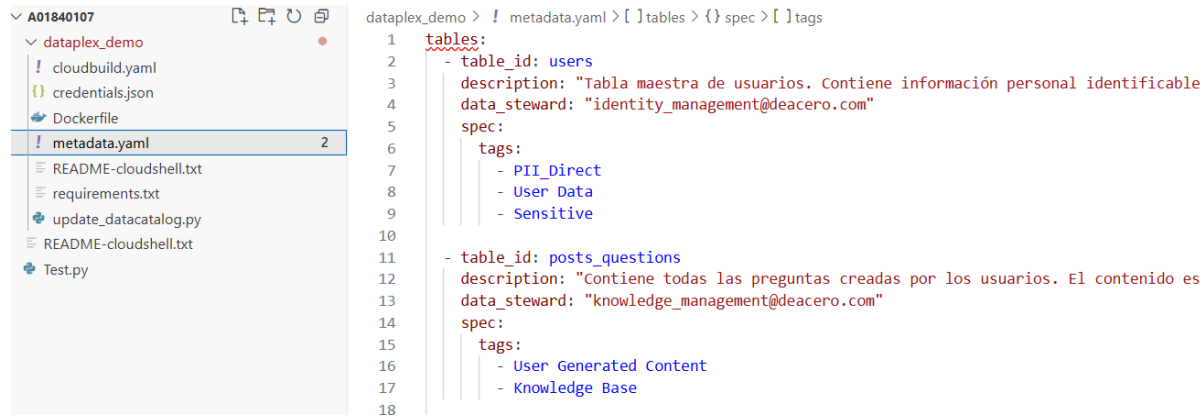
2. Se asocia la zona a Bigquery y se realiza la consulta (3 tablas de los assets)

ID de elemento	so-bq-asset
Tipo	Conjunto de datos de BigQuery
Estado	✓ Activo
Estado de seguridad	✓ Listo
Estado de la detección	✓ Programada calculado el 29 de septiembre de 2025, 1:00:29 a.m. UTC-6
Estado del recurso	Listo

Consulta sin título	Ejecutar	Guardar	Descargar	Compartir	Programa	Abrir en	Más
<pre>1 -- Tabla de usuarios 2 SELECT * 3 FROM `bigquery-public-data.stackoverflow.users` 4 LIMIT 10; 5 6 -- Tabla de preguntas 7 SELECT * 8 FROM `bigquery-public-data.stackoverflow.posts_questions` 9 LIMIT 10; 10 11 -- Tabla de respuestas 12 SELECT * 13 FROM `bigquery-public-data.stackoverflow.posts_answers` 14 LIMIT 10;</pre>							
✓ Se completó la consulta							
Todos los resultados							
Tiempo transcurrido		Declaraciones procesadas		Estado del trabajo			
2 s		3		✓ SUCCESS			
Estado	Hora de finalización	SQL		Etapas finalizadas	Bytes p	Acción	
✓	1:05 a.m. [2:1]	SELECT *		2	3.14 GE	Ver resultados	
✓	1:05 a.m. [7:1]	SELECT *		2	37.17 C	Ver resultados	
✓	1:05 a.m. [12:1]	SELECT *		2	28.62 C	Ver resultados	

Parte 2: Catalogación Automatizada con YAML

1. Crear el archivo YAML, que contiene los metadatos de las 3 tablas



```
dataplex_demo > ! metadata.yaml > [ ] tables > { } spec > [ ] tags
1 tables:
2   - table_id: users
3     description: "Tabla maestra de usuarios. Contiene información personal identificable"
4     data_steward: "identity_management@deacero.com"
5     spec:
6       tags:
7         - PII_Direct
8         - User Data
9         - Sensitive
10
11   - table_id: posts_questions
12     description: "Contiene todas las preguntas creadas por los usuarios. El contenido es"
13     data_steward: "knowledge_management@deacero.com"
14     spec:
15       tags:
16         - User Generated Content
17         - Knowledge Base
18
```

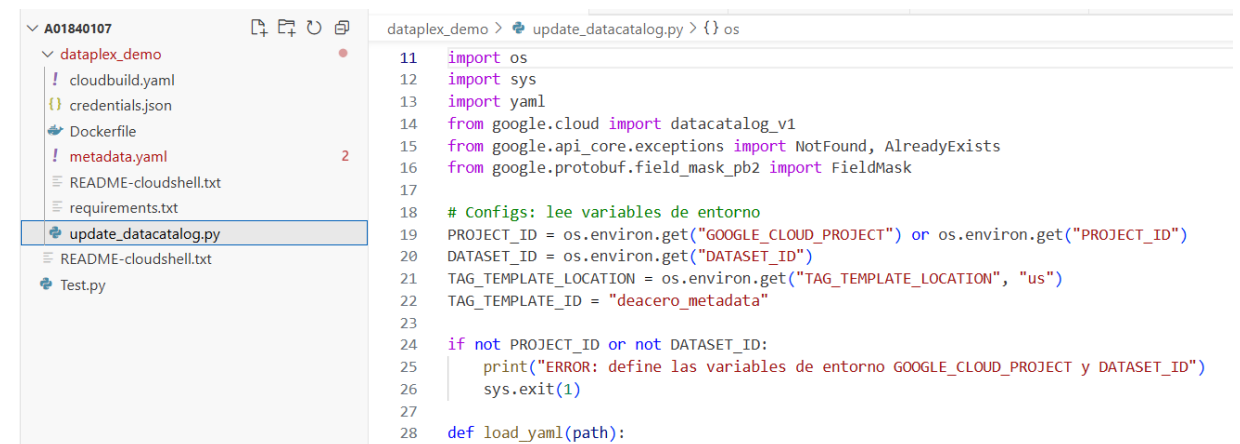
2. Se creó un script llamado update_datacatalog.py que hace lo siguiente:

Leer el archivo YAML.

Conectar con Data Catalog usando la biblioteca de cliente de Google Cloud (google-cloud-datacatalog).

Actualizar las descripciones de las tablas.


Aplicar los tags definidos en el YAML a cada tabla.



```
dataplex_demo > update_datacatalog.py > { } os
11 import os
12 import sys
13 import yaml
14 from google.cloud import datacatalog_v1
15 from google.api_core.exceptions import NotFound, AlreadyExists
16 from google.protobuf.field_mask_pb2 import FieldMask
17
18 # Configs: lee variables de entorno
19 PROJECT_ID = os.environ.get("GOOGLE_CLOUD_PROJECT") or os.environ.get("PROJECT_ID")
20 DATASET_ID = os.environ.get("DATASET_ID")
21 TAG_TEMPLATE_LOCATION = os.environ.get("TAG_TEMPLATE_LOCATION", "us")
22 TAG_TEMPLATE_ID = "deacero_metadata"
23
24 if not PROJECT_ID or not DATASET_ID:
25     print("ERROR: define las variables de entorno GOOGLE_CLOUD_PROJECT y DATASET_ID")
26     sys.exit(1)
27
28 def load_yaml(path):
```

Parte 3: Configuración de Aspect Types en Dataplex

1. Definir el Aspect Type y aplicarlo a las tablas desde Dataplex

data governance aspect 

Nombre visible	data governance aspect	Ubicación	us (múltiples regiones en Estados Unidos)
ID de tipo de aspecto	data-governance-aspect	Etiquetas	-
Descripción	-	Fecha de creación	28 sept 2025 10:07:38
ID de proyecto	governance-stackoverflow-demo	Última modificación	28 sept 2025 10:29:51

Plantilla

Entradas de ejemplo

Filtro

Ingresar el nombre o el valor de la propiedad

Nombre	Tipo	Descripción	Valores de enum/texto	Presencia	Obsoleta
Owner	Texto	Owner (email) para gobernanza	-	Opcional	No
Freshness	Enum	Frecuencia de actualización	daily, weekly y monthly	Opcional	No

Plantilla	Entradas de ejemplo	
Nombre de la entrada	Descripción	Fecha de creación
users	-	26 de septiembre de 2025
posts_answers	-	26 de septiembre de 2025
posts_questions	-	26 de septiembre de 2025

Parte 4: Políticas de Enmascaramiento

1. Se creo una copia de las tablas del data set para poder editar los esquemas y proteger las columnas sensibles.

Metadatos

ID de taxonomía de la etiqueta de política	6780339236347713080
Fecha de creación	28 sept 2025 11:33:41
Modificada	28 sept 2025 11:33:41
ID de proyecto	governance-stackoverflow-demo
Nombre visible del proyecto	governance-stackoverflow-demo
Ubicación	us (múltiples regiones en Estados Unidos)

Etiquetas de política

Las etiquetas de política son etiquetas con políticas de acceso que se pueden aplicar a los subrecursos, por ejemplo, a las columnas de BigQuery.

Administrar políticas de datos

<input checked="" type="checkbox"/>	Nombre ↑	ID	Reglas de enmascaramiento de datos	Descripción
<input checked="" type="checkbox"/>	 pii_user_info	5100175554015200424 	Hash (SHA256)	display_nam

2. Se realizó la consulta para verificar que efectivamente esas columnas estuvieran protegidas

```
SELECT display_name, location
FROM `bigquery-public-data.stackoverflow-demo.users`
LIMIT 10;
```

Parte 5: Implementación de una Regla de Calidad de Datos

1. Crear la Regla de Calidad en Dataplex en la sección de análisis de calidad de datos

<input type="checkbox"/>	Nombre visible	Etiquetas	Tipo	Estado	Nombre de la tabla	Alcance	
<input type="checkbox"/>	dq_posts_questions_owner_check		Calidad de los datos	 1 dimensión 1 regla con errores	posts_questions	Datos completos	

2. Se ejecuta el Job de calidad y se verifican los resultados

Resultados más recientes del estado de la calidad de los datos

Completeness	 Con errores
--------------	---

Parte 6: Dockerización y CI/CD Pipeline

1. Se crea un script con los requirements

```
update_datacatalog.py  Test.py  ! metadata.yaml 2  requirements.txt  x  README-cloudshe  ◆  [
dataplex_demo > requirements.txt
1  google-cloud-bigquery
2  google-cloud-datacatalog
3  pyyaml
```

2. Se crea un Dockerfile en donde se copian los archivos del proyecto al contenedor , se instalan las dependencias y se configuran las credenciales de GCP. Se construye y ejecuta el contenedor (bash).

```
Test.py  ! metadata.yaml 2  requirements.txt  README-cloudshell.txt  Dockerfile  x
dataplex_demo > Dockerfile
2  FROM python:3.11-slim
3
4  # Directorio de trabajo dentro del contenedor
5  WORKDIR /app
6
7  # Copiar archivos del proyecto
8  COPY requirements.txt .
9  COPY update_datacatalog.py .
10 COPY credentials.json .
11 COPY metadata.yaml .
12
13 # Definir variables de entorno necesarias
14 ENV GOOGLE_CLOUD_PROJECT=governance-stackoverflow-demo
15 ENV DATASET_ID=so_dataset_2
16 ENV GOOGLE_APPLICATION_CREDENTIALS=/app/credentials.json
17
18 # Instalar dependencias
19 RUN pip install --no-cache-dir -r requirements.txt
20
21 # Comando por defecto al iniciar el contenedor
22 CMD ["python", "update_datacatalog.py", "metadata.yaml"]
```

3. Crea un archivo cloudbuild.yaml (en CloudBuild) para automatizar la construcción y despliegue del contenedor, se ejecuta.

```
! metadata.yaml 2  requirements.txt  README-cloudshell.txt  Dockerfile  ! cloudbuild.yaml X  ◆  □  ..

dataplex_demo > ! cloudbuild.yaml > [ ] steps > [ ] env
1  steps:
2      # Paso 1: Construir la imagen Docker
3      - name: 'gcr.io/cloud-builders/docker'
4        args: ['build', '-t', 'gcr.io/$PROJECT_ID/dataplex-catalog:latest', '.']
5
6      # Paso 2: Subir la imagen al Container Registry
7      - name: 'gcr.io/cloud-builders/docker'
8        args: ['push', 'gcr.io/$PROJECT_ID/dataplex-catalog:latest']
9
10     # Paso 3: Ejecutar el script Python dentro de la imagen
11     - name: 'gcr.io/$PROJECT_ID/dataplex-catalog:latest'
12       entrypoint: 'python'
13       args: ['update_datacatalog.py', 'metadata.yaml']
14       env:
15         - 'GOOGLE_CLOUD_PROJECT=$PROJECT_ID'
16         - 'DATASET_ID=so_dataset_2'
17         - 'GOOGLE_APPLICATION_CREDENTIALS=/app/credentials.json'
18
```


Parte 7: Análisis de Gobernanza

- El linaje de datos describe cómo la información se mueve y se transforma dentro de un sistema. En este proyecto, la tabla **users** almacena los datos de los usuarios, **posts_questions** registra las preguntas que cada usuario genera, y **posts_answers** contiene las respuestas vinculadas tanto a las preguntas como a los usuarios correspondientes. Visualizar este linaje permite a un Community Manager entender la relación entre usuarios y su actividad, identificar problemas de integridad de datos, y tomar decisiones informadas sobre moderación, contenido y estrategia de comunidad. Garantiza transparencia, trazabilidad y confiabilidad en el manejo de la información.