

Trabajo 1 – Almacenamiento y Recuperación de Información

Grupo 6

Javier Patiño Serna
Marcela Díaz Cordero
Simón Madrid Álvarez

Universidad EAFIT
Maestría en ciencia de datos y analítica
Edwin Montoya

Medellín, Colombia
Septiembre 2022

Trabajo 1: Almacenamiento y recuperación de datos

A continuación, se describen los pasos realizados en el trabajo con sus respectivos pantallazos y códigos

1. Fuentes de datos

Se eligieron dos fuentes de datos (dos datasets):

- a. El primer dataset se obtuvo en Kaggle

Link:

[https://www.kaggle.com/datasets/sevgisarac/temperature-change?select=Environment Temperature change E All Data NOFLAG.csv](https://www.kaggle.com/datasets/sevgisarac/temperature-change?select=Environment+Temperature+change+E+All+Data+NOFLAG.csv)

Descripción: Corresponde a la variación en °C a lo largo de los años (Desde 1961 hasta 2019)

- b. El segundo dataset se obtuvo en la página pronosticosyalertas.gov.co

Link:

<http://www.pronosticosyalertas.gov.co/datos-abiertos-ideam>

Descripción: corresponde al pronóstico del tiempo en Colombia (discriminado por departamentos y municipios) en la semana del 7 de septiembre al 11 de septiembre.

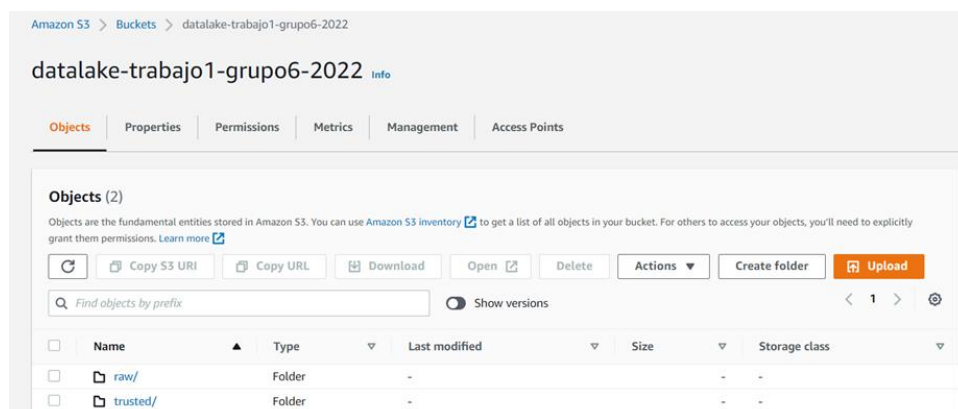
2. S3: Ingesta

La ingesta a la zona raw se hizo manual en un bucket S3 llamado "datalake-trabajo1-grupo6-2022"

Link:

<https://s3.console.aws.amazon.com/s3/buckets/datalake-trabajo1-grupo6-2022?region=us-east-1&tab=objects>

En este bucket se definieron 2 zonas:



3. Glue

Se ejecutaron dos Crawler (mundo y Colombia). Se creo la base de datos “cambioclimatico” en la cual se agregaron las 2 tablas resultantes de los crawler.

Luego de su catalogación se actualizó el esquema de ambas tablas

ANTES: Nombres de columnas propios del dataset

DESPUES: Actualización de esquemas

Schema (18)

View and manage the table schema.

Filter schemas

<input type="checkbox"/>	#	Column name	Data type
<input type="checkbox"/>	1	cod_div	bigint
<input type="checkbox"/>	2	latitud	double
<input type="checkbox"/>	3	longitud	double
<input type="checkbox"/>	4	region	string
<input type="checkbox"/>	5	departamento	string
<input type="checkbox"/>	6	municipio	string
<input type="checkbox"/>	7	fecha	string
<input type="checkbox"/>	8	hora	string
<input type="checkbox"/>	9	temperatura	double
<input type="checkbox"/>	10	velocidad del viento	double
<input type="checkbox"/>	11	direccion del viento	double
<input type="checkbox"/>	12	presion	double
<input type="checkbox"/>	13	punto de rocío	double
<input type="checkbox"/>	14	cobertura total nubosa	double
<input type="checkbox"/>	15	precipitacion (mm/h)	double
<input type="checkbox"/>	16	probabilidad de tormenta	double
<input type="checkbox"/>	17	humedad	double
<input type="checkbox"/>	18	pronostico	string

Schema (18)

View and manage the table schema.

Filter schemas

<input type="checkbox"/>	#	Column name	Data type
<input type="checkbox"/>	1	cod_div	bigint
<input type="checkbox"/>	2	latitud	double
<input type="checkbox"/>	3	longitud	double
<input type="checkbox"/>	4	region	string
<input type="checkbox"/>	5	departamento	string
<input type="checkbox"/>	6	municipio	string
<input type="checkbox"/>	7	fecha	string
<input type="checkbox"/>	8	hora	string
<input type="checkbox"/>	9	temperatura	double
<input type="checkbox"/>	10	velocidad del viento	double
<input type="checkbox"/>	11	direccion del viento	double
<input type="checkbox"/>	12	presion	double
<input type="checkbox"/>	13	punto de rocío	double
<input type="checkbox"/>	14	cobertura total nubosa	double
<input type="checkbox"/>	15	precipitacion	double
<input type="checkbox"/>	16	probabilidad de tormenta	double
<input type="checkbox"/>	17	humedad	double
<input type="checkbox"/>	18	pronostico	string

Quedaron los siguientes crawler:

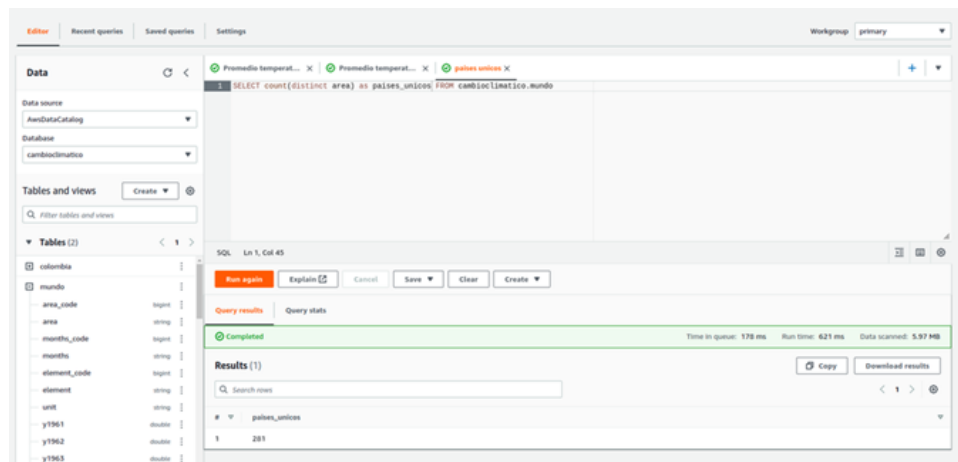
The screenshot shows the AWS Glue console interface. On the left is a navigation menu with options like Data Catalog, Databases, Tables, Stream schema registries, Schemas, Connections, Crawlers, Clusters, Catalog settings, Data Integration and ETL, AWS Glue Studio, Jobs, Interactive Sessions, Notebooks, Data classification tools, Sensitive data detection, Record matching, Triggers, Workflows, Blueprints, Security configurations, and Legacy pages. The main panel displays the 'cambioclimatico' database. It includes a 'Database properties' section with fields for Name, Description, Location, and Created on (UTC). Below this is a 'Tables (3)' section with a table listing three tables: 'category_prueba', 'colombia', and 'mundo'. Each table entry shows its name, database, location, classification, deprecated status, and a 'View data' link.

El crawler category _prueba corresponde a una base de datos creada desde athena con el siguiente código

The screenshot shows the Athena SQL editor interface. On the left is a sidebar with 'Data' and 'Tables and views' sections. The main area contains a SQL query. The query starts with a 'CREATE TABLE' statement for 'category_prueba' with a 'TEXTFILE' format and a 'LOCATION' pointing to an S3 bucket. It then includes a 'SELECT' statement that joins data from the 'cambioclimatico.colombia' table with a subquery that calculates the average temperature by department and date.

4. Consultas en Athena.

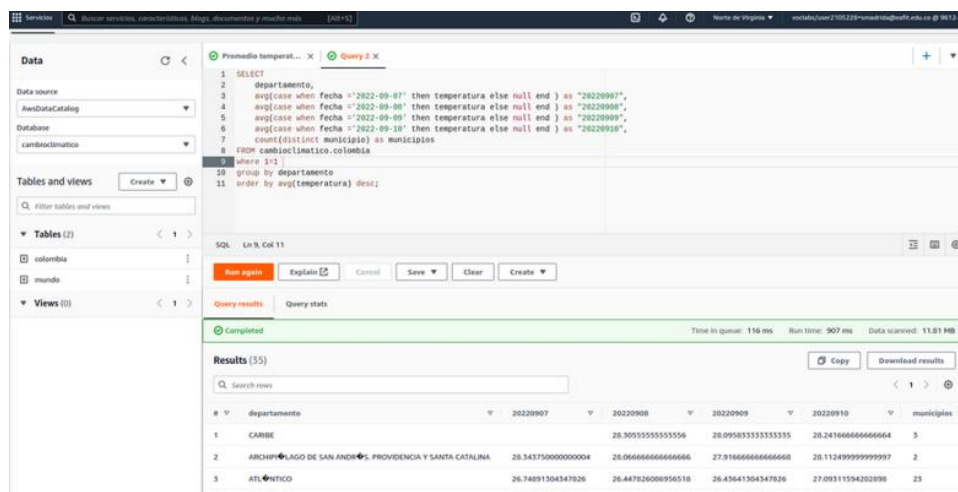
Se comprobó que las tablas mundo y colombia tienen datos:



Se ejecutaron las siguientes consultas en colombia:

Esta primera consulta permite identificar el promedio de temperatura por departamento para la semana de 7 al 10 de septiembre. Adicional, permite identificar la cantidad de municipios para cada departamento:

```
SELECT
    departamento,
    avg(case when fecha = '2022-09-07' then temperatura else null end ) as "20220907",
    avg(case when fecha = '2022-09-08' then temperatura else null end ) as "20220908",
    avg(case when fecha = '2022-09-09' then temperatura else null end ) as "20220909",
    avg(case when fecha = '2022-09-10' then temperatura else null end ) as "20220910",
    count(distinct municipio) as municipios
FROM cambioclimatico.colombia
where 1=1
group by departamento
order by avg(temperatura) desc;
```



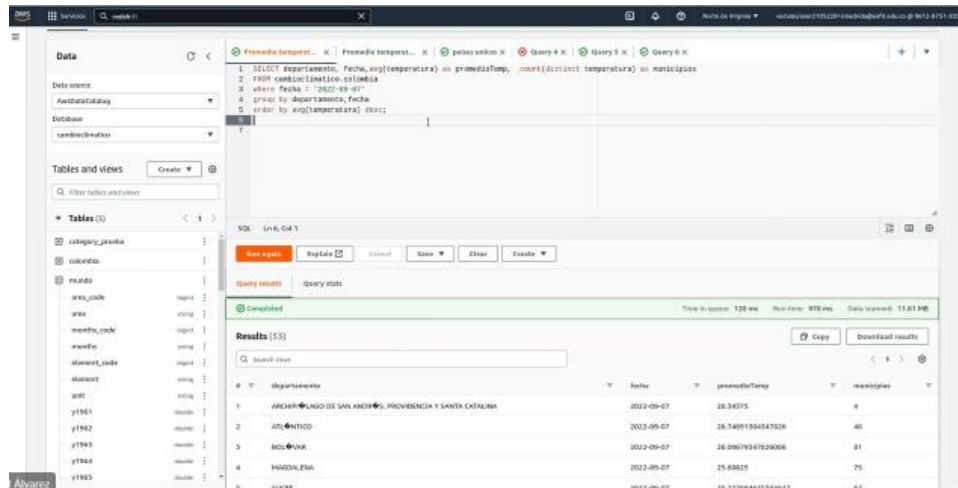
El segundo query ejecutado permite identificar el promedio para el día 7 de septiembre:

```
SELECT
    departamento,
```

```

fecha,
avg(temperatura) as promedioTemp,
count(distinct temperatura) as municipios
FROM cambioclimatico.colombia
where fecha = '2022-09-07'
group by departamento,fecha
order by avg(temperatura) desc;

```



The screenshot shows the Amazon Redshift console interface. On the left, there's a sidebar with 'Data' and 'Tables and views' sections. The 'Data' section shows the 'cambioclimatico' database and the 'colombia' table. The 'Tables and views' section shows a list of tables including 'category_paqueta', 'colombia', 'mundo', 'area_code', 'area', 'month_code', 'month', 'element_code', 'element', 'unit', 'unit_code', 'y1961', 'y1962', 'y1963', 'y1964', and 'y1965'. The main area displays a SQL query and its results. The query is:
 1 SELECT departamento, fecha, avg(temperatura) as promedioTemp, count(distinct temperatura) as municipios
 2 FROM cambioclimatico.colombia
 3 where fecha = '2022-09-07'
 4 group by departamento, fecha
 5 order by avg(temperatura) desc;
 The results are shown in a table with 5 columns: departamento, fecha, promedioTemp, and municipios. The results are sorted by promedioTemp in descending order.

#	departamento	fecha	promedioTemp	municipios
1	ARCHIPELAGO DE SAN ANDRÉS, PROVIDENCIA Y SANTA CATALINA	2022-09-07	28.34375	4
2	ATLÁNTICO	2022-09-07	26.74091334547628	40
3	BOLÍVAR	2022-09-07	26.08679347529056	81
4	MAGDALENA	2022-09-07	25.83625	75
5	SUCRE	2022-09-07	25.122288125338614	49

5. Redshift:

Luego de crear el cluster de redshift se cosntruyó una nueva tabla llamada colombia_rsf con el siguiente código:

```

CREATE table colombia_rsf(
  cod_div varchar(100),
  latitud decimal(30,20),
  longitud varchar(200),
  region varchar(50),
  departamento varchar(100),
  municipio varchar(50),
  fecha varchar(50),
  hora varchar(50),
  temperatura decimal(30,20),
  velocidad_viento decimal(30,20),
  direccion_viento decimal(30,20),
  presion decimal(30,20),
  punto_rocio decimal(30,20),
  cobertura decimal(30,20),
  precipitacion decimal(30,20),
  prob_tormenta decimal(30,20),
  humedad decimal(30,20),
  pronostico varchar(50)
);

```

Y se agregaron datos a la tabla desde el bucket S3:

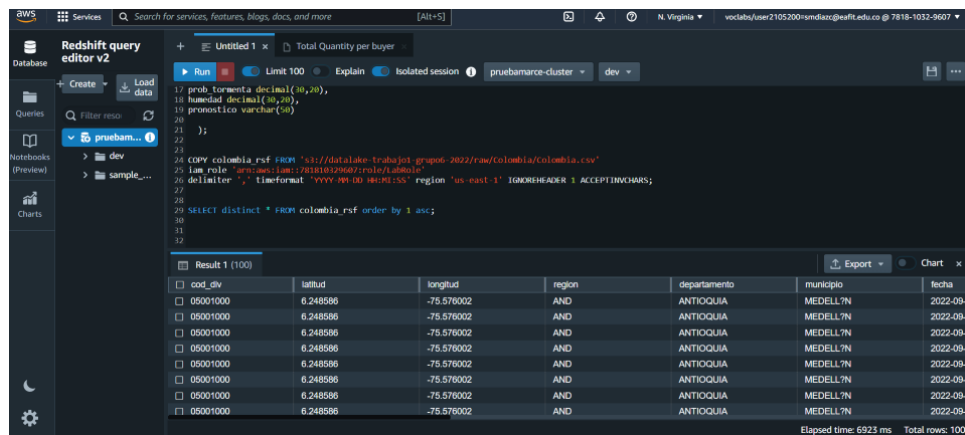
```

COPY colombia_rsf
FROM 's3://datalake-trabajo1-grupo6-2022/raw/Colombia/Colombia.csv'
iam_role 'arn:aws:iam::781810329607:role/LabRole'

```

```
delimiter ',' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-east-1' IGNOREHEADER 1
ACCEPTINVCHARS;
```

```
SELECT distinct * FROM colombia_rsf order by 1 asc;
```



Se ejecutan algunas consultas en redshif con los siguientes códigos:

```
SELECT DISTINCT region, departamento, round(avg (humedad),3) as prom_humedad
from colombia_rsf
group by region, departamento
```

```
31:
32: SELECT DISTINCT region, departamento, round(avg (humedad),3) as prom_humedad
33: from colombia_rsf
34: group by region, departamento
--
```

Result 1 (51)

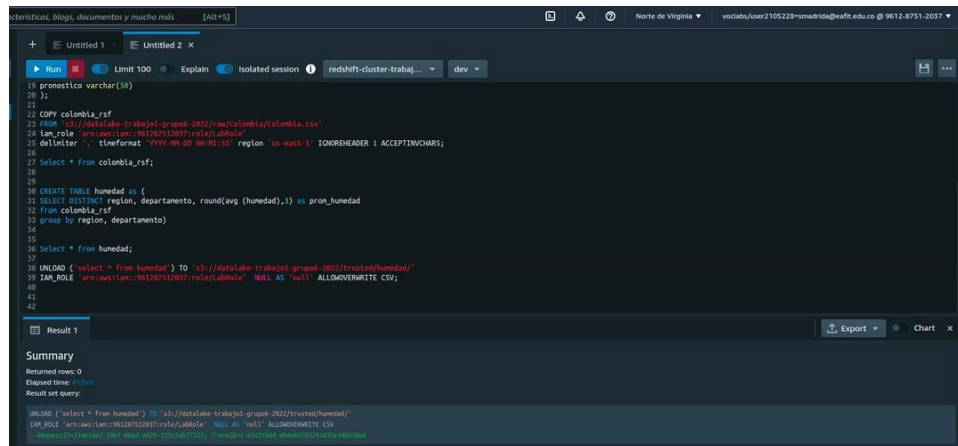
region	departamento	prom_humedad
AND	NORTE DE SANTANDER	83.301
AND	CUNDINAMARCA	79.956
AND	CESAR	88.822
AND	SANTANDER	86.111
ORI	CASANARE	79.167
AMA	CAQUET?	80.745
CAR	C?RDOBA	87.698
PAC	VALLE DEL CAUCA	94.09
AND	PUTUMAYO	85.347
PAC	NARI?O	85.864
PAC	CAUCA	91.942
ORI	GUAVIARE	84.949
---	---	---

Se creó la tabla humedad y se almacenó el resultado en S3 :

```
CREATE TABLE humedad as (
SELECT DISTINCT region, departamento, round(avg (humedad),3) as prom_humedad
from colombia_rsf
group by region, departamento)
```

```
SELECT * FROM humedad
```

```
UNLOAD ('select * from humedad') TO 's3://datalake-trabajo1-grupo6-2022/trusted/humedad/'  
IAM_ROLE 'arn:aws:iam::961287512037:role/LabRole' NULL AS 'null' ALLOWOVERWRITE CSV;
```



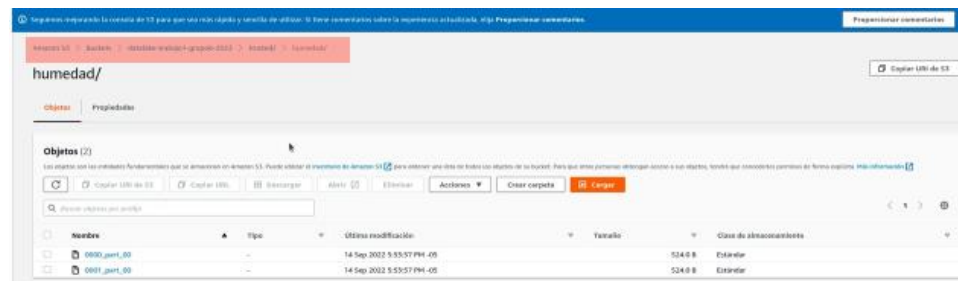
The screenshot shows a Redshift console window with a SQL query editor and a results pane. The query in the editor is as follows:

```
19 pronostico varchar(50)  
20 ;  
21  
22 COPY colombia_rsf  
23 FROM 's3://datalake-trabajo1-grupo-2022/rw/colombia/colombia.csv'  
24 IAM_ROLE 'arn:aws:iam::961287512037:role/LabRole'  
25 delimiter ',' timeformat 'YYYY-MM-DD HH:MM:SS' region 'us-east-1' IGNOREHEADER 1 ACCEPTINVCHARS;  
26  
27 Select * from colombia_rsf;  
28  
29  
30 CREATE TABLE humedad as (  
31 SELECT DISTINCT region, departamento, round(avg(humedad),3) as pron_humedad  
32 from colombia_rsf  
33 group by region, departamento)  
34  
35  
36 select * from humedad;  
37  
38 UNLOAD ('select * from humedad') TO 's3://datalake-trabajo1-grupo-2022/trusted/humedad/'  
39 IAM_ROLE 'arn:aws:iam::961287512037:role/LabRole' NULL AS 'null' ALLOWOVERWRITE CSV;  
40  
41  
42
```

The results pane shows a summary of the query execution:

Summary
Returned rows: 0
Elapsed time: 0:10m
Result set query:
UNLOAD ('select * from humedad') TO 's3://datalake-trabajo1-grupo-2022/trusted/humedad/'
IAM_ROLE 'arn:aws:iam::961287512037:role/LabRole' NULL AS 'null' ALLOWOVERWRITE CSV

Los datos de humedad quedaron almacenados en el bucket S3 de la zona trusted (directorio humedad)



6. EMR

Hive

Se hicieron consultas desde hive a las tablas de glue de manera efectiva

The screenshot shows the Hive query interface. The query is as follows:

```
SELECT departamento, fecha, avg(temperatura) as promedioTemp, count(distinct temperatura) as municipios
FROM cambioclimatico.colombia
where fecha = '2022-09-07'
group by departamento, fecha
order by avg(temperatura) desc;
```

The results are displayed in a table with the following columns: departamento, fecha, promedioTemp, and municipios. The results are sorted by promedioTemp in descending order.

departamento	fecha	promedioTemp	municipios
1 ARIHPOLAGO DE SAN ANDRÉS, PROVIDENCIA Y SANTA CATALINA	2022-09-07	28.34375	4
2 ATLANTICO	2022-09-07	26.74891304347825	48
3 BOLIVAR	2022-09-07	26.006793478260864	81
4 MAGDALENA	2022-09-07	25.808249999999999	75
5 SUCRE	2022-09-07	25.72788461538462	62
6 CORDOBA	2022-09-07	25.726666666666667	64
7 VICHADA	2022-09-07	25.51875	27
8 VAUPES	2022-09-07	25.422916666666662	35
9 AMAZONAS	2022-09-07	25.367045454545456	50
10 GUANAJA	2022-09-07	25.084722222222226	47
11 LA GUAJIRA	2022-09-07	24.668333333333333	73
12 ARAUCA	2022-09-07	24.655357142857135	46
13 GUANAJARE	2022-09-07	24.309374999999996	25
14 CESAR	2022-09-07	24.022000000000002	95

Jupyterhub (pyspark)

Por último, se creó un bucket llamado notebooks-trabajo6-grupo1 donde quedaron almacenados los notebooks que se trabajaron desde spark.

The screenshot shows the AWS S3 console interface for the bucket 'notebooks-trabajo1-grupo6'. The 'Objects' tab is selected, showing a list of objects. The list contains one object named 'jupyter/' which is a folder.

Name	Type	Last modified	Size	Storage class
jupyter/	Folder	-	-	-

El link público para este bucket es:

<https://s3.console.aws.amazon.com/s3/buckets/notebooks-trabajo1-grupo6?region=us-east-1&tab=objects>

En estos notebooks se extrajo información desde s3 a jupyter hub, se procesó dicha información y se guardaron los resultados nuevamente en s3 en la zona trusted (Ver notebook llamado Trabajo1_grupo6.ipynb)

trusted/

Copy S3 URI

Objects Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix Show versions < 1 > Settings

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	humedad/	Folder	-	-	-
<input type="checkbox"/>	pysparkdaraframes/	Folder	-	-	-