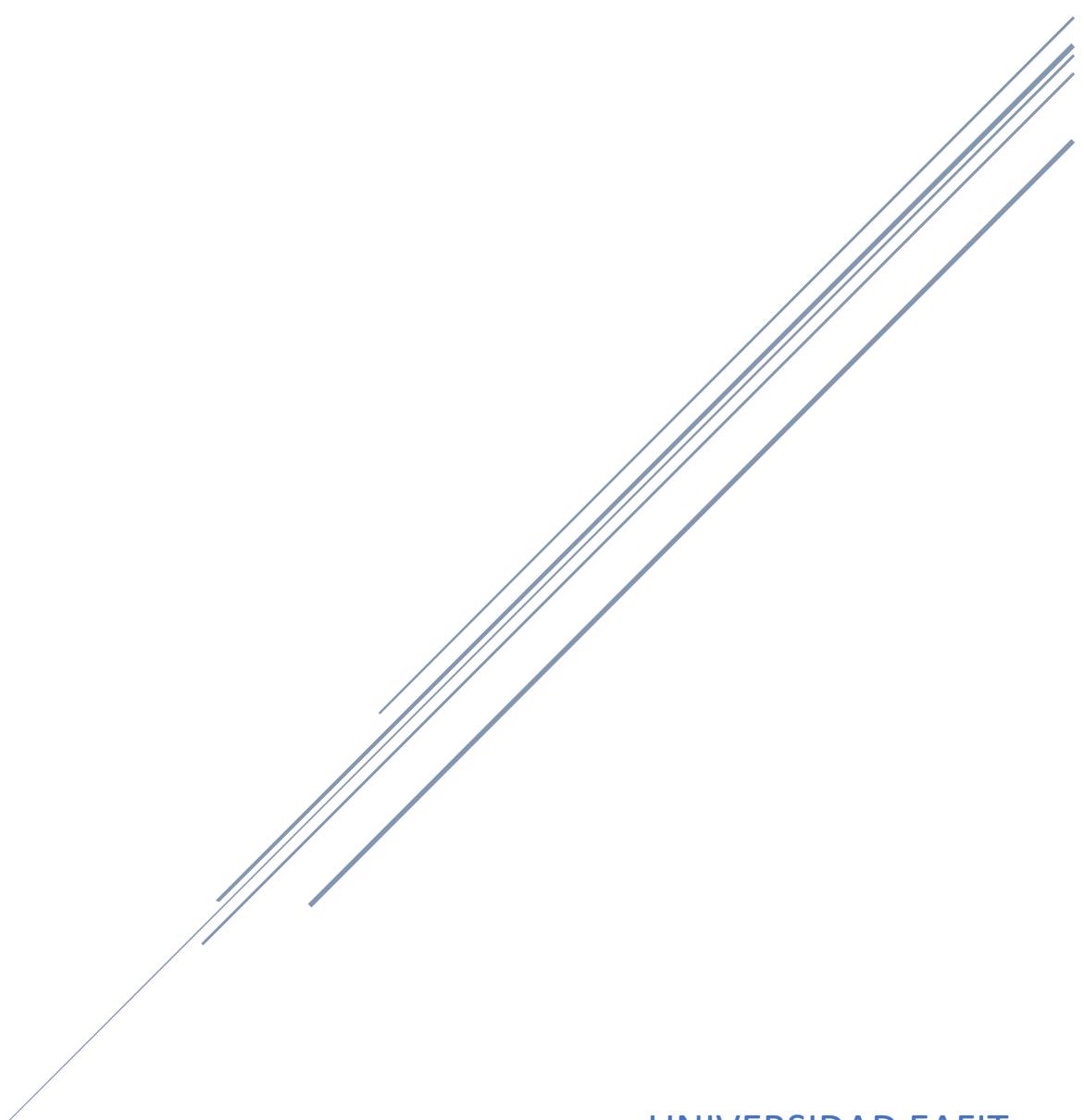


# PROYECTO FINAL

## SISTEMAS DISTRIBUIDOS



UNIVERSIDAD EAFIT  
2025

Automatización del proceso de Captura, Ingesta, Procesamiento y Salida de datos

Proyecto 3: Tópicos Especiales en Telemática

Autores:

Juan Pablo Rúa Cartagena

Santiago Sánchez Carvajal

Yasir Enrique Blandon Varela

Docente Académico:

Edwin Nelson Montoya Munera

UNIVERSIDAD EAFIT

Departamento de Informática y Sistemas

ST0263 Tópicos Especiales en Telemática Sistemas Distribuidos

Medellín – Antioquia

20/11/2025

# Contenido

IMPLEMENTACIÓN DE UNA ARQUITECTURA DE INGESTA, PROCESAMIENTO Y ANÁLISIS DE DATOS	3
1. INTRODUCCIÓN .....	3
2. OBJETIVOS DEL PROYECTO.....	4
2.1 Objetivo General .....	4
2.2 Objetivos Específicos.....	4
3. CONTEXTO GENERAL Y JUSTIFICACIÓN DEL PROYECTO .....	5
4. ARQUITECTURA GENERAL DEL SISTEMA.....	6
4.1 Captura e Ingesta de Datos.....	6
4.2 Procesamiento ETL.....	6
4.3 Análisis Avanzado.....	6
4.4 Servicios Utilizados.....	6
5. PROCESOS DE INGESTA AUTOMÁTICA DE DATOS .....	7
6. PROCESOS AUTOMÁTICOS – AWS GLUE.....	8
6.1 Job: descargaURL .....	8
6.2 Job: descargaAPI .....	8
6.3 Job: rds-s3 .....	8
7. PROCESAMIENTO ETL EN APACHE SPARK UTILIZANDO AMAZON EMR .....	9
8. AWS S3 .....	11
9. COMPONENTES DE BASE DE DATOS.....	15
9.1 Base de Datos Relacional AWS RDS .....	16
9.2 Base de Datos en AWS Glue Data Catalog .....	16
10. ANÁLISIS AVANZADO .....	17
11. CONSULTAS ANALÍTICAS MEDIANTE AMAZON ATHENA .....	18
12. EXPLICACIÓN DE SCRIPTS.....	19
12.1 datos_api.py.....	20
12.2 datos_url.py .....	20
12.3 rds_s3.py .....	20
12.4 analítica.py .....	20
13. GUÍA PASO A PASO (EJECUCIÓN).....	21
14. CONCLUSIONES .....	23
REFERENCIAS.....	24

# IMPLEMENTACIÓN DE UNA ARQUITECTURA DE INGESTA, PROCESAMIENTO Y ANÁLISIS DE DATOS

## 1. INTRODUCCIÓN

El estudio, tratamiento y análisis de grandes volúmenes de datos se ha convertido en uno de los pilares fundamentales de los sistemas modernos de ingeniería, particularmente en áreas como salud pública, analítica institucional y toma de decisiones basada en evidencia. En este contexto, las arquitecturas de datos distribuidos representan un componente esencial para la administración eficiente de información proveniente de múltiples fuentes externas e internas, permitiendo automatizar procesos de ingesta, integración, transformación y análisis, garantizando a su vez escalabilidad, resiliencia y disponibilidad continua. El presente documento expone el diseño, la construcción y la implementación integral de un sistema distribuido orientado al procesamiento automatizado de datos relacionados con la pandemia de COVID-19 en Colombia. Su propósito fundamental es reproducir un entorno cercano a los que se encuentran en la industria, en donde flujos complejos de información deben ser administrados de manera robusta, sistemática y replicable.

Para ello, el proyecto se fundamenta en una infraestructura distribuida basada en los servicios nativos de Amazon Web Services (AWS), que permiten no solo la automatización completa de los procesos de ingesta de datos, sino también la ejecución de tareas ETL (Extract, Transform, Load) a gran escala en clústeres administrados (EMR). La información proviene de dos grandes tipos de fuentes: datos abiertos del Ministerio de Salud de Colombia, ofrecidos tanto vía archivos descargables como mediante API, y datos provenientes de un motor de base de datos relacional PostgreSQL desplegado en Amazon RDS. Estos conjuntos de datos heterogéneos se integran en un data lake estructurado, permitiendo mantener una organización coherente del ciclo de vida de la información.

La totalidad del sistema fue diseñada bajo el criterio de ejecución automatizada, lo cual implica que, una vez desplegada la infraestructura requerida, los procesos operan sin intervención humana, logrando reproducir un flujo realista de ingeniería de datos profesional. Se emplearon herramientas como AWS Glue, Amazon EMR, Amazon Athena y Amazon S3, que en conjunto conforman una solución coherente para ingestión, catalogación, procesamiento y consulta de datos. Asimismo, se desarrollaron scripts en Python de forma modular, lo que facilita su administración y su ejecución tanto dentro de los servicios AWS como localmente en entornos de desarrollo.

## 2. OBJETIVOS DEL PROYECTO

### 2.1 Objetivo General

Diseñar e implementar una arquitectura distribuida que permita ingestión automática, almacenamiento estructurado, procesamiento ETL y análisis avanzado de datos de COVID-19 en Colombia utilizando servicios administrados de AWS.

### 2.2 Objetivos Específicos

- Capturar datos de COVID-19 desde fuentes externas (API REST y archivos descargables).
- Ingestar datos relacionales complementarios desde una base de datos RDS (PostgreSQL).
- Almacenar todos los datos en un data lake en Amazon S3 en su respectiva zona RAW.
- Ejecutar procesos ETL y analítica avanzada utilizando Spark sobre AWS EMR de forma automatizada.
- Transformar y enriquecer los datos para almacenarlos en la zona TRUSTED y REFINED.
- Facilitar consultas analíticas mediante Amazon Athena.
- Proveer resultados finales almacenados en S3 para consumo por API Gateway o herramientas analíticas.

### 3. CONTEXTO GENERAL Y JUSTIFICACIÓN DEL PROYECTO

Durante el desarrollo del presente proyecto se examinaron los distintos procesos que conforman el ciclo de vida de un sistema analítico: captura, ingesta, almacenamiento, procesamiento y entrega de resultados. Aunque en los ejercicios prácticos se trabajó con cantidades limitadas de datos y ejemplos simplificados, la realidad empresarial exige el diseño de soluciones capaces de manejar volúmenes amplios, fuentes diversas y procesos no manuales. Por ello, se definió como objetivo construir un prototipo funcional que reprodujera un flujo de ingeniería de datos real, específicamente orientado al manejo de información epidemiológica.

El caso seleccionado responde a la importancia histórica y analítica de los datos referentes a la pandemia de COVID-19 en Colombia. El Ministerio de Salud ofrece datos públicos actualizados sobre la evolución del virus, disponibles tanto en formato de archivo como por medio de API.

A su vez, existe información complementaria que puede ser almacenada en motores de bases de datos relacionales, permitiendo enriquecer los análisis y ofrecer una visión integral sobre el comportamiento epidemiológico a nivel departamental y municipal. De esta manera, la diversidad de fuentes permite replicar un escenario genuino de integración de datos heterogéneos.

La elección de AWS responde a su capacidad de ofrecer servicios administrados que facilitan la orquestación de procesos sin necesidad de aprovisionar infraestructura física. A través de Amazon S3 es posible estructurar un data lake altamente escalable; mediante AWS Glue se administra la ingesta automatizada, la catalogación y la ejecución de jobs ETL; por medio de Amazon RDS se emula un sistema transaccional; y con Amazon EMR se realiza el procesamiento distribuido de grandes volúmenes de información.

Asimismo, Amazon Athena permite realizar consultas SQL directamente sobre los datos almacenados sin necesidad de servidores adicionales. En conjunto, estos servicios posibilitan la construcción de un entorno distribuido robusto, modular y reproducible para fines académicos y profesionales.

## **4. ARQUITECTURA GENERAL DEL SISTEMA**

La arquitectura implementada consta de los siguientes componentes:

### **4.1 Captura e Ingesta de Datos**

- Fuente 1: API pública del Ministerio de Salud Endpoint JSON de datos abiertos.
- Fuente 2: Archivo CSV descargable desde Datos.gov
- Fuente 3: Base de datos relacional PostgreSQL en AWS RDS

### **4.2 Procesamiento ETL**

- Clúster emr-proyecto3 configurado con Apache Spark.
- Jobs de Spark ejecutados como Steps automáticos.
- Transformaciones y enriquecimiento almacenado en la zona TRUSTED.

### **4.3 Análisis Avanzado**

- Script analítica.py para generación de indicadores departamentales.
- Resultados almacenados en la zona REFINED.
- Consultas mediante Athena.
- Disponibilidad opcional vía API Gateway.

### **4.4 Servicios Utilizados**

- Amazon S3
- AWS Glue (ETL Jobs, Crawlers, Data Catalog)
- Amazon RDS / Aurora PostgreSQL
- Amazon EMR
- Amazon Athena
- Amazon API Gateway (opcional)
- Amazon IAM para roles y permisos

## 5. PROCESOS DE INGESTA AUTOMÁTICA DE DATOS

Con el objetivo de garantizar que la captura de datos no dependa de la intervención humana, se desarrollaron tres scripts en Python destinados a la ingesta automática de las fuentes del sistema. Estos scripts fueron almacenados tanto en un bucket especial de AWS Glue como en un repositorio local de trabajo, asegurando la capacidad de edición, despliegue y actualización.

El script `datos_api.py` establece una conexión directa con el endpoint JSON del Ministerio de Salud utilizando la librería Requests. Desde allí, descarga los datos mediante streaming y sube el contenido sin almacenarlo en disco intermedio, utilizando el cliente de Amazon S3 provisto por Boto3. Este método resulta eficiente al manejar grandes volúmenes, ya que reduce el uso de memoria y garantiza que los datos lleguen íntegramente a la zona RAW.

Por otra parte, el script `datos_url.py` realiza una tarea equivalente, pero orientada a la descarga del archivo CSV desde la URL pública. En este caso se emplea también Requests en modo streaming, además de la capacidad de cargar directamente el contenido al bucket en formato comprimido (archivos `.csv` o `.gz`). La automatización asegura que, incluso si el tamaño del archivo crece con el tiempo, el sistema puede procesarlo sin modificaciones.

El tercer script, `rds_s3.py`, es ejecutado por un job en AWS Glue específicamente configurado para leer la tabla catalogada correspondiente a la base de datos en RDS. Este script emplea un `GlueContext`, transforma los datos en un `DynamicFrame` y los exporta como archivos parquet en la zona RAW. La utilización de Glue permite además conservar los metadatos estructurales en el Data Catalog, donde posteriormente pueden ser consumidos por Athena o por otros procesos.

Para ejecutar automáticamente estos scripts, se configuraron tres Jobs en AWS Glue: `descargaAPI`, `descargaURL` y `rds-s3`. Cada uno se asocia directamente a su respectivo script y efectúa su tarea de forma independiente, garantizando que la ingestión de cada fuente se mantenga actualizada en el tiempo. Adicionalmente, se configuraron dos crawlers en AWS Glue para catalogar tanto los datos provenientes de RDS como los datos ya disponibles en S3, lo que facilita la consulta posterior mediante Athena y la integración en procesos ETL.

## 6. PROCESOS AUTOMÁTICOS – AWS GLUE

Se implementaron tres ETL Jobs, cada uno ejecutando un script específico:

### 6.1 Job: descargaURL

- Ejecuta: datos\_url.py
- Descarga archivo CSV desde Datos.gov.

### 6.2 Job: descargaAPI

- Ejecuta: datos\_api.py
- Descarga JSON desde la API pública.

### 6.3 Job: rds-s3

- Ejecuta: rds\_s3.py
- Extrae datos desde RDS utilizando el Glue Data Catalog.

The screenshot shows the AWS Glue Studio interface. On the left, there's a sidebar with navigation links for AWS Glue, Data Catalog, Data Integration and ETL, and Legacy pages. The main area is titled 'AWS Glue Studio' and shows a 'Create job' section with three options: 'Visual ETL' (selected), 'Notebook', and 'Script editor'. Below this is a 'Example jobs' section and a table titled 'Your jobs (3)'. The table lists the following information:

Job name	Type	Created by	Last modified	AWS Glue version	Action
descargaURL	Python shell	Script	18/11/2025, 4:14:43 a. m.	-	Upgraded with AI
rds-s3	Glue ETL	Script	17/11/2025, 4:25:34 p. m.	4.0	Upgraded with AI
descargaAPI	Python shell	Script	16/11/2025, 6:27:04 p. m.	-	Upgraded with AI

## 7. PROCESAMIENTO ETL EN APACHE SPARK UTILIZANDO AMAZON EMR

Uno de los elementos centrales del proyecto es el procesamiento distribuido de los datos mediante Apache Spark. Para ello se creó un clúster EMR denominado emr-proyecto3, configurado con las capacidades necesarias para ejecutar jobs de Spark y Steps automatizados. Este clúster se conectó directamente al bucket proyecto3datalake, lo que permitió leer y escribir archivos en las distintas zonas del data lake.

The screenshot shows the AWS Amazon EMR console. The top navigation bar includes the AWS logo, a search bar, and account information (ID de cuenta: 4264-9140-6849, vclabs/user3862742@sanchezc5@nafit.edu.co). Below the navigation is a breadcrumb trail: Amazon EMR > EMR en EC2: Clústeres > emr-proyecto3. The main content area is titled 'emr-proyecto3'. It has a 'Resumen' section with tabs for 'Propiedades' (selected), 'Acciones de arranque', 'Instancias (hardware)', 'Pasos', 'Aplicaciones', 'Configuraciones', 'Monitorización', 'Eventos', and 'Etiquetas (0)'. The 'Propiedades' tab displays various cluster details:

- Información del clúster**: ID del clúster j-2D2VPJMURLDZ, ARN del clúster arn:aws:elasticmapreduce:us-east-1:426491406849:cluster/j-2D2VPJMURLDZ, Configuración del clúster Grupos de instancias, Capacidad 1 Primary (Principal) | 1 Principal | 1 Tarea.
- Aplicaciones**: Versión de Amazon EMR emr-7.10.0, Aplicaciones instaladas Hadoop 3.4.1, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Livy 0.8.0, Pig 0.17.0, Spark 3.5.5, Tez 0.10.2.
- Administración de clústeres**: Destino del registro en Amazon S3 proyecto3datalake, IU de aplicación persistente Servidor de historial de Spark, Servidor de línea de tiempo de YARN, UI de Tez.
- Estado y hora**: Estado Esperando, Hora de creación 19 de noviembre de 2025 1:47 (UTC-05:00), Tiempo transcurrido 19 minutos, 16 segundos, DNS público del nodo principal ec2-98-84-163-54.compute-1.amazonaws.com, Conectarse al nodo principal mediante SSH, Conectarse al nodo principal mediante SSM.

At the bottom of the page are links for CloudShell, Comentarios, and footer links: © 2025, Amazon Web Services, Inc. o sus filiales. Privacidad, Términos, Preferencias de cookies.

*Clúster creado*

Los procesos ETL desarrollados tienen como finalidad integrar la información proveniente de la API, los archivos CSV y la base de datos relacional, unificándola bajo un esquema coherente para análisis posterior. Esta etapa incluye tareas como limpieza de datos, estandarización de columnas, manejo de valores faltantes, transformación de tipos, enriquecimiento mediante joins con los datos de RDS, generación de métricas preliminares y creación de nuevos campos derivados.

**Instancias creadas:**

Instancias (4) Información										
<input type="button" value="Conectar"/> Estado de la instancia <input type="button" value="Todos los ..."/> <input type="button" value="Acciones"/> Lanzar instancias										
Estado de la instancia = running <input type="button" value="X"/> <input type="button" value="Quitar los filtros"/>										
	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estado de la al.	Zona de dispon...	DNS de IPv		
<input type="checkbox"/>	i-00a492039464f4fc5	i-00a492039464f4fc5	En ejecución <input type="radio"/> <input checked="" type="radio"/>	m5.xlarge	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-44-223-		
<input type="checkbox"/>	EC2-covid	i-0e05b9b150212fb93	En ejecución <input type="radio"/> <input checked="" type="radio"/>	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1f	ec2-44-201-		
<input type="checkbox"/>		i-0367ba242d7aa7af4	En ejecución <input type="radio"/> <input checked="" type="radio"/>	m5.xlarge	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-98-84-1		
<input type="checkbox"/>		i-0f93037635e320f48	En ejecución <input type="radio"/> <input checked="" type="radio"/>	m5.xlarge	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-54-175-		

Instancias (4) Información										
<input type="button" value="Conectar"/> Estado de la instancia <input type="button" value="Todos los ..."/> <input type="button" value="Acciones"/> Lanzar instancias										
Estado de la instancia = running <input type="button" value="X"/> <input type="button" value="Quitar los filtros"/>										
	dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica	Direcciones I...	Monitoreo	Nombre del grupo d...	Nombre de...		
c	ec2-44-223-5-83.comp...	44.223.5.83	-	-	disabled	ElasticMapReduce-slave	projeto3emr			
f	ec2-44-201-71-74.com...	44.201.71.74	-	-	disabled	default,ec2-rds-1	projeto3			
c	ec2-98-84-163-54.com...	98.84.163.54	-	-	disabled	ElasticMapReduce-master	projeto3emr			
c	ec2-54-175-5-5.comput...	54.175.5.5	-	-	disabled	ElasticMapReduce-slave	projeto3emr			

El resultado de este proceso se almacena en la zona trusted/ del data lake, específicamente en la carpeta trusted/enriquecido. Allí se generan archivos parquet, optimizando el desempeño de lectura en Spark y Athena. Este formato, ampliamente utilizado en sistemas de ingeniería de datos, permite la división física de columnas, facilitando consultas parciales de alto rendimiento.

## 8. AWS S3

### 1) Bucket aws-glue-assets-426491406849-us-east-1

Contiene:

- **Carpeta scripts/**

Guarda todos los scripts ejecutados por los Jobs de Glue.

- **Carpeta SparkHistoryLog/**

Para almacenamiento del historial de ejecuciones de aplicaciones Spark.

### 2) Bucket principal del Data Lake: proyecto3datalake

Organizado en zonas:

- Zona raw/

The screenshot shows the Amazon S3 console interface. The main view displays the contents of the 'raw' folder within the 'proyecto3datalake' bucket. The folder contains two empty subfolders: 'covid/' and 'covid/'. The left sidebar provides navigation links for 'Amazon S3', 'Buckets de uso general', 'Storage Lens', and 'Características destacadas'. The top navigation bar includes links for 'Buckets', 'Actions', and 'Create bucket'.

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
covid/	Carpeta	-	-	-
covid/	Carpeta	-	-	-

The screenshot shows the Amazon S3 console interface, specifically the contents of the 'covid/' folder within the 'raw' folder of the 'proyecto3datalake' bucket. The folder is currently empty. The left sidebar provides navigation links for 'Amazon S3', 'Buckets de uso general', 'Storage Lens', and 'Características destacadas'. The top navigation bar includes links for 'Buckets', 'Actions', and 'Create bucket'.

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
	-	-	-	-

**Amazon S3**

Buckets de uso general:

- Buckets de directorio
- Buckets de tablas
- Buckets vectoriales
- Concesiones de acceso
- Puntos de acceso (buckets de uso general, sistemas de archivos FSx)
- Puntos de acceso (buckets de directorio)
- Puntos de acceso del objeto
- Lambdas
- Puntos de acceso de varias regiones
- Opciones por lotes
- Analizador de acceso de IAM para S3

Configuración de bloques de acceso público correspondiente a esta cuenta

**Storage Lens**

StorageLens

**api/**

**Objetos** | Propiedades

**Objetos (1)**

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el inventario de Amazon S3 para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Copiar URI de S3

Nombre | Tipo | Última modificación | Tamaño | Clase de almacenamiento

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
covid_19_simple.json	json	16 Nov 2025 6:27:07 PM -05	1.1 MB	Estandar

**Amazon S3**

Buckets de uso general:

- Buckets de directorio
- Buckets de tablas
- Buckets vectoriales
- Concesiones de acceso
- Puntos de acceso (buckets de uso general, sistemas de archivos FSx)
- Puntos de acceso (buckets de directorio)
- Puntos de acceso del objeto
- Lambdas
- Puntos de acceso de varias regiones
- Opciones por lotes
- Analizador de acceso de IAM para S3

Configuración de bloques de acceso público correspondiente a esta cuenta

**Storage Lens**

Parámetros

Grupos de Storage Lens

**url/**

**Objetos** | Propiedades

**Objetos (1)**

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el inventario de Amazon S3 para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Copiar URI de S3

Nombre | Tipo | Última modificación | Tamaño | Clase de almacenamiento

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
casos_covid.csv	gz	16 Nov 2025 6:17:05 PM -05	87.3 MB	Estandar

**Amazon S3**

Buckets de uso general:

- Buckets de directorio
- Buckets de tablas
- Buckets vectoriales
- Concesiones de acceso
- Puntos de acceso (buckets de uso general, sistemas de archivos FSx)
- Puntos de acceso (buckets de directorio)
- Puntos de acceso del objeto
- Lambdas
- Puntos de acceso de varias regiones
- Opciones por lotes
- Analizador de acceso de IAM para S3

Configuración de bloques de acceso público correspondiente a esta cuenta

**Storage Lens**

Parámetros

Grupos de Storage Lens

Configuración de AWS Organizations

Características destacadas

AWS Marketplace para S3

**rds/**

**Objetos** | Propiedades

**Objetos (9)**

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el inventario de Amazon S3 para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Copiar URI de S3

Nombre | Tipo | Última modificación | Tamaño | Clase de almacenamiento

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
part-00000-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:21 PM -05	813.0 B	Estandar
part-00001-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00002-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00003-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00004-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00005-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00006-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00007-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar
part-00008-ca2-5533-7fd-467b-5fb-62f625d4b53-2000.parquet	parquet	17 Nov 2025 4:22:19 PM -05	1.9 KB	Estandar

- Zona trusted

Amazon S3 > Buckets > proyecto3datalake > trusted/

**Objetos (1)**

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
enriquecido/	Carpeta			

Amazon S3 > Buckets > proyecto3datalake > trusted/ > enriquecido/

**Objetos (2)**

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
_SUCCESS	-	19 Nov 2025 2:00:20 AM -05	0 B	Estándar
c000.snapy.parquet	parquet	19 Nov 2025 1:59:31 AM -05	83.4 MB	Estándar

- Zona refined

Amazon S3 > Buckets > proyecto3datalake > refined/

**Objetos (1)**

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
analisis/	Carpeta			

Amazon S3

Buscar [Alt+S]

Estados Unidos (Norte de Virginia) ID de cuenta: 4264-9140-6849  
voclabs/user3862742:ssanchez5@eafit.edu.co

Amazon S3 > Buckets > proyecto3datalake > refined/ > analysis/

Copiar URI de S3

Objetos Propiedades

Objetos (2)

Copiar URI de S3 Copiar URL Descargar Abrir Eliminar Acciones Crear carpeta Cargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el inventario de Amazon S3 para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. Más información.

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
_SUCCESS	-	19 Nov 2025 2:00:56 AM -05	0 B	Estándar
part-00000-6abbb898-69ab-4de4-8722-f625778c2fb5-c000.snappy.parquet	parquet	19 Nov 2025 2:00:55 AM -05	2.2 KB	Estándar

## Otros directorios

- jupyter/ → notebooks utilizados en EMR o Studio.
- scripts/ → Copias locales de los scripts Python.
- resultados/athenas/ → resultados de consultas ejecutadas en Athena.

## 9. COMPONENTES DE BASE DE DATOS

### 9.1 Base de Datos Relacional AWS RDS

Nombre: rds-covid

Tabla principal importada desde pgAdmin:

casos\_municipios

Campos:

- categoria
- nombre\_departamento
- camas\_uci\_disponibles
- poblacion
- nombre\_municipio
- codigo\_divipola

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying the database structure under 'covidRDS/postgres@RDS'. The 'Tables' section contains one entry: 'casos\_municipios'. The right pane shows a query results window with the following content:

Query History

```
1 SELECT * FROM casos_municipios
```

Data Output

	codigo_divipola	nombre_municipio	nombre_departamento	poblacion	camas_uci_disponibles	categoria
1	11001	BOGOTA	CUNDINAMARCA	7743955	2500	Distrito Capital
2	05001	MEDELLIN	ANTIOQUIA	2533424	1200	Especial
3	76001	CALI	VALLE DEL CAUCA	2252616	850	Especial
4	08001	BARRANQUILLA	ATLANTICO	1274250	600	Especial
5	13001	CARTAGENA	BOLIVAR	1028736	450	Distrito Turistico
6	68001	BUARAMANGA	SANTANDER	591130	300	Primera
7	17001	MANIZALES	CALDAS	434403	200	Primera
8	66001	PEREIRA	RISARALDA	467269	250	Primera

## 9.2 Base de Datos en AWS Glue Data Catalog

Nombre: db\_covid

Tablas:

1. covidrds\_public\_casos\_municipios
2. datos-finalesanalisis

Columnas:

- nombre\_departamento
- casos\_totales
- fallecidos\_totales
- tasa\_letalidad\_pct
- casos\_por\_100k\_hab
- promedio\_camas\_uci

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a navigation sidebar with sections like AWS Glue, Data Catalog, Data Integration and ETL, and Legacy pages. The main area displays the 'db-covid' database properties and a list of tables.

**Database properties:**

Name	Description	Location	Created on (UTC)
db-covid	-	-	November 17, 2025 at 19:40:10

**Tables (2):**

Name	Database	Location	Classification	Deprecated	View data	Data quality	Column stats...
covidrds_public_cas	db-covid	covidRDS.public.cas	postgresql	-	-	View data quality	View statistics
datos-finalesanalisis	db-covid	s3://proyecto3data1	Parquet	-	Table data	View data quality	View statistics

## 10. ANÁLISIS AVANZADO

Una vez finalizadas las transformaciones y el enriquecimiento de datos, se implementó un proceso analítico mediante un script adicional denominado analítica.py. Este script se encarga de cargar la información desde la zona trusted, procesar los datos departamentales y generar indicadores epidemiológicos relevantes para la interpretación del comportamiento del COVID-19 en Colombia.

Entre los cálculos realizados se encuentran el número total de casos por departamento, el número total de fallecidos, la cantidad de recuperados, el promedio departamental de camas UCI disponibles, la tasa de letalidad expresada en porcentaje y la tasa relativa de casos por cada cien mil habitantes. Para ello, se emplean agregaciones de Spark DataFrames como count, sum y avg, acompañadas del uso de funciones condicionales para discriminar estados de recuperación. Posteriormente, se construye una vista temporal que permite ejecutar consultas por medio de SparkSQL, facilitando la clasificación de los departamentos según diferentes criterios.

El resultado finalmente producido consiste en un conjunto de datos refinados con los diez departamentos que presentan la mayor tasa de casos por habitante. Este archivo es almacenado en la zona refined/ del data lake, en el directorio refined/analisis, nuevamente en formato parquet comprimido. A partir de allí puede ser consumido por Athena o por una API externa según los requerimientos analíticos.

## 11. CONSULTAS ANALÍTICAS MEDIANTE AMAZON ATHENA

Athena constituye una herramienta indispensable dentro del flujo, al permitir realizar consultas SQL sobre archivos ubicados en S3 sin la necesidad de motores de base de datos tradicionales. En este proyecto, Athena está asociada al Data Catalog generado por AWS Glue, lo que permite tener tablas definidas con metadatos coherentes, tanto para los datos provenientes de RDS como para los resultados refinados.

El sistema permite consultar directamente tablas como covidrds\_public\_casos\_municipios o datos-finalesanalisis, y generar resultados que luego son almacenados automáticamente en la carpeta resultados/athenas del bucket proyecto3datalake. Esto facilita la explotación final de los datos para informes, visualizaciones o procedimientos externos, y demuestra el valor de un modelo distribuido de consulta sin servidores.

The screenshot shows the Amazon Athena Editor interface. On the left, there's a sidebar titled 'Datos' with sections for 'Origen de datos' (AWS Data Catalog), 'Catalogo' (None), and 'Base de datos' (db-covid). Below that are 'Tablas y vistas' and 'Resultados'. The main area is titled 'Consulta 1' and contains a SQL query: 'SELECT \* FROM "db-covid"."datos-finalesanalisis" LIMIT 300'. The results section shows a table with 6 rows of data:

#	nombre_departamento	casos_totales	fallecidos_totales	tasa_mortalidad_pct	casos_por_100k_hab	promedio_camas_usd
1	SANTANDER	142642	5568	2.56	24520.04	500.0
2	CUNDINAMARCA	1388137	30179	1.6	24320.08	2500.0
3	CAJAMARCA	84478	1245	1.47	19445.92	200.0
4	VALLE DEL CAUCA	406751	8909	2.19	18058.83	850.0
5	BOLIVAR	155326	2347	1.44	15895.82	450.0
6	RISARALDA	73507	1664	2.18	15731.2	250.0

## 12. EXPLICACIÓN DE SCRIPTS

### 12.1 datos\_api.py

Funcionalidad:

- Consumir API REST del Ministerio de Salud.
- Descargar el JSON en modo streaming para mayor eficiencia.
- Enviar el contenido directamente a S3 sin guardar en disco.

### 12.2 datos\_url.py

Funcionalidad:

- Descargar el archivo CSV publicado por Datos.gov.
- Utilizar streaming para archivos grandes.
- Subir el contenido comprimido (.gz) directamente a S3.

### 12.3 rds\_s3.py

Este script ejecutado desde Glue:

- Conecta con la tabla catalogada de RDS.
- Convierte el resultado en un DynamicFrame.
- Lo almacena en S3 como Parquet con Snappy.

### 12.4 analítica.py

Realiza la parte analítica:

- Carga de parquet desde TRUSTED.
- Agrupaciones y cálculos estadísticos.
- Consulta SQL para obtener ranking departamental.

## 13. GUÍA PASO A PASO (EJECUCIÓN)

El proyecto se divide en 3 fases:

- Ingesta de datos
- Procesamiento
- Consumo

A continuación, presentaremos una guía general de cómo llevar a cabo cada una de las fases

### 1. Fase de ingestá

Paso	Componente	Acción Clave
1. Infraestructura RDS	RDS/VPC	Crear la tabla dim_municipios en PostgreSQL y llenarla con los 8 registros esenciales.
2. Conexión de Red	Security Group	Asegurar que el rds-ec2-1 tenga una regla de entrada auto-referenciada para <b>Todo el Tráfico TCP</b> (0-65535) para permitir la comunicación de Glue/EMR.
3. Ingesta URL	Glue Job	Crear y ejecutar el Job Python Shell (descargaURL) para descargar el archivo casos_covid.gz y guardarlo en s3://.../raw/url/.
4. Ingesta RDS	Glue Job (Spark)	Crear y ejecutar el Job Spark (rds-s3) utilizando la conexión Postgresql connection para leer

		dim_municipios y guardarlo en formato Parquet en s3://.../raw/rds/.
--	--	---

## 2. Fase de procesamiento

Paso	Componente	Objetivo
1. Configuración EMR	EMR Cluster	Lanzar un clúster estable (ej. M7g) con Terminación Automática.
2. EMR Step 1 (JOIN)	etl_join.py	JOIN de COVID (raw/url/casos_covid.gz) con Municipios (raw/rds/) usando PySpark. El script maneja el renombrado dinámico de la columna DIVIPOLA.
3. Salida Trusted	S3 Trusted	Resultado: Data enriquecida guardada en s3://.../trusted/covid_enriquecido.
4. EMR Step 2 (Analítica)	etl_analitica_refined.py	Leer la zona Trusted. Ejecutar el análisis descriptivo (Tasa de Letalidad, Casos por 100K) utilizando <b>DataFrames Pipelines</b> y <b>SparkSQL</b> .
5. Salida Refined	S3 Refined	Resultado: Ranking de análisis guardado en s3://.../refined/analisis/.

### **3. Fase de consumo**

Paso	Componente	Acción Clave
<b>1. Catalogación</b>	<b>Glue Crawler</b>	Crear y ejecutar el CrawlerRefined_Analisis apuntando a s3://.../refined/analisis/.
<b>2. Consulta SQL</b>	<b>Athena</b>	Verificar que la tabla (ej. datos-finales-analisis) esté disponible en la base de datos db-covid y ejecutar la consulta de <i>ranking</i> .
<b>3. Acceso Web</b>	<b>Lambda &amp; API Gateway</b>	Crear la función Lambda (Python/boto3) para consultar la tabla de Athena y devolver los resultados en formato JSON. <b>Exponer la función vía API Gateway</b> .

## 14. CONCLUSIONES

El desarrollo de este proyecto permitió comprender con profundidad la complejidad y relevancia de implementar una arquitectura distribuida para la gestión de datos en un contexto real. La utilización integrada de varios servicios de AWS puso de manifiesto la importancia de elegir adecuadamente las tecnologías que permitan automatizar procesos, manejar volúmenes significativos de información y garantizar la disponibilidad continua de los datos en todas las etapas del flujo. La experiencia adquirida en la ingesta automática desde fuentes heterogéneas confirmó que la combinación de datos abiertos y bases de

datos relacionales enriquece significativamente las capacidades analíticas y permite ofrecer una visión más precisa sobre los fenómenos estudiados.

El proyecto evidenció también el papel central que desempeñan los data lakes en la gestión moderna de información, particularmente gracias a su flexibilidad para almacenar datos sin necesidad de esquemas rígidos. La estructuración en zonas RAW, trusted y refined facilita la separación entre las etapas del ciclo de vida de los datos y contribuye a la trazabilidad y calidad del producto final. Del mismo modo, el uso de Amazon EMR y Apache Spark permitió experimentar con procesamiento distribuido a gran escala, demostrando la potencia de estas herramientas para ejecutar tareas ETL y de análisis avanzado sobre conjuntos amplios de datos epidemiológicos.

Asimismo, la integración con AWS Glue y Amazon Athena reveló el papel fundamental de los catálogos de metadatos y los motores de consulta sin servidor, ya que ofrecen una forma eficiente de acceder, explorar y validar los resultados generados por el sistema. La automatización lograda en los procesos de ingesta y transformación aseguró una operación fluida, replicable y confiable, alineada con las prácticas profesionales de la ingeniería de datos. La experiencia completa fortaleció la comprensión de los desafíos reales asociados al manejo de sistemas distribuidos y al diseño de soluciones escalables basadas en la nube.

Finalmente, la construcción conjunta de esta arquitectura permitió al equipo aplicar, en un caso práctico y realista, muchos de los conceptos aprendidos teóricamente, integrando conocimientos de sistemas distribuidos, bases de datos, computación en la nube, procesamiento masivo de datos y analítica.

## REFERENCIAS

<https://youtu.be/ZFns7fvBCH4?si=hu5Y34JDB9yY7bsd>

<https://github.com/airscholar/EMR-for-data-engineers/tree/main>