

UNIR - Universidad Internacional de la Rioja
Ciudad de México

ACTIVIDAD 1: Gobierno de Datos y Gestión de Datos con Herramientas de Microsoft

Alumno: Cuenca Roa, Leonard Jose

Grupo: 1001

Equipo 01C

ÍNDICE

Objetivo 1	3
Instalación del software Dremio	3
Carga de Datos	4
Espacios de trabajo	7
Crear los datasets personalizados	8
Objetivos 2	13
Tarea de realización de ETL y modelado de datos	13
Descarga de datos Ejemplo de COVID-19 de EE. UU.	13
Descargar Microsoft Visual Studio	14
Descargar Microsoft SQL Server	14
Preparación de los datos del archivo dataset COVID 03-03-2023.csv:	15
Extracción de los datos del archivo dataset COVID 03-03-2023.csv:	17
Transformación de los datos del archivo dataset COVID 03-03-2023.csv:	19
Carga de los datos del archivo dataset COVID 03-03-2023.csv:	20
Carga datos de ejemplos propios de Dremio (Opcional)	23
Ficheros Parquet	23
Un ejemplo para explicarlo	23

Objetivo 1

Instalación del software Dremio

Se procedió a instalar la herramienta Dremio utilizando docker, dado que se dispone de un equipo Mac para realizar las prácticas.

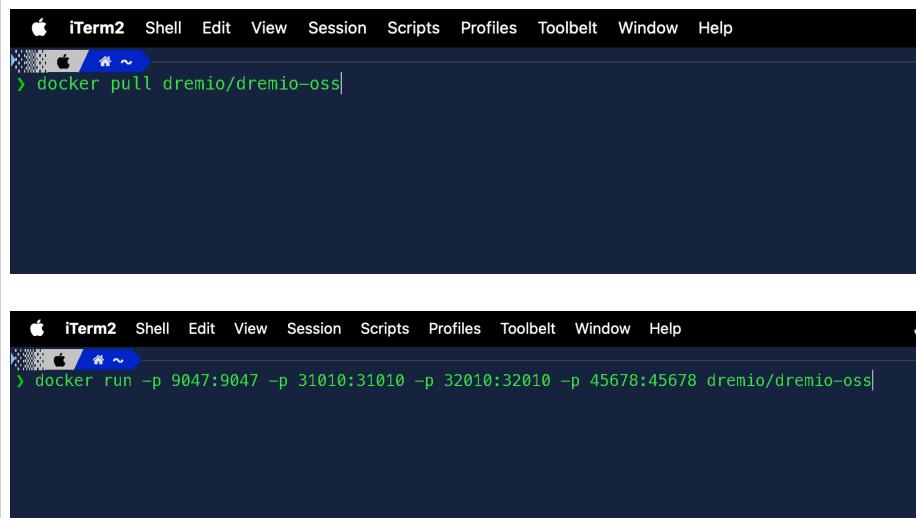
Para habilitar Dremio en modo escritorio, se ejecutaron los siguientes comandos en el siguiente orden:

```
docker pull dremio/dremio-oss
```

```
docker run -p 9047:9047 -p 31010:31010 -p 32010:32010 -p 45678:45678 dremio/dremio-oss
```

Se realizó la validación de la instalación usando la interfaz de docker para comprobar si el contenedor fue instalado y ejecutado apropiadamente.

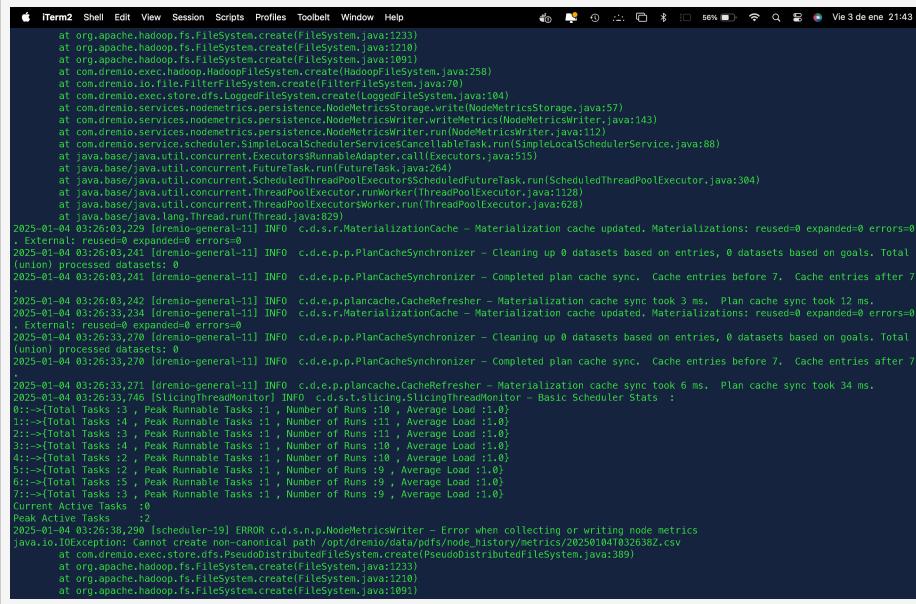
Evidencia 1



```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
> docker pull dremio/dremio-oss

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
> docker run -p 9047:9047 -p 31010:31010 -p 32010:32010 -p 45678:45678 dremio/dremio-oss
```

Evidencia 2



```
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:123)
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:109)
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:258)
at com.dremio.io.file.FileSystem.create(FilterFileSystem.java:70)
at com.dremio.executor.store.dfs.LoggedFileSystem.create(LoggedFileSystem.java:104)
at com.dremio.services.nodemetrics.persistence.NodemetricsStorage.write(NodemetricsStorage.java:57)
at com.dremio.services.nodemetrics.persistence.NodemetricsWriter.run(NodemetricsWriter.java:143)
at com.dremio.service.nodemetrics.persistence.NodemetricsWriter.run(NodemetricsWriter.java:12)
at com.dremio.service.scheduler.SimpleLocalSchedulerService$CancelableTask.run(SimpleLocalSchedulerService.java:88)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:51)
at java.util.concurrent.FutureTask.run(FutureTask.java:264)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:620)
at java.lang.Thread.run(Thread.java:929)

2025-01-04 03:26:03.229 [dremio-general-1] INFO c.d.s.r.MaterializationCache - Materialization cache updated. Materializations: reused=0 expanded=0 errors=0
, External: reused=0 expanded=0 errors=0
2025-01-04 03:26:03.241 [dremio-general-1] INFO c.d.e.p.p.PlanCacheSynchronizer - Cleaning up 0 datasets based on entries, 0 datasets based on goals. Total (union) processed datasets: 0
2025-01-04 03:26:03.241 [dremio-general-1] INFO c.d.e.p.p.PlanCacheSynchronizer - Completed plan cache sync. Cache entries before 7. Cache entries after 7
2025-01-04 03:26:03.242 [dremio-general-1] INFO c.d.e.p.p.PlanCache.CacheRefresher - Materialization cache sync took 3 ms. Plan cache sync took 12 ms.
2025-01-04 03:26:03.234 [dremio-general-1] INFO c.d.s.r.MaterializationCache - Materialization cache updated. Materializations: reused=0 expanded=0 errors=0
, External: reused=0 expanded=0 errors=0
2025-01-04 03:26:03.270 [dremio-general-1] INFO c.d.e.p.p.PlanCacheSynchronizer - Cleaning up 0 datasets based on entries, 0 datasets based on goals. Total (union) processed datasets: 0
2025-01-04 03:26:03.270 [dremio-general-1] INFO c.d.e.p.p.PlanCacheSynchronizer - Completed plan cache sync. Cache entries before 7. Cache entries after 7
2025-01-04 03:26:03.271 [dremio-general-1] INFO c.d.e.p.p.PlanCache.CacheRefresher - Materialization cache sync took 6 ms. Plan cache sync took 34 ms.
2025-01-04 03:26:03.746 [SlicingThreadMonitor] INFO c.d.s.t.slicing.SlicingThreadMonitor - Basic Scheduler Stats :
0::=>(Total Tasks :3 , Peak Runnable Tasks :1 , Number of Runs :10 , Average Load :1.0)
1::=>(Total Tasks :4 , Peak Runnable Tasks :1 , Number of Runs :11 , Average Load :1.0)
2::=>(Total Tasks :3 , Peak Runnable Tasks :1 , Number of Runs :11 , Average Load :1.0)
3::=>(Total Tasks :4 , Peak Runnable Tasks :1 , Number of Runs :11 , Average Load :1.0)
4::=>(Total Tasks :2 , Peak Runnable Tasks :1 , Number of Runs :10 , Average Load :1.0)
5::=>(Total Tasks :2 , Peak Runnable Tasks :1 , Number of Runs :9 , Average Load :1.0)
6::=>(Total Tasks :5 , Peak Runnable Tasks :1 , Number of Runs :9 , Average Load :1.0)
7::=>(Total Tasks :3 , Peak Runnable Tasks :1 , Number of Runs :9 , Average Load :1.0)
Current Active Tasks :0
Peak Active Tasks :2
2025-01-04 03:26:03.290 [Scheduler-19] ERROR c.d.s.r.NodeMetricsWriter - Error when collecting or writing node metrics
java.io.IOException: Cannot create non-canonical path /opt/dremio/data/pdfs/node_history/metrics/20250104T032603Z.csv
at com.dremio.executor.store.dfs.PseudoDistributedFileSystem.create(PseudoDistributedFileSystem.java:309)
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:123)
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:109)
at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:1091)
```

Se procedió a instalar la herramienta Dremio utilizando docker, dado que se dispone de un equipo Mac para realizar las prácticas.

Para habilitar Dremio en modo escritorio, se ejecutaron los siguientes comandos en el siguiente orden:

docker pull dremio/dremio-oss

docker run -p 9047:9047 -p 31010:31010 -p 32010:32010 -p 45678:45678 dremio/dremio-oss

Se realizó la validación de la instalación usando la interfaz de docker para comprobar si el contenedor fue instalado y ejecutado apropiadamente.

Evidencia 3

The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with icons for Containers, Images, Volumes, Builds, and Docker Scout, along with an 'Add Extensions' button. The main area is titled 'Containers' with a 'Give feedback' link. It displays container usage statistics: 'Container CPU usage 5.67% / 800% (8 CPUs available)' and 'Container memory usage 1.88GB / 3.74GB'. A 'Show charts' button is also present. Below this, a table lists the running container: 'kind_hypatia' (image: dremio/dremio-oss, status: Running, ports: 31010:31010, 32010:32010, 45678:45678, 9047:9047), with a 'CPU (%)' of 5.67% and started 1 hour ago. There are 'Actions' and 'Show less' buttons for each row.

Carga de Datos

Posteriormente, se accedió a la interfaz de Dremio a través del navegador web en la dirección:

<http://localhost:9047>

Con el propósito de cumplir con el objetivo 1, descrito en el enunciado, procederemos a cargar cada conjunto de datos en la plataforma Dremio. Una vez cargados, generaremos para cada uno una wiki, siguiendo las indicaciones del documento provisto

Evidencia 1

The screenshot shows the Dremio web interface. The left sidebar has sections for 'Datasets' (Leasan, Spaces (0)), 'Sources' (No data sources yet, Add source), and other settings. The main area shows a list of datasets under the space '@Leasan.datasets': 'books' (Jobs: 1), 'Licencias_Locales_202104' (Jobs: 0), 'Locales_202104' (Jobs: 0), and 'Terrazas_202104' (Jobs: 0). There are 'Filter...' and '+' buttons at the top right of the dataset list.

Posteriormente, se accedió a la interfaz de Dremio a través del navegador web en la dirección:

<http://localhost:9047>

Con el propósito de cumplir con el objetivo 1, descrito en el enunciado, procederemos a cargar cada conjunto de datos en la plataforma Dremio. Una vez cargados, generaremos para cada uno una wiki, siguiendo las indicaciones del documento provisto

Evidencia 2 Wikipedia Books

The screenshot shows the Dremio web interface with the URL localhost:9047/home/%40Leosan/datasets.books/wiki?tipVersion=0005291638931065&version=0005291638.... The main panel displays the 'books' dataset from the '@Leosan.datasets' space. The 'Details' tab is selected, showing the following information:

- Wiki**: Análisis de Libros. Este es el dataset de archivo book.js.
- Descripción**: Tenemos un conjunto de datos que describen los atributos principales de una lista de más de 800 libros.
- Se describe los atributos de la tabla**:

Nombre Atributo	Tipo	Descripción
id	entero	campo que indica la clave primaria que permite identificar el registro
title	string	campo que indica el título del libro
isbn	entero	campo que indica el registro único en el sistema bibliotecario
pageCount	entero	campo que indica el total de páginas que contiene el libro
publishedDate	date	campo que indica la fecha de publicación
thumbnailUrl	string	campo que indica el enlace de compra en amazon
shortDescription	string	campo que indica una descripción corta del libro
longDescription	string	campo que indica una descripción larga del libro
status	string	campo que indica el estatus del libro si está publicado o no
authors	array	campo que indica el nombre de autores
categories	array	campo que indica los nombre de la categoría perteneciente
- Overview**: Shows statistics: Jobs (last 30 days) 1, Descendants 0, Created 03/01/2025, 22:10:50, Last updated 03/01/2025, 22:10:50.

Evidencia 3 Wikipedia Licencia 2021

The screenshot shows the Dremio web interface with the URL localhost:9047/home/%40Leosan/datasets.Licencias_Locales_202104/wiki?tipVersion=0007389060722506&version=0007389060722506.... The main panel displays the 'Licencias_Locales_202104' dataset from the '@Leosan.datasets' space. The 'Details' tab is selected, showing the following information:

- Wiki**: Análisis de Licencias Locales 2021. Este es el dataset del archivo Licencias_Locales_202104.csv.
- Descripción**: Tenemos un conjunto de datos que describen los atributos principales de la licencia de establecimientos comerciales del año 2021 con mas de 132173 registros.
- Se describe los atributos de la tabla**:

Nombre Atributo	Tipo	Descripción
id_local	entero	campo que indica la clave primaria que permite identificar el registro
id_distrito_local	entero	campo clave foranea
desc_distrito_local	string	campo que indica la descripción local
id_barrio_local	entero	campo clave foranea
desc_barrio_local	string	campo que indica la descripción barrio local
cod_barrio_local	string	campo que indica el código del barrio local
id_seccion_censal_local	entero	campo clave foranea
desc_seccion_censal_local	string	campo que indica una descripción censal local
coordenada_x_local	decimal	campo que indica la coordenada x
coordenada_y_local	decimal	campo que indica la coordenada y
id_tipo_acceso_local	entero	campo clave foranea
- Overview**: Shows statistics: Jobs (last 30 days) 4, Descendants 0, Created 03/01/2025, 22:23:13, Last updated 03/01/2025, 22:23:13.

Posteriormente, se accedió a la interfaz de Dremio a través del navegador web en la dirección:

<http://localhost:9047>

Con el propósito de cumplir con el objetivo 1, descrito en el enunciado, procederemos a cargar cada conjunto de datos en la plataforma Dremio. Una vez cargados, generaremos para cada uno una wiki, siguiendo las indicaciones del documento provisto

Evidencia 4 Wikipedia Locales 2021

The screenshot shows the Dremio web interface with the URL http://localhost:9047/home/%40Leosan/datasets.Locales_202104/wiki?tipVersion=0000287164470648&version=00.... The page title is "Locales_202104" under the "@Leosan.datasets" space. The "Details" tab is selected. The "Wiki" section contains the following text:
Análisis de Locales 2021
Este es el dataset del archivo Locales_202104.csv
Descripción: Tenemos un conjunto de datos que describen los atributos principales de los locales de establecimientos comerciales del año 2021 con mas de 148000 registros.

Se describe los atributos de la tabla

Nombre Atributo	Tipo	Descripción
id_local	entero	campo que indica la clave primaria que permite identificar el registro
id_distrito_local	entero	campo clave foranea
desc_distrito_local	string	campo que indica la descripción local
id_barrio_local	entero	campo clave foranea
desc_barrio_local	string	campo que indica la descripción barrio local
cod_barrio_local	string	campo que indica el código del barrio local
id_sección_censal_local	entero	campo clave foranea
desc_sección_censal_local	string	campo que indica una descripción censal local
coordenada_x_local	decimal	campo que indica la coordenada x
coordenada_y_local	decimal	campo que indica la coordenada y
id_tipo_acceso_local	entero	campo clave foranea

The "Overview" panel on the right shows the following statistics:
"@Leosan".datasets."Locales_202104"
No Label 0
Jobs (last 30 days) 2
Descendants 0
Created 03/01/2025, 22:59:54
Last updated 03/01/2025, 22:59:54

Evidencia 5 Wikipedia Terrazas

The screenshot shows the Dremio web interface with the URL http://localhost:9047/home/%40Leosan/datasets.Terrazas_202104/wiki?tipVersion=18873dac-3f3f-66d1-fac3-888ab6d33.... The page title is "Terrazas_202104" under the "@Leosan.datasets" space. The "Details" tab is selected. The "Wiki" section contains the following text:
Análisis de Terrazas 2021
Este es el dataset del archivo Terrazas_202104.csv
Descripción: Tenemos un conjunto de datos que describen los atributos principales de las terrazas de establecimientos comerciales del año 2021 con mas de 6000 registros.

Se describe los atributos de la tabla

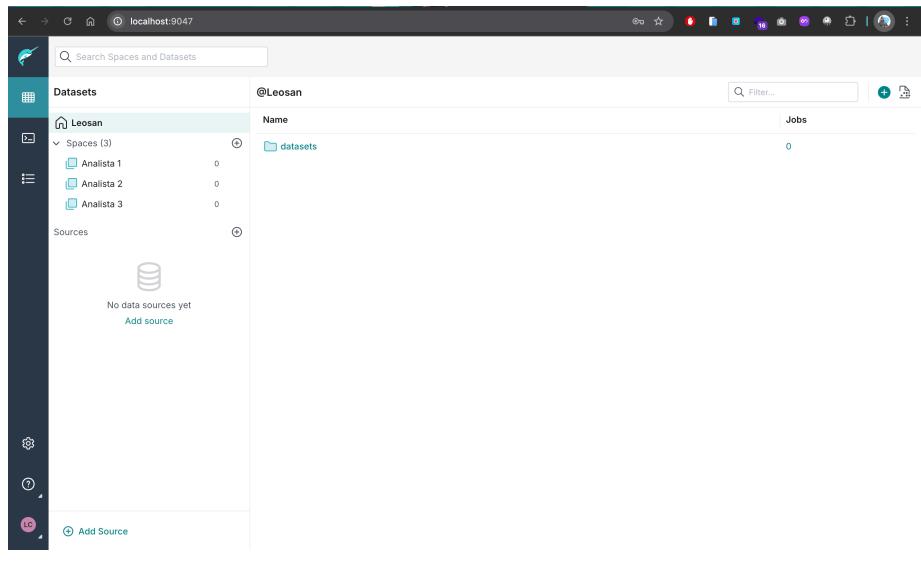
Nombre Atributo	Tipo	Descripción
id_terrazza	entero	campo que indica la clave primaria que permite identificar el registro
id_local	entero	campo clave foranea
id_distrito_local	entero	campo clave foranea
desc_distrito_local	string	campo que indica la descripción del distrito local
id_barrio_local	entero	campo clave foranea
desc_barrio_local	string	campo que indica la descripción del barrio local
id_ndp_edificio	entero	campo clave foranea
id_clase_ndp_edificio	entero	campo clave foranea
id_vial_edificio	entero	campo clave foranea
clase_vial_edificio	string	campo que describe la clase del edificio
desc_vial_edificio	string	campo que describe la vialidad del edificio

The "Overview" panel on the right shows the following statistics:
"@Leosan".datasets."Terrazas_202104"
No Label 0
Jobs (last 30 days) 1
Descendants 0
Created 03/01/2025, 23:07:25
Last updated 03/01/2025, 23:07:25

Espacios de trabajo

Con el fin de avanzar en el objetivo 1, se crearon los espacios de trabajo siguiendo las especificaciones del documento. Como prueba de ello, se adjunta la evidencia de la generación de Dremio

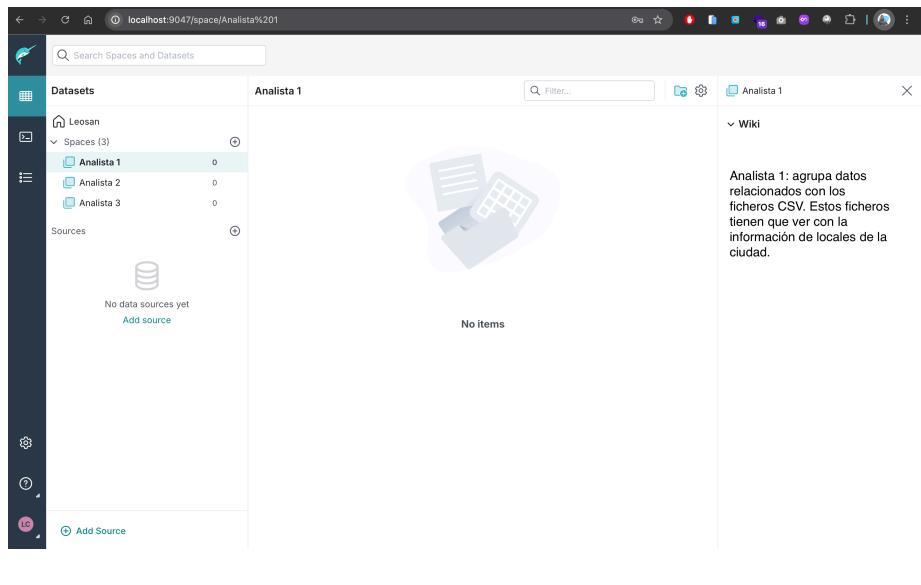
Evidencia 1



The screenshot shows the Dremio interface at localhost:9047. On the left, there's a sidebar with a navigation menu. Under 'Datasets', it shows a workspace named 'Leosan' which contains three spaces: 'Analista 1', 'Analista 2', and 'Analista 3'. Below this, under 'Sources', it says 'No data sources yet' with a link to 'Add source'. On the right, there's a main panel titled '@Leosan' with a table header 'Name'. It lists a single dataset named 'datasets' with 0 rows and 0 jobs. There are also 'Filter...' and '+' buttons.

Se generó los wiki por cada espacio de trabajo Analista 1

Evidencia 2



This screenshot shows the Dremio interface at localhost:9047/space/Analista%201. The left sidebar shows the 'Leosan' workspace with three spaces: 'Analista 1', 'Analista 2', and 'Analista 3'. The right panel is titled 'Analista 1'. It features a 'Wiki' section with a description: 'Analista 1: agrupa datos relacionados con los ficheros CSV. Estos ficheros tienen que ver con la información de locales de la ciudad.' Below this, there's a 'No items' message. A small icon of a document and keyboard is visible.

Se generó los wiki por cada espacio de trabajo Analista 2

Evidencia 3

The screenshot shows the Leosan interface with the title 'Analista 2'. On the left, there's a sidebar with 'Datasets' and 'Sources' sections. Under 'Datasets', there's a 'Spaces (3)' section with three entries: 'Analista 1', 'Analista 2' (which is selected), and 'Analista 3'. Each entry has a count of '0' next to it. Below this is a 'Sources' section with a 'No data sources yet' message and a 'Add source' button. In the center, there's a large placeholder icon for a document or dataset. On the right, there's a 'Wiki' section with the following text:
Analista 2: agrupa datos relacionados con los ficheros JSON. Estos ficheros tienen información sobre las librerías de la ciudad.

Se generó los wiki por cada espacio de trabajo Analista 3

Evidencia 4

The screenshot shows the Leosan interface with the title 'Analista 3'. The layout is identical to Evidence 3, with a sidebar for 'Datasets' and 'Sources'. The 'Spaces (3)' section shows 'Analista 1', 'Analista 2', and 'Analista 3' with counts of '0'. The 'Sources' section has a 'No data sources yet' message and a 'Add source' button. The central area is a placeholder for a document. The right side features a 'Wiki' section with the following text:
Analista 3: agrupa datos relacionados con los ficheros Open Data. Estos ficheros tienen información sobre el clima de la ciudad.

Crear los datasets personalizados

Se solicitó realizar una serie de modificaciones al dataset Terrazas_202104. Estas incluyen eliminar todos los campos que comiencen con 'id_' excepto 'id_terrazas', así como eliminar el campo 'Escalera'. Además, se debe crear un nuevo campo denominado 'Superficie_TO', el cual será el resultado de sumar los valores de los campos 'Superficie_ES' y 'Superficie_RA'. Finalmente, la consulta realizada debe guardarse con el nombre 'Terreza_001' en el espacio de trabajo 'Analista 1'

Evidencia 1

The screenshot shows a database interface with a query editor and a results table.

SQL Script:

```

1 SELECT id_terrazas, desc_distrito_local, desc_barrio_local, clase_vial_edificio, nom_edificio, Cod_Postal,
2     coordenada_x_local, coordenada_y_local, desc_tipo_acceso_local, desc_situacion_local,
3     secuencia_local_RC, coordenada_x_agrupacion, coordenada_y_agrupacion, rotulo, desc_período_terrazas,
4     desc_situacion_local,
5     CAST(NULLIF(Trim(REPLACE(Superficie_ES, ',', ',')), '') AS DECIMAL(10,2)) +
6     CAST(NULLIF(Trim(REPLACE(Superficie_RA, ',', ',')), '') AS DECIMAL(10,2)) AS Superficie_TO,
7     Fecha_config_ult_decreto_resol, DESC_CLASE, DESC_NOMBRE, nom_terrazas, num_terrazas,
8     desc Ubicacion_terraza, hora_ini_L1_es, hora_fin_L1_es, hora_ini_L1_ra, hora_fin_L1_ra, hora_ini_VS_es,
9     hora_fin_VS_es FROM "@Leosan".datasets."Terrazas_202104"

```

Results Table:

c_período_terrazas	desc_situacion_local	# Superficie_TO
anal	Abierto	null	24.03	24/03/2014	CALLE	AI
anal	Abierto	42.12	04/04/2014	AVENIDA	AI	
anal	Abierto	22.32	05/05/2014	CALLE	AI	
anal	Abierto	58.96	07/06/2014	CALLE	AI	
anal	Abierto	42.88	07/06/2014	CALLE	AI	
anal	Abierto	149.67	18/06/2014	PLAZA	AI	
anal	Abierto	45.37	15/08/2014	CALLE	AI	
anal	Abierto	24.86	15/08/2014	CALLE	AI	

Evidencia 2

The screenshot shows a dataset management interface with a sidebar and a main panel.

Sidebar:

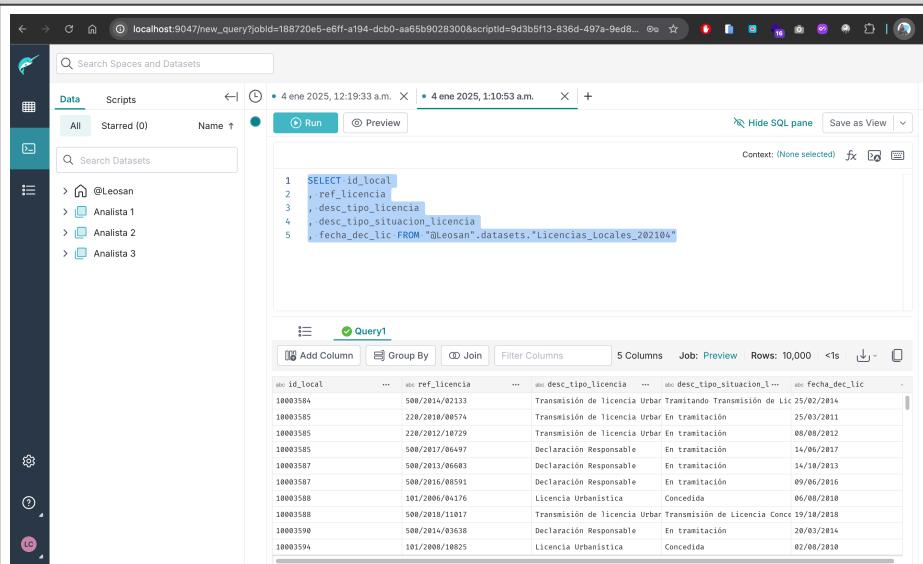
- Search bar: Search Spaces and Datasets
- Data: Selected
- Scripts: Available
- All: Available
- Starred (0)
- Name ↑
- Run: Available
- Preview: Available
- Context: (None selected)
- Hide SQL pane
- Save as View

Main Panel:

- Datasets:**
 - Leosan (Space)
 - Analista 1 (Dataset, 1 row)
 - Analista 2 (Dataset, 0 rows)
 - Analista 3 (Dataset, 0 rows)
- Analista 1:**
 - Name: Terreza_001
 - Jobs: 2
- Sources:**
 - No data sources yet
 - Add source

Se solicitó modificar el dataset "Licencias_Locales_202104" eliminando todas las columnas, excepto "id_local", "ref_licencia", "desc_tipo_licencia", "desc_tipo_situacion_licencia" y "fecha_dec_lic". Una vez realizada esta limpieza, se debe guardar el nuevo dataset con el nombre "Licencias_002" en la ubicación de trabajo correspondiente al usuario "Analista 1"

Evidencia 1



```

1 SELECT id_local
2 , ref_licencia
3 , desc_tipo_licencia
4 , desc_tipo_situacion_licencia
5 , fecha_dec_lic FROM "@Leosan".datasets."Licencias_Locales_202104"

```

The screenshot shows a database interface with a sidebar containing datasets like '@Leosan', 'Analista 1', 'Analista 2', and 'Analista 3'. The main area displays a SQL query in the 'Query1' tab:

```

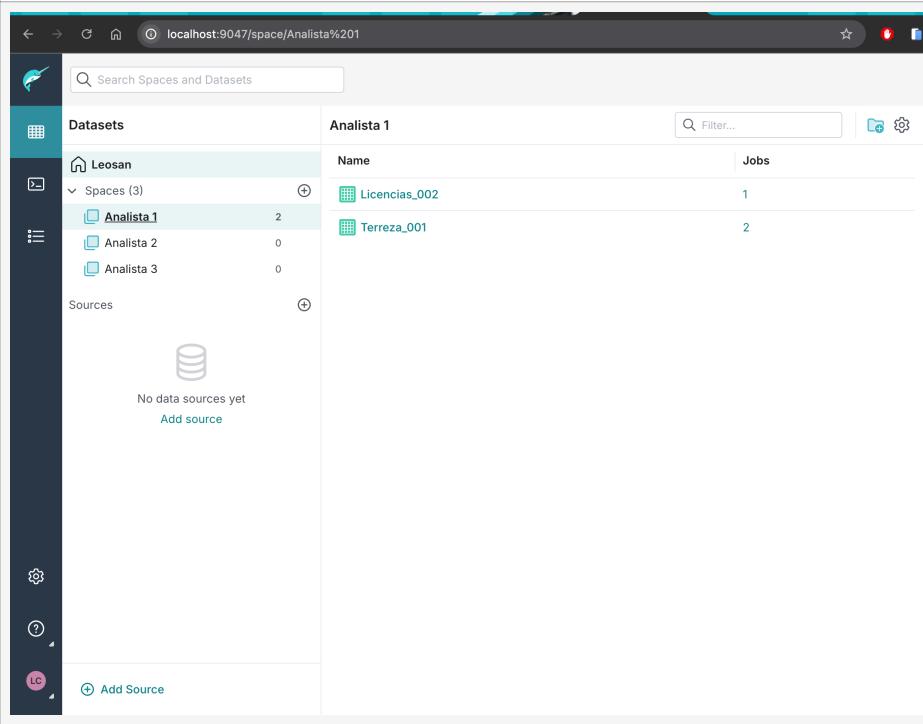
1 SELECT id_local
2 , ref_licencia
3 , desc_tipo_licencia
4 , desc_tipo_situacion_licencia
5 , fecha_dec_lic FROM "@Leosan".datasets."Licencias_Locales_202104"

```

Below the query, a preview pane shows the results of the query, which include columns: id_local, ref_licencia, desc_tipo_licencia, desc_tipo_situacion_licencia, and fecha_dec_lic. The data rows are as follows:

id_local	ref_licencia	desc_tipo_licencia	desc_tipo_situacion_licencia	fecha_dec_lic
10003584	500/2014/02133	Transmisión de licencia Urbana	Tramitando Transmisión de Lic.	25/02/2014
10003585	220/2010/00574	Transmisión de licencia Urbana	En tramitación	25/03/2011
10003585	220/2012/10729	Transmisión de licencia Urbana	En tramitación	08/08/2012
10003585	500/2013/00497	Declaración Responsable	En tramitación	14/06/2013
10003587	500/2013/00603	Declaración Responsable	En tramitación	14/10/2013
10003587	500/2015/00591	Declaración Responsable	En tramitación	09/06/2016
10003588	101/2006/00176	Licencia Urbanística	Concedida	06/08/2010
10003588	500/2013/11017	Transmisión de licencia Urbana	Transmisión de Licencia Concesión	19/10/2013
10003590	500/2014/00638	Declaración Responsable	En tramitación	20/03/2014
10003594	101/2005/10025	Licencia Urbanística	Concedida	02/08/2010

Evidencia 2



The screenshot shows a dataset management interface with a sidebar showing spaces like 'Leosan' and 'Analista 1'. The main area shows a table of datasets:

Name	Jobs
Licencias_002	1
Terreza_001	2

At the bottom, there is a button labeled '+ Add Source'.

Se requiere realizar un análisis de datos combinando dos conjuntos de información. El primer paso es acceder al conjunto de datos "Terrazas_202104". A continuación, se debe vincular este conjunto con otro denominado "Licencia_002". La conexión entre ambos se establecerá utilizando el campo común "id_local", mediante un tipo de unión interna. El resultado de esta operación debe almacenarse como un nuevo conjunto de datos llamado "Licencias_Terrazas_003" en el espacio de trabajo asignado al "Analista 2"

Evidencia 1

```

2 FROM (
3 secuencial_local_Pc, Escalera, id_planta_agrupado, id_local_agrupado, coordenada_x_agrupacion, coordenada_y_agrupacion, rotulo,
4 id_perodo_terraza, desc_perodo_terraza, id_situacion_terraza, desc_situacion_terraza, Superficie_ES, Superficie_RA,
5 Fecha_confir_ult_decreto_resol, id_ndp_terraza, id_clase_ndp_terraza, ID_VIAL, DESC_CLASE, DESC_NOMBRE, nom_terraza, num_terraza,
6 cal_terraza, desc ubicacion_terraza, hora_ini_L1_es, hora_fin_L1_es, hora_ini_L2_ra, hora_fin_L2_ra, hora_ini_VS_es, hora_fin_VS_ra,
7 hora_ini_VS_ra, hora_fin_VS_ra, mesas_aux_ra, mesas_aux_es, mesas_es, mesas_ra, sillas_es, sillas_ra
8 ) nested_0
9 FROM "@Leosan".datasets.Terrazas_202104 AS Terrazas_202104
10 ) nested_0
11 INNER JOIN "Analista 1".Licencias_002 AS join_licencias_002 ON nested_0.id_local = join_licencias_002.id_local;
12

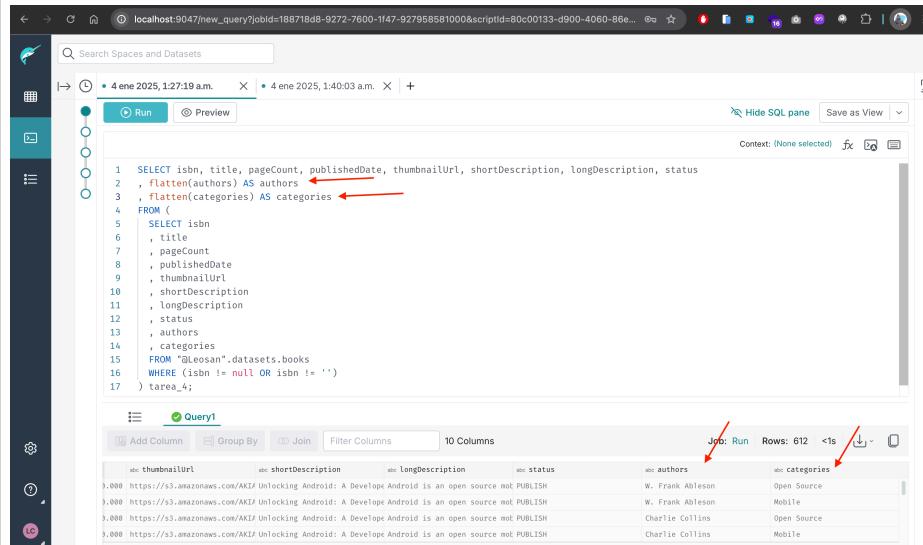
```

Evidencia 2

Name	Jobs
Licencias_Terrazas_003	0

Se solicita modificar un conjunto de datos llamado "books" siguiendo estos pasos: En primer lugar, se elimina un campo innecesario llamado "_id". Luego, se filtran los datos para excluir todos los libros que no tengan un número ISBN. A continuación, se aplica una transformación llamada "unnest" a las columnas "authors" y "categories", lo que probablemente implica descomponer listas o arreglos de autores y categorías en filas individuales para facilitar su análisis. Finalmente, se guarda este nuevo conjunto de datos modificado con el nombre "Books_001" en un área de trabajo específica llamada "Analista 3". En resumen, se está preparando un dataset más limpio y estructurado para realizar análisis posteriores, aplicando los unnest o para este caso un flatten podemos realizar búsquedas mas rápidas separando ambos campos en registros únicos por usuario y categorías.

Evidencia 1



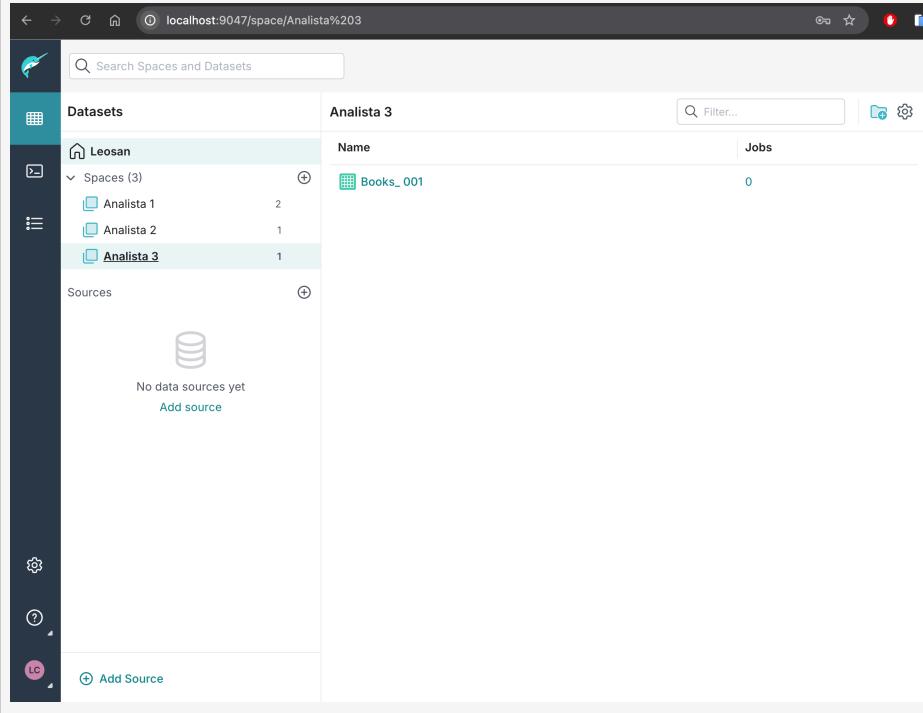
```

1  SELECT isbn, title, pageCount, publishedDate, thumbnailUrl, shortDescription, longDescription, status
2  , flatten(authors) AS authors
3  , flatten(categories) AS categories
4  FROM (
5    SELECT isbn
6    , title
7    , pageCount
8    , publishedDate
9    , thumbnailUrl
10   , shortDescription
11   , longDescription
12   , status
13   , authors
14   , categories
15   FROM "Leosan".datasets.books
16   WHERE (isbn != null OR isbn != '')
17 ) tarea_4;

```

The screenshot shows a database query editor interface. The top part displays a SQL query with two `FLATTEN` clauses pointing to the `authors` and `categories` columns. The bottom part shows the results of the query in a table format. The table has columns: `abc_thumbnailUrl`, `abc_shortDescription`, `abc_longDescription`, `abc_status`, `abc_authors`, and `abc_categories`. The data includes rows for books by Frank Ableson and Charlie Collins, with their respective authors and categories listed under each book entry.

Evidencia 2



The screenshot shows a dataset management interface. On the left, there's a sidebar with 'Datasets' and 'Sources'. Under 'Datasets', there are three spaces: 'Leosan', 'Analista 1', 'Analista 2', and 'Analista 3', with 'Analista 3' currently selected. On the right, there's a table titled 'Analista 3' with columns 'Name' and 'Jobs'. A new dataset named 'Books_001' is listed under 'Name' with a value of '0' under 'Jobs'. Below the table, it says 'No data sources yet' and has a button 'Add source'.

Objetivos 2

Tarea de realización de ETL y modelado de datos

Descarga de datos Ejemplo de COVID-19 de EE. UU.

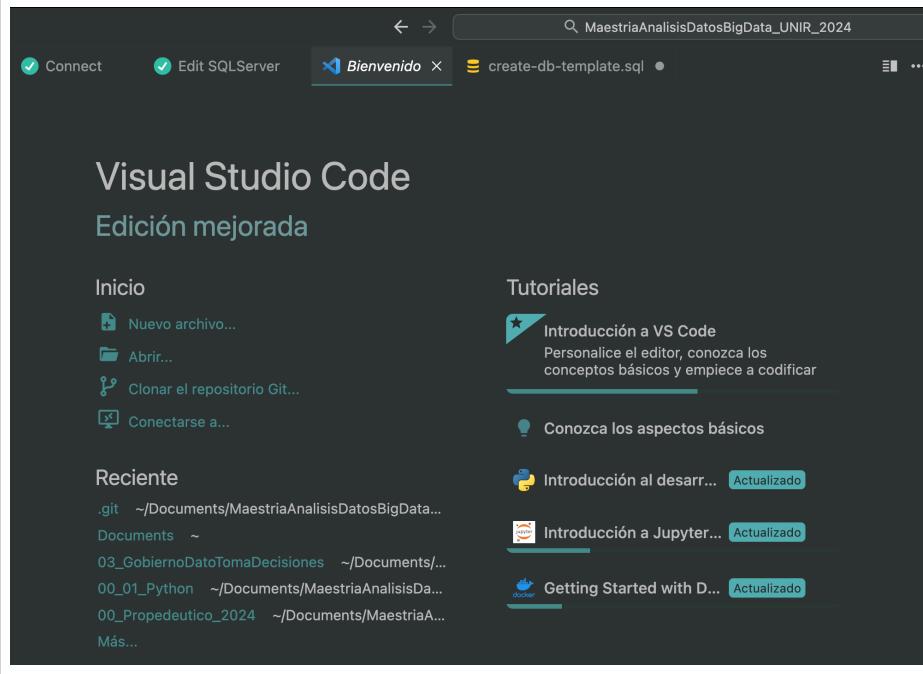
Como ejemplo para la aplicación práctica detallada en el documento, he seleccionado el conjunto de datos proporcionado por el portal oficial https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_daily_reports/03-03-2023.csv Este conjunto de datos será utilizado para aplicar lo que se requiere para esta práctica que es usar Microsoft Visual Studio, Descargar Microsoft SQL Server, Realización de ETL desde Excel a SQL Server y construcción del modelo de datos, solo con una variante usare Python en vez de excel para crear el ETL que cumpla con la practica y el concepto de ETL, que esta extracción , transformación y carga de los datos, usare el dataset en csv, llamado **03-03-2023.csv**

Evidencia 1

Descargar Microsoft Visual Studio

Me siento mas familiarizado con Visual Studio Code, por lo que realizaré esta práctica apoyando en paquetes y desarrollo de código en Python para realizar las operaciones necesarias.

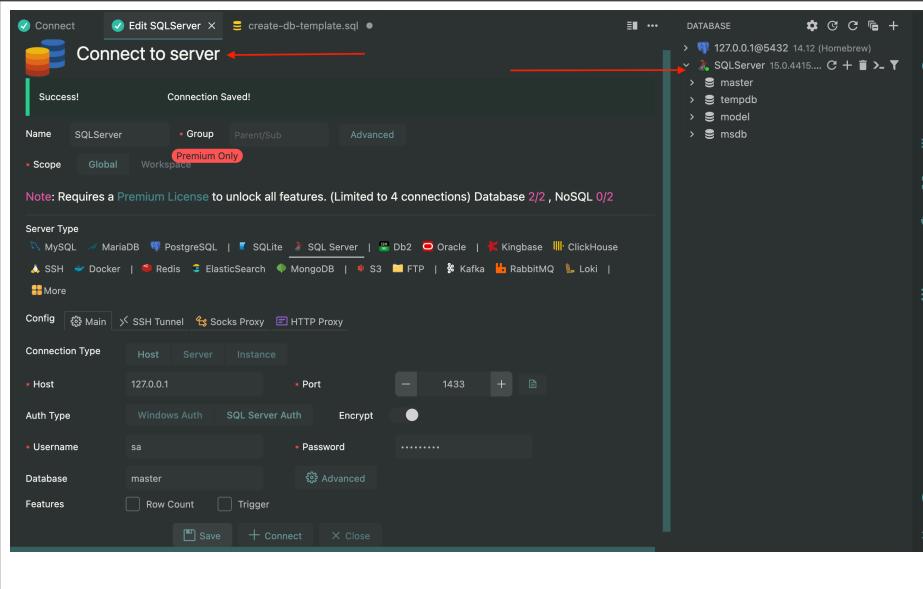
Evidencia 1



Descargar Microsoft SQL Server

Como tengo un equipo Mac, la manera de usar sql server es instalando Docker por lo que dejaré las evidencias necesarias para ir cumpliendo los objetivos de esta práctica.

Evidencia 1



Para alcanzar este objetivo, llevaré a cabo las siguientes actividades, las cuales considero fundamentales, ya que se han validado a lo largo de mi aprendizaje:

Preparación de los datos del archivo dataset COVID_03-03-2023.csv:

- **Limpieza de datos:** se corregirá errores tipográficos, valores nulos, formatos inconsistentes, etc. Usando Python, generaré métodos que pueda validar valores nulos en campo **integer** remplazándolos con valores Cero (0) o en caso de **string** comillas dobles (''), en formato de fechas generar el método para el formato de fecha adecuado.
- **Normalización de datos:** Me aseguraré de que los datos estén en un formato adecuado para la base de datos (por ejemplo, fechas en formato estándar, numero enteros, validación de decimales, cada registro tengo su clave primaria).
- **Diseño de las tablas en SQL Server:** Definiré la estructura de las tablas que me permita generar un Data Warehouse ó un Data Smart apropiado para resolver una serie de incógnitas a problemas planteados.

Desarrollo de Modelado

Genero Bases Datos

The screenshot shows the DBeaver interface with the following details:

- Left Panel:** A large gray area labeled "Genero Bases Datos".
- Top Bar:** Shows "Edit SQLServer" and several file icons.
- SQL Editor:** Displays the command: "create database [practica_dataset]".
- Output Window:** Shows the result: "309ms".
- Right Panel:** A tree view of the database structure.
 - Root: 127.0.0.1@5432 (14.12 Homebrew)
 - SQlServer (15.0.4415.2 Developer Edition)
 - master
 - tempdb
 - model
 - msdb
 - practica_dataset
 - dbo
 - Query
 - Tables (1)
 - t_origen_dataset
 - Views
 - Functions
 - Procedures

Desarrollo de Modelado

Genero Tabla Origen

```

1 -- Active: 1736056347863@127.0.0.1@1433@practica_dataset@dbo > create-table-template.sql
2 CREATE TABLE t_origen_dataset(
3     id int IDENTITY(1,1) primary key,
4     fips INTEGER,
5     admin2 NVARCHAR(255),
6     province_state NVARCHAR(255),
7     country_region NVARCHAR(255),
8     last_update DATE,
9     lat FLOAT,
10    long FLOAT,
11    confirmed INTEGER,
12    deaths INTEGER,
13    recovered INTEGER,
14    active NVARCHAR(255),
15    combined_key NVARCHAR(255),
16    incident_rate FLOAT,
17    case_fatality_ratio FLOAT
18 ); 126ms
19 EXECUTE sp_addextendedproperty N'MS_Description', '[table_comment]', N'user', N'dbo', N'table'
20 t_origen_dataset

```

RESULTS GRID (SHOWING COLUMNS AND DATA TYPES)

	id	fips	admin2	province_state	country_region	last_update
Q	int	int	nvarchar(255)	nvarchar(255)	nvarchar(255)	date

Desarrollo de Modelado

Genero Tabla Hechos

Tabla de Hechos:

- id_caso: Identificador único para cada caso.
- fips: Código FIPS del lugar.
- fecha: Fecha de actualización del dato.
- confirmed: Número de casos confirmados.
- deaths: Número de muertes.
- recovered: Número de recuperados.
- active: Estado activo del caso.
- incident_rate: Tasa de incidencia.
- case_fatality_ratio: Tasa de letalidad.
- fk_pais: Clave foránea a la dimensión País.
- fk_region: Clave foránea a la dimensión Región.

```

1 -- Active: 1736056347863@127.0.0.1@1433@practica_dataset@dbo > create-table-template.sql
2 CREATE TABLE t_hechos(
3     id_caso int IDENTITY(1,1) primary key,
4     fips NVARCHAR(255),
5     fecha DATE,
6     confirmed INTEGER,
7     deaths INTEGER,
8     recovered DATE,
9     active NVARCHAR(255),
10    incident_rate FLOAT,
11    case_fatality_ratio INTEGER,
12    fk_pais INTEGER,
13    fk_region INTEGER
14 ); 58ms
15 EXECUTE sp_addextendedproperty N'MS_Description', '[table_comment]', N'user', N'dbo', N'table'
16 EXECUTE sp_addextendedproperty N'MS_Description', '[column_comment]', N'user', N'dbo', N'table'
17

```

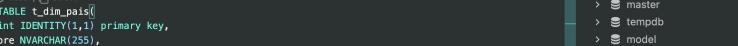
RESULTS GRID (SHOWING COLUMNS AND DATA TYPES)

	id_caso	fips	fecha	confirmed	deaths	recovered	active	incident_rate	case_fatality_ratio	fk_pais	fk_region
Q	int	nvarchar(255)	date	integer	integer	date	nvarchar(255)	float	integer	integer	integer

Desarrollo de Modelado

Genero Tabla Dimensiones

```
Users > leonard > dbclient > storage > 1736056347863@127.0.0.1@1433@practica_dataset@dbo > create-table-template.sql > ...  
1 Active: 1736056347863@127.0.0.1@1433@practica_dataset@dbo SQLServer SQLServer  
2 CREATE TABLE t_dim_pais(  
3     id int IDENTITY(1,1) primary key,  
4     nombre NVARCHAR(255),  
5     acronimo NVARCHAR(255),  
6 )  
7 ► Run  
8 EXECUTE sp_addextendedproperty N'MS_Description', '[table_comment]', N'user', N'dbo', N'table'  
9 ► Run  
EXECUTE sp_addextendedproperty N'MS_Description', '[column_comment]', N'user', N'dbo', N'table  
9
```



```
Users > leonard > .dbclient > storage > 1736056347863@127.0.0.1@1433@practica_dataset@dbo > create-table-template.sql > ...
1 Active: 1736056347863@127.0.0.1@1433@practica_dataset@dbo SqlServer SQLServer
2 CREATE TABLE t_dim_region(
3     id int IDENTITY(1,1) primary key,
4     nombre NVARCHAR(255),
5     acronymo NVARCHAR(10),
6 ); 43ms
7 ► Run
8 EXECUTE sp_addextendedproperty N'MS_Description', '[table_comment]', N'user', N'dbo', N'table'
9 EXECUTE sp_addextendedproperty N'MS_Description', '[column_comment]', N'user', N'dbo', N'table'
9
```

Extracción de los datos del archivo dataset COVID_03-03-2023.csv:

- **Lenguajes de programación:** Considerare utilizar el lenguaje Python para crear scripts personalizados que extraigan los datos.

Desarrollo Script Python para extracción, transformación y carga de datos para generar DW

Scripts

Extracion_ETL_data.py

```
01_Cuatriestre_03_GobiernoDatomaDecisiones > Actividades > Actividad 1 > Resultados > dev.py > Extraccion_ETL_data.py > ...  
import pandas as pd  
from conex import connect_sql  
  
# Realizar la extracción para el modelo real de la predicción  
df = pd.read_csv("U:/Users/leonard/Desktop/Documentos/MaestriaAnalisisDatosBigData_MIR_2024/01_Cuatriestre/03_GobiernoDatomaDecisiones/Actividades/Actividad 1/Resultados/dev_py/03032023.csv")  
  
connection = connect_sql("conectar_sql_server")  
server = "localhost"  
database = "practica_dataset"  
username = "sa"  
password = "admin1"  
port = 1433  
  
connection.cursor()  
  
connection_sql.consulta_sql_server("DELETE FROM t_origin_dataset", connection, 0)  
  
# Inicia la extracción de datos  
print("Extracción Iniciada.")  
total_datos = 0  
for index, row in df.iterrows():  
  
    SQL_QUERY = "INSERT INTO t_origin_dataset (tips, admin2, province_state, country_region, last_update, lat, long, confirmed, deaths, recovered, active, combined_key, incident_rate, case_fatality_ratio) VALUES (%sTIPS%, %sAdm2%, %sProv%, %sCountry%, %sLastUpdate%, %sLat%, %sLong%, %sConfirmed%, %sDeaths%, %sRecovered%, %sActive%, %sKey%, %sRate%, %sCFR%)"  
    connection_sql.insert_data_origin_sql_server(SQL_QUERY, row, connection)  
    total_datos += 1  
  
    print("Extracción Finalizada.")  
    print("")  
    print("Un total de {} datos extraídos exitosamente del dataset.".format(total_datos))  
  
connection_sql.cerrar_sql_server(connection)  
  
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS  
01_Cuatriestre_03_GobiernoDatomaDecisiones > dev.py [5:23] at 2023-03-01 10:00 O  
/Users/leonard/.pyenv/versions/3.10.6/bin/python /Users/leonard/Desktop/Documentos/MaestriaAnalisisDatosBigData_MIR_2024/01_Cuatriestre/03_GobiernoDatomaDecisiones/Actividades/Actividad 1/Resultados/dev_py/Extraccion_ETL_data.py  
Conexión exitosa.  
Extracción Iniciada.  
Extracción Finalizada  
  
Un total de 4016 datos extraídos exitosamente del dataset.  
La conexión ha finalizado.
```

Desarrollo Script Python para extracción, transformación y carga de datos para generar DW

Evidencia 1

```
SELECT * FROM t_origen_dataset LIMIT 100
+-----+-----+-----+-----+-----+-----+-----+-----+
| id   | tips | admin2 | province_state | country_region | last_update | date    | lat    | long   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 88437 | 0    | NaN    | NaN    | Afghanistan    | 2023-03-04 | 33.93911 | 67.709953|
| 88438 | 0    | NaN    | NaN    | Albania        | 2023-03-04 | 41.1533  | 20.1683 |
| 88439 | 0    | NaN    | NaN    | Algeria        | 2023-03-04 | 28.0339  | 1.6596  |
| 88440 | 0    | NaN    | NaN    | Andorra        | 2023-03-04 | 42.5063  | 1.5218  |
| 88441 | 0    | NaN    | NaN    | Angola         | 2023-03-04 | -11.2027 | 17.8739 |
| 88442 | 0    | NaN    | NaN    | Antarctica     | 2023-03-04 | -71.9499 | 23.347   |
| 88443 | 0    | NaN    | NaN    | Antigua and Barbuda | 2023-03-04 | 17.0608  | -61.7964 |
| 88444 | 0    | NaN    | NaN    | Argentina      | 2023-03-04 | -38.4161 | -63.6167 |
| 88445 | 0    | NaN    | NaN    | Armenia        | 2023-03-04 | 40.0691  | 45.0382 |
+-----+-----+-----+-----+-----+-----+-----+-----+
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
NIR_2024/01_Cuatrimestre/03_GobiernoDatoTomaDecisiones/Actividades/Actividad 1/Resultados/dev_py/Extracion_
ETL_data.py"
Conexión exitosa.
Extracción Iniciada.
Extracción Finalizada.

Un total de 4016 datos extraídos exitosamente del dataset.
La conexión ha finalizado.
```

Evidencia 2

```
SELECT * FROM t_dim_pais LIMIT 100
+-----+-----+-----+-----+
| id   | nombre | acronimo |
+-----+-----+-----+
| 202  | Afghanistan | (NULL) |
| 203  | Albania    | (NULL) |
| 204  | Algeria    | (NULL) |
| 205  | Andorra    | (NULL) |
| 206  | Angola     | (NULL) |
| 207  | Antarctica | (NULL) |
| 208  | Antigua and Barbuda | (NULL) |
| 209  | Argentina  | (NULL) |
| 210  | Armenia    | (NULL) |
+-----+-----+-----+
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
ETL_data.py"
Conexión exitosa.
Extracción Iniciada.
Extracción Finalizada.

Un total de 4016 datos extraídos exitosamente del dataset.
La conexión ha finalizado.
```

Desarrollo Script Python para extracción, transformación y carga de datos para generar DW

Evidencia 3

The screenshot shows a data extraction interface. On the left, there's a table titled "t_dim_region" with columns: id (integer), nombre (varchar(255)), and acronimo (varchar(255)). The table contains 599 rows, with the first few rows shown:

		id	nombre	acronimo
	>	2996	Abruzzo	(NULL)
	>	2997	Acre	(NULL)
	>	2998	Adygea Republic	(NULL)
	>	2999	Aguascalientes	(NULL)
	>	3000	Aichi	(NULL)
	>	3001	Akita	(NULL)
	>	3002	Alabama	(NULL)
	>	3003	Alagoas	(NULL)

Below the table, the terminal window shows the following output:

```
SELECT * FROM t_dim_region LIMIT 100
Search Results | Export | Cost: 4ms
Total 599
* id integer
  nombre varchar(255)
  acronimo varchar(255)

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
ETL_data.py"
Conexión exitosa.
Extracción Iniciada.
Extracción Finalizada

Un total de 4016 datos extraídos exitosamente del dataset.
La conexión ha finalizado.
```

Transformación de los datos del archivo dataset COVID 03-03-2023.csv:

- **Limpieza adicional:** considerare si es necesario, realizar una limpieza más profunda de los datos durante la transformación.
- **Cálculos:** considerare si es necesario, realizar cálculos o agregaciones (por ejemplo, calcular totales, promedios).
- **Formateado:** Considerare si es necesario, ajustar los formatos de los datos para que coincidan con los tipos de datos definidos en las tablas de SQL Server.

Carga de los datos del archivo dataset COVID 03-03-2023.csv:

- **Herramientas:** Considerare usar el lenguaje Python para la carga de los datos en una tabla de Origen para luego almacenar nuestros datos en nuestro Datawarehouse diseñado previamente para solucionar algunas necesidades hipotéticas que se consideran para este ejercicio.

Usando Python se generó el script para extraer los datos que necesitamos desde la tabla **t_origen_dataset** a la tabla **t_hechos** que será nuestra Data Smart que usaremos para dar respuesta a la problemática planteada, se realizó así con el propósito de tener datos limpios y adecuados, para este caso contamos con un dataset de 4016 registros de datos crudos y en efecto tenemos un total de 4016 registros de datos limpios en nuestra tabla de **t_hechos**.

Script

```
create-table-template.sql ✘ connexion_sql.py ✘ ETL_data.py M ✘ README.MD
01_Cuatrimestre > 03_GobiernoDatosTomaDecisiones > Actividades > Actividad 1 > Resultados > dev_py > ETL_data.py ...
51
52     total_datos = 0
53
54     for t_hechos in lista_t_hechos: # type: ignore
55         datos_a_insertar = {'fips': t_hechos[0], 'fecha':t_hechos[1], 'confirmed':t_hechos[2], 'deaths':t_hechos[3], 'recovered':t_hechos[4]}
56         print(" ")
57         print(" --- data N°{} ".format(total_datos))
58         print(datos_a_insertar)
59         print(" ")
60         total_datos=total_datos + 1
61         SQL_QUERY = "INSERT INTO t_hechos (fips, fecha, confirmed, deaths, recovered, active, incident_rate, case_fatality_ratio, fk_pais, fk_region) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
62         connexion_sql.insert_data_sql_server(SQL_QUERY, datos_a_insertar, connection)
63
64     connexion_sql.cerrar_sql_server(connection)
65
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
{'fips': 0, 'fecha': datetime.date(2023, 3, 4), 'confirmed': 535, 'deaths': 0, 'recovered': 0, 'active': '0', 'incident_rate': 0, 'case_fatality_ratio': 0.0, 'fk_pais': 2610, 'fk_region': 9920}

--- data N°4013
{'fips': 0, 'fecha': datetime.date(2023, 3, 4), 'confirmed': 11945, 'deaths': 2159, 'recovered': 0, 'active': '0', 'incident_rate': 40.04099354817252, 'case_fatality_ratio': 18.07450816241105, 'fk_pais': 2611, 'fk_region': 9920}

--- data N°4014
{'fips': 0, 'fecha': datetime.date(2023, 3, 4), 'confirmed': 343079, 'deaths': 4057, 'recovered': 0, 'active': '0', 'incident_rate': 1866.18701654856, 'case_fatality_ratio': 1.182526473494443, 'fk_pais': 2612, 'fk_region': 9920}

--- data N°4015
{'fips': 0, 'fecha': datetime.date(2023, 3, 4), 'confirmed': 264127, 'deaths': 5668, 'recovered': 0, 'active': '0', 'incident_rate': 1777.086034264987, 'case_fatality_ratio': 2.145937310374177, 'fk_pais': 2613, 'fk_region': 9920}

La conexión ha finalizado.
```

Evidencia 1

SELECT * FROM t_hechos LIMIT 100									
		fips	fecha	confirmed	deaths	active	incident_rate	case_fatality_ratio	fk_pais
0	>	1	2023-03-04	209362	7896	0	537.8138256649571	4	2413
0	>	2	0	334227	3598	0	11620.925707137396	1	2414
0	>	3	0	271469	6881	0	619.0707938235357	3	2415
0	>	4	0	47875	165	0	61962.07856079726	0	2416
0	>	5	0	105277	1933	0	320.3193012381489	2	2417
0	>	6	0	2023-03-04	11	0	0	0	2418
0	>	7	0	2023-03-04	9106	146	0	9298.668409443671	2
0	>	8	0	2023-03-04	10044125	130463	0	22225.501819200127	1
0	>	9	0	2023-03-04	446819	8721	0	15078.761920253344	2
0	>	10	0	2023-03-04	232619	228	0	54337.53795842093	0
0	>	11	0	2023-03-04	3908129	8493	0	48141.52500615915	0
0	>	12	0	2023-03-04	105021	90	0	42760.99348534202	0
0	>	13	0	2023-03-04	1800236	2783	0	35191.78965887988	0
0	>	14	0	2023-03-04	881911	1322	0	50208.42584685454	0
0	>	15	0	2023-03-04	286897	253	0	53575.53688141923	0
0	>	16	0	2023-03-04	2877260	7338	0	43398.24130077377	0

Con este diseño, deseo poder responder las siguientes problemáticas total de incidencias registradas para el día 03 de Marzo del 2023 COVID 19

Cantidad de países infectados:

```
1 -- Acti
2   ▶ Run | - If you are an premium user, can show definition by hover
3
4 SELECT COUNT(DISTINCT fk_pais) AS num_paises_infectados
5 FROM t_hechos
6 WHERE confirmed > 0;|
```

t_hechos X

Search Results

num_paises_infectados

201

Cuántos muertos por el COVID-19

```
SELECT SUM(deaths) AS total_muertes
FROM t_hechos LIMIT 100
```

Search Results

total_muertes

6877327

Con este diseño, deseo poder responder las siguientes problemáticas total de incidencias registradas para el día 03 de Marzo del 2023 COVID 19

Evolución de casos por país y tiempo:

```
▶ Run | + Tab | JSON | Select
✓ 20 SELECT nombre, fecha, SUM(confirmed) AS casos_totales
21 FROM t_hechos
22 INNER JOIN t_dim_pais ON t_hechos.fk_pais = t_dim_pais.id
23 GROUP BY nombre, fecha
24 ORDER BY nombre, fecha; 8ms
```

Result(RO) ×

Search Results Export

< 1 2 3 > Total 210

nombre	fecha	casos_totales
Afghanistan	2023-03-04	209362
Albania	2023-03-04	334427
Algeria	2023-03-04	271469
Andorra	2023-03-04	47875
Angola	2023-03-04	105277

Tasa de letalidad por región

```
▶ Run | + Tab | JSON | Select
✓ 20 SELECT nombre, fecha, SUM(confirmed) AS casos_totales
21 FROM t_hechos
22 INNER JOIN t_dim_pais ON t_hechos.fk_pais = t_dim_pais.id
23 GROUP BY nombre, fecha
24 ORDER BY nombre, fecha; 8ms
```

Result(RO) ×

Search Results Export

< 1 2 3 > Total 210

nombre	fecha	casos_totales
Afghanistan	2023-03-04	209362
Albania	2023-03-04	334427
Algeria	2023-03-04	271469
Andorra	2023-03-04	47875
Angola	2023-03-04	105277

Carga datos de ejemplos propios de Dremio (Opcional)

Evidencia 1

The screenshot shows the Dremio interface with a query results page. The left sidebar shows datasets like '@Leosan', 'Analista 1', 'Analista 2', 'Analista 3', 'Analista General', 'Empleados', and 'Empleados'. The main area displays a query titled 'Query1' with the following SQL code:

```
1 SELECT * FROM "@Leosan"."Employee Attrition"
```

The results table has 10 columns and 15 rows. The columns are labeled: 'Emp ID', '...', 'satisfaction_level', 'last_evaluation', 'number_project', 'Age', 'Rate', 'Job Satisfaction', 'Job Level', and 'Job Role'. The data looks like this:

Emp ID	...	satisfaction_level	last_evaluation	number_project	Age	Rate	Job Satisfaction	Job Level	Job Role
1		0.38	0.86	5	35	0.50	0.52	0.50	0.50
2		0.8	0.88	7	35	0.50	0.52	0.50	0.50
3		0.11	0.88	5	35	0.50	0.52	0.50	0.50
4		0.72	0.87	5	35	0.50	0.52	0.50	0.50
5		0.37	0.52	2	35	0.50	0.52	0.50	0.50
6		0.41	0.5	2	35	0.50	0.52	0.50	0.50
7		0.1	0.77	6	35	0.50	0.52	0.50	0.50
8		0.92	0.85	5	35	0.50	0.52	0.50	0.50
9		0.89	1	5	35	0.50	0.52	0.50	0.50
10		0.42	0.53	2	35	0.50	0.52	0.50	0.50
11		0.45	0.54	2	35	0.50	0.52	0.50	0.50

On the right side, there's an 'Overview' panel for the dataset 'Employee Attrition' and a 'Columns (10)' panel listing the columns with their descriptions.

Ficheros Parquet

- Parquet es una forma muy eficiente de guardar datos cuando tienes muchos de ellos y necesitas buscar cosas específicas rápidamente. Es como tener una biblioteca súper organizada para tus datos.

Un ejemplo para explicarlo

Imagina que tenemos una biblioteca tradicional (base de datos por filas), buscas un libro buscando en cada estante (fila) hasta encontrarlo. Parquet es como una biblioteca súper organizada donde todos los libros del mismo género (columna) están juntos en una misma sección. Esto hace que encontrar un libro específico sea mucho más rápido.

Diferencia entre Json y Parquet

Una diferencia de los formatos tradicionales como CSV o JSON que almacenan datos por filas, Parquet organiza los datos por columnas.

Otra diferencia es la compresión eficiente, al almacenar datos de forma columnar, Parquet puede aplicar técnicas de compresión más eficientes, reduciendo significativamente el tamaño de los archivos.

Se puede diferenciar por usar el procesamiento paralelo porque al ser columnar facilita el procesamiento paralelo de los datos, lo que es ideal para tareas de big data.