



DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
Inteligencia de negocios 202220 – Proyecto 2
PROFESORA: Haydemar Núñez

Nombres	Apellidos	Código	Login
María Sofía	Álvarez López	201729031	ms.alvarezl
Brenda Catalina	Barahona Pinilla	201812721	bc.barahona
Álvaro Daniel	Plata Márquez	201820098	ad.plata

Proyecto 1 – Etapa 2: Automatización analítica de textos

El objetivo de este proyecto es analizar indicadores socioeconómicos y dimensiones municipales de Colombia. El proyecto busca aprovechar los datos abiertos para generar conocimiento útil que aporte a diferentes tipos de actores, desde el ciudadano normal hasta el experto en temáticas allí trabajadas, pasando por actores como periodistas, entre otros. De igual manera, quieren ampliar la gama de análisis que se pueden observar en la página web de la iniciativa

1. Identificación de necesidades analíticas

A partir de entrevistas realizadas a personas relacionadas con el proyecto que están trabajando, los datos compartidos y la retroalimentación a la consultoría realizada en semestres anteriores, se identificaron y documentaron algunos requerimientos analíticos, utilizando la matriz de requerimientos de negocio, los temas analíticos, análisis requeridos, procesos de negocio, fuentes de datos y datos requeridos siguiendo la metodología Kimball.

A continuación, se presentan los temas analíticos propuestos, con sus respectivos análisis requeridos/inferidos y las justificaciones respectivas, que se pensaron para este proyecto. Para encontrar la tabla completa, remítase al archivo EntregaAnalisisRequeridos.xlsx

Tema analítico: Conflicto armado.

Tema analítico	Análisis requeridos o inferidos	Categoría del análisis - Tablero de control, análisis OLAP, Minería de datos	Procesos de negocio	Fuentes de datos y datos
Conflicto armado	Análisis de homicidios * año * departamento * causa * municipio * sexo	Análisis OLAP	Registro de afectaciones en el conflicto armado	Excel ConflictoArmado
	Análisis de número de personas secuestradas * año * departamento * municipio	Análisis OLAP	Registro de demografía y población	Excel DemografíaYPoblación
	Visualización de las afectaciones del conflicto con base en el análisis 1 y 2	Tablero de Control	Registro de afectaciones en el conflicto armado y Registro de demografía y población	Datamart sobre el conflicto armado

Una de los temas de mayor interés en Colombia es el análisis del conflicto armado, ya que ha sido una de las crisis más grandes que ha existido en la historia de América del Sur. Siendo una guerra asimétrica, resulta conveniente analizar la cantidad de homicidios (por año - porque hay años en que el impacto del conflicto fue más fuerte - , por departamento - porque hay departamentos en que el conflicto es más fuerte y tiene mayor incidencia - , por causa - porque resulta conveniente entender las razones principales por las cuales ocurren los homicidios en el conflicto armado, con el fin de intentar mitigarlo - y por sexo - porque es fundamental entender la incidencia del conflicto en este ámbito). De manera similar, el número de personas secuestradas puede analizarse por indicadores análogos. Esto también es fundamental, porque hubo años y regiones del país en los cuales el número de secuestros alcanzó cifras altamente preocupantes.

Una de las mejores formas de visualizar esto es, por ejemplo, utilizando mapas de burbujas. Estos son una excelente herramienta de visualizar los sitios con mayor presencia de un fenómeno en particular. El mapa mostrado a continuación, por ejemplo, ilustra la proporción de surfistas en el mundo:

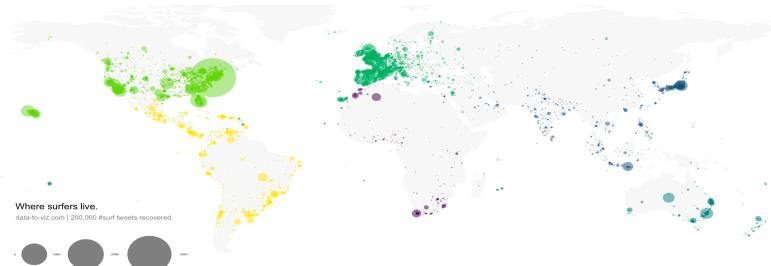


Figura 1: Número de surfistas en el mundo.

Note que para poder realizar adecuadamente estos diagramas, es necesario que los indicadores estén normalizados a la población del país.

Tema analítico: Educación.

Educación	Análisis de cobertura neta * año * mes * departamento * municipio	Análisis OLAP	Registro de datos sobre educación	Excel Educación
	Visualización del análisis indicadores educativos con base en el análisis 1	Tablero de Control	Registro de datos sobre educación	Datamart sobre la educación

El segundo tema analítico de interés es la educación. Siendo la educación uno de los principales motores de los países en vía de desarrollo para lograr la igualdad, resulta de especial interés analizar la cobertura neta de educación a nivel nacional. Es común que las áreas rurales tengan desarrollos educativos mucho más elementales y precarios que las grandes ciudades: no sólo porque acceder a estos municipios es más difícil; sino que, dado el bajo desarrollo social y económico, muy pocos profesores están dispuestos a mudarse a esas áreas. Lograr ampliar la cobertura neta de educación a nivel nacional resulta crucial puesto que es un indicador fundamental del avance económico y social del país. De manera similar al caso del conflicto armado, una visualización adecuada sería un mapa de burbujas, o un gráfico de barras corriente que evidencie el porcentaje de cobertura (normalizado a la población) para cada uno de los municipios/departamentos en cuestión.

Tema analítico: Desempleo.

Desempleo	Análisis del indicador de desempleo * año * municipio	Análisis OLAP	Registro de mediciones de desempleo	Excel MedicionedeDesempleoDepartamental
	Análisis del indicador de desempleo * año * departamento	Análisis OLAP	Registro de mediciones de desempleo	Excel MedicionedeDesempleoMunicipal
	Visualización de análisis indicadores de desempleo municipal	Tablero de Control	Registro de mediciones de desempleo	Datamart sobre el conflicto armado
	Visualización de análisis indicadores de desempleo para departamentos	Tablero de Control	Registro de mediciones de desempleo	Datamart sobre desempleo

Gran parte de la población colombiana es bastante joven. Muchas veces, al graduarse de la Universidad o del colegio, o incluso aquellos que desertan de la educación, se ven enfrentados a una época de desempleo prolongada mientras logran conseguir un empleo. La dinámica del desempleo varía cada año (pues también depende de factores macroeconómicos externos como la migración forzada), y por municipio (dependiendo de la edad de la población, el nivel educativo, el número de plazas de empleo disponibles y la cantidad de desplazados que el municipio/departamento recibe a diario). De manera similar al caso del conflicto armado, una visualización adecuada sería un mapa de burbujas, o una tabla cuadrada de área proporcional, como la que se presenta en la figura 2.

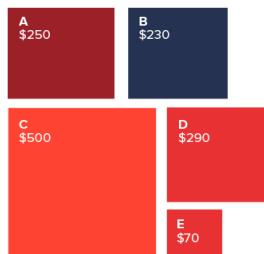


Figura 2: Tabla cuadrada de área proporcional.

Esta tabla podría mostrar (normalizado a la población) para cada municipio o departamento, la tasa de desempleo. Es interesante pues rápidamente el usuario puede notar cuál es el departamento/municipio de mayor incidencia (en el caso de la figura 2, sería la variable correspondiente al cuadrado C).

Tema analítico: Salud.

Salud	Análisis del nivel de satisfacción del sistema de salud * año * departamento * municipio	Análisis OLAP	Registro de datos sobre la salud	Excel Salud
	Análisis de población afiliada a regímenes * año * departamento * municipio * tipo de régimen	Análisis OLAP	Registro de demografía y población	Excel DemografiaYPoblacion
	Visualización de análisis indicadores de salud con base en el análisis 1 y 2	Tablero de Control	Registro de datos sobre la salud y Registro de demografía y población	Datamart sobre salud

Como país en vía de desarrollo, el indicador de la salud en Colombia es fundamental para entender la evolución del país hacia una cobertura igualitaria de los servicios básicos y asegurar una mayor expectativa de vida (y disminuir tasas como muertes de niños y jóvenes por enfermedades prevenibles también). Para ello, hay tres análisis que resulta conveniente analizar en este ámbito. El primero es el análisis del nivel de satisfacción del sistema de salud por año, departamento y municipio. Resulta fundamental que los usuarios sientan que están bien atendidos y que la salud en su municipio y departamento es favorable. Un buen indicador del manejo de la salud es la percepción de los ciudadanos.

Otro análisis que resulta relevante es el número de habitantes (normalizado a la población del departamento) que están afiliados a regímenes de salud, así como el tipo de régimen. Como punto

básico, el gobierno debe asegurar que los usuarios tengan un régimen de salud (así sea subsidiado). No obstante, debido a las personas sin hogar, a los desplazados forzados o inmigrantes ilegales, siempre va a haber un porcentaje de la población que no esté afiliada. También puede dejar de haber personas afiliadas si los recursos no son suficientes para financiar la salud de todos los habitantes. Analizar esto es fundamental para determinar el avance económico y social del país. Para poder visualizar esta información, es necesario utilizar tableros de control que permitan mostrar (por ejemplo, en un mapa), la satisfacción promedio de los usuarios por departamento, así como la cantidad de usuarios afiliados a un régimen (o cierto régimen de salud). Asimismo, la visualización de la tabla cuadrada de área proporcional también sería valiosa en este ámbito.

Tema analítico: Vivienda y servicios públicos

Vivienda y servicios públicos	Análisis del indicador de cobertura * año * departamento * municipio * tipo de servicio	Análisis OLAP	Registro de vivienda y servicios públicos	Excel ViviendayServiciosPublicos
	Análisis de penetración de banda ancha * año * departamento * municipio	Análisis OLAP	Registro de demografía y población	Excel DemografiaYPoblacion
	Visualización de análisis indicadores de servicios públicos con base en el análisis 1 y 2	Tablero de Control	Registro de vivienda y servicios públicos y Registro de demografía y población	Datamart sobre Vivienda y servicios públicos

Es importante analizar a su vez la cobertura de vivienda y servicios básicos de la población. Lo ideal es que todos los habitantes tengan acceso a estas facilidades; no obstante, debido a que aún queda un gran trecho por recorrer en este sentido, aún un gran porcentaje de la población no tiene acceso a vivienda, servicios básicos o los dos. Es importante normalizar este indicador a la población del departamento/municipio para que sea confiable.

Por otro lado, la cobertura de internet es fundamental. Es crucial que el país esté interconectado debido a la globalización a la que nos enfrentamos hoy en día. En aras de evaluar la conexión de todo el país al internet (y por ende, a un mayor y más amplio acceso a la información), la penetración de banda es un indicador fundamental. Para analizar y revisar ambos indicadores, es posible usarlo utilizando un mapa de burbujas o de calor (como el de la figura 3), dependiendo de la cobertura normalizada de vivienda y servicios públicos, o de penetración normalizada de banda ancha, para cada departamento/municipio.

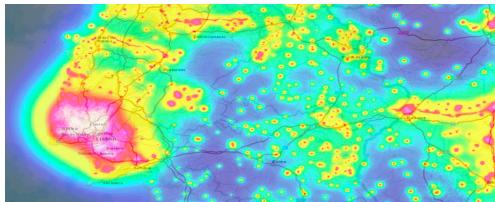


Figura 3: Ejemplo de mapa de calor.

2. Modelado de DataMarts

A partir de las fuentes de datos anexas a esta entrega y los requerimientos analíticos identificados en el literal anterior, se realizó un modelo dimensional que representa los tipos de requerimientos del cliente.

Con esto en mente, lo primero que se hizo fue elaborar el modelo multidimensional, con los nombres de atributos, llaves primarias, llaves foráneas y roles claramente definidos. El modelo obtenido fue el de la figura 4.

Para la creación de este modelo dimensional, seguimos el siguiente proceso:

1. *Seleccionar el proceso de negocio:* Para los requerimientos que deseamos realizar, deseamos colocar gran enfoque en los siguientes procesos de negocio: registro de Homicidios, registro de personas secuestradas, registro de la proporción de satisfacción de salud, registro de afiliados al sistema de salud y registro de la medición de desempleo departamental.
2. *Seleccionar granularidad:* En general, el nivel de granularidad de la mayoría de procesos es parecido. Escogimos este nivel de granularidad para satisfacer los requerimientos de análisis que identificamos anteriormente. Al analizar los datos nos damos cuenta que en estos podemos realizar una distinción de algunos indicadores según la ubicación y la temporalidad.
 - a. Registro de homicidios:
Nivel de granularidad: a nivel de especificación de temporalidad, demografía, ubicación y causa
 - b. Registro de personas secuestradas:
Nivel de granularidad: A nivel de especificación de temporalidad y de ubicación
 - c. Registro de desempleo:
Nivel de granularidad: A nivel de especificación de temporalidad y de ubicación

- d. **Registro de Satisfacción de salud:**
Nivel de granularidad: A nivel de especificación de temporalidad, de ubicación y nivel de satisfacción
 - e. **Registro de afiliado:**
Nivel de granularidad: A nivel de especificación de temporalidad , de ubicación y tipo de afiliación.
3. *Identificar dimensiones:* Para todos los procesos mencionados anteriormente decidimos realizar las vistas por fecha (mes y/o año) y lugar (departamento). Para los procesos de registro de homicidios, registro de la proporción de satisfacción de salud y registro de afiliados al sistema de salud se colocaron más dimensiones:
- a. Proceso de registro de homicidios: Dimensión de Causa de homicidios y Dimensión Sexo
 - b. Proceso de registro de la proporción de satisfacción de salud: Dimensión nivel de satisfacción (Muy satisfecho, Muy insatisfecho, satisfecho, etc.)
 - c. Proceso de registro de afiliados al sistema de salud: Dimensión tipo régimen, el cual especifica a que sistema de salud se hace referencia, por ejemplo Afiliado al régimen contributivo, afiliado al régimen subsidiado, afiliado a regímenes especiales y afiliados al SGSSS
4. *Identificar los hechos:*
- a. **Registro de homicidios:**
Porcentaje de homicidios (medida no aditiva). Esta es una medida no aditiva puesto que lo único que podemos obtener es un recuento de las filas, no es posible realizar cálculos adicionales sobre porcentajes.
 - b. **Registro de personas secuestradas:**
Porcentaje de personas secuestradas (medida no aditiva). Esta es una medida no aditiva puesto que lo único que podemos obtener es un recuento de las filas, no es posible realizar cálculos adicionales sobre porcentajes.
 - c. **Registro de desempleo:**
Porcentaje de personas que están desempleadas (medida no aditiva). Esta es una medida no aditiva puesto que lo único que podemos obtener es un recuento de las filas, no es posible realizar cálculos adicionales sobre porcentajes.
 - d. **Registro de Satisfacción de salud:**
Porcentaje de personas que tienen un determinado nivel de satisfacción con el sistema de salud: (medida no aditiva). Esta es una medida no aditiva puesto que lo único que podemos obtener es un recuento de las filas, no es posible realizar cálculos adicionales sobre porcentajes.
 - e. **Registro de afiliado:**
Porcentaje de personas afiliadas al régimen de salud (medida no aditiva). Esta es una medida no aditiva puesto que lo único que podemos obtener es un recuento de las filas, no es posible realizar cálculos adicionales sobre porcentajes.

Ahora, para cada uno de los atributos de las dimensiones planteadas, elegimos el tipo de manejo de historia, o si es de variación lenta, como se ve a continuación:

1. **Dimensión tipo afiliación:** El primer atributo es la llave primaria. El atributo de tipo régimen no necesita manejar historia. Basta simplemente con manejar el cambio y ya, puesto lo que importa al negocio es tener cuál es el régimen actual de salud de los individuos. El histórico de este atributo no es necesario.
2. **Dimensión nivel satisfacción:** El primer atributo es la llave primaria. Para el atributo de nivel satisfacción vale la pena llevar un manejo de historia tipo 3. Con ello, la entidad agrega un nuevo atributo para conocer cuál fue la anterior calificación de los usuarios y ver si, en los últimos meses, el servicio ha mejorado o empeorado.
3. **Dimensión ubicación:** Esta dimensión tiene una llave primaria, un código de departamento y un código de municipio. Es muy poco probable que alguno de los dos últimos atributos cambie en el tiempo. Por tanto, no se maneja historia en esta dimensión.
4. **Dimensión sexo:** Esta es una dimensión que tiene llave primaria y sexo. Aunque no muchas personas cambian su sexo, es posible que esto ocurra. No debe manejarse historia para este atributo, pues en ningún momento se busca hacer análisis sobre cambios de sexo.
5. **Dimensión causa homicidio:** Esta es una dimensión que tiene llave primaria y causa. La causa de un homicidio puede variar, dependiendo de los análisis de medicina legal, y por

tanto puede ser importante llevar un récord de ello, porque puede ser información relevante para el negocio y para las investigaciones legales. Para ello, se puede usar un manejo de historia tipo 2, tal que haya un inicio y fin de una causa de homicidio determinada asignada y se pueda reconstruir la historia para análisis judiciales posteriores.

6. **Dimensión Fecha:** Utilizada por las otras dimensiones y la tabla de hechos. No requiere manejo de historia porque es más una dimensión que no será utilizada por sí sola ni maneja información del negocio.

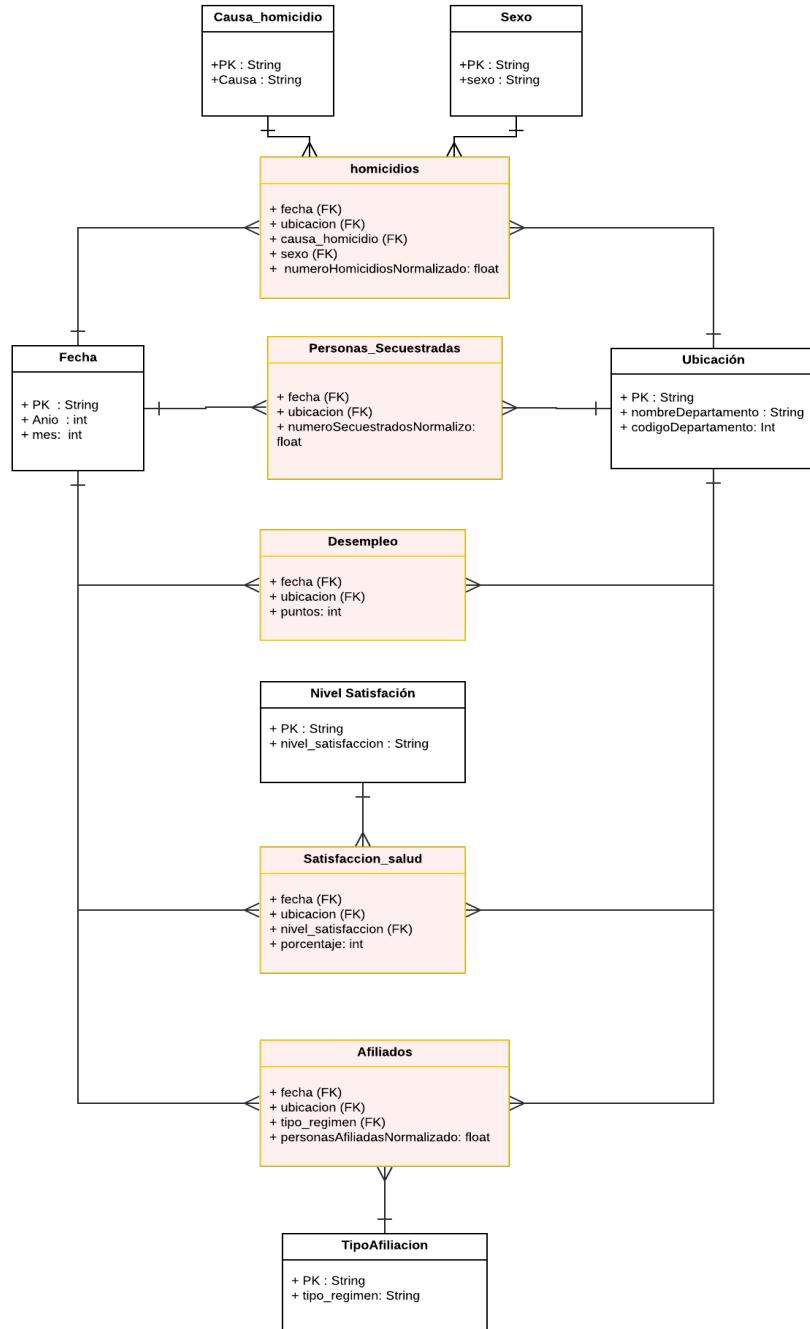


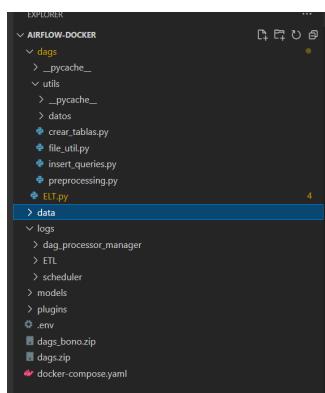
Figura 4: Modelo propuesto para el proyecto.

3. Entendimiento de los datos y proceso ETL

La sección a continuación describe el entendimiento y perfilamiento de los archivos de datos brindados por infraestructura visible:

Para la creación del proceso ETL en el proyecto seguimos los siguientes pasos:

- a. **Creación de la estructura del proyecto en la máquina virtual asignada:** Creamos la estructura del proyecto basándonos en la misma estructura del laboratorio 5. La estructura es la siguiente:

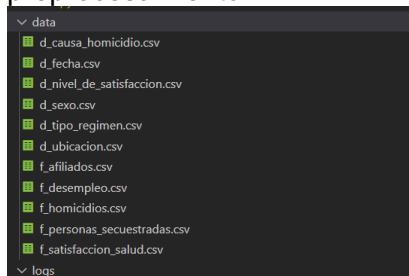


Los archivos en esta estructura cumplen las mismas funciones que en el laboratorio 5. Explicaremos más adelante el contenido y función de los más relevantes.

- b. Creamos el servidor de Airflow con la misma imagen de Docker utilizada en el laboratorio 5. Este servidor se crea con la ejecución de los siguientes comandos desde la carpeta raíz del proyecto:

 - docker-compose up airflow-init
 - docker-compose up

c. Agregamos en la carpeta /data del proyecto los archivos .csv resultados del preprocesamiento.



- d. Creamos el archivo crear-tablas.py, que se encargará de ser el script de creación de tablas del modelo multidimensional

```

1  CREATE TABLE IF NOT EXISTS `tbl_crm` (
2      `id` int(11) NOT NULL AUTO_INCREMENT,
3      `name` varchar(100) NOT NULL,
4      `email` varchar(100) NOT NULL,
5      `password` varchar(100) NOT NULL,
6      `mobile` varchar(100) NOT NULL,
7      `address` varchar(100) NOT NULL,
8      `city` varchar(100) NOT NULL,
9      `state` varchar(100) NOT NULL,
10     `zip` varchar(100) NOT NULL,
11     `country` varchar(100) NOT NULL,
12     `status` tinyint(1) NOT NULL,
13     `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
14     `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
15     PRIMARY KEY (`id`)
16 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17
18  CREATE TABLE IF NOT EXISTS `tbl_crm_user` (
19      `id` int(11) NOT NULL AUTO_INCREMENT,
20      `name` varchar(100) NOT NULL,
21      `email` varchar(100) NOT NULL,
22      `password` varchar(100) NOT NULL,
23      `mobile` varchar(100) NOT NULL,
24      `address` varchar(100) NOT NULL,
25      `city` varchar(100) NOT NULL,
26      `state` varchar(100) NOT NULL,
27      `zip` varchar(100) NOT NULL,
28      `country` varchar(100) NOT NULL,
29      `status` tinyint(1) NOT NULL,
30      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
31      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
32      PRIMARY KEY (`id`)
33 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
34
35  CREATE TABLE IF NOT EXISTS `tbl_crm_department` (
36      `id` int(11) NOT NULL AUTO_INCREMENT,
37      `name` varchar(100) NOT NULL,
38      `status` tinyint(1) NOT NULL,
39      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
40      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
41      PRIMARY KEY (`id`)
42 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
43
44  CREATE TABLE IF NOT EXISTS `tbl_crm_designation` (
45      `id` int(11) NOT NULL AUTO_INCREMENT,
46      `name` varchar(100) NOT NULL,
47      `status` tinyint(1) NOT NULL,
48      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
49      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
50      PRIMARY KEY (`id`)
51 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
52
53  CREATE TABLE IF NOT EXISTS `tbl_crm_branch` (
54      `id` int(11) NOT NULL AUTO_INCREMENT,
55      `name` varchar(100) NOT NULL,
56      `status` tinyint(1) NOT NULL,
57      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
58      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
59      PRIMARY KEY (`id`)
60 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
61
62  CREATE TABLE IF NOT EXISTS `tbl_crm_satisfaction` (
63      `id` int(11) NOT NULL AUTO_INCREMENT,
64      `satisfaction` tinyint(1) NOT NULL,
65      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
66      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
67      PRIMARY KEY (`id`)
68 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
69
70  CREATE TABLE IF NOT EXISTS `tbl_crm_satisfaction_detail` (
71      `id` int(11) NOT NULL AUTO_INCREMENT,
72      `satisfaction_id` int(11) NOT NULL,
73      `satisfaction_detail` tinyint(1) NOT NULL,
74      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
75      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
76      PRIMARY KEY (`id`)
77 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
78
79  CREATE TABLE IF NOT EXISTS `tbl_crm_satisfaction_detail_value` (
80      `id` int(11) NOT NULL AUTO_INCREMENT,
81      `satisfaction_detail_id` int(11) NOT NULL,
82      `value` varchar(100) NOT NULL,
83      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
84      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
85      PRIMARY KEY (`id`)
86 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
87
88  CREATE TABLE IF NOT EXISTS `tbl_crm_time` (
89      `id` int(11) NOT NULL AUTO_INCREMENT,
90      `time` time NOT NULL,
91      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
92      `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
93      PRIMARY KEY (`id`)
94 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
95
96  CREATE TABLE IF NOT EXISTS `tbl_crm_time_region` (
97      `id` int(11) NOT NULL AUTO_INCREMENT,
98      `region` varchar(100) NOT NULL,
99      `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
100     `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
101     PRIMARY KEY (`id`)
102 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

- e. Creamos el archivo file_util.py, donde definimos las funciones de escritura y lectura de csv:

```

dags > utils > file_util.py
1 import pandas as pd
2
3 def cargar_datos(nombre):
4     df = pd.read_csv("/opt/airflow/data/" + nombre + ".csv", sep=',', encoding='utf-8', encoding_errors='ignore', index_col=False)
5     return df
6
7 def guardar_datos(df, nombre):
8     df.to_csv('/opt/airflow/data/' + nombre + '.csv', encoding='utf-8', encoding_errors='ignore', sep=',', index=False)

```

- f. Creamos el archivo insert_queries.py, donde definimos las funciones que transformarán el contenido de archivos csv a queries de inserción en SQL

```

from utils.utils import cargar_datos
# city insertion
def insert_query_ciudad(**kwargs):
    insert = f'INSERT INTO fecha (Fecha_Key,Año,Mes) VALUES '
    insertQuery = ''
    insertQuery += insert
    insertQuery += cargar_datos(kwargs['csv_path'])
    for index, row in dataFrame.iterrows():
        insertQuery += insert + f'({row["fecha_key"]},{row["Año"]},{row["Mes"]});\n'
    return insertQuery

def insert_query_ubicacion(**kwargs):
    insert = f'INSERT INTO ubicacion (Ubicacion_Key,Departamento,Codigo_Entidad) VALUES '
    insertQuery = ''
    insertQuery += insert
    insertQuery += cargar_datos(kwargs['csv_path'])
    for index, row in dataFrame.iterrows():
        insertQuery += insert + f'({row["ubicacion_key"]},{row["Departamento"]},{row["Codigo_Entidad"]});\n'
    return insertQuery

def insert_query_genero(**kwargs):
    insert = f'INSERT INTO genero (Genero_Key,Genero) VALUES '
    insertQuery = ''
    insertQuery += insert
    dataFrame = cargar_datos(kwargs['csv_path'])
    for index, row in dataFrame.iterrows():
        insertQuery += insert + f'({row["sexo_key"]},{row["Sexo"]});\n'
    return insertQuery

def insert_query_causa_homicidio(**kwargs):
    insert = f'INSERT INTO causa_homicidio (Causa_Homicidio_Key,Causa) VALUES '
    insertQuery = ''
    insertQuery += insert
    dataFrame = cargar_datos(kwargs['csv_path'])
    for index, row in dataFrame.iterrows():
        insertQuery += insert + f'({row["causa_homicidio_key"]},{row["causa"]});\n'
    return insertQuery

def insert_query_homicidios(**kwargs):
    insert = f'INSERT INTO homicidios (Homicidio_Key,Fecha_Key,Ubicacion_Key,Causa_Homicidio_Key,Genero_Key,Numeros_Homicidios) VALUES '
    insertQuery = ''
    insertQuery += insert
    dataFrame = cargar_datos(kwargs['csv_path'])
    for index, row in dataFrame.iterrows():
        insertQuery += insert + f'({row["indice"]},{row["fecha_key"]},{row["ubicacion_key"]},{row["causa_homicidio_key"]},{row["sexo_key"]});\n'
    return insertQuery

def insert_query_personas_secuestreadas(**kwargs):
    insert = f'INSERT INTO personas_secuestreadas (Personas_Secuestreadas_Key,Fecha_Key,Ubicacion_Key,Numeros_Secuestreados) VALUES '
    insertQuery = ''

```

- g. Creamos un DAG que consta de 2 pasos que utilizará el operador de Airflow llamado "PostgresOperator" el cual es el encargado de manejar conexiones y procesos relacionados con bases de datos PostgreSQL. Estas tareas consisten en crear las tablas de la base de datos y posteriormente poblarlas, todo esto utilizando los scripts descritos anteriormente.
- h. Para realizar el proceso ETL, definimos el archivo "ELT.py" que contiene el código correspondiente a la implementación del DAG:

```

dag > dag.py ->
1 # 1.0 ejecutar el script
2 !file_util.py
3 file_airflow.providers.postgres_operator import PostgresOperator
4 file_airflow.operators.bash import BashOperator
5 file_airflow.operators.python import PythonOperator
6 file_airflow.operators.python import BranchPythonOperator
7 file_airflow.operators import DatabaseTable
8
9 # Función (1)
10 # Utilizamos la función para cargar los datos
11 # Utilizamos la función para crear las tablas
12 # Utilizamos la función para insertar los datos
13
14 default_args = {
15     'owner': 'DataEng',
16     'start_date': '2023-01-01',
17     'end_date': '2023-01-01',
18     'retries': 0,
19     'retry_delay': timedelta(minutes=1),
20     'depends_on_past': False,
21     'catchup': False
22 }
23
24 # # Tarea 1.0: preprocesamiento
25 # Tarea 1.1: crear la conexión a la base de datos
26 # Tarea 1.2: crear la conexión a la base de datos
27 # Tarea 1.3: crear la conexión a la base de datos
28 # Tarea 1.4: crear la conexión a la base de datos
29
30 # Tarea 1.5: ejecutar el script
31
32 # Tarea 2.0: crear la tabla de fact_order
33 # Tarea 2.1: insertar los datos en la base de datos
34 # Tarea 2.2: insertar los datos en la base de datos
35 # Tarea 2.3: ejecutar el script
36
37 # Tarea 3.0: ejecutar el script
38
39 # Tarea 4.0: ejecutar el script
40
41 # Tarea 5.0: ejecutar el script
42
43 # Tarea 6.0: ejecutar el script
44
45 # Tarea 7.0: ejecutar el script
46
47 # Tarea 8.0: ejecutar el script
48
49 # Tarea 9.0: ejecutar el script
50
51 # Tarea 10.0: ejecutar el script
52
53 # Tarea 11.0: ejecutar el script
54
55 # Tarea 12.0: ejecutar el script
56
57 # Tarea 13.0: ejecutar el script
58
59 # Tarea 14.0: ejecutar el script
60
61 # Tarea 15.0: ejecutar el script
62
63 # Tarea 16.0: ejecutar el script
64
65 # Tarea 17.0: ejecutar el script
66
67 # Tarea 18.0: ejecutar el script
68
69 # Tarea 19.0: ejecutar el script
70
71 # Tarea 20.0: ejecutar el script
72
73 # Tarea 21.0: ejecutar el script
74
75 # Tarea 22.0: ejecutar el script
76
77 # Tarea 23.0: ejecutar el script
78
79 # Tarea 24.0: ejecutar el script
80
81 # Tarea 25.0: ejecutar el script
82
83 # Tarea 26.0: ejecutar el script
84
85 # Tarea 27.0: ejecutar el script
86
87 # Tarea 28.0: ejecutar el script
88
89 # Tarea 29.0: ejecutar el script
90
91 # Tarea 30.0: ejecutar el script
92
93 # Tarea 31.0: ejecutar el script
94
95 # Tarea 32.0: ejecutar el script
96
97 # Tarea 33.0: ejecutar el script
98
99 # Tarea 34.0: ejecutar el script
100
101 # Tarea 35.0: ejecutar el script
102
103 # Tarea 36.0: ejecutar el script
104
105 # Tarea 37.0: ejecutar el script
106
107 # Tarea 38.0: ejecutar el script
108
109 # Tarea 39.0: ejecutar el script
110
111 # Tarea 40.0: ejecutar el script
112
113 # Tarea 41.0: ejecutar el script
114
115 # Tarea 42.0: ejecutar el script
116
117 # Tarea 43.0: ejecutar el script
118
119 # Tarea 44.0: ejecutar el script
120
121 # Tarea 45.0: ejecutar el script
122
123 # Tarea 46.0: ejecutar el script
124
125 # Tarea 47.0: ejecutar el script
126
127 # Tarea 48.0: ejecutar el script
128
129 # Tarea 49.0: ejecutar el script
130
131 # Tarea 50.0: ejecutar el script
132
133 # Tarea 51.0: ejecutar el script
134
135 # Tarea 52.0: ejecutar el script
136
137 # Tarea 53.0: ejecutar el script
138
139 # Tarea 54.0: ejecutar el script
140
141 # Tarea 55.0: ejecutar el script
142
143 # Tarea 56.0: ejecutar el script
144
145 # Tarea 57.0: ejecutar el script
146
147 # Tarea 58.0: ejecutar el script
148
149 # Tarea 59.0: ejecutar el script
150
151 # Tarea 60.0: ejecutar el script
152
153 # Tarea 61.0: ejecutar el script
154
155 # Tarea 62.0: ejecutar el script
156
157 # Tarea 63.0: ejecutar el script
158
159 # Tarea 64.0: ejecutar el script
160
161 # Tarea 65.0: ejecutar el script
162
163 # Tarea 66.0: ejecutar el script
164
165 # Tarea 67.0: ejecutar el script
166
167 # Tarea 68.0: ejecutar el script
168
169 # Tarea 69.0: ejecutar el script
170
171 # Tarea 70.0: ejecutar el script
172
173 # Tarea 71.0: ejecutar el script
174
175 # Tarea 72.0: ejecutar el script
176
177 # Tarea 73.0: ejecutar el script
178
179 # Tarea 74.0: ejecutar el script
180
181 # Tarea 75.0: ejecutar el script
182
183 # Tarea 76.0: ejecutar el script
184
185 # Tarea 77.0: ejecutar el script
186
187 # Tarea 78.0: ejecutar el script
188
189 # Tarea 79.0: ejecutar el script
190
191 # Tarea 80.0: ejecutar el script
192
193 # Tarea 81.0: ejecutar el script
194
195 # Tarea 82.0: ejecutar el script
196
197 # Tarea 83.0: ejecutar el script
198
199 # Tarea 84.0: ejecutar el script
200
201 # Tarea 85.0: ejecutar el script
202
203 # Tarea 86.0: ejecutar el script
204
205 # Tarea 87.0: ejecutar el script
206
207 # Tarea 88.0: ejecutar el script
208
209 # Tarea 89.0: ejecutar el script
210
211 # Tarea 90.0: ejecutar el script
212
213 # Tarea 91.0: ejecutar el script
214
215 # Tarea 92.0: ejecutar el script
216
217 # Tarea 93.0: ejecutar el script
218
219 # Tarea 94.0: ejecutar el script
220
221 # Tarea 95.0: ejecutar el script
222
223 # Tarea 96.0: ejecutar el script
224
225 # Tarea 97.0: ejecutar el script
226
227 # Tarea 98.0: ejecutar el script
228
229 # Tarea 99.0: ejecutar el script
230
231 # Tarea 100.0: ejecutar el script
232
233 # Tarea 101.0: ejecutar el script
234
235 # Tarea 102.0: ejecutar el script
236
237 # Tarea 103.0: ejecutar el script
238
239 # Tarea 104.0: ejecutar el script
240
241 # Tarea 105.0: ejecutar el script
242
243 # Tarea 106.0: ejecutar el script
244
245 # Tarea 107.0: ejecutar el script
246
247 # Tarea 108.0: ejecutar el script
248
249 # Tarea 109.0: ejecutar el script
250
251 # Tarea 110.0: ejecutar el script
252
253 # Tarea 111.0: ejecutar el script
254
255 # Tarea 112.0: ejecutar el script
256
257 # Tarea 113.0: ejecutar el script
258
259 # Tarea 114.0: ejecutar el script
260
261 # Tarea 115.0: ejecutar el script
262
263 # Tarea 116.0: ejecutar el script
264
265 # Tarea 117.0: ejecutar el script
266
267 # Tarea 118.0: ejecutar el script
268
269 # Tarea 119.0: ejecutar el script
270
271 # Tarea 120.0: ejecutar el script
272
273 # Tarea 121.0: ejecutar el script
274
275 # Tarea 122.0: ejecutar el script
276
277 # Tarea 123.0: ejecutar el script
278
279 # Tarea 124.0: ejecutar el script
280
281 # Tarea 125.0: ejecutar el script
282
283 # Tarea 126.0: ejecutar el script
284
285 # Tarea 127.0: ejecutar el script
286
287 # Tarea 128.0: ejecutar el script
288
289 # Tarea 129.0: ejecutar el script
290
291 # Tarea 130.0: ejecutar el script
292
293 # Tarea 131.0: ejecutar el script
294
295 # Tarea 132.0: ejecutar el script
296
297 # Tarea 133.0: ejecutar el script
298
299 # Tarea 134.0: ejecutar el script
300
301 # Tarea 135.0: ejecutar el script
302
303 # Tarea 136.0: ejecutar el script
304
305 # Tarea 137.0: ejecutar el script
306
307 # Tarea 138.0: ejecutar el script
308
309 # Tarea 139.0: ejecutar el script
310
311 # Tarea 140.0: ejecutar el script
312
313 # Tarea 141.0: ejecutar el script
314
315 # Tarea 142.0: ejecutar el script
316
317 # Tarea 143.0: ejecutar el script
318
319 # Tarea 144.0: ejecutar el script
320
321 # Tarea 145.0: ejecutar el script
322
323 # Tarea 146.0: ejecutar el script
324
325 # Tarea 147.0: ejecutar el script
326
327 # Tarea 148.0: ejecutar el script
328
329 # Tarea 149.0: ejecutar el script
330
331 # Tarea 150.0: ejecutar el script
332
333 # Tarea 151.0: ejecutar el script
334
335 # Tarea 152.0: ejecutar el script
336
337 # Tarea 153.0: ejecutar el script
338
339 # Tarea 154.0: ejecutar el script
340
341 # Tarea 155.0: ejecutar el script
342
343 # Tarea 156.0: ejecutar el script
344
345 # Tarea 157.0: ejecutar el script
346
347 # Tarea 158.0: ejecutar el script
348
349 # Tarea 159.0: ejecutar el script
350
351 # Tarea 160.0: ejecutar el script
352
353 # Tarea 161.0: ejecutar el script
354
355 # Tarea 162.0: ejecutar el script
356
357 # Tarea 163.0: ejecutar el script
358
359 # Tarea 164.0: ejecutar el script
360
361 # Tarea 165.0: ejecutar el script
362
363 # Tarea 166.0: ejecutar el script
364
365 # Tarea 167.0: ejecutar el script
366
367 # Tarea 168.0: ejecutar el script
368
369 # Tarea 169.0: ejecutar el script
370
371 # Tarea 170.0: ejecutar el script
372
373 # Tarea 171.0: ejecutar el script
374
375 # Tarea 172.0: ejecutar el script
376
377 # Tarea 173.0: ejecutar el script
378
379 # Tarea 174.0: ejecutar el script
380
381 # Tarea 175.0: ejecutar el script
382
383 # Tarea 176.0: ejecutar el script
384
385 # Tarea 177.0: ejecutar el script
386
387 # Tarea 178.0: ejecutar el script
388
389 # Tarea 179.0: ejecutar el script
390
391 # Tarea 180.0: ejecutar el script
392
393 # Tarea 181.0: ejecutar el script
394
395 # Tarea 182.0: ejecutar el script
396
397 # Tarea 183.0: ejecutar el script
398
399 # Tarea 184.0: ejecutar el script
400
401 # Tarea 185.0: ejecutar el script
402
403 # Tarea 186.0: ejecutar el script
404
405 # Tarea 187.0: ejecutar el script
406
407 # Tarea 188.0: ejecutar el script
408
409 # Tarea 189.0: ejecutar el script
410
411 # Tarea 190.0: ejecutar el script
412
413 # Tarea 191.0: ejecutar el script
414
415 # Tarea 192.0: ejecutar el script
416
417 # Tarea 193.0: ejecutar el script
418
419 # Tarea 194.0: ejecutar el script
420
421 # Tarea 195.0: ejecutar el script
422
423 # Tarea 196.0: ejecutar el script
424
425 # Tarea 197.0: ejecutar el script
426
427 # Tarea 198.0: ejecutar el script
428
429 # Tarea 199.0: ejecutar el script
430
431 # Tarea 200.0: ejecutar el script
432
433 # Tarea 201.0: ejecutar el script
434
435 # Tarea 202.0: ejecutar el script
436
437 # Tarea 203.0: ejecutar el script
438
439 # Tarea 204.0: ejecutar el script
440
441 # Tarea 205.0: ejecutar el script
442
443 # Tarea 206.0: ejecutar el script
444
445 # Tarea 207.0: ejecutar el script
446
447 # Tarea 208.0: ejecutar el script
448
449 # Tarea 209.0: ejecutar el script
450
451 # Tarea 210.0: ejecutar el script
452
453 # Tarea 211.0: ejecutar el script
454
455 # Tarea 212.0: ejecutar el script
456
457 # Tarea 213.0: ejecutar el script
458
459 # Tarea 214.0: ejecutar el script
460
461 # Tarea 215.0: ejecutar el script
462
463 # Tarea 216.0: ejecutar el script
464
465 # Tarea 217.0: ejecutar el script
466
467 # Tarea 218.0: ejecutar el script
468
469 # Tarea 219.0: ejecutar el script
470
471 # Tarea 220.0: ejecutar el script
472
473 # Tarea 221.0: ejecutar el script
474
475 # Tarea 222.0: ejecutar el script
476
477 # Tarea 223.0: ejecutar el script
478
479 # Tarea 224.0: ejecutar el script
480
481 # Tarea 225.0: ejecutar el script
482
483 # Tarea 226.0: ejecutar el script
484
485 # Tarea 227.0: ejecutar el script
486
487 # Tarea 228.0: ejecutar el script
488
489 # Tarea 229.0: ejecutar el script
490
491 # Tarea 230.0: ejecutar el script
492
493 # Tarea 231.0: ejecutar el script
494
495 # Tarea 232.0: ejecutar el script
496
497 # Tarea 233.0: ejecutar el script
498
499 # Tarea 234.0: ejecutar el script
500
501 # Tarea 235.0: ejecutar el script
502
503 # Tarea 236.0: ejecutar el script
504
505 # Tarea 237.0: ejecutar el script
506
507 # Tarea 238.0: ejecutar el script
508
509 # Tarea 239.0: ejecutar el script
510
511 # Tarea 240.0: ejecutar el script
512
513 # Tarea 241.0: ejecutar el script
514
515 # Tarea 242.0: ejecutar el script
516
517 # Tarea 243.0: ejecutar el script
518
519 # Tarea 244.0: ejecutar el script
520
521 # Tarea 245.0: ejecutar el script
522
523 # Tarea 246.0: ejecutar el script
524
525 # Tarea 247.0: ejecutar el script
526
527 # Tarea 248.0: ejecutar el script
528
529 # Tarea 249.0: ejecutar el script
530
531 # Tarea 250.0: ejecutar el script
532
533 # Tarea 251.0: ejecutar el script
534
535 # Tarea 252.0: ejecutar el script
536
537 # Tarea 253.0: ejecutar el script
538
539 # Tarea 254.0: ejecutar el script
540
541 # Tarea 255.0: ejecutar el script
542
543 # Tarea 256.0: ejecutar el script
544
545 # Tarea 257.0: ejecutar el script
546
547 # Tarea 258.0: ejecutar el script
548
549 # Tarea 259.0: ejecutar el script
550
551 # Tarea 260.0: ejecutar el script
552
553 # Tarea 261.0: ejecutar el script
554
555 # Tarea 262.0: ejecutar el script
556
557 # Tarea 263.0: ejecutar el script
558
559 # Tarea 264.0: ejecutar el script
560
561 # Tarea 265.0: ejecutar el script
562
563 # Tarea 266.0: ejecutar el script
564
565 # Tarea 267.0: ejecutar el script
566
567 # Tarea 268.0: ejecutar el script
568
569 # Tarea 269.0: ejecutar el script
570
571 # Tarea 270.0: ejecutar el script
572
573 # Tarea 271.0: ejecutar el script
574
575 # Tarea 272.0: ejecutar el script
576
577 # Tarea 273.0: ejecutar el script
578
579 # Tarea 274.0: ejecutar el script
580
581 # Tarea 275.0: ejecutar el script
582
583 # Tarea 276.0: ejecutar el script
584
585 # Tarea 277.0: ejecutar el script
586
587 # Tarea 278.0: ejecutar el script
588
589 # Tarea 279.0: ejecutar el script
590
591 # Tarea 280.0: ejecutar el script
592
593 # Tarea 281.0: ejecutar el script
594
595 # Tarea 282.0: ejecutar el script
596
597 # Tarea 283.0: ejecutar el script
598
599 # Tarea 284.0: ejecutar el script
600
601 # Tarea 285.0: ejecutar el script
602
603 # Tarea 286.0: ejecutar el script
604
605 # Tarea 287.0: ejecutar el script
606
607 # Tarea 288.0: ejecutar el script
608
609 # Tarea 289.0: ejecutar el script
610
611 # Tarea 290.0: ejecutar el script
612
613 # Tarea 291.0: ejecutar el script
614
615 # Tarea 292.0: ejecutar el script
616
617 # Tarea 293.0: ejecutar el script
618
619 # Tarea 294.0: ejecutar el script
620
621 # Tarea 295.0: ejecutar el script
622
623 # Tarea 296.0: ejecutar el script
624
625 # Tarea 297.0: ejecutar el script
626
627 # Tarea 298.0: ejecutar el script
628
629 # Tarea 299.0: ejecutar el script
630
631 # Tarea 300.0: ejecutar el script
632
633 # Tarea 301.0: ejecutar el script
634
635 # Tarea 302.0: ejecutar el script
636
637 # Tarea 303.0: ejecutar el script
638
639 # Tarea 304.0: ejecutar el script
640
641 # Tarea 305.0: ejecutar el script
642
643 # Tarea 306.0: ejecutar el script
644
645 # Tarea 307.0: ejecutar el script
646
647 # Tarea 308.0: ejecutar el script
648
649 # Tarea 309.0: ejecutar el script
650
651 # Tarea 310.0: ejecutar el script
652
653 # Tarea 311.0: ejecutar el script
654
655 # Tarea 312.0: ejecutar el script
656
657 # Tarea 313.0: ejecutar el script
658
659 # Tarea 314.0: ejecutar el script
660
661 # Tarea 315.0: ejecutar el script
662
663 # Tarea 316.0: ejecutar el script
664
665 # Tarea 317.0: ejecutar el script
666
667 # Tarea 318.0: ejecutar el script
668
669 # Tarea 319.0: ejecutar el script
670
671 # Tarea 320.0: ejecutar el script
672
673 # Tarea 321.0: ejecutar el script
674
675 # Tarea 322.0: ejecutar el script
676
677 # Tarea 323.0: ejecutar el script
678
679 # Tarea 324.0: ejecutar el script
680
681 # Tarea 325.0: ejecutar el script
682
683 # Tarea 326.0: ejecutar el script
684
685 # Tarea 327.0: ejecutar el script
686
687 # Tarea 328.0: ejecutar el script
688
689 # Tarea 329.0: ejecutar el script
690
691 # Tarea 330.0: ejecutar el script
692
693 # Tarea 331.0: ejecutar el script
694
695 # Tarea 332.0: ejecutar el script
696
697 # Tarea 333.0: ejecutar el script
698
699 # Tarea 334.0: ejecutar el script
700
701 # Tarea 335.0: ejecutar el script
702
703 # Tarea 336.0: ejecutar el script
704
705 # Tarea 337.0: ejecutar el script
706
707 # Tarea 338.0: ejecutar el script
708
709 # Tarea 339.0: ejecutar el script
710
711 # Tarea 340.0: ejecutar el script
712
713 # Tarea 341.0: ejecutar el script
714
715 # Tarea 342.0: ejecutar el script
716
717 # Tarea 343.0: ejecutar el script
718
719 # Tarea 344.0: ejecutar el script
720
721 # Tarea 345.0: ejecutar el script
722
723 # Tarea 346.0: ejecutar el script
724
725 # Tarea 347.0: ejecutar el script
726
727 # Tarea 348.0: ejecutar el script
728
729 # Tarea 349.0: ejecutar el script
730
731 # Tarea 350.0: ejecutar el script
732
733 # Tarea 351.0: ejecutar el script
734
735 # Tarea 352.0: ejecutar el script
736
737 # Tarea 353.0: ejecutar el script
738
739 # Tarea 354.0: ejecutar el script
740
741 # Tarea 355.0: ejecutar el script
742
743 # Tarea 356.0: ejecutar el script
744
745 # Tarea 357.0: ejecutar el script
746
747 # Tarea 358.0: ejecutar el script
748
749 # Tarea 359.0: ejecutar el script
750
751 # Tarea 360.0: ejecutar el script
752
753 # Tarea 361.0: ejecutar el script
754
755 # Tarea 362.0: ejecutar el script
756
757 # Tarea 363.0: ejecutar el script
758
759 # Tarea 364.0: ejecutar el script
760
761 # Tarea 365.0: ejecutar el script
762
763 # Tarea 366.0: ejecutar el script
764
765 # Tarea 367.0: ejecutar el script
766
767 # Tarea 368.0: ejecutar el script
768
769 # Tarea 369.0: ejecutar el script
770
771 # Tarea 370.0: ejecutar el script
772
773 # Tarea 371.0: ejecutar el script
774
775 # Tarea 372.0: ejecutar el script
776
777 # Tarea 373.0: ejecutar el script
778
779 # Tarea 374.0: ejecutar el script
780
781 # Tarea 375.0: ejecutar el script
782
783 # Tarea 376.0: ejecutar el script
784
785 # Tarea 377.0: ejecutar el script
786
787 # Tarea 378.0: ejecutar el script
788
789 # Tarea 379.0: ejecutar el script
790
791 # Tarea 380.0: ejecutar el script
792
793 # Tarea 381.0: ejecutar el script
794
795 # Tarea 382.0: ejecutar el script
796
797 # Tarea 383.0: ejecutar el script
798
799 # Tarea 384.0: ejecutar el script
800
801 # Tarea 385.0: ejecutar el script
802
803 # Tarea 386.0: ejecutar el script
804
805 # Tarea 387.0: ejecutar el script
806
807 # Tarea 388.0: ejecutar el script
808
809 # Tarea 389.0: ejecutar el script
810
811 # Tarea 390.0: ejecutar el script
812
813 # Tarea 391.0: ejecutar el script
814
815 # Tarea 392.0: ejecutar el script
816
817 # Tarea 393.0: ejecutar el script
818
819 # Tarea 394.0: ejecutar el script
820
821 # Tarea 395.0: ejecutar el script
822
823 # Tarea 396.0: ejecutar el script
824
825 # Tarea 397.0: ejecutar el script
826
827 # Tarea 398.0: ejecutar el script
828
829 # Tarea 399.0: ejecutar el script
830
831 # Tarea 400.0: ejecutar el script
832
833 # Tarea 401.0: ejecutar el script
834
835 # Tarea 402.0: ejecutar el script
836
837 # Tarea 403.0: ejecutar el script
838
839 # Tarea 404.0: ejecutar el script
840
841 # Tarea 405.0: ejecutar el script
842
843 # Tarea 406.0: ejecutar el script
844
845 # Tarea 407.0: ejecutar el script
846
847 # Tarea 408.0: ejecutar el script
848
849 # Tarea 409.0: ejecutar el script
850
851 # Tarea 410.0: ejecutar el script
852
853 # Tarea 411.0: ejecutar el script
854
855 # Tarea 412.0: ejecutar el script
856
857 # Tarea 413.0: ejecutar el script
858
859 # Tarea 414.0: ejecutar el script
860
861 # Tarea 415.0: ejecutar el script
862
863 # Tarea 416.0: ejecutar el script
864
865 # Tarea 417.0: ejecutar el script
866
867 # Tarea 418.0: ejecutar el script
868
869 # Tarea 419.0: ejecutar el script
870
871 # Tarea 420.0: ejecutar el script
872
873 # Tarea 421.0: ejecutar el script
874
875 # Tarea 422.0: ejecutar el script
876
877 # Tarea 423.0: ejecutar el script
878
879 # Tarea 424.0: ejecutar el script
880
881 # Tarea 425.0: ejecutar el script
882
883 # Tarea 426.0: ejecutar el script
884
885 # Tarea 427.0: ejecutar el script
886
887 # Tarea 428.0: ejecutar el script
888
889 # Tarea 429.0: ejecutar el script
890
891 # Tarea 430.0: ejecutar el script
892
893 # Tarea 431.0: ejecutar el script
894
895 # Tarea 432.0: ejecutar el script
896
897 # Tarea 433.0: ejecutar el script
898
899 # Tarea 434.0: ejecutar el script
900
901 # Tarea 435.0: ejecutar el script
902
903 # Tarea 436.0: ejecutar el script
904
905 # Tarea 437.0: ejecutar el script
906
907 # Tarea 438.0: ejecutar el script
908
909 # Tarea 439.0: ejecutar el script
910
911 # Tarea 440.0: ejecutar el script
912
913 # Tarea 441.0: ejecutar el script
914
915 # Tarea 442.0: ejecutar el script
916
917 #
```

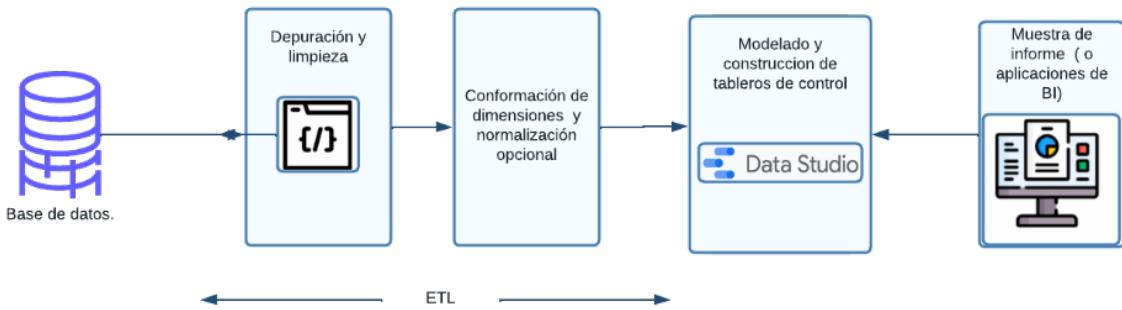


Figura: Arquitectura de solución propuesta para este proyecto.

La arquitectura es sencilla. Primero, partimos de las fuentes de datos: los archivos .csv proporcionados por infraestructura visible. A continuación, estos datos pasan por un proceso que prepara los datos para subirlos a la base de datos, denominado ETL, el cual fue implementado en *Python* utilizando Airflow, como se describió en la sección anterior. A continuación, hay una capa de presentación, en donde los datos están disponibles para realizar una consulta. En este caso, la capa de presentación corresponde la base de datos *PostgreSQL*, que es el paso del tercer cuadrado de la figura 5. Por último, se encuentra el modelado y construcción de los tableros de control, los cuales pueden ser accedidos a partir del servicio Data Studio de Google.

Implementación de los tableros de control :

La idea de esta parte del proyecto era utilizar GoogleDataStudio conectándose a la base de datos donde se alojan los datos que representan el modelo multidimensional propuesto, para la elaboración de los tableros de control.

Puede encontrar los tableros de control desarrollados en el siguiente link <https://datastudio.google.com/reporting/c775cee7-acb5-427e-98e3-51ef65a4c058>

El tablero de control se realizó en GoogleDataStudio. En este puede ver en la parte izquierda que están las opciones para realizar los filtros correspondientes. En sí, aca hay dos tableros de control, en el primero se muestran los hechos del proceso de registro del conflicto armado y en el segundo se muestran los hechos del proceso de registro de Salud. A continuación, se muestra una imagen del tablero de control. Para pasar al segundo tablero de control, debe darle clic en el número 2 de la barra vertical izquierda.



Figura 6: Implementación del tablero de control.

El usuario puede elegir uno de los indicadores que desea visualizar, haciendo click en la barra del indicador, como se muestra en la figura a continuación:



Como se observa en la figura 6, también puede discriminar por departamento y fecha, para limitar su análisis dependiendo de ciertos factores y/o características.

Recién se ingresa al tablero, la tabla de pie muestra indicadores de homicidios (Según causa) para todo el territorio nacional. La gráfica de abajo presenta la evolución poblacional y el mapa de la derecha permite una visualización, por departamento, donde el tamaño de la burbuja representa la incidencia del conflicto en dicho territorio.

El usuario puede discriminar por departamento también si hace click en alguna de las burbujas, con lo que obtendrá los resultados específicos para cada departamento, como se muestra en la figura a continuación para Vichada:



Por defecto, se presentan los valores en todos los años si el usuario no selecciona un año o rango de años en particular.

El tablero de salud es bastante similar: en la tabla de Pie, se presentan el porcentaje nacional de afiliados o satisfacción (dependiendo de la métrica elegida por el usuario en la opción de la izquierda), tal que los departamentos de mayores afiliados y satisfacción tendrán una mayor cobertura en el pie:



De manera similar, los usuarios pueden discriminar por departamento, indicador y fecha en el tablero de salud. Para explorar cada una de estas opciones, remítase al link de arriba.

8. Descripción de las actividades realizadas:

- Identificación de las necesidades analíticas: Brenda, Alvaro, Sofía
- Modelado del datamarts: Brenda, Alvaro, Sofía
- Entender las fuentes de datos: Brenda, Alvaro, Sofía (5 horas)
- Diseñar e implementar el proceso de ETL: Brenda, Alvaro, Sofía (2 horas)
- Proponer arquitectura de decisión: Brenda, Alvaro (1 hora)
- Implementar los tableros de control: Brenda, Alvaro (6 horas)
- Preparacion del video: Brenda, Sofia, Alvaro (1:15 horas)
- Realización del video: Sofia (1 hora)
- Realización del informe: Brenda, Sofía, Alvaro (2 horas)

- Repartición de puntos:

Si tuviéramos que repartir 100 puntos entre los integrantes del grupo, repartiríamos 33.33 a cada uno.

Referencias:

https://www.data-to-viz.com/graph/IMG/Surfer_bubble.png