



Proyecto Final

Product Development

José Armando Barrios León - 20007192

Lilian Rebeca Carrera Lemus - 20008077

Ruben Dario González Monterroso - 20003314

Vidal Baez Fortunato - 20002076

Table of Contents

| | |
|--|--------------|
| 1. Introduccion | 1 |
| 2. Estructura Utilizada | 2,3,4 |
| 2.1.1 Estructura y Folders..... | 4,5 |
| 3. Airflow..... | 6 |
| Dags..... | 7 |
| 4. Connections..... | 8 |
| 5. ETL Process..... | 8 |
| 6. Comprobando DB..... | 9 |
| 7. Dashboard..... | 10 |

1. Introduccion












Instrucciones: se busca evaluar todos los conocimientos adquiridos por los estudiantes a lo largo del semestre.

Covid 19 Dashboard muestras las informaciones sobre los casos de covid-19 contagios, recuperaciones y muertes dado por pais y region para analizar los datos estadiscamente.

2. Estructura utilizada

2.1 docker-compose

Contiene todas las configuraciones del proyectos tales como la creacion de postgres 9.6 , asi como las configuraciones del web server , DB y los volúmenes(rutas):

| Name | Date modified | Type | Size |
|---|---------------------|---------------|------|
|  docker-compose.yml | 12/11/2020 6:49 PM | YML File | 2 KB |
|  requirements.txt | 11/23/2020 6:46 PM | Text Document | 1 KB |
|  .gitignore | 11/23/2020 6:46 PM | Text Document | 2 KB |
|  Dockerfile | 11/23/2020 6:46 PM | File | 3 KB |
|  README.md | 11/23/2020 6:46 PM | MD File | 1 KB |
|  monitor | 12/11/2020 9:51 PM | File folder | |
|  dags | 12/11/2020 7:30 PM | File folder | |
|  data transformation | 12/11/2020 7:29 PM | File folder | |
|  script | 12/11/2020 7:12 PM | File folder | |
|  dashboard | 12/11/2020 1:03 AM | File folder | |
|  config | 12/11/2020 12:38 AM | File folder | |

- /dags:/usr/local/airflow/dags: donde se encuentras los dags del proyectos
- /home/airflow/monitor: en esta ruta se encuentras los archivos que son cargados a la base de datos.(después de ser cargados los archivos son borrados del folder).

```

version: '3.7'
services:
  postgres:
    image: postgres:9.6
    environment:
      - POSTGRES_USER=airflow
      - POSTGRES_PASSWORD=airflow
      - POSTGRES_DB=airflow
    logging:
      options:
        max-size: 10m
        max-file: "3"
  webserver:
    build: .
    restart: always
    depends_on:
      - postgres
    environment:
      - LOAD_EX=n
      - EXECUTOR=Local
    logging:
      options:
        max-size: 10m
        max-file: "3"
    volumes:
      - ./dags:/usr/local/airflow/dags
      - ./monitor:/home/airflow/monitor
    ports:
      - "8080:8080"
    command: webserver
    healthcheck:
      test: ["CMD-SHELL", "[ -f /usr/local/airflow/airflow-webserver.pid ]"]
      interval: 30s
      timeout: 30s
      retries: 3
  db:
    image: mysql:5.7
    volumes:
      #- ./db_data:/var/lib/mysql
      - ./script/casos_covid.sql:/docker-entrypoint-initdb.d/1.sql
    restart: always
    ports:
      - 3308:3306
    environment:
      MYSQL_ROOT_PASSWORD: pruebal23
      MYSQL_DATABASE: Mapamundi
      MYSQL_USER: usrmapa
      MYSQL_PASSWORD: pruebal23

```

- build: . Esto indica que el web server utilizar el Docker file localizado en el folder para su creación.
- El host db contiene la imagen de la base de datos mysql a utilizar para ser llenada con los datasets

```
dashboard:
  depends_on:
    - db
  build: ./dashboard
  ports:
    - 3838:3838
  volumes:
    - ./dashboard/Code:/srv/shiny-server/
    - ./dashboard/logs:/var/log/shiny-server/
```

- El host dashboard contiene la configuración del shiny server, este es cargado mediante un Docker file dentro de la carpeta dashboard. Se le mapean unas carpetas en donde se deben agregar los archivos del dashboard y los archivos log que genera.

```
CREATE DATABASE IF NOT EXISTS Mapamundi;
USE Mapamundi;

/* -----
   Creación de estructura de la base de datos
   ----- */








# Información de casos acumulados
CREATE TABLE IF NOT EXISTS casos_covid(
  Pais          VARCHAR(50),
  Estado        VARCHAR(50),
  Latitud       FLOAT,
  Longitud      FLOAT,
  Fecha         DATETIME,
  Confirmados   INT,
  Recuperados   INT,
  Fallecidos    INT
);

TRUNCATE TABLE casos_covid;
```




- /docker-entrypoint-initdb.d/1.sql: este es el script que se utilizar para crear las tablas en la base de datos.

2.1.1 Estructuras y folders






Instructions: En este renglon se describe cada folders que se utilizar en el proyecto.

| Name | Date modified | Type | Size |
|---|--------------------|-------------|------|
|  __pycache__ | 12/11/2020 9:40 PM | File folder | |
|  __init__.py | 11/23/2020 6:46 PM | PY File | 1 KB |
|  consolidate_dag.py | 11/23/2020 6:46 PM | PY File | 2 KB |
|  new_sales.py | 12/11/2020 1:52 AM | PY File | 4 KB |
|  sales_dag.py | 11/23/2020 6:46 PM | PY File | 3 KB |
|  task_test.py | 11/23/2020 6:46 PM | PY File | 2 KB |
|  Transformacion_data.py | 12/11/2020 9:40 PM | PY File | 6 KB |




- Dags: Contienes todos los Directed Acyclic Graph que utiliza el Proyecto.

| Name | Date modified | Type | Size |
|---|---------------------|-------------|------|
|  Code | 12/11/2020 1:12 AM | File folder | |
|  logs | 12/11/2020 11:19 PM | File folder | |
|  Dockerfile | 12/11/2020 11:12 PM | File | 1 KB |

- Dashboard: En esta carpeta contiene todos los tableros que se utilizan en el proyecto.

| | | | |
|---|--------------------|----------------------|--------|
|  data transformation.zip | 12/11/2020 5:57 PM | Archivo WinRAR Z... | 276 KB |
|  data_transformation.py | 12/11/2020 7:28 PM | PY File | 2 KB |
|  time_series_covid19_confirmed_global.csv | 11/30/2020 6:57 PM | Microsoft Excel C... | 351 KB |
|  time_series_covid19_deaths_global.csv | 11/30/2020 6:57 PM | Microsoft Excel C... | 246 KB |
|  time_series_covid19_recovered_global.csv | 11/30/2020 6:57 PM | Microsoft Excel C... | 314 KB |

- Data Transformacion: Contiene los archivos *.cvs que será cargado a la BD.
- Monitor: En este folder se colocan los archivos que será cargado por airflow, este folder tiene un sensor que verificar si los archivos se encuentra y lo elimina luego de ser cargados

| | | | |
|--|--------------------|--------------------|------|
|  casos_covid.sql | 12/11/2020 7:12 PM | JetBrains DataGrip | 1 KB |
|  entrypoint.sh | 11/23/2020 6:46 PM | Shell Script | 5 KB |
|  schema.sql | 11/23/2020 6:46 PM | JetBrains DataGrip | 1 KB |

- Script: Contiene todos los script a utilizar para la creación de la base de datos.

3. Airflow

Al ejecutar la instrucción Docker-compose up se levantan todos los servicios, se observa lo siguiente:

```
[2020-12-12 04:47:06,656] [settings.py:253] INFO - settings.configure_orm(): Using pool settings. pool_size=5, max_overflow=10, pool_recycle=1800, pid=192
```

```
[2020-12-12 04:47:06,676] [settings.py:253] INFO - settings.configure_orm(): Using pool settings. pool_size=5, max_overflow=10, pool_recycle=1800, pid=1
```

```
[2020-12-12 04:47:07,090] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:07,952] [(scheduler_job.py:1344)] INFO - Starting the scheduler
```

```
[2020-12-12 04:47:07,952] [(scheduler_job.py:1352)] INFO - Running execute loop for ~ 1 seconds
```

```
[2020-12-12 04:47:07,953] [(scheduler_job.py:1353)] INFO - Processing each file at most ~ 1 times
```

```
[2020-12-12 04:47:07,953] [(scheduler_job.py:1356)] INFO - Searching for files in /usr/local/airflow/dags
```

```
[2020-12-12 04:47:08,061] [(scheduler_job.py:1353)] INFO - There are 5 files in /usr/local/airflow/dags
```

```
[2020-12-12 04:47:08,205] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:08,206] [(dagbag.py:403)] INFO - Filling up the DagBag from /usr/local/airflow/dags
```

```
[2020-12-12 04:47:08,420] [(scheduler_job.py:1409)] INFO - Resetting orphaned tasks for active dag runs
```

```
[2020-12-12 04:47:08,468] [(dag_processing.py:550)] INFO - Launched DagFileProcessorManager with pid: 156
```

```
[2020-12-12 04:47:08,535] [(__main__.py:53)] INFO - Configured default timezone <timezone> (UTC)
```

```
[2020-12-12 04:47:08,554] [settings.py:253] INFO - settings.configure_orm(): Using pool settings. pool_size=5, max_overflow=10, pool_recycle=1800, pid=156
```

```
Running the Gunicorn Server with:
```

```
Workers: 4 sync
```

```
Host: 0.0.0.0:8080
```

```
Timeout: 120
```

```
Logfiles: - -
```

```
[2020-12-12 04:47:10,866] [settings.py:253] INFO - settings.configure_orm(): Using pool settings. pool_size=5, max_overflow=10, pool_recycle=1800, pid=192
```

```
[2020-12-12 04:47:10 +0000] [192] [INFO] Starting gunicorn 19.10.0
```

```
[2020-12-12 04:47:10 +0000] [192] [INFO] Listening at http://0.0.0.0:8080 (192)
```

```
[2020-12-12 04:47:10 +0000] [192] [INFO] Using worker: sync
```

```
[2020-12-12 04:47:10 +0000] [213] [INFO] Booting worker with pid: 213
```

```
[2020-12-12 04:47:10 +0000] [214] [INFO] Booting worker with pid: 214
```

```
[2020-12-12 04:47:10 +0000] [215] [INFO] Booting worker with pid: 215
```

```
[2020-12-12 04:47:11 +0000] [216] [INFO] Booting worker with pid: 216
```

```
[2020-12-12 04:47:12,340] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:12,343] [(dagbag.py:403)] INFO - Filling up the DagBag from /usr/local/airflow/dags
```

```
[2020-12-12 04:47:12,358] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:12,360] [(dagbag.py:403)] INFO - Filling up the DagBag from /usr/local/airflow/dags
```

```
[2020-12-12 04:47:12,493] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:12,495] [(dagbag.py:403)] INFO - Filling up the DagBag from /usr/local/airflow/dags
```

```
[2020-12-12 04:47:12,696] [(__init__.py:51)] INFO - Using executor LocalExecutor
```

```
[2020-12-12 04:47:12,698] [(dagbag.py:403)] INFO - Filling up the DagBag from /usr/local/airflow/dags
```

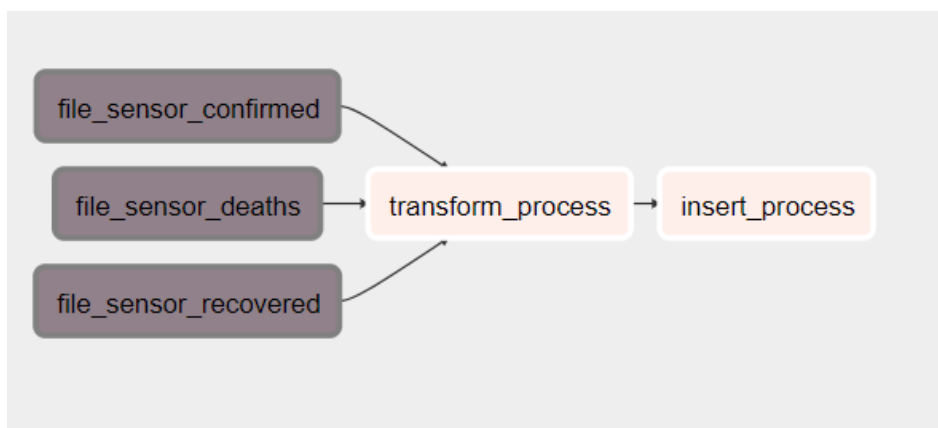

Dags

| Airflow | | | | | | | |
|--|-------------------------|----------|------------------------|--------------|----------|----------|-------|
| DAGs Data Profiling Browse Admin Docs About 2020-12-12 05:45:59 UTC | | | | | | | |
| DAGs | | | | | | | |
| Search: <input type="text"/> | | | | | | | |
| | DAG | Schedule | Owner | Recent Tasks | Last Run | DAG Runs | Links |
| | consolidate_data_dag | @5*1*1* | airflow, obed espinosa | | | | |
| | new_sales_dag | @5*1*1* | airflow | | | | |
| | sales_ingestion_dag | @5*1*1* | obed espinosa | | | | |
| | task_test | @5*1*1* | obed espinosa | | | | |
| | Transformacion_data_dag | @5*1*1* | airflow | | | | |

Showing 1 to 5 of 5 entries

Hide Paused DAGs

El proyecto usa el dag llamado transformación_data_dag, el cual esta organizado en tres bloque el primero realiza las lecturas de los archivos *.csv al momento del sensor detecta los archivo en el carpeta(monitor), el segundo bloque se encarga de unificar los archivos y realizar las transformaciones necesarias y el ultimo paso realizar un insert de manera línea a la base de datos que se encuentra en MySQL.



4. Connection:

 This Tab contiene todas las conexiones utilizadas e el Proyecto.

Airflow_db: esta conexión se utiliza para cargar los datos a la base de datos de Mysql mediante airflow.

The screenshot shows the 'Connection [edit]' form in the Airflow web interface. The form is titled 'Connection [edit]' and has tabs for 'List', 'Create', and 'Edit'. The 'Edit' tab is active. The form contains the following fields:

- Conn Id *: airflow_db
- Conn Type: MySQL
- Host: db
- Schema: Mapamundi
- Login: usmapa
- Password: *****
- Port: 3306
- Extra: (empty)

At the bottom of the form, there are four buttons: 'Save' (green), 'Save and Add Another' (light blue), 'Save and Continue Editing' (light blue), and 'Cancel' (red).

Fs_default: esta conexión se utilizar en la parte de los sensores que monitorea los archivos.

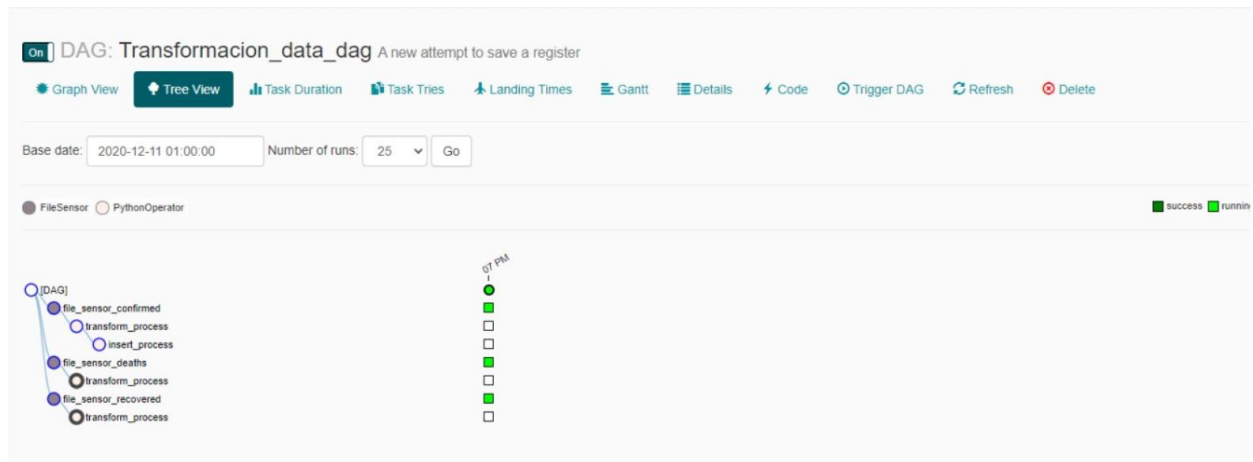
The screenshot shows the 'Connection [edit]' form in the Airflow web interface for a file system connection. The form is titled 'Connection [edit]' and has tabs for 'List', 'Create', and 'Edit'. The 'Edit' tab is active. The form contains the following fields:

- Conn Id *: fs_default
- Conn Type: File (path)
- Host: (empty)
- Schema: (empty)
- Login: usmapa
- Password: *****
- Port: (empty)
- Extra: {"path": "/home/airflow/monitor/"}

At the bottom of the form, there are four buttons: 'Save' (green), 'Save and Add Another' (light blue), 'Save and Continue Editing' (light blue), and 'Cancel' (red).

5. ETL Process

En este tab se realizan las cargas hacia MySQL.



6. Validacion de Datos:

Se verificar que los datos están en la BD.

✓ `select * from casos_covid;`

Output Mapamundi.casos_covid

| | Pais | Estado | Latitud | Longitud | Fecha | Confirmados | Recuperados | Fallecidos |
|----|---------------------|------------------------------|----------|----------|---------------------|-------------|-------------|------------|
| 1 | Afghanistan | Afghanistan | 33.9391 | 67.71 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 2 | Albania | Albania | 41.1533 | 20.1683 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 3 | Algeria | Algeria | 28.0339 | 1.6596 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 4 | Andorra | Andorra | 42.5063 | 1.5218 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 5 | Angola | Angola | -11.2027 | 17.8739 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 6 | Antigua and Barbuda | Antigua and Barbuda | 17.0608 | -61.7964 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 7 | Argentina | Argentina | -38.4161 | -63.6167 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 8 | Armenia | Armenia | 40.0691 | 45.0382 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 9 | Australia | Australian Capital Territory | -35.4735 | 149.012 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 10 | Australia | New South Wales | -33.8688 | 151.209 | 2020-01-22 00:00:00 | 0 | 0 | 0 |
| 11 | Australia | Northern Territory | -12.4634 | 130.846 | 2020-01-22 00:00:00 | 0 | 0 | 0 |

Para la visualización de los datos utilizamos shiny en R studio con las siguientes librerías:

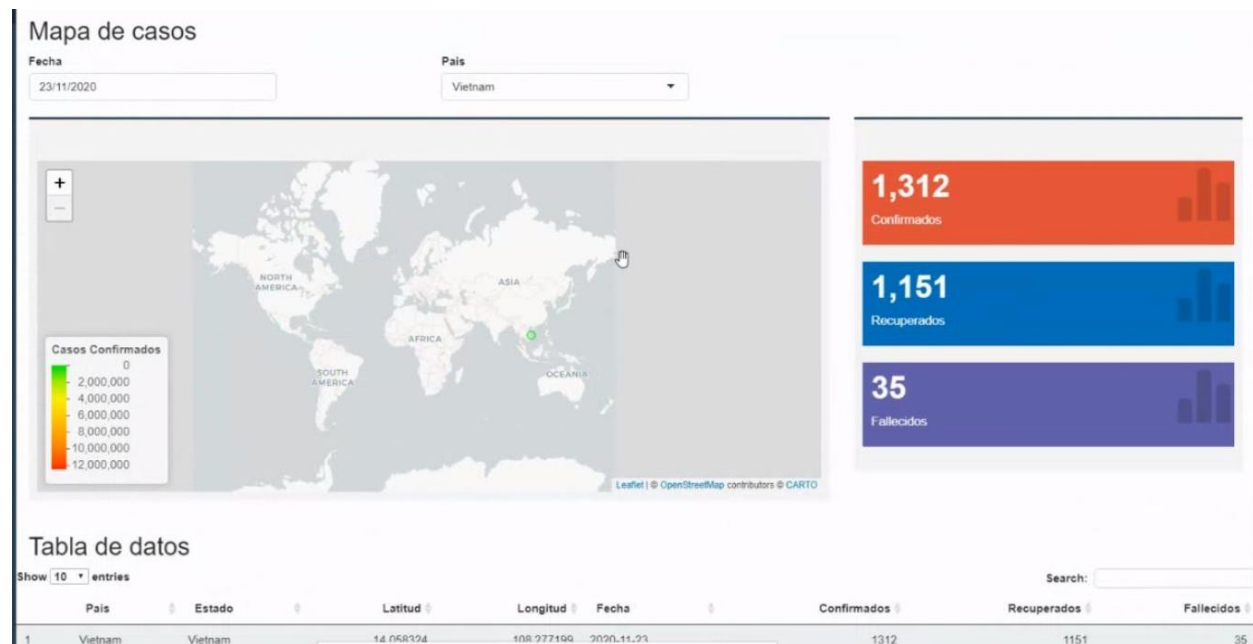
library(leaflet): devuelve un widget de mapa Folleto, que almacena una lista de objetos que se pueden modificar o actualizar más tarde.

library(shinythemes): Contiene algunas several Bootstrap themes de <http://bootswatch.com/>, las cuales son almacenadas para ser usada con Shiny.

library(dashboardthemes): Permite la creación manual de temas y logotipos para ser utilizados en aplicaciones creadas utilizando el paquete 'shinydashboard'

library(shinydashboard): hace que sea fácil de usar Shiny para crear dashboards como estos: Dashboard. Dashboard bus.

Covid-19 Map: el grafico muestras el mapa de los casos de covid-19 totalizados y el mismo actualiza las informaciones según lo seleccionado en los parámetros de fecha y Pais.



También contiene una tabla donde se visualiza la totalidad de los casos por país.

Grafica de Linea : en esta grafica de líneas se muestra las el incrementos de los casos asi como la cantidad de recuperados y fallecidos por dia desde que comenzó la pandemia(Covid-19).

