

DESEMBARQUE DE CAPTURAS MARÍTIMAS EN ARGENTINA

ETL para la carga de datasets en una base de datos
PostgreSQL

- **Presentación basada en repositorio:**
<https://github.com/JuanCapuano/Postgres-etl.git>



GRUPO N° 7

- Falvo, Santiago
- Santiesteban, Augusto
- Bernardi, Manuel
- Tosco, Santiago
- Capuano, Juan
- Cocchi, Santiago
- Banegas, Valentín
- Navarro, Pablo



Propósito del Proyecto

Este proyecto implementa un proceso **ETL (Extract, Transform, Load)** para la carga y análisis de datos relacionados con desembarque de capturas marítimas en Argentina. Utiliza herramientas como Docker, PostgreSQL, pgAdmin y Apache Superset para facilitar la gestión, análisis y visualización de datos.

El objetivo principal es proporcionar una solución escalable y reproducible para analizar datos de desembarque de capturas marítimas por grupo especie, departamento y provincia, permitiendo la creación de tablas interactivas y gráficos personalizados.

Los datasets utilizados en este proyecto pueden descargarse desde el portal oficial de datos abiertos del gobierno de Argentina:

<https://datos.gob.ar/dataset>

Levantamiento de Servicios en Docker

Se crea un archivo .env.db en la raíz del proyecto con las siguientes variables de entorno:

```
1 #Definimos cada variable
2 DATABASE_HOST=db
3 DATABASE_PORT=5432
4 DATABASE_NAME=postgres
5 DATABASE_USER=postgres
6 DATABASE_PASSWORD=postgres
7 POSTGRES_INITDB_ARGS="--auth-host=scram-sha-256 --auth-local=trust"
8 # Configuración para inicializar postgres
9 POSTGRES_PASSWORD=${DATABASE_PASSWORD}
10 PGUSER=${DATABASE_USER}
11 # Configuración para inicializar pgadmin
12 PGADMIN_DEFAULT_EMAIL=postgres@postgresql.com
13 PGADMIN_DEFAULT_PASSWORD=${DATABASE_PASSWORD}
14 # Configuración para inicializar superset
15 SUPERSET_SECRET_KEY=your_secret_key_here
```

Levantamiento de Servicios en Docker

El archivo docker-compose.yml define los siguientes servicios:

```
networks:
  net:
    external: false

volumes:
  postgres-db:
    external: false

▷ Run All Services
services:
  ▷ Run Service
  db:
    image: postgres:alpine
    env_file:
      - .env.db
    restart: unless-stopped
    environment:
      - POSTGRES_INITDB_ARGS=--auth-host=md5 --auth-local=trust
    healthcheck:
      # Prueba de salud para el contenedor
      test: [ "CMD-SHELL", "pg_isready" ]
      interval: 10s
      timeout: 2s
      retries: 5
    ports:
      - 5432:5432
    volumes:
      - postgres-db:/var/lib/postgresql/data
      - ./scripts:/docker-entrypoint-initdb.d
      - ./datos:/datos
    networks:
      - net
```

```
superset:
  image: apache/superset:4.0.0
  restart: unless-stopped
  env_file:
    - .env.db
  ports:
    - 8088:8088
  depends_on:
    db:
      condition: service_healthy
  networks:
    - net
```

```
pgadmin:
  image: dpage/pgadmin4
  restart: unless-stopped
  env_file:
    - .env.db
  ports:
    - 5050:80
  depends_on:
    db:
      condition: service_healthy
  networks:
    - net
```

Levantamiento de Servicios en Docker

Se crea el archivo [init.sh](#) con los siguientes comandos:

```
#!/bin/bash
echo "Inicializamos el usuario de superset"
docker compose exec -it superset superset fab create-admin \
--username admin \
--firstname Superset \
--lastname Admin \
--email admin@superset.com \
--password admin
echo "Migramos la base de datos"
docker compose exec -it superset superset db upgrade
echo "Seteamos los Roles"
docker compose exec -it superset superset init
```

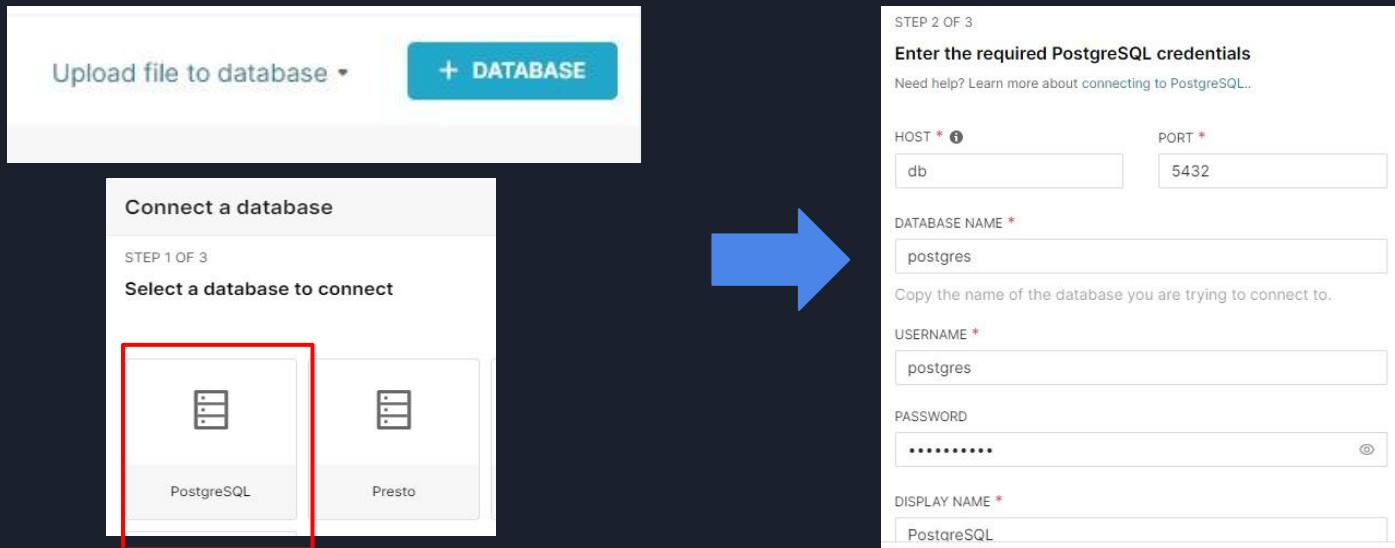
Se ejecutan los siguientes comandos para iniciar los contenedores:

```
docker compose up -d
. init.sh
```

Conexión con Apache Superset

Una vez levantados los servicios, podemos acceder a superset en <http://localhost:8088/> con el usuario “admin” y la contraseña “admin” (configurado en init.sh).

Dentro, en Settings, podemos hacer una conexión con la base de datos creada:



Upload file to database ▾ + DATABASE

Connect a database

STEP 1 OF 3

Select a database to connect

PostgreSQL Presto

STEP 2 OF 3

Enter the required PostgreSQL credentials

Need help? Learn more about connecting to PostgreSQL..

HOST * ⓘ db PORT * 5432

DATABASE NAME * postgres

Copy the name of the database you are trying to connect to.

USERNAME * postgres

PASSWORD

DISPLAY NAME * PostgreSQL

Carga de Datos desde un .csv

Los datasets utilizados son: [Desembarque de captura de especies marítimas en 2019](#), [provincias](#) y [departamentos](#).

Se crean las tablas finales para estos grupos de datos:

```
/*  
Borro las tablas si existen  
*/  
DROP TABLE IF EXISTS public.pesca;  
DROP TABLE IF EXISTS public.departamento;  
DROP TABLE IF EXISTS public.provincia;
```

```
CREATE TABLE public.provincia (  
  id BIGINT,  
  nombre VARCHAR,  
  nombre_completo VARCHAR,  
  centroide_lat FLOAT,  
  centroide_lon FLOAT,  
  categoria VARCHAR  
);
```

```
CREATE TABLE public.departamento (  
  id BIGINT,  
  nombre VARCHAR,  
  nombre_completo VARCHAR,  
  centroide_lat FLOAT,  
  centroide_lon FLOAT,  
  categoria VARCHAR,  
  provincia_id BIGINT  
);  
  
CREATE TABLE public.pesca (  
  id SERIAL,  
  fecha VARCHAR(10),  
  flota VARCHAR(100),  
  puerto VARCHAR(100),  
  latitud FLOAT,  
  longitud FLOAT,  
  categoria VARCHAR(50),  
  especie VARCHAR(100),  
  especie_agrupada VARCHAR(100),  
  captura BIGINT,  
  departamento_id BIGINT  
);
```




Carga de Datos desde un .csv

Se agregan las restricciones de las claves primarias y foráneas:

```
ALTER TABLE public.pesca
ADD CONSTRAINT pesca_pk PRIMARY KEY (id);

ALTER TABLE public.departamento
ADD CONSTRAINT departamento_pk PRIMARY KEY (id);

ALTER TABLE public.provincia
ADD CONSTRAINT provincia_pk PRIMARY KEY (id);

ALTER TABLE public.departamento
ADD CONSTRAINT fk_departamento_provincia FOREIGN KEY (provincia_id) REFERENCES provincia (id);

ALTER TABLE public.pesca
ADD CONSTRAINT fk_pesca_departamento FOREIGN KEY (departamento_id) REFERENCES departamento (id);

/*
```

Carga de Datos desde un .csv

Para cargar los datos desde los csv, se utilizan tablas temporales, lo que nos permite quitar o modificar atributos antes de agregarlos definitivamente a la base de datos.

Las tablas temporales contienen todos los atributos que posee el csv.

```
CREATE TEMPORARY TABLE temp_departamentos (  
  categoria VARCHAR,  
  centroide_lat FLOAT,  
  centroide_lon FLOAT,  
  fuente VARCHAR,  
  id VARCHAR,  
  nombre VARCHAR,  
  nombre_completo VARCHAR,  
  provincia_id VARCHAR,  
  provincia_interseccion FLOAT,  
  provincia_nombre VARCHAR  
);
```

```
CREATE TEMPORARY TABLE provincias_temp (  
  categoria VARCHAR,  
  centroide_lat FLOAT,  
  centroide_lon FLOAT,  
  fuente VARCHAR,  
  id VARCHAR,  
  iso_id VARCHAR,  
  iso_nombre VARCHAR,  
  nombre VARCHAR,  
  nombre_completo VARCHAR  
);
```

```
CREATE TEMPORARY TABLE pesca_temp (  
  fecha VARCHAR,  
  flota VARCHAR,  
  puerto VARCHAR,  
  provincia VARCHAR,  
  provincia_id VARCHAR,  
  departamento VARCHAR,  
  departamento_id VARCHAR,  
  latitud FLOAT,  
  longitud FLOAT,  
  categoria VARCHAR,  
  especie VARCHAR,  
  especie_agrupada VARCHAR,  
  captura BIGINT  
);
```

Carga de Datos desde un .csv

Ahora, se copian (o extraen) los datos del .csv dentro de las tablas temporales:

```
COPY provincias_temp
FROM '/datos/provincias.csv' DELIMITER ',' CSV HEADER;
```

```
INSERT INTO
    public.provincia (
        id,
        nombre,
        nombre_completo,
        centroide_lat,
        centroide_lon,
        categoria
    )
```

```
SELECT
    id::INTEGER,
    nombre,
    nombre_completo,
    centroide_lat,
    centroide_lon,
    categoria
FROM provincias_temp;
```

```
COPY temp_departamentos
FROM '/datos/departamentos.csv' DELIMITER ',' CSV HEADER;
```

```
INSERT INTO
    public.departamento (
        id,
        nombre,
        nombre_completo,
        centroide_lat,
        centroide_lon,
        categoria,
        provincia_id
    )
```

```
SELECT
    id::INTEGER,
    nombre,
    nombre_completo,
    centroide_lat,
    centroide_lon,
    categoria,
    provincia_id::INTEGER
FROM temp_departamentos;
```

Carga de Datos desde un .csv

```
COPY pesca_temp  
FROM '/datos/captura-puerto-flota-2019-utf8.csv' DELIMITER ',' CSV HEADER;
```

```
INSERT INTO  
    public.pesca (  
        fecha,  
        flota,  
        puerto,  
        latitud,  
        longitud,  
        categoria,  
        especie,  
        especie_agrupada,  
        captura,  
        departamento_id  
    )
```

```
SELECT  
    fecha,  
    flota,  
    puerto,  
    latitud,  
    longitud,  
    categoria,  
    especie,  
    especie_agrupada,  
    captura,  
    departamento_id::INTEGER  
  
FROM pesca_temp  
WHERE departamento_id::BIGINT IN (SELECT id FROM departamento);
```

Carga de Datos a las tablas definitivas

Desde las tablas temporales se cargan los datos a las tablas definitivas de la base de datos:

```
INSERT INTO public.provincia (  
    id,  
    nombre,  
    nombre_completo,  
    centroide_lat,  
    centroide_lon,  
    categoria  
)  
SELECT DISTINCT  
    id::BIGINT,  
    nombre,  
    nombre_completo,  
    centroide_lat,  
    centroide_lon,  
    categoria  
FROM provincias_temp  
WHERE id IS NOT NULL  
    AND id::BIGINT NOT IN (SELECT id FROM public.provincia);
```

```
INSERT INTO public.departamento (  
    id,  
    nombre,  
    nombre_completo,  
    centroide_lat,  
    centroide_lon,  
    categoria,  
    provincia_id  
)  
SELECT DISTINCT  
    id::BIGINT,  
    nombre,  
    nombre_completo,  
    centroide_lat,  
    centroide_lon,  
    categoria,  
    provincia_id::BIGINT  
FROM temp_departamentos  
WHERE id IS NOT NULL  
    AND id::BIGINT NOT IN (SELECT id FROM public.departamento);
```



Carga de Datos a las tablas definitivas

```
INSERT INTO public.pesca (  
    fecha,  
    flota,  
    puerto,  
    latitud,  
    longitud,  
    categoria,  
    especie,  
    especie_agrupada,  
    captura,  
    departamento_id  
)  
SELECT  
    fecha,  
    flota,  
    puerto,  
    latitud,  
    longitud,  
    categoria,  
    especie,  
    especie_agrupada,  
    captura,  
    departamento_id::BIGINT  
FROM pesca_temp  
WHERE departamento_id::BIGINT IN (SELECT id FROM departamento);
```



Consultas a la base de datos

Se crean consultas personalizadas en base a los datos cargados en la base de datos:

```
-- Ejemplo: Total de captura por provincia
```

```
SELECT
  p.nombre AS provincia,
  SUM(pe.captura) AS total_captura
FROM public.pesca pe
JOIN public.departamento d ON pe.departamento_id = d.id
JOIN public.provincia p ON d.provincia_id = p.id
GROUP BY p.nombre
ORDER BY total_captura DESC;
```

```
-- Ejemplo: Top 3 de especies más capturadas a nivel nacional
```

```
SELECT
  pe.especie,
  SUM(pe.captura) AS total_captura
FROM public.pesca pe
GROUP BY pe.especie
ORDER BY total_captura DESC
LIMIT 3;
```



Consultas a la base de datos

```
-- Ejemplo: Cantidad de especies distintas capturadas  
-- por provincia y departamento
```

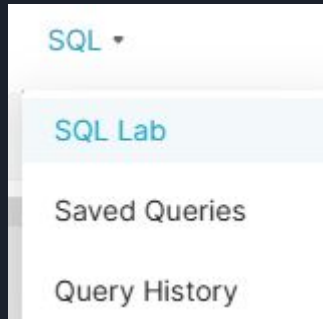
```
SELECT  
    p.nombre AS provincia,  
    d.nombre AS departamento,  
    COUNT(DISTINCT pe.especie) AS cantidad_especies  
FROM pesca pe  
JOIN departamento d ON pe.departamento_id = d.id  
JOIN provincia p ON d.provincia_id = p.id  
GROUP BY p.nombre, d.nombre  
ORDER BY cantidad_especies DESC;
```




Visualización de las consultas en Superset

Realizando las consultas dentro de Apache Superset, se pueden visualizar los datos obtenidos de una manera más interactiva.

Utilizamos SQL Lab para realizar la consulta.



Visualización de las consultas en Superset

CONSULTA: Total de captura por provincia

provincia	total_captura
Buenos Aires	1057951814
Chubut	150842968
Santa Cruz	104576918
Río Negro	18168114
Tierra del Fuego, Antártida e Islas del Atlántico Sur	2004026

CONSULTA: Top 3 especies más capturadas a nivel nacional

especie	total_captura
Merluza hubbsi	110627294
Langostino	104464594
Mero	74661212

Visualización de las consultas en Superset

CONSULTA: Cantidad de especies distintas capturadas por provincia y departamento

provincia	departamento	cantidad_especies
Buenos Aires	General Pueyrredón	70
Buenos Aires	Necochea	33
Buenos Aires	General Lavalle	32
Río Negro	San Antonio	24
Chubut	Biedma	22
Santa Cruz	Deseado	19
Tierra del Fuego, Antártida e Islas del Atlántico Sur	Ushuaia	16
Buenos Aires	La Costa	16
Chubut	Escalante	14
Buenos Aires	Castelli	14
Chubut	Rawson	12
Buenos Aires	Coronel de Marina Leonardo Rosales	7
Buenos Aires	Bahía Blanca	6
Chubut	Florentino Ameghino	3
Santa Cruz	Magallanes	1