



# Visualización de información a nivel global con Elasticsearch y Kibana

Grupo 2 - Iñigo Murga, Jon Cañadas y Mikel García



## Reto abordado

Este proyecto consiste en la visualización de datos a nivel global. Para ello utilizamos Elasticsearch y Kibana.

- Generación correcta de los datos.
- Intento de integración de seguridad.
- Analizar el CSV.
- Generar visualizaciones correspondientes.

# `docker-compose.yml`

---

```
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
      - bootstrap.memory_lock=true
      - xpack.security.enabled=false
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - esdata:/usr/share/elasticsearch/data
    ports:
      - "9200:9200"

  kibana:
    image: docker.elastic.co/kibana/kibana:${STACK_VERSION}
    container_name: kibana
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      - xpack.security.enabled=false
    depends_on:
      - elasticsearch
    ports:
      - "5601:5601"

volumes:
  esdata:
    driver: local
```

# **docker-composeSEC.yml**

---

```
services:
  setup:
    image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
    volumes:
      - certs:/usr/share/elasticsearch/config/certs
    user: "0"
    command: >
      bash -c '
        if [ x${ELASTIC_PASSWORD} == x ]; then
          echo "Set the ELASTIC_PASSWORD environment variable in the .env file";
          exit 1;
        elif [ x${KIBANA_PASSWORD} == x ]; then
          echo "Set the KIBANA_PASSWORD environment variable in the .env file";
          exit 1;
        fi;
        if [ ! -f config/certs/ca.zip ]; then
          echo "Creating CA";
          bin/elasticsearch-certutil ca --silent --pem -out config/certs/ca.zip;
          unzip config/certs/ca.zip -d config/certs;
        fi;
        if [ ! -f config/certs/certs.zip ]; then
          echo "Creating certs";
          echo -ne \
            "instances:\n"\
            "  - name: es01\n"\
            "    dns:\n"\
            "      - es01\n"\
            "      - localhost\n"\
            "    ip:\n"\
            "      - 127.0.0.1\n"\
            "  - name: kibana\n"\
            "    dns:\n"\
            "      - kibana\n"\
            "      - localhost\n"\
            "    ip:\n"\
            "      - 127.0.0.1\n"\
          > config/certs/instances.yml;
          bin/elasticsearch-certutil cert --silent --pem -out config/certs/certs.zip --in config/certs/instances.yml --ca-cert config/certs/ca/ca.crt --ca-key config/certs/ca/ca.key;
          unzip config/certs/certs.zip -d config/certs;
        fi;
        echo "Setting file permissions"
        chown -R root:root config/certs;
        find . -type d -exec chmod 750 {\} \;;
        find . -type f -exec chmod 640 {\} \;;
        echo "Waiting for Elasticsearch availability";
        until curl -s --cacert config/certs/ca/ca.crt https://es01:9200 | grep -q "missing authentication credentials"; do sleep 30; done;
        echo "Setting kibana_system password";
      '
```

```
until curl -s -X POST --cacert config/certs/ca/ca.crt -u "elastic:${ELASTIC_PASSWORD}" -H "Content-Type: application/json" https://es01:9200/_security/user/kibana_system/_password -d "{\"password\":\"\"}";
echo "All done!";

healthcheck:
  test: ["CMD-SHELL", "[ -f config/certs/es01/es01.crt ]"]
  interval: 1s
  timeout: 5s
  retries: 120

es01:
  depends_on:
    setup:
      condition: service_healthy
  image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
  volumes:
    - certs:/usr/share/elasticsearch/config/certs
    - es01data:/usr/share/elasticsearch/data
  ports:
    - ${ES_PORT}:9200
  environment:
    - node.name=es01
    - cluster.name=${CLUSTER_NAME}
    - discovery.type=single-node
    - ELASTIC_PASSWORD=${ELASTIC_PASSWORD}
    - bootstrap.memory_lock=true
    - xpack.security.enabled=true
    - xpack.security.http.ssl.enabled=true
    - xpack.security.http.ssl.key=certs/es01/es01.key
    - xpack.security.http.ssl.certificate=certs/es01/es01.crt
    - xpack.security.http.ssl.certificate_authorities=certs/ca/ca.crt
    - xpack.security.transport.ssl.enabled=true
    - xpack.security.transport.ssl.key=certs/es01/es01.key
    - xpack.security.transport.ssl.certificate=certs/es01/es01.crt
    - xpack.security.transport.ssl.certificate_authorities=certs/ca/ca.crt
    - xpack.security.transport.ssl.verification_mode=certificate
    - xpack.license.self_generated.type=${LICENSE}
  healthcheck:
    test:
      [
        "CMD-SHELL",
        "curl -s --cacert config/certs/ca/ca.crt https://localhost:9200 | grep -q 'missing authentication credentials'",
      ]
    interval: 10s
    timeout: 10s
```

retries: 120

kibana:

depends\_on:

es01:

condition: service\_healthy

image: docker.elastic.co/kibana/kibana:\${STACK\_VERSION}

volumes:

- certs:/usr/share/kibana/config/certs

- kibanadata:/usr/share/kibana/data

ports:

- \${KIBANA\_PORT}:5601

environment:

- SERVERNAME=kibana

- ELASTICSEARCH\_HOSTS=https://es01:9200

- ELASTICSEARCH\_USERNAME=kibana\_system

- ELASTICSEARCH\_PASSWORD=\${KIBANA\_PASSWORD}

- ELASTICSEARCH\_SSL\_CERTIFICATEAUTHORITIES=config/certs/ca/ca.crt

healthcheck:

test:

[

"CMD-SHELL",

"curl -I -s --cacert config/certs/ca/ca.crt https://localhost:5601 | grep -q 'HTTP/1.1 302 Found'",

]

interval: 10s

timeout: 10s

retries: 120

volumes:

certs:

driver: local

es01data:

driver: local

kibanadata:

driver: local



**.env**

—

COMPOSE\_PROJECT\_NAME=visualizacion

ELASTIC\_PASSWORD=elasticdemo

KIBANA\_PASSWORD=kibanademo

STACK\_VERSION=8.8.2

CLUSTER\_NAME=docker-cluster

LICENSE=basic

ES\_PORT=9200

KIBANA\_PORT=5601

# CSV Elegido

---



## Descripción

**Contiene diferentes datos del año 2023 divididos por países, incluyendo indicadores económicos, sociales, sanitarios y ambientales. Este dataset nos permite realizar análisis comparativos entre países y explorar posibles relaciones entre distintas variables de desarrollo.**

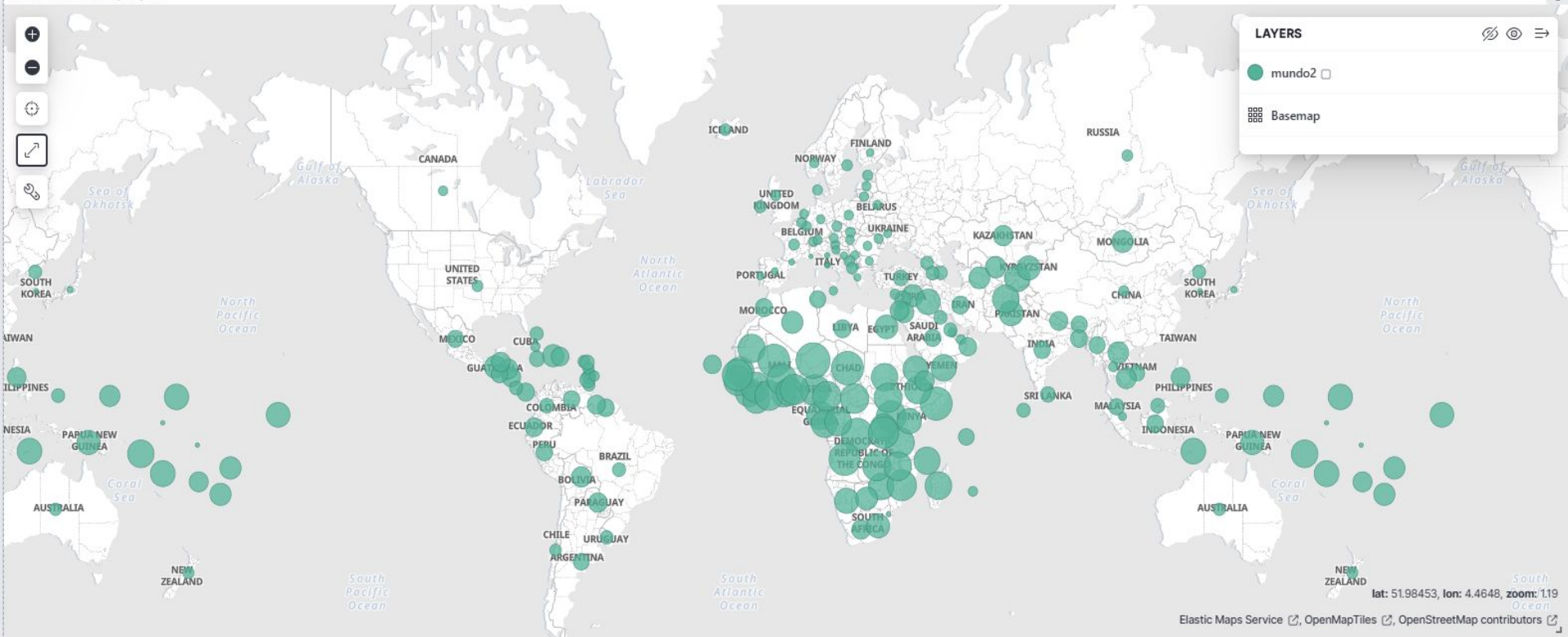
## Columnas Utilizadas

- **País:** nombre del país
- **Tasa de natalidad:** número de nacimientos por cada 1000 personas
- **Tasa de fertilidad:** número promedio de hijos por mujer
- **Mortalidad infantil:** muertes de menores de 1 año por cada 1000 nacidos vivos
- **Esperanza de vida:** Promedio de años que se espera que viva una persona

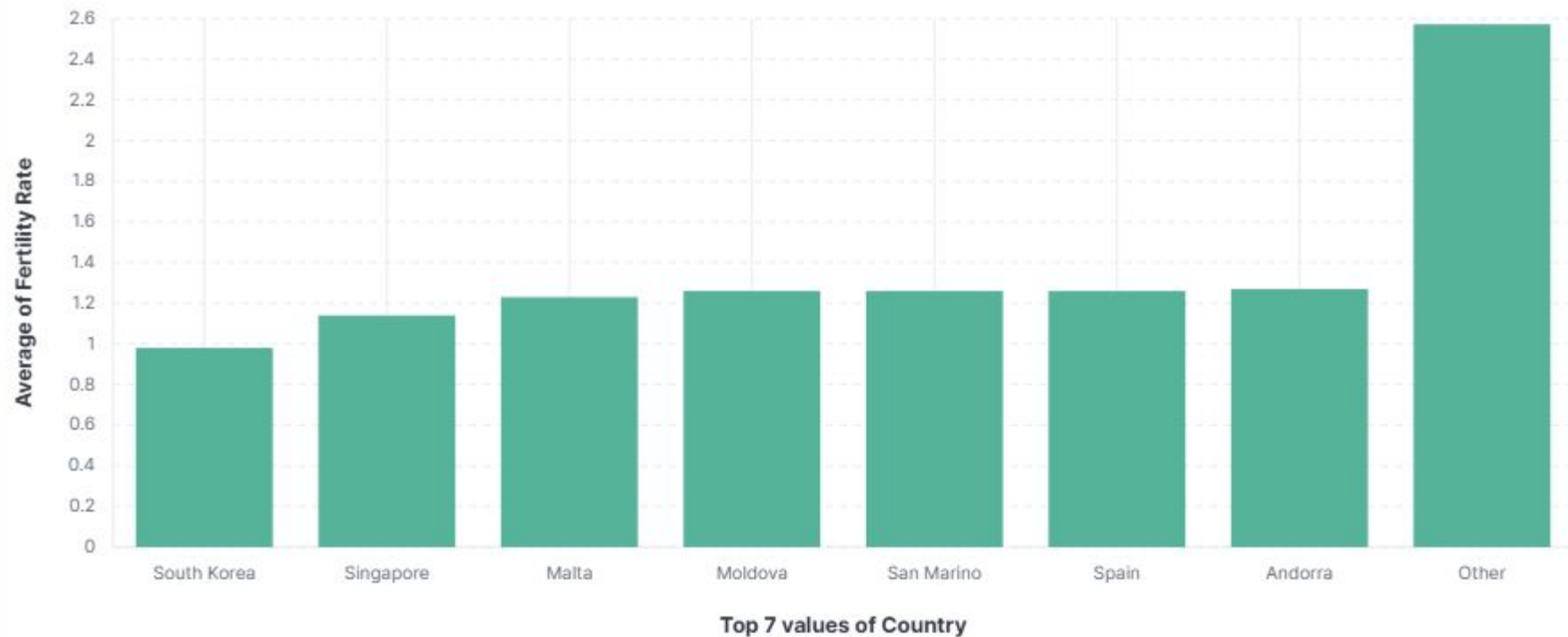
# Visualizaciones Elegidas

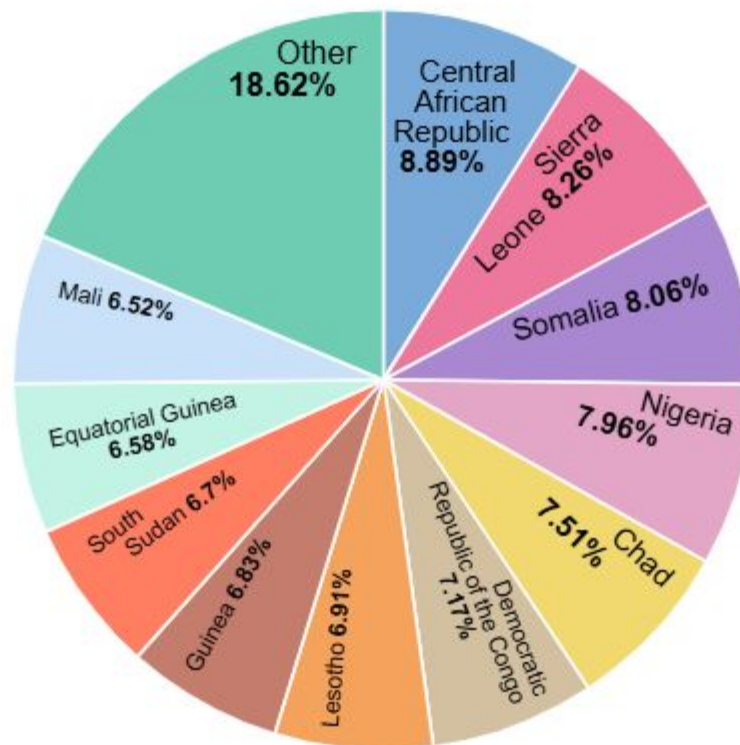
---

# Tasa de natalidad por país



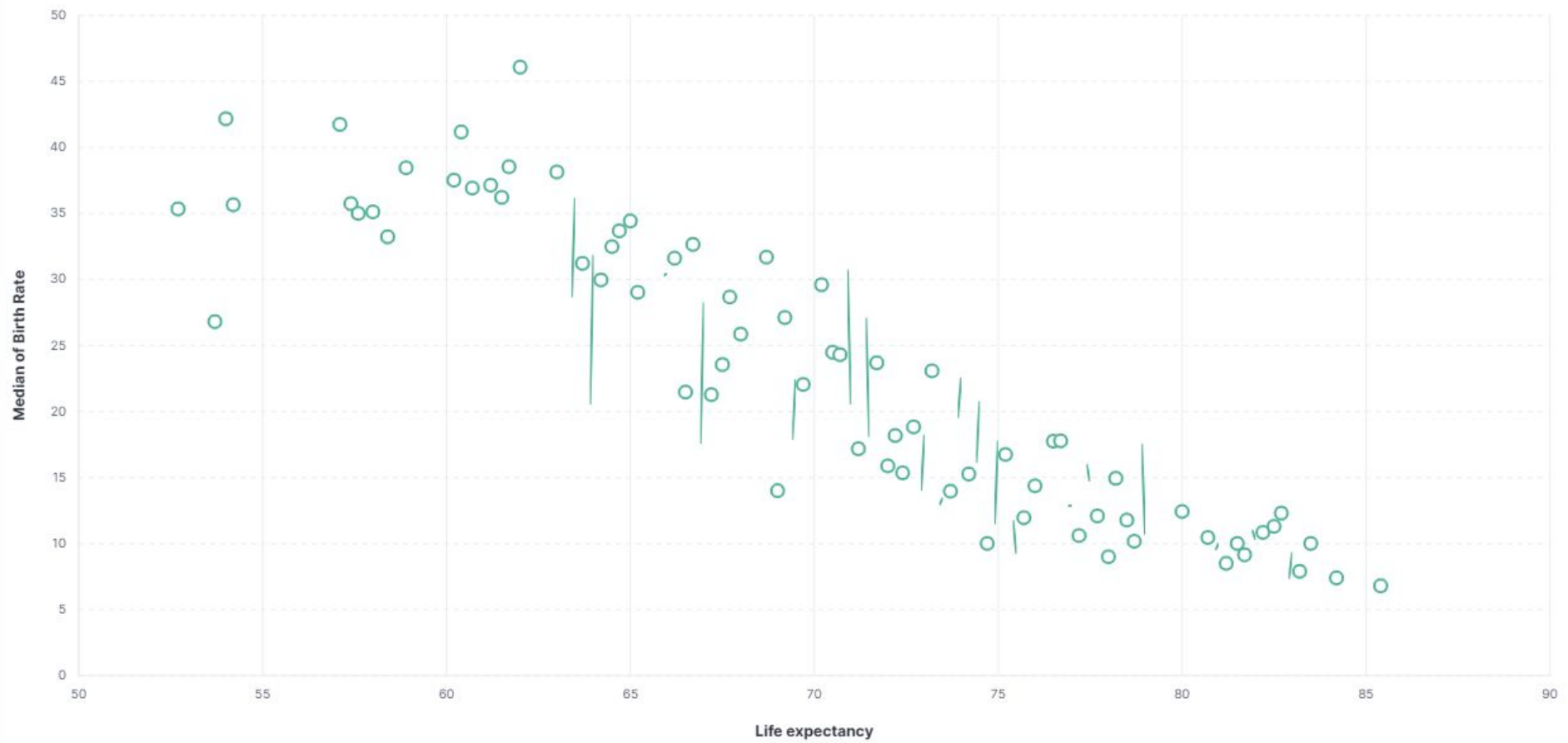
## Fertilidad por país







Esperanza de vida con tasa de natalidad





## Problemas y Posibles futuras mejoras

- Al probar a integrar el sistema con seguridad hemos tenido múltiples problemas como no poder acceder a elastic o tardía al lanzar el sistema para acceder a kibana.
- Al analizar qué visualizaciones realizar, tuvimos problemas con el mapa geográfico por lo que tuvimos que buscar un CSV que se adecuara.
- Al intentar añadir widgets al dashboard, como nuestro dataset cuenta con muchos países, la visualización de los gráficos no era buena. Esto ocurría porque acababa agrupando muchos países restantes en el campo “otros” provocando que fuese ilegible.
- ✓ Para evitar acceso no deseados estaría bien la introducción de seguridad.
- ✓ Se podría realizar un lavado del CSV haciendo diversas modificaciones como la eliminación de columnas no utilizadas.
- ✓ Implementar más nodos de Elasticsearch para un mayor rendimiento y escalabilidad.



## Alternativas



Grafana

