

# Proyecto Final

## Análisis de datos

Díaz Carlos  
Palomo Wendy  
Yépez Alexis

### I. DEFINICIÓN DEL CASO DE ESTUDIO

Se obtendrán análisis y visualizaciones de los siguientes casos de estudio:

#### A. *Pulso político en 20 ciudades principales de Ecuador*

El ámbito político para el Ecuador en el año 2021 se ha caracterizado por los casos de corrupción que han salido a la luz, también un punto importante ha sido el posicionamiento del nuevo presidente “Guillermo Lasso” lo cual ha desencadenado criterios de aceptación y rechazo de diferentes frentes políticos y ciudadanos.

#### B. *Pulso político por provincias en Ecuador*

La recopilación de datos en cada una de las provincias acerca del uso de la red social Twitter por lo que se filtrara la información por palabras y las coordenadas de cada provincia, donde se analizara la cantidad de Tweets realizados por cada usuario con relación a la política del país.

#### C. *Juegos en línea por países.*

Desde su aparición, los juegos en línea han ido creciendo en gran manera, tanto en aceptación como en geografía. Cada país cuenta con una serie de gañeras, streamers e influencers que dan a conocer juegos en línea en cuanto salen al público. Su interacción en redes sociales siempre es relevante frente a otros temas y es una actividad que ira incrementando conforme el paso de los años.

#### D. *Tema definido por el estudiante*

Recopilación de datos de estudiantes de la Escuela Politécnica Nacional que consta de provincia, cantón y demás datos relevantes para analizar en donde se concentra la mayor cantidad de estudiantes de la universidad y cuáles son las provincias más comunes de donde vienen los estudiantes foráneos a estudiar.

#### E. *Eventos o noticias mundiales*

Se recopiló información de noticieros internacionales que tienen una página en Facebook, desde la cual se obtuvo la información de cada post realizado con el objetivo de analizar el tráfico generado a sus seguidores. Buscando ver cuál de ellos puede tener más interacciones por publicación realizada.

### II. OBJETIVOS GENERALES Y ESPECÍFICOS

#### A. *Objetivos Generales*

Recopilar datos para analizarlos a través de herramientas dedicadas.

#### B. *Objetivos Específicos*

- Analizar los casos de estudio para el análisis
- Crear una arquitectura para recopilación de datos
- Desarrollar un cronograma de actividades.
- Dividir las actividades y herramientas a utilizar por integrante tomando como referencia la arquitectura.
- Recopilar datos de las fuentes establecidas en la arquitectura
- Realizar un análisis gráfico usando las herramientas establecidas en la arquitectura.
- Documentar todo el análisis realizado por cada integrante del grupo.

### III. DESCRIPCIÓN DEL EQUIPO DE TRABAJO Y ACTIVIDADES REALIZADAS POR CADA UNO

El equipo está conformado por 3 personas:

#### A. *Carlos Díaz*

- Crear la arquitectura del proyecto
- Recopilar datos de redes sociales (Facebook, Twitter, TikTok) para el caso de estudio pertinente
- Guardar las bases de datos en el gestor CouchDB, MongoDB, SQLServer.
- Concentrar los datos en MongoDB Atlas

## B. Wendy Palomo

- Crear la arquitectura del proyecto
- Crear los dashboard de visualización de los datos concentrados en MongoDB Atlas
- Realizar el análisis de datos en Power BI

## C. Alexis Yépez

- Crear la arquitectura del proyecto
- Recolectar datos desde Twitter del pulso político de 20 ciudades del Ecuador
- Concentrar los datos provenientes de Twitter, CouchDB y Mysql en Elasticsearch
- Crear los dashboard de visualizaciones de las fuentes de datos mencionadas en el punto anterior con Kibana

## IV. CRONOGRAMA DE ACTIVIDADES

OBJETIVOS	ACTIVIDADES	SEMANA 1	SEMANA 2	SEMANA 3	TOTAL HORAS
Objetivo 1. Definir una arquitectura que indique el flujo que lleva la recolección de datos hasta su interpretación.	Definir el número de base de datos SQL y NoSQL.	3			3
	Definir las herramientas a utilizar.	3			3
	Definir los recursos a utilizar.	3			3
	Definir el tiempo que tendrá el proyecto.	3			3
Objetivo 2. Recolectar los datos de diferentes fuentes y almacenarlos en distintas bases de datos SQL y No SQL usando distintos scripts.	Crear scripts para la recolección de datos.	6			6
	Ejecutar los scripts.	25	25		50
	Buscar archivos CSV.	6			6
	Analizar los datos recopilados.		10	10	20
Objetivo 3. Concentrar los datos en una herramienta de colección de datos masivos.	Cargar los datos a una BD.	6	4		10
	Crear índices en Elasticsearch.	6	4		10
	Crear scripts para transferir los datos a Elasticsearch.		6	6	12
	Cargar los datos en una herramienta de visualización.		4	4	8
Objetivo 4. Generar dashboards o visualizaciones que muestren todos los tipos de datos recolectados.	Realizar la limpieza de los datos recopilados.		2	2	4
	Crear las visualizaciones respectivas.		10	10	20
	Interpretar las visualizaciones del dashboard.		12	12	24
	Documentar los resultados obtenidos.		5	5	10
Objetivo 5. Interpretar las distintas visualizaciones y dashboards obtenidos y reportar los resultados obtenidos con los datos recolectados.	Realizar las conclusiones de los resultados.		5	5	10
Total horas por semana		49	47	87	163

Tabla 1: Cronograma de actividades

## V. RECURSO Y HERRAMIENTAS UTILIZADAS

### A. PYTHON

Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Para este proyecto en específico se está haciendo uso de las siguientes librerías:

- pymongo: utilizada para poder conectarnos a MongoDB.
- tweepy: librería fácil de usar para acceder a la API de Twitter.
- couchdb: utilizada para poder conectarnos a CouchDB.
- pymysql: utilizada para poder conectarnos a MySQL.
- pandas: es una biblioteca escrita como extensión de NumPy para manipulación y análisis de datos.

## B. APACHE COUCHDB

CouchDB es un sistema de gestión de bases de datos que aprovecha lo mejor de las bases de datos documentales para alcanzar el excelente rendimiento de las bases de datos relacionales.

El protocolo que emplea CouchDB en su replicación de datos se implementa en una variedad de proyectos y productos que abarcan todos los entornos informáticos desde clústers de servidores distribuidos globalmente, pasando por teléfonos móviles hasta navegadores web.



Figura 1: CouchDB

### C. MONGODB

Es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++, que en lugar de guardar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico.



Figura 2: MongoDB

### D. LOGSTASH

Logstash es parte del ElasticStack. Es una herramienta Open Source que nos permite centralizar la recogida de información, normalizarla y redistribuirla.



Figura 4: logstash

### E. ELASTICSEARCH

Elasticsearch es un motor de analítica y análisis distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados.



**Figura 5:** ElasticSearch

### F. KIBANA

Kibana es una aplicación de frontend gratuita y abierta que se encuentra sobre el Elastic Stack y proporciona capacidades de visualización de datos y de búsqueda para los datos indexados en Elasticsearch.



**Figura 6:** Kibana

### G. POWER BI

Power BI es una solución de análisis empresarial basado en la nube, que permite unir diferentes fuentes de datos, analizarlos y presentar un análisis de estos a través de informes y paneles. Con Power BI se tiene de manera fácil acceso a datos dentro y fuera de la organización casi en cualquier dispositivo.



**Figura 7:** Power BI

### H. MICROSOFT SQL SERVER

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) que admite una amplia variedad de aplicaciones de procesamiento de transacciones, inteligencia empresarial y análisis en entornos informáticos corporativos.



**Figura 8:** SQL Sever

### I. MONGODB ATLAS

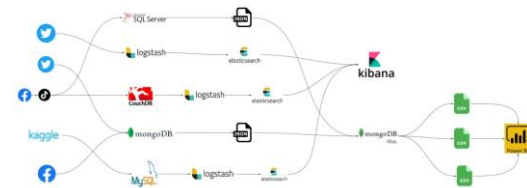
MongoDB Atlas es un servicio de Cloud Database (o Base de Datos en la Nube), que te permite crear y administrar tu BBDD Mongo desde cualquier lugar del mundo, a través de su plataforma



**Figura 9:** MongoDB Atlas

## VI. ARQUITECTURA DE LA SOLUCIÓN

La arquitectura planteada para el proyecto consta de las siguientes partes, los datos serán recabados de diferentes redes sociales como Facebook, Twitter, Tik Tok y de otras fuentes como Kaggle. Por otra parte, para almacenar estos datos en primera instancia se utilizará bases de datos relacionales como SQL Server, MySQL y no relacionales como MongoDB, CouchDB y Elasticsearch. Posteriormente se tendrá dos concentradores de datos: Elasticsearch y MongoDB Atlas, a estos se dirigirán los datos que se encuentran en las bases de datos mencionadas anteriormente. En última instancia la arquitectura contiene como elementos visualizadores de datos a Power BI y Kibana, donde se representarán de forma gráfica todos los datos y así facilitar el proceso de análisis.



**Figura 10:** Arquitectura

## VII. EXTRACCIÓN DE DATOS

A. Pulso político en 20 ciudades principales de Ecuador

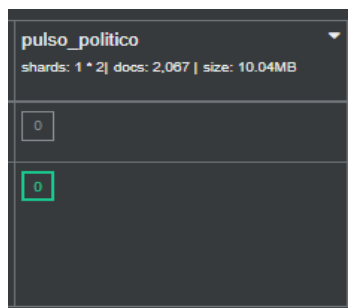
La recopilación de datos del caso de A, se la realizo desde la red social Twitter. Para esto usamos Logstash, es necesario configurar el archivo .conf y pasarle como parámetros las credenciales de desarrollador que Twitter nos ofrece para acceder a sus datos.

```
input {
  twitter {
    # add your data
    consumer_key => "s1F9DjnrIGRI3CeY5c3T6Aagj7"
    consumer_secret =>
    "ciQcHaJ1Wec4Wuchjrc6Ein7QFDngAf1Z4GTA82MnukX86RnJ"
    oauth_token =>
    "1251569508222935046-80W1M6STcxvJIi0aV2009RMy0196F"
    oauth_token_secret =>
    "tU901KKCCtmhBQjpZQLB16b2Rvffflf4I3wgjfw8nMQno"
    keywords => ["ecuador", "pulso politico", "quito",
    "guayaquil", "ambato",
    "milagro", "cuenca", "loja", "santo domingo", "machala",
    "durán", "manta", "portoviejo", "esmeraldas", "quevedo",
    "riobamba", "ibarra", "babahoyo", "sangolquí", "latacunga",
    "tulcan", "pasaje"]
    full_tweet => true
  }
}

output {
  elasticsearch {
    index => pulso_politico
    document_type => twitter
  }
}
```

**Figura 11:** Archivo de configuración de logstash

Definimos el nombre de las ciudades como palabras después de Ecuador y pulso político ya que están relacionadas con los datos que nos interesa recabar.



**Figura 12:** Índice creado en elasticsearch

```
{
  "id_str": "42479983",
  "profile_background_image_url":
  "http://abs.twimg.com/images/themes/theme13/bg.gif",
  "listed_count": 1851,
  "profile_use_background_image": false,
  "id": 42479983,
  "location": "Quito - Ecuador",
  "friends_count": 1659,
  "followers_count": 454998,
  "profile_sidebar_border_color": "000000",
  "profile_image_url_https":
  "https://pbs.twimg.com/profile_images/142495864351829254/BGATAEBA_normal.jpg",
  "contributors_enabled": false,
  "profile_background_image_url_https":
  "https://abs.twimg.com/images/themes/theme13/bg.gif",
  "url": "http://www.ecuadorinmediato.com",
  "utc_offset": null,
  "withheld_in_countries": [ -
  ],
  "created_at": "Mon May 25 20:25:55 +0000 2009",
  "notifications": null,
  "time_zone": null,
  "profile_sidebar_fill_color": "000000",
  "name": "EcuadorInmediato",
  "screen_name": "ecuadorinmediato",
  "verified": false,
  "profile_background_tile": false,
  "profile_link_color": "#6699FF",
  "description": "Medio de Comunicación | Cuenta oficial del Primer Periódico Digital de Ecuador fundado el 10 del Agosto de 2004.",
  "statuses_count": 209376,
  "translator_type": "none"
},
{
  "id_str": "143794330808382855",
  "text": "¡ATENCIÓN URGENTE! Ecuador Esta gigantografía resintió a presidente BlasóGuillermo contra @mancorran y la echó- https://t.co/Wd9re0kxv",

```

**Figura 13:** Datos almacenados en el índice

## B. Pulso político por provincias en Ecuador

La recopilación de datos efectuada para cada provincia fue realizada con el lenguaje Python y se hizo uso de la API de twitter para poder

realizar una recolección por ubicación y términos de palabras que permitan obtener como resultado el pulso político en las diferentes ciudades del Ecuador y a su vez almacenar todo esto en la base de datos no relacional MongoDB.

Para conseguir esto se utilizaron las librerías de pymongo que permite establecer una conexión con MongoDB sobre la cual se pueda realizar una inserción de datos y estos a su vez queden almacenados en ella. Se utilizo la librería de tweepy con la cual se puede establecer una conexión con Twitter, para lo cual es necesario tener unas credenciales que son otorgadas desde la plataforma de desarrollo de Twitter.

```
from pymongo import MongoClient
from pymongo.errors import ConnectionFailure
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import json

'''=====API TWITTER====='''
ckey = "VEROCB2cKlWdz1FRUC6xQnAEj"
csecret = "uyLI7NilyXfRXAeSnifYvwCe7AFW7FfcBQajUSprpvyktpAtW"
atoken = "2353244641-3MqbkpQHEJigHmzB7y0T1iVtw1VrXgpoSeUgF"
asecret = "HQLYghs2n7y0iBFTbRzUJmRc71Krb8gkgUdun3s0EA1"

'''=====CONEXION MONGODB====='''

CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db_mongo = CLIENT["pulso_politico_prov"]
provincia = db_mongo["cotopaxi"]

'''=====INSERCIÓN DE DATOS====='''
class listener(StreamListener):
    def on_data(self, data):
        dictTweet = json.loads(data)
        try:

```

**Figura 14:** Script para recopilar datos

Luego procedemos a insertar los datos en MongoDB con el siguiente script con el cual vamos estableciendo el identificador necesario para su almacenamiento y después enviamos una confirmación sobre su estado, también colocamos un mensaje para notificar si existió un error en la inserción de datos.

```
'''=====INSERCIÓN DE DATOS====='''
class listener(StreamListener):
    def on_data(self, data):
        dictTweet = json.loads(data)
        try:
            dictTweet['_id'] = str(dictTweet['id'])
            doc = provincia.insert(dictTweet)
            print("SAVED" + str(doc) + ">" + str(data))
        except:
            print("Already exists")
            pass
        return True

    def on_error(self, status):
        print(status)

'''=====AUTH====='''
auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
twitterStream = Stream(auth, listener())
```

**Figura 15:** Script para inserción de datos en MongoDB

Además de las credenciales es necesario establecer los términos de búsqueda en nuestro caso fue necesario describir las palabras que se relacionen con el Pulso Político en Ecuador actualmente, adicional a esto fue necesario establecer las ubicaciones de cada provincia por coordenadas.

```
'''=====AUTH====='''
auth = OAuthHandler(key, csecret)
auth.set_access_token(token, ascret)
twitterStream = Stream(auth, listener())

'''=====TRACK====='''
# Palabras para realizar búsqueda relacionada con el pulso político en Ecuador
twitterStream.filter(track=['Pulso Político', 'Bullerino Lasso', 'Presidente', 'Alcald',
                           'Embajadores', 'Política', 'Borrero', 'Vicepresidentes', 'Corrupción'])

'''=====LOCATION====='''

# COSTA
# twitterStream.filter(locations=[-81.0114, -2.5076, -80.1999, -1.6607]) #Santa Elena
# twitterStream.filter(locations=[-80.5634, -3.8643, -79.1019, -0.8367]) #Guayas
# twitterStream.filter(locations=[-80.1029, 0.8618, -78.4267, 1.4696]) #Esmeraldas
# twitterStream.filter(locations=[-79.5484, -0.6987, -78.7463, 0.0191]) #Santo Domingo
# twitterStream.filter(locations=[-81.0848, -1.9515, -79.4019, 0.3707]) #Manabí
# twitterStream.filter(locations=[-80.4678, -3.8901, -79.3654, -3.0471]) #El Oro
# twitterStream.filter(locations=[-79.8775, -2.133, -79.0816, -0.5374]) #Los Rios

# SIERRA
# twitterStream.filter(locations=[-79.3709, -0.7211, -77.8397, 0.3273]) #Pichincha
# twitterStream.filter(locations=[-79.2758, 0.1229, -77.8104, 0.8766]) #Imbabura
# twitterStream.filter(locations=[-80.4847, -4.7409, -79.1007, -3.3296]) #El Ja
# twitterStream.filter(locations=[-79.7637, -3.6204, -78.4256, -2.5544]) #Azuay
# twitterStream.filter(locations=[-78.074092, 0.356681, -77.795461, 0.588566]) #Bolíver
# twitterStream.filter(locations=[-78.5494, 0.3567, -77.5255, 1.1979]) #Carchi
# twitterStream.filter(locations=[-79.3393, 1.22, -78.3809, -0.3508]) #Cotacachi
# twitterStream.filter(locations=[-79.2545, -2.5679, -78.3585, -1.4301]) #Chimborazo
# twitterStream.filter(locations=[-79.8279, 2.8369, -78.3679, 2.2169]) #El Dorado
# twitterStream.filter(locations=[-78.937982, -1.523145, -78.110204, -0.98147]) #Tungurahua

# AMAZONIA
# twitterStream.filter(locations=[-78.18, -2.689, -75.585, -1.8038]) #Pastaza
# twitterStream.filter(locations=[-77.9795, -0.6558, -75.2233, 0.6621]) #Sucumbios
# twitterStream.filter(locations=[-79.4386, -5.0159, -78.3677, -3.3619]) #Zamora Chinchipe
# twitterStream.filter(locations=[-78.5878, -2.8554, -77.6434, -2.0052]) #Morona Santiago
# twitterStream.filter(locations=[-78.4273, -1.2407, -77.027, 0.8543]) #Napo
# twitterStream.filter(locations=[-77.4703, -0.6410, -76.0810, -0.0703]) #Orellana
```

**Figura 16:** Términos de búsqueda

Después de todo este proceso podemos visualizar nuestros datos almacenados en MongoDB Compass

Collections						
Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Manabí	108	3.9 KB	421.1 KB	1	36.0 KB	
azuay	39	4.1 KB	158.4 KB	1	36.0 KB	
bolíver	9	4.7 KB	42.1 KB	1	36.0 KB	
canar	17	4.4 KB	74.7 KB	1	36.0 KB	
carchi	13	3.9 KB	50.1 KB	1	36.0 KB	
cotacachi	55	3.9 KB	213.2 KB	1	36.0 KB	
el oro	140	3.6 KB	521.6 KB	1	36.0 KB	
galapagos	4	2.6 KB	11.3 KB	1	36.0 KB	
guayas	115	3.9 KB	453.8 KB	1	36.0 KB	
los rios	119	4.0 KB	481.2 KB	1	36.0 KB	
pichincha	108	4.1 KB	447.9 KB	1	36.0 KB	

**Figura 17:** Términos de búsqueda

### C. Juegos en línea por países.

En este apartado se ha implementado un script de Python para la extracción de datos desde la red social TikTok. Esta provee de un API para acceder a sus datos por medio de hashtags, esta es la parte que se especifica en el script para que los datos tengan relación con las palabras clave.

```
import couchdb
from TikTokApi import TikTokApi
import json

'''=====couchdb====='''
server = couchdb.Server('http://admin:Joel$2021*@localhost:5984/')
try:
    # Nombre de la base de datos por provincia
    db = server.create('juegos_ec')
except:
    db = server['juegos_ec']

# Instancia de la api
api = TikTokApi.get_instance()
# Términos de búsqueda
search_term = ["fornite", "Ecuador"]

tiktoks = api.search_for_hashtags(search_term, count=30)

for tiktok in tiktoks:
    doc = db.save(tiktok)
    print(tiktok)
```

**Figura18:** Script de extracción de datos

El parámetro del país del cual se requiere la información será el que cambia en cada consulta, así como también el del nombre del o los juegos que se desee recopilar.

Databases	
juegos_ar	13.0 KB
juegos_br	13.0 KB
juegos_cn	12.7 KB
juegos_ec	12.9 KB
juegos_es	13.0 KB
juegos_in	12.7 KB
juegos_jp	12.7 KB
juegos_ru	13.0 KB
juegos_uk	10.9 KB
juegos_usa	12.7 KB

**Figura 19:** Bases de datos creadas en CouchDB

Dado que la concentración de datos se da en Elasticsearch, se envía los datos a un índice desde CouchDB por medio de logstash con el siguiente archivo de configuración:

```
input{
  couchdb_changes{
    username=>"alexis"
    password=>"-.261727-."
    db=>"juegos_usa"
  }
}
output {
  elasticsearch {
    index => "juegos_usa"
  }
}
```

**Figura 20:** Archivo de configuración

El parámetro que va a cambiar en cada ejecución será el del nombre de la base de datos de CouchDB y el nombre del índice al que irá la información. Se creará un índice por cada BD, en total 10 por cada país consultado.

**Figura 21:** Índices creados por cada BD

#### D. Tema definido por el estudiante

La recopilación de datos se omite en este paso, sin embargo, es necesario importar el archivo tip sql a MySQL para que se cree la base de datos allí.

DEST_Codigo	DEST_Cod_Estudiante	DEST_AnoNac	DEST_Genero	DEST_Ubi_Pais	DEST_Ubi_Provincia
5	A4AAA3A4IOLA	1995	MASCULINO	ECUADOR	CARCHI
6	A4AB347C76CUAD	1988	FEMENINO	ECUADOR	CARCHI
7	A4AB3A7A74PONL	1980	FEMENINO	ECUADOR	CARCHI
8	A4AB3C3C6PILA	1986	MASCULINO	ECUADOR	CARCHI
9	A4AB47BB8ALUNN	NULL	NULL	NULL	NULL
10	A4AB47BB8ALUNN	1983	MASCULINO	ECUADOR	CARCHI
11	A4AB48CA7ACABS	NULL	NULL	NULL	NULL
12	A4AB637BABBIENA	1991	MASCULINO	ECUADOR	CARCHI
13	A4AB63BA4REV	1991	MASCULINO	ECUADOR	CARCHI
14	A4AB644C3ERAA	1990	MASCULINO	ECUADOR	CARCHI
15	A4AB64BB7CBUSA	1990	MASCULINO	ECUADOR	CARCHI
16	A4AB64BCBACAO	1990	MASCULINO	ECUADOR	CARCHI
17	A4AB6677CAPOZY	1996	MASCULINO	ECUADOR	CARCHI
18	A4AB6AAB4BESP	1989	FEMENINO	ECUADOR	CARCHI
19	A4AB6BB38GUE	1986	MASCULINO	ECUADOR	CARCHI
20	A4AB7343A6REV	1993	MASCULINO	ECUADOR	CARCHI

**Figura 22:** datos importados

La siguiente parte consiste la transferencia de datos a elasticsearch mediante logstash por medio del siguiente archivo de configuración:

```
input {
  jdbc {
    jdbc_connection_string => "jdbc:mysql://localhost:3306/
danna Paola"
    jdbc_user => "alexseik"
    jdbc_password => "-.261727-."
    jdbc_driver_library => "C:\Program Files (
x86)\MySQL\Connector J
8.0\mysql-connector-java-8.0.7-dmr-bin.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    statement => "SELECT * FROM dannapaola_1627692327956"
  }
}
output {
  stdout { codec => json_lines }
  elasticsearch {
    "hosts" => "localhost:9200"
    "index" => "mysql-dannapaola"
    "document_type" => "data"
  }
}
```

**Figura 23:** Script Base de Datos Elasticsearch

**Figura 24:** Script Base de Datos Elasticsearch

#### E. Eventos o noticias mundiales

Para la recolección de datos sobre este tema se decidió realizar la búsqueda de información en esta red social y almacenar toda la información recolectada en MongoDB.

### 1. Búsqueda en la página de Facebook de ABC News

```
from facebook_scraper import get_posts
from pymongo import MongoClient
import time
from pymongo.errors import ConnectionFailure

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT['noticias_internacionales']
noticias = db['ABCNews']

i = 1
for post in get_posts('ABCNews', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']

    for post in get_posts('ABCNews', pages=30, extra_info=True):
```

**Figura 25:** Script búsqueda en Facebook ABC News

### 2. Búsqueda de información en BBC News

```
from facebook_scraper import get_posts
from pymongo import MongoClient
import time
from pymongo.errors import ConnectionFailure

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT['noticias_internacionales']
noticias = db['BBCNewsMundo']

i = 1
for post in get_posts('BBCNewsMundo', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']

    for post in get_posts('BBCNewsMundo', pages=30, extra_info=True):
```

**Figura 26:** Script búsqueda en Facebook BBC News

### 3. Búsqueda de información en Bloomberg

```
from facebook_scraper import get_posts
from pymongo import MongoClient
import time
from pymongo.errors import ConnectionFailure

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT['noticias_internacionales']
noticias = db['bloombergmarkets']

i = 1
for post in get_posts('bloombergmarkets', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']

    for post in get_posts('bloombergmarkets', pages=30, extra_info=True):
```

**Figura 27:** Script búsqueda en Facebook Bloomberg



#### 4. Búsqueda de información en CNN en español

```
from facebook_scraper import get_posts
from pymongo import MongoClient
import time
from pymongo.errors import ConnectionFailure

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["bloombergmarkets"]

i = 1
for post in get_posts('bloombergmarkets', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 28:** Script búsqueda en Bloomberg

#### 5. Búsqueda de información en ESPN

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["ESPN"]

i = 1
for post in get_posts('ESPN', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 29:** Script búsqueda en Facebook ESPN

#### 6. Búsqueda de información en New York Times

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["nytimes"]

i = 1
for post in get_posts('nytimes', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 30:** Script búsqueda en Facebook New York Times

#### 7. Búsqueda de información en RT News

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["RTnews"]

i = 1
for post in get_posts('RTnews', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 31:** Script búsqueda en Facebook RT News

#### 8. Búsqueda de información en Univision

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["univision"]

i = 1
for post in get_posts('univision', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 32:** Script búsqueda en Facebook Univision

#### 9. Búsqueda de información en USA Today

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT["noticias_internacionales"]
noticias = db["usatoday"]

i = 1
for post in get_posts('usatoday', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        doc['reactions'] = post['reactions']
    except:
```

**Figura 32:** Script búsqueda en Facebook USA Today

## 10. Búsqueda de información en Agencia ANZA

```
import ...

'''=====CONEXION MONGODB====='''
CLIENT = MongoClient("mongodb://localhost:27017/")
try:
    CLIENT.admin.command('ismaster')
    print('MongoDB connection: Success')
except ConnectionFailure as cf:
    print('MongoDB connection: failed', cf)
db = CLIENT['noticias-internacionales']
noticias = db['AgenciaANSA']

i = 1
for post in get_posts('AgenciaANSA', pages=30, extra_info=True):
    print(i)
    i = i + 1
    time.sleep(1)
    id = post['post_id']
    doc = {}
    doc['id'] = id
    mydate = post['time']

    try:
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        try:
            doc['reactions'] = post['reactions']
        except:
            pass
```

**Figura 33:** Script búsqueda en Facebook Agencia Ansa

Los resultados se almacenaron en MongoDB Compass creando una colección por cada búsqueda por tanto se obtuvieron 10 colecciones. Se emplearon las librerías de pymongo y para realizar la búsqueda se empleó la librería de Facebook\_scraper.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Manabi	108	3.9 KB	421.1 KB	1	36.0 KB	
azulay	39	4.1 KB	159.4 KB	1	36.0 KB	
bolivar	9	4.7 KB	42.1 KB	1	36.0 KB	
canar	17	4.4 KB	74.7 KB	1	36.0 KB	
carchi	13	3.9 KB	50.1 KB	1	36.0 KB	
cotacachi	55	3.9 KB	213.2 KB	1	36.0 KB	
eloro	145	3.6 KB	521.6 KB	1	36.0 KB	
galapagos	4	2.8 KB	11.3 KB	1	36.0 KB	
guayas	115	3.9 KB	452.8 KB	1	36.0 KB	
losrios	119	4.0 KB	481.2 KB	1	36.0 KB	
pichincha	89	4.1 KB	447.2 KB	1	36.0 KB	

**Figura 34:** Mongo DB

## VIII. ANÁLISIS DE LA INFORMACIÓN

### A. Pulso político en 20 ciudades principales de Ecuador

Al hacer el análisis de los datos recabados de las principales ciudades del Ecuador, se ha obtenido que la palabra corrupción y Guillermo Lasso han sido las más mencionadas y compartidas por los usuarios. Campos como los nombres de usuario que retuitearon un post nos sirve para ver que los usuarios de las ciudades de la Sierra ecuatoriana han sido los que más lo han hecho.

```
"user_mentions": [ -
  {
    "id_str": "300390462",
    "screen_name": "LassoGuillermo",
    "name": "Guillermo Lasso",
    "indices": [ -
      69,
      84
    ],
    "id": 300390462
  },
  {
    "id_str": "336140672",
    "screen_name": "maemontanov",
    "name": "Mae Montaña",
    "indices": [ -
      92,
      104
    ],
    "id": 336140672
  }
],
"urls": [ -
  {
    "display_url": "twitter.com/i/web/status/1...",
    "indices": [ -
      116,
      139
    ],
    "url": "https://t.co/W6dreDWYx",
    "expanded_url": "https://twitter.com/i/web/status/1437943330898382855"
  }
],
"hashtags": [ -
```

**Figura 35:** Datos de usuarios de twitter

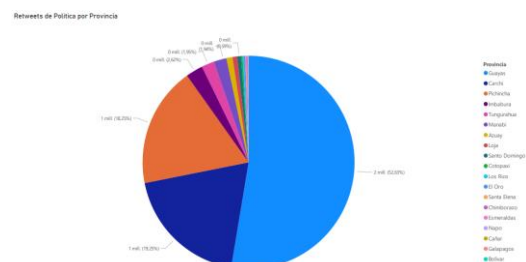
### B. Pulso político por provincias en Ecuador

La limpieza de datos fue lo primordial antes de realizar el análisis de estos, se la hizo mediante un Query de Power BI se procedió a borrar espacios vacíos, suprimir columnas innecesarias, de esta manera solo se conservó los datos necesarios como los son:

- Place.country: Para confirmar que los datos hayan sido extraídos de Ecuador.
- Place.full\_name: Filtra la provincia de donde proceden los tweets al cual se le cambio el nombre a solo 'provincias'.
- Retweet count: Filtra el total de retweets de cada tweet.

place.country	Provincia	Retweets de Publica
1	EC	48
2	EC	42
3	EC	177
4	EC	97
5	EC	217
6	EC	3803
7	EC	188
8	EC	2080
9	EC	200
10	EC	451
11	EC	97
12	EC	3803
13	EC	22
14	EC	710
15	EC	414
16	EC	321
17	EC	618

**Figura 36:** Bases de datos en PowerBI



**Figura 37:** Grafico Circulas de las provincias que más han twitteado



### C. Juegos en línea por países.

Con el cvs del database que extraemos de Facebook de los países como Bolivia, Cuba, Uruguay, etc se analizó el número de jugadores por millones que existen en cada país e igualmente previamente se realizó la debida limpieza de datos.

Con el segundo dataset se puede analizar:

- Country: país de origen
- Population: población por millones en el país en el campo Country.
- Languages: el idioma que hablan los pobladores del país.
- Likes. Los me gustas que obtuvo
- Comments: el número de veces que tubo

Consultas [4]

Nombre consulta	Origen de datos	Tipo de consulta	Tabla	Campos	Criterios	Criterios de consulta
juegos_bolivia	Borrador	Diseño de consultas	juegos	juegos	juegos	juegos
juegos_cuba	Borrador	Diseño de consultas	juegos	juegos	juegos	juegos
juegos_universo	Borrador	Diseño de consultas	juegos	juegos	juegos	juegos
juegosOnline	Tabla	Acción	juegosOnline	juegosOnline	juegosOnline	juegosOnline

**Figura 38:** Base de Datos CVS de los JuegosOnline

*D. Tema definido por el estudiante*

El data set escogido está limpio por lo que no hubo la necesidad de limpiar los datos. Para una mejor comprensión se transfirió el data set a phpMyadmin donde podemos observar cada uno de los campos con mayor determinación y para realizar las visualizaciones se utilizó la herramienta de Tableau.

```
import couchdb
from TikTokApi import TikTokApi

'''=====CONEXION COUCHDB====='''
server = couchdb.Server('http://admin:Joel$2021*@localhost:5984/')
try:
    db = server.create('juegos_ec')
except:
    db = server['juegos_ec']

'''=====INSTANCIA A LA API====='''
api = TikTokApi.get_instance()

'''=====TERMINOS DE BUSQUEDA====='''
search_term = ["fornite", "Ecuador"]
tiktoks = api.search_for_hashtags(search_term, count=30)

'''=====INSERCIÓN DE DATOS====='''
for tiktok in tiktoks:
    doc = db.save(tiktok)
    print(tiktok)
```

**Figura 39:** Script conexión a Couch DB.

```

1 # Import packages
2 import argparse
3 import time
4
5 # Create parser
6 parser = argparse.ArgumentParser()
7 parser.add_argument('--url', type=str, required=True, help='URL of the server')
8 parser.add_argument('--port', type=int, required=True, help='Port of the server')
9
10 # Parse arguments
11 args = parser.parse_args()
12 url = args.url
13 port = args.port
14
15 # Create connection
16 connection = psycopg2.connect(database='postgres', user='postgres', password='postgres', host=url, port=port)
17 cursor = connection.cursor()
18
19 # Create table
20 cursor.execute('CREATE TABLE IF NOT EXISTS test (id SERIAL, name VARCHAR(255))')
21
22 # Insert data
23 for i in range(1000):
24     cursor.execute('INSERT INTO test (name) VALUES (%s)', ('test %s' % i,))
25
26 # Close connection
27 cursor.close()
28 connection.close()
29
30 # Print message
31 print('Data inserted successfully')
32
33 # End of script

```

**Figura 40:** Script conexión a SQL Server.

### E. Eventos o noticias mundiales

1) Facebook

Los obtuvimos los datos mediante la red social Facebook, los cuales solo se analizó las estadísticas de los me gusta, reacciones, comentarios, publicaciones.

## 2) Kaggle

Los datos obtenidos de la página Kaggle, relacionados con las vacunas contra el Covid-19 se los almaceno en SQLite en una base de datos llamada “países\_vacunados” y una tabla llamada “vacunados”.

```

C:\Users\lfo\Documents> EPMU_datososqlite3
SQLite version 3.34.1 2021-01-20 14:10:07
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode csv
sqlite> .import country_vaccinations.csv vacunados
sqlite> .schema vacunados
Error: unknown command or invalid arguments: ".schema". Enter ".help" for help
sqlite> .schema vacunados
CREATE TABLE IF NOT EXISTS "vacunados"(
  "country" TEXT,
  "iso_code" TEXT,
  "date" TEXT,
  "total_vaccinations" TEXT,
  "people_vaccinated" TEXT,
  "people_fully_vaccinated" TEXT,
  "daily_vaccinations_raw" TEXT,
  "daily_vaccinations" TEXT,
  "total_vaccinations_per_hundred" TEXT,
  "people_vaccinated_per_hundred" TEXT,
  "people_fully_vaccinated_per_hundred" TEXT,
  "daily_vaccinations_per_million" TEXT,
  "vaccines" TEXT,
  "source_name" TEXT,
  "source_website" TEXT
);
sqlite> .save paises_vacunados.db
sqlite>

```

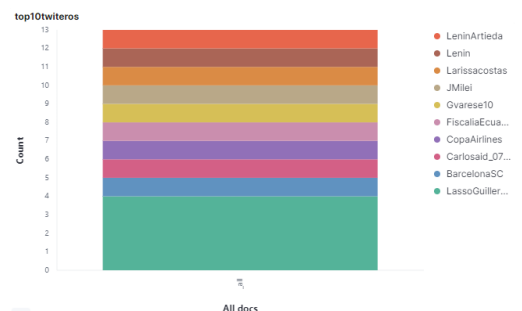
### Figura 41: Convertir de csv a db con SQLite

### 3) Statista

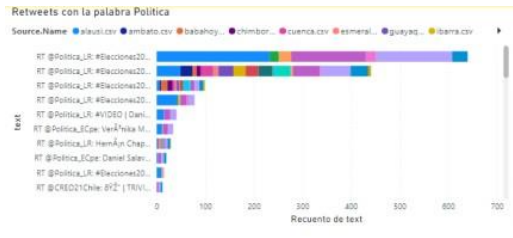
Los datos obtenidos de la página Statista se los analizo manualmente, ya que había celdas que interferían al momento de realizar las gráficas.

## IX. VISUALIZACIÓN DE INFORMACIÓN

A. Pulso político en 20 ciudades principales de Ecuador



**Figura 42:** Top10 usuarios twiteros

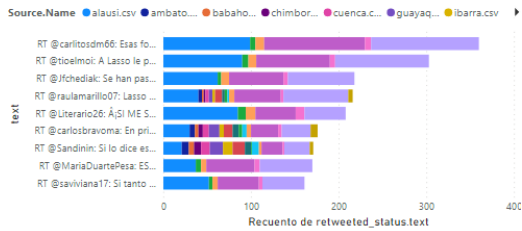


1531

Conteo de Retweets con la palabra Politi

**Figura 43:** Retweets con la palabra Política.

Lasso Nombrado en Twitter

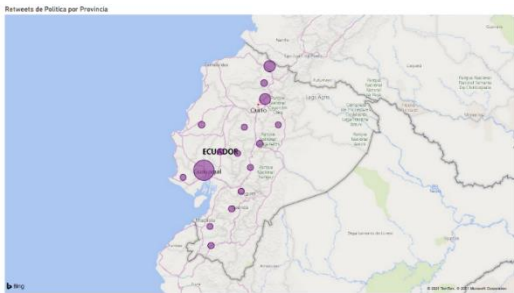


13,39 mil

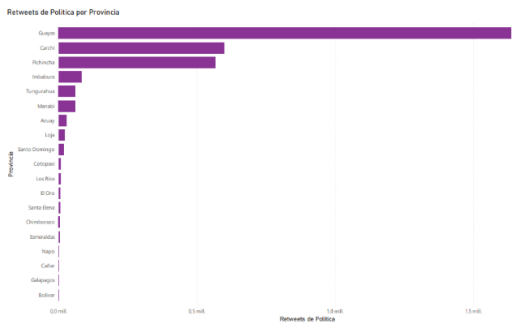
Conteo de Lasso Nombrado en Twitter

**Figura 44:** Conteo en twitter.

## B. Pulso político por provincias en Ecuador

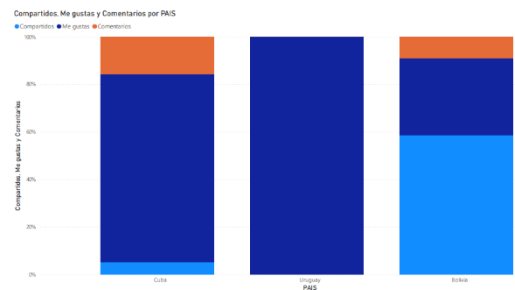


**Figura 45:** Total de Retweets por provincial



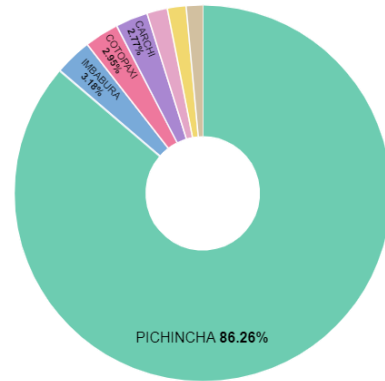
**Figura 46:** Total de tweets en todas las provincias

## C. Juegos en línea por países.



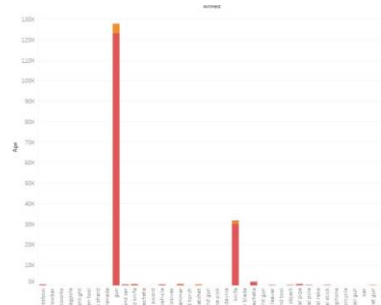
**Figura 47:** JuegosOnline.csv grafica de jugadores en Cuba, Uruguay y Bolivia

## D. Estudiantes por provincia

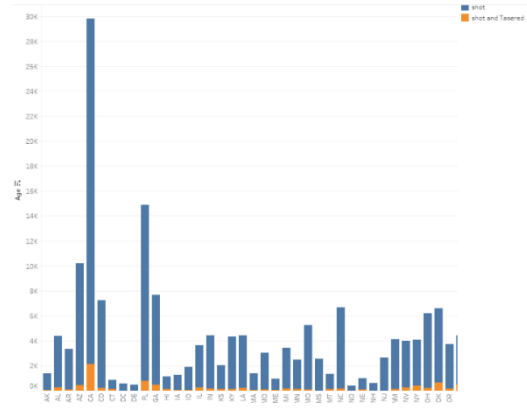


**Figura 48:** resultados de estudiantes matriculado en la EPN por provincias

## E. Tema definido por el estudiante



**Figura 50:** Enfrentamientos.

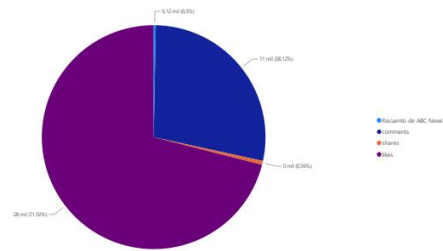


**Figura 51:** Armas que utilizan los delincuentes.

## F. Eventos o noticias mundiales

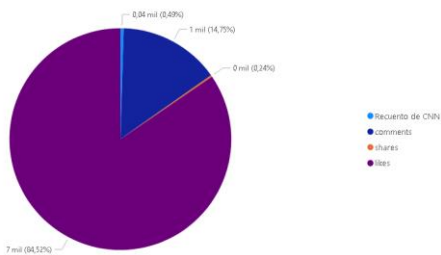
### 1) Facebook

Recuento de ABC News, comments, shares y likes

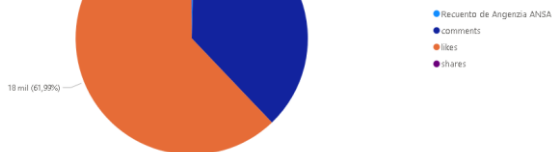


**Figura 52: ABC News**

Recuento de CNN, comments, shares y likes

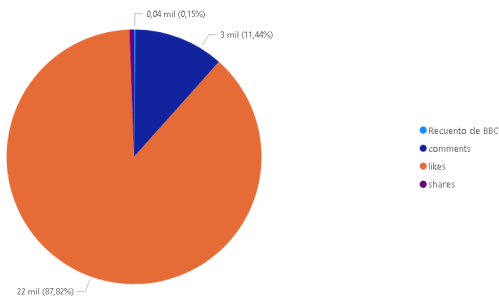


**Figura 53: CNN**



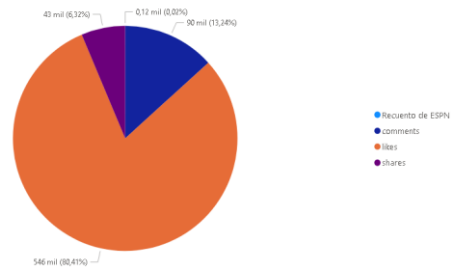
**Figura 54: Agencia ANSA**

Recuento de BBC, comments, likes y shares



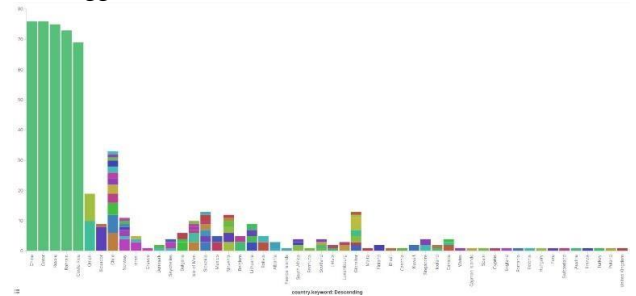
**Figura 55: BBC News**

RECuento de ESPN, COMMENTS, LIKES Y SHARES



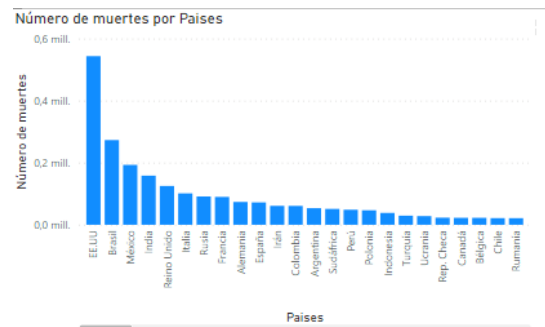
**Figura 56: ESPN**

### 2) Kaggle



**Figura 57: Personas completamente vacunadas por país**

### 3) Statista



**Figura 61 Número de personas fallecidas confirmadas con Covid-19 por país**

## X. RESULTADOS OBTENIDOS

### A. Pulso político en 20 ciudades principales de Ecuador

#### 1) Retweets por ciudad.

Los usuarios de Twitter ubicados en las ciudades de la región Sierra son los que más postean temas relacionados a la corrupción y al nuevo gobierno de Guillermo Lasso.

### *B. Pulso político por provincias en Ecuador*

Se obtuvo que las provincias donde se realizan más tuits y retuits son Guayas y Pichincha. Se alude este resultado a que al ser estas provincias urbanizadas casi en su totalidad cuentan con personas de un estrato social el cual les permite estar en contacto y al día con las noticias que suscitan en redes sociales

### *C. Juegos en línea por países.*

En base a los resultados obtenidos, se puede concluir que los juegos en línea más populares hoy en día son Fornite, Free Fire y Call of Duty. Esto tomando en cuenta que son juegos multiplataforma y por lo tanto más usuarios tienen acceso a la interacción con ellos. El que encabeza la lista es Fornite, dado que la influencia de gamers y streamers hace que este juego tenga una tendencia sobre el público juvenil y parcialmente en adultos.

### *D. Tema definido por el estudiante*

El análisis gráfico de la base de datos matriculados en la EPN permite observar que la mayoría de los estudiantes foráneos provienen de la región sierra, es relativamente pequeño el porcentaje de estudiantes que provienen de la región costa y al ser una universidad pública ubicada en la capital, la gran parte de sus estudiantes pertenecen a Pichincha.

### *E. Eventos o noticias mundiales*

En relación con las noticias y fuentes de información, el análisis de este caso de estudio permitió concluir que el público tiene una tendencia a receptar y poner su credibilidad en el medio de comunicación ESPN.

Los resultados obtenidos de los países que son vacunados diariamente. Dando como primer lugar al país de Qatar, con más personas vacunadas diariamente.

## **XI. CONCLUSIONES Y RECOMENDACIONES**

Python es un lenguaje de programación que nos permite la recolección y tratado de datos de una manera rápida y eficaz, es de gran ayuda para recopilar datos e insertarlos en bases de datos relacionales como las no relacionales. Este proceso de recolección de datos permite capturar información que puede ser utilizada posteriormente para obtener información relevante acerca de algún tema en específico.

Gracias al análisis de las gráficas se puede analizar que la población oriental tiende a tener menor índice de población gamer en plataformas

online, y que los países que más generan ingresos económicos son los países latinoamericanos.

Las herramientas como Power BI y Tableau ayudan a tener una mejor visualización de los datos recolectados ya que facilita el entendimiento y el aprendizaje.

## **XII. DESAFÍOS Y PROBLEMAS ENCONTRADOS**

Un problema para resaltar es la recopilación de datos por geolocalización en la red social Twitter, puesto que en ciertas provincias no se tuitea o no cuentan con una conectividad a internet y no se pudo recolectar la cantidad de datos propuestos.

Otro problema al realizar una importación directamente a SQL Server es que de vez en cuando los dataset que se usan se crean dentro de las veces como caracteres tipo String lo que limita al momento de querer crear estadísticas en el análisis.

La recolección de datos es una de las tareas más complicadas ya que se debe establecer ciertos parámetros y seleccionar bien a nuestro objetivo porque el resultado obtenido va a ser empleado posteriormente para un análisis. Su almacenamiento y debe ser acorde a las herramientas que vamos a emplear para su análisis y la visualización de datos.

## **XIII. ENLACE DE GITHUB DEL PROYECTO**

<https://github.com/JoelDiaz93/ProyectoFinal-AD>

## **XIII. REFERENCIAS**

- [1] Available:  
<http://www.halcyon.com/pub/journals/21ps03-vidmar>
- [2] Apache.(2020). Logstash [Online]. Available:  
<https://www.elastic.co/es/logstash>
- [3] A.Robledano.(2020). ¿Qué es MySQL? [Online]. Available:  
<https://openwebinars.net/blog/que-es-mysql/>
- [4] Apache.(2020). Kibana [Online]. Available:  
<https://www.elastic.co/es/what-is/kibana>
- [5] M.Alvarez. (2003). ¿Qué es Python?[Online]. Available:  
<https://desarrolloweb.com/articulos/1325.php>
- [6] IONOS. (2020). CouchDB [Online]. Available:  
<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/presentacion-de-couchdb/>
- [7] Elastic. (2021). Elasticsearch [Online]. Available:  
<https://www.elastic.co/es/what-is/elasticsearch>