Proyecto Final Análisis

Jhordy Aguas, Melani Barrera, Cristian Paredes, Dilan Obando, Escuela Politécnica Nacional (EPN),
Quito – Ecuador
melani.barrera@epn.edu.ec
cristian.paredes@epn.edu.ec
jhordy.aguas@epn.edu.ec
dilan.obando@epn.edu.ec

***Resume*n – El proyecto tiene como objetivo principal desarrollar una arquitectura de Data Lake que integre datos provenientes de al menos 12 fuentes diferentes, incluyendo redes sociales, webscraping y archivos estáticos. Para lograr esto, se utilizarán diversas bases de datos NoSQL y relacionales para almacenar y gestionar los datos recopilados. Como concentrador de datos, se empleará Microsoft SQL Server, MySQL o MongoDB Atlas, conectado a PowerBI para generar dashboards interactivos y visualizaciones dinámicas. Se realizarán análisis de datos utilizando técnicas como minería de textos, análisis de sentimientos y análisis geoespacial para extraer información relevante**

I. Introducción

recopilar, almacenar y analizar datos de diversas fuentes se ha vuelto crucial para la toma de decisiones efectiva en una amplia gama de campos. Con el fin de aprovechar al máximo esta vasta cantidad de datos, se requiere una infraestructura robusta que permita la integración y el análisis eficientes de múltiples fuentes de datos. Este proyecto se centra en el diseño e implementación de una arquitectura de Data Lake que abarca una variedad de fuentes de datos, desde redes sociales como Facebook y Twitter hasta archivos estáticos como Kaggle e INEC. A través de esta arquitectura, se busca generar insights y visualizaciones valiosas para abordar diversas temáticas de interés, como el pulso político en Ecuador, el comportamiento de los juegos en línea a nivel mundial y eventos relevantes a nivel global.

II. Desarrollo de Contenidos

- Definición del caso de estudio

  El caso de estudio se centra en analizar y comprender diversas temáticas mediante el uso de un Data Lake, que integra datos de múltiples fuentes, incluyendo redes sociales, webscraping y archivos estáticos.
  Metodología: Recopilación de Datos: Se recopilarán datos de al menos 12 fuentes diferentes, incluyendo redes sociales como Facebook, Twitter, TikTok y LinkedIn, así como webscraping de sitios relevantes y archivos estáticos como Kaggle e INEC.
  Arquitectura de Data Lake: Se diseñará una arquitectura de Data Lake que incluirá al menos 3 bases de datos NoSQL y 3 bases de datos relacionales para almacenar los datos recopilados. Microsoft SQL Server, MySQL o MongoDB Atlas se utilizarán como concentrador de datos para centralizar y gestionar la información.
  Visualización de Datos: Se establecerán conexiones entre el concentrador de datos y PowerBI para generar dashboards interactivos y visualizaciones dinámicas que permitan explorar y analizar los datos de manera efectiva.
  Análisis de Datos: Se aplicarán técnicas de análisis de datos, como minería de textos, análisis de sentimientos y análisis geoespacial, para extraer información relevante de los datos recopilados.
  Interpretación de Resultados: Se interpretarán los resultados obtenidos de los análisis realizados y se presentarán conclusiones y recomendaciones basadas en estos hallazgos, con el objetivo de proporcionar información valiosa para la toma de decisiones en las áreas temáticas seleccionadas.

- Objetivo General
  - Investigar y analizar diversas temáticas utilizando un Data Lake que integra datos de múltiples fuentes, con el fin de generar insights y visualizaciones útiles para la toma de decisiones

- Objetivos específicos
  - Recopilar datos de al menos 12 fuentes diferentes, incluyendo redes sociales (Facebook, Twitter, TikTok, LinkedIn), webscraping, archivos estáticos (Kaggle, INEC, etc.).
  - Diseñar una arquitectura de Data Lake que incluya al menos 3 bases de datos NoSQL y 3 bases de datos relacionales para almacenar los datos recopilados.
  - Crear conexiones entre el concentrador de datos y PowerBI para generar dashboards interactivos y visualizaciones dinámicas.
  - Aplicar técnicas de análisis de datos, como minería de textos, análisis de sentimientos y análisis geoespacial, para extraer información relevante de los datos recopilados.
  - Interpretar los resultados obtenidos de los análisis realizados y presentar conclusiones y recomendaciones basadas

en estos hallazgos.

- Descripción del equipo de trabajo y actividades realizadas por cada uno.

  Las actividades que realizo cada uno fue Extracción por Cristian Paredes, la transformación Melani Barrera y la carga por Jhordy Aguas, y la visualización por Dilan Obando.
  Función para limpiar los caracteres no deseados de una cadena y convertirlo de csv a json def limpiar cadena(cadena):

- Cronograma de Actividades

Lunes 26: empezamos a buscar las fuentes de los datos, de los diferentes temas.

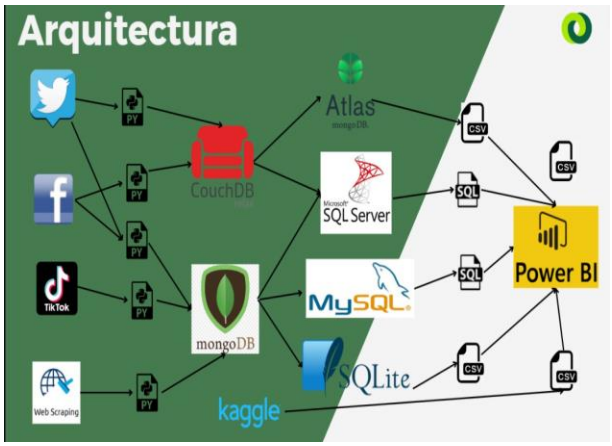
Miércoles 28: La conexión entre la base de datos relacionales y bases de daros NoSql.

Viernes: realizar los dashboards de cada caso analizado.

Sábado: Revisamos lo que ya teníamos realizado, y comenzamos a analizar cada parte y mejorar si fuese el caso.

Domingo y Lunes 04 de marzo realizamos el video de la explicación del análisis de los casos de estudio, cual fue su proceso y como lo obtuvismos.

- Arquitectura



- Recurso y Herramientas utilizadas.

  Bases de Datos Relacionales, Bases de Datos NoSQL, Frameworks de Desarrollo, Couch DB, Power BI
  Visual Studio Code.

Ilustraciones Código del proceso

```
# LIBRERIAS
# Twitter
import tweepy    Import "tweepy" could not be resolved
import couchdb    Import "couchdb" could not be resolved
```

```
# CONSTANTES
limit = 1

ciudades = ['Guayaquil', 'Quito', 'Cuenca', 'Santo Domingo', 'Machala', 'Durán', 'Manta', 'Portoviejo', 'Loja', 'Ambato',
            'Esmeraldas', 'Quevedo', 'Riobamba', 'Milagro', 'Ibarra', 'La Libertad', 'Babahoyo', 'Sangolquí', 'Daule',
            'Latacunga']
paises = ['Ecuador', 'Colombia', 'Peru', 'España', 'México', 'Bolivia', 'Argentina', 'Cuba', 'Panama', 'Venezuela',
          'Chile', 'Paraguay', 'Uruguay']
```

```
# CREDENCIALES
# CouchDB
cdbURL = "http://Cristian:123456@localhost:5984/"
# MongoDB
mdbURL = "mongodb://localhost:27017"
# MongoDBAtlas
mdbAtlasURL = "mongodb+srv://cristian1:Andygago77-18@cluster0.zvjujrq.mongodb.net/File0?retryWrites=true&w=majority"
# MYSQL
usermySQL = 'root'
passwordmySQL = '123456'
# Twitter
bearertoken = "AAAAAAAAAAAAAAAAAAAAAPV5sgEAAAAATEM%2Bl8RMMSFcgWmgnjiIr0bt1s%3D4zzyAoVQNFvvahlh8AwoEIohwb9TJogYN6WVm1M9swKvDTSoji"
ckey = "g0Lk6fb0KUT9G532hf8PfufMI"
csecret = "AwFl7im14ORxpOtkjhZgtv2YsXGJv9AxO1kr9244/kIjcWv1mA"
atoken = "1763995073319124704-oiSrvjtcesTrhFeE0ZcZRQlTotMeRt"
asecret = "MFOPdTJBsirlHkIOIcGdyQBIocuvbN6yQy7yc1B1Z77PV"
# SQL Server
servidorSQLServer = "CRISTIAN\SQLEXPRESS"
userSQLServer = ""
passwordSQLServer = ""
```

```
# FUNCIÓN PARA OBTENER DATOS DE TWITTER DE JUEGOS POR PAISES
# Devuelve una lista de diccionarios
def generateTwitterDataByCountry(pais):
    topic = 'videojuegos'

    if pais == "Ecuador":
        direccion = [-92.21,-5.02,-75.19,1.88]
    elif pais == "Colombia":
        direccion = [-82.12,-4.23,-66.85,16.06]
    elif pais == "Peru":
        direccion = [-84.64,-20.2,-68.65,-0.04]
    elif pais == 'España':
        direccion = [-18.39,27.43,4.59,43.99]
    elif pais == 'México':
        direccion = [-118.6,14.39,-86.49,32.72]
    elif pais == 'Bolivia':
        direccion = [-69.65,-22.9,-57.45,-9.67]
    elif pais == 'Argentina':
        direccion = [-73.6,-55.2,-53.6,-21.8]
    elif pais == 'Cuba':
        direccion = [-85.17,19.63,-73.92,23.48]
    elif pais == 'Panama':
        direccion = [-83.05,7.03,-77.14,9.87]
    elif pais == 'Venezuela':
        direccion = [-73.35,0.65,-59.54,15.92]
    elif pais == 'Chile':
        direccion = [-109.7,-56.7,-66.1,-17.5]
    elif pais == 'Paraguay':
        direccion = [-62.64,-27.61,-54.26,-19.29]
    elif pais == 'Uruguay':
        direccion = [-58.49,-35.78,-53.08,-30.09]
    else:
        print("\nError: No ha ingresado ningún país!\n")

    lista = []

    class listener(tweepy.Stream):
        i = 0
```

```
# FUNCIÓN PARA OBTENER DATOS DE TWITTER DE PULSO POLÍTICO POR CIUDADES
# Devuelve una lista de diccionarios
def generateTwitterDataByCity(ciudad):
    topic = 'política'
    if ciudad == "Guayaquil":
        direccion = [-79.95912,-2.287573,-79.856351,-2.053362]
    elif ciudad == "Quito":
        direccion = [-78.619545,-0.365889,-78.441315,-0.047208]
    elif ciudad == "Cuenca":
        direccion = [-79.5983,-3.1761,-78.8471,-2.5578]
    elif ciudad == "Santo Domingo":
        direccion = [-79.5484,-0.6987,-78.7461,0.0191]
    elif ciudad == "Machala":
        direccion = [-80.02869,-3.354746,-79.84235,-3.190935]
    elif ciudad == 'Durán':
        direccion = [-79.878002,-2.339698,-79.67956,-2.094379]
    elif ciudad == 'Manta':
        direccion = [-80.912795,-1.135691,-80.663985,-0.928689]
    elif ciudad == 'Portoviejo':
        direccion = [-80.5625,-1.202987,-80.316762,-0.929944]
    elif ciudad == 'Loja':
        direccion = [-79.2699,-4.1329,-79.1374,-3.8501]
    elif ciudad == 'Ambato':
        direccion = [-78.937982,-1.470916,-78.5368,-1.1098]
    elif ciudad == 'Esmeraldas':
        direccion = [-79.835349,0.566715,-79.404573,1.048625]
    elif ciudad == 'Quevedo':
        direccion = [-79.616489,-1.194387,-79.368954,-0.939314]
    elif ciudad == 'Riobamba':
        direccion = [-78.723592,-1.710588,-78.595487,-1.633713]
    elif ciudad == 'Milagro':
        direccion = [-79.645094,-2.234423,-79.512052,-1.987807]
    elif ciudad == 'Ibarra':
        direccion = [-78.17586,0.260163,-78.025016,0.499894]
    elif ciudad == 'La Libertad':
        direccion = [-78.554745,-0.234735,-78.520918,-0.211145]
    elif ciudad == 'Babahoyo':
        direccion = [-79.671471,-2.133047,-79.201259,-1.619683]
```

```python
# FUNCIÓN PARA OBTENER DATOS DE FACEBOOK
# Devuelve una lista de diccionarios
def generateFacebookData(topic):
    lista = []
    i = 0
    for post in get_posts(topic, pages=50, extra_info=True):
        i += 1
        dato = {}
        dato["_id"] = post["post_id"]
        mydate = post["time"]
        dato["texto"] = post["text"]
        dato["date"] = mydate.timestamp()
        dato["likes"] = post["likes"]
        dato["comments"] = post["comments"]
        dato["shares"] = post["shares"]
        try:
            dato["reactions"] = post["reactions"]
        except:
            dato["reactions"] = {}
        dato["post_url"]=post["post_url"]
        lista.append(dato)
        if i == limit:
            break
    print("Generación de datos de Facebook completada.")
    return lista
```

```python
# FUNCIÓN PARA OBTENER DATOS DE TIKTOK
# Devuelve una lista de diccionarios
def generateTikTokData(topic):
    scraping = "tiktok-scraper hashtag " + str(topic) + " -n " + str(limit) + " -t all -f " + str(topic)
    os.system(scraping)
    archivo = topic + ".json"
    with open(archivo, encoding = "utf8") as file:
        data = json.load(file)
    lista = []
    for num in range(len(data)):
        dato = {}
        dato["_id"] = data[num]["id"]
        dato["secretID"] = data[num]["secretID"]
        dato["text"] = data[num]["text"]
        dato["createTime"] = data[num]["createTime"]
        dato["webVideoUrl"] = data[num]["webVideoUrl"]
        dato["videoUrl"] = data[num]["videoUrl"]
        dato["videoUrlNoWaterMark"] = data[num]["videoUrlNoWaterMark"]
        dato["diggCount"] = data[num]["diggCount"]
        dato["shareCount"] = data[num]["shareCount"]
        dato["playCount"] = data[num]["playCount"]
        dato["commentCount"] = data[num]["commentCount"]
        lista.append(dato)
    print("Generación de datos de Tiktok completada.")
    return lista
```

```python
# FUNCIÓN PARA OBTENER DATOS DE WEBSCRAPING DEL COVID
# Devuelve una lista de diccionarios
def generateWebScrapingCovid():
    url = "https://www.bbc.com/news/coronavirus"
    lista = []
    req = requests.get(url)
    if req.status_code == 200:
        html = BeautifulSoup(req.text, "html.parser")
        tabla = html.select("#lx-stream")[0]
        articulos = tabla.select("div>ol>li>article")
        for articulo in articulos:
            dato = {}
            if articulo.select_one("div[class*='gel-3']>a>div>img") != None:
                cadena = articulo.select_one("span[id^='title_']").get("id")
                dato["_id"] = cadena[cadena.index("_")+1:]
                dato["titulo"] = articulo.select_one("span[id^='title_']").get_text()
                dato["fecha"] = articulo.select_one("span[class='qa-post-auto-meta']").get_text()
                dato["imagen"] = articulo.select_one("div[class*='gel-3']>a>div>img").get("src")
                dato["articulo"] = "https://www.bbc.com" + articulo.select_one("div[class*='gel-3']>a").get("href")
                dato["texto"] = articulo.select_one("div[class*='gel-5']>p").get_text()
                lista.append(dato)
    print("Generación de Webscraping de datos de covid de la BBC completada.")
    return lista
```

➢ Arquitectura de la solución

Bases de Datos Relacionales: MySQL, PostgreSQL, Microsoft SQL Server.
Bases de Datos NoSQL: MongoDB, Cassandra, Redis.
Plataforma de Nube: AWS, Google Cloud Platform, Microsoft Azure.
Herramientas de Orquestación: Apache Airflow, Apache NiFi.
Contenedorización: Docker, Kubernetes.
Herramientas de Monitoreo: Prometheus, Grafana.
Frameworks de Desarrollo: Flask, Django, Express.js.

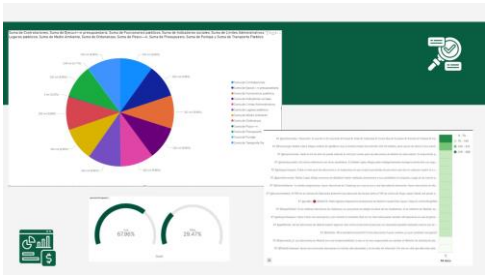
➢ Extracción de datos

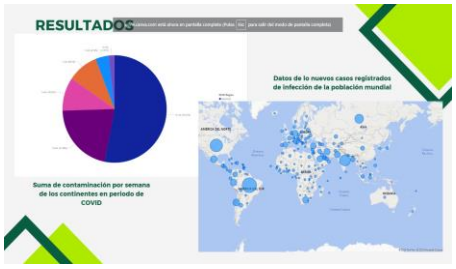
Web Scraping: BeautifulSoup, Scrapy, Selenium.
APIs: Facebook Graph API, Twitter API, TikTok API, LinkedIn API.
Integración de Datos: Apache Kafka, Apache Nifi.
Importación de Archivos: pandas (Python), Spark.
Análisis de Información:

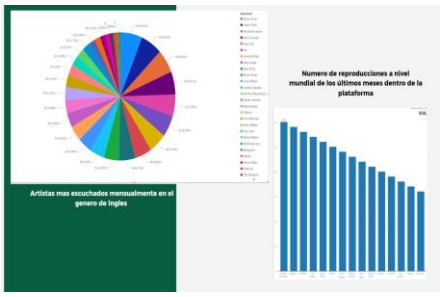
Minería de Texto: NLTK (Natural Language Toolkit), spaCy, TextBlob.
Análisis de Sentimientos: VADER (Valence Aware Dictionary and sEntiment Reasoner), IBM Watson Tone Analyzer.
Procesamiento de Datos: pandas (Python), NumPy, Spark.
Machine Learning: scikit-learn, TensorFlow, PyTorch.
Análisis Geoespacial: GeoPandas, Leaflet.js.
Visualización de Información:

PowerBI, Tableau, QlikView.
Bibliotecas de Visualización: Matplotlib, Seaborn, Plotly (Python), D3.js, Highcharts.
Mapas Interactivos: Google Maps API, Mapbox, Leaflet.js.
Dashboards Interactivos: Dash (Python), Streamlit.

➢ Resultados obtenidos







III. Conclusiones

- La integración de múltiples fuentes de datos, incluyendo redes sociales, webscraping y archivos estáticos, permite obtener una visión holística y completa de los temas analizados.
- El diseño de una arquitectura de Data Lake con múltiples bases de datos NoSQL y relacionales permite gestionar eficientemente la gran cantidad y variedad de datos recopilados, aunque también implica desafíos en términos de mantenimiento y escalabilidad.
- La conexión entre el concentrador de datos y PowerBI facilita la creación de dashboards interactivos y visualizaciones dinámicas, lo que permite explorar y comunicar de manera efectiva los hallazgos obtenidos a partir del análisis de datos.
- Los resultados obtenidos del análisis de datos ofrecen insights significativos sobre el pulso político en diferentes regiones de Ecuador, la popularidad de los juegos en línea en distintos países y otros temas de interés, lo que puede ser de utilidad para la toma de decisiones en diversos ámbitos.

IV. Recomendaciones

- Optimización de la Arquitectura de Datos, continuar evaluando y refinando la arquitectura de Data Lake para mejorar la eficiencia y la escalabilidad, considerando la posibilidad de implementar herramientas de orquestación y automatización para simplificar los procesos de recopilación, almacenamiento y procesamiento de datos.
- Actualización Periódica de los Dashboards y Visualizaciones, mantener los dashboards y visualizaciones actualizadas de forma regular con nuevos datos y análisis, con el fin de garantizar que la información presentada sea relevante y útil para la toma de decisiones en tiempo real.

- Link de github del proyecto.

https://github.com/Cristiann-Paredes/Analisis_Proyecto.git

V. Referencias

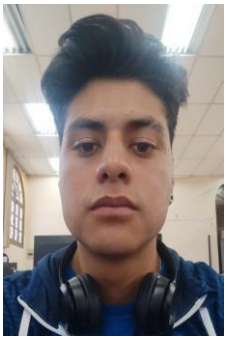
[1]
"Search | Kaggle," *Kaggle.com*, 2024. https://www.kaggle.com/search (accessed Mar. 05, 2024).
[2]
"Facebook - Inicia sesión o regístrate," *Facebook*, 2024. https://www.facebook.com/?locale=es_LA (accessed Mar. 05, 2024).

[3]
"TikTok - Make Your Day," *Tiktok.com*, 2024. https://www.tiktok.com/es/ (accessed Mar. 05, 2024).

VI. BIOGRAFÍAS

Jhordy Aguas, nació en Cayambe Ecuador el 21 de junio de 2002. Realizó sus estudios en el colegio “Nelson Torres”, obteniendo el título de Bachillerato General. En la actualidad es estudiante de Tecnología Superior en Desarrollo de software en la Escuela Politécnica
Nacional. Actualmente está cursando tercer semestre y realiza sus pasantías en PhpOtk. Aspira a ser desarrollador Full-Stack.
(jhordy.aguas@epn.edu.ec)

Melani Barrera, nació en Quito Ecuador el 15 de diciembre del 2001. Realizó sus estudios en el colegio "Gran Colombia ", obteniendo el título de Bachiller en Auxiliar en Contabilidad. En la actualidad es estudiante de Tecnología superior en Desarrollo de software en la Escuela Politécnica Nacional.
(melani.barrera@epn.edu.ec)



Dilan Obando, nació en Quito Ecuador el 20 de abril del 2000. Realizó sus estudios en el colegio "Central Técnico ", obteniendo el título de Bachiller en Electrónica y Control es estudiante de Tecnología superior en Desarrollo de software en la Escuela Politécnica Nacional.
(dilan.obando@epn.edu.ec)

Cristian Paredes, nació en Quito-Ecuador el 18 de junio 2000.

Actualmente se encuentra cursado cuarto semestre de la carrera de desarrollo de software y aspira a ser desarrollador frontend.

(cristian.parede@epn.edu.ec)