UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN

INGENIERÍA CIVIL INFORMÁTICA

GESTIÓN DE DATOS (IN1232C)

Informe técnico

PROYECTO SEMESTRAL 2025-II

Estudiantes: Tomás Fell

Sofía Bastias

Profesor: Lorenzo Paredes

Fecha: 25/11/2025

Introducción

Ante la abundante cantidad de información obtenida de los reportes diarios realizados sobre COVID-19 de los años anteriores, se necesita aplicar distintos métodos de procesamiento de datos para poder elaborar un análisis exhaustivo global del comportamiento entre los contagiados, recuperados y muertos durante cierto periodo de tiempo.

De esta forma será imprescindible que se utilicen librerías como Pandas, NumPy, Matplotlib y Seaborn en Python para poder procesar, analizar y visualizar la gran cantidad de datos de las que dispondremos para este estudio con tal de facilitar la comprensión de estos y posteriormente mejorar el análisis de tendencias epidemiológicas globales, entendiendo su comportamiento ante búsqueda de patrones.

Objetivos:

- Limpieza y preparación de los datos.
- Análisis exploratorio y perfilado.
- Visualizar de forma avanzada los datos.
- Generar Dashboard final con todos los datos anteriores.
- Optimizar y documentar todo el procesamiento.

Desarrollo

Etapa 1: Limpieza y preparación de datos.

Objetivo: Comprender la estructura del dataset y generar una base consolidada y limpia.
Rango temporal: 1 mes de datos.
Dificultad: Básica – operaciones individuales sobre un solo archivo CSV.

1. Cargar y visualizar los primeros 5 registros del archivo 08-02-2021.csv.

Se cargan los primeros 5 registros que datan de la fecha 08-02-2021, ya que se escogió como datos el mes de Agosto del 2021.

2. Mostrar el número total de filas y columnas del DataFrame.

A través de la función .shape se obtuvo que el dataframe DFAgosto2021 contiene 124434 filas y 14 columnas.

3. Describir los tipos de datos (dtypes) y convertir las columnas necesarias (por ejemplo, fechas).

Gracias a la función de .info podemos obtener los Dtype de cada columna. En la Figura 1, podemos ver que Last_Update tiene como Dtype object, así que convertimos la columna a Dtype datetime como se puede ver en la Figura 2.

```
DFAgosto2021.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 124434 entries, 0 to 124433
Data columns (total 14 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   FIPS                 101246 non-null  float64
 1   Admin2               101401 non-null  object
 2   Province_State       118885 non-null  object
 3   Country_Region       124434 non-null  object
 4   Last_Update          124434 non-null  object
 5   Lat                  121644 non-null  float64
 6   Long_                121644 non-null  float64
 7   Confirmed            124434 non-null  int64
 8   Deaths               124434 non-null  int64
 9   Recovered            3704 non-null    float64
 10  Active               3704 non-null    float64
 11  Combined_Key         124434 non-null  object
 12  Incident_Rate        121644 non-null  float64
 13  Case_Fatality_Ratio  123128 non-null  float64
dtypes: float64(7), int64(2), object(5)
memory usage: 13.3+ MB
```

Figura 1

```
 4   Last_Update          124434 non-null  datetime64[ns]
```

Figura 2

4. Detectar y mostrar valores nulos o faltantes por columna.

Con .isnull().sum() obtenemos todos los valores nulos por columna lo que nos permite decidir qué columnas son importantes para nuestros análisis de datos. En este caso, como podemos ver en la Figura 3, FIPS, Admin2, Lat y Long_ son columnas que tienen muchos valores nulos, al igual que Recorvered y Active. Pero a diferencia de los dos mencionados, los primeros tres no tienen tanta relevancia a la hora de realizar un análisis de los datos, por lo cual podemos descartarlos sin problema. tendrán tanta relevancia en el estudio, por lo cual las eliminaremos

	0
FIPS	23188
Admin2	23033
Province_State	5549
Country_Region	0
Last_Update	0
Lat	2790
Long_	2790
Confirmed	0
Deaths	0
Recovered	120730
Active	120730
Combined_Key	0
Incident_Rate	2790
Case_Fatality_Ratio	1306

Figura 3

5. Eliminar columnas irrelevantes (por ejemplo, códigos FIPS o coordenadas si no se usarán).

Como dijimos anteriormente, descartamos las tres columnas mencionadas con la función .drop para eliminar columnas. Y en el caso de las columnas Recovered y Active, se reemplaza los valores NaN por 0, así tenemos un mejor cálculo cuando obtengamos la columna active_cases.

6. Estandarizar nombres de columnas (usar formato snake_case).

Ahora vamos a quitar cualquier mayúscula, espacio o carácter especial para solo dejar el nombre de la columna de forma que está todo normalizado, en minúsculas y separada con guión bajo como se ve en la Figura 4.

```
Index(['admin2', 'province_state', 'country_region', 'last_update',
       'confirmed', 'deaths', 'recovered', 'active', 'combined_key',
       'incident_rate', 'case_fatality_ratio'],
      dtype='object')
```

Figura 4

7. Homogeneizar nombres de países (ej. “US” → “United States”).

Se transforman nombres con abreviación como US a United States para tener un mayor entendimiento de los datos.

8. Convertir la columna Last_Update al formato YYYY-MM-DD.

Se transforma la columna de last_update para que tenga el formato de año-mes-días, como se ve en la Figura 5.

	last_update
0	2021-08-02
1	2021-08-02
2	2021-08-02
3	2021-08-02
4	2021-08-02

Figura 5

9. Crear una columna active_cases = Confirmed - Deaths - Recovered.

Se crea la columna active_cases, según la fórmula dada, para mostrarla junto a las demás columnas como se ve en la Figura 6.

case_fatality_ratio	active_cases
4.567427	58178.0
1.845689	421.0
2.486614	52264.0
0.872053	340.0
2.373000	4402.0
3.300077	25.0
2.142935	248943.0
2.005305	5734.0
2.419355	0.0
0.737179	9291.0

Figura 6

10. Guardar el DataFrame limpio como covid_clean_enero2020.csv e indicar su tamaño en MB.

Se usa la función .to_csv para guardar el dataframe en un archivo .csv para posteriormente, importando la librería humanize que contiene la función .naturalsize, obtener el tamaño de este. El archivo termina pesando 13.8745 MB.

Etapa 2: Análisis exploratorio y perfilado.

Objetivo: Explorar la evolución de los datos y realizar análisis comparativos.
Rango temporal: 6 meses de datos
Dificultad: Media – combinar múltiples archivos y realizar cálculos agregados.

En este caso hacemos el mismo procedimiento de limpieza que en la etapa anterior solo que ahora nuestro datos son desde Mayo hasta Octubre del 2021.

1. ¿Cuáles son los 10 países con más casos confirmados acumulados durante el semestre?

Se calcula el máximo número de casos confirmados, ordenados por país y se grafica obtiene el siguiente gráfico.

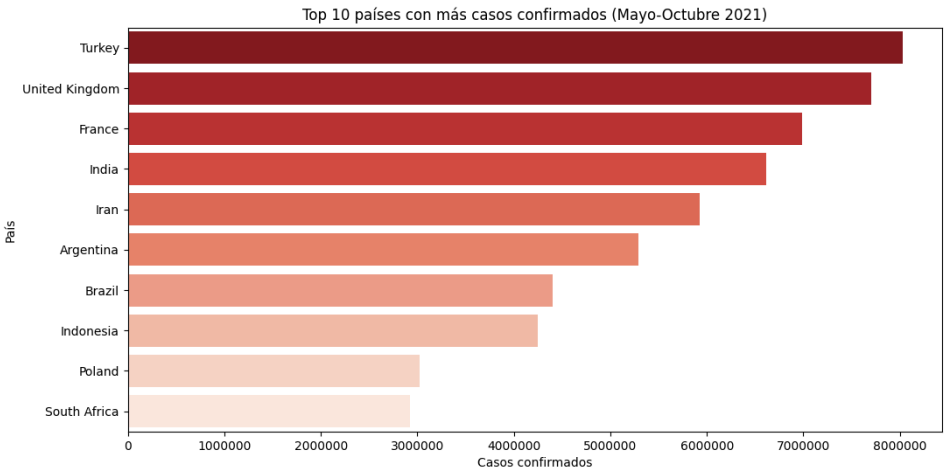

Figura 7

2. ¿Qué países presentan mayor tasa de letalidad (Deaths / Confirmed * 100)?

Se calculó la tasa de letalidad dividiendo la cantidad de muertos por la cantidad de casos confirmados, en caso de no existir casos confirmados se rellena con valores nulos.

country_region	tasa_letalidad
Vanuatu	24.773551
MS Zaandam	22.222222
Yemen	19.313898
Mexico	9.597985
Belgium	9.517121
Peru	8.555551
Sudan	7.463922
Syria	7.047397
Egypt	5.758984
Ecuador	5.707743

Figura 8

3. ¿Cuántos países no registran recuperados en los datos analizados?

Se creó un dataframe auxiliar en el que se agregaron los datos en los cuales la columna de recuperados fuera de 0 o nula y se calculo la cantidad de filas que este tenia dando un total de 201 países que no contaban con recuperados durante el periodo de Mayo a Octubre del 2021

4. ¿Qué país latinoamericano presenta la mayor cantidad de casos activos en junio de 2020?

Se creó un arreglo con los países de Latinoamérica y mediante este se filtró los datos y se sumó los datos de casos activos obteniendo el siguiente gráfico en el que se aprecia que Brasil es el país latinoamericano con más casos activos, superando la suma del resto

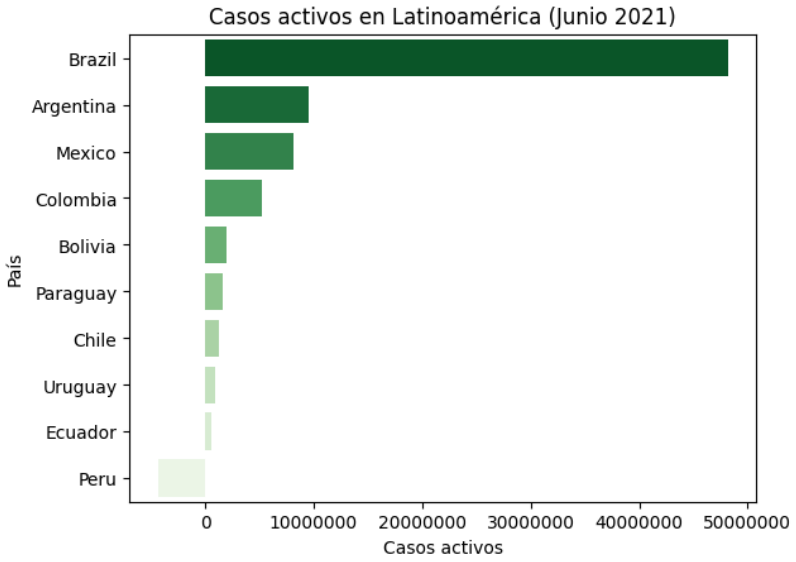

Figura 9

5. ¿Cómo evolucionaron los casos confirmados en Chile entre enero y junio? (gráfico de líneas).

Se tomaron los datos relacionados a Chile y con estos se realizó un gráfico con el que se muestra una tendencia al alza de los casos confirmados.

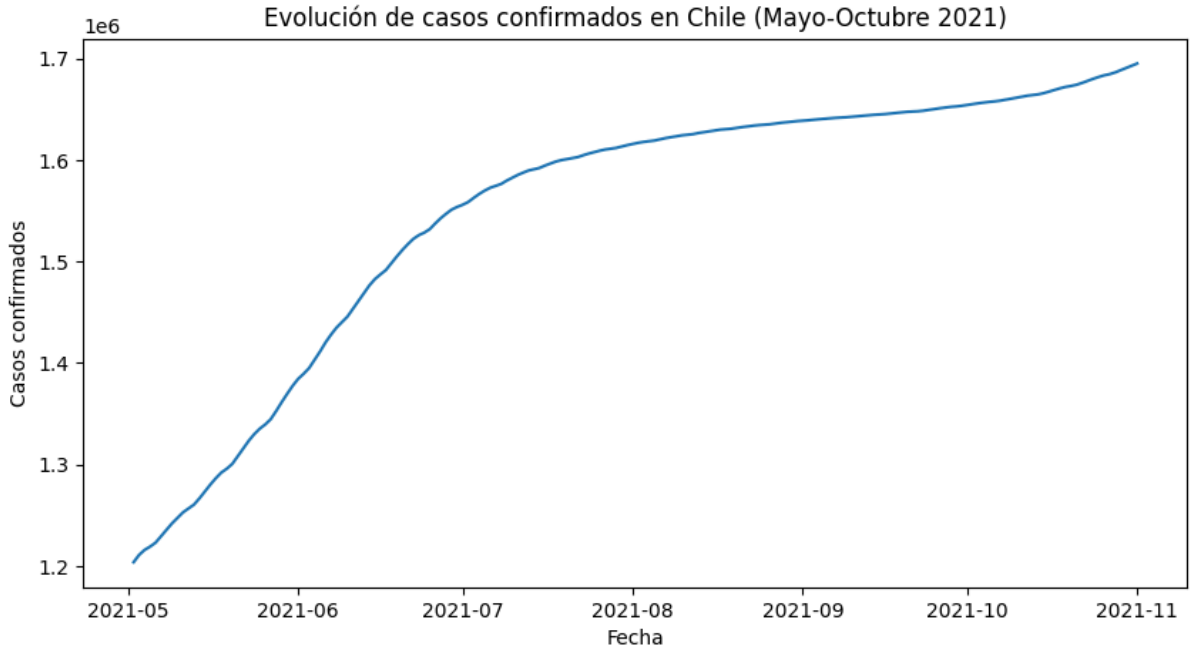

Figura 10

6. ¿Cuál fue la fecha con más nuevos casos a nivel mundial durante este período?

Se analizaron los datos mediante un dataframe auxiliar sumando los datos de los nuevos casos y se obtuvo que el día del periodo con más contagios fue el dos de mayo del 2021.

7. ¿Existe correlación entre casos confirmados y fallecidos? (gráfico de dispersión + regresión).

Se realizó el siguiente gráfico con los datos de casos confirmados y de fallecidos. Y se obtuvo un coeficiente de correlación de 0.89 lo que indica una fuerte correlación positiva entre los caso confirmados y los fallecidos, lo que indica que los test fueron muy confiables y se aplicaron correctamente, pero existieron muy pocos falsos positivos y falsos negativos.

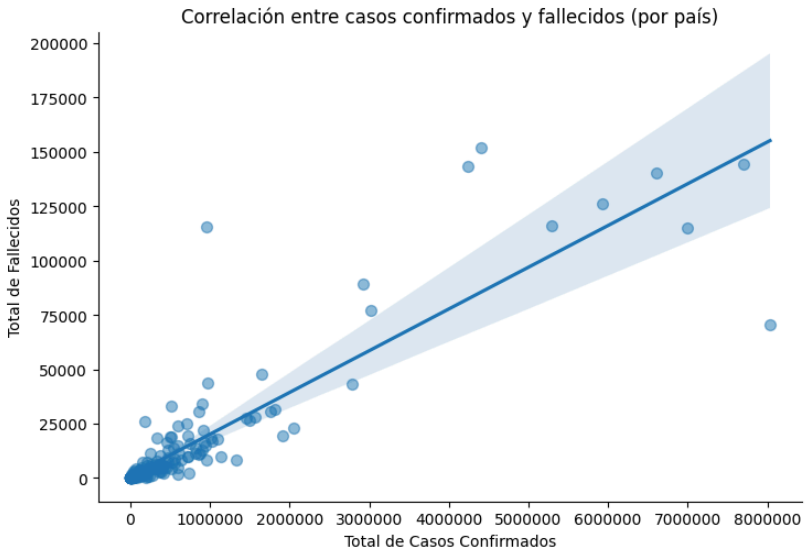

Figura 11

8. Mostrar el Top 10 de países con mayor crecimiento porcentual de casos entre mayo y junio.

Para esto se crearon tres dataframe auxiliares, uno para cada uno de los meses y uno para analizar cuál fue el crecimiento porcentual y posteriormente se realizó un gráfico con los 10 países que mostraron el mayor crecimiento porcentual.

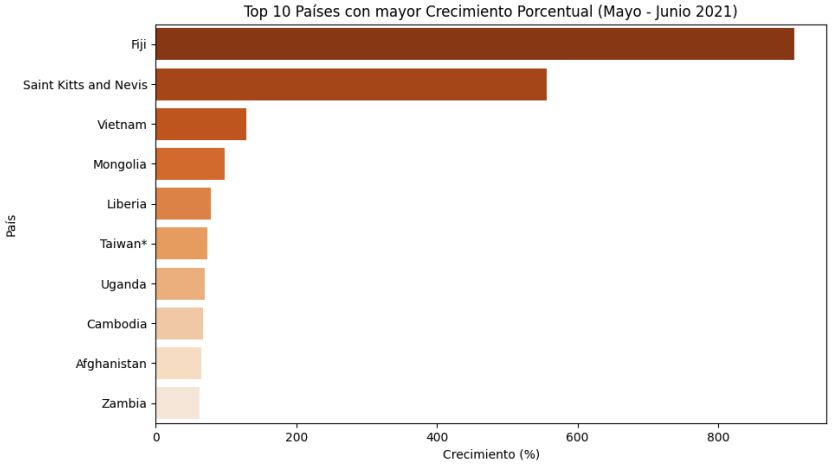

Figura 12

9. Identificar países con rebrote (un día sin casos y luego un incremento posterior).

Se analizó los casos activos, en caso de que se encuentre dos días consecutivos con una diferencia abismal de cantidad de casos activos, esto quiere decir que después de tener un día con 0 casos, pasan a tener más de 0, lo que significa un rebrote. Los países más conocidos que salieron con posibles rebrotes son: Australia, Brasil, Canadá, Chile, China, Colombia, Francia, Alemania, India, Italia, Japón, México, Perú, Rusia, etc.

10. Generar un reporte de perfilado automático (ydata-profiling o pandas_profiling) que incluya distribuciones, correlaciones y resumen de calidad de datos.

Se instaló la librería ydata-profiling y se usó esta para generar el reporte perfilado en un archivo html llamado “perfilado.html” disponible en el repositorio de github.

Etapa 3

Objetivo: Profundizar en el análisis visual, comparando regiones y tendencias.
Rango temporal: 2 años de datos
Dificultad: Media-alta – manipulación de datasets grandes y generación de gráficos agregados.

En este caso hacemos el mismo procedimiento de limpieza que en la etapa 1 solo que ahora nuestro datos son desde inicios del 2021 hasta finales del 2022.

1. Evolución temporal global de casos confirmados, activos y fallecidos (líneas).

Se asignaron las fechas del periodo al eje x y los casos activos, confirmados y fallecidos al eje y, al usar un solo eje x la gran cantidad de casos confirmados ocupa las otras dos.

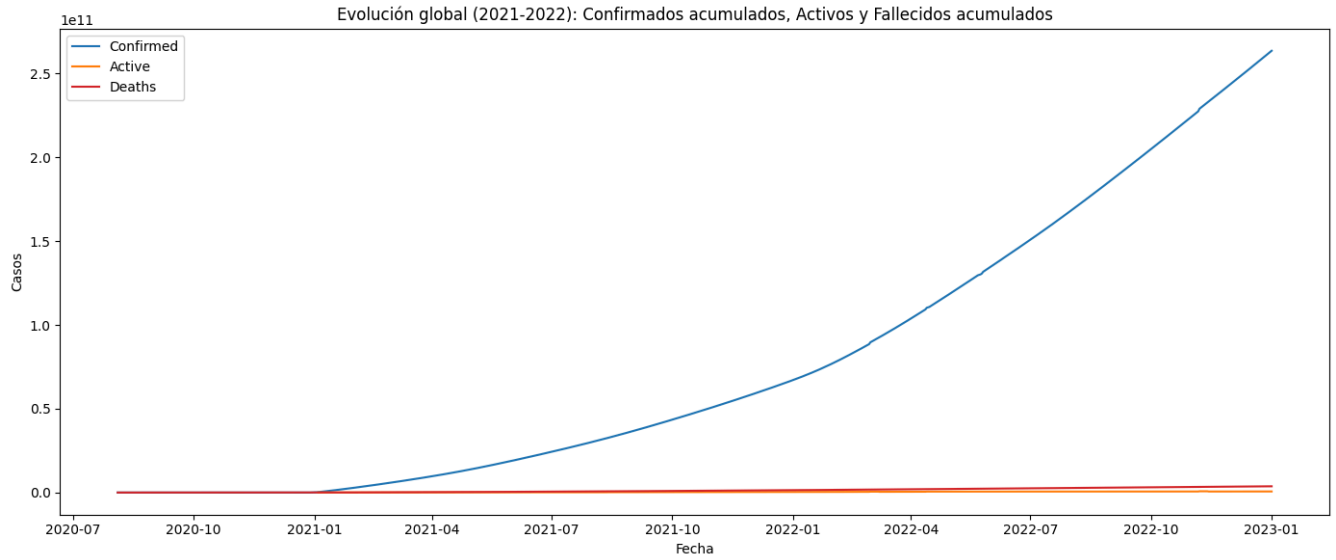

Figura 13

2. Comparativa Top 10 países con más casos confirmados (barras).

Se usó las funciones de graficado asignando los países al eje y la cantidad total de los casos confirmados al eje x, mostrando así solo los 10 países con más casos confirmados.

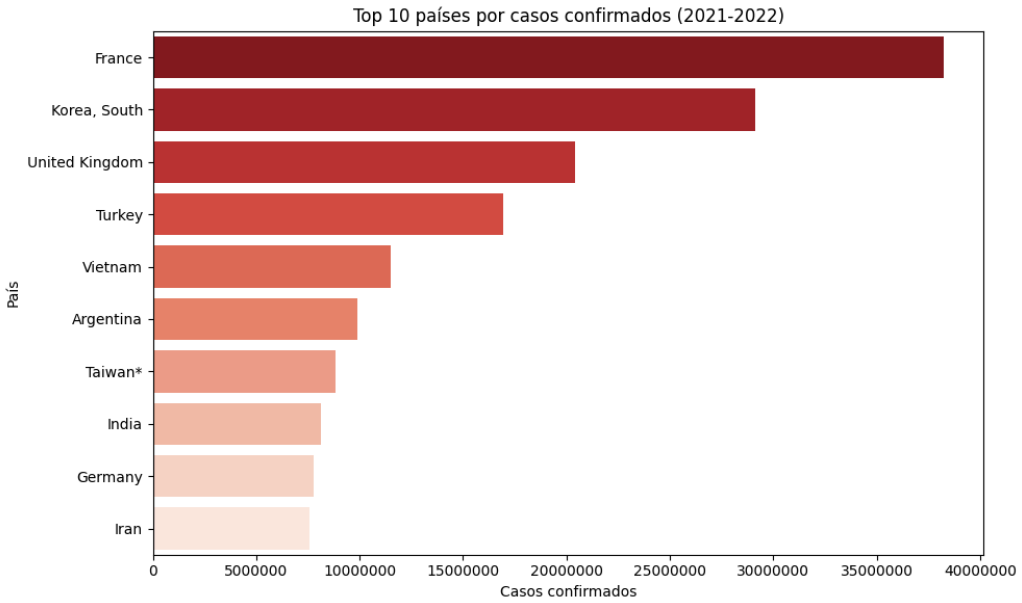

Figura 14

3. Heatmap de correlaciones entre columnas relevantes (confirmados, fallecidos, activos, ratio).

Se tomaron las variables relevantes que se pidieron y se calculó la correlación entre estas, mostrando que con excepción de los fallecidos todos tienen una correlación fuertemente positiva, lo que se explica porque el Covid19 tiene una mortalidad del 2%.

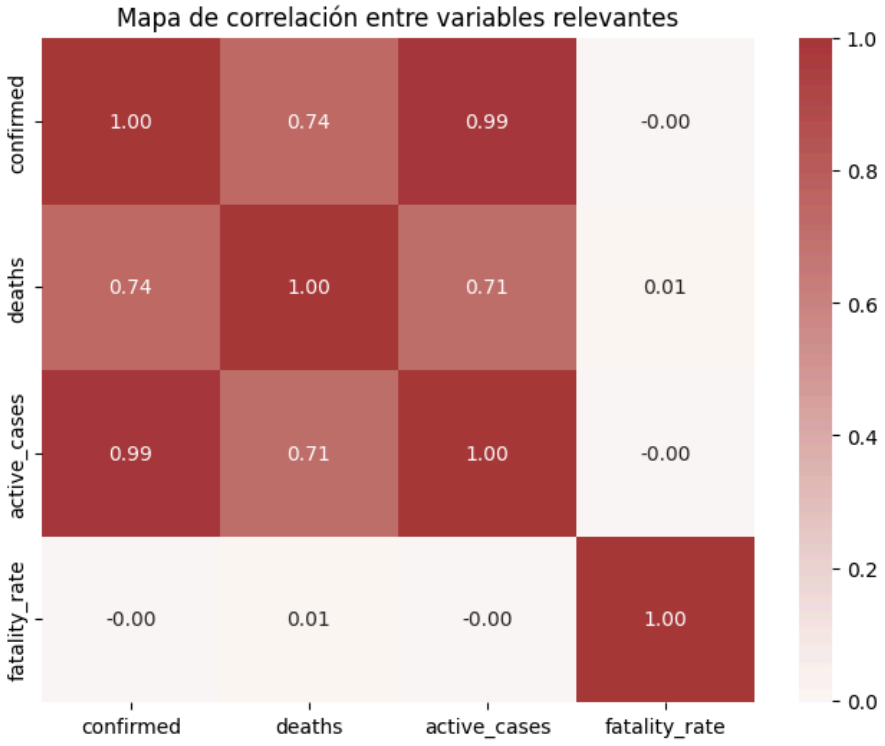

Figura 15

4. Gráfico de barras horizontales comparando tasas de letalidad por continente.

Se creó un diccionario de continentes y mediante este se asignó la columna de continentes al dataframe usando el país como guía, luego se calculó la tasa de letalidad total del periodo por continente.

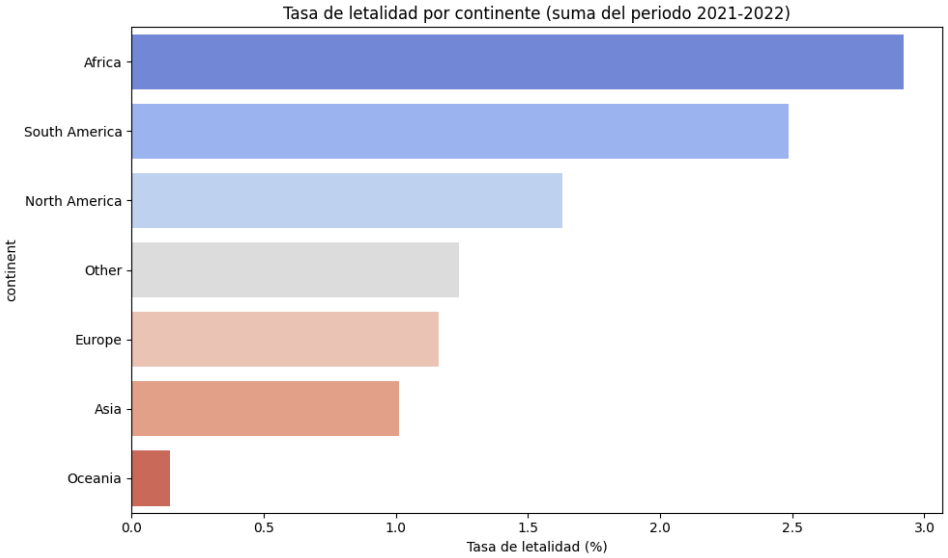

Figura 16

5. Mapa o gráfico geográfico que muestra la incidencia por continente o país (opcional).

Se instalaron e importaron las librerías geopandas y geodataset. Y usando las columnas correspondiente a latitud y longitud(en este dataset no eliminamos las columnas mencionadas anteriormente para poder guiarnos a la hora de graficar el mapa) del dataset para correlacionar con el mapa y el número de casos confirmados.

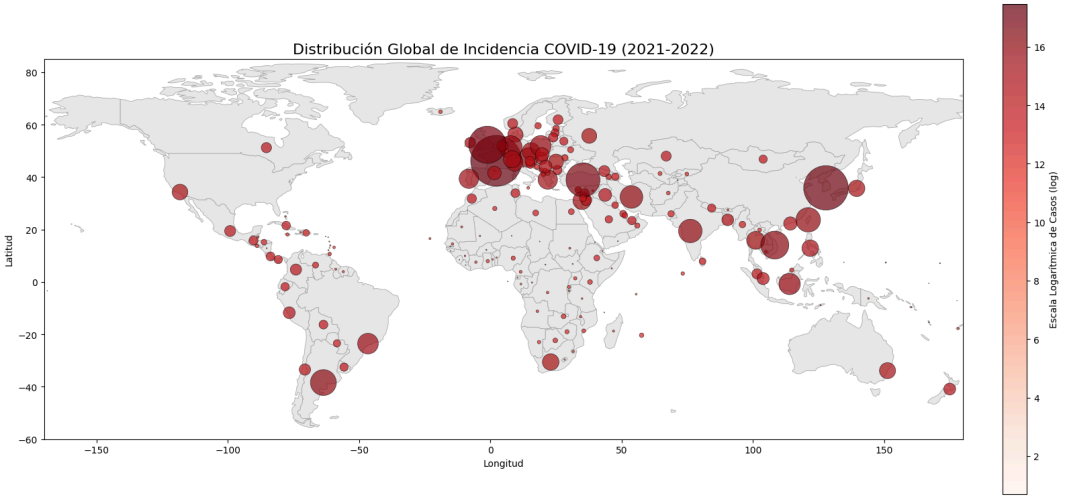

Figura 17

Etapa 4

Objetivo: Integrar análisis, visualizaciones y optimización en una herramienta interactiva.
Rango temporal: Datos globales completos (2020–2022).
Dificultad: Alta – integración total y automatización del flujo de análisis.

Se decide ocupar Streamlit y como servidor ngrok que es de forma gratuita mediante la generación de un Token, para crear un dashboard interactivo que contiene:

Filtros por continente, país y rango de fechas.

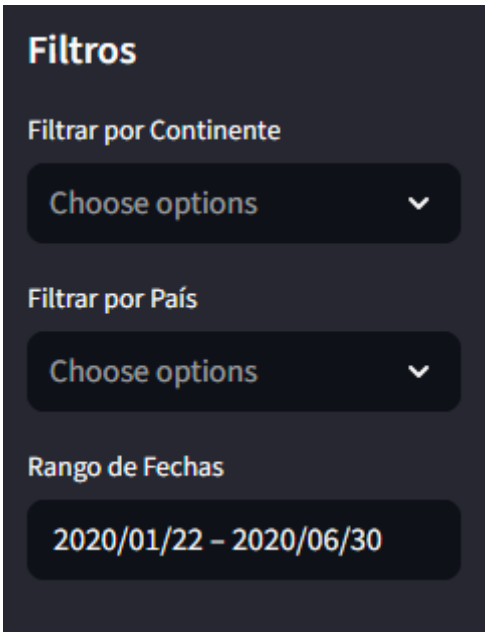

Figura 18

Total confirmados, Total Fallecidos y Tasa de Letalidad.

Figura 19

Gráfico evolutivo de casos confirmados, activos, recuperados y fallecidos.

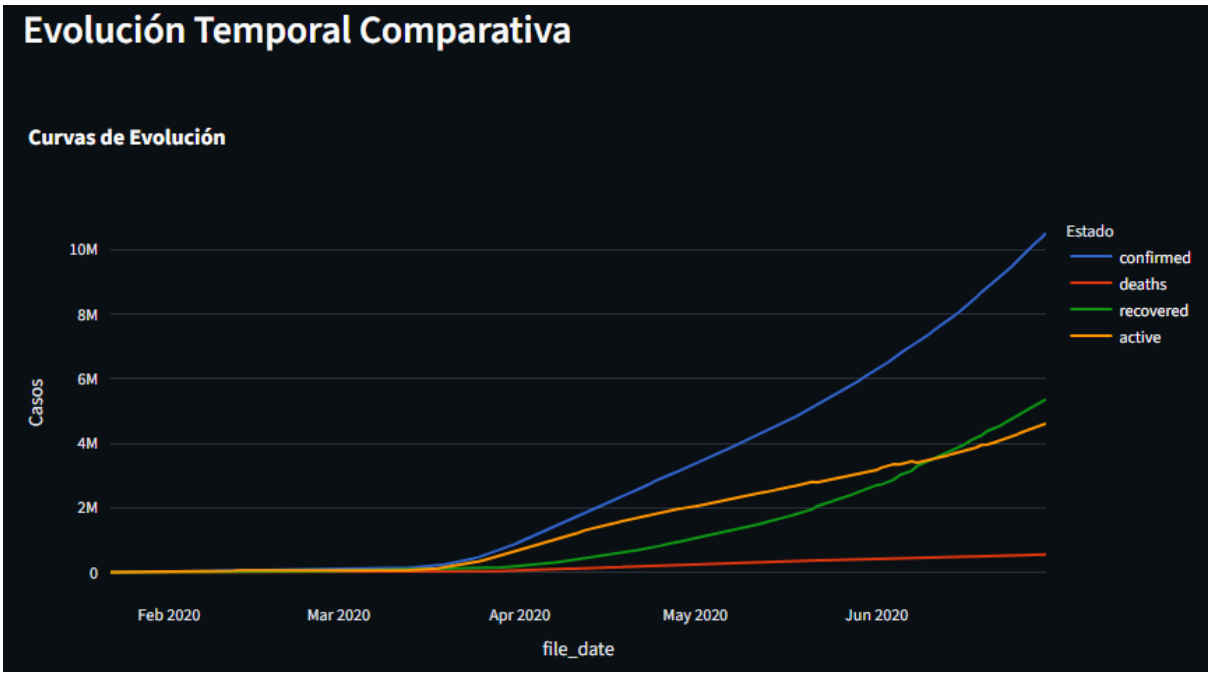

Figura 20

Mapa Global con casos activos.

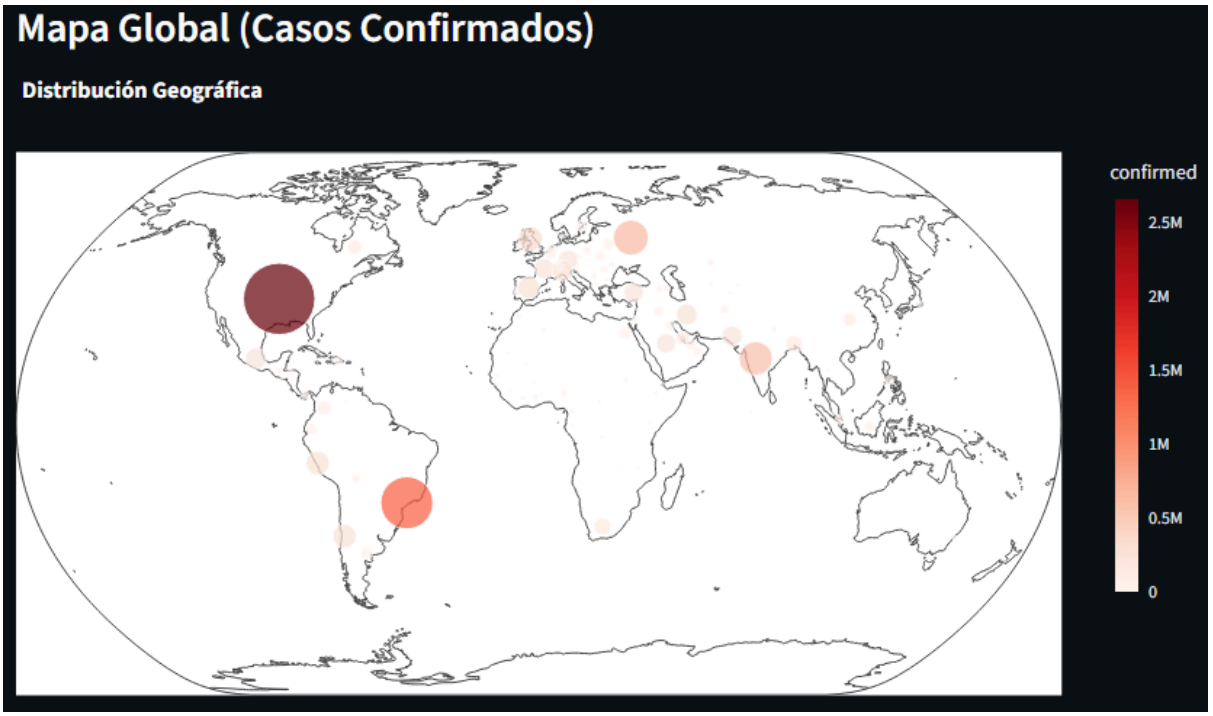

Figura 21

Ranking de Países con mayor casos activos.

Figura 22

Conclusiones automáticas con índices de rebrote y tasa de crecimiento.

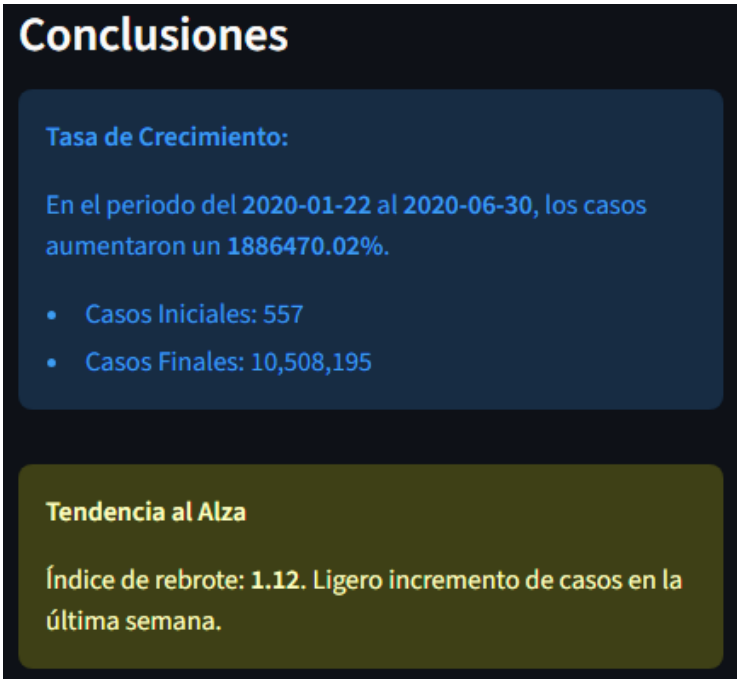

Figura 23

Etapa 5 (implementado en Etapa 3)

Objetivo: Mejorar la eficiencia del código y demostrar dominio técnico.

1. Lectura eficiente

Para la lectura eficiente se usó el método chunksize que ayuda a procesar datasets grandes haciéndolo por partes en vez de procesarlos completos como lo haría pandas normalmente. De esta forma se logra dividir el datasets en fragmentos lo que permite optimizar el uso de la memoria y el tiempo por sobretodo.

En este caso podemos visualizar el tiempo de carga del dataset antes de la optimización siendo de 155.1 segundos. Y después con el método chunksize nos da un tiempo de 92.9 segundos. Esto significa una reducción del 40.1% en tiempo.

2. Conversión de tipos

En el caso de la conversión de tipos se hizo a tres grandes astype con el fin de poder reducir la memoria ocupada. Para las columnas Int64 (confirmed, deaths, recovered, active_cases) las convertimos a Int32 y lo mismo hacemos con las columnas de tipo Float64(incident_rate y case_fatality_ratio) convirtiéndolas en Float32. Ahora para las columnas de tipo object(country_region y province_state) las convertimos al tipo category.

Esto nos hizo pasar de una memoria de uso de 1033.4 MB a un 666.5 MB, lo que significa en una reducción del 35.5% de la memoria inicial usada para el dataset.

3. Uso de índices y operaciones vectorizadas.

Para evidenciar el tiempo que se optimiza entre cada operación usamos de referencia al cálculo de la tasa de fatalidad. Esto nos deja en que al ocupar la operación .apply(que itera fila por fila) ocupa un mayor tiempo de ejecución a diferencia de la función vectorizada de NumPy, siendo su tiempo de ejecución de un 30.7 segundos y la optimizada de un 0.05 segundos. Esto significa una reducción del 99.8% del tiempo de ejecución.

Resultados Principales y Gráficos

A continuación se mostrarán todos los resultados y gráficos(la mayoría mencionadas anteriormente) principales obtenidos en todas las etapas anteriores, obviando al dashboard.

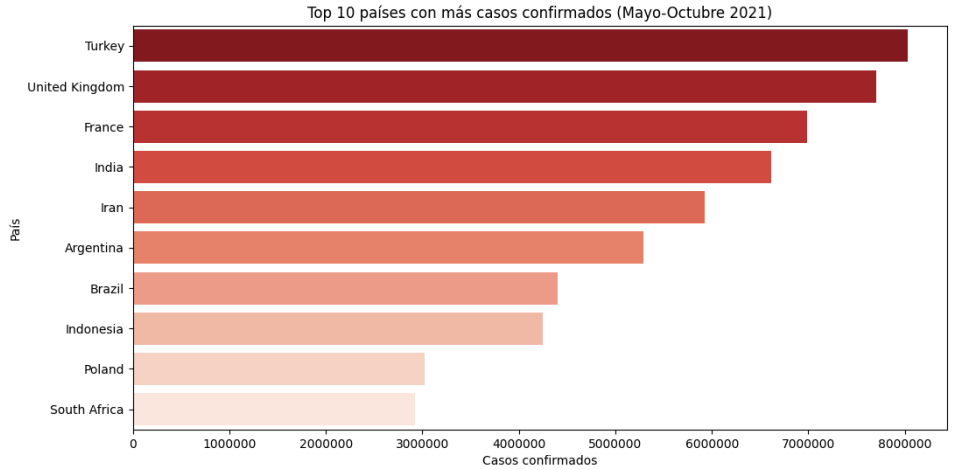

Figura 7

Entre los meses de mayo y octubre del 2021 podemos evidenciar que Turquía encabeza como el país con más casos confirmados durante esa fecha siendo el Top1, seguido por el Reino Unido. Además podemos ver que South Africa es el Top10 de países con más casos confirmados.

country_region	tasa_letalidad
Vanuatu	24.773551
MS Zaandam	22.222222
Yemen	19.313898
Mexico	9.597985
Belgium	9.517121
Peru	8.555551
Sudan	7.463922
Syria	7.047397
Egypt	5.758984
Ecuador	5.707743

Figura 8

Como podemos ver en la tabla, Vanuatu tiene la mayor tasa de letalidad donde entre 10 confirmados, 2 o 3 de ellos mueren por covid.

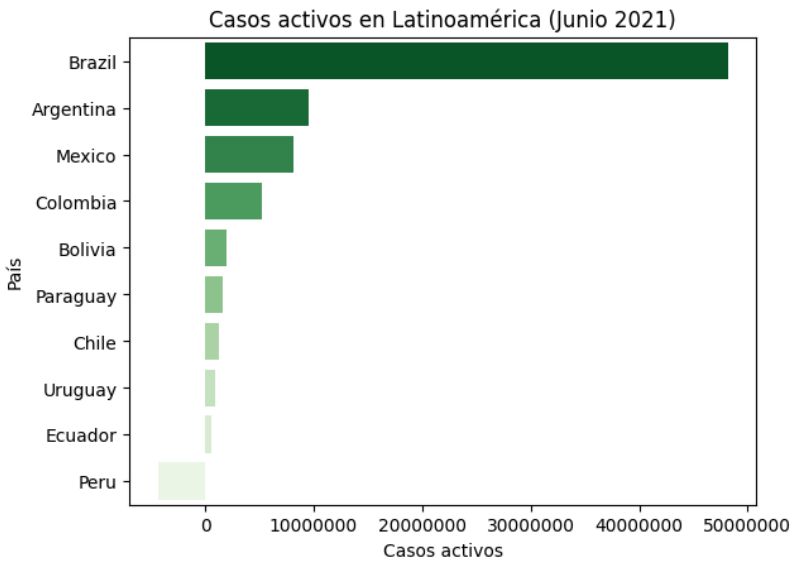

Figura 9

Entre los países latinoamericanos Brasil lleva la delantera como el país con mayor casos activos en junio del 2021, esto siendo congruente con la cantidad de habitantes que hay en cada país. Por ejemplo Argentina igual cuenta con una población grande y se ve reflejada en la cantidad de casos activos, al menos en la fecha señalada.

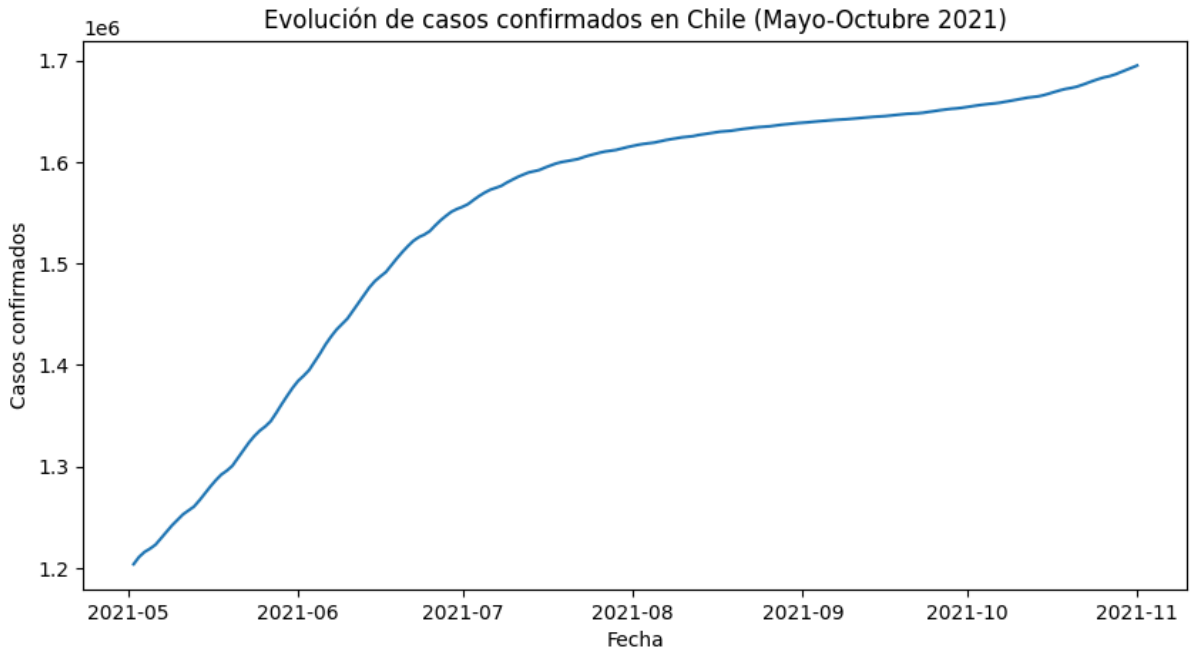

Figura 10

Vemos el comportamiento de los casos confirmados en 2021 desde mayo a octubre de Chile, este crecimiento solo fomenta el alto índice de rebrote que hubo en ese año, ante la vuelta a la normalidad que hubo a nivel nacional en ese momento. Esa vuelta trajo consigo nuevos casos de covid.

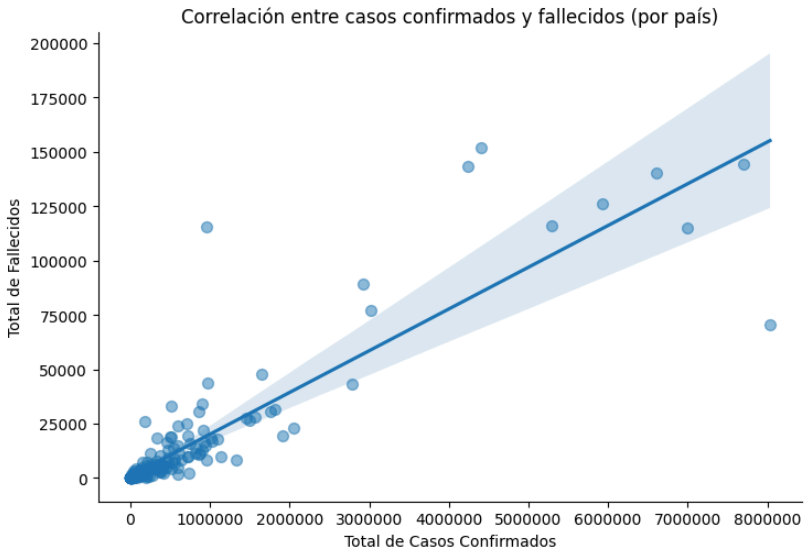

Figura 11

Al analizar la correlación entre los casos confirmados y fallecidos, tenemos un coeficiente de 0.89 lo que significa una relación positiva entre ellos. A medida que los contagios aumentan, el número de fallecidos también va a aumentar.

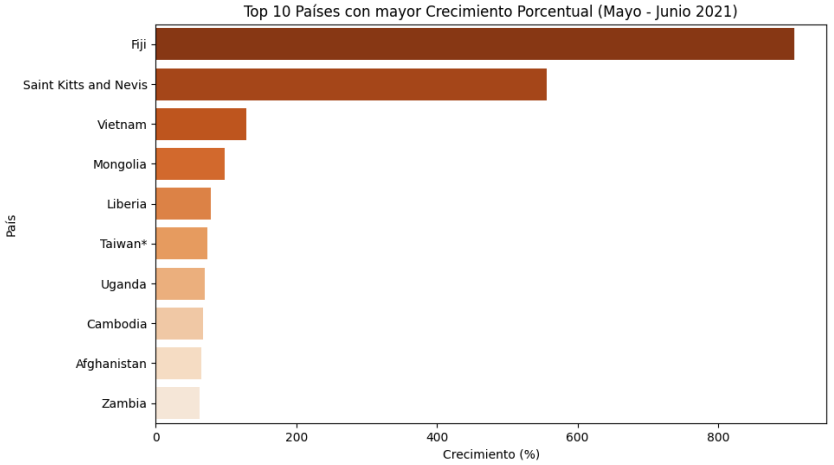

Figura 12

En este top vemos los países que tuvieron un crecimiento porcentual de casos bastante grandes en solo un mes. Esto puede estar considerando la velocidad en la que se contagia una población o como los rebrotes afectan a un país. Por ejemplo la isla Fiji contó con un 800% de crecimiento en solo 1 mes a comparación del segundo puesto que solo tendrá casi un 600% de crecimiento.

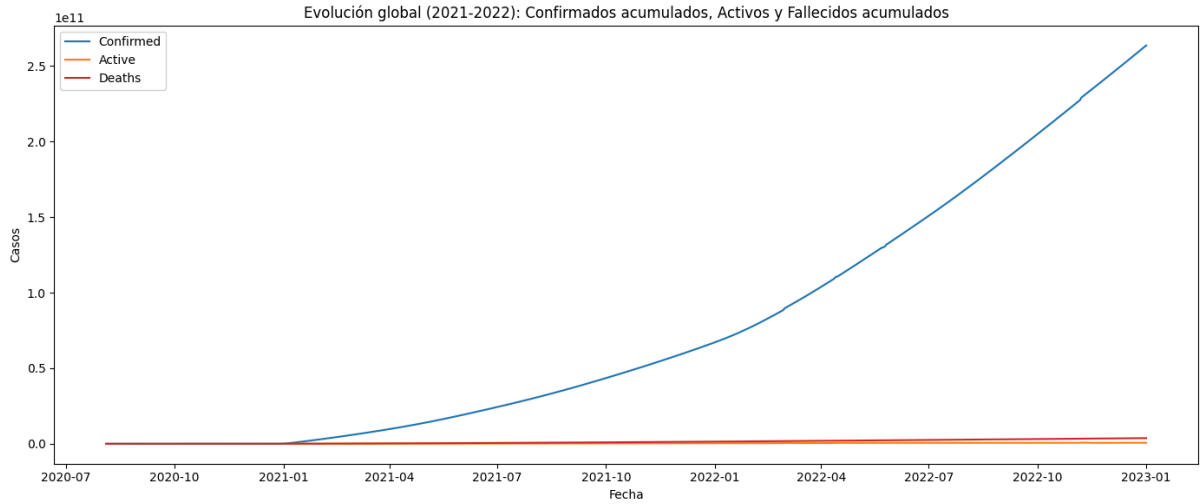

Figura 13

En este gráfico podemos ver la evolución global tanto de los casos confirmados, que permanecen activos y los fallecidos. Con esto podemos concluir que aun si hay muchos casos confirmados, pocos o nadie muere a medida que avanza el tiempo y mucho menos se mantiene activos. Esto es consecuencia de las campañas de vacunas que se realizaron en plena pandemia mundial lo que bajó los índices de casos activos y de muertes por covid.

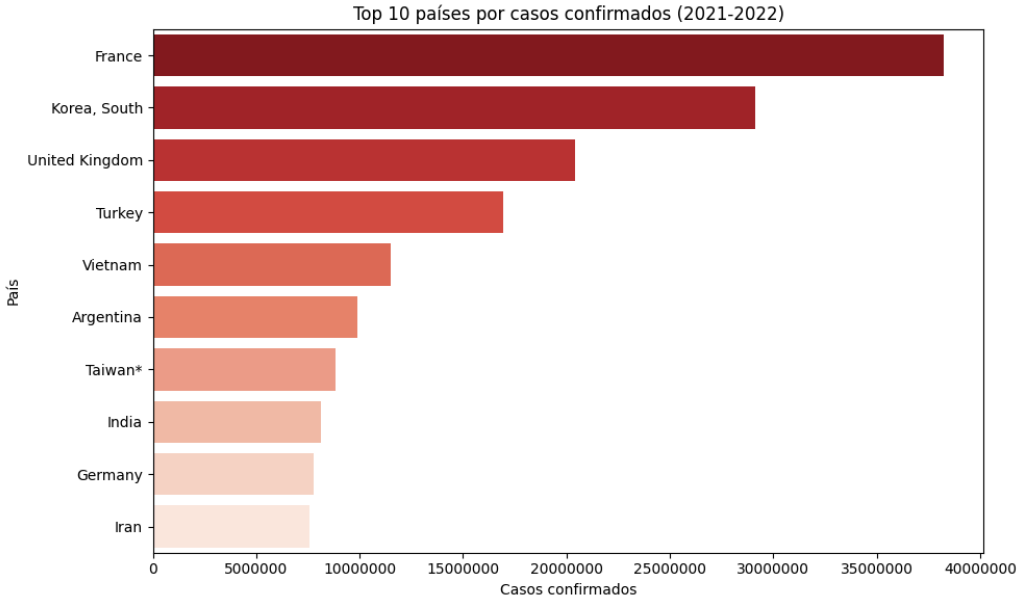

Figura 14

Este gráfico representa el top10 de países con mayores casos confirmados pero correspondiente a los años 2021 y 2022. Por lo que en base a estos datos, Francia tiene una diferencia abismal con el resto de países, lo que lo hace el país con mayor casos confirmados en esos años.

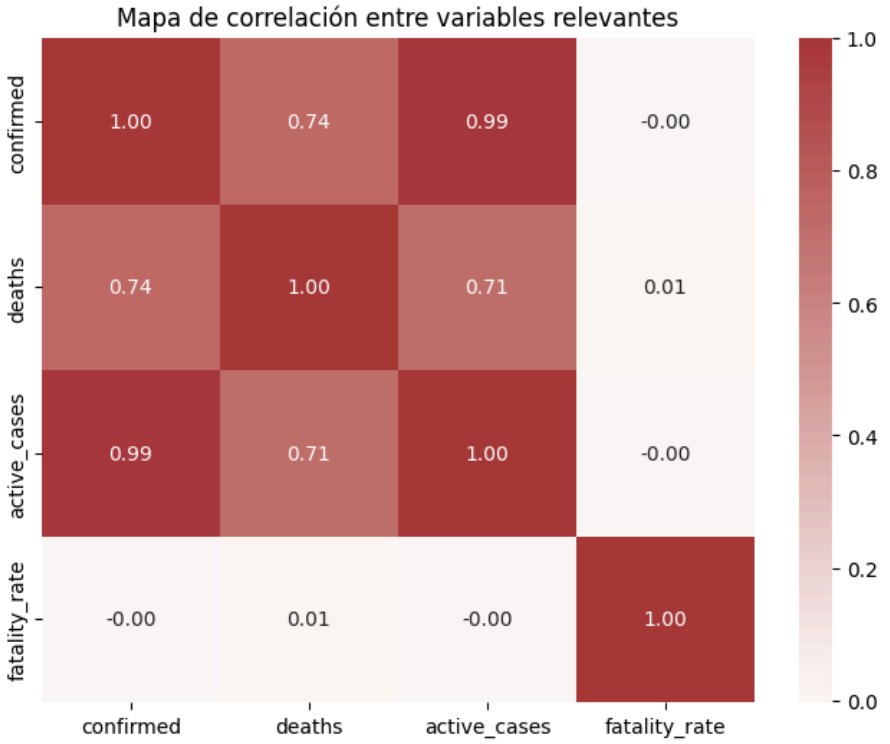

Figura 15

Entre las columnas “confirmed”, “active_cases” y “deaths” podemos ver una gran correlación, siendo solo la más baja la de “active_cases” y “deaths” con un 0.71 en ambas. Y la más alta entre “confirmed” y “active_cases” con un 0.99, lo que significa que a mayor caso confirmado, mayor es la posibilidad de que sea un caso activo, al menos en los tiempos en donde se evaluó el dataset(2021-2022).

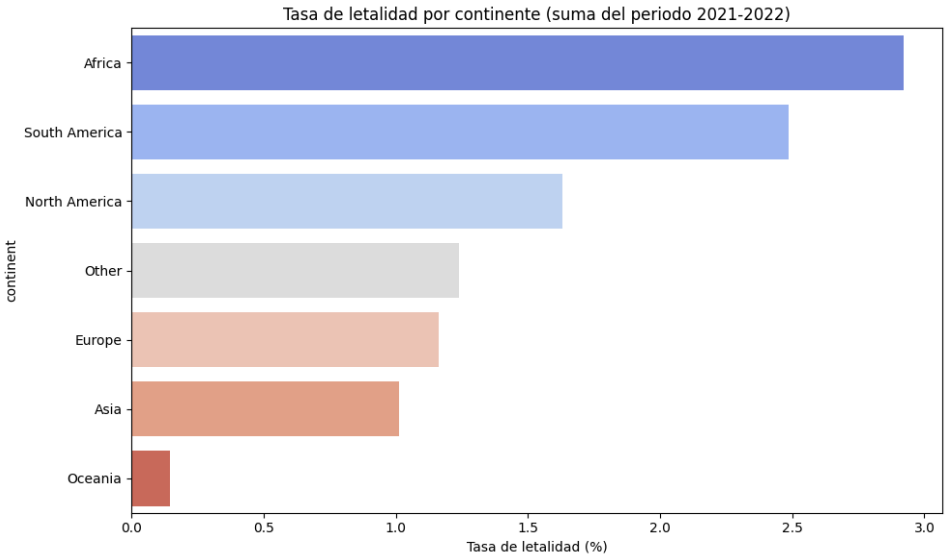

Figura 16

Entre los años señalados vemos a la mayoría de continentes más desarrollados con una tasa de letalidad menor a la que muestran continentes vulnerables como África o Sudamérica, quizás porque supieron tomar mejores medidas ante los posibles

rebrote, ya que fueron los primeros países en tener el contagio por covid-19. Oceanía por su parte tuvo una muy baja tasa, ya sea por su ubicación tan alejada del resto.

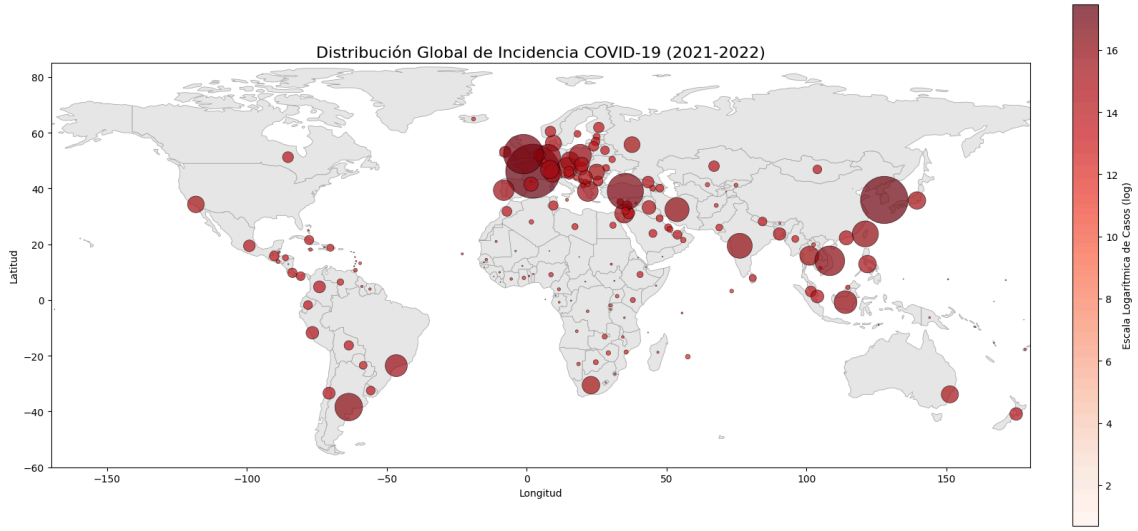

Figura 17

Mapa global encargado de mostrar los casos confirmados a nivel mundial, evidenciando las áreas de mayor contagio y como el covid-19 llegó a afectar a un montón de países, sobre todo al continente Europeo gracias a la facilidad de desplazamiento entre los países europeos. Esto le generó varios casos de contagio incluso en los años donde la humanidad estuvo volviendo a la normalidad, por lo que tuvieron muchos índices de rebrote.

Evidencias de Optimización

1. Lectura eficiente

Antes de optimizar:

```
Tiempo: 155.18858456611633 segundos
Memoria: 1208.714566230774 MB
```

Después de optimizar:

```
Tiempo OPTIMIZADO: 92.98890709877014 segundos
Memoria OPTIMIZADA: 1208.714566230774 MB
```

2. Conversión de tipos

Antes de optimizar:

```
Memoria: 1033.47096824646 MB
```

Después de optimizar:

```
Memoria OPTIMIZADA: 666.5488595962524 MB
Reducción de memoria: 366.9221086502075 MB
```

```
Porcentaje de optimizacion 35.504
```

3. Uso de índices y operaciones vectorizadas.

Antes de optimizar:

```
Tiempo: 30.73193073272705 segundos
```

Después de optimizar:

```
Tiempo OPTIMIZADO 0.05966472625732422 segundos
```

Conclusiones

La trata de datasets tan grandes es un proceso completamente delicado en donde se necesita tener mucho cuidado ante la limpieza de datos, siendo esta la parte más importante del análisis de datos. El estudio de Johns Hopkins que reúne todos los datos obtenidos de todos los países por casos de covid-19, esto nos permite analizar el comportamiento del covid-19 y como los países tomaron medidas ante el incremento desmesurado de contagios a inicios de este.

Viendo los resultados anteriormente mencionados y documentados, ante diferentes gráficos, podemos ver la directa correlación entre los casos confirmados y fallecidos a inicios de los reportes diarios de contagio(más específicamente en la época de marzo de 2020). Sin embargo a medida que el tiempo fue avanzando, la tasa de letalidad fue bajando drásticamente, ya sea por el accionar de los diferentes gobiernos de cada país, como del confinamiento mundial que sufrió la humanidad en aquellos tiempos donde el covid-19 tenía su apogeo.

En términos de análisis con librerías de python para la ciencia de datos, podemos afirmar que el procesamiento de datos grandes es imposible sin técnicas de optimización, ya que los tiempos al procesar estos toman más del esperado ante la gran cantidad de datos, ya que la cantidad de reportes diarios generados para el estudio son demasiados.

Aprendizajes:

- Procesar grandes cantidades de datos
- Realizar distintos tipos de gráficos.
- Manejar diferentes técnicas de optimización.
- Obtener conclusiones ante el análisis de datos.