

Integración de bases de datos No Relacionales para el análisis de mortalidad en EE.UU.



Docente

Mg. Felipe Gaston Vásquez Morales

Alumno

Maximiliano Alejandro Sepulveda Rubilar

Benjamín Alonso Fernández Andrade

Módulo

ICC529-1

Github

(todos los entregables)

<https://github.com/Mizuhar4/Proyecto-Final-TBD.git>

Temuco, Chile.

Julio, 2025

1 Descripción del proyecto.....	3
2 Metodología.....	4
Diseño de la base de datos:.....	4
Descripción procesos ETL:.....	6
Scripts utilizados:.....	7
clickhouse_setup.py.....	7
neo4j_setup.py.....	8
Consulta 1:.....	9
Causas de muerte en Clickhouse:.....	9
Causas de muerte en Neo4j:.....	9
Consulta 2:.....	10
Clickhouse:.....	10
Neo4j:.....	10
Descripción de la integración entre ambos:.....	11
integration.py.....	11
3 Resultado y análisis detallado.....	15
Análisis 1:.....	15
Análisis 2:.....	16
Análisis 3:.....	17
Análisis 4:.....	18
Análisis 5:.....	19
Análisis General:.....	20
4 Conclusión.....	21
5 Bibliografía.....	22

1 Descripción del proyecto

En el presente proyecto, nos dedicamos a combinar datos de dos bases de datos no relacionales ClickHouse y Neo4j con el objetivo principal de almacenar, consultar y examinar información sobre las principales causas de muerte en Estados Unidos, con el propósito de comprender mejor las tendencias de mortalidad a lo largo de diferentes períodos y regiones.

Para este proyecto elegimos dos conjuntos de datos públicos el primero es `Death_Rates1900-2013.csv`, incluye datos históricos de tasas de mortalidad desde 1900 hasta 2013, organizados por causa de muerte y año, con valores que reflejan el promedio de muertes por cada 100.000 habitantes. El segundo es `leading_cause_death.csv` el cual cubre registros desde 1999 hasta 2013, ofreciendo detalles por estado, causa de muerte y número de fallecimientos, estos datos permiten analizar tendencias a largo plazo y explorar los patrones geográficos con mayor precisión. [1]

Como motores de base de datos nos guiamos por ClickHouse para el primer conjunto por su diseño donde destaca en manejar grandes volúmenes de datos y ofrece un rendimiento excepcional frente a otros en consultas analíticas sobre nuestras series temporales históricas, esto resulta ideal para analizar más de un siglo de tasas de mortalidad y ser capaz de detectar años con cambios significativos en ciertas enfermedades.

Por otro lado como segundo motor elegimos Neo4j para el segundo conjunto debido a que su modelo de grafos simplifica la representación de relaciones complejas entre entidades como estados, causas de muerte y años, esto nos facilita visualizar cómo ciertas enfermedades impactan regiones específicas en diferentes períodos y analizar la relación entre factores geográficos y tendencias de mortalidad, además Neo4j permite consultas naturalmente orientadas a relaciones, algo más difícil de lograr con bases de datos tabulares tradicionales.

Al integrar ambos motores de base de datos mediante un proceso de extracción, transformación y carga (ETL), nos permite generar indicadores de valor como el promedio de muertes por causa a nivel nacional y de forma más específica el número de fallecimientos por estado en un año determinado. Asimismo posibilita la exploración de relaciones temporales y geográficas a través de consultas conjuntas, aportando una herramienta potente para el análisis de datos históricos de salud pública en Estados Unidos.

Diseño de la base de datos:

Para cumplir con los requisitos del proyecto, se diseñaron esquemas básicos y optimizados para cada uno de los motores de bases de datos seleccionados, en ClickHouse se creó una tabla denominada `death_rates`, estructurada para almacenar información de tasas de mortalidad. Esta tabla contiene las columnas: `year` (año del registro), `cause` (causa de muerte), `death_rate` (tasa de mortalidad por cada 100.000 habitantes).

Como diseño de la base de datos en el caso de ClickHouse creamos la tabla `death_rates` donde su estructura, es la siguiente:

Tabla N°1

name	type
year	UInt16
cause	String
death_rate	Float32

En el caso del `year` decidimos usar el tipo `UInt16` ya que al ser de 16 bits se pueden almacenar valores desde 0 hasta 65,535 que nos da suficiente para almacenar años como 1900-2025 o incluso más allá de ese rango, así logrando ahorrar espacio en disco y memoria.

En el caso de Neo4j, se utilizó un modelo de grafos para representar las relaciones entre las entidades donde definimos los siguientes nodos:

- State (representando a los estados de EE.UU.)
- Cause (causas de muerte)
- Year (años correspondientes a los registros).

Las relaciones fueron modeladas de la siguiente forma: un estado se conecta con una causa de muerte mediante la relación `[REPORTED]`, la cual incluye la propiedad `deaths` (cantidad de fallecimientos registrados) y cada causa se vincula con un año mediante la relación `[OCCURRED_IN]`. Esta estructura permite navegar de manera intuitiva por los datos y analizar las relaciones geográficas y temporales de forma eficiente.

A continuación en la Figura N°1 se ilustra como se ve la estructura de la base de datos donde elegimos reducir el conjunto de datos a un único estado (Florida) y a un año específico, manteniendo todas las causas de muerte asociadas. Esta representación permite observar la densidad de conexiones entre el nodo Florida y las diversas causas de muerte.

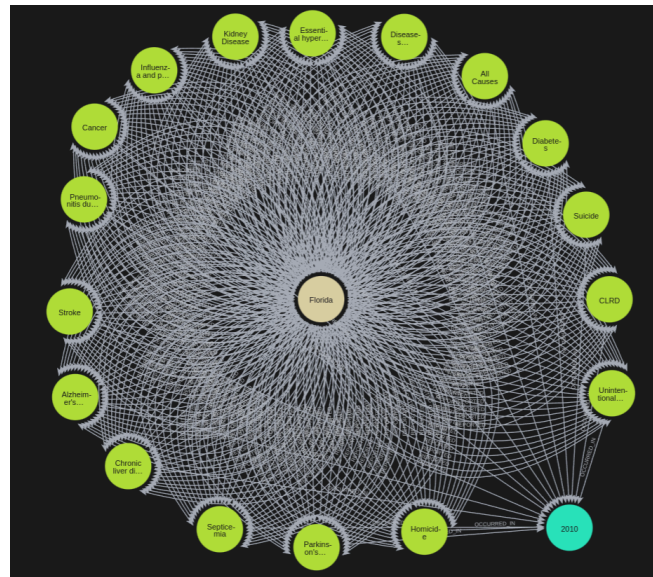


Figura N°1

A continuación en la Figura N°2 ocultamos todas la mayoría causas de muerte para que se aprecie una versión simplificada de la estructura, de esta forma se aprecia claramente como el nodo florida se conecta con el nodo cáncer a través de la relación REPORTED y como este último se vincula con el nodo 2010 mediante la relación OCCURRED_IN reflejando una relación temporal y causal bien definida.

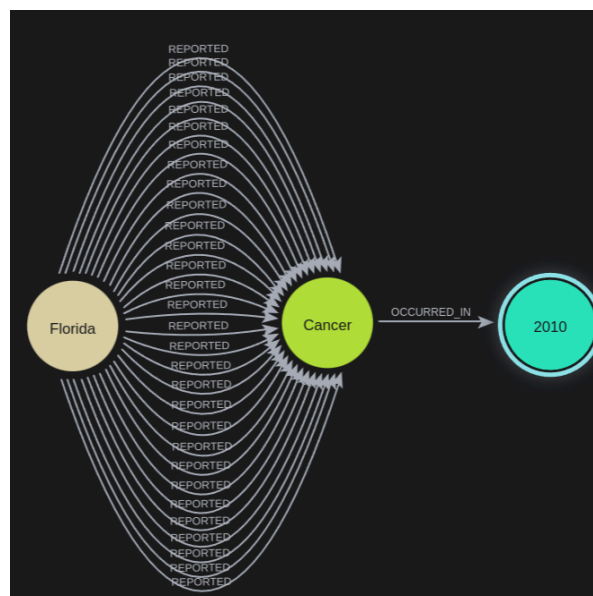


Figura N°2

Descripción procesos ETL:

1. Extracción:

La etapa de extracción consistió en obtener los datos brutos desde dos archivos CSV. El primero, `Death_Rates1900-2013.csv`, contiene tasas de mortalidad ajustadas por edad, asociadas a diversas causas de muerte en Estados Unidos, registradas entre los años 1900 y 2013. Esta fuente fue utilizada para alimentar la base de datos ClickHouse. El segundo archivo, `leading_cause_death.csv`, presenta el número absoluto de muertes por causas principales, desglosado por estado y por año. Estos datos fueron empleados para construir el grafo de relaciones en Neo4j. Ambos archivos fueron leídos utilizando Python, con apoyo de la biblioteca pandas, lo que permitió una lectura eficiente y flexible para su posterior procesamiento.

2. Transformación:

En esta etapa se aplicaron múltiples procesos de limpieza y normalización para asegurar la calidad y consistencia de los datos antes de ser cargados en sus respectivas bases. En primer lugar, se filtraron las columnas relevantes: para el archivo `leading_cause_death.csv` se conservaron solamente las columnas `STATE`, `CAUSE_NAME`, `YEAR` y `DEATHS`. En paralelo, se estandarizaron los nombres de causas en ambos archivos para facilitar su posterior correspondencia. Además, se verificó la validez de los datos, convirtiendo los campos `DEATHS` y `YEAR` a tipos numéricos y eliminando registros incompletos o mal formateados.

Como parte de la transformación semántica, se definió un diccionario de mapeo (`CAUSE_MAP`) que asocia explícitamente los nombres de causas en ambos datasets, permitiendo una correcta integración de la información. Además, se realizaron transformaciones específicas: en ClickHouse se calculó el promedio nacional de las tasas de mortalidad por causa y año, mientras que en Neo4j se agruparon los registros por estado y causa para obtener la suma total de muertes por año. También se eliminaron duplicados y registros irrelevantes, como aquellos etiquetados como "United States", que no representaban un estado individual.

3. Carga:

En la etapa de carga, los datos fueron insertados en sus respectivas bases utilizando scripts automatizados en Python. Para ClickHouse, se cargaron los datos limpios en una tabla llamada `death_rates` utilizando el conector `clickhouse_connect`. Esta tabla contiene las columnas `year`, `cause`, `death_rate` y una columna adicional `average_age` con valores simulados, que ilustra un caso de transformación adicional. Por otro lado, en Neo4j se construyó un modelo de grafo mediante sentencias Cypher ejecutadas a través de `neo4j-python-driver`. En este modelo se definieron nodos para `State`, `Cause` y `Year`, con relaciones del tipo `(State)-[:REPORTED {deaths, year}]->(Cause)` y `(Cause)-[:OCCURRED_IN]->(Year)`, permitiendo un análisis relacional detallado y visualizable.

Scripts utilizados:

“Durante el proceso de carga y estructuración de datos se implementaron dos scripts principales, uno para cada motor de base de datos no relacional: clickhouse_setup.py para ClickHouse y neo4j_setup.py para Neo4j.”

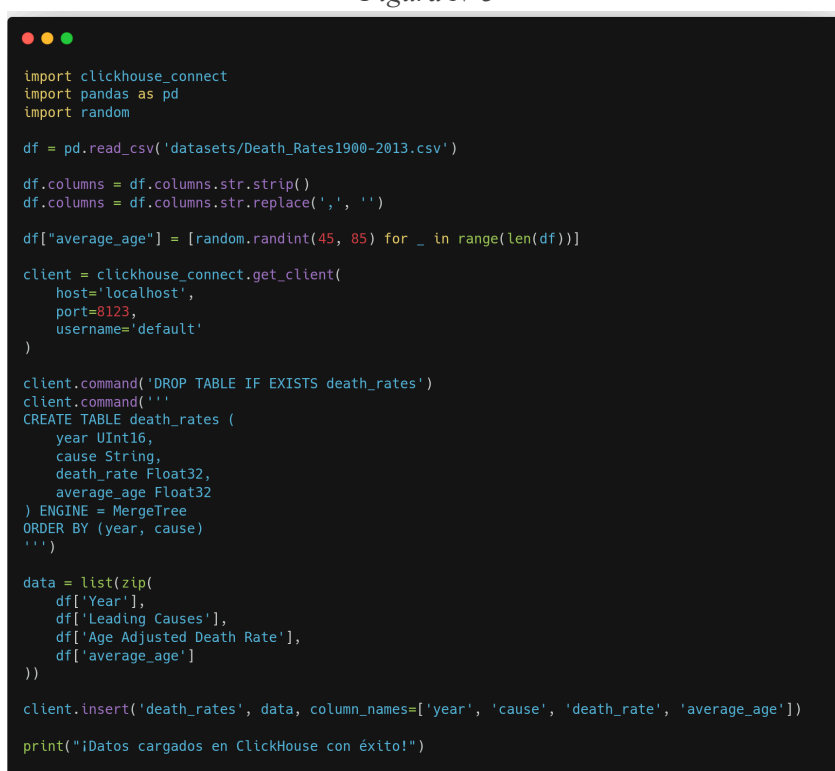
clickhouse_setup.py

Este script tiene como objetivo **cargar los datos del archivo Death_Rates1900-2013.csv** en una tabla de ClickHouse. Su funcionalidad se divide en los siguientes pasos:

- **Lectura y limpieza del dataset:**
 - Se utiliza pandas para cargar el archivo CSV.
 - Se limpian los nombres de columnas para eliminar espacios y comas que puedan interferir en el procesamiento.
- **Simulación de datos adicionales:**
 - Se agrega una columna ficticia average_age con valores aleatorios entre 45 y 85 años, con el fin de ilustrar un ejemplo de transformación de datos.
- **Conexión a ClickHouse:**
 - Se establece la conexión utilizando clickhouse_connect, sin requerir autenticación por contraseña.
- **Creación de la tabla:**
 - Se define la tabla death_rates, que incluye los campos: year, cause, death_rate y average_age.
 - Se usa el motor MergeTree y se ordena por (year, cause).
- **Inserción de los datos:**
 - Los registros procesados se insertan directamente en la tabla mediante client.insert.

“A continuación en la Figura N°3 se muestra el script de python”

Figura N°3



```
import clickhouse_connect
import pandas as pd
import random

df = pd.read_csv('datasets/Death_Rates1900-2013.csv')

df.columns = df.columns.str.strip()
df.columns = df.columns.str.replace(',', '')

df["average_age"] = [random.randint(45, 85) for _ in range(len(df))]

client = clickhouse_connect.get_client(
    host='localhost',
    port=8123,
    username='default'
)

client.command('DROP TABLE IF EXISTS death_rates')
client.command('''
CREATE TABLE death_rates (
    year UInt16,
    cause String,
    death_rate Float32,
    average_age Float32
) ENGINE = MergeTree
ORDER BY (year, cause)
''')

data = list(zip(
    df['Year'],
    df['Leading Causes'],
    df['Age Adjusted Death Rate'],
    df['average_age']
))

client.insert('death_rates', data, column_names=['year', 'cause', 'death_rate', 'average_age'])

print("¡Datos cargados en ClickHouse con éxito!")
```

neo4j_setup.py

Este script tiene como objetivo **cargar el archivo leading_cause_death.csv** en una base de datos Neo4j, modelando los datos como un grafo semántico. Su funcionalidad principal es la siguiente:

- **Lectura y preprocesamiento del dataset:**
 - Se extraen solo las columnas relevantes: STATE, CAUSE_NAME, YEAR, y DEATHS.
 - Se eliminan los registros nulos y se convierte el campo DEATHS a tipo entero.
- **Conexión a Neo4j:**
 - Se establece una conexión mediante neo4j.GraphDatabase.driver utilizando credenciales básicas.
- **Inserción de datos en formato de grafo:**
 - Para cada fila, se ejecuta una transacción que crea o actualiza los siguientes elementos:
 - Nodo (State {name})
 - Nodo (Cause {name})
 - Nodo (Year {value})
 - Relación (:State)-[:REPORTED {year, deaths}]->(:Cause)
 - Relación (:Cause)-[:OCCURRED_IN]->(:Year)
- **Carga iterativa:**
 - Se itera sobre cada fila del DataFrame y se ejecuta la transacción Cypher para construir el grafo completo.

“A continuación en la Figura N°4 se muestra el script de python”

Figura N°4

```
from neo4j import GraphDatabase
import pandas as pd

uri = "neo4j://127.0.0.1:7687"
user = "neo4j"
password = "pascual1"

df = pd.read_csv("datasets/leading_cause_death.csv")
df = df[['STATE', 'CAUSE_NAME', 'YEAR', 'DEATHS']]
df = df.dropna()
df['DEATHS'] = df['DEATHS'].apply(lambda x: str(x).isdigit())
df['DEATHS'] = df['DEATHS'].astype(int)
df['YEAR'] = df['YEAR'].astype(int)

driver = GraphDatabase.driver(uri, auth=(user, password))

def insert_data(tx, state, cause, year, deaths):
    tx.run("""
        MERGE (s:State {name: $state})
        MERGE (c:Cause {name: $cause})
        MERGE (y:Year {value: $year})
        MERGE (s)-[r:REPORTED {year: $year}]->(c)
        SET r.deaths = $deaths
        MERGE (c)-[:OCCURRED_IN]->(y)
    """, state=state, cause=cause, year=year, deaths=deaths)

with driver.session() as session:
    for _, row in df.iterrows():
        session.execute_write(insert_data, row['STATE'], row['CAUSE_NAME'], row['YEAR'],
                               row['DEATHS'])

print("¡Datos cargados en Neo4j con éxito!")
driver.close()
```


“Primero para que las consultas de cada base de datos tengan sentido buscamos cosas en común”

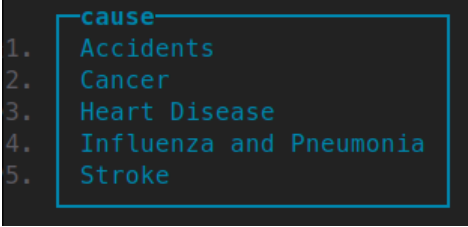
Consulta 1:

“Ver causas de muerte en común de cada Dataset”

Causas de muerte en Clickhouse:

```
SELECT DISTINCT cause FROM death_rates  
ORDER BY cause;
```

```
SELECT DISTINCT cause  
FROM death_rates  
ORDER BY cause ASC
```



```
1. cause  
2. Accidents  
3. Cancer  
4. Heart Disease  
5. Influenza and Pneumonia  
6. Stroke
```

Figura N°5

Causas de muerte en Neo4j:

- Visualizar datos en forma de tabla:

```
MATCH (c:Cause) RETURN DISTINCT c.name ORDER BY c.name
```

- Visualizar datos en forma de grafos:

```
MATCH (c:Cause)  
WITH DISTINCT c  
RETURN c  
ORDER BY c.name
```

Figura N°6



“Permite ver las causas en común para luego integrarlo y hacer análisis más relevantes”

Consulta 2:

“Mayor causas de muertes desde el año 1999”

Clickhouse:

```
SELECT
  year,
  cause,
  AVG(death_rate) AS avg_death_rate
FROM death_rates
WHERE year >= 1999
GROUP BY
  year,
  cause
ORDER BY
  year ASC,
  avg_death_rate DESC
```

	year	cause	avg_death_rate
1.	1999	Heart Disease	266.5
2.	1999	Cancer	200.8000030517578
3.	1999	Stroke	61.599998474121094
4.	1999	Accidents	35.29999923706055
5.	1999	Influenza and Pneumonia	23.5
6.	2000	Heart Disease	257.6000061035156
7.	2000	Cancer	199.60000610351562
8.	2000	Stroke	60.900001525878906
9.	2000	Accidents	34.900001525878906
0.	2000	Influenza and Pneumonia	23.700000762939453
1.	2001	Heart Disease	249.5
2.	2001	Cancer	196.5
3.	2001	Stroke	58.400001525878906
4.	2001	Accidents	35.70000076293945
5.	2001	Influenza and Pneumonia	22.200000762939453
6.	2002	Heart Disease	244.60000610351562
7.	2002	Cancer	194.3000030517578
8.	2002	Stroke	57.20000076293945
9.	2002	Accidents	37.099998474121094

Figura N°7

Neo4j:

- Visualizar datos en forma de tabla:

```
MATCH (:State)-[r:REPORTED]->(c:Cause)-[:OCCURRED_IN]->(y:Year)
WHERE y.value >= 1999 AND r.deaths IS NOT NULL
WITH y.value AS year, c.name AS cause, SUM(r.deaths) AS total_deaths
RETURN year, cause, total_deaths
ORDER BY year ASC, total_deaths DESC
```

- Visualizar datos en forma de grafos:

```
MATCH (:State)-[r:REPORTED]->(c:Cause)-[:OCCURRED_IN]->(y:Year)
WHERE y.value >= 1999 AND r.deaths IS NOT NULL
WITH c, y, SUM(r.deaths) AS total_deaths
RETURN c, y, total_deaths
ORDER BY y.value ASC, total_deaths DESC
LIMIT 100
```

Figura N°8

	year	cause	total_deaths
1	1999	"All Causes"	147276265
2	1999	"Diseases of Heart"	38742275
3	1999	"Cancer"	33815957
4	1999	"Stroke"	8639181
5	1999	"CLRD"	7898724
6	1999	"Unintentional Injuries"	6922158

“Asocia los años en los que hay más muertes”

Descripción de la integración entre ambos:

La conexión entre ClickHouse y Neo4j fue implementada con un script en Python (integration.py) que realiza la consulta, el cruce y el análisis de la información proveniente de las dos bases. Tal como se ha mencionado, la finalidad del script es combinar la información estructurada (tasas de mortalidad promedio, definidos por causa y año) almacenada en ClickHouse con la información relacional (muertes por causa, estado y año) almacenada en Neo4j, lo que permite tener una visión más integrada entre las distintas enfermedades y su impacto en Estados Unidos.

El script hace conexiones al mismo tiempo con ClickHouse y con Neo4j mediante los drivers que ofrecen. Después establece diferentes funciones para obtener causas de muerte comunes en ambas bases, consultar qué años hay para cada causa. También extrae las tasas de mortalidad promedio definidas con causa y año a partir de la información guardada en ClickHouse y extrae las cifras de muertes por estado a partir de la información guardada en Neo4j y crea un encuentro de los resultados para evitar las duplicaciones.

El programa tiene un menú interactivo en el que va guiando al usuario en la selección de una causa y un año en concreto para realizar el análisis. Una vez seleccionados, se muestran estadísticas de las tasas de mortalidad, las tasas nacionales promedio y una lista de los estados en orden decreciente desde aquellas causas de muertes.

Como funcionalidad adicional, el menú incluye una opción para generar un resumen combinado con todas las causas y años disponibles, integrando promedios nacionales desde ClickHouse y totales de muertes desde Neo4j. Este resumen se guarda también como archivo CSV y sirve como insumo para análisis estadísticos más amplios.

integration.py

```
import clickhouse_connect
from neo4j import GraphDatabase
import pandas as pd

clickhouse = clickhouse_connect.get_client(host='localhost', port=8123)

neo4j_uri = "neo4j://127.0.0.1:7687"
neo4j_user = "neo4j"
neo4j_password = "pascual1"
neo4j_driver = GraphDatabase.driver(neo4j_uri, auth=(neo4j_user, neo4j_password))

CAUSE_MAP = {
    "Stroke": "Stroke",
    "Cancer": "Cancer",
    "Influenza and pneumonia": "Influenza and Pneumonia",
    "Diseases of Heart": "Heart Disease",
    "Unintentional Injuries": "Accidents"
}

def get_common_causes():
    query = "SELECT DISTINCT cause FROM death_rates"
    clickhouse_causes = set(row[0] for row in clickhouse.query(query).result_rows)

    with neo4j_driver.session() as session:
        result = session.run("MATCH (c:Cause) RETURN DISTINCT c.name AS cause")
```

```

neo4j_causes = set(record["cause"] for record in result)

neo4j_causes_mapped = [c for c in CAUSE_MAP if CAUSE_MAP[c] in clickhouse_causes and c in neo4j_causes]
return sorted(neo4j_causes_mapped)

def get_common_years(cause_neo4j, cause_clickhouse):
    query = f"SELECT DISTINCT year FROM death_rates WHERE cause = '{cause_clickhouse}' ORDER BY year"
    clickhouse_years = set(row[0] for row in clickhouse.query(query).result_rows)

    with neo4j_driver.session() as session:
        result = session.run(
            """
            MATCH (s:State)-[r:REPORTED]->(c:Cause {name: $cause})
            RETURN DISTINCT r.year AS year
            ORDER BY year
            """, cause=cause_neo4j)
        neo4j_years = set(record["year"] for record in result if record["year"] is not None)

    return sorted(clickhouse_years.intersection(neo4j_years))

def get_national_death_rate(cause_clickhouse, year):
    query = f"""
    SELECT AVG(death_rate) AS avg_rate
    FROM death_rates
    WHERE cause = '{cause_clickhouse}' AND year = {year}
    """
    result = clickhouse.query(query)
    if result.result_rows:
        return result.result_rows[0][0]
    return None

def get_state_deaths_from_neo4j(cause_neo4j, year):
    with neo4j_driver.session() as session:
        query = """
        MATCH (s:State)-[r:REPORTED]->(c:Cause {name: $cause})
        WHERE r.year = $year
        RETURN s.name AS state, r.deaths AS deaths
        ORDER BY deaths DESC
        """
        result = session.run(query, cause=cause_neo4j, year=year)
        return pd.DataFrame([dict(record) for record in result])

def mostrar_menu(lista_opciones, titulo):
    print(f"\nSeleccione {titulo}:")
    for i, item in enumerate(lista_opciones, 1):
        print(f"{i}. {item}")
    print("Ingrese el número correspondiente (o 'q' para volver al menú): ", end="")

def export_combined_summary_to_csv():
    print("\nExtrayendo datos de Neo4j y ClickHouse...")

    causas = get_common_causes()
    rows = []

    for causa_neo in causas:

```

```

causa_ch = CAUSE_MAP[causa_neo]
años = get_common_years(causa_neo, causa_ch)

for año in años:
    with neo4j_driver.session() as session:
        result = session.run("""
            MATCH (s:State)-[r:REPORTED]->(c:Cause {name: $causa})
            WHERE s.name <> 'United States' AND r.year = $año
            RETURN r.deaths AS deaths
            """, causa=causa_neo, año=año)
        muertes = [record["deaths"] for record in result if record["deaths"] is not None]
        total_neo4j = sum(muertes) if muertes else 0
        estados = len(muertes)

    query = f"""
        SELECT AVG(death_rate) AS avg_rate
        FROM death_rates
        WHERE cause = '{causa_ch}' AND year = {año}
        """
    result = clickhouse.query(query)
    avg_ch = result.result_rows[0][0] if result.result_rows else None

    rows.append({
        "causa": causa_neo,
        "año": año,
        "muertes_totales_neo4j": total_neo4j,
        "tasa_promedio_nacional_clickhouse": round(avg_ch, 2) if avg_ch else None,
        "estados_reportados": estados
    })

df = pd.DataFrame(rows)
df = df.sort_values(by=["causa", "año"])
df.to_csv("resumen_integrado_causas.csv", index=False)
print("CSV generado: 'resumen_integrado_causas.csv'")

def main():
    while True:
        print("\n=== MENÚ PRINCIPAL ===")
        print("1. Analizar causa y año")
        print("2. Exportar resumen combinado (Neo4j + ClickHouse)")
        print("3. Salir")
        opcion = input("Seleccione una opción: ").strip()

        if opcion == "1":
            causas = get_common_causes()
            if not causas:
                print("No hay causas comunes.")
                continue

            mostrar_menu(causas, "la causa para analizar")
            op = input().strip()
            if op.lower() == 'q':
                continue
            if not op.isdigit() or not (1 <= int(op) <= len(causas)):
                print("Opción inválida.")
                continue
            causa_neo = causas[int(op) - 1]

```

```

causa_ch = CAUSE_MAP[causa_neo]

años = get_common_years(causa_neo, causa_ch)
if not años:
    print("No hay años comunes.")
    continue

mostrar_menu(años, "el año para analizar")
op_año = input().strip()
if op_año.lower() == 'q':
    continue
if not op_año.isdigit() or not (1 <= int(op_año) <= len(años)):
    print("Año inválido.")
    continue
año = años[int(op_año) - 1]

print(f"\nAnálisis de: {causa_neo} ({causa_ch}) en el año {año}")

national_rate = get_national_death_rate(causa_ch, año)
if national_rate is not None:
    print(f"\nTasa nacional promedio: {national_rate:.2f} muertes por 100,000 habitantes\n")
else:
    print("No se encontró la causa en ClickHouse.")
    continue

df = get_state_deaths_from_neo4j(causa_neo, año)
df = df[df['state'] != 'United States']
df = df.groupby('state', as_index=False)['deaths'].sum()
df = df.sort_values(by='deaths', ascending=False)

if not df.empty:
    print("Top muertes por estado:\n")
    print(df.head(51))
    nombre_csv = f"salida_{causa_neo.replace(' ', '_')}_año.csv"
    df.to_csv(nombre_csv, index=False)
    print(f"\nResultados guardados en '{nombre_csv}'")
else:
    print("No se encontraron datos en Neo4j.")

elif opcion == "2":
    export_combined_summary_to_csv()

elif opcion == "3":
    print("Saliendo...")
    break
else:
    print("Opción inválida.")

if __name__ == "__main__":
    main()

```

3 Resultado y análisis detallado

Análisis 1:

“Top muertes por año usando ambas bases de datos con integration.py”

Figura N°9

```
> python integration.py

=== MENÚ PRINCIPAL ===
1. Analizar causa y año
2. Exportar resumen combinado (Neo4j + ClickHouse)
3. Salir
Seleccione una opción: 1

Seleccione la causa para analizar:
1. Cancer
2. Diseases of Heart
3. Influenza and pneumonia
4. Stroke
5. Unintentional Injuries
Ingrese el número correspondiente (o 'q' para volver al menú): 1

Seleccione el año para analizar:
1. 1999
2. 2000
3. 2001
4. 2002
5. 2003
6. 2004
7. 2005
8. 2006
9. 2007
10. 2008
11. 2009
12. 2010
13. 2011
14. 2012
15. 2013
Ingrese el número correspondiente (o 'q' para volver al menú): 2

Análisis de: Cancer (Cancer) en el año 2000

Tasa nacional promedio: 199.60 muertes por 100,000 habitantes

Top muertes por estado:

state deaths
4      California 53158
9      Florida    39183
32     New York   37198
43     Texas      33300
38     Pennsylvania 30161
13     Illinois   25365
35     Ohio       24988
22     Michigan   19798
30     New Jersey 18073
33     North Carolina 15786
21     Massachusetts 14027
10     Georgia    13690
46     Virginia   13528
14     Indiana    12842
42     Tennessee  12339
25     Missouri   12144
47     Washington 10668

Resultados guardados en 'salida_Cancer_2000.csv'
```

Se comprueba que el promedio de muertes por habitantes de Clickhouse, su aproximado es correcto según las muertes registradas de la base de datos de Neo4J. Esto permite ver las regiones más afectadas por ciertas enfermedades o causas y en qué año pasa esto.

Análisis 2:

“Analizar el csv generado de la integración completa de las dos bases de datos”

Figura N°10

```
=== MENÚ PRINCIPAL ===  
1. Analizar causa y año  
2. Exportar resumen combinado (Neo4j + ClickHouse)  
3. Salir  
Seleccione una opción: 2  
  
Extrayendo datos de Neo4j y ClickHouse...  
CSV generado: 'resumen_integrado_causas.csv'
```

Tabla N°2

causa	año	muerter_totales_neo4j	tasa_promedio_nacional_clickhouse	estados_reportados
Cancer	1999	549838	200.8	51
Diseases of Heart	1999	725192	266.5	51
Influenza and pneumonia	1999	63730	23.5	51
Stroke	1999	167366	61.6	51
Unintentional Injuries	1999	97860	35.3	51

Analizando la Tabla N°2, las enfermedades cuya fuente es cardiaca ocuparon en el año 1999, el primer lugar de todas las causas de muerte en EE. UU., ya que se registraron más de 725.000 personas fallecidas lo que hace que la tasa media de mortalidad nacional fuese de 266,5 muertes por cada 100.000 habitantes. En un segundo lugar está el cáncer, con aproximadamente 550.000 muertes y una tasa de 200,8, lo que da cuenta de que, si bien la guerra fue un gran impacto, el hecho de que estas enfermedades de origen cardiovascular se situarán en el primer lugar significa que su peso se convierte en el máximo impacto con respecto a la salud pública en ese año.

El accidente, Unintentional Injuries, representaron cerca de 97.800 muertes, y el accidente cerebrovascular, Stroke, con 167.000 muertes y la tasa de 61,6. Y, por último, la influenza y la neumonía alcanzaron unas 63.700 muertes, representando la causa con menor mortalidad de entre las causas analizadas, con una tasa de sólo 23,5.

Para cada una de las causas se informaron datos de los 51 estados, lo que hace que se pueda ofrecer datos de cobertura nacional y, por tanto, resultados válidos. Por otro lado, la información extraída puede informarnos de manera muy clara sobre la distribución de las causas de muerte más comunes durante el periodo y cómo se puede utilizar como punto de referencia para poder analizar factores en años posteriores.

Análisis 3:

“Analizar el csv generado de la integración completa de las dos bases de datos causa: cáncer”

Tabla N°3

causa	año	muerter_totales_neo4j	tasa_promedio_nacional_clickhouse	estados_reportados
Cancer	1999	549838	200.8	51
Cancer	2000	553091	199.6	51
Cancer	2001	553768	196.5	51
Cancer	2002	557271	194.3	51
Cancer	2003	556902	190.9	51
Cancer	2004	553888	186.8	51
Cancer	2005	559312	185.1	51
Cancer	2006	559888	181.8	51
Cancer	2007	562875	179.3	51
Cancer	2008	565469	176.4	51
Cancer	2009	567628	173.5	51
Cancer	2010	574743	172.8	51
Cancer	2011	576691	169.0	51
Cancer	2012	582623	166.5	51
Cancer	2013	584881	163.2	51

Muerter totales por cancer durante los años

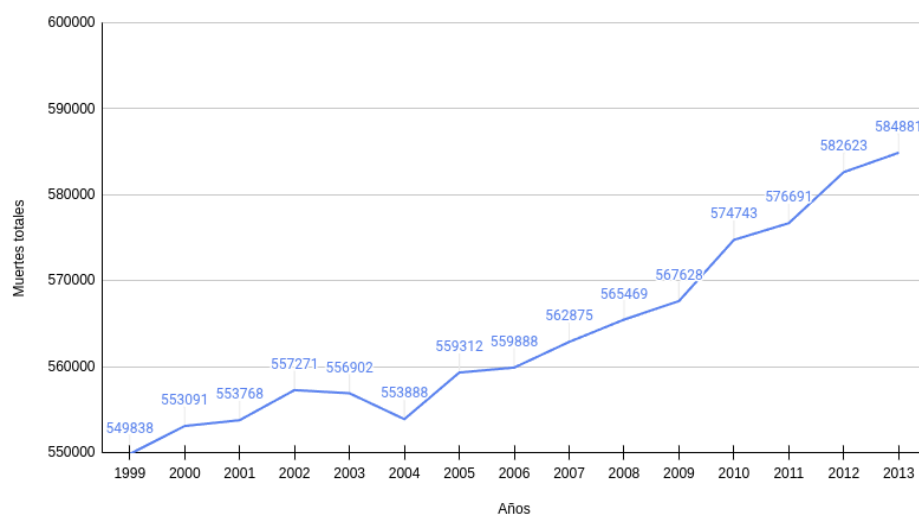


Gráfico N°1

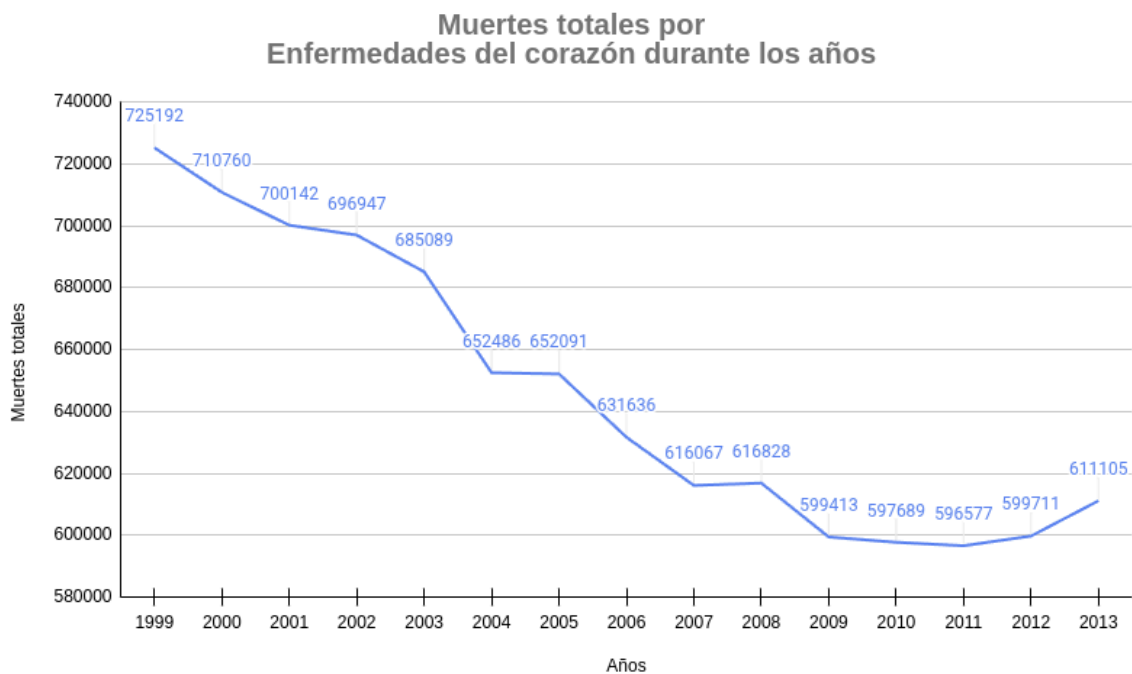
Estudiando lo presentado en la Tabla N°2 y Gráfico N°1 De 1999 a 2013 el número total de muertes por causa de cáncer en territorio estadounidense creció de 549.838 a 584.881. Un leve crecimiento con la obvia de que podría tener relación con el aumento de la población.

Este moderado aumento no impidió que la tasa de mortalidad nacional media bajase de 200,8 a 163,2 muertes por cada 100.000 habitantes. Algo que tal vez puede sugerir un mejor rendimiento en lo que respecta a la prevención, a la detección precoz y al tratamiento.

Análisis 4:

“Analizar el csv generado de la integración completa de las dos bases de datos causa: enfermedades del corazón”

Gráfico N°2



Al observar el Gráfico N°2 en el año 1999 se produjeron más de 725 mil muertes debido a las enfermedades del corazón en los Estados Unidos. En tanto en 2013, este número bajó a 611.105 muertes, mostrando una tendencia sostenida hacia la baja en cifras absolutas.

La tasa nacional promedio de mortalidad también tendió a la baja de manera relevante de 266.5 en 1999 a 169.8 en el año 2013. Esto constituirá una mejora en la salud cardiovascular, tal vez como consecuencia de cambios en el estilo de vida lo que podría incluir el acceso a distintos tratamientos y las campañas de prevención.

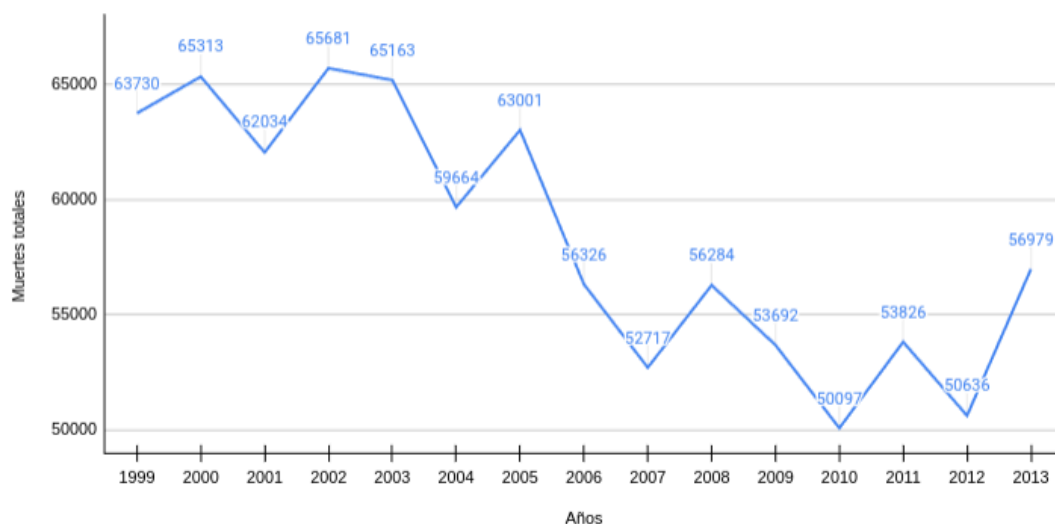
Análisis 5:

“Analizar el csv generado de la integración completa de las dos bases de datos causa cáncer”

Tabla N°4

causa	año	muerter_totales_neo4j	tasa_promedio_nacional_clickhouse	estados_reportados
Influenza and pneumonia	1999	63730	23.5	51
Influenza and pneumonia	2000	65313	23.7	51
Influenza and pneumonia	2001	62034	22.2	51
Influenza and pneumonia	2002	65681	23.2	51
Influenza and pneumonia	2003	65163	22.6	51
Influenza and pneumonia	2004	59664	20.4	51
Influenza and pneumonia	2005	63001	21.0	51
Influenza and pneumonia	2006	56326	18.4	51
Influenza and pneumonia	2007	52717	16.8	51
Influenza and pneumonia	2008	56284	17.6	51
Influenza and pneumonia	2009	53692	16.5	51
Influenza and pneumonia	2010	50097	15.1	51
Influenza and pneumonia	2011	53826	15.7	51
Influenza and pneumonia	2012	50636	14.5	51
Influenza and pneumonia	2013	56979	15.9	51

Gráfico N°2



Viendo la Tabla N°4 y el Gráfico N°3 En el año 1999, las defunciones por gripe y pulmonía alcanzaron las 63.730 Con el pasar de los años el número exhibió ciertos altibajos no muy marcados, tocando un punto bajo en 2010 (50.097 fallecimientos) y un pequeño aumento allá por 2013 (56.979 fallecimientos).

La tasa media a nivel nacional también dibujó una trayectoria en descenso cayendo de 23.5 en 1999 hasta 15.9 en 2013. Esto da cuenta de una mejora global en la forma de prevenir o tratar estas dolencias respiratorias.

Análisis General:

Este proyecto busca entender mejor las razones principales detrás de los fallecimientos en EE. UU., uniendo información de dos fuentes: Neo4j y ClickHouse. Al juntar estos datos, podemos ver cómo cambian las muertes por distintas enfermedades, tanto en todo el país como en cada estado. Esto nos ayuda a hacer análisis más detallados que si usáramos solo una fuente.

Algo clave que vemos con este análisis es cómo cambian las tasas de mortalidad con el tiempo. Por ejemplo, las muertes por cáncer y problemas del corazón han bajado constantemente, probablemente por mejoras en la atención médica, tratamientos mejores y campañas para prevenir. En cambio, otras causas, como los accidentes, suben o se mantienen igual, mostrando que necesitamos hacer algo al respecto.

Neo4j nos da datos sobre cuántas personas mueren en cada estado y año. ClickHouse, por su parte, nos da las tasas de muerte ajustadas por la cantidad de gente que vive en todo el país. Esto nos ayuda a tener una imagen más completa: con Neo4j vemos dónde hay más problemas y con ClickHouse podemos comparar las causas de muerte de forma justa, sin que influya la cantidad de personas que viven en cada lugar.

Al juntar las dos bases, también podemos comprobar si la información es correcta. Por ejemplo, si vemos que hay más muertes en Neo4j, pero la tasa baja en ClickHouse, podría ser porque ha crecido la población o porque los casos se concentran en estados con menos gente. Así, tenemos una visión más real y con más contexto.

En resumen, el proyecto muestra lo útil que es combinar tecnologías NoSQL: Neo4j, que sirve para entender relaciones complicadas, y ClickHouse, que es bueno para analizar grandes cantidades de datos organizados. Con esta forma de trabajar, pudimos sacar, transformar, conectar y mostrar los datos de manera eficiente, lo que nos ayudó a entender mejor cómo son los patrones de mortalidad en el país.

4 Conclusión

En este proyecto, logramos integrar exitosamente las bases de datos no relacionales ClickHouse y Neo4j, permitiendo un análisis conjunto de las principales causas de mortalidad en Estados Unidos mediante datos históricos y geográficos, consideramos que ClickHouse fue una elección acertada para procesar las series temporales de tasas de mortalidad de 1900 a 2013, gracias a su optimización para grandes volúmenes de datos y consultas analíticas de alto rendimiento, superando alternativas como MongoDB o PostgreSQL en este contexto.

Por su parte, Neo4j se destacó al modelar relaciones entre estados, causas de muerte y años, ofreciendo una visualización clara de patrones regionales y temporales, algo menos eficiente en sistemas tabulares y en bases relacionales.

A lo largo del desarrollo nos dimos cuenta la ventaja de optar por mantener datos separados en cada motor para aprovechar sus fortalezas: ClickHouse para análisis rápidos de datos históricos cuantitativos y Neo4j para explorar conexiones complejas, como identificar estados con aumentos simultáneos de muertes por influenza y neumonía, un análisis más intuitivo en grafos que en tabulares con múltiples uniones. El proceso ETL, apoyado en scripts como `clickhouse_setup.py` y `neo4j_setup.py`, facilitó la generación de indicadores clave, como promedios de fallecimientos, enriqueciendo el estudio y análisis de la mortalidad.

Esta integración demuestra que combinar tecnologías especializadas es una estrategia robusta para manejar datos heterogéneos, sin embargo, identificamos limitaciones en la visualización de relaciones en Neo4j, sugiriendo la necesidad de explorar herramientas avanzadas en futuros trabajos y proyecto de un estilo similar. Finalizando este proyecto valida la efectividad de bases de datos no relacionales especializadas en el análisis de datos complejos en este caso particular de la salud pública.

5 Bibliografía

- [1] Kaggle. (n.d.). Leading Causes of Death USA Dataset, Recuperado el 6 de julio de 2025, de https://www.kaggle.com/datasets/kingburrito666/leading-causes-of-death-usa/data?select=leading_cause_death.csv.
- [2] ClickHouse. (n.d.). ClickHouse documentation. Recuperado el 6 de julio de 2025, de <https://clickhouse.com/docs>
- [3] Neo4j. (n.d.). Neo4j documentation. Recuperado el 6 de julio de 2025, de <https://neo4j.com/docs>
- [4] Neo4j. (n.d.). CSV import. Recuperado el 6 de julio de 2025, de <https://neo4j.com/docs/getting-started/data-import/csv-import>
- [5] ClickHouse. (n.d.). CSV and TSV data formats. Recuperado el 6 de julio de 2025, de <https://clickhouse.com/docs/integrations/data-formats/csv-tsv>
- [6] Microsoft. (n.d.). Relational data ETL. Recuperado el 6 de julio de 2025, de <https://learn.microsoft.com/es-es/azure/architecture/data-guide/relational-data/etl>