ANÁLISIS DEL APORTE DE LAS ENERGÍAS RENOVABLES EN LA GENERACIÓN ELÉCTRICA EN COLOMBIA

Talento Tech
Inteligencia Artificial
Nivel Explorador

Alba López

Ulises Linares

Luis Bertel

2025

Índice

Resumen

Este estudio analiza la evolución del aporte de las energías renovables en la generación eléctrica en Colombia entre los años 2010 y 2025, en el contexto de la transición energética y los compromisos nacionales e internacionales de sostenibilidad. Para ello, se utilizó un conjunto de datos con registros mensuales de generación eléctrica por tipo de fuente energética, inicialmente extraído desde la plataforma Kaggle (2010–2022) y posteriormente actualizado mediante técnicas de web scraping hasta el año 2025. La investigación se desarrolló bajo el enfoque del ciclo de vida de proyectos de Machine Learning, e incluyó procesos de limpieza de datos, análisis exploratorio, visualización de tendencias y análisis multidimensional. Las herramientas principales empleadas fueron Python y bibliotecas como Pandas, NumPy, Matplotlib y Seaborn. Los resultados permiten identificar patrones de crecimiento, estacionalidad y participación relativa de las fuentes renovables —como solar, eólica, hidroeléctrica y otras— en la matriz eléctrica colombiana. Este análisis proporciona insumos relevantes para la toma de decisiones en materia de política energética, planificación sectorial y evaluación del avance del país hacia un sistema eléctrico más diversificado y sostenible.

1. Introducción

En el contexto actual de transición energética y lucha contra el cambio climático, el análisis de las fuentes utilizadas para la generación de electricidad se ha convertido en una herramienta fundamental para comprender la evolución del sistema energético global. La electricidad es un insumo esencial para prácticamente todas las actividades productivas, sociales y tecnológicas, por lo que su disponibilidad, accesibilidad y sostenibilidad resultan determinantes para el desarrollo económico y el bienestar social.

Los sistemas eléctricos están experimentando una transformación profunda impulsada por el avance tecnológico, la creciente penetración de energías renovables, las preocupaciones ambientales y los compromisos internacionales de descarbonización. En este escenario, evaluar la participación relativa de las distintas fuentes de generación —como las tecnologías renovables (hidroeléctrica, solar, eólica, geotérmica), la energía nuclear y los combustibles fósiles (carbón, gas natural, petróleo)— permite no solo entender el comportamiento histórico del sector, sino también anticipar escenarios futuros y sus implicancias a nivel ambiental y económico.

Desde una perspectiva ambiental, el tipo de fuente energética utilizada en la generación eléctrica determina en gran medida el volumen de emisiones de gases de efecto invernadero (GEI), así como otros impactos relacionados con el uso del agua, el suelo y la biodiversidad. Por ello, el análisis de la matriz de generación resulta clave para monitorear el progreso hacia metas climáticas y para identificar oportunidades de mejora en términos de eficiencia y sostenibilidad.

Desde el punto de vista económico, la diversificación de fuentes y la integración de tecnologías limpias pueden contribuir a reducir la vulnerabilidad frente a fluctuaciones de precios en los mercados energéticos internacionales, mejorar la seguridad energética y estimular la innovación y el desarrollo de nuevas cadenas de valor. Además, una planificación energética basada en datos empíricos permite a los gobiernos y organismos reguladores diseñar políticas públicas más eficaces, promover inversiones sostenibles y garantizar el acceso equitativo a servicios energéticos modernos.

En este sentido, el estudio sistemático de los datos sobre generación eléctrica clasificados por tipo de fuente constituye una base sólida para la toma de decisiones informadas, el diseño de políticas integradas y la construcción de sistemas eléctricos más resilientes, sostenibles y socialmente inclusivos.

1.1. Contexto

Colombia se encuentra en una etapa clave de transformación de su matriz energética, impulsada por la necesidad de diversificar las fuentes de generación eléctrica, reducir la dependencia de recursos hidroeléctricos y fósiles, y avanzar

hacia un sistema energético más sostenible y resiliente frente al cambio climático. Históricamente, el país ha contado con una alta participación de la generación hidroeléctrica —superior al 60 % en varios años—, lo cual ha permitido mantener una baja intensidad de emisiones de carbono en el sector eléctrico en comparación con otras economías de América Latina y el mundo. No obstante, esta dependencia ha expuesto al sistema a riesgos asociados con la variabilidad climática, especialmente durante los períodos de sequía causados por fenómenos como El Niño, que comprometen la estabilidad del suministro energético.

Con el objetivo de diversificar su matriz energética y fortalecer su resiliencia, Colombia ha establecido un marco normativo robusto para la promoción de las fuentes no convencionales de energía renovable (FNCER), particularmente las tecnologías solar fotovoltaica y eólica. La Ley 1715 de 2014, junto con desarrollos posteriores como la Ley 2099 de 2021 y las subastas de contratos de largo plazo organizadas por el Ministerio de Minas y Energía, han creado condiciones favorables para la inversión en este tipo de tecnologías. En línea con sus compromisos internacionales, como el Acuerdo de París y su Contribución Determinada a Nivel Nacional (NDC), el país se ha trazado metas ambiciosas para aumentar la participación de energías renovables en su matriz y reducir las emisiones del sector energético.

A pesar de estos avances, el desarrollo e integración de las FNCER enfrenta diversos desafíos técnicos, económicos y sociales. Entre ellos se destacan la necesidad de fortalecer la infraestructura de transmisión, adaptar la regulación del mercado eléctrico para facilitar la participación de fuentes intermitentes, resolver aspectos relacionados con la consulta previa en territorios de comunidades étnicas y garantizar la estabilidad financiera de los proyectos a largo plazo. Estos desafíos requieren una planificación energética integral, basada en evidencia empírica y en análisis comparativos que permitan identificar tendencias, oportunidades y barreras.

En este contexto, el presente análisis se basa en un conjunto de datos proporcionado por la Agencia Internacional de Energía (AIE), que reúne información mensual sobre la generación eléctrica de múltiples países —incluido Colombia— entre 2010 y 2025. El dataset está estructurado por año, mes, tipo de fuente energética y volumen de generación expresado en gigavatios-hora (GWh). Esta información permite cuantificar y caracterizar la evolución del aporte de las energías renovables a la generación eléctrica nacional, así como comparar el desempeño colombiano con el de otras naciones en la región y el mundo.

El análisis de esta información no solo proporciona una visión detallada del proceso de transición energética en Colombia, sino que también contribuye a la formulación de políticas públicas, al diseño de estrategias de inversión y a la promoción de un sistema eléctrico más eficiente, sostenible e inclusivo.

1.2. Definición del problema

1.2.1. Objetivo espécifico

Analizar la evolución y el aporte de las fuentes de energía renovable en la generación eléctrica en Colombia entre los años 2010 y 2025, con base en datos estadísticos comparativos, a fin de identificar tendencias, avances y desafíos en el marco de la transición energética del país.

1.2.2. Objetivos espécificos

- Caracterizar la participación de las diferentes fuentes renovables (solar, eólica, hidroeléctrica, geotérmica y otras) en la matriz de generación eléctrica de Colombia durante el período 2010–2025.
- Identificar los principales desafíos y oportunidades asociados a la integración de fuentes renovables en el sistema eléctrico colombiano, en relación con factores técnicos, regulatorios y ambientales.

1.3. preguntas orientadora

¿Cómo ha evolucionado el aporte de las energías renovables en la generación eléctrica en Colombia entre 2014 y 2025, y qué factores explican su desempeño?

2. Materiales y métodos

2.1. Enfoque metodológico

Este estudio se enmarca dentro del ciclo de vida de un proyecto de Machine Learning, específicamente en las etapas de adquisición de datos, limpieza, análisis exploratorio, visualización y análisis multidimensional. Aunque no se realiza modelado predictivo, se adoptan herramientas y metodologías propias de la ciencia de datos para estructurar un análisis riguroso, reproducible y orientado a la extracción de conocimiento a partir de datos energéticos. El enfoque se limita exclusivamente al caso de Colombia.

2.2. Conjunto de datos

El análisis se basa en un conjunto de datos inicialmente obtenido desde el portal Kaggle, el cual contiene información mensual de generación eléctrica clasificada por tipo de fuente energética, país, año y mes, expresada en gigavatios-hora (GWh). Esta base cubre el período comprendido entre 2010 y 2022. Para extender el alcance temporal hasta 2025, se aplicaron técnicas de web scraping sobre fuentes oficiales y sitios especializados, lo que permitió actualizar el conjunto de datos con los registros más recientes disponibles.

El dataset final fue consolidado y almacenado en formato CSV, lo cual facilitó su manipulación y análisis mediante herramientas de programación. Se realizó una curaduría manual y automatizada para garantizar la coherencia, completitud y calidad de los datos utilizados.

El conjunto de datos utilizado contiene registros mensuales de generación eléctrica para distintos países, incluyendo Colombia, y presenta la siguiente estructura de variables en la tabla 1:

2.3. Herramientas tecnológicas

El procesamiento y análisis de los datos se llevaron a cabo en el lenguaje de programación Python, utilizando el entorno Jupyter Notebook por su versatilidad en la integración de código, visualizaciones y documentación. Las principales bibliotecas empleadas fueron:

- **Pandas:** para la manipulación y estructuración tabular de los datos.
- **NumPy:** para el manejo de operaciones numéricas.
- **Matplotlib y Seaborn:** para la creación de visualizaciones descriptivas y comparativas.
- **Scikit-learn**: para métodos de reducción de dimensiones, como PCA.

2.4. Proceso metodológico

El proceso seguido se puede desglosar en las siguientes etapas [García and Molina, 2018]:

1. Adquisición y consolidación de datos:
    - Descarga inicial del dataset desde Kaggle.

Variable	Descripción
COUNTRY	Nombre del país al que corresponde el registro.
CODE_TIME	Código temporal en formato abreviado que indica el mes y año (por ejemplo, JAN2010).
TIME	Representación en texto del mes y año (por ejemplo, January 2010).
YEAR	Año correspondiente al registro.
MONTH	Mes correspondiente al registro, en formato numérico (1–12).
MONTH_NAME	Nombre del mes correspondiente al registro.
PRODUCT	Tipo de fuente energética utilizada para la generación eléctrica (e.g., Hydro, Wind, Solar, etc.).
VALUE	Cantidad de electricidad generada en gigavatios-hora (GWh).
DISPLAY_ORDER	Indicador numérico utilizado para el ordenamiento visual de los productos.
yearToDate	Acumulado de generación eléctrica en GWh desde el inicio del año hasta el mes actual.
previousYearToDate	Acumulado de generación eléctrica para el mismo periodo del año anterior.
share	Porcentaje de participación del producto en la generación total del país (en formato decimal, por ejemplo 0.12 = 12 %).

Tabla 1: Categorias del conjunto de datos

- Actualización de datos mediante técnicas de web scraping para ampliar el rango temporal hasta 2025.
- Fusión de fuentes y estandarización del formato general.

2. Limpieza de datos:
   - Revisión de datos faltantes, valores atípicos o inconsistentes.
   - Conversión de tipos de datos y normalización de etiquetas (por ejemplo, estandarización de nombres de fuentes energéticas).
   - Filtro de registros para conservar únicamente la información correspondiente a Colombia.
3. Análisis exploratorio de datos (EDA):
   - Estadísticas descriptivas generales sobre la generación eléctrica total y por tipo de fuente.
   - Evaluación de la evolución temporal de la participación de energías renovables frente a no renovables.
4. Visualización de tendencias:
   - Representación gráfica de series temporales, porcentajes de participación, acumulados anuales y estacionales.
   - Comparación de patrones interanuales y detección de puntos de inflexión relevantes.
5. Análisis multidimensional:
   - Aplicación de técnicas para explorar las relaciones entre variables como tipo de generación, volumen mensual, variabilidad estacional, y evolución anual.
   - Uso de gráficos de calor, diagramas de dispersión múltiple o métodos de reducción de dimensiones (como PCA), en caso de ser necesarios.

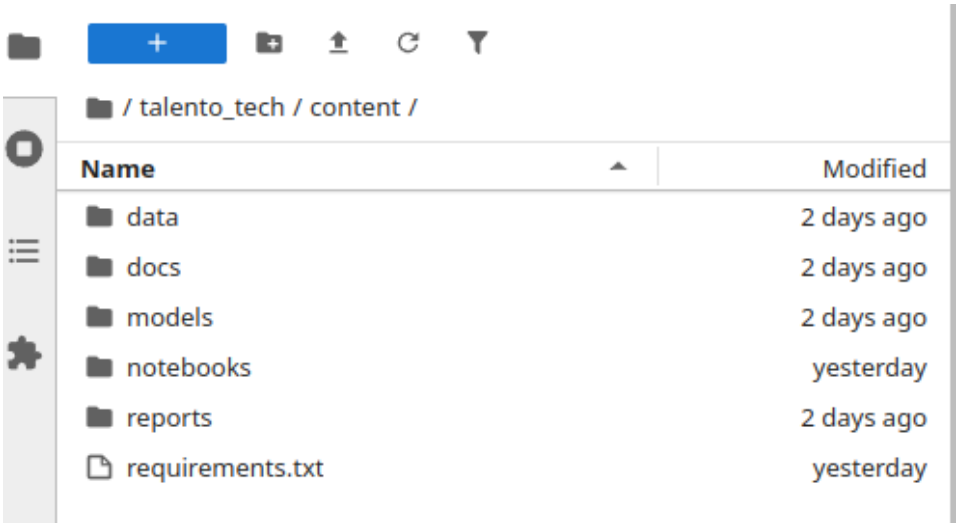

Figura 1: Estructura cookiecutter para proyecto de ciencia de datos

3. Resultados

En la definición de la estructura del proyecto se ha utilizado el scaffold cookiecutter Data Science el cual es una estrcutura lógica, flexible y estándar para compartir proyecto de ciencia de datos. La estructura se muestra en la figura 1:

3.1. Enriquecimiento de la información

Se ha mejorado el conjunto de datos adicionando los años 2023, 2024 y los meses de enero y febrero para el 2025. Se ha utilizado técnicas de web scrapping para obtener la información adicional.

```
# Import necessary libraries
import os
import csv
import requests
from pprint import pprint

# Set show information about API request
VERBOSE = True

# Define API endpoints for getting available years, products, and countries
api_list_template = 'https://api.iea.org/mes/list/%s'

# Define API endpoint for getting monthly data for a specific country, year, month, and product
api_information_template =
↪  'https://api.iea.org/mes/latest/month?COUNTRY=%s&YEAR=%s&MONTH=%s&PRODUCT=%s&share=true'

# Get lists of available years, products, and countries from the API
years = requests.get(api_list_template % 'YEAR').json()
products = requests.get(api_list_template % 'PRODUCT').json()
countries = requests.get(api_list_template % 'COUNTRY').json()
```

```
20
21  # Define the header row for the CSV file that will store the data
22  header = [
23  'COUNTRY',           # Name of the country
24  'CODE_TIME',         # A code that represents the month and year (e.g., JAN2010 for January 2010)
25  'TIME',              # The month and year in a more human-readable format (e.g., January 2010)
26  'YEAR',              # The year of the data point
27  'MONTH',             # The month of the data point as a number (1-12)
28  'MONTH_NAME',        # The month of the data point as a string (e.g., January)
29  'PRODUCT',           # The type of energy product (e.g., Hydro, Wind, Solar)
30  'VALUE',             # The amount of electricity generated in gigawatt-hours (GWh)
31  'DISPLAY_ORDER',     # The order in which the products should be displayed
32  'yearToDate',        # The amount of electricity generated for the current year up to the current month
    ↪  in GWh
33  'previousYearToDate',# The amount of electricity generated for the previous year up to the current
    ↪  month in GWh
34  'share'              # The share of the product in the total electricity generation for the country in
    ↪  decimal format
35  ]
36
37  # Check if data.csv file exists
38  if os.path.isfile('data.csv'):
39      # Check if file is empty
40      if os.stat('data.csv').st_size != 0:
41          # Read CSV file and get the last line of it
42          with open('data.csv', 'r') as csv_file:
43          last_line = csv_file.readlines()[-1].split(',')
44
45          # Extract the information from the last line get their index values
46          index_last_year = years.index(int(last_line[3]))
47          index_last_month = int(last_line[4])
48          index_last_country = countries.index(last_line[0])
49          index_last_product = products.index(last_line[6]) + 1
50
51      else:
52          # If CSV file is empty
53          index_last_year, index_last_month, index_last_country, index_last_product = 0, 1, 0, 0
54
55  # Open CSV file for writing
56  with open('data.csv', 'a+', newline='') as csv_file:
57      writer = csv.DictWriter(csv_file, fieldnames=header)
58
59      # Scrape the data and write it to the CSV file
60      for year in years[index_last_year:]:
61          for month in range(index_last_month, 13):
62              for country in countries[index_last_country:]:
63                  # Replace apostrophes in the country name with %27 to create a valid URL
64                  country = country.replace('\'', '%27')
65
66                  for product in products[index_last_product:]:
67                      # Send an API request to get monthly data for the current country, year, month and
                        ↪  product
68                      response = requests.get(
```

```
                        api_information_template % (country, year, month, product)
                    )

                    # Check if the API response is OK
                    if response.ok:
                        # Parse the response JSON
                        response = response.json()


                        # if response has data
                        if len(response['latest']) != 0:

                            print(response)

                            # Create a dictionary of the data to write to the CSV file
                            result = dict()

                            # Extract the data from the response and add it to the result dictionary
                            for key in response['latest'][0].keys():
                                result[key] = response['latest'][0][key]

                            # Add year-to-date, previous-year-to-date and share data to the result
                             ↪ dictionary
                            result['yearToDate'] = response['yearToDate']
                            result['previousYearToDate'] = response['previousYearToDate']
                            result['share'] = response['share']

                            # Write the result dictionary to the CSV file
                            writer.writerow(result)

                            # If verbose mode is on, print the result for this month
                            if VERBOSE:
                                pprint(result, sort_dicts=False)
                                print('__________________________')

        index_last_product = 0
        index_last_month = 1
        index_last_country = 0
```

En la figura 2 se muestra el web scrapping en funcionamiento descargando la información de los años faltantes.

3.2. Limpieza de datos

Para la limpieza de datos se ha utilizado el archivo con la información de producción de energía electríca desde el año 2010 hasta 2025 inclusive.

3.2.1. Importación de las librerías y asignación del directorio de trabajo

```
# import library
import pandas as pd
import os
```

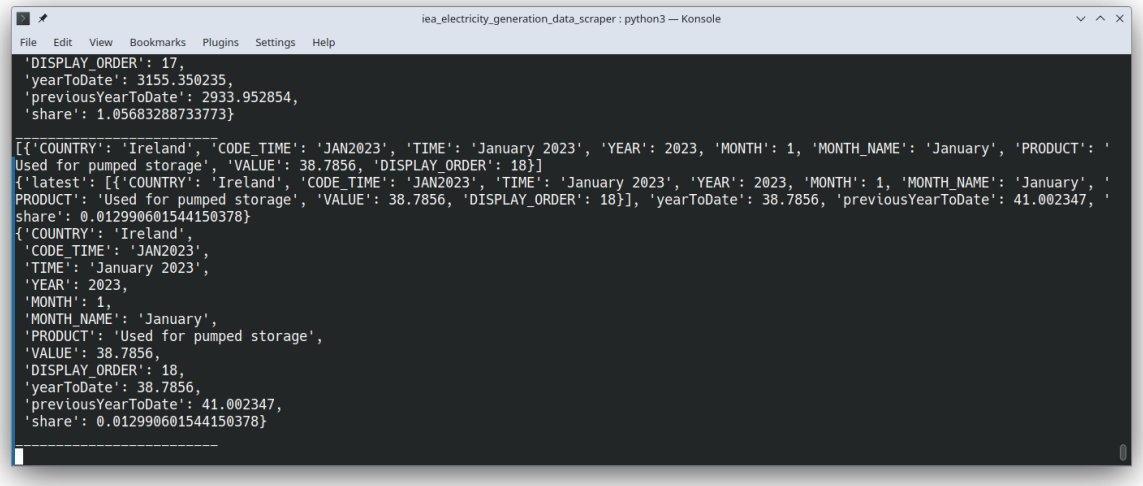

Figura 2: Web scrapping descargando la información de los años 2024 y 2025

```
4
5  # set default directory
6  os.chdir('/home/lbertel/code/talento_tech/content')
7  !ls
```

```
data  docs  models  notebooks  reports requirements.txt
```

3.2.2. Carga del archivo con datos en crudo

```
1  # load raw file
2  df_data_all_country = pd.read_csv('data/raw/data.csv')
3
4  # print all data
5  df_data_all_country.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 213138 entries, 0 to 213137
Data columns (total 12 columns):
#   Column              Non-Null Count   Dtype
---  ------              --------------   -----
0   COUNTRY             213138 non-null  object
1   CODE_TIME           213138 non-null  object
2   TIME                213138 non-null  object
3   YEAR                213138 non-null  int64
4   MONTH               213138 non-null  int64
```

```
5   MONTH_NAME          213138 non-null  object
6   PRODUCT             213138 non-null  object
7   VALUE               213138 non-null  float64
8   DISPLAY_ORDER       213138 non-null  int64
9   yearToDate          213138 non-null  float64
10  previousYearToDate  196016 non-null  float64
11  share               213138 non-null  float64
dtypes: float64(4), int64(3), object(5)
memory usage: 19.5+ MB
```

3.2.3. Recuperar los registros asociados al país Colombia

```
# select row with COUNTRY is equal Colombia
df_data_colombia = df_data_all_country[df_data_all_country['COUNTRY'] == 'Colombia']

# print first record of Colombia
df_data_colombia.head(10)

# size dataframe
len(df_data_colombia)
```

3104

3.2.4. Recuperar los registros asociados al país Colombia

```
    # select row with COUNTRY is equal Colombia
    df_data_colombia = df_data_all_country[df_data_all_country['COUNTRY'] == 'Colombia']

    # print first record of Colombia
    df_data_colombia.head(10)

    # size dataframe
    len(df_data_colombia)
```

3104

3.2.5. Eliminar categorias no necesarias en el estudio

```
# show columns after delete
print(df_data_colombia.columns)

# delete columns
df_temp = df_data_colombia.drop(['COUNTRY', 'CODE_TIME', 'TIME', 'MONTH_NAME', 'DISPLAY_ORDER',
 ↪  'yearToDate', 'previousYearToDate', 'share'], axis=1)

```

```
# show columns before delete
print(df_temp.columns)
print('')
print('---------------------------------')
print('')

df_temp.info()
```

```
Index(['COUNTRY', 'CODE_TIME', 'TIME', 'YEAR', 'MONTH', 'MONTH_NAME',
'PRODUCT', 'VALUE', 'DISPLAY_ORDER', 'yearToDate', 'previousYearToDate',
'share'],
dtype='object')
Index(['YEAR', 'MONTH', 'PRODUCT', 'VALUE'], dtype='object')

---------------------------------

<class 'pandas.core.frame.DataFrame'>
Index: 3104 entries, 46557 to 212133
Data columns (total 4 columns):
#   Column   Non-Null Count  Dtype
---  ------   --------------  -----
0   YEAR     3104 non-null   int64
1   MONTH    3104 non-null   int64
2   PRODUCT  3104 non-null   object
3   VALUE    3104 non-null   float64
dtypes: float64(1), int64(2), object(1)
memory usage: 121.2+ KB
```

3.2.6. Eliminar registros con valores nulos y que en la categoría PRODUCT tenga totales

```
# delete row with null values
len(df_temp)
df_without_null = df_temp.dropna(how='any')

# size dataframe without null
len(df_without_null)

# eliminate row with PRODUCT containt Total
filter_total = df_without_null[~df_without_null['PRODUCT'].str.contains('Total', case=False, na=False)]

len(filter_total)
```

```
2706
```

3.2.7. Guardar el dataframe en un archivo

```
# save the new dataframe
filter_total.to_csv('data/processed/colombia_data.csv', index=False)

# information dataframe
filter_total.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2706 entries, 46557 to 212133
Data columns (total 4 columns):
#   Column   Non-Null Count  Dtype
---  ------   --------------  -----
0   YEAR     2706 non-null   int64
1   MONTH    2706 non-null   int64
2   PRODUCT  2706 non-null   object
3   VALUE    2706 non-null   float64
dtypes: float64(1), int64(2), object(1)
memory usage: 105.7+ KB
```

3.2.8. Mostrar información del dataset ya limpio

```
# dimension of dataframe
filter_total.shape

# missing values
filter_total.isnull().sum(axis=0)
```

```
YEAR       0
MONTH      0
PRODUCT    0
VALUE      0
dtype: int64
```

3.3. Análisis multidimensional

3.3.1. Importacioón de librerías y Carga del archivo de trabajo

```
# import library
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
```

```
8
9  # set default directory
10 os.chdir('/home/lbertel/code/talento_tech/content')
11
12 # set style matplotlib
13 plt.style.use('tableau-colorblind10')
14
15 # load data
16 colombia_df = pd.read_csv('data/processed/colombia_data.csv')
17 colombia_df.head(10)
```

	YEAR	MONTH	PRODUCT	VALUE
0	2014	1	Hydro	3903.977
1	2014	1	Wind	5.648
2	2014	1	Solar	1.065
3	2014	1	Coal	521.938
4	2014	1	Oil	139.219
5	2014	1	Natural gas	1031.146
6	2014	1	Combustible renewables	99.721
7	2014	1	Net electricity production	5702.714
8	2014	1	Electricity supplied	5555.847
9	2014	1	Distribution losses	536.164

3.3.2. Cantidad de energía clasificada según tipo

```
1  # energy clasification dataset
2  order = colombia_df.groupby('PRODUCT').mean()['VALUE'].sort_values(ascending=False).index
3
4  fig, ax = plt.subplots(figsize=(8, 8))
5  fig.suptitle('Cantidad de energía clasificada según tipo')
6
7  sns.barplot(data=colombia_df, x='VALUE', y='PRODUCT', ax=ax, estimator='mean', errorbar=None,
   ↪  order=order)
8  ax.set_xlabel('Cantidad de Energía [GWh]')
9  ax.set_ylabel('Tipo de energía')
10
11 plt.tight_layout()
```

3.3.3. Evolución de la producción neta de energía en Colombia (Resolución Anual)

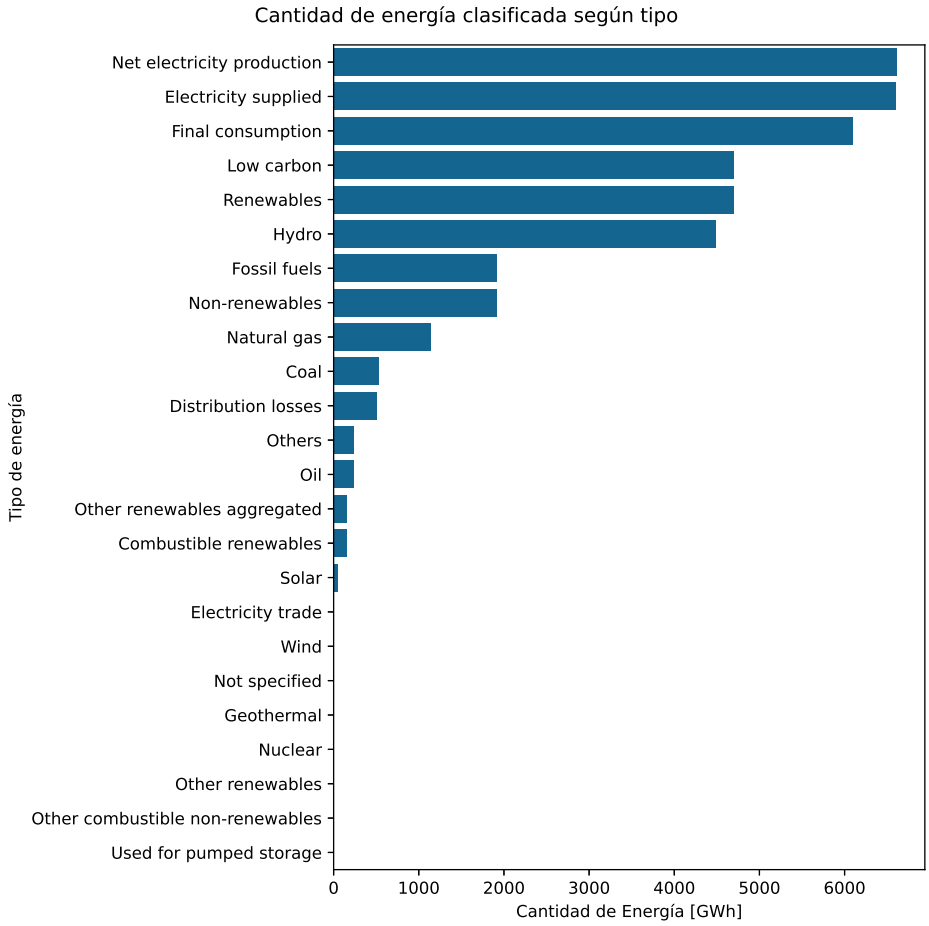

Figura 3: Cantidad de energía clasificada por tipo

```python
# energy production 2014-2025
filt = (colombia_df['PRODUCT'] == 'Net electricity production')
df_net = colombia_df.loc[filt]
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción neta de energía en Colombia (Resolución Anual)')

sns.pointplot(data=df_net, x='YEAR', y='VALUE', ax=ax, estimator='mean', errorbar=None)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')

plt.tight_layout()
```

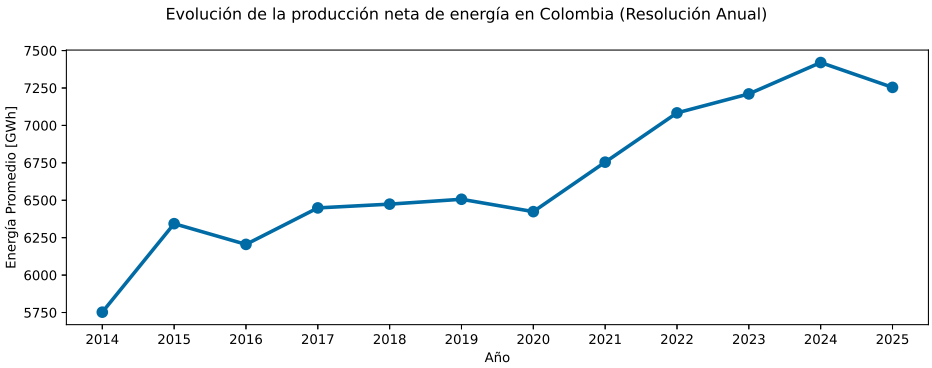

Figura 4: Evolución de la producción neta de energía en Colombia

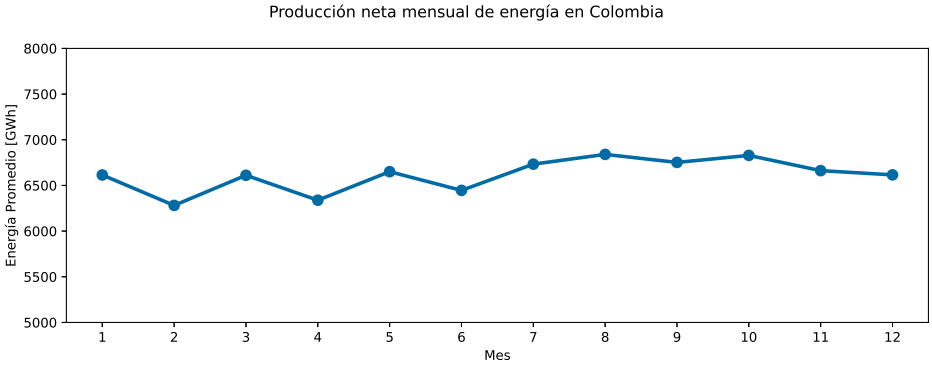

Figura 5: Producción neta mensual de energía en Colombia

3.3.4. Producción neta mensual de energía en Colombia

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Producción neta mensual de energía en Colombia')

sns.pointplot(data=df_net, x='MONTH', y='VALUE', ax=ax, estimator='mean', errorbar=None)
ax.set_xlabel('Mes')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_ylim([5000, 8000])

plt.tight_layout()
```

3.3.5. Matriz energética de Colombia

```
# select
filt = ['Wind', 'Solar', 'Other renew. agg.', 'Others', 'Nuclear', 'Natural gas', 'Hydro', 'Coal']
```

```
df_gen = colombia_df[colombia_df['PRODUCT'].isin(filt)]

matrix = df_gen.groupby('PRODUCT').sum()['VALUE'].sort_values(ascending=False)
matrix
```

```
PRODUCT
Hydro          601415.683695
Natural gas    153592.481385
Coal            71158.873666
Others          31527.442435
Solar            7138.600132
Wind              629.372654
Nuclear             0.000000
Name: VALUE, dtype: float64
```

```
labels = matrix.index

fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Matriz energética de Colombia')

ax.pie(x=matrix, labels=labels, autopct='%.0f%%', rotatelabels=True,startangle=180,
 ↪  colors=sns.color_palette("pastel"));
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5), labels=["{} -
 ↪  {:.2f}%".format(i,j/sum(matrix)*100) for i,j in zip(labels,matrix)], frameon=False)

plt.tight_layout()
```

3.3.6. Energía según tipo de generación

```
order = df_gen.groupby('PRODUCT').mean()['VALUE'].sort_values(ascending=False).index

fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Energía según tipo de generación')

sns.barplot(data=df_gen, x='VALUE', y='PRODUCT', ax=ax, estimator='mean', errorbar=None, order=order)
ax.set_xlabel('Energía promedio [GWh]')
ax.set_ylabel('Tipo de generación')

plt.tight_layout()
```

3.3.7. Evolución de la producción de energía en Colombia según tipo de generación

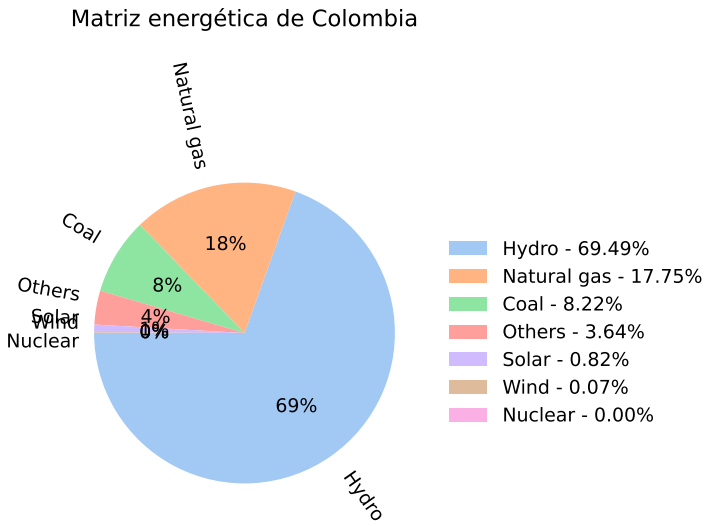

Figura 6: Matriz energética de Colombia

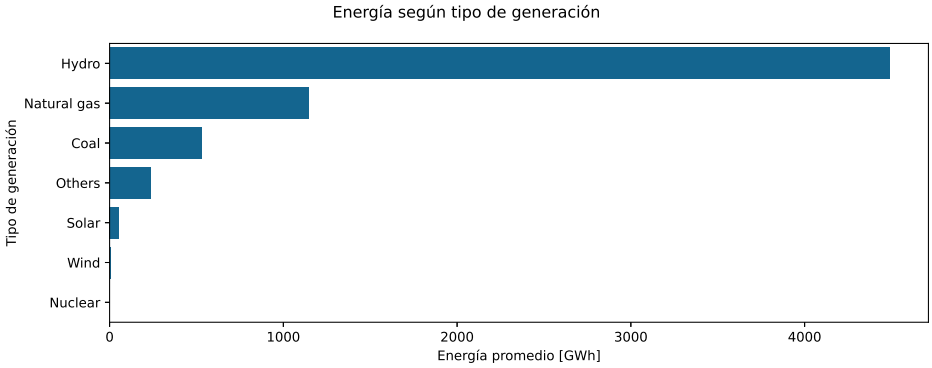

Figura 7: Energía según tipo de generación

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia según tipo de generación')

sns.lineplot(data=df_gen, x='YEAR', y='VALUE', ax=ax, hue='PRODUCT', estimator='mean', errorbar=None,
 ↪  hue_order=order,)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_xlim([2014, 2025])
ax.set_ylim([0, 6000])
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

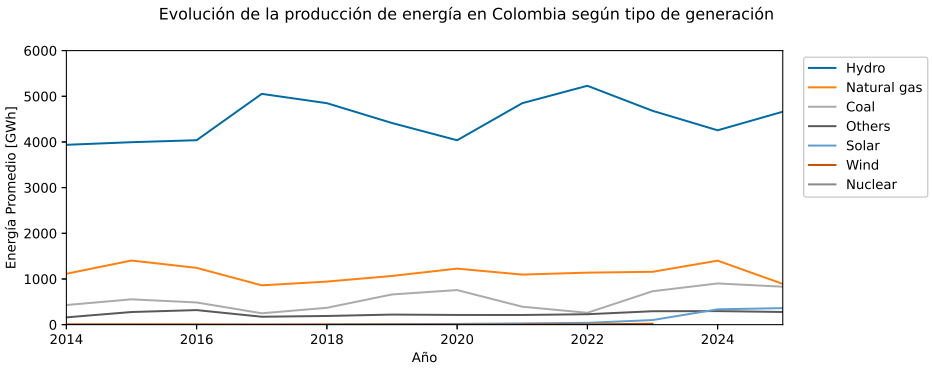

Figura 8: Evolución de la producción de energía en Colombia según tipo de generación

3.3.8. Clasificación de la generación por tipo y año

```
df_gen_year = df_gen.groupby(['YEAR','PRODUCT']).mean()
df_gen_year.sort_values(['YEAR', 'VALUE'], ascending=[True, False], inplace=True)
df_gen_year.head(14)
```

		MONTH	VALUE
YEAR	PRODUCT		
2014	Hydro	6.5	3937.424333
	Natural gas	6.5	1113.783417
	Coal	6.5	429.340833
	Others	6.5	159.849250
	Wind	6.5	5.790167
	Solar	6.5	0.745583
2015	Hydro	6.5	3994.836750
	Natural gas	6.5	1404.155417
	Coal	6.5	554.467500
	Others	6.5	276.451667
	Wind	6.5	5.624750
	Solar	6.5	0.745583
2016	Hydro	6.5	4038.266250
	Natural gas	6.5	1243.133167

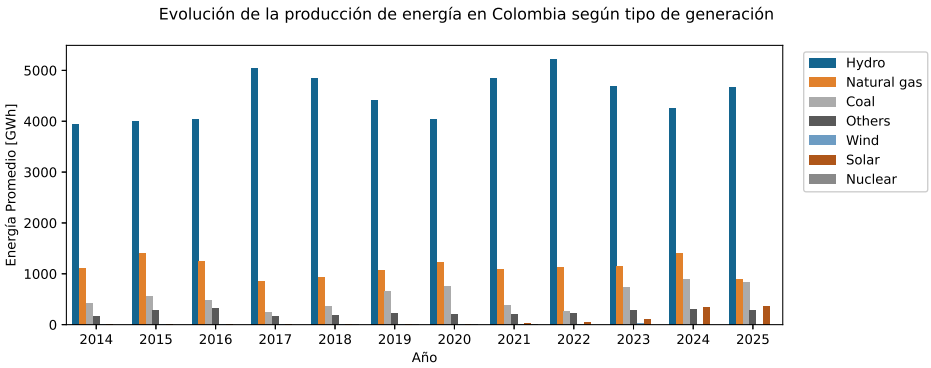

Figura 9: Evolución de la producción de energía en Colombia según tipo de generación

3.3.9. Evolución de la producción de energía en Colombia según tipo de generación

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia según tipo de generación')

sns.barplot(data=df_gen_year.reset_index(), x='YEAR', y='VALUE', hue='PRODUCT')
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

3.3.10. Clasificación de las energías renovables y no renovables en Colombia

```
filt = ['Non-renewables', 'Renewables']
df_nr = colombia_df[colombia_df['PRODUCT'].isin(filt)]
df_nr.head()
```

	YEAR	MONTH	PRODUCT	VALUE
12	2014	1	Renewables	4010.411
13	2014	1	Non-renewables	1692.303
30	2014	2	Renewables	3703.603
31	2014	2	Non-renewables	1661.258
48	2014	3	Renewables	4313.454

3.3.11. Totales de energias renovables y no renovables

```
suma = df_nr.groupby('PRODUCT').sum()['VALUE'].sort_values(ascending=False)
suma
```

```
PRODUCT
Renewables        629704.771240
Non-renewables    256278.797486
Name: VALUE, dtype: float64
```

3.3.12. Proporción de producción de energía en Colombia Renovables vs No renovables

```
fig, ax = plt.subplots(figsize=(5, 5))
fig.suptitle('Proporción de producción de energía en Colombia Renovables vs No renovables')

ax.pie(x=suma, labels=suma.index, autopct='%.1f%%', startangle=90, colors=sns.color_palette("pastel"))

plt.tight_layout()
```

3.3.13. Evolución de la producción de energía en Colombia. Renovable vs no renovable (2014-2025)

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia. Renovable vs no renovable
↪  (2014-2025)')

sns.lineplot(data=df_nr, x='YEAR', y='VALUE', ax=ax, hue='PRODUCT', estimator='mean', errorbar=None)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_xlim(2014, 2026)
ax.set_ylim([0, 6000])
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

3.3.14. Descriminación de energías renovables

```
filt = ['Wind', 'Solar', 'Other renewables', 'Hydro', 'Geothermal', 'Combustible renewables']
df_ren = colombia_df[colombia_df['PRODUCT'].isin(filt)]
df_ren.head()
```

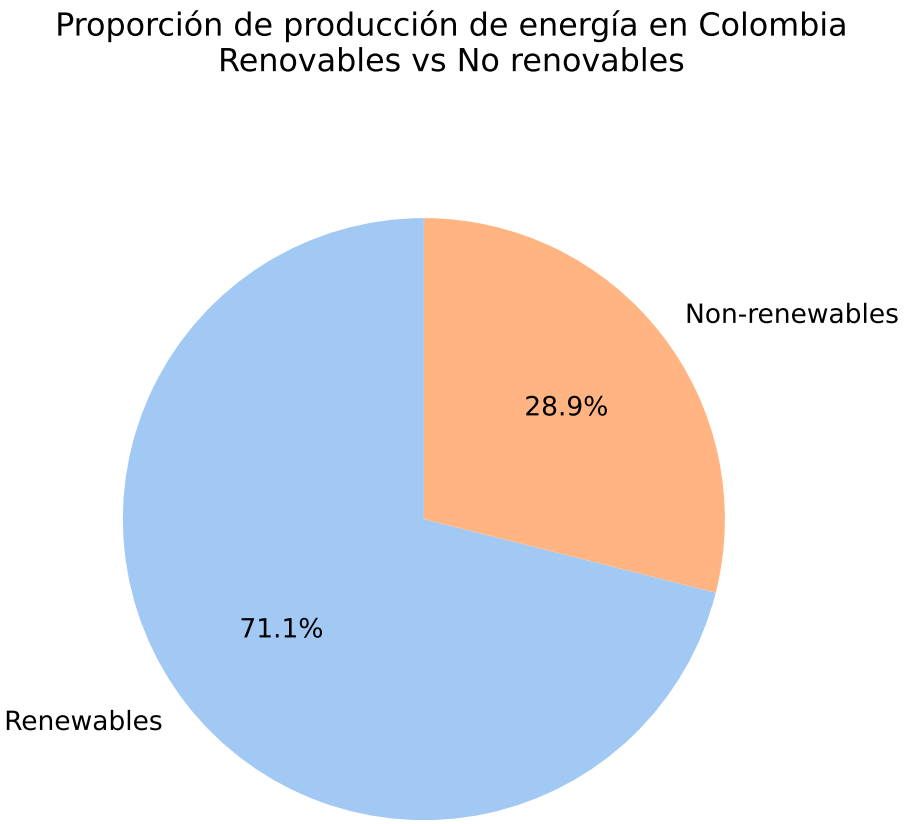

Figura 10: Proporción de producción de energía en Colombia Renovables vs No renovables

	YEAR	MONTH	PRODUCT	VALUE
0	2014	1	Hydro	3903.977
1	2014	1	Wind	5.648
2	2014	1	Solar	1.065
6	2014	1	Combustible renewables	99.721
18	2014	2	Hydro	3598.260

3.3.15. Acumulado de energías renovables

```
suma = df_ren.groupby('PRODUCT').sum()['VALUE'].sort_values(ascending=False)
suma
```

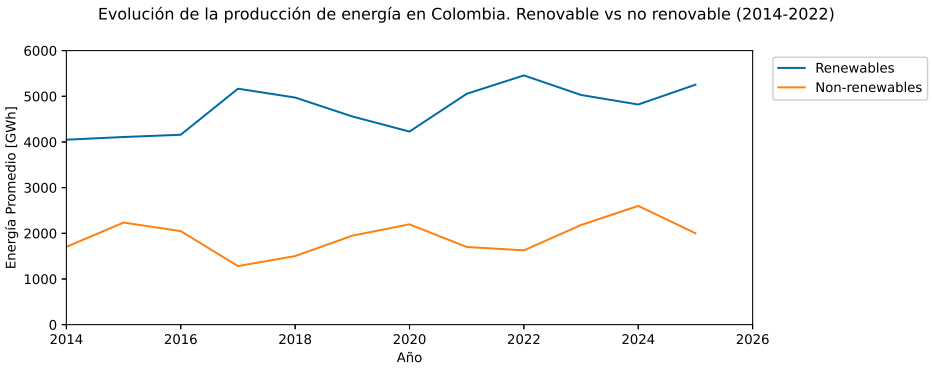

Figura 11: Evolución de la producción de energía en Colombia. Renovable vs no renovable (2014-2025)

```
PRODUCT
Hydro                      601415.683695
Combustible renewables      20521.114759
Solar                        7138.600132
Wind                          629.372654
Geothermal                      0.000000
Other renewables                0.000000
Name: VALUE, dtype: float64
```

3.3.16. Producción de energía de fuentes Renovables

```
order = df_ren.groupby('PRODUCT').mean()['VALUE'].sort_values(ascending=False).index

fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Producción de energía. Fuentes Renovables')

sns.barplot(data=df_ren, x='VALUE', y='PRODUCT', ax=ax, estimator='mean', errorbar=None, order=order)
ax.set_xlabel('Energía promedio [GWh]')
ax.set_ylabel('Tipo de generación')

plt.tight_layout()
```

3.3.17. Evolución de la producción de energía en Colombia. Fuentes Renovables (2014-2025)

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia. Fuentes Renovables (2014-2025)')

sns.lineplot(data=df_ren, x='YEAR', y='VALUE', ax=ax, hue='PRODUCT', estimator='mean', errorbar=None)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_xlim(2014, 2026)
```

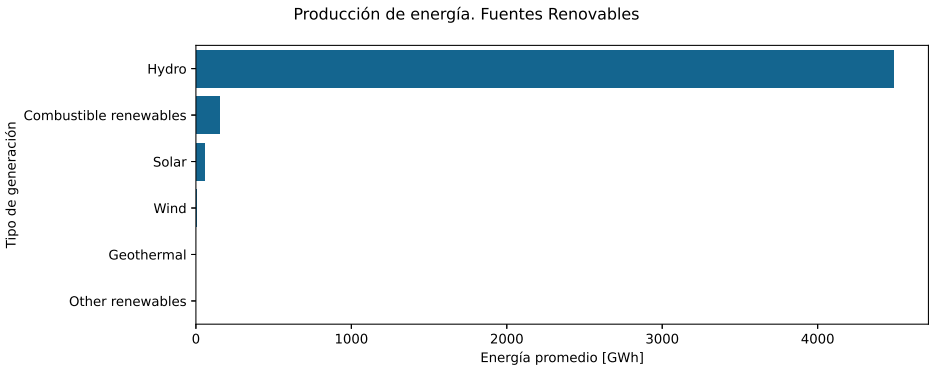

Figura 12: Producción de energía de fuentes Renovables

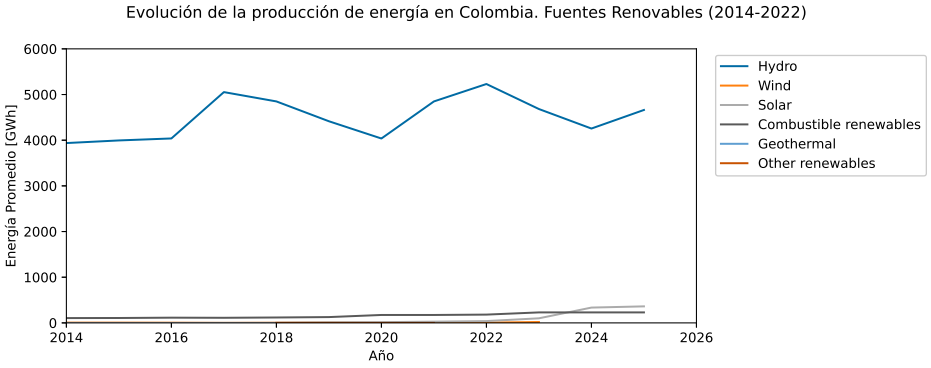

Figura 13: Evolución de la producción de energía en Colombia. Fuentes Renovables (2014-2025)

```
ax.set_ylim([0, 6000])
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

3.3.18. Producción de energía. Fuentes Renovables. No hidraulica

```
order = df_ren[df_ren["PRODUCT"] !=
 ↪  "Hydro"].groupby('PRODUCT').mean()['VALUE'].sort_values(ascending=False).index

fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Producción de energía. Fuentes Renovables. No hidraulica')

sns.barplot(data=df_ren[df_ren["PRODUCT"] != "Hydro"], x='VALUE', y='PRODUCT', ax=ax, estimator='mean',
 ↪  errorbar=None, order=order)
ax.set_xlabel('Energía promedio [GWh]')
ax.set_ylabel('Tipo de generación')
```

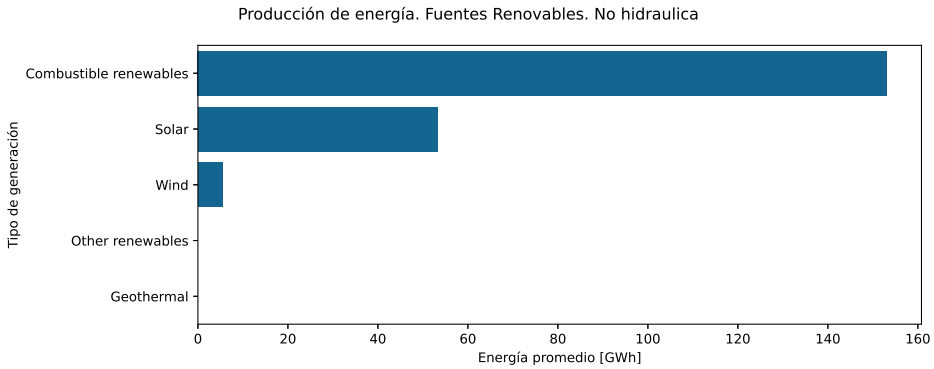

Figura 14: Producción de energía. Fuentes Renovables. No hidraulica

```

plt.tight_layout()
```

3.3.19. Evolución de la producción de energía en Colombia fuentes renovables (2014-2025) no hidraulica

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia. Fuentes Renovables (2014-2022). No hidraulica')

sns.lineplot(data=df_ren[df_ren["PRODUCT"] != "Hydro"], x='YEAR', y='VALUE', ax=ax, hue='PRODUCT', estimator='mean', errorbar=None)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_xlim(2014, 2026)
# ax.set_ylim([0, 6000])
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

3.3.20. Producción de energía de fuentes no renovables

```
filt = ['Coal', 'Natural gas', 'Fossil fuels', 'Nuclear']
df_notren = colombia_df[colombia_df['PRODUCT'].isin(filt)]
df_notren.head()
```

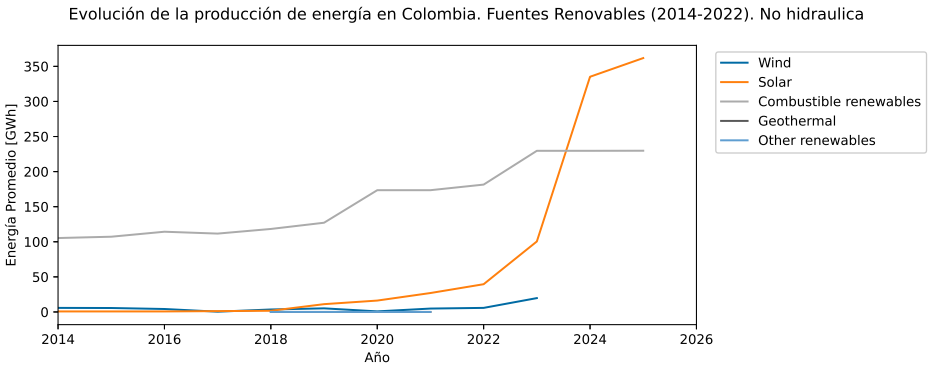

Figura 15: Producción de energía. Fuentes Renovables. No hidraulica

	YEAR	MONTH	PRODUCT	VALUE
3	2014	1	Coal	521.938
5	2014	1	Natural gas	1031.146
17	2014	1	Fossil fuels	1692.303
21	2014	2	Coal	413.943
23	2014	2	Natural gas	1095.052

3.3.21. Producción de energía de fuentes no renovables por tipo

```
suma = df_notren.groupby('PRODUCT').sum()['VALUE'].sort_values(ascending=False)
suma
```

```
PRODUCT
Fossil fuels    256278.797486
Natural gas     153592.481385
Coal             71158.873666
Nuclear              0.000000
Name: VALUE, dtype: float64
```

3.3.22. Producción de energía de fuentes no renovables

```
order = df_notren.groupby('PRODUCT').mean()['VALUE'].sort_values(ascending=False).index

fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Producción de energía. Fuentes No Renovables')

sns.barplot(data=df_notren, x='VALUE', y='PRODUCT', ax=ax, estimator='mean', errorbar=None,
 ↪  order=order)
```

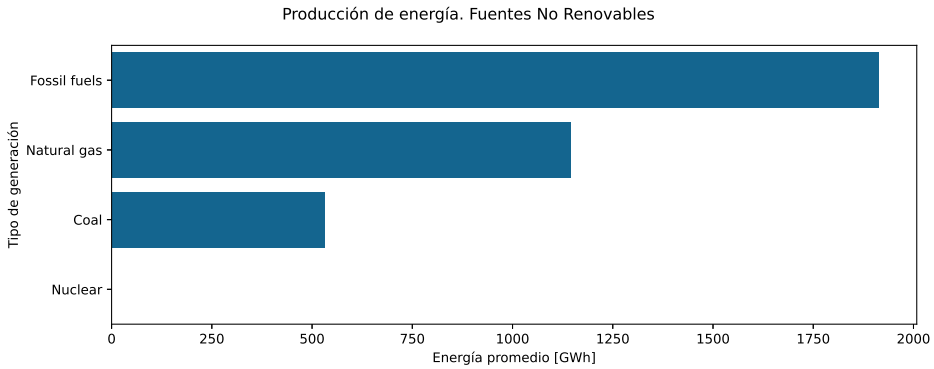

Figura 16: Producción de energía de fuentes no renovables

```
ax.set_xlabel('Energía promedio [GWh]')
ax.set_ylabel('Tipo de generación')

plt.tight_layout()
```

3.3.23. Evolución de la producción de energía en Colombia de fuentes no renovables (2014-2025)

```
fig, ax = plt.subplots(figsize=(10, 4))
fig.suptitle('Evolución de la producción de energía en Colombia. Fuentes No Renovables (2014-2022)')

sns.lineplot(data=df_notren, x='YEAR', y='VALUE', ax=ax, hue='PRODUCT', estimator='mean',
 ↪  errorbar=None)
ax.set_xlabel('Año')
ax.set_ylabel('Energía Promedio [GWh]')
ax.set_xlim(2014, 2026)
# ax.set_ylim([0, 6000])
ax.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.tight_layout()
```

3.4. Análisis de datos con aprendizaje supervisado

3.4.1. Importación de librerías y carga del archivo de trabajo

```
# import library
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import seaborn as sns
```

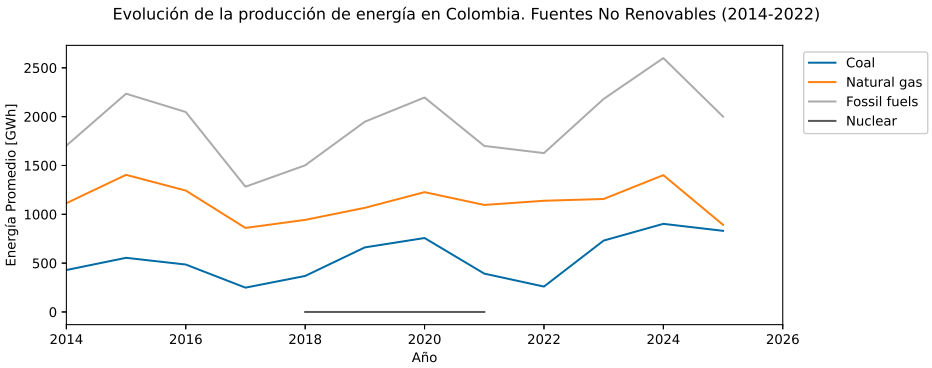

Figura 17: Evolución de la producción de energía en Colombia de fuentes no renovables (2014-2025)

```
import os

# set default directory
os.chdir('/home/lbertel/code/talento_tech/content')

# set style matplotlib
plt.style.use('tableau-colorblind10')

# load data
colombia_df = pd.read_csv('data/processed/colombia_data.csv')
colombia_df.head(10)
```

	YEAR	MONTH	PRODUCT	VALUE
0	2014	1	Hydro	3903.977
1	2014	1	Wind	5.648
2	2014	1	Solar	1.065
3	2014	1	Coal	521.938
4	2014	1	Oil	139.219
5	2014	1	Natural gas	1031.146
6	2014	1	Combustible renewables	99.721
7	2014	1	Net electricity production	5702.714
8	2014	1	Electricity supplied	5555.847
9	2014	1	Distribution losses	536.164

3.4.2. Regresión lineal de la producción eléctrica en Colombia

```
# Group by year and add up total production
df_total = colombia_df.groupby('YEAR')['VALUE'].sum().reset_index()
```

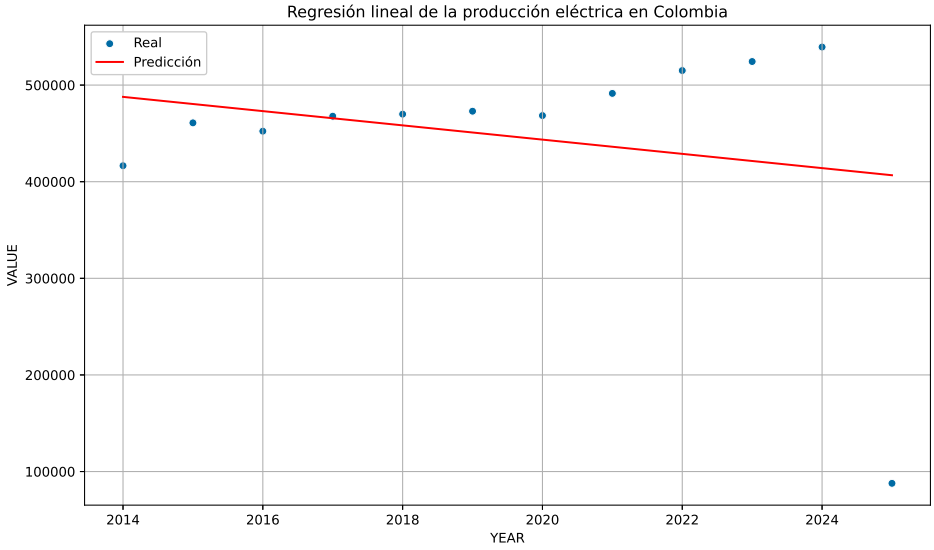

Figura 18: Regresión lineal de la producción eléctrica en Colombia

```
# Create regression model
X = df_total[['YEAR']]
y = df_total['VALUE']
model = LinearRegression().fit(X, y)

# Predictions
df_total['PREDICCION'] = model.predict(X)

# Show results
plt.figure(figsize=(10,6))
sns.scatterplot(data=df_total, x='YEAR', y='VALUE', label='Real')
sns.lineplot(data=df_total, x='YEAR', y='PREDICCION', color='red', label='Predicción')
plt.title('Regresión lineal de la producción eléctrica en Colombia')
plt.grid(True)
plt.legend()
plt.show()
```

3.5. Análisis de datos con aprendizaje no supervisado

3.5.1. Importación de librerías y carga del archivo de trabajo

```
    # import library
    import pandas as pd
    import numpy as np
    from sklearn.linear_model import LinearRegression
    import matplotlib.pyplot as plt
```

```
    import seaborn as sns
    import os

    # set default directory
    os.chdir('/home/lbertel/code/talento_tech/content')

    # set style matplotlib
    plt.style.use('tableau-colorblind10')

    # load data
    colombia_df = pd.read_csv('data/processed/colombia_data.csv')
    colombia_df.head(10)
```

	YEAR	MONTH	PRODUCT	VALUE
0	2014	1	Hydro	3903.977
1	2014	1	Wind	5.648
2	2014	1	Solar	1.065
3	2014	1	Coal	521.938
4	2014	1	Oil	139.219
5	2014	1	Natural gas	1031.146
6	2014	1	Combustible renewables	99.721
7	2014	1	Net electricity production	5702.714
8	2014	1	Electricity supplied	5555.847
9	2014	1	Distribution losses	536.164

3.5.2. Clustering de años según patrón energético (PCA + KMeans)

```
df_pivot = colombia_df.pivot_table(index='YEAR', columns='PRODUCT', values='VALUE',
 ↪  aggfunc='sum').fillna(0)

# 2. data scaling
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df_pivot)

# 3. Reducing dimensions with PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(data_scaled)

# 4. Group by with KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(pca_result)

# 5. show result
plt.figure(figsize=(10, 6))
```

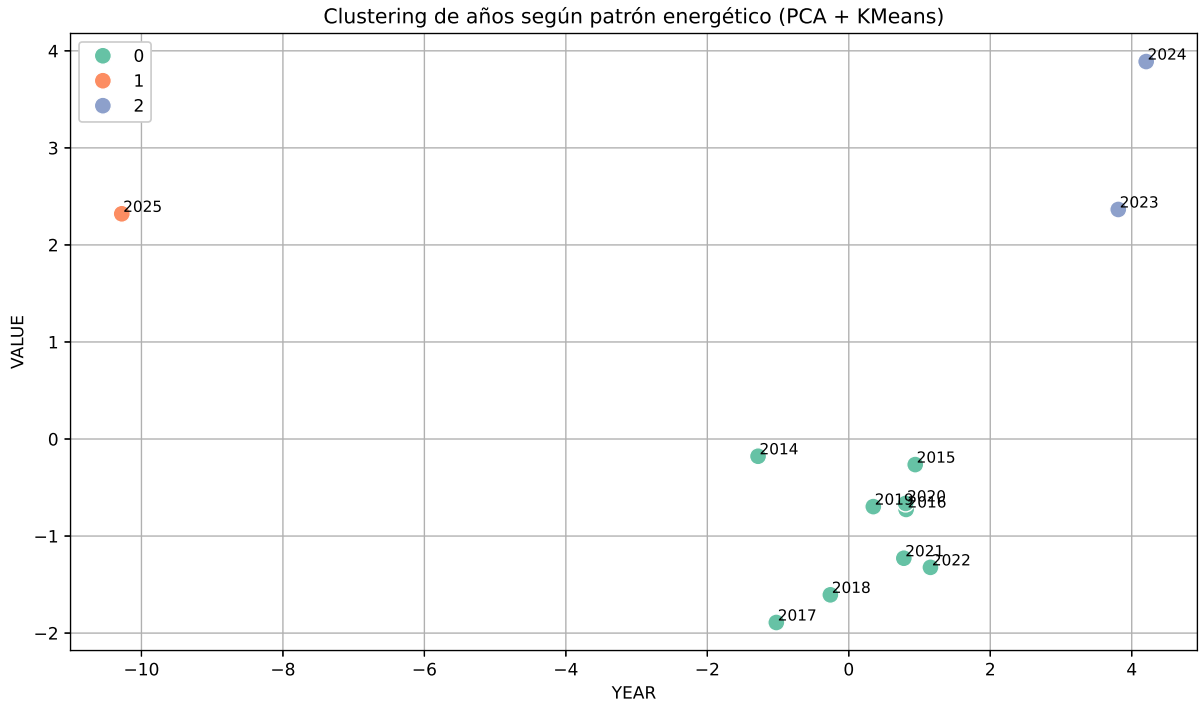

Figura 19: Clustering de años según patrón energético (PCA + KMeans)

```python
sns.scatterplot(x=pca_result[:, 0], y=pca_result[:, 1], hue=clusters, palette='Set2', s=100)

# Añadir etiquetas de años
for i, year in enumerate(df_pivot.index):
plt.text(pca_result[i, 0]+0.02, pca_result[i, 1]+0.02, str(year), fontsize=9)

plt.title('Clustering de años según patrón energético (PCA + KMeans)')
plt.xlabel('YEAR')
plt.ylabel('VALUE')
plt.grid(True)
plt.tight_layout()
plt.show()
```

4. Conclusiones

- La generación hidroeléctrica domina ampliamente entre las fuentes renovables.
- Las fuentes como solar y eólica aún representan un porcentaje muy bajo, aunque con potencial de crecimiento.
- La energía basada en fósiles también tiene un peso relevante, especialmente el gas natural.
- Se observa un crecimiento sostenido de la producción entre 2014 y 2018.

- El análisis completo de 2019 a 2025 permitirá ver el impacto de políticas o proyectos recientes.

Referencias

[García and Molina, 2018] García, J. and Molina, J. M. (2018). *Ciencia de datos: técnicas analíticas y aprendizaje estadístico*.