



# Especificación de requisitos de software

Proyecto: Green Data  
Revisión[2]

## Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
[Fecha]	[Rev]		[Firma o sello]

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña [Nombre]	Fdo. D./Dña [Nombre]

## Contenido

FICHA DEL DOCUMENTO

3

<b>CONTENIDO</b>	<b>4</b>
<b>1 INTRODUCCIÓN</b>	<b>5</b>
1.1 Propósito	6
1.2 Alcance	6
1.3 Personal involucrado	7
1.4 Definiciones, acrónimos y abreviaturas	7
1.5 Referencias	7
1.6 Resumen	7
<b>2 DESCRIPCIÓN GENERAL</b>	<b>8</b>
2.1 Perspectiva del producto	8
2.2 Funcionalidad del producto	8
2.3 Características de los usuarios	8
2.4 Restricciones	8
2.5 Suposiciones y dependencias	8
2.6 Evolución previsible del sistema	8
<b>3 REQUISITOS ESPECÍFICOS</b>	
<b>3.2 Requisitos funcionales</b>	<b>9</b>
3.2.1 Requisito funcional 1	13
3.2.2 Requisito funcional 2	16
3.2.3 Requisito funcional 3	
3.2.4 Requisito funcional	
<b>3.4 Otros requisitos</b>	
<b>4 APÉNDICES</b>	

# 1 Introducción

## Objetivo General

Demostrar relación entre el consumo energético y la emisión de CO2 por país, junto con su impacto en el medio ambiente.

## Finalidad

Servir como guía a las naciones y empresas de todo el mundo para generar conciencia sobre el impacto humano en el medio ambiente Objetivos del estudio:

Objetivos del estudio:

- 1- Analizar consumo energético y emisiones de CO2 de cada país.
- 2- Encontrar los mayores productores de energías renovables
- 4- Mostrar crecimiento en las emisiones de CO2 por nación
- 5- Mostrar países con mayor calidad de aire para mejores condiciones de vida.
- 6 -Verificar qué países alcanzarían una reducción de CO2 para el 2030 según el Acuerdo de París a través de una regresión lineal.
- 7- Demostrar cómo el nivel de CO2 influye en el medio ambiente.
- 8- Generar conciencia sobre la situación ecológica actual y futura.
- 9- Proponer sugerencias para reducir las emisiones de CO2.

## Indicadores de Medición (KPI'S):

1. Índice de generación de combustibles fósiles
2. Intensidad de carbono del consumo
3. Indicador de energía renovables (Proporción de energía renovable y proporción de electricidad baja en carbono)
4. Emisiones CO2 per cápita
5. Índice de autonomía energética (Energía producida/ Energía consumida)
6. Indicador de calidad del aire
7. Variación de temperatura
8. Variación del nivel del mar y masa glacial
9. Consumo o producción en base al PBI

## 1.1 Propósito

- El cambio climático se ha acelerado a niveles sin precedentes como consecuencia de las actividades humanas, siendo una de las mayores responsables la necesidad de energía obtenida a partir de diversas fuentes de combustibles fósiles. El impacto del desarrollo energético en el ambiente y los consumos generados, atraen a las compañías a tomar acción en cómo intervenir en estas problemáticas. Lo cual lleva a mediciones de consumo y generación para intervenir o mejorar dicha generación/consumo.

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para poder demostrar visualmente el consumo energético y generación de CO<sub>2</sub>, tanto de manera geoespacial como particularmente en gráficos. Proponer modelos de como ha sido el consumo a lo largo del tiempo y como será a futuro. Qué intervenciones hacer y cuales son los mejores países y los peores, tanto en el consumo como generación de energías más limpias. Éste será supervisado por el Product Owner.

## 1.2 Alcance

Esa especificación esta dirigida para el desarrollo de funcionalidades sobre diferentes ideas propuestas en base a data sets y para profundizar la automatización de esta información, la cual tiene por objetivo principal en gestionarla para obtener lo siguiente: Alcance:

- Consumo y producción de energía por país
- Plantas y tipos de generación de energía por país
- Emisiones de gases de efecto invernadero globales y por país
- Ranking de consumo energético y generación de energías limpias.
- Predicción de emisiones de CO<sub>2</sub>
- Impacto en el medio ambiente Fuera de Alcance:
- Propuestas o soluciones para revertir la contaminación
- Huella de carbono per cápita
- Relación con el contexto político, económico y social
- Tasa de mortalidad por contaminación del aire
- Otros tipos de segmentación distintos a la nacionalidad, tales como la edad, cultura, etc.

### 1.3 Personal involucrado

Nombre	Julián Mediavilla
Rol	Data Warehouse Architect
Nombre	Diego Esteban Barrios
Rol	Data Engineer
Nombre	Elizabeth Morales
Rol	Machine Learning Engineer
Nombre	Gabriela Huaman Arévalo
Rol	Data Analyst

### 1.4 Definiciones, acrónimos y abreviaturas

[Inserte aquí el texto]

*Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.*

### 1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
[Ref.]	[Título]	[Ruta]	[Fecha]	[Autor]

*Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.*

### 1.6 Resumen

Este documento consta de tres secciones.

En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general, con el fin de conocer lo que se va a realizar, los datos asociados. De manera detallada en cada parte y como se ha trabajado.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el trabajo

## 2 Descripción general

### 2.1 Perspectiva del producto

En el sitio WEB que presentaremos con nuestros alcances y objetivos desarrollados, se permitirá la interacción y visualización de forma rápida y eficaz. Con direccionamiento a los documentos, desarrollo y GitHub.

### 2.2 Funcionalidad del producto

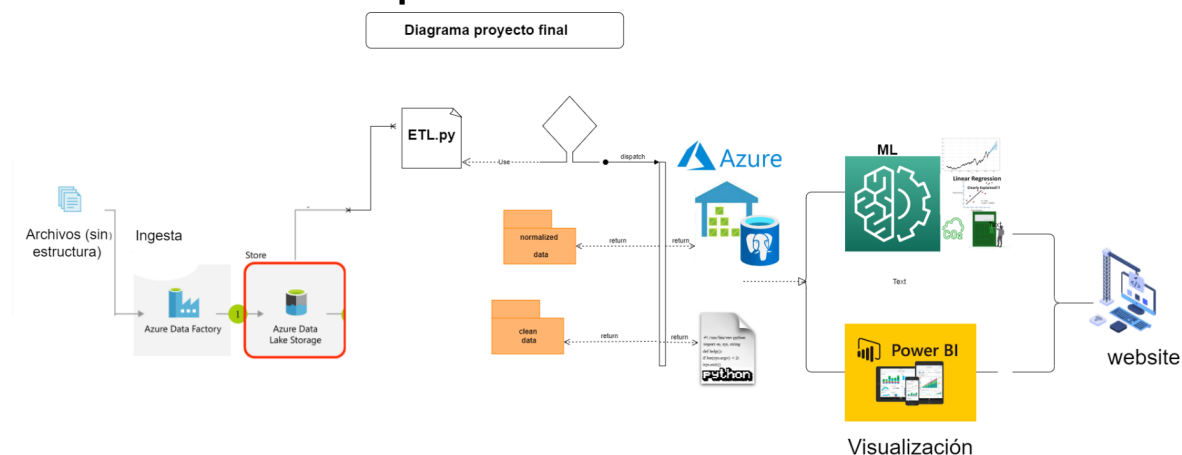


Ilustración Nro 1

### 2.3 Características de los usuarios

Tipo de usuario	Product Owner
Formación	Tecnología

### 2.4 Restricciones

- Interfaz para ser usada con internet.
- Uso de Dominio.
- HTML. Css, Jupyter.
- El servidor debe ser funcional.

### 2.5 Suposiciones y dependencias

Se asume que los requisitos aquí descritos son estables

### 3 Requisitos funcionales

En cuanto a este tipo de requisitos podemos definir dos partes:

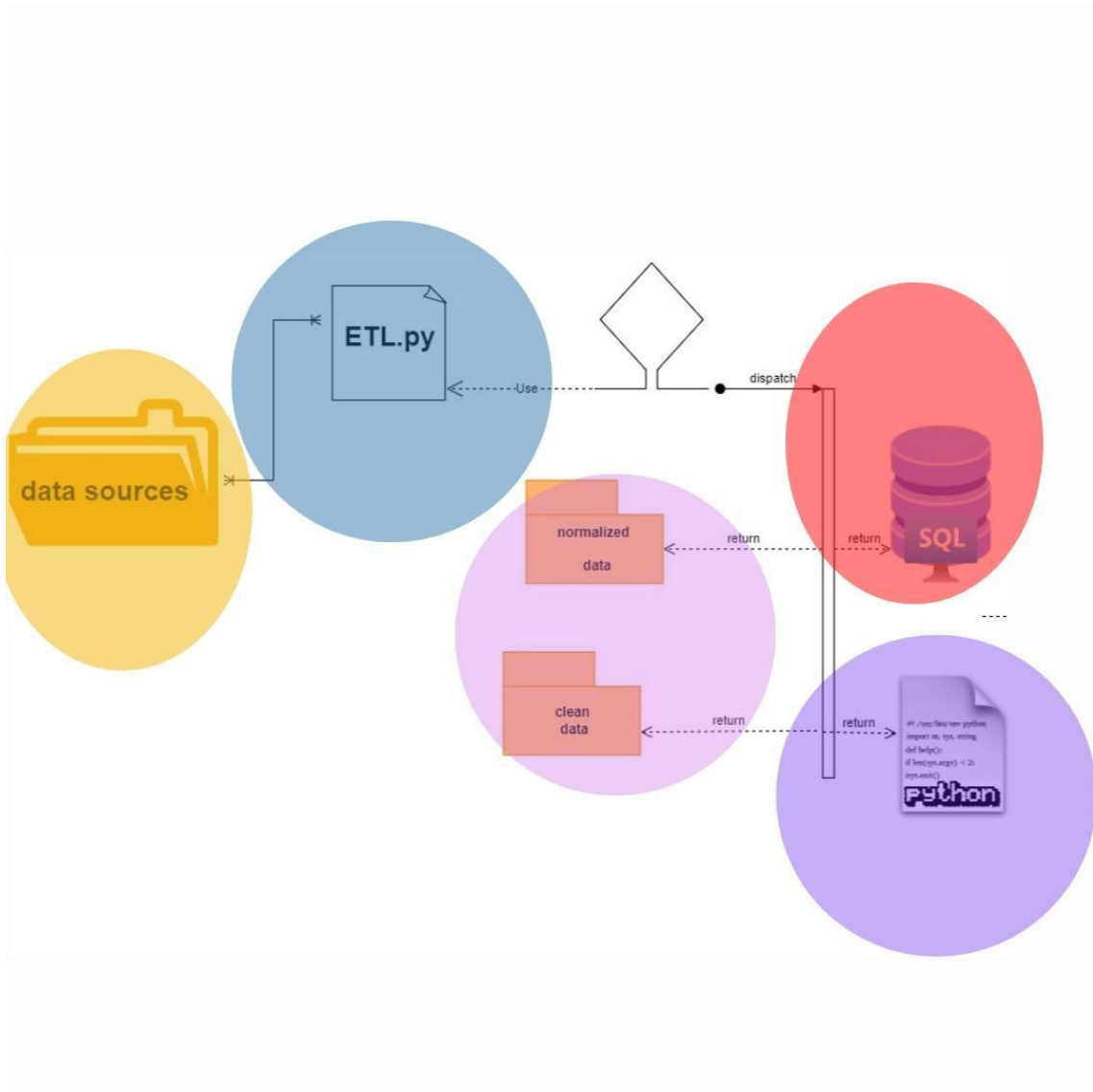


Ilustración Nro. 2

### EDA (Exploratory Data Analysis):





<https://docs.google.com/spreadsheets/d/1n70koKz9wxkRcucVv7bPbKpV19krkr9k/edit#gid=701609892>

## Plataforma en la nube: AZURE

Nos permite la creación de nuestro data lake y la transformación de la data para utilizarla en power bi y python.

## Descripción de requisitos del software

## Recursos

Reciente Favorito

Nombre	Tipo	Última consulta
 pf10	Cuenta de almacenamiento	hace 19 minutos
 DF10	Grupo de recursos	hace 20 minutos
 pfml0867533929	Almacén de claves	hace 13 horas
 pfml	Azure Machine Learning	hace 14 horas
 pfml8412426062	Cuenta de almacenamiento	hace 21 horas
 Azure subscription 1	Suscripción	hace 21 horas
 pf10sql	Base de datos SQL	hace 21 horas
 data	Centro de Private Link de Synapse	hace 21 horas
 PF10	Grupo de recursos	hace 21 horas
 pf10	Base de datos SQL	hace 21 horas
 pf10	SQL Server	hace 21 horas
 cs2100320021d94c10b	Cuenta de almacenamiento	hace 23 horas


























Nombre	Modificado	Nivel de acceso	Estado del archivo	Tipo de blob	Tamaño
<input type="checkbox"/>  airpollution.csv	17/8/2022, 10:19:12	Frecuente (inferido)		Blob en bloques	5.04 MiB
<input type="checkbox"/>  API_EN.ATM.CO2EKT_DS2_en_csv_v2_4354173.csv	18/8/2022, 7:25:16	Frecuente (inferido)		Blob en bloques	147 KiB
<input type="checkbox"/>  brent-daily.csv	17/8/2022, 11:21:18	Frecuente (inferido)		Blob en bloques	156.69 KiB
<input type="checkbox"/>  brent-monthly.csv	17/8/2022, 11:21:15	Frecuente (inferido)		Blob en bloques	7.46 KiB
<input type="checkbox"/>  brent-weekly.csv	17/8/2022, 11:21:16	Frecuente (inferido)		Blob en bloques	32.33 KiB
<input type="checkbox"/>  brent-year.csv	17/8/2022, 11:21:15	Frecuente (inferido)		Blob en bloques	643 B
<input type="checkbox"/>  conflictos internacionales.csv	17/8/2022, 12:10:36	Frecuente (inferido)		Blob en bloques	41.12 MiB
<input type="checkbox"/>  data.json	17/8/2022, 13:41:28	Frecuente (inferido)		Blob en bloques	30.89 KiB
<input type="checkbox"/>  energyco2.csv	17/8/2022, 10:15:20	Frecuente (inferido)		Blob en bloques	5.59 MiB
<input type="checkbox"/>  fossil-fuel-co2-emissions-by-nation.csv	17/8/2022, 11:25:29	Frecuente (inferido)		Blob en bloques	740.9 KiB
<input type="checkbox"/>  glaciers.csv	17/8/2022, 11:11:30	Frecuente (inferido)		Blob en bloques	1.1 KiB
<input type="checkbox"/>  global_power_plant_database.csv	17/8/2022, 10:15:26	Frecuente (inferido)		Blob en bloques	11.42 MiB
<input type="checkbox"/>  Guerras.csv	17/8/2022, 11:45:22	Frecuente (inferido)		Blob en bloques	554.47 KiB
<input type="checkbox"/>  Metadata_Country_API_EN.ATM.CO2EKT_DS2_en_csv_v2...	18/8/2022, 7:24:57	Frecuente (inferido)		Blob en bloques	57.61 KiB
<input type="checkbox"/>  Metadata_Indicator_API_EN.ATM.CO2EKT_DS2_en_csv_v...	18/8/2022, 7:24:57	Frecuente (inferido)		Blob en bloques	474 B
<input type="checkbox"/>  Metadata_Indicator_API_EN.ATM.CO2EKT_DS2_en_csv_v...	18/8/2022, 7:24:57	Frecuente (inferido)		Blob en bloques	474 B
<input type="checkbox"/>  Muertes por contaminación del aire 2016.csv	17/8/2022, 11:12:01	Frecuente (inferido)		Blob en bloques	878.05 KiB
<input type="checkbox"/>  natural gas - daily.csv	17/8/2022, 11:25:07	Frecuente (inferido)		Blob en bloques	335.74 KiB
<input type="checkbox"/>  natural gas - monthly.csv	17/8/2022, 11:25:02	Frecuente (inferido)		Blob en bloques	15.29 KiB
<input type="checkbox"/>  nivel del mar.csv	17/8/2022, 11:29:52	Frecuente (inferido)		Blob en bloques	5.79 KiB
<input type="checkbox"/>  owid-energy-consumption-source.csv	17/8/2022, 10:15:17	Frecuente (inferido)		Blob en bloques	8.1 MiB
<input type="checkbox"/>  owid-energy-datadictionary.csv	17/8/2022, 10:15:05	Frecuente (inferido)		Blob en bloques	23.79 KiB
<input type="checkbox"/>  poblacion.csv	17/8/2022, 11:22:15	Frecuente (inferido)		Blob en bloques	476.55 KiB
<input type="checkbox"/>  temperatura - annual.csv	17/8/2022, 11:27:10	Frecuente (inferido)		Blob en bloques	4.84 KiB
<input type="checkbox"/>  temperatura - monthly.csv	17/8/2022, 11:27:11	Frecuente (inferido)		Blob en bloques	67.41 KiB
<input type="checkbox"/>  wti-daily.csv	17/8/2022, 11:21:19	Frecuente (inferido)		Blob en bloques	161.47 KiB
<input type="checkbox"/>  wti-monthly.csv	17/8/2022, 11:21:15	Frecuente (inferido)		Blob en bloques	7.7 KiB
<input type="checkbox"/>  wti-weekly.csv	17/8/2022, 11:21:15	Frecuente (inferido)		Blob en bloques	33.47 KiB
<input type="checkbox"/>  wti-year.csv	17/8/2022, 11:21:15	Frecuente (inferido)		Blob en bloques	657 B

Ilustración Nro. 3

## ETL(Extract Transform and load )

En cuanto a la automatización de los se decidió optar por Python:

### 3.1.1 Requisito data base en Azure:

En este caso se ha utilizado pgAdmin y Python, para el caso de la conexión con azure se aplico lo siguiente.

#### Base de datos PostgreSQL y conexión:

##### Conexión con Azure y permisos:

En Azure debemos agregar nuestras IP's, para permitir que el firewall permita la conexión.

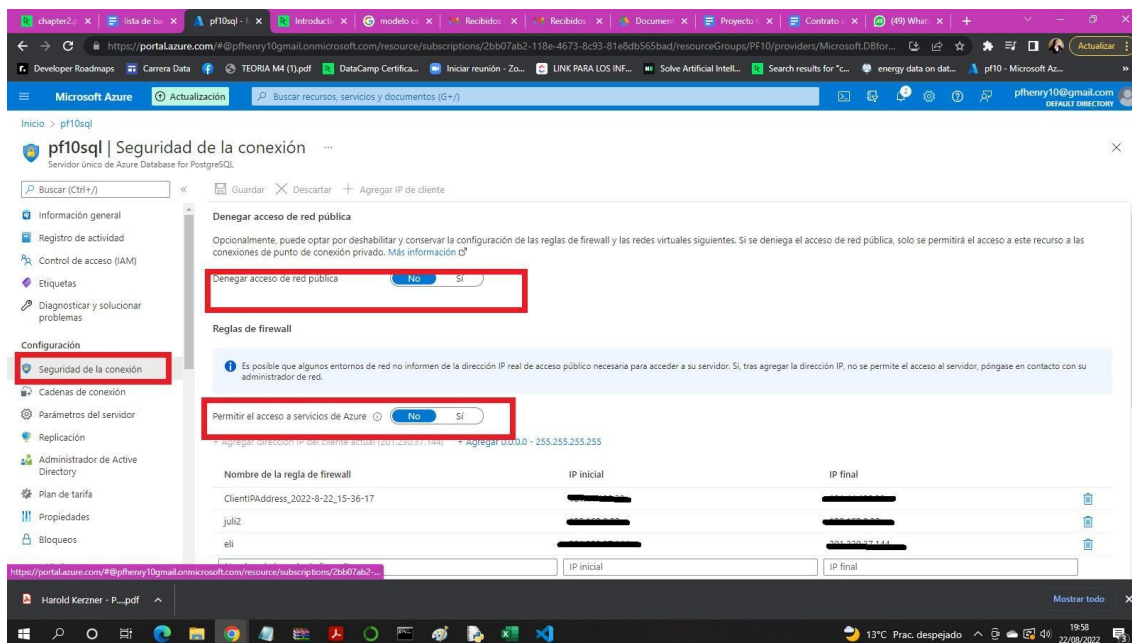


Ilustración Nro. 4

En Azure debemos agregar nuestras IP's, para permitir que el firewall no interrumpa la conexión entre Azure y nuestra computadora..

## Configuración de SSL

**i** La aplicación de conexiones SSL en su servidor puede requerir una configuración adicional para las aplicaciones que se conecten a él. [Más información](#)

Aplicar conexión SSL

**HABILITADO** DESHABILITADO

Debemos permitir también que se pueda acceder a los servicios de Azure.

Permitir el acceso a servicios de Azure ☐ No ☒ Sí

+ Agregar dirección IP del cliente actual (201.230.37.144) + Agregar 0.0.0.0 - 255.255.255.255

Nombre de la regla de firewall	IP inicial	IP final
ClientIPAddress_2022-8-22_15-36-17		181.44.129.20
juli2		192.168.0.32
eli		201.230.37.144
Nombre de la regla de firewall	IP inicial	IP final

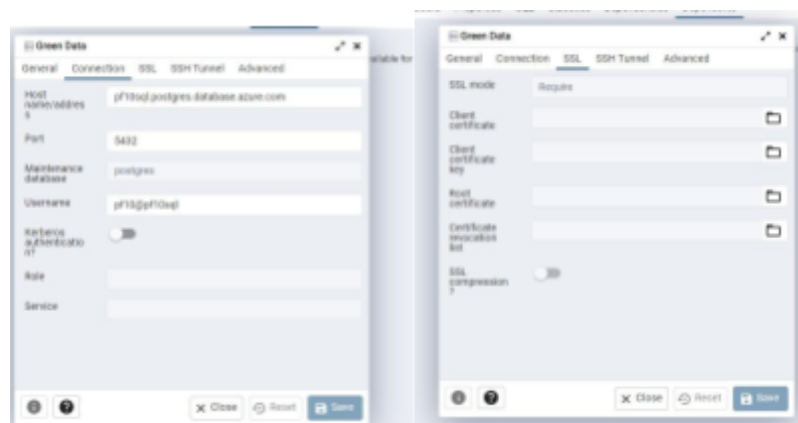
Reglas de red virtual + Agregar una red virtual existente + Crear nueva red virtual

Nombre de regla	Red virtual	Subred	Intervalo de direcci...	Estado del punto de...	Grupo de recursos	Id. de suscripción	Estado
No hay ningún resultado.							

## Ilustración Nro. 5

### Conexión con pgAdmin:

Si bien la información no necesita pgAdmin, facilita la visualización de la data desde una herramienta Desktop conectada en Azure. Como se muestra, esos son los pasos para la conexión entre pgAdmin y Azure.



## Ilustración Nro. 6



### Conexión Python con Azure:

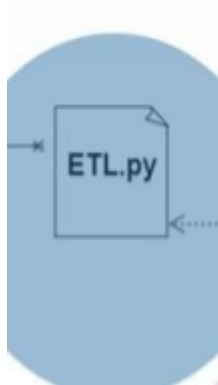
En el caso de la conexión con Python debemos incluir librerías para poder hacer la conexión con la nube de nuestra base de datos postgres. La función de try y except nos permite conectar y crear la base de datos "greendata" o retornar un mensaje de error que no ha logrado generar la conexión.

**Librerías:** *Pandas, psycopg2, os, sqlalchemy, csv.*

```
import pandas as pd
import psycopg2
import os
from psycopg2 import Error
import sqlalchemy
import csv
database = 'postgres'
user = 'pf10@pf10sql'
password = '!123henryfinal'
host = 'pf10sql.postgres.database.azure.com'
port = '5432'
try:
    #Conexión al servidor
    conn_string = "host=%s dbname=%s user=%s password=%s" % (host, database, user, password)
    conn = psycopg2.connect(conn_string)
    conn.autocommit = True
    cursor = conn.cursor()
    #Crear nuestra base de datos
    cursor.execute("CREATE DATABASE greendata;")
    print("Se ha creado la base de datos greendata")
    conn.commit()
    conn.close()
except (Exception, psycopg2.Error) as error:
    print(error)
```

**Ilustración Nro. 7**

### 3.1.2 Requisito ETL py:



Extracción de la información:

Los archivos que se utilizaron fueron los csv, para trabajar con pandas. Es importante recalcar que en la Ilustración Nro 8 se muestra que se va llamando por columnas para modificar los valores de índices, nombres por medio de la estandarización de las columnas y nombres de los Dataframes. Las tablas han sido definidas según los objetivos, alcances y KPI's que se han definido.

Se definió como tabla de hechos y tabla de dimensiones:

#### Dimensiones:

- Países
- Región
- Plantas generadoras
- Tipo de planta
- Calendario

#### Hechos:

- Hechos consumo mundial
- Hechos consumo por país
- Hechos de producción por país.
- Hechos de producción mundial.
- Hechos de emisión por país de CO2
- Hechos de emisión mundial de CO2



```
#ETL
# Calendario
años = range(1980,2021)
calendario = pd.DataFrame({"Años": años})
# Localización
loc_global = pd.DataFrame(pd.read_csv(archivo3))
mundo = pd.DataFrame(pd.read_csv(archivo7))
loc_global = loc_global[['country', 'country_long']].drop_duplicates().reset_index(drop=True)
loc_global.rename(columns={'country_long': 'name'}, inplace=True, errors='raise')
loc_final = pd.merge(loc_global, mundo, on='name', how='outer')
loc_final = loc_final[['country', 'name', 'region']]
loc_final.rename(columns={"country": "Codigo", "name": "Pais", "region": "Region"})
# Tablas producción mundo y países
produccion = pd.DataFrame(pd.read_csv(archivo5))
produccion = produccion[['year', 'country', 'coal_prod_change_pct', 'coal_production', 'gas_prod_change_pct', 'gas_production', 'oil_prod_change_pct', 'oil_production']]
produccion = produccion[produccion['year'] >= 1980]
produccion = produccion.rename(columns={'year': 'Año', 'country': 'Pais', 'coal_production': 'Prod_carbon', 'coal_prod_change_pct': 'Prod_carb_%_año'})
lista_world = ["World"]
lista_borrar_paises = []
for x in produccion['Pais']:
    if x.endswith(("BP"), ("Shift"), ("EIA"), ("27"), ("Ember"), "Islands", "countries"),):
        lista_borrar_paises.append(x)
    lista_aux = ["Africa", "Antarctica", "Asia", "Europe", "Oceania", "USSR", "World", "North America", "URSS"]
    lista_borrar_paises = lista_borrar_paises + lista_aux
produccion_mundo = produccion[produccion.Pais.isin(lista_world)]
produccion_paises = produccion[~produccion.Pais.isin(lista_borrar_paises)]
```

Ilustración Nro. 8

```
# Tablas generación mundo y países
generacion = pd.DataFrame(pd.read_csv(archivo5))
generacion = generacion[generacion['year'] >= 1980]
generacion.rename(columns={"country": "Pais", "year": "Año", "iso_code": "Id_Pais", "population": "Poblacion", "gdp": "PBI", "biofuel_electricity": "Gen_Elec"}, inplace=True)
generacion_mundo = generacion[generacion.Pais.isin(lista_world)]
generacion_paises = generacion[~generacion.Pais.isin(lista_borrar_paises)]
# Tabla misiones de CO2 por país
emision_co2 = pd.DataFrame(pd.read_csv(archivo1))
emision_co2_pais = emision_co2[emision_co2['Country'] != 'World']
emision_co2_pais = emision_co2_pais[emision_co2_pais['Country'].str.contains("Africa|Asia|Oceania|America|Europe|EU28|G20|G7|countries|Zone|EIA|Shift")]
emision_co2_pais_filtrado = emision_co2_pais[['Year', 'Country', 'CO2_emission']]
emision_co2_pais_filtrado = emision_co2_pais_filtrado.groupby(['Year', 'Country'], as_index=False).sum()
energias = pd.DataFrame(pd.read_csv(archivo5))
energias_paises = energias[energias['country'] != 'World']
energias_paises_filtrado = energias_paises[energias_paises['country'].str.contains("Africa|Asia|Oceania|America|Europe|EU28|G20|G7|countries|Zone|EIA|Shift")]
energias_paises_filtrado = energias_paises_filtrado[energias_paises_filtrado['year'] >= 1980]
energias_paises_filtrado.reset_index(inplace=True, drop=True)
energias_paises_filtrado = energias_paises_filtrado[['year', 'country', 'carbon_intensity_elec']]
energias_paises_filtrado.rename(columns={'year': 'Year', 'country': 'Country'}, inplace=True)
emision_pais = emision_co2_pais_filtrado.merge(energias_paises_filtrado, how='inner', on=['Year', 'Country'])
emision_pais.rename(columns={'Year': 'Año', 'Country': 'Pais', 'CO2_emission': 'Emision_CO2', 'carbon_intensity_elec': 'Intensidad_carbono'}, axis=1, inplace=True)
# Tabla mundial: creación de columnas
# Emisión CO2 mundial
emision_co2_mundial = emision_co2[emision_co2['Country'] == 'World']
emision_co2_mundial = emision_co2_mundial.groupby('Year', as_index=False).sum()
emision_co2_mundial_final = emision_co2_mundial[['Year', 'CO2_emission']]
# Variación masa glaciares
glaciares = pd.DataFrame(pd.read_csv(archivo2))
glaciares = glaciares[glaciares['Year'] >= 1980]
glaciares.drop(columns=['Number of observations'], inplace=True)
```

Ilustración Nro. 9

Los Dataframes que se han creado han sido los siguientes:

- Calendario
- Tablas producción mundo y países
- Tablas consumos mundo y países
- Tablas generación mundo y países
- Tabla misiones de CO2 por país
- Tabla mundial: creación de columnas
- Emisión CO2 mundial
- Variación masa glaciares
- Variación masa glaciares
- Intensidad de emisiones
- Variación de temperatura



- Creación tabla mundial
- Tabla plantas de energía
- Tabla tipo de planta
- Tabla calidad del aire



#### Estandarización de la data:

Para los nombres se ha decidido lo siguiente:

Columnas: Mayúscula en la primera letra y después minúscula

Ejemplo: Year -> Año

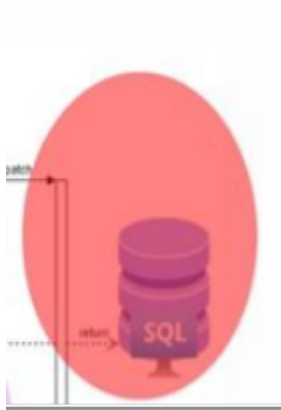
Palabras: Todas deben estar en español

Ejemplo: country' -> 'País'

Dos palabras: El uso de subguión debe incluirse para unir las:

Ejemplo: Id\_País

## Conversión de Dataframe y carga de la información como tablas a la base de datos de pgAdmin:



Carga de información a la base de datos:

```
#Conexión a la base de datos y carga de tablas
try:
    conn_string = "host=%s dbname=%s user=%s password=%s" % (host, database, user, password)
    conn = psycopg2.connect(conn_string)
    conn.autocommit = True
    cursor = conn.cursor()
    engine = sqlalchemy.create_engine('postgresql://pf10@pf10sql:123henryfinal@pf10sql.postgres.database.azure.com/greendata')
    calendario.to_sql('calendario', con=engine, if_exists='replace', index=False)
    loc_final.to_sql('localizacion', con=engine, if_exists='replace', index=False)
    produccion_mundo.to_sql("prod_mundial", con=engine, if_exists='replace', index=False)
    produccion_paises.to_sql("prod_pais", con=engine, if_exists='replace', index=False)
    consumos_mundo.to_sql("cons_mundial", con=engine, if_exists='replace', index=False)
    consumos_paises.to_sql("cons_pais", con=engine, if_exists='replace', index=False)
    generacion_mundo.to_sql("gen_mundial", con=engine, if_exists='replace', index=False)
    generacion_paises.to_sql("gen_pais", con=engine, if_exists='replace', index=False)
    emision_pais.to_sql("emision_pais", con=engine, if_exists='replace', index=False)
    emision_mundial.to_sql("emision_mundial", con=engine, if_exists='replace', index=False)
    planta_energia.to_sql("planta_energia", con=engine, if_exists='replace', index=False)
    tipo_planta.to_sql("tipo_planta", con=engine, if_exists='replace', index=False)
    calidad_aire.to_sql("calidad_aire", con=engine, if_exists='replace', index=False)
    conn.commit()
    conn.close()
except (Exception, psycopg2.Error) as error:
    print(error)
```

### Ilustración Nro 10

En esta parte del código cada uno de los Dataframes han sido convertidos a un tabla en la base de datos de pgAdmin. Por medio de la librería **sqlalchemy**, luego la data pasa a ser subida a la base de datos y se cierra la información. En caso que no se realice la acción aparece un mensaje de error.