PECIER

Comité Nacional Peruano de la CIER

Curso Virtual

Python para el Análisis de Datos y la Automatización en el Sector Eléctrico

27, 29 y 31 de Octubre, 03, 05, 07, 10 y 12 de Noviembre 2025.

MARVIN COTO – FACILITADOR
E-mail: mcotoj@gmail.com

PECIER

Comité Nacional Peruano de la CIER

SESIÓN 6
Parte 1

PECIER
Comité Nacional Peruano de la CIER

Introducción a Pandapower

Veremos qué es Pandapower y para qué sirve en el análisis de redes eléctricas.

¿Qué es Pandapower?

PECIER
Comité Nacional Peruano de la CIER

Pandapower es una biblioteca de Python para análisis de sistemas eléctricos.

Se basa en pandas para estructuras de datos y en numba para acelerar cálculos numéricos, y permite modelar redes con un enfoque orientado a objetos y altamente intuitivo.

PECIER
Comité Nacional Peruano de la CIER

¿Por qué usar Pandapower?

- Automatizar estudios eléctricos comunes, como flujos de potencia.
- Permite integración con otras bibliotecas (como scikit-learn (IA), matplotlib, etc.).
- Ideal para prototipado y simulaciones.

Instalación de Pandapower

PECIER
Comité Nacional Peruano de la CIER

Antes de poder usar Pandapower, debemos tener un entorno de Python funcionando.

Se recomienda crear un entorno virtual para mantener los paquetes organizados, o en su defecto utilizar Colab.

Instalación de Pandapower

PECIER
Comité Nacional Peruano de la CIER

En caso de que se desee activar un entorno virtual:
Los entornos virtuales permiten tener diferentes versiones de Python y bibliotecas en la misma ocmputadora, para evitar conflictos entre ellas.
La forma de realizarlo es:

```
python -m venv pandapower_env
source pandapower_env/bin/activate # Linux/Mac
# pandapower_env\Scripts\activate # Windows

pip install pandapower
```

```
!pip install psycopg2-binary
!pip install pandapower
```

PECIER
Comité Nacional Peruano de la CIER

Bibliotecas esenciales

Explicación

Estas bibliotecas serán nuestras herramientas principales:

```
import pandapower as pp             # Biblioteca básica para modelar y simular redes
import pandapower.networks as pn    # Redes de prueba integradas
import pandas as pd                 # ¡Pandas!
import matplotlib.pyplot as plt     # ¡Visualización!
```

PECIER
Comité Nacional Peruano de la CIER

Bibliotecas esenciales

Explicación
Estas bibliotecas serán nuestras herramientas principales:

```
from pandapower.networks.create_examples import example_simple

net = example_simple()

print(net)

!pip install python-igraph # Visualiza grafos y redes

!pip install igraph
```

PECIER
Comité Nacional Peruano de la CIER

Crear una Red Eléctrica Básica

Explicación paso a paso

Vamos a crear una red con:

- Dos barras (una Slack, una PQ)
- Una carga
- Una línea
- Un generador externo

PECIER
Comité Nacional Peruano de la CIER

Crear una Red Eléctrica Básica

Código

```
net = pp.create_empty_network()  # Crear red vacía

bus1 = pp.create_bus(net, vn_kv=20., name="Barra Slack")
bus2 = pp.create_bus(net, vn_kv=20., name="Barra Carga")

pp.create_ext_grid(net, bus=bus1, vm_pu=1.02, name="Conexión Red")  # Slack
pp.create_load(net, bus=bus2, p_mw=5, q_mvar=2, name="Carga")
pp.create_line(net, from_bus=bus1, to_bus=bus2, length_km=10,
               std_type="NAYY 4x50 SE", name="Línea 1")

pp.runpp(net)  # Ejecuta flujo de potencia

print(net.res_bus)
print(net.res_line)
```

PECIER
Comité Nacional Peruano de la CIER

Visualización de la Red

Podemos visualizar la red de dos formas:

```
pp.plotting.simple_plot(net) # Vista básica (matplotlib)
pp.plotting.simple_plotly(net) # Vista interactiva (plotly)
```

`simple_plotly()` requiere `plotly`. Si no está instalado, ejecuta `pip install plotly`

PECIER
Comité Nacional Peruano de la CIER

Redes de ejemplo

Una forma de visualizar las posibilidades de PandaPower es explorar las redes de ejemplo que ya están programadas.

```
import pandapower.networks as pn
import pandapower.plotting.plotly as pf_plotly

net = pn.mv_oberrhein()  # Red de ejemplo

# Visualización
pf_plotly.simple_plotly(net)
```

```
# Lista básica de cantidad de elementos
for element in ['bus', 'line', 'trafo', 'load', 'sgen', 'gen', 'ext_grid', 'switch', 'shunt']:
    if element in net and not net[element].empty:
        print(f"{element.capitalize()}: {len(net[element])} elementos")
```

PECIER
Comité Nacional Peruano de la CIER

Redes de ejemplo

Una forma de visualizar las posibilidades de PandaPower es explorar las redes de ejemplo que ya están programadas.

```
print(net.bus.head())
```

```
# Flujo de carga
# Porcentaje de carga de cada línea

import pandapower as pp

pp.runpp(net) # Aquí estamos ejecutando el flujo de carga
print(net.res_bus[["vm_pu", "va_degree"]])
print(net.res_line[["loading_percent"]])
```

PECIER
Comité Nacional Peruano de la CIER

Redes de ejemplo

Aspectos que se pueden explorar: ¿Hay alguna línea con carga mayor al 80%?

```
sobrecargadas=net.res_line[net.res_line["loading_percent"] > 80]
print(f"{len(sobrecargadas)}")

# ¿Cuáles son? Es decir, cuáles son los índices que cumplen.
net.res_line.index[net.res_line["loading_percent"] > 80].tolist()
```

PECIER
Comité Nacional Peruano de la CIER

Redes de ejemplo

Analizar pérdidas técnicas (por efecto Joule)

```
pp.runpp(net)
net.res_line["p_loss_mw"] = net.res_line["pl_mw"] + net.res_line["ql_mvar"] * 0  # si solo interesa
potencia activa
print(net.res_line[["p_loss_mw"]])
print("Total pérdidas:", net.res_line["p_loss_mw"].sum(), "MW")
```

PECIER
Comité Nacional Peruano de la CIER

Redes de ejemplo

Agregar una carga:

```
pp.create_load(net, bus=7, p_mw=0.5, q_mvar=0.2)
pp.runpp(net) # Vuelve a ejecutar flujo de carga para ver reflejado el cambio

print(net.res_line[["loading_percent"]])


# Existen transformadores cargados a más del 100%

trafos_sobrecargados = net.res_trafo[net.res_trafo["loading_percent"] > 100]
print(trafos_sobrecargados)
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia6_pandopower1

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

SESIÓN 6
Parte 2

Creación de Redes Eléctricas Simples con Pandapower

PECIER
Comité Nacional Peruano de la CIER

Objetivos de la Lección

1. Entender la estructura de datos de Pandapower para modelado de redes
2. Aprender a crear componentes básicos de sistemas de potencia
3. Construir redes eléctricas desde cero
4. Realizar simulaciones básicas de flujo de potencia
5. Visualizar y analizar resultados

Estructura Básica de una Red en Pandapower

PECIER
Comité Nacional Peruano de la CIER

Componentes principales:

```
!pip install pandapower['all’]

import pandapower as pp
net = pp.create_empty_network()  # Red vacía

# Barras (nodos)
bus1 = pp.create_bus(net, vn_kv=20., name="Barra 1")
bus2 = pp.create_bus(net, vn_kv=0.4, name="Barra 2")

# Generación (Slack bus)
pp.create_ext_grid(net, bus=bus1, vm_pu=1.0, name="Red Externa")
```

Estructura Básica de una Red en Pandapower

PECIER
Comité Nacional Peruano de la CIER

Componentes principales:

```
# Cargas
pp.create_load(net, bus=bus2, p_mw=0.1, q_mvar=0.05, name="Carga Residencial")

# Líneas
pp.create_line(net, from_bus=bus1, to_bus=bus2, length_km=0.5,
               std_type="NAYY 4x50 SE", name="Línea 1")

# Transformadores
pp.create_transformer(net, hv_bus=bus1, lv_bus=bus2,
                      std_type="0.63 MVA 20/0.4 kV", name="Trafo 1")
```

Componentes Básicos y su Creación

PECIER
Comité Nacional Peruano de la CIER

Barras (Buses)

Las barras son los nodos de la red donde se conectan los elementos.

```
# Creación de barras con diferentes niveles de voltaje
bus_slack = pp.create_bus(net, vn_kv=115, name="Slack Bus", type="b")
bus_pv = pp.create_bus(net, vn_kv=23, name="PV Bus", type="b")
bus_pq = pp.create_bus(net, vn_kv=0.48, name="PQ Bus", type="b")

# Visualizar barras creadas
print(net.bus)
```

Componentes Básicos y su Creación

PECIER
Comité Nacional Peruano de la CIER

Slack Bus (Barra de Referencia)

```
pp.create_ext_grid(net, bus=bus_slack, vm_pu=1.02,
name="Subestación Principal",
                   max_p_mw=100, min_p_mw=0,
max_q_mvar=50, min_q_mvar=-50)
print(net.ext_grid)
```

Componentes Básicos y su Creación

PECIER
Comité Nacional Peruano de la CIER

Cargas (PQ)

```
# Diferentes tipos de carga
pp.create_load(net, bus=bus_pq, p_mw=1.5, q_mvar=0.75, name="Residencial",
               const_z_percent=0, const_i_percent=100)  # Corriente constante

pp.create_load(net, bus=bus_pq, p_mw=5.0, q_mvar=2.5, name="Comercial",
               const_z_percent=100, const_i_percent=0)  # Impedancia constante

# Carga con factor de potencia
pp.create_load_from_cosphi(net, bus=bus_pq, sn_mva=3.0, cos_phi=0.9,
                           name="Industrial", mode="underexcited")
```

Componentes Básicos y su Creación

PECIER
Comité Nacional Peruano de la CIER

Líneas de Transmisión/Distribución

```
# Creación de líneas con tipos estándar
line1 = pp.create_line(net, from_bus=bus_slack, to_bus=bus_pv, length_km=15,
                       std_type="149-AL1/24-ST1A 10.0", name="Línea Tronsmisión")

# Línea con parámetros personalizados
line2 = pp.create_line_from_parameters(
    net, from_bus=bus_pv, to_bus=bus_pq, length_km=0.2, name="Línea Distribución",
    r_ohm_per_km=0.162, x_ohm_per_km=0.083, c_nf_per_km=235, max_i_ka=0.142)

print(net.line)
```

Componentes Básicos y su Creación

PECIER
Comité Nacional Peruano de la CIER

Transformadores

```
# Transformador estándar
trafo1 = pp.create_transformer(
    net, hv_bus=bus_slack, lv_bus=bus_pv, name="Trafo Principal",
    std_type="25 MVA 110/20 kV")

# Transformador con tap variable
trafo2 = pp.create_transformer_from_parameters(
    net, hv_bus=bus_pv, lv_bus=bus_pq, name="Trafo Distribución",
    sn_mva=0.63, vn_hv_kv=23, vn_lv_kv=0.48, vkr_percent=4.5,
    vk_percent=12, pfe_kw=1.5, i0_percent=0.3, tap_pos=0,
    tap_neutral=0, tap_min=-2, tap_max=2, tap_step_percent=1.5)
```

Generación Distribuida

PECIER
Comité Nacional Peruano de la CIER

Generadores Síncronos

```
pp.create_gen(net, bus=bus_pv, p_mw=2.0, vm_pu=1.02, name="Generador Diésel",
              min_p_mw=0, max_p_mw=3, min_q_mvar=-1, max_q_mvar=1)
```

Generación Distribuida

PECIER
Comité Nacional Peruano de la CIER

Fuentes Renovables (PV)

```
pp.create_sgen(net, bus=bus_pq, p_mw=0.5, q_mvar=0, name="Solar PV",
               type="PV", k=1.2, rx=0.4, current_source=True)
```

Inspección de la Red Creada

PECIER
Comité Nacional Peruano de la CIER

```
import pandas as pd
# Resumen de la red
print(f"La red contiene:\n"
      f"- {len(net.bus)} barras\n"
      f"- {len(net.line)} líneas\n"
      f"- {len(net.trafo)} transformadores\n"
      f"- {len(net.load)} cargas\n"
      f"- {len(net.ext_grid)} barras slack\n"
      f"- {len(net.gen)} generadores")

# Mostrar todos los elementos
print("\nComponentes de la red:")
for element in net.keys():
    if isinstance(net[element], pd.DataFrame) and not net[element].empty:
        print(f"{element}: {len(net[element])} elementos")
```

Inspección de la Red Creada

PECIER
Comité Nacional Peruano de la CIER

```
import pandas as pd
# Resumen de la red
print(f"La red contiene:\n"
      f"- {len(net.bus)} barras\n"
      f"- {len(net.line)} líneas\n"
      f"- {len(net.trafo)} transformadores\n"
      f"- {len(net.load)} cargas\n"
      f"- {len(net.ext_grid)} barras slack\n"
      f"- {len(net.gen)} generadores")

# Mostrar todos los elementos
print("\nComponentes de la red:")
for element in net.keys():
    if isinstance(net[element], pd.DataFrame) and not net[element].empty:
        print(f"{element}: {len(net[element])} elementos")
```

Visualización de la Red

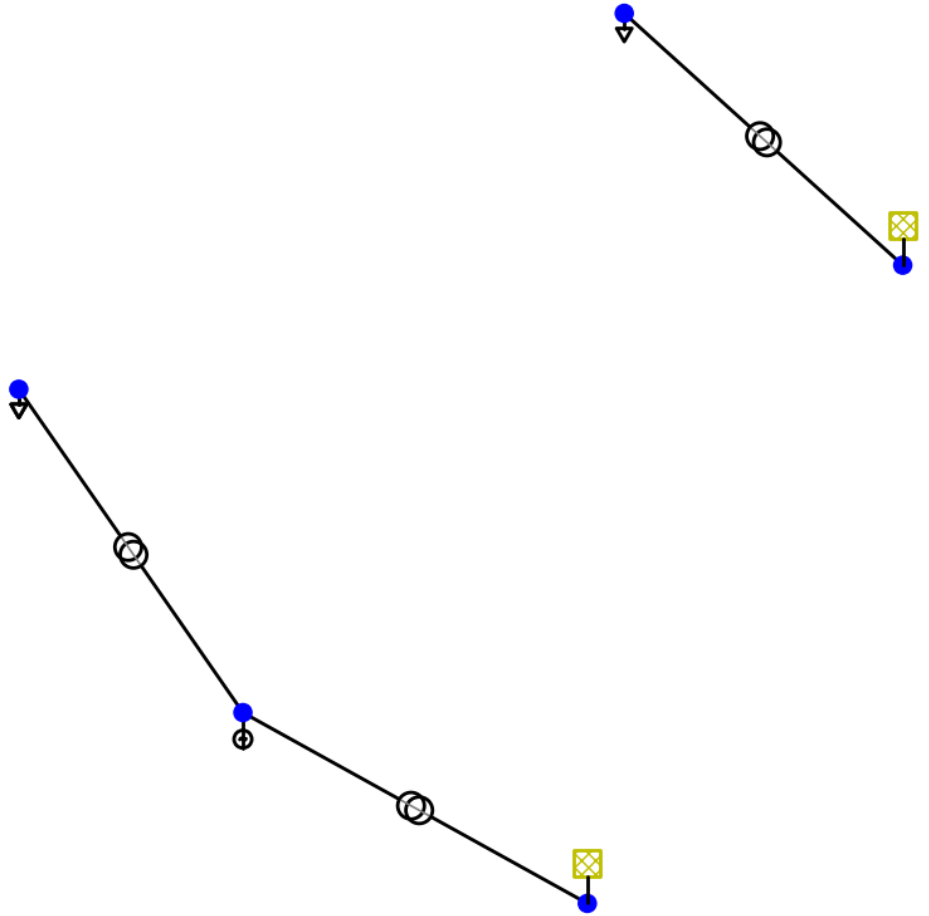
PECIER
Comité Nacional Peruano de la CIER

```
# Diagrama unifilar simple
pp.plotting.simple_plot(net, plot_loads=True, plot_gens=True,
                        trafo_size=1.5, line_width=1.2)

# Diagrama interactivo (requiere plotly)
try:
    import plotly
    pp.plotting.simple_plotly(net, show=True)
except ImportError:
    print("Instala plotly para visualización interactiva: pip install plotly")
```

Visualización de la Red

PECIER
Comité Nacional Peruano de la CIER



PECIER
Comité Nacional Peruano de la CIER

Próximos Pasos

Veamos el ejemplo de:
https://github.com/e2nIEE/pandapower/blob/develop/tutorials/shortcircuit/shortcircuit.ipynb

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia6_pandapower2

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

SESIÓN 1
Parte 3

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

El término capacidad de acogida fotovoltaica se define como la capacidad fotovoltaica máxima que puede conectarse a una red específica, sin dejar de cumplir los códigos de red y los principios de planificación de red pertinentes.
Aquí presentaremos un algoritmo básico para calcular la capacidad de alojamiento FV con pandapower.
La idea básica del cálculo de la capacidad de acogida es aumentar la instalación fotovoltaica hasta que se produzca una violación de cualquier principio o restricción de planificación. Para analizar la capacidad de acogida, necesitamos tres elementos básicos:
1.Evaluación de las violaciones de las restricciones
2.Elegir los puntos de conexión de las nuevas plantas fotovoltaicas
3.Definición de la potencia instalada de las nuevas plantas fotovoltaicas

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de violaciones de restricciones

Nuestra función de ejemplo que evalúa la violación de restricciones se define como:

```
import pandapower as pp

def violations(net):
    pp.runpp(net)
    if net.res_line.loading_percent.max() > 50:
        return (True, "Línea Sobrecargada")
    elif net.res_trafo.loading_percent.max() > 50:
        return (True, "Transformador \n Sobercargado")
    elif net.res_bus.vm_pu.max() > 1.04:
        return (True, "Problemas \n Tensión")
    else:
        return (False, None)
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de violaciones de restricciones

La función ejecuta un flujo de potencia y, a continuación, comprueba la carga de la línea y del transformador (ambas deben ser inferiores al 50%) y el aumento de tensión (que debe ser inferior a 1,04 pu). La función devuelve un indicador booleano para señalar si se ha violado alguna restricción, así como una cadena que indica el tipo de violación de la restricción.

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Elección de un bus de conexión

Si se instalan nuevas plantas fotovoltaicas, hay que elegir un bus de conexión. En este caso, elegimos un bus al azar de cada uno de los buses que tienen una conexión de carga:

```
from numpy.random import choice

def chose_bus(net):
    return choice(net.load.bus.values)
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Elección del tamaño de la instalación fotovoltaica

La función que devuelve un tamaño de planta se da como:

```
from numpy.random import normal

def get_plant_size_mw():
    return normal(loc=0.5, scale=0.05)
```

Esta función devuelve un valor aleatorio de una distribución normal con una media de 0,5 MW y una desviación típica de 0,05 MW. En función de la información existente, también sería posible utilizar otras distribuciones de probabilidad, como una distribución de Weibull, o extraer valores de tamaños de centrales existentes.

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento

Ahora utilizaremos estos bloques para evaluar la capacidad de alojamiento en una red genérica. Utilizaremos como ejemplo la red MV Oberrhein del paquete pandapower networks:

```
import pandapower.networks as nw
def load_network():
    return nw.mv_oberrhein(scenario="generation")
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento

La capacidad de alojamiento se evalúa:

```
import pandas as pd

iterations = 50
results = pd.DataFrame(columns=["instalada", "problema"])

for i in range(iterations):
    net = load_network()
    installed_mw = 0
    while 1:
        violated, violation_type = violations(net)
        if violated:
            results.loc[i] = [installed_mw, violation_type]
            break
        else:
            plant_size = get_plant_size_mw()
            pp.create_sgen(net, chose_bus(net), p_mw=plant_size, q_mvar=0)
            installed_mw += plant_size
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento

Este algoritmo añade nuevas plantas fotovoltaicas hasta que se produce una violación de alguna restricción. A continuación, guarda la capacidad fotovoltaica instalada. Esto se lleva a cabo durante un número de iteraciones (aquí: 50) para obtener una distribución de los valores de capacidad de alojamiento en función de los puntos de conexión y los tamaños de las plantas.

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento

Los resultados pueden visualizarse utilizando matplotlib y seaborn:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.rc('xtick', labelsize=18)
plt.rc('ytick', labelsize=18)
plt.rc('legend', fontsize=18)
plt.rc('axes', labelsize=20)
plt.rcParams['font.size'] = 20

import seaborn as sns
sns.set_style("whitegrid", {'axes.grid' : False})
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10,5))
ax = axes[0]
sns.boxplot(results.instalada, width=.1, ax=ax, orient="v")
ax.set_ylabel("Capacidad instalada [MW]")
# ¿Cuánta potencia se puede instalar antes de que ocurra un problema?
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento

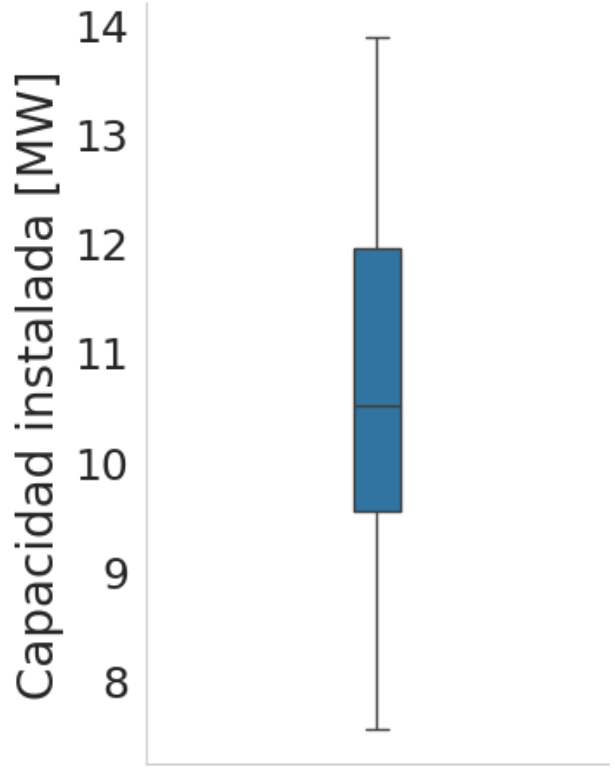
Los resultados pueden visualizarse utilizando matplotlib y seaborn:

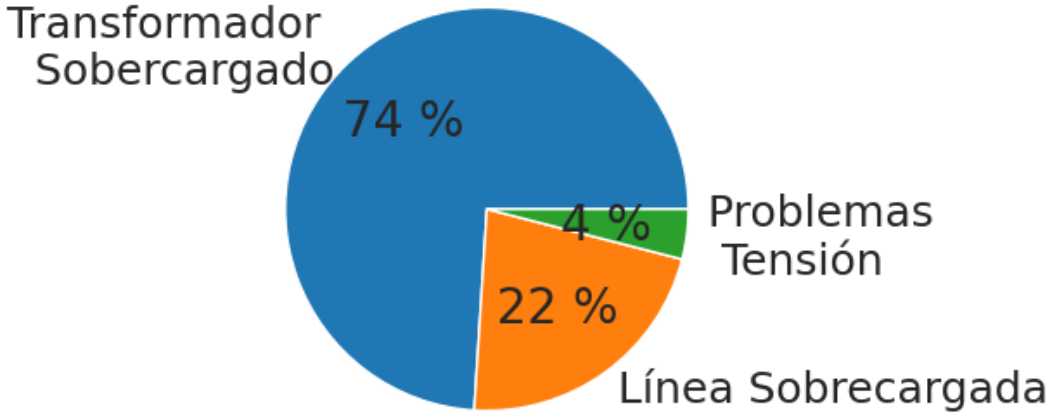
```
ax = axes[1]
ax.axis("equal")
results.problema.value_counts().plot(kind="pie", ax=ax,  autopct=lambda x:"%.0f %%"%
ax.set_ylabel("")
ax.set_xlabel("")
sns.despine()
plt.tight_layout()
```

Capacidad de alojamiento

PECIER
Comité Nacional Peruano de la CIER

Evaluación de la capacidad de alojamiento





La aplicación en estudios de casos más complejos podrían incluir el control Q de plantas fotovoltaicas, controladores de tomas de transformador, distribución más sofisticada de plantas fotovoltaicas, distribución de probabilidad de diferentes buses, etc.

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia6_pandapower3

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

E-mail: pecier@cier.org