

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **FLUJOS ÓPTIMO DE POTENCIA**

**HERRAMIENTA COMPUTACIONAL PARA EL CÁLCULO DE  
FLUJOS ÓPTIMOS DE POTENCIA EN CORRIENTE ALTERNA,  
CONSIDERANDO DIFERENTES FORMULACIONES  
MATEMÁTICAS PARA LAS RESTRICCIONES DEL PROBLEMA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
“INGENIERIA ELÉCTRICA”**

**ALEXANDER VINICIO LLAMUCA ACOSTA**

**[alexander.llamuca@epn.edu.ec](mailto:alexander.llamuca@epn.edu.ec)**

**DIRECTOR: DR. NELSON VICTORIANO GRANDA GUTIÉRREZ**

**[nelson.granda@epn.edu.ec](mailto:nelson.granda@epn.edu.ec)**

**DMQ, Septiembre 2023**

## **CERTIFICACIONES**

Yo, ALEXANDER VINICIO LLAMUCA ACOSTA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**ALEXANDER VINICIO LLAMUCA ACOSTA**

Certifico que el presente trabajo de integración curricular fue desarrollado por ALEXANDER VINICIO LLAMUCA ACOSTA, bajo mi supervisión.

---

**DR. NELSON VICTORIANO GRANDA GUTIÉRREZ**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como los productos resultantes del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ALEXANDER VINICIO LLAMUCA ACOSTA

DR. NELSON VICTORIANO GRANDA GUTIÉRREZ

## **DEDICATORIA**

A mis padres, Vinicio y Alexandra.

A mis hermanos, David, Ariel, Abigail y Valentina.

## **AGRADECIMIENTO**

A Dios por guiarme a lo largo de este camino, brindándome sabiduría y fortaleza para poder alcanzar este objetivo tan importante de mi vida.

A mis padres, Vinicio y Alexandra por siempre brindarme su amor, apoyo, cariño, aliento y sobre todo por creer y confiar en mis capacidades para poder culminar de manera exitosa mis estudios en tan prestigiada universidad.

A mis hermanos por siempre estar presentes con su cariño y motivación para crecer día a día como persona.

A mis abuelitos, Alicia, Ángel, Miguel, por su apoyo, amor y disposición de estar siempre para brindarme su ayuda. Un agradecimiento especial a mi abuelita Mariana que me guía y protege desde el cielo.

A mis amigos quienes de una u otra manera me han brindado su apoyo.

Al Dr. Nelson Granda, por ser un guía durante el desarrollo de este trabajo.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT .....	VIII
<b>1 INTRODUCCIÓN .....</b>	<b>1</b>
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Marco teórico.....	2
1.3.1 Flujo Óptimo de Potencia en Corriente Alterna (OPF-AC) .....	2
1.3.2 Función objetivo.....	3
1.3.2.1 Minimizar costo total de generación. ....	3
1.3.2.2 Linealización de la función de costos. ....	4
1.3.3 Restricciones de igualdad.....	5
1.3.4 Restricciones de desigualdad .....	5
<b>2 METODOLOGÍA .....</b>	<b>8</b>
2.1 Descripción de la herramienta computacional .....	8
2.1.1 Modelación de la Función Objetivo .....	8
2.1.2 OPF-AC en términos de potencia .....	9
2.1.2.1 Balance de potencia en términos de potencia activa y reactiva.....	10
2.1.3 Balance de potencia en términos de corriente.....	12
2.1.3.1 Formulación matemática de OPF-AC con balance de potencia en términos de corriente 13	
2.1.4 OPF-AC con relajación de ecuaciones del flujo de potencia .....	17
2.1.4.1 Linealización de las ecuaciones del flujo de potencia [17]. ....	17
<b>3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>23</b>
3.1 Resultados.....	23
3.1.1 Sistema IEEE de 14 barras .....	23
3.1.2 Sistema Nacional Interconectado (SNI).....	31
3.1.2.1 Escenario Lluvioso .....	31

3.1.2.2	Escenario Seco.....	35
3.2	CONCLUSIONES .....	40
3.3	Recomendaciones .....	40
4	REFERENCIAS BIBLIOGRÁFICAS.....	42
5	ANEXOS .....	44
	ANEXO I. MANUAL DE USUARIO.....	44
	ANEXO II. DATOS DEL SISTEMA IEEE 14 BARRAS .....	51
	ANEXO III. TABLAS DE RESULTADOS DEL SISTEMA IEEE 14 BARRAS .....	54
	ANEXO IV. TABLAS DE RESULTADOS DEL SISTEMA NACIONAL INTERCONETADO (SNI).....	64

## RESUMEN

En el presente Trabajo de Integración Curricular (**TIC**) se desarrolla una herramienta computacional, mediante el uso del lenguaje de programación de Python, que permite dar solución al problema del flujo óptimo de potencia en corriente alterna (**OPF-AC**), y se presenta también una recopilación de información relacionada con el problema: su formulación matemática y métodos de solución.

Se realizará la implementación de varios modelos matemáticos de OPF-AC mediante el uso de la librería PYOMO de Python. Se tendrán en cuenta alternativas de modelación de las restricciones del balance nodal, límites de flujos de potencia. Además, se implementará el modelo del OPF-AC considerando la relajación cuadrática de las restricciones del flujo de potencia. Se presenta una sección para el manejo adecuado del programa donde el usuario podrá seleccionar de manera sencilla los diferentes modelos de la función objetivo, casos de estudio.

Esta herramienta es aplicada al sistema de prueba IEEE de 14 barras y al Sistema Nacional Interconectado (SNI) ecuatoriano considerando dos escenarios hidrológicos: lluvioso y seco, para los cuales se obtienen: el costo total de generación, la potencia de generación y la demanda del sistema, que son presentados mediante gráficas y tablas con sus respectivos análisis.

**PALABRAS CLAVE:** Flujo óptimo de potencia, Pyomo, Python, OPF-AC, SNI, ecuaciones del flujo de potencia, formulación matemática, modelos.



## ABSTRACT

In the present document corresponding to the Curricular Integration Work (TIC) a computational tool is developed using the Python programming language that allows solving the problem of the Optimal Power Flow in alternating current (OPF-AC), and it is presented a collection of information related to the problem: mathematical formulations and solution methods.

The implementation of several mathematical models of OPF-AC will be carried out using the PYOMO Python library. Alternatives for modeling the restrictions of the nodal balance, limits of power flows will be considered. In addition, the OPF-AC model will be implemented considering the quadratic relaxation of the power flow restrictions. A section is presented for the proper management of the program where the user can easily select the different models of objective function, case studies.

This tool is applied to the IEEE 14-Bus system and to the Ecuadorian National Interconnected System (SNI) considering two hydrological scenarios: rainy and dry, for which the results of the total generation cost, generation power and demand of the system, which are presented through graphs and tables with their respective analysis.

**KEYWORDS:** Optimal Power Flow, Pyomo, Python, OPF-AC, SNI, power flow equations, mathematical formulation, models.

# 1 INTRODUCCIÓN

El Despacho Económico (**DE**) se puede describir como el proceso de selección de los generadores que deben ser despachados para lograr que la operación del Sistema Eléctrico de Potencia (**SEP**) sea la más económica, tomando en cuenta la demanda eléctrica total más las pérdidas de potencia y los límites técnicos de los generadores [1]. Para el cálculo del DE no se consideran detalles específicos de la red eléctrica a la se encuentran conectados los generadores, más bien se asocian todos sus efectos a la demanda total y las pérdidas [2].

El Flujo Óptimo de Potencia (**OPF**) tiene como objetivo principal determinar el punto de operación óptimo del SEP con el propósito de minimizar su costo total de operación, las pérdidas totales de la red, maximizar el nivel de seguridad o la combinación de cualquiera de ellas. La obtención de estos objetivos es de vital importancia para las empresas eléctricas dado que su operación, por ley, debe cumplir con el mínimo usos de recursos y garantizar un alto nivel de seguridad [3], [4].

El OPF es un problema de optimización no lineal, no convexo, que describe la operación del SEP en estado estacionario, debido a que se requiere determinar el valor óptimo de las variables de control considerando un conjunto de restricciones no lineales de operación. Por tanto, para la formulación del OPF se requiere de: i) Función objetivo, la cual se deberá optimizar, ii) Conjunto de restricciones de igualdad, que corresponden al balance de potencia y, iii) Conjunto de restricciones de desigualdad, que describen los límites operativos de los componentes del SEP [4], [5].

El flujo óptimo de potencia de corriente alterna (**OPF-AC**) considera las ecuaciones completas del flujo de potencia a través de los elementos de transmisión. Comparado la formulación matemática de un OPF de corriente continua (**OPF-DC**) con el OPF-AC, éste último tiene mayores beneficios dado que es más preciso, y, además, considera la potencia reactiva, las pérdidas en el sistema de transmisión y modelación detallada de equipamiento como HVDC, FACTS, cambiadores de tomas, compensación reactiva, etc. Sin embargo, las ecuaciones del flujo de potencia AC son cuadráticas debido a que la potencia depende del cuadrado del voltaje, por lo que la resolución es más compleja al tener un problema de optimización no lineal, no convexo [1].

Para el desarrollo de la herramienta computacional para resolver el problema del OPF-AC se empleará la librería Pyomo que permite formular modelos de optimización mediante varios componentes como: función objetivo, variables, diferentes restricciones, ecuaciones

lineales como no lineales, y, además, permite trabajar con datos de Excel tanto para la inicialización de rutinas de programación como para la impresión de resultados.

## 1.1 Objetivo general

Desarrollar una herramienta computacional para la solución del flujo óptimo de potencia en corriente alterna considerando diferentes formulaciones matemáticas para las distintas restricciones del problema.

## 1.2 Objetivos específicos

1. Implementar un modelo OPF – AC considerando diferentes formulaciones matemáticas para la restricción del flujo de potencia, utilizando la librería PYOMO de Python.
2. Implementar un modelo OPF – AC tomando en cuenta diferentes alternativas para modelar la restricción de balance nodal, utilizando la librería PYOMO de Python.
3. Implementar un modelo OPF – AC que implemente diferentes alternativas de relajación de las restricciones del problema, mediante el uso de la librería PYOMO de Python.
4. Aplicar la herramienta computacional desarrollada al sistema IEEE de 14 barras, y al Sistema Nacional Interconectado ecuatoriano.

## 1.3 Marco teórico

### 1.3.1 Flujo Óptimo de Potencia en Corriente Alterna (OPF-AC)

El OPF-AC es un problema de optimización que tiene como objetivo encontrar un punto de operación en estado estable del SEP garantizando el mínimo costo de operación mientras se cumplen las restricciones de igualdad y desigualdad [6], para asegurar una operación confiable y, además, poner en práctica mercados eficientes [7].

La formulación general del OPF-AC se describe de la siguiente manera:

$$\min f(x) \tag{1.1}$$

Sujeto a:

$$g(x) = 0 \tag{1.2}$$

$$h^{min} \leq h(x) \leq h^{max} \tag{1.3}$$

Donde,  $f(x)$  representa la función objetivo a minimizar, por ejemplo, el costo total de generación o las pérdidas totales del sistema,  $g(x)$  representa las restricciones de igualdad que, generalmente, son las ecuaciones de balance de potencia nodal, y  $h(x)$  representa las restricciones de desigualdad, que contiene los límites técnicos de operación [8].

El OPF-AC es empleado tanto en la operación como en la planificación del SEP, por ejemplo, para la planificación de expansión, la gestión de la red, los mercados diarios y para el control en tiempo real [9].

### 1.3.2 Función objetivo

En la formulación del OPF la función objetivo es de suma importancia ya que es necesaria para realizar el proceso de optimización matemática dando como resultado la solución óptima del problema [10]. Existen algunos tipos de función objetivo donde su elección depende del propósito del estudio, por ejemplo, minimización del costo total de generación, de las pérdidas o de los cortes de carga [11].

#### 1.3.2.1 Minimizar costo total de generación.

Para minimizar el costo total de generación se tiene la siguiente función objetivo [10]:

$$FO = \min \sum_{n=1}^{N_{gen}} F_n(P_{gen\ n}) \quad (1.4)$$

Donde:

$F_n(P_{gen\ n})$  Es la función de costo de generación del generador  $n$ . [USD/h]

$N_{gen}$  Es el número de generadores del SEP.

$P_{gen\ n}$  Es la potencia activa generada por cada generador  $n$ .

El costo de generación, como función de la potencia eléctrica generada, usualmente se expresa como [10]:

$$F_n(P_{gen\ n}) = a_n * P_{gen\ n}^2 + b_n * P_{gen\ n} + c_n \quad (1.5)$$

Donde:

$a_n$  Coeficiente cuadrático de costos de la unidad  $n$ .

$b_n$  Coeficiente lineal de costos de la unidad  $n$ .

$c_n$  Coeficiente de costos constante de la unidad  $n$ .

### 1.3.2.2 Linealización de la función de costos.

En la figura 1, se muestra la función de costos de generación que, en general, es una función cuadrática; sin embargo, para facilitar el proceso de solución puede representarse mediante una aproximación lineal por segmentos [2].

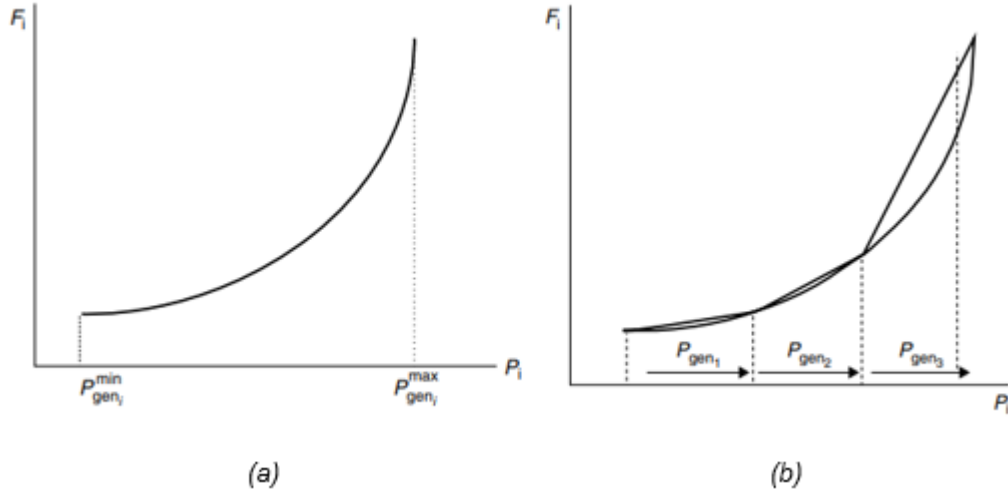


Figura 1.- Función de costos no lineal: (a) cuadrática, (b) aproximación lineal por segmentos [2].

Por tanto, la función de costos de generación puede ser expresada como un conjunto de funciones lineales distribuidas en tres segmentos donde la variable  $P_{gen\ n}$  es reemplazada por tres nuevas variables descritas como  $P_{gen\ n1}, P_{gen\ n2}, P_{gen\ n3}$ . Cada segmento tendrá designado un valor de pendiente  $s_{n1}, s_{n2}, s_{n3}$ , entonces la función de costos se expresa como la suma de la potencia de la generación mínima más la multiplicación de cada uno de los costos por sus respectivas pendientes [2].

$$F_i(P_{gen\ n}) = P_{gen\ n}^{min} + s_{n1} * P_{gen\ n1} + s_{n2} * P_{gen\ n2} + s_{n3} * P_{gen\ n3} \quad (1.6)$$

$$0 \leq P_{gen\ n,k} \leq P_{gen\ n,k}^{min} \quad \forall k = 1,2,3 \quad (1.7)$$

Donde:

- $F_i(P_{gen\ n})$  Es la función de costo del generador  $n$ . [USD/h]
- $P_{gen\ n}$  Es la potencia activa generada por cada generador  $n$ .
- $P_{gen\ n}^{min}$  Es la potencia activa mínima generada por el generador  $n$ .
- $s_{n,k}$  Valor de la pendiente del generador  $n$  correspondiente a cada segmento  $k$ .

### 1.3.3 Restricciones de igualdad

Por lo general, el balance de potencia se emplea como restricción de igualdad  $g(x)$ , es decir, se debe cumplir que la potencia total generada sea igual a la potencia total de demanda más las pérdidas [12]:

$$\sum_{\substack{n \in \Omega \\ n=1}}^{N \in \Omega} P_{gen\ n} = \sum_{i=1}^N P_{d\ i} + P_i \quad \forall i \in N \quad (1.8)$$

$$\sum_{n=1}^{N \in \Omega} Q_{gen\ n} = \sum_{i=1}^N Q_{d\ i} + Q_i \quad \forall i \in N \quad (1.9)$$

Donde:

$P_{gen\ n}$	Potencia activa del generador $n$ conectado en la barra $i$ . [MW]
$Q_{gen\ n}$	Potencia reactiva del generador $n$ conectado en la barra $i$ . [MVar]
$P_{d\ i}$	Potencia activa de demanda conectada en la barra $i$ . [MW]
$Q_{d\ i}$	Potencia reactiva de demanda conectada en la barra $i$ . [MVar]
$P_i$	Potencia activa neta inyectada en la barra $i$ . [MW]
$Q_i$	Potencia reactiva neta inyectada en la barra $i$ . [MVar]
$\Omega$	Conjunto de generadores conectados en la barra $i$ .

Los términos de  $P_i$  y  $Q_i$  se pueden calcular como la sumatoria de los flujos de potencia AC a través de los elementos que se conectan a la barra  $i$ , y pueden ser expresados en coordenadas polares o en coordenadas rectangulares [12].

### 1.3.4 Restricciones de desigualdad

Este conjunto de restricciones considera los límites máximos y mínimos tanto de las variables de control como de estado [8]. A continuación, se presentan cada uno de estos límites operativos:

#### **Límites técnicos de generación:**

*Potencia activa:*

$$P_{gen\ n}^{min} \leq P_{gen\ n} \leq P_{gen\ n}^{max} \quad \forall i \in N \quad (1.10)$$

Donde:

$P_{gen\ n}^{max}$	Valor de potencia activa máxima de generación de la unidad $n$ conectada a la barra $i$ . [MW]
--------------------	--

$P_{gen\ n}^{min}$  Valor de potencia activa mínima de generación de la unidad  $n$  conectada a la barra  $i$ . [MW]

**Potencia reactiva:**

$$Q_{gen\ n}^{min} \leq Q_{gen} \leq Q_{gen\ n}^{max} \quad \forall i \in N \quad (1. 11)$$

Donde:

$Q_{gen\ n}^{max}$  Valor de potencia reactiva máxima de generación de la unidad  $n$  conectada a la barra  $i$ . [MVar]

$Q_{gen\ n}^{min}$  Valor de potencia reactiva mínima de generación de la unidad  $n$  conectada a la barra  $i$ . [MVar]

**Límites de magnitud de voltaje:**

Cuando se ocupan restricciones del flujo de potencia expresadas en coordenadas polares, se expresan como:

$$V_i^{min} \leq V_i \leq V_i^{max} \quad \forall i \in N \quad (1. 12)$$

Cuando se ocupan restricciones del flujo de potencia expresadas en coordenadas rectangulares, se expresan como:

$$V_i^{min} \leq \sqrt{(V_i^r)^2 + (V_i^m)^2} \leq V_i^{max} \quad \forall i \in N \quad (1. 13)$$

Donde:

$V_i^{max}$  Valor de voltaje máximo en la barra  $i$ .

$V_i^{min}$  Valor de voltaje mínimo en la barra  $i$ .

$V_i^r$  Valor real del voltaje en la barra  $i$ .

$V_i^m$  Valor imaginario del voltaje en la barra  $i$ .

**Límites del ángulo de voltaje:**

$$\delta_i^{min} \leq \delta_i \leq \delta_i^{max} \quad \forall i \in N \quad (1. 14)$$

Donde:

$\delta_i^{max}$  Valor del ángulo máximo del voltaje en la barra  $i$ .

$\delta_i^{min}$  Valor del ángulo mínimo del voltaje en la barra  $i$ .

**Límites de flujo de potencia aparente en líneas de transmisión o transformadores:**

$$S_{ij}^{min} \leq S_{ij} \leq S_{ij}^{max} \quad \forall i, j \in N \quad (1. 15)$$

$$S_{ij}^{min^2} \leq P_{ij}^2 + Q_{ij}^2 \leq S_{ij}^{max^2} \quad \forall i, j \in N \quad (1. 16)$$

$$P_{ij} = G_{ij}V_i^2 - V_i * V_j(B_{ij} \sin(\delta_{ij}) + G_{ij} \cos(\delta_{ij})) \quad \forall i, j \in N \quad (1. 17)$$

$$Q_{ij} = -B_{ij}V_i^2 - V_i * V_j(G_{ij} \sin(\delta_{ij}) - B_{ij} \cos(\delta_{ij})) \quad \forall i, j \in N \quad (1. 18)$$

Donde:

$S_{ij}^{max}$  Valor de potencia aparente máxima transmitida de la barra  $i$  hacia la barra  $j$ .

$S_{ij}^{min}$  Valor de potencia aparente mínima transmitida de la barra  $i$  hacia la barra  $j$ .

$P_{ij}$  Valor de potencia activa transmitida de la barra  $i$  hacia la barra  $j$ .

$Q_{ij}$  Valor de potencia reactiva transmitida de la barra  $i$  hacia la barra  $j$ .



## **2 METODOLOGÍA**

Para el desarrollo de la herramienta computacional que permita resolver el problema del OPF-AC, considerando diferentes formulaciones matemáticas, se empleó el entorno Spyder y la librería Pyomo, que permite realizar la formulación de diferentes modelos matemáticos de optimización [13].

### **2.1 Descripción de la herramienta computacional**

Se implementan cuatro diferentes formulaciones matemáticas para resolver el OPF-AC:

- i. Ecuación de balance considerando potencias activa y reactiva con las ecuaciones del flujo de potencia en coordenadas polares,
- ii. Ecuación de balance considerando potencias activa y reactiva con las ecuaciones del flujo de potencia en coordenadas rectangulares,
- iii. Ecuación de balance considerando potencias activa y reactiva con la linealización de las ecuaciones del flujo de potencia, y,
- iv. Ecuación de balance considerando corrientes con el flujo de potencia expresado en coordenadas rectangulares.

Además, se implementan dos formas para expresar a la función objetivo: la primera que representa a la función de costos de generación mediante una ecuación cuadrática y, la segunda que expresa a la función de costos de manera lineal. Entonces, el tiempo de solución del programa cambiara de acuerdo con el tipo de función seleccionada.

Para encontrar la solución de los diferentes casos, se hace uso del solver IPOPT (Interior Point Optimizer), el cual es un software de código abierto que tiene la capacidad de resolver problemas de optimización no lineal de un alto grado [14].

#### **2.1.1 Modelación de la Función Objetivo**

Para este caso, se cuenta con la implementación de dos modelos de función objetivo, la función de costos cuadrática y la función de costos lineal, las cuales serán empleadas para todos los casos del OPF-AC y su diferencia se podrá observar tanto en el tiempo que se demora el programa para obtener una solución óptima como en el valor del costo total del sistema.

```

78 #----- SELECCIÓN DE LA FUNCIÓN OBJETIVO -----
79
80 def objective(model):
81     costoTotal=0
82     costoTotal=sum(model.USD[j] for j in model.G)
83     return costoTotal
84 model.OBJ = pyo.Objective(rule=objective)
85 def cost_function(model, l):
86     costo=0
87     if MetFO==1:
88         costo=model.c2[1]*model.P[1]*model.P[1]+model.c1[1]*model.P[1]+model.c0[1] # FUNCIÓN OBJETIVO CUADRÁTICA
89     else:
90         costo=model.c2[1]+model.c1[1]*model.P[1] #FUNCIÓN OBJETIVO LINEALIZADA
91     return model.USD[1]==costo
92 model.cost_function = pyo.Constraint(model.G,rule=cost_function)
93

```

Figura 2.- Código de la función objetivo.

Se puede observar en la figura 2 el desarrollo del código correspondiente a la función objetivo, donde se tiene dos funciones, la denominada `objective` que contiene el costo total del sistema en dólares y, la otra función `cost_function` que contiene la formulación de las funciones objetivos empleando el condicional `if, else`, que en el caso de elegir en el archivo 'ejecutable.py' la opción #1 en la variable `MetFO` se utilizarla la ecuación cuadrática (1.5) para la resolución del problema, o caso contrario seleccionado la opción #0 se hace uso de la ecuación lineal (1.6).

### 2.1.2 OPF-AC en términos de potencia

Para este modelo se considera la restricción de igualdad correspondiente al balance nodal en términos de potencia activa y reactiva, que se encuentran representadas en las ecuaciones (1.8) y (1.9) respectivamente.

```

#-----RETRICCIÓN DE IGUALDAD -----
def Balance_P_def2(model):
    P=sum(model.P[j]
          for j in model.G)

    Pd=sum(model.demP[d]
           for d in model.D)
    Pl=sum(model.FLP[n]
           for n in model.N)

    return P==Pd+Pl #P DE GENERACIÓN = P DE DEMANDA + P DE PÉRDIDAS

model.Balance_P2 = pyo.Constraint(rule=Balance_P_def2, doc = ' Balance de Potencia Activa Nodal')

def balance_potencia2Q(model):
    Q=sum(model.Q[j]
          for j in model.G)

    Qd=sum(model.demQ[d]
           for d in model.D)
    Ql=sum(model.FLQ[n]
           for n in model.N)
    return Q==Qd+Ql #Q DE GENERACIÓN = Q DE DEMANDA + Q DE PÉRDIDAS

model.bal_pot2Q = pyo.Constraint(rule=balance_potencia2Q,doc = ' Balance de Potencia Reactiva Nodal')

```

Figura 3.- Código del Balance Nodal.

En la figura 3 se muestra el desarrollo de las ecuaciones del balance nodal en el programa, donde tanto para la potencia activa como reactiva se crea su respectiva función, se puede

observar que se crea las variables que obtendrán los valores totales de potencia de generación, demanda y pérdidas del sistema para al final formar la ecuación que describe el balance, es decir, la potencia de generación total debe ser igual a la potencia de demanda más las de pérdidas.

### 2.1.2.1 Balance de potencia en términos de potencia activa y reactiva

Para este modelo las restricciones de igualdad se expresan en las ecuaciones (2.1) y (2.1) de la siguiente manera [9]:

$$\sum_{n=1}^{N \in \Omega} P_{gen\ n} = \sum_{i=1}^N P_{di} + P_i \quad \forall i \in N \quad (2.1)$$

$$\sum_{n=1}^{N \in \Omega} Q_{gen\ n} = \sum_{i=1}^N Q_{di} + Q_i \quad \forall i \in N \quad (2.2)$$

Además, se plantean dos formas de representar las ecuaciones del flujo de potencia de corriente alterna, la primera expresando a los voltajes en coordenadas polares y, la segunda con los voltajes en coordenadas rectangulares, por lo cual el modelo modificará la restricción de los límites del voltaje dependiente del caso seleccionado. A continuación, se presenta el desarrollo de las expresiones matemáticas:

#### Expresadas en coordenadas polares:

Para esta formulación se utiliza la representación en forma polar del voltaje, donde  $V_i$  es la magnitud del voltaje y,  $\delta_i$  es el ángulo en la barra  $i$ .

Los términos correspondientes a  $P_i$  y  $Q_i$  de las ecuaciones (2.1) y (2.2) representan a las ecuaciones del flujo de potencia de corriente alterna, que se expresan en forma polar de la siguiente manera [15]:

$$P_i = V_i \sum_{j=1}^N V_j * (G_{ij} * \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)) \quad \forall i \in N \quad (2.3)$$

$$Q_i = V_i \sum_{j=1}^N V_j * (G_{ij} * \sin(\delta_i - \delta_j) - B_{ij} * \cos(\delta_i - \delta_j)) \quad \forall i \in N \quad (2.4)$$

Donde:

$V_i, V_j$  Magnitud del voltaje en la barra  $i$  y  $j$ , respectivamente.

$\delta_i, \delta_j$  Ángulos del voltaje en la barra  $i$  y  $j$ , respectivamente

$G_{ij}$  Valor de conductancia del elemento  $i - j$  de la matriz  $Y_{barra}$ .  $[1/\Omega]$

$B_{ij}$  Valor de suceptancia del elemento  $i - j$  de la matriz  $Y_{barra}$ .  $[1/\Omega]$

```
# EXPRESADO EN COORDENADAS POLARES:
if MetResolucion==1:
def Calculo_Potencia_Barra(model,n):
    suma=0
    for j in model.N:
        suma+=(np.real(model.Yb[n,j])*pyo.cos(model.delta[n]-model.delta[j])+np.imag(model.Yb[n,j])*pyo.sin(model.delta[n]-model.delta[j]))*model.v[j]
    return model.FLP[n]==model.v[n]*suma*model.baseMVA
model.Calculo_Potencia_Barra=pyo.Constraint(model.N, rule=Calculo_Potencia_Barra, doc = 'Flujo de potencia activa')
def Calculo_Potencia_BarraQ(model,n):
    suma=0
    for j in model.N:
        suma+=(np.real(model.Yb[n,j])*pyo.sin(model.delta[n]-model.delta[j])-np.imag(model.Yb[n,j])*pyo.cos(model.delta[n]-model.delta[j]))*model.v[j]
    return model.FLQ[n]==model.v[n]*suma*model.baseMVA
model.Calculo_Potencia_BarraQ=pyo.Constraint(model.N, rule=Calculo_Potencia_BarraQ, doc = 'Flujo de potencia reactiva')
```

Figura 4.- Representación de las ecuaciones del flujo de potencia en coordenadas polares.

En la figura 4 se puede observar la implementación de las ecuaciones (2.3) y (2.4), donde se define dos nuevas funciones para el cálculo del flujo potencia activa y reactiva empleando un lazo for para obtener los valores de potencia en cada una de las barras que conforman el sistema, la formulación del sumatorio de las ecuaciones (2.3) y (2.4), cada una con sus respectivos valores de voltaje, ángulo, conductancia y suceptancia, se encuentran desarrolladas en la variable suma, que luego retorna la variable *model.FLP* donde se multiplica al sumatorio por su respectivo voltaje correspondiente a la barra *i* para completar todos los términos de las ecuaciones. Entonces, este modelo se utilizará cuando el usuario seleccione la opción #1 en el método de resolución en el archivo 'ejecutable.py'.

### Expresadas en coordenadas rectangulares:

En esta formulación, se exprese de forma rectangular al voltaje, es decir, en una parte real y una imaginaria que se representa por  $V_i^r$  y  $V_i^m$  respectivamente. Entonces, las ecuaciones de la potencia activa y reactiva inyectadas en cada barra del sistema se expresan de la siguiente manera [15]:

$$P_i = \sum_{j=1}^N [V_i^r * (G_{ij} * V_j^r - B_{ij} * V_j^m) + V_i^m * (G_{ij} * V_j^m + B_{ij} * V_j^r)] \quad \forall i \in N \quad (2.5)$$

$$Q_i = \sum_{j=1}^N [V_i^m * (G_{ij} * V_j^r - B_{ij} * V_j^m) - V_i^r * (G_{ij} * V_j^m + B_{ij} * V_j^r)] \quad \forall i \in N \quad (2.6)$$

Donde:

- $V_i^r$  Valor de la parte real del voltaje en la barra *i*.
- $V_i^m$  Valor de la parte imaginaria del voltaje en la barra *i*.
- $V_j^r$  Valor de la parte real del voltaje en la barra *j*.
- $V_j^m$  Valor de la parte imaginaria del voltaje en la barra *j*.
- $G_{ij}$  Valor de conductancia del elemento  $i - j$  de la matriz  $Y_{barra}$ .

$B_{ij}$  Valor de susceptancia del elemento  $i - j$  de la matriz  $Y_{barra}$ .

```
#EXPRESADO EN COORDENADAS RECTANGULARES
if MetResolucion==2:
    def Calculo_Potencia_Barra(model,n):
        suma=0
        vrn=model.v[n]*pyo.cos(model.delta[n])
        vjn=model.v[n]*pyo.sin(model.delta[n])
        for j in model.N:
            Gnm=np.real(model.Yb[n,j])
            Bnm=np.imag(model.Yb[n,j])
            vrjm=model.v[j]*pyo.cos(model.delta[j])
            vjnm=model.v[j]*pyo.sin(model.delta[j])
            suma+=vrn*(Gnm*vrjm-Bnm*vjnm)+vjn*(Gnm*vjnm+Bnm*vrjm)
        return model.FLP[n]==suma*model.baseMVA
    model.Calculo_Potencia_Barra=pyo.Constraint(model.N, rule=Calculo_Potencia_Barra, doc = 'Flujo de potencia activa')
    def Calculo_Potencia_BarraQ(model,n):
        suma=0
        vrn=model.v[n]*pyo.cos(model.delta[n])
        vjn=model.v[n]*pyo.sin(model.delta[n])
        for j in model.N:
            Gnm=np.real(model.Yb[n,j])
            Bnm=np.imag(model.Yb[n,j])
            vrjm=model.v[j]*pyo.cos(model.delta[j])
            vjnm=model.v[j]*pyo.sin(model.delta[j])
            suma+=vjn*(Gnm*vrjm-Bnm*vjnm)-vrn*(Gnm*vjnm+Bnm*vrjm)
        return model.FLQ[n]==suma*model.baseMVA
    model.Calculo_Potencia_BarraQ=pyo.Constraint(model.N, rule=Calculo_Potencia_BarraQ, doc = 'Flujo de potencia reactiva')
```

Figura 5.- Código desarrollado para las ecuaciones del flujo en coordenadas rectangulares.

En la figura 5 se muestra el uso de dos funciones, una para encontrar la de potencia activa y otra para la potencia reactiva por cada barra del sistema donde, en ambos casos, se empieza definiendo una variable suma, luego se procede a definir cada uno de los valores de la parte real e imaginaria del voltaje para la barra  $i$  y  $j$ , respectivamente,. Además, se puede observar que se emplea un lazo for que permitirá obtener el valor total de la potencia neta en cada barra. Las ecuaciones (2.5) y (2.6) se formulan en la variable suma que se forma a partir de las variables reales e imaginarias del voltaje junto con los valores de susceptancias y conductancias obtenidas en la matriz  $Y_{barra}$  del sistema y, finalmente el valor total de potencia de cada una de las barras retorna en la variable  $model.FLP$  y  $model.FLQ$ , respectivamente, empleadas para cumplir con el balance nodal de potencia. Para resolver el problema con esta formulación matemática de los flujos se debe seleccionar la opción #2 en el método de resolución en el archivo ejecutable.

### 2.1.3 Balance de potencia en términos de corriente

En cuanto a este modelo se emplea el balance de corriente nodal para cada una de las barras del sistema, donde se representará la corriente en coordenadas rectangulares, es decir, contara con un valor real y otro imaginario expresados como:  $i_i = i_i^r + i_i^m$  [16].

Las ecuaciones que expresan el flujo de corriente por líneas y transformadores en función de coordenadas rectangulares se describen de la siguiente manera [12]:

$$i_{ij}^r = g_{ij}^{net} * V_i^r - b_{ij}^{net} * V_i^m + G_{ij} * V_j^r - B_{ij} * V_j^m \quad \forall (i,j) \in K \quad (2.7)$$

$$i_{ij}^m = g_{ij}^{net} * V_i^m + b_{ij}^{net} * V_i^r + G_{ij} * V_j^m + B_{ij} * V_j^r \quad \forall (i,j) \in K \quad (2.8)$$

Donde:

$V_i^r$	Valor de la parte real del voltaje en la barra $i$ .
$V_i^m$	Valor de la parte imaginaria del voltaje en la barra $i$ .
$V_j^r$	Valor de la parte real del voltaje en la barra $j$ .
$V_j^m$	Valor de la parte imaginaria del voltaje en la barra $j$ .
$g_{ij}^{net}$	Valor de conductancia del elemento $i - j$ de la matriz $Y_{net}$ .
$b_{ij}^{net}$	Valor de susceptancia del elemento $i - j$ de la matriz $Y_{net}$ .
$K$	Conjunto de líneas de transmisión

Por tanto, las ecuaciones que definen al balance de corriente nodal para cada barra del sistema se escriben de la siguiente manera:

$$i_i^r - \left( \sum_{(i,j) \in K} i_{ij}^r + G_{ij}^{sh} * V_i^r - B_{ij}^{sh} * V_i^m \right) = 0 \quad (2.9)$$

$$i_i^m - \left( \sum_{(i,j) \in K} i_{ij}^m + G_{ij}^{sh} * V_i^m + B_{ij}^{sh} * V_i^r \right) = 0 \quad (2.10)$$

Donde:

$i_i^r$	Valor de la parte real de corriente en la barra $i$ .
$i_i^m$	Valor de la parte imaginaria de corriente en la barra $i$ .
$G_{ij}^{sh}$	Valor de conductancia shunt del elemento $i - j$ de la matriz $Y_{barra}$ .
$B_{ij}^{sh}$	Valor de susceptancia shunt del elemento $i - j$ de la matriz $Y_{barra}$ .

### 2.1.3.1 Formulación matemática de OPF-AC con balance de potencia en términos de corriente

Para este modelo la ecuación (2.11) representa la función de costos cuadrática, las restricciones de igualdad se expresan en las ecuaciones (2.12) y (2.13), mientras que las ecuaciones (2.14) a (2.19) representan las restricciones de desigualdad. Por tanto, el OPF-AC en términos de corriente se formula de la siguiente manera [16]:

$$F_n(P_{gen\ n}) = a_n * P_{gen\ n}^2 + b_n * P_{gen\ n} + c_n \quad (2.11)$$

s.a:

$$i_i^r = \left( \sum_{(i,j) \in K} i_{ij}^r + G_{ij}^{sh} * V_i^r - B_{ij}^{sh} * V_i^m \right) \quad \forall i \in N \quad (2.12)$$

$$i_i^m = \left( \sum_{(i,j) \in K} i_{ij}^r + G_{ij}^{sh} * V_i^m + B_{ij}^{sh} * V_i^r \right) \quad \forall i \in N \quad (2.13)$$

$$P_{gen\ n} - (V_i^r * i_i^r + V_i^m * i_i^m) = P_{di} \quad (2.14)$$

$$Q_{gen\ n} - (V_i^r * i_i^r - V_i^m * i_i^m) = Q_{di} \quad (2.15)$$

$$P_{gen\ n}^{min} \leq P_{gen\ n} \leq P_{gen\ n}^{max} \quad \forall i \in N \quad (2.16)$$

$$Q_{gen\ n}^{min} \leq Q_{gen\ n} \leq Q_{gen\ n}^{max} \quad \forall i \in N \quad (2.17)$$

$$V_i^{min} \leq \sqrt{(V_i^r)^2 + (V_i^m)^2} \leq V_i^{max} \quad \forall i \in N \quad (2.18)$$

$$\sqrt{(i_{ij}^r)^2 + (i_{ij}^m)^2} \leq i_{ij}^{max} \quad \forall i, j \in K \quad (2.19)$$

Donde, las ecuaciones (2.16) y (2.17) contienen los límites de potencia activa y reactiva en cada barra, (2.14) y (2.15) representan el balance de potencia con el flujo de potencia en términos de corriente y voltaje de forma rectangular, los límites de voltaje se expresan en (2.18) y, los límites correspondientes al flujo de corriente por líneas y transformadores se muestra en la ecuación (2.19).

El desarrollo de este modelo parte del cálculo de los valores de corriente que fluyen a través de líneas de transmisión y transformadores correspondientes a las ecuaciones (2.7) y (2.8) que permiten formar las ecuaciones del balance nodal de corriente, para luego cumplir tanto con las restricciones de igualdad como de desigualdad que conforman el modelo total. A continuación, se muestra la implementación dentro del programa del modelo:

```

#----- CÁLCULO DEL FLUJO DE CORRIENTE POR LINEAS Y TRANSFORMADORES -----
if MetResolucion==3:
def Calculo_corriente_R_linea_from(model,l):
    n=int(py0.value(model.lpos[0,l]))
    m=int(py0.value(model.lpos[1,l]))
    Vnr= model.v[n]*py0.cos(model.delta[n])
    Vnj=model.v[n]*py0.sin(model.delta[n])
    Vmr= model.v[m]*py0.cos(model.delta[m])
    Vmj= model.v[m]*py0.sin(model.delta[m])
    Gnet12 = np.real(model.Ynet[n,m])
    Bnet12 = np.imag(model.Ynet[n,m])
    G12 = np.real(model.Yb[n,m])
    B12 = np.imag(model.Yb[n,m])
    return model.ILfromR[l]==Gnet12*Vnr-Bnet12*Vnj+G12*Vmr-B12*Vmj
def Calculo_corriente_I_linea_from(model,l):
    n=int(py0.value(model.lpos[0,l]))
    m=int(py0.value(model.lpos[1,l]))
    Vnr= model.v[n]*py0.cos(model.delta[n])
    Vnj=model.v[n]*py0.sin(model.delta[n])
    Vmr= model.v[m]*py0.cos(model.delta[m])
    Vmj= model.v[m]*py0.sin(model.delta[m])
    Gnet12 = np.real(model.Ynet[n,m])
    Bnet12 = np.imag(model.Ynet[n,m])
    G12 = np.real(model.Yb[n,m])
    B12 = np.imag(model.Yb[n,m])
    return model.ILfromJ[l]==Gnet12*Vnj+Bnet12*Vnr+G12*Vmj+B12*Vmr
def Calculo_corriente_R_linea_to(model,l):
    n=int(py0.value(model.lpos[0,l]))
    m=int(py0.value(model.lpos[1,l]))
    Vnr= model.v[n]*py0.cos(model.delta[n])
    Vnj=model.v[n]*py0.sin(model.delta[n])
    Vmr= model.v[m]*py0.cos(model.delta[m])
    Vmj= model.v[m]*py0.sin(model.delta[m])
    Gnet21 = np.real(model.Ynet[m,n])
    Bnet21 = np.imag(model.Ynet[m,n])
    G21 = np.real(model.Yb[m,n])
    B21 = np.imag(model.Yb[m,n])
    return model.ILtoR[l]==Gnet21*Vmr-Bnet21*Vmj+G21*Vnr-B21*Vnj
def Calculo_corriente_I_linea_to(model,l):
    n=int(py0.value(model.lpos[0,l]))
    m=int(py0.value(model.lpos[1,l]))
    Vnr= model.v[n]*py0.cos(model.delta[n])
    Vnj=model.v[n]*py0.sin(model.delta[n])
    Vmr= model.v[m]*py0.cos(model.delta[m])
    Vmj= model.v[m]*py0.sin(model.delta[m])
    Gnet21 = np.real(model.Ynet[m,n])
    Bnet21 = np.imag(model.Ynet[m,n])
    G21 = np.real(model.Yb[m,n])
    B21 = np.imag(model.Yb[m,n])
    return model.ILtoJ[l]==Gnet21*Vmj+Bnet21*Vmr+G21*Vnj+B21*Vnr

```

Figura 6.- Código implementado para el cálculo del flujo de corriente por líneas y trafos.

En la figura 6 se muestra el desarrollo de las ecuaciones (2.7) y (2.8) para encontrar los valores de corriente real e imaginaria que fluye a través de las líneas de transmisión y transformadores del sistema, donde las expresiones *from* corresponden a los valores de corriente que fluye desde la barra *i* hacia *j*, mientras que las funciones *to* contienen los valores que fluyen desde la barra *j* hacia *i*.



```

#----- BALANCE DE CORRIENTE NODAL PARA CADA BARRA -----
#CORRIENTE POR BARRA REAL:
def Calculo_Corriente_Barra_R(model,n):
    InI=0
    Gshunt = 0
    Bshunt = 0
    # SUMATORIO DEL NODO DE LA CORRIENTE
    posAnt=[]
    posAnt2=[]
    for l in model.L:
        pos0=int(pyo.value(model.lpos[0,1]))
        pos1=int(pyo.value(model.lpos[1,1]))
        if pos0==n:
            exist_count = list(posAnt).count(pos0)
            exist_count2 = list(posAnt2).count(pos1)
            if exist_count > 0 and exist_count2 > 0: # TOMAMOS EN CUENTA LAS LÍNEAS EN PARALELO
                print("Paralelo: Pos0:", pos0, "Pos1:", pos1)
            else:
                InI=InI+model.IlfromR[l]
                Gshunt+=np.real(model.Yb[pos0,pos1])+np.real(model.Ynet[pos0,pos1])
                Bshunt+=np.imag(model.Yb[pos0,pos1])+np.imag(model.Ynet[pos0,pos1])
                posAnt.append(pos0)
                posAnt2.append(pos1)
        if pos1==n:
            exist_count = list(posAnt).count(pos0)
            exist_count2 = list(posAnt2).count(pos1)
            if exist_count > 0 and exist_count2 > 0:
                print("Paralelo: Pos0:", pos0, "Pos1:", pos1)
            else:
                InI=InI-model.IlfromR[l]
                Gshunt+=np.real(model.Yb[pos1,pos0])+np.real(model.Ynet[pos1,pos0])
                Bshunt+=np.imag(model.Yb[pos1,pos0])+np.imag(model.Ynet[pos1,pos0])
                posAnt.append(pos0)
                posAnt2.append(pos1)
    if testcase!='FOP/testcases/5BARRAS.xlsx':
        for l in model.TRANSF:
            pos0=int(pyo.value(model.tpos[0,1]))
            pos1=int(pyo.value(model.tpos[1,1]))
            if pos0==n:
                InI=InI+model.ITfromR[l]
                posAnt.append(pos0)
                posAnt2.append(pos1)
            if pos1==n:
                InI=InI-model.ITfromR[l]
                posAnt.append(pos1)
                posAnt2.append(pos0)
    InI=InI+Gshunt*model.v[n]*pyo.cos(model.delta[n])-Bshunt*model.v[n]*pyo.sin(model.delta[n])
    return model.IBr[n]==InI

```

Figura 7.- Código correspondiente al balance nodal de corriente real.

Una vez encontrados los valores que se muestra en la figura 6 se obtiene las ecuaciones (2.12) y (2.13) empleando dos funciones para encontrar la corriente de barra real e imaginaria, respectivamente, en la figura 7 se puede observar el código correspondiente al desarrollo de la ecuación (2.12) con la que se obtiene el valor real de corriente en cada una de las barras del sistema, donde se emplea un lazo for que permite realizar la sumatoria de las corrientes de líneas y transformadores calculadas anteriormente, donde lo primero que se realiza es verificar la posición de las líneas para luego con el condicional if, else se compruebe que la línea pertenezca a una barra y, que no se vuelva a leer la misma línea dos veces, para de manera posterior en la variable *InI* ir almacenando y sumando los valores de corriente que pertenecen a esa barra, lo cual se realiza de la misma forma en caso de que exista transformadores dentro del sistema, finalmente en la variable *InI* se puede observar la formulación de la ecuación (2.12) que retorna en la variable *model.Ibr* con el valor real de la corriente de cada barra del sistema, de igual forma se desarrolla el código para obtener los valores imaginarios de la corriente en cada barra.

```

#EXPRESADO EN TÉRMINOS DE CORRIENTE
def Calculo_Flujo_Barra_P(model,n):
    vr=model.v[n]*pyo.cos(model.delta[n])
    vj=model.v[n]*pyo.sin(model.delta[n])
    ir=model.IBr[n]
    ij=model.IBj[n]
    P1=vr*ir
    P2=vj*ij
    return model.FLP[n]==(P1+P2)*model.baseMVA
model.Calculo_Flujo_Barra_P=pyo.Constraint(model.N, rule=Calculo_Flujo_Barra_P, doc = 'Flujo de potencia activa')
def Calculo_Flujo_Barra_Q(model,n):
    vr=model.v[n]*pyo.cos(model.delta[n])
    vj=model.v[n]*pyo.sin(model.delta[n])
    ir=model.IBr[n]
    ij=model.IBj[n]
    Q1=vj*ir
    Q2=vr*ij
    return model.FLQ[n]==(Q1-Q2)*model.baseMVA
model.Calculo_Flujo_Barra_Q=pyo.Constraint(model.N, rule=Calculo_Flujo_Barra_Q, doc = 'Flujo de potencia reactiva')

```

Figura 8.- Código del flujo de potencia en términos de corriente y voltaje.

Como se pueda observar en la figura 8 la potencia activa y reactiva se expresan en las variables *model.FLP* y *model.FLQ* respectivamente, donde se realiza la multiplicación del voltaje por la corriente que representan a los términos que restan a la potencia de generación de las ecuaciones (2.14) y (2.15), en el código los valores de voltaje se definen en forma rectangular, los valores de corrientes son los que se obtienen a partir de las ecuaciones (2.12) y (2.13) tanto su parte real como imaginaria y, posteriormente, se forman las ecuaciones de potencia, donde para la primera función correspondiente a la potencia activa se expresa como la suma de sus valores, mientras que para la potencia reactiva se expresan como la resta y, además, se intercambia en la multiplicación los valores del voltaje.

#### 2.1.4 OPF-AC con relajación de ecuaciones del flujo de potencia

Este modelo del OPF-AC se formula con las ecuaciones características del flujo de potencia de corriente alterna linealizadas cuyo desarrollo se muestra en la siguiente sección.

##### 2.1.4.1 Linealización de las ecuaciones del flujo de potencia [17].

Los flujos de potencia entre barras (*i,j*) se expresa de la siguiente manera:

$$P_{ij} = (V_i^2 - V_i * V_j * \cos(\delta_{ij})) * G_{ij} - V_i * V_j * B_{ij} \sin(\delta_{ij}) \quad \forall i, j \in N \quad (2.20)$$

$$Q_{ij} = -(V_i^2 - V_i * V_j * \cos(\delta_{ij})) * B_{ij} - V_i * V_j * G_{ij} \sin(\delta_{ij}) \quad \forall i, j \in N \quad (2.21)$$

Considerando que la diferencia entre los ángulos de las barras es pequeña cuando se expresan en radianes, y usando la expansión en series de Taylor de las funciones seno y coseno hasta el término cuadrático, se tiene:

$$\cos(\delta_i - \delta_j) \approx 1 - \frac{\delta_{ij}^2}{2} \quad (2.22)$$

$$\sin(\delta_i - \delta_j) \approx \delta_{ij} \quad (2.23)$$

Para desacoplar  $V$  y  $\delta$ , se asume que la magnitud del voltaje es cercana a 1 p.u ( $V \approx 1$  p.u):

$$V_i * V_j * \delta_{ij} \approx \delta_{ij} \quad (2.24)$$

$$V_i * V_j * \delta_{ij}^2 \approx \delta_{ij}^2 \quad (2.25)$$

Al reemplazar las expresiones de (2.22) a (2.26) en las ecuaciones (2.20) y (2.21) se obtiene las siguientes ecuaciones:

$$P_{ij} = G_{ij} * (V_i^2 - V_i * V_j) - B_{ij} * \delta_{ij} + \frac{1}{2} * G_{ij} * \delta_{ij}^2 \quad \forall i, j \in N \quad (2.26)$$

$$Q_{ij} = -B_{ij} * (V_i^2 - V_i * V_j) - G_{ij} * \delta_{ij} + \frac{1}{2} * (-B_{ij}) * \delta_{ij}^2 \quad \forall i, j \in N \quad (2.27)$$

Tomando en cuenta que  $V^2$  es una variable independiente, se emplea la siguiente transformación matemática para el término de la magnitud de voltaje no lineal ( $V_{ij} = V_i - V_j$ ):

$$V_i^2 - V_i * V_j = V_i^2 - \left( \frac{V_i^2 + V_j^2}{2} - \frac{V_{ij}^2}{2} \right) = \frac{V_i^2 - V_j^2}{2} + \frac{V_{ij}^2}{2} \quad (2.28)$$

Por tanto, aplicando la transformación matemática (2.28) en las ecuaciones (2.26) y (2.27) se tiene el modelo de la red linealizado con  $Q$  y  $V$  expresado de la siguiente manera:

$$P_{ij}^{\Lambda} = G_{ij} * \frac{V_i^2 - V_j^2}{2} - B_{ij} * \delta_{ij} + P_{ij}^L \quad (2.29)$$

$$Q_{ij}^{\Lambda} = -B_{ij} * \frac{V_i^2 - V_j^2}{2} - G_{ij} * \delta_{ij} + Q_{ij}^L \quad (2.30)$$

$$P_{ij}^L = \frac{1}{2} G_{ij} * (\delta_{ij}^2 + V_{ij}^2) \quad (2.31)$$

$$Q_{ij}^L = \frac{1}{2} (-B_{ij}) * (\delta_{ij}^2 + V_{ij}^2) \quad (2.32)$$

Donde:

$P_{ij}^{\Lambda}$  Aproximación lineal del flujo de potencia activa.

$Q_{ij}^{\Lambda}$  Aproximación lineal del flujo de potencia reactiva.

$P_{ij}^L$  Pérdidas de potencia activa en el elemento  $i - j$ .

$Q_{ij}^L$  Pérdidas de potencia reactiva en el elemento  $i - j$ .

Las ecuaciones (2.29) y (2.30) representan las ecuaciones del flujo de potencia linealizadas, a excepción de las ecuaciones (2.31) y (2.32) que representan las pérdidas donde se incluyen el efecto de las magnitudes de tensión.

Para expresar las ecuaciones de las pérdidas de tipo lineal se toma en consideración los valores de condiciones iniciales de  $V = V_0$  y  $\delta = \delta_0$ , la ecuación (2.31) de pérdidas de potencia se puede expresar en términos del ángulo y de la magnitud de voltaje como:

$$P_{ij}^L = \frac{1}{2} G_{ij} * \delta_{ij}^2 + \frac{1}{2} G_{ij} * V_{ij}^2 = P_{ij,\delta}^L + P_{ij,V}^L \quad (2.33)$$

Al aplicar el desarrollo de la serie de Taylor de primer orden, el término  $P_{ij,\delta}^L$  se linealiza de la siguiente manera:

$$P_{ij,\delta}^L = \frac{1}{2} G_{ij} * \delta_{ij}^2 \approx G_{ij} * \delta_{ij,0} * \delta_{ij} - \frac{1}{2} G_{ij} * \delta_{ij,0}^2 \quad (2.34)$$

En cuanto a  $P_{ij,V}^L$  al ser una función del voltaje, primero se aplica la transformación de  $V^2$  y luego se aplica el desarrollo de la serie de Taylor de primer orden:

$$\begin{aligned} P_{ij,V}^L &= \frac{1}{2} G_{ij} * V_{ij}^2 \approx \frac{G_{ij}}{2} * \frac{(V_i + V_j)^2}{(V_{i,0} + V_{j,0})^2} V_{ij}^2 = \frac{G_{ij}}{2} * \frac{(V_i^2 - V_j^2)^2}{(V_{i,0} + V_{j,0})^2} \\ &= G_{ij} * \frac{V_{i,0} - V_{j,0}}{V_{i,0} + V_{j,0}} (V_i^2 - V_j^2) - \frac{G_{ij}}{2} (V_{i,0} - V_{j,0})^2 \end{aligned} \quad (2.35)$$

Por tanto, las ecuaciones linealizadas de las pérdidas se expresan de la siguiente manera:

$$P_{ij}^L = (G_{ij} * \delta_{ij,0} * \delta_{ij} - \frac{1}{2} G_{ij} * \delta_{ij}^2) + (G_{ij} * \frac{V_{i,0} - V_{j,0}}{V_{i,0} + V_{j,0}} (V_i^2 - V_j^2) - \frac{G_{ij}}{2} (V_{i,0} - V_{j,0})^2) \quad (2.36)$$

$$Q_{ij}^L = ((-B_{ij}) * \delta_{ij,0} * \delta_{ij} - \frac{1}{2} (-B_{ij}) * \delta_{ij}^2) + ((-B_{ij}) * \frac{V_{i,0} - V_{j,0}}{V_{i,0} + V_{j,0}} (V_i^2 - V_j^2) - \frac{(-B_{ij})}{2} (V_{i,0} - V_{j,0})^2) \quad (2.37)$$

Finalmente, las ecuaciones del flujo de potencial por líneas de transmisión y transformadores de forma lineal con pérdidas se muestran a continuación [18]:

$$\begin{aligned} P_{ij}^A &= G_{ij} * \frac{V_i^2 - V_j^2}{2} - B_{ij} * \delta_{ij} + (G_{ij} * \delta_{ij,0} * \delta_{ij} - \frac{1}{2} G_{ij} * \delta_{ij}^2) + (G_{ij} * \frac{V_{i,0} - V_{j,0}}{V_{i,0} + V_{j,0}} (V_i^2 \\ &\quad - V_j^2) - \frac{G_{ij}}{2} (V_{i,0} - V_{j,0})^2) \end{aligned} \quad (2.38)$$

$$Q_{ij}^{\Lambda} = -B_{ij} * \frac{V_i^2 - V_j^2}{2} - G_{ij} * \delta_{ij} + ((-B_{ij}) * \delta_{ij,0} * \delta_{ij} - \frac{1}{2}(-B_{ij}) * \delta_{ij}^2) + ((-B_{ij}) * \frac{V_{i,0} - V_{j,0}}{V_{i,0} + V_{j,0}}(V_i^2 - V_j^2) - \frac{(-B_{ij})}{2}(V_{i,0} - V_{j,0})^2) \quad (2.39)$$

Donde:

$P_{ij}^{\Lambda}$  Aproximación lineal del flujo de potencia activa.

$Q_{ij}^{\Lambda}$  Aproximación lineal del flujo de potencia reactiva.

$V_{(i,j),0}$  Valor de voltaje de condiciones iniciales.

$\delta_{ij,0}$  Valor del ángulo de voltaje de condiciones iniciales.

Para este modelo las ecuaciones de la potencia activa y reactiva inyectadas en cada barra empleadas para el balance nodal de potencia se expresan de la siguiente manera [17]:

$$P_i = \sum_{(i,j) \in K} P_{ij}^{\Lambda} + \left( \sum_{j=1}^N G_{ij} \right) * V_i^2 \quad \forall i \in N \quad (2.40)$$

$$Q_i = \sum_{(i,j) \in K} Q_{ij}^{\Lambda} + \left( \sum_{j=1}^N -B_{ij} \right) * V_i^2 \quad \forall i \in N \quad (2.41)$$

Donde, los segundos términos de las ecuaciones (2.40) y (2.41), representan los flujos de potencia por los elementos shunt del sistema, son líneas dado a que  $V^2$  es considera una variable independiente [17].

A continuación, se presenta la implementación de este modelo para la resolución del OPF-AC con la relajación de las ecuaciones del flujo de potencia:

```

456     if MetResolucion==0:
457         def Calculo_Potencia_de_Linea(model,l):
458             Pos0=int(pyo.value(model.lpos[0,l]))
459             Pos1=int(pyo.value(model.lpos[1,l]))
460             Gij=np.real(model.Yb[Pos0,Pos1])
461             Bij=np.imag(model.Yb[Pos0,Pos1])
462             Vi=model.v[Pos0]
463             Vj=model.v[Pos1]
464             Vi0=1
465             Vj0=1
466             Deltaij=model.delta[Pos0]-model.delta[Pos1]
467             Deltaij0=0
468             Vij=Vi-Vj
469             PijL=((Gij*Deltaij0*Deltaij-0.5*Gij*Deltaij**2)+(Gij*(Vi0-Vj0/Vi0+Vj0)*(Vi**2-Vj**2))-0.5*Gij*(Vi0-Vj0)**2)
470             return model.Pij[l]==(Gij*((Vi**2-Vj**2)/2)-Bij*Deltaij+PijL)*model.baseMVA
471
472         def Calculo_Potencia_de_Linea2(model,l):
473             Pos0=int(pyo.value(model.lpos[0,l]))
474             Pos1=int(pyo.value(model.lpos[1,l]))
475             Gji=np.real(model.Yb[Pos1,Pos0])
476             Bji=np.imag(model.Yb[Pos1,Pos0])
477             Vi=model.v[Pos1]
478             Vj=model.v[Pos0]
479             Vi0=1
480             Vj0=1
481             Deltaji=model.delta[Pos1]-model.delta[Pos0]
482             Deltaji0=0
483             Vji=Vj-Vi
484             PjiL=(Gji*Deltaji0*Deltaji-0.5*Gji*Deltaji**2)+Gji*(Vi0-Vj0/Vi0+Vj0)*(Vi**2-Vj**2)-0.5*Gji*(Vi0-Vj0)**2
485             return model.Pji[l]==(Gji*((Vi**2-Vj**2)/2)-Bji*Deltaji+PjiL)*model.baseMVA

```

Figura 9.- Código desarrollado de las ecuaciones linealizadas del flujo de potencia por líneas de transmisión.

En la figura 9 se muestra la implementación de dos funciones, para desarrollar las ecuaciones (2.38) y (2.39), donde la primera realiza el cálculo del flujo de potencia que va desde la barra  $i$  hacia la barra  $j$  y, la segundo calcula el flujo de manera inversa. Como se puede observar se empieza definiendo cada una de las variables de voltajes, susceptancias, conductancias para formar las ecuaciones, donde la primero que se realiza es la formulación de ecuación (2.36) en la línea 469 que representa a las pérdidas de potencia y, posteriormente, se completa la ecuación en la variable  $model.Pij$  obteniendo así los valores del flujo de potencia activa por las líneas de transmisión que conforman el sistema de estudio. De igual manera se plantea las funciones para realizar el cálculo para transformadores.

```

# LINEALIZADAS (RELAJACIÓN)
if MetResolucion==0:
    def Calculo_Potencia_Barra(model,n):
        sumaG=0
        sumPij=0
        for l in model.L:
            Pos0=int(pyo.value(model.lpos[0,l]))
            Pos1=int(pyo.value(model.lpos[1,l]))
            if Pos0==n or Pos1==n:
                sumPij+=(model.Pij[l]+model.Pji[l])
        if testcase!='FOP/testcases/SBARRAS.xlsx':
            for t in model.TRANSF:
                sumPij+=(model.PijT[t]+model.PjiT[t])
        for j in model.N:
            sumaG+=(np.real(model.Yb[n,j]))
        return model.FLP[n]==(model.v[n]*model.v[n]*sumaG*model.baseMVA+sumPij)
    model.Calculo_Potencia_Barra=pyo.Constraint(model.N, rule=Calculo_Potencia_Barra, doc = 'Flujo de potencia activa')
    def Calculo_Potencia_BarraQ(model,n):
        sumaB=0
        sumQij=0
        for l in model.L:
            Pos0=int(pyo.value(model.lpos[0,l]))
            Pos1=int(pyo.value(model.lpos[1,l]))
            if Pos0==n or Pos1==n:
                sumQij+=model.Qij[l]+model.Qji[l]
        if testcase!='FOP/testcases/SBARRAS.xlsx':
            for t in model.TRANSF:
                Pos0=int(pyo.value(model.lpos[0,l]))
                Pos1=int(pyo.value(model.lpos[1,l]))
                if Pos0==n or Pos1==n:
                    sumQij+=model.QijT[t]+model.QjiT[t]
        for j in model.N:
            sumaB+=(-1*np.imag(model.Yb[n,j]))
        return model.FLQ[n]==(model.v[n]*model.v[n]*sumaB*model.baseMVA+sumQij)
    model.Calculo_Potencia_BarraQ=pyo.Constraint(model.N, rule=Calculo_Potencia_BarraQ, doc = 'Flujo de potencia reactiva')

```

Figura 10.- Código implementado para las potencias inyectas en cada barra con la relajación de las ecuaciones del flujo de potencia.

La implantación de las ecuaciones (2.40) y (2.41) dentro del programa, se muestra en la figura 10 donde se formula dos funciones para encontrar los valores tanto de la potencia activa y reactiva inyectadas en cada barra del sistema, para lo cual se hace uso de un lazo *for* que permite identifica las posición de cada una de las líneas del sistema para realizar el sumatorio de las potencias de líneas y transformadores que llegan a cada barra junto con el segundo término de cada ecuación que se expresa en la variable *suma*, para de manera posterior formar la ecuación total en la variable *model.FLP*, para el caso de la potencia activa, obteniendo de esa manera los valores totales de la potencia activa y reactiva inyectada en cada una de las barras que luego se emplean en el balance nodal de potencia.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

#### 3.1 Resultados

Para analizar y verificar que la herramienta computación desarrollada funciona de manera correcta, se aplica a dos casos de estudio, el sistema IEEE de 14 barras y el Sistema Nacional Interconectado (SNI) ecuatoriano.

##### 3.1.1 Sistema IEEE de 14 barras

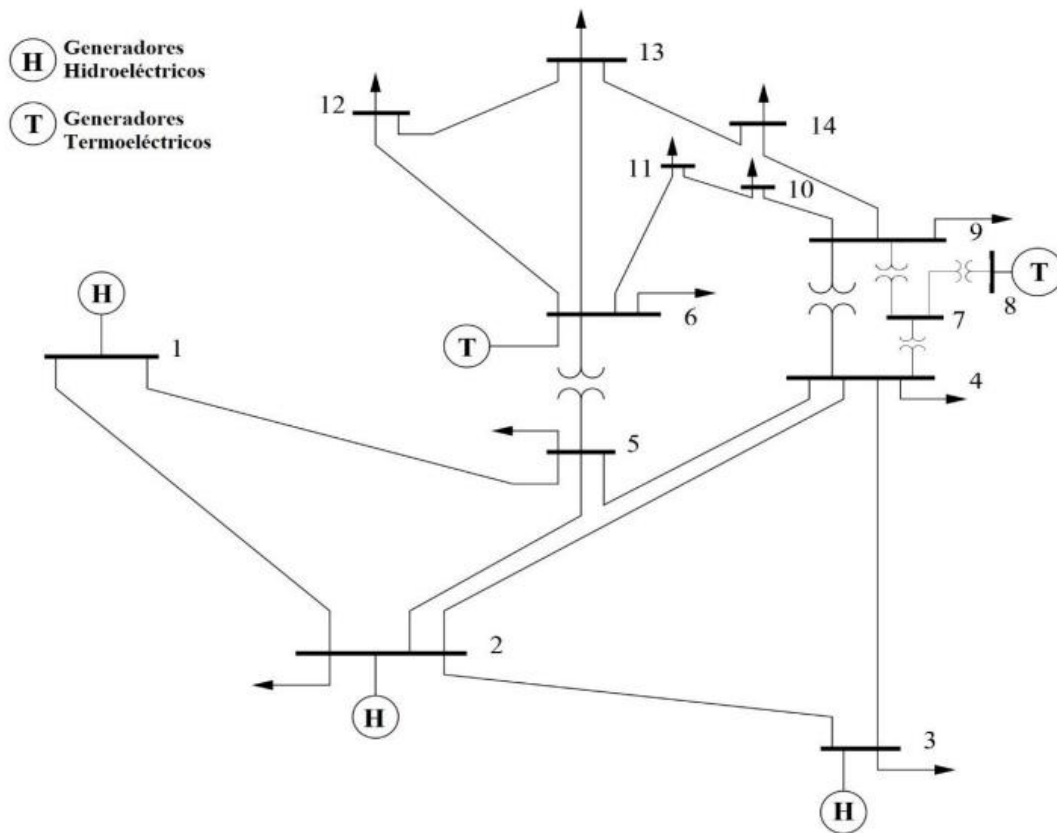


Figura 11.- Modelo del sistema de prueba IEEE de 14 barras [19].

Los parámetros de entrada del sistema se presentan en el Anexo II. A continuación, se presentan los resultados obtenidos seleccionando las siguientes opciones en el programa:

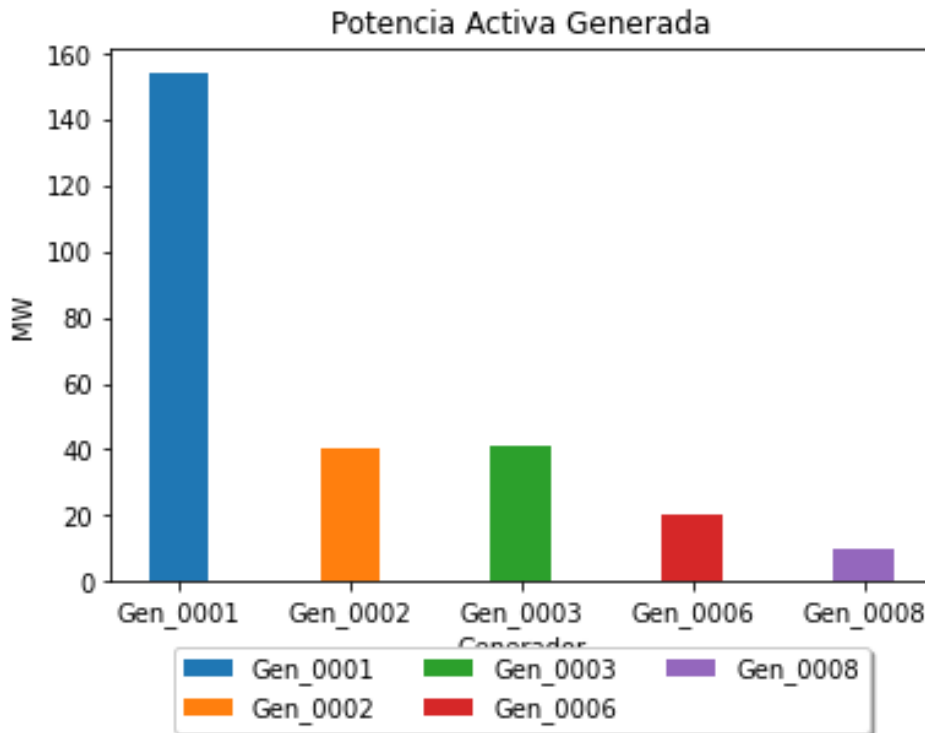
- Función objetivo: Cuadrática.
- Método de resolución: OPF-AC expresando las ecuaciones del flujo de potencia en coordenadas polares.



**Tabla 3. 1** Resultados del OPF-AC polar.

<b>Dato</b>	<b>Valor</b>
<b>Costo total del sistema</b>	8178.87 [USD/h]
<b>Potencia Generada</b>	265.41 [MW]
<b>Demanda</b>	259 [MW]

En la tabla 3.1 se muestran los resultados obtenidos al ejecutar el caso del OPF-AC con las restricciones del flujo de potencia expresado de forma polar, donde el sistema genera un total de 265.41 [MW] para satisfacer la demanda y su costo de generación es igual a 8178.87 [USD/h]. Además, se tiene que la potencia de generación es mayor a la demanda, ya que el sistema considera pérdidas, cumpliendo de manera correcta con la restricción del balance nodal de potencia.



*Figura 12.-* Potencia de generación.

En la figura 12 se muestra el aporte de los 5 generadores que forman parte del sistema, los generadores hidroeléctricos son los encargados de cubrir en mayor porcentaje a la demanda, debido a que su costo de generación es menor, donde el que mayor potencia entrega es el generador conectado a la primera barra (Gen\_001) con un total de 154.02 [MW] abasteciendo el 59.46% del total de la demanda. Mientras que los generadores termoeléctricos son los que menor potencia aportan al sistema esto debido a que su costo de generación es elevado.

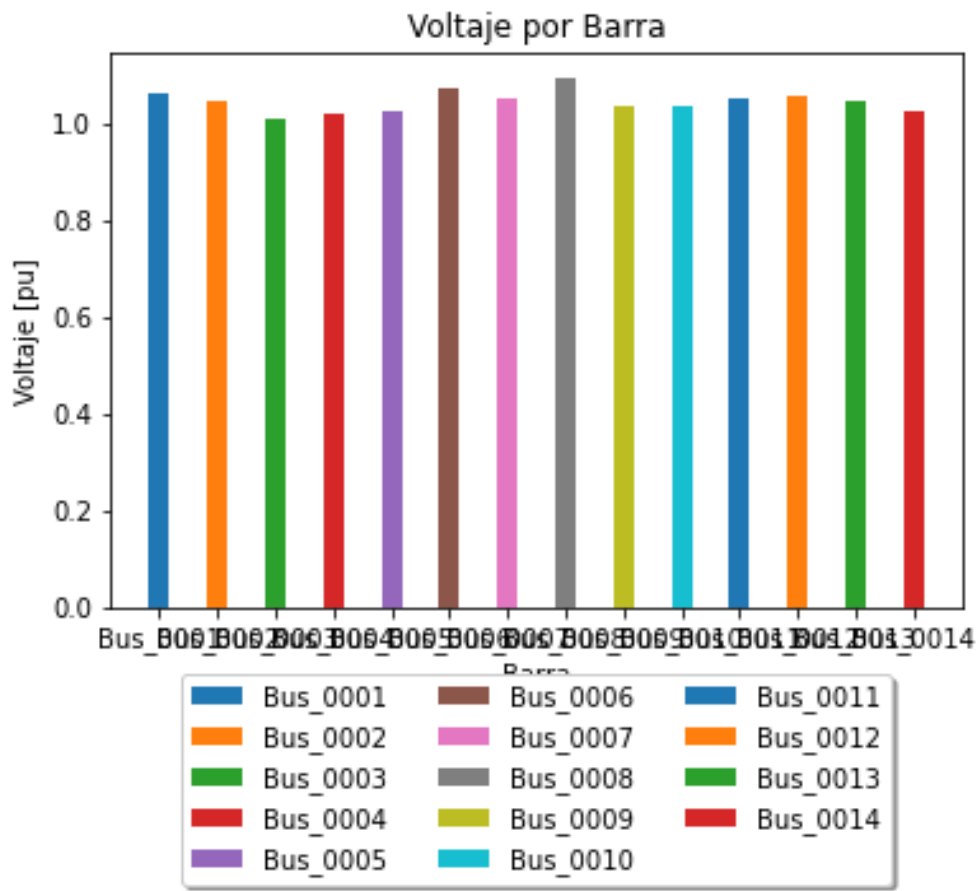


Figura 13.- Valores del voltaje en cada barra del sistema.

Como se observa en la figura 13 los voltajes en cada barra toman valores mayores al 1,0 p.u, cumpliendo de manera correcto los límites de voltaje que para este sistema deben estar entre 0,94 a 1,06 p.u.

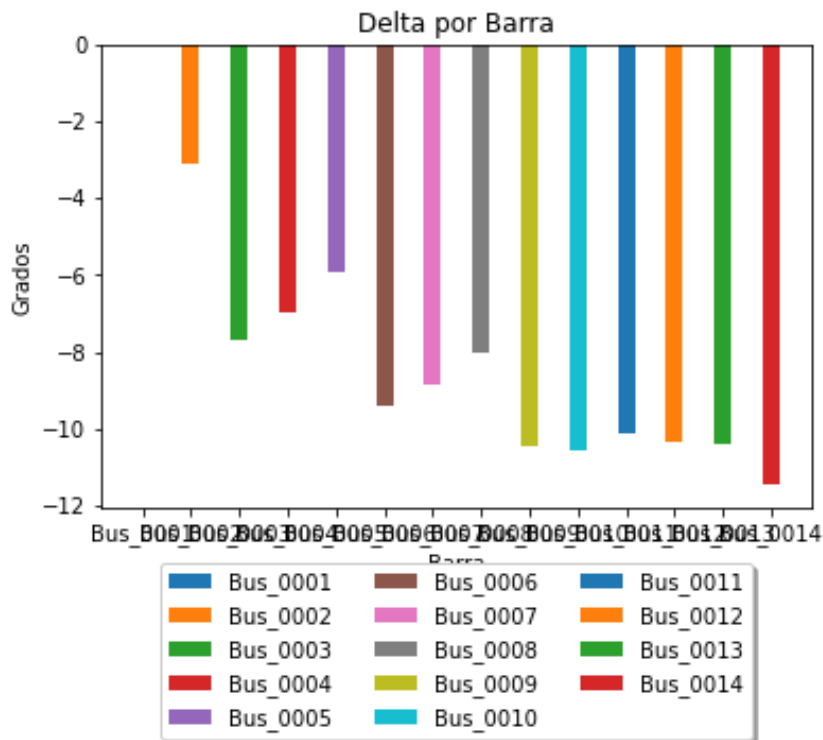


Figura 14.- Resultados de los ángulos de voltaje en cada barra.

En la figura 14 se presenta los valores que toman los ángulos de los voltajes por barra, donde se muestra que la barra 1 toma un valor del ángulo igual a cero ya que esta es considerada como la barra slack del sistema, mientras que las demás barras toman valores de ángulos no tan elevados que permiten tener una operación segura del sistema.

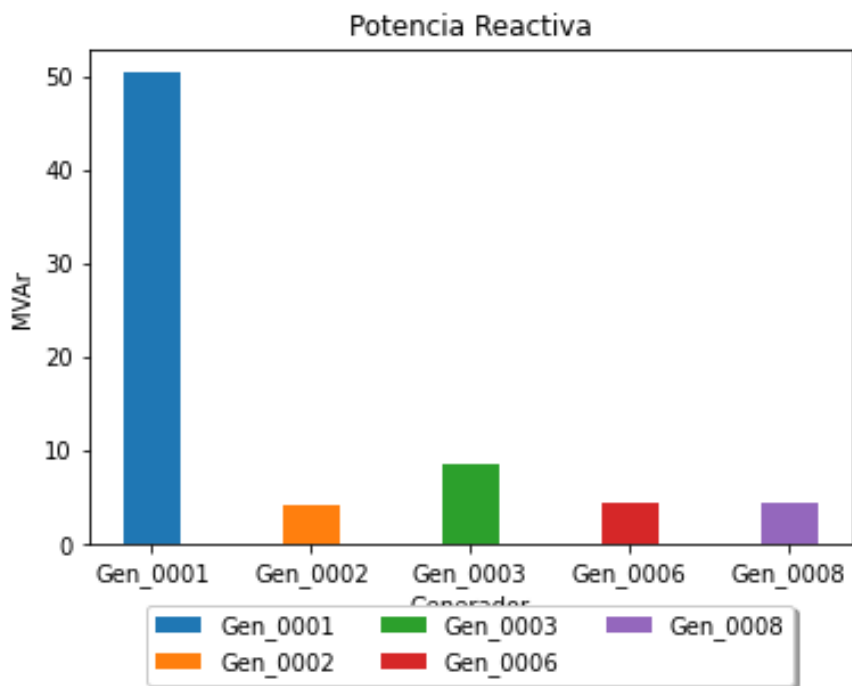


Figura 15.- Valores de potencia reactiva de cada generador.

La figura 15 muestra la potencia reactiva producida por los generadores al sistema, donde el valor más alto de reactivos es entregado por el generador conectado a la barra 1, y en sí esta potencia es la cargada de generar pérdidas al sistema ya que el programa realiza un flujo de potencia de corriente alterna.

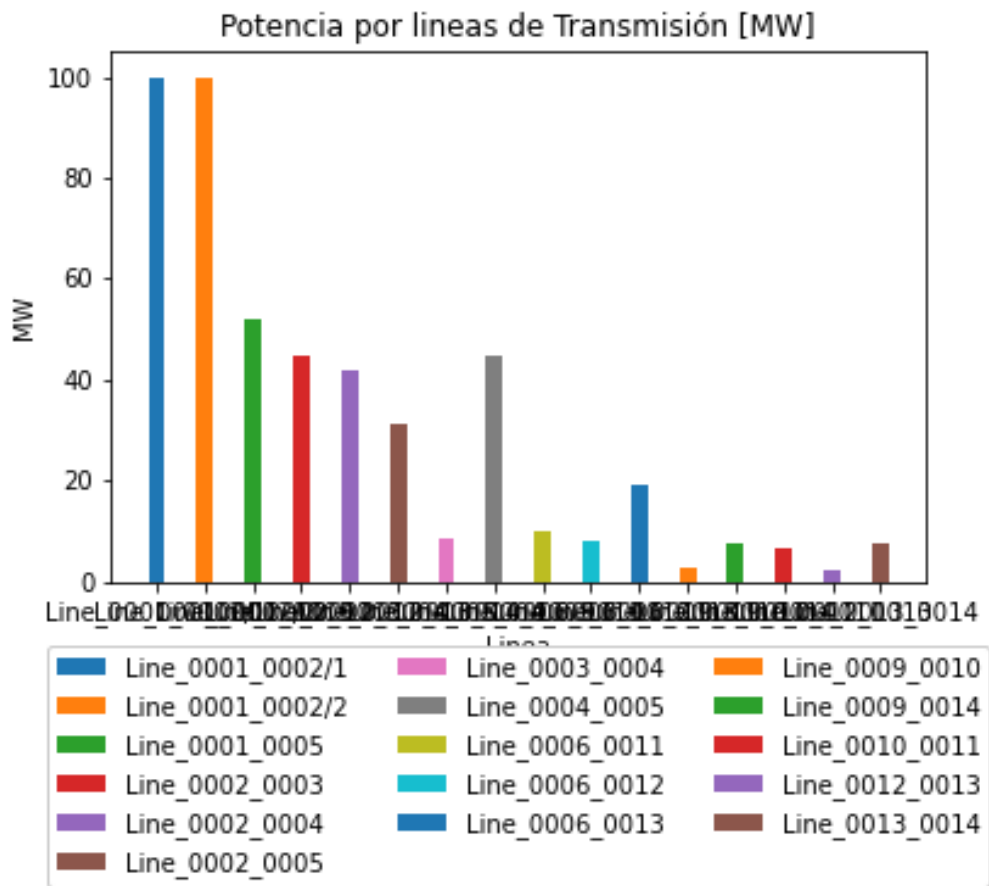


Figura 16.- Valores del flujo de potencia por las líneas del sistema.

En la figura 16 se puede observar los valores de potencia que fluye a través de cada línea de transmisión (LT) del sistema, donde el flujo de potencia por las líneas que parten de la barra 1 son las de mayor valor debido a que el generador conectado a esa barra es el que mayor potencia entrega por ende deber fluir una mayor potencia por esas líneas para que ayuden al transporte del total de la potencia para lograr a satisfacer la demanda y que el sistema trabaje en condiciones seguras.

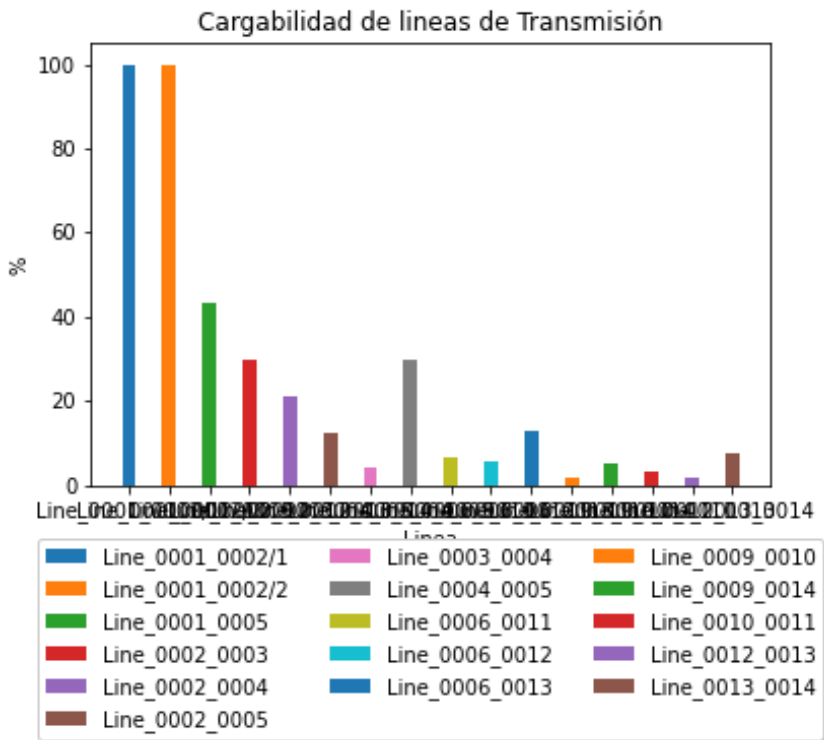


Figura 17.- Cargabilidad de las líneas de transmisión del sistema.

En cuanto a la cargabilidad de las LT del sistema, se muestra en la figura 17 que por las líneas que parten de la barra 1 se tiene que el flujo de potencia alcanza la capacidad máxima de transporte de las líneas, mientras que las demás líneas se encuentran por debajo del 50% de su potencia máxima logrando así cumplir con los límites de operación.

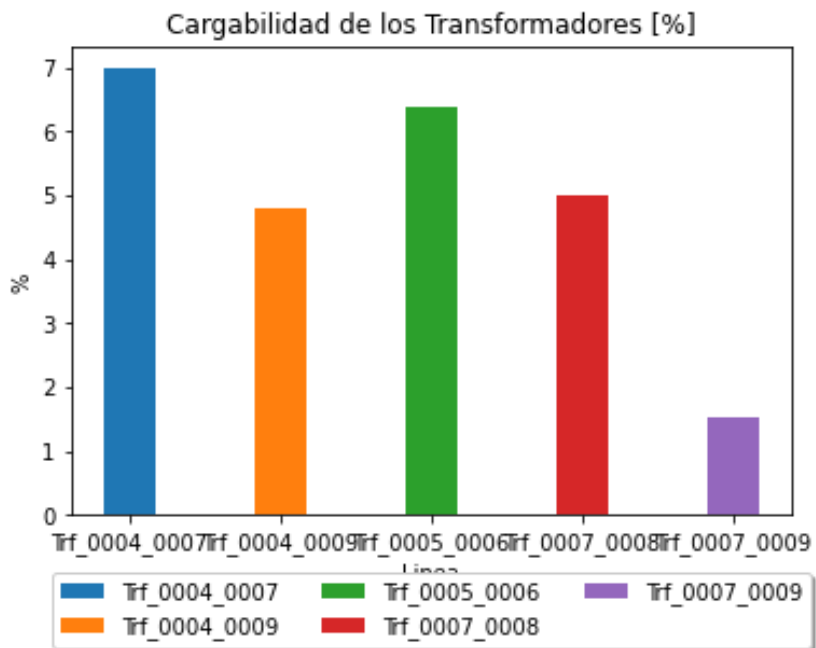


Figura 18.- Porcentaje de cargabilidad de los transformadores del sistema.

En la figura 18 se puede observar los resultados del porcentaje de cargabilidad de los 5 transformadores que componen el sistema, donde se muestra que los transformadores no se encuentran sobrecargados y operan en valores menores a su capacidad máxima cumpliendo así un operación confiable dentro de sus límites.

A continuación, se presentan los resultados obtenidos empleando cada una de las formulaciones matemáticas del OPF-AC:

**Tabla 3. 2** Resultados con función de costos cuadrática.

<b>Datos</b>	<b>Función de Costos Cuadrática</b>			
	<b>Balance de Potencia</b>			<b>Balance de Corriente</b>
<b>Restricciones de Flujos de potencia</b>	<b>Coordenadas polares</b>	<b>Coordenadas rectangulares</b>	<b>Linealización (Relajación)</b>	<b>Coordenadas rectangulares</b>
<b>Costo Total [USD/h]</b>	8178.87	8178.87	8401.13	8206.54
<b>Potencia de Generación [MW]</b>	265.41	265.41	270.417	266.242
<b>Demanda [MW]</b>	259	259	259	259
<b>Tiempo de ejecución [seg]</b>	0.103	0.110	0.124	0.135

En la tabla 3.2 se muestran los resultados del costo de generación, potencia de generación y el tiempo de ejecución que se obtiene con la herramienta computacional empleando la función de costos cuadrática en los diferentes modelos del OPF-AC, donde para los dos primeros casos cuanto se trabaja con balance nodal de potencia la diferencia está en el tiempo que se demora el programa para obtener una solución óptima debido a la formulación de las ecuaciones del flujo de potencia dando un costo de 8178.87 [USD/h] para un potencia generada de 265.41 [MW], en cuanto al modelo con la relajación de las ecuaciones del flujo de potencia se presenta una mayor potencia de generación por lo cual el costo aumenta. Para el caso donde se emplea balance nodal de corriente el tiempo de ejecución es mayor en comparación con otros casos debido al cálculo de las corrientes y,

además, presenta un costo mayor a los casos bases ya que la potencia de generación aumenta tomando un valor de 8206.54 [MW].

**Tabla 3. 3** Resultados con función de costos lineal.

<b>Datos</b>	<b>Función de Costos Linealizada</b>			
	<b>Balance de Potencia</b>			<b>Balance de Corriente</b>
<b>Restricciones de Flujos de potencia</b>	<b>Coordenadas polares</b>	<b>Coordenadas rectangulares</b>	<b>Linealización (Relajación)</b>	<b>Coordenadas rectangulares</b>
<b>Costo Total [USD/h]</b>	6391.53	6391.53	6450.31	6405.38
<b>Potencia de Generación [MW]</b>	264.56	264.56	267.5	265.253
<b>Demanda [MW]</b>	259	259	259	259
<b>Tiempo de ejecución [seg]</b>	0.079	0.080	0.090	0.104

Los resultados que se muestran en la tabla 3.3 son los obtenidos haciendo uso de la función de costos linealizada representada en 50 segmentos, para el caso donde se expresa de forma diferente a las ecuaciones del flujo de potencia se tiene un costo de 6391.53 [USD/h] para una generación total de 264.56 [MW]. El modelo que realiza los cálculos de corriente del sistema para realizar el balance toma un tiempo de 0.104 [seg] para encontrar una solución óptima siendo el mayor en comparación a los demás casos al igual que cuando se expresa la función de costos de forma cuadrática, y la generación de potencia es de 265.253 [MW] dando un costo total igual a 6405.38 [USD/h]. En comparación con los resultados obtenidos en la tabla 3.2 se puede observar que el costo de generación del sistema disminuye producto de la linealización de la función objetivo y, además, al emplear menos recursos para resolver el problema el tiempo de ejecución de igual manera disminuye obteniendo de manera más sencilla la solución del OPF-AC para cualquiera de sus formulaciones matemáticas.

### 3.1.2 Sistema Nacional Interconectado (SNI)

El SNI ecuatoriano es un sistema con un gran número de generados, líneas de transmisión y elementos, por tanto, la resolución del problema OPF-AC se vuelve más complejo. A continuación, se presentan los resultados obtenidos considerando dos escenarios: Seco y Lluvioso

#### 3.1.2.1 Escenario Lluvioso

El SNI cuenta con un gran número de centrales hidroeléctricas que para esta temporada van a contar con un gran caudal que permitirá trabajar a sus diferentes unidades con su capacidad máxima ayudando a cubrir la demanda de manera adecuada. En cuanto a la demanda se tiene varias cargas conectadas a una misma barra, por lo cual para ahorrar recursos en la programación se las modela en su valor total dentro de una sola carga, donde la demanda total del sistema es de 3210.794 [MW].

Para este caso se selecciona las siguientes opciones dentro del programa:

- Función de costos cuadrática
- Método de resolución: OPF-AC con las restricciones del flujo de potencia expresadas de forma polar.

**Tabla 3. 4** Resultados obtenidos del OPF-AC con balance de corriente nodal.

<b>Dato</b>	<b>Valor</b>
<b>Costo total del sistema</b>	12712.87 [USD/h]
<b>Potencia Generada</b>	3361.337 [MW]
<b>Demanda</b>	3210.794 [MW]

La potencia total entregada por las unidades que confirman el SNI para este caso es de 3361.337 [MW], con lo cual se cubre adecuadamente a la demanda, dando un costo total de generación igual a 12712.87 [USD/h]. A continuación se presentan las gráficas obtenidas al aplicar este caso al SNI con sus respectivas descripciones que ayudaran a comprender la operación del sistema:



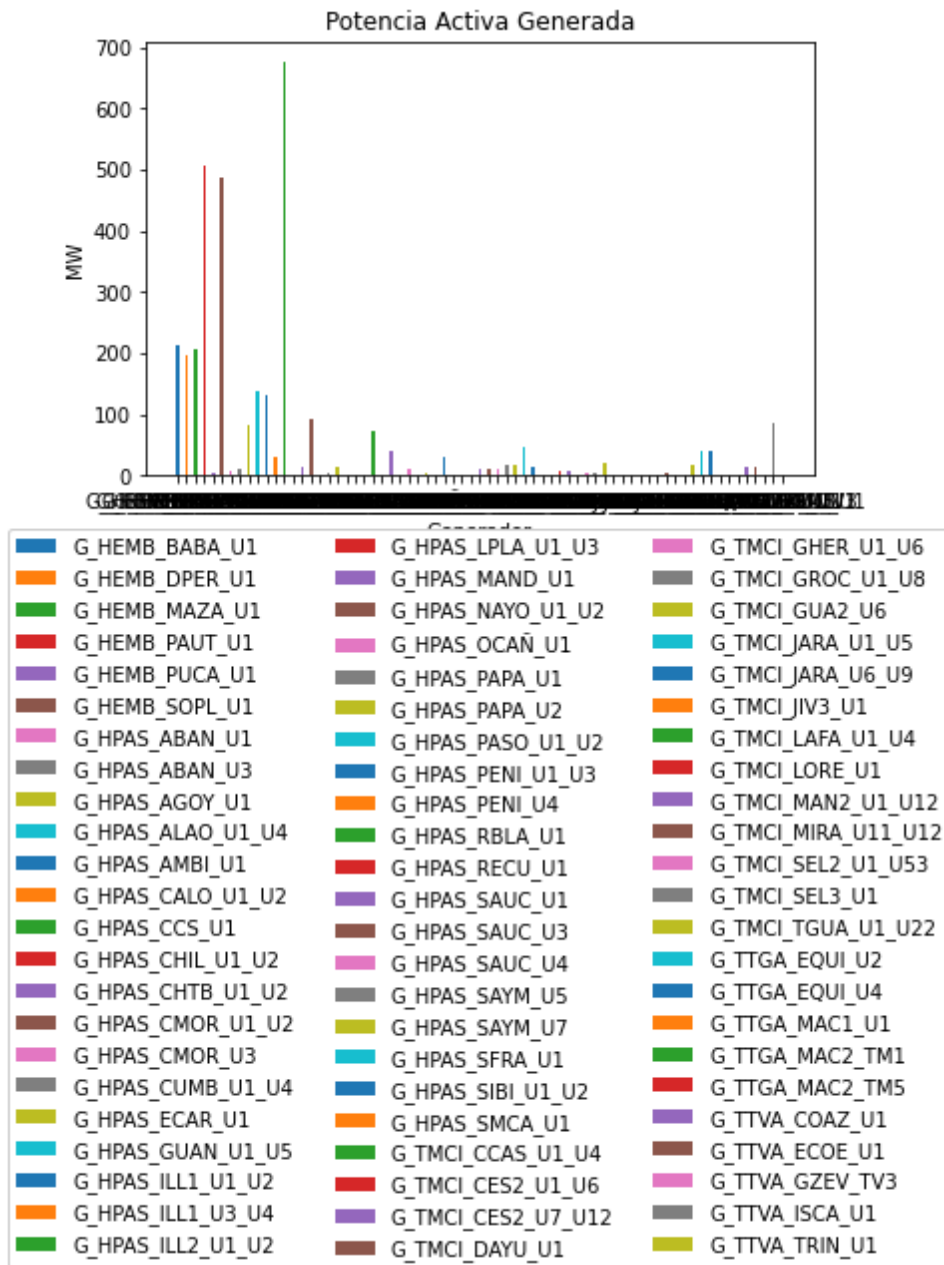


Figura 19.- Potencia entregada por cada generador del SIN.

En la figura 19 se muestra el aporte de cada generador al sistema para lograr abastecer la demanda donde el despacho total de generación es mayor al valor de la demanda. Como se puede observar las centrales hidroeléctricas son las de mayor participación, destacándose de manera clara el aporte de Coco Codo Sinclair y Paute.

La central de mayor generación es Coco Codo Sinclair con un aporte de 674.95 [MW], seguida de Paute-Molino con 506.54 [MW] y Sopladora con una potencia de 487.35 [MW], está última despachada al máximo de su capacidad.

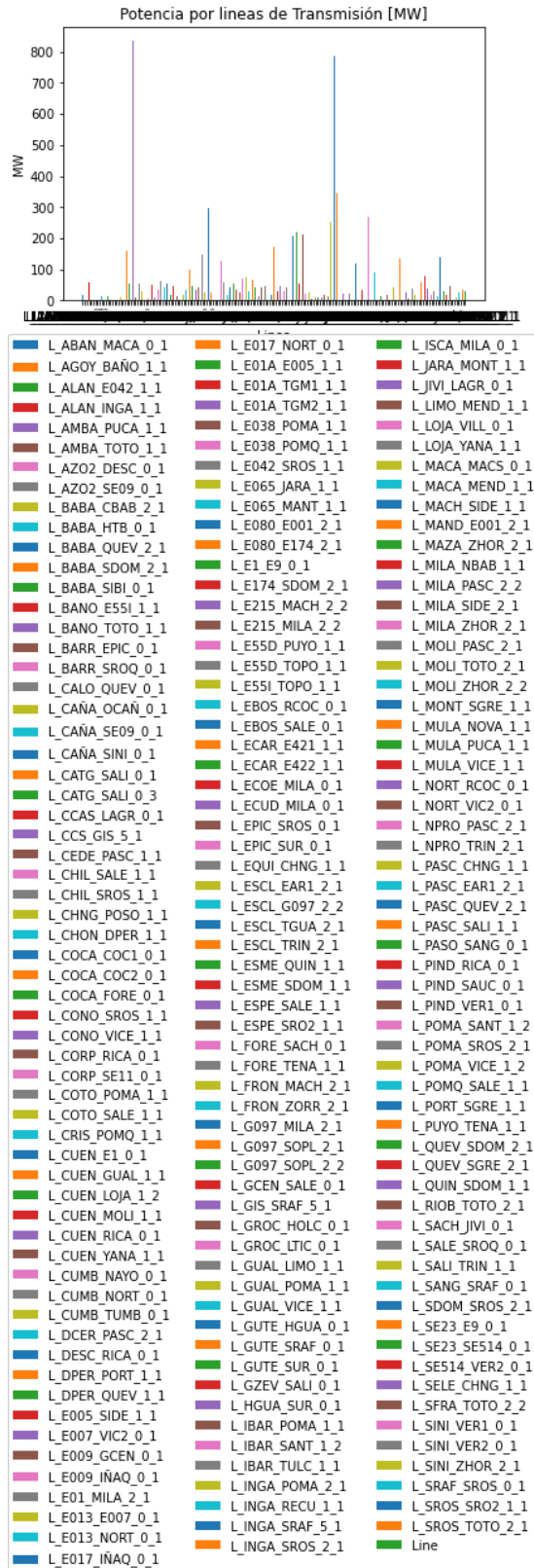


Figura 20.- Flujo de potencia por líneas de transmisión del sistema.

La potencia que fluye a través de las líneas de transmisión (L/T) se muestra en la figura 20, donde se destaca la L/T Coca Codo Sinclair – San Rafael y L/T San Rafael – El Inga

con un flujo de 838.66 [MW] y 836.8 [MW] respetivamente, encargadas de transportar la potencia generada por Coca Codo Sinclair. En cuanto a la cargabilidad de las líneas, todas se encuentran trabajando dentro de sus límites, es decir, no se tiene líneas con sobrecargas.

**Tabla 3. 5 Resultados del SNI en temporada lluviosa.**

<b>Datos</b>	<b>Función de Costos Cuadrática</b>			
	<b>Balance de Potencia</b>			<b>Balance de Corriente</b>
<b>Restricciones de Flujos de potencia</b>	<b>Coordenadas polares</b>	<b>Coordenadas rectangulares</b>	<b>Linealización (Relajación)</b>	<b>Coordenadas rectangulares</b>
<b>Costo Total [USD/h]</b>	12712.87	12712.87	13224.43	12771.65
<b>Potencia de Generación [MW]</b>	3361.337	3361.337	3528.588	3360.958
<b>Demanda [MW]</b>	3210.794	3210.794	3210.794	3210.794
<b>Tiempo de ejecución [seg]</b>	60.59	65.27	58.18	69.98

En la tabla 3.5 se muestran los resultados obtenidos con las diferentes formulaciones del OPF-AC, donde se observa que en las formulaciones en forma polar y rectangular, los costos y potencia generada son los mismos; la diferencia que existe entre estos dos métodos es el tiempo de ejecución donde las ecuaciones del flujo de potencia en forma rectangular toma más tiempo, que este caso es de 65.27 [seg]. En cuanto a los resultados del OPF-AC expresado con el balance de corriente presenta resultados diferentes ya que se tiene una mayor generación de potencia lo que eleva el costo total de generación; y además, debido a que se realizan cálculos de corrientes, encontrar la solución toma un tiempo de 69.98 [seg]. Sin embargo, como se observa los resultados presentan valores cercanos con lo cual se garantiza un correcto funcionamiento de la herramienta computacional desarrollada.

### 3.1.2.2 Escenario Seco.

En cuanto a la temporada seca, las centrales hidroeléctricas resultan afectadas ya que no tienen la suficiente cantidad de agua para la producción de energía eléctrica, siendo principalmente perjudicadas las de embalse debido a que la disminución del caudal no permite aprovechar la capacidad máxima de sus unidades, quedando algunas fuera de servicio. En cuanto a la demanda total del sistema es igual que el caso anterior.

Para este escenario se realizan ciertas modificaciones en el parque de generación: se deja fuera de servicio a 3 unidades de Paute, una unidad San Francisco y una unidad de Agoyán.

Para este caso se selecciona las siguientes opciones dentro del programa:

- Función de costos cuadrática
- Método de resolución: OPF-AC con balance nodal de corriente.

**Tabla 3. 6** Resultados obtenidos del OPF-AC con balance de corriente nodal.

<b>Dato</b>	<b>Valor</b>
<b>Costo total del sistema</b>	13147.91 [USD/h]
<b>Potencia Generada</b>	3369.107 [MW]
<b>Demanda</b>	3210.794 [MW]

Para la temporada seca, considerando la formulación del OPF-AC con la restricción del balance en términos de corriente, se tiene un despacho de potencia igual a 3369.107 [MW] dando un costo total de generación igual a 13147.91 [USD/h] que en comparación con el caso presentado en la sección anterior se tiene un mayor precio como consecuencia de un aporte mayor de potencia por parte de las centrales térmicas.

A continuación se presentan las gráficas obtenidas al aplicar este caso al SNI:

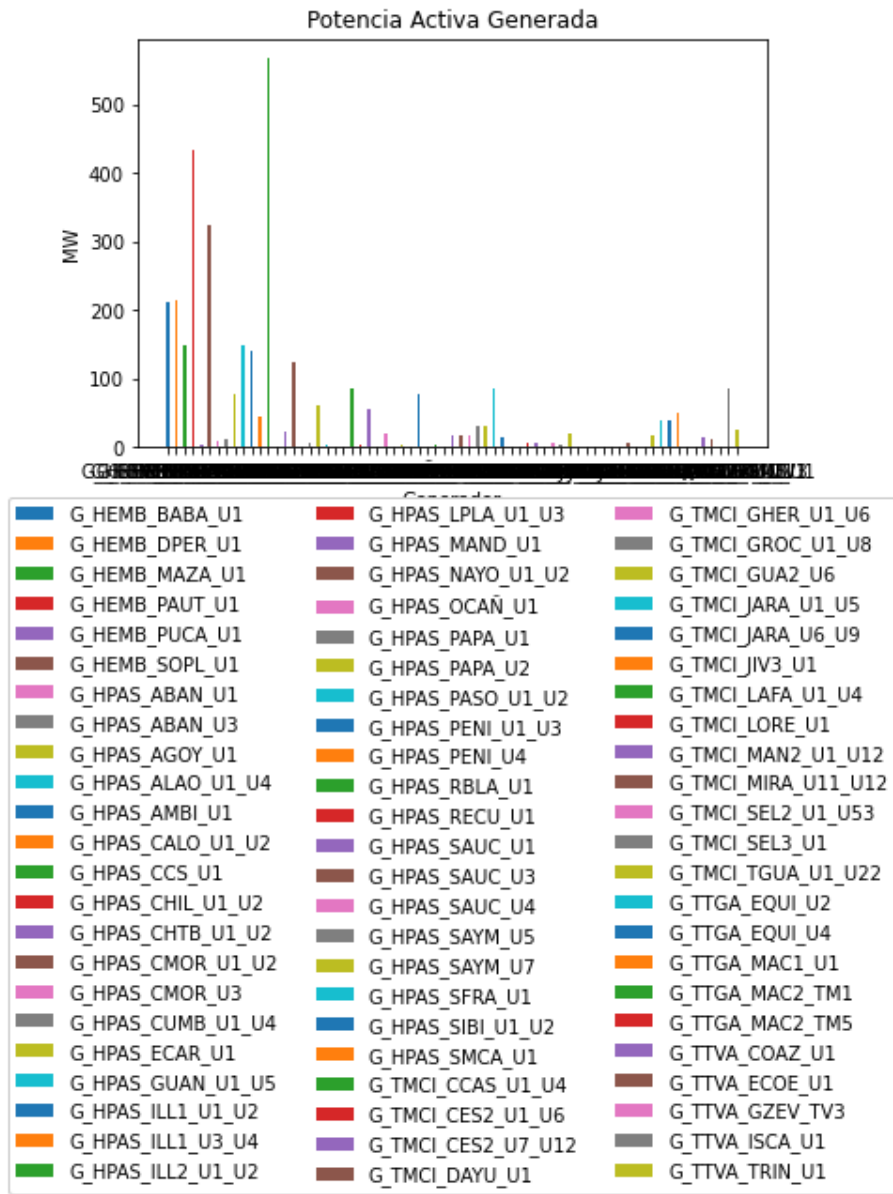


Figura 21.- Potencia de generación por cada unidad del SNI.

En la figura 21 se muestra el despacho de potencia, donde las centrales Coca Codo Sinclair, Paute Molino y Sopladora continúan siendo las que mayor potencia entregan, cubriendo alrededor del 48% de la demanda, esto debido a que el principal objetivo del OPF-AC es minimizar el costo de operación sistema y por tanto, se debe aprovechar toda la capacidad de las centrales que presentan menor costo de generación. Sin embargo, para este caso se puede observar una mayor participación de las centrales térmicas para cubrir el porcentaje restante de la demanda.

El despacho de potencia de las principales centrales para este escenario son: 567.54 [MW] por parte de Coca Codo Sinclair, 324 [MW] y 435.03 [MW] de Sopladora y Paute-Molino, respectivamente. Al comparar los valores de potencia obtenidos en la temporada lluviosa

se presenta una claramente disminución en el aporte de potencia pero de igual forma se logra cumplir con la demanda empleando un despacho mayor por parte de las otras centrales que forman parte del sistema.

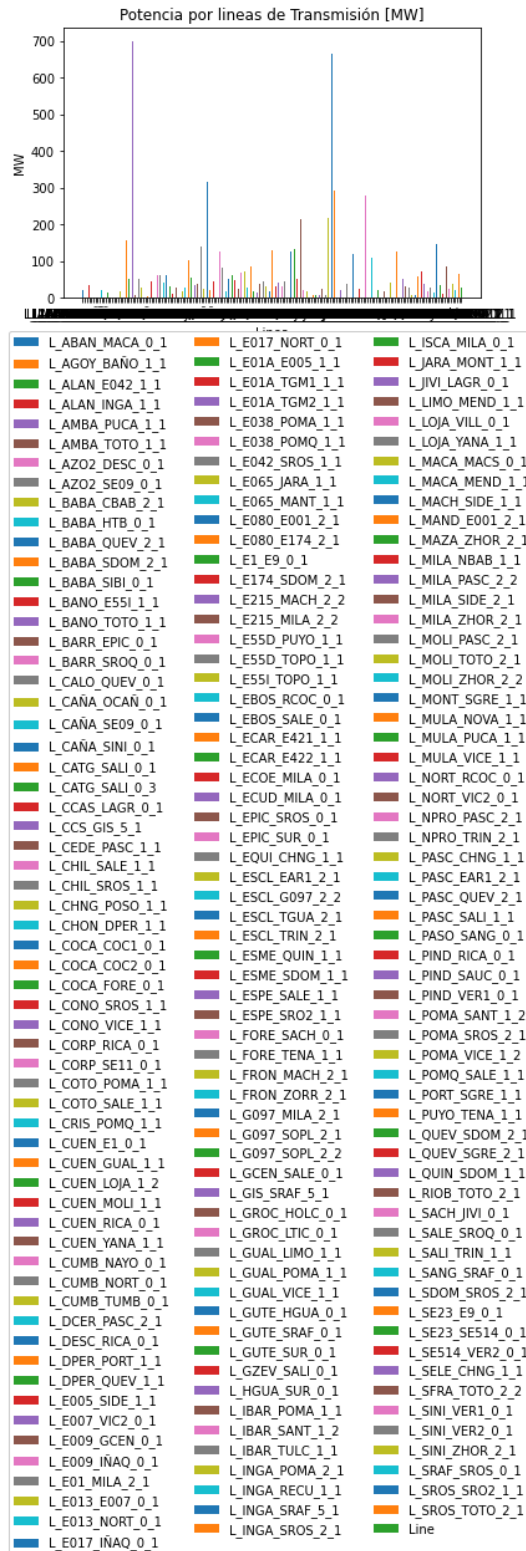


Figura 22.- Resultados del flujo de potencia por L/Ts del SNI.

De igual forma que en el caso anterior las líneas de transmisión que se destacan en la figura 22 son las correspondientes a las L/Ts que forman parte del sistema de transmisión de extra alta tensión de 500 [kV] como lo son Coca Codo Sinclair – San Rafael y San Rafael – El Inga que presentan un flujo similar mayor a 600 [MW] que son las encargadas de transportar y aprovechar de la mejor manera el total de la potencia generada por la central hidroeléctrica Coca Codo Sinclair. En cuanto a las cargabilidad de las líneas ninguna se encuentra trabajando por encima de su capacidad máxima cumpliendo de forma correcta con sus límites permitiendo encontrar una solución óptima para el sistema.

**Tabla 3. 7** Resultados del SNI en temporada seca.

<b>Datos</b>	<b>Función de Costos Cuadrática</b>			
	<b>Balance de Potencia</b>			<b>Balance de Corriente</b>
<b>Restricciones de Flujos de potencia</b>	<b>Coordenadas polares</b>	<b>Coordenadas rectangulares</b>	<b>Linealización (Relajación)</b>	<b>Coordenadas rectangulares</b>
<b>Costo Total [USD/h]</b>	13122.96	13122.96	13670.66	13147.91
<b>Potencia de Generación [MW]</b>	3356.631	3356.631	3397.798	3369.107
<b>Demanda [MW]</b>	3210.794	3210.794	3210.794	3210.794
<b>Tiempo de ejecución [seg]</b>	60.43	63.73	57.93	71.20

En la tabla 3.7 se presentan los resultados de los diferentes modelos del OPF-AC aplicado al SNI en temporada seca, donde el costo de generación es similar para los casos donde se modifica la forma de expresar el flujo de potencia tomando un valor de 13122.96 [USD/h] cuando el sistema genera una potencia de 3356.631 [MW] y, además, la diferencia en ambos casos se nota en el tiempo que tarda el programa en encontrar una solución óptima. Sin embargo, el modelo que emplea más tiempo para resolver el problema es cuando se selecciona la opción del balance de corriente con un tiempo de 71.20 [seg]. En comparación con los resultados obtenidos con hidrología lluviosa se puede observar un

aumento del costo de generación del sistema como producto de una participación mayor de las centrales térmicas en el aporte de potencia para satisfacer a la demanda.

Cabe destacar también que la herramienta computacional para los casos del SNI cuenta con archivos que permiten almacenar los datos iniciales del modelo, debido a que es un sistema complejo con varios parámetros, por lo cual ayudan a disminuir el tiempo que tarda el programa en encontrar la solución óptima.



## 3.2 CONCLUSIONES

- Se desarrolló una herramienta computacional que permite resolver el flujo óptimo de potencia de corriente alterna considerando diferentes formulaciones matemáticas, empleando lenguaje de programación de Python junto con la librería Pyomo. La herramienta fue aplicada a los sistemas IEEE de 14 barras y el SNI ecuatoriano.
- Se observó en los resultados que, al tratarse de un problema de optimización de costos que depende de la potencia de generación, el programa se enfoca en aprovechar la máxima potencia de los generadores de menor costos para lograr obtener una solución óptima que se encuentre dentro de cada uno de los límites técnicos.
- En el caso del sin, las centrales hidroeléctricas son las que mayor participación tienen para satisfacer la demanda ya que su costo de generación es menor en comparación a las centrales térmicas, y además, son las centrales con mayor capacidad instalada en nuestro país.
- En cuanto a los resultados obtenidos del OPF-AC con balance de potencia nodal se tiene valores iguales tanto en el costo total de generación como en el valor de potencia generada y su diferencia se ve reflejada en el tiempo que se demora el programa en encontrar la solución óptimo debido a la formulación matemática seleccionada.
- Para el caso del SNI se hace uso adicional de archivos que permiten almacenar los datos del modelo al ejecutar por primera vez el programa, esto con el objetivo de no saturar el programa ya que se trata de un sistema complejo con gran cantidad de elementos, y también ayudará a que el tiempo para encontrar la solución sea menor.
- De acuerdo con los resultados obtenidos en los diferentes casos de estudio se puede concluir que la herramienta computacional desarrollada funciona de manera correcta.

## 3.3 Recomendaciones

- Para tener una simulación correcta del programa se debe verificar tener instaladas todas las librerías necesarias, y además, contar con el solucionador (solver) adecuado para optimización no lineal, dentro del entorno Anaconda.
- En cuanto a las formulaciones de los diferentes casos del OPF-AC se recomienda revisar diferentes autores que permitan tener un mayor conocimiento del problema y encontrar las ecuaciones adecuadas para facilitar la programación.
- Utilizar SEP de prueba pequeños al inicio para poder comprobar de una manera más sencilla y rápida el correcto funcionamiento del programa, para posteriormente, emplearlo en sistemas más complejos.

## 4 REFERENCIAS BIBLIOGRÁFICAS

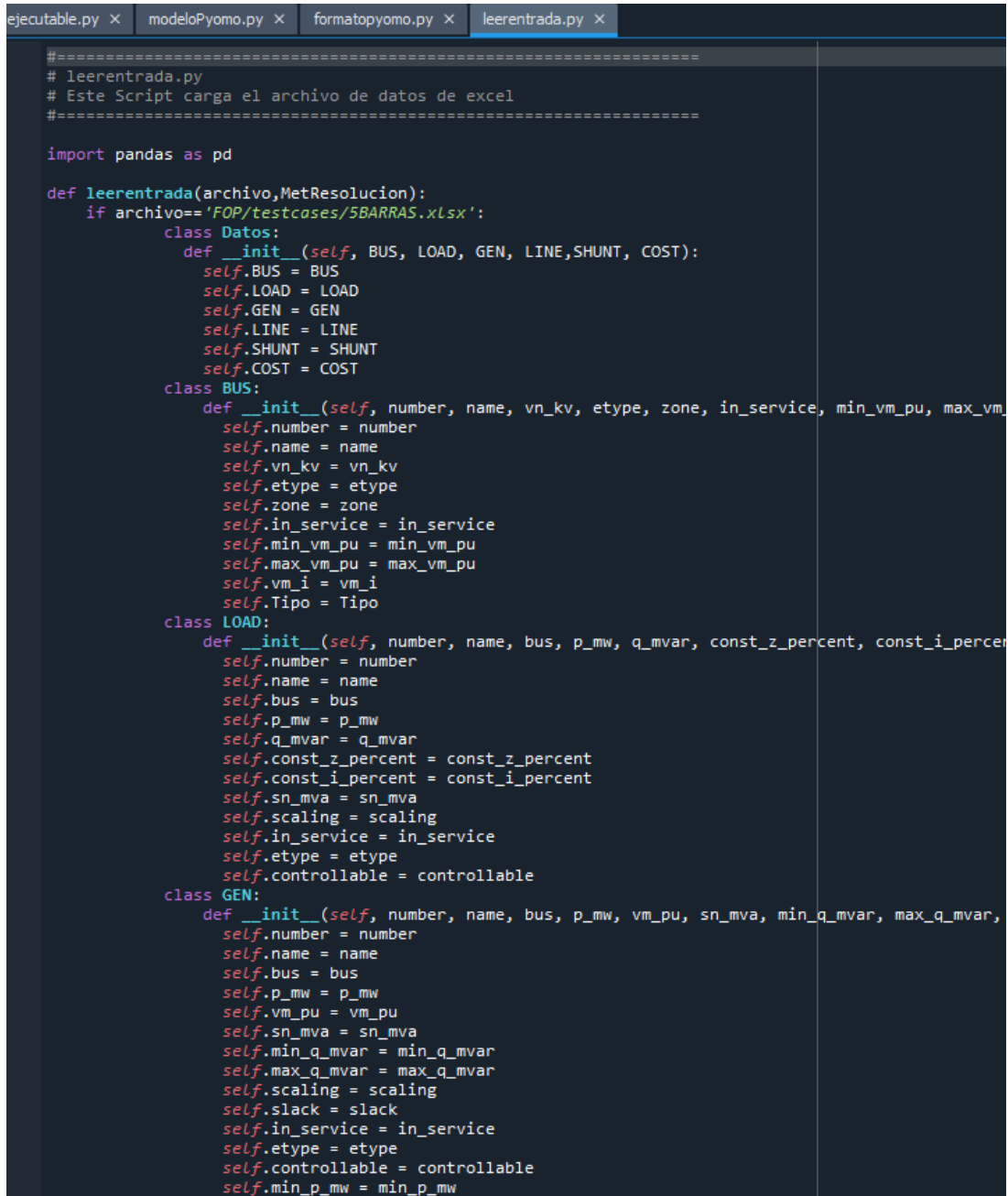
- [1] S. Chatzivasileiadis, Optimization in Modern Power Systems, First edition. Copenhagen, Denmark, 2018.
- [2] A. J. Wood, B. F. Wollenberg, y G. B. Sheblé, POWER GENERATION, OPERATION, AND CONTROL, THIRD EDITION. Hoboken, New Jersey, 2014.
- [3] J. F. Marley, D. K. Molzahn, y I. A. Hiskens, «Solving Multiperiod OPF Problems Using an AC-QP Algorithm Initialized With an SOCP Relaxation», IEEE Transactions on Power Systems, vol. 32, n.º 5, pp. 3538-3548, sep. 2017, doi: 10.1109/TPWRS.2016.2636132.
- [4] K. Frauendorfer, H. Glavitsch, y R. Bacher, Eds., Optimization in Planning and Operation of Electric Power Systems: Lecture Notes of the SVOR/ASRO Tutorial Thun, Switzerland, October 14–16, 1992. Heidelberg, Germany, 1993.
- [5] M. A. Ríos, C. J. Zapata, y O. Gómez, «Flujo de Potencia Óptimo con Restricciones de Estabilidad de Voltaje», p. 8, may 2014.
- [6] M. Jereminov, A. Pandey, y L. Pileggi, «Equivalent circuit formulation for solving AC optimal power flow», IEEE Transactions on Power Systems, vol. 34, n.º 3, pp. 2354-2365, 2018.
- [7] M. B. Cain, R. P. O'neill, y A. Castillo, «History of optimal power flow and formulations», Federal Energy Regulatory Commission, vol. 1, pp. 1-36, 2012.
- [8] B. Sereeter, C. Vuik, C. Witteveen, y P. Palensky, «Optimal power flow formulations and their impacts on the performance of solution methods», en 2019 IEEE Power & Energy Society General Meeting (PESGM), IEEE, 2019, pp. 1-5.
- [9] A. S. Nair, S. Abhyankar, S. Peles, y P. Ranganathan, «Computational and numerical analysis of ac optimal power flow formulations on large-scale power grids», Electric Power Systems Research, vol. 202, p. 107594, 2022.
- [10] C. López Servin, «Estudio de flujos de potencia óptimos de C.A. empleando el método de punto interior.», Instituto Politécnico Nacional, México, 2016.
- [11] H. Glavitsch y R. Bacher, «Optimal power flow algorithms», Analysis and control system techniques for electric power systems, vol. 41, pp. 135-206, 1991.

- [12] S. A. Sadat y K. Kim, Numerical Performance of Different Formulations for Alternating Current Optimal Power Flow. 2021. doi: 10.36227/techrxiv.16578962.
- [13] W. E. Hart et al., Pyomo-optimization modeling in python, vol. 67. Springer, 2017.
- [14] «Ipopt: Documentación». <https://coin-or.github.io/Ipopt/> (accedido 7 de julio de 2023).
- [15] S. Frank y S. Rebennack, «A Primer on Optimal Power Flow: Theory, Formulation, and Practical Examples». Colorado School of Mines, octubre de 2012.
- [16] «A Current-Voltage Successive Linear Programming Approach to Solving the ACOPF - Buscar con Google». [https://www.google.com/search?q=A+Current-Voltage+Successive+Linear+Programming+Approach+to+Solving+the+ACOPF&rlz=1C1CHBF\\_esEC843EC843&oq=A+Current-Voltage+Successive+Linear+Programming+Approach+to+Solving+the+ACOPF&aqs=chrome..69i57.650j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=A+Current-Voltage+Successive+Linear+Programming+Approach+to+Solving+the+ACOPF&rlz=1C1CHBF_esEC843EC843&oq=A+Current-Voltage+Successive+Linear+Programming+Approach+to+Solving+the+ACOPF&aqs=chrome..69i57.650j0j7&sourceid=chrome&ie=UTF-8) (accedido 7 de julio de 2023).
- [17] Z. Yang, H. Zhong, A. Bose, T. Zheng, Q. Xia, y C. Kang, «A Linearized OPF Model with Reactive Power and Voltage Magnitude: A Pathway to Improve the MW-Only DC OPF», IEEE Transactions on Power Systems, vol. PP, jun. 2017, doi: 10.1109/TPWRS.2017.2718551.
- [18] D. K. Molzahn y I. A. Hiskens, «A Survey of Relaxations and Approximations of the Power Flow Equations», EES, vol. 4, n.º 1-2, pp. 1-221, feb. 2019, doi: 10.1561/31000000012.
- [19] J. Valerino, R. Núñez, A. Díaz, y M. Banderas, «La estabilidad de la tensión de los sistemas eléctricos de potencia utilizando el programa “PSAF”»; Voltage stability of electric power system using the program “PSAF”», Ingeniería Energética, vol. 33, pp. 84-93, ago. 2012.

## 5 ANEXOS

### ANEXO I. MANUAL DE USUARIO

El programa desarrollado está formado por varios archivos o scripts que cumplen diferentes funciones para poder plantear y resolver el problema del flujo óptimo de potencia de corriente alterna en sus diferentes formulaciones matemáticas. A continuación, se presenta la descripción de cada script:



```
#####
# leerentrada.py
# Este Script carga el archivo de datos de excel
#####

import pandas as pd

def leerentrada(archivo, MetResolucion):
    if archivo=='FOP/testcases/SBARRAS.xlsx':
        class Datos:
            def __init__(self, BUS, LOAD, GEN, LINE, SHUNT, COST):
                self.BUS = BUS
                self.LOAD = LOAD
                self.GEN = GEN
                self.LINE = LINE
                self.SHUNT = SHUNT
                self.COST = COST
        class BUS:
            def __init__(self, number, name, vn_kv, etype, zone, in_service, min_vm_pu, max_vm_pu, vm_i, Tipo):
                self.number = number
                self.name = name
                self.vn_kv = vn_kv
                self.etype = etype
                self.zone = zone
                self.in_service = in_service
                self.min_vm_pu = min_vm_pu
                self.max_vm_pu = max_vm_pu
                self.vm_i = vm_i
                self.Tipo = Tipo
        class LOAD:
            def __init__(self, number, name, bus, p_mw, q_mvar, const_z_percent, const_i_percent, sn_mva, scaling, in_service, etype, controllable):
                self.number = number
                self.name = name
                self.bus = bus
                self.p_mw = p_mw
                self.q_mvar = q_mvar
                self.const_z_percent = const_z_percent
                self.const_i_percent = const_i_percent
                self.sn_mva = sn_mva
                self.scaling = scaling
                self.in_service = in_service
                self.etype = etype
                self.controllable = controllable
        class GEN:
            def __init__(self, number, name, bus, p_mw, vm_pu, sn_mva, min_q_mvar, max_q_mvar, scaling, slack, in_service, etype, controllable, min_p_mw):
                self.number = number
                self.name = name
                self.bus = bus
                self.p_mw = p_mw
                self.vm_pu = vm_pu
                self.sn_mva = sn_mva
                self.min_q_mvar = min_q_mvar
                self.max_q_mvar = max_q_mvar
                self.scaling = scaling
                self.slack = slack
                self.in_service = in_service
                self.etype = etype
                self.controllable = controllable
                self.min_p_mw = min_p_mw
```

Figura 23.- Ventana del script 'leerentrada.py'.

En la figura 23 se muestra el script que permite trabajar con los datos de los archivos de Excel que contiene todos los parámetros de los diferentes sistemas eléctricos. Lo primero que se realiza es la importación de la librería pandas, que es empleada para el manejo de datos, luego se crea la función *leerentrada* para agrupar los datos del sistema en diferentes clases dependiendo del tipo de elemento que, posteriormente, serán empleados para darles el formato necesario para poder trabajar con la librería Pyomo.

```

1 #=====
2 # formatopyomo.py
3 # Script de Python para escribir un archivo de datos para PYOMO
4 #=====
5 import os
6
7 import math as mt
8 import FOP.Nuevo.Ybarra as Yp
9 def formatopyomo(ptc,mod, MetResolucion,testcase):
10     import numpy as np
11     nGen=0
12     nDemanda=len(ptc.LOAD.name)
13     nBuses=len(ptc.BUS.name)
14     nLines=len(ptc.LINE.name)
15     if testcase!='FOP/testcases/5BARRAS.xlsx':
16         nTrafos=len(ptc.TRAFO.name)
17     C2=[]
18     C1=[]
19     C0=[]
20     Nseg=[]
21     demP=[]
22     demQ=[]
23
24     min_p=[]
25     max_p=[]
26
27     min_q=[]
28     max_q=[]
29
30     thetamax=[]
31     thetamin=[]
32
33     Voltmax=[]
34     Voltmin=[]
35
36     Pijmax=[]
37     PijTmax=[]
38
39     nG_Slack=[]
40
41     for i in range(len(ptc.GEN.name)):
42         C0.append(ptc.COST.cp0_eur[i])
43         C1.append(ptc.COST.cp1_eur_per_mw[i])
44         C2.append(ptc.COST.cp2_eur_per_mw2[i])
45         Nseg.append(ptc.COST.Nseg[i])
46         min_p.append(ptc.GEN.min_p_mw[i])
47         max_p.append(ptc.GEN.max_p_mw[i])
48         min_q.append(ptc.GEN.min_q_mvar[i])
49         max_q.append(ptc.GEN.max_q_mvar[i])
50         nG_Slack.append(ptc.GEN.slack[i])

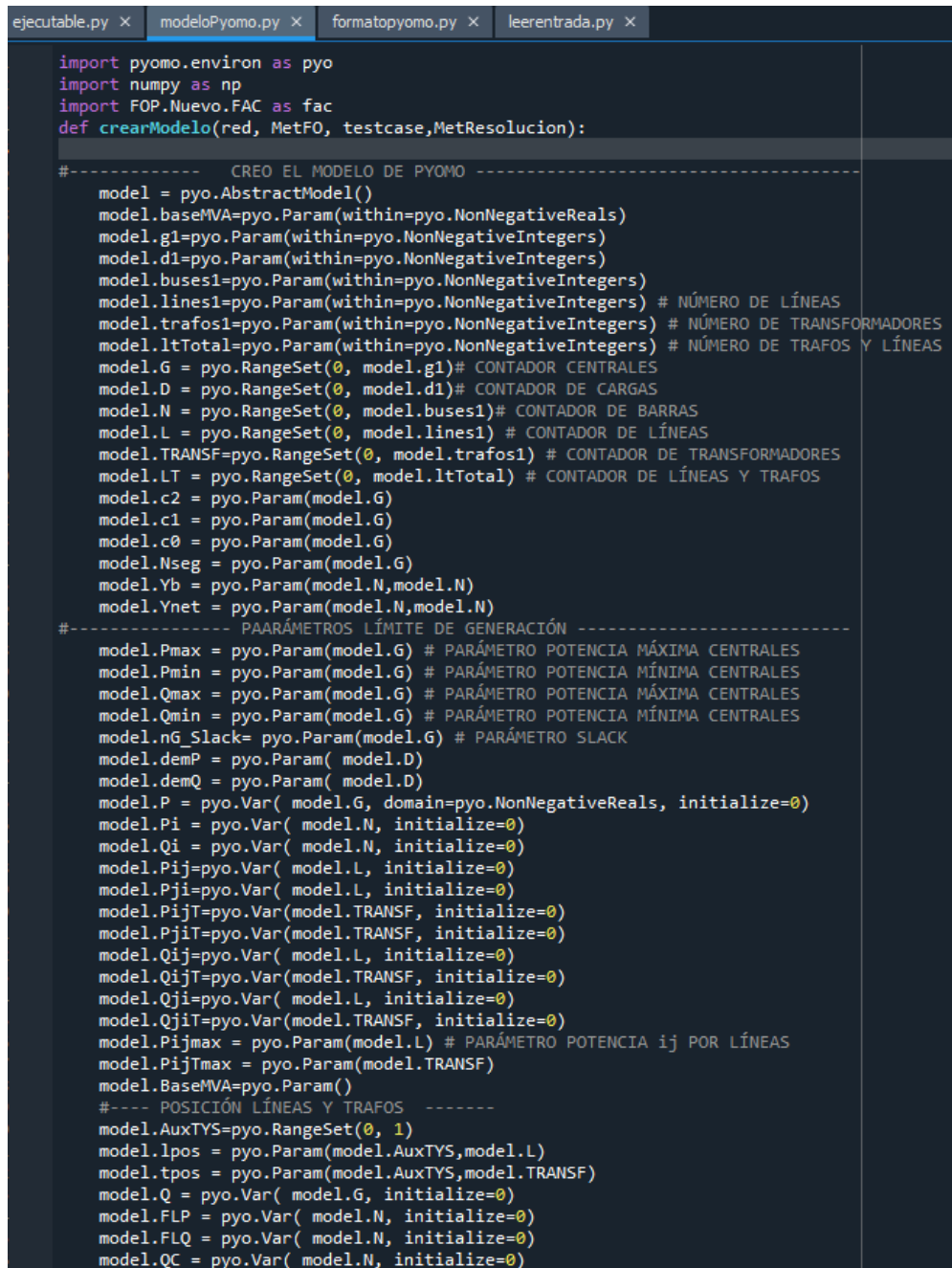
```

Figura 24.- Ventana del script 'formatopyomo.py'.

En la figura 24 se presenta el archivo donde se realiza la implementación de la función *formatopyomo* que se utiliza para poder trabajar con las diferentes variables en el formato de Pyomo, donde para cada una de las variables del sistema, por ejemplo, número barras, costos, potencias, se crea diferentes diccionarios con sus respectivos nombres que,

posteriormente, serán empleados para formar las diferentes ecuaciones de cada uno de los casos del problema.

Luego, una vez ya realizo la lectura y la modificación del formato de los datos, se desarrolla el script denominado modeloPyomo, que se muestra en la figura 25, donde se crea las diferentes funciones con las restricciones de igualdad y desigualdad para formar y resolver el OPF-AC dependiendo de la formulación matemática seleccionado.



```
ejecutable.py x modeloPyomo.py x formatopyomo.py x leerentrada.py x
import pyomo.environ as pyo
import numpy as np
import FOP.Nuevo.FAC as fac
def crearModelo(red, MetFO, testcase, MetResolucion):
#----- CREO EL MODELO DE PYOMO -----
model = pyo.AbstractModel()
model.baseMVA=pyo.Param(within=pyo.NonNegativeReals)
model.g1=pyo.Param(within=pyo.NonNegativeIntegers)
model.d1=pyo.Param(within=pyo.NonNegativeIntegers)
model.buses1=pyo.Param(within=pyo.NonNegativeIntegers)
model.lines1=pyo.Param(within=pyo.NonNegativeIntegers) # NÚMERO DE LÍNEAS
model.trafos1=pyo.Param(within=pyo.NonNegativeIntegers) # NÚMERO DE TRANSFORMADORES
model.ltTotal=pyo.Param(within=pyo.NonNegativeIntegers) # NÚMERO DE TRAFOS Y LÍNEAS
model.G = pyo.RangeSet(0, model.g1)# CONTADOR CENTRALES
model.D = pyo.RangeSet(0, model.d1)# CONTADOR DE CARGAS
model.N = pyo.RangeSet(0, model.buses1)# CONTADOR DE BARRAS
model.L = pyo.RangeSet(0, model.lines1) # CONTADOR DE LÍNEAS
model.TRANSF=pyo.RangeSet(0, model.trafos1) # CONTADOR DE TRANSFORMADORES
model.LT = pyo.RangeSet(0, model.ltTotal) # CONTADOR DE LÍNEAS Y TRAFOS
model.c2 = pyo.Param(model.G)
model.c1 = pyo.Param(model.G)
model.c0 = pyo.Param(model.G)
model.Nseg = pyo.Param(model.G)
model.Yb = pyo.Param(model.N,model.N)
model.Ynet = pyo.Param(model.N,model.N)
#----- PAARÁMETROS LÍMITE DE GENERACIÓN -----
model.Pmax = pyo.Param(model.G) # PARÁMETRO POTENCIA MÁXIMA CENTRALES
model.Pmin = pyo.Param(model.G) # PARÁMETRO POTENCIA MÍNIMA CENTRALES
model.Qmax = pyo.Param(model.G) # PARÁMETRO POTENCIA MÁXIMA CENTRALES
model.Qmin = pyo.Param(model.G) # PARÁMETRO POTENCIA MÍNIMA CENTRALES
model.nG_Slack= pyo.Param(model.G) # PARÁMETRO SLACK
model.demP = pyo.Param( model.D)
model.demQ = pyo.Param( model.D)
model.P = pyo.Var( model.G, domain=pyo.NonNegativeReals, initialize=0)
model.Pi = pyo.Var( model.N, initialize=0)
model.Qi = pyo.Var( model.N, initialize=0)
model.Pij=pyo.Var( model.L, initialize=0)
model.Pji=pyo.Var( model.L, initialize=0)
model.Pijt=pyo.Var(model.TRANSF, initialize=0)
model.Pjit=pyo.Var(model.TRANSF, initialize=0)
model.Qij=pyo.Var( model.L, initialize=0)
model.Qijt=pyo.Var(model.TRANSF, initialize=0)
model.Qji=pyo.Var( model.L, initialize=0)
model.Qjijt=pyo.Var(model.TRANSF, initialize=0)
model.Pijmax = pyo.Param(model.L) # PARÁMETRO POTENCIA ij POR LÍNEAS
model.Pijtmax = pyo.Param(model.TRANSF)
model.BaseMVA=pyo.Param()
#---- POSICIÓN LÍNEAS Y TRAFOS -----
model.AuxTYS=pyo.RangeSet(0, 1)
model.lpos = pyo.Param(model.AuxTYS,model.L)
model.tpos = pyo.Param(model.AuxTYS,model.TRANSF)
model.Q = pyo.Var( model.G, initialize=0)
model.FLP = pyo.Var( model.N, initialize=0)
model.FLQ = pyo.Var( model.N, initialize=0)
model.QC = pyo.Var( model.N, initialize=0)
```

Figura 25.- Ventana del script 'modeloPyomo.py'.

Como se observa en la figura 25 el programa parte de la importación de algunas librerías necesarias para poder desarrollar el código. Luego, se tiene el desarrollo de una función denominada *crearModelo* donde se crea el modelo abstracto de Pyomo, debido a que no se trabaja con valores fijos, con los diferentes parámetros, variables de estado y de control, restricciones para formular el problema del OPF.

Dentro de los parámetros se puede observar que se definen los datos de costos, potencia máximas y mínimas, demandas, posiciones de líneas y transformadores, y, entre otras variables más.

Se presenta también las variables de estado donde se tiene el voltaje de barras como su ángulo, mientras que en las variables de control se presentan la potencia activa y reactiva de generación.

Luego de la creación del modelo se desarrolla el código de la función objetivo y de las diferentes restricciones del problema, que se muestra en las figuras de la sección 2.1 con su respectiva descripción.

A continuación se presenta como se debe hacer uso de la herramienta computacional:

El archivo principal de la herramienta computacional se denomina como `ejecutable.py` que se muestra en la figura 26, desde este script se pone en marcha al programa para obtener la solución del OPF-AC

Como primer paso para poder ejecutar el programa se muestra en la línea 12 de la figura 26 que se debe seleccionar el caso de estudio donde se presenta los siguientes sistemas de potencia:

- i. La red IEEE de 14 barras.
- ii. La red IEEE de 5 barras.
- iii. El Sistema Nacional Interconectado (SNI) en temporada lluviosa.
- iv. El Sistema Nacional Interconectado (SNI) en temporada seca.

Los nombres de los casos de estudio se pueden observar en la figura 26 a partir de la línea 10 hasta la 15, presentando un rango de opciones desde la #0 hasta #4 dependiente del caso de estudio que se desee ejecutar, para lo cual en la línea 17 se presenta la variable *testcase* donde se debe modificar el número que se encuentra entre corchetes al lado de *archivo*.



El siguiente paso es escoger la función objetivo, ya sea esta lineal (opción #0) o cuadrática (opción #1), en la variable *MetFO* presente en la línea 22 se debe modificar el número para elegir con cual función de costos va a trabajar el programa.

Por último, se debe seleccionar la formulación matemática para la resolución del OPF-AC en la variable *MetResolucion* que se encuentra en la línea 27 donde se puede elegir desde la opción #0 hasta la #3 y, además, a partir de la línea 29 se muestra en comentarios la descripción para poder identificar con facilidad cada una de las opciones.

```

1  from pyomo.opt import SolverFactory
2  import time
3  from FOP.leerentrada import leerentrada
4  from FOP.formatopyomo import formatopyomo
5  import FOP.guardarres as res
6  import FOP.modeloPyomo as modeloC
7  tic = time.time()
8
9  #----- ESCOGER EL ARCHIVO DE PRUEBA (0-1-2-3) -----
10 archivo = [
11
12     'IEEE_14_bus_system_pwlV2.xlsx', # 0 Red IEEE 14 barras
13     '5BARRAS.xlsx',                # 1 5 barras
14     'SNI_Lluvioso.xlsx',          # 2 SNI Lluvioso
15     'SNI_Seco.xlsx',               # 3 SNI Seco
16 ]
17 testcase = 'FOP/testcases/' + archivo[1]
18
19 #----- FLUJO ÓPTIMO DE POTENCIA AC -----
20
21 # ESCOGER EL MÉTODO DE RESOLUCIÓN:
22 MetFO=1 # SELECCIONAR LA FUNCIÓN OBJETIVO
23 '''
24 #0:LÍNEAL
25 #1:CUADRÁTICA
26 '''
27 MetResolucion=2 # SELECCIONAR LA FORMULACIÓN DEL OPF-AC
28 '''
29 #0: OPF-AC con Relajación de Ecuaciones del Flujo de Potencia.
30 #1: OPF-AC con Restricción de flujos de potencia expresado en Coordenadas Polares.
31 #2: OPF-AC con Restricción de flujos de potencia expresado en Coordenadas Rectangulares.
32 #3: OPF-AC con Ecuación de Balance Nodal usando Corrientes.
33 '''
34 solver='ipopt'
35 out=0
36 opt=({'solver':solver,'out':out})
37 testcase = testcase
38 mod = 'FOPAC'
39 ptc = leerentrada(testcase, MetResolucion) # LEER EL CASO DE ESTUDIO
40 data = formatopyomo(ptc, mod, MetResolucion, testcase)
41 optimise = SolverFactory(opt['solver'])
42
43 print("MODELO")
44 model=modeloC.crearModelo(ptc, MetFO, testcase, MetResolucion) # CREA MODELO DE PYOMO
45 print("DATOS")
46 i=model.create_instance(data) # INIALIZACIÓN
47 solver=SolverFactory('ipopt') # SOLUCIONADOR DE MODELOS NO LINEALES
48 print("RESOLVER")
49 results = solver.solve(i) # RESOLVER MODELO
50 print(results)
51 res.guardarSalida(i, ptc, testcase, MetResolucion) # IMPRESIÓN RESULTADOS
52 toc = time.time() - tic
53 print('Tiempo:', toc) # IMPRESIÓN DEL TIEMPO TOTAL DE EJECUCIÓN.

```

Figura 26.- Ventana del script 'ejecutable.py'.

Una vez que se ingresa de manera correcta todos los datos anteriores descritos, lo primero que realiza el programa es la creación del modelo en la línea 44, donde se formulan de las diferentes ecuaciones, luego en la línea 46 se procede a inicializar, donde se vuelve a crear una nueva instancia del modelo ingresando los datos que se encuentran en el diccionario de PYOMO lo cual da forma a las ecuaciones, que posteriormente serán resueltas con el solver *ipopt* que se muestra en la línea 47. Finalmente, una vez que el programa resuelve el problema se imprimen los resultados y el tiempo que se demoró en encontrar una solución óptima lo cual se puede encontrar en la línea 51 y 52.

En la figura 26 se muestra un ejemplo del correcto ingreso de los datos anterior donde el programa se encuentra configurado para emplear la función de costos cuadrática en el caso del IEEE de 5 barras empleando la formulación del balance de potencia con las ecuaciones del flujo de potencia expresadas en forma polar.

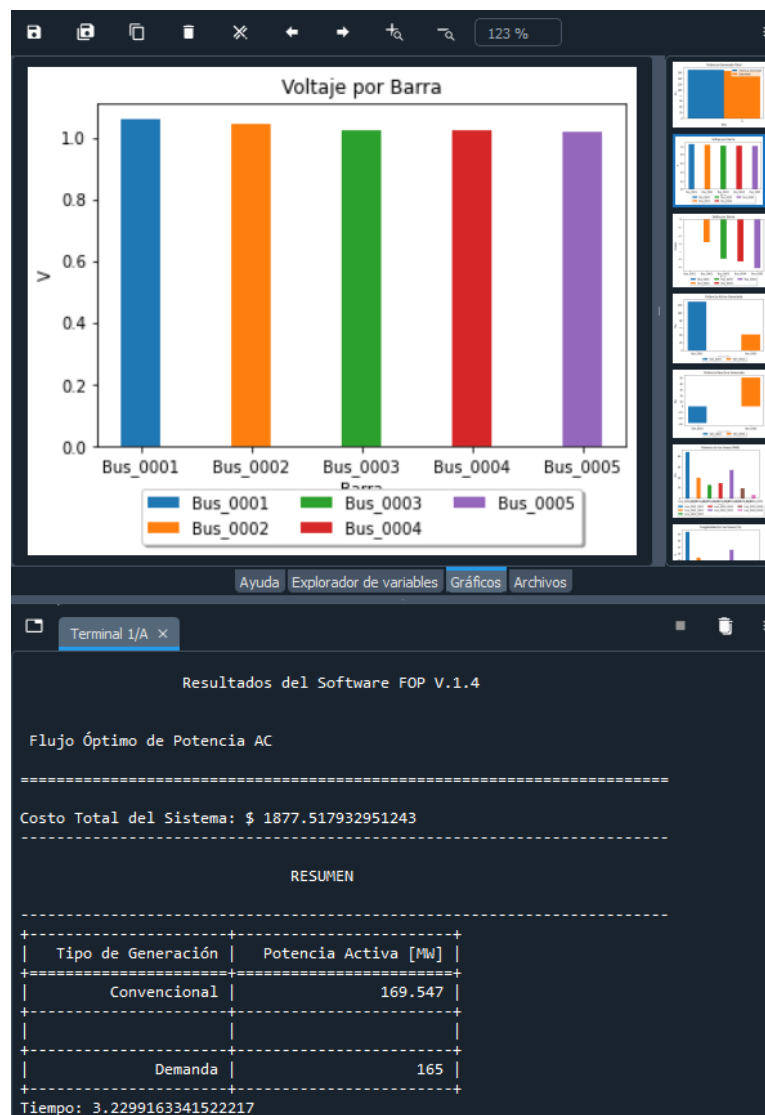


Figura 27.- Presentación e impresión de resultados.

En la figura 27 se puede observar la impresión de los resultados una vez que se manda a correr el programa, los cuales se presentan en la parte derecha de la ventana principal, donde en la parte superior se encuentran los diferentes resultados de manera gráfica y, en la parte inferior se imprimen los resultados del costo total del sistema y un cuadro resumen que presenta los valores de la potencia de generación, demanda con sus respectivas unidades y, al final se muestra el tiempo total de ejecución para encontrar la solución. Además, cabe destacar que los resultados también se los puede encontrar dentro de la carpeta que contiene el programa en un archivo de Excel denominado como *Resultados.xlsx*.

## ANEXO II. DATOS DEL SISTEMA IEEE 14 BARRAS

*Tabla 1. Datos del voltaje y tipo de barras.*

<b>Barra</b>	<b>Voltaje Nominal [kV]</b>	<b>Voltaje [p.u.]</b>	<b>Tipo</b>
<b>Bus_0001</b>	132	1.06	Slack
<b>Bus_0002</b>	132	1.045	PV
<b>Bus_0003</b>	132	1.01	PV
<b>Bus_0004</b>	132	-	PQ
<b>Bus_0005</b>	132	-	PQ
<b>Bus_0006</b>	33	1.07	PV
<b>Bus_0007</b>	1.0	-	PQ
<b>Bus_0008</b>	11	1.09	PV
<b>Bus_0009</b>	33	-	PQ
<b>Bus_0010</b>	33	-	PQ
<b>Bus_0011</b>	33	-	PQ
<b>Bus_0012</b>	33	-	PQ
<b>Bus_0013</b>	33	-	PQ
<b>Bus_0014</b>	33	-	PQ

*Tabla 2. Datos de cargas.*

<b>Carga</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>Load_0002</b>	Bus_0002	21.7	12.7
<b>Load_0003</b>	Bus_0003	94.2	19
<b>Load_0004</b>	Bus_0004	47.8	-3.9
<b>Load_0005</b>	Bus_0005	7.6	1.6
<b>Load_0006</b>	Bus_0006	11.2	7.5
<b>Load_0009</b>	Bus_0009	29.5	16.6
<b>Load_0010</b>	Bus_0010	9	5.8
<b>Load_0011</b>	Bus_0011	3.5	1.8
<b>Load_0012</b>	Bus_0012	6.1	1.6
<b>Load_0013</b>	Bus_0013	13.5	5.8
<b>Load_0014</b>	Bus_0014	14.9	5

**Tabla 3.** Datos de capacidad de los generadores.

<b>Generador</b>	<b>Voltaje [p.u.]</b>	<b>Qmin [MVar]</b>	<b>Qmax [MVar]</b>	<b>Pmin [MW]</b>	<b>Pmax [MW]</b>
<b>Gen_0001</b>	1.06	-160	300	30	320
<b>Gen_0002</b>	1.045	-40	50	40	180
<b>Gen_0003</b>	1.01	-22	40	25	195
<b>Gen_0006</b>	1.07	-15	24	20	245
<b>Gen_0008</b>	1.09	-15	24	10	150

**Tabla 4.** Datos de líneas de transmisión.

<b>Línea</b>	<b>from_bus</b>	<b>to_bus</b>	<b>r [<math>\Omega</math>/km]</b>	<b>x [<math>\Omega</math>/km]</b>	<b>b [nf/km]</b>
<b>Line_0001_0002/1</b>	Bus_0001	Bus_0002	6.753542	20.61956	401.9064167
<b>Line_0001_0002/2</b>	Bus_0001	Bus_0002	6.753542	20.61956	401.9064167
<b>Line_0001_0005</b>	Bus_0001	Bus_0005	9.414187	38.8625	749.0074167
<b>Line_0002_0003</b>	Bus_0002	Bus_0003	8.187537	34.49428	666.7993333
<b>Line_0002_0004</b>	Bus_0002	Bus_0004	10.12509	30.722	569.3675
<b>Line_0002_0005</b>	Bus_0002	Bus_0005	9.922968	30.29685	517.60675
<b>Line_0003_0004</b>	Bus_0003	Bus_0004	11.67582	29.80027	526.741
<b>Line_0004_0005</b>	Bus_0005	Bus_0004	2.326104	7.337246	194.86375
<b>Line_0006_0011</b>	Bus_0011	Bus_0006	1.034332	2.166021	0
<b>Line_0006_0012</b>	Bus_0006	Bus_0012	1.33849	2.785771	0
<b>Line_0006_0013</b>	Bus_0013	Bus_0006	0.7203735	1.41864	0
<b>Line_0009_0010</b>	Bus_0009	Bus_0010	0.3464109	0.9202054	0
<b>Line_0009_0014</b>	Bus_0014	Bus_0009	1.384228	2.944439	0
<b>Line_0010_0011</b>	Bus_0010	Bus_0011	0.8935243	2.091643	0
<b>Line_0012_0013</b>	Bus_0012	Bus_0013	2.405819	2.176693	0
<b>Line_0013_0014</b>	Bus_0013	Bus_0014	1.861428	3.789938	0

**Tabla 5.** Datos de transformadores.

<b>Transformador</b>	<b>Barra HV</b>	<b>Barra LV</b>	<b>HV [kV]</b>	<b>LV [kV]</b>	<b>r [p.u]</b>	<b>x [p.u]</b>
<b>Trf_0004_0007</b>	Bus_0004	Bus_0007	132	1	0	0.20912
<b>Trf_0004_0009</b>	Bus_0004	Bus_0009	132	33	0	0.55618
<b>Trf_0005_0006</b>	Bus_0005	Bus_0006	132	33	0	0.25202
<b>Trf_0007_0008</b>	Bus_0008	Bus_0007	11	1	0	0.17615
<b>Trf_0007_0009</b>	Bus_0009	Bus_0007	33	1	0	0.11001

## ANEXO III. TABLAS DE RESULTADOS DEL SISTEMA IEEE 14 BARRAS

**Método de resolución: OPF-AC expresado en coordenadas polares:**

- Función objetivo: Cuadrática.

*Tabla 6. Resultados del voltaje y ángulo de barras.*

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	5.76E-29	1.06
<b>Bus_0002</b>	-3.09368	1.045
<b>Bus_0003</b>	-7.67141	1.01
<b>Bus_0004</b>	-6.97413	1.020304
<b>Bus_0005</b>	-5.89708	1.023783
<b>Bus_0006</b>	-9.42924	1.07
<b>Bus_0007</b>	-8.87935	1.052727
<b>Bus_0008</b>	-7.99817	1.09
<b>Bus_0009</b>	-10.472	1.035305
<b>Bus_0010</b>	-10.5693	1.033777
<b>Bus_0011</b>	-10.1208	1.047983
<b>Bus_0012</b>	-10.3212	1.053707
<b>Bus_0013</b>	-10.4071	1.047239
<b>Bus_0014</b>	-11.4656	1.022284

*Tabla 7. Resultados de la potencia de generación.*

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>Gen_0001</b>	Bus_0001	154.0219	50.33281
<b>Gen_0002</b>	Bus_0002	40	4.241804
<b>Gen_0003</b>	Bus_0003	41.38838	8.640134
<b>Gen_0006</b>	Bus_0006	20	4.357596
<b>Gen_0008</b>	Bus_0008	10	4.357596

- Función objetivo: Lineal.

**Tabla 8.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	2.59E-33	1.06
<b>Bus_0002</b>	-0.08998	1.045
<b>Bus_0003</b>	-6.10497	1.01
<b>Bus_0004</b>	-4.92992	1.020243
<b>Bus_0005</b>	-3.96575	1.024014
<b>Bus_0006</b>	-7.46225	1.07
<b>Bus_0007</b>	-6.85441	1.052739
<b>Bus_0008</b>	-5.97324	1.09
<b>Bus_0009</b>	-8.457	1.035376
<b>Bus_0010</b>	-8.5629	1.033845
<b>Bus_0011</b>	-8.13394	1.048025
<b>Bus_0012</b>	-8.35094	1.053705
<b>Bus_0013</b>	-8.43341	1.047257
<b>Bus_0014</b>	-9.46859	1.022336

**Tabla 9.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAR]</b>
<b>Gen_0001</b>	Bus_0001	51.7912	47.19699
<b>Gen_0002</b>	Bus_0002	157.7693	5.109548
<b>Gen_0003</b>	Bus_0003	25	8.305978
<b>Gen_0006</b>	Bus_0006	20	4.317652
<b>Gen_0008</b>	Bus_0008	10	4.317652



**Método de resolución: OPF-AC expresado en coordenadas rectangulares:**

- Función objetivo: Cuadrática.

**Tabla 10.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	-3.1E-30	1.06
<b>Bus_0002</b>	-3.09368	1.045
<b>Bus_0003</b>	-7.67141	1.01
<b>Bus_0004</b>	-6.97413	1.020304
<b>Bus_0005</b>	-5.89708	1.023783
<b>Bus_0006</b>	-9.42924	1.07
<b>Bus_0007</b>	-8.87935	1.052727
<b>Bus_0008</b>	-7.99817	1.09
<b>Bus_0009</b>	-10.472	1.035305
<b>Bus_0010</b>	-10.5693	1.033777
<b>Bus_0011</b>	-10.1208	1.047983
<b>Bus_0012</b>	-10.3212	1.053707
<b>Bus_0013</b>	-10.4071	1.047239
<b>Bus_0014</b>	-11.4656	1.022284

**Tabla 11.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>Gen_0001</b>	Bus_0001	154.0219	50.33281
<b>Gen_0002</b>	Bus_0002	40	4.241804
<b>Gen_0003</b>	Bus_0003	41.38838	8.640134
<b>Gen_0006</b>	Bus_0006	20	4.357596
<b>Gen_0008</b>	Bus_0008	10	4.357596

- Función objetivo: Lineal.

**Tabla 12.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	7.14E-33	1.06
<b>Bus_0002</b>	-0.08998	1.045
<b>Bus_0003</b>	-6.10497	1.01
<b>Bus_0004</b>	-4.92992	1.020243
<b>Bus_0005</b>	-3.96575	1.024014
<b>Bus_0006</b>	-7.46225	1.07
<b>Bus_0007</b>	-6.85441	1.052739
<b>Bus_0008</b>	-5.97324	1.09
<b>Bus_0009</b>	-8.457	1.035376
<b>Bus_0010</b>	-8.5629	1.033845
<b>Bus_0011</b>	-8.13394	1.048025
<b>Bus_0012</b>	-8.35094	1.053705
<b>Bus_0013</b>	-8.43341	1.047257
<b>Bus_0014</b>	-9.46859	1.022336

**Tabla 13.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAR]</b>
<b>Gen_0001</b>	Bus_0001	51.7912	47.19699
<b>Gen_0002</b>	Bus_0002	157.7693	5.109548
<b>Gen_0003</b>	Bus_0003	25	8.305978
<b>Gen_0006</b>	Bus_0006	20	4.317652
<b>Gen_0008</b>	Bus_0008	10	4.317652

**Método de resolución: OPF-AC con relajación de las ecuaciones del flujo de potencia:**

- Función objetivo: Cuadrática.

**Tabla 14.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	-1.543E-31	1.06
<b>Bus_0002</b>	-2.8889222	1.045
<b>Bus_0003</b>	-7.0516475	1.01
<b>Bus_0004</b>	-6.6386319	1.02068087
<b>Bus_0005</b>	-5.6228026	1.0241466
<b>Bus_0006</b>	-9.100472	1.07
<b>Bus_0007</b>	-8.4494697	1.05297767
<b>Bus_0008</b>	-7.4407226	1.09
<b>Bus_0009</b>	-10.076377	1.03563831
<b>Bus_0010</b>	-10.185671	1.03406683
<b>Bus_0011</b>	-9.764506	1.04814004
<b>Bus_0012</b>	-9.9876739	1.05372113
<b>Bus_0013</b>	-10.069041	1.04729838
<b>Bus_0014</b>	-11.094768	1.02251026

**Tabla 15.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAR]</b>
<b>Gen_0001</b>	Bus_0001	151.6298	65.94547
<b>Gen_0002</b>	Bus_0002	40	4.994232
<b>Gen_0003</b>	Bus_0003	47.33988	8.997325
<b>Gen_0006</b>	Bus_0006	20	4.499087
<b>Gen_0008</b>	Bus_0008	11.44771	4.499087

- Función objetivo: Lineal.

**Tabla 16.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	-1.1E-38	1.06
<b>Bus_0002</b>	-0.05778	1.045
<b>Bus_0003</b>	-6.07616	1.01
<b>Bus_0004</b>	-4.90437	1.020245
<b>Bus_0005</b>	-3.94249	1.02402
<b>Bus_0006</b>	-7.43824	1.07
<b>Bus_0007</b>	-6.82925	1.052741
<b>Bus_0008</b>	-5.94808	1.09
<b>Bus_0009</b>	-8.43204	1.035379
<b>Bus_0010</b>	-8.53811	1.033847
<b>Bus_0011</b>	-8.10954	1.048027
<b>Bus_0012</b>	-8.32687	1.053705
<b>Bus_0013</b>	-8.40927	1.047257
<b>Bus_0014</b>	-9.44399	1.022338

**Tabla 17.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>Gen_0001</b>	Bus_0001	53.63724	36.47483
<b>Gen_0002</b>	Bus_0002	158.8623	3.992375
<b>Gen_0003</b>	Bus_0003	25	7.665145
<b>Gen_0006</b>	Bus_0006	20	4.071304
<b>Gen_0008</b>	Bus_0008	10	4.071304

**Método de resolución: OPF-AC con balance de corriente nodal.**

- Función objetivo: Cuadrática.

**Tabla 18.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	-3.3E-28	1.06
<b>Bus_0002</b>	-3.09368	1.045
<b>Bus_0003</b>	-7.67141	1.01
<b>Bus_0004</b>	-6.97413	1.020304
<b>Bus_0005</b>	-5.89708	1.023783
<b>Bus_0006</b>	-9.42924	1.07
<b>Bus_0007</b>	-8.87935	1.052727
<b>Bus_0008</b>	-7.99817	1.09
<b>Bus_0009</b>	-10.472	1.035305
<b>Bus_0010</b>	-10.5693	1.033777
<b>Bus_0011</b>	-10.1208	1.047983
<b>Bus_0012</b>	-10.3212	1.053707
<b>Bus_0013</b>	-10.4071	1.047239
<b>Bus_0014</b>	-11.4656	1.022284

**Tabla 19.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>Gen_0001</b>	Bus_0001	154.8532	51.00893
<b>Gen_0002</b>	Bus_0002	40	4.268229
<b>Gen_0003</b>	Bus_0003	41.38838	8.652677
<b>Gen_0006</b>	Bus_0006	20	4.36256
<b>Gen_0008</b>	Bus_0008	10	4.36256

**Tabla 20.** Resultados del flujo de corriente por líneas y transformadores.

<b>L/T o Transformador</b>	<b>from_bus</b>	<b>to_bus</b>	<b>I_FROM</b>	<b>I_TO</b>
<b>Line_0001_0002/1</b>	Bus_0001	Bus_0002	0.945161	0.94191
<b>Line_0001_0002/2</b>	Bus_0001	Bus_0002	0.945161	0.94191
<b>Line_0001_0005</b>	Bus_0001	Bus_0005	0.490004	0.494497
<b>Line_0002_0003</b>	Bus_0002	Bus_0003	0.434266	0.441862
<b>Line_0002_0004</b>	Bus_0002	Bus_0004	0.398833	0.399636
<b>Line_0002_0005</b>	Bus_0002	Bus_0005	0.29856	0.301371
<b>Line_0003_0004</b>	Bus_0003	Bus_0004	0.094472	0.083581
<b>Line_0004_0005</b>	Bus_0005	Bus_0004	0.442801	0.441132
<b>Line_0006_0011</b>	Bus_0011	Bus_0006	0.115501	0.115501
<b>Line_0006_0012</b>	Bus_0006	Bus_0012	0.081784	0.081784
<b>Line_0006_0013</b>	Bus_0013	Bus_0006	0.198896	0.198896
<b>Line_0009_0010</b>	Bus_0009	Bus_0010	0.025796	0.025796
<b>Line_0009_0014</b>	Bus_0014	Bus_0009	0.073926	0.073926
<b>Line_0010_0011</b>	Bus_0010	Bus_0011	0.078409	0.078409
<b>Line_0012_0013</b>	Bus_0012	Bus_0013	0.022344	0.022344
<b>Line_0013_0014</b>	Bus_0013	Bus_0014	0.081073	0.081073
<b>Trf_0004_0007</b>	Bus_0004	Bus_0007	0.231354	0.231354
<b>Trf_0004_0009</b>	Bus_0004	Bus_0009	0.119686	0.119686
<b>Trf_0005_0006</b>	Bus_0005	Bus_0006	0.337868	0.337868
<b>Trf_0007_0008</b>	Bus_0008	Bus_0007	0.231344	0.231344
<b>Trf_0007_0009</b>	Bus_0009	Bus_0007	0.307667	0.307667

- Función objetivo: Lineal.

**Tabla 21.** Resultados del voltaje y ángulo de barras.

<b>Barra</b>	<b>Ángulo [grados]</b>	<b>Voltaje [p.u.]</b>
<b>Bus_0001</b>	1.1E-31	1.06
<b>Bus_0002</b>	0.069042	1.045
<b>Bus_0003</b>	-5.96268	1.01
<b>Bus_0004</b>	-4.80377	1.020253
<b>Bus_0005</b>	-3.85087	1.024043
<b>Bus_0006</b>	-7.3437	1.07
<b>Bus_0007</b>	-6.73016	1.052747
<b>Bus_0008</b>	-5.84898	1.09
<b>Bus_0009</b>	-8.33373	1.03539
<b>Bus_0010</b>	-8.44048	1.033857
<b>Bus_0011</b>	-8.01344	1.048032
<b>Bus_0012</b>	-8.23206	1.053705
<b>Bus_0013</b>	-8.3142	1.04726
<b>Bus_0014</b>	-9.34709	1.022345

**Tabla 22.** Resultados de la potencia de generación.

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAR]</b>
<b>Gen_0001</b>	Bus_0001	47.08552	48.22614
<b>Gen_0002</b>	Bus_0002	163.1674	4.463015
<b>Gen_0003</b>	Bus_0003	25	8.565693
<b>Gen_0006</b>	Bus_0006	20	4.349296
<b>Gen_0008</b>	Bus_0008	10	4.349296

**Tabla 23.** Resultados del flujo de corriente por líneas y transformadores.

<b>L/T o Transformador</b>	<b>from_bus</b>	<b>to_bus</b>	<b>I_FROM</b>	<b>I_TO</b>
<b>Line_0001_0002/1</b>	Bus_0001	Bus_0002	0.214641	0.2687
<b>Line_0001_0002/2</b>	Bus_0001	Bus_0002	0.214641	0.2687
<b>Line_0001_0005</b>	Bus_0001	Bus_0005	0.336935	0.348905
<b>Line_0002_0003</b>	Bus_0002	Bus_0003	0.555893	0.559482
<b>Line_0002_0004</b>	Bus_0002	Bus_0004	0.491718	0.490026
<b>Line_0002_0005</b>	Bus_0002	Bus_0005	0.403631	0.402617
<b>Line_0003_0004</b>	Bus_0003	Bus_0004	0.127523	0.124426
<b>Line_0004_0005</b>	Bus_0005	Bus_0004	0.394834	0.393688
<b>Line_0006_0011</b>	Bus_0011	Bus_0006	0.114398	0.114398
<b>Line_0006_0012</b>	Bus_0006	Bus_0012	0.081618	0.081618
<b>Line_0006_0013</b>	Bus_0013	Bus_0006	0.198209	0.198209
<b>Line_0009_0010</b>	Bus_0009	Bus_0010	0.027275	0.027275
<b>Line_0009_0014</b>	Bus_0014	Bus_0009	0.074938	0.074938
<b>Line_0010_0011</b>	Bus_0010	Bus_0011	0.07737	0.07737
<b>Line_0012_0013</b>	Bus_0012	Bus_0013	0.022218	0.022218
<b>Line_0013_0014</b>	Bus_0013	Bus_0014	0.080271	0.080271
<b>Trf_0004_0007</b>	Bus_0004	Bus_0007	0.232952	0.232952
<b>Trf_0004_0009</b>	Bus_0004	Bus_0009	0.120786	0.120786
<b>Trf_0005_0006</b>	Bus_0005	Bus_0006	0.334768	0.334768
<b>Trf_0007_0008</b>	Bus_0008	Bus_0007	0.231243	0.231243
<b>Trf_0007_0009</b>	Bus_0009	Bus_0007	0.308933	0.308933



## ANEXO IV. TABLAS DE RESULTADOS DEL SISTEMA NACIONAL INTERCONETADO (SNI)

*Método de resolución: OPF-AC expresado en coordenadas polares:*

- Función objetivo: Cuadrática.
- Temporada Lluviosa

*Tabla 24. Resultados de la potencia de generación.*

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVar]</b>
<b>G_HEMB_BABA_U1</b>	Bus 11	212.2169487	44.32817834
<b>G_HEMB_DPER_U1</b>	Bus 03	197.0779452	0.677167162
<b>G_HEMB_MAZA_U1</b>	Bus 24	205.2407144	44.32817834
<b>G_HEMB_PAUT_U1</b>	Bus 32	506.5492562	44.89053839
<b>G_HEMB_PUCA_U1</b>	Bus 71	3.576536362	9.522961062
<b>G_HEMB_SOPL_U1</b>	Bus 22	487.3500048	-6.033836664
<b>G_HPAS_ABAN_U1</b>	Bus 25	7.730537321	44.32817834
<b>G_HPAS_ABAN_U3</b>	Bus 26	9.818898513	44.32817834
<b>G_HPAS_AGOY_U1</b>	Bus 53	81.58155285	7.254299614
<b>G_HPAS_ALAO_U1_U4</b>	Bus 73	137.2299716	44.32817834
<b>G_HPAS_AMBI_U1</b>	Bus 76	130.1777763	44.32817834
<b>G_HPAS_CALO_U1_U2</b>	Bus 07	30.65170577	44.32817834
<b>G_HPAS_CCS_U1</b>	Bus 78	674.9562183	-1.672380658
<b>G_HPAS_CHIL_U1_U2</b>	Bus 65	1.2422E-06	44.32817834
<b>G_HPAS_CHTB_U1_U2</b>	Bus 21	14.68052386	44.32817834
<b>G_HPAS_CMOR_U1_U2</b>	Bus 37	93.73908023	44.32817834
<b>G_HPAS_CMOR_U3</b>	Bus 80	0.002452298	44.32817834
<b>G_HPAS_CUMB_U1_U4</b>	Bus 67	4.016425942	44.32817834
<b>G_HPAS_ECAR_U1</b>	Bus 60	13.12853081	44.32817834
<b>G_HPAS_GUAN_U1_U5</b>	Bus 66	1.14699E-06	44.32817834
<b>G_HPAS_ILL1_U1_U2</b>	Bus 85	0.798283063	44.32817834
<b>G_HPAS_ILL1_U3_U4</b>	Bus 86	1.105035052	44.32817834
<b>G_HPAS_ILL2_U1_U2</b>	Bus 70	72.89044092	44.32817834
<b>G_HPAS_LPLA_U1_U3</b>	Bus 77	1.725875112	44.32817834
<b>G_HPAS_MAND_U1</b>	Bus 12	41.28716341	44.32817834
<b>G_HPAS_NAYO_U1_U2</b>	Bus 68	2.00241E-07	44.32817834

<b>G_HPAS_OCAÑ_U1</b>	Bus 43	9.546628455	44.32817834
<b>G_HPAS_PAPA_U1</b>	Bus 69	2.159999008	44.32817834
<b>G_HPAS_PAPA_U2</b>	Bus 81	4.079999027	44.32817834
<b>G_HPAS_PASO_U1_U2</b>	Bus 64	4.65297E-07	44.32817834
<b>G_HPAS_PENI_U1_U3</b>	Bus 74	30.8469164	44.32817834
<b>G_HPAS_PENI_U4</b>	Bus 83	1.238679247	44.32817834
<b>G_HPAS_RBLA_U1</b>	Bus 84	2.070904359	44.32817834
<b>G_HPAS_RECU_U1</b>	Bus 61	1.85852E-07	44.32817834
<b>G_HPAS_SAUC_U1</b>	Bus 38	9.524090352	44.32817834
<b>G_HPAS_SAUC_U3</b>	Bus 39	9.524090466	44.32817834
<b>G_HPAS_SAUC_U4</b>	Bus 40	9.524090466	44.32817834
<b>G_HPAS_SAYM_U5</b>	Bus 41	17.22640639	44.32817834
<b>G_HPAS_SAYM_U7</b>	Bus 42	17.21963317	44.32817834
<b>G_HPAS_SFRA_U1</b>	Bus 55	46.96394608	2.790708858
<b>G_HPAS_SIBI_U1_U2</b>	Bus 20	13.88482504	44.32817834
<b>G_HPAS_SMCA_U1</b>	Bus 82	1.154763881	44.32817834
<b>G_TMCI_CCAS_U1_U4</b>	Bus 59	0	44.32817834
<b>G_TMCI_CES2_U1_U6</b>	Bus 08	6.99999993	44.32817834
<b>G_TMCI_CES2_U7_U12</b>	Bus 09	6.99999993	44.32817834
<b>G_TMCI_DAYU_U1</b>	Bus 56	0	44.32817834
<b>G_TMCI_GHER_U1_U6</b>	Bus 63	4.99999995	44.32817834
<b>G_TMCI_GROC_U1_U8</b>	Bus 50	3.99999996	44.32817834
<b>G_TMCI_GUA2_U6</b>	Bus 62	19.9999998	44.32817834
<b>G_TMCI_JARA_U1_U5</b>	Bus 04	0	44.32817834
<b>G_TMCI_JARA_U6_U9</b>	Bus 05	0	44.32817834
<b>G_TMCI_JIV3_U1</b>	Bus 58	0	44.32817834
<b>G_TMCI_LAFA_U1_U4</b>	Bus 75	0	44.32817834
<b>G_TMCI_LORE_U1</b>	Bus 57	0	44.32817834
<b>G_TMCI_MAN2_U1_U12</b>	Bus 02	0	44.32817834
<b>G_TMCI_MIRA_U11_U12</b>	Bus 06	4.99999995	44.32817834
<b>G_TMCI_SEL2_U1_U53</b>	Bus 47	0	44.32817834
<b>G_TMCI_SEL3_U1</b>	Bus 48	0	44.32817834
<b>G_TMCI_TGUA_U1_U22</b>	Bus 45	17.99999982	44.32817834
<b>G_TTGA_EQUI_U2</b>	Bus 51	39.9999996	44.32817834
<b>G_TTGA_EQUI_U4</b>	Bus 49	39.9999996	44.32817834

<b>G_TTGA_MAC1_U1</b>	Bus 19	0	8.538122687
<b>G_TTGA_MAC2_TM1</b>	Bus 17	0	44.32817834
<b>G_TTGA_MAC2_TM5</b>	Bus 18	0	44.32817834
<b>G_TTVA_COAZ_U1</b>	Bus 16	15.04000013	44.32817834
<b>G_TTVA_ECOE_U1</b>	Bus 14	12.99999987	44.32817834
<b>G_TTVA_GZEV_TV3</b>	Bus 52	0	16.28515404
<b>G_TTVA_ISCA_U1</b>	Bus 15	84.80031689	44.32817834
<b>G_TTVA_TRIN_U1</b>	Bus 46	0	23.69456893

**Método de resolución: OPF-AC con balance nodal de corriente:**

- Función objetivo: Cuadrática.
- Temporada Seca

**Tabla 25. Resultados de la potencia de generación.**

<b>Generador</b>	<b>Barra</b>	<b>P[MW]</b>	<b>Q[MVAr]</b>
<b>G_HEMB_BABA_U1</b>	Bus 11	212.6092	36.47952
<b>G_HEMB_DPER_U1</b>	Bus 03	213.3	0.301634
<b>G_HEMB_MAZA_U1</b>	Bus 24	150	36.47952
<b>G_HEMB_PAUT_U1</b>	Bus 32	435.038	41.25244
<b>G_HEMB_PUCA_U1</b>	Bus 71	3.471053	9.471599
<b>G_HEMB_SOPL_U1</b>	Bus 22	324	-14.3823
<b>G_HPAS_ABAN_U1</b>	Bus 25	9.25414	36.47952
<b>G_HPAS_ABAN_U3</b>	Bus 26	13.062	36.47952
<b>G_HPAS_AGOY_U1</b>	Bus 53	78.53519	7.17706
<b>G_HPAS_ALAO_U1_U4</b>	Bus 73	149.1581	36.47952
<b>G_HPAS_AMBI_U1</b>	Bus 76	140.0074	36.47952
<b>G_HPAS_CALO_U1_U2</b>	Bus 07	44.4343	36.47952
<b>G_HPAS_CCS_U1</b>	Bus 78	567.5421	-3.99007
<b>G_HPAS_CHIL_U1_U2</b>	Bus 65	2.160819	36.47952
<b>G_HPAS_CHTB_U1_U2</b>	Bus 21	22.99326	36.47952
<b>G_HPAS_CMOR_U1_U2</b>	Bus 37	125.2304	36.47952
<b>G_HPAS_CMOR_U3</b>	Bus 80	1.2	36.47952
<b>G_HPAS_CUMB_U1_U4</b>	Bus 67	5.288103	36.47952
<b>G_HPAS_ECAR_U1</b>	Bus 60	59.79311	36.47952

<b>G_HPAS_GUAN_U1_U5</b>	Bus 66	2.22572	36.47952
<b>G_HPAS_ILL1_U1_U2</b>	Bus 85	0.798917	36.47952
<b>G_HPAS_ILL1_U3_U4</b>	Bus 86	1.106253	36.47952
<b>G_HPAS_ILL2_U1_U2</b>	Bus 70	85.34088	36.47952
<b>G_HPAS_LPLA_U1_U3</b>	Bus 77	4.793685	36.47952
<b>G_HPAS_MAND_U1</b>	Bus 12	54.57066	36.47952
<b>G_HPAS_NAYO_U1_U2</b>	Bus 68	5.58E-08	36.47952
<b>G_HPAS_OCAÑ_U1</b>	Bus 43	20.25862	36.47952
<b>G_HPAS_PAPA_U1</b>	Bus 69	2.16	36.47952
<b>G_HPAS_PAPA_U2</b>	Bus 81	4.08	36.47952
<b>G_HPAS_PASO_U1_U2</b>	Bus 64	1.64E-07	36.47952
<b>G_HPAS_PENI_U1_U3</b>	Bus 74	77.13559	36.47952
<b>G_HPAS_PENI_U4</b>	Bus 83	1.252047	36.47952
<b>G_HPAS_RBLA_U1</b>	Bus 84	3.0312	36.47952
<b>G_HPAS_RECU_U1</b>	Bus 61	1.95E-07	36.47952
<b>G_HPAS_SAUC_U1</b>	Bus 38	18.13713	36.47952
<b>G_HPAS_SAUC_U3</b>	Bus 39	18.13713	36.47952
<b>G_HPAS_SAUC_U4</b>	Bus 40	18.13713	36.47952
<b>G_HPAS_SAYM_U5</b>	Bus 41	32.09974	36.47952
<b>G_HPAS_SAYM_U7</b>	Bus 42	32.06693	36.47952
<b>G_HPAS_SFRA_U1</b>	Bus 55	86.47524	2.73865
<b>G_HPAS_SIBI_U1_U2</b>	Bus 20	14.535	36.47952
<b>G_HPAS_SMCA_U1</b>	Bus 82	1.848104	36.47952
<b>G_TMCI_CCAS_U1_U4</b>	Bus 59	0	36.47952
<b>G_TMCI_CES2_U1_U6</b>	Bus 08	7	36.47952
<b>G_TMCI_CES2_U7_U12</b>	Bus 09	7	36.47952
<b>G_TMCI_DAYU_U1</b>	Bus 56	0	36.47952
<b>G_TMCI_GHER_U1_U6</b>	Bus 63	5	36.47952
<b>G_TMCI_GROC_U1_U8</b>	Bus 50	4	36.47952
<b>G_TMCI_GUA2_U6</b>	Bus 62	20	36.47952
<b>G_TMCI_JARA_U1_U5</b>	Bus 04	0	36.47952
<b>G_TMCI_JARA_U6_U9</b>	Bus 05	0	36.47952
<b>G_TMCI_JIV3_U1</b>	Bus 58	0	36.47952
<b>G_TMCI_LAFA_U1_U4</b>	Bus 75	0	36.47952
<b>G_TMCI_LORE_U1</b>	Bus 57	0	36.47952

<b>G_TMCI_MAN2_U1_U12</b>	Bus 02	0	36.47952
<b>G_TMCI_MIRA_U11_U12</b>	Bus 06	5	36.47952
<b>G_TMCI_SEL2_U1_U53</b>	Bus 47	0	36.47952
<b>G_TMCI_SEL3_U1</b>	Bus 48	0	36.47952
<b>G_TMCI_TGUA_U1_U22</b>	Bus 45	18	36.47952
<b>G_TTGA_EQUI_U2</b>	Bus 51	40	36.47952
<b>G_TTGA_EQUI_U4</b>	Bus 49	40	36.47952
<b>G_TTGA_MAC1_U1</b>	Bus 19	50	8.525196
<b>G_TTGA_MAC2_TM1</b>	Bus 17	0	36.47952
<b>G_TTGA_MAC2_TM5</b>	Bus 18	0	36.47952
<b>G_TTVA_COAZ_U1</b>	Bus 16	15.04	36.47952
<b>G_TTVA_ECOE_U1</b>	Bus 14	13	36.47952
<b>G_TTVA_GZEV_TV3</b>	Bus 52	0	16.23992
<b>G_TTVA_ISCA_U1</b>	Bus 15	84.80032	36.47952
<b>G_TTVA_TRIN_U1</b>	Bus 46	25	23.56501