

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**HERRAMIENTAS COMPUTACIONALES PARA LA REALIZACIÓN  
DE FLUJOS DE POTENCIA EN SISTEMAS ELÉCTRICOS DE  
TRANSMISIÓN Y DISTRIBUCIÓN**

**HERRAMIENTA COMPUTACIONAL PARA LA REALIZACIÓN DE  
FLUJOS DE POTENCIA EN SISTEMAS ELÉCTRICOS DE  
TRANSMISIÓN**

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

**ROBERTO GABRIEL SUMBA CACERES**

**roberto.sumba@epn.edu.ec**

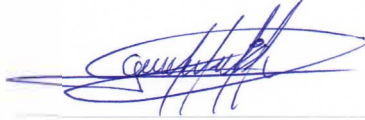
**DIRECTOR: HUGO NEPTALÍ ARCOS MARTÍNEZ**

**hugo.arcos@epn.edu.ec**

**DMQ, abril 2023**

## **CERTIFICACIONES**

Yo, ROBERTO GABRIEL SUMBA CACERES 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.



**Roberto Gabriel Sumba Caceres**

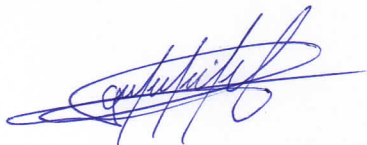
Certifico que el presente trabajo de integración curricular fue desarrollado por ROBERTO GABRIEL SUMBA CACERES, bajo mi supervisión.



**Phd. Hugo Arcos**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) 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.



ROBERTO GABRIEL SUMBA CACERES



HUGO NEPTALÍ ARCOS MARTÍNEZ

## DEDICATORIA

A todas aquellas personas especiales que ya no están físicamente conmigo, pero que siempre estarán en mi corazón y en mi mente. A aquella persona con la que tenía planes y sueños que no pudimos cumplir juntos por su trágica partida, a mi abuela que siempre me inspiró con su sabiduría y amor incondicional, y a mi hermano que se fue demasiado pronto, pero me dejó valiosas enseñanzas sobre la vida y la importancia de valorar cada momento. Este logro no sería posible sin el apoyo y guía de todas estas personas que, aunque no puedan celebrar conmigo en este momento, siempre estarán presentes en mi vida y en mis recuerdos.

## AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento al Dr. Hugo Arcos, mi tutor de TIC, por su guía, apoyo y paciencia a lo largo de todo el proceso. Desde el primer día, he admirado la pasión y dedicación que muestra en su enseñanza, lo cual ha sido fundamental para mi crecimiento académico y personal. Su cariño y compromiso con la carrera y sus estudiantes son evidentes en la forma en que ha brindado sus conocimientos y en la paciencia que ha tenido conmigo en momentos de dificultad.

También quisiera agradecer a mis hermanos, quienes han sido mi motor para seguir estudiando y superarme día a día. A mis amigos, quienes siempre han estado a mi lado, brindándome su consejo y acompañándome en las etapas difíciles que se presentaron en este camino.

De igual manera, no puedo olvidar agradecer a la Escuela Politécnica Nacional, por brindarme una formación académica de calidad y por ser una fuente constante de desafíos y dudas que, gracias a su exigencia, me han llevado a aclararlas y a creer que lo imposible se puede hacer posible.

Por último, pero no menos importante, quiero agradecer a mis padres, quienes han estado a mi lado en todo momento, a pesar de todas las dificultades y conflictos que hemos tenido. Me alegra saber que, juntos, hemos podido solucionarlos y seguir adelante como una familia unida.

Una vez más, gracias por todo su apoyo, orientación y motivación durante esta etapa de mi vida. Lo apreciaré por siempre.

Atentamente,

Gabriel Sumba

# ÍNDICE DE CONTENIDO

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VI
ABSTRACT.....	VII
1. Descripción del componente desarrollado .....	1
1.1. Objetivo general.....	1
1.2. Objetivos específicos .....	1
1.3. Alcance .....	1
1.4. Marco teórico .....	2
1.4.1. Conceptos Básicos .....	2
1.4.2. Modelación de Componentes de Sistemas Eléctricos de Potencia .....	4
1.4.3. Solución de Flujos de Potencia .....	20
2. Metodología.....	31
2.1. Estructura de la Herramienta de Software.....	32
2.1.1. Datos de entrada .....	34
2.1.2. Matriz de admitancia.....	36
2.1.3. Programación de los algoritmos de solución de flujos de potencia.....	37
2.2. Implementación en lenguaje de programación Python .....	40
2.2.1. Módulos del Software Desarrollado en Python .....	41
3. Resultados, Conclusiones y Recomendaciones .....	49
3.1. Resultados .....	49
3.1.1. Resultados de la aplicación desarrollada en Python.....	49
3.1.2. Comparación de Resultados entre PowerFactory y Aplicación desarrollada.....	55
3.2. Conclusiones .....	58
3.3. Recomendaciones .....	59
4. Referencias Bibliográficas .....	60
5. Anexos .....	61

## RESUMEN

En el presente Trabajo de Integración Curricular se ha desarrollado una herramienta computacional para la solución de flujos de potencia para sistemas eléctricos mallados. La herramienta cumple con los objetivos de implementar modelos matemáticos de transformadores, generadores y líneas de transmisión para estructurar modelos de redes de transmisión y determinar flujos de potencia, así como implementar un algoritmo mediante programación en lenguaje de código abierto para la realización de flujos de potencia utilizando los métodos de Gauss Seidel, Newton Raphson desacoplado y Flujo linealizado (DC).

Los resultados obtenidos con la herramienta desarrollada han sido comparados con los resultados obtenidos en el software comercial PowerFactory de Digsilent, lo cual ha permitido corroborar la utilidad del software programado en código abierto. La herramienta conceptualizada para generar un ambiente lúdico permite a docentes de sistemas de potencia de ingeniería eléctrica aplicar nuevos métodos de enseñanza en la solución de flujos de potencia, haciendo al proceso de enseñanza - aprendizaje más interactivo y práctico. En conclusión, el desarrollo de esta herramienta computacional representa una valiosa contribución para la formación de los estudiantes en esta importante área del conocimiento de la ingeniería eléctrica.

**PALABRAS CLAVE:** Herramienta computacional, Modelos matemáticos, Flujos de potencia, Comparación de resultados, Enseñanza de ingeniería eléctrica, análisis de sistemas eléctricos de potencia.

## **ABSTRACT**

In the present Curricular Integration Work, a computational tool has been developed for the solution of power flows for meshed electrical systems. The tool meets the objectives of implementing mathematical models of transformers, generators, and transmission lines to structure transmission network models and determine power flows, as well as implementing an algorithm through programming in open source language for the realization of power flows using the methods of Gauss Seidel, Newton Raphson decoupled and linearized flow (DC).

The results obtained with the developed tool have been compared with the result obtained in the commercial software PowerFactory from DlgSILENT, which has allowed corroborating the usefulness of the open source programmed software. The tool conceptualized to generate a playful environment allows electrical engineering power systems teachers to apply new teaching methods in the solution of power flows, making the teaching-learning process more interactive and practical. In conclusion, the development of this computational tool represents a valuable contribution to the training of students in this important area of electrical engineering knowledge.

**KEYWORDS:** Computational tool, Mathematical models, Power flows, Result comparison, Electrical engineering education, Power system analysis.



# **1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO**

El no disponer de herramientas informáticas específicas para mejorar el aprendizaje de los métodos de solución de flujos de potencia para estudiantes de ingeniería en electricidad, constituye una limitación en el proceso de formación académica. Por tal razón se ha creado una herramienta computacional que brinda un ambiente interactivo, flexible y lúdico, adaptado a las necesidades académicas.

La herramienta computacional permite al docente mejorar sus métodos de enseñanza y así desarrollar de mejor manera las capacidades y destrezas de sus alumnos en esta importante área de conocimiento de la ingeniería eléctrica.

## **1.1. Objetivo general**

Elaborar una Herramienta de software de código abierto, programada en lenguaje de programación Python, que permita realizar simulaciones de flujos de potencia de sistemas eléctricos de prueba.

## **1.2. Objetivos específicos**

1. Implementar los modelos matemáticos de transformadores, generadores y líneas de transmisión que permitan estructurar modelos de redes de transmisión para los cuales determinar flujos de potencia.
2. Implementar un algoritmo, mediante programación en lenguaje de código abierto, que permita la realización de flujos de potencia mediante los métodos de Gauss Seidel, Newton Raphson desacoplado y Flujo DC.
3. Analizar y corroborar los resultados de la solución de flujos de potencia obtenidos con los métodos de Gauss Seidel, Newton Raphson desacoplado y flujo DC. Para el efecto se compararán los resultados obtenidos con resultados del software comerciales PowerFactory de DigSILENT.

## **1.3. Alcance**

El presente Trabajo de Integración Curricular entregará como principal producto una herramienta computacional adaptada a las necesidades de aprendizaje de estudiantes que se encuentren cursando materias del área de Sistemas Eléctricos de Potencia. Además, el programa desarrollado con código abierto Python permitirá a los usuarios ver los resultados de voltaje de cada iteración que se haya realizado en el Método Gauss-Seidel, o la matriz jacobiana y voltajes en cada iteración en el método de Newton Raphson.

El programa desarrollado será utilizado para simular flujos de potencia en los sistemas de prueba IEEE de 9, 14 y 39 barras; y el sistema de 4 barras de Grainger & Stevenson. Los resultados obtenidos de las simulaciones serán validados por el software comercial PowerFactory de DlgSILENT.

## 1.4. Marco teórico

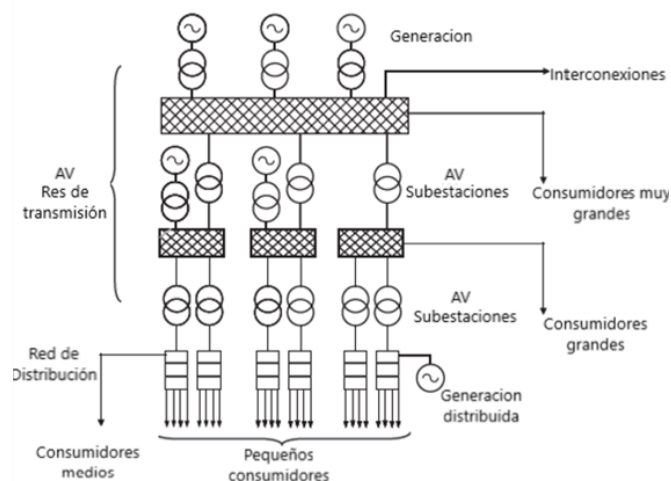
En esta sección se presentan los conceptos sobre la modelación matemática de los elementos que conforman un sistema eléctrico de potencia para la transmisión de energía eléctrica, su integración en un modelo matemático de sistema, la estructuración del sistema de ecuaciones a ser resuelto y los métodos de solución para la obtención de flujos de potencia.

### 1.4.1. Conceptos Básicos

#### 1.4.1.1. Sistema eléctrico de Potencia

Un sistema eléctrico de potencia (SEP) se define como la interconexión de elementos eléctricos, mecánicos y electrónicos, que convierte un tipo de energía primaria en energía eléctrica. Dicha energía es transportada para luego distribuirla hacia los consumidores finales.

Un sistema eléctrico de potencia está constituido por tres partes: Generación, transmisión y distribución como se muestra en la Figura 1.1.



**Figura 1.1.** Partes del Sistema Eléctrico de Potencia [1].

#### Generación

Es la parte del SEP encargada de convertir algún tipo de energía (cinética, potencial, nuclear, térmica, solar, etc.) en energía eléctrica. El Dispositivo que se utiliza con más

frecuencia para dicha conversión es el denominado generador, el cual utiliza el principio de conversión electromagnética de energía, es decir, utiliza el movimiento de un conductor (circuito cerrado) dentro de un campo magnético para inducir un voltaje en dicho conductor [2]. Para el caso de un generador, la turbina acoplada al rotor mueve el campo magnético generado por una corriente de excitación inyectada a los devanados del rotor para inducir voltaje en los devanados del estator de la máquina. Para la conversión de energía no solo se utiliza generadores, también se utiliza paneles solares que absorben la radiación solar para producir electricidad que puede ser almacenada o transferida al sistema de potencia.

Las instalaciones donde se encuentran los dispositivos utilizados para la conversión de energía se denominan centrales eléctricas, las cuales en la mayoría de las veces se ubican cerca de las fuentes de energía primaria como es el caso de las centrales hidroeléctricas, geotérmicas, biogás, etc. [1].

### Transmisión

La transmisión o red de transmisión es la parte encargada de transportar grandes cantidades de energía eléctrica desde los convertidores de energía (centrales eléctricas) hacia los consumidores mediante conductores de alta tensión. Es decir, la transmisión utiliza conductores (líneas) como un medio físico por el que fluye la energía hacia los centros de consumo.

Para evitar pérdidas en el transporte de energía debido a las grandes distancias físicas que existen entre la producción y consumo de energía, se eleva el nivel de voltaje a través de transformadores. Un transformador eléctrico es un dispositivo que mediante la acción de un campo magnético cambia el voltaje de la red, manteniendo la frecuencia y la potencia [3].

Otro componente fundamental en el sistema de transmisión son las subestaciones, las cuales tienen 3 funciones principales: permiten la interconexión de líneas, son los nodos de transformación que alimenta a la red de distribución y es donde se ubican los equipos de medición y protección del sistema [1]. Por lo general, las líneas de alto voltaje alimentan a las subestaciones, donde se reduce el voltaje a un nivel de utilización.

### Distribución

La distribución o redes de distribución permiten entregar energía eléctrica a los consumidores finales (hogares, industrias, a la iluminación urbana, iluminación ornamental, etc.). Por seguridad de los consumidores la energía es repartida a niveles bajos de voltaje, esto se logra por medio de los transformadores de distribución.

## 1.4.2. Modelación de Componentes de Sistemas Eléctricos de Potencia

### 1.4.2.1. Transformadores

El transformador es un elemento del sistema eléctrico de potencia que idealmente, eleva o disminuye el nivel de voltaje sin afectar la potencia que se está suministrando. Cuando el transformador eleva el voltaje, la corriente disminuye o en otra instancia si el transformador reduce el voltaje, la corriente incrementa. Esto para mantener la misma potencia en ambos lados del transformador.

En un sistema de potencia los transformadores se utilizan para elevar el voltaje de las terminales de los generadores a niveles de transmisión con el objetivo de transportar la energía eléctrica a largas distancias con bajas pérdidas. Por último, un transformador reduce el voltaje de transmisión a uno de utilización para los usuarios [3].

#### Transformador monofásico

Un modelo aproximado del transformador debe considerar las pérdidas que ocurren en la operación de un transformador como son [3]:

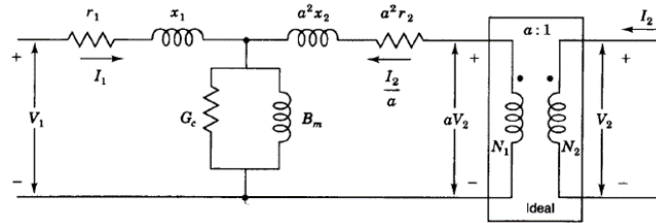
- Pérdidas en el cobre, debido a la oposición al paso de la corriente en los devanados del transformador.
- Pérdidas por corrientes parásitas, debido al calentamiento del núcleo del transformador.
- Pérdidas por histéresis, relacionadas al reordenamiento de los dominios magnéticos en el núcleo en cada semiciclo.
- Flujos dispersos, Son los flujos que escapan del núcleo y que producen una autoinductancia en las bobinas primaria y secundaria.

En la práctica, cuando se deja abierto el lado secundario del transformador, el lado primario toma algo de corriente de la fuente, esa corriente se denomina corriente de excitación y es la suma de la corriente de pérdidas en el núcleo  $I_c$  y la corriente de magnetización  $I_m$ .

$$I_\phi = I_c + I_m \quad (1.1)$$

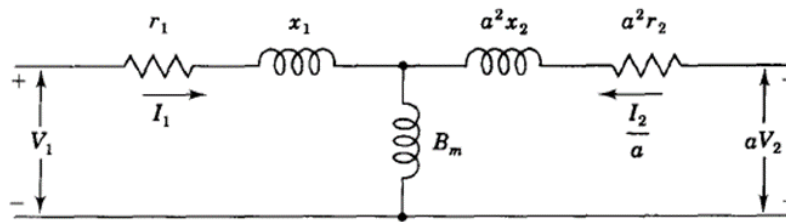
La corriente de pérdida en el núcleo explica la pérdida por histéresis y por corrientes parásitas y da lugar a la resistencia  $R_c = \frac{1}{G_c}$  [4].

La corriente de magnetización establece el flujo mutuo en el núcleo, su comportamiento es como el de una bobina por lo cual da lugar a una reactancia de magnetización  $X_m = \frac{1}{B_m}$ .



**Figura 1.2.** Circuito equivalente de un transformador junto con la corriente de excitación y magnetización [4].

En un sistema de potencia se tiene que las pérdidas por el calentamiento del núcleo se consideran despreciables y los demás parámetros se describirán a continuación.



**Figura 1.3.** Circuito equivalente de un transformador de dos devanados [5].

$r_1$ : Resistencia del devanado primario.

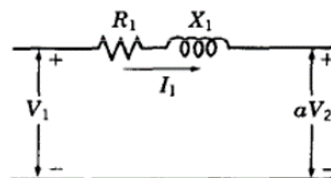
$x_1$ : Reactancia de dispersión del devanado primario.

$a^2 x_2$ : Reactancia de dispersión del devanado secundario reflejado al lado primario.

$a^2 r_2$ : Resistencia del devanado secundario reflejado al lado primario.

$B_m$ : Susceptancia de magnetización.

Para simplificar más el circuito se desprecia la corriente de excitación ya que su valor es muy pequeño en comparación con las elevadas corrientes que requieren las cargas por lo cual su circuito equivalente quedaría de la siguiente manera.



**Figura 1.4.** Circuito equivalente sin considerar la corriente de excitación [5].

Donde:

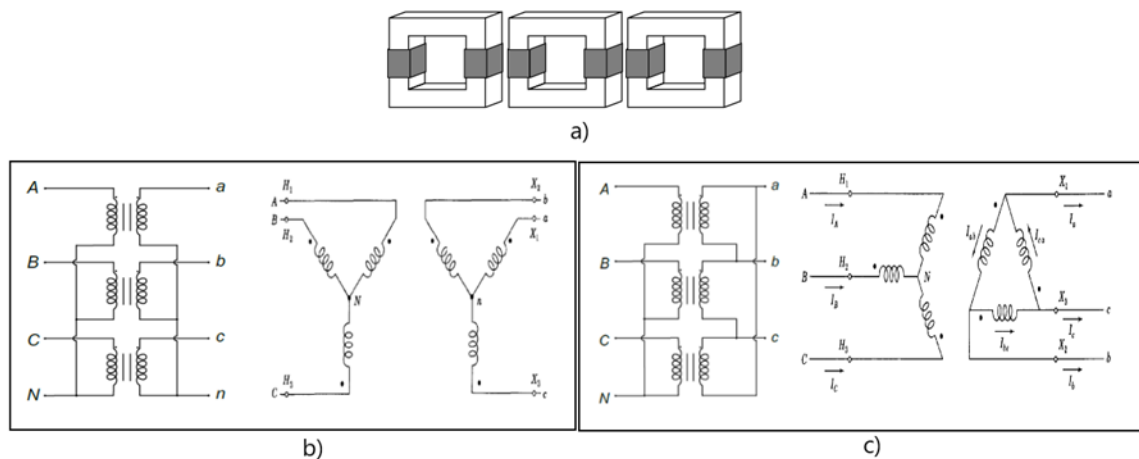
$$R_1 = r_1 + a^2 r_2 \quad X_1 = x_1 + a^2 x_2 \quad (1.2)$$

Para obtener las resistencias y reactancias en por unidad se debe obtener la impedancia base considerando como voltaje base, el voltaje nominal de donde están reflejadas las resistencias y reactancias del circuito equivalente del transformador. Por ejemplo, si la reactancia y la resistencia equivalente del transformador se encuentran reflejado en el lado primario entonces como voltaje base se utilizará el voltaje nominal del lado primario del transformador. Una ventaja de utilizar valores en por unidad en un transformador es que no se requiere referir la impedancia del transformador hacia el otro lado ya que tiene el mismo valor.

Como dato adicional los fabricantes suelen especificar la impedancia del transformador, generador, etc. en porcentaje o en por unidad, cuyas cantidades bases son los valores nominales de placa.

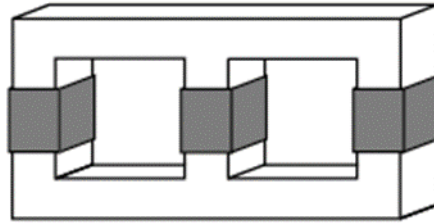
Transformador trifásico.

Un transformador trifásico se puede formar por tres transformadores monofásicos que por un lado estén conectados en Y o Δ y por el otro lado, de igual manera, conectados en Y o Δ como se muestra en la Figura 1.5.



**Figura 1.5.** a) Bancos trifásico de transformadores monofásicos b) Diagramas de conexión YY. c) Diagramas de conexión YΔ [5].

Existen varias conexiones además de YY y YΔ de la Figura 1.5 como la conexión ΔY y la conexión ΔΔ. No solo se puede usar tres transformadores monofásicos, también se puede usar una unidad trifásica como se muestra en la Figura 1.6.



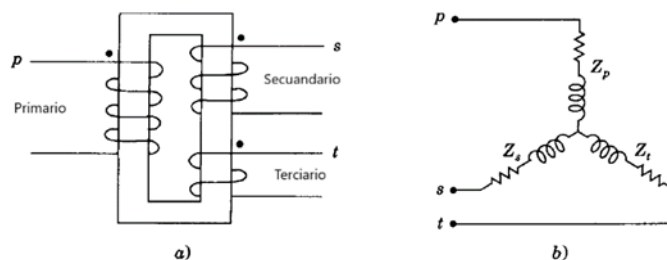
**Figura 1.6.** Unidad trifásica de tres fases sobre una misma estructura de acero [1].

Ambos transformadores operan en base a la misma teoría con la diferencia que la unidad trifásica es más económica debido a la menor cantidad de acero en el núcleo y además ocupa menos espacio que el banco de transformadores. Por otro lado, el banco trifásico permite que uno de sus transformadores monofásicos pueda ser removido en caso de presentarse una falla, los dos restantes pueden seguir operando a menores KVA, cosa que no pasa en una unidad trifásica, si sucede una falla en la unidad trifásica todo el banco se perdería y dejaría de operar [5].

Transformador de tres devanados.

Muy a menudo en los sistemas de potencia los transformadores suelen tener tres devanados, permitiéndoles operar a tres niveles de voltajes diferentes con capacidades diferentes. Es decir, la potencia nominal entre el primario y el secundario es distinta a la potencia entre el primario y terciario. Las impedancias del transformador generalmente son expresadas en por unidad, sobre las mismas bases de voltaje y potencia.

En la Figura 1.7 se puede ver el diagrama esquemático de un transformador monofásico de tres devanados junto con su circuito equivalente despreciando la corriente de magnetización. Se debe mencionar que el punto de conexión en común de los devanados no se debe asumir que es el neutro en común.



**Figura 1.7.** a) Diagrama esquemático del transformador trifásico b) Circuito equivalente

Las pruebas de cortocircuito del transformador permiten determinar las impedancias de dispersión de cada devanado ( $Z_p, Z_s, Z_t$ ), de la siguiente manera [5]:

Se cortocircuita uno de los tres lados del transformador mientras que otro lado es alimentado por una fuente variable permitiendo así obtener la corriente especificada del transformador, el devanado que queda libre se lo deja abierto. Mediante las lecturas de voltímetro, amperímetro y vatímetro colocados en el lado de la fuente se puede obtener las impedancias de dispersión:

$Z_{ps}$  Medición en el primario, secundario cortocircuitado y terciario abierto.

$Z_{pt}$  Medición en el primario, terciario cortocircuitado y secundario abierto.

$Z_{st}$  Medición en la secundaria, terciario cortocircuitado y primario abierto.

Las impedancias obtenidas por las pruebas de cortocircuito se pueden referir al voltaje de un lado del transformador para obtener la impedancia por separado como se muestra a continuación.

$$Z_{ps} = Z_p + Z_s \quad (1.3)$$

$$Z_{pt} = Z_p + Z_t \quad (1.4)$$

$$Z_{st} = Z_s + Z_t \quad (1.5)$$

Donde:  $Z_p$ ,  $Z_s$  y  $Z_t$  son las impedancias en los devanados primario, secundario y terciario del circuito equivalente del transformador. Mediante las ecuaciones mostradas anteriormente se puede obtener.

$$Z_p = \frac{1}{2}(Z_{ps} + Z_{pt} - Z_{st}) \quad (1.6)$$

$$Z_s = \frac{1}{2}(Z_{pt} + Z_{st} - Z_{ps}) \quad (1.7)$$

$$Z_t = \frac{1}{2}(Z_{ps} + Z_{st} - Z_{pt}) \quad (1.8)$$

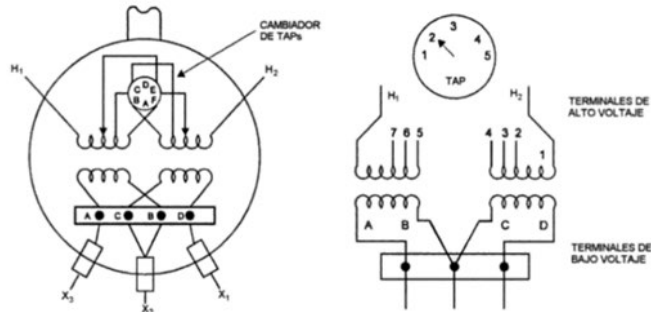
Un transformador trifásico de tres devanados puede formarse por tres transformadores monofásicos de tres devanados donde los primarios y secundarios son conectados en Y y los terciarios en  $\Delta$ .

### Transformador con taps.

El sistema eléctrico de potencia no es ideal, por tal motivo, existe variaciones de voltaje que no son deseados y por tal motivo se incorpora en el SEP los transformadores con taps. Estos transformadores son construidos de tal manera puedan cambiar su relación de transformación para aumentar o disminuir el voltaje en sus terminales. Es decir, aumentar

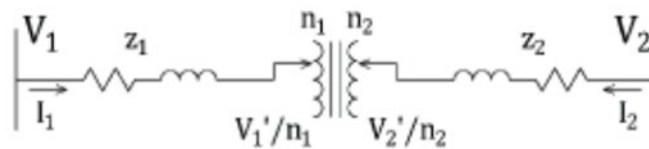


o disminuir el número de vueltas del transformador mediante el cambio de taps [6], el cambiador de taps generalmente se ubica en el lado de alto voltaje del transformador y para maniobrarlo se debe desconectar la carga.



**Figura 1.8.** Cambiador de Derivación (TAPS) [6].

La Figura 1.8 muestra de manera generalizada un transformador con taps en ambos lados.



**Figura 1.9.** Transformador de dos devanados con taps en ambos lados [7].

Donde:

$z_1$ : Impedancia del lado primario.

$z_2$ : Impedancia del lado secundario.

$n_1$ : Posición del taps del lado primario.

$n_2$ : Posición del taps del lado secundario.

La relación de voltajes ( $V'_1, V'_2$ ) y corrientes del transformador de la Figura 1.9 es la siguiente:

$$\frac{V_1 - z_1 I_1}{n_1} = \frac{V_2 - z_2 I_2}{n_2} \quad (1.9)$$

$$\frac{I_1}{n_2} = -\frac{I_2}{n_1} \quad (1.10)$$

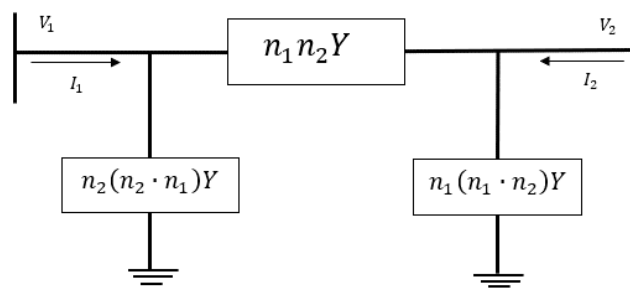
Mediante las ecuaciones se pueden obtener  $I_1$  e  $I_2$  :

$$I_1 = n_1 n_2 Y (V_1 - V_2) + n_2 (n_2 - n_1) Y V_1 \quad (1.11)$$

$$I_2 = n_1 n_2 Y (V_1 - V_2) + n_1 (n_1 - n_2) Y V_2 \quad (1.12)$$

$$Y = \frac{1}{z_2 n_1^2 + n_2^2 z_1} \quad (1.13)$$

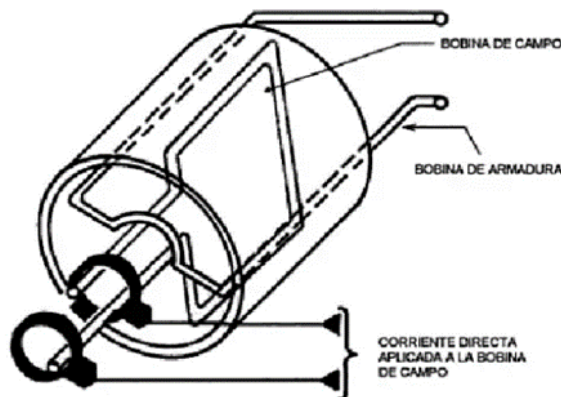
De estas ecuaciones se puede obtener el  $\pi$  del transformador de la siguiente manera.



**Figura 1.10.** Modelo  $\pi$  del transformador de dos devanados con taps en ambos lados [7].

#### 1.4.2.2. Generadores Sincrónicos

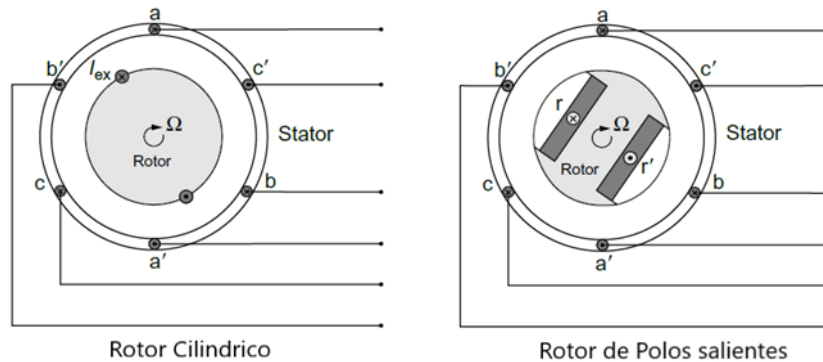
Los generadores sincrónicos son máquinas cuyo propósito es el de convertir la energía mecánica en energía eléctrica AC, constituido por 2 partes principales; una parte estacionaria y una móvil. La parte estacionaria conocida como estator o armadura tiene ranuras donde se ubica las bobinas de armadura, la parte móvil del generador sincrónico se la conoce como rotor y es la parte que rota dentro del estator, al igual que el estator, el rotor tiene ranuras donde se ubican las bobinas de campo.



**Figura 1.11.** Generador sincrónico

Las bobinas de campo son alimentadas por una fuente DC (excitatriz), usualmente constituida por una fuente AC y elementos activos que permitan la conversión de AC a DC.

Según la construcción del rotor de la maquina sincrónica se puede dividir en: rotor cilíndrico y rotor de polos salientes. La diferencia se encuentra en su velocidad, bobinas y entre otras características.

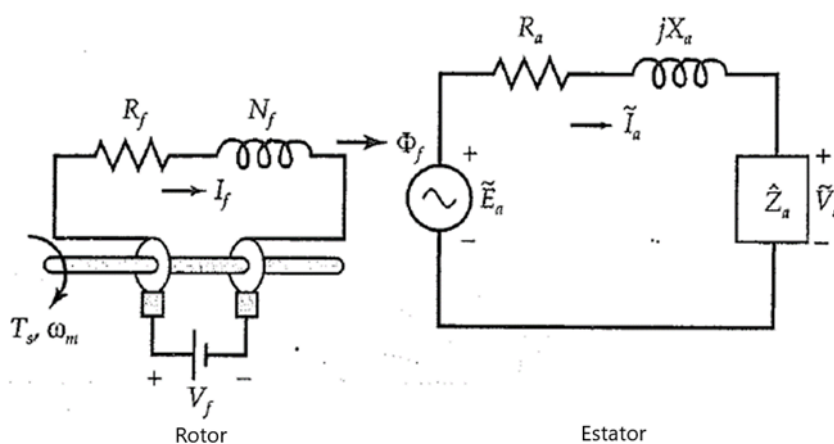


**Figura 1.12.** Tipos de máquinas sincrónicas

Rotor cilíndrico

El rotor cilíndrico tiene en toda su periferia varias ranuras que llevan el bobinado de campo. Esto con el fin de adaptarlo a máquinas de alta velocidad como es el caso de los turbogeneradoras. Usualmente este tipo de rotor tiene de 2 a 4 polos.

El circuito equivalente de la maquina sincrónica de rotor cilíndrico se puede ver en la Figura 1.13. El voltaje en las terminales del generador  $V_a$  usualmente es menor que el voltaje generado, esto debido a la caída de voltaje ( $I_a R_a$ ) en la resistencia de la armadura y la caída de voltaje ( $jI_a X_a$ ) en la reactancia de dispersión.



**Figura 1.13.** Circuito equivalente del generador de rotor cilíndrico por fase [4].

$E_a$ : Voltaje generado por bobinado

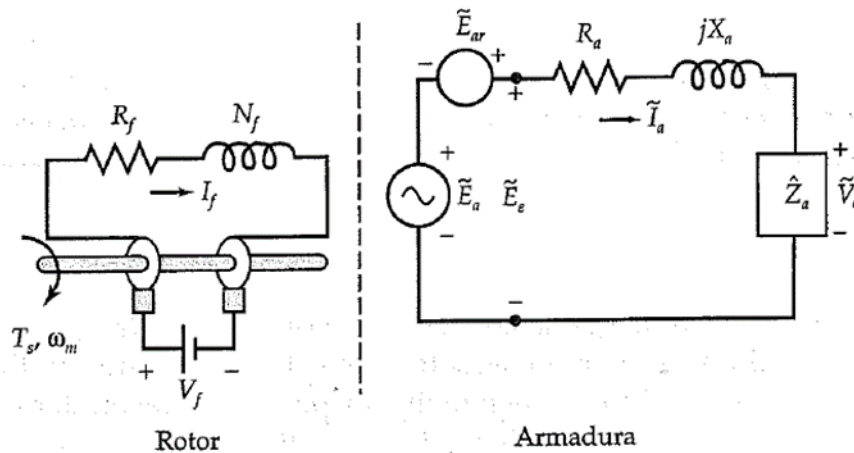
$I_a$ : Corriente de armadura cuando se conecta una carga.

$R_a$ : Resistencia de la armadura.

$X_a$ : Reactancia de dispersión.

Usualmente el voltaje en las terminales del generador es menor que el voltaje generado, sin embargo, esto depende del factor de potencia de la carga; si el factor de potencia está en adelanto  $V_a$  será mayor que  $E_a$ ; si el factor de potencia está en atraso o es igual a la unidad  $V_a$  es menor que  $E_a$ .

La corriente que circula por las bobinas de armadura produce un flujo el cual interactúa con el flujo del rotor, autoinduciendo fuerza electromotriz en las bobinas de armadura  $E_{ar}$ , que esta en atraso con respecto a  $I_a$  en  $90^\circ$ . Por consiguiente, el circuito equivalente se modifica como se muestra en la Figura 1.14.



**Figura 1.14.** Circuito equivalente considerando la reacción de armadura [4].

Las expresiones obtenidas del circuito equivalente son:

$$E_e = V_a + I_a(R_a + jX_a) \quad (1.14)$$

Y donde la reactancia de armadura se expresa como:

$$E_{ar} = -jI_a X_m \quad (1.15)$$

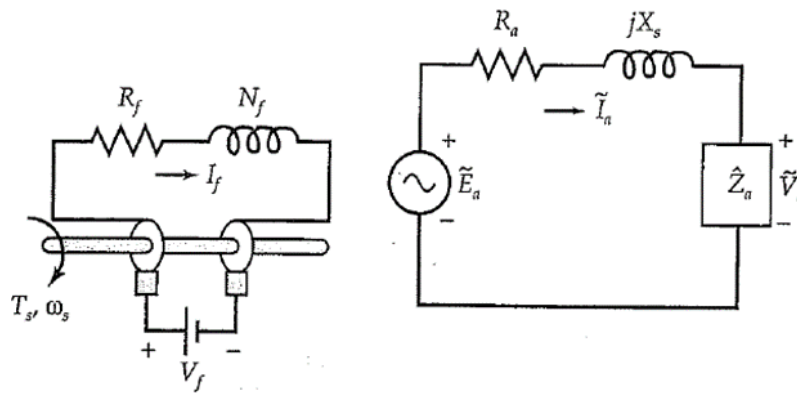
Donde  $X_m$  es la reactancia de magnetización, una constante proporcional.

$$V_a = E_a - I_a(R_a + j(X_m + X_a)) \quad (1.16)$$

El circuito se puede simplificar aún más si se considera la reactancia síncrona  $X_s$  como:

$$X_s = X_m + X_a \quad (1.17)$$

Y por ende el circuito equivalente se podría representar de la siguiente manera.



**Figura 1.15.** Circuito equivalente del generador de rotor cilíndrico por fase [4].

Donde el voltaje de armadura se puede hallar mediante,

$$V_a = E_a - I_a(R_a + jX_s) \quad (1.18)$$

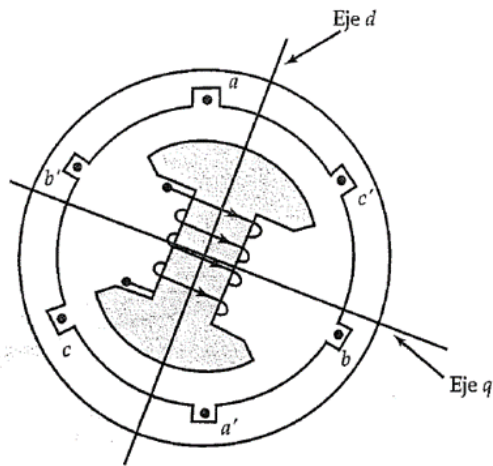
### Rotor de polos salientes

El rotor de polos salientes es el más utilizado en aplicaciones de baja velocidad como es el caso de los hidro-generadores y grupos electrógenos a diésel, según la necesidad puede tener más de 4 polos. Físicamente el rotor de polos salientes tiene un entrehierro más grande como se ve en la Figura 1.16.

Para la modelación del circuito equivalente la reactancia síncrona se separa en dos reactancias; la reactancia de eje directo  $x_d$  y la reactancia de eje de cuadratura  $x_q$ . Al igual que se presentaba la reacción de armadura en la máquina de rotor cilíndrico, en este caso se produce dos fuerzas electromotrices; en el eje directo  $E_d$  y en el eje de cuadratura  $E_q$ , que se expresan como:

$$E_d = -jI_d X_q \quad (1.19)$$

$$E_q = -jI_q X_d \quad (1.20)$$



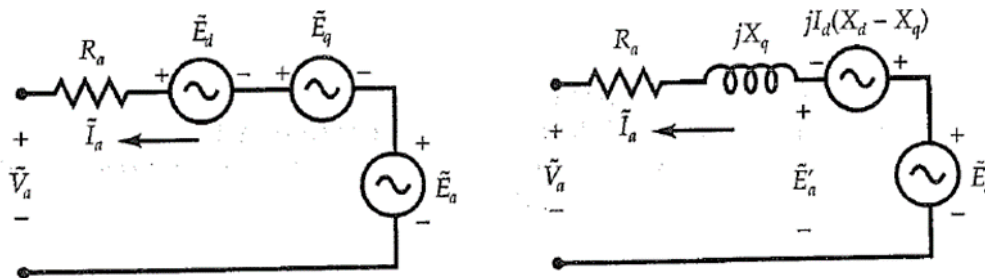
**Figura 1.16.** Generador sincrónico de polos salientes (2 polos) [4].

El voltaje en las terminales del generador  $V_a$  se puede hallar mediante:

$$V_a = E_a + E_d + E_q - I_a R_a \quad (1.21)$$

Sustituyendo  $E_d$  y  $E_q$  de la ecuación 1.1 y expresando  $jI_d X_d$  como  $jI_d X_q + jI_d (X_d - X_q)$  se tiene que,

$$E_a = E'_a + jI_d (X_d - X_q) \quad (1.22)$$



**Figura 1.17.** Circuito equivalente del generador sincrónico de rotor de polos saliente [4].

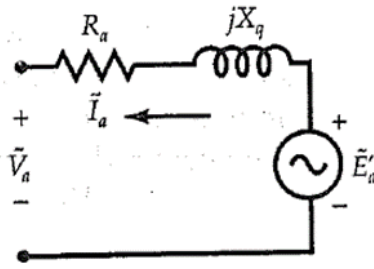
Donde,

$$E'_a = V_a + I_a R_a + I_a X_q \quad (1.23)$$

y

$$I_a = I_d + I_q \quad (1.24)$$

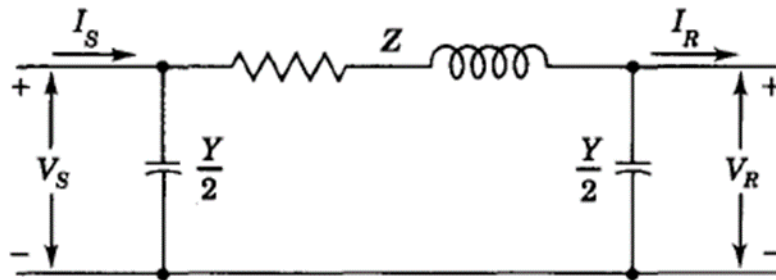
De esta manera el circuito equivalente del generador sincrónico de rotor de polos salientes quedaría como se muestra en la Figura 1.18.



**Figura 1.18.** Circuito equivalente del generador síncrono de polos salientes [4].

### 1.4.2.3. Líneas de transmisión

En un sistema de potencia existen tres tipos de líneas de transmisión según su longitud: líneas de transmisión de longitud corta (menores a 80 km), de longitud media (entre 80 y 240 km) y de longitud larga (mayores a 240 km). Para el caso de estudios de flujos de potencia lo más usual es considerar líneas de transmisión de longitud media. En la Figura 1.19 se presenta el circuito equivalente de una línea de transmisión media (circuito nominal  $\pi$ ), donde la admitancia (propiedad capacitiva de la línea) se coloca en los extremos de la línea en partes iguales para poder representar la distribución uniforme del parámetro capacitivo de la línea [5].



**Figura 1.19.** Circuito equivalente de una línea de transmisión media

Donde,

$V_R$ : Voltaje en el extremo receptor

$V_s$ : Voltaje en el extremo generador

$Y$ : Capacitancia de la línea de transmisión.

$I_s$ : corriente en el extremo generador

$I_R$ : Corriente en el extremo receptor

$Z$ : Impedancia de la línea de transmisión (resistencia y reactancia).

La conductancia en una línea de transmisión no es considerada en los modelos de líneas ya que toma en cuenta las corrientes de fugas presentes en los aisladores de la línea, esta corriente puede llegar a ser despreciable además de ser un valor que cambia conforme se modifican los factores externos como la contaminación en los aisladores y las condiciones atmosféricas.

#### 1.4.2.4. Diagrama unifilar

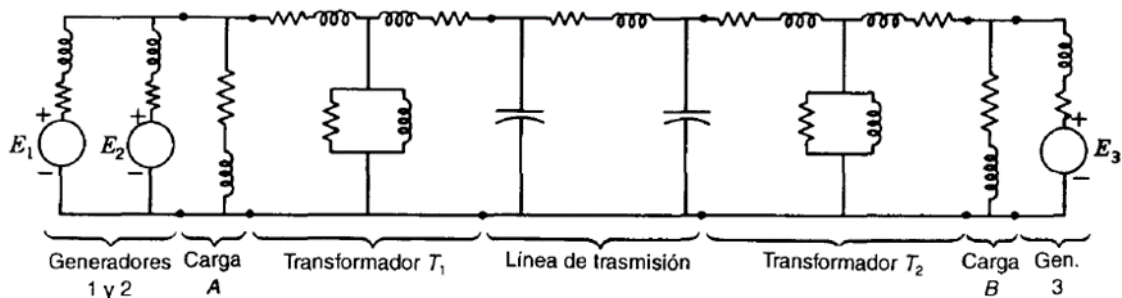
De forma práctica, la representación de un sistema de potencia se realiza mediante un diagrama unifilar o diagrama de una sola línea, donde los conductores son representados por una línea y los componentes como transformadores, generadores, interruptores de potencia, etc., son representados por símbolos estandarizados publicados por ANSI o IEEE [5].



**Figura 1.20.** Diagrama unifilar de un sistema eléctrico de potencia [5].

En la Figura 1.20 se muestra un diagrama unifilar de un sistema con 3 generadores estereos, 2 cargas (A y B) y dos transformadores (YY y YΔ).

Un sistema eléctrico de potencia es modelado mediante la unión de sus componentes (transformadores, generadores, líneas de transmisión, etc.) con su respectivo equivalente monofásico ya que se considera que los sistemas de potencia son trifásicos balanceados. Los circuitos equivalentes monofásicos se pueden combinar de manera de formar un diagrama de impedancias del sistema completo, como se muestra en la Figura 1.21.



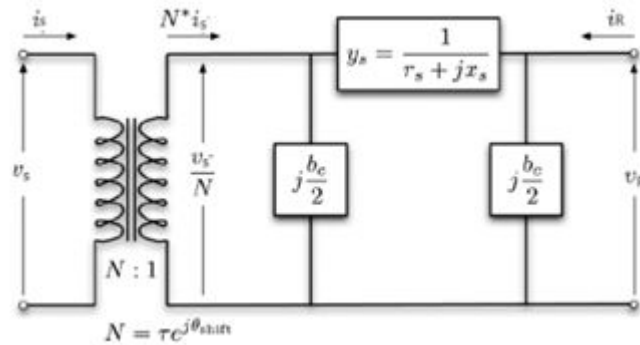
**Figura 1.21.** Diagrama de impedancias del diagrama unifilar de la Figura 1.20



El diagrama de impedancias se puede reducir considerando el modelo reducido de los componentes que conforman el sistema de potencia de la Figura 1.25.

#### 1.4.2.5. Rama Generalizada

Las líneas de transmisión y transformadores con taps pueden ser modelados como un modelo de rama en común [8], considerando los modelos  $\pi$  de la Figura 1.10 y la Figura 1.19, se obtiene la Figura 1.22.



**Figura 1.22.** Modelo generalizado de una rama del Sistema Eléctrico de Potencia.[8].

Donde:

$i_s$  Corriente inyectada en el extremo de envío

$v_s$  Voltaje en el extremo de envío

$i_R$  Corriente inyectada en el extremo de recepción

$v_R$  Voltaje en el extremo de recepción.

Las corrientes inyectadas tanto en las barras de envío como las de recepción de las ramas se pueden expresar en términos de la matriz de admitancia de rama de 2x2,  $Y_{br}$ ; y sus respectivos voltajes  $v_s$  y  $v_R$ , como:

$$\begin{bmatrix} i_s \\ i_R \end{bmatrix} = Y_{br} \begin{bmatrix} v_s \\ v_R \end{bmatrix} \quad (1.25)$$

$Y_{br}$ , matriz 2x2 puede denotarse con el elemento de admitancia en serie,  $y_s = 1/z_s$ , como se muestra a continuación:

$$Y_{br} = \begin{bmatrix} \left(y_s + j\frac{b_c}{2}\right)t^2 & -\frac{y_s}{t} \\ -\frac{y_s}{t} & y_s + j\frac{b_c}{2} \end{bmatrix} \quad (1.26)$$

$$Y_{br} = \begin{bmatrix} Y_{SS} & Y_{SR} \\ Y_{RS} & Y_{RR} \end{bmatrix} \quad (1.27)$$

Donde, los elementos de la diagonal,  $y_{SS}$  y  $y_{RR}$ , corresponden a los elementos propios de la barra de envío y recepción respectivamente. Mientras que los elementos fuera de la diagonal representan los elementos conectados entre la barra de envío y recepción

Tener un modelo generalizado de la rama de un sistema de potencia permite calcular la matriz de admitancias de un sistema de manera eficiente [8].

#### 1.4.2.6. Admitancias Shunt

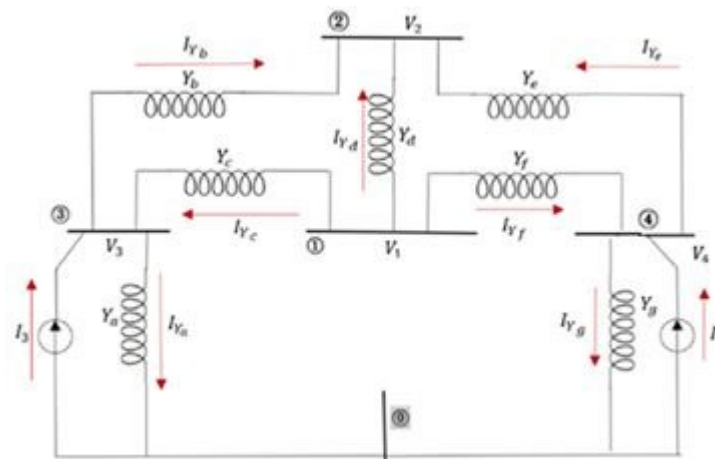
Los capacitores o inductores conectados en derivación a las barras pueden modelarse como una admitancia fija conectada a tierra.

$$y_{sh} = g_{sh} + jb_{sh} \quad (1.28)$$

Esta admitancia se suma a las correspondientes admitancias propias  $y_{SS}$  o  $y_{RR}$  de la matriz  $Y_{br}$ , ya que corresponde a elementos propios de cada barra y así obtener la matriz de admitancias de todo el sistema eléctrico de potencia propuesto.

#### 1.4.2.7. Relación de Voltajes y Corrientes de un sistema de transmisión

Teniendo como referencia la Figura 1.23 donde, los números de nodo están dentro de círculos;  $Y$ , es la admitancia (el inverso de la impedancia);  $V$ , es el voltaje en el nodo y tierra (nodo ①) y entre el nodo ③-① y ④-① se encuentran las fuentes de corriente ( $I_3$  e  $I_4$ ).



**Figura 1.23.** Diagrama de admitancias del circuito de referencia para obtención de las ecuaciones de nodo [5].

Se utiliza la admitancia en vez de la impedancia ya que resulta más conveniente para aplicar las leyes de corrientes de Kirchhoff en cada nodo como se muestra a continuación.

Leyes de corrientes de Kirchhoff en el nodo 1.

$$V_1(Y_c + Y_d + Y_f) - Y_d V_2 - Y_c V_3 - Y_f V_4 = 0 \quad (1.29)$$

Leyes de corrientes de Kirchhoff en el nodo 2.

$$-Y_d V_1 + (Y_b + Y_d + Y_e) V_2 - Y_b V_3 - Y_e V_4 = 0 \quad (1.30)$$

Leyes de corrientes de Kirchhoff en el nodo 3.

$$-Y_c V_1 - Y_b V_2 + (Y_a + Y_b + Y_c) V_3 = I_3 \quad (1.31)$$

Leyes de corrientes de Kirchhoff en el nodo 4.

$$-Y_f V_1 - Y_e V_2 + (Y_e + Y_f + Y_g) V_4 = I_4 \quad (1.32)$$

El nodo de referencia, nodo  $\textcircled{0}$ , no se le aplica las leyes de corriente de Kirchhoff ya que esta no brindaría información adicional. De aquí, se concluye que el número de ecuaciones de nodo requerida es un nodo menos de todos los nodos en el sistema.

Las 4 ecuaciones de nodo se pueden expresar de manera matricial de la siguiente manera:

$$\begin{bmatrix} Y_c + Y_d + Y_f & -Y_d & -Y_c & -Y_f \\ -Y_d & Y_b + Y_d + Y_e & -Y_b & -Y_e \\ -Y_c & -Y_b & Y_a + Y_b + Y_c & 0 \\ -Y_f & -Y_e & 0 & Y_e + Y_f + Y_g \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_3 \\ I_4 \end{bmatrix} \quad (1.33)$$

De la matriz se puede deducir que de manera general para un circuito de 4 nodos y el nodo de referencia la ecuación del análisis nodal se tiene:

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \quad (1.34)$$

La primera matriz de la izquierda se le denomina  $Y_{barra}$ , matriz de admitancias de barra. Y para formar los elementos de matriz  $Y_{barra}$  se sigue las siguientes reglas.

- Los elementos de la diagonal  $Y_{ii}$  es igual a la suma de todas las admitancias conectadas en el nodo  $i$ .
- Los elementos fuera de la diagonal  $Y_{ij}$  es igual al negativo de las admitancias conectadas entre el nodo  $i$  y  $j$ .

Este criterio se puede expandir para cualquier número de nodos presentes en un sistema eléctrico de potencia.

### 1.4.3. Solución de Flujos de Potencia

Un estudio de flujo de potencia es un análisis en estado estable el cual permite determinar voltajes, corrientes y transferencias de potencia activa y reactiva de un sistema eléctrico bajo determinadas condiciones de carga y generación. La finalidad de los estudios de flujos de potencia es simular diferentes condiciones operativas en las que se considera al sistema completo como ante la presencia de eventos de indisponibilidad de sus elementos.

#### 1.4.3.1. Ecuaciones del Análisis de Flujo de Potencia

La ecuación básica para el análisis de flujos de potencia se deriva de las ecuaciones nodales del sistema eléctrico de potencia. Por ejemplo, para un sistema de cuatro barras, se tiene:

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \quad (1.35)$$

La ecuación nodal de la barra o nodo  $i$  se puede escribir como:

$$I_i = \sum_{j=1}^n Y_{ij} V_j \quad (1.36)$$

La relación entre la potencia real y reactiva en por unidad suministrada al sistema en la barra  $i$  y la corriente en por unidad inyectada en el sistema por esa barra es:

$$S_i = V_i I_i^* = P_i + jQ_i \quad (1.37)$$

Donde:

$V_i$ : Voltaje en por unidad de la barra  $i$ .

$I_i^*$ : Complejo conjugado de la corriente inyectada en la barra  $i$ .

$P_i$  y  $Q_i$ : Potencia activa y reactiva en por unidad.

También:

$$Y_{ij} = G_{ij} + j B_{ij} \quad (1.38)$$

$$V_i = |V_i| \angle \delta_{ij} \quad (1.39)$$

Por lo tanto:

$$S_i = V_i \angle \theta_i \left[ \sum_{j=1}^n (G_{ij} + j B_{ij}) (V_j \angle \delta_j) \right]^* \quad (1.40)$$

$$P_i + Q_i = V_i \sum_{j=1}^n (G_{ij} + j B_{ij}) \cdot V_j \angle (\delta_i - \delta_j) \quad (1.41)$$

La ecuación puede expandirse e igualar su parte real (potencia activa) y parte imaginaria (potencia reactiva) para obtener:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \delta_{ij} + B_{ij} \sen \delta_{ij}) V_j \approx P_{Gi} - P_{Ci} \quad (1.42)$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \delta_{ij} + B_{ij} \cos \delta_{ij}) V_j \approx Q_{Gi} - Q_{Ci} \quad (1.43)$$

Los errores de potencia es la diferencia entre la potencia especificada y la potencia calculada.

$$\Delta P_i = P_i^{esp} - P_i^{cal} \quad (1.44)$$

$$\Delta Q_i = Q_i^{esp} - Q_i^{cal} \quad (1.45)$$

La potencia activa y reactiva, el módulo y ángulo del voltaje, son variables asociadas con cada barra. En un estudio de flujos de potencia, dos de las cuatro variables son definidas y las otras dos son desconocidas, pudiéndose plantear, en cada nodo, el mismo número de ecuaciones que de incógnitas.

Las variables conocidas y desconocidas dependen del tipo de barra las cuales se califican en tres tipos, como se detalla a continuación:

1. **Barra de carga (barra P-Q):** Es una barra en la cual la potencia activa y reactiva son especificadas y el voltaje será calculado. Las barras de carga son barras que no cuentan con generadores. En esta barra  $V$  y  $\delta$  son desconocidos.
2. **Barra de generación (barra PV):** Es una barra en la cual la magnitud del voltaje es definida y se mantiene constante, por otra parte, el despacho económico proporciona el valor de potencia activa entregada por cada generador del sistema. En esta barra  $Q$  y  $\delta$  son desconocidos.

3. **Barra Oscilante (barra  $V\delta$ ):** Es una barra especial de generación conocida como barra de referencia. En esta barra se asume la magnitud y el ángulo del voltaje por ejemplo  $1\angle 0^\circ$  pu, en tanto que P y Q son variables desconocidas.

La Tabla 1.1. presenta las variables calculadas y especificadas de cada tipo de barra.

**Tabla 1.1.** Variables especificadas (Esp) y calculadas (Cal) [6].

Tipo de Barra	$V$	$\theta$	$P_G$	$Q_G$	$P_C$	$Q_C$	# ecuaciones
$PQ$	Cal	Cal	Esp	Esp	Esp	Esp	$2(N - N_g - 1)$
$PV$	Esp	Cal	Esp	Cal	Esp	Esp	$N_g$
$V\delta$	Esp	Esp	Cal	Cal	Esp	Esp	0
							$2N - N_g - 2$

Las ecuaciones para el análisis de flujo de potencia son ecuaciones no lineales por lo tanto no se pueden resolver analíticamente, requiriéndose de algoritmos iterativos para su resolución.

El procedimiento para resolver un flujo de potencia consiste en una serie de pasos estándares que se resumen a continuación:

1. Estructurar la matriz de admitancias de barra  $Y_{BARRA}$ .
2. Hacer una estimación inicial de los voltajes de cada barra del sistema, en módulo y ángulo.
3. Determinar las desviaciones de potencia desde las ecuaciones de balance.
4. Actualizar los voltajes estimados en función de los algoritmos numéricos conocidos, como Newton-Raphson, Gauss-Seidel, entre otros.
5. Repetir el paso 4 hasta que las desviaciones de la solución sean mínimas.

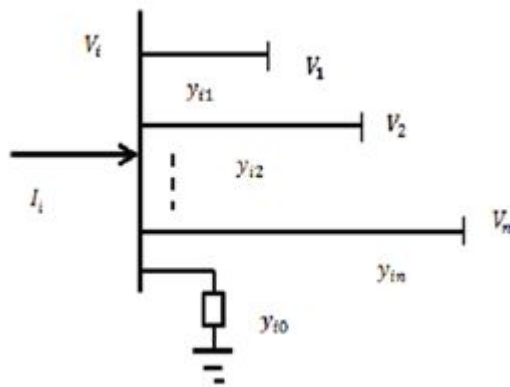
#### 1.4.3.2. Método de Gauss-Seidel

Es una técnica numérica iterativa para resolver ecuaciones no lineales de flujo de potencia en coordenadas rectangulares. En este método, las ecuaciones no lineales del flujo de potencia se pueden resolver iterativamente evaluando la solución inicial. Luego, esta solución estimada se usa junto con la ecuación originales para calcular una estimación

nueva y más precisa. El proceso continúa hasta que la diferencia entre estimaciones sucesivas es extremadamente insignificante.

Las ecuaciones son resueltas utilizando una combinación de cálculos iterativos y la matriz de admitancia, como se muestra a continuación:

Considerando una barra del sistema, como la mostrada en la Figura 1.24, las líneas de transmisión son representadas por el modelo equivalente  $\pi$ ; y las admitancias de cada línea se presentan con valores en por unidad sobre una base de potencia en común en MVA.



**Figura 1.24.** Diagrama de admitancias del circuito de referencia para obtención de las ecuaciones de nodo [9].

Aplicando la Ley de Corrientes de Kirchhoff a la barra  $i$  se tiene:

$$I_i = y_{i0}V_i + y_{i1}(V_i - V_1) + y_{i2}(V_i - V_2) + \dots + y_{in}(V_i - V_n) \quad (1.46)$$

$$I_i = (y_{i0} + y_{i1} + y_{i2} + \dots + y_{in})V_i - y_{i1}V_1 - y_{i2}V_2 - \dots - y_{in}V_n \quad (1.47)$$

De manera general:

$$I_i = V_i \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij}V_j \quad j \neq i \quad (1.48)$$

Por otro lado, la corriente en la barra  $i$  puede ser hallada mediante su respectiva potencia real y reactiva de la siguiente manera:

$$P_i + jQ_i = V_i I_i^* \quad (1.49)$$

$$I_i = \frac{P_i - jQ_i}{V_i^*} \quad (1.50)$$

Igualando las ecuaciones de corriente de la barra  $i$ , se obtiene:

$$\frac{P_i - jQ_i}{V_i^*} = V_i \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij} V_j \quad j \neq i \quad (1.51)$$

A partir de esta relación, se puede obtener la siguiente formulación matemática para la resolución por técnicas iterativas de Gauss-Seidel.

$$V_i^{[k]} = \frac{1}{\sum_{j=0}^n y_{ij}} \left[ \frac{P_i^{esp} - jQ_i^{esp}}{V_i^{[k-1]^*}} + \sum_{j=1}^n y_{ij} V_j \right] \quad j \neq i \quad (1.52)$$

Donde,  $k$  es el número de iteraciones del voltaje que se está calculando y  $k - 1$  es la iteración anterior a  $k$ .

Reescribiendo la ecuación en términos de la matriz de admitancias  $Y_{barras}$  en la que los elementos fuera de la diagonal son  $Y_{ij} = -y_{ij}$  y los elementos en la diagonal son  $Y_{ii} = \sum y_{ij}$  la ecuación pasa a ser:

$$V_i^{[k]} = \frac{1}{Y_{ii}} \left[ \frac{P_i^{esp} - jQ_i^{esp}}{V_i^{[k-1]^*}} - \sum_{j=1, j \neq i}^n Y_{ij} V_j \right] \quad (1.53)$$

Y las potencias activas y reactiva son:

$$P_i^{[k]} = \Re \left\{ V_i^{[k-1]^*} \left[ \sum_{j=1, j \neq i}^n Y_{ij} V_j^{[k]} + V_i^{[k-1]} Y_{ii} \right] \right\} \quad (1.54)$$

$$Q_i^{[k]} = -\Im \left\{ V_i^{[k-1]^*} \left[ \sum_{j=1, j \neq i}^n Y_{ij} V_j^{[k]} + V_i^{[k-1]} Y_{ii} \right] \right\} \quad (1.55)$$

$Y_{ii}$  incluye la admitancia a tierra de la susceptancia de carga y cualquier otra admitancia fija a tierra.

Debido a que en la barra oscilante se especifica los valores de voltaje  $V$  y  $\delta$ , esta no es considerada en el proceso iterativo de cálculo. Para las barras P-Q (barras de carga), donde  $P_i^{esp}$  y  $Q_i^{esp}$  son conocidas, se comienza con una estimación inicial y  $V_i^{[k]}$  es obtenido en cada iteración.

Finalmente, para las barras PV (barras de generación), donde  $P_i^{esp}$  y  $|V_i|$  son especificados,  $Q_i^{[k]}$  se obtiene para luego utilizarla en el cálculo de  $V_i^{[k]}$ , el voltaje en magnitud es una cantidad especificada y debe corregirse de la siguiente manera:



$$V_{i,corr}^k = |V_i| \frac{V_i^{[k]}}{|V_i^{[k]}|} \quad (1.56)$$

La velocidad de convergencia se puede incrementar aplicando el factor de aceleración  $\alpha$  a la solución aproximada obtenida de cada iteración como se muestra en la siguiente ecuación, para las barras P-Q:

$$V_{iac}^{[k]} = V_i^{[k-1]} + \alpha [V_i^{[k]} - V_i^{[k-1]}] \quad (1.56)$$

### 1.4.3.3. Método de Newton Raphson Desacoplado [5]

El método de Newton Raphson es una técnica iterativa que permite resolver sistemas de ecuaciones no lineales mediante la expansión de las funciones (ecuaciones) en series de Taylor. Para su solución, de manera general se procede de la siguiente manera:

Sea la función  $f(x)$ , una función no lineal y diferenciable en un intervalo, la expansión en series de Taylor será:

$$f(x) = f(x_0) + \frac{df(x_0)}{dx} \Delta x + \frac{d^2f(x_0)}{dx^2} \Delta x^2 + \dots + \frac{d^n f(x_0)}{dx^n} \Delta x^n \quad (1.57)$$

Donde,  $\Delta x = x - x_0$  y  $x_0$  es una aproximación a la raíz de la función  $f(x)$ . Dicha aproximación permite asumir que  $\Delta x$  es casi despreciable, por lo tanto, las derivadas de orden superior tenderán a cero, quedando la ecuación (1.57) de la siguiente manera:

$$f(x) \cong f(x_0) + \frac{df(x_0)}{dx} (x - x_0) = 0 \quad (1.58)$$

Despejando  $x$  de la ecuación (1.58) se tiene que:

$$x = x_0 + \frac{f(x_0)}{\frac{df(x_0)}{dx}} \quad (1.59)$$

$$x = x_0 + \frac{f(x_0)}{\left. \frac{df(x)}{dx} \right|_{x=x_0}} \quad (1.60)$$

$$x = x_0 + \frac{f(x_0)}{f'(x_0)} \quad (1.61)$$

El valor obtenido de la ecuación (1.61) es nuevamente utilizado para el cálculo del nuevo valor de  $x$ , pudiéndose de forma general reescribir:

$$x^{[k+1]} = x^{[k]} + \frac{f(x^{[k]})}{f'(x^{[k]})} \quad (1.62)$$

Donde,  $k$  es el número de iteración.

Las iteraciones terminan cuando la diferencia entre la actual y anterior iteración es aproximadamente cero o inferior a una tolerancia especificada  $\varepsilon$ .

$$\Delta x^{[k]} = x^{[k]} - x^{[k-1]} \approx 0$$

El proceso se puede aplicar para un sistema de ecuaciones no lineales de  $n$  incógnitas como es el caso de las ecuaciones de flujos de potencia, donde se tiene dos incógnitas en cada ecuación no lineal. Por tal motivo se puede utilizar el método de Newton Raphson para la solución de flujos de potencia de la siguiente manera:

Las ecuaciones (1.42) y (1.43) se reescriben como:

$$P_i = V_i \sum_{\substack{j=1 \\ j \neq i}}^n (G_{ij} \cos \delta_{ij} + B_{ij} \operatorname{sen} \delta_{ij}) V_j + V_i^2 G_{ij} \quad (1.62)$$

$$Q_i = V_i \sum_{\substack{j=1 \\ j \neq i}}^n (G_{ij} \operatorname{sen} \delta_{ij} + B_{ij} \cos \delta_{ij}) V_j - V_i^2 B_{ij} \quad (1.63)$$

Los errores de potencia, ecuaciones (1.44) y (1.45), se pueden expandir en series de Taylor como:

$$\Delta P_i = \sum_{j=1}^n \frac{dP_i}{d\delta_j} \Delta \delta_j + \sum_{j=1}^n \frac{dP_i}{dV_j} \Delta V_j \quad (1.64)$$

$$\Delta Q_i = \sum_{j=1}^n \frac{dQ_i}{d\delta_j} \Delta \delta_j + \sum_{j=1}^n \frac{dQ_i}{dV_j} \Delta V_j \quad (1.65)$$

Donde,

$$j = 1, \dots, n \\ j \neq \text{barra oscilante}$$

Expresando las ecuaciones (1.57) y (1.58) de forma matricial para  $n$  barras, en donde el segundo término es multiplicado y dividido por el módulo del voltaje  $V_j$ , se obtiene:

$$\begin{bmatrix} \Delta P_i \\ \vdots \\ \Delta P_n \\ \Delta Q_i \\ \vdots \\ \Delta Q_n \end{bmatrix} = \begin{bmatrix} \frac{dP_i}{d\delta_i} & \dots & \frac{dP_i}{d\delta_n} \\ \vdots & J_{11} & \vdots \\ \frac{dP_n}{d\delta_i} & \dots & \frac{dP_n}{d\delta_n} \\ \hline \frac{dQ_i}{d\delta_i} & \dots & \frac{dQ_i}{d\delta_n} \\ \vdots & J_{21} & \vdots \\ \frac{dQ_n}{d\delta_i} & \dots & \frac{dQ_n}{d\delta_n} \end{bmatrix} \begin{bmatrix} \Delta\delta_i \\ \vdots \\ \Delta\delta_n \\ \frac{\Delta V_i}{V_i} \\ \vdots \\ \frac{\Delta V_n}{V_n} \end{bmatrix} \quad \begin{matrix} i = 1, \dots, n \\ i \neq \text{barra osc.} \end{matrix} \quad (1.66)$$

La matriz formada por  $J_{11}$ ,  $J_{12}$ ,  $J_{21}$  y  $J_{22}$ , de la ecuación (1.66), es conocida como Jacobiano y se representa por la letra  $J$ . Esta matriz puede simplificarse mediante el principio de desacoplamiento de un sistema eléctrico de potencia, el cual menciona que [10]:

“Cuando un sistema eléctrico de potencia de generación-transmisión opera en estado estable existe una débil dependencia entre la potencia activa y la magnitud del voltaje, y entre la potencia reactiva y el ángulo del voltaje”.

Es decir que las derivadas parciales de  $J_{12}$  y  $J_{21}$  pueden ser despreciables con lo cual, la ecuación (1.66) se puede reducir, de manera general a:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_{11} & (J_{12} \approx 0) \\ (J_{21} \approx 0) & J_{22} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta V/V \end{bmatrix} \quad (1.67)$$

$$\Delta P = J_{11} \cdot \Delta\delta \quad (1.68)$$

$$\Delta Q = J_{22} \cdot \frac{\Delta V}{V} \quad (1.69)$$

De esta manera las ecuaciones a resolverse son más sencillas para su resolución.

Los vectores las variables,  $\Delta\delta$  y  $\Delta V/V$  deben ser actualizados en cada iteración ya que los valores iniciales que se les da en un principio no son exactos, el proceso iterativo será:

$$\delta_i^{[k]} = \delta_i^{[k-1]} + \Delta\delta_i^{[k]}$$

$$V_i^{[k]} = V_i^{[k-1]} + \Delta V_i^{[k]}$$

Para,

$$\begin{matrix} i = 1, \dots, n \\ i \neq \text{barra osc.} \end{matrix}$$

Los voltajes y ángulos hallados en cada iteración se utilizarán para hallar el jacobiano  $J_{11}$  y  $J_{22}$ .

Para los elementos fuera de la diagonal se tiene ( $i \neq j$ ):

$$\frac{dP_i}{d\delta_j} = V_i V_j (G_{ij} \cos \delta_{ij} - B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.70)$$

$$V_j \frac{dP_i}{dV_j} = V_i V_j (G_{ij} \cos \delta_{ij} + B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.71)$$

$$\frac{dQ_i}{d\delta_j} = -V_j \frac{dP_i}{d\delta_j} = -V_i V_j (G_{ij} \cos \delta_{ij} + B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.72)$$

$$V_j \frac{dQ_i}{dV_j} = \frac{dP_i}{dV_j} = V_i V_j (G_{ij} \cos \delta_{ij} - B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.73)$$

Y para los elementos dentro de la diagonal ( $i = j$ ):

$$\frac{dP_i}{d\delta_i} = -Q_i - V_i^2 B_{ii} \quad (1.74)$$

$$V_j \frac{dP_i}{dV_i} = P_i + V_i^2 G_{ii} \quad (1.75)$$

$$\frac{dQ_i}{d\delta_i} = P_i - V_i^2 G_{ii} \quad (1.76)$$

$$V_j \frac{dQ_i}{dV_j} = Q_i - V_i^2 B_{ii} \quad (1.77)$$

El proceso iterativo se termina cuando la diferencia entre la iteración  $k$  actual y anterior es aproximadamente cero o inferior a una tolerancia especificada  $\varepsilon$ .

Al igual que en el método de Gauss Seidel, la barra oscilante no se considera ya que el módulo y el ángulo de su voltaje son valores especificados.

De manera general una vez hallados los valores de voltaje y ángulos de cada barra, independiente del método utilizado, se puede hallar los flujos de potencial de las ramas conectadas entre las barras  $i$  y  $j$ , de la siguiente manera:

$$P_{ij} = V_i \left( G_{ij} (V_i - V_j \cos(\theta_{ij})) + V_j B_{ij} \operatorname{sen}(\theta_{ij}) \right) \quad (1.78)$$

$$Q_{ij} = V_i \left( B_{ij} (V_i - V_j \cos(\theta_{ij})) - V_j G_{ij} \operatorname{sen}(\theta_{ij}) - V_i \frac{y}{2} \right) \quad (1.79)$$

Las pérdidas en las ramas se determinan como:

$$Pl_{ij} = P_{ij} + P_{ji} \quad (1.80)$$

$$Ql_{ij} = Q_{ij} + Q_{ji} \quad (1.81)$$

Y las pérdidas totales como:

$$Pl_{totales} = \sum Pl_{ij} \quad (1.82)$$

$$Ql_{totales} = \sum Ql_{ij} \quad (1.83)$$

#### 1.4.3.4. Flujo de Potencia Linealizado o DC [10]

Un flujo de potencia DC es un método no iterativo, usado en la planificación de expansión del sistema de transmisión y análisis rápidos de transferencia de potencia [10]. Este método permite convertir las ecuaciones no lineales del flujo de potencia, ecuaciones (1.42) y (1.43), en ecuaciones lineales, mediante tres suposiciones que se detallan a continuación:

1. Se asume que los voltajes de barra  $V_i$  y  $V_j$ , en el sistema son iguales a 1pu, este supuesto se realiza debido a que en este método no interesa conocer con exactitud el voltaje, por tanto, tampoco la potencia reactiva, es decir:

$$P_i = \sum_{j=1}^n (G_{ij} \cos \delta_{ij} + B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.84)$$

$$Q_i = \sum_{j=1}^n (G_{ij} \operatorname{sen} \delta_{ij} + B_{ij} \cos \delta_{ij}) \approx 0 \quad (1.85)$$

2. Las ramas se consideran sin pérdidas, es decir,  $R_{ij} \ll X_{ij}$ . Siempre y cuando se trate de un sistema de transmisión, este supuesto no se puede considerar en el sistema de distribución ya que es difícil garantizar que la resistencia sea despreciable en niveles de medio y bajo voltaje.

En términos de admitancias,  $G_{ij} \ll B_{ij}$ . Es decir,  $G_{ij} \approx 0$  y por lo tanto la ecuación (1.84) será:

$$P_i = \sum_{j=1}^n (B_{ij} \operatorname{sen} \delta_{ij}) \quad (1.86)$$

3. Las diferencias angulares entre las barras  $i$  y  $j$ , son muy pequeñas, por tanto, la función trigonometría  $\operatorname{sen} \delta_{ij}$  es igual a  $(\delta_i - \delta_j)$  en radianes. En consecuencia:

$$P_i = \sum_{j=1}^n B_{ij} (\delta_i - \delta_j) \quad (1.87)$$

La ecuación (1.62) puede expresarse de forma matricial para  $n$  barras como:

$$\begin{bmatrix} P_i \\ \vdots \\ P_n \end{bmatrix} = [B] \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix}, \quad \begin{array}{l} i = 1, \dots, n \\ i \neq \text{barra osc.} \end{array}$$

Donde  $B$  es la parte imaginaria de la matriz de admitancias del sistema completo  $Y_{bus}$  y  $P_i$  es la potencia específica de cada barra. De la ecuación las únicas incógnitas son los ángulos los cuales se calcularán realizando su respectivo despeje.

Los flujos por las líneas se podrán calcular mediante la siguiente fórmula [11]:

$$P_{ij} = \frac{\delta_i - \delta_j}{x_{ij}}$$

Cabe mencionar que estas suposiciones se pueden aplicar a un sistema de transmisión siempre y cuando este no este sobrecargado, ya que, en esta última condición, los voltajes disminuyen como efecto las resistencias de las ramas.

## 2. METODOLOGÍA

Un enfoque metodológico para el desarrollo de software es esencial para abordar de manera efectiva un problema dado, ya que seguir una metodología permite alcanzar los objetivos específicos de forma exitosa siguiendo una serie de pasos establecidos, como se describirán a continuación.

Para desarrollar un software computacional que resuelva flujos de potencia en sistemas de transmisión, se utiliza metodologías ágiles, asegurándose que la herramienta cumpla con los objetivos especificados en el presente Trabajo de Integración Curricular.

Los softwares comerciales, como PowerFactory de DlgSILENT, no ofrecen la posibilidad de observar el proceso que se sigue para llegar a la solución de flujos de potencia, en este sentido se establece una desventaja en el ámbito del aprendizaje, ya que los resultados intermedios de cada iteración no pueden ser observados y analizados. En este contexto no es posible comparar la convergencia de la solución de los diferentes métodos y los estudiantes no pueden analizar sus ventajas y desventajas en los diferentes casos de estudio.

Desarrollar herramientas computacionales específicas que permitan mejorar el aprendizaje de los métodos de solución de flujos de potencia, de esta forma se aporta al desarrollo de las habilidades técnicas de los estudiantes de la Carrera de Electricidad. La disponibilidad de este tipo de herramientas permite a los docentes enriquecer sus estrategias de enseñanza y crear un mejor entorno de aprendizaje para sus estudiantes.

La creación de herramientas en lenguaje de programación Python ofrece una gran variedad de beneficios para los usuarios, permitiéndoles aprovechar las estructuras de datos incorporadas, su simplicidad y versatilidad para crear aplicaciones eficientes. Los usuarios pueden utilizar bibliotecas ya desarrolladas por otros usuarios y así ahorrar tiempo al evitar escribir códigos desde cero. Python al ser de código abierto, es gratuito y puede ser utilizado y distribuido libremente, lo que lo hace accesible a una amplia comunidad de usuarios. Al utilizar un lenguaje de código abierto los desarrolladores pueden compartir, reutilizar o expandir código de proyectos, simplificando el desarrollo de aplicaciones reduciendo tiempo y costos.

En el presente Trabajo de Integración Curricular (TIC) se aprovecha los beneficios que ofrece el lenguaje de programación Python para la creación de una aplicación que permita ingresar los modelos matemáticos de transformadores, generadores y líneas de transmisión, para la estructuración de modelos de red de transmisión representada por su

matriz de admitancias para cada caso de estudio propuesto. El modelo de red permite la obtención de soluciones de flujo de potencia mediante la aplicación de los métodos de Gauss Seidel, Newton Raphson desacoplado y flujo DC, a través de sus respectivos algoritmos implementados en lenguaje de programación Python.

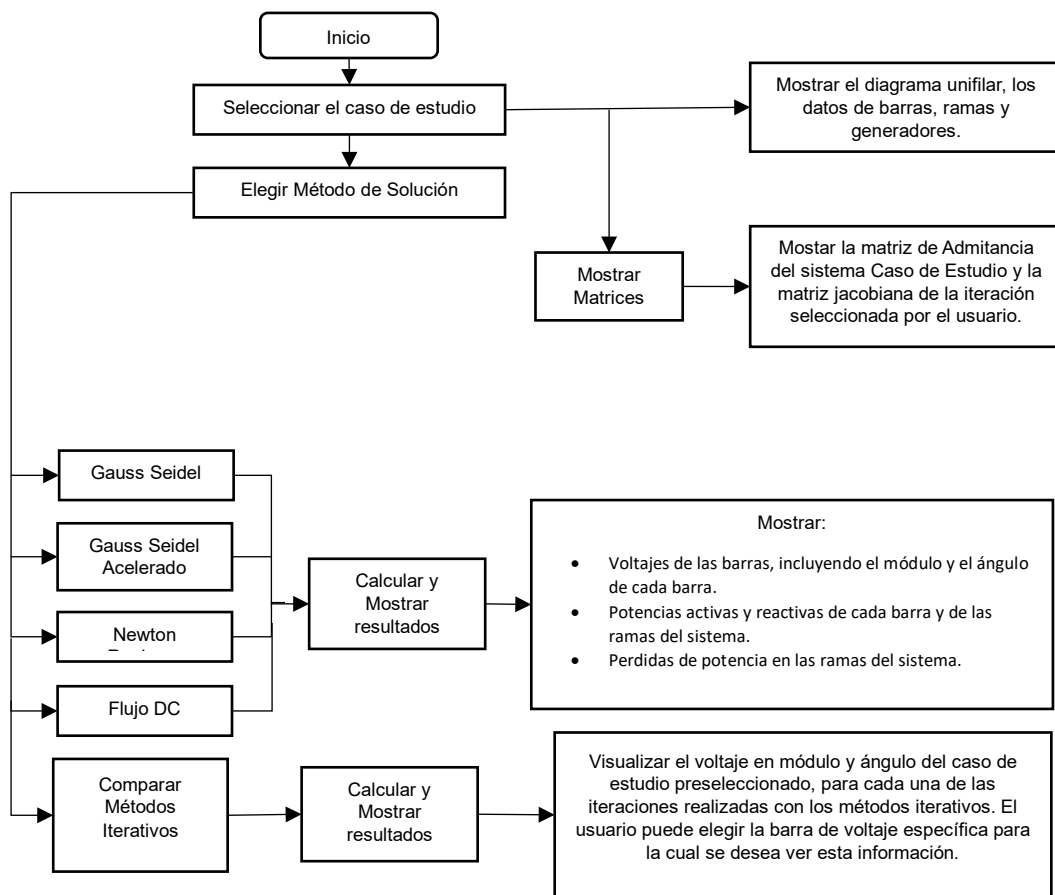
La aplicación desarrollada tiene la capacidad de mostrar resultados intermedios durante el proceso de solución, presentar gráficos de evolución de la convergencia para cada método y brinda acceso a los datos obtenidos tras la solución de flujos de potencia, esto se logra mediante una interfaz interactiva, intuitiva y fácil de navegar con el fin de que el usuario fortalezca sus habilidades en relación a la solución de flujos de potencia de sistemas eléctricos de transmisión.

## **2.1. Estructura de la Herramienta de Software**

Una vez identificado el problema y su solución, se inicia el diseño de una interfaz amigable y fácil de usar que brinde al usuario una visión detallada y completa del estado del sistema eléctrico. De esta manera, facilitar al usuario el análisis y evaluación más precisa y eficiente del caso de estudio.

En la Figura 2.1 se presenta el diagrama de flujo que será utilizado para la elaboración del código de la aplicación. Este diagrama proporciona una representación gráfica de los pasos y procesos involucrados en la solución del problema y permite una comprensión clara y concisa del funcionamiento de la aplicación.



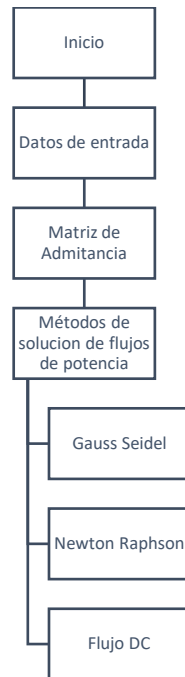


**Figura 2.1.** Diagrama flujos de la aplicación.

El programa comienza con la presentación de la ventana principal, donde el usuario puede elegir el caso de estudio que desea analizar. Una vez seleccionado el sistema, se puede abrir una ventana secundaria mediante un botón. En esta ventana secundaria se muestra la matriz de admitancias y una lista desplegable con las iteraciones realizadas en el método Newton Raphson. El usuario puede elegir la iteración que desea visualizar y, en consecuencia, obtendrá la matriz jacobiana correspondiente de dicha iteración. De esta manera, se puede profundizar en el análisis de cada iteración.

Regresando a la pantalla principal, además de elegir el caso de estudio, se puede elegir uno de los cuatro métodos de solución disponibles. También, al igual que antes, se puede acceder a una segunda pantalla que muestra los siguientes valores: voltajes, en modulo y ángulo, de cada barra; potencia activa y reactiva de cada barra y de las ramas del sistema; y perdidas de potencias en las ramas. Si se ha seleccionado un método iterativo se visualizará un gráfico que muestra la convergencia hacia la solución. Esta ventana secundaria tendrá el objetivo de brindar una visión detallada y completa del sistema eléctrico seleccionado previamente para su análisis y comprensión.

En general, la aplicación constara de los siguientes elementos, los cuales se muestran en la Figura 2.2 y serán detallados a continuación.



**Figura 2.2.**Diagrama de la estructura general de la aplicación

### 2.1.1. Datos de entrada

Independiente de caso de estudio, los datos de entrada para la aplicación están organizadas en tres matrices: una con los datos de las barras, otra con los datos de las ramificaciones y la tercera matriz con los datos de los generadores del caso de estudio seleccionado.

Cada columna de la matriz corresponderá a un dato en específico según los tres tipos de matriz, como se mostrará a continuación.

#### 2.1.1.1. Datos de barras

La estructura de los datos de barras del sistema se presenta en la Tabla 2.1, y el índice de cada columna corresponde con un término específico que se detalla en la Tabla 2.2

**Tabla 2.1.** Matriz Datos de Barras

Barra_i	Tipo	Pd	Qd	Gs	Bs	Vm	Va
0	3	50	30.99	0	0	1	0
1	1	170	105.35	0	0	1	0

**Tabla 2.2.** Índices de las columnas de los datos de Barras

Barra_i	Numeración de la barra
Tipo	Tipo de barras
Pd	Potencia Real Demandada (MW)
Qd	Potencia Reactiva Demandada (MVar)
Gs	Conductancia shunt (MW a Voltaje 1 pu)
Bs	Susceptancia shunt (MW a Voltaje 1 pu)
Vm	Magnitud del voltaje en pu
Va	Ángulo del voltaje en grados

En el capítulo 1 se describen tres tipos de barras, que se identificaran en la aplicación desarrollada con los números 1,2 y 3 respectivamente: barra  $PQ$ , barra  $PV$  y barra  $V\delta$ . Adicionalmente, la potencia real y reactiva demandada se refiere a la potencia consumida por la carga conectada en la barra correspondiente.

### 2.1.1.2. Datos de Ramificaciones

Para los datos de ramificaciones del sistema se considera las barras a las que se encuentran conectadas y demás datos como se ve en la Tabla 2.3, con el detalle mostrado en la Tabla 2.4.

**Tabla 2.3.** Matriz Datos de ramificaciones.

fbus	tbus	r	x	B	Ratio	Estado
0	1	0.01008	0.0504	0.1025	0	1

**Tabla 2.4.** Índices de las columnas de los datos de ramificaciones

fbus	Barra desde
tbus	Barra hasta
r	Resistencia de la rama (pu)
x	Reactancia de la rama (pu)
b	Susceptancia de la rama (pu)
ratio	Relación de vueltas del transformador (pu)
Estados	Estado inicial de la rama, 1 en servicio, 0 fuera de servicio

### 2.1.1.3. Datos de Generadores

En el caso de los generadores la estructura de datos se presenta en la Tabla 2.5 y el detalle en la Tabla 2.6.

**Tabla 2.5.** Matriz Datos de generadores.

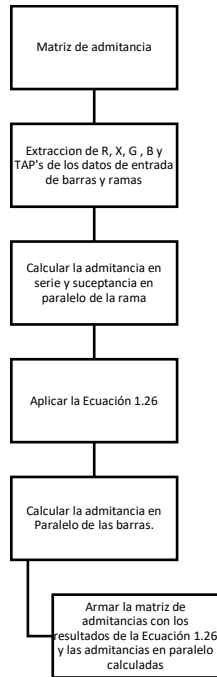
barra	Pg	Qg	Vg	Estado
0	0	0	1	1

**Tabla 2.6.** Índices de las columnas de los datos de ramificaciones.

Barra	Barra donde se encuentra el generador
Pg	Potencia activa de salida del generador (MW)
Qg	Potencia reactiva de salida del generador (MW)
Vg	Magnitud del voltaje, setpoint (p.u.)
Estado	Estado inicial del generador, 1 en servicio, 0 fuera de servicio

### 2.1.2. Matriz de admitancia

En la sección 1.4.2.5 y 1.4.2.7 se explica la modelación utilizada en el planteamiento del problema de flujos de potencia de un sistema eléctrico, sobre dicha base es posible estructurar un algoritmo que permite obtener la matriz de admitancia de barras de cualquier caso de estudio. En la siguiente figura se resume en forma de diagrama de bloques el procedimiento seguido para la obtención de mencionada matriz.



**Figura 2.3.** Algoritmo para armar la matriz de admitancia

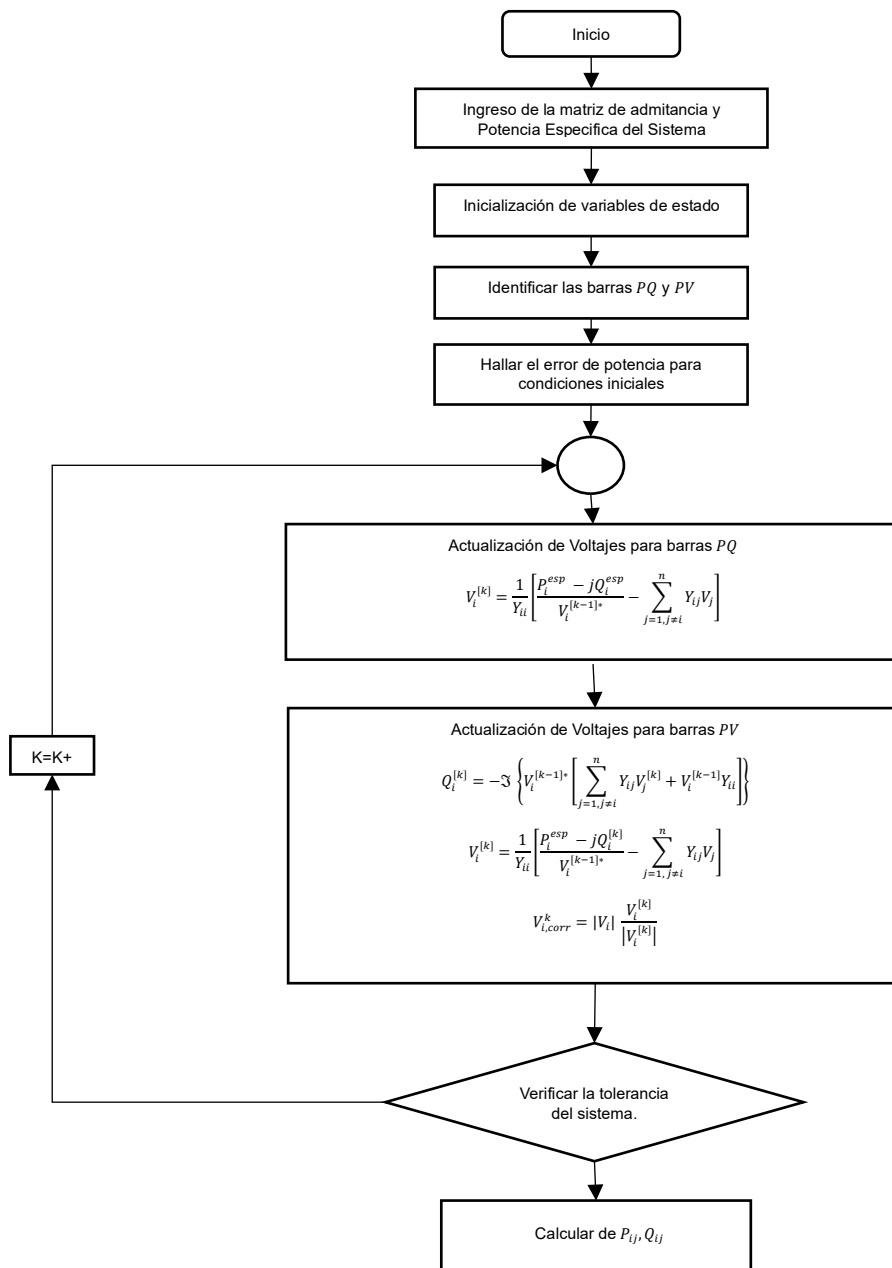
### 2.1.3. Programación de los algoritmos de solución de flujos de potencia

La programación de los métodos de solución de flujos de potencia se guía en diagramas de flujo que permiten estructurar los algoritmos de solución de flujos de potencia según los métodos presentados en el Capítulo 1.

#### 2.1.3.1. Gauss Seidel

Una vez que se ha establecido la matriz de admitancia del sistema, es esencial calcular la potencia especificada en cada barra, que es la diferencia entre la potencia aparente generada ( $P_g + Q_g$ ) y la potencia aparente demandada ( $P_d + Q_d$ ).

La Figura 2.4 se presenta el diagrama de flujo del algoritmo del método de Gauss-Seidel para resolver un problema de flujo de potencia.



**Figura 2.4.** Diagrama de flujo del método de Gauss-Seidel

Dentro del algoritmo mostrado podemos agregar el factor de aceleración  $\alpha$  después de la actualización de voltajes para las barras PQ.

### 2.1.3.2. Newton Raphson Desacoplado

Antes de comenzar con las iteraciones del método de Newton Raphson Desacoplado, es necesario realizar el cálculo de la potencia especificada y la matriz de admitancia, al igual que en el método Gauss Seidel.

El algoritmo presentado en la Figura 2.5 [12], toma en cuenta las derivadas parciales de las matrices  $J_{11}$  y  $J_{22}$ , mientras que las derivadas parciales de las matrices  $J_{12}$  y  $J_{21}$  son despreciadas.

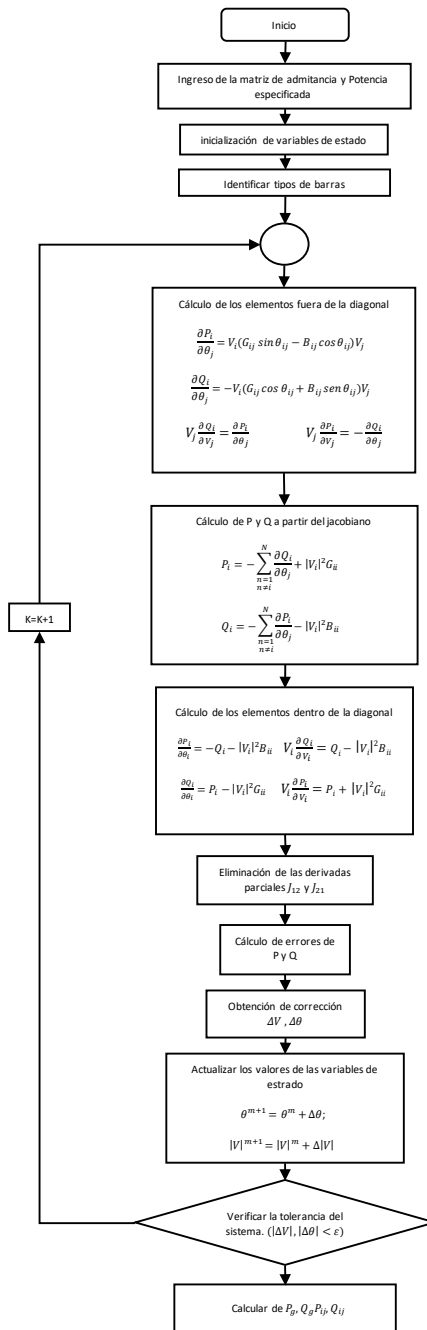
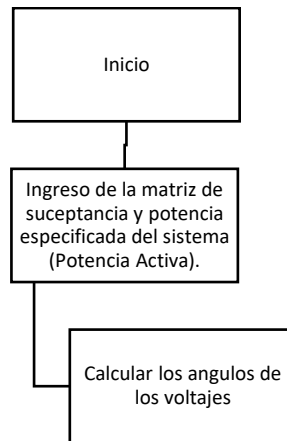


Figura 2.5. Diagrama de flujo del método de Newton Raphson Desacoplado

### 2.1.3.3. Flujo DC

Para resolver el flujo de potencia linealizado, es necesario considerar la parte imaginaria de la matriz de admitancia y los ángulos de los voltajes en condiciones iniciales. Esto se debe a los criterios asumidos por el método, que se explican en la sección 1.4.3.3.



**Figura 2.6.** Diagrama de flujo del método flujo de potencia linealizado

El flujo DC es una solución más sencilla que otros métodos debido a que no es iterativa.

## 2.2. Implementación en lenguaje de programación Python

Tkinter es un módulo de Python utilizado para la creación de interfaces gráficas, es compatible con una amplia variedad de sistemas operativos, lo que lo hace atractivo para aquellos que desarrollan aplicaciones destinadas a diferentes plataformas. La utilidad de Tkinter es amplia, desde la creación de aplicaciones simples hasta complejas con características avanzadas como gráficos y animaciones. Por esta razón, la interface de la herramienta de software desarrollada en el presente Trabajo de Integración Curricular es programada utilizando Tkinter.

La programación de la herramienta de software utiliza bibliotecas disponibles de Python, por lo tanto, para su uso es necesario que el usuario tenga instaladas las librerías que se mencionan a continuación:

### NumPy

Es una biblioteca de Python que se utiliza para trabajar con matrices de N dimensiones. Es ampliamente utilizada para el análisis de datos o la comunidad científicas.

### SciPy



SciPy es una biblioteca de Python para cálculo científico y análisis numérico que complementa a NumPy y proporciona funciones avanzadas para optimización. En este TIC, se utilizará el submódulo 'scipy.sparse' que ofrece herramientas para manipular eficientemente matrices dispersas. Este módulo incluye funciones para realizar cálculos complejos en matrices dispersas, como multiplicación de matrices, solución de sistemas de ecuaciones lineales, entre otros.

### Matplotlib

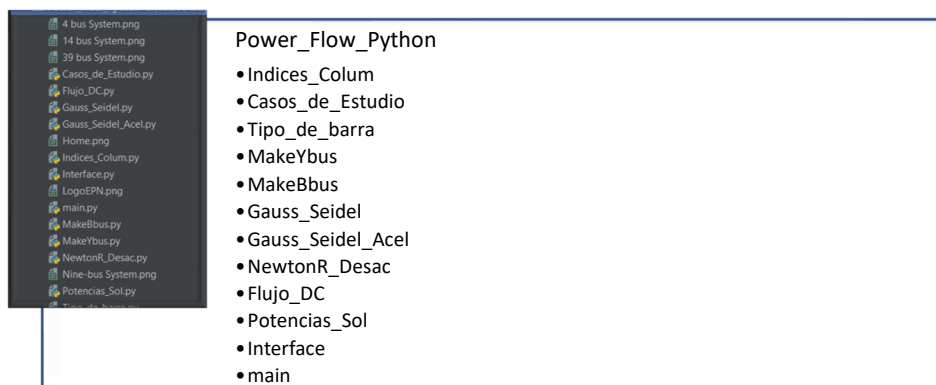
Matplotlib es una herramienta que permite crear visualizaciones de datos en Python de manera sencilla y flexible. Cuenta con una gran variedad de funciones y gráficos como: los gráficos de barras, gráficos 3D y mapas. Además, Matplotlib permite personalizar y controlar cada aspecto de la visualización, como el estilo, las etiquetas y leyendas.

### PyPower

PyPower es un paquete de software que se caracteriza por ser muy accesible y ofrece módulos que pueden ser modificados de acuerdo con la utilidad que se desee en la solución de flujos de potencia. Este paquete cuenta con una amplia gama de funciones y algoritmos que se adaptan a las necesidades de los usuarios, y su código puede ser personalizado y modificado para mejorar su rendimiento o para ajustarse a requisitos específicos de solución de problemas en sistemas eléctricos. Para este TIC, no es necesario su instalación.

## 2.2.1. Módulos del Software Desarrollado en Python

La aplicación consta de los módulos presentados en la Figura 2.7, los cuales se detallan a continuación.



**Figura 2.7.** Módulos de la aplicación.

### 2.2.1.1. Indices\_Colum

El módulo “Indices\_Colum” se ha definido para establecer constantes que indican los índices de las columnas en las matrices de los datos de entrada. De esta manera, se facilita la comprensión y accesibilidad de todo el código del programa, ya que estas constantes se encuentran en todos los módulos.

### 2.2.1.2. Casos\_de\_Estudio

El módulo “Casos\_de\_Estudio” es una sección de la aplicación donde se almacenan los datos de los diferentes sistemas de potencia IEEE de 9, 14 y 39 barras, así como el sistema de 4 barras referenciado en [5].

Los datos de los sistemas están almacenados en funciones que devuelven 3 matrices: *bus*, *branch* y *gen*. La matriz *bus* contiene los datos de las barras; la matriz *branch* los datos de las ramas; y, la matriz *gen* los datos de los generadores.

```
def Caso4b():
    #-----Datos de las barras-----
    # barra_i, tipo, Pd, Qd, Gs, Bs, Vm, Va
    bus=array([
    [0, 3, 50, 30, 99, 0, 0, 1, 0],
    [1, 1, 170, 105, 35, 0, 0, 1, 0],
    [2, 1, 200, 123, 94, 0, 0, 1, 0],
    [3, 2, 80, 49, 58, 0, 0, 1, 0.02, 0]
    ])

    # fbus, tbus, r, x, b, ratio, angle, status
    branch = array([
    [0, 1, 0.01008, 0.0504, 0.1025, 0, 0, 1],
    [0, 2, 0.00744, 0.0372, 0.0775, 0, 0, 1],
    [1, 3, 0.00744, 0.0372, 0.0775, 0, 0, 1],
    [2, 3, 0.01272, 0.0636, 0.1275, 0, 0, 1]
    ])

    ## Datos de los generadores
    # barra, Pg, Qg, Vg, mBase, status
    gen = array([
    [0, 0, 0, 1, 0, 1],
    [3, 318, 0, 1, 0.02, 0, 1]
    ])
    return bus, branch, gen
```

Figura 2.8. Módulo “Caso\_de\_Estudio”.

### 2.2.1.3. Tipo\_de\_barras

El módulo almacena una función que toma como entrada las matrices *bus* y *gen*, definidas en el módulo “Caso\_de\_Estudio”, y forma una lista de índices para las barras  $V\delta$ ,  $PV$ ,  $PQ$ . Si por algún motivo no fue definida la barra  $V\delta$ , la función elige la primera barra  $PV$  como  $V\delta$  y se quita de la lista de barras  $PV$ .

### 2.2.1.4. MakeYbus

La función construye la matriz de admitancia a partir de las matrices bus y branch. Primero, calcula el número de barras y líneas en el sistema, después calcula los elementos de la matriz de admitancia de cada rama utilizando la resistencia y reactancia de la línea.

Además, se calcula la susceptancia conectada en paralelo a la barra y se asignan valores de relación de transformación a las ramas (1 en caso de líneas).

A continuación, se calculan las admitancias en paralelo  $Y_{ff}$ ,  $Y_{ft}$ ,  $Y_{tf}$  y  $Y_{tt}$  y se calcula la admitancia en derivación. Se construyen las matrices de conexión  $C_f$  y  $C_t$  que se usan para conectar las ramas con las barras "desde" y "hasta", respectivamente.

#### **2.2.1.5. MakeBbus**

El módulo permite obtener la matriz de susceptancia del sistema a partir de las matrices "bus" y "branch". El módulo calcula la matriz  $B$  que representa la relación entre la potencia real inyectada en cada barra y los ángulos de voltaje de cada barra. Mediante este método se obtienen los valores necesarios para realizar un análisis de flujo de potencia en DC más preciso.

#### **2.2.1.6. Gauss\_Seidel**

El módulo "Gauss\_Seidel" implementa el algoritmo de Gauss-Seidel descrito en la Figura 2.4. La función toma como entrada la matriz de admitancia, la potencia especificada, el vector de los voltajes iniciales, los índices de cada tipo de barra ( $P\delta$ ,  $PV$  y  $PQ$ ) y una variable "Var". La variable "Var" almacena el valor ingresado por el usuario y se utiliza para guardar el valor de los voltajes en cada iteración de la barra "Var" hasta que el método termine las iteraciones.

El algoritmo de Gauss Seidel se realiza en un bucle que se ejecuta hasta que se alcance la tolerancia o se alcance el número máximo de iteraciones. En cada iteración, los voltajes en las barras  $PQ$  y  $PV$  se actualizan para después evaluar la función  $F$ , que representa el desequilibrio de la red, si el desequilibrio es menor que la tolerancia, el algoritmo convergió.

Finalmente, la función devuelve el vector voltaje de la última iteración, el número de iteraciones, el desequilibrio del sistema (errores de potencia) y los voltajes de barra correspondiente a la variable "Var" de cada iteración.

#### **2.2.1.7. Gauss\_Seidel\_Acel**

El módulo es similar al módulo "Gauss\_Seidel". La única diferencia es la inclusión del factor de aceleración en el algoritmo. Este módulo fue creado con el propósito de mostrar la diferencia que hace la aplicación del factor de aceleración en el algoritmo de Gauss-Seidel.

### 2.2.1.8. NewtonR\_Desac

La función implementa un método de Newton para resolver el flujo de potencia en un sistema eléctrico. La función toma como entrada la matriz de admitancia del sistema eléctrico, la potencia especificada, los voltajes iniciales, una lista de índices que indican el tipo de barra y dos variables llamadas "Var" e "it\_jac".

La variable "Var" desempeña un papel similar al mencionado en el método anterior, mientras que "it\_jac" almacena el valor especificado por el usuario y permite acceder a la matriz jacobiana en la iteración especificada por "it\_jac".

En cada iteración, se evalúa la matriz Jacobiana, que representa las derivadas parciales de los flujos de potencia en relación con los ángulos de voltaje y las magnitudes de voltaje. Luego, se calculan los errores en el ángulo y la magnitud de voltaje. A continuación, se actualizan los valores de voltaje y ángulo para calcular los errores en el flujo de potencia. Finalmente, se comparan estos errores con una tolerancia previamente establecida, y si se cumplen los criterios, el proceso termina.

### 2.2.1.9. Flujo\_DC

La función dentro del módulo implementa el método de solución del flujo de potencia DC y se utiliza para obtener los ángulos de voltaje de todas las barras del sistema con excepción de la barra de referencia. Para el efecto, la función toma como entrada la matriz B del sistema, la potencia activa especificada, los ángulos iniciales de los voltajes de las barras y la lista de los índices que indican el tipo de barra.

La función inicializa los ángulos de voltajes inicial "Va0" y luego actualiza los ángulos de las barras que no son de referencia utilizando una solución de sistemas lineales basada en la matriz de admitancia y las inyecciones de potencia. Finalmente, se devuelve el vector de ángulos de voltaje en radianes.

La línea de código:

$$Va[pvpq] = \text{spsolve}((B[pvpq.T, pvpq]), ((Pbus[pvpq]) - (B[pvpq.T, ref] * Va0[ref])))$$

Realiza una operación de solución de sistemas de ecuaciones lineales mediante la función *spsolve* de la librería *scipy.sparse.linalg* y el resultado de la operación se almacena en la submatriz *Va[pvpq]* que contiene los ángulos de voltaje de las barras que no son de referencia. En resumen, esta línea de código se utiliza para calcular los ángulos de voltaje de las barras en un sistema de flujo de potencia directo (DC) utilizando el método de solución de sistemas de ecuaciones lineales.

### **2.2.1.10. Potencias\_Sol**

El módulo *Potencias\_sol* almacena las funciones:

*flujos\_Ramifi* - permite calcular el flujo de potencia activa y reactiva a través de las ramas del sistema y las pérdidas correspondientes, utilizando los voltajes obtenidos en la solución de flujos de potencia.

*Potencia\_Barras* - permite calcular la potencia que se inyecta en cada barra del sistema, utilizando los resultados obtenidos en la solución del flujo de potencia.

*flujos\_RamifiDC* - utiliza los voltajes obtenidos en la solución de flujos para calcular la potencia activa que fluye a través de las ramas y las pérdidas asociadas.

*Potencia\_BarrasDC* - permite obtener la potencia activa inyectada en cada barra.

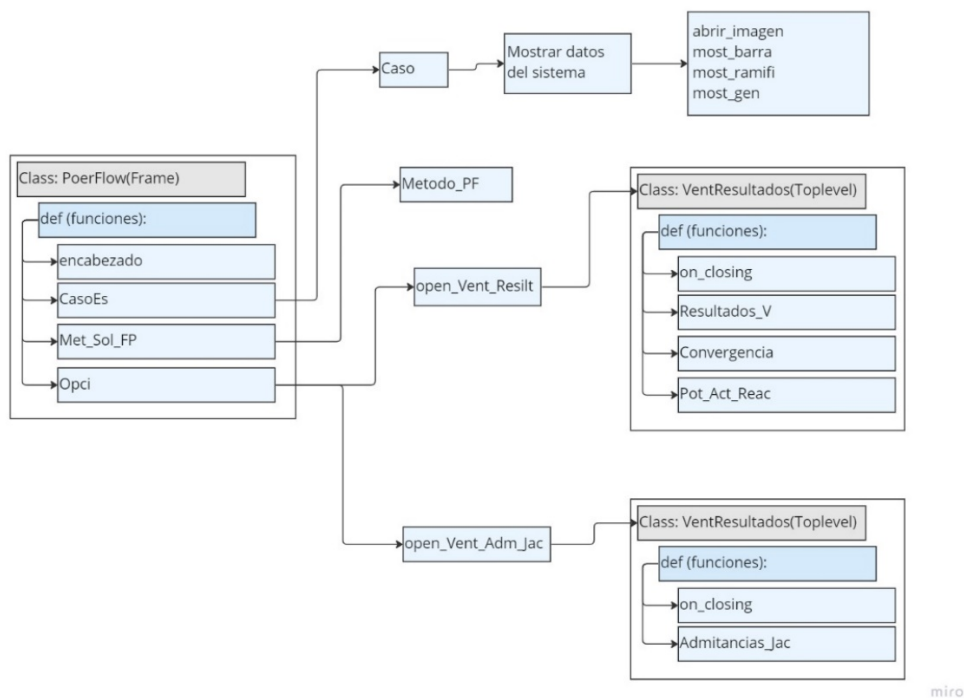
Las funciones en este módulo utilizan las ecuaciones del análisis de flujos de potencia explicados en el marco teórico de este documento.

### **2.2.1.11. Interface**

El módulo en cuestión ha sido desarrollado utilizando programación orientada a objetos, con el objetivo de ofrecer un código de fácil comprensión para los usuarios. Para lograrlo, se ha utilizado clases, funciones y funciones especiales, todas ellas para maximizar la legibilidad y claridad del código.

En la Figura 2.9 se puede observar la estructura del módulo, es decir, la organización de las clases y funciones que lo componen. Esta figura es de utilidad para aquellos usuarios que necesiten conocer el funcionamiento interno del módulo y que desean hacer cambios.

Cabe destacar que el uso de programación orientada a objetos ofrece numerosas ventajas, como la reutilización del código ya escrito y la posibilidad de crear abstracciones que simplifiquen la comprensión del problema.



**Figura 2.9.** Diagrama de clases y funciones del módulo "Interface"

En la Figura 2.9 se puede observar dos tipos de elementos: aquellos con fondo gris corresponden a las clases que se encargan de construir la interfaz gráfica de la ventana principal y las ventanas secundarias. Por otro lado, aquellos con fondo azul corresponden a funciones que permiten mostrar los resultados obtenidos después de la solución del flujo de potencia.

Varias funciones del módulo permiten la visualización clara de los resultados obtenidos después de la solución del flujo de potencia, entre ellas, las que permiten la presentación de los voltajes y potencia de cada barra y rama del sistema en forma de tablas. Estas funciones también permiten realizar la comparación de los voltajes de una barra seleccionada por el usuario para los diferentes métodos iterativos disponibles. Esta comparación es particularmente útil en la evaluación de la precisión y la eficiencia de los métodos iterativos de solución. En general, estas funciones contribuyen significativamente a la claridad y eficiencia de la presentación de los resultados obtenidos después de la solución del flujo de potencia, lo que a su vez facilita la comparación y la toma de decisiones.

En resumen, el módulo incluye clases y funciones que permiten la construcción de una interfaz gráfica de usuario, así como la presentación clara y eficiente de los resultados obtenidos después de la solución del flujo de potencia. El uso de una interfaz gráfica hace que el proceso de aprendizaje sea mucho más amigable e intuitivo para el usuario.

### 2.2.1.12. Main

La función `main()` es el punto de entrada de la aplicación y se encarga de crear la ventana principal y los widgets necesarios, mediante las funciones explicadas en el módulo *Interface*. Es importante señalar que este módulo es esencial para el correcto funcionamiento de la aplicación y en tal sentido se recomienda revisar la documentación del anexo (manual de usuario).

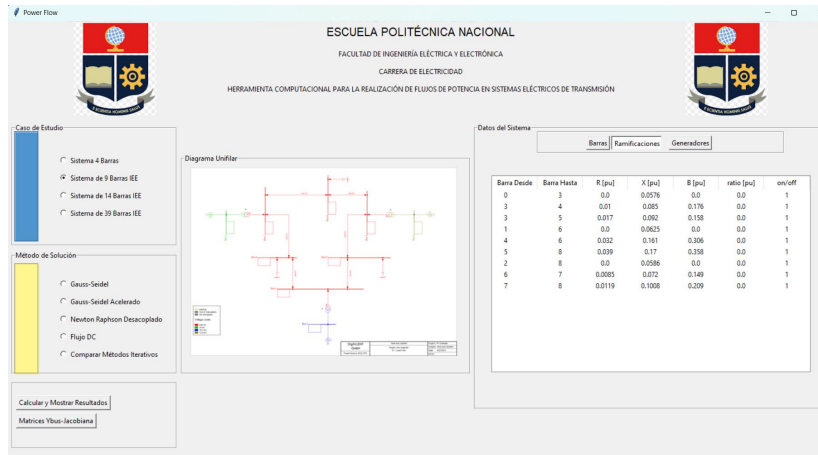


Figura 2.10. Ventana principal construida con la clase "PowerFlow"

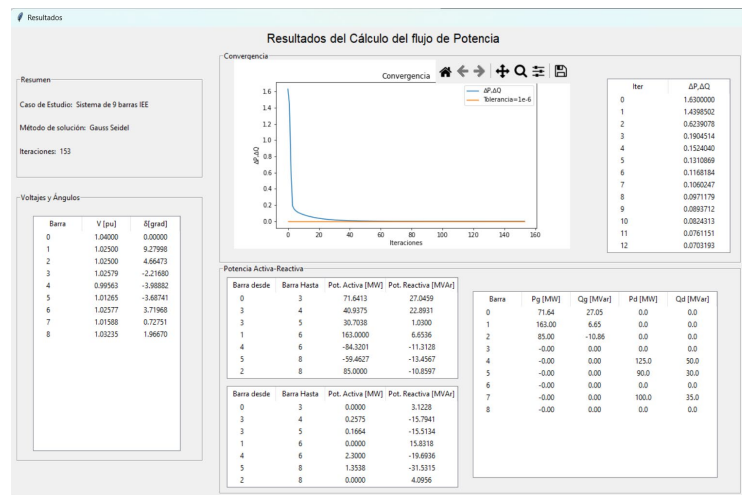
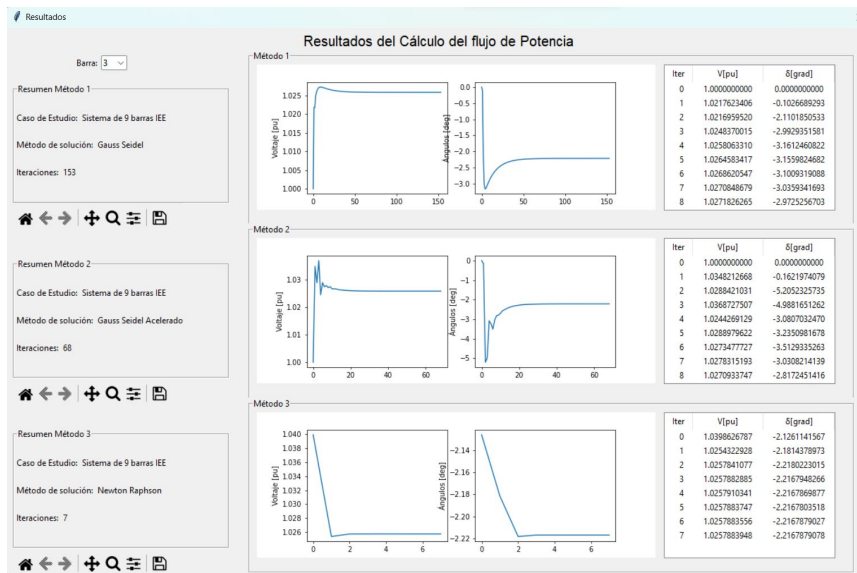
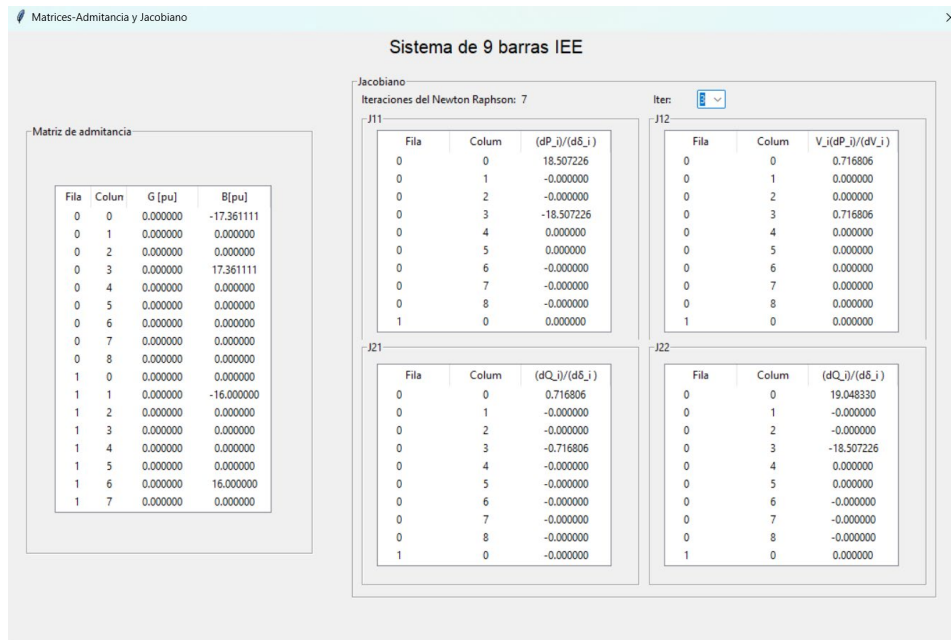


Figura 2.11. Ventanas secundarias construida con la clase "VentResultados" para visualizar voltajes y potencias del sistema.



**Figura 2.12.** Ventana secundaria construida con la clase VentResultados para la comparación de los métodos iterativos



**Figura 2.13.** Ventana secundaria construida con la clase "Admi\_Jac".

Se destaca el hecho de que la aplicación tiene la capacidad de mostrar la matriz jacobiana completa, lo que proporciona una visión detallada de la solución del flujo de potencia. Sin embargo, las iteraciones del método de Newton Raphson desacoplado solo utilizan las matrices  $J_{11}$  y  $J_{22}$ .



## **3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES**

### **3.1. Resultados**

En esta sección, se presentan los resultados obtenidos mediante la herramienta computacional desarrollada, se tomará como referencia el sistema de 9 barras IEEE con el objetivo de brindar una guía clara y útil para el usuario.

Se evalúa la precisión de la herramienta desarrollada utilizando sistemas de prueba estandarizados: sistema de 9 barras, sistema de 14 barras y sistema de 39 barras de IEEE (también el sistema ejemplo de 4 barras descrito en [5]). La comparación de los resultados obtenidos con la herramienta desarrollada toma como patrón los resultados obtenidos en el software PowerFactory.

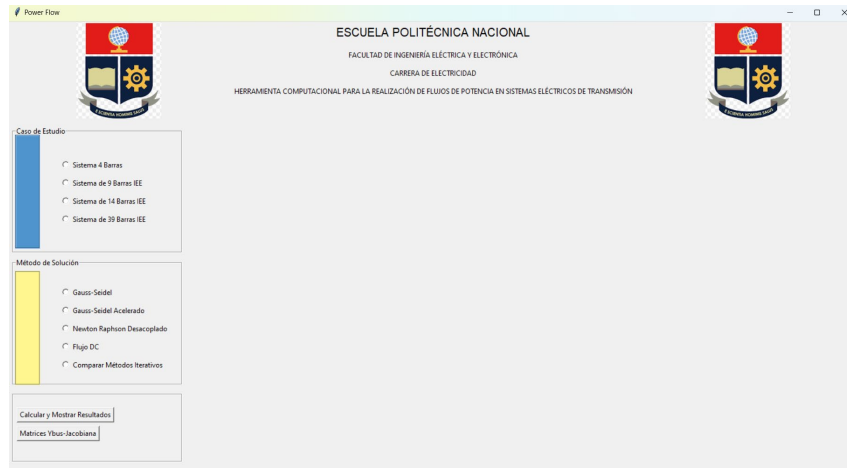
#### **3.1.1. Resultados de la aplicación desarrollada en Python**

Antes de iniciar la simulación del sistema de 9 barras IEEE, es recomendable revisar la guía de usuario presentada en la sección de anexos de este documento. En dicha guía, se explica los requisitos y necesidades de la aplicación para su funcionamiento.

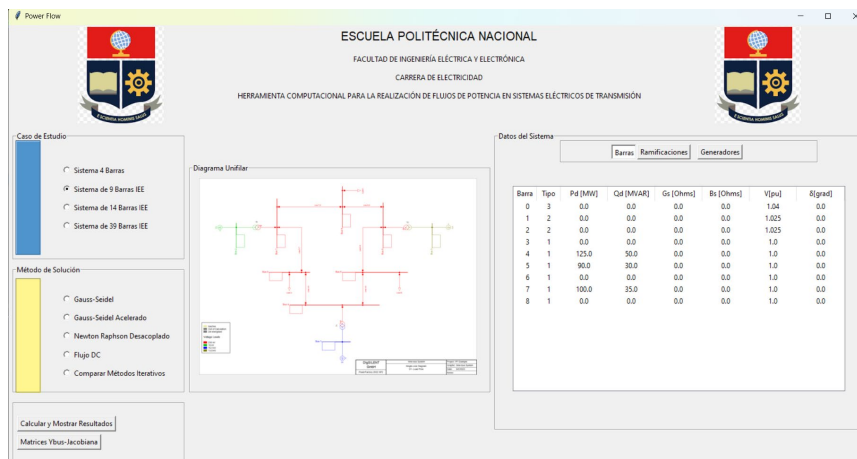
Como dato adicional, es importante tener en cuenta que en la aplicación el primer índice se considera como cero, por lo que la barra 1 en el sistema real corresponde a la barra 0 en la aplicación desarrollada. Esto aplica para todas las barras, ramificaciones y generadores presentes en el sistema.

##### **3.1.1.1. Ventana principal**

Después de ejecutar el módulo principal *main*, se mostrará la ventana principal (ver Figura 3.1), donde en la parte izquierda se encuentran los casos de estudio y los métodos de solución disponible. En este caso, se seleccionará el caso "*Sistema de 9 barras IEEE*", con motivo de explicar los resultados que se pueden visualizar en la aplicación. Al seleccionar el caso mencionado, se mostrará automáticamente en la parte derecha de la ventana principal, el diagrama unifilar y los datos de entrada del sistema, los cuales se pueden visualizar en las pestañas de datos de barras, ramificaciones o generadores que se pueden seleccionar según las necesidades del usuario (ver Figura 3.2).



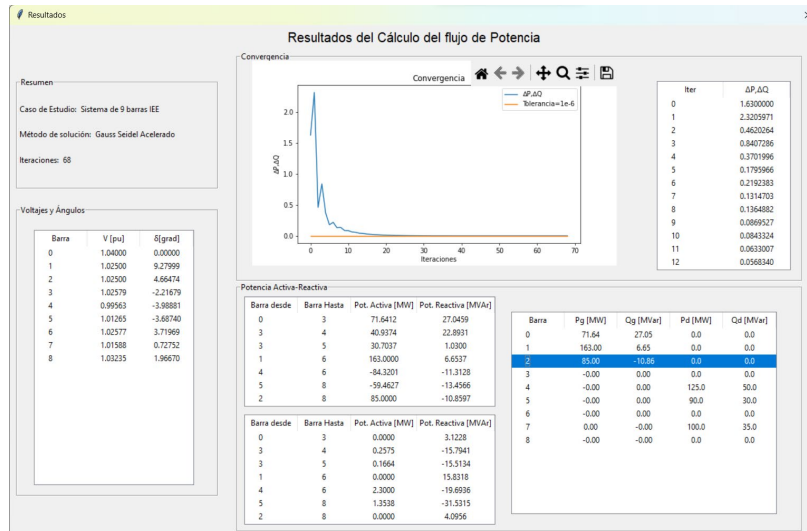
**Figura 3.1.** Ventana principal de la aplicación desarrollada.



**Figura 3.2.** Ventana principal, diagrama unifilar y datos de entrada

### 3.1.1.2. Ventana secundaria, Resultados y comparación de métodos iterativos.

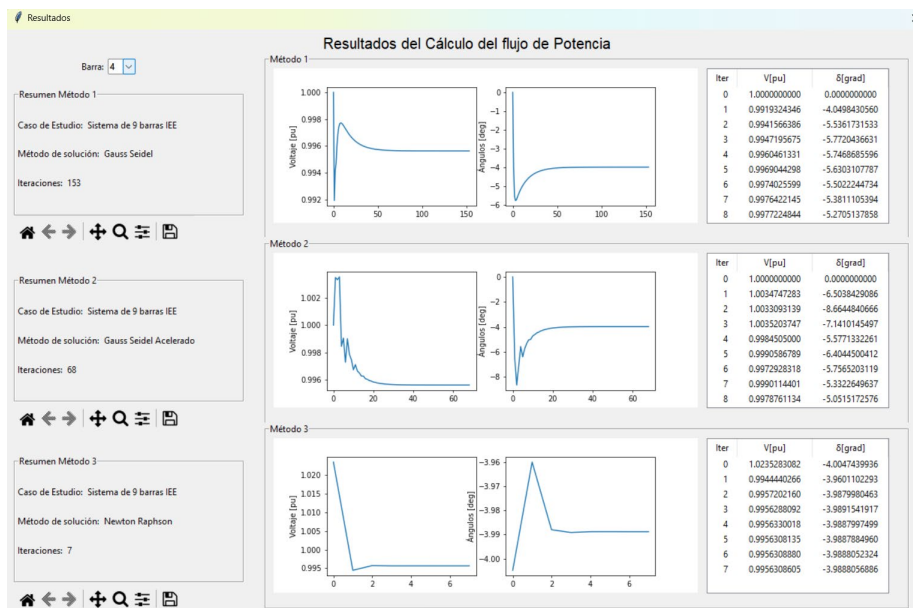
Después de seleccionar el caso de estudio, es posible elegir uno de los cuatro métodos de solución de flujo de potencia: Gauss-Seidel, Gauss-Seidel acelerado, Newton Raphson desacoplado y Flujo DC. Al presionar el botón "Calcular y Mostrar Resultados" se mostrarán los resultados del caso seleccionado en una ventana secundaria. En el ejemplo de la Figura 3.3 se seleccionó el método de Gauss Seidel con factor de aceleración, se realizaron 68 iteraciones, esto se puede observar en el cuadro resumen. La gráfica de convergencia muestra como el máximo error de potencia se acerca a la tolerancia de forma abrupta debido al método escogido. Además, se puede ver los flujos de potencia de cada rama y las pérdidas asociadas, así como la potencia generada y demanda del sistema después de la simulación del flujo de potencia.



**Figura 3.3.** Ventana secundaria, Voltajes y Potencias

Al regresar a la pantalla principal, se puede seleccionar el método "Flujo DC" y al presionar el botón "Calcular y Mostrar Resultados" se mostrará la ventana secundaria (ver Figura 3.3). La diferencia con los métodos iterativos es que no se mostrará la gráfica de convergencia ya que este método no requiere iteraciones.

En la ventana principal, en los métodos de solución, se cuenta con una opción adicional "Comparar Metodos iterativos", la cual compara todos los métodos iterativos del caso de estudio seleccionado. Al presionar el botón "Calcular y Mostrar Resultados", se abrirá la ventana que se presenta en la Figura 3.4.



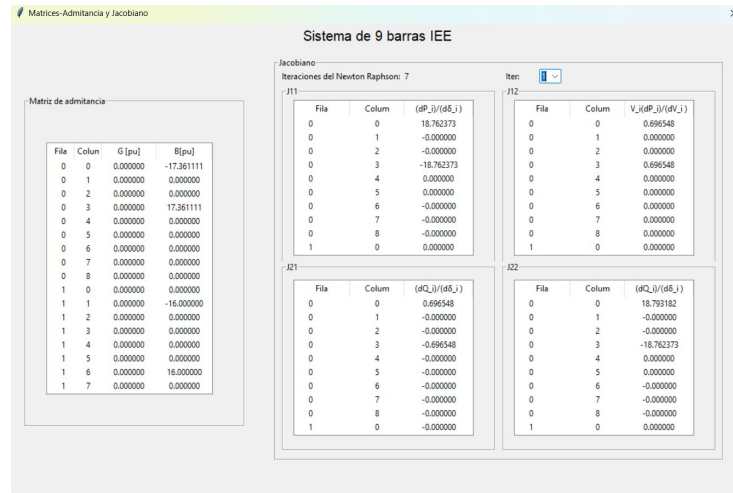
**Figura 3.4.** Ventana secundaria, método "Comparar métodos iterativos".

La ventana está compuesta por tres cuadros resumen que detallan el caso de estudio, el método utilizado y el número de iteraciones llevadas a cabo en cada método. Además, se incluye una lista desplegable que permite seleccionar la barra que se desea analizar para ver la convergencia del módulo y ángulo del voltaje de los tres métodos iterativos disponibles.

La Figura 3.4 proporciona información valiosa sobre el caso de estudio "Sistema de 9 barras IEEE". En particular, se puede observar que este caso converge después de 153, 68 y 7 iteraciones en los métodos de Gauss Seidel, Gauss Seidel acelerado y Newton Raphson desacoplado, respectivamente. Es interesante notar que el método de Gauss Seidel con factor de aceleración requiere menos iteraciones que el caso sin aceleración, aunque gráficamente se observa diferencias en los módulos y ángulos de voltaje antes de que la solución se estabilice. Por otro lado, el método Newton Raphson desacoplado requiere el menor número de iteraciones en comparación con los otros métodos y no muestra distorsiones considerables antes de la convergencia.

### **3.1.1.3. Ventana secundaria, Matriz de admitancia y Matriz Jacobiana.**

Para acceder a la última función de la aplicación, es necesario cerrar la ventana secundaria previa y seleccionar el botón "*Matrices Newton Raphson/Ybus*". Al hacerlo, se abrirá una segunda ventana en la que se puede observar la matriz de admitancia, se incluye una lista desplegable que permite visualizar las iteraciones realizadas por el método Newton Raphson desacoplado para el caso de estudio seleccionado. Al seleccionar una de estas iteraciones en la lista desplegable, se mostrará la matriz jacobiana completa. Cabe destacar que, aunque la ventana secundaria (Figura 3.5) muestra la matriz jacobiana completa, el código del método solo utiliza las submatrices  $J_{11}$  y  $J_{22}$  de la matriz jacobiana para la solución del caso de estudio.



**Figura 3.5.** Ventana secundaria, Matriz de admitancia y matriz jacobiana del método Newton Raphson desacoplado

### 3.1.1.4. Modificación de los datos de Entrada.

Dentro de la librería del programa desarrollado, es posible modificar los datos de entrada, como los datos de las barras, ramificaciones y generadores, mediante el modulo "*Casos\_de\_Estudio*". Este módulo contiene cuatro casos de estudio predefinidos, pero se puede personalizar para adaptarse a necesidades específicas. Por ejemplo, en el sistema de 9 barras IEEE, se puede cambiar el estado de una rama específica, como la rama 3-4, a apagado y ajustar los valores de voltaje de los generadores para observar cómo afecta el sistema con esta modificación.

```
# barra_i tipo Pd Qd Gs Bs Vm Va baseKV
bus = array([
[0, 3, 0, 0, 0, 0, 1.070, 0, 345],
[1, 2, 0, 0, 0, 0, 1.065, 0, 345],
[2, 2, 0, 0, 0, 0, 1.065, 0, 345],
[3, 1, 0, 0, 0, 0, 1, 0, 345],
[4, 1, 125, 50, 0, 0, 1, 0, 345],
[5, 1, 90, 30, 0, 0, 1, 0, 345],
[6, 1, 0, 0, 0, 0, 1, 0, 345],
[7, 1, 100, 35, 0, 0, 1, 0, 345],
[8, 1, 0, 0, 0, 0, 1, 0, 345]
])

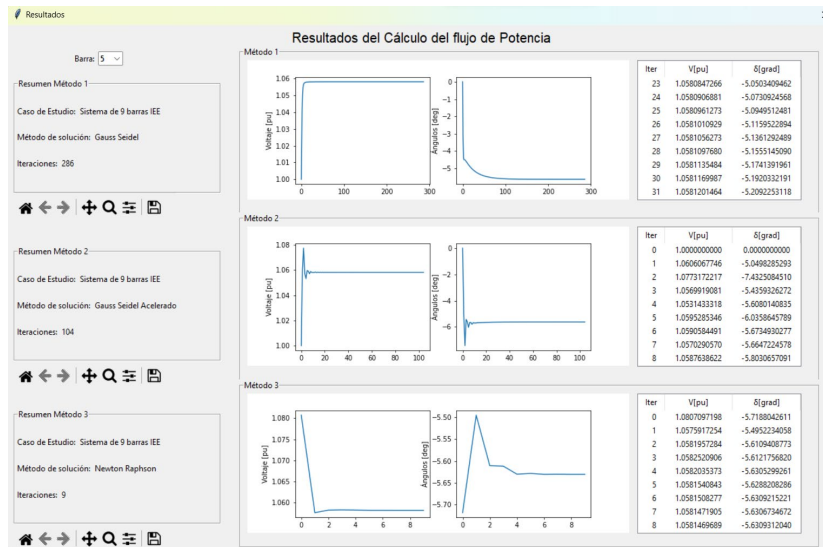
branch = array([
[0, 3, 0, 0.0576, 0, 0, 0, 1],
[3, 4, 0.010, 0.085, 0.176, 0, 0, 0],
[3, 5, 0.017, 0.092, 0.158, 0, 0, 1],
[1, 6, 0, 0.0625, 0, 0, 0, 1],
[4, 6, 0.0320, 0.1610, 0.306, 0, 0, 1],
[5, 8, 0.0390, 0.17, 0.3580, 0, 0, 1],
[2, 8, 0, 0.0586, 0, 0, 0, 1],
[6, 7, 0.0085, 0.072, 0.149, 0, 0, 1],
[7, 8, 0.0119, 0.1008, 0.209, 0, 0, 1]
])

# barra, Pg, Qg, Vg, mBase, status
gen = array([
[0, 0, 0, 1.070, 0, 1],
[1, 163, 0, 1.065, 0, 1],
[2, 85, 0, 1.065, 0, 1]
])
```

**Figura 3.6.** Datos de entrada modificados para el sistema de 9 barras IEEE

después de modificar los datos de entrada, es posible realizar una simulación y así poder observar como el sistema varía ante esas modificaciones. En efecto, la Figura 3.7 muestra

que la solución converge en la interacción 286 para el método de Gauss Seidel, en la interacción 104 para el método de Gauss Seidel con factor de aceleración y en la interacción 9 para el método de Newton Raphson.



**Figura 3.7.** Comparación de los métodos iterativos del sistema de 9 barras IEEE con datos de entrada modificados

Comparación de resultados del sistema IEEE 9 barras modificado con el software de referencia PowerFactory

Para validar los resultados obtenidos con la aplicación desarrollada, se realizó una comparación cuantitativa con los resultados obtenidos por el software comercial Power Factory. Las Tablas 3.1 y 3.2 muestra los valores del módulo y ángulo de voltaje, así como la potencia generada después de la simulación en los dos softwares. Se presentan los errores relativos obtenidos en la misma tabla.

**Tabla 3.1.** Comparación de los valores de voltaje, la aplicación desarrollada vs. PowerFactory

Barra_i	Voltaje, PowerFactory		Voltaje, Python		Errores	
	[pu]	[deg]	[pu]	[deg]	Modulo	ángulo
Bus 1	1.07	0	1.07	0	0	0
Bus 2	1.065	-1.3	1.065	-1.25292355	0	-0.03621265
Bus 3	1.065	-1.8	1.065	-1.80130478	0	0.00072488
Bus 4	1.072	-2.2	1.07181763	-2.166745	-0.00017012	-0.01511591
Bus 5	0.902	-18.2	0.90216381	-18.2487122	0.00018161	0.0026765
Bus 6	1.058	-5.6	1.06E+00	-5.63E+00	0.00013845	0.005517
Bus 7	1.035	-6.6	1.03498745	-6.55595761	-1.2126E-05	-0.00667309
Bus 8	1.036	-7.8	1.04E+00	-7.78E+00	-5.8929E-05	-0.00232221
Bus 9	1.066	-4.3	1.06640592	-4.3149614	0.00038079	0.0034794

**Tabla 3.2.** Comparación de los valores de potencia generada, aplicación desarrollada vs. PowerFactory

Barra_i	PowerFactory		Python		Errores	
	[MW]	[Mvar]	[MW]	[Mvar]	P. Activa	P. Reactiva
Bus 1	75.277677	-1.953476	75.2772667	-1.95296087	-5.4505E-06	-0.0002637
Bus 2	163.00001	58.690913	162.999479	58.6900236	-3.2454E-06	-1.5154E-05
Bus 3	85.000002	-0.690058	84.9999227	-0.69028669	-9.3294E-07	0.00033141

Se puede verificar en las Tablas 3.1 y 3.2 que los errores relativos son bajos y se encuentran dentro de los límites aceptables. Esto confirma que la aplicación desarrollada es confiable y puede ser utilizada con seguridad.

### **3.1.2. Comparación de Resultados entre PowerFactory y Aplicación desarrollada.**

Se realizó una comparación de los resultados obtenidos mediante la aplicación desarrollada y los resultados del software de referencia PowerFactory. Esta comparación permitió identificar los errores máximos en los resultados obtenidos con la aplicación desarrollada para los casos de estudio considerados. A continuación, se muestra dichos errores.

#### **3.1.2.1. Sistema de 4 barras**

El sistema de cuatro barras se presenta como un ejemplo en el libro de Grainger & Stevenson Jr. Este sistema está compuesto por 2 generadores, 4 barras y 4 cargas, y los detalles relevantes se encuentran en el ANEXO I.

**Tabla 3.3.** Comparación de errores relativos entre la aplicación desarrollada y PowerFactory para el sistema de 4 barras

Método	Voltaje en las barras		Potencia Generada	
	[V]	[deg]	Pg.	Qg.
<b>Gauss Seidel</b>	1.2384E-06	4.5295E-06	1.092E-06	3.886E-05
<b>Gauss Seidel Acelerado</b>	1.2281E-06	4.7218E-06	4.6572E-07	3.8826E-05
<b>NRD</b>	1.1145E-06	7.104E-06	1.4828E-06	3.7123E-05
<b>Flujo DC</b>	-----	9E-08	0	-----

### 3.1.2.2. Sistema de 9 barras IEEE

El sistema de 9 barras descrito en el libro de P. Anderson & A. Fouad es un modelo estandarizado IEEE. La estandarización de este sistema proporciona un marco de referencia claro para los resultados esperados, lo que permite una comparación precisa tendiente a evaluar la efectividad de la herramienta desarrollada.

Los datos de los generadores, cargas, transformadores y líneas del sistema se detallan en el ANEXO II. Estos datos fueron obtenidos de la documentación del ejemplo Nine-bus System de PowerFactory. Es importante señalar que, según documentación propia de Power Factory, en los resultados de los ángulos a partir de la barra 4 se debe restar 150 grados debido al desfaseamiento del transformador T1.

**Tabla 3.4.** Comparación de errores relativos entre la aplicación desarrollada y PowerFactory para el sistema de 9 barras.

Método	Voltaje en las barras		Potencia Generada	
	[V]	[deg]	Pg.	Qg.
<b>Gauss Seidel</b>	4.2957E-06	3.5737E-05	3.5203E-06	1.9763E-05
<b>Gauss Seidel Acelerado</b>	4.2858E-06	2.4535E-05	2.4162E-06	1.8902E-05
<b>NRD</b>	4.2858E-06	5.2918E-06	9.9105E-08	1.8134E-05
<b>Flujo DC</b>	----- ---	1.2663E-07	1.2663E-07	-----

### 3.1.2.3. Sistema de 14 barras IEEE

El sistema de 14 barras está estandarizado por IEEE, lo que lo convierte en una referencia útil para validar los resultados obtenidos por la aplicación desarrollada en comparación con el software comercial PowerFactory. Los detalles del sistema se encuentran en el ANEXO



III, donde se muestran los datos extraídos de la documentación de los ejemplos de PowerFactory.

**Tabla 3.5.** Comparación de errores relativos entre la aplicación desarrollada y PowerFactory para el sistema de 14 barras

Método	Voltaje en las barras		Potencia Generada	
	[V]	[deg]	Pg.	Qg.
<b>Gauss Seidel</b>	5.05889E-06	2.3736E-06	2.15749E-06	3.03922E-05
<b>Gauss Seidel Acelerado</b>	5.05889E-06	1.40845E-06	1.865E-06	2.97712E-05
<b>NRD</b>	0.000264466	0.000786993	2.30574E-05	0.003457688
<b>Flujo DC</b>	-----	0.020651855	0	-----

### 3.1.2.4. Sistema de 39 barras IEEE

El sistema de 39 barras de Nueva Inglaterra es un modelo simplificado del sistema de transmisión de alta tensión en Estados Unidos. Fue presentado por primera vez en 1970 y desde entonces se ha utilizado con frecuencia para la investigación científica y publicaciones. El anexo IV muestra los datos del sistema extraídos de la documentación de los ejemplos de PowerFactory.

**Tabla 3.6.** Comparación de errores relativos entre la aplicación desarrollada y PowerFactory para el sistema de 39 barras

Método	Voltaje en las barras		Potencia Generada	
	[V]	[deg]	Pg.	Qg.
<b>Gauss Seidel</b>	5.05889E-06	0.094295175	0.03164557	0.094419138
<b>Gauss Seidel Acelerado</b>	5.05889E-06	1.40845E-06	0.031645571	0.094419138
<b>NRD</b>	0.000264466	0.000786993	0.03164263	0.094438373
<b>Flujo DC</b>	-----	0.020651855	0	-----

Como se puede evidenciar en las tablas, los errores relativos de los sistemas analizados no superan el 5% por lo cual se puede afirmar que la aplicación desarrollada tiene una alta precisión en sus resultados.

## 3.2. Conclusiones

1. La implementación del algoritmo de flujos de potencia mediante los métodos de Gauss Seidel, Newton Raphson desacoplado y Flujo DC, ha permitido tener a disposición una herramienta de cálculo que brinda opciones para la resolución de sistemas eléctricos de potencia de prueba. Al haberse realizado la implementación en un lenguaje de código abierto, se posibilita la adaptación y mejora del algoritmo para futuros proyectos y aplicaciones.
2. La aplicación desarrollada para la solución de flujos de potencia ha demostrado precisión en la validación de los resultados obtenidos en comparación con el software comercial PowerFactory. Esto indica que la aplicación es confiable en la determinación de flujos de potencia de sistemas eléctricos. Es importante destacar que la precisión en la solución de flujos de potencia es fundamental para garantizar la estabilidad y operación óptima de los sistemas eléctricos.
3. El desarrollo de esta herramienta computacional permitirá a los profesores de ingeniería eléctrica implementar métodos de enseñanza didácticos para la solución de flujos de potencia, permitiendo a los estudiantes un aprendizaje interactivo y práctico en el campo de la ingeniería eléctrica.

### 3.3. Recomendaciones

1. Se recomienda que, antes de utilizar la aplicación desarrollada, los usuarios instalen todas las librerías descritas en el presente trabajo de titulación. Esto garantizará el correcto funcionamiento de la herramienta y permitirá que los usuarios puedan aprovechar al máximo todas las funcionalidades que ofrece. Es importante mantener estas librerías actualizadas y revisar periódicamente la documentación oficial de las mismas para estar al tanto de posibles actualizaciones o mejoras.
2. Recomendamos a los usuarios interesados en programación en Python que aprendan a usar programas de manejo de datos como MySQL para poder manipular datos de entrada desde la interfaz, lo que permitirá un manejo más eficiente de los datos en el programa desarrollado. Se recomienda también buscar tutoriales y recursos en línea para aprender a integrar Python con MySQL y aprovechar todas las posibilidades que ofrece esta combinación en el manejo de datos. Con esta habilidad, los usuarios podrán desarrollar programas más completos y efectivos para la solución de problemas en diferentes áreas.

## 4. REFERENCIAS BIBLIOGRÁFICAS

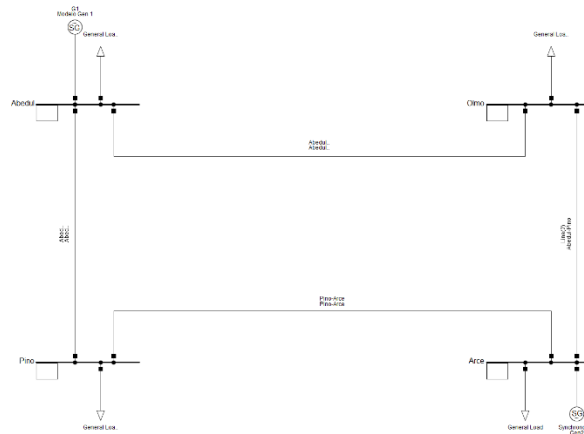
- [1] A. Gómez, A. Conejo y C. Cañizares, *Electric and Energy Systems Analysis and Operation*, New York: CRC Press, 2009.
- [2] A. Fitzgerald, C. Kingsley y S. Umans, *Máquinas eléctricas*, México: Mc Graw Hill, 2004.
- [3] S. Chapman, *Máquinas eléctricas*, Bogotá: Mc Graw Hill, 200.
- [4] B. Gurú, *Máquinas eléctricas y transformadores*, México: Oxford University Press, 2006.
- [5] J. Grainger y W. Stevenson, *Análisis de sistema de potencia*, México: Mc Graw Hill, 1996.
- [6] G. Enríquez, *El libro practico de los generadores, transformadores y motores electricos*, México: Limusa, 2000.
- [7] J. Játiva, *Apuntes de clases de Sistemas Eléctricos de Potencia*, Escuela Politécnica Nacional, 2020.
- [8] R. Zimmerman, C. Murillo y R. Thomas, «MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education,» *Power Systems*, vol. 26, nº 1, pp. 12-19, Febrero 2011.
- [9] A. Aioboman, P. James, I. Araga, C. Wamdeo y I. Okakwu, «Contingency Analysis on the Nigerian Power Systems Network,» *IEEE PES/IAS Power África*, vol. 1, nº 1, pp. 70-75, Agosto 2019.
- [10] G. Arguello, *Análisis y Control de Sistemas Eléctricos de Potencia*, Quito: Escuela Politécnica Nacional, 1990.
- [11] A. Wood y B. Wollenberg, *Power generation operation and control*, New York: John Wiley y Sons Inc, 1984.
- [12] J. Játiva, G. Constante y C. Cabrera, «Flujo de Potencia por Newton-Raphson con el Jacobiano Calculado en las Ecuaciones de Errores de Potencia,» *Revista Politécnica*, vol. 33, nº 1, enero 2014.

## 5. ANEXOS

### ANEXO I

#### DATOS DEL SISTEMA 4 BARRAS

Los datos que se muestran a continuación fueron extraídos de [5]:



**Figura A. 1** Diagrama del sistemas de 4 barras

**Tabla A. 1** Datos de Barras del sistema de 4 Barras

Barras	Generación		Carga		V, pu	V, grad	Tipo
	P,MW	Q,Mvar	P,MW	Q,Mvar			
1	----	-----	50	30.99	1	0	P $\delta$
2	0	0	170	105.35	1	0	PQ
3	0	0	200	123.94	1	0	PQ
4	318	....	80	49.58	1.02	0	PV

**Tabla A. 2** Datos de las ramas del sistema 4 barras

Barra, Desde	Barra, Hasta	R, pu	X, pu	B, pu
1	2	0.01008	0.0504	0.1025
1	3	0.00744	0.0372	0.0775
2	4	0.00744	0.0372	0.0775
3	4	0.01272	0.0636	0.1275

## ANEXO II

### DATOS DEL SISTEMA 9 BARRAS IEEE

Los datos que se muestran a continuación fueron extraídos de la documentación anexada con el ejemplo Nine-bus System de DigSILENT PowerFactory:

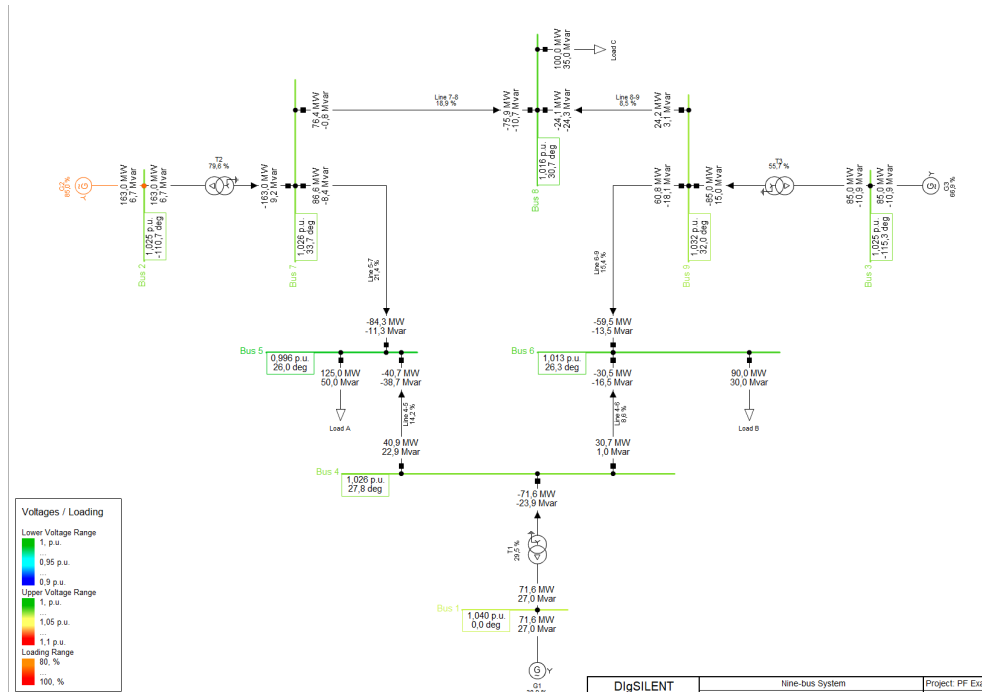


Figura A. 2 Diagrama unifilar del sistema de 9 barras IEEE

Tabla A. 3 Datos de Barras del sistema de 9 barras IEEE

Barras	Generación		Carga		V, pu	V, grad	Tipo
	P,MW	Q,Mvar	P,MW	Q,Mvar			
1	----	----	50	30.99	1.04	0	P $\delta$
2	163	0	170	105.35	1.025	0	PV
3	85	0	200	123.94	1.025	0	PV
4	0	0	80	49.58	1	0	PQ
5	0	0	125	50	1		PQ
6	0	0	90	30	1		PQ
7	0	0	0	0	1		PQ
8	0	0	100	35	1		PQ
9	0	0	0	0	1		PQ

**Tabla A. 4** Datos de las ramas del sistema 9 barras

<b>Barra, Desde</b>	<b>Barra, Hasta</b>	<b>R, pu</b>	<b>X, pu</b>	<b>B, pu</b>	<b>t, pu</b>
1	4	0	0.0576	0	0
2	7	0	0.0625	0	0
3	9	0	0.0586	0	0
4	5	0.0100	0.0850	0.176	0
4	6	0.0170	0.0920	0.158	0
5	7	0.0320	0.1610	0.306	0
6	9	0.0390	0.1700	0.3580	0
7	8	0.0085	0.0720	0.149	0
8	9	0.0119	0.1008	0.209	0





**Tabla A. 6** Datos de las ramas del sistema 14 barras

<b>Barra, Desde</b>	<b>Barra, Hasta</b>	<b>R, pu</b>	<b>X, pu</b>	<b>B, pu</b>	<b>t, pu</b>
1	2	0.01938	0.05917	0.0528	0
1	5	0.05403	0.22304	0.0492	0
2	3	0.04699	0.19797	0.0438	0
2	4	0.05811	0.17632	0.0374	0
2	5	0.05695	0.17388	0.034	0
3	4	0.06701	0.17103	0.0346	0
4	5	0.01335	0.04211	0.0128	0
4	7	0	0.20912	0	0.978
4	9	0	0.55618	0	0.969
5	6	0	0.25202	0	0.932
6	11	0.09498	0.1989	0	0
6	12	0.12291	0.25581	0	0
6	13	0.06615	0.13027	0	0
7	8	0	0.17615	0	0
7	9	0	0.11001	0	0
9	10	0.03181	0.0845	0	0
9	14	0.12711	0.27038	0	0
10	11	0.08205	0.19207	0	0
12	13	0.22092	0.19988	0	0
13	14	0.17093	0.34802	0	0

# ANEXO IV

## DATOS DEL SISTEMA 39 BARRAS IEEE

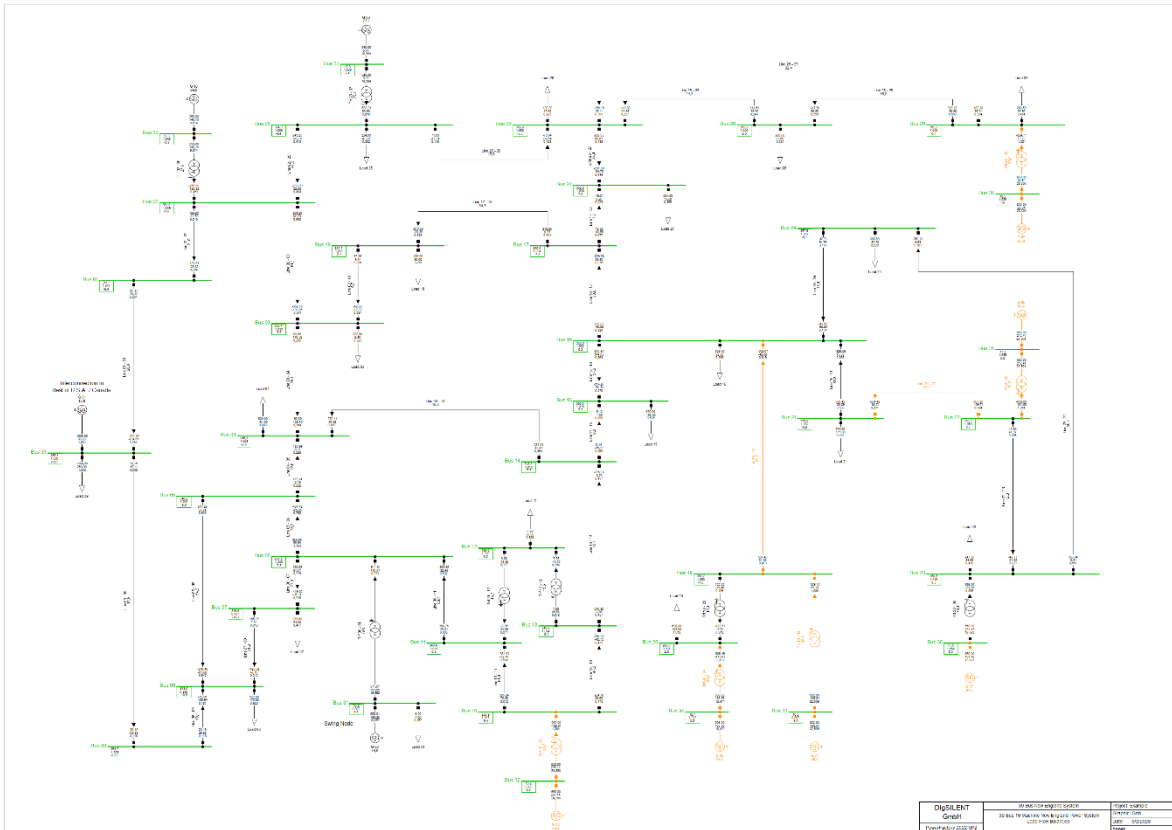


Figura A. 4 Diagrama unifilar del sistema de 39 barras IEEE.

**Tabla A. 7** Datos de Barras del sistema de 39 barras IEEE.

Barras	Generación		Carga		V, pu	V, grad	Tipo
	P,MW	Q,Mvar	P,MW	Q,Mvar			
1	0	0	0	0	1	0	PQ
2	0	0	0	0	1	0	PQ
3	0	0	322	2.4	1	0	PQ
4	0	0	500	184	1	0	PQ
5	0	0	0	0	1	0	PQ
6	0	0	0	0	1	0	PQ
7	0	0	233.8	84	1	0	PQ
8	0	0	522	176	1	0	PQ
9	0	0	0	0	1	0	PQ
10	0	0	0	0	1	0	PQ
11	0	0	0	0	1	0	PQ
12	0	0	7.5	88	1	0	PQ
13	0	0	0	0	1	0	PQ
14	0	0	0	0	1	0	PQ
15	0	0	320	153	1	0	PQ
16	0	0	329	32.3	1	0	PQ
17	0	0	0	0	1	0	PQ
18	0	0	158	30	1	0	PQ
19	0	0	0	0	1	0	PQ
20	0	0	680	103	1	0	PQ
21	0	0	274	115	1	0	PQ
22	0	0	0	0	1	0	PQ
23	0	0	247.5	84.6	1	0	PQ
24	0	0	308.6	-92.2	1	0	PQ
25	0	0	224	47.2	1	0	PQ
26	0	0	139	17	1	0	PQ
27	0	0	281	75.5	1	0	PQ
28	0	0	206	27.6	1	0	PQ
29	0	0	283.5	26.9	1	0	PQ
30	250	0	0	0	1.0475	0	P $\delta$
31	0	0	9.2	4.6	0.982	0	PV
32	650	0	0	0	0.9831	0	PV
33	632	0	0	0	0.9972	0	PV
34	508	0	0	0	1.0123	0	PV
35	650	0	0	0	1.0493	0	PV
36	560	0	0	0	1.0635	0	PV
37	540	0	0	0	1.0278	0	PV
38	830	0	0	0	1.0265	0	PV
39	1000	0	1104	250	1.03	0	PV

**Tabla A. 8** Datos de las ramas del sistema 39 barras IEEE

<b>Barra, Desde</b>	<b>Barra, Hasta</b>	<b>R, pu</b>	<b>X, pu</b>	<b>B, pu</b>	<b>t, pu</b>
1	2	0.0035	0.0411	0.6987	0
1	39	0.001	0.025	0.75	0
2	3	0.0013	0.0151	0.2572	0
2	25	0.007	0.0086	0.146	0
2	30	0	0.0181	0	1.025
3	4	0.0013	0.0213	0.2214	0
3	18	0.0011	0.0133	0.2138	0
4	5	0.0008	0.0128	0.1342	0
4	14	0.0008	0.0129	0.1382	0
5	6	0.0002	0.0026	0.0434	0
5	8	0.0008	0.0112	0.1476	0
6	7	0.0006	0.0092	0.113	0
6	11	0.0007	0.0082	0.1389	0
6	31	0	0.025	0	1.07
7	8	0.0004	0.0046	0.078	0
8	9	0.0023	0.0363	0.3804	0
9	39	0.001	0.025	1.2	0
10	11	0.0004	0.0043	0.0729	0
10	13	0.0004	0.0043	0.0729	0
10	32	0	0.02	0	1.07
12	11	0.0016	0.0435	0	1.006
12	13	0.0016	0.0435	0	1.006
13	14	0.0009	0.0101	0.1723	0
14	15	0.0018	0.0217	0.366	0
15	16	0.0009	0.0094	0.171	0
16	17	0.0007	0.0089	0.1342	0
16	19	0.0016	0.0195	0.304	0
16	21	0.0008	0.0135	0.2548	0
16	24	0.0003	0.0059	0.068	0
17	18	0.0007	0.0082	0.1319	0
17	27	0.0013	0.0173	0.3216	0
19	20	0.0007	0.0138	0	1.06
19	33	0.0007	0.0142	0	1.07
20	34	0.0009	0.018	0	1.009
21	22	0.0008	0.014	0.2565	0
22	23	0.0006	0.0096	0.1846	0
22	35	0	0.0143	0	1.025
23	24	0.0022	0.035	0.361	0
23	36	0.0005	0.0272	0	1
25	26	0.0032	0.0323	0.513	0
25	37	0.0006	0.0232	0	1.025
26	27	0.0014	0.0147	0.2396	0

26	28	0.0043	0.0474	0.7802	0
26	29	0.0057	0.0625	1.029	0
28	29	0.0014	0.0151	0.249	0
29	38	0.0008	0.0156	0	1.025

**ANEXO V**  
MANUAL DE USUARIO

# Manual de Usuario PowerFlow



Elaborado por: Roberto  
Gabriel Sumba Caceres

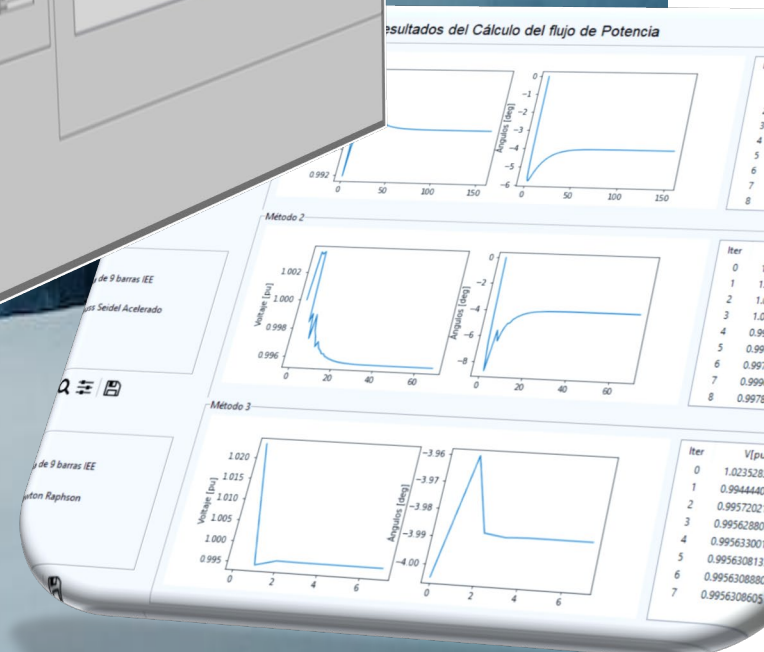
Director: Dr. Hugo Arcos

UNIVERSIDAD TÉCNICA NACIONAL  
INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
CARRERA DE ELECTRICIDAD  
ANÁLISIS DE FLUJOS DE POTENCIA EN SISTEMAS ELÉCTRICOS DE TRANSMISIÓN

Datos del Sistema

Barra Ramificaciones Generadores

Barra Desde	Barra Hasta	R (pu)	X (pu)	B (pu)	ratio (pu)	coef
0	3	0.0	0.0376	0.0	0.0	1
3	4	0.01	0.085	0.176	0.0	1
3	5	0.017	0.082	0.130	0.0	1
1	6	0.0	0.0625	0.0	0.0	1
4	6	0.032	0.161	0.306	0.0	1
5	8	0.039	0.17	0.358	0.0	1
2	8	0.0	0.0566	0.0	0.0	1
6	7	0.0025	0.072	0.149	0.0	1
7	8	0.0119	0.1000	0.209	0.0	1



## 1. introducción.

El programa desarrollado en el presente Trabajo de Integración Curricular proporciona soluciones precisas y eficientes para el análisis de flujos de potencia en sistemas eléctricos de potencia. El programa utiliza los métodos de Gauss-Seidel, Gauss-Seidel acelerado, Newton Raphson desacoplado y flujo DC para obtener soluciones rápidas y precisas. Además, la interfaz del programa es intuitiva y fácil de usar, lo que permite al usuario analizar y ver gráficamente el estado del sistema seleccionado. En resumen, este programa es una herramienta valiosa para los ingenieros eléctricos y estudiantes que necesitan analizar sistemas eléctricos de potencia con precisión y eficiencia.

## 2. Requerimientos des sistema

Para poder utilizar la aplicación sin inconvenientes, es necesario instalar previamente algunos paquetes en el sistema operativo. Específicamente, se requiere tener instalados los paquetes de Python, SciPy, Numpy y Matplotlib.

Antes de proceder con la instalación de la aplicación, es importante asegurarse de tener Python instalado en el sistema. En caso de no tenerlo, se puede seguir los siguientes pasos para instalarlo en diferentes sistemas operativos:

- Windows:
  - Descargar el instalador de Python desde el sitio web oficial: <https://www.python.org/downloads/windows/>.
  - Ejecutar el archivo descargado y seguir las instrucciones del instalador para completar la instalación
- Mac:
  - Descarga el instalador de Python desde el sitio web oficial : <https://www.python.org/downloads/mac-osx/>.
  - Ejecutar el archivo descargado y seguir las instrucciones del instalado para completar la instalación.
- Linux:
  - Abrir la terminal y ejecutar el siguiente comando para instalar Python: `sudo apt-get install python`
  - Si se utiliza una distribución de Linux diferente, consultar la documentación para encontrar el método adecuado para instalar Python.

Una vez que Python está instalado, se puede proceder con la instalación de los paquetes adicionales necesarios para la aplicación, como se mencionó anteriormente.



### 3. Instalación de Python

Para instalar los paquetes de Python necesarios para utilizar la aplicación en diferentes sistemas operativos, se puede seguir los siguientes pasos:

- Windows:
  - Abrir la línea de comandos o terminal en el sistema.
  - Ejecutar el siguiente comando para instalar los paquetes de Numpy y SciPy: `pip install numpy scipy`.
  - Ejecutar el siguiente comando para instalar el paquete de Matplotlib: `pip install matplotlib`.
- Mac:
  - Abrir la línea de comandos o terminal en el sistema.
  - Ejecutar el siguiente comando para instalar los paquetes de Numpy y SciPy: `pip install numpy scipy`.
  - Ejecutar el siguiente comando para instalar el paquete de Matplotlib: `pip install matplotlib`.
- Linux:
  - Abrir la terminal en el sistema.
  - Ejecutar el siguiente comando para instalar los paquetes de Numpy y SciPy: `sudo apt-get install python-numpy python-scipy`.
  - Ejecutar el siguiente comando para instalar el paquete de Matplotlib: `sudo apt-get install python-matplotlib`.

Es importante mencionar que, en algunas distribuciones de Linux, es posible que sea necesario utilizar un gestor de paquetes diferente al comando 'apt-get' para instalar los paquetes necesarios. En cualquier caso, es recomendable buscar la documentación del sistema operativo o distribución específica para asegurarse de utilizar el método adecuado de instalación.

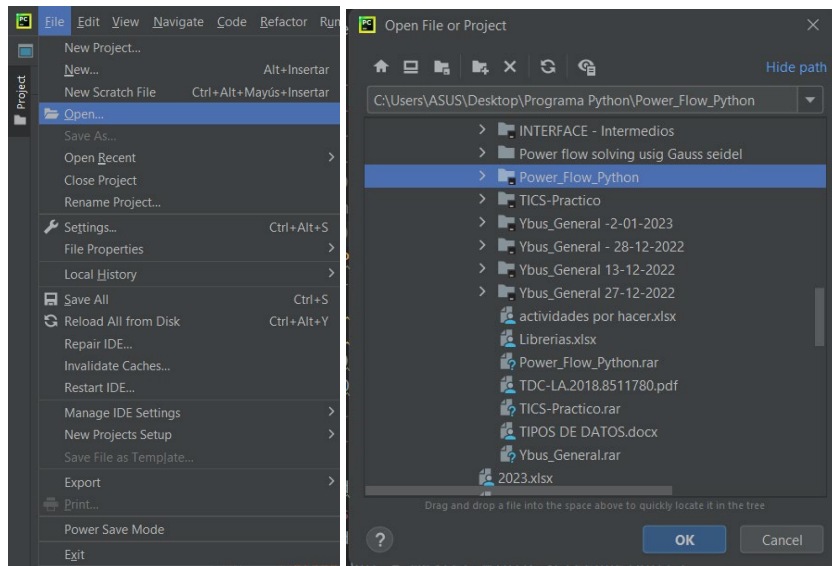
Por último, es fundamental contar con una IDE para el desarrollo de aplicaciones en Python. Una IDE es una herramienta fundamental para el desarrollo de aplicaciones en Python. Permite una edición más eficiente del código, una depuración más efectiva, y facilita el proceso de ejecución y prueba. Es importante señalar que existen diferentes IDEs para Python, cada una con sus propias características y herramientas. Algunas de las IDEs más utilizadas en la actualidad incluyen PyCharm, Visual Studio Code, Spyder, Jupyter Notebook, entre otras.

En general, la instalación de una IDE para Python puede variar según la distribución del sistema operativo y la versión de la IDE. Sin embargo, gracias a la comunidad de desarrolladores, es posible encontrar en línea guías detalladas y tutoriales que explican paso a paso el proceso de instalación para cada IDE.

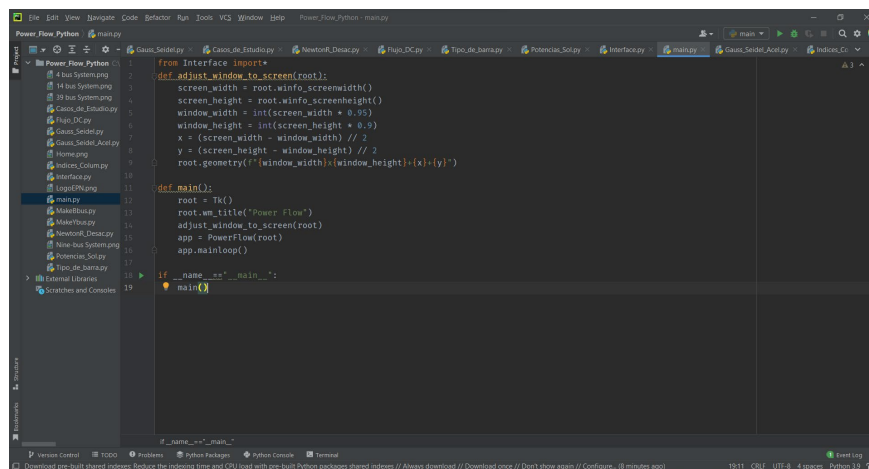
#### 4. Configuración de IDE Pycharm.

La aplicación desarrollada en Python se ha ejecutado utilizando la versión comunitaria de PyCharm. A continuación, se detallan los pasos necesarios para llevar a cabo esta operación.

1. Abrir PyCharm.
2. En la barra de menú principal dar clic en "File".
3. Seleccionar "Open" en el menú desplegable.
4. Navega hasta la ubicación donde se encuentra el proyecto "**Power\_Flow\_Python**" y selecciónalo.



Una vez que el proyecto se ha abierto en Pycharm, se puede comenzar a trabajar en él.



El proyecto elegido, que se refiere a la aplicación en cuestión, cuenta con diversos módulos que posibilitan llevar a cabo análisis de los flujos de potencia en sistemas eléctricos. A continuación, se describirán dichos módulos.

## 5. Módulos de la aplicación desarrollada para flujos de potencia

Módulo "Indices\_Colum":

Este módulo establece constantes que indican los índices de las columnas en las matrices de los datos de entrada. De esta manera, se facilita la comprensión y accesibilidad del código del programa.

Módulo "Casos\_de\_Estudio":

Este módulo almacena los datos de los diferentes sistemas de potencia, como los sistemas IEEE de 9, 14 y 39 barras, así como el sistema de 4 barras referenciado en [5]. Los datos están almacenados en funciones que devuelven tres matrices: bus, branch y gen. Estas matrices pueden ser modificadas según considere el usuario.

Módulo "Tipo\_de\_barras":

Este módulo almacena una función que forma una lista de índices para las barras  $V\delta$ , PV y PQ a partir de las matrices bus y gen. Si no se define la barra  $V\delta$ , la función elige la primera barra PV como  $V\delta$  y se quita de la lista de barras PV.

Módulo "MakeYbus":

Este módulo construye la matriz de admitancia a partir de las matrices bus y branch. Calcula los elementos de la matriz de admitancia de cada rama utilizando la resistencia y reactancia de la línea, además de la susceptancia conectada en paralelo a la barra. También calcula las admitancias en paralelo y se construyen las matrices de conexión  $C_f$  y  $C_t$  que se usan para conectar las ramas con las barras.

Módulo "MakeBbus":

Este módulo permite obtener la matriz de susceptancia del sistema a partir de las matrices bus y branch. Calcula la matriz B que representa la relación entre la potencia real inyectada en cada barra y los ángulos de voltaje de cada barra.

Módulo "Gauss\_Seidel":

Este módulo implementa el algoritmo de Gauss-Seidel para resolver el flujo de potencia en un sistema eléctrico. El algoritmo se realiza en un bucle que se ejecuta hasta que se alcance la tolerancia o se alcance el número máximo de iteración.

Módulo "Gauss\_Seidel\_Acel":

Este módulo es similar al módulo "Gauss\_Seidel", pero incluye el factor de aceleración en el algoritmo.

Módulo "NewtonR\_Desac":

Este módulo implementa el método de Newton para resolver el flujo de potencia en un sistema eléctrico. La función toma como entrada la matriz de admitancia del sistema eléctrico, la potencia especificada, los voltajes iniciales, una lista de índices que indican el tipo de barra y dos variables llamadas "Var" e "it\_jac". En cada iteración, se evalúa la matriz Jacobiana y se resuelve un sistema de ecuaciones lineales para actualizar los voltajes en las barras PQ y PV.

Módulo "Interface":

El módulo contiene elementos de programación, como clases y funciones, que posibilitan la creación de una interfaz de usuario visual y la presentación efectiva y legible de los resultados obtenidos tras resolver el análisis de flujo de potencia. Asimismo, el uso de una interfaz gráfica hace que el proceso de aprendizaje sea más agradable y sencillo para el usuario.

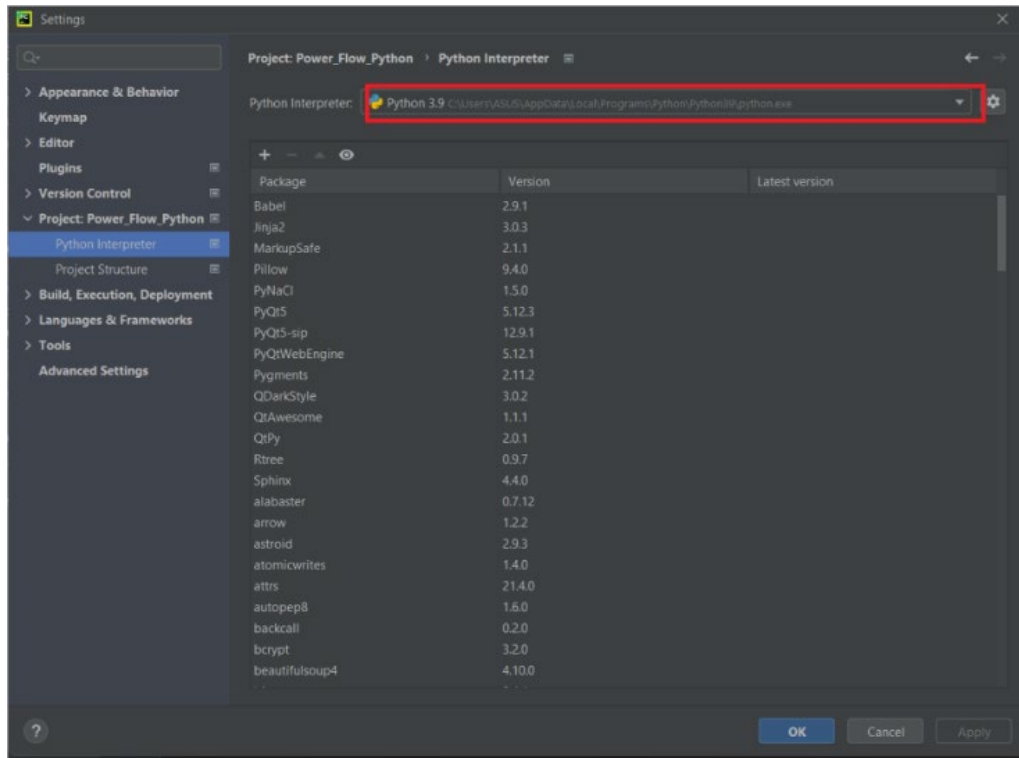
Módulo "main":

El módulo principal el cual al ejecutar el depurador dará inicio a la interfaz gráfica de la aplicación.

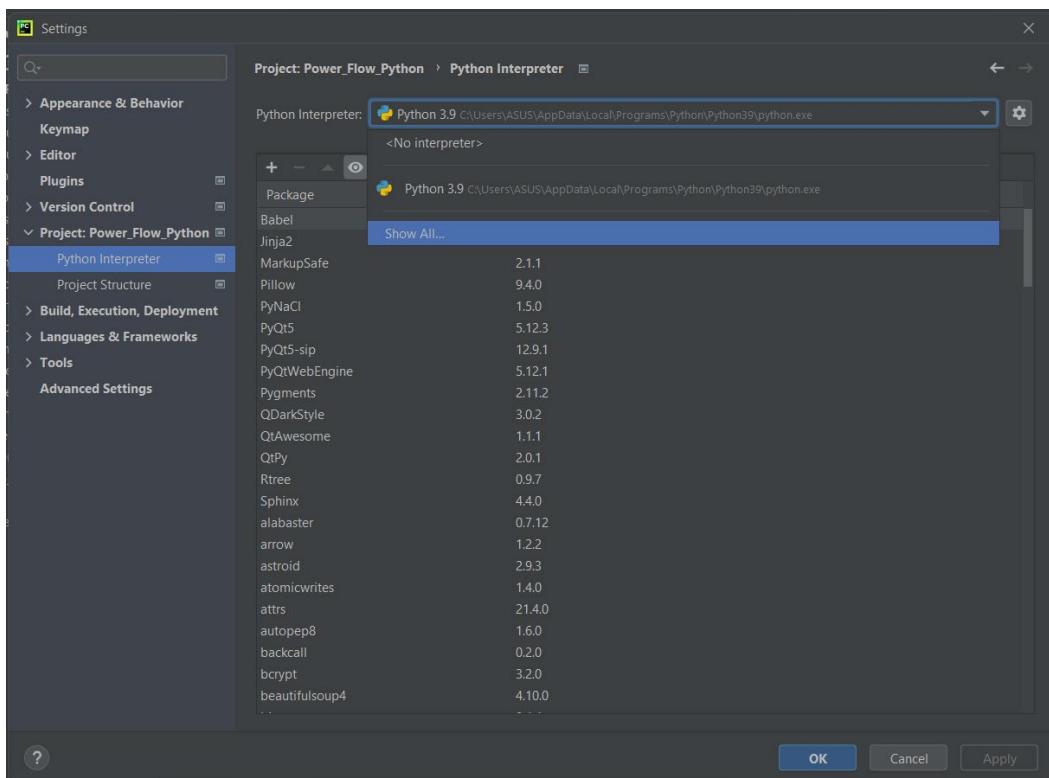
## 6. Configuración del Intérprete de Python

Una vez que se ha proporcionado una breve explicación de cada módulo, se procede a configurar PyCharm para utilizar todas las librerías instaladas en Python de la siguiente manera.

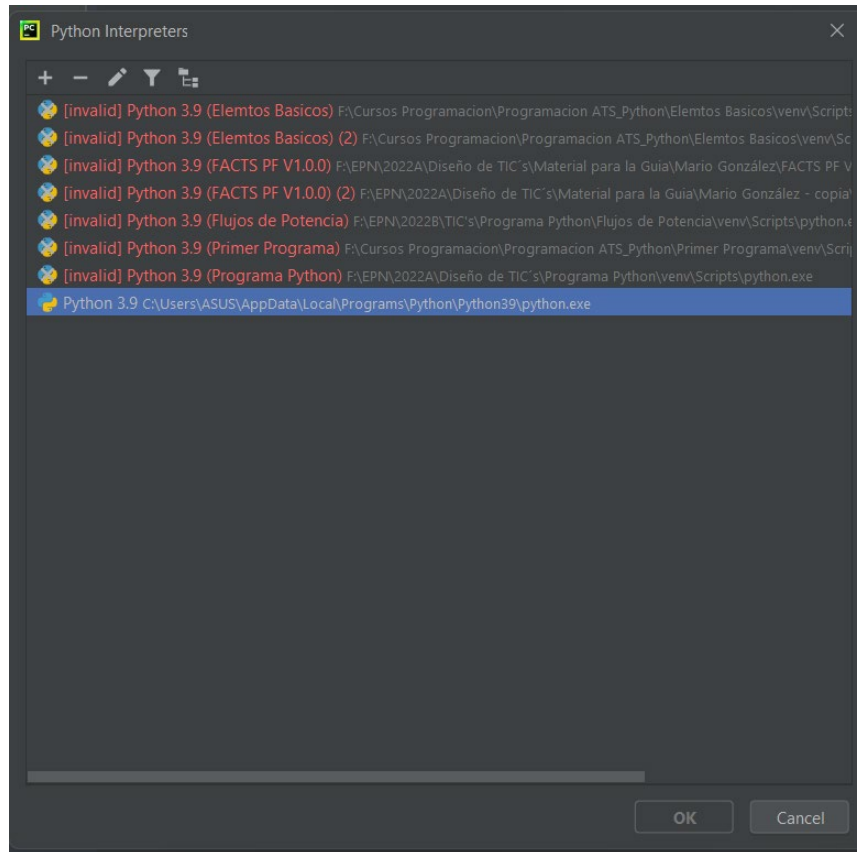
- 6.1. Seleccione "File" en la barra de menú y luego seleccione "Settings".
- 6.2. En el menú de configuración, seleccione "Project" y luego seleccione el nombre del proyecto.
- 6.3. Seleccione "Python Interpreter" en la lista de opciones.
- 6.4. Asegúrese de que la versión de Python que está utilizando este seleccionada en la parte superior de la pantalla. Si no está seleccionada, seleccione la versión correcta como se detalla a continuación (en caso de tener seleccionada ignore los siguientes pasos).



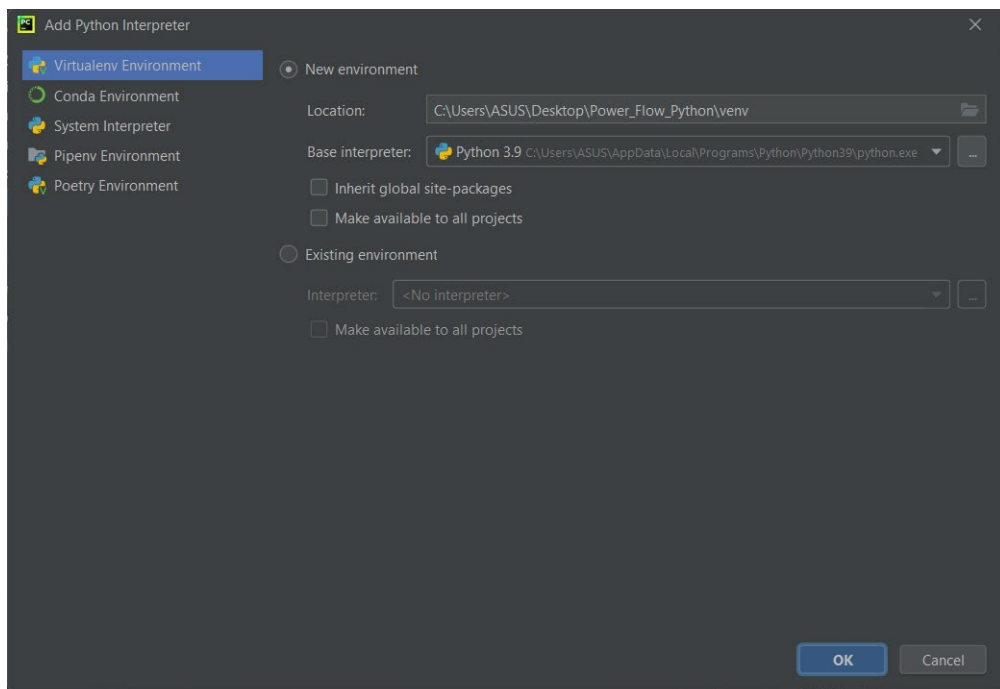
6.5. En la parte inferior de la pantalla, seleccione “Show all” para ver todos los intérpretes de Python.



6.6. Puede buscar y seleccionar el intérprete de Python con la siguiente extensión, mostrada en la imagen a continuación.



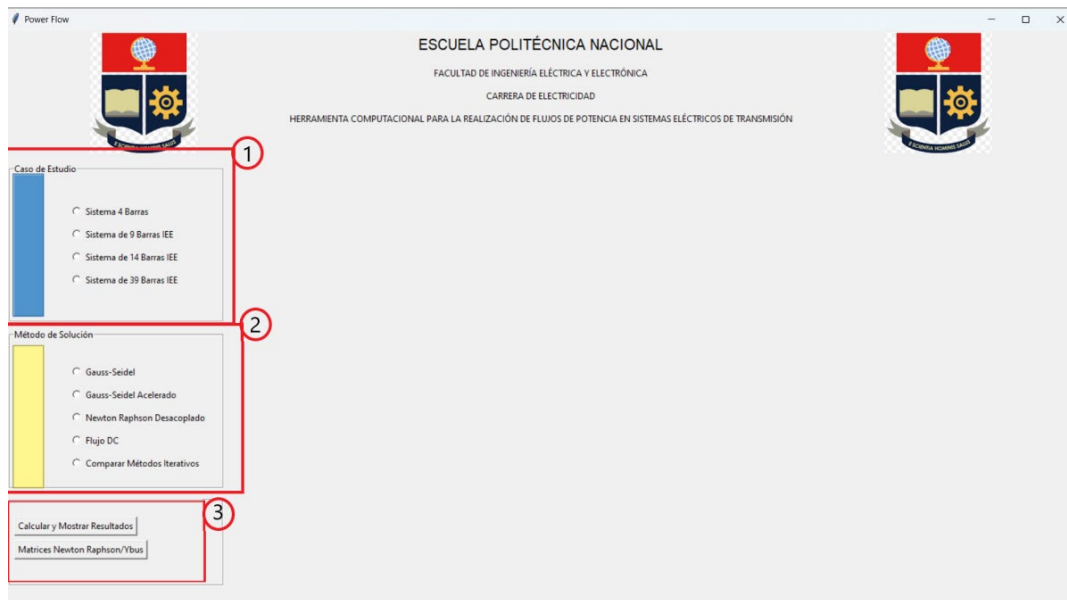
- 6.7. Si no se encuentra la extensión solicitada del paso anterior, de da clic en el boto" + " en la parte superior derecha de la pantalla.
- 6.8. A continuación, seleccione "Virtualenv Environment" y elija la ubicación del proyecto, es decir, la carpeta "vent". Luego, seleccione la ubicación que se muestra en la imagen en "Base Interpreter".



6.9. Por último, da clic en "OK" a todas las pantallas abiertas hasta este momento y Pycharm estará listo ejecutar la aplicación desarrollada para las soluciones de lujos de potencia.

## 7. Uso de la aplicación:

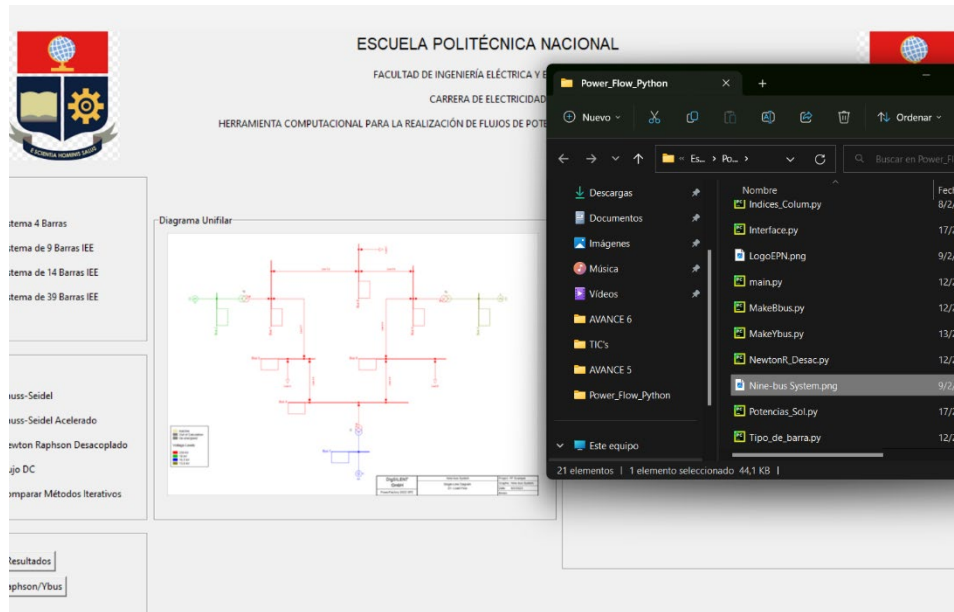
Para abrir la interfaz gráfica de la aplicación y acceder a la ventana principal, es necesario ejecutar el módulo "main".



La ventana principal se divide en tres secciones: (1) "Casos de estudio", (2) "Método de solución" y (3) "Botones de ejecución".

7.1. En la sección "Casos de estudio" de la aplicación, el usuario puede elegir entre cuatro opciones diferentes de análisis de sistemas eléctricos predefinidos. Una vez seleccionado el caso de estudio, se mostrará automáticamente el diagrama unifilar y los datos del sistema eléctrico correspondientes a ese caso en particular.

- a. Diagrama unifilar: Para ver el diagrama unifilar, simplemente haga clic en él en la aplicación. La carpeta que contiene la imagen del diagrama se abrirá automáticamente y podrá abrir la imagen con el visualizador de imágenes de su preferencia en el sistema.



- b. Datos del sistema: En la sección ubicada a la derecha del diagrama unifilar, encontrará tres botones que permiten al usuario visualizar los datos de barras, ramificaciones o generadores en tablas, según lo que necesite observar.



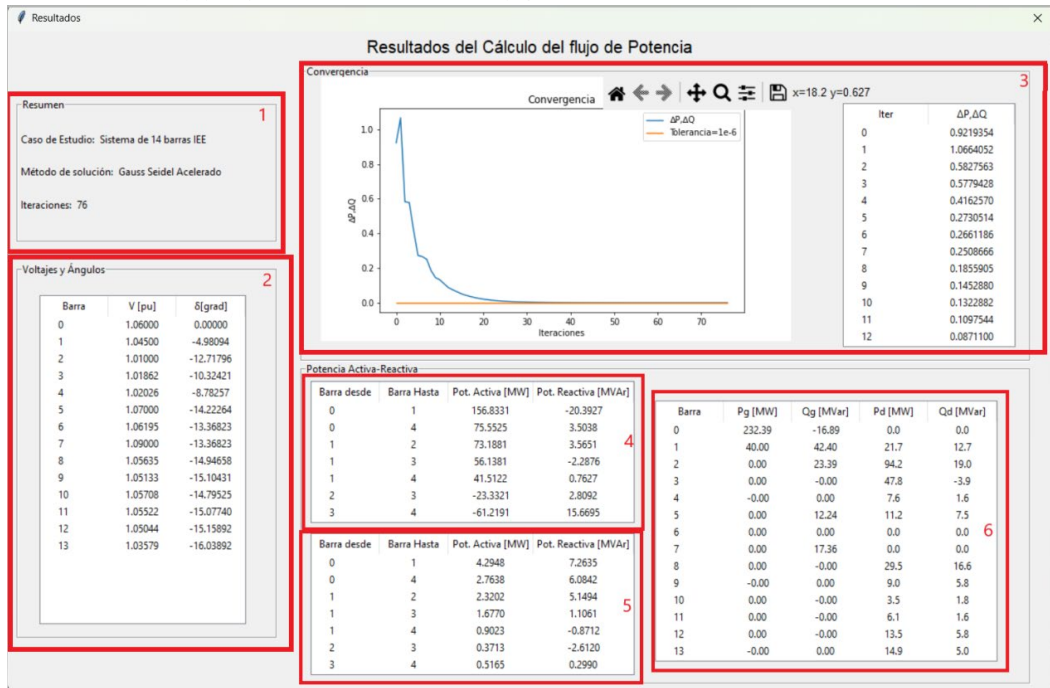
7.2. En la sección "Métodos de solución", el usuario puede elegir el método de solución para el caso de estudio preseleccionado. Las opciones incluyen el método de Gauss-Seidel, Gauss-Seidel acelerado, Newton-Raphson desacoplado y flujo linealizado (DC). La última opción en la sección, "Comparar métodos iterativos", permite ejecutar el caso de estudio con los métodos iterativos de Gauss-Seidel, Gauss-Seidel acelerado y Newton-Raphson desacoplado.

7.3. En esta sección se presentan dos botones, denominados "Calcular y mostrar resultados" y "Matrices Newton Raphson/Ybus"

- a. Calcular y mostrar resultados: Al seleccionar el caso de estudio y el método en la ventana principal y presionar el botón "Calcular y mostrar resultados", se abrirá una segunda pantalla donde se mostrarán:
- (1) un cuadro resumen.
  - (2) el módulo y ángulo de cada barra del sistema.

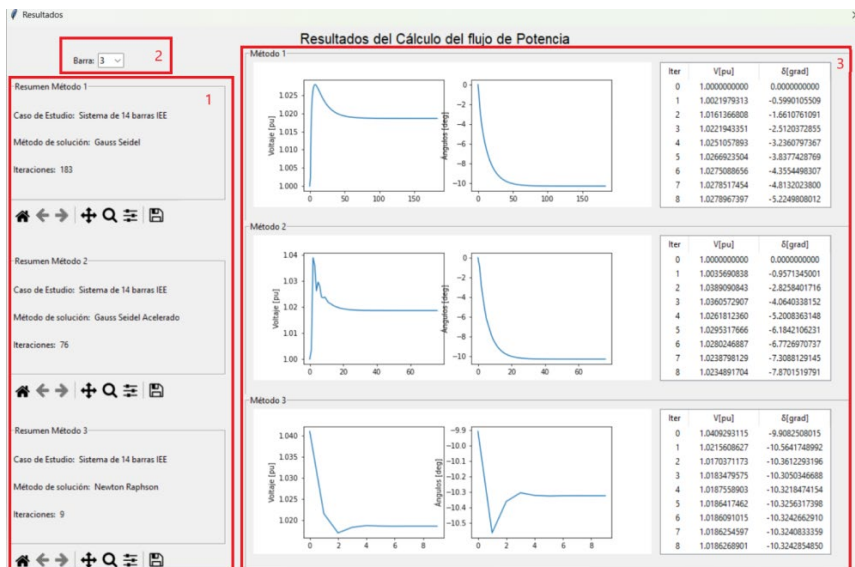


- (3) una gráfica de convergencia y tabla.
- (4) el flujo de potencia en cada rama.
- (5) las pérdidas en cada rama.
- (6) la potencia demandada y generada en cada barra.



En la sección "Método de solución" de la ventana principal, para comparar los métodos iterativos se debe seleccionar "Comparar métodos iterativos" y luego presionar el botón "Calcular y mostrar resultados". Como resultado, se mostrarán:

- (1) Tres cuadros resumen que indicarán el caso de estudio, el método de solución utilizado y el número de iteraciones realizadas.
- (2) Una lista desplegable que permitirá al usuario elegir la barra para analizar.
- (3) Tres gráficas que muestran la relación entre el voltaje y las iteraciones para cada método iterativo.



- b. Matrices Newton Raphson/Ybus: Tan solo es necesario seleccionar en la ventana principal, el caso de estudio, y presionar el botón para mostrar la ventana donde se puede observar:
- (1) la matriz de admitancias (Ybus) del sistema seleccionado previamente.
  - (2) una lista desplegable con las iteraciones en la que converge a una solución el caso preseleccionado por el método de Newton Raphson.
  - (3) seleccionada la iteración se muestra los valores de la matriz jacobiana correspondiente a la iteración seleccionada.

Matrices-Admitancia y Jacobiano

Sistema de 14 barras IEE

Jacobiano

Iteraciones del Newton Raphson: 9

Iter: 3

Matriz de admitancia

Fila	Column	G [pu]	B [pu]
0	0	6.025029	-19.447070
0	1	-4.999132	15.263087
0	2	0.000000	0.000000
0	3	0.000000	0.000000
0	4	-1.025897	4.234984
0	5	0.000000	0.000000
0	6	0.000000	0.000000
0	7	0.000000	0.000000
0	8	0.000000	0.000000
0	9	0.000000	0.000000
0	10	0.000000	0.000000
0	11	0.000000	0.000000
0	12	0.000000	0.000000
0	13	0.000000	0.000000
1	0	-4.999132	15.263087
1	1	9.521324	-30.270715
1	2	-1.135019	4.781863

J11

Fila	Column	(dP <sub>i</sub> )/(dδ <sub>i</sub> )
0	0	22.013962
0	1	-17.323525
0	2	0.000000
0	3	0.000000
0	4	-4.690438
0	5	0.000000
0	6	0.000000
0	7	0.000000
0	8	0.000000
0	9	0.000000

J12

Fila	Column	V <sub>i</sub> (dP <sub>i</sub> )/(dV <sub>i</sub> )
0	0	9.094618
0	1	-4.050159
0	2	0.000000
0	3	0.000000
0	4	-0.394668
0	5	0.000000
0	6	0.000000
0	7	0.000000
0	8	0.000000
0	9	0.000000

J21

Fila	Column	(dQ <sub>i</sub> )/(dδ <sub>i</sub> )
0	0	-4.444827
0	1	4.050159
0	2	-0.000000
0	3	-0.000000
0	4	0.394668
0	5	-0.000000
0	6	-0.000000
0	7	-0.000000
0	8	-0.000000
0	9	-0.000000

J22

Fila	Column	(dQ <sub>i</sub> )/(dV <sub>i</sub> )
0	0	21.687494
0	1	-17.323525
0	2	0.000000
0	3	0.000000
0	4	-4.690438
0	5	0.000000
0	6	0.000000
0	7	0.000000
0	8	0.000000
0	9	0.000000

Nota: Cada grafica tiene asociada la herramienta mostrada en la siguiente imagen, la cual permite configurar la marea que se visualiza estas gráficas.

