UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE TECNOLOGIA E INGENIERÍA ELÉCTRICA

DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL DE FLUJO DE CARGA IMPLEMENTADO EN UNA PLATAFORMA G.I.S.

Por:

Daniel Vitulli Ballesteros

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
Como requisito parcial para optar al título de
Ingeniero Electricista

Sartenejas, Septiembre de 2013

UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE TECNOLOGIA E INGENIERÍA ELÉCTRICA

DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL DE FLUJO DE CARGA IMPLEMENTADO EN UNA PLATAFORMA G.I.S.

Por:

Daniel Vitulli Ballesteros

Realizado con la asesoría de:

Tutor Académico: Paulo De Oliveira
Tutor Industrial: Francisco Ochoa

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
Como requisito parcial para optar al título de
Ingeniero Electricista

Sartenejas, Noviembre de 2013.

ACTA DE EVALUACIÓN

Resumen

Las compañías eléctricas presentan normalmente múltiples problemas para acceder a la información del estado del sistema eléctrico de distribución, desde el manejo de los datos debido a la incompatibilidad entre formatos de los distintos software que utilizan hasta la accesibilidad de la fuente de datos para correr Flujos de Carga que muchos casos tienen que pasar entre departamentos asilados para poder ejecutarse. Este proyecto presenta una aplicación de analisis numérico (Flujo de Carga) con soporte geográfico constituyendo una opción válida para superar los inconvenientes derivados de la asimetría en la información. Para ello se ha desarrollado una aplicación fusionando herramientas de software libre de distintos propósitos para integrar en un mismo objetivo la información básica requerida para la planeación, diseño, mantenimiento y operación de cualquier red eléctrica de distribución que cumpla con la estructura del Modelo de Datos normalizado propuesto.

De forma resumida el proyecto fue realizado bajo un lenguaje de programación de alto nivel "Python" buscando diseñar un sistema computacional de software libre que pueda obtener datos en tiempo real desde un Modelo específico actualizable, y por medio de un sistema web accesible desde cualquier computador lograr visualizar de forma geográficamente referenciada un Flujo de Carga del circuito que se desea estudiar; esto gracias a la creación de código en formato KML (Keyhole Markup Language). Otro enfoque importante que busca el proyecto es promover el uso del software libre debido a la gran cantidad de ventajas que para una empresa este representa tanto en tiempo como en inversión de capital. El proyecto se enfoca en ser una herramienta de carácter investigativo, que pueda servir como base para la posterior creación o adecuación de la empresa interesada para sus requerimientos particulares y así lograr una mejora en la visión a futuro para el crecimiento y operación del servicio eléctrico prestado a los ciudadanos, y es importante recalcar que no existe actualmente en Venezuela herramientas que analicen sistemas de dsitribución y potencias con detalles geográficos referenciados que sean de software libre, únicamente propietarios y bajo licencias costosas para la nación.

Esta tesis se ha desarrollado en el Instituto de Energia de la Universidad Simon Bolivar (INDENE) en el contexto del Proyecto MCTI-FONACIT-PEI: Desarrollo de una Herramienta Computacional en Codigo Abierto para Estudios del Sistema Electrico de Transmision y Distribucion de Venezuela. No.2012000092

DEDICATORIA

A mi querida esposa.

A mi familia y amigos.

A mi suegra hermosa allá en el cielo.

AGRADECIMIENTOS

Es difícil escribir estas palabras y expresar el agradeciomiento a todas las personas que me ayudaron para lograr este pequeño paso. Le doy gracias a mi esposa, que siempre estuvo a mi lado en esta larga faena, apoyándome, dándome fuerzas para no renunciar aunque el camino fuera muy empinado.

Agradezco a mis padres, por todo su apoyo incondicional, su comprensión y cariño. Al Señor Gino y la Señora Martha, que han sido como unos sugundos padres para mí.

A mis hermanos, que fueron columna de apoyo tanto económico como moral, muchas gracias.

A mis amigos, Kike, Dani, Jose Gregorio, Gabriela, estoy sumamente agradecido por todo lo que me han brindado alrededor de los años.

Gracias especiales a mis tutores, Francisco Ochoa y Paulo De Oliveira por su guía en la realización del proyecto, así como al ingeniero Carlos Ruiz que me llevó paso a paso por el camino correcto. Gracias de Verdad.

A una persona que tengo en el corazón y que Dios se la llevó para ser un ángel en el cielo, mi querida suegra Josefina, muchas gracias por tu apoyo, tu comprensión, tu paciencia. Mis triunfos son tuyos también.

Finalmente a todas las personas que no he nombrado y que me han ayudado.

¡Muchas Gracias!

Índice general

Índice general v
Lista de abreviaturas x
Lista de símbolos xii
Introducción……..1
Descripción de la Empresa……...5
1. Capítulo 1: Marco Teórico 11
1.1. Sistema de Información Geográfico o GIS 11
1.2. KML (Keyhole Markup Language) 13
1.3. Análisis de Flujo de Carga. 14
1.3.1. Ecuaciones. 17
1.3.2. El Método Numérico de Newton-Raphson. 18
2. Capítulo 2: Paquetes Computacionales para Estudio de Sistemas Eléctricos, Sistemas de Información Geográfica y Lenguajes de Programación 21
2.1. Definición de Software 21
2.1.1. Software privativo, Propietario o de Código Cerrado. 22
2.1.2. Software Libre 23
2.1.3. Programas de Software Libre para un Sistema de Información Geográfica y Cálculo de Flujos de Carga 25
2.1.4. Programas de Software Privativo para un Sistema de Información Geográfica y Cálculo de Flujos de Carga. 27
2.2. Elección del lenguaje de programación a utilizar en el proyecto 28
2.2.1. Python 31
3. Capítulo 3: Aplicación de Software para el análisis del flujo de carga 33
3.1. Formato del caso de estudio en PyPower 33
3.1.1. Uso de Pypower y Funciones de Importancia de la Librería. 37
3.1.2. Función ppoption 38
3.1.3. Función runpf 39
4. Capítulo 4: Aplicación del Software para Referenciar Geográficamente e Implantación Web. 42
4.1. OpenLayers. 42
4.1.1. Código Importante OpenLayers 43

4.2. Modelo-Vista-Controlador 44
4.3. Pyramid: 45
4.3.1. Uso de Pyramid para un Proyecto Base. 46
5. Capítulo 5: Metodología aplicada. 50
5.1. Reconocimiento de la Estructura de la Base de Datos 51
5.1.1. La Tabla Branch 52
5.1.2. La Tabla element_parameters: 54
5.1.3. La Tabla Node 55
5.1.4. La tabla node_capacity 56
5.1.5. Tabla circuit 56
5.1.6. Tabla circuit_factors 57
5.2. Verificación de Radialidad en las Ramas del Sistema: 58
5.3. Conversión del Tipo de Coordenadas: 60
5.4. Algoritmo de Obtención de Datos en tiempo real de la Base de Datos y Conversión de Datos para utilizar la librería Pypower y correr el Flujo de Carga. 61
5.5. Algoritmo para la creación del código en formato KML 63
5.6. Montaje del Resultado en Pyramid 64
5.6.1. Setup.py 65
5.6.2. _init_.py 65
5.6.3. Views.py 66
6. Capítulo 6: Resultados 67
6.1. Respuestas de los Circuitos 70
Conclusiones y Recomendaciones 80
Referencias bibliográficas 83
7. ANEXO A: Caso de Estudio 4 Barras. 85
8. ANEXO B: Caso de Estudio Circuito del Modelo de Datos. 86
9. ANEXO C: Generación del KML. 88
10. ANEXO D: Plantillas web y configuración archivos de Pyramid. 93
11. ANEXO E: Conversión de Datos del Modelo hacia PyPower. 104
12. ANEXO F: Kersting, Capítulo 4, Sección 4.2. 107
13. ANEXO G: Resultados de los Flujos de Carga, Circuitos 9001-9005. 111

Índice de cuadros

6.1. Cuadro 6.1. Voltajes en p.u. y ángulos en grados del Circuito 9001. 64

6.1. Cuadro 6.2. Información de Flujos de Potencia en las Ramas del Circuito 9001. 68

Índice de figuras

Figura 1.1. Ejemplo de un Sistema de Información Geográfico. 6
Figura 1.2. Ejemplo de una vista satelital ofrecida por Google Maps. 7
Figura 1.3. Datos e incógnitas según el tipo de barra 10
Figura 2.1. Paquetes de código abierto y libre para análisis de sistemas de potencia. (Milano F. , 2010) 20
Figura 2.2. Comparación de lenguajes de programación. (INDENE-USB, 2010) 24
Figura 3.1. Usó de la Librería PyPower para el caso de estudio de 9 barras. 32
Figura 4.1. Relaciones dentro del Modelo-Vista-Controlador. 39
Figura 4.2. Interfaz de proyecto inicial en Pyramid 43
Figura 5.1. Diagrama de Flujo de la Metodología General del Proyecto. 44
Figura 5.2. Tablas que conforman la Base de Datos Normalizada. 46
Figura 5.3. Campos de la Tabla “branch”. 47
Figura 5.4. Campos de la Tabla “branch” modificados. 47
Figura 5.5. Campos de la tabla “element_parameters”. 48
Figura 5.6. Campos de la tabla “node”. 49
Figura 5.7. Campos de la tabla “node”. 50
Figura 5.8. Campos de la tabla “node_capacity”. 50
Figura 5.9. Campos de la tabla “circuit”. 51
Figura 5.10. Campos de la tabla “circuit_factors”. 51
Figura 5.11. Tablas de la Base de Datos Original. 52
Figura 5.12. Diagrama de Flujo de la Verificación de Radialidad de los Circuitos 53
Figura 5.13. Diagrama de Flujo de la Conversión a Formato PyPower y del Flujo de Carga. 55
Figura 5.14. Diagrama de Flujo de la creación del script KML. 57
Figura 6.1. Interfaz Principal de Acceso al Sistema. 61
Figura 6.2. Resultados del Flujo de Carga bajo un mapa en OpenLayers del Circuito 9001. 62
Figura 6.3. Resultados del Flujo de Carga del Circuito 9001 entregados por PyPower. 63
Figura 6.4. Leyenda y Criterio del 66% utilizado para la interfaz del mapa. 63
Figura 6.5. Circuito 9001 Referenciado Geográficamente con resultados del Flujo de Carga. 64
Figura 6.6. Voltaje en p.u. del circuito 9001 con respecto a las barras del sistema. 66

Figura 6.7. Ángulos en grados del circuito 9001 con respecto a las barras del sistema. 66
Figura 6.8. Voltajes del circuito 9001 referenciados por coordenadas de longitud y latitud. 66
Figura 6.9. Información de nodos en el mapa. 68
Figura 6.10. Información de Ramas en el mapa. 69
Figura 6.11. Resumen del Circuito 9001 entregado por PyPower. 69

Lista de abreviaturas

AC	Alternate Currente "Corriente Alterna"
API	Del inglés Application Programming Interface.
ASM	Lenguaje de Ensamblado.
CADAFE	Compañía Anónima de Administración y Fomento Eléctrico
CORPOELEC	Corporación Eléctrica Nacional
DC	Direct Current "Corriente Directa"
EA	Análisis de estabilidad de pequeña señal
EDC	Electricidad de Caracas
EMT	Transitorio electromagnético
ENELVEN	C.A. Energía Eléctrica de Venezuela.
FONACIT	Fondo Nacional de Ciencia, Tecnología e Innovación.
FUNINDES	Fundación de Investigación y Desarrollo de la Universidad Simón Bolívar.
GIS	Sistema de Información Geográfico
GPL	Licencia Pública General del inglés General Public License.
GUI	Interfaz Gráfica de Usuario
GRASS	Programa de manejo de información geográfico.
HTML	HyperText Markup Language "Lenguaje de Etiquetas de hipertexto"
IEEE	Instituto de Ingenieros Eléctricos y Electricistas
KML	Del Inglés Keyhole Markup Language
InterPSS	Simulador de Sistemas de Potencia basado en tecnología de Internet
IP	Etiqueta numérica del inglés Internet Protocol.
MATLAB	Matrix Laboratory "Laboratorio de Matrices"
MCTI	Ministerio de Ciencia, Tecnología e Innovación.
MVC	Modelo-Vista-Controlador.
OOP	Object Oriented Programming
OPF	Optimal Power Flow
PDVSA	Petróleos de Venezuela S.A.

PEI	Programa de Entrenamiento para la Investigación.
PF	Power Flow "Flujo de carga"
PQ	Tipo de barra a la que se le conoce la potencia y reactiva consumida
PSAT	Power System Analysis Toolbox "Caja de análisis de sistemas potencia"
PV	Tipo de barra con módulo de voltaje y potencia generada y consumida conocidos
SEN	Sistema Eléctrico Nacional
SLACK	Barra del sistema de referencia cuyo voltaje y ángulo es conocido.
URL	Localizador de recursos uniforme del inglés uniform resource locator
SQL	Del inglés structured query language.
UTM	Sistema de Coordenadas del inglés Universal Transverse Mercator
WGS84	Sistema de Coordenadas del inglés World Geodetic System 84
XML	Del Inglés eXtensible Markup Language

Lista de símbolos

%.	Porcentaje
Hz.	Hertz
θ	Angulo del voltaje
G	Conductancia
Δ	Diferencial.Puede ser de potencia activa o reactiva
Y	elemento de la matriz de admitancias o la matriz en sí según el caso
Z.	Impedancia
kA.	Kilo Amper
kV.	Kilo Voltio
km.	Kilómetros
MW	Mega Vatio
MVA.	Mega Voltio Amper
Mvar.	Mega Voltio Amper Reactivo
Ω.	Ohmio
pu.	Por Unidad
P	Potencia activa
Pd	Potencia activa demandada
Pg	Potencia activa generada
$P_{inyectada}$	Potencia activa inyectada en una barra
Q	Potencia reactiva
Qd	Potencia reactiva demandada
Qg	Potencia reactiva generada
$Q_{inyectada}$	Potencia reactiva inyectada en una barra
B	Susceptancia

X_0	Vector de partida para los procesos iterativos.
fp	Factor de Potencia
fu	Factor de Utilización.

Introducción

Día a día las compañías eléctricas deben lidiar con problemas de diversas índoles que suelen colocar trabas para el correcto funcionamiento del servicio. Debido a esto, actividades como el diseño y la planificación del sistema para prever una correcta expansión del mismo, identificación de las zonas críticas de la red, e inclusive los tiempos de respuesta del trabajo de campo pueden verse afectadas y merman su eficiencia. Si aunamos a este hecho, que el manejo de los datos utilizado no es ni eficiente ni fácilmente actualizable, llegamos a un punto donde el buen uso de los sistemas computarizados toma una gran importancia.

Las empresas suelen adquirir licencias con programas que les facilite el control de todos estos aspectos, sin embargo, las mismas además de ser costosas, suelen tener limitantes a la hora de trabajar con las data, y normalmente su capacidad de personalización hacia los requerimientos particulares necesarios son pocos o inexistentes. En este orden de ideas, el software libre ofrece grandes beneficios, sin embargo, incluso en las versiones OpenSource de sistemas de control eléctrico el manejo de la data para exportación es muy limitado y prácticamente ningún software ofrece posibilidades de vincular el mundo eléctrico con el mundo real, siendo este el bastón de apoyo de este proyecto a realizar.

Antecedentes

Existe una cantidad considerable de software orientado al análisis de sistemas de potencia y distribución, de hecho ha habido un auge creciente dentro de este contexto, sin embargo, la mayoría pertenece al tipo de software denominado “privativo”. Empresas como PDVSA utilizan programas como ETAP que poseen gran cantidad de módulos de cálculo pero las licencias suelen costar alrededor de $22.000. Otro Software bastante utilizado por las empresas son los de la compañía DigSilent pero como en el caso del ETAP las licencias son bastante onerosas teniendo un promedio de $15.000 c/u. Entes como CADAFE utilizan un software denominado PADEE, pero está basado en Licencias AUTOCAD y de forma similar la EDC trabaja con los software SIMIP y SIGRED que son basados en licencias Bentley y ENELVEN con FeederAll de ABB.

Hay otros programas privativos que analizan sistemas de potencia y distribución, sin embargo, muchos de los lenguajes utilizados para los mismos son considerados cerrados como es el caso del lenguaje de programación "M", licenciado por MATLAB (del inglés MATrix LABoratory). En ellos están escritos programas como el desarrollado por el profesor Federico Milano PSAT (Power System Analisys Toolbox), con el cual se pueden realizar Análisis de Cortocircuitos y Flujos de Carga. También existen herramientas OpenSource para análisis eléctricos como los es INTERPSS del cual se hablará en capítulos posteriores, sin embargo, muy pocas herramienta colocan una referencia geográfica real de los nodos a ser analizados y mostrarlos en consecuencia en un mapa ya que las herramientas para mostrar Sistemas de Información Geográficos suelen ser tratados para funciones distintas.

Justificación y planteamiento del problema

Hoy día debido al tamaño y complejidad de los circuitos eléctricos es necesario el uso de algún software computacional, y con esto las empresas eléctricas se enfrentan a tener que tomar una decisión para la elección de los mismos. Normalmente las opciones de software libre son mucho más personalizables y por ende adaptables a las necesidades de las empresas pero suelen también ser incompletas o con muchos errores para datos masivos, por otro lado, las de software privativo suelen ser costosas y menos personalizables, aunque suelen poseer muchos más módulos de cálculo y también ser testeados rigurosamente. Sin embargo, no existe prácticamente ningún software que vincule el mundo eléctrico con el mundo real, uno donde se pueda plasmar visualmente la ubicación del mismo en un mapamundi, y que en consecuencia ayude a mejorar la planificación de expansión de la red a la cual se presta servicio, o ser accesible vía web desde cualquier computador de la empresa con los permisos necesarios y por ende una mayor accesibilidad a los datos y resultados. A fin de buscar solución a este inconveniente el INDENE (Instituto de Energía de Universidad Simón Bolívar) se plantea la elaboración de una herramienta bajo un lenguaje abierto (Python) y bajo un criterio de estandarización de la información en un Modelo de Datos que puede ser actualizable en tiempo real a futuro bajo un sistema SCADA (del inglés Supervisory Control And Data Acquisition) que cumpla con la estructura requerida. Mayor información del proyecto puede ser encontrado en la página web del Instituto (INDENE-USB, 2013).

En Venezuela es común ver que la empresa que brinda el servicio eléctrico presenta limitaciones grandes en el manejo de los datos lo que conlleva a una escasa actualización por parte de los mismos y una gran dificultad para el uso de nuevas herramientas de trabajo por incompatibilidad entre los formatos, así como múltiples programas de licencias libre y propietario para el manejo de funciones distintas. Con la realización de este proyecto y una posible adaptación y expansión del mismo, se podrá optimizar todos los procesos de análisis y se podrá integrar en una sola herramienta las funciones normalmente asociadas por separado a varios software incompatibles entre sí.

Este proyecto pretende mostrar las virtudes de un Lenguaje de Programación como Python en la aplicación de sistemas eléctricos, incluir y proponer el software libre para aprovechar todos los beneficios que una empresa debe tener en cuenta, y adaptar en una forma sencilla lo geográfico con lo eléctrico, fomentando un complemento al diseño, planificación y operación que debe ser aprovechado en futuros emprendimientos eléctricos que sean realizados en el país.

Objetivo General.

El objetivo general de este trabajo es la integración de diversos módulos en software libre para la ejecución de un Flujo de Carga bajo el método del Newton-Raphson y la posterior visualización de los circuitos analizados en una plataforma web de forma geográficamente referenciada bajo un patrón de arquitectura de Modelo-Vista-Controlador.

Objetivos específicos.

Algunos de los objetivos específicos de este trabajo de grado son:

- Capacitación y Adiestramiento en Python y PostgreSQL.
- Analizar los Modelos de Datos suministrado por INDENE para la identificación de los parámetros requeridos del proyecto.
- Adaptar y Relacionar los Modelos de Datos para el funcionamiento de la herramienta.
- Realizar algoritmo de búsqueda de radialidad en los Circuitos del Modelo.
- Convertir tipos de coordenadas para el uso en OpenLayers.

- Generación de Algoritmo para conversión de los datos del Modelo al formato PyPower.
- Creación de resultados en código KML.
- Capacitación en Pyramid.
- Creación de plantillas web y montaje del servidor web.

Descripción de la Empresa.

La Fundación de Investigación y Desarrollo de la Universidad Simón Bolívar (FUNINDES) fue creada en 1986 para funcionar de vínculo entre los sectores privados y públicos de Venezuela y las capacidades de desarrollo de la Universidad Simón Bolívar y como ente a logrado fomentar proyectos de desarrollo en las áreas de naturaleza científica, humanística y gerencial.

La participación activa de FUNINDES ha logrado la contratación de 2000 proyectos en los últimos 5 años, lo que representan 500.000 horas hombre que han sido aprovechadas por empresas y organizaciones. Hoy día la Fundación representa la mejor opción del empresario para atender las exigencias de la Ley Orgánica de Ciencia, Tecnología e Innovación.

Entre los objetivos de la Fundación Tenemos los siguientes (FUNINDES-USB, 2013):

1. "Evaluación, selección y control de tecnología.
2. Adaptación y asimilación de tecnologías extranjeras.
3. Sustitución de tecnología importada por otras de origen nacional.
4. Buscar, obtener y difundir información acerca de las tecnologías disponibles y de las condiciones de su comercialización y transferencia.
5. Recuperar y procurar un uso adecuado del acervo científico-tecnológico de la USB.
6. Gestionar la concepción y ejecución de planes y programas dirigidos a la actualización, el mejoramiento profesional y formación de recursos humanos.
7. Realizar labores de consultoría, asesoría, estudios e investigación aplicada en cualquier área del conocimiento que posean las distintas dependencias académicas de la USB.
8. Dictado de cursos, seminarios, talleres de perfeccionamiento, capacitación y actualización profesional en distintas áreas del conocimiento."

La estructura organizativa de la empresa viene dada por (FUNINDES-USB, 2013):

"Presidencia

Es la máxima autoridad ejecutiva y tiene a su cargo la dirección, supervisión y control de las funciones asignadas a las gerencias y demás instancias y órganos de la Fundación.

La Presidencia cuenta con cuatro unidades de Staff, las cuales se encargan de brindar apoyo a toda la Fundación en aspectos prioritarios para su funcionamiento. Estas son:

Consultoría Jurídica

Tiene como función velar por la aplicación de las normativas legales que estén relacionadas con cada uno de los actos y negocios de FUNINDES, a fin de asegurar la absoluta legalidad de la organización, en sus vinculaciones tanto internas como externas.

Coordinación de Recursos Humanos .

Ejecuta los procesos concernientes a la administración de los Recursos Humanos y asesora a gerentes y supervisores en la selección, evaluación del desempeño, capacitación y desarrollo del personal a fin de asegurar las mejores condiciones de trabajo con base a la misión, visión y valores de la Fundación. .

Oficina de Atención Ciudadana .

Está encargada de atender todas las inquietudes de quienes interactúan con la fundación, orientándolos y ayudándolos a canalizarlas y resolverlas a la brevedad posible.

Coordinación de Gestión de Calidad .

Desarrolla un sistema de gestión de calidad para asegurar y controlar los procesos y productos desarrollados por FUNINDES. .

Gerencia de Planificación y Desarrollo .

La Gerencia de Planificación y Desarrollo (GPD) tiene como misión, según los Estatutos de FUNINDES (cláusula décima primera): .

"Gestionar la vinculación entre la oferta real o potencial de las capacidades concentradas en la Universidad Simón Bolívar y la demanda de las mismas, atendiendo a las políticas y lineamientos dictados por la Universidad Simón Bolívar y dentro del alcance y objeto de la Fundación". .

Para cumplir con esta misión la Gerencia de Planificación y Desarrollo cuenta con tres coordinaciones:

Coordinación de Mercadeo y Promoción .

Esta coordinación realizar actividades pertinentes a la vinculación de la Universidad con su entorno. Para ello detecta las necesidades de los clientes, establece los alcances del servicio solicitado, ubica el equipo de trabajo más idóneo dentro de la USB, propicia convenios y contratos, coordina las relaciones cliente-profesor para esbozar la oferta de servicio, coordina la actualización de los canales de comunicación existentes e implementa nuevos instrumentos de promoción. .

Coordinación de Ofertas .

Una vez culminado el proceso de mercadeo, esta coordinación prepara el ensamblaje de las ofertas de servicio de FUNINDES (proyectos, cursos y servicios de laboratorios) y asesora a las Unidades Operativas y profesores a elaborar la propuesta que mejor satisfaga las expectativas del

cliente y produzca los mejores beneficios a los objetivos tanto de la Fundación como de la Universidad Simón Bolívar.

.

Coordinación de Enlace con Laboratorios .

Esta coordinación se encarga de articular las acciones logísticas, operativas y administrativas que facilitan la prestación de los Servicios de Laboratorio, mediante la vinculación eficiente entre la universidad, FUNINDES y el mercado, asegurando la entrega de los servicios contratados de manera oportuna y satisfactoria a los clientes.

Gerencia de Operaciones .

La Gerencia de Operaciones (GO) tiene como misión, según los Estatutos de FUNINDES (cláusula décima segunda). .

"La gestión de políticas financieras, administrativas y de control administrativo, requerida para los fines y cumplimiento de las funciones de sus funciones"

Su principal actividad es administrar eficientemente el capital humano y financiero, establecer controles correspondientes para la utilización eficiente y productiva de dichos recursos y ofrecer así un adecuado soporte a los productos de FUNINDES, asegurando la rentabilidad y satisfacción de todos sus clientes. .

La Gerencia de Operaciones se compone de cuatro coordinaciones:

Coordinación de Administración .

Esta Coordinación tiene como misión ejecutar los procesos administrativos de la Fundación, aplicando para ello las normas y procedimientos establecidos por la organización y velando por el

cumplimiento de las regulaciones gubernamentales y la legislación tributaria, a fin de asegurar el uso y aprovechamiento óptimo de los recursos.

Coordinación de Contabilidad .

Esta Coordinación tiene a su cargo la responsabilidad de crear y mantener los registros contables de las transacciones financieras de la Fundación de tal manera que se facilite la elaboración de los estados financieros, de acuerdo con las normas contables y los parámetros del sistema de información establecidos. .

Coordinación de Seguimiento Ejecutivo .

Esta Coordinación es el pilar fundamental en el control del buen desarrollo del proyecto, desde la contratación hasta la entrega de los productos previstos, asegurando para ello los requerimientos logísticos administrativos, el cumplimiento de los tiempos estipulados, el control de los ingresos y egresos de acuerdo con lo presupuestado y facilitando la relación armónica y productiva entre los profesores de la USB y los clientes.

Coordinación de Servicios Generales .

La Coordinación Servicios Generales tiene como misión primordial formular e implementar las estrategias y el sistema de información que soporten la operación de todos los procesos de FUNINDES, así como asegurar la prestación de servicios de apoyo logístico y servicios generales que satisfagan los requerimientos de clientes internos y externos.

Coordinación de Informática .

Se encarga de articular acciones que garanticen el funcionamiento de redes sistemas y equipos de

tecnología de información, con el objetivo de proveer una infraestructura tecnológica acorde con las necesidades de FUNINDES, de acuerdo a las normas y procedimientos establecidos de manera eficaz y eficiente.”

Capítulo 1: Marco Teórico

Uno de los problemas que las compañías eléctricas deben enfrentar para el análisis, planificación y desarrollo de la red, es el manejo de la data para correr Flujos de Carga. Para lograr un resultado más confiable deben tener información actualizada, estandarizada para su software particular y accesible por todo el grupo de trabajo. Sin embargo, dichas premisas no siempre suelen cumplirse y especialmente en el caso particular de Venezuela con la creación de la Corporación Eléctrica Nacional (**CORPOELEC**), donde diversas empresas se fusionaron y con ellas sus datos, la adecuación de los diversos formatos, confiabilidad de la data e implantación de la misma para manejo de Flujos de Carga se han visto dramáticamente afectados. Aunado a esto, también existe una separación importante del mundo eléctrico y el real, ya que el software utilizado comúnmente en sistemas de información geográfica son usados con otros fines distintos a ver los flujos de potencia y condiciones de la red.

La idea de este proyecto es la de unir el mundo eléctrico y el mundo geográfico, buscando obtener una aplicación donde exista un modelo de datos estandarizado y actualizable en tiempo real, de una accesibilidad mejorada vía web pudiendo ser aplicable a cualquier parte del mundo, y donde se logre la integración visual del circuito con un mapa de coordenadas.

1.1. Sistema de Información Geográfico o GIS

De sus siglas en inglés Geographic Information System, son un conjunto estructurado de unidades que contribuyen ordenadamente a manipular, analizar, capturar, almacenar y desplegar la información geográficamente referenciada y su principal objetivo es la ayudar a la resolución de problemas de gestión geográfica y planificación. Con dichos sistemas los usuarios tienen la

capacidad de generar consultas donde los resultados son mostrados de forma dinámica. Sus temas de aplicación son múltiples: Marketing, Planificación Urbana, Logística, Arqueología, Investigaciones Científicas, Análisis Estadístico de diversas índoles, hasta para evaluaciones de impactos ambientales o desastres naturales. Un ejemplo de estos sistemas puede observarse en la Figura 1.1.

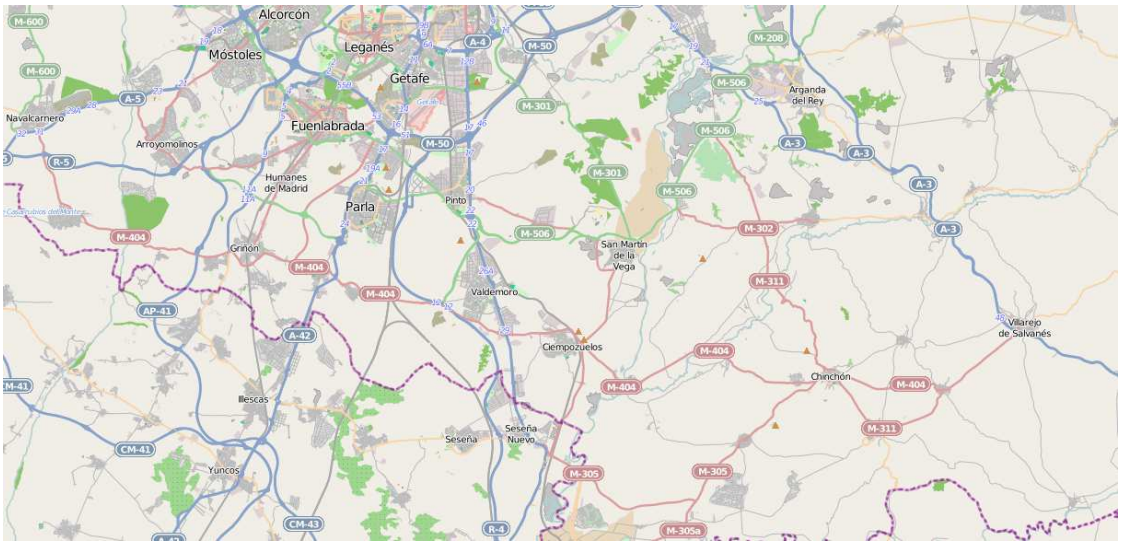


Figura 1.1. Ejemplo de un Sistema de Información Geográfico.

Debido a esto se han realizado avances en el desarrollo de software por medio de la Geomática utilizando tecnologías GIS como herramienta. La Geomática puede ser definida (Ruiz C.,2012) de la siguiente forma: “es un término científico moderno que hace referencia a un conjunto de ciencias en las cuales se integran los medios para la captura, tratamiento, análisis, interpretación, difusión y almacenamiento de información geográfica. Este término, nacido en Canadá, ya es parte de las normas de estandarización ISO”, y corresponde a la fusión del prefijo proveniente del griego “Geo” que significa “Tierra” con informática.

En los últimos años la cartografía dirigida a entornos web ha aumentado vertiginosamente con el desarrollo del internet y las redes de comunicación, haciendo accesible a usuarios una cantidad enorme de información y posibilidades de edición de la misma con aplicaciones como Google Maps u OpenStreetMap. Por lo general, se ofrecen al público imágenes aéreas satelitales o de callejones que facilitan la ubicación geográfica de calles, instituciones, hoteles o cualquier

elemento como los mostrados en la Figura 1.2. Existen muchos fuentes de datos desde donde estos software pueden cargar los raster o elementos vectoriales, y uno de ellos estandarizado y creado por Google, lleva las siglas KML (del inglés Keyhole Markup Language) el cual fue elegido para la realización del proyecto.

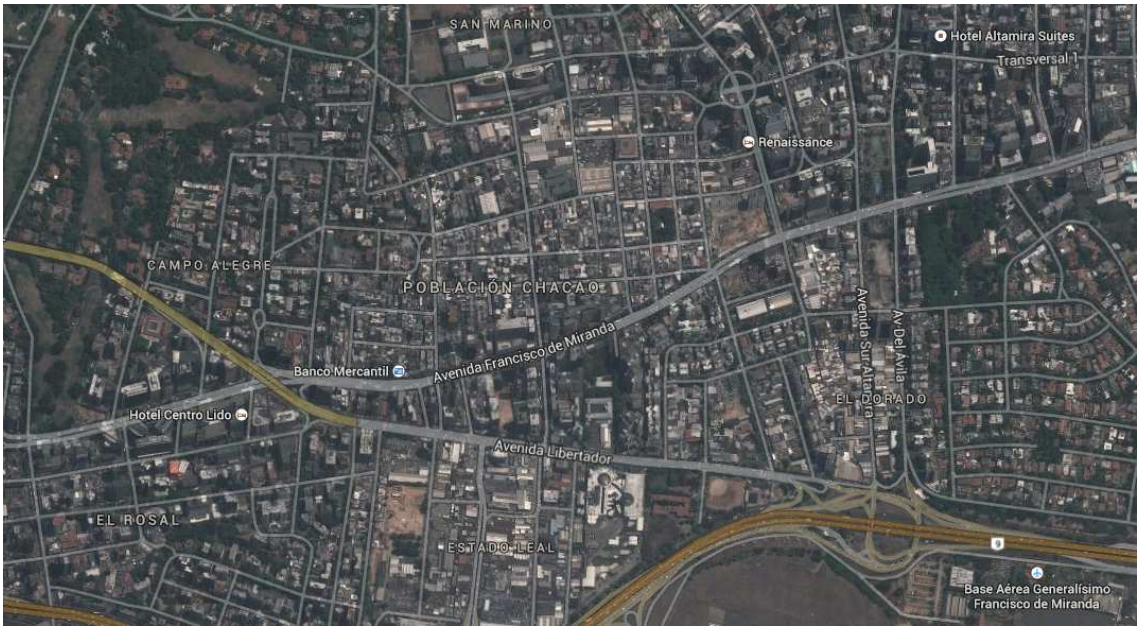


Figura 1.2. Ejemplo de una vista satelital ofrecida por Google Maps.

1.2. KML (Keyhole Markup Language)

Es un lenguaje que codifica los documentos con etiquetas anidadas, basada en XML, donde se pueden representar información geográfica tanto para 2 ó 3 Dimensiones. Los Ficheros KML muestran información de una característica de objeto, como lo puede ser una línea, un punto y en su forma más básica poseen etiquetas que identifican un nombre o título, una descripción del objeto y unas coordenadas o par de coordenadas asociadas al mismo. Una estructura de ejemplo se muestra a continuación:

```
<Placemark>
    <name>Oviedo</name>
    <description>Ciudad de Oviedo, capital de Asturias</description>
    <Point>
        <coordinates>-5.843868,43.360758,0</coordinates>
    </Point>
</Placemark>
```

La etiqueta “Placemark” hace referencia a un punto del mapa o forma, una marca de lugar o referencia. Cada etiqueta posee una entrada y una llegada, que delimita el contenido asociada a la misma y se precede con el símbolo “/”. El contenido de la etiqueta “name”,”description” y “Point”, se encuentran dentro de la información relacionada a la etiqueta “Placemark”, y a su vez estas etiquetas pueden poseer de contenido otras etiquetas anidadas. Esto trae como consecuencia una estructura de fácil lectura e interpretación para contenido web.

La etiqueta “name”, posee el título del punto, “description”, la información descriptiva relacionada al mismo y dentro de la etiqueta “Point”, se apunta otra de nombre “coordinates” que almacena la información del punto bajo el sistema de coordenadas WGS84 del que se hablará posteriormente.

Para el tratamiento de Líneas se incluye dentro del “Placemark” la etiqueta “LineString” y se incluyen las coordenadas donde se desea llevar las líneas del mapa separadas por comas, ejemplo de esto es mostrado a continuación:

```
<LineString>
<coordinates>
        -66.60291478, 10.45270181 , 0 -66.60240721 , 10.45221904 , 0
</coordinates>
</LineString>
```

Donde el valor “0” que acompaña a cada par de coordenadas indica la altura del suelo que se desea dibujar la línea.

1.3. Análisis de Flujo de Carga.

El correcto funcionamiento del sistema eléctrico es vital para que todos los elementos que hacen uso del mismo funcionen correctamente, para ello es imprescindible tener un control sobre el estado de la red, garantizar un servicio que sea confiable e ininterrumpido, donde dichos valores de estado se mantengan lo más parecidos a la referencia deseable. Los estudios de flujo de carga son de gran importancia para ese control, además si se espera una expansión del sistema eléctrico éstos nos sirven como indicadores de dónde hay que atacar los problemas, es decir, es

un arma que nos permite planear y diseñar las modificaciones al sistema (Stevenson, W., & Grainger, J., 1996).

Antes del año 1940, los sistemas eléctricos poseían pocas interconexiones, en consecuencia solían ser típicamente de topología radial y debido a esto el análisis de los mismos no necesitaban de herramientas complicadas de cálculo. Sin embargo, a medida que se fueron conociendo las ventajas de crear sistemas mallados, y las interconexiones crecían, los cálculos referidos a los sistemas en estudio crecieron en complejidad. Como golpe de suerte, este crecimiento coincidió con el advenimiento de las computadoras digitales, y en 1947 se hace mención por primera vez en el artículo "Machine computations of power network performance", AIEE Transactions, vol. 66, (Dunstan, 1947), de la relación entre una computadora y análisis de flujos de potencia. Gracias a esos avances, hoy día disponemos de documentación suficiente y potencial de hardware para atacar problemas con topologías de red complejas y de gran envergadura.

En líneas generales, el propósito principal de un flujo de carga es conseguir el estado de una red, dígase como estado, los valores de voltaje nodales con su respectivo ángulo, y en consecuencia todos los flujos de potencia en la red dado unos niveles de demanda y generación conocidos.

Para resolver un problema de flujo de carga es necesario establecer las ecuaciones de balance de potencias tanto activas como reactivas en la mayoría de los nodos que así sean posible. Esto es posible ya que la topología de la red no varía, y los flujos de potencia vienen fuertemente vinculados a los valores de generación y demanda que en cada nodo del circuito se presente. Los nodos del sistema podemos clasificarlos en alguno de los siguientes tipos (Stevenson, W., & Grainger, J., 1996):

Nodos de Carga o Barras PQ: Son aquellos nodos donde se sirve suplir una carga directamente conectada, y los valores de potencia activa (P) y potencia reactiva (Q), son conocidos y constantes para cada momento de estudio e independientes del valor de voltaje de la barra (ya que los valores de tensión normalmente son regulados por los cambiadores de toma de los transformadores cercanos a la misma).

Barras de Voltaje Controlado o Barras PV: En este tipo de barra el voltaje es controlado y por ende conocido. Esto refiere al tipo de barra donde existe generación colocada y puede existir

o no carga asociada. El voltaje en los terminales es regulado automáticamente con el RAV, y la potencia controlada a través del gobernador de velocidad.

Barra de Compensación o Barra SLACK: Este tipo de barra presenta una modelación como fuente de voltaje independiente. En los sistemas de potencia en corriente AC, se suele definir un ángulo de referencia normalmente asociado a algún generador del sistema que sea robusto y que las alteraciones que pudieran presentarse que pongan en riesgo la estabilidad afecten en poca proporción al mismo. A partir de aquí todos los ángulos del sistema se miden con respecto al de la barra de este generador en particular. Así que para este tipo de barra se conocen los datos de voltaje y ángulo, más no de potencia, incluso tenga su correspondiente gobernador de velocidad, este dato queda asignado por el resultado de un balance de potencias del sistema una vez corrido el flujo de carga.

Lo dicho anteriormente queda resumido (Delgado S. ,2013) en la Figura 1.3.

Tipo de barra	Datos	Incógnitas
Barras PQ	Potencia activa y reactiva demandada: Pd y Qd, cero si no existe ni carga ni generación.	Voltaje con modulo y ángulo: |V| y θ
Barras PV	Potencia activa generada y demandada (en caso de haber carga) y modulo del voltaje en terminales: Pg, Pd y |V|	Potencia reactiva generada y ángulo del voltaje: Qg y θ
Barras Slack	Voltaje en modulo y ángulo: |V| y θ	Flujo de potencia activa y reactiva generada: Pg y Qg

Figura 1.3. Datos e incógnitas según el tipo de barra (Delgado S., 2012)

Adicional es necesario conocer completamente la topología de la red, para el caso particular de este proyecto las líneas de transmisión son tratadas como líneas cortas y se desprecian las Admitancias Shunt del sistema ya que obviarlas comúnmente da márgenes de error aproximados a 0.0001. Los valores son representados normalmente en por unidad y la unidad para los ángulos será en grados.

1.3.1. Ecuaciones.

Inicialmente es necesario realizar un balance de potencia en todos los nodos de la red, es decir, necesitamos verificar que en un nodo si existe flujo de potencia generado (pudiera ser cero) y existe flujo de potencia demandado (pudiera ser cero), la diferencia entre ambos debe ser igual a la potencia inyectada hacia el sistema, esto debe realizarse tanto para potencia activa como reactiva, nodo por nodo. Visto en forma de ecuación podemos expresarlo de la siguiente forma:

$$\Delta P_i = P_{gi} - P_{di} - P_{inyectadai} \quad (1.1)$$

$$\Delta Q_i = Q_{gi} - Q_{di} - Q_{inyectadai} \quad (1.2)$$

En las ecuaciones anteriores se describe un balance de potencia por nodo donde el subíndice "i" indica el número de barra en estudio y P_g, P_d y $P_{inyectada}$ son la potencia activa generada, demandada e inyectada respectivamente. Para que se cumpla el balance de potencia los términos ΔP_i y ΔQ_i deben ser iguales a cero, entonces el problema se resume a resolver la expresión de las Potencias Inyectadas, dadas por las siguientes ecuaciones descritas por Milano en la sección "Classical Power Flow Equations" (Milano F. , 2010):

$$P_{inyectadai} = V_i^2 G_{ii} + \sum_{k=1, k\neq i}^{n} V_i V_k [G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik}] \quad (1.3)$$

$$Q_{inyectadai} = -V_i^2 B_{ii} + \sum_{k=1, k\neq i}^{n} V_i V_k [G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik}] \quad (1.4)$$

θ_{ik} se refiere a la diferencia de ángulo entre las barras (ik) del sistema y G_{ik} y B_{ik} la Conductancia y Susceptancia sacadas de la Matriz de Inductancias de la red. Como G_{ik} y B_{ik} son

conocidas ya que la estructura de la red es parte de los datos que se tienen del problema, quedarían como incógnitas los valores de Voltaje, Ángulo y Potencias Activas y Reactivas inyectadas por cada nodo del sistema.

1.3.2. El Método Numérico de Newton-Raphson.

Para resolver las ecuaciones de balance de potencias activa y reactiva, nos encontramos con el escenario que las expresiones resultantes son no lineales, dónde existe una dependencia entre el ángulo y el módulo del voltaje por cada nodo del sistema y que representan lo que se conoce como una ecuación transcendente (García, M., 2009) donde la solución de las mismas no pueden llegarse por métodos únicamente algebraicos, por esta razón se utiliza lo comúnmente denominado "Métodos Numéricos" que son procedimientos iterativos , donde destaca la Resolución de Newton-Raphson por la exactitud de los resultados. Dicho método con el avance de la computación y el poder de procesamiento mejorado que ha sido tendencia en estos últimos años, nos permite resolver redes eléctricas de gran envergadura, como los utilizados por las compañías eléctricas en los Sistemas Interconectados cosa que era para la década de los años 70 fuera de alcance. Según (Milano F., 2010), éste método numérico posee la ventaja que los tiempos de ejecución son menores a costa de un exponencial aumento en los recursos de procesamiento, y cómo en el caso particular de este proyecto las redes eléctricas son tomados de un Modelo de Datos donde los circuitos son dinámicos y de tamaños considerables y la Librería PyPower nos facilita la opción a usar dicho método, se ha tomado como referencia para ser utilizado.

Según Stevenson en la sección 9.3 (Stevenson, W., & Grainger, J., 1996), este método tiene como base la expansión de series de Taylor. Las series de Taylor nos indican que para una ecuación T(x) se puede expresar de la siguiente forma:

$$T(x) \cong T(x_0) + \frac{\partial T}{\partial x}\Big|_{x_0}(x - x_0) + \frac{1}{2}\frac{\partial^2 T}{\partial x^2}\Big|_{x_0}(x - x_0)^2 + \cdots \quad (1.5)$$

Para simplicidad se pueden despreciar los términos de derivada con orden superior a 1:

$$T(x) \cong T(x_0) + \frac{\partial T}{\partial x}\Big|_{x_0} (x - x_0) \qquad (1.6)$$

Si igualamos a cero la expresión anterior y despejamos la variable x, obtenemos:

$$x = x_0 - [J(x_0)]^{-1} T(x_0) \qquad (1.7)$$

En la expresión 1.7 la Matriz J, se denomina "Matriz Jacobiana" y posee las derivadas parciales de la ecuación, además es dependiente de un valor x_o que definimos para el cálculo como un valor inicial que se encuentre cercano a la variable solución. Para el cálculo de Flujos de Carga, se suele trabajar con información en por unidad, y una vez definidas las bases del sistema se suelen tener valores de voltaje cercanos a 1 y ángulos cercanos a una referencia puesta en 0 grados (que es exactamente el valor de los datos conocidos en el nodo Slack del circuito). Por ende éste es el valor por defecto utilizado en la mayoría de los casos para el vector de arranque $\bar{x}$aunque no implica que sea la solución más óptima de partida. Por otro lado, la función $T(\bar{x})$ queda definida como un vector con todas las ecuaciones de balance de potencia tanto activas como reactivas de circuito.

$$\bar{x} = \begin{bmatrix} \theta_i \\ \theta_j \\ \theta_k \\ \cdots \\ V_i \\ V_k \end{bmatrix} \qquad T(\bar{x}) = \begin{bmatrix} \Delta P_i \\ \Delta P_j \\ \Delta P_k \\ \cdots \\ \Delta Q_i \\ \Delta Q_k \end{bmatrix} \qquad (1.8)$$

El Newton Raphson como método numérico es iterativo, la idea principal es partir de un valor inicial, realizar el método y comparar el resultado final tomando en cuenta un valor de tolerancia. Luego si no es cumplida, el vector de valores iniciales de la nueva iteración será el final de la iteración anterior, así sucesivamente, hasta que se cumpla la tolerancia exigida, esto es representado de la siguiente forma:

$$|T(x)| < \delta \quad (1.9)$$

Donde δ típicamente presenta valores que rondan los $1x10^{-6}$ y $1x10^{-8}$. Para nuestro proyecto fue elegido un $\delta = 1x10^{-6}$

En la época de los años 70, la elaboración de la matriz Jacobiana (J) para redes de gran tamaño e interconexiones era considerada un gran problema, ya que la sección que más recursos computacionales de cálculo representa del método es esta. Esto es debido a que dependiendo de cada tipo de barra dado por la topología de la red se insertan ecuaciones de balance e incógnitas al tamaño de la matriz. En el caso de las barras PQ se insertan 2 ecuaciones de balance (representadas por potencia reactiva y activa inyectadas) y 2 incógnitas (representadas por el valor del módulo del Voltaje y ángulo del nodo), para el caso de barras PV es insertado 1 ecuación de balance y 1 incógnita. Por último se crea el Jacobiano por derivar las ecuaciones con respecto a todas las variables del sistema. La expresión queda reflejada en la ecuación 1.10 a continuación:

$$J(\bar{x}) = \begin{bmatrix} \frac{\partial \Delta P_i}{\partial \theta_i} & \frac{\partial \Delta P_i}{\partial \theta_j} & \frac{\partial \Delta P_i}{\partial \theta_k} & \cdots & \frac{\partial \Delta P_i}{\partial V_i} & \frac{\partial \Delta P_i}{\partial V_k} \\ \frac{\partial \Delta P_j}{\partial \theta_i} & \frac{\partial \Delta P_j}{\partial \theta_j} & \frac{\partial \Delta P_j}{\partial \theta_k} & \cdots & \frac{\partial \Delta P_j}{\partial V_i} & \frac{\partial \Delta P_j}{\partial V_k} \\ \frac{\partial \Delta P_k}{\partial \theta_i} & \frac{\partial \Delta P_k}{\partial \theta_j} & \frac{\partial \Delta P_k}{\partial \theta_k} & \cdots & \frac{\partial \Delta P_k}{\partial V_i} & \frac{\partial \Delta P_k}{\partial V_k} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \Delta Q_i}{\partial \theta_i} & \frac{\partial \Delta Q_i}{\partial \theta_j} & \frac{\partial \Delta Q_i}{\partial \theta_k} & \cdots & \frac{\partial \Delta Q_i}{\partial V_i} & \frac{\partial \Delta Q_i}{\partial V_k} \\ \frac{\partial \Delta Q_k}{\partial \theta_i} & \frac{\partial \Delta Q_k}{\partial \theta_j} & \frac{\partial \Delta Q_k}{\partial \theta_k} & \cdots & \frac{\partial \Delta Q_k}{\partial V_i} & \frac{\partial \Delta Q_k}{\partial V_k} \end{bmatrix} \qquad (1.10)$$

Capítulo 2: Paquetes Computacionales para Estudio de Sistemas Eléctricos, Sistemas de Información Geográfica y Lenguajes de Programación

El crecimiento constante de los sistemas eléctricos ha creado la necesidad de mejorar los procesos de cálculo para las empresas encargadas y así tener el mayor control posible sobre la red eléctrica, tanto para conocer el estado de la red como para una mejor percepción geográfica del mismo, en consecuencia han aparecido en el mercado una serie de software tanto de carácter libre como privado para atacar este problema. En este capítulo se mostraran herramientas computacionales dedicadas a este fin, sin embargo, es menester definir los conceptos básicos de software tanto privado como libre; así como sus ventajas, desventajas y características primordiales.

2.1. Definición de Software

Una de las definiciones más formales las ofrece la IEEE en el estándar 729: "la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo" (IEEE Std., 1993). Tomando este enunciado como base entendemos que no sólo se refiere al código fuente destinado a ejecutar un procedimiento, sino que es todo el equipamiento lógico o programática asociada a un hardware específico.

2.1.1. Software privativo, Propietario o de Código Cerrado.

Esta clase de software es caracterizado por ser cualquier programa informático donde el cliente o usuario tiene restricciones a la hora de utilizarlo, modificarlo o redistribuirlo; cuyo código fuente tiene limitaciones de acceso al mismo. El término propietario, privativo, privado, no libre, con propietario, semilibre o de código cerrado suelen usarse para referirse básicamente al mismo tipo de software.

Usar software privativo suele tener una serie de ventajas con respecto al libre, entre ellas las más notables son (Culebro, M. & Gómez, W. & Torres, S., 2006):

Control de Calidad: Es común que la compañía que desarrolla el software privado tenga un departamento que se encargue específicamente de realizar pruebas directas sobre el mismo, buscando eliminar así la mayor cantidad de bugs para el usuario final.

Recursos a la investigación y Personal Altamente Capacitado: Las compañías suelen contratar programadores de muy alto nivel así como la inversión de recursos a investigar cuál es la tendencia del uso del mismo.

Adaptación del usuario: Normalmente la curva de aprendizaje asociada al uso del producto es mucho más sencilla que la del software libre, esto está relacionado directamente con la cantidad de usuarios que lo utilizan

Software de aplicación muy particular: Existen software que son diseñados para labores muy específicas, y normalmente el usuario que lo utiliza no encuentra reemplazo en distribución de software libre.

Campo de Expansión: Hay compañías de software que ofrecen paquetes con descuento para promocionar su herramienta a Universidades o Socios Comerciales. Esto genera una amplia difusión del mismo.

Variedad de Software: En el caso particular de los sistemas operativos privativos, suelen tener una mayor cantidad de software para su plataforma que para su contraparte libre.

Entre las desventajas del software privativo se encuentran:

Código Fuente Secreto: Las compañías suelen proteger su código con recelo, y debido a ello no es posible modificarlo para buscar solucionar problemas con el mismo como resultados errados, o mal funcionamiento en general. Es necesario esperar que la propia compañía busque solucionar el problema con una actualización o parche.

Cursos de Aprendizaje Costosos: Es habitual que los cursos de aprendizaje de los mismos presenten costos elevados.

Imposibilidad de Compartir e Ilegalidad de Copias: Para realizar estas acciones es necesario adquirir las licencias exigidas por la compañía distribuidora.

2.1.2. Software Libre

Este tipo de software a diferencia del privativo, puede ser distribuido, modificado, copiado o usado por cualquier persona o ente jurídico. Es importante no confundir el software gratuito con el software libre, ya que no solamente se trata de una cuestión de precios, sino de la libertad en el uso (Stallman, 2004). Para ello se puede clasificar el software libre en cuatro grupos de libertad:

Libertad 0: Este permite usar el programa para cualquier propósito

Libertad 1: Acceso al código fuente que te permite estudiar el funcionamiento y adaptarlo a las necesidades particulares deseadas.

Libertad 2: Potestad para redistribuir copias de la herramienta

Libertad 3: Acceso al código fuente para mejorarlo y así publicarlo para el bien de toda la comunidad.

Debido a esto cuando nos referimos a software libre, debemos en lo posible evitar el uso de términos como gratuito o de regalo, ya que los mismos tergiversan el hecho de se trata de un tema de libertad de uso.

Entre las ventajas que presentan el software libre podemos indicar:

- **Bajo Costo de Adquisición y Libre Uso:** Los software Libres suelen tener entre sus características ser de libre licencia y distribución. En algunos casos la persona o

empresa puede cancelar un servicio o soporte técnico, sin embargo la herramienta no presenta costo asociado a la adquisición del mismo. Las empresas grandes o el Gobierno puede verse beneficiados por el ahorro asociado a la implantación de la herramienta a nivel macro con respecto al costo de las licencias privativas.

- **Requisitos de Hardware menores en comparación con el software de uso privado:** Ya que el software puede ser modificado, se pueden generar versiones donde los requisitos sean menores para su implementación; por ejemplo, los sistemas Linux asociados a servidores pueden ser manejados sin interfaz gráfica, por ende una reducción de requisitos de hardware asociados.

- **Adaptación del Software:** El software libre permite al usuario personalizar a su gusto y necesidad la herramienta, ya que se permite el acceso al código fuente.

- Si hablamos de las desventajas del software libre podemos referirnos a:

- **Proceso de aprendizaje más lento:** Normalmente las personas que necesitan aprender a usar software libre y ya han usado software propietario previamente, tendrán una curva de aprendizaje mayor. Tomando en cuenta las estadísticas mundiales del uso de software libre este caso es mucho más común de lo pensado.

- **El software libre no tiene garantía directa del autor:** A diferencia del software propietario donde la compañía se hace responsable de la herramienta, en las licencias libres no aplica de la misma forma.

- **GUI menos agradables:** Las interfaces gráficas de usuario y multimedia suelen ser menos compatibles y amigables al cliente.

- **El usuario debe tener nociones de programación:** La administración del sistema en muchas ocasiones recae en programar scripts para automatizar tareas.

- **La cantidad de distribuciones pueden ocasionar confusión:** Debido a la libertad que ofrece el software libre, una consecuencia directa es la cantidad de distribuciones de la misma. En ciertos casos esto puede ser visto de forma positiva, sin embargo, para muchos usuarios esto puede crear confusiones a la hora de elegir el producto deseado.

2.1.3. Programas de Software Libre para un Sistema de Información Geográfica y Cálculo de Flujos de Carga.

Como se indicó anteriormente, las empresas eléctricas se han visto obligadas para mantener un correcto control sobre el estado del sistema a utilizar cierto software que les permitan de forma eficiente correr flujos de carga, realizar estudios de cortocircuito, entre otras actividades. De igual forma, poseer un criterio de referencia geográfica de la red, mejora la percepción del problema y comunicación entre el departamento encargado del análisis y el del trabajo de campo. A continuación se hará referencia a ciertos programas de software libre que contribuyen para tal fin y una breve descripción de algunos de ellos.

Según (Milano F. , 2010) existen varios paquetes de código abierto y libre que permiten el análisis de sistemas de potencia. En la Figura 2.1 veremos una lista de ellos dónde las casillas de verificación corresponden al tipo de análisis que el mismo puede realizar, y las columnas son las iniciales del nombre en inglés de la característica del cálculo o de aspectos estéticos asociados:

- PF: Flujo de carga convencional.
- CPF: Continuación de flujo de carga y/o análisis de estabilidad de voltaje.
- OPF: Flujo de carga óptimo.
- EA: Análisis de estabilidad de pequeña señal.
- TDS: Simulaciones en el dominio del tiempo para análisis de estabilidad transitoria.
- EMT: Transitorio electromagnético.
- GUI: Interfaz de Usuario Gráfica.
- CAD: Editor de Diagrama Unifilar asistido por Ordenador.

Package	Language	PF	CPF	OPF	EA	TDS	EMT	GUI	CAD
AMES	Java	✓		✓					
EST	Matlab	✓			✓	✓			✓
InterPSS	Java	✓				✓		✓	✓
MatDyn	Matlab	✓				✓			
MatEMTP	Matlab					✓	✓	✓	✓
Matpower	Matlab	✓		✓					
ObjectStab	Modelica	✓			✓	✓		✓	✓
OpenDSS	Delphi	✓				✓		✓	✓
PAT	Matlab	✓			✓	✓			✓
PSAT	Matlab, Octave	✓	✓	✓	✓	✓		✓	✓
PST	Matlab	✓	✓		✓	✓			
Pylon	Python	✓		✓				✓	✓
UWPFLOW	C	✓	✓						
VST	Matlab	✓	✓		✓	✓		✓	

Figura 2.1. Paquetes de código abierto y libre para análisis de sistemas de potencia. (Milano F. , 2010)

InterPSS: InterPSS es un proyecto de software libre que permite realizar múltiples tareas eléctricas, todas ellas destinadas a mejorar el diseño, análisis, diagnóstico y operación de los sistemas eléctricos. Su núcleo está programado en lenguaje Java, y funciona tanto para sistemas en Microsoft Windows como en Linux.

MatPower: Es un Paquete de Matlab que permite realizar Flujos de Carga y Flujos de Carga Óptimos. Fue Desarrollado como parte de un proyecto de nombre PowerWeb. Los desarrolladores pretenden que la librería sea dirigida a Educadores e Investigadores como herramienta de simulación con opción a modificaciones para sus necesidades.

Quantum GIS: Es un Sistema de Información Geográfica de software libre Multiplataforma (Funciona bajo ambientes Windows, Linux y MacOS) que permite el manejo de formatos vectoriales, extensiones espaciales de PostgreSQL y archivos Raster. Fue programado en C++ y usa para su GUI (Interfaz Gráfica de Usuario) una biblioteca de QT. Una de las potentes funcionalidades del programa está en permitir plugins tanto de C++ como de Python.

GRASS: Como el Quantum GIS, el sistema GRASS es un software GIS, bajo licencia GPL (software libre), es multiplataforma y puede manejar tanto Raster como Vectorial. Inicialmente fue desarrollado por el Laboratorio de Investigación de Ingeniería de la Construcción del Ejército de los Estados Unidos (USA-CERL).

2.1.4. Programas de Software Privativo para un Sistema de Información Geográfica y Cálculo de Flujos de Carga.

En el ámbito de programas con licencia de propietario para el cálculo eléctrico y GIS tenemos los siguientes (Ochoa F., 2013):

"**•AGORA - Advanced Grid Observation Reliable Algorithms - http://www.elequant.com/products/agora/:** Softwareque permitea la industriade energía eléctricagestionarlas redes de energíay restaurarel poder después deapagones.

• **AMTECH Power Software - http://www.amtech-power.com/**: Software de Diseño yverificación para sistemas de altay baja tensiónde distribución.NormasIECyNECCovers, coordinación de los dispositivosde protección.

• **IPSA Power Ltd Power System Analysis Software - http://www.ipsa-power.com**: Software para eldiseño, operación yplanificaciónde la generacióneléctrica, los sistemas de transmisión y distribución.

• **Electranix Corporation - http://www.electranix.com**: PSCAD esuna plataformapara la creación desimulaciones deenergía eléctrica ysistemas electrónicos de potencia, controlesyprotecciones.

• **Fractal Power Engineering Software - http://www.fractal.hr/index_en.htm**: El desarrollode energíade ingeniería de software, flujo de cargay análisisde corto circuito, con una interfazgráfica de usuario.

• **Powerex - http://powerex.150m.com**: Softwarede distribución de energía, flujo de cargay los estudiosde corto circuito.

• **Operation Technology, Inc. - http://www.etap.com:**Los diseñadores y desarrolladoresdeETAPPowerStation, software para el análisisy diseño desistemas yPSMSde energía eléctrica, sistema de administración de energíaen tiempo real.

• **PLECS - Power Electronics Modelling - http://www.plexim.com:** Simulación decircuitos eléctricosen el entornoMATLAB/Simulink. Está especialmentediseñado parasistemas de transmisiónelectrónica de potenciay sistemas de unidades

•**Distribution Networks Real Time Restoration - http://digilander.libero.it/stockbroker/adiscon.html**: Softwarepara la restauracióndel motoren tiempo realy la reconfiguraciónde las redesde distribución eléctricaa gran escala. ”

DbMAP 3D Suite: Es una Suite Profesional de diseño, creación y publicación geográfica el cual permite visualizar vía web campos vectoriales o geometrías almacenadas en Oracle Spatial and Locator o alguna otra Base de Datos Relacional. Es multiplataforma y un software de licencia propietaria por ABACO Group.

GeoStratum: GeoStratum es un Sistema de información Geográfico que utiliza un visor cartográfico para mostrar información geofererenciada. La idea principal del producto es poder añadir capas de información propias de un tema o negocio específico como lo pueden ser infraestructuras de suministros eléctricos, planes urbanísticos, líneas de transporte, etc.

2.2. Elección del lenguaje de programación a utilizar en el proyecto

Aunque existen varios software tipo libre que pudiéramos utilizar para el análisis de sistemas de potencia, el alcance de este proyecto va dirigido a elaborar de forma muy específica y con una estructura Modelo-Vista-Controlador (que tocaremos en temas posteriores), un software que permita con la estructura de datos existente, correr flujos de Carga y Mostrarlos en la Web referenciado geográficamente. Por esta razón es necesario elegir un lenguaje de programación que permita unificar ambos conceptos. Pero la elección de dicho lenguaje posee varios puntos a tomar en cuenta, según (Silvana, 2013), tenemos los siguientes:

1. **Nivel del lenguaje:** este punto se refiere al hecho de que a medida que los lenguajes de programación se alejan del lenguaje nativo de la máquina, suelen ser más didácticos y entendibles, pero al mismo tiempo se observan otras desventajas. Mientras el lenguaje sea de más bajo nivel, más se parecerá al lenguaje nativo, como es el caso del lenguaje ensamblador o ASM (del inglés assembly language), por lo que su ejecución es más directa. En general todos los niveles de lenguaje presentan pros y contras, en el próximo punto nos referiremos a unos de ellos.

2. **Rapidez:** Se pueden considerar 3 aspectos, y se pueden clasificar de la siguiente forma:

- **Rapidez de corrida:** este punto, depende entre otras cosas a la calidad de los algoritmos y la eficiencia del código utilizado, la capacidad del hardware donde se está corriendo el programa y el nivel de lenguaje asociado al mismo.

- **Rapidez de compilación:** los programas son traducidos a un lenguaje que la máquina logra entender, el tiempo en el cual se logra esto es el que definimos como rapidez de compilación.

- **Rapidez de escritura del programa:** en este punto se tienen en cuenta dos aspectos, primero, el nivel del lenguaje, ya que existen lenguajes de programación donde es mucho más engorroso a nivel de número de líneas y abstracción de las mismas que otros, ejemplo claro de esto es el lenguaje ensamblador con respecto por ejemplo a python. Como segundo punto importante también tenemos la experiencia que el programador posea, ya que hay múltiples formas de hacer lo mismo, unas que implican más líneas de código que otras.

3. **Popularidad:** mientras más popular sea un lenguaje de programación, mayor es la posibilidad de encontrar documentación acerca de la misa, soluciones de bugs, librerías, posibilidades de volverse multiplataforma o la propia continuidad del mismo con el tiempo.

4. **Disponibilidad de librerías:** este punto es muy importante, y es uno del que basamos la elección del lenguaje usado para este proyecto. Las librerías son paquetes que poseen una función específica y normalmente buscan facilitar y aumentar la eficiencia el uso del lenguaje al no exigir programar la función buscada, sino simplemente usarla con la menor cantidad de líneas de código.

En el anexo 14 de (INDENE-USB, 2010), podemos observar un análisis profundo del tema con otros criterios adicionales, en ella, se buscaba ponderar de forma numérica diversos lenguajes de programación actuales el cual queda resumido en la Figura 2.2:

Criterio	C/C++	Java	PHP	Python	Ruby	.NET
Medición según Tiobe	5	5	4	2	0	3
Interfaces Usuario	3	5	2	5	4	5
Interoperatividad OS	2	4	3	5	5	1
Neutralidad y Estándares	5	3	2	4	2	3
Plataformas MVC	1	4	3	5	5	1
Dinamismo	2	3	4	5	5	3
Entornos de Desarrollo	5	5	3	4	4	3
Documentación/Soporte	5	5	4	5	4	5
Componentes/Escalabilidad	5	3	4	5	4	3
Métricas/Rendimiento	3	4	3	4	3	4
Puntaje total	36	41	32	44	35	32

Figura 2.2. Comparación de lenguajes de programación. (INDENE-USB, 2010)

En ella las filas significan lo siguiente:

Medición de Tiobe: índice de popularidad

Interfaces de Usuario: este índice califica la calidad de interacción con el usuario.

Interoperatividad OS: se refiere a la capacidad multiplataforma del software

Neutralidad Tecnológica y Estándares: busca medir que tan independiente es el lenguaje usado al desarrollo del proyecto de decisiones corporativas que pudieran afectarlo.

Plataformas MVC: índice que mide la posibilidad de implantar plataformas del tipo Modelo-Vista-Controlador (Punto importante para el proyecto del que se hablara en el capítulo 4).

Dinamismo: facilidad de usar el lenguaje

Entornos de Desarrollo: son herramientas para facilitar el desarrollo del programador.

Documentación y Soporte: información disponible para el programador para su aprendizaje y resolución de conflictos.

Componentes y Escalabilidad: Cantidad de Librerías y probabilidad de obsolencia del lenguaje

Métricas y Rendimientos: Mide el consumo de recursos de hardware del lenguaje y velocidad de ejecución de códigos similares entre lenguajes.

Podemos observar a Python como el lenguaje de mayor puntuación general obtenida. Este hecho y otros adicionales como disponibilidad de librerías eléctricas para el alcance de este proyecto así como paquetes de creación de archivos KML destinados a referenciar geográficamente, hace que la conclusión sea utilizar a python como lenguaje de la herramienta buscada.

2.2.1. Python

Es un lenguaje de programación libre, que permite la orientación a objetos, multiplataforma y facilita la forma de realización de los scripts ya que es multiparadigma. La licencia asociada es de Python Software Foundation License que es compatible con la

licencia pública del GNU. Su fundador Guido van Rossum, lo elaboró a finales de la década de los 80 buscando un sucesor al lenguaje de programación ABC. El mismo era capaz de manejar excepciones y para principios de los 90 ya estaban presentes las clases con herencia, los tipos modulares y manejo de funciones.

Entre los beneficios que python ofrece en comparación con otras alternativas tenemos que el código es mucho más reducido y leíble, además de tener una curva de aprendizaje bastante rápida, tiene excelente documentación, múltiples librerías, consumo de recursos reducido, estabilidad y al ser multiplataforma se adapta a los tiempos actuales donde el desarrollo de sistemas operativos y tablets tienen un auge creciente .

Capítulo 3: Aplicación de Software para el análisis del flujo de carga

Para la realización del Flujo de Carga del proyecto se aprovechó una librería de python llamada PyPower (Lincoln, Richard; Power System Engineering Research Center, 2009), dicho paquete de programación permite al usuario realizar Flujos de Carga por varios métodos de análisis numéricos tales como Newton Raphson, Gauss Seidel o Desacoplado Rápido. La idea del uso del paquete consiste en generar inicialmente lo que llaman "caso de estudio", el mismo debe contener los datos del sistema eléctrico a analizar bajo un cierto formato específico que será explicado a continuación.

Para efectos prácticos se hará la explicación con el caso más sencillo que viene incluido con PyPower, que es el caso de (4) cuatro barras y (2) dos generadores, además dicho ejemplo se encuentra referenciado en el libro de Power System Analysis (Stevenson & Grainger, 1994), si se desea obtener información al respecto del caso de prueba remitirse al Anexo A (Delgado S.,2013). Si se desea referencia del código de caso de prueba generado automáticamente tomando datos del Modelo de Base de Datos existente por favor remitirse al Anexo B, donde se coloca el caso del circuito 9001.

3.1. Formato del caso de estudio en PyPower

Toda la información del caso de estudio de PyPower se almacena en lo que comúnmente se conoce como diccionario de datos, dichos datos presentan etiquetas con información directa del circuito de la red, como puede ser datos de ramas, nodos, generadores, potencias base del sistema en MVA, costos asociados a la generación. En nuestro caso específico el modelo de datos no presentaba información de los costos asociados a la generación y los

datos de las ramas eran presentados por circuitos independientes, por dicha razón el nodo inicial al circuito (que debería coincidir con el nodo más próximo a la subestación), fue tomado como barra Slack del sistema, y no existe generación distribuida en los mismos.

A continuación mostramos la información del caso de estudio:

1. **'bus':** Este arreglo presenta la información de todas las barras asociadas al sistema. El formato que debe presentar esta sección con respecto al caso de 4 barras y 2 generadores es la mostrada a continuación:

```
ppc["bus"] = array([
                [0, 3, 50,  30.99,  0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
                [1, 1, 170, 105.35, 0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
                [2, 1, 200, 123.94, 0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
                [3, 2, 80,  49.58,  0, 0, 1, 1, 0, 230, 1, 1.1, 0.9]
                 ])
```

La explicación de las columnas (Delgado S. ,2013) se muestran a continuación y se leen de izquierda a derecha, conservando dicho orden:

- Número de la barra (entero positivo y único para cada barra).
- Tipo de barra: 1 si es PQ, 2 si es PV, 3 para la barra slack y 4 en caso de estar aislada.
- Potencia activa demandada [MW].
- Potencia reactiva demandada [MVAr].
- Conductancia shunt demandada en la barra cuando V=1.0 p.u. [MW].
- Susceptancia shunt inyectada en la barra cuando V=1.0 p.u. [MVAr].
- Número del área (entero positivo).
- Magnitud del voltaje [p.u.].
- Ángulo del voltaje [°].

- Voltaje base [kV].
- Número de la zona (entero positivo).
- Voltaje máximo permitido en la barra [p.u.].
- Voltaje mínimo permitido en la barra [p.u.].

2. **'branch':** En este arreglo se especifican las ramas y sus características, el formato para el caso de estudio quedaría de la siguiente forma:

```
ppc["branch"] = array([
      [0, 1, 0.01008, 0.0504, 0.1025, 250, 250, 250, 0, 0, 1, -360, 360],
      [0, 2, 0.00744, 0.0372, 0.0775, 250, 250, 250, 0, 0, 1, -360, 360],
      [1, 3, 0.00744, 0.0372, 0.0775, 250, 250, 250, 0, 0, 1, -360, 360],
      [2, 3, 0.01272, 0.0636, 0.1275, 250, 250, 250, 0, 0, 1, -360, 360]
   ])
```

La información de las columnas (Delgado S., 2013) leídas de igual forma de izquierda a derecha son:

- Número de la barra de salida (entero positivo).
- Número de la barra de llegada (entero positivo).
- Resistencia de la rama [p.u.].
- Reactancia de la rama [p.u.].
- Susceptancia total de la rama [p.u.].
- Límite de potencia de régimen permanente [MVA].
- Límite de potencia a corto plazo [MVA].
- Límite de potencia de emergencia [MVA].
- Relación de transformación (cero para las líneas y menor a uno (1) para los transformadores).
- Angulo de desfasaje (cero para las líneas y según el tipo de conexión para los transformadores).

- Estado de servicio: uno (1) para las ramas en servicio y cero (0) para las que se encuentran fuera de servicio.
- Ángulo mínimo de diferencia entre la fase del voltaje en la barra de salida y en la barra de destino [°].
- Ángulo máximo de diferencia entre la fase del voltaje en la barra de salida y en la barra de destino [°].

3. **'gen':** Arreglo con los datos de los generadores del sistema. El formato para el caso de estudio mostrado quedaría así:

```
ppc["gen"] = array([
    [3, 318, 0, 100, -100, 1.02, 100, 1, 318, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0,   0, 100, -100, 1,    100, 1, 0,   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        ])
```

Los correspondientes a los generadores (Delgado S., 2013) serían los siguientes: "

- Número de la barra en la que se encuentra conectado el generador (entero positivo).
- Potencia activa generada [MW].
- Potencia reactiva generada [MVAr].
- Límite máximo de potencia reactiva [MVAr].
- Límite mínimo de potencia reactiva [MVAr].
- Consigna de voltaje del generador [p.u.].
- Base de potencia aparente del generador [MVA].
- Estado de servicio: mayor a cero para generadores en servicio y menor o igual a cero para generadores fuera de servicio.
- Límite máximo de potencia activa [MW].
- Límite mínimo de potencia activa [MW].

- El resto de los datos de este arreglo son referentes a características más específicas de la máquina y sus curvas de capacidad, a las cuales no se tiene acceso, por lo que se establecen en cero para ser obviadas."

4. **'baseMVA':** Esta llave particular es un número real positivo y representa la potencia aparente base del sistema en MVA, para el caso de estudio la expresión del formato quedaría expresado así:

```
ppc["baseMVA"] = 100.0
```

Una vez que el caso de prueba es creado y guardado en un Array de la librería Numpy, es posible utilizarlo directamente para correr el flujo de carga bajo la función de la librería de PyPower llamada runpf. Cada función de la librería posee un código computacionalmente complejo cuya explicación no corresponde al alcance de este proyecto, sin embargo para una mejor comprensión del uso de la herramienta se mostraran brevemente a continuación las funciones más importantes relacionadas con esta importante librería.

3.1.1. Uso de Pypower y Funciones de Importancia de la Librería.

Para correr los flujos de carga con PyPower es únicamente necesario definir 2 funciones, uno que cargue las opciones de la corrida y la forma como se desea la salida, y otro que corre el flujo de carga (por supuesto es necesario tener el caso de estudio de forma adicional). Sin embargo, aunque para el usuario es transparente, revisando el código de dichas funciones podemos observar múltiples llamados a otras funciones de forma condicionada, tanto por los datos colocados en las opciones, como necesarias siempre en todos los procedimientos numéricos (transformación del formato de entrada en matrices

necesarias). En la figura 3.1, sacada de la documentación en la página web de PyPower (Lincoln, R., 2011), se muestra la corrida de un flujo de carga en sólo 4 líneas de código, desde In [2] a In [5], donde inicialmente se guarda en una variable la salida de una función que contiene el caso de estudio de 9 barras incluido en su API; posteriormente en la variable ppopt se guarda la salida de una función donde se especifica el tipo de algoritmo que se desea correr (en este caso particular es de Desacoplado Rápido indicado por el número 2), posterior a esto se corre el flujo de carga y se guarda el resultado en la variable r para mostrarse con una última función llamada printpf.

```
In [2]: ppc = case9()

In [3]: ppopt = ppoption(PF_ALG=2)

In [4]: r = runpf(ppc, ppopt)

In [5]: printpf(r)
```

Figura 3.1. Usó de la Librería PyPower para el caso de estudio de 9 barras.

A continuación la explicación de las funciones más importantes utilizadas para este proyecto:

3.1.2. Función ppoption.

Este es un módulo que crea un diccionario de opciones. En ella se especifican múltiples cosas, desde las características del Flujo de Carga a ser corrido, como las opciones de salida que se desean obtener.

Los parámetros del uso de esta función se definen a continuación:

- **"PF_ALG:"** Se usa para indicar el tipo de algoritmo para la resolución del flujo de carga deseado. Su valor por defecto es el método de Newton Raphson o

valor (1). También se dispone del Método de Desacoplo Rápido versión XB utilizando (2), Desacoplo Rápido versión BX para el (3) y Gauss Siedel el valor (4) respectivamente.

- **"PF_TOL:"** Con este parámetro se especifica el nivel de tolerancia para encontrar la convergencia del algoritmo. Los métodos numéricos son procesos iterativos, donde se busca que todas las variables en estudio se encuentre con una diferencia con respecto a la solución anterior dada por el valor de la tolerancia. En el caso de nuestro estudio se utilizó el valor de Tolerancia de 1e-6, es decir, 0.000001.

- **"PF_MAX_IT:"** Es el número máximo de iteraciones deseado, esto debido a que existe la posibilidad que debido a la topología de los datos del sistema, el método numérico sea incapaz de conseguir la respuesta deseada, entonces el método se vuelve un loop infinito, y nunca termina de realizar las iteraciones. Por ende es posible modificarle la cantidad máxima de veces que se usa el método. Es necesario indicar que si la cantidad de iteraciones es muy baja, hay posibilidad que no se llegue a cumplir el criterio de tolerancia al finalizar el proceso y llegar a un resultado errado y sin convergencia.

3.1.3. Función runpf

Módulo principal encargado directamente de realizar el Flujo de Carga. El mismo internamente para realizarlo funciona en 3 bloques principales, el primero que toma los datos de entrada y los reestructura en matrices de forma que puedan ser utilizados para cada método elegido. El segundo bloque hace referencia al método numérico seleccionado y realiza el proceso numérico asociado iterativo hasta conseguir o no convergencia. El resultado de la convergencia se almacena dentro del diccionario de datos generados luego del final del proceso donde se activa una bandera binaria con el dato 0 si no consigue

converger o 1 si es caso contrario. Al final, se realiza un proceso para guardar la data de la corrida del Flujo de Carga a un diccionario como se ha comentado y se hace un llamado a una función que se encarga de arreglar los resultados y mostrarlos en la salida de forma ordenada.

Entre los parámetros que recibe tenemos los siguientes:

- **Caso de estudio:** este debe contener el diccionario de datos creado con el formato adecuado previamente descrito. La librería viene incluida con varios casos de estudio.

- **Opciones vinculadas al Flujo de Carga y Salida de Datos:** Es el diccionario de datos resultado de correr la función ppoption bajo los parámetros indicados en el mismo.1

- **Nombre para archivo de salida (resultados):** Si es asignado, se genera en la misma dirección desde donde se corre el módulo un archivo con el nombre asignado. De existir el archivo, éste será eliminado y creado nuevamente. En él se genera un reporte con los resultados completos del flujo de carga, tanto niveles de voltaje en por unidad y ángulo con respecto a la barra slack, como pérdidas del sistema en MW, MVAR, Flujos de Potencia en las ramas, así como un resumen de la cantidad de Barras, Ramas, Generadores del Sistema, Barra de mayor y menor voltaje en módulo entre otros.

- **Nombre para archivo de salida (caso de estudio):** de igual forma si es incluido, en la dirección desde donde se corre el módulo se genera o sobrescribe un archivo de nombre igual al proporcionado, con extensión .py (de python) con el caso de estudio proporcionado. Esto es de gran utilidad cuando se están trabajando con toma de datos dinámicas como es el caso de nuestro proyecto y los circuitos a elegir varían en tiempo real, y se desea obtener el caso de prueba

para ser corrido en cualquier otra computadora del mundo sin necesidad a tener acceso directo a la base de datos.

Capítulo 4: Aplicación del Software para Referenciar Geográficamente e Implantación Web.

Uno de los objetivos de este proyecto va dirigido a lograr implementar una fusión entre el cálculo eléctrico y la realidad geográfica del circuito asociado, de esta manera, además de mejorar el resultado visual, si los datos topológicos y de potencia tomados del Modelo son actualizados en tiempo real por algún sistema, se tiene una localización geográfica del punto del circuito con problemas, ayudando así a la planificación de la solución, diseño de expansión del circuito, entre otros aspectos.

4.1. OpenLayers.

Una de las librerías más utilizadas para mostrar mapas dinámicos en los navegadores web más modernos es OpenLayers. Está escrita en el lenguaje Java, lo que la hace más accesible por ser estándar en la publicación de las plantillas web e implementa una API JavaScript muy similar a la que nos puede ofrecer Google Maps, pero a diferencia del mismo éste es un software gratuito que ha sido elaborado por la comunidad de código abierto. Ésta librería ha sido publicada bajo la licencia FreeBSD.

Desde su creación por MetaCarta en 2006, OpenLayers ha sido una biblioteca Java Licenciada con soporte de múltiples fuentes de datos geográficos, entre ellos acepta como datos de entrada para una capa vectorial el formato KML, que es generado como parte del código del proyecto. El script completo utilizado en las plantillas de Pyramid es algo largo para ser mostrado, por ende se dejará disponible para su análisis en el Anexo C de este mismo libro. Sin embargo a continuación se explicará partes esenciales del código en su implementación.

4.1.1. Código Importante OpenLayers

Inicialmente se define un estilo para el mapa en la plantilla:

```
#map { width: 100%;
height: 80%;
border: 1px solid black;
    }
```

Básicamente se le indica las características visuales que va a presentar el mapa. Luego se genera una variable con las opciones del comportamiento del mismo:

```
var options = {maxResolution: "auto",
minResolution: "auto",
numZoomLevels: 50,
            };
```

Como puede observarse las etiquetas involucradas se refieren a los límites de resolución y niveles de enfoque de Zoom. Una vez definido lo anterior se puede proceder a crear un nuevo mapa de tipo OpenLayers con el siguiente código donde hacemos uso de la variable "options" y el estilo "map" previamente creados:

```
map = new OpenLayers.Map('map',options);
```

Y posterior a esto se crean 2 capas vectoriales; la primera que llamamos "wms" que posee el contenido del mapa, direccionado a la base de datos de OpenStreetMaps (Mapa de código libre, accesible a modificaciones por los usuarios a nivel mundial):

```
var wms = new OpenLayers.Layer.OSM("OpenStreetMap", null, {
transitionEffect: "resize",
attribution: "&copy; <a
href='http://www.openstreetmap.org/copyright'>OpenStreetMap</a> contributors"
 })
```

Y la segunda capa que posee el contenido de los nodos y ramas con los resultados del flujo de carga cuya fuente de datos se saca del código KML generado por python (el cual será explicado en el capítulo 5) que fue denominado "sundials":

```
var sundials = new OpenLayers.Layer.Vector("KML", {
projection: map.displayProjection,
strategies: [new OpenLayers.Strategy.Fixed()],
```

```
protocol: new OpenLayers.Protocol.HTTP({
      url: "/kml/${request.matchdict.uid}",
format: new OpenLayers.Format.KML({
extractStyles: true,
extractAttributes: true
                })
            })
        });
```

Donde la parte del código que llama a la “URL” "/kml/${request.matchdict.uid}" ejecuta el script en python que genera como respuesta el código requerido en formato KML. Una vez teniendo ambas capas vectoriales sólo faltaría agregarlo al mapa previamente creado:

```
map.addLayers([wms, sundials]);
```

Dentro de la plantilla existe muchas más funciones como la que indica los eventos que se ejecutan al cargar el mapa, centrar las coordenadas a la extensión de los nodos, generar la ventana que se despliegan al hacer el evento clic dentro de los elementos del circuito (ramas, nodos), entre otros, pero se salen del alcance que se busca tener con este proyecto.

4.2. Modelo-Vista-Controlador

La elección del lenguaje de programación en python, además de lo expuesto en el capítulo 2, obedece a que en este proyecto se buscó como exigencia que la arquitectura de software fuera dirigido a un Modelo-Vista-Controlador, y en esta característica particular python cumple perfectamente ya que es uno de los lenguajes por excelencia para trabajar con dicho patrón.

A nivel informático, existen patrones o guías para la realización de los proyectos que comúnmente son llamados Arquitectura de Software, éstos permiten manejar la estructura del programa para su elaboración, el Modelo-Vista-Controlador es una de ellas. En esta se busca separar en 3 grupos bien definidos la distribución del programa. Un primer grupo donde se define un Modelo de Datos, que básicamente es la fuente proveedora de información al programa. El segundo grupo viene dado por las Vistas del programa y va

dedicado a la interfaz de usuario, el producto donde el individuo interactúa con la herramienta. Como último grupo se definen los Controladores del programa; éstos a groso modo se refieren a toda la lógica con la cual el programa funciona y al mismo tiempo servir de mediador entre las interacciones del usuario y el Modelo de Datos.

En la Figura 3.1. (Plotz, M., 2010). Se muestra la interrelación de los grupos de la arquitectura MVC.

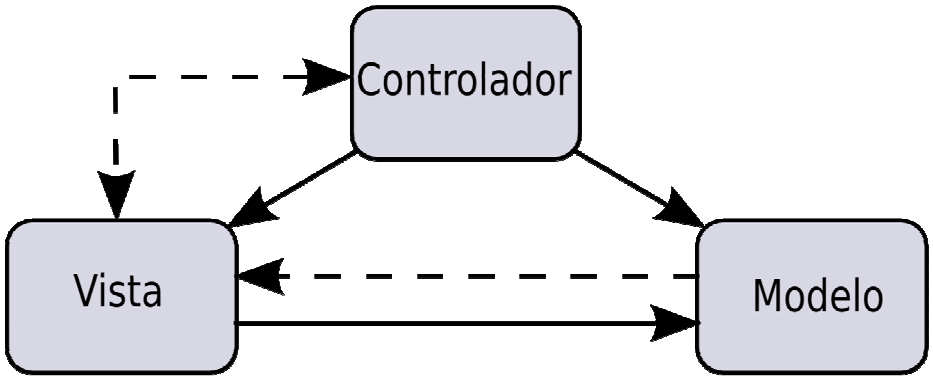


Figura 4.1. Relaciones dentro del Modelo-Vista-Controlador.

Este tipo de arquitectura tiene sus inicios en los años 70 donde Trygve Reenskaug lo presento durante su visita al Xerox Parc. La forma del patrón MVC ha cambiado con los años adaptándose a las necesidades de los clientes actuales integrándolas a aplicaciones web. Debido a esto se han creado muchos Frameworks que utilizan dicho patrón, tanto comerciales como de uso libre. En este proyecto se utiliza uno de ellos, Opensource y cuyo lenguaje de programación es Python, su nombre es Pyramid.

4.3. Pyramid:

Hemos elegido para la realización del proyecto un Framework Web de Código Libre llamado Pyramid. El mismo fue escrito en Python y está basado en Web Server Gateway Interface. Es compatible con el Modelo Vista Controlador, y para efectos de la herramienta cumple con todos los requerimientos necesarios para implementarlo a nivel productivo.

Algunos de los puntos que hacen a Pyramid tan atrayente (Ruiz C.,2013) es que además de ser ligero, es un Framework de uso general, por ende no es importante el tamaño del proyecto que se está realizando ya que el Framework puede servir como opción más que suficiente para suplir todas las necesidades del cliente. Los principios de diseño que basan al programa se pueden resumir en los siguientes:

Simplicidad: Con Pyramid se es libre de utilizar cualquier tecnología disponible, lo único que se exige son conocimientos básicos para poder utilizarlas. La herramienta se estructura de forma ordenada, para generar mayor facilidad en el aprendizaje.

Minimalismo: La herramienta es más que suficiente para satisfacer las necesidades de cualquier aplicativo web.

Documentación: Presente con información actualizada y tutoriales ejemplos para su implementación.

Velocidad: Framework de alta velocidad de respuesta, con posibilidades de manejo de plantillas diversas.

Confiabilidad y Apertura: La herramienta ha sido testeado exhaustivamente antes de cualquiera de las publicaciones de sus respectivas versiones, y es de licencia abierta permisiva a modificaciones de ser necesarias.

4.3.1. Uso de Pyramid para un Proyecto Base.

El uso de la herramienta pasa por diferentes pasos, desde la instalación de un entorno visual, hasta la configuración de los módulos y creación de las plantillas web. Se resumirán y explicarán los pasos a continuación (Ruiz C.,2013):

Inicialmente es necesario tener instalado en el servidor web una versión de Python mayor o igual a la versión 2.6. Hecho esto, como primer paso es necesario instalar "virtualenv" que nos va a facilitar la creación de entornos virtuales, y con ellos podremos probar múltiples versiones de sitios web, sin alterar las configuraciones originales del PC.

Para instalarlo, la manera más sencilla es usando un ejecutable llamado “easy_install” que se encuentra en la carpeta de ejecutables que se instalan con Python. Por medio de la consola de comandos escribimos lo siguiente:

```
easy_install virtualenv
```

Esta simple instrucción se conectará al internet y buscará la versión de virtualenv más compatible para instalarla. Acto seguido es necesario crear nuestro ambiente virtual, para hacerlo en la consola de comandos escribimos la primera línea del siguiente código y lo ejecutamos:

```
$ virtualenv --no-site-packages env
New Python executable in env/bin/Python
Installing setuptools.....done.
```

Una vez hecho esto podemos proceder a instalar el Pyramid dentro del entorno virtual, para ello con la consola de comandos nos situamos dentro de la carpeta creada previamente que llamamos env, y ejecutamos el siguiente código:

```
c:\env> Scripts\easy_install pyramid
```

Con la realización de los pasos previos ya tendremos instalado el Pyramid dentro de nuestro entorno virtual, podemos entonces crear nuestra aplicación web. Para esto necesitamos seguir otra serie de comandos, que se resumen en lo siguiente:

Creamos el proyecto:

```
> Scripts\pcreate -s starter proyecto
```

Al hacer esto se generan una serie de archivos dentro de una carpeta de nombre proyecto. Continuo a esto necesitamos correr el siguiente código donde “python.exe” se refiere al ejecutable que se generó en el entorno virtual, y “setup.py” al archivo generado después de crear el proyecto dentro de la carpeta “proyecto”.

```
>cd proyecto
> ..\Scripts\python.exe setup.py develop
```

Ya podemos correr nuestro programa con el siguiente comando.

```
..\Scripts\pserve development.ini
```

Veremos la siguiente información en pantalla:

```
Starting server in PID 16601.
serving on http://0.0.0.0:6543
```

Con esto logramos abrir nuestra aplicación en el navegador visitando el siguiente “URL”: **http://localhost:6543/**, con ello veremos una aplicación básica creada automáticamente por el asistente con su plantilla web y tendrá la siguiente apariencia:

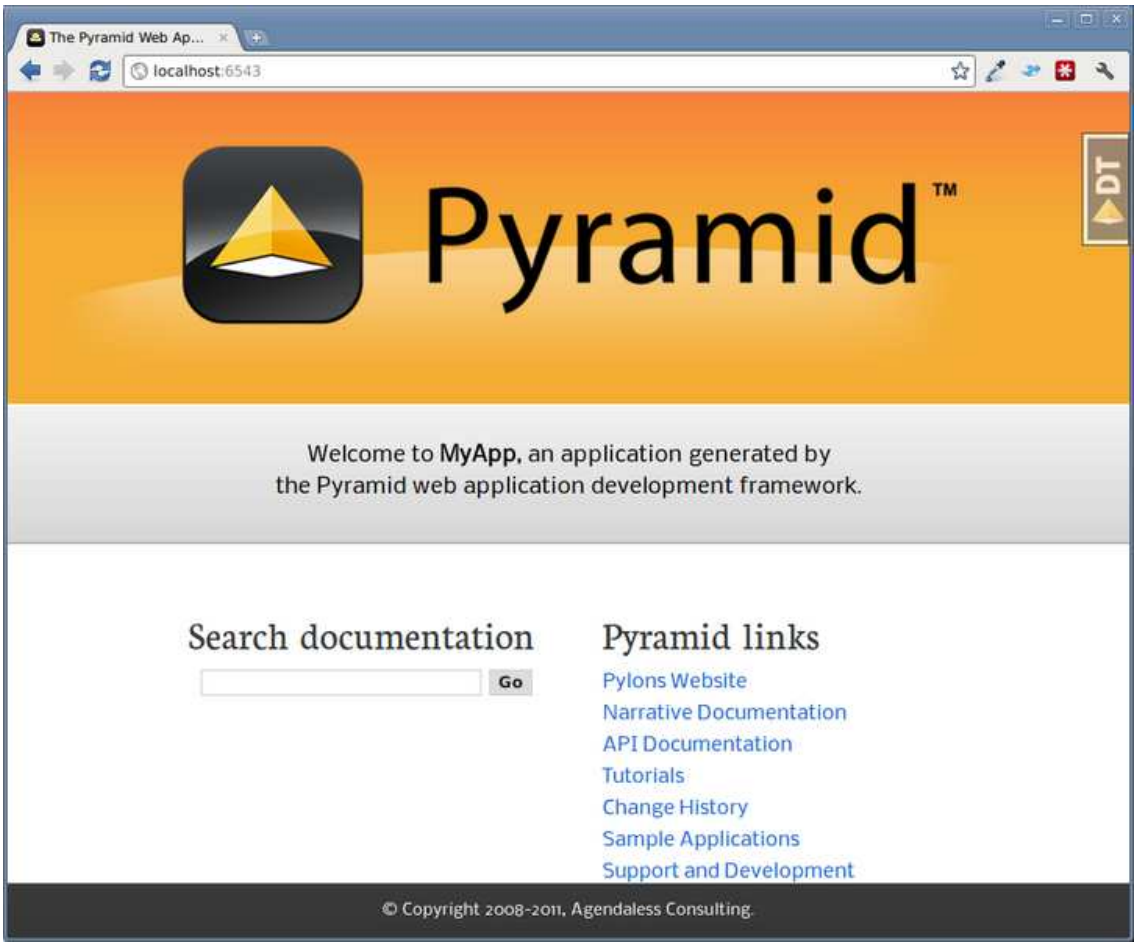


Figura 4.2. Interfaz de proyecto inicial en Pyramid

Ahora es necesario hacer varios cambios al código y crear las plantillas necesarias para implementar OpenLayers dentro de nuestra aplicación web. El código de las plantillas y las configuraciones de los archivos del sistema son muy largos y engorrosos y explicarlos a detalles está fuera del alcance esperado de este proyecto, sin embargo, se puede encontrar la información del código en el Anexo D.

Capítulo 5: Metodología aplicada.

En los capítulos anteriores se definieron las bases de las herramientas usadas para el proyecto, a continuación se presentará la metodología usada en todo el proceso de diseño, evaluación y desarrollo del mismo. Los pasos generales pueden describirse en varios procesos anidados. En la figura 5.1, se representa un Diagrama de Flujo de los mismos:

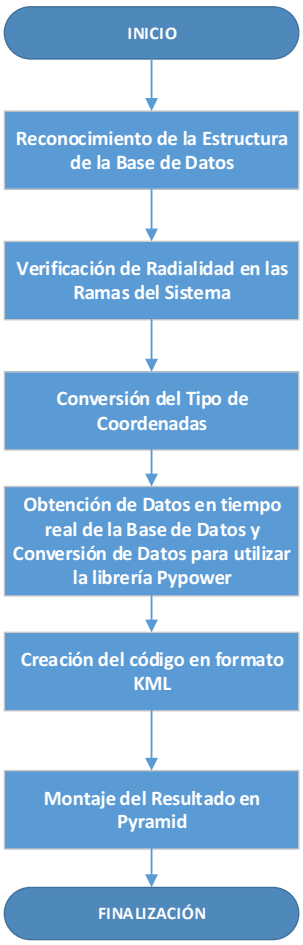


Figura 5.1. Diagrama de Flujo de la Metodología General del Proyecto.

5.1. Reconocimiento de la Estructura de la Base de Datos

Todo lo relacionado con el Modelo de Datos que se utilizó en este proyecto fue trabajado en PostgreSQL, bajo el cliente de conexión llamado pgAdmin III, específicamente la versión 1.16. Para un mejor entendimiento de esta sección se deben poseer conocimientos básicos de conceptos de base de datos como, tablas, registros, campos, campos primarios o tipos de datos comunes. En la página oficial del PostgreSQL, se encuentran muy buenos tutoriales básicos del tema (Martinez R., 2009).

Al iniciar el proyecto se facilitaron 2 Bases de Datos, la primera de donde se tomó toda la fuente de información para crear la segunda, cuyas tablas ya se encuentran normalizada; esta última es la que se utilizó para la realización final de la herramienta. Dicha fuente de datos se entregó sin ningún tipo de documentación y tuvo que ser verificada campo por campo, para la identificación de los datos necesarios para el diseño de la herramienta.

La base de datos utilizada se conforma por 13 tablas como se puede observar en la Figura 5.2. Los datos necesarios para hacer la corrida del flujo de carga pueden ser sacados únicamente de las tablas branch (ramas del sistema), node (nodos del sistema), node_capacity (relacionado a el valor de las cargas del sistema), element_parameters (tabla calibre del conductor asociado a la rama) y circuit_factors (tabla de factores del circuito como el factor de utilización y factor de potencia). El resto de las tablas van destinadas a relacionar los circuitos con una región, subestación, circuito y un grupo de tablas para agregar resultados de flujo de carga directamente a la Base de Datos que se encuentran vacías y no son utilizadas para este proyecto.

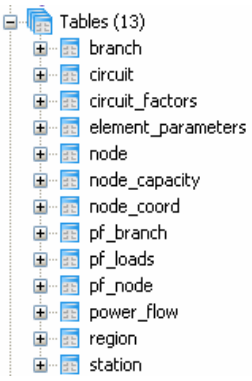


Figura 5.2. Tablas que conforman la Base de Datos Normalizada.

5.1.1. La Tabla Branch

La tabla “branch” era la encargada de indicar todas las ramas del circuito. Los campos asociados a la misma se muestran en la figura 5.3. Se puede observar un campo de identificación de rama general en la tabla “id”, dato único, aunque no utilizado para el proyecto. Seguido a esto dos campos que representan un valor entero positivo con los números de nodo inicial y final, bajo el nombre de “node0_id” y “node1_id” respectivamente. Dichos números están relacionados directamente con los “id” de la tabla “node”, que se mostrará en breve. A continuación un campo llamado “elem” que nos indica el tipo de rama presente en el circuito. Las ramas identificadas como líneas simples de distribución poseen un valor alfanumérico con letra “L”. Los datos de todas las Ramas tipo Switch son identificadas con el valor alfanumérico “S” y los datos de rama referentes a una conexión de Transformadores se identifican con un valor alfanumérico “T”. El campo “descripción” muestra un breve código descriptivo de la rama. Otro campo de interés es “params”, que posee un entero positivo que relaciona el “id” de la tabla “element_parameters”; con dicha tabla se sacan los datos del calibre del conductor, o valores de impedancia del Transformador asociados a la rama.

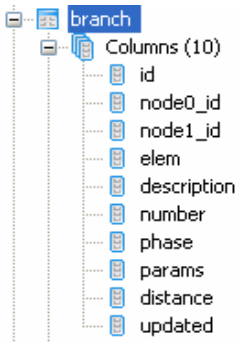


Figura 5.3. Campos de la Tabla “branch”.

El resto de los campos no son representativos para la herramienta en desarrollo. Cabe acotar que para efectos de simplicidad y eficiencia en el programa a la hora de hacer las consultas SQL (evitar consultas SQL anidadas en la medida de lo posible), se modificó la estructura de dicha tabla agregándole los siguientes campos (ver Figura 5.4), “status y status2” que se explicará en el apartado siguiente, “r1, x1,r0 y x0” que son utilizados para el cálculo eléctrico del Flujo de Carga, “amps y amps_emergency” como los datos de amperaje nominal y amperaje de emergencia respectivamente, usados para verificar los estados de las líneas con respecto al criterio del 66%, que se explicará en siguientes apartados.

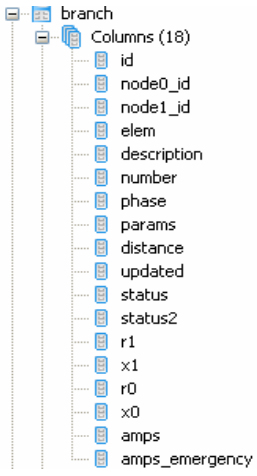


Figura 5.4. Campos de la Tabla “branch” modificados.

En el proceso de elaboración del proyecto se encontraron inconvenientes para lograr la convergencia de los flujos de carga. Después de varios análisis se concluyó que los datos de los parámetros “r1”y “x1” asociados a las ramas estaban mal indexados desde la base de datos original, por esto, se procedió a manualmente manipular esos valores y se colocaron para el caso de las líneas de distribución los valores r1= “0.3029”, x1= “0.25798” tomados del valor de secuencia positiva de la reducción de la matriz primitiva a valores de secuencia en ohmios por cada kilómetro en la sección 4.2 del libro “Distribution System Modeling and Analysis” (Kersting W., 2002), para mayor información referirse al Anexo F de este mismo libro.

5.1.2. La Tabla element_parameters:

Esta tabla tiene información, básicamente de los calibres de conductores e impedancias de transformadores, con los datos eléctricos asociados a los mismos, resistencias de secuencia positiva (negativa asumida iguales) y de secuencia cero, así como las reactancias. También muestra los valores de corriente en Amperios nominales y de emergencia. En la Figura 5.5 se muestran los campos asociados a la tabla.

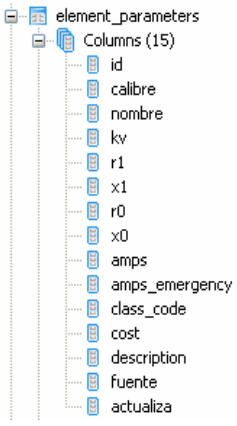


Figura 5.5. Campos de la tabla “element_parameters”.

5.1.3. La Tabla Node

Esta tabla presenta la información asociada a todos los nodos del sistema. En la Figura 5.6 puede observarse la estructura del mismo. Presenta los campos “id” identificando el nodo con un valor único entero positivo, el campo “circuit_id”, que nos relaciona el nodo con un código de circuito, “name” muestra un código que vincula el nodo con la información del mismo en la primera base de datos, y el campo “kv” indica el nivel de Tensión en Kilovoltios de dicha barra. El resto de los campos no muestran información relevante al proyecto.

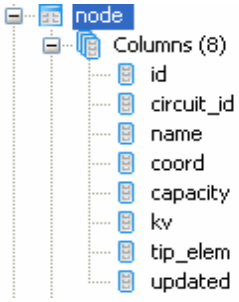


Figura 5.6. Campos de la tabla “node”.

Para Facilitar búsquedas SQL anidadas y lograr la integración de referencia geográfica, se modificó el contenido de dicha tabla. En la figura 5.7 se muestran los cambios a la estructura, y como se puede observar se agregaron los campos de “cargas_kva” asociada a la Potencia Aparente Concentrada como carga en el nodo; información que fue relacionada y sacada de la Tabla “node_capacity”, los campos “coord_x , coord._y”, muestra información de las coordenadas asociadas a los nodos. Dicha información no se encontraba disponible en el Modelo de Datos a trabajar y se tuvo que relacionar y extraer de la Base de datos Original no Normalizada. Los datos de las coordenadas se encontraban referenciados en un sistema cartesiano denominado “UTM, La Canoa, Huso 19”, los campos “latitud y longitud” que referencian una conversión de las coordenadas UTM al WGS84, que es un sistema de coordenadas geográficas universal con el que trabajan servicios como Google Maps o OpenStreetMap, y por último el campo “rotación” que es un valor que corresponde a la inclinación visual del objeto (no utilizado en este proyecto).

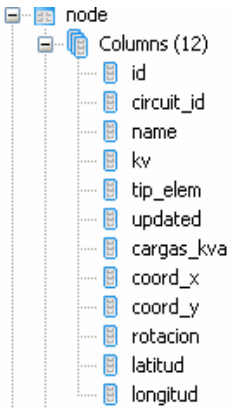


Figura 5.7. Campos de la tabla “node”.

5.1.4. La tabla node_capacity

Esta tabla presenta la información de la potencia aparente concentrada en los nodos del sistema como cargas. Normalmente es la información de la capacidad del transformador, ya que a falta de mediciones reales de la carga se suele hacer un aproximado del consumo como los KVA del Transformador multiplicado por un factor de utilización, típicamente de 0.5. La información de los campos de la tabla es mostrada en la Figura 5.8.

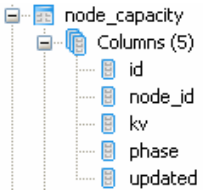


Figura 5.8. Campos de la tabla “node_capacity”.

5.1.5. Tabla circuit

La información que proporciona ésta tabla nos permite conocer el identificador de subestación relacionado con cada circuito, y el “id” del primer nodo que representa a los mismos, es decir, el nodo más cercano a la subestación que en nuestra herramienta

utilizaremos como nodo de referencia o Barra Slack. Dicha representación viene dada por los campos “station_id” y “first_node” respectivamente. En la Figura 5.9 se puede observar la estructura de la tabla:

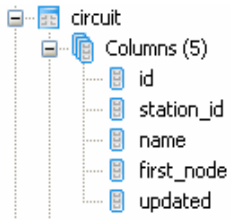


Figura 5.9. Campos de la tabla “circuit”.

5.1.6. Tabla circuit_factors

Ésta tabla relaciona un campo “id_circuito” con la tabla “circuit”, y muestra tanto los valores del factor de potencia asignados a cada circuito como el factor de utilización que va a representar al sistema, dados respectivamente por los campos “fp” y “fu”. La Figura 5.10 muestra la estructura del mismo:

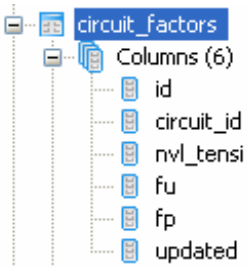


Figura 5.10. Campos de la tabla “circuit_factors”.

La base de datos original fue utilizada en varias ocasiones para este proyecto, tanto para buscar la relación de las coordenadas con las barras de los circuitos, como para lograr identificar la estructura de la base de datos normalizada. Para referencia se muestra en la Figura 5.11 las tablas que componen dicho Modelo de Datos.

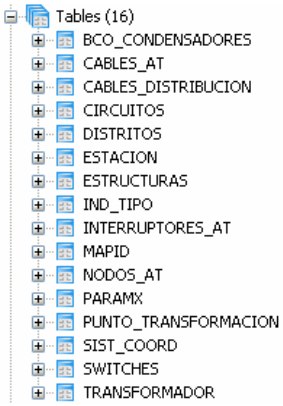


Figura 5.11. Tablas de la Base de Datos Original.

5.2. Verificación de Radialidad en las Ramas del Sistema:

El alcance del proyecto va dirigido al análisis de sistemas eléctricos de distribución, y estos son típicamente de topología radial. En este mismo orden de ideas, se realizó un algoritmo que permitía verificar con los datos de las ramas del sistema y clasificados por circuito la radialidad de los mismos. El diagrama de flujo del algoritmo es mostrado en la figura 5.12 a continuación:

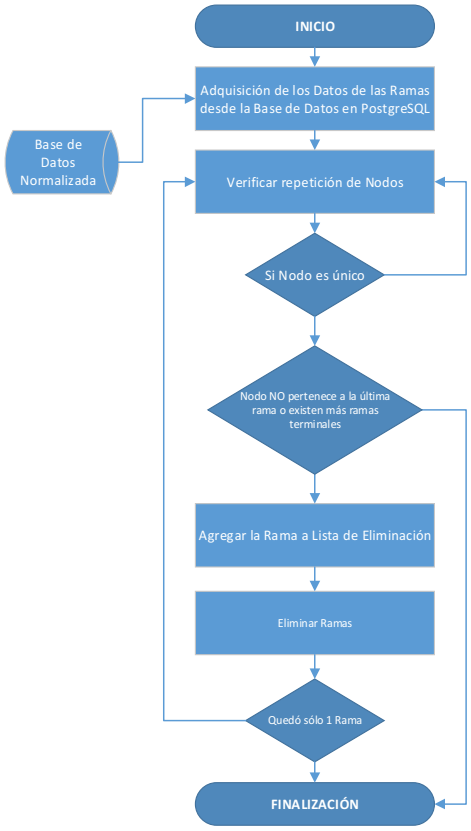


Figura 5.12. Diagrama de Flujo de la Verificación de Radialidad de los Circuitos.

Como se puede observar, inicialmente se leen los datos directamente de la base de datos y comienza una verificación de los nodos encontrados en todo el circuito. Si el nodo aparece más de 1 vez significa que no es terminal. La idea es ir eliminando progresivamente las ramas donde los nodos terminales aparecen y así lograr llegar a una única rama. Sólo se buscaban las soluciones con circuitos radiales, si más de 1 rama quedaba luego del procedimiento tendrían que ser analizados de forma individual ya que pudieran existir loops o islas de radialidad por la falla en una conexión entre ramas. La búsqueda de la mejor solución para este tipo de problema está fuera del alcance de este proyecto.

El algoritmo de este proceso fue repetido para los 854 circuitos de la base de datos y para 3 condiciones distintas en el estado de los Switch, esto debido a que al momento de la entrega de las 2 Bases de Datos, no se entregó documentación del mismo, y después de

analizar bien el Modelo de Datos, se encontró relacionar las ramas de tipo Switch con la base de datos original, para buscar algún campo que indicara el estado (si se encontraba abierto o cerrado). Gracias a esto se consiguieron 2 campos de referencia y se probaron 3 hipótesis: la primera, que todos los registros donde aparecían ramas del tipo Switch se encontraban cerradas, y las otras 2 correspondían a los estados que los campos "status y status2" de la tabla "branch" que fueron sacados de la base de datos original.

Después del análisis se encontró que bajo la primera premisa existían 129 circuitos que cumplían la radialidad del sistema, mientras que con el estado dado por el campo "status" 128 circuitos eran radiales y bajo el campo "status2" fueron sólo 51 circuitos. Por ende se asumió que cualquier Switch que se encuentra en la base de datos normalizada está en estado de conexión "cerrado".

5.3. Conversión del Tipo de Coordenadas:

Después de cotejar la Base de Datos a utilizar, no se encontraron valores de coordenadas referenciadas geográficamente en la misma vinculada a los nodos del sistema. Debido a esto hubo un proceso relacional entre la base de datos origen y la base de datos normalizada para conseguir dichos valores. Esto fue realizado con un algoritmo con varias búsquedas SQL incluidas. De aproximadamente 220.000 nodos presentes, se logró conseguir las coordenadas de 180.000 nodos, sin embargo dichas coordenadas se encontraban referenciadas en un sistema cartesiano de siglas UTM (del inglés Universal Transverse Mercator), más específicamente UTM, La Canoa, Huso 19, que representó una referencia geográfica para Venezuela hace unos años. Dicho sistema de coordenadas es incompatible con el utilizado por OpenStreetMaps con OpenLayers, que fue la librería elegida para el proyecto, por ende se tuvo que hacer una conversión de los 180.000 pares de coordenadas hacia el sistema llamado WGS84 (siglas del inglés World Geodetic System 84), usado comúnmente por Google Maps y OpenStreetMaps y hoy día es estándar mundial.

Para dicho Fin se utilizó un Sistema GIS de código abierto llamado Quantum GIS, que es multiplataforma y fue referido en el capítulo 2. Una vez obtenida la conversión, se

realizó un archivo de inserción de instrucciones SQL por cada nodo del sistema hacia la tabla “branch” a los campos “latitud” y “longitud” respectivamente.

5.4. Algoritmo de Obtención de Datos en tiempo real de la Base de Datos y Conversión de Datos para utilizar la librería Pypower y correr el Flujo de Carga.

Una vez identificados los datos necesarios para la realización del flujo de carga, se realizó un algoritmo que ingresaba a la fuente de datos, los tomaba y guardaba en listas de Python para posteriormente generar un elemento tipo Array con el formato específico que utiliza la librería de Pypower para poder funcionar. El diagrama de flujo de dicho algoritmo es mostrado en la Figura 5.13.

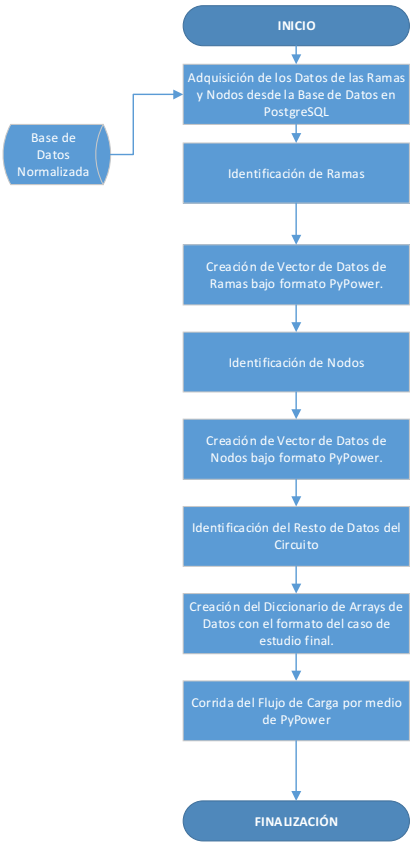


Figura 5.13. Diagrama de Flujo de la Conversión a Formato PyPower y del Flujo de Carga.

Inicialmente se hicieron las búsquedas de todos los datos requeridos (Número de Circuito, Factor de Potencia y Utilización Asociado al Circuito, Nivel de Tensión Base, Información de los nodos asociados al circuito en estudio, Ramas del circuito excluyendo valores incoherentes asociados a problemas con la información del modelo de datos), se analizaron y se organizó en vectores de nodos y ramas con el formato indicado en el capítulo 3. El código de dicha conversión es mostrado en el Anexo E debido a la extensión del mismo.

Una vez que se tiene a disponibilidad los datos en el formato de PyPower, se corre el Flujo de Carga respectivo y los datos de salida son almacenados en un diccionario de datos que tiene la particularidad de ser diccionario de diccionarios, es decir, como parte de los datos almacenados bajo el llamado a una etiqueta de datos del diccionario, se accedía a otro de forma anidada. Bajo este contexto hubo un análisis de la información de salida completa, dando como resultado las siguientes rutas de información (Bajo la premisa de que el diccionario inicial es denominado con la letra “r”):

Datos de Nodos:	r[0]["order"]["int"]["bus"]
Datos de Ramas:	r[0]["order"]["int"]["branch"]
Datos de Generadores:	r[0]["order"]["int"]["gen"]
Datos de Indexación Original de Nodos:	r[0]["order"]["bus"]["i2e"]

Toda la información de salida del flujo de carga están indicadas en Matrices de información direccionados en esas rutas de memoria. Los resultados son mostrados con una numeración de nodos distinta a la sacada del Modelo de Datos, esto por ser parte de una reordenación interna dentro del algoritmo de la librería de PyPower. Sin embargo el programa conserva la información de indexado original con la cual se puede revertir el proceso y mostrar la información conservada.

5.5. Algoritmo para la creación del código en formato KML

Una vez corrido el flujo de carga y generado los resultados es necesario crear un tipo de código especial denominado KML para lograr una capa vectorial de datos con los resultados y posteriormente utilizarlos para su visualización en el mapa. El diagrama de Flujo de la sección del código para dicho fin se observa en la figura 5.14.

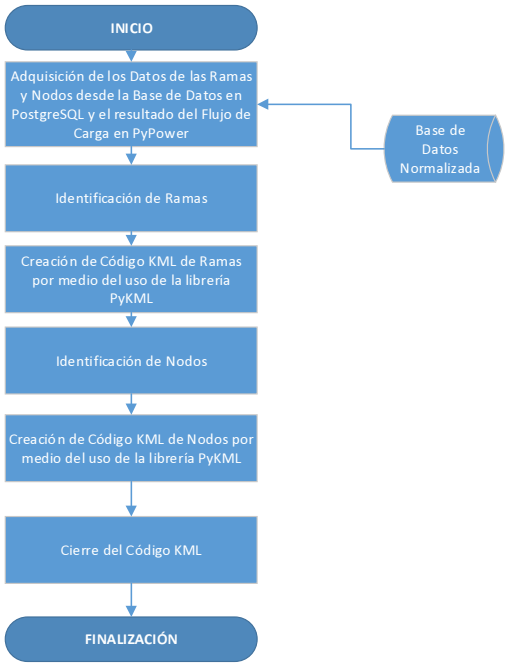


Figura 5.14. Diagrama de Flujo de la creación del script KML.

Para el desarrollo del código se utilizó una librería de Python llamada PyKML , con el que se puede crear, manipular y verificar código KML (Tyler E., 2011). En ella se escribe el código que se desea generar como funciones anidadas. Un ejemplo sencillo de su uso se muestra a continuación:

```
pm1 = KML.Placemark(
KML.name("Hello World!"),
KML.Point(
KML.coordinates("-64.5253,18.4607")
        )
```

)

En el código anterior se declara una variable con la información de una etiqueta "Placemark" y su contenido, pero se realiza declarando una función KML.Placemark cuyo datos de entradas son múltiples funciones anidadas (KML.name, KML.Point). Cada vez que se declara una función se busca que al final se genere un código dentro de la estructura de etiquetas al estilo KML.

En el caso específico del proyecto, se generó un documento KML con 2 estilos diferentes, cada uno con sus características y direccionado hacia una referencia externa con la imagen del nodo o barra, sea que tiene presente una carga o es sólo un nodo de paso. Esto puede ser observado en el siguiente script:

doc = KML.kml(KML.Document(KML.Name("Flujo de Carga"),KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("https://dl.dropboxusercontent.com/u/43356135/pin4.png")),),id="linea",)),KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("https://dl.dropboxusercontent.com/u/43356135/pin5.png")),),id="carga",))

La función KML.Name coloca el nombre del archivo, KML.Style, genera la etiqueta de estilos necesaria para personalizar los puntos a mostrar, y KML.href indica la dirección de donde va a ser tomada la ruta del ícono de la imagen.

Continuo a esto, se realizaron ciclos condicionados donde se clasificaban los nodos (con carga o sin carga) y las ramas (Líneas de Transmisión, Switchs o Transformadores) y se agregaban al código almacenado en "doc" previamente declarado. El script de este proceso es mostrado en también en el Anexo C.

5.6. Montaje del Resultado en Pyramid

Este es el último proceso del proyecto, en el mismo se busca montar los resultados obtenidos del flujo de carga y la generación del archivo KML a una página web, donde se muestre un mapa con una capa vectorial de información referenciada geográficamente de los nodos y las ramas del sistema. Para ello se utilizó una librería de java llamada

OpenLayers tratada en el capítulo 4 para cargar y visualizar capas vectoriales de mapas en la web.

Una vez seguidos los pasos mostrados en el capítulo 4, se elaboraron 2 páginas web cuyos códigos son mostrados en el Anexo D; la primera muestra un cuadro combinado donde se elige el circuito de estudio y al hacerlo, se direcciona a un URL con el valor del circuito que ejecuta el código Python para el análisis. Éste script toma los datos en tiempo real del circuito y devuelven el KML necesario y el informe del flujo de carga dado por el PyPower para mostrar una segunda página, y cargar con OpenLayers en el mapa dicho código.

Fue necesario modificar los siguientes archivos de la instalación del proyecto de Pyramid:

5.6.1. Setup.py

La sección de los requerimientos fue modificada para incluir diversas librerías y el código agregado es el siguiente:

requires = ['pyramid', 'pyramid_debugtoolbar', 'waitress', 'pypower', 'lxml', 'pykml',]

5.6.2. _init_.py

Dentro de “main” fueron agregadas las siguientes 3 rutas, las mismas ejecutan código Python y muestran en su respectivo caso las páginas web del proyecto:

config.add_route('circuito', '/circuito/{uid}')

config.add_route('resultado', '/resultado/{uid}')

config.add_route('kml', '/kml/{uid}')

La variable del circuito en estudio fue almacenada por el indicador {uid}, y el combo desplegable donde se selecciona el circuito es el encargado de entregar el valor al navegador

5.6.3. Views.py

En este archivo fueron incluidas las vistas que toman el valor del circuito y ejecutan el código Python completo. Adicional en ellas se vinculan las plantillas web que van a ser mostradas. El código de las mismas está disponible en el Anexo D por simplicidad.

Adicional todos los archivos de estilos para las páginas web e imágenes estáticas fueron almacenados dentro de la carpeta denominada “static” del proyecto.

Capítulo 6: Resultados

La fase final del proyecto incluía el montaje vía web de los diversos Flujos de Carga seleccionados. Una de las dificultades encontradas era que los circuitos que se disponían en la Base de Datos tenían errores en las coordenadas, ya que habían desplazamientos muy probablemente asociados a la conversión del tipo UTM Huso 19 hacia WGS84 ya discutido en el capítulo anterior, o es posible que la información original sea incorrecta. Por este motivo se seleccionó cercana a la ubicación de la subestación "Los Ángeles, Circuito 17" y se crearon 5 circuitos bajo coordenadas controladas y cuyas cargas fueron manipuladas para conseguir el resultado visual esperado del estado del sistema.

Se generaron 2 plantillas web para utilizar el sistema. Para obtener a la interfaz creada es necesario primero iniciar el servidor web de Pyramid, como se mostró en el capítulo 4. Luego por medio de la URL "http://localhost:6543/" si se está trabajando en la misma máquina donde está instalado Pyramid o sustituyendo "localhost" por la dirección IP o dominio de referencia indicado se desplegará la pantalla mostrada en la Figura 6.1 :

Figura 6.1. Interfaz Principal de Acceso al Sistema.

En ella se debe seleccionar tanto el Tipo de Análisis requerido que en este caso únicamente se encuentra disponible el Flujo de Carga Newthon Raphson y posteriormente la subestación y el circuito en análisis deseado mediante el cuadro combinado haciendo un clic en el mismo. Al hacer esto, se direcciona a un URL específico que ejecuta los procesos de Python que toman los datos en tiempo real desde la Base de Datos y corre un Flujo de Carga por el método de Newton-Raphson para luego generar el código KML que será mostrado en un mapa de OpenLayers de una segunda plantilla web. En la Figura 6.2 se puede observar lo dicho anteriormente del circuito 90001:

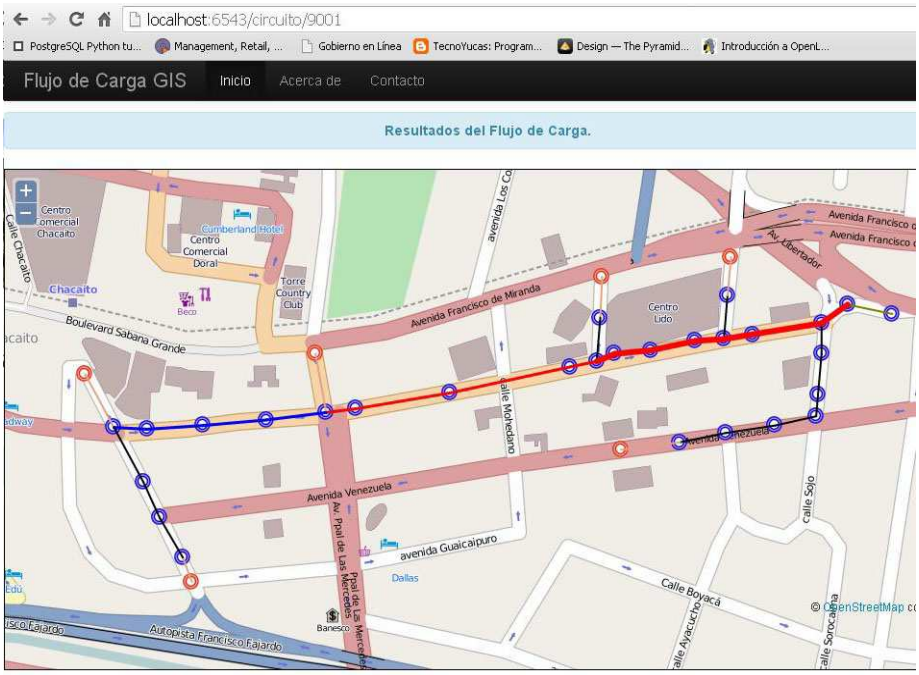


Figura 6.2. Resultados del Flujo de Carga bajo un mapa en OpenLayers del Circuito 9001.

Adicional se genera un reporte de resultados que es anexada en la parte inferior de dicha plantilla como la mostrada en la Figura 6.3:

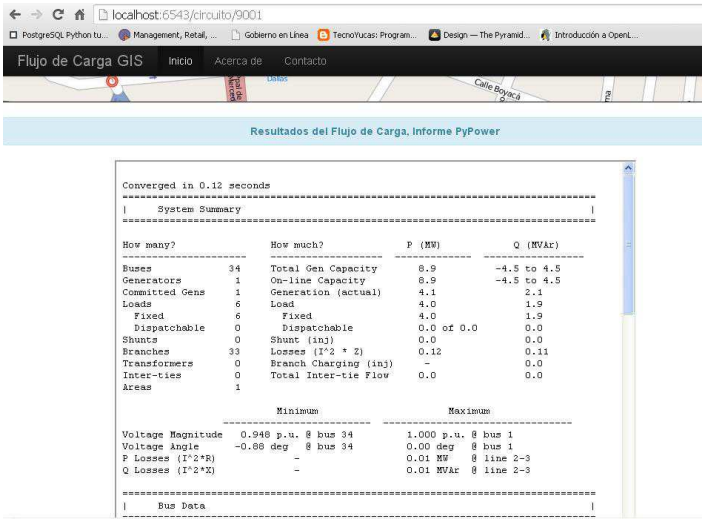


Figura 6.3. Resultados del Flujo de Carga del Circuito 9001 entregados por PyPower.

Los resultados visuales del mapa son interpretados por las siguientes leyendas, basando el color de las líneas de transmisión bajo el criterio del 66 % de la corriente que implica mantener el circuito en condiciones normales aunque se pierda una de las tres fases del sistema. El criterio puede observarse en la Figura 6.4 a continuación:

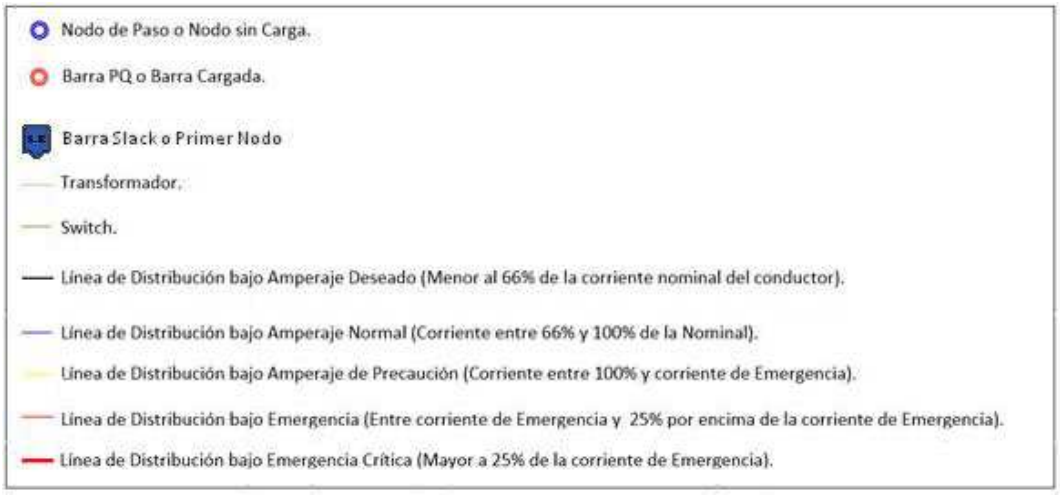


Figura 6.4. Leyenda y Criterio del 66% utilizado para la interfaz del mapa.

6.1. Respuestas de los Circuitos

Con el diseño individual de los circuitos se buscó mostrar todos los posibles casos de amperaje de rama con criterio del 66% indicado anteriormente. Entre los resultados del proceso tenemos el circuito 9001 que está correspondido por las barras indicadas en la Figura 6.5. Se puede ver que desde la salida inicial, los conductores han sido violados y se encuentran bajo emergencia crítica (color rojo grueso), esto motivado por el consumo de todas las ramas que van por el troncal del sistema. Posteriormente los niveles de corriente disminuyen por tramo entrando de emergencia (color rojo fino) a amperaje normal (color azul) y al final en corriente de amperaje deseado (color negro). El resultado de los voltajes nodales en por unidad y ángulos en grados se observan en el Cuadro 6.1.

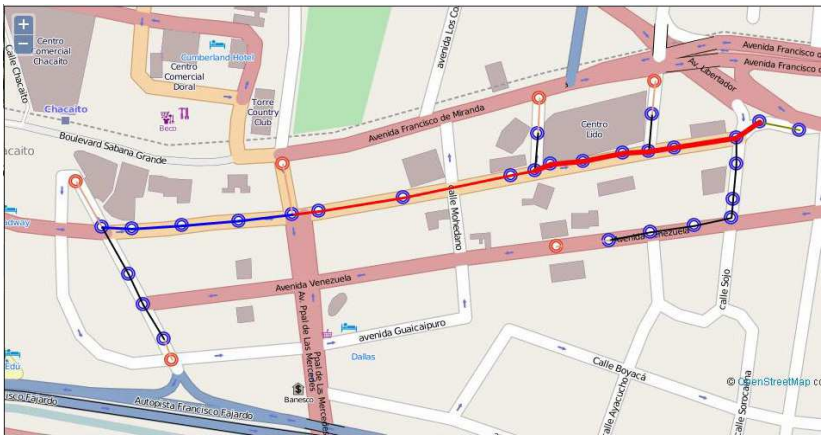


Figura 6.5. Circuito 9001 Referenciado Geográficamente con resultados del Flujo de Carga.

Cuadro 6.1. Voltajes en p.u. y ángulos en grados del Circuito 9001.

NODO	VOLTAJE (p.u.)	ANGULO (grados)
1	1,0000	0,0000
2	0,9960	-0,0550
3	0,9920	-0,1100
4	0,9920	-0,1190
5	0,9910	-0,1270
6	0,9910	-0,1350
7	0,9900	-0,1430
8	0,9900	-0,1510

9	0,9890	-0,1600
10	0,9860	-0,3250
11	0,9890	-0,1580
12	0,9860	-0,2060
13	0,9850	-0,2170
14	0,9820	-0,4360
15	0,9830	-0,2440
16	0,9810	-0,2810
17	0,9760	-0,3500
18	0,9740	-0,3800
19	0,9730	-0,3940
20	0,9690	-0,6580
21	0,9710	-0,4160
22	0,9680	-0,4670
23	0,9640	-0,5340
24	0,9600	-0,5860
25	0,9560	-0,8430
26	0,9570	-0,6260
27	0,9560	-0,6510
28	0,9540	-0,6720
29	0,9540	-0,6860
30	0,9520	-0,8090
31	0,9530	-0,6950
32	0,9510	-0,7220
33	0,9510	-0,7320
34	0,9480	-0,8840

Los niveles de voltaje en por unidad y ángulo con respecto a los nodos del sistema son mostrados en las Figuras 6.6 y 6.7 así como en la Figura 6.8 la magnitud de los voltajes referenciados por los valores de coordenadas en el cual han sido asignados. Se puede ver que al ser un sistema radial y no existir compensación en las barras sino únicamente barras PQ, se produce una caída de tensión constante desde la barra Slack del Circuito representada en este caso por el nodo 1 hacia el resto del sistema. Con respecto a los ángulos también se observa una disminución gradual desde la barra Slack, teniendo importante participación los nodos 10, 14, 20, 25, 30 y 34 al ser barras PQ, cuyo consumo de potencia activa repercute directamente en las diferencias angulares del circuito.

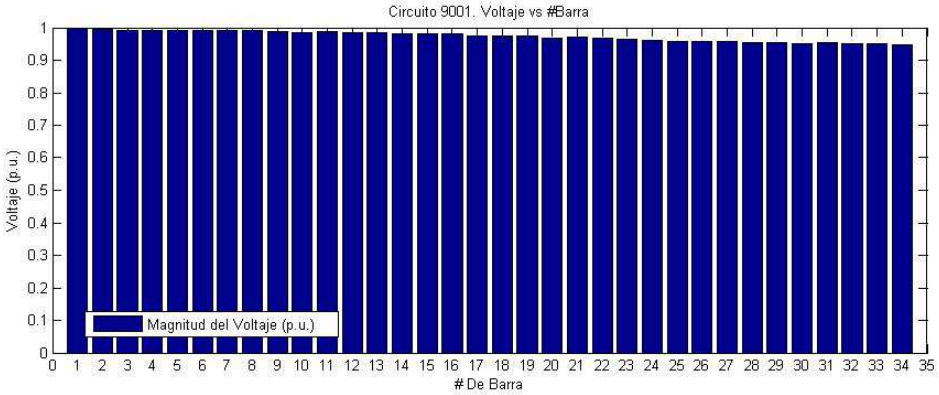


Figura 6.6. Voltaje en p.u. del circuito 9001 con respecto a las barras del sistema.

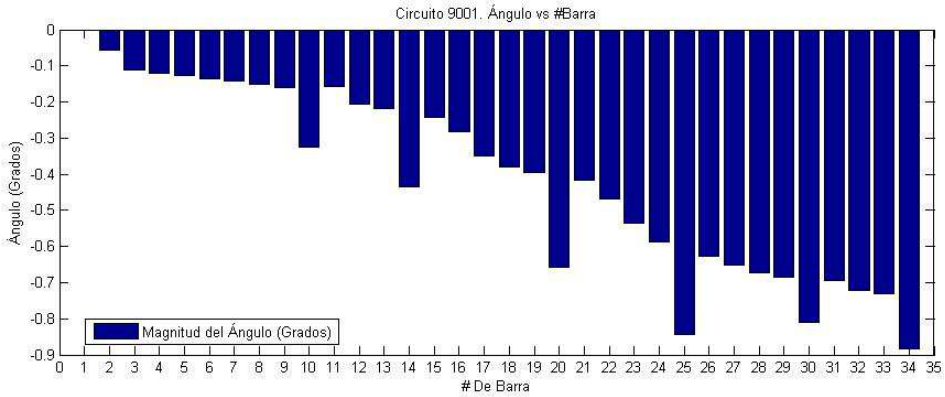
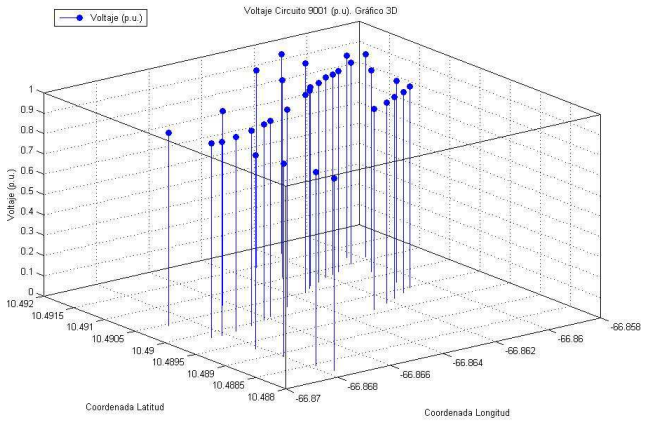


Figura 6.7. Ángulos en grados del circuito 9001 con respecto a las barras del sistema.



6.8. Voltajes del circuito 9001 referenciados por coordenadas de longitud y latitud.

Con respecto a los flujos de potencia el resumen de todas las ramas de las pérdidas activas y reactivas de las mismas así como las inyecciones de una barra a otra quedan resumidas en el Cuadro 6.2. Es importante destacar que en el Cuadro 6.2 no se hacen distinciones de las ramas que son del tipo "Línea" a las ramas que son del tipo "Transformador" y son mostradas ambas:

Cuadro 6.2. Información de Flujos de Potencia en las Ramas del Circuito 9001.

			INYECCION DESDE BUS		INYECCION HASTA BUS		PÉRDIDAS	
RAMA #	DESDE	HASTA	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
0	34	33	-0.49	-0.24	0.50	0.24	0.001	0.00
1	29	30	0.41	0.20	-0.40	-0.20	0.000	0.00
2	29	28	-0.90	-0.44	0.90	0.44	0.001	0.00
3	24	25	0.86	0.42	-0.85	-0.41	0.002	0.01
4	19	20	0.90	0.44	-0.90	-0.44	0.002	0.01
5	16	15	-2.71	-1.34	2.71	1.34	0.006	0.01
6	15	12	-2.71	-1.34	2.72	1.35	0.006	0.01
7	12	13	0.77	0.38	-0.77	-0.37	0.000	0.00
8	13	14	0.77	0.37	-0.76	-0.37	0.001	0.00
9	12	11	-3.48	-1.73	3.49	1.73	0.010	0.01
10	11	3	-3.49	-1.73	3.50	1.74	0.010	0.01
11	2	1	-4.11	-2.04	4.12	2.05	0.014	0.01
12	3	4	0.59	0.29	-0.59	-0.29	0.000	0.00
13	4	5	0.59	0.29	-0.59	-0.29	0.000	0.00
14	5	6	0.59	0.29	-0.59	-0.29	0.000	0.00
15	6	7	0.59	0.29	-0.59	-0.29	0.000	0.00
16	7	8	0.59	0.29	-0.59	-0.29	0.000	0.00
17	8	9	0.59	0.29	-0.59	-0.29	0.000	0.00
18	9	10	0.59	0.29	-0.58	-0.28	0.001	0.00
19	24	23	-1.76	-0.86	1.77	0.87	0.006	0.00
20	2	3	4.11	2.04	-4.09	-2.03	0.014	0.01
21	31	29	-0.50	-0.24	0.50	0.24	0.000	0.00
22	33	32	-0.50	-0.24	0.50	0.24	0.000	0.00
23	21	18	-1.78	-0.88	1.79	0.88	0.004	0.00
24	28	27	-0.90	-0.44	0.90	0.44	0.001	0.00
25	27	26	-0.90	-0.44	0.91	0.44	0.001	0.00
26	18	19	0.90	0.44	-0.90	-0.44	0.001	0.00

27	18	17	-2.69	-1.33	2.69	1.33	0.005	0.00
28	17	16	-2.69	-1.33	2.71	1.34	0.011	0.01
29	26	24	-0.91	-0.44	0.91	0.44	0.002	0.00
30	23	22	-1.77	-0.87	1.78	0.88	0.007	0.01
31	32	31	-0.50	-0.24	0.50	0.24	0.001	0.00
32	22	21	-1.78	-0.88	1.78	0.88	0.006	0.00

También a nivel de interfaz visual es posible verificar los datos del Flujo de Carga en el mapa, esto se logra haciendo clic sobre los nodos del circuito, dando información relevante respecto a los niveles de voltaje, ángulo, nombre del nodo y valores de carga de Potencia Activa “P” y potencia reactiva “Q”. Si no existe carga en la barra y es un nodo de paso los valores de P y Q se mostraran en cero ambos respectivamente. Además el usuario puede acceder a información de las ramas al hacer clic sobre ellas, y de igual forma se desplegará un cuadro con datos relevantes como la relación de los nodos de la rama, corrientes de la rama, corrientes nominales y de emergencia, distancia en kilómetros de la rama, impedancia de la misma, pérdidas activas y reactivas así como los flujos de potencia asociados a ellas. Esto puede ser observado en las Figuras 6.9 y 6.10 respectivamente.

Figura 6.9. Información de nodos en el mapa.

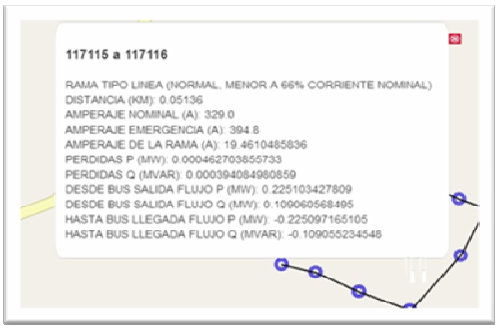


Figura 6.10. Información de Ramas en el mapa.

En la Figura 6.11 se muestra un resumen del circuito 9001 entregado por PyPower (por esta razón se encuentra en inglés), donde se observa la cantidad de nodos, barras, generadores, voltajes y ángulos máximos y mínimos entre otros. Por razones de espacio no es mostrada la información de los circuitos 9002, 9003, 9004 y 9005, sin embargo, en el Anexo G, podrán ver el resumen del Flujo de Carga de todos los circuitos con sus correspondientes gráficas.

Circuito 9001

```
Converged in 0.08 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    --------------------  -------------  -----------------
Buses               34   Total Gen Capacity       8.9          -4.5 to 4.5
Generators           1   On-line Capacity         8.9          -4.5 to 4.5
Committed Gens       1   Generation (actual)      4.1               2.1
Loads                6   Load                     4.0               1.9
  Fixed              6     Fixed                  4.0               1.9
  Dispatchable       0     Dispatchable           0.0 of 0.0        0.0
Shunts               0   Shunt (inj)              0.0               0.0
Branches            33   Losses (I^2 * Z)         0.12              0.11
Transformers         0   Branch Charging (inj)     -                0.0
Inter-ties           0   Total Inter-tie Flow     0.0               0.0
Areas                1

                          Minimum                      Maximum
                     -------------------------  -------------------------------
Voltage Magnitude   0.948 p.u. @ bus 34          1.000 p.u. @ bus 1
Voltage Angle      -0.88 deg   @ bus 34          0.00 deg   @ bus 1
P Losses (I^2*R)             -                  0.01 MW    @ line 2-3
Q Losses (I^2*X)             -                  0.01 MVAr  @ line 2-3

================================================================================
```

Figura 6.11. Resumen del Circuito 9001 entregado por PyPower.

Cabe destacar en la realización y posteriores resultados del proyecto los siguientes aspectos:

Python fue elegido para la realización de este proyecto, por múltiples razones, entre ellas la facilidad de aprendizaje y uso de las librerías científicas adecuadas pero una de las razones principales fue por la elección del tipo de arquitectura de software seleccionado de Modelo-Vista-Controlador ya que Python se maneja perfectamente con este patrón. Dicha forma de manejo de los datos en el proceso de desarrollo se pudo constatar su importancia ya que separa perfectamente en secciones cada parte del software y permite a su vez identificar problemas del código, por ejemplo, si existiera problemas en los tiempos de ejecución de algún proceso, se buscaría el controlador específico que es ajeno al Modelo de Datos o a los códigos asociados a las Vistas, y en consecuencia se atacaría más rápidamente el inconveniente; esto pasó reiteradamente durante el desarrollo del software ya que se separó el código que creaba el formato del caso de estudio en PyPower del módulo que corría el Flujo de Carga y creaba el script en formato KML y por ende se identificaban las secciones problemáticas con mayor facilidad y por ende rapidez. .

El Modelo de Datos entregado por el Instituto de Energía de la Universidad Simón Bolívar (INDENE) para el proyecto venía normalizado ya que las tablas del mismo presentaban un orden bien definido para lo que se necesitaba bajo un modelo relacional entre ellos (los datos de las ramas , nodos, circuitos, parámetros de conductores, entre otros venían especificados en tablas con datos propios de cada uno y a su vez bajo un índice relacional entre ellas si así era necesario, con esto se evitan datos redundantes o registros duplicados además de facilitar una posible actualización de los datos en ella), sin embargo, a medida que se fue trabajando con el mismo se encontraron incoherencias en la información suministrada, como datos de ramas que se apuntaban a nodos repetidos, o indexación de los calibres de conductor incoherentes que ocasionaban una no convergencia del Flujo de Carga en los circuitos, también luego de crear el algoritmo de verificación de radialidad de los circuitos del sistema se observó que apenas alrededor del 15% de ellos cumplían con esa condición que al ser una Base de Datos de Distribución sin generación distribuida típicamente debería cumplirse, obteniendo en vez islas radiales o circuitos mallados.

Otro inconveniente del Modelo de Datos era la falta de información geográfica, ya que inicialmente se entregaron las tablas relacionadas a los nodos del sistema sin coordenadas incluidas, lo que motivó a la verificación de la Base de Datos de origen desde donde se creó el modelo normalizado y aunque se logró relacionar más del 80% de los nodos, no fue suficiente para obtener circuitos completos, únicamente 6 circuitos que representan el 0.7 % completaron tanto la condición de radialidad como todas la información geográfica para su visualización. A este hecho se une también que una vez terminado el proceso de despliegue del circuito de forma referenciada los nodos que se muestran parecen tener problemas de ubicación con respecto al plano real de la zona, es decir, puntos de nodos desplazados de las avenidas o calles, ramas de dimensiones incoherentes que no se relacionan con los datos del Modelo de Datos, o nodos ubicados en puntos confusos. Esto puede ser causado muy probablemente por problemas en la conversión de las coordenadas desde el sistema UTM Huso 19 hacia WGS84, sin embargo, es mucho más probable que sean por información incorrecta desde la fuente de origen de datos y es por esta razón que al final se tuvo que crear circuitos inventados y así mostrar el potencial de la herramienta elaborada.

A nivel del Flujo de Carga, los datos originales mostraron circuitos muy poco cargados donde prácticamente en todas las ramas de los mismos no se llegaba ni al 50% del amperaje nominal. Cómo la finalidad principal de este proyecto es de investigación, y en líneas generales lo que se trata de mostrar es la capacidad del software creado, los circuitos fueron editados para ser cargados de modo que se mostraran todas las condiciones del criterio del 66%, es por esta razón que se ven casos donde la corriente de la rama supera inclusive el 25% de la corriente de emergencia del mismo, y existen ramales donde el amperaje está en niveles deseados poco comunes en una red real.

De igual forma se buscó que el circuito mostrado tenga caídas de tensión creíbles, llegando hasta un máximo del 10 % en por unidad en el circuito 9002, registrándose un mínimo de 0.901 p.u. en la barra 3702. Cómo los circuitos no presentan compensación la caída de voltaje es continua a medida que nos movemos desde la barra Slack hacia las ramas terminales, por supuesto tomando en cuenta el aporte de los nodos de carga para dicha diferencia de potencial.

Los tiempos de respuesta del algoritmo para el cálculo del flujo de carga son bastante eficientes, y como se están trabajando con circuitos independientes se reducen aún más los mismos ya que se trabajan con menos nodos, ramas y cargas, y cabe acotar que la topología radial hace que la matriz generada en el proceso presente numerosos valores nulos que simplifican muchísimo los cálculos, a diferencia de las redes múltiples malladas.

La estructura del Modelo de Datos permite realizar Flujo de Carga Trifásicos de una forma ordenada, sin embargo existe una carencia de información acerca de compensaciones y generación distribuida en el mismo. Por esta razón en la elaboración del algoritmo se obvio cualquier referencia a ellos y se trabajó con cargas nodales concentradas, es por eso que no existen barras PV en los circuitos. Esto puede ser una limitante a la hora del análisis de las redes de distribución, sin embargo, PyPower permite realizar dichas labores y ya que se ha trabajado con una Arquitectura de Software MVC, se puede modificar el controlador que crea el caso de estudio bajo el formato PyPower y adaptarlo a futuras modificaciones a el Modelo de Datos para que incluya dichas alternativas.

El servidor web utilizado, permitió fácilmente la adaptación de las plantillas y los tiempos de ejecución y carga son cortos y dependientes en gran medida de los subprocesos anteriores asociados, además de permitir el Modelo-Vista-Controlador. Su aplicación inicial es algo engorrosa ya que la metodología de trabajo es distinta a la de la mayoría de Frameworks Web, sin embargo, posee un potencial enorme una vez la curva de aprendizaje es superada, ya que la estructura de trabajo en el mismo es muy ordenada.

La interfaz gráfica usada fue la de plantillas web, que hoy día su ejecución es conocida ampliamente, donde únicamente se necesita la selección del circuito en estudio por medio de un cuadro combinado que despliega las opciones. El resto de las operaciones son todas realizadas por los controladores del sistema interactuando con el servidor web de Pyramid, así que para el usuario final es muy agradable e intuitivo de usar. Para el momento de la elaboración de este trabajo de grado se tienen 5 circuitos de prueba tomados directamente del Modelo de Datos, sin embargo, la información del cuadro combinado es actualmente estática y sólo muestra esos circuitos. Si se desea a futuro expandir la herramienta para un

indeterminado número de circuitos es necesario generar un validador que lo haga al cargar la página, como puede ser el caso de algún código JavaScript.

Conclusiones y Recomendaciones

La necesidad de utilizar herramientas computacionales para el análisis de los sistemas eléctricos de potencia es esencial. El constante avance en los sistemas de información y el hardware ha permitido un aumento en los procesos de cálculo y en consecuencia los métodos de estudio para el control en el área de la ingeniería eléctrica se han adaptado progresivamente y el resultado de este proyecto es sólo un ejemplo de esas capacidades. Para dichos fines existen diversos software tanto libres como propietarios, ejemplo de ello INTERPSS o el ETAP, sin embargo, la mayoría de éstos cómo se expresó en el capítulo 2, presentan limitaciones y prácticamente muy pocos relacionan lo geográfico con lo analítico; es por esto que parte del proceso de adaptación implica la utilización de Lenguajes de Programación para acoplar a las necesidades reales de las empresas los software que les permitan eficientemente el diseño, planificación y control de sus sistemas a bajo costo.

El principal Objetivo de este proyecto fue la elaboración de una herramienta computacional para el Análisis de Flujo de Carga sobre una referencia geográfica detallada bajo el uso de herramientas de software libre. Es de gran importancia recalcar tres virtudes que este proyecto ofrece, la primera y más importante es que se abre la posibilidad de obtener bajo la expansión del mismo, un software que logre analizar los múltiples aspectos de un sistema eléctrico bajo un costo nulo de licencia para las empresas, así casos particulares como los del ETAP o DigSilent-Power Factory cuyas licencias superan los $15.000 cada una pudieran a futuro ser sustituídas por herramientas similares que no amerite un gran golpe para las compañías más que el de la instrucción del personal o soporte técnico, de hecho en Venezuela, todas las herramientas de Análisis que incluyen soporte para Sistemas de Información Geográfico son de tipo propietario, como lo es el caso de PADEE de licencia AUTOCAD utilizado por CADAFE, SIMIP y SIGRED de licencia BENTLEY utilizado por la EDC, y FeederAll de licencia ABB empleado por ENELVEN, éste proyecto sería el primero de distribución Libre, lo que le aporta un mayor peso al mismo.

Una segunda virtud es la posibilidad de correr Flujos de Carga con información dinámica, ya que al ser aplicada en una Base de Datos y si existen herramientas que vinculen un sistema SCADA con el mismo, se lograría el análisis para obtener el estado del sistema en tiempo real, lo que permitiría un mejor control de problemas, tiempos de reacción reducidos de los departamentos involucrados y en líneas generales capacidades de diseño, planificación de expansión superiores. Una tercera virtud y no menos importante es la capacidad de ser visualizado vía web, lo que mejora la accesibilidad a los datos incluso desde cualquier parte del mundo por supuesto bajo las condiciones de seguridad que ello conlleven, esto trae como consecuencia una mayor información del estado del sistema y tiempos de reacción más cortos cuando el personal encargado del siniestro no se encuentra dentro del área de trabajo y suceda un inconveniente.

Finalmente, pensando en la mejora y continuación del proyecto por parte de un tercero se recomiendan las siguientes acciones:

- Verificar la veracidad de la información suministrada por el Modelo de Datos, o en su defecto por la Base de Datos fuente, ya que se encontraron múltiples problemas de indexación, incoherencias de datos o información corrupta para su análisis y esto redujo la confiabilidad de que los resultados correspondan con la realidad.

- Examinar los circuitos y solucionar los problemas de radialidad encontrados, ya que sólo el 15% de los mismos cumplían con este requisito, lo que indica que la información no es confiable. Para este fin se debe verificar las diversas islas radiales independientes o los diversos circuitos mallados y encontrar un criterio adecuado para cerrar o abrir los Switchs que hacen que dichos circuitos vuelvan a ser radiales.

- Ubicar la información de las coordenadas de los nodos para mostrarlos vía web, ya que sólo el 0.7% de los circuitos totales tenían completas las coordenadas en todas sus barras y cumplían con ser circuitos radiales.

- Modificar el Modelo de Datos para que en su estructura contemplen las compensaciones o las generaciones distribuidas (barras tipo PV), para que el modelo sea más completo y confiable, además de lograr una mayor capacidad de análisis para el diseño de la red o simple expansión del mismo.

- Mejorar el apartado visual de los componentes del sistema ya que por cuestiones de tiempo se trataron a los transformadores y a los Switchs del sistema cómo líneas de colores específicos bajo su leyenda en vez de con su figura eléctrica correspondiente.

Referencias bibliográficas

Culebro, M., Gómez, W., & Torres, S. (2006). *Software libre vs software propietario. Ventajas y desventajas.* Creative Commons.

Dunstan, L. A. (Enero de 1947). Machine Computation of Power Network Performance. *Transactions of the American Institute of Electrical Engineers* , 610-624.

Delgado, S. (Noviembre, 2013). Aplicación del estandar IEC-61970 al SEN y Analisis de Flujo de Carga en software abierto. Caracas, Miranda, Venezuela.

FUNINDES-USB. (2013). ¿Quiénes Somos?. Recuperado el 19 de Septiembre de 2013, de http://www.funindes.usb.ve/

García, M. (2009). Ecuaciones Trascendentes en Algebras de Banach. Ciudad de la Habana, Cuba.

IEEE Std. (1993). *IEEE Software Engineering Standard: Glossary of Software Engineering Terminology*. IEEE Computer Society Press, capitulo 2

INDENE-USB. (2010). *Anteproyecto para el desarrollo de herramienta computacional para estudios de planificación corto-mediano plazo de sistemas eléctricos de distribución.* Caracas.

INDENE-USB. (2013). Proyecto MCTI-FONACIT-PEI N° 2953: Desarrollo de una Herramienta Computacional en Codigo Abierto para Estudios del Sistema Electrico de Transmision y Distribucion de Venezuela. Recuperado el 19 de Septiembre de 2013, de http://www.indene.usb.ve/pei2953.html

Kersting W., (2002). *Distribution System Modeling and Analysis*. CRC Press.

Lincoln, R. (2011). *PyPower Usage*. Recuperado el 19 de Septiembre de 2013, de https://github.com/rwl/PYPOWER/blob/master/doc/usage.rst#id3

Lincoln, Richard ; Power System Engineering Research Center. (2009). *PyPower.* Recuperado el 19 de Septiembre de 2012, de http://pypi.python.org/pypi/PYPOWER/4.0.1

Martinez, R. (2009). *PostgreSQL*. Recuperado el 19 de Septiembre de 2013, de http://www.postgresql.org.es/documentacion

Milano, F. (2010). *Power System Modelling and Scripting.* Springer.

Milano, F., Zhou, M., & Hou, G. (2009). Open Model For Exchanging Power System Data. *IEEE PES General Meeting.*

Ochoa, F. J. (Marzo de 2012). Estudios de Cortocircuito balanceado en sistemas de distribución sobre una plataforma libre. Caracas, Miranda, Venezuela.

Plotz, M. (2010). *Principles Of MVC for PHP Developers*. Recuperado el 19 de Septiembre de 2013, de: http://www.htmlgoodies.com/beyond/php/article.php/3912211

Ruiz C. (2012) *Breve Introducción a Pyramid.* Caracas.

Ruiz C. (2013) *Geomática Libre, presentación OSM.* Recuperado el 19 de Septiembre de 2013, de http://atmantree.com/go/2013/08/breve-introduccion-a-pyramid/

Stallman, R. M. (2004). *Software libre para una sociedad libre.* Traficantes de Sueños.

Stevenson, W. (1975). *Análisis de Sistemas Eléctricos de Potencia.*McGraw-Hill.

Stevenson, W., & Grainger, J. (1994). *Power System Analysis.* McGraw-Hill.

Tyler E. (2011). *PyKML*. Recuperado el 19 de Septiembre de 2013, de http://pythonhosted.org/pykml/

ANEXO A: Caso de Estudio 4 Barras.

Caso de Estudio de 4 barras de Grainger & Stevenson

Formato de los casos de estudio de PyPower. Caso cuatro (4) barras.

```
# Copyright (C) 1996-2011 Power System Engineering Research Center
# Copyright (C) 2010-2011 Richard Lincoln
#
# PYPOWER is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published
# by the Free Software Foundation, either version 3 of the License,
# or (at your option) any later version.
#
# PYPOWER is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with PYPOWER. If not, see <http://www.gnu.org/licenses/>.

"""Power flow data for 4 bus, 2 gen case from Grainger & Stevenson.
"""

from numpy import array

def case4gs():
    """Power flow data for 4 bus, 2 gen case from Grainger & Stevenson.
    Please see L{caseformat} for details on the case file format.

    This is the 4 bus example from pp. 337-338 of I{"Power System Analysis"},
by John Grainger, Jr., William Stevenson, McGraw-Hill, 1994.

    @return: Power flow data for 4 bus, 2 gen case from Grainger & Stevenson.
    """
ppc = {"version": '2'}

    ##-----  Power Flow Data  -----##
    ## system MVA base
ppc["baseMVA"] = 100.0

    ## bus data
    # bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
ppc["bus"] = array([
        [0, 3, 50,  30.99,  0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
        [1, 1, 170, 105.35, 0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
        [2, 1, 200, 123.94, 0, 0, 1, 1, 0, 230, 1, 1.1, 0.9],
        [3, 2, 80,  49.58,  0, 0, 1, 1, 0, 230, 1, 1.1, 0.9]
    ])

    ## generator data
    # bus, Pg, Qg, Qmax, Qmin, Vg, mBase, status, Pmax, Pmin, Pc1, Pc2,
    # Qc1min, Qc1max, Qc2min, Qc2max, ramp_agc, ramp_10, ramp_30, ramp_q, apf
ppc["gen"] = array([
        [3, 318, 0, 100, -100, 1.02, 100, 1, 318, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0,   0, 100, -100, 1,    100, 1, 0,   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    ])

    ## branch data
    #fbus, tbus, r, x, b, rateA, rateB, rateC, ratio, angle, status, angmin, angmax
ppc["branch"] = array([
        [0, 1, 0.01008, 0.0504, 0.1025, 250, 250, 250, 0, 0, 1, -360, 360],
```

```
    [0, 2, 0.00744, 0.0372, 0.0775, 250, 250, 250, 0, 0, 1, -360, 360],
    [1, 3, 0.00744, 0.0372, 0.0775, 250, 250, 250, 0, 0, 1, -360, 360],
    [2, 3, 0.01272, 0.0636, 0.1275, 250, 250, 250, 0, 0, 1, -360, 360]
])
```

ANEXO B: Caso de Estudio Circuito del Modelo de Datos.

Caso de Estudio Circuito 9001

```
def estudio2():

    ## PYPOWER Case Format : Version 2
ppc = {'version': '2'}

    ##-----  Power Flow Data  -----##
    ## system MVA base
ppc['baseMVA'] = 4.45

    ## bus data
    # bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
ppc['bus'] = array([
    [1, 3, 0, 0, 0, 0, 1, 1, 0, 4.8, 1, 1.1, 0.9],
    [2, 1, 0, 0, 0, 0, 1, 0.99614236, -0.055025686, 4.8, 1, 1.1, 0.9],
    [3, 1, 0, 0, 0, 0, 1, 0.99228564, -0.11047916, 4.8, 1, 1.1, 0.9],
    [4, 1, 0, 0, 0, 0, 1, 0.99173444, -0.11863447, 4.8, 1, 1.1, 0.9],
    [5, 1, 0, 0, 0, 0, 1, 0.99118326, -0.12679886, 4.8, 1, 1.1, 0.9],
    [7, 1, 0, 0, 0, 0, 1, 0.99008096, -0.14315489, 4.8, 1, 1.1, 0.9],
    [8, 1, 0, 0, 0, 0, 1, 0.98952984, -0.15134657, 4.8, 1, 1.1, 0.9],
    [9, 1, 0, 0, 0, 0, 1, 0.98897874, -0.15954739, 4.8, 1, 1.1, 0.9],
    [11, 1, 0, 0, 0, 0, 1, 0.98898094, -0.15815616, 4.8, 1, 1.1, 0.9],
    [12, 1, 0, 0, 0, 0, 1, 0.98567693, -0.20615283, 4.8, 1, 1.1, 0.9],
    [13, 1, 0, 0, 0, 0, 1, 0.98495253, -0.21693319, 4.8, 1, 1.1, 0.9],
    [15, 1, 0, 0, 0, 0, 1, 0.98309791, -0.24363601, 4.8, 1, 1.1, 0.9],
    [16, 1, 0, 0, 0, 0, 1, 0.9805193, -0.28131636, 4.8, 1, 1.1, 0.9],
    [17, 1, 0, 0, 0, 0, 1, 0.9758789, -0.34964273, 4.8, 1, 1.1, 0.9],
    [18, 1, 0, 0, 0, 0, 1, 0.97381694, -0.38021901, 4.8, 1, 1.1, 0.9],
    [19, 1, 0, 0, 0, 0, 1, 0.97286693, -0.3944817, 4.8, 1, 1.1, 0.9],
    [21, 1, 0, 0, 0, 0, 1, 0.97141808, -0.4157936, 4.8, 1, 1.1, 0.9],
    [22, 1, 0, 0, 0, 0, 1, 0.9679918, -0.46692029, 4.8, 1, 1.1, 0.9],
    [23, 1, 0, 0, 0, 0, 1, 0.96353879, -0.5339285, 4.8, 1, 1.1, 0.9],
    [26, 1, 0, 0, 0, 0, 1, 0.95747206, -0.62624526, 4.8, 1, 1.1, 0.9],
    [30, 1, 0.405, 0.19615, 0, 0, 1, 0.95171571, -0.80917536, 4.8, 1, 1.1, 0.9],
    [28, 1, 0, 0, 0, 0, 1, 0.9544781, -0.67224424, 4.8, 1, 1.1, 0.9],
    [29, 1, 0, 0, 0, 0, 1, 0.95359764, -0.68582832, 4.8, 1, 1.1, 0.9],
    [31, 1, 0, 0, 0, 0, 1, 0.95301553, -0.69480163, 4.8, 1, 1.1, 0.9],
    [32, 1, 0, 0, 0, 0, 1, 0.95126935, -0.72178745, 4.8, 1, 1.1, 0.9],
    [33, 1, 0, 0, 0, 0, 1, 0.95059033, -0.7323087, 4.8, 1, 1.1, 0.9],
    [34, 1, 0.495, 0.239739, 0, 0, 1, 0.94828124, -0.88409094, 4.8, 1, 1.1, 0.9],
    [14, 1, 0.765, 0.370506, 0, 0, 1, 0.98150252, -0.43565978, 4.8, 1, 1.1, 0.9],
    [10, 1, 0.585, 0.283328, 0, 0, 1, 0.9863548, -0.32530831, 4.8, 1, 1.1, 0.9],
    [25, 1, 0.855, 0.414095, 0, 0, 1, 0.95615473, -0.84332801, 4.8, 1, 1.1, 0.9],
    [27, 1, 0, 0, 0, 0, 1, 0.95588694, -0.65056176, 4.8, 1, 1.1, 0.9],
    [20, 1, 0.9, 0.43589, 0, 0, 1, 0.96875291, -0.65843274, 4.8, 1, 1.1, 0.9],
    [6, 1, 0, 0, 0, 0, 1, 0.9906321, -0.13497232, 4.8, 1, 1.1, 0.9],
    [24, 1, 0, 0, 0, 0, 1, 0.9601143, -0.58589618, 4.8, 1, 1.1, 0.9],
])

    ## generator data
    # bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin Pc1 Pc2 Qc1min Qc1max Qc2min Qc2max ramp_agc ramp_10
ramp_30 ramp_q apf
ppc['gen'] = array([
    [1, 4.12014, 2.05374, 4.45, -4.45, 1, 4.45, 1, 8.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
])

    ## branch data
    # fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax Pf Qf Pt Qt
ppc['branch'] = array([
```

```
    [34, 33, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.4950, -0.2397, 0.4956, 0.2416],
    [29, 30, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.4054, 0.1974, -0.4050, -0.1962],
    [29, 28, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.9023, -0.4402, 0.9030, 0.4408],
    [24, 25, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.8567, 0.4197, -0.8550, -0.4141],
    [19, 20, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.9018, 0.4419, -0.9000, -0.4359],
    [16, 15, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -2.7052, -1.3396, 2.7114, 1.3449],
    [15, 12, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -2.7114, -1.3449, 2.7176, 1.3502],
    [12, 13, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.7668, 0.3752, -0.7663, -0.3747],
    [13, 14, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.7663, 0.3747, -0.7650, -0.3705],
    [12, 11, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -3.4844, -1.7253, 3.4946, 1.7341],
    [11, 3, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -3.4946, -1.7341, 3.5048, 1.7428],
    [2, 1, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -4.1062, -2.0419, 4.1201, 2.0537],
    [3, 4, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5874, 0.2872, -0.5872, -0.2870],
    [4, 5, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5872, 0.2870, -0.5869, -0.2868],
    [5, 6, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5869, 0.2868, -0.5866, -0.2865],
    [6, 7, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5866, 0.2865, -0.5863, -0.2863],
    [7, 8, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5863, 0.2863, -0.5860, -0.2860],
    [8, 9, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5860, 0.2860, -0.5857, -0.2858],
    [9, 10, 0.0075, 0.0251, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.5857, 0.2858, -0.5850, -0.2833],
    [24, 23, 0.00585028, 0.00498269, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -1.7644, -0.8645, 1.7699, 0.8692],
    [2, 3, 0.00292514, 0.00249134, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 4.1062, 2.0419, -4.0923, -2.0300],
    [31, 29, 0.00351017, 0.00298961, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.4967, -0.2426, 0.4969, 0.2428],
    [33, 32, 0.0040952, 0.00348788, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.4956, -0.2416, 0.4959, 0.2419],
    [21, 18, 0.0040952, 0.00348788, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -1.7825, -0.8799, 1.7864, 0.8832],
    [28, 27, 0.00468023, 0.00398615, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.9030, -0.4408, 0.9042, 0.4418],
    [27, 26, 0.00526525, 0.00448442, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.9042, -0.4418, 0.9055, 0.4429],
    [18, 19, 0.00321766, 0.00274048, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, 0.9026, 0.4426, -0.9018, -0.4419],
    [18, 17, 0.00234011, 0.00199307, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -2.6890, -1.3258, 2.6939, 1.3300],
    [17, 16, 0.00526525, 0.00448442, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -2.6939, -1.3300, 2.7052, 1.3396],
    [26, 24, 0.00877542, 0.00747403, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.9055, -0.4429, 0.9077, 0.4448],
    [23, 22, 0.00760537, 0.00647749, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -1.7699, -0.8692, 1.7770, 0.8753],
    [32, 31, 0.0105305, 0.00896884, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -0.4959, -0.2419, 0.4967, 0.2426],
    [22, 21, 0.00585028, 0.00498269, 0, 11.125, 11.125, 11.125, 0, 0, 1, -360, 360, -1.7770, -0.8753, 1.7825, 0.8799],
  ])

return ppc
```

ANEXO C: Generación del KML.

Generación del Archivo KML (Código Python).

```
d = request.matchdict['uid']
conn=psycopg2.connect('host=192.168.0.108 port=5432 dbname=bd_edc_02_daniel user=indene password=indene01');
cur = conn.cursor();
converge=[]
sql_query = 'SELECT fu,fp FROM pps.circuit_factors WHERE circuit_id ='+str(d)
cur.execute(sql_query);
fu_fp = cur.fetchall();
cur.execute("SELECT first_node FROM pps.circuit WHERE id = '"+str(d)+"';");
primer_nodo = cur.fetchall();
ppc=loadfromdb(d);
ppopt=ppoption(PF_ALG=1, PF_MAX_IT= 100, TOL=1e-6)
#Corrida del flujo de carga#
r=runpf(ppc,ppopt, fname='resultados'+d +".txt", solvedcase='estudio2')
#Creacion de el archivo KML
i=0;
################################KML NODOS#####################################
# Ambos estudios para referenciar al KML con el dibujo correcto
        #estudio NO HAY CARGA
doc = KML.kml(KML.Document(KML.Name("Flujo de
Carga"),KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("https://dl.dropboxusercontent.com/u/43356135/pin4.png")),)
,id="linea",)),KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("https://dl.dropboxusercontent.com/u/43356135/pin5.pn
g")),),id="carga",),KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("https://dl.dropboxusercontent.com/u/43356135/po
werlinepole.png")),),id="subestacion",))
nodoi=(int(r[0]["order"]["bus"]["i2e"].tolist()[i]));
sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoi) +"'"
cur.execute(sql_query);
variosnodoi = cur.fetchall();
        #estudio si es barra slack
if float(nodoi) == float(primer_nodo[0][0]):
        descripcion = "PRIMER NODO (BARRA SLACK): " + str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) + "<br/>" +
"VOLTAJE, P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" + "ANGULO EN GRADOS: " + str(r[0]["order"]["int"]["bus"][i][8])
+ "<br/>" + "CARGA P (MW): " + str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA Q (MVAR): " +
str(float(r[0]["order"]["int"]["bus"][i][3])) ;
        pm1 =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#subestacion"),KML.description(descripcion),K
ML.Point(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))
        print("entre en nodo slack")
        #estudio HAY CARGA
if (r[0]["order"]["int"]["bus"][i][2] == float(0)) and (float(nodoi) != float(primer_nodo[0][0])):
        descripcion = "NODO SIN CARGA: " + str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) + "<br/>" + "VOLTAJE,
P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" + "ANGULO EN GRADOS: " + str(r[0]["order"]["int"]["bus"][i][8]) + "<br/>" +
"CARGA P (MW): " + str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA Q (MVAR): " +
str(float(r[0]["order"]["int"]["bus"][i][3])) ;
        pm1 =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#linea"),KML.description(descripcion),KML.Poi
nt(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))
        #estudio HAY CARGA
if r[0]["order"]["int"]["bus"][i][2] != float(0):
        descripcion = "NODO CON CARGA: " + str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) + "<br/>" + "VOLTAJE,
P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" + "ANGULO EN GRADOS: " + str(r[0]["order"]["int"]["bus"][i][8]) + "<br/>" +
"CARGA P (MW): " + str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA Q (MVAR): " +
str(float(r[0]["order"]["int"]["bus"][i][3])) ;
        pm1 =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#carga"),KML.description(descripcion),KML.Poi
nt(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))
```

```
                #print(descripcion)
        style=KML.Style(KML.IconStyle(KML.scale(1.2),KML.Icon(KML.href("http://maps.google.com/mapfiles/kml/shapes/shade
d_dot.png")),),id="linea",)
        doc.Document.append(pm1)
        for w in r[0]["order"]["int"]["bus"]:
                nodoi=(int(r[0]["order"]["bus"]["i2e"].tolist()[i]));
                sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoi) +"'"
                cur.execute(sql_query);
                variosnodoi = cur.fetchall();
                if i!=0:
                        #estudio si es barra slack
                        if float(nodoi) == float(primer_nodo[0][0]):
                                descripcion = "PRIMER NODO (BARRA SLACK): " +
str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) + "<br/>" + "VOLTAJE, P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" +
"ANGULO EN GRADOS: " + str(r[0]["order"]["int"]["bus"][i][8]) + "<br/>" + "CARGA P (MW): " +
str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA Q (MVAR): " + str(float(r[0]["order"]["int"]["bus"][i][3])) ;
                                pm1 =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#subestacion"),KML.description(descripcion),K
ML.Point(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))

                        #estudio NO HAY CARGA
                        if (r[0]["order"]["int"]["bus"][i][2] == float(0)) and (float(nodoi) != float(primer_nodo[0][0])):
                                descripcion = "NODO SIN CARGA: " + str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) +
"<br/>" + "VOLTAJE, P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" + "ANGULO EN GRADOS: " +
str(r[0]["order"]["int"]["bus"][i][8]) + "<br/>" + "CARGA P (MW): " + str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA
Q (MVAR): " + str(float(r[0]["order"]["int"]["bus"][i][3])) ;
                                pm =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#linea"),KML.description(descripcion),KML.Poi
nt(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))
                        #estudio HAY CARGA
                        if r[0]["order"]["int"]["bus"][i][2] != float(0):
                                descripcion = "NODO CON CARGA: " + str(int(r[0]["order"]["bus"]["i2e"].tolist()[i])) +
"<br/>" + "VOLTAJE, P.U: " + str(r[0]["order"]["int"]["bus"][i][7]) + "<br/>" + "ANGULO EN GRADOS: " +
str(r[0]["order"]["int"]["bus"][i][8]) + "<br/>" + "CARGA P (MW): " + str(float(r[0]["order"]["int"]["bus"][i][2])) + "<br/>" + "CARGA
Q (MVAR): " + str(float(r[0]["order"]["int"]["bus"][i][3])) ;
                                pm =
KML.Placemark(KML.name(int(r[0]["order"]["bus"]["i2e"].tolist()[i])),KML.styleUrl("#carga"),KML.description(descripcion),KML.Poi
nt(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0][2])))))

                        doc.Document.append(pm)
        i=i+1;

        ################################ KML RAMAS ###################################

        numeronodo=[]
        i=0;
        for q in r[0]["order"]["int"]["bus"]:
                numeronodo.append(int(r[0]["order"]["int"]["bus"][i][0]));
                i=i+1;

        i=0;

        for ww in r[0]["order"]["int"]["branch"]:
                nodoiindex=numeronodo.index(int(r[0]["order"]["int"]["branch"][i][0]));
                nodojindex=numeronodo.index(int(r[0]["order"]["int"]["branch"][i][1]));

                nodoi=(int(r[0]["order"]["bus"]["i2e"].tolist()[nodoiindex]));
                nodoj=(int(r[0]["order"]["bus"]["i2e"].tolist()[nodojindex]));

                sql_query = 'SELECT id,node0_id, node1_id, elem, distance,amps,amps_emergency,r1,x1 FROM pps.branch
WHERE (node0_id =' +str(nodoi) + ') AND (node1_id = '+str(nodoj)+')'
                cur.execute(sql_query);
                ramas = cur.fetchall();
                ramas_list = list(ramas);
                #Si son Lineas

                if ramas_list[0][3] == 'L':
                        #Aqui se confirma si esta violada la linea.
```

```
                    sql_query= "SELECT nvl_tensi FROM pps.circuit_factors WHERE circuit_id = '"+str(d)+"'"
                    cur.execute(sql_query);
                    voltaje_base = cur.fetchall();
                    sbase= float(r[0]["order"]["int"]["gen"][0][3]);
                    zbase = (float(voltaje_base[0][0])**2/(sbase));

                    sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoi) +"'"
                    cur.execute(sql_query);
                    variosnodoi = cur.fetchall();


                    sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoj) +"'"
                    cur.execute(sql_query);
                    variosnodoj = cur.fetchall();


                    voltajenodoi=float(r[0]["order"]["int"]["bus"][nodoiindex][7])
                    angulonodoi=float(r[0]["order"]["int"]["bus"][nodoiindex][8])
                    voltajenodoj=float(r[0]["order"]["int"]["bus"][nodojindex][7])
                    angulonodoj=float(r[0]["order"]["int"]["bus"][nodojindex][8])
                    voltajerealnodoi=voltajenodoi*math.cos(angulonodoi*0.05304532925)*(float(variosnodoi[0][0]))
                    votajeimagnodoi=voltajenodoi*math.sin(angulonodoi*0.05304532925)
                    voltajerealnodoj=voltajenodoj*math.cos(angulonodoj*0.05304532925)*(float(variosnodoj[0][0]))
                    votajeimagnodoj=voltajenodoj*math.sin(angulonodoj*0.05304532925)
                    difvoltajeijreal=voltajerealnodoi-voltajerealnodoj
                    difvoltajejimag=votajeimagnodoi-votajeimagnodoj

                    xt = math.sqrt(difvoltajeijreal*difvoltajeijreal + difvoltajejimag*difvoltajejimag)


          amperaje_rama=math.fabs((xt)/math.sqrt((float((ramas_list[0][7]*ramas_list[0][4]))*float((ramas_list[0][7]*ramas_list[0][4]))
)+((float((ramas_list[0][8]*ramas_list[0][4]))*float((ramas_list[0][8]*ramas_list[0][4])))))*1000)
                    if (amperaje_rama < float(ramas_list[0][5])*0.666): #Normal
                              descripcion = "RAMA TIPO LINEA (NORMAL, MENOR A 66% CORRIENTE
NOMINAL) " + "<br/>" + "DISTANCIA (KM):  " + str(float(ramas_list[0][4])) + "<br/>" + "AMPERAJE NOMINAL (A): " +
str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " + str(float(ramas_list[0][6])) + "<br/>" + "AMPERAJE DE
LA RAMA (A): " + str(amperaje_rama) + "<br/>" + "PERDIDAS P (MW): " + str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" +
"PERDIDAS Q (MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" + "DESDE BUS SALIDA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q (MW): " +
str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                              pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0
"+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff000000"),KML.widt
h("2"))))

                    if (amperaje_rama > float(ramas_list[0][5])*0.666 and amperaje_rama < float(ramas_list[0][5])):
#Normal llegando al limite
                              descripcion = "RAMA TIPO LINEA (ENTRE 66 % Y CORRIENTE NOMINAL)  " +
"<br/>" + "DISTANCIA (KM):  " + str(float(ramas_list[0][4])) + "<br/>" + "AMPERAJE NOMINAL (A): " +
str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " + str(float(ramas_list[0][6])) + "<br/>" + "AMPERAJE DE
LA RAMA (A): " + str(amperaje_rama) + "<br/>" + "PERDIDAS P (MW): " + str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" +
"PERDIDAS Q (MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" + "DESDE BUS SALIDA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q (MW): " +
str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                              pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0
"+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ffff0000"),KML.width
("3"))))

                    if (amperaje_rama > float(ramas_list[0][5]) and amperaje_rama < float(ramas_list[0][6])): #Precaucion
                              descripcion = "RAMA TIPO LINEA (ENTRE CORRIENTE NOMINAL Y CORRIENTE
EMERGENCIA)  " + "<br/>" + "DISTANCIA (KM):  " + str(float(ramas_list[0][4])) + "<br/>" + "AMPERAJE NOMINAL (A): " +
str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " + str(float(ramas_list[0][6])) + "<br/>" + "AMPERAJE DE
```

```
LA RAMA (A): " + str(amperaje_rama) + "<br/>" + "PERDIDAS P (MW): " + str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" +
"PERDIDAS Q (MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" + "DESDE BUS SALIDA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q (MW): " +
str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                                        pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0
"+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff00ffff"),KML.width(
"3"))))

                                if (amperaje_rama > float(ramas_list[0][6]) and amperaje_rama < float(ramas_list[0][6])*1.25):
#EMERGENCIA CRITICA
                                        descripcion = "RAMA TIPO LINEA (ENTRE CORRIENTE EMERGENCIA Y 125%
CORRIENTE EMERGENCIA)  " + "<br/>" + "DISTANCIA (KM):  " + str(float(ramas_list[0][4])) + "<br/>" + "AMPERAJE
NOMINAL (A): " + str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " + str(float(ramas_list[0][6])) + "<br/>"
+ "AMPERAJE DE LA RAMA (A): " + str(amperaje_rama) + "<br/>" + "PERDIDAS P (MW): " +
str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" + "PERDIDAS Q (MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" +
"DESDE BUS SALIDA FLUJO P (MW): " + str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q
(MW): " + str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                                        pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0
"+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff0000ff"),KML.width
("3"))))

                                if (amperaje_rama > float(ramas_list[0][6])*1.25): #EMERGENCIA CRITICA
                                        descripcion = "RAMA TIPO LINEA (ENTRE CORRIENTE EMERGENCIA Y 125%
CORRIENTE EMERGENCIA)  " + "<br/>" + "DISTANCIA (KM):  " + str(float(ramas_list[0][4])) + "<br/>" + "AMPERAJE
NOMINAL (A): " + str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " + str(float(ramas_list[0][6])) + "<br/>"
+ "AMPERAJE DE LA RAMA (A): " + str(amperaje_rama) + "<br/>" + "PERDIDAS P (MW): " +
str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" + "PERDIDAS Q (MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" +
"DESDE BUS SALIDA FLUJO P (MW): " + str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q
(MW): " + str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                                        pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0
"+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff0000ff"),KML.width
("6"))))


                        # Si son Transformadores
                        if ramas_list[0][3] == 'T':
                                sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoi) +"'"
                                cur.execute(sql_query);
                                variosnodoi = cur.fetchall();


                                sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoj) +"'"
                                cur.execute(sql_query);
                                variosnodoj = cur.fetchall();
                                if (float(fu_fp[0][0])>1.2):
                                        descripcion = "RAMA TIPO TRANSFORMADOR (TRANSFORMADOR OVERLOAD)"
+ "<br/>" + "IMPEDANCIA DEL TRX EN P.U:  " + (str(float(ramas_list[0][7])) + " + j" +str(float(ramas_list[0][8])));
                                        pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0 "+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff00a8ff"))))

                                if (float(fu_fp[0][1])<=1.2):

                                        descripcion = "RAMA TIPO TRANSFORMADOR" + "<br/>" + "IMPEDANCIA DEL TRX
EN P.U:  " + (str(float(ramas_list[0][7])) + " + j" +str(float(ramas_list[0][8])));
```

```
                                        pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0 "+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff84a2e8"))))
                        # Si son Switches
                        if ramas_list[0][3] == 'S':
                                sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoi) +"'"
                                cur.execute(sql_query);
                                variosnodoi = cur.fetchall();


                                sql_query = "SELECT kv,latitud,longitud FROM pps.node WHERE id ='" +str(nodoj) +"'"
                                cur.execute(sql_query);
                                variosnodoj = cur.fetchall();
                                descripcion = "RAMA TIPO SWITCH " + "<br/>" + "ESTADO: CERRADO " + "<br/>" +
"AMPERAJE NOMINAL (A): " + str(float(ramas_list[0][5])) + "<br/>" + "AMPERAJE EMERGENCIA (A): " +
str(float(ramas_list[0][5])) + "<br/>" + "PERDIDAS P (MW): " + str(r[0]["order"]["int"]["branch"][i][2])+ "<br/>" + "PERDIDAS Q
(MVAR): " + str(r[0]["order"]["int"]["branch"][i][3])+ "<br/>" + "DESDE BUS SALIDA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][13])+ "<br/>" + "DESDE BUS SALIDA FLUJO Q (MW): " +
str(r[0]["order"]["int"]["branch"][i][14])+ "<br/>" + "HASTA BUS LLEGADA FLUJO P (MW): " +
str(r[0]["order"]["int"]["branch"][i][15])+ "<br/>" + "HASTA BUS LLEGADA FLUJO Q (MVAR): " +
str(r[0]["order"]["int"]["branch"][i][16]) ;
                                pm = KML.Placemark(KML.name(str(nodoi) +" a
"+str(nodoj)),KML.description(descripcion),KML.LineString(KML.coordinates(str(float(variosnodoi[0][1]))+","+str(float(variosnodoi[0]
[2]))+",0 "+str(float(variosnodoj[0][1]))+","+str(float(variosnodoj[0][2]))+",0")),KML.Style(KML.LineStyle(KML.color("#ff008080"))))

                        doc.Document.append(pm)

                        i=i+1;


        print(os.path.dirname(__file__)+'\\resultados', os.path.dirname(__file__)+'\\static\\resultados')
        shutil.copy2(os.path.dirname(__file__)+'\\resultados', os.path.dirname(__file__)+'\\static\\resultados')

        # output a KML file (named based on the Python script)
        outfile = file(os.path.dirname(__file__)+'\\static\\'+os.path.basename(os.path.splitext(__file__)[0])+'_'+d+'.kml','w')

        outfile.write(etree.tostring(doc, pretty_print=True))
```

ANEXO D: Plantillas web y configuración archivos de Pyramid.

Código de la plantilla mytemplate.pt

```
<!DOCTYPEhtml>
<!--
savedfromurl=(0130)file:///C:/Documents%20and%20Settings/Daniel%20Vitulli/Mis%20documentos/Dropbox/prueba%20repositorio%
20html/repositorio_indene.htm-->
<htmllang="en"><head><metahttp-equiv="Content-Type"content="text/html;charset=UTF-8">
<metacharset="utf-8">
<title>FlujodeCargaINDENE,U.S.B.</title>
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<metaname="description"content="">
<metaname="author"content="">

<!--Lestyles-->
<linkhref="${request.static_url('prueba3:static/bootstrap.css')}"rel="stylesheet">
<styletype="text/css">
body{
padding-top:60px;
padding-bottom:40px;
}
.sidebar-nav{
padding:9px0;
}

@media(max-width:980px){
/*Enableuseoffloatednavbartext*/
.navbar-text.pull-right{
float:none;
padding-left:5px;
padding-right:5px;
}
}
</style>
<linkhref="${request.static_url('prueba3:static/bootstrap-responsive.css')}">

<!--HTML5shim,forIE6-8supportofHTML5elements-->
<!--[ifltIE9]>
<scriptsrc="../assets/js/html5shiv.js"></script>
<![endif]-->

<!--Favandtouchicons-->
<linkrel="apple-touch-icon-precomposed"sizes="144x144"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-144-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"sizes="114x114"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-114-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"sizes="72x72"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-72-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-57-precomposed.png">
<linkrel="shortcuticon"href="http://twitter.github.io/bootstrap/assets/ico/favicon.png">
<scriptsrc="${request.static_url('prueba3:static/dpt.js')}"></script>
<script>
functionllamarCircuito(){
vare=document.getElementById("combo1");
circuito=e.options[e.selectedIndex].value;
if(circuito!=0)
window.location.href="/circuito/"+circuito;
```

```
}
</script>

</head>
<body><divclass="container"style="text-align:center;">
<imgalign="center"src="${request.static_url('prueba3:static/cromoconlogos.jpg')}"alt="indene">
</div>
<br>
<br>
<divclass="navbarnavbar-inversenavbar-fixed-top">
<divclass="navbar-inner">
<divclass="container-fluid">
<buttontype="button"class="btnbtn-navbar"data-toggle="collapse"data-target=".nav-collapse">
<spanclass="icon-bar"></span>
<spanclass="icon-bar"></span>
<spanclass="icon-bar"></span>
</button>
<aclass="brand"href="/">FlujodeCargaGIS</a>
<divclass="nav-collapsecollapse">
<ulclass="nav">
<liclass="active"><ahref="http://www.indene.usb.ve">Inicio</a></li>
<li><ahref="http://www.indene.usb.ve">Acercade</a></li>
<li><ahref="http://www.indene.usb.ve">Contacto</a></li>
</ul>
</div><!--/.nav-collapse-->
</div>
</div>
</div>
<divclass="container-fluid"align="center">
<divclass="row-fluid"align="center">
<divclass="span13"align="center">
<aname="objetivo1"></a>
<divclass="well"align="center">
<h3align="center">DesarrollodeunaHerramientaComputacionaldeFlujodeCargaImplementadoenunaPlataformaGeográfica</h3>
</div>
<br>
<divclass="row-fluid"align="center">
<divclass="alertalert-info"align="center">
<iclass="icon-folder-open"></i>
<strong>SeleccioneelTipodeAnálisisquedeseaejecutar:</strong>
</div><!--/span-->

</div><aname="objetivo2"></a>
<divclass="span10"align="left">
<li>
<label>
<inputtype="radio"checked="true"name="optionsRadios"value="option1">
<span>FlujodeCarga-MétodoNewthonRaphson</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>FlujodeCarga-MétodoGaussSiedel</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>FlujodeCarga-MétodoDesacopladoRápido</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>FlujodeCargaMultifilar</span>
</label>
</li>
```

```
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>FlujodeCargaÓptimo</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>FlujodeCargaconArmónicos</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>EstudiodeCortoCircuito-Trifásico</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>EstudiodeCortoCircuito-Monofásico</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>EstimadordeEstados</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>AnálisisdeEstabilidaddeVoltaje</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>Análisisdeestabilidaddepequeñaseñal</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>Simulacioneseneldominiodeltiempoparaanálisisdeestabilidadtransitoria</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>Transitorioelectromagnético</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>EstimacióndeDemanada</span>
</label>
</li>
<li>
<label>
<inputtype="radio"disabled="disabled"name="optionsRadios"value="option1">
<span>ConfiguraciónÓptimadelaRed</span>
</label>
</li>
</div>
<br>
<br>
```

```
<br>
<br>
<divclass="alertalert-info"align="center">
<strong>SeleccioneunCircuitoparaejecutarelFlujodeCargayReferenciarGeográficamente.</strong>
</div><!--/span-->
<divclass="wellsidebar-nav"align="center">
<ulclass="navnav-list">
<selectid="estados"onChange="DropdownEstado()">
<optionvalue="00">--ESTADO--</option>
<optionvalue="01">DISTRITOCAPITAL</option>
<optionvalue="02">AMAZONAS</option>
<optionvalue="03">ANZOATEGUI</option>
<optionvalue="04">APURE</option>
<optionvalue="05">ARAGUA</option>
<optionvalue="06">BARINAS</option>
<optionvalue="07">BOLIVAR</option>
<optionvalue="08">CARABOBO</option>
<optionvalue="09">COJEDES</option>
<optionvalue="10">DELTAAMACURO</option>
<optionvalue="11">FALCON</option>
<optionvalue="12">GUARICO</option>
<optionvalue="13">LARA</option>
<optionvalue="14">MERIDA</option>
<optionvalue="15">MIRANDA</option>
<optionvalue="16">MONAGAS</option>
<optionvalue="17">NUEVAESPARTA</option>
<optionvalue="18">PORTUGUESA</option>
<optionvalue="19">SUCRE</option>
<optionvalue="20">TACHIRA</option>
<optionvalue="21">TRUJILLO</option>
<optionvalue="22">YARACUY</option>
<optionvalue="23">ZULIA</option>
<optionvalue="24">VARGAS</option>
</select>
<selectid="municipios"onChange="DropdownMunicipios()">
<optionvalue="0000">--MUNICIPIO--</option>
</select>
<selectclass="input-large"id="subestacion"name="subestacion">
<optionvalue="10000">--SUBESTACIÓN--</option>
<optionvalue="10001">ALTAMIRA</option>
<optionvalue="10002">ANAUCO</option>
<optionvalue="10003">ANGELES</option>
<optionvalue="10004">ANTIMANO</option>
<optionvalue="10005">ARTIGAS</option>
<optionvalue="10006">AVILANES</option>
<optionvalue="10007">BARBECHO</option>
<optionvalue="10008">BELLAVISTA</option>
<optionvalue="10009">BELLOMONTE</option>
<optionvalue="10010">BOLEITA</option>
<optionvalue="10011">BOULEVARD</option>
<optionvalue="10012">C.STA.MONICA</option>
<optionvalue="10013">CASADA</option>
<optionvalue="10014">CAFETAL</option>
<optionvalue="10015">CAICAGUANA</option>
<optionvalue="10016">CANDILITO</option>
<optionvalue="10017">CAOBOS</option>
<optionvalue="10018">CARABALLEDA</option>
<optionvalue="10019">CARAYACA</option>
<optionvalue="10020">CARICUAO</option>
<optionvalue="10021">CARRIZALES</option>
<optionvalue="10022">CASARAPA</option>
<optionvalue="10023">CASINO</option>
<optionvalue="10024">CASTAN</option>
<optionvalue="10025">CASTELLANA</option>
<optionvalue="10026">CATIA</option>
<optionvalue="10027">CAUCAGUITA</option>
<optionvalue="10028">CAURIMARE</option>
<optionvalue="10029">CHAGUARAMOS</option>
<optionvalue="10030">CHUSPA</option>
```

<optionvalue="10031">COCHE12.47kV</option>
<optionvalue="10032">COCHE4.8kV</option>
<optionvalue="10033">COLON</option>
<optionvalue="10034">COLONIATOVAR</option>
<optionvalue="10035">CONDE</option>
<optionvalue="10036">CORTGUAYABO</option>
<optionvalue="10037">DELICIAS</option>
<optionvalue="10038">DONBOSCO</option>
<optionvalue="10039">DOSCAMINOS12.47kV</option>
<optionvalue="10040">DOSCAMINOS4.8kV</option>
<optionvalue="10041">ELEGGUA</option>
<optionvalue="10042">ESCUELAMILITAR</option>
<optionvalue="10043">ESMERALDA</option>
<optionvalue="10044">FLORIDA</option>
<optionvalue="10045">FUERTO</option>
<optionvalue="10046">FUNCERES</option>
<optionvalue="10047">GRANADA</option>
<optionvalue="10048">GUAIRA</option>
<optionvalue="10049">GUAIRITA</option>
<optionvalue="10050">GUARATARO</option>
<optionvalue="10051">HOYOLASTAPIAS</option>
<optionvalue="10052">HUMBOLDT</option>
<optionvalue="10053">JUANMIRANDA</option>
<optionvalue="10054">JUNKO</option>
<optionvalue="10055">L.CARABALLO</option>
<optionvalue="10056">LASABANA</option>
<optionvalue="10057">LAGUNITA</option>
<optionvalue="10058">LONGAESPA_A</option>
<optionvalue="10059">LOSCARACAS</option>
<optionvalue="10060">MACARACUAY</option>
<optionvalue="10061">MARAPA</option>
<optionvalue="10062">MARE</option>
<optionvalue="10063">MARQUES</option>
<optionvalue="10064">MAURY</option>
<optionvalue="10065">MERCEDES</option>
<optionvalue="10066">MIRANDA</option>
<optionvalue="10067">MONTALBAN</option>
<optionvalue="10068">NORTE</option>
<optionvalue="10069">PALOVERDE</option>
<optionvalue="10070">PALOSGRANDES</option>
<optionvalue="10071">PARACOTOS</option>
<optionvalue="10072">PARAISO</option>
<optionvalue="10073">PETARE</option>
<optionvalue="10074">PICURE</option>
<optionvalue="10075">PINAR</option>
<optionvalue="10076">PINEDA</option>
<optionvalue="10077">PLACER</option>
<optionvalue="10078">PRADOSDELESTE</option>
<optionvalue="10079">PROPATRIA</option>
<optionvalue="10080">RINCON</option>
<optionvalue="10081">ROSAL12.47kV</option>
<optionvalue="10082">ROSAL4.8kV</option>
<optionvalue="10083">RUICES</option>
<optionvalue="10084">SABANAGRANDE</option>
<optionvalue="10085">SANAGUSTIN</option>
<optionvalue="10086">SANANTONIO</option>
<optionvalue="10087">SANBERNARDINO</option>
<optionvalue="10088">SANTACRUZ</option>
<optionvalue="10089">SANTAMONICA</option>
<optionvalue="10090">SANTAROSA</option>
<optionvalue="10091">SURESTE</option>
<optionvalue="10092">TACAGUA</option>
<optionvalue="10093">TAMANACO</option>
<optionvalue="10094">TODASANA</option>
<optionvalue="10095">TRAPICHITO12.47kV</option>
<optionvalue="10096">TRAPICHITO8.3kV</option>
<optionvalue="10097">TRINIDAD</option>
<optionvalue="10098">URBINA</option>
<optionvalue="10099">--YAGUARA--</option>

```
</select>

<selectclass="input-large"id="combo1"name="combo1"onChange="llamarCircuito()">
<optionvalue="0">--CIRCUITO--</option>
<optionvalue="9001">Circuito9001</option>
<optionvalue="9002">Circuito9002</option>
<optionvalue="9003">Circuito9003</option>
<optionvalue="9004">Circuito9004</option>
<optionvalue="9005">Circuito9005</option>
<optionvalue="9006">Circuito9006</option>
</select>
</ul>
</div><!--/.well-->
</div><!--/.fluid-container-->
<footer>
<p>©INDENEU.S.B.2013</p>
</footer>
</div><!--/.fluid-container-->

<!--Lejavascript
==========================================-->
<!--Placedattheendofthedocumentsothepagesloadfaster-->

<scriptsrc="${request.static_url('prueba3:static/jquery.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-transition.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-alert.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-modal.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-dropdown.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-scrollspy.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-tab.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-tooltip.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-popover.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-button.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-collapse.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-carousel.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-typeahead.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-combobox.js')}"></script>

</body></html>
```

Código de la plantilla kml_view1

```
<htmllang="es">
<head>
<metahttp-equiv="Content-Type"content="text/html;charset=UTF-8">
<metacharset="utf-8">
<title>FlujodeCargaINDENE,U.S.B.</title>
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<metaname="description"content="">
<metaname="author"content="">
<metahttp-equiv="Content-Type"content="text/html;charset=utf-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0">
<metaname="apple-mobile-web-app-capable"content="yes">
<linkhref="${request.static_url('prueba3:static/bootstrap.css')}"media="screen"rel="stylesheet"type="text/css">
<linkhref="${request.static_url('prueba3:static/bootstrap-combobox.css')}"media="screen"rel="stylesheet"type="text/css">
<linkrel="stylesheet"href="style.css"type="text/css">
<!--Lestyles-->
<linkhref="${request.static_url('prueba3:static/bootstrap.css')}"rel="stylesheet">
<styletype="text/css">
html,body{
padding-top:60px;
padding-bottom:40px;
height:100%;
}
#map{
width:100%;
height:80%;
```

```
border:1pxsolidblack;
}
.olPopupp{margin:0px;font-size:.9em;}
.olPopuph2{font-size:1.2em;}
.sidebar-nav{
padding:9px0;
}

@media(max-width:980px){
/*Enableuseoffloatednavbartext*/
.navbar-text.pull-right{
float:none;
padding-left:5px;
padding-right:5px;
}
}
</style>
<scriptsrc="http://www.openlayers.org/api/OpenLayers.js"></script>

<scripttype="text/javascript">
$(document).ready(function(){
$('.combobox').combobox();
});
varlon=5;
varlat=40;
varzoom=5;
varmap,select;
varoptions={
maxResolution:"auto",
minResolution:"auto",
numZoomLevels:50,
};
functioninit(){
map=newOpenLayers.Map('map',options);

varwms=newOpenLayers.Layer.OSM("OpenStreetMap",null,{
transitionEffect:"resize",
attribution:"&copy;<ahref='http://www.openstreetmap.org/copyright'>OpenStreetMap</a>contributors"
})

varsundials=newOpenLayers.Layer.Vector("KML",{
projection:map.displayProjection,
strategies:[newOpenLayers.Strategy.Fixed()],
protocol:newOpenLayers.Protocol.HTTP({
//url:"${request.static_url('prueba3:static/runpypower2.kml')}",
url:"/kml/${request.matchdict.uid}",
format:newOpenLayers.Format.KML({
extractStyles:true,
extractAttributes:true
})
})
});

map.addLayers([wms,sundials]);
select=newOpenLayers.Control.SelectFeature(sundials);
sundials.events.on({
"featureselected":onFeatureSelect,
"featureunselected":onFeatureUnselect,
"loadend":centrar

});

map.addControl(select);
select.activate();
//map.zoomToExtent(newOpenLayers.Bounds(wms.getExtent()))
//map.zoomToExtent(newOpenLayers.Bounds(68.774414,11.381836,123.662109,34.628906));
}

functioncentrar(event){
```

```
map.zoomToExtent(event.object.getDataExtent());
}
functionrefrescar(sundials){
sundials.loaded=false;
sundials.loaded=true;
//settingvisibilitytotrueforcesareloadofthelayer//
sundials.setVisibility(true);
//therefreshwillforceittogetthenewKMLdata//
sundials.setUrl("${request.static_url('prueba3:static/runpypower2.kml')}");

}
functiononPopupClose(evt){
select.unselectAll();
}
<!--
functiononFeatureSelect(event){
varfeature=event.feature;
//SinceKMLisuser-generated,donaiveprotectionagainst
//Javascript.
varcontent="<h2>"+feature.attributes.name+"</h2>"+feature.attributes.description;
if(content.search("<script")!=-1){
content="ContentcontainedJavascript!Escapedcontentbelow.<br>"+content.replace(/</g,"&lt;");
}
popup=newOpenLayers.Popup.FramedCloud("chicken",
feature.geometry.getBounds().getCenterLonLat(),
newOpenLayers.Size(100,100),
content,
null,true,onPopupClose);
feature.popup=popup;
map.addPopup(popup);
}
functiononFeatureUnselect(event){
varfeature=event.feature;
if(feature.popup){
map.removePopup(feature.popup);
feature.popup.destroy();
deletefeature.popup;
}
}
</script>
<scripttype="text/javascript">
functionreadfile(){
alert(document.getElementById('iframe').contentDocument.body.firstChild.innerHTML);
}
</script>
<linkhref="${request.static_url('prueba3:static/bootstrap-responsive.css')}">
<!--HTML5shim,forIE6-8supportofHTML5elements-->
<!--[ifltIE9]>
<scriptsrc="../assets/js/html5shiv.js"></script>
<![endif]-->
<!--Favandtouchicons-->
<linkrel="apple-touch-icon-precomposed"sizes="144x144"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-144-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"sizes="114x114"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-114-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"sizes="72x72"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-72-
precomposed.png">
<linkrel="apple-touch-icon-precomposed"href="http://twitter.github.io/bootstrap/assets/ico/apple-touch-icon-57-precomposed.png">
<linkrel="shortcuticon"href="http://twitter.github.io/bootstrap/assets/ico/favicon.png">
</head>
<divclass="container"style="text-align:center;">
<imgalign="center"src="${request.static_url('prueba3:static/cromoconlogos.jpg')}"alt="indene">
</div>
<br>
<br>
<bodyonload="init()">
<divclass="navbarnavbar-inversenavbar-fixed-top">
<divclass="navbar-inner">
<divclass="container-fluid">
```

```
<buttontype="button"class="btnbtn-navbar"data-toggle="collapse"data-target=".nav-collapse">
<spanclass="icon-bar"></span>
<spanclass="icon-bar"></span>
<spanclass="icon-bar"></span>
</button>
<aclass="brand"href="/">FlujodeCargaGIS</a>
<divclass="nav-collapsecollapse">
<ulclass="nav">
<liclass="active"><ahref="http://www.indene.usb.ve">Inicio</a></li>
<li><ahref="http://www.indene.usb.ve">Acercade</a></li>
<li><ahref="http://www.indene.usb.ve">Contacto</a></li>
</ul>
</div><!--/.nav-collapse-->
</div>
</div>
</div>
<aname="objetivo1"></a>
<divclass="well">
<h2align="center">DesarrollodeunaHerramientaComputacionaldeFlujodeCargaImplementadoenunaPlataformaG.I.S.</h2>
</div>
<br>
<tableclass="tabletable-stripedtable-bordered">
<thead>
<divclass="alertalert-info"align="center">
<iclass="icon-folder-open"></i>
<strong>ResultadosdelFlujodeCarga.</strong>
</div>
<divid="map"class="smallmap"></div>
<divid="docs"align="middle">
<br/>
<imgalign="center"src="${request.static_url('prueba3:static/leyenda.jpg')}"alt="indene">
<palign="center"><b>Leyenda</b></p>
<br/>

<divclass="alertalert-info"align="center">
<iclass="icon-folder-open"></i>
<strong>ResultadosdelFlujodeCarga,InformePyPower</strong>
</div>
<iframeid='iframe'src="/resultado/${request.matchdict.uid}"width="700"height="600"></iframe>
</div>
<br/>
<divclass="span10"align="center">
<hr>
<footer>
<p>©INDENEU.S.B.2013</p>
</footer>
</div><!--/.fluid-container-->
<!--Lejavascript
==================================================-->
<!--Placedattheendofthedocumentsothepagesloadfaster-->
<scriptsrc="${request.static_url('prueba3:static/jquery.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-transition.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-alert.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-modal.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-dropdown.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-scrollspy.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-tab.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-tooltip.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-popover.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-button.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-collapse.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-carousel.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-typeahead.js')}"></script>
<scriptsrc="${request.static_url('prueba3:static/bootstrap-combobox.js')}"></script>

</body>
</html>
```

Configuración archivo development.ini

```
###
# app configuration
# http://docs.pylonsproject.org/projects/pyramid/en/latest/narr/environment.html
###

[app:main]
use = egg:prueba3

pyramid.reload_templates = true
pyramid.debug_authorization = false
pyramid.debug_notfound = false
pyramid.debug_routematch = false
pyramid.default_locale_name = en
#pyramid.includes =
  #pyramid_debugtoolbar

# By default, the toolbar only appears for clients from IP addresses
# '127.0.0.1' and '::1'.
# debugtoolbar.hosts = 127.0.0.1 ::1

###
# wsgi server configuration
###

[server:main]
use = egg:waitress#main
host = 0.0.0.0
port = 6543

###
# logging configuration
# http://docs.pylonsproject.org/projects/pyramid/en/latest/narr/logging.html
###

[loggers]
keys = root, prueba3

[handlers]
keys = console

[formatters]
keys = generic

[logger_root]
level = INFO
handlers = console

[logger_prueba3]
level = DEBUG
handlers =
qualname = prueba3

[handler_console]
class = StreamHandler
args = (sys.stderr,)
level = NOTSET
formatter = generic

[formatter_generic]
format = %(asctime)s %(levelname)-5.5s [%(name)s][%(threadName)s] %(message)s
```

Configuración Archivo setup.py

```
import os

from setuptools import setup, find_packages
```

```
here = os.path.abspath(os.path.dirname(__file__))
README = open(os.path.join(here, 'README.txt')).read()
CHANGES = open(os.path.join(here, 'CHANGES.txt')).read()

requires = [
            'pyramid',
            'pyramid_debugtoolbar',
            'waitress',
            'pypower',
            'lxml',
            'pykml',
            ]

setup(name='prueba3',
version='0.0',
description='prueba3',
    long_description=README + '\n\n' + CHANGES,
classifiers=[
      "Programming Language :: Python",
      "Framework :: Pyramid",
      "Topic :: Internet :: WWW/HTTP",
      "Topic :: Internet :: WWW/HTTP :: WSGI :: Application",
      ],
author='',
    author_email='',
url='',
keywords='web pyramid pylons',
packages=find_packages(),
    include_package_data=True,
    zip_safe=False,
    install_requires=requires,
    tests_require=requires,
    test_suite="prueba3",
    entry_points="""\
    [paste.app_factory]
main = prueba3:main
    """,
    )
```

ANEXO E: Conversión de Datos del Modelo hacia PyPower.

Conversión de datos de la Base de Datos hacia formato PyPower.

```
import psycopg2
import itertools
from numpy import array
from numpy import arccos
from numpy import sin
from decimal import Decimal

def loadfromdb(m):

	# Connect to an existing database
	conn=psycopg2.connect('host=192.168.0.108 port=5432 dbname=bd_edc_02_daniel user=indene password=indene01');
	# Busco el primer nodo del circuito 1
	cur = conn.cursor();

	cur.execute("SELECT first_node FROM pps.circuit WHERE id = '"+str(m)+"';");
	primer_nodo = cur.fetchall();

	cur.execute("SELECT nvl_tensi FROM pps.circuit_factors WHERE circuit_id = '"+str(m)+"';");
	voltaje_base = cur.fetchall();

	cur.execute("SELECT fp FROM pps.circuit_factors WHERE circuit_id = '"+str(m)+"';");
	factor_potencia = cur.fetchall();

	cur.execute("SELECT fu FROM pps.circuit_factors WHERE circuit_id = '"+str(m)+"';");

	factor_utili = cur.fetchall();

	sql_query = 'SELECT id,name,kv,cargas_kva FROM pps.node WHERE circuit_id = \''+str(m)+'\'
	cur.execute(sql_query);
	nodes_id_tuple = cur.fetchall();
	i=0;
	nodes_id_list = [];
	mva_base=0;
	for q in nodes_id_tuple:
		nodes_id_list.append(nodes_id_tuple[i][0]);
		mva_base=mva_base+nodes_id_tuple[i][3];
		i=i+1;


	mva_base=float(mva_base)/1000;

	# if mva_base == 0:
		# mva_base=1;

	sql_query = 'SELECT id,node0_id, node1_id, elem, distance,r1,x1,r0,x0,amps,amps_emergency FROM pps.branch WHERE
(node0_id IN (' + ','.join(map(str, nodes_id_list)) + ') OR node1_id IN (' + ','.join(map(str, nodes_id_list)) + ')) AND node0_id <>
node1_id'
	cur.execute(sql_query);
	branches_nodos_tuple = cur.fetchall();
	branches_nodos_list = list(branches_nodos_tuple);

	nodes_id_list = [];
	nodes_id_list = list(nodes_id_tuple);


	## bus data
	# bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
	nodos = [];
```

```
	i=0;

	for w in nodes_id_list:
		if (float(nodes_id_list[i][0]) != float(primer_nodo[0][0])):

	nodos.append([nodes_id_list[i][0],1,(float(nodes_id_list[i][3])*float(factor_potencia[0][0])*float(factor_utili[0][0]))/1000,(flo
at(nodes_id_list[i][3])*(sin(arccos(float(factor_potencia[0][0])))*float(factor_utili[0][0]))/1000),0,0,1,1,0,(float(voltaje_base[0][0])),1,1.1
,0.9]);

		if (float(nodes_id_list[i][0]) == float(primer_nodo[0][0])):

	nodos.append([nodes_id_list[i][0],3,(float(nodes_id_list[i][3])*float(factor_potencia[0][0])*float(factor_utili[0][0]))/1000,(flo
at(nodes_id_list[i][3])*(sin(arccos(float(factor_potencia[0][0])))*float(factor_utili[0][0]))/1000),0,0,1,1,0,(float(voltaje_base[0][0])),1,1.1
,0.9]);
		i=i+1;

	zbase = (float(voltaje_base[0][0])**2/(mva_base));

	ramas= []
	i=0;
	## branch data
	#fbus, tbus, r, x, b, rateA, rateB, rateC, ratio, angle, status, angmin, angmax
	for ww in branches_nodos_list:

		if ((branches_nodos_list[i][3] == 'L') or (branches_nodos_list[i][3] == 'S')):
			# ramas.append([branches_nodos_list[i][1], branches_nodos_list[i][2],
float('{:.15f}'.format((float(branches_nodos_list[i][4])*(float(branches_nodos_list[i][5])/zbase)))),
float('{:.15f}'.format(float(branches_nodos_list[i][4])*(float(branches_nodos_list[i][6])/zbase))), 0, mva_base*2.5, mva_base*2.5,
mva_base*2.5, 0, 0, 1, -360, 360]);
			ramas.append([branches_nodos_list[i][1], branches_nodos_list[i][2],
float('{:.10f}'.format((float(branches_nodos_list[i][4])*(float(branches_nodos_list[i][5])))))/zbase,
float('{:.10f}'.format(float(branches_nodos_list[i][4])*(float(branches_nodos_list[i][6]))))/zbase, 0, mva_base*2.5, mva_base*2.5,
mva_base*2.5, 0, 0, 1, -360, 360]);

		if (branches_nodos_list[i][3] == 'T'):
			ramas.append([branches_nodos_list[i][1], branches_nodos_list[i][2],
float('{:.10f}'.format(float(branches_nodos_list[i][5]))), float('{:.10f}'.format(float(branches_nodos_list[i][6]))), 0, mva_base*2.5,
mva_base*2.5, mva_base*2.5, 0, 0, 1, -360, 360]);

		i=i+1;

	a=0;
	nodosborrar=[];
	largoramas=len(ramas);

	for wwww in nodos:
		i=0;
		contador=0;
		for www in branches_nodos_list:

			if (float(nodos[a][0]) == float(ramas[i][0])) or ((float(nodos[a][0]) == float(ramas[i][1]))) :
				contador = contador+1;
			if (contador > 0):
				break
			if (i == (largoramas-1)):
				nodosborrar.append(nodos[a]);

			i=i+1;
		a=a+1;

	i=0;
	for wwwww in nodosborrar:
		nodos.remove(nodosborrar[i]);
		i=i+1;


	ppc = {"version": '2'}
	##----- Power Flow Data -----##
	## system MVA base
```

```
            ppc["baseMVA"] = (mva_base);

            ppc["bus"] = array(nodos);
            ## generator data
            # bus, Pg, Qg, Qmax, Qmin, Vg, mBase, status, Pmax, Pmin, Pc1, Pc2,
            # Qc1min, Qc1max, Qc2min, Qc2max, ramp_agc, ramp_10, ramp_30, ramp_q, apf
            ppc["gen"] = array([[primer_nodo[0][0], 0, 0, (mva_base), -(mva_base), 1, (mva_base), 1, mva_base*2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]]);

            ppc["branch"] = array(ramas);

            return ppc
```

ANEXO F: Kersting, Capítulo 4, Sección 4.2.

Ejemplo de donde se tomó el dato de la impedancia de las líneas en los circuitos (Kersting, Capítulo 4, Sección 4.2)

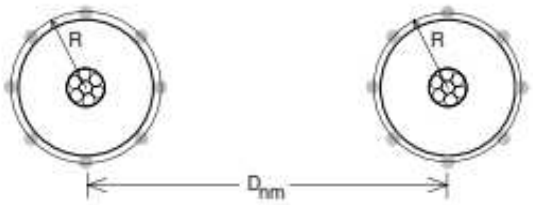


FIGURE 4.10
Distances between concentric neutral cables.

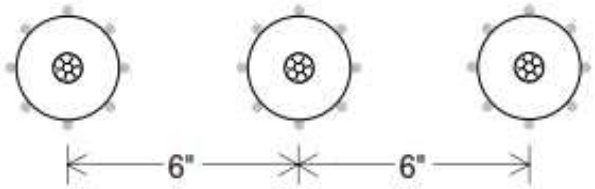


FIGURE 4.11
Three-phase concentric neutral cable spacing.

radius R, and therefore it may be assumed that D_{ij} in Equation 4.84 is equal to D_{nn}. For cables in conduit that assumption is not valid.

In applying the modified Carson's equations, the numbering of conductors and neutrals is important. For example, a three-phase underground circuit with an additional neutral conductor must be numbered as:

1 = phase conductor #1
2 = phase conductor #2
3 = phase conductor #3
4 = neutral of conductor #1
5 = neutral of conductor #2
6 = neutral of conductor #3
7 = additional neutral conductor (if present)

Example 4.2

Three concentric neutral cables are buried in a trench with spacings as shown in Figure 4.11. The cables are 15 kV, 250,000 CM stranded all-aluminum with 13 strands of #14 annealed, coated copper wires (1/3 neutral). The outside diameter of the cable over the neutral strands is 1.29 inches (Appendix B). Determine the phase impedance matrix and the sequence impedance matrix.

© 2002 by CRC Press LLC

SOLUTION

The data for the phase conductor and neutral strands from a conductor data table (Appendix A) are

250,000 AA phase conductor:
- GMR = 0.0171 ft.
- Diameter = 0.567 inches
- Resistance = 0.4100 Ω/mile

#14 copper neutral strands:
- GMR_s = 0.00208 ft.
- Resistance = 14.8722 Ω/mile

Diameter (d_s) = 0.0641 inches

The radius of the circle passing through the center of the strands (Equation 4.82) is

$$R = \frac{d_{cd} - d_s}{24} = 0.0511 \text{ ft}$$

The equivalent GMR of the concentric neutral is computed by:

$$GMR_{cn} = \sqrt[k]{GMR_s \cdot k \cdot R^{k-1}} = \sqrt[13]{0.00208 \cdot 13 \cdot 0.0511^{13-1}} = 0.0486 \text{ ft}$$

The equivalent resistance of the concentric neutral is

$$r_{cn} = \frac{r_s}{k} = \frac{14.8722}{13} = 1.1438 \ \Omega/\text{mile}$$

The phase conductors are numbered 1, 2, and 3. The concentric neutrals are numbered 4, 5, and 6. The conductor-to-conductor and concentric neutral-to-concentric neutral spacings are

$$D_{12} = D_{21} = D_{45} = D_{54} = 0.5 \text{ ft}$$
$$D_{23} = D_{32} = D_{56} = D_{65} = 0.5 \text{ ft}$$
$$D_{31} = D_{13} = D_{64} = D_{46} = 1.0 \text{ ft}$$

The spacings between conductors and their concentric neutrals are

$$D_{14} = D_{25} = D_{36} = R = 0.0511 \text{ ft}$$

Since the radius R is much smaller than the spacings between cables, the distances between concentric neutrals and adjacent phase conductors are

© 2002 by CRC Press LLC

just the center-to-center distances between conductors:

$$D_{15} = D_{51} = 0.5$$
$$D_{26} = D_{62} = 0.5 \text{ ft}$$
$$D_{61} = D_{16} = 1.0.$$

The self impedance for the cable in position 1 is

$$\hat{z}_{11} = 0.0953 + 0.41 + j0.12134 \cdot \left(\ln\frac{1}{0.0171} + 7.93402\right)$$
$$= 0.5053 + j1.4564 \ \Omega/\text{mile}$$

The self impedance for the concentric neutral for Cable #1 is

$$\hat{z}_{44} = 0.0953 + 1.144 + j0.12134 \cdot \left(\ln\frac{1}{0.0486} + 7.93402\right)$$
$$= 1.2393 + j1.3296 \ \Omega/\text{mile}$$

The mutual impedance between Cable #1 and Cable #2 is

$$\hat{z}_{12} = 0.0953 + j0.12134 \cdot \left(\ln\frac{1}{0.5} + 7.93402\right) = 0.0953 + j1.0468 \ \Omega/\text{mile}$$

The mutual impedance between Cable #1 and its concentric neutral is

$$\hat{z}_{14} = 0.0953 + j0.12134 \cdot \left(\ln\frac{1}{0.0511} + 7.93402\right) = 0.0953 + j1.3236 \ \Omega/\text{mile}$$

The mutual impedance between the concentric neutral of Cable #1 and the concentric neutral of Cable #2 is

$$\hat{z}_{45} = 0.0953 + j0.12134 \cdot \left(\ln\frac{1}{0.5} + 7.93402\right) = 0.0953 + j1.0468 \ \Omega/\text{mile}$$

Continuing the application of the modified Carson's equations results in a 6×6 primitive impedance matrix. This matrix in partitioned (Equation 4.33) form is

$$[\hat{z}_{ij}] = \begin{bmatrix} .5053 + j1.4564 & .0953 + j1.0468 & .0953 + j.9627 \\ .0953 + j1.0468 & .5053 + j1.4564 & .0953 + j1.0468 \\ .0953 + j.9627 & .0953 + j1.0468 & .5053 + j1.4564 \end{bmatrix} \Omega/\text{mile}$$

$$[\hat{z}_{ss}] = \begin{bmatrix} .0953 + j1.3236 & .0953 + j1.0468 & .0953 + j.9627 \\ .0953 + j1.0468 & .0953 + j1.3236 & .0953 + j1.0468 \\ .0953 + j.9627 & .0953 + j1.0468 & .0953 + j1.3236 \end{bmatrix} \Omega/\text{mile}$$

© 2002 by CRC Press LLC

$$[\hat{z}_{nj}] = [\hat{z}_{in}]^T$$

$$[\hat{z}_{nn}] = \begin{bmatrix} 1.2391 + j1.3296 & .0953 + j1.0468 & .0953 + j.9627 \\ .0953 + j1.0468 & 1.2391 + j1.3296 & .0953 + j1.0468 \\ .0953 + j.9627 & .0953 + j1.0468 & 1.2391 + j1.3296 \end{bmatrix} \Omega/\text{mile}$$

Using the Kron reduction results in the phase impedance matrix:

$$[z_{abc}] = [\hat{z}_{ij}] - [\hat{z}_{in}] \cdot [\hat{z}_{nn}]^{-1} \cdot [\hat{z}_{nj}]$$

$$[z_{abc}] = \begin{bmatrix} 0.7981 + j0.4463 & 0.3191 + j0.0328 & 0.2849 - j0.0143 \\ 0.3191 + j0.0328 & 0.7891 + j0.4041 & 0.3191 + j0.0328 \\ 0.2849 - j0.0143 & 0.3191 + j0.0328 & 0.7981 + j0.4463 \end{bmatrix} \Omega/\text{mile}$$

The sequence impedance matrix for the concentric neutral three-phase line is determined using Equation 4.66:

$$[z_{012}] = [A_s]^{-1} \cdot [z_{abc}] \cdot [A_s]$$

$$[z_{012}] = \begin{bmatrix} 1.4105 + j0.4665 & -0.0028 - j0.0081 & -0.0056 + j0.0065 \\ -0.0056 + j0.0065 & 0.4874 + j0.4151 & -0.0264 + j0.0451 \\ -0.0028 - j0.0081 & 0.0523 + j0.0003 & 0.4874 + j0.4151 \end{bmatrix} \Omega/\text{mile}$$

4.2.2 Tape-Shielded Cables

Figure 4.12 shows a simple detail of a tape-shielded cable. The cable consists of a central phase conductor covered by a thin layer of nonmetallic semiconducting screen to which is bonded the insulating material. The insulation is covered by a semiconducting insulation screen. The shield is bare copper

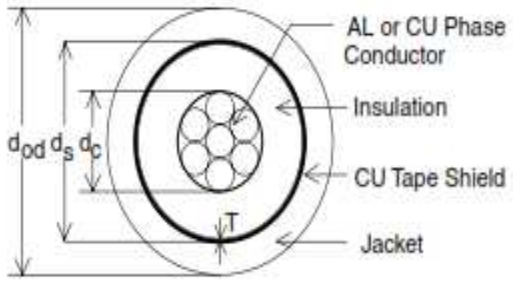


FIGURE 4.12
Tape-shielded cable.

© 2002 by CRC Press LLC

ANEXO G: Resultados de los Flujos de Carga, Circuitos 9001-9005.

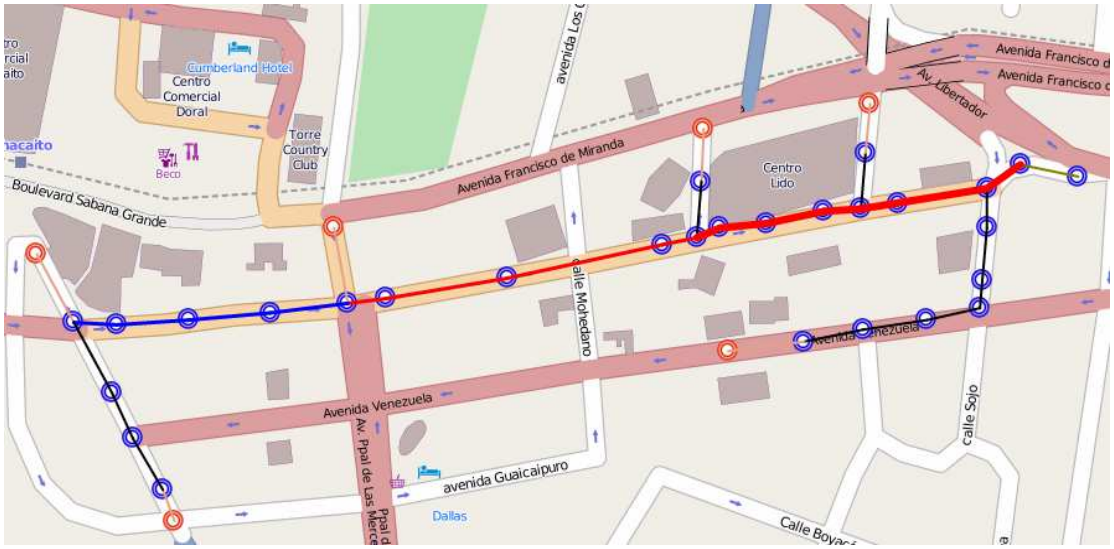
Circuito 9001

Resultados del Flujo de Carga:

```
Converged in 0.08 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    --------------------  -------------  -----------------
Buses              34    Total Gen Capacity       8.9          -4.5 to 4.5
Generators          1    On-line Capacity         8.9          -4.5 to 4.5
Committed Gens      1    Generation (actual)      4.1               2.1
Loads               6    Load                     4.0               1.9
  Fixed             6      Fixed                  4.0               1.9
  Dispatchable      0      Dispatchable           0.0 of 0.0        0.0
Shunts              0    Shunt (inj)              0.0               0.0
Branches           33    Losses (I^2 * Z)         0.12              0.11
Transformers        0    Branch Charging (inj)     -                0.0
Inter-ties          0    Total Inter-tie Flow     0.0               0.0
Areas               1

                          Minimum                      Maximum
                  -------------------------  ---------------------------------
Voltage Magnitude   0.948 p.u. @ bus 34         1.000 p.u. @ bus 1
Voltage Angle      -0.88 deg   @ bus 34         0.00 deg   @ bus 1
P Losses (I^2*R)             -                 0.01 MW    @ line 2-3
Q Losses (I^2*X)             -                 0.01 MVAr  @ line 2-3

================================================================================
|     Bus Data                                                                 |
================================================================================
 Bus      Voltage          Generation             Load
  #   Mag(pu) Ang(deg)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
----- ------- --------  --------  --------  --------  --------
    1  1.000    0.000       -         -         -         -
    2  0.996   -0.055       -         -         -         -
    3  0.992   -0.110       -         -         -         -
    4  0.992   -0.119       -         -         -         -
    5  0.991   -0.127       -         -         -         -
    7  0.990   -0.143       -         -         -         -
    8  0.990   -0.151       -         -         -         -
    9  0.989   -0.160       -         -         -         -
   11  0.989   -0.158       -         -         -         -
   12  0.986   -0.206       -         -         -         -
   13  0.985   -0.217       -         -         -         -
   15  0.983   -0.244       -         -         -         -
   16  0.981   -0.281       -         -         -         -
   17  0.976   -0.350       -         -         -         -
   18  0.974   -0.380       -         -         -         -
   19  0.973   -0.394       -         -         -         -
   21  0.971   -0.416       -         -         -         -
   22  0.968   -0.467       -         -         -         -
   23  0.964   -0.534       -         -         -         -
   26  0.957   -0.626       -         -         -         -
   30  0.952   -0.809       -         -        0.41      0.20
   28  0.954   -0.672       -         -         -         -
   29  0.954   -0.686       -         -         -         -
   31  0.953   -0.695       -         -         -         -
   32  0.951   -0.722       -         -         -         -
   33  0.951   -0.732       -         -         -         -
   34  0.948   -0.884       -         -        0.49      0.24
   14  0.982   -0.436       -         -        0.77      0.37
   10  0.986   -0.325       -         -        0.58      0.28
   25  0.956   -0.843       -         -        0.85      0.41
   27  0.956   -0.651       -         -         -         -
   20  0.969   -0.658       -         -        0.90      0.44
    6  0.991   -0.135       -         -         -         -
   24  0.960   -0.586       -         -         -         -
                          --------  --------  --------  --------
               Total:       4.12      2.05      4.00      1.94

================================================================================
|     Branch Data                                                              |
================================================================================
Brnch   From   To    From Bus Injection   To Bus Injection     Loss (I^2 * Z)
  #     Bus    Bus    P (MW)   Q (MVAr)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
-----  -----  -----  --------  --------  --------  --------  --------  --------
   0     34     33     -0.49     -0.24      0.50      0.24     0.001      0.00
   1     29     30      0.41      0.20     -0.40     -0.20     0.000      0.00
   2     29     28     -0.90     -0.44      0.90      0.44     0.001      0.00
   3     24     25      0.86      0.42     -0.85     -0.41     0.002      0.01
   4     19     20      0.90      0.44     -0.90     -0.44     0.002      0.01
```

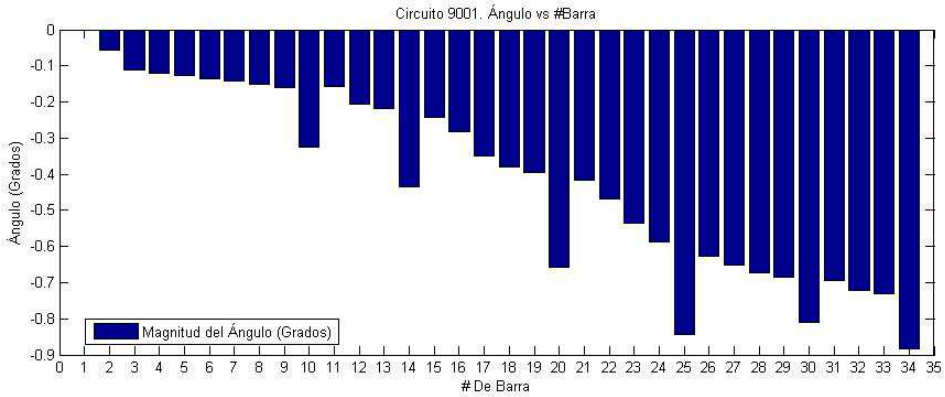
5	16	15	-2.71	-1.34	2.71	1.34	0.006	0.01
6	15	12	-2.71	-1.34	2.72	1.35	0.006	0.01
7	12	13	0.77	0.38	-0.77	-0.37	0.000	0.00
8	13	14	0.77	0.37	-0.76	-0.37	0.001	0.00
9	12	11	-3.48	-1.73	3.49	1.73	0.010	0.01
10	11	3	-3.49	-1.73	3.50	1.74	0.010	0.01
11	2	1	-4.11	-2.04	4.12	2.05	0.014	0.01
12	3	4	0.59	0.29	-0.59	-0.29	0.000	0.00
13	4	5	0.59	0.29	-0.59	-0.29	0.000	0.00
14	5	6	0.59	0.29	-0.59	-0.29	0.000	0.00
15	6	7	0.59	0.29	-0.59	-0.29	0.000	0.00
16	7	8	0.59	0.29	-0.59	-0.29	0.000	0.00
17	8	9	0.59	0.29	-0.59	-0.29	0.000	0.00
18	9	10	0.59	0.29	-0.58	-0.28	0.001	0.00
19	24	23	-1.76	-0.86	1.77	0.87	0.006	0.00
20	2	3	4.11	2.04	-4.09	-2.03	0.014	0.01
21	31	29	-0.50	-0.24	0.50	0.24	0.000	0.00
22	33	32	-0.50	-0.24	0.50	0.24	0.000	0.00
23	21	18	-1.78	-0.88	1.79	0.88	0.004	0.00
24	28	27	-0.90	-0.44	0.90	0.44	0.001	0.00
25	27	26	-0.90	-0.44	0.91	0.44	0.001	0.00
26	18	19	0.90	0.44	-0.90	-0.44	0.001	0.00
27	18	17	-2.69	-1.33	2.69	1.33	0.005	0.00
28	17	16	-2.69	-1.33	2.71	1.34	0.011	0.01
29	26	24	-0.91	-0.44	0.91	0.44	0.002	0.00
30	23	22	-1.77	-0.87	1.78	0.88	0.007	0.01
31	32	31	-0.50	-0.24	0.50	0.24	0.001	0.00
32	22	21	-1.78	-0.88	1.78	0.88	0.006	0.00
						Total:	0.115	0.11

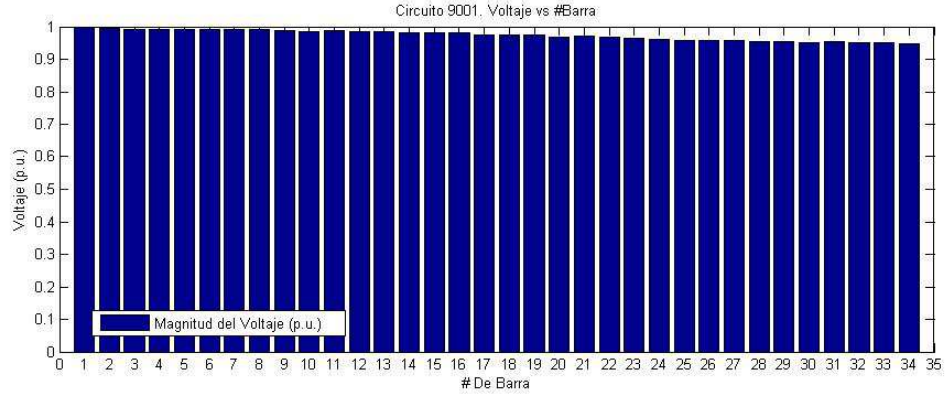
Mapa referenciado geográficamente del circuito:

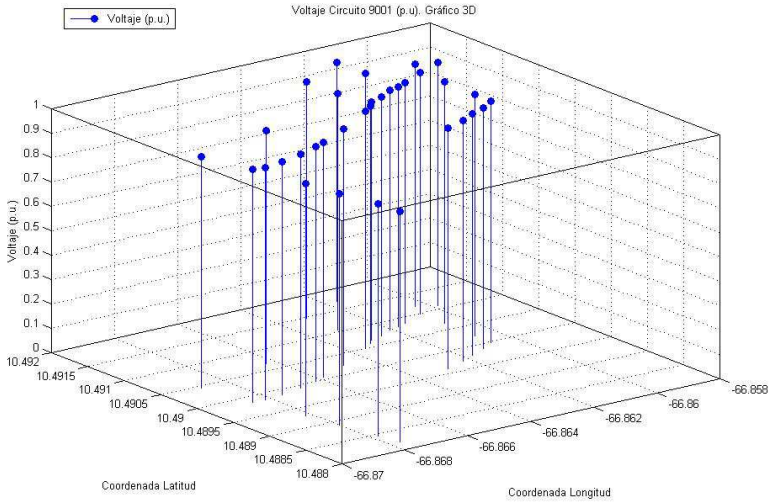


Circuito 9001 representado en OpenStreetMaps.

Gráficas de Voltajes, Ángulos y Gráfico 3D con respecto a las barras del circuito:







Circuito 9002

Resultados del Flujo de Carga:

```
Converged in 0.11 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses             37     Total Gen Capacity       7.0          -3.5 to 3.5
Generators         1     On-line Capacity         7.0          -3.5 to 3.5
Committed Gens     1     Generation (actual)      3.2               1.6
Loads              4     Load                     3.1               1.5
```

Fixed	4	Fixed	3.1	1.5
Dispatchable	0	Dispatchable	0.0 of 0.0	0.0
Shunts	0	Shunt (inj)	0.0	0.0
Branches	36	Losses (I^2 * Z)	0.09	0.10
Transformers	0	Branch Charging (inj)	-	0.0
Inter-ties	0	Total Inter-tie Flow	0.0	0.0
Areas	1			

	Minimum	Maximum
Voltage Magnitude	0.901 p.u. @ bus 3702	1.000 p.u. @ bus 102
Voltage Angle	-1.63 deg @ bus 3702	0.00 deg @ bus 102
P Losses (I^2*R)	-	0.04 MW @ line 2802-2902
Q Losses (I^2*X)	-	0.03 MVAr @ line 2802-2902

Bus Data

Bus #	Voltage Mag(pu)	Voltage Ang(deg)	Generation P (MW)	Generation Q (MVAr)	Load P (MW)	Load Q (MVAr)
102	1.000	0.000	-	-	-	-
202	0.997	-0.043	-	-	-	-
302	0.995	-0.066	-	-	-	-
402	0.994	-0.090	-	-	-	-
502	0.992	-0.114	-	-	-	-
602	0.990	-0.138	-	-	-	-
702	0.989	-0.162	-	-	-	-
902	0.996	-0.062	-	-	-	-
1002	0.994	-0.081	-	-	-	-
1102	0.993	-0.100	-	-	-	-
1202	0.991	-0.120	-	-	-	-
1302	0.990	-0.139	-	-	-	-
1402	0.989	-0.159	-	-	-	-
1502	0.987	-0.178	-	-	-	-
1602	0.987	-0.190	-	-	-	-
1802	0.987	-0.186	-	-	-	-
1902	0.986	-0.193	-	-	-	-
2002	0.986	-0.201	-	-	-	-
2102	0.985	-0.208	-	-	-	-
2202	0.985	-0.216	-	-	-	-
2302	0.984	-0.224	-	-	-	-
2402	0.983	-0.231	-	-	-	-
2502	0.983	-0.239	-	-	-	-
2602	0.982	-0.247	-	-	-	-
2702	0.982	-0.254	-	-	-	-
2802	0.981	-0.262	-	-	-	-
3002	0.907	-1.364	-	-	-	-
3102	0.907	-1.372	-	-	-	-
3202	0.906	-1.380	-	-	-	-
3302	0.906	-1.388	-	-	-	-
3402	0.905	-1.397	-	-	-	-
3502	0.905	-1.405	-	-	-	-
3602	0.904	-1.413	-	-	-	-
1702	0.982	-0.500	-	-	0.85	0.41
2902	0.908	-1.355	-	-	0.04	0.02
802	0.978	-0.799	-	-	1.75	0.85
3702	0.901	-1.627	-	-	0.49	0.24
		Total:	3.24	1.63	3.15	1.53

Branch Data

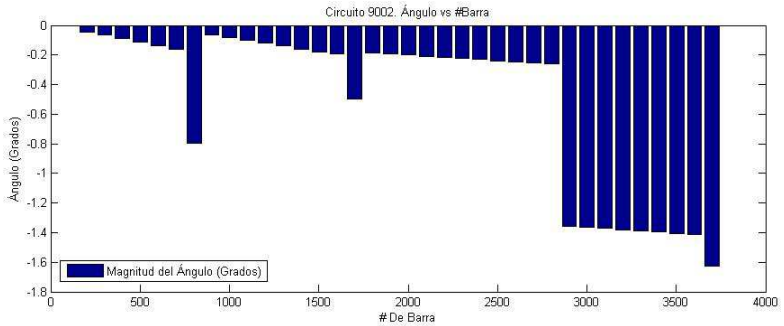
Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
0	102	202	3.24	1.63	-3.23	-1.62	0.009	0.01
1	202	302	1.78	0.89	-1.77	-0.89	0.003	0.00
2	302	402	1.77	0.89	-1.77	-0.89	0.003	0.00
3	402	502	1.77	0.89	-1.77	-0.88	0.003	0.00
4	502	602	1.77	0.88	-1.77	-0.88	0.003	0.00
5	602	702	1.77	0.88	-1.76	-0.88	0.003	0.00
6	702	802	1.76	0.88	-1.76	-0.85	0.009	0.03
7	202	902	1.45	0.73	-1.45	-0.73	0.002	0.00
8	902	1002	1.45	0.73	-1.45	-0.73	0.002	0.00
9	1002	1102	1.45	0.73	-1.45	-0.73	0.002	0.00
10	1102	1202	1.45	0.73	-1.45	-0.73	0.002	0.00
11	1202	1302	1.45	0.73	-1.45	-0.73	0.002	0.00
12	1302	1402	1.45	0.73	-1.44	-0.72	0.002	0.00
13	1402	1502	1.44	0.72	-1.44	-0.72	0.002	0.00
14	1502	1602	0.86	0.42	-0.86	-0.42	0.001	0.00
15	1602	1702	0.86	0.42	-0.85	-0.41	0.002	0.01
16	1502	1802	0.58	0.30	-0.58	-0.30	0.000	0.00
17	1802	1902	0.58	0.30	-0.58	-0.30	0.000	0.00
18	1902	2002	0.58	0.30	-0.58	-0.30	0.000	0.00
19	2002	2102	0.58	0.30	-0.58	-0.30	0.000	0.00
20	2102	2202	0.58	0.30	-0.58	-0.30	0.000	0.00
21	2202	2302	0.58	0.30	-0.58	-0.30	0.000	0.00
22	2302	2402	0.58	0.30	-0.58	-0.30	0.000	0.00
23	2402	2502	0.58	0.30	-0.58	-0.30	0.000	0.00
24	2502	2602	0.58	0.30	-0.58	-0.30	0.000	0.00
25	2602	2702	0.58	0.30	-0.58	-0.30	0.000	0.00

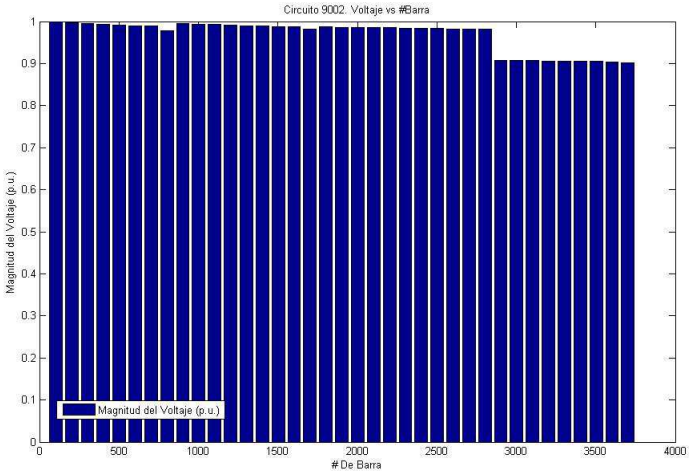
26	2702	2802	0.58	0.30	-0.58	-0.30	0.000	0.00
27	2902	3002	0.50	0.24	-0.50	-0.24	0.000	0.00
28	3002	3102	0.50	0.24	-0.50	-0.24	0.000	0.00
29	3102	3202	0.50	0.24	-0.50	-0.24	0.000	0.00
30	3202	3302	0.50	0.24	-0.50	-0.24	0.000	0.00
31	3302	3402	0.50	0.24	-0.50	-0.24	0.000	0.00
32	3402	3502	0.50	0.24	-0.50	-0.24	0.000	0.00
33	3502	3602	0.50	0.24	-0.50	-0.24	0.000	0.00
34	3602	3702	0.50	0.24	-0.49	-0.24	0.001	0.00
35	2802	2902	0.58	0.30	-0.54	-0.27	0.038	0.03
						Total:	0.089	0.10

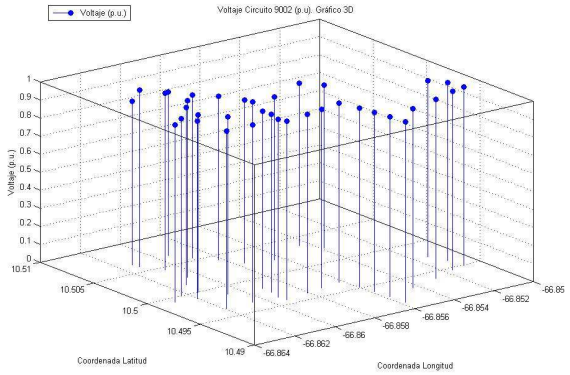
Mapa referenciado geográficamente del circuito:

Circuito 9002 representado en OpenStreetMaps.

Gráficas de Voltajes, Ángulos y Gráfico 3D con respecto a las barras del circuito:







Circuito 9003

Resultados del Flujo de Carga:

```
Converged in 0.16 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses                32  Total Gen Capacity       6.1          -3.0 to 3.0
Generators            1  On-line Capacity         6.1          -3.0 to 3.0
Committed Gens        1  Generation (actual)      2.8               1.4
Loads                 4  Load                     2.7               1.3
```

Fixed	4	Fixed	2.7	1.3
Dispatchable	0	Dispatchable	0.0 of 0.0	0.0
Shunts	0	Shunt (inj)	0.0	0.0
Branches	31	Losses (I^2 * Z)	0.07	0.08
Transformers	0	Branch Charging (inj)	-	0.0
Inter-ties	0	Total Inter-tie Flow	0.0	0.0
Areas	1			

	Minimum	Maximum
Voltage Magnitude	0.959 p.u. @ bus 32003	1.000 p.u. @ bus 1003
Voltage Angle	-0.80 deg @ bus 28003	0.00 deg @ bus 1003
P Losses (I^2*R)	-	0.01 MW @ line 5003-6003
Q Losses (I^2*X)	-	0.01 MVAr @ line 7003-8003

Bus Data

Bus #	Voltage Mag(pu)	Voltage Ang(deg)	Generation P (MW)	Generation Q (MVAr)	Load P (MW)	Load Q (MVAr)
1003	1.000	0.000	-	-	-	-
2003	0.997	-0.037	-	-	-	-
3003	0.995	-0.075	-	-	-	-
4003	0.992	-0.113	-	-	-	-
5003	0.989	-0.151	-	-	-	-
6003	0.987	-0.189	-	-	-	-
7003	0.986	-0.201	-	-	-	-
9003	0.985	-0.216	-	-	-	-
10003	0.983	-0.243	-	-	-	-
11003	0.981	-0.269	-	-	-	-
12003	0.980	-0.296	-	-	-	-
13003	0.978	-0.323	-	-	-	-
14003	0.977	-0.333	-	-	-	-
16003	0.977	-0.340	-	-	-	-
17003	0.975	-0.358	-	-	-	-
18003	0.974	-0.375	-	-	-	-
19003	0.973	-0.393	-	-	-	-
20003	0.972	-0.410	-	-	-	-
21003	0.971	-0.428	-	-	-	-
22003	0.969	-0.446	-	-	-	-
23003	0.968	-0.463	-	-	-	-
24003	0.967	-0.481	-	-	-	-
25003	0.966	-0.499	-	-	-	-
26003	0.965	-0.516	-	-	-	-
27003	0.964	-0.526	-	-	-	-
29003	0.964	-0.525	-	-	-	-
30003	0.964	-0.534	-	-	-	-
31003	0.963	-0.542	-	-	-	-
8003	0.980	-0.558	-	-	0.85	0.41
28003	0.960	-0.800	-	-	0.63	0.31
32003	0.959	-0.798	-	-	0.58	0.28
15003	0.973	-0.619	-	-	0.68	0.33
		Total:	2.82	1.41	2.75	1.33

Branch Data

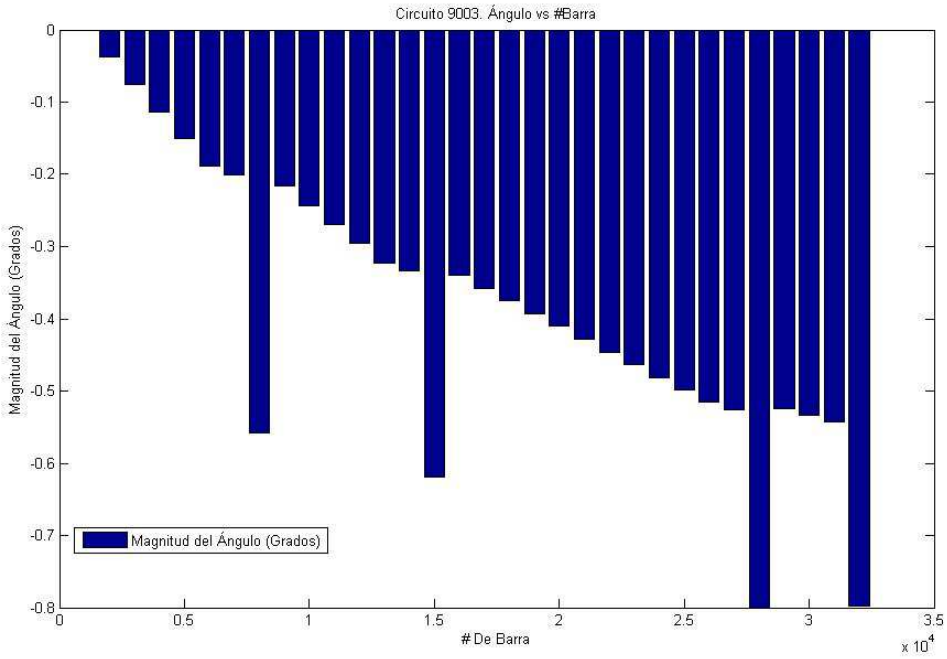
Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
0	1003	2003	2.82	1.41	-2.81	-1.40	0.007	0.01
1	2003	3003	2.81	1.40	-2.80	-1.39	0.007	0.01
2	3003	4003	2.80	1.39	-2.80	-1.39	0.007	0.01
3	4003	5003	2.80	1.39	-2.79	-1.38	0.007	0.01
4	5003	6003	2.79	1.38	-2.78	-1.38	0.007	0.01
5	6003	7003	0.86	0.42	-0.86	-0.42	0.001	0.00
6	7003	8003	0.86	0.42	-0.85	-0.41	0.002	0.01
7	6003	9003	1.93	0.96	-1.92	-0.95	0.003	0.00
8	9003	10003	1.92	0.95	-1.92	-0.95	0.003	0.00
9	10003	11003	1.92	0.95	-1.92	-0.95	0.003	0.00
10	11003	12003	1.92	0.95	-1.91	-0.94	0.003	0.00
11	12003	13003	1.91	0.94	-1.91	-0.94	0.003	0.00
12	13003	14003	0.68	0.33	-0.68	-0.33	0.000	0.00
13	14003	15003	0.68	0.33	-0.67	-0.33	0.001	0.00
14	13003	16003	1.23	0.61	-1.23	-0.61	0.001	0.00
15	16003	17003	1.23	0.61	-1.23	-0.61	0.001	0.00
16	17003	18003	1.23	0.61	-1.23	-0.61	0.001	0.00
17	18003	19003	1.23	0.61	-1.23	-0.61	0.001	0.00
18	19003	20003	1.23	0.61	-1.23	-0.60	0.001	0.00
19	20003	21003	1.23	0.60	-1.23	-0.60	0.001	0.00
20	21003	22003	1.23	0.60	-1.22	-0.60	0.001	0.00
21	22003	23003	1.22	0.60	-1.22	-0.60	0.001	0.00
22	23003	24003	1.22	0.60	-1.22	-0.60	0.001	0.00
23	24003	25003	1.22	0.60	-1.22	-0.60	0.001	0.00
24	25003	26003	1.22	0.60	-1.22	-0.60	0.001	0.00
25	26003	27003	0.63	0.31	-0.63	-0.31	0.000	0.00
26	27003	28003	0.63	0.31	-0.63	-0.31	0.001	0.00
27	26003	29003	0.59	0.29	-0.59	-0.29	0.000	0.00
28	29003	30003	0.59	0.29	-0.59	-0.29	0.000	0.00
29	30003	31003	0.59	0.29	-0.59	-0.29	0.000	0.00
30	31003	32003	0.59	0.29	-0.58	-0.28	0.001	0.00

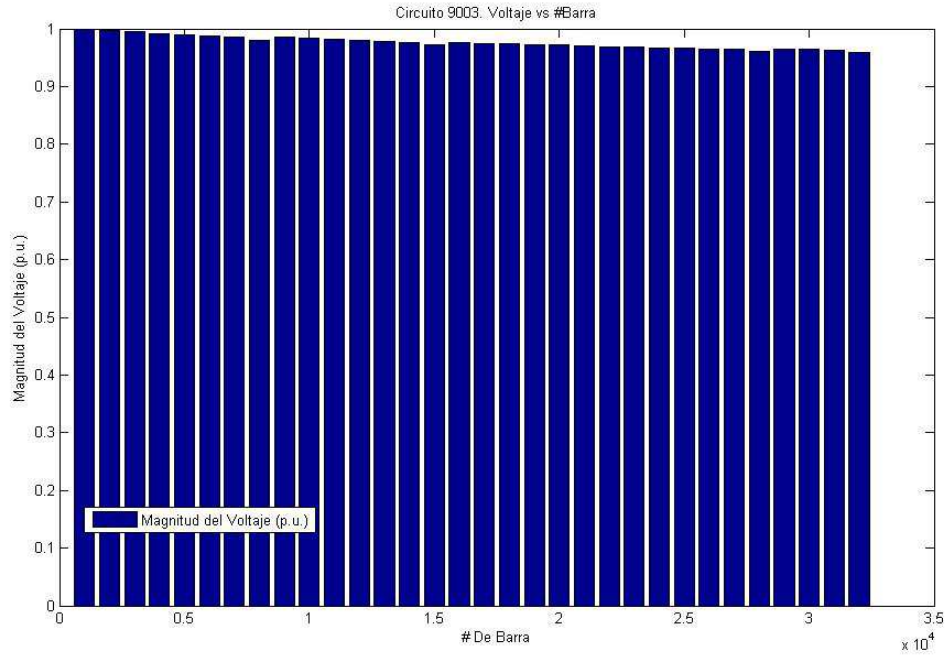
	--------	--------
Total:	0.071	0.08

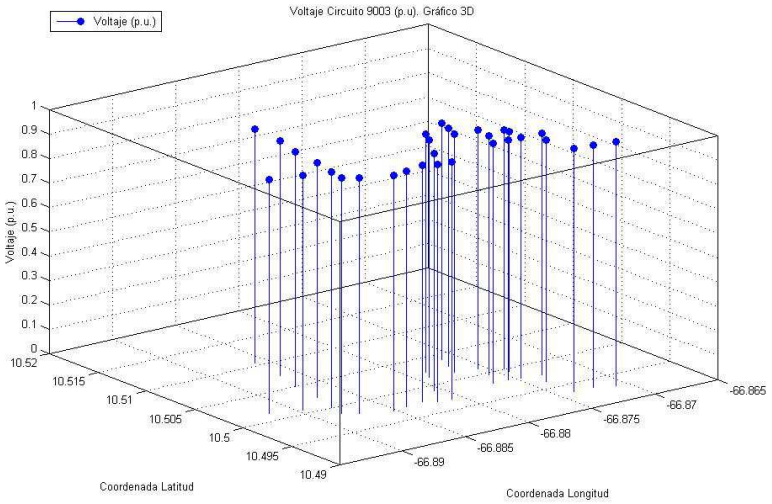
Mapa referenciado geográficamente del circuito:

Circuito 9003 representado en OpenStreetMaps.

Gráficas de Voltajes, Ángulos y Gráfico 3D con respecto a las barras del Circuito:







Circuito 9004

Resultados del Flujo de Carga:

```
Converged in 0.06 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    --------------------  -------------  -----------------
Buses                37  Total Gen Capacity       8.6          -4.3 to 4.3
Generators            1  On-line Capacity         8.6          -4.3 to 4.3
Committed Gens        1  Generation (actual)      4.0               2.0
Loads                 4  Load                     3.9               1.9
  Fixed               4    Fixed                  3.9               1.9
  Dispatchable        0    Dispatchable           0.0 of 0.0        0.0
Shunts                0  Shunt (inj)              0.0               0.0
Branches             36  Losses (I^2 * Z)         0.09              0.10
Transformers          0  Branch Charging (inj)     -                0.0
Inter-ties            0  Total Inter-tie Flow     0.0               0.0
Areas                 1

                          Minimum                      Maximum
                 -------------------------  --------------------------------
Voltage Magnitude   0.970 p.u. @ bus 350004       1.000 p.u. @ bus 10004
Voltage Angle      -0.79 deg   @ bus 250004       0.00 deg   @ bus 10004
P Losses (I^2*R)             -                  0.01 MW    @ line 20004-30004
Q Losses (I^2*X)             -                  0.02 MVAr  @ line 240004-250004

================================================================================
|     Bus Data                                                                 |
================================================================================
 Bus      Voltage          Generation             Load
  #   Mag(pu) Ang(deg)   P (MW)    Q (MVAr)    P (MW)    Q (MVAr)
----- ------- --------  --------  --------  --------  --------
10004  1.000    0.000       -         -         -         -
20004  0.996   -0.053       -         -         -         -
30004  0.993   -0.106       -         -         -         -
40004  0.991   -0.125       -         -         -         -
50004  0.990   -0.144       -         -         -         -
60004  0.989   -0.163       -         -         -         -
70004  0.987   -0.181       -         -         -         -
80004  0.986   -0.200       -         -         -         -
90004  0.985   -0.219       -         -         -         -
100004  0.984   -0.239       -         -         -         -
110004  0.982   -0.258       -         -         -         -
120004  0.981   -0.277       -         -         -         -
130004  0.980   -0.288       -         -         -         -
140004  0.980   -0.299       -         -         -         -
160004  0.980   -0.285       -         -         -         -
170004  0.980   -0.294       -         -         -         -
180004  0.979   -0.302       -         -         -         -
200004  0.985   -0.211       -         -         -         -
210004  0.983   -0.246       -         -         -         -
220004  0.981   -0.268       -         -         -         -
230004  0.980   -0.290       -         -         -         -
240004  0.978   -0.312       -         -         -         -
260004  0.982   -0.259       -         -         -         -
270004  0.981   -0.273       -         -         -         -
```

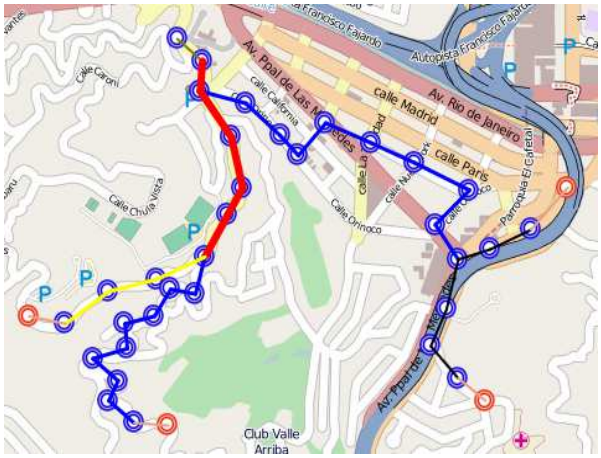
280004	0.980	-0.286	-	-	-	-
290004	0.979	-0.300	-	-	-	-
300004	0.978	-0.313	-	-	-	-
310004	0.977	-0.326	-	-	-	-
320004	0.977	-0.340	-	-	-	-
330004	0.976	-0.353	-	-	-	-
340004	0.975	-0.367	-	-	-	-
360004	0.990	-0.141	-	-	-	-
370004	0.988	-0.176	-	-	-	-
250004	0.971	-0.787	-	-	1.57	0.76
190004	0.977	-0.477	-	-	0.58	0.28
350004	0.970	-0.653	-	-	0.94	0.46
150004	0.976	-0.528	-	-	0.77	0.37
			3.96	1.97	3.87	1.87

Total:

Branch Data

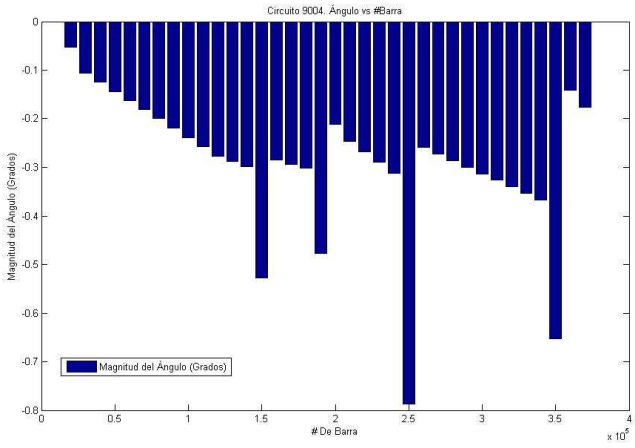
Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
0	10004	20004	3.96	1.97	-3.94	-1.96	0.013	0.01
1	20004	30004	3.94	1.96	-3.93	-1.95	0.013	0.01
2	30004	40004	1.37	0.67	-1.37	-0.67	0.002	0.00
3	40004	50004	1.37	0.67	-1.36	-0.67	0.002	0.00
4	50004	60004	1.36	0.67	-1.36	-0.67	0.002	0.00
5	60004	70004	1.36	0.67	-1.36	-0.67	0.002	0.00
6	70004	80004	1.36	0.67	-1.36	-0.67	0.002	0.00
7	80004	90004	1.36	0.67	-1.36	-0.67	0.002	0.00
8	90004	100004	1.36	0.67	-1.36	-0.67	0.002	0.00
9	100004	110004	1.36	0.67	-1.36	-0.66	0.002	0.00
10	110004	120004	1.36	0.66	-1.35	-0.66	0.002	0.00
11	120004	130004	0.77	0.38	-0.77	-0.38	0.000	0.00
12	130004	140004	0.77	0.38	-0.77	-0.37	0.000	0.00
13	140004	150004	0.77	0.37	-0.76	-0.37	0.001	0.00
14	120004	160004	0.59	0.29	-0.59	-0.29	0.000	0.00
15	160004	170004	0.59	0.29	-0.59	-0.29	0.000	0.00
16	170004	180004	0.59	0.29	-0.59	-0.29	0.000	0.00
17	180004	190004	0.59	0.29	-0.58	-0.28	0.001	0.00
18	30004	360004	2.56	1.28	-2.56	-1.27	0.005	0.00
19	360004	370004	2.56	1.27	-2.55	-1.27	0.005	0.00
20	370004	200004	2.55	1.27	-2.55	-1.26	0.005	0.00
21	200004	210004	2.55	1.26	-2.54	-1.26	0.005	0.00
22	210004	260004	0.95	0.47	-0.95	-0.47	0.001	0.00
23	260004	270004	0.95	0.47	-0.95	-0.47	0.001	0.00
24	270004	280004	0.95	0.47	-0.95	-0.47	0.001	0.00
25	280004	290004	0.95	0.47	-0.95	-0.47	0.001	0.00
26	290004	300004	0.95	0.47	-0.95	-0.47	0.001	0.00
27	300004	310004	0.95	0.47	-0.95	-0.47	0.001	0.00
28	310004	320004	0.95	0.47	-0.95	-0.47	0.001	0.00
29	320004	330004	0.95	0.47	-0.95	-0.47	0.001	0.00
30	330004	340004	0.95	0.47	-0.95	-0.46	0.001	0.00
31	340004	350004	0.95	0.46	-0.94	-0.46	0.002	0.01
32	210004	220004	1.59	0.79	-1.58	-0.79	0.002	0.00
33	220004	230004	1.58	0.79	-1.58	-0.78	0.002	0.00
34	230004	240004	1.58	0.78	-1.58	-0.78	0.002	0.00
35	240004	250004	1.58	0.78	-1.57	-0.76	0.006	0.02
					Total:		0.087	0.10

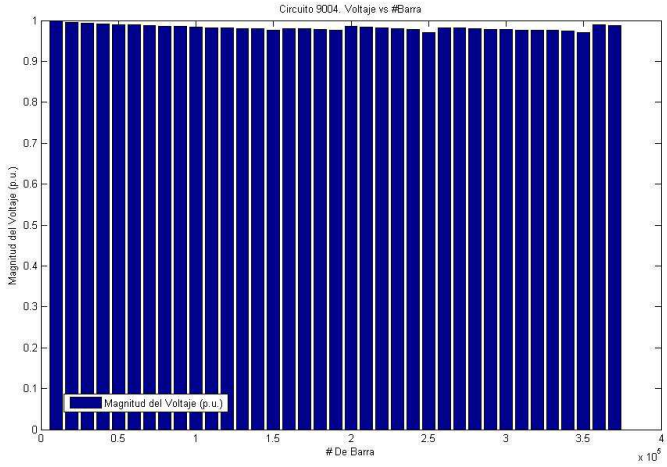
Mapa referenciado geográficamente del circuito:

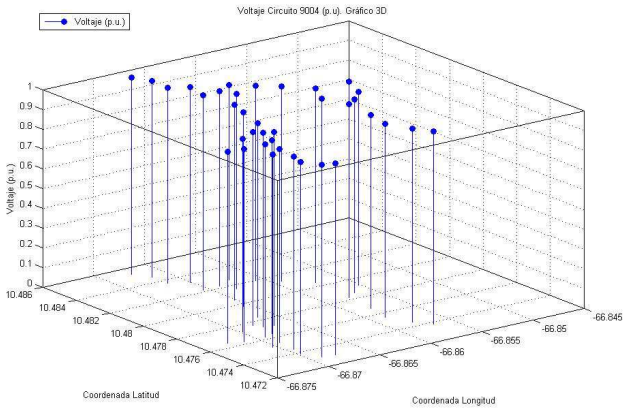


Circuito 9004 representado en OpenStreetMaps.

Gráficas de Voltajes, Ángulos y Gráfico 3D con respecto a las barras del circuito:







Circuito 9005

Resultados del Flujo de Carga:

```
Converged in 0.08 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    --------------------  -------------  -----------------
Buses              39    Total Gen Capacity       4.2          -2.1 to 2.1
Generators          1    On-line Capacity         4.2          -2.1 to 2.1
Committed Gens      1    Generation (actual)      2.0               1.0
Loads               3    Load                     1.9               0.9
  Fixed             3      Fixed                  1.9               0.9
  Dispatchable      0      Dispatchable           0.0 of 0.0        0.0
Shunts              0    Shunt (inj)              0.0               0.0
Branches           38    Losses (I^2 * Z)         0.07              0.08
Transformers        0    Branch Charging (inj)     -                0.0
Inter-ties          0    Total Inter-tie Flow     0.0               0.0
Areas               1

                          Minimum                      Maximum
                 -------------------------  ---------------------------------
Voltage Magnitude   0.956 p.u. @ bus 2700005       1.000 p.u. @ bus 100005
Voltage Angle      -1.08 deg   @ bus 2700005       0.00 deg   @ bus 100005
P Losses (I^2*R)             -                  0.00 MW    @ line 2600005-2700005
Q Losses (I^2*X)             -                  0.01 MVAr  @ line 2600005-2700005

================================================================================
|     Bus Data                                                                 |
================================================================================
 Bus      Voltage          Generation             Load
  #   Mag(pu) Ang(deg)    P (MW)    Q (MVAr)    P (MW)    Q (MVAr)
----- ------- --------  --------  --------  --------  --------
100005  1.000    0.000       -         -         -         -
200005  0.998   -0.025       -         -         -         -
300005  0.996   -0.051       -         -         -         -
400005  0.994   -0.077       -         -         -         -
500005  0.993   -0.102       -         -         -         -
600005  0.991   -0.128       -         -         -         -
700005  0.989   -0.154       -         -         -         -
800005  0.987   -0.180       -         -         -         -
900005  0.985   -0.206       -         -         -         -
1000005  0.983   -0.233       -         -         -         -
1100005  0.982   -0.259       -         -         -         -
1200005  0.980   -0.285       -         -         -         -
1300005  0.978   -0.312       -         -         -         -
1400005  0.976   -0.338       -         -         -         -
1500005  0.974   -0.365       -         -         -         -
1600005  0.973   -0.378       -         -         -         -
1700005  0.972   -0.391       -         -         -         -
1800005  0.972   -0.404       -         -         -         -
1900005  0.971   -0.416       -         -         -         -
2000005  0.970   -0.429       -         -         -         -
2100005  0.969   -0.442       -         -         -         -
2200005  0.968   -0.455       -         -         -         -
2300005  0.967   -0.468       -         -         -         -
2400005  0.966   -0.481       -         -         -         -
2500005  0.965   -0.494       -         -         -         -
2600005  0.965   -0.507       -         -         -         -
2800005  0.973   -0.379       -         -         -         -
2900005  0.972   -0.393       -         -         -         -
3000005  0.971   -0.407       -         -         -         -
```

3100005	0.970	-0.422	-	-	-	-
3200005	0.969	-0.436	-	-	-	-
3300005	0.968	-0.450	-	-	-	-
3400005	0.967	-0.464	-	-	-	-
3500005	0.967	-0.478	-	-	-	-
3600005	0.966	-0.480	-	-	-	-
3700005	0.966	-0.536	-	-	0.09	0.04
3800005	0.966	-0.491	-	-	-	-
2700005	0.956	-1.079	-	-	0.90	0.44
3900005	0.957	-1.062	-	-	0.90	0.44
			---	---	---	---
		Total:	1.96	0.99	1.89	0.92

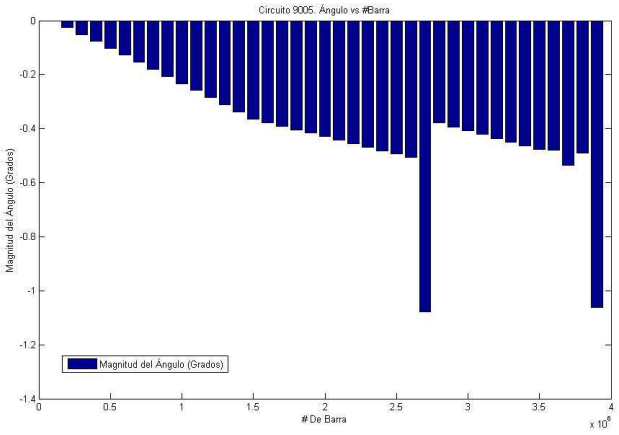
Branch Data

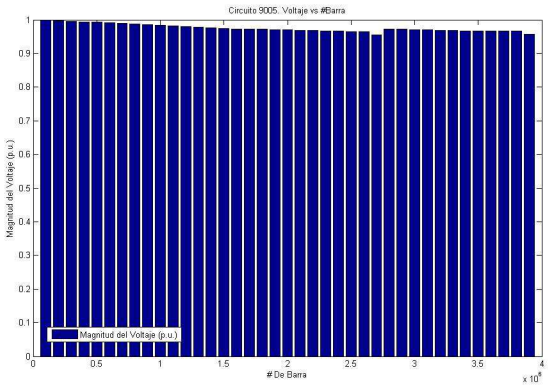
Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
0	100005	200005	1.96	0.99	-1.95	-0.99	0.003	0.00
1	200005	300005	1.95	0.99	-1.95	-0.99	0.003	0.00
2	300005	400005	1.95	0.99	-1.95	-0.98	0.003	0.00
3	400005	500005	1.95	0.98	-1.95	-0.98	0.003	0.00
4	500005	600005	1.95	0.98	-1.94	-0.98	0.003	0.00
5	600005	700005	1.94	0.98	-1.94	-0.98	0.003	0.00
6	700005	800005	1.94	0.98	-1.94	-0.97	0.003	0.00
7	800005	900005	1.94	0.97	-1.93	-0.97	0.003	0.00
8	9000051000005		1.93	0.97	-1.93	-0.97	0.003	0.00
9	10000051100005		1.93	0.97	-1.93	-0.97	0.003	0.00
10	11000051200005		1.93	0.97	-1.92	-0.96	0.003	0.00
11	12000051300005		1.92	0.96	-1.92	-0.96	0.003	0.00
12	13000051400005		1.92	0.96	-1.92	-0.96	0.003	0.00
13	14000051500005		1.92	0.96	-1.91	-0.95	0.003	0.00
14	15000051600005		0.91	0.46	-0.91	-0.46	0.001	0.00
15	16000051700005		0.91	0.46	-0.91	-0.45	0.001	0.00
16	17000051800005		0.91	0.45	-0.91	-0.45	0.001	0.00
17	18000051900005		0.91	0.45	-0.91	-0.45	0.001	0.00
18	19000052000005		0.91	0.45	-0.91	-0.45	0.001	0.00
19	20000052100005		0.91	0.45	-0.91	-0.45	0.001	0.00
20	21000052200005		0.91	0.45	-0.91	-0.45	0.001	0.00
21	22000052300005		0.91	0.45	-0.91	-0.45	0.001	0.00
22	23000052400005		0.91	0.45	-0.91	-0.45	0.001	0.00
23	24000052500005		0.91	0.45	-0.90	-0.45	0.001	0.00
24	25000052600005		0.90	0.45	-0.90	-0.45	0.001	0.00
25	26000052700005		0.90	0.45	-0.90	-0.44	0.004	0.01
26	15000052800005		1.00	0.50	-1.00	-0.50	0.001	0.00
27	28000052900005		1.00	0.50	-1.00	-0.50	0.001	0.00
28	29000053000005		1.00	0.50	-1.00	-0.50	0.001	0.00
29	30000053100005		1.00	0.50	-1.00	-0.50	0.001	0.00
30	31000053200005		1.00	0.50	-1.00	-0.50	0.001	0.00
31	32000053300005		1.00	0.50	-1.00	-0.49	0.001	0.00
32	33000053400005		1.00	0.49	-1.00	-0.49	0.001	0.00
33	34000053500005		1.00	0.49	-0.99	-0.49	0.001	0.00
34	35000053600005		0.09	0.04	-0.09	-0.04	0.000	0.00
35	36000053700005		0.09	0.04	-0.09	-0.04	0.000	0.00
36	35000053800005		0.90	0.45	-0.90	-0.45	0.001	0.00
37	38000053900005		0.90	0.45	-0.90	-0.44	0.004	0.01
						Total:	0.068	0.08

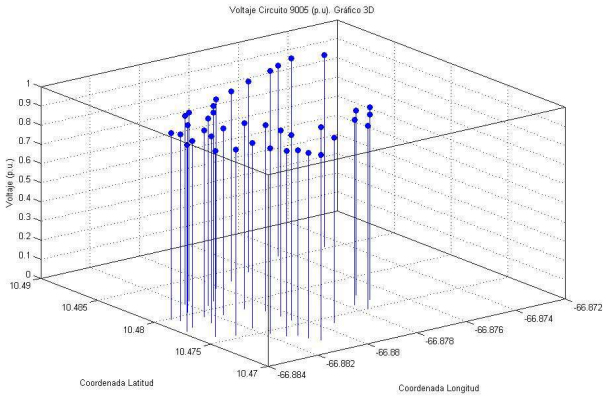
Mapa referenciado geográficamente del circuito:

Circuito 9005 representado en OpenStreetMaps.

Gráficas de Voltajes, Ángulos y Gráfico 3D con respecto a las barras del circuito:






Voltaje Circuito 9005 (p.u). Gráfico 3D
Voltaje (p.u.)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10.49
10.485
10.48
10.475
10.47
Coordenada Latitud
-66.884
-66.882
-66.88
-66.878
-66.876
-66.874
-66.872
Coordenada Longitud