

SEMINARIO DE INVESTIGACIÓN  
“HERRAMIENTAS SOFTWARE PARA EL ANÁLISIS DE  
MICRO-REDES”

ANA MARÍA CRUZ ROSAS  
JOAN GERARDO ORTEGA GARZÓN  
JOHN MAURICIO MORENO NIÑO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA 2014

SEMINARIO DE INVESTIGACIÓN  
“HERRAMIENTAS SOFTWARE PARA EL ANÁLISIS DE  
MICRO-REDES”

ANA MARÍA CRUZ ROSAS  
JOAN GERARDO ORTEGA GARZÓN  
JOHN MAURICIO MORENO NIÑO

Director

IVÁN DAVID SERNA SUÁREZ

*MS.c en Ingeniería Eléctrica*

Codirectores

GILBERTO CARRILLO CAICEDO

*Ingeniero Industrial. Ph.D*

GABRIEL ORDOÑEZ PLATA

*Ingeniero Electricista. Ph.D*

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA 2014

*Le dedico primeramente mi trabajo a Dios, quien me lo ha dado todo de forma perfecta, su inmenso amor es quien obra en mi vida y no tengo más que agradecerle.*

*De igual forma, dedico este triunfo a mis padres **Jime José Cruz Wilchez** y **Marisela Rosas Torres**, a ellos les debo mi hermosa vida, con su amor, apoyo y confianza han sabido educarme y sacarme adelante, a mis hermanos **Karen Lucia**, **Jimmy Leonardo** y **Juliana Andrea**, quienes adoro profundamente. A toda mi familia no me queda más que agradecer por su amor, ayuda y comprensión, esta solo es una más de las metas cumplidas en este viaje de magia y amor.*

*A mi mejor amigo y novio **Cristhian Duarte** quien ha sabido sostenerme en todos mis momentos, gracias por su amor y colaboración en la realización de este trabajo, nuestros sueños no tendrán fin.*

*A mis compañeros de trabajo **Joan** y **Mauricio**, juntos creamos y llevamos a cabo este proyecto, su paciencia, comprensión, esfuerzo y lucha fueron fundamentales para llegar al éxito, nuestra entrega ha sido total y he aquí este inmenso resultado.*

*A mis queridos amigos y colegas **Ruben**, **Ingrid**, **Ginny**, **Esteban**, **Hency**, **Camila**, **Fabian**, **Yuri**, **Edgar**, **Sonia**, **Carlos**, **Liceth** y a todos aquellos quienes en este camino encontré y me brindaron una sonrisa con momentos inolvidables, gracias por lo que aportaron en mi vida, por haberme escuchado y dejado compartir con todos; gracias a todas las personas que han cruzado por ella y han creado una huella que me fortalece como persona, bendiciones y alegrías para todos.*

*A mis profesores, quienes entregan su tiempo, conocimientos y educación para la formación de los futuros profesionales, su dedicación y apoyo siempre serán recordados.*

**Ana María Cruz Rosas**

---

# AGRADECIMIENTOS

*Primero que todo dar gracias a Dios, ya que con El todo y sin El nada , por estar en nuestros caminos, presente en cada paso que damos, por fortalecer nuestros corazones e iluminar nuestras mentes y habernos puesto juntos para lograr esta meta, siendo el soporte unos a otros.*

*Agradecemos a nuestro director de proyecto, el Profesor Iván Serna, por su apoyo, su calidad humana, su paciencia, por instruirnos y guiarnos en este proceso.*

*A nuestros compañeros de estudio, quienes son nuestros amigos y colegas, con quienes compartimos buenos y malos momentos, quienes nos apoyaron y por muchos motivos pasaron por nuestras vidas y dejaron gratos recuerdos, gracias a todos, sería imposible nombrar a tantos, más en nuestras mentes siempre tendrán un espacio.*

*Por último y a quien más nos debemos, nuestras familias, padres, madres y hermanos, quienes en este momento reciben junto a nosotros esta gran alegría, gracias a ustedes hoy estamos aquí, por su esfuerzo, su dedicación, su sacrificio, su inmenso amor y entrega total para que nosotros podamos salir adelante, Dios los bendiga siempre, los amamos.*

# CONTENIDO

INTRODUCCIÓN	15
<hr/>	
1. OBJETIVOS	17
<hr/>	
2. SEMINARIO DE INVESTIGACIÓN	18
<hr/>	
2.1. APRENDIENDO A USAR HERRAMIENTAS SOFTWARE . . . . .	19
2.2. RETOS DE LAS HERRAMIENTAS . . . . .	19
2.2.1. Python. • 2.2.2. Pypower. • 2.2.3. ATP. • 2.2.4. GridLab-D (GLD).	
2.3. ESTUDIO Y METODOLOGÍA PEDAGÓGICA . . . . .	23
2.3.1. Generación de la asignatura.	
2.4. INTERVENCIÓN Y GUÍA DEL DIRECTOR . . . . .	24
<hr/>	
3. CREACIÓN DEL MATERIAL PARA EL SEMINARIO	26
<hr/>	
3.1. CREACIÓN DE GUÍAS O LECTURAS POR SESIÓN . . . . .	27
3.2. CREACIÓN DE EJERCICIOS . . . . .	27
3.3. PRINCIPALES RETOS CON LA CREACIÓN DE ARCHIVOS . . . . .	27
3.4. RESULTADOS . . . . .	28
<hr/>	
4. MANEJO Y DESCRIPCION DEL MATERIAL PARA EL SEMINARIO	29
<hr/>	
4.1. DESCRIPCIÓN DE LA GUÍA . . . . .	30
4.2. MANEJO DE LA GUÍA . . . . .	30
4.3. DESCRIPCIÓN DEL EJERCICIO POR SESIÓN . . . . .	31
4.4. MANEJO DEL EJERCICIO FINAL . . . . .	31

5. CONCLUSIONES	33
6. RECOMENDACIONES	36
BIBLIOGRAFÍA	37
ANEXOS	39

# ANEXOS

A. PYTHON	40
B. PYPOWER	43
C. ATP	45
D. GRIDLAB-D	47

---

# GLOSARIO

IEEE	Instituto de Ingeniería Eléctrica y Electrónica
ATP	Alternative Transients Program
EMTP	Electro Magnetic Transient Program
BPA	Boneville Power Administration
TACS	Transient Analysis of Control System
DOE	Departamento de Energía de EE.UU.
PNNL	Laboratorio Nacional del Noroeste del Pacífico
NR	Newton Raphson
[IP_N]	IPython notebook
[N:M]	Distanciamiento de caracteres
FFT	Transformada rápida de Fourier
GLD	GridLab-D
FBS	Forward Sweep Backward
PW	PowerWorld
HVAC	Sistema de climatización
ZIP	Carga constante ( Corriente cte, Impedancia cte, Potencia cte)
RTP	Precio en Tiempo real
CREG	Comisión de Regulación de Energía y Gas

## RESUMEN

### **TÍTULO:**

SEMINARIO DE INVESTIGACIÓN “HERRAMIENTAS SOFTWARE PARA EL ANÁLISIS DE MICRO-REDES” <sup>1</sup>

### **AUTOR:**

ANA MARIA CRUZ ROSAS, JOAN GERARDO ORTEGA GARZÓN, JOHN MAURICIO MORENO NIÑO<sup>2</sup>

### **PALABRAS CLAVE:**

Diseño y simulación de sistemas de transmisión y distribución, modelado de elementos de una micro-red, análisis financiero, Python, ATP, PyPower, GridLab-D.

### **DESCRIPCIÓN:**

Una tendencia global en los sistemas eléctricos está asociada con la implementación de micro-redes. Sin embargo limitantes técnicos y económicos para disponer de un laboratorio de experimentación llevan a buscar otras alternativas, como por ejemplo, herramientas software que permitan diseñar ambientes que encierren las características y el comportamiento de las micro-redes para analizar su funcionamiento. Los estudiantes en sus últimos semestres académicos buscan que los conocimientos adquiridos sean lo más cercanos a lo usado en la vida profesional real, es por ello que este seminario instruye en herramientas software de gran utilidad que son usadas por empresas de energía eléctrica para correr sus flujos de potencia, detectar sus fallas y realizar sus análisis de costo/beneficio.

En este trabajo se presenta la creación completa de una asignatura, basada en el aprendizaje de cuatro herramientas software ( Python, ATP, PyPower y GridLab-D), como resultado se entrega el material con la temática y metodología a desarrollar en el transcurso de un semestre académico y se dejan como anexos las carpetas que contienen las guías de instalación paso a paso de cada herramienta, las lecturas y temáticas propuestas para cada sesión con ejercicios ejemplo solucionados y simulados, ejercicios propuesto como método de evaluación para el desarrollo individual por parte del estudiante en cada semana de estudio y un ejercicio final que concentra lo aprendido de cada herramienta durante las sesiones, todos los ejercicios simulados y desarrollados para el docente.

---

<sup>1</sup>Trabajo de grado.

<sup>2</sup>Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.  
Director: Iván Serna, Codirectores: Gilberto Carrillo, Gabriel Ordoñez.

## ABSTRACT

**TITLE:**

RESEARCH SEMINAR "SOFTWARE TOOLS FOR ANALYSIS OF MICRO-NETWORKS"<sup>3</sup>

**AUTHORS:**

ANA MARIA CRUZ ROSAS, JOAN GERARDO ORTEGA GARZÓN, JOHN MAURICIO MORENO NIÑO <sup>4</sup>

**KEY WORDS:**

Design and simulation of transmission and distribution systems, modeling elements of a smart-grid, financial analysis, Python, ATP, PyPower, GridLab-D.

**DESCRIPTION:**

A global trend in power systems is associated with the implementation of smart-grid. However technical and economic limitations to dispose of an experimental laboratory raise the need to other alternatives, an example is the software tools which allow design environments that enclose the characteristics and behavior of micro-networks to analyze their performance. Students in their last semesters seeking that the foreground are as closest is used a real professional life, that is why this seminar instructed in valuable software tools that are used by power companies to run their power flows, identify its flaws and perform its cost / benefit.

In this work the complete creation of a course based on learning four software tools (Python, ATP, and PyPower GridLab-D) occurs, as a result the material is delivered to the topic and methodology be developed in the course of an academic semester and allowed as annexes in folders guides Installation Walkthrough of each tool, the readings and proposed themes for each session with exercises solved and simulated example, proposed as the method of evaluation for individual development by the student in each week of study exercises and a final exercise that focuses on lessons learned from each tool during sessions, all exercises are simulated and developed for teaching.

---

<sup>3</sup>Research work.

<sup>4</sup>Faculty of Physical-Mechanic Engineering. School of Electrical, Electronical and Telecommunications Engineering. Advisor: Iván Serna. Co-advisor: Gilberto Carrillo, Gabriel Ordoñez.

---

# INTRODUCCIÓN

Los sistemas eléctricos del país se encuentran en continuo desarrollo, y por lo general la implementación de nuevas tecnologías en métodos de medición y facturación energética resultan en cambios favorables para los comercializadores y usuarios. Las micro-redes son la vanguardia tecnológica para la distribución de energía eléctrica (incluyendo el aprovechamiento de fuentes locales de energía), por lo tanto, se han desarrollado herramientas software para el análisis y simulación de sus características previendo ventajas y desventajas en su estructura, configuración, elementos y dispositivos que pueden hacer parte de la formación de una micro-red. Con la ayuda de estas herramientas software se pueden analizar los modelos de los equipos que conforman las micro-redes y crear nuevos modelos según las necesidades de los estudios.

En este trabajo de grado se investigaron y estudiaron cuatro (4) herramientas software que permiten el diseño, la creación, el análisis y la simulación de micro-redes y sistemas de potencia.

Este trabajo de grado expone la utilidad y algunos de los beneficios del uso de software libre. Richard Stallman (Stallman, 2004) establece que los usuarios deben tener la libertad para “ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software”. Más específicamente define cuatro (4) características:

Libertad 0: la libertad para ejecutar el programa sea cual sea el propósito

Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a las necesidades del usuario—el acceso al código fuente es condición indispensable para esto.

Libertad 2: la libertad para redistribuir copias.

## *Introducción*

Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad —el acceso al código fuente es condición indispensable para esto.

Estas libertades le proporcionan al software libre una serie de ventajas con respecto al software privado. Algunas de estas son:

- Tienen un muy bajo costo (en algunos casos nulo) de adquisición y son de libre uso. Se encuentran al escrutinio público, lo que facilita encontrar y corregir errores, además, al haber una mayor cantidad de personas involucradas en un proyecto se incrementa la innovación tecnológica que se puede alcanzar.
- A diferencia de los programas privativos, una herramienta de software libre no requiere de un sistema operativo específico, “lo que se traduce en un mejor soporte -de manera general- para las versiones antiguas de software y de plataformas de hardware o software más minoritarias” (Culebro, Gómez, & Torres, 2006).
- Aumentan las posibilidades de aprendizaje de los usuarios.

Como resultado del proyecto se creó una serie de material educativo y de apoyo correspondiente a un seminario tipo taller dedicado al análisis y estudio de micro-redes, conformado por lecturas guía, ejemplos y ejercicios propuestos como método de evaluación. Este seminario fue creado para ser desarrollado en sesiones o clases divididas en cuatro fases, una por herramienta con énfasis en la metodología de clase teórico-práctica y generando un espacio de formación y refuerzo hacia la programación y simulación de sistemas eléctricos reales mediante el uso de herramientas software de código abierto.

---

## CAPÍTULO 1

---

# OBJETIVOS

### OBJETIVO GENERAL

- Desarrollar las competencias cognitivas, pragmáticas y ontológicas, necesarias para el análisis de micro-redes a través de herramientas software sin cargos por distribución y/o de código abierto.

### OBJETIVOS ESPECIFICOS

- Analizar la operación óptima y probabilística de los sistemas de potencia, bajo el paradigma de las micro-redes, por medio de Python/Pypower.
- Analizar esquemas financieros y tarifarios de la demanda de energía en sistemas de distribución, bajo el paradigma de las micro-redes, por medio de GridLab-D.
- Analizar el modelado de circuitos, generación de código fuente y la simulación de circuitos creados mediante el uso de herramientas de ATP.
- Analizar el comportamiento de los equipos y elementos que componen un sistema de distribución empleando Python.

---

## CAPÍTULO 2

---

# SEMINARIO DE INVESTIGACIÓN

Este trabajo consiste en la creación de una asignatura basada en el aprendizaje de cuatro herramientas software (Python, PyPower, ATP y GridLab-D) para el análisis y estudio de micro-redes. A continuación se desea plasmar las experiencias recogidas durante en la creación del plan de asignatura y sus contenidos. La metodología de enseñanza, la elaboración, diseño y redacción del material de trabajo.

## 2.1 APRENDIENDO A USAR HERRAMIENTAS SOFTWARE

El primer paso que se llevó a cabo en el proceso de realización del proyecto tuvo que ver con el reconocimiento e investigación de las herramientas software en las que se fundamenta la realización del seminario, desde el inicio del proceso se encontró una nueva experiencia, la cual consistió en la utilización de software y lenguaje de código abierto sobre los cuales se realizó este trabajo. Debido a la poca masificación de la herramientas, en algunas ocasiones, encontrar información sobre su instalación y uso resultó una tarea tediosa, de concentración y dedicación.

La instalación de los paquetes software de cada herramienta de la mano de nuestro director de proyecto, así como la información brindada para el aprendizaje del manejo y uso de los software fue trabajado en conjunto, para así realizar las guías de instalación y fundamentos de trabajo de cada herramienta.

En las sesiones programadas por el director de proyecto de grado, orientadas a la dirección del proyecto, se presentaban las principales inquietudes en el funcionamiento del software y posibles soluciones. Progresivamente se resolvieron los inconvenientes con cada una de estas herramientas, en ocasiones necesitando repasar los temas presentados en algunas de las asignaturas vistas durante el transcurso de la carrera.

## 2.2 RETOS DE LAS HERRAMIENTAS

Cada herramienta de estudio presenta en particular diferentes lenguajes de programación y modos de funcionamiento, de manera que en cada paso de aprendizaje se presentaron diferentes retos al enfrentarse con su uso y manejo del programa.

La interfaz gráfica que presenta cada uno de los programas es diferente, algunos de ellos cuentan con editores de texto como es el caso de Python y PyPower, consola de programación MS-DOS como es el caso GridLab-D y otros con entorno gráfico como es el caso de ATPDraw.

A continuación se presenta el análisis realizado por los autores respecto a los principales retos abarcados con cada uno de los software y la experiencia del trabajo realizado basada en los objetivos que se querían lograr con cada una de las herramientas.

**2.2.1 Python** Al introducirse en el contexto de la programación, es inevitable recordar aquellos primeros pasos dados en cursos como C, C ++ o similares, en los cuales se estructuraban líneas de código con un propósito específico, en un entorno de programación no muy amigable. En esta ocasión se hablará concretamente del lenguaje de programación conocido como Python.

Cuando se enfrentan situaciones nuevas es natural abrumarse por la incertidumbre que esto genera y quizá esta es la sensación que se experimenta al escuchar por primera vez la palabra Python. Luego de unos minutos y asociando este termino con un lenguaje de programación, se concluye que podría tratarse del mismo lenguaje usado en los software Matlab o Scilab muy utilizados en ingeniería y de los cuales se tienen mayores referencias.

Por otro lado, luego de un proceso de documentación riguroso se consigue un acercamiento menos traumático frente al lenguaje, y se logra comprender las cualidades que este ofrece y que lo hacen tan atractivo frente a otros lenguajes. Quizá una de las mayores dificultades consistió en identificar la terminología asociada con la colección de paquetes para Python. Otra dificultad se presentó en el manejo de la sintaxis que, aunque se parece mucho a la de otros software de análisis científico y de ingeniería, cambia ligeramente en el modo de escritura y de asignación de variables, puesto que en ocasiones se realiza de manera mas semántica (lo que quiere decir) que funcional (lo que se desea hacer). Y por último, también se encontró dificultades para definir el entorno en el cual se desarrollaron los ejercicios y demás actividades propuestas, estas dificultades fueron finalmente sobrellevadas y solucionadas.

---

**2.2.2 Pypower** La instalación de PyPower fue un proceso sencillo debido a que es un paquete de Python, el entorno en el cual se puede trabajar el modulo es variado, las guías fueron desarrolladas en el Notebook de Python, en general en cualquier editor de texto plano con extensión \*.py se puede crear el código que represente el sistema de potencia al cual se le desea implementar la simulación para conocer el flujo clásico u optimo de potencia.

PyPower es un modulo de Python, por lo que el lenguaje es ampliamente conocido y permite el mismo manejo que cualquier código que utiliza módulos, librerías y funciones. Usado para resolver problemas de flujos de potencia para corriente alterna y corriente continua, PyPower es similar a la herramienta de MatLab: Matpower, pero diseñada y programada para Python lo cual tiene ciertas ventajas, ya que es de uso y distribución libre. Las características, funciones y ventajas del uso de PyPower están consignadas en la primera lectura guía, esta explica las

## 2.2. RETOS DE LAS HERRAMIENTAS

diferentes formas en las que se puede obtener el flujo de potencia y el flujo optimo de potencia en un sistema.

En la ultima guía se explica en que consiste la solución del flujo probabilístico en un sistema de potencia, allí esta consignada la teoría necesaria para que conozcan algunas de las incertidumbre presentes en los elementos de un sistema de potencia y como intervienen en los resultados de un flujo de carga y su analisis. La forma como se pueden modelar o caracterizar estos parámetros resultan en el uso de ecuaciones y distribuciones de probabilidad, se entregan las bases para realizar el análisis posterior de un sistema de potencia el cual requiere de todas las herramientas aprendidas en las sesiones de PyPower y la practica en el lenguaje de programación Python.

Con PyPower se pueden generar análisis financieros basado en el cambio de precios debidos a la elasticidad de la demanda. El uso de funciones externas y comandos que guardan salidas especificas para un análisis posterior hacen de PyPower un paquete completo y de importancia a la hora de realizar análisis de cargas y solución de flujos de potencia.

---

**2.2.3 ATP** El proceso realizado en el estudio de la herramienta ATP comienza con el proceso de instalación del software ATP-ATPDraw, el cual se llevó a cabo de manera sencilla y sin ningún inconveniente. De cualquier forma, este proceso se consignó de la manera más detallada posible en la guía de instalación, para garantizar su correcta instalación.

Se exploró el entorno de trabajo que proporcionaba la herramienta, ATPDraw, la cual presenta al usuario un entorno amigable y sencillo a pesar de no tenerse conocimiento previo acerca del software ATP y su lenguaje de programación.

Las características principales que conforman el ATPDraw, así como su modo de operación y componentes hacen parte de la descripción que se desarrolla en la guía introducción a ATP-Draw, la cual se realizó obteniendo información de diferentes fuentes bibliográficas. Así mismo, la orientación del director de proyecto fue fundamental para establecer las características y procesos que se deben llevar a cabo con el uso de la herramienta; para realizar el proceso de simulación, adecuado, y para distinguir los diferentes tipos de herramientas de edición y los diferentes tipos de componentes eléctricos que hacen parte del ATP.

Para ampliar el uso del software se realizó el estudio de rutinas auxiliares disponibles en el programa, tales como la rutina MODELS en donde se crearon elementos de medida y

control. La rutina DATA BASE MODULE proporcionó la creación de elementos que no se encontraban dentro del menú de componentes del software. Estos detalles hacen de ATPDraw una herramienta poderosa y de gran interés para los ingenieros analistas, proporciona una amplia gama de aplicaciones útiles en el proceso de análisis de sistemas eléctricos de potencia.

---

*2.2.4 GridLab-D (GLD)* Esta herramienta en particular presentó desde su inicio una serie de retos, su falta de interfaz gráfica amigable con el usuario y un lenguaje de programación exclusivo de GLD, estos retos con lectura previa y experiencia en simulaciones pueden superarse con facilidad y el usuario aprende a interactuar con la consola.

La instalación del software se realizó dependiendo de la arquitectura del computador (32 o 64 bits). La información concerniente a GridLab-D pueden encontrarse en su página oficial creada por el Departamento de Energía de EE.UU. (DOE) en el Laboratorio Nacional del Noroeste del Pacífico (PNNL), en colaboración con la industria y el mundo académico. Allí se pueden encontrar los links de descarga y la plataforma sourceforge, en donde se concentran archivos, foros, wikis, y algunos códigos de prueba.

La instalación de GridLab-D necesitó algunos complementos, los cuales por defecto se encuentran en la colección de paquetes de Anaconda para Python, que ya estaban en el equipo, sin embargo en la guía de instalación no se tomó en cuenta la previa instalación de otros programas con recursos y complementos compartidos, por lo tanto el usuario debe omitir algunos pasos asignados a estos complementos específicos como lo es el MinGW.

Cuando la instalación está completa y correcta se puede acceder a la consola de trabajo de GLD, la cual es un entorno con una ventana de trabajo de fondo negro, parecido a la interfaz de la consola en Windows (MS-DOS). GLD presenta un entorno de trabajo basado en el importe y carga de archivos desde la consola.

Las primeras simulaciones que se lograron en la guía uno se generaron usando los archivos ejemplo que trae el software, sin embargo se hizo necesario leer toda la información suministrada por la plataforma de la página oficial para entender los procesos de prueba, las líneas de comando de ayuda y la simulación de archivos.

### 2.3 ESTUDIO Y METODOLOGÍA PEDAGÓGICA

En el desarrollo del proyecto se tuvieron en cuenta aspectos como el uso y el estudio a futuro del material realizado por parte de otras personas, por ello previendo que un docente o grupo de estudiantes tomen el material que resultó del seminario de investigación y decidan emplearlo en una clase, se pensó en el tipo de metodología pedagógica, la cual pueden variar y cambiar según la persona que quiere convertirse en el tutor o docente.

Algunos consideran que cualquiera que haya sido un estudiante puede enseñar, pero esto no es totalmente cierto. Cada docente o guía de un curso debe adoptar una metodología de enseñanza enfocada en el aprendizaje y la ayuda al estudiante, una metodología no es algo que uno puede recibir de los demás, pues se puede definir como el complejo resultado de la instrucción personal, la experiencia académica y la reflexión acerca de los resultados que se esperan.

Las metodologías pedagógicas pueden variar y cambiar según la persona que quiere convertirse en docente para una clase o grupo, pero en general existen ciertos aspectos importantes que se deben cubrir al escoger la metodología y generar unos objetivos, entre ellos están:

- Ganar la atención y el respeto de los estudiantes
- Generar expectativas y curiosidad por las temáticas a tratar en el curso
- Tener el material pertinente, ordenado y previamente aprendido con gran dominio
- Herramientas audiovisuales y material de apoyo.
- Proporcionar tiempo para la práctica y la retroalimentación

Teniendo en cuenta los anteriores aspectos, se realizó el material de la asignatura y se enfocó en una metodología pedagógica teórico-práctica, en donde, teniendo en cuenta el enfoque de seminario, se espera una participación activa de los estudiantes, mayor que la del docente. El docente inicia la sesión con la explicación de la temática y luego sin demora da uno o varios ejemplos del tema, a mediados de la clase se propone un ejercicio el cual se basa en la explicación teórica, hablada o presentada en herramientas audiovisuales y se procede a invitar a los estudiantes a culminar el ejercicio de manera individual y autónoma, generando así dudas que pueden ser contestadas en la misma sesión, dejando como trabajo individual y de repaso el ejercicio hasta la siguiente clase.

Esta metodología teórico-práctica permite la relación e interacción entre el maestro y el estu-

## Capítulo 2. SEMINARIO DE INVESTIGACIÓN

dianete, abriendo el espacio para la práctica y comprobar que tanto se comprendió la temática expuesta.

---

*2.3.1 Generación de la asignatura* Una asignatura se puede definir como un conjunto de actividades de trabajo académico organizadas por uno o varios docentes con propósitos formativos, en torno a una temática y/o problemática específico que se desarrolla a partir de la relación entre estudiantes y profesores. (Lineamientos para el seminario de investigación como modalidad de trabajo para el desarrollo del trabajo de grado, Vicerectoría Académica, Universidad Industrial de Santander, Septiembre, 2007, 16).

Para la generación de la asignatura fue necesario desde el inicio hacer reconocimiento de diferentes aspectos pedagógicos y técnicos. La fase inicial fue escoger la temática y empezar a interactuar y especializarse en los temas. Fue necesario que la realización del material fuera totalmente valida y generada bajo un índice alto de conocimiento, por ello se aprendió y elaboró el material que se entregó para la realización de la asignatura.

La elaboración de este material tiene un amplio trasfondo ya que la parte de aprendizaje propio e investigación fue altamente compleja debido a las capacidades requeridas por las herramientas software que se implementaron. Para realizar este seminario los autores tuvieron que estar en capacidad de razonar lógicamente y abrir la mente a la programación de diferentes lenguajes. Para ello se tomaron experiencias vividas en el transcurso de la carrera profesional en donde diferentes asignaturas introdujeron a la programación.

Por otra parte en el proceso de investigación hubo un choque debido a la poca información que existe sobre las herramientas. La orientación del director de proyecto fue fundamental para establecer el proceso de búsqueda de la información necesaria que debió llevarse a cabo en Internet, con esto se logró la recopilación de información para comenzar a ordenar, traducir y analizar la información.

### 2.4 INTERVENCIÓN Y GUÍA DEL DIRECTOR

La asesoría y dirección del profesor jugo un papel trascendental en el desarrollo del seminario de investigación ya que se contó con la experiencia del docente en la estructuración y creación de las lecturas guías, ejemplos y ejercicios que son de gran utilidad y aplicación en el proceso de formación propuesta.

#### 2.4. INTERVENCIÓN Y GUÍA DEL DIRECTOR

Además de esto se contó con su asesoría en el manejo de las herramientas de simulación y en el material bibliográfico, que fue de gran utilidad para comprender distintos paquetes y funciones que permitieron ampliar el campo de aplicación de las herramientas de simulación integradas en el desarrollo del seminario.

Finalmente se realizaron entregas semanales de las guías para la revisión por parte del profesor y la asesoría en el desarrollo del módulo teórico y práctico, todo esto basado en el planteamiento del problema y los niveles de logro propuestos para el desarrollo del seminario.

---

## CAPÍTULO 3

---

# CREACIÓN DEL MATERIAL PARA EL SEMINARIO

El contenido de este capítulo presenta un resumen general de las actividades propuestas, los ejercicios de evaluación y los niveles de logro que se desean cumplir durante el seminario. El material anexo encierra la temática del seminario desarrollada para cada una de las herramientas software en el análisis y estudio de micro-redes, el cual puede ser empleado como base para diseñar trabajos futuros enfocados en el mejoramiento de los métodos de enseñanza de las herramientas computacionales orientadas a cualquier tema particular.

### 3.1 CREACIÓN DE GUÍAS O LECTURAS POR SESIÓN

La creación de las guías y lecturas diseñadas para la explicación general de la herramienta, constituye el eje de los conocimientos e ideas para el desarrollo de cada una de las sesiones dispuestas, estas son una guía detallada del manejo de los software desde su proceso de instalación hasta la aplicación en el análisis y estudio de micro-redes. Los conocimientos adquiridos mediante el desarrollo de ejemplos, llevados a cabo en cada sección y el planteamiento de ejercicios finales que aumentan el nivel como medida de evaluación o sustentación de las temáticas concentran la esencia de la lectura. Uno de los objetivos por cumplir al aplicar los ejercicios dentro de la clase es el de generar dudas y considerar su solución, en caso de no ser terminado se propone como trabajo personal con el compromiso de solución para la siguiente sesión.

### 3.2 CREACIÓN DE EJERCICIOS

La creación de ejercicios tiene como fundamento el afianzamiento de los conocimientos dados por el expositor durante el proceso de aprendizaje, a lo largo de las sesiones o clases basado en la temática de la lectura. En cada una se tienen como base los ejemplos que se desarrollan a medida que se avanza en la ejecución de la guía de trabajo, los ejemplos proporcionan el funcionamiento básico y detallado de las características principales del software en cada nivel del proceso y son necesarios para poder adquirir los conocimientos que servirán como herramientas para afrontar el problema planteado al final de cada sesión del seminario. En segundo plano los ejercicios sirven para llevar al estudiante dentro de un proceso de evaluación propia y ser el indicador de cumplimiento de los niveles de logros propuestos.

### 3.3 PRINCIPALES RETOS CON LA CREACIÓN DE ARCHIVOS

Los principales retos que se afrontaron en el momento de llevar a cabo la creación y constitución de material de trabajo, tuvieron que ver con las fuentes bibliográficas y las bases teóricas que explican o definen las herramientas software trabajadas en el seminario, esto se debió a que la información a la cual se pudo acceder con facilidad en Internet no se encontraba actualizada y muchas veces no cubría con detalles los procesos importantes del funcionamiento del software en estudio. Aun así siempre se tuvo la colaboración del director de proyecto quien nos abrió una ventana inagotable de recurso bibliográfico, enseñándonos bibliotecas virtuales

### *Capítulo 3.* CREACIÓN DEL MATERIAL PARA EL SEMINARIO

de importantes instituciones y revistas científicas, lo cual abrió un panorama de posibilidades y de allí quedo en nuestras manos la investigación y síntesis de contenido para crear las guías. Este proceso sirvió como experiencia en la creación de todo el material en general, siempre se buscó tener un alto nivel de detalle que sirviera para el usuario novato que se introduce en los lenguajes de programación y específicamente en las herramientas informáticas para el estudio de micro-redes, tomando el material de apoyo para que se motive a aprender y desarrollarse en los ambientes de programación y en los tipos de herramientas computacionales presentadas en el desarrollo del seminario.

## 3.4 RESULTADOS

El resultado físico del trabajo realizado contiene el desarrollo de lecturas guías, archivos de simulación de ejemplos y ejercicios planteados en el desarrollo del seminario, material bibliográfico. El resultado académico que se logro es poder introducir y explicar el funcionamiento de herramientas informáticas gratuitas (software libre), que le permitan al usuario abarcar y afianzar su proceso de formación en el uso de herramientas de simulación como soporte al conocimiento teórico adquirido durante su proceso de formación académica.

---

## CAPÍTULO 4

---

# MANEJO Y DESCRIPCION DEL MATERIAL PARA EL SEMINARIO

Un seminario es un espacio donde unos estudiantes, con la guía y ayuda de un profesor, construyen de manera conjunta una serie de conocimientos sobre algún tema específico. El intercambio de ideas es fundamental y caracteriza la modalidad seminario, se pueden lograr grandes avances y conclusiones cuando la participación de los asistentes es más activa, haciendo que el éxito del seminario dependa de que tan enriquecedora es la participación de los estudiantes, esta tiene que ser mayor que la del profesor, este ultimo sera el encargado de proponer los temas y actividades, dirigiendo hacia un buen rumbo las clases.

Las lecturas se centran en adquirir habilidades por medio de la elaboración de los ejemplos en clase, individuales o grupales, en donde lo que se busca es situar al estudiante frente a un problema, y por medio del material brindado y asesoría del profesor o monitor encargado de la sesión, solucionar dudas e inconvenientes con el software, generando en ellos una retroalimentación positiva que de como resultado la solución del ejercicio propuesto. El estudiante deberá resolver algunos ejemplos en el tiempo establecido, mientras que los ejercicios de mayor nivel que se encuentran como trabajo de la semana, tienen un tiempo de respuesta entre clase y clase para que el alumno lo analice y de respuesta de manera autónoma como trabajo personal.

Se creo un ejercicio final por cada herramienta vista durante el desarrollo del seminario, este ejercicio concentra los conocimientos adquiridos durante las sesiones teóricas y los ejercicios semanales de evaluación, para su realización se programó una semana completa dedicada al análisis y realización del mismo.

## 4.1 DESCRIPCIÓN DE LA GUÍA

La guía presentada para el desarrollo del seminario contiene la siguiente estructura:

- **Objetivo:** Abarca de manera general el uso de la herramienta de trabajo.
- **Competencia:** Plantea el parámetro sobre el cual será desarrollada la guía y evaluada la sesión..
- **Palabras claves:** Brinda información de ciertas definiciones que puedan facilitar el proceso de aprendizaje.
- **Resumen:** Aborda la descripción general de la guía y su contenido.
- **Introducción:** Presenta los temas de estudio haciendo un énfasis en la importancia y las prestaciones que se pueden obtener con la herramienta de trabajo.
- **Instalación:** Describe la correcta instalación del software que se va a manejar (presente solamente en la guía instalación).
- **Desarrollo del contenido temático:** Plantea el desarrollo conceptual y estructural del programa como lo es el entorno, estructura, herramientas, interfaces gráficas que utiliza, proceso de simulación.
- **Ejercicio:** se realizar los primeros pasos en el proceso de interacción y afianzamiento con el programa en estudio.
- **Ejemplo:** Es formulado al final del contenido de cada lectura guía, el cual es clave en el proceso de aprendizaje y afianzamiento de los niveles de logro.
- **Referencias:** En esta parte se encuentran los principales textos guías que fueron utilizados para el desarrollo del contenido en cada lectura guía.
- **Niveles de logro:** Son los resultados que se pretenden alcanzar y cumplir con la realización de las actividades desarrolladas en la guía por parte del usuario.
- **Anexos:** En estos se encuentra el material de trabajo, junto con los resultados de las simulaciones que tuvieron lugar en el desarrollo de la lectura guía.

## 4.2 MANEJO DE LA GUÍA

Para realizar un manejo adecuado y eficiente del material realizado durante el seminario, se aconseja al usuario o docente la lectura detallada de las guías que se entregan como resultado

### 4.3. DESCRIPCIÓN DEL EJERCICIO POR SESIÓN

de este proyecto de grado. Debe ser preparada la sesión con anterioridad, es aconsejable el uso de alguna herramienta audiovisual para la explicación del contenido de la guía, así mismo se aconseja que los estudiantes que tomen el seminario tengan su propio equipo de computo ya que se hace indispensable que puedan seguir el ejemplo de la clase digitando los códigos y corriendo las simulaciones durante el desarrollo de cada actividad, además de llevar el orden especificado en los procesos de simulación e interacción de los diferentes tipos de herramientas, es importante la conexión del maestro con sus alumnos y la programación dirigida. El trabajo en equipo requiere especial cuidado para que todos los asistentes puedan tener excelentes resultados y no se cree una sesión donde solo el profesor es quien realiza las simulaciones y los asistentes solo son observadores pasivos. Como la modalidad es seminario se recomienda la asistencia de un docente y un auxiliar, quien con los conocimientos en las herramientas informáticas puedan guiar y ayudar a los estudiantes con sus simulaciones y las dudas generadas por el trabajo en clase.

### 4.3 DESCRIPCIÓN DEL EJERCICIO POR SESIÓN

El ejercicio propuesto por semana abarca la mayor parte de las rutinas implementadas en el desarrollo del seminario, haciendo énfasis en la importancia de la herramienta. Con la implementación y solución del ejercicio se pretende abarcar gran parte de los niveles de logro, además de motivar a el estudiante con las bases dadas pero realizando algunos cambios tenga un pensamiento lógico de programación y de esta forma supere el reto propuesto en el ejercicio. Estos son pensados como una parte del ejercicio final del estudio de cada herramienta, por lo cual se puede decir que los estudiantes construyen una torre de conocimientos los cuales los harán capaces de resolver nuevos retos a medida que las sesiones de las herramientas avanzan y el nivel crece.

### 4.4 MANEJO DEL EJERCICIO FINAL

Para lograr la ejecución y un óptimo aprovechamiento del ejercicio final, se aconseja al usuario haber realizado los ejemplos y ejercicios propuestos en el desarrollo del seminario, lo cual hará que el ejercicio final sea la integración y culminación de un exitoso proceso de formación.

El ejercicio final consiste en la concentración de la temática vista durante tres sesiones de teoría y un ejercicio por clase que eleva el nivel sesión tras sesión, los ejercicios propuestos para el estudiante en cada clase forman una parte esencial del proceso para realizar el ejercicio final,

#### *Capítulo 4.* MANEJO Y DESCRIPCION DEL MATERIAL PARA EL SEMINARIO

por ello es necesario que no se tenga inconveniente con los conceptos dados en el transcurso de las clases y se hayan afianzado de manera continua los códigos y la facilidad para interactuar con la herramienta software, realizando sin mayores inconvenientes las simulaciones y creando en algunos casos los códigos para archivos de entrada y salida según se especifique.

---

## CAPÍTULO 5

---

# CONCLUSIONES

- Hace algunos años la necesidad de herramientas software de simulación para el análisis de sistemas de potencia eran enormes, esto llevó al desarrollo de diferentes software computacionales que hoy en día proporcionan métodos avanzados de solución, rapidez y eficacia en los procesos. Uno de los análisis con mas necesidad de recursos fue la solución del flujo de potencia, al día de hoy este problema ya fue resuelto y es cubierto por una gama de herramientas de diferentes características las cuales ahora crean la pregunta de cuál escoger y colocar en proceso, pues en estos momentos el enigma no es cómo lograr correr un flujo de carga, sino de qué forma es más eficiente. En lo referente a este tema se tomaron como referencia diferentes fuentes y se llegó a la conclusión de que los programas escritos en Python son algunos de los más apropiados a la hora de desarrollar herramientas de análisis de sistemas de potencia, ya que la cantidad de librerías disponibles en este lenguaje es amplia. Tanto las librerías matemáticas como gráficas, se encuentran en aumento y su popularidad es cada vez mayor, estos pueden ser utilizados en diferentes sistemas operativos y presentan un alto nivel de rendimiento. De esta forma se utilizó el paquete de simulación para el flujo de potencia, PyPower, el cual nos presentó una herramienta con grandes ventajas y un número amplio de aplicaciones y posibles análisis de datos.
- Uno de los mayores choques iniciales con las herramientas software fue el trabajo en un entorno consola, sin una interfaz gráfica llamativa o dotada de amplias herramientas, pero aun así de una enorme capacidad de procesamiento, compilación, lectura y escritura de datos y archivos. Si bien algunos programas de software libre no están tan avanzados ni son tan sofisticados como otros de propietario, con el tiempo se ha ido incrementando

## *Conclusiones*

el interés en ellos, haciendo que muchas personas incluso consideren al software como un derecho que tiene la sociedad, lo que puede suponer que en los próximos años la calidad y cantidad de programas de este tipo van a aumentar significativamente debido al aumento en la popularidad de sus adeptos y las ventajas que presenta al cubrir las necesidades de análisis y simulación de datos. Esto se observa con el paquete de simulación PyPower, que es una poderosa herramienta para solucionar flujos de potencia. Además este tipo de herramienta tiene la ventaja académica de permitir al usuario estudiar y modificar el código, lo que le permite al estudiante analizar el algoritmo de solución ya sea con intenciones de mejorarlo o reproducirlo.

- GridLab-D presenta la opción de realizar análisis de flujo de cargas clásico, diseño y modelado de sistemas de transmisión y distribución, control de cargas, respuesta a la demanda y un módulo de mercados que permite simular un entorno de subasta generando archivos de precios que pueden ser leídos en diferentes tipos de facturación para obtener un análisis de precios. Este software presenta capacidades únicas que permiten al usuario realizar un análisis completo de una micro-red y generar diferentes tipos de archivos de salida. Comprender el lenguaje de programación GLD es sencillo, los códigos de los sistemas de potencia no necesitan más que un pensamiento lógico estructurado para generar un archivo de simulación a partir de una estructura o sistemas físico con algunos datos y parámetros necesarios para las simulaciones, GLD está en constante crecimiento y cambio debido a su origen de software libre que permite ampliar sus usos. Gracias a la posibilidad de mejora a cada momento, GLD resulta ser una herramienta ideal para ingenieros diseñadores, estudiantes de ingeniería y analistas de micro-redes.
- Usar herramientas software para el análisis de micro-redes constituye un recurso muy valioso, dado que facilita la comparación con modelos reales establecidos para predecir fallas, identificar falencias y poder actuar rápidamente para brindar una solución eficaz en tiempo real.
- Python presenta propiedades muy atractivas que facilitan el aprendizaje de forma interactiva, así como características especiales en su sintaxis haciendo que los programas escritos en su lenguaje, sean muy legibles. En el entorno de desarrollo integrado Notebook de IPython es posible escribir y correr código, graficar e incluso documentar sobre la marcha, gracias a la interfaz de usuario amigable y cómoda que ofrece Notebook para aprovechar eficientemente las características de IPython.
- Las prestaciones del ATP permiten aplicar esta herramienta en un amplio espectro que abarca desde casos muy simples, (cuyo principal objetivo es generalmente representar las

características más importantes de un determinado proceso transitorio) hasta ejemplos muy complejos y sofisticados.

- El ATP es una herramienta desarrollada en entorno Windows que puede ser aplicada en un elevado y creciente número de estudios, y personalizada por el usuario, para aplicaciones muy específicas. Las principales prestaciones de los scripts, se pueden ampliar fácilmente mediante la adición de programas, ya sea desarrollados especialmente para usuarios ATP o por que sirven de apoyo a otras herramientas de simulación.
- Las prestaciones de los distintos programas que forman el paquete de simulación ATP permiten ampliar la librería de componentes disponibles en el programa ATPDraw de tal forma que un usuario puede aprovechar el potencial de esta herramienta y adaptar el paquete a su medida de forma rápida y eficaz. El presente trabajo ha mostrado como utilizar este potencial en la simulación de sistemas de distribución, aunque muchos de los módulos desarrollados pueden ser de gran utilidad en otro tipo de aplicaciones.

---

## CAPÍTULO 6

---

# RECOMENDACIONES

- El material desarrollado mediante este proyecto proporciona una amplia fuente de información acerca de las herramientas software estudiadas para el análisis de micro-redes, por lo cual se hace hincapié en el manejo apropiado para la realización de una asignatura tipo seminario, teniendo en cuenta el orden de las sesiones, el crecimiento en información y nivel de los ejercicios propuestos para los estudiantes. En el plan de trabajo se explicó la metodología que se quería implementar de haber sido posible la realización del seminario, tiempos de clase, número de sesiones y niveles de logro para impartir el conocimiento relacionado con estos software libres.
- Es indispensable crear un ambiente dinámico de aprendizaje, se recomienda la creación de material visual llamativo, como lo puede ser la presentación en diapositivas de la introducción a las herramientas software y la explicación de las lecturas con imágenes, siguiendo los ejemplos propuestos dentro de las mismas. También se ha hablado de la necesidad de que cada estudiante siga los ejemplos desde un equipo de cómputo y lleve el paso del profesor a la hora de la creación de códigos ejemplo y su simulación para comprender y obtener ese razonamiento lógico que se necesita en la programación de un lenguaje, generando así las bases para enfrentar y solucionar el ejercicio por sesión complementario del ejercicio final, el cual requiere las capacidades en los lenguajes de programación enseñados durante las sesiones.

---

# BIBLIOGRAFÍA

ALLEN Downeey JEREY Elkner, CHRISHD Meyers. 2002. Aprende a Pensar Como un Programador. Green Tea Press, 1(April), Pag 2007, 312.

AATEEQ, M., HABIB, H., UMER, A., & REHMAN, M.U. C++ or Python? Which One to Begin with: A Learner's Perspective of: Teaching and Learning in Computing and Engineering (LaTiCE), International Conference on. 2014 (April), Pages 6469.

CULEBRO, M., Gómez, W., & Torres, S. Software libre vs software propietario. Ventajas y desventajas. Creative Commons, (2006), P 65

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

JASON C. Fuller, CHAS-SIN David P, Pacific Northwest (National Laboratory), . 2012. GridLAB-D, A Unique Tool to Design the Smart Grid. U.S. Department of Energy, Oce of electricity Delivery and Energy reliability (OE1), 1(November), 4.

JASON C. Fuller, CHAS-SIN David P. , Pacific Northwest (National Laboratory), GridLAB-D, The next revolution in power systems simulation. U.S. Department of Energy, Oce of electricity Delivery and Energy, Oce of electricity Delivery and Energy reliability (OE1, 3(November), 2013, Pag 28

LASTRA, Ing. BIANCHI Raul , 1996, ATP Para Inexpertos, IITRE Facultad de Ing. U.de La

## *BIBLIOGRAFÍA*

Plata, 1(February), 69.

MILANO, F., ZHOU, M., & Hou, G. Open Model For Exchanging Power System Data. IEEE PES General Meeting.(2009), P 14.

SANDOVAL Andrés, DOWNEY Allen , JEFFREY Elkner. 2003. Introducción a la programación con Python (2ed). Ponticia Universidad Bolivariana, 2002 (January), Pag 341.

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

VELASCO, Juan A. Martinez. El paquete de simulación ATP. Nuevas prestaciones. Universitat Politècnica de Catalunya, 1(January), 2011. Pag 82.

VICERECTORIA ACADÉMICA, Universidad Industrial de Santander,Lineamientos para el seminario de investigación como modalidad de trabajo para el desarrollo del trabajo de grado, Septiembre, 2007, P 16.

---

# ANEXOS

**NOTA REMISORIA:** LOS ANEXOS COMPLETOS CORRESPONDIENTES A ESTE TRABAJO DE GRADO PUEDEN SOLICITARSE A LA BIBLIOTECA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.

---

## ANEXOS A

---

# PYTHON

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. En los últimos años ha crecido y en la actualidad es uno de los lenguajes de programación más empleados para el desarrollo de software. Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar Windows, Mac OS X y Linux, además, se trata de un lenguaje potente, flexible y con una sintaxis clara y concisa, no requiere dedicar tiempo a su compilación debido a que es un lenguaje interpretado.

Hay buenas razones para elegir Python como lenguaje de programación principal. Dentro de las razones se encuentra que Python es fácil de aprender. Además, cuenta con las estructuras de datos de alto nivel, eficientes, que le permiten escribir operaciones complejas en un menor número de declaraciones que en C, C ++ o Java.

### A.1 GUÍA DE INSTALACIÓN ANACONDA

Anaconda es una colección gratuita de paquetes de gran alcance para Python 2.7, 3.4 (u otras versiones), que permite la gestión a gran escala de datos, análisis y visualización para el estudio y aprendizaje de economía, análisis científico e ingeniería. Esta distribución tiene incluido dentro de sus paquetes intérpretes de lenguaje Python, así como también un entorno de desarrollo integrado denominado, Notebook IPython (N\_IP) que es un entorno computacional interactivo multitarea basado en la web y que actúa como medio facilitador para el desarrollo de la mayoría de las actividades propuestas, pues permite realizar todas las operaciones que generalmente se realizan en Python, ofreciendo una interfaz más amigable y un

## A.2. PYTHON: INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN

manejo ordenado en la escritura del código, lo que resulta mucho más limpio y sencillo.

### A.2 PYTHON: INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN

Esta lectura expone de forma general los conceptos básicos que conforman la sintaxis de Python, algunos de sus tipos estructurados, de control y algunas funciones mediante líneas de código escritas en Nootbook IPython a modo de ejemplos. Esto facilita el estudio porque permite identificar el contenido y desarrollarlo paralelamente. adicionalmente al finalizar la lectura se construye un ejemplo que abarca en gran parte la temática expuesta y se propone un ejercicio como desafío al contenido de la lectura.

### A.3 PYTHON: BIBLIOTECA NUMPY

Esta lectura expone de forma general las características básicas que conforman la biblioteca Numpy de Python, su estructura sintáctica, módulos y algunas funciones de interés, presentado el contenido con imágenes y ejemplos. Adicionalmente se mencionan algunos de los conceptos fundamentales que el lector (estudiante) debe considerar a la hora de continuar con el aprendizaje de este atractivo lenguaje de programación. La segunda lectura afronta elementos del lenguaje que se encuentran dentro del paquete fundamental para computación científica e ingeniería conocido como Numpy. Este paquete es análogo a una biblioteca numérica y proporciona uno o varios objetos para la composición multidimensional array. Objetos derivados (tales como matrices) y una variedad de rutinas para la realización de operaciones de forma rápida con arrays, que incluye matemática, lógica, transformadas de Fourier y álgebra lineal básica. Adicionalmente al finalizar la lectura se construye un ejemplo que abarca en gran parte la temática expuesta y se propone un ejercicio como desafío al contenido de la lectura.

### A.4 PYTHON: ARCHIVOS

Esta lectura proporciona la temática de uno de los usos más representativos de Python, el cual tiene que ver con el procesamiento de datos generados en otro tipo de formato. Por lo tanto se requiere conocer cómo llevar argumentos hacia un archivo de procesamiento por lotes o guion

## *ANEXOS A.* PYTHON

de Python. Importar y exportar archivos, además, de otros procesos como correr programas externos desde Python.

---

## ANEXOS B

---

# PYPOWER

Para el análisis y solución de flujos de potencia existe la librería de Python llamada PyPower (Lincoln, Richard; Power System Engineering Research Center, 2009), dicho paquete de programación permite al usuario realizar flujos de carga y flujo óptimo de grandes redes de potencia ya sea por los métodos de análisis numéricos tales como Newton Raphson, Gauss Seidel o Fast Decoupled. PyPower proporciona una interfaz de línea de comandos (CLI) y una interfaz de programación de aplicaciones Python (API). Las características que hacen del paquete PyPower un software ideal son: la rapidez de compilación, rapidez de escritura del programa, popularidad y disponibilidad de librerías.

### **B.1** GUÍA DE INSTALACIÓN PYPOWER

En esta lectura se mostrara la herramienta computacional Python, enfocado en la instalación del paquete para la simulación y solución de flujo de potencia PyPower, se define el concepto básico de software libre. Así como sus ventajas, desventajas y características primordiales. Al final de la lectura se repasa su funcionamiento desarrollando un ejercicio de flujo de potencia clásico.

## B.2 INTRODUCCIÓN AL PAQUETE DE SIMULACIÓN PYPOWER

El desarrollo de esta guía está basada en el Flujo de carga mediante el uso de la librería de Python llamada PyPower, dicho paquete de programación permitirá realizar Flujos de Carga por varios métodos de análisis, como lo son el flujo de carga con corriente alterna o corriente continua. 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 en el desarrollo de la guía. Además, de la realización de estudios financieros y de bases de datos de flujos de potencia, que permitan el almacenamiento de información de los resultados obtenidos para su posterior uso.

## B.3 FLUJO PROBABILÍSTICO

En la guía encontrará la temática relacionada al flujo probabilístico el cual consiste en generar la solución de un flujo de potencia teniendo en cuenta las incertidumbres sobre los parámetros de los elementos del sistema. En esta lectura se definen conceptos como: tipos de funciones usadas para caracterizar la variabilidad de un parámetro, método de simulación de Monte Carlo, distribuciones de probabilidad. Se reutilizará código Python estudiado en guías pasadas para generar funciones que faciliten procesos para el desarrollo del ejercicio. Como objetivo se desea que el lector comprenda el planteamiento de problema del flujo probabilístico y genere una solución basándose en los conceptos adquiridos en el módulo de PyPower.

---

## ANEXOS C

---

# ATP

El ATP (Alternative Transients Program) es una potente herramienta de simulación estándar, utilizada por la industria de la energía eléctrica para el análisis de fenómenos transitorios y de estado estable que pueden ocurrir en sistemas eléctricos. Actualmente es empleado en un elevado número de aplicaciones que permiten el desarrollar de módulos que representar de forma simple y eficaz elementos o componentes de sistemas de potencia.

### C.1 GUÍA DE INSTALACIÓN ATP

El contenido de esta guía presenta la instalación de la versión recientemente desarrollada del ATP-ATPDraw 5.6. Para ello se proporciona una guía paso a paso para su correcta instalación, prestando atención a los detalles y complementos de la instalación, ya que son indispensables para su correcto funcionamiento y cuya finalidad es facilitar el uso del ATP mediante la opciones que ofrece el ATPDraw. El cual es un pre-procesador gráfico del ATP-EMTP en la plataforma MS Windows. Además se abarca en las prestaciones e importancia del programa en estudio.

### C.2 INTRODUCCIÓN A ATP-ATPDRAW

El contenido de esta guía presenta un resumen de los paquetes más importantes que conforman la herramienta de simulación ATPDraw. Se describen sus principales herramientas

incorporadas en el menú principal y menú de componentes, así como sus principales características. Se desarrolla un ejemplo de simulación en el entorno de trabajo del ATPDraw con el fin de brindar al usuario la guía correcta de edición y simulación que se debe llevar a cabo en los procesos involucrados con el desarrollo de la herramienta. El manejo adecuado de esta herramienta brinda al estudiante una gran ayuda en su proceso de formación como complemento en el proceso de formación académico.

### **C.3** INTRODUCCIÓN A MODELS

Esta guía presenta el estudio del componente MODELS y MODELS Type 94. La cual comienza con una descripción sobre los aspectos básicos de su funcionamiento y aplicaciones, además, de la importancia del lenguaje de programación y su estructura para el proceso de edición y simulación de nuevos componentes de control o de medida. Se desarrolla un ejemplo donde se involucra el proceso de construcción de un modelo y su simulación. Además de un tipo especial de MODELS Type 94 el cual se realizara la descripción detallada de un ejemplo y se describen las maneras en que los modelos se interconectan con el ATPDraw.

### **C.4** FUNCIONES AVANZADAS EN ATP

En esta guía se explora el entorno de las rutinas auxiliares de las que dispone el software ATP. Como lo son \$PARAMETER y DATA BASE MODULE, los cuales son de gran utilidad en el proceso de desarrollar componentes que no se encuentran definidos dentro del paquete de simulación. Se describen las principales características y usos más comunes, también se entrega la información básica del lenguaje de programación utilizado y se explica con ejemplos el proceso de edición que se debe llevar a cabo en el proceso de implementación de cada rutina auxiliar en el ATPDraw.

---

## ANEXOS D

---

# GRIDLAB-D

GridLab-D (GLD) es una nueva herramienta de simulación de sistemas de distribución de energía y análisis de variables en series de tiempo. Proporciona información valiosa para los usuarios que diseñan y operan sistemas de distribución y desean tomar ventaja de las últimas tecnologías de la energía. Incorpora técnicas de modelado avanzado, con algoritmos de alto rendimiento para ofrecer lo mejor en el uso final de modelado.

GridLab-D es un entorno de simulación flexible que puede integrarse con una variedad de gestión y análisis de datos de las herramientas de terceros. El núcleo de GridLab-D tiene un algoritmo avanzado que coordina simultáneamente el estado de millones de dispositivos independientes, cada uno de los cuales está descrito por múltiples ecuaciones diferenciales. Las ventajas de este algoritmo sobre simuladores basados en diferencias finitas tradicionales son:

1. Se ocupa de situaciones inusuales con mucha más precisión.
2. Se encarga de escalas de tiempo muy dispares, que van desde sub-segundos a muchos años.
3. Es muy fácil integrarse con nuevos módulos y sistemas de terceros. En su forma más simple, GridLab-D examina en detalle la interacción de cada parte de un sistema de distribución con otro.

## D.1 GUÍA DE INSTALACIÓN GRIDLAB-D

En esta guía se presenta el proceso de instalación del programa GridLab-D directamente de la pagina oficial, mediante vínculo: <http://www.gridlabd.org/>. Para ello se contempla el proceso estructurado para su correcta descarga e instalación. Es importante prestar atención a los detalles y complementos de la instalación ya que son indispensables para su optimo funcionamiento.

## D.2 INTRODUCCIÓN A GRIDLAB-D

En esta lectura de introducción se explica el porqué de usar el software, sus ventajas y desventajas, principales características y usos más comunes, también se entrega la información básica del lenguaje de programación de GridLab-D y se explica con algunos ejemplos simples el uso y funcionamiento de la consola GLD. Se proporciona la plantillas para caracterizar en codigo GLD algunos de sus elementos. Al terminar la temática de la sección se resuelve un ejemplo y se plantea un ejercicio para resolver de forma individual basándose en los conocimientos adquiridos durante la sesión y el repaso de esta lectura.

## D.3 MICRO-REDES, SISTEMA DE TRANSMISIÓN Y DISTRIBUCIÓN EN GRIDLAB-D

En esta lectura encontrara la temática relacionada a la definición de las micro-redes y su importancia en el desarrollo de los sistemas energéticos, inteligentes y amables con el medio ambiente. Se explicara como GridLab-D sirve como una nueva herramienta de simulación para pequeños y grandes sistemas de transmisión y distribución comunes y de energías renovables. Se planteando un ejercicio que concentra la temática tratada y pone en practica los conceptos adquiridos por los asistentes al seminario. Los temas centrales son el modelamientos de los elementos y dispositivos de un sistemas de potencia y caracterización de equipos y cargas, se adjuntan las plantillas de caracterización de los electrodomésticos y se resuelve el flujo de potencia por los dos métodos de solución de los cuales dispone GLD. La lectura termina la lectura dando un ejemplo de un sistema de distribución de prueba de la IEEE.

## D.4 MERCADOS ELÉCTRICOS, SUBASTA Y FACTURACIÓN

En esta lectura encontrara la temática relacionada a los mercados de energía eléctrica y respuesta a la demanda como una herramienta de control por parte el usuario, para controlar el consumo de las cargas y emplear el uso racional de la energía eléctrica. Se muestra como GLD es una nueva herramienta de simulación para pequeños y grandes sistemas de transmisión y distribución comunes y de energías renovables, se termina la lectura dando un ejemplo de un sistema de distribución de prueba de la IEEE y el ejercicio correspondiente a la temática para realizar por parte del estudiante de forma individual. Los temas centrales son los tipos de facturación y la respuesta a la demanda, de estas herramientas se puede obtener un análisis financiero ligado a la obtención de precio en diferentes modos de simulación de subasta.