

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y SISTEMAS
SECCIÓN DE POSGRADO



**PROPUESTA DE UNA BASE
COMPUTACIONAL PARA DESARROLLO
DE SOFTWARE PARA SISTEMAS DE
ENERGÍA ELÉCTRICA**

TESIS

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO
EN CIENCIAS CON MENCIÓN EN:
INGENIERÍA DE SISTEMAS**

JORGE ISAAC RODRIGUEZ RODRIGUEZ

LIMA - PERÚ
2007

*Dedico este trabajo a
toda mi familia y amigos.*

*En especial a
mis padres Isaac y Judith.*

AGRADECIMIENTOS

A Dios, que me permitió alcanzar mis objetivos.

A mis padres Isaac Rodríguez Quevedo y Judith Rodríguez Gamarra, fuentes de inspiración y ejemplo de constante superación, por el cariño y preocupación mostrados a lo largo de estos años. A mis hermanos José, Carlos y Johana por el continuo cariño expresado en el decorrer de estos años. A Denisse Lilian Malpartida Damián por el cariño y comprensión durante los años en que realicé la Maestría.

Al profesor MSc. Zalatiel Carranza Ávalos por la orientación, apoyo e incentivo, cruciales para el desarrollo del presente trabajo. Al grupo de profesores de la Sección de Posgrado FIIS - UNI: MSc. Alberto Un Jan, MSc. Celedonio Méndez, Dr. David Mauricio, Dra. Gloria Teresita Huamaní, Dr. Pedro Espinoza Haro y MSc. Tino Reyna, por las enseñanzas vertidas tanto dentro como fuera de las aulas de clase. A los profesores de la COPPE / UFRJ (Universidade Federal do Rio de Janeiro): PhD. Djalma Mosqueira Falcão y PhD. Glauco Nery Taranto, por la amistad y comprensión durante el desarrollo del presente informe.

A mis amigos de la FIIS - UNI: Antonio Suárez, Carlos Loli, Carlos Rodríguez, Daniel Llanos, Oscar Castro, Rubén Borja, y a todos mis colegas de aulas que con su amistad hicieron de mi permanencia en la Maestría un recuerdo inolvidable. A los programas de fomento CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) y CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) por el apoyo financiero.

ÍNDICE

DESCRIPTORES TEMÁTICOS	vii
RESUMEN	viii
ABSTRACT	ix
INTRODUCCIÓN	1
Generalidades	1
Formulación del problema	6
Hipótesis	8
Objetivos	8
Justificación	9
Metodología	10
Estructura del informe	10
1. CAPÍTULO I MARCO TEÓRICO	12
1.1. Introducción	12
1.2. Software	14
1.3. Ingeniería de Software	19
1.4. Paradigma de Orientación a Objetos (POO)	26
1.5. Unified Modeling Language (UML)	34
1.6. Sistemas distribuidos	42
1.7. Common Object Request Broker Architecture	45

2. CAPÍTULO II SISTEMAS ELÉCTRICOS DE ENERGÍA	49
2.1. Introducción.....	49
2.2. Términos básicos de electricidad	50
2.3. Análisis de circuitos eléctricos	53
2.4. Sistemas Eléctricos de Energía (SEE)	67
2.5. Flujo de potencia	75
2.6. Estabilidad de SEE.....	83
3. CAPÍTULO III BASE COMPUTACIONAL DE SOFTWARE PARA SEE.....	91
3.1. Introducción.....	91
3.2. Estado actual de los simuladores de SEE.....	92
3.3. Revisión bibliográfica.....	94
3.4. Requerimientos del sistema de software	101
3.5. Análisis orientado a objetos.....	104
3.6. Diseño orientado a objetos.....	109
3.7. Implementación	116
4. CAPÍTULO IV EXPANSIÓN DE LA TRANSMISIÓN.....	117
4.1. Introducción.....	117
4.2. Formulación matemática clásica o DC	119
4.3. Técnicas de optimización utilizadas	121
4.4. Algoritmos genéticos	125
4.5. Formulación matemática propuesta o AC	132
4.6. Metodología AC propuesta.....	135
4.7. Implementación de la metodología AC extendiendo la base computacional propuesta	142
5. CAPÍTULO V ANÁLISIS DE RESULTADOS	145
5.1. Introducción.....	145
5.2. Sistema eléctrico WECC	146
5.3. Sistema eléctrico Garver	158

CONCLUSIONES Y RECOMENDACIONES	171
Conclusiones.....	171
Recomendaciones.....	172
GLOSARIO DE TÉRMINOS	173
BIBLIOGRAFÍA	175
ANEXOS	188
A. Datos de los sistemas de prueba	189
B. Índice de Tablas	193
C. Índice de Figuras	194

DESCRIPTORES TEMÁTICOS

1. Ingeniería de Software
2. Paradigma de Orientación a Objetos.
3. Unified Modeling Language.
4. Common Object Request Broker Architecture.
5. Simulación de Sistemas Eléctricos de Energía.
6. Estabilidad de Sistemas Eléctricos de Energía.
7. Expansión de la transmisión eléctrica.

RESUMEN

En este trabajo se presenta la aplicación de conceptos relacionados a la Ingeniería de Software - como el Paradigma de Orientación a Objetos (POO), el *Unified Modeling Language* (UML) y la especificación *Common Object Request Broker Architecture* (CORBA) - en la propuesta de una base computacional capaz de dar soporte al desarrollo integrado de un amplio conjunto de simuladores de Sistemas Eléctricos de Energía (SEE). Las estructuras de clases presentes en esta base representan no sólo a los elementos físicos que componen el SEE, si no también a conceptos abstractos como las metodologías de las diversas simulaciones aplicadas en el SEE incluyendo sus operaciones matemáticas. Dos prototipos son implementados para probar la validez de la propuesta. El primer prototipo implementa una versión reducida de la base computacional usando una arquitectura cliente - servidor siguiendo la especificación CORBA. El cliente, implementado utilizando Java, es un editor gráfico para la creación / edición de diagramas que representen al SEE. El servidor, implementado utilizando C++, es el encargado de los cálculos de la simulación "flujo de potencia". El segundo prototipo implementa la versión actual de la base computacional sin la utilización de CORBA y utilizando C++. Este prototipo considera también las simulaciones "estabilidad permanente" y "estabilidad transitoria". Los resultados obtenidos son comparados satisfactoriamente con los resultados de simuladores comerciales. La facilidad y utilidad de cómo extender y reutilizar la base propuesta se demuestra al resolver una formulación del análisis complejo "expansión de la transmisión" con resultados satisfactorios.

ABSTRACT

This work presents the use of Software Engineering concepts – such as Object-Oriented Paradigm (OOP), Unified Modeling Language (UML) and Object Request Broker Architecture (CORBA) specification – in a proposal of a computational base able to give support to the integrated development of an ample set of Electric Power Systems (EPS) simulators. The class structures in this base represent not only the physical elements that compose the EPS, but also to abstract concepts like the methodologies of the diverse analyses or simulations applied in the EPS included their mathematical operations. Two prototypes are implemented to prove the present proposal. The first prototype implements a reduced version of the computational base using a client – server architecture according to CORBA specification. The client, implemented using Java, is a graphical editor for the creation/edition of diagrams that represent the EPS. The server, implemented using C++, is in charge to make the corresponding calculations for a simulation “power flow”. The second prototype implements the present version of the computational base without the use of CORBA, using C++. This prototype also considers the simulations "small-signal stability" and "transient stability". The results of each one of these simulations are compared satisfactorily with the results of commercial simulators. The facility and utility of how to extend and to reuse the computational base proposed are demonstrated when solving a formulation of the complex analysis "expansion network planning" with satisfactory results.

INTRODUCCIÓN

GENERALIDADES

La electricidad es una necesidad primaria de las personas, necesaria para proveer luz artificial y como fuente de energía para la mayoría de equipos que tornan nuestra vida más confortable. La cantidad necesaria de electricidad que requiere una población es llamada demanda eléctrica. Por razones técnico – económicas, la electricidad no puede ser almacenada en grandes cantidades, por lo que tiene que ser producida y entregada en el instante justo en que es requerida. Esto sólo es posible gracias al adecuado funcionamiento de los llamados Sistemas Eléctricos de Energía (SEE).

Un SEE o Sistema Eléctrico de Potencia (SEP), es un conjunto de elementos físicos y personas que interactúan entre sí con la finalidad de atender completamente la demanda eléctrica con adecuados niveles de seguridad, calidad e incurriendo en el menor costo total posible. El término “seguridad” está relacionado con la capacidad del sistema de seguir funcionando ante la presencia de diversos disturbios, producto de la influencia del medio ambiente o del irregular funcionamiento de algún elemento físico. El término “calidad” está relacionado con la necesidad de que algunas propiedades físicas cuantificables (de los elementos físicos) se encuentren dentro de determinados límites, de modo que aseguren un adecuado funcionamiento de los equipos de la demanda eléctrica. Es natural deducir que la búsqueda de niveles de seguridad y calidad en un SEE es conflictuante con la búsqueda de la minimización de su costo total.

El adecuado funcionamiento de los SEE es garantizado mediante estudios de expansión y operación de los mismos, estudios que sólo son posibles gracias a la utilización de modelos. Un modelo es una abstracción, una representación simplificada de un sistema que facilita su estudio, permite efectuar simulaciones y obtener conclusiones útiles, conclusiones que sólo serán válidas para la mencionada representación simplificada. Un modelo es necesario dada la imposibilidad de considerar todos los aspectos de la compleja realidad.

Los modelos utilizados para los diversos estudios concernientes a los SEE han evolucionado a través del tiempo. Antes de la década del 50 eran utilizados modelos análogos físicos que en escala reducida representaban ciertos aspectos correspondientes a grandes áreas de los SEE; llamados analizadores de red. Estos modelos se encontraban dentro de laboratorios y servían para cálculos y simulaciones que hoy serían considerados demasiados simples y aproximados. A finales de la década del 50 comenzó el uso de modelos simbólicos, específicamente modelos matemáticos. Estos modelos utilizan símbolos para designar propiedades físicas permitiendo expresar cientos de relaciones que pueden ser resueltas con la utilización de métodos implementados en un computador digital (métodos computacionales). Esta última característica fue fundamental para que en las décadas del 60 y del 70 el desarrollo de métodos computacionales para simulación de los SEE ganase un enorme impulso.

El comportamiento de los elementos físicos que conforman un SEE, es fuertemente dependiente del fenómeno físico cuya influencia se desea analizar. Dado que existen una gran variedad de fenómenos físicos de interés que pueden actuar en un SEE, y las limitaciones propias de los computadores en estas décadas (60 y 70), se hizo necesario el desarrollo de diferentes modelos matemáticos y diferentes métodos computacionales, originando el surgimiento de especializaciones bien diferenciadas dentro del área de simulación digital de los SEE.

Especializaciones como “flujo de potencia óptimo” [DOMM68], “transitorios electromagnéticos” [DOMM69], “estimación de estados” [SCHW70], “flujo de potencia desacoplado” [STOT74], devinieron en simuladores digitales (*softwares*) que priorizaban las características propias de su respectiva especialización e prácticamente desconsideraban características correspondientes a otras especializaciones. Estos *softwares* fueron implementados por programadores solitarios utilizando lenguajes de programación procedural (Fortran, C). El gran suceso obtenido en esas décadas de la programación procedural creó en el desarrollo de *software* para SEE un caso extremo de “*statu quo*” (“mantengamos el estado actual”), motivo por el cual estos *softwares* han ido creciendo y logrando un alto nivel de sofisticación y madurez, convirtiéndose en *softwares* de nivel comercial [CEPE97] [CEPE99] [CEPE01].

En los últimos años la aplicación de una nueva reglamentación en el sector eléctrico trajo consigo un proceso, aún inconcluso, de reestructuración y privatización de los SEE [ILIC98]. El Perú no es ajeno a esta realidad [LCE92] [RLCE93]. En las actuales estructuras la generación, transmisión y distribución de la energía eléctrica son consideradas separadas y pasan de ser propiedad estatal para ser propiedad del capital privado (agentes privados) fiscalizados por el estado (agente regulador). La electricidad es ahora considerada una mercadería y comercializada a través de transacciones económicas, donde los agentes privados concurren entre sí ya no buscando minimizar sus costos sino maximizar sus respectivos lucros, creando así un ambiente competitivo.

El establecimiento de un ambiente competitivo genera nuevos desafíos y/o nuevas problemáticas, originando la aparición de nuevos elementos físicos y métodos de solución. Operativamente, también exige el máximo aprovechamiento de los SEE, forzando una operación cada vez más próxima de los límites operacionales, insertando nuevos factores que incrementan las posibles fuentes de disturbios.

Las decisiones de los diversos agentes son soportadas por simulaciones digitales. En el Perú prácticamente todos los simuladores digitales (*softwares*) utilizados son de origen extranjero. El único desarrollo nacional documentado fue encontrado en [PAJA02], el cual es un *software* confiable y de gran porte mas que al igual que sus congéneres [CEPE97] [CEPE99] [CEPE01] no fue diseñado para una realidad cambiante y cada vez más compleja como la vivida actualmente.

Es en este contexto que los agentes necesitan de mecanismos modernos y confiables para analizar (simular) los SEE, que representen de forma eficiente sus nuevas características tanto técnicas como económicas. Muchas áreas de investigación deben ser impulsadas, desde áreas técnicas relativas a los SEE pasando por investigaciones que envuelvan cuestiones económicas en conjunto con aspectos técnicos, hasta áreas como la Ingeniería de Software [AGOS02].

Conceptos relacionados a la Ingeniería de Software (la aplicación de la ingeniería en el desarrollo de *software*) [PRES02], como el “paradigma de orientación a objetos” (POO) (paradigma que representa en objetos *software*, objetos que se imaginan existen en el mundo real) [TAYL97]; el “*Unified Modeling Language*” (UML) (lenguaje de modelado orientado a objetos) [BOOC98], los “sistemas distribuidos” (colección de computadores autónomos conectados a través de una red que funcionan como un único e integrado sistema de computación) [COUL00], la especificación “*Common Object Request Broker Architecture*” (CORBA) (especificación de una arquitectura desarrollada en un ambiente distribuido y heterogéneos que utiliza el POO) [LANG02], etc., han sido ampliamente aceptados en el mundo de los negocios consiguiendo suceso en los últimos años, facilitando a las personas a escribir, distribuir, reutilizar y extender paquetes existentes de *software*. Se han realizado diversas investigaciones para evaluar la utilización de algunos de estos conceptos para la simulación digital de los SEE, las cuales pueden ser divididas en dos generaciones.

La primera generación comenzó a inicios de la década del 90 y buscaba demostrar la viabilidad de la utilización del POO para la realización de una simulación en particular [NEYE90] [HAVA94] [ZHOU96] [MANZ99]. A finales de la década del 90, los avances de las arquitecturas de *hardware* y consolidación de los compiladores basados en el POO hicieron que la preocupación con la aplicabilidad del POO adquiriese menor relevancia; dando origen a la segunda generación, donde el gran desafío pasó a ser el de encontrar una estructura de clases que represente adecuadamente al SEE y que sirva como base para una implementación integrada de diversos simuladores [AGOS02] [MOHA04] [RODR05] [MANZ05].

El presente trabajo se enmarca dentro de esta segunda generación y utiliza los conceptos de la Ingeniería de Software (POO, UML, sistemas distribuidos, CORBA, etc.) para proponer una base computacional capaz de dar soporte a la construcción integrada de un amplio conjunto de simuladores digitales de SEE. Un único *software* capaz de realizar todas las simulaciones de interés en los SEE aún es una utopía, por lo que para demostrar la efectividad de la base propuesta, uno de sus componentes principales es implementado proporcionando posterior soporte a la implementación de tres métodos computacionales representativos de tres especialidades diferentes: el “flujo de potencia” (el cual determina un estado particular del SEE) [MONT83], la “estabilidad permanente” (el cual determina el comportamiento del SEE ante ocurrencia de una pequeña perturbación) [KUND04] y la “estabilidad transitoria” (el cual determina el comportamiento del SEE ante la ocurrencia de una gran perturbación) [KUND04].

Un problema actual en el área de Ingeniería Eléctrica que requiere de varias especialidades para su resolución es el “planeamiento de la expansión” (problema de optimización que consiste en determinar el mejor conjunto de elementos físicos que deben ser adicionados al SEE para que una demanda futura sea atendida convenientemente incurriendo en el menor costo posible) [LATO03].

El hecho de que varias especialidades sean necesarias para la resolución de un problema lo torna de “naturaleza global”. Quien intente resolver este tipo de problemas deberá tener presente que las especialidades no son independientes unas de otras; que nuestra mente deseando comprender mejor el problema para resolverlo, es quien lo divide y clasifica siguiendo las llamadas especialidades; y que tanto más granular sea la especialización, mayor será el peligro de perder la visión global del problema. Una manera de minimizar esta dificultad es formar un equipo de especialistas de diversas áreas constituyendo un equipo multidisciplinar para resolver un determinado problema. El objetivo es llevar al estudio del problema las diferentes maneras de encarar la realidad por los especialistas de cada área del conocimiento humano. Este aspecto es muchas veces más importante que los conocimientos específicos de cada área.

El problema del “planeamiento de la expansión” es un problema de naturaleza global. Este problema es actualmente resuelto en varias etapas; en cada una de las cuales son utilizados diferentes simuladores digitales; sin embargo, se reconoce la necesidad de una formulación como un problema de optimización del todo y no sólo de las partes consideradas aisladamente [SILV05]. El presente trabajo demuestra la facilidad y utilidad de cómo extender la base computacional propuesta, al resolver una formulación que integra dos etapas del planeamiento de la expansión: “expansión de la transmisión” y “análisis del comportamiento [dinámico]”, en una sola llamada “expansión de la transmisión AC”, formulación para la cual no existe, hasta la actualidad, métodos de solución implementados o reportados en la literatura especializada.

FORMULACIÓN DEL PROBLEMA

La propuesta de una base computacional para desarrollo de simuladores digitales (*software*) para SEE es un problema multidisciplinario, que puede ser resuelto con la resolución de los siguientes sub-problemas:

Sub-problemas

- Representar al SEE, sus diversos elementos físicos y la forma cómo estos colaboran para proveer un comportamiento de alto nivel (descripción topológica), que sea útil para diversas especialidades dentro del área de la simulación digital de los SEE.
- Representar de una manera integrada métodos computacionales tradicionales de simulación pertenecientes a diferentes especialidades como el “flujo de potencia”, la “estabilidad permanente” y la “estabilidad transitoria.
- Representar métodos computacionales de naturaleza global, como la “expansión de la transmisión AC”, de una manera en que reutilice métodos computacionales tradicionales.

Formulación de los sub-problemas

- Computacionalmente el problema de la descripción topológica consiste en modelar el SEE como estructuras que sean independientes de las simulaciones que se requieran realizar.
- Matemáticamente el problema de “flujo de potencia” consiste en resolver un sistema de ecuaciones algebraicas no lineales, el problema de “estabilidad permanente” consiste en resolver un sistema de ecuaciones diferenciales lineales y el problema de “estabilidad transitoria” consiste en resolver un sistema de ecuaciones diferenciales no lineales en conjunto con un sistema de ecuaciones algebraicas no lineales.
- Matemáticamente el problema de la “expansión de la transmisión AC” consiste en resolver un problema de optimización no lineal entero mixto con una función objetivo que minimiza el costo incurrido sujeto a restricciones técnicas que consideren el comportamiento dinámico de los elementos físicos confortantes del SEE.

HIPÓTESIS

Hipótesis

Es posible obtener un modelo (estructura) computacional del SEE que sirva como base para la implementación de una gran variedad de simuladores tradicionalmente implementados separadamente.

Sub-Hipótesis

- Es posible obtener un modelo computacional que represente los diversos elementos físicos del SEE y la forma cómo estos colaboran para proveer un comportamiento de alto nivel que sea útil para diversas especialidades dentro del área de la simulación de los SEE.
- Es posible obtener un modelo computacional (único) que represente de una manera integrada métodos computacionales tradicionales de simulación pertenecientes a diferentes especialidades como el “flujo de potencia”, la “estabilidad permanente” y la “estabilidad transitoria”.
- Es posible extender los modelos computacionales propuestos para la solución de problemas de naturaleza global (que necesiten de diferentes especialidades), como la “expansión de la transmisión AC”.

OBJETIVOS

Objetivo general:

- Contribuir con el desarrollo de *software* nacional para simulación de SEE dentro del contexto de ambientes competitivos.

Objetivos específicos:

- Aplicar conceptos de Ingeniería de Software para proponer una base computacional que pueda ser extendida por diversos especialistas, permitiéndoles la implementación integrada de diversas metodologías computacionales para simulación de SEE.

- Implementar parcialmente la base computacional, para dar soporte a la implementación de los siguientes métodos computacionales tradicionales: “flujo de potencia”, “estabilidad permanente” y “estabilidad transitoria”.
- Proponer una formulación para “la expansión de la transmisión AC”. Implementar un método computacional que resuelva la formulación propuesta reutilizando y/o extendiendo la base computacional anteriormente implementada.
- Aplicar el prototipo de *software* desarrollado en el estudio de algunos sistemas eléctricos prácticos.

JUSTIFICACIÓN

Un requerimiento principal de cualquier sociedad moderna es la adecuada operación de su SEE. Este objetivo natural e importante, es garantizado con las diversas simulaciones digitales efectuadas a los SEE. La simulación de SEE es una tarea cuya dificultad tiende a incrementarse con el crecimiento del sistema, interconexiones con otros SEE y principalmente con el establecimiento de un ambiente competitivo.

En un ambiente competitivo los requerimientos de los simuladores cambian a la velocidad que la realidad cambia, por lo que se torna importante poseer estructuras flexibles que asimilen en un tiempo hábil estos cambios, cada vez más constantes.

La utilización [de los avances obtenidos en los últimos 20 años en el área] de la Ingeniería de Software en el proceso de desarrollo de *software* para simulación digital de los SEE se presenta como una buena alternativa para la consecución de este objetivo, por lo que investigaciones que exploren las ventajas y desventajas de su aplicabilidad deben ser impulsadas.

METODOLOGÍA

En el desarrollo del presente trabajo se utilizó la metodología de investigación científica, basada en las técnicas de análisis y síntesis.

Inicialmente se elaboró una revisión bibliográfica, para obtener el estado del arte referente al modelado de los Sistemas Eléctricos de Energía desde la óptica de la Ingeniería de Software.

Seguidamente se propuso una plataforma; la cual fue parcialmente implementada y sirvió como base para la implementación integrada de métodos computacionales correspondientes a las siguientes especialidades: “flujo de potencia”, “estabilidad permanente” y “estabilidad transitoria”.

Luego se elaboró una revisión bibliográfica para obtener el estado del arte referente al problema de la “expansión de la transmisión AC”; y se propuso una [correspondiente] formulación, que fue resuelta con la implementación de un método computacional extendiendo la base computacional existente.

Finalmente, el prototipo de *software* implementado se probó utilizando sistemas eléctricos prácticos.

ESTRUCTURA DEL INFORME

El desarrollo del informe comienza en el capítulo 1, con la presentación de conceptos básicos correspondientes a la Ingeniería de Software que serán utilizados en capítulos posteriores. Se detallan los conceptos correspondientes a *software*, Ingeniería de Software, POO, UML, sistemas distribuidos y la especificación CORBA.

En el capítulo 2 se presentan los conceptos básicos de electricidad necesarios para entender la estructura, comportamiento y simulación de los SEE. En este capítulo se comentan resumidamente los métodos para análisis de circuitos eléctricos que incorporan sistemas de control; se

describe resumidamente el funcionamiento de los SEE; se presentan los fundamentos matemáticos de la simulación más importante en un SEE: el flujo de potencia; y se comentan muy superficialmente dos simulaciones relevantes: la estabilidad permanente y la estabilidad transitoria.

En el capítulo 3 se presenta la revisión bibliográfica sobre la utilización de conceptos presentados en el capítulo 1 en el desarrollo de simuladores digitales de SEE. En este capítulo también se presenta una propuesta de una plataforma que servirá como base computacional para la representación e implementación de las simulaciones tradicionales presentadas en el capítulo anterior.

En el capítulo 4 se presentan las formulaciones actuales utilizadas en la resolución del problema de la “expansión de la transmisión”. En este capítulo también se presenta una formulación académica para el problema mencionado adicionando restricciones de estabilidad permanente y transitoria, restricciones que hasta la actualidad no han podido ser incorporadas. Se presenta también un método computacional para resolver la formulación propuesta, extendiendo la base computacional presentada en el capítulo anterior.

En el capítulo 5 se presentan y comentan los resultados de la implementación de la base propuesta en el capítulo 3 para un sistema eléctrico ampliamente utilizado en la literatura especializada, el *Western Electricity Coordinating Council (WECC)*. Estos resultados se compararon a los resultados obtenidos de simuladores digitales comerciales. En este capítulo también se presentan y comentan los resultados de la implementación de la formulación propuesta en el capítulo 4 para el sistema eléctrico Garver, sistema para el cual la respuesta óptima global sin la inclusión de restricciones de estabilidad ya es conocida.

Finalmente se presentan las conclusiones del presente trabajo, así como recomendaciones para posibles trabajos futuros.

CAPÍTULO I

MARCO TEÓRICO

1.1. INTRODUCCIÓN

El impacto del *software* de computadora [PRES02] (o simplemente *software*) en nuestra sociedad y en la cultura es cada vez más profundo. Él está inmerso en sistemas de todo tipo, desde aquellos que nos conducen a la toma de decisiones comerciales hasta aquellos que sirven de base para la investigación científica moderna y resolución de problemas de ingeniería.

En los últimos 50 años el *software* ha pasado de ser una herramienta de análisis de información a ser una industria por sí misma. A diferencia de la industria del *hardware* que encontró su camino de suceso por medio de la utilización de componentes, la industria del *software* aún está en camino de descubrir las formas en que [la curva que representa] su desarrollo finalmente se estabilice en un nivel sin fallas. Los primeros *softwares* desarrollados (hasta aproximadamente la década del 70) no atingían grandes dimensiones (considerando las propias limitaciones del *hardware* existente) En esa época no se tenía claro las etapas del proceso de desarrollo y el modelado no utilizaba métodos o formalismos, siendo expresados generalmente por textos o lenguaje natural. Prácticamente la única herramienta utilizada era el diagrama de flujo.

En la década de los 80 importantes marcos fueron fijados en la industria del *software*: surge el concepto de Ingeniería de Software, Dijkstra escribe sobre programación estructurada, Codd describe el modelo

relacional, Ritchie desarrolla el lenguaje C y De Marco populariza el análisis estructurado. Es así que se formalizan las bases (proceso, métodos y herramientas) de la Ingeniería de Software, proporcionando un marco de trabajo en el desarrollo de *softwares* con compromisos de calidad.

Los avances listados son resumidos generalmente en una metodología denominada de "estructurada". Actualmente se poseen óptimos *softwares* desarrollados con la utilización de esta metodología, mas aún existen problemas inherentes a la propia metodología. Por ejemplo: existen dificultades en garantizar compatibilidad entre las fases del proceso de desarrollo, los modelos creados no son expresivos siendo de difícil comprensión por los usuarios, la modificación de modelos requiere grandes esfuerzos, etc. Estas dificultades sumadas al incremento de complejidad, urgencia y requisitos de adaptabilidad en los *softwares* actuales, llevó a repensar los pros y contras de esta abordaje.

En la década de los 90 la consolidación del POO permitió que la Ingeniería de Software madurase completamente como ingeniería, cubriendo muchas de las dificultades mencionadas recientemente. El POO proveyó las bases para la aparición (o desarrollo) de procesos y métodos que exploten su utilización en el desarrollo de *software*. Permitted también la aparición de una estandarización para la documentación de los mencionados procesos y métodos: el UML. El POO abrió las puertas para diversos abordajes en el desarrollo de *software*, como por ejemplo el de sistemas distribuidos y la utilización de especificaciones de arquitectura como CORBA. Esto no quiere decir que existe la perfección completa con la utilización del POO, problemas aún existen, mas es innegable que con su utilización se ha evolucionado mucho.

En este capítulo se describen los principales conceptos de la Ingeniería del Software empleados en los capítulos posteriores, presentándolos de una manera simple, para un mejor acompañamiento y entendimiento del presente informe.

1.2. SOFTWARE

1.2.1. DEFINICIÓN

El término *software* es definido como: “la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema basado en computador” [IEEE90]. Un sistema basado en computador es definido como “un conjunto o disposición de elementos que están organizados para realizar un objetivo predefinido procesando información” [PRES02].

El *software* es el producto que diseñan y construyen los ingenieros de *software*. Producto que permite al *hardware* de la computadora desempeñar trabajo útil. Producto que abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números y texto. Producto que brinda al usuario información resultante que mejora de algún modo su modo de vivir.

El papel del *software* ha sufrido un cambio significativo durante un periodo de tiempo superior a 50 años. Enormes mejoras en rendimiento del *hardware*, profundos cambios de arquitecturas informáticas, grandes aumentos de memoria y capacidad de almacenamiento y una gran variedad de opciones de entrada y salida han conducido a sistemas más sofisticados y más complejos basados en computadora.

El programador solitario de antaño ha sido reemplazado por un equipo de especialistas del *software*, cada uno centrado en una parte de la tecnología requerida para entregar una aplicación concreta. Y de este modo, las cuestiones que se preguntaba el programador solitario son las mismas cuestiones que nos preguntamos cuando construimos sistemas modernos basados en computadoras: ¿Por qué lleva tanto tiempo terminar el *software*?, ¿Por qué son tan elevados los costos de desarrollo?, ¿Por qué nos resulta difícil constatar el progreso conforme se desarrolla el *software*?

1.2.2. CARACTERÍSTICAS

Para poder comprender lo que es el *software*, es importante examinar sus características que lo diferencian de otras cosas que el hombre puede construir. Cuando se construye *hardware*, por ejemplo, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce fácilmente en una forma física; mas el *software* es un elemento del sistema que es lógico en lugar de físico. Por tanto el *software* tiene unas características considerablemente distintas a las del *hardware*:

- El *software* se desarrolla, no se fabrica en un sentido clásico. En el *software*, la buena calidad se garantiza prácticamente mediante un buen diseño. La relación entre el número de personas que trabajan en el desarrollo de *software* y la productividad total no es lineal. Los costos del *software* se encuentran casi totalmente en la ingeniería. Esto significa que los proyectos de *software* no se pueden gestionar como si fueran proyectos de fabricación.
- El *software* no se estropea. En las Figuras 1.1 y 1.2 se describen la proporción de fallas como una función del tiempo para el *hardware* y el *software*, respectivamente. Se aprecia claramente que ambos poseen muchas fallas al principio de su vida (atribuibles a defectos del diseño o fabricación). Sin embargo, conforme pasa el tiempo, el *hardware* empieza a desgastarse y su tasa de fallas se incrementa, lo que no sucede con el *software* (curva ideal). La implicación es clara, el *software* no se estropea. ¡Pero se deteriora!. El deterioro es originado por la introducción de nuevos defectos a cada cambio de requisitos durante la etapa de mantenimiento (curva real).
- Aunque la industria tiende a ensamblar componentes, la mayoría del *software* se construye a medida. Los componentes reutilizables se han creado para que el ingeniero pueda concentrarse en elementos verdaderamente innovadores de un diseño. En el mundo del

hardware, la reutilización de componentes es una parte natural del proceso de ingeniería. En el mundo del *software* es algo que sólo ha comenzado a lograrse a escala amplia. El componente de *software* debería diseñarse e implementarse para que pueda volver a ser reutilizado en muchos programas diferentes.

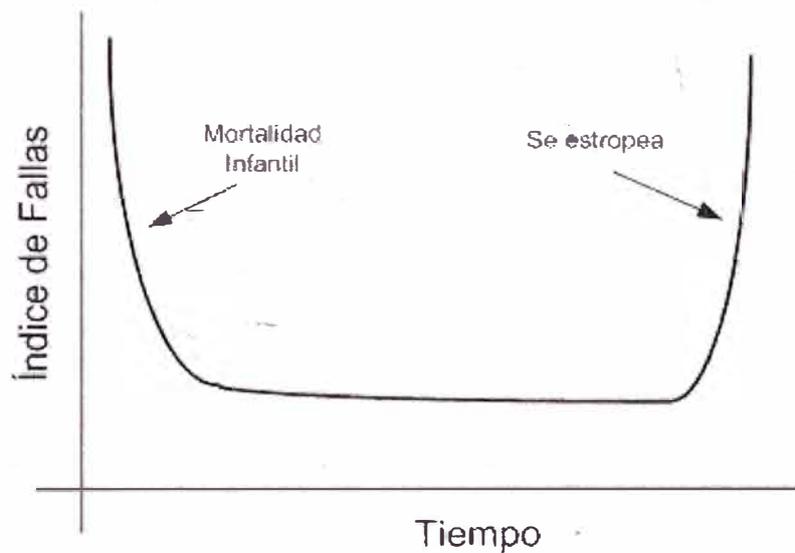


Figura 1.1: Curva de fallas del hardware.

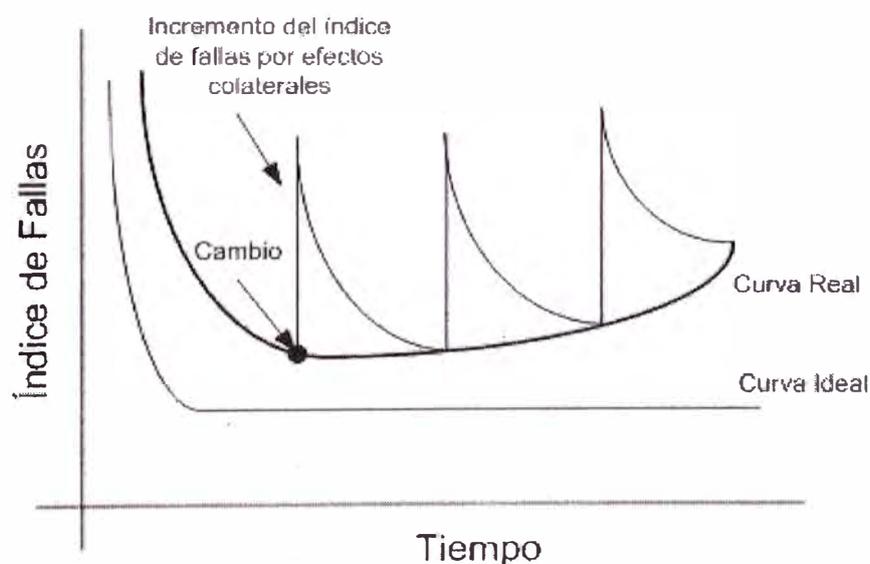


Figura 1.2: Curva de fallas real e idealizada del software

1.2.3. APLICACIONES DEL SOFTWARE

El *software* puede aplicarse en cualquier situación en la que pueda definirse un algoritmo (s). Se entiende por algoritmo a un conjunto ordenado y finito de operaciones para hallar la solución a un problema.

Definir categorías genéricas para las aplicaciones de *software* que sean significativas es una tarea que se complica en la medida que aumenta la complejidad del *software*. Una tentativa de clasificación puede ser lograda considerando el contenido y el determinismo de los datos procesados. El contenido se refiere al significado y a la forma de los datos de entrada y salida. El determinismo se refiere a la predecibilidad del orden y del tiempo de llegada de los datos. Con estas consideraciones, las siguientes áreas del *software* pueden indicar la amplitud de las aplicaciones potenciales:

- Software de sistemas. Es el *software* desarrollado para servir a otro (s) *software* (por ejemplo: ciertos componentes del sistema operativo, compiladores, editores, etc.). El área del *software* de sistemas se caracteriza por una fuerte interacción con el *hardware*, gran utilización por múltiples usuarios, sofisticada gestión de procesos, operación concurrente planificada, compartimiento de recursos, estructura de datos complejas y múltiples interfaces externas.
- Software de tiempo real. Es el *software* que coordina / analiza y controla sucesos del mundo real en un rango de milisegundos a segundos (tiempo real). Esta compuesto por un componente de adquisición de datos, uno de análisis, uno de control / salida y uno de monitorización que coordina los demás componentes.
- Software de ingeniería y científico. Este tipo de *software* está caracterizado por los algoritmos de "manejo de números". Las aplicaciones recorren una amplia gama de especialidades. Sin embargo, las nuevas aplicaciones del área de ingeniería / ciencia se han alejado de los algoritmos convencionales numéricos. El diseño

asistido por computadora, la simulación de sistemas y otras aplicaciones interactivas, han comenzado a coger características del *software* de tiempo real e incluso del *software* de sistemas.

- Software de inteligencia artificial. Es el *software* que hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Los sistemas expertos (también llamados sistemas basados en el conocimiento), reconocimiento de patrones (imágenes y voz), redes neuronales artificiales, prueba de teoremas y juegos son aplicaciones representativas de esta categoría.
- Software de gestión. El proceso de la información comercial constituye la mayor de las áreas de aplicación del *software*. Los sistemas de información de gestión (SIG) acceden a una o más bases de datos que contienen información comercial, reestructurando los datos existentes y/o realizando cálculos interactivos (por ejemplo, el procesamiento de transacciones en puntos de ventas), para facilitar las operaciones comerciales o gestionar la toma de decisiones.
- Software empotrado. El *software* empotrado reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo (por ejemplo, el control de las teclas de un horno de microondas).
- Software de computadores personales. El procesamiento de textos, las hojas de cálculo, multimedia, aplicaciones financieras, de negocios y personales son algunos ejemplos de esta categoría de *software*.
- Software basado en Web. Una página Web buscada por un explorador es un *software* que incorpora instrucciones ejecutables y datos. En esencia, la red viene a ser una gran computadora que proporciona un recurso *software* casi ilimitado.

1.3. INGENIERÍA DE SOFTWARE

1.3.1. DEFINICIÓN

Una de las primeras definiciones del término “Ingeniería de Software” aceptada fue propuesta por Fritz Bauer en [BAUE72]: “[La Ingeniería de Software] es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente *software* que sea fiable y que funcione eficientemente sobre máquinas reales”.

Una definición más completa actualmente aceptada es presentada en [IEEE90]: “[La Ingeniería de Software] es (1) la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del *software*, es decir, la aplicación de la ingeniería al *software*; y (2) el estudio de los métodos de (1)”.

1.3.2. CAPAS DE LA INGENIERÍA DE SOFTWARE

Cualquier enfoque de ingeniería (incluida la Ingeniería de Software) debe apoyarse sobre un compromiso de organización de calidad. Con esta finalidad la ingeniería del software estudia y aplica diversas tecnologías agrupándolas en 3 capas principales (proceso, métodos y herramientas), tal y como es mostrado en la Figura 1.3:

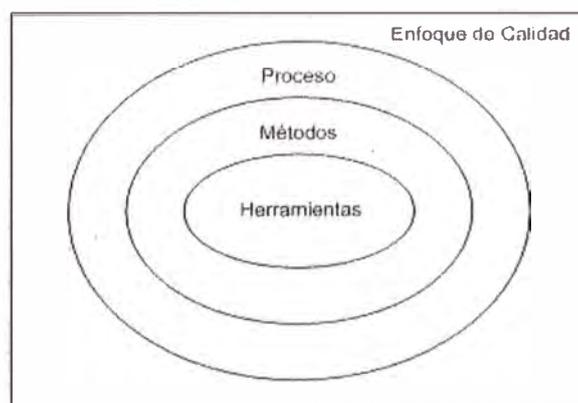


Figura 1.3: Capas de la ingeniería de software.

Capa de procesos

El fundamento de la Ingeniería de Software es la capa de proceso. El proceso es un conjunto de pasos predecibles que deben ser seguidos para el desarrollo racional y oportuno del *software*. El proceso es importante porque proporciona estabilidad, control y organización a una actividad que puede, si no se controla, volverse caótica. El proceso define un marco de trabajo para un conjunto de áreas claves de proceso. Las áreas claves del proceso forman la base del control de gestión de proyectos del *software* y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos de trabajo y se asegura la calidad. La calidad de un *software* puede ser cuantificada evaluando la conformidad del proceso para con un modelo de 5 niveles (inicial, repetible, definido, gestionado y optimizado) llamado de *Capability Maturity Model (CMM)*.

Capa de métodos.

Los métodos de la Ingeniería de Software indican “cómo” construir técnicamente el *software*. Los métodos abarcan una gran gama de tareas que incluyen y soportan cada una de las diversas etapas del proceso. Los métodos de la Ingeniería de Software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Capa de herramientas.

Las herramientas de la Ingeniería de Software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del *software* llamado *Computer-Aided Software Engineering (CASE)*.

1.3.3. EL PROCESO DE SOFTWARE

Para construir la Ingeniería de Software adecuadamente, se debe definir un proceso [de desarrollo] de *software*. Independientemente del área de aplicación, tamaño o complejidad del *software*, el trabajo que se asocia a la Ingeniería de software se puede dividir en tres fases genéricas:

- Fase de definición. Esta fase se centra sobre el qué, es decir, en identificar qué información ha de ser procesada, qué función, comportamiento y rendimiento se desea, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del *software*. Aunque los métodos aplicados durante esta fase variarán, de alguna manera tendrán lugar tres tareas principales: análisis de sistemas, planificación del proyecto del *software* y análisis de los requisitos.
- Fase de desarrollo técnico. Esta fase se centra en el cómo, es decir, en definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de *software*, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de realizarse la prueba. Aunque los métodos aplicados durante esta fase variarán, tres tareas específicas técnicas deberían ocurrir siempre: diseño del *software*, generación de código (implementación) y prueba del *software*.
- Fase de mantenimiento. Esta fase se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del entorno del *software* y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Durante esta fase se encuentran cuatro tipos de cambios: corrección, adaptación, mejora y /o prevención.

1.3.4. MODELOS DE PROCESO DEL SOFTWARE

Para resolver los problemas reales de una industria, un ingeniero de *software* o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe las fases genéricas del proceso, utilizando métodos y herramientas. Esta estrategia a menudo se llama modelo de proceso o paradigma de Ingeniería de Software. Se selecciona un modelo (o una combinación de modelos) según la naturaleza de la aplicación, los métodos y herramientas a utilizar, y los controles y entregas requeridos.

Entre los principales modelos de proceso tenemos el modelo lineal secuencial, el de construcción de prototipos, el espiral y el basado en componentes:

Modelo lineal secuencial.

Este modelo es llamado algunas veces de “ciclo de vida básico” o “modelo en cascada”. Tal y como muestra la Figura 1.4, este modelo sugiere un enfoque sistemático y secuencial para el desarrollo del *software* que comienza en un nivel de sistemas y progresa con el análisis, diseño, implementación, pruebas y mantenimiento. El principal problema de este modelo radica en el hecho que requiere explícitamente todos los requisitos al inicio del proyecto. Este modelo es el más antiguo y más extensamente utilizado, y pese a tener debilidades es significativamente mejor que un enfoque hecho al azar para el desarrollo de *software*.



Figura 1.4: Modelo lineal secuencial.

Modelo de construcción de prototipos.

Este modelo es adecuado cuando se tiene un conjunto de objetivos generales para el *software*, pero no es posible identificar los requisitos detallados de entrada, proceso o salida. Tal y como muestra la Figura 1.5 este modelo comienza con la recolección de requisitos, apareciendo un “diseño rápido” que lleva a la construcción de un prototipo, que será evaluado para posteriormente ser utilizado sólo para refinar los requisitos del *software* a desarrollar.

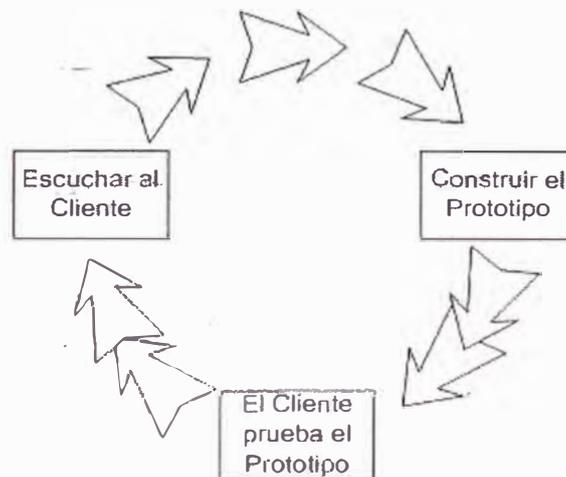


Figura 1.5: Modelo de construcción de prototipos.

Modelo espiral.

Este modelo conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. En el modelo espiral, el *software* se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado. A diferencia del modelo de proceso clásico que termina cuando se entrega el *software*, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del *software* [de computadora]. A pesar de las facilidades presentadas, este modelo no ha sido utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos.

Modelo basado en componentes.

Se entiende por componente a un conjunto de elementos lógicos con una funcionalidad común, de manera que sean tan independientes de otros componentes como sea posible. El componente de *software* debería diseñarse e implementarse para que pueda volver a ser reutilizado en muchos programas diferentes.

El modelo basado en componentes incorpora muchas de las características del modelo en espiral. Sin embargo, configura aplicaciones desde componentes preparados de *software*. Los componentes creados en proyectos anteriores se almacenan en bibliotecas. Tal y como muestra la Figura 1.6 inicialmente se identifican las componentes candidatos; luego las bibliotecas existentes son examinadas para determinar si estos componentes ya existen, caso así sea, se extraen de la biblioteca y se vuelven a reutilizar. Se compone así la primera iteración, mediante los componentes extraídos y los componentes nuevos construidos para cumplir las necesidades únicas de la aplicación. Este modelo conduce a la reutilización del *software*, proporcionando grandes beneficios a los ingenieros de *software*.

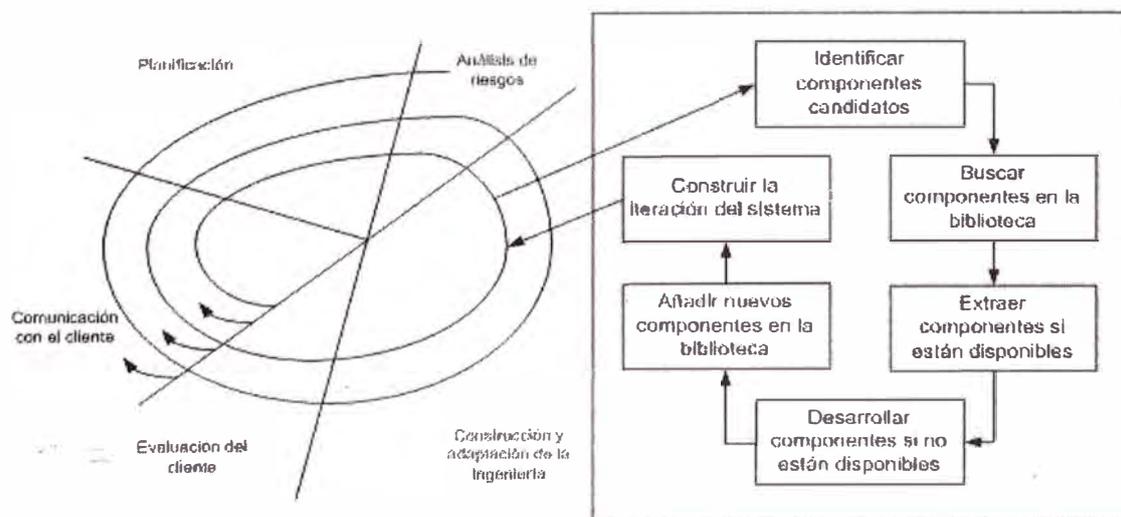


Figura 1.6: Modelo basado en componentes.

1.3.5. MÉTODOS DE DESARROLLO DE SOFTWARE

A través de los años se han propuesto muchos métodos dentro de la Ingeniería de Software. Sin embargo, actualmente dos tendencias dominan el panorama: los métodos formales y los métodos que sacrifican parte de una formalidad en compensación a rapidez de desarrollo, los cuales son llamados de ágiles. Entre los métodos formales encontramos los métodos estructurados y los métodos orientados a objetos.

Los métodos estructurados están basados en la comprensión del *software* como un conjunto de funciones (procedimientos) que ejecutan procesamientos sobre datos. Los datos y los procedimientos son tratados separadamente, de forma que los datos son vistos como elementos pasivos que van sufriendo una serie de procesamientos hasta conseguir el resultado deseado. Los métodos estructurados pueden aún ser divididos en [métodos para] análisis estructurado y diseño estructurado.

El análisis estructurado es un método de modelado clásico que, mediante una notación semi-estándar, crea modelos que representen el contenido y el flujo de los datos, partiendo el sistema funcionalmente, y de acuerdo a los distintos comportamientos requeridos, establece la esencia de lo que se debe construir. Este análisis no es un método sencillo aplicado siempre de la misma manera, sino, es una amalgama que ha evolucionado durante los últimos 30 años [DEMA79] [YOUR89].

El diseño estructurado es un método de modelado clásico que acrecentó aspectos procedimentales a los trabajos que se concentraban en criterios para el desarrollo de programas modulares y métodos para refinar las estructuras del *software* de manera descendente. El diseño estructurado establece modelos [de diseño] que especifican cómo los modelos de análisis han de ser construidos [MILL72] [GANN89], siendo los procedimientos finalmente implementados en funciones y agrupados generalmente en módulos utilizando lenguajes de programación procedurales (C, Fortran).

1.4. PARADIGMA DE ORIENTACIÓN A OBJETOS (POO)

1.4.1. DEFINICIÓN

El POO es definido como un paradigma de desarrollo de *software* basado en la tecnología de objetos. Se entiende por paradigma a un conjunto de conocimientos y creencias que forman una visión del mundo, en torno a una teoría en determinado periodo histórico [WIKI06]. La tecnología de objetos es una tecnología que representa entidades existentes en el mundo real [tan fielmente como sea posible] como objetos *software* que se interrelacionan para producir una solución al problema que se está tratando de resolver. Es interesante notar que los objetos existen en la naturaleza mucho antes de existir cualquier tipo de aplicación que los represente dentro de sistemas basados en computador (*software*).

Durante muchos años el término [Programación] “Orientada a Objetos” (OO) se utilizó para referirse a un enfoque de desarrollo de *software* que utilizase un lenguaje de programación orientado a objetos (C++, Eiffel, Smalltalk, etc.). Actualmente el término POO encierra una completa visión de la Ingeniería de Software, constituyendo lo que tradicionalmente se llama de Ingeniería de Software Orientada a Objetos. Este paradigma hace sentir su impacto en las etapas principales (por no decir todas) del proceso, originando los términos de “Análisis Orientado a Objeto” (AOO), “Diseño Orientado a Objetos” (DOO) e Implementación o “Programación Orientada a Objetos”. Este paradigma también influye en la formulación de métodos, los cuales se presentan generalmente agrupados en las llamadas “Metodologías orientadas a Objetos” (MOO), y en conjunto con las herramientas orientadas a objetos dan soporte a las principales etapas del proceso. Tanto el proceso como los métodos mencionados serán comentados posteriormente, luego de presentar los conceptos básicos sobre los cuales está basado el POO.

1.4.2. CONCEPTOS BÁSICOS

Las definiciones de los conceptos básicos del POO han sido fuente de diversos debates a través de la historia. Sin embargo, sumado a los conceptos de “objeto” y “clase”, existen tres características de los sistemas orientados a objetos que los hacen únicos: encapsulación, herencia y polimorfismo [TAYL97]. A continuación se describen brevemente todos estos conceptos básicos:

Objeto

Un objeto *software* (o simplemente objeto) es una abstracción, individualizada e identificable, que modela el concepto presente en la realidad humana ocupando espacio físico (entidad real) o lógico (entidad abstracta) [TAYL97].

La abstracción es una de las principales formas que los humanos usan para lidiar con la complejidad. Es un proceso mental para ver una entidad con diferentes grados de detalle. Tal y como se muestra en la Figura 1.7 la abstracción permite denotar características esenciales de una entidad que lo diferencien de otras, estableciendo un alcance o frontera relativa elegida por el observador, [BOOC97].

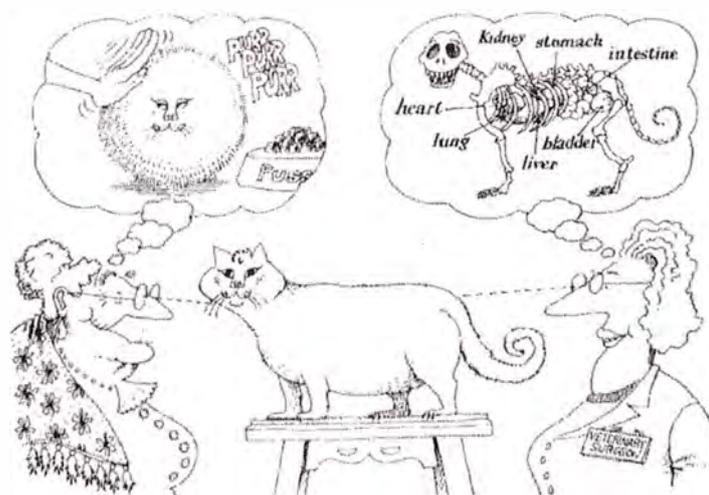


Figura 1.7: Ejemplo de abstracción.

Los objetos modelan entidades del mundo real, siendo cada uno de ellos caracterizado por un estado y un comportamiento. El estado es una condición durante la vida de un objeto, es decir, cómo sus propiedades (formalmente llamados de atributos) están durante un determinado momento de su ciclo de vida. El estado de un objeto representa los resultados acumulativos de su comportamiento. El comportamiento es el conjunto de operaciones (formalmente llamados de métodos) que describen como el objeto actúa o responde ante la recepción de un estímulo (formalmente llamado de mensaje). La identidad del objeto es el atributo que lo diferencia de todos los otros objetos.

Clase

La realidad puede ser representada por un número de objetos comunicándose unos con otros. Algunos de estos objetos poseen características comunes que permiten agruparlos. Una clase es la representación de un conjunto de objetos que comparten la misma estructura de atributos, métodos, relaciones y semántica [TAYL97]. Tal y como lo muestra la Figura 1.8, una clase es una plantilla de la cual se pueden derivar (formalmente llamado de instanciar) objetos.

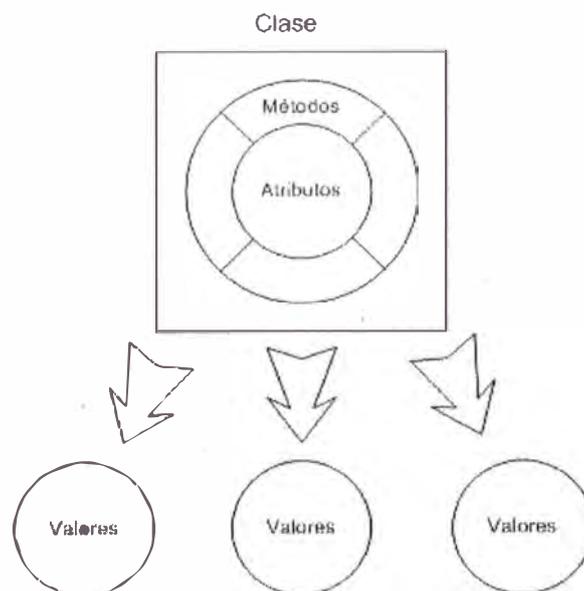


Figura 1.8: Una clase y sus instancias.

Encapsulamiento

Es la propiedad de juntar los datos (atributos) y operaciones (métodos) en un mismo paquete (clase), de modo que estos datos sólo pueden ser accedidos vía la interfase definida por la clase [BOOC97].

La interfase de una clase consiste en la declaración de todos los métodos que la clase contendrá. Un objeto para enviar un mensaje a otro, no tiene que preocuparse con los detalles internos en el objeto receptor, sólo con la interfase del objeto receptor. Tal y como muestra la Figura 1.9 el encapsulamiento oculta los detalles de la implementación interna de un objeto, reduciendo los efectos colaterales cuando ocurren cambios.

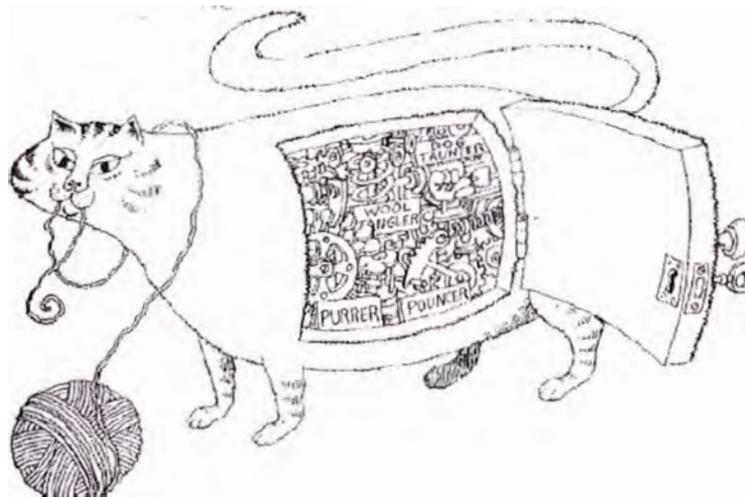


Figura 1.9: Ejemplo de encapsulamiento.

Herencia

Es el mecanismo de reutilización de atributos y métodos definidas en clases generales por clases más específicas [BOOC97].

La herencia trabaja con el concepto de jerarquía de clases, es decir, una subclase Y hereda todos los atributos y métodos asociados con su superclase X. Esto quiere decir que todas las estructuras de datos y algoritmos originalmente diseñados e implementados para X están inmediatamente disponibles para Y. La reutilización se realiza directamente.

Es importante destacar que en cada nivel de la jerarquía de clases, pueden añadirse nuevos atributos y métodos a aquellos que ya han sido heredados de niveles superiores de la jerarquía. Un ejemplo de herencia puede ser observado en la Figura 1.10.

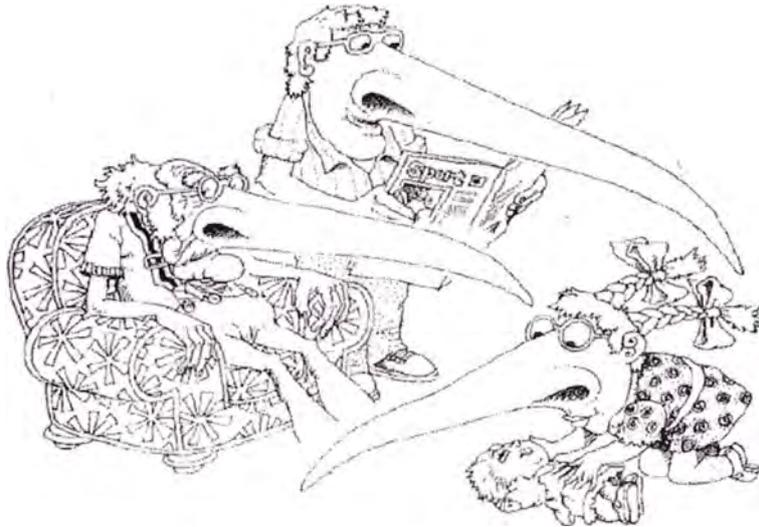


Figura 1.10: Ejemplo de herencia

Polimorfismo

Es la propiedad de un programa orientado a objetos de discernir, de entre los métodos homónimos, aquel que debe ser ejecutado [PRES02].

El polimorfismo trabaja con el concepto de jerarquía de clases, donde las subclases implementan operaciones diferentes que las superclases, de modo que una operación puede tener diferentes implementaciones en diversos puntos de la jerarquía de clases, desde que se mantenga la misma descripción.

Para entender el polimorfismo, considere una aplicación convencional que debe dibujar o un círculo o un cuadrado. Considere una superclase llamada Gráfico, de la cual se derivan dos subclases: Círculo y Cuadrado. La superclase Gráfico define un método llamado dibujar que será implementado diferentemente en cada una de las subclases, debido a que los dibujos resultantes deben ser diferentes.

1.4.3. PROCESO ORIENTADO A OBJETOS

Si bien de entre los modelos anteriormente comentados en el ítem 1.3.4 no existe uno que se adapte completamente para su uso con el POO, se reconoce que la utilización de un modelo evolutivo que fomente el ensamblaje (reutilización) de componentes es una buena elección. Una de las mejores propuestas de un modelo de proceso orientado a objetos es el *Rational Unified Process* (RUP) [BOOC99]. El POO enfatiza la creación de clases que encapsulan tanto los datos como los algoritmos que se utilizan para manejar los datos. Si se diseñan y se implementan adecuadamente, estas clases son reutilizables por las diferentes aplicaciones y arquitecturas de *software*.

En la Figura 1.6 se presentó un diagrama del modelo de desarrollo basado en componentes (ítem 1.3.4), un modelo genérico que puede ser adaptado para un modelo de proceso orientado a objetos (Figura 1.11), con la particularidad que cuando el componente (clase) no se encuentra en la biblioteca, el desarrollador de *software* utiliza un conjunto de métodos (metodología) orientado a objetos para crearlo.

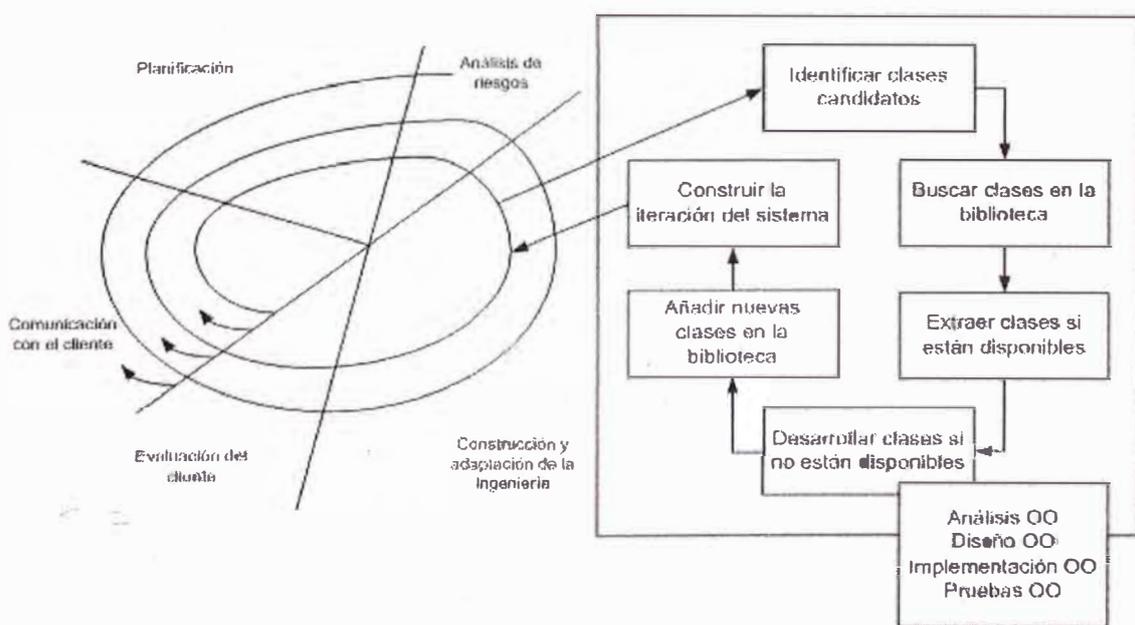


Figura 1.11: Modelo de proceso orientado a objetos.

1.4.4. MÉTODOS ORIENTADOS A OBJETOS

Los métodos OO son definidos para alcanzar los objetivos de cada una de las etapas del proceso. En la Figura 1.11 se puede apreciar que el desarrollo técnico está compuesto básicamente del AOO, DOO y la Programación OO, los cuales son comentados a continuación:

- **Análisis Orientado a Objetos.** El principal objetivo de esta etapa es refinar y estructurar los requerimientos del usuario. El resultado del análisis será un modelo conceptual con especificaciones en el lenguaje de los desarrolladores, formando un paquete que se denomina Modelo de Análisis. Los paquetes permiten organizar el modelo en piezas manejables que representan abstracciones de subsistemas y conjunto de clases. El Modelo de Análisis incluye las clases de análisis y la forma en que estas interactúan.
- **Diseño Orientado a Objetos.** El principal objetivo de esta etapa es confeccionar un modelo que especifique la manera en que se construirá el sistema para cumplir con los requerimientos establecidos. Se deben identificar la arquitectura, especificar subsistemas; así como los objetos pertinentes, clasificarlos dentro de las clases en la granularidad correcta, definir interfases de clases y jerarquías de herencia y establecer relaciones clave entre ellos. El diseño debe ser específico al problema que se tiene entre manos, pero suficientemente general para adaptarse a problemas y requerimientos futuros.
- **Programación Orientada a Objetos.** La etapa final del desarrollo técnico OO es la programación. El proceso de programación involucra la conversión de un diseño OO en un código de programa. Efectivamente, esto significa que las clases definidas en el diseño deben ser convertidas en clases expresadas en lenguajes de programación orientados a objetos (C++, Java, etc.).

Los métodos OO generalmente se presentan agrupados en las denominadas “Metodologías Orientadas a Objetos” (MOO). Una lista de las principales MOO propuestas en la literatura es presentada en la Tabla 1.1, y una comparación entre ellas puede ser encontrada en [BOUZ97].

Tabla 1.1: Metodologías orientadas a objetos.

Nombre	Autores	Fecha
BOOCH	G. Booch	1991
HOOD	HOOD Technical Group	1993
OOA	S. Shlaer, S. Mellor	1988/1992
OOA/OOD	T. Coad, E. Yourdon	1991
OMT	J. Rumbaugh, M. Blaha, W. Premerlani, etc.	1991
OOSE	I. Jacobson, M. Christerson, P. Jonson, G. Overgaard	1992
OOM	M. Bouzeghoub, A. Rochfeld	1993/1994
FUSION	D. Coleman, P. Arnold, S. Bodoff, etc.	1994

1.4.5. VENTAJAS DEL POO

La aplicación del POO en el desarrollo de *software* trae muchas ventajas, las cuales pueden ser resumidas en:

- *Fácil reaprovechamiento del código* – clases bien definidas (abstraídas e implementadas) hoy, reducen tiempo de desarrollo mañana. Clases bien definidas se muestran útiles para la resolución de problemas diferentes al problema original.
- *Facilidad en el mantenimiento del software* – clases bien definidas originan que los mantenimientos listados en 1.3.3 pasen a ser predominantemente evolutivos, dejando de ser correctivos.
- *Fácil representación del mundo real* – “un aspecto interesante de la teoría de objetos es su característica intrínseca de analizar el mundo tal y como él es, permitiendo organizar resultados de manera más fácil y natural”, es decir, el POO busca representar el mundo real tal y como él realmente se presenta (ya que finalmente el mundo está formado de entidades objeto).

1.5. UNIFIED MODELING LANGUAGE (UML)

1.5.1. DEFINICIÓN

Es un lenguaje de modelado visual utilizado para especificar, visualizar, construir y documentar los artefactos de *software* [BOOC98], es decir, el UML funciona como un estándar para documentación de *software* desarrollado con la OO.

Este tipo de abordaje es importante, pues evita que el desarrollador del *software* sea el único a conocer cómo el *software* funciona evitando que él sea la única persona que pueda manipularlo. El UML es una forma de representación visual que tiene el propósito de modelar, configurar, mantener y controlar informaciones sobre un *software*, posibilitando que cualquier persona pueda comprenderlo y manipularlo.

De entre las metodologías listadas en la Tabla 1.1, las más dominantes fueron (y tal vez aún son) la BOOCH, OMT y OOSE. Los respectivos autores (Booch, Rumbaugh y Jacobson) decidieron unir fuerzas y en junio de 1996 lanzaron una unión de sus métodos llamado de “Método Unificado”, la cual traía consigo adicionalmente una propuesta de estandarización para el modelado. Esta propuesta fue adoptada posteriormente por la *Object Management Group* (OMG) y gracias a una participación activa de la comunidad de Ingeniería de Software, surge un lenguaje de modelado (UML) bien definido, expresivo, poderoso y genéricamente aplicable. En los últimos años nuevas versiones fueron presentadas a la comunidad, siendo la versión actual la [UML] 2.0. Es importante resaltar que el UML no es un proceso ni tampoco un método. El UML soporta varios procesos (lineal, prototipos, componentes, etc.), así como también muchos métodos OO (BOOCH, OMT, OOSE, etc.).

No es objetivo de este trabajo presentar detalles de todos los aspectos y notaciones del UML, mas sí presentar lo necesario para el entendimiento de los próximos capítulos.

1.5.2. VISIÓN GENERAL DEL UML

Para conocer la estructura del UML en la Figura 1.12 se muestra un vista general de todos sus componentes, de manera que resulte más fácil la comprensión de cada uno de ellos.

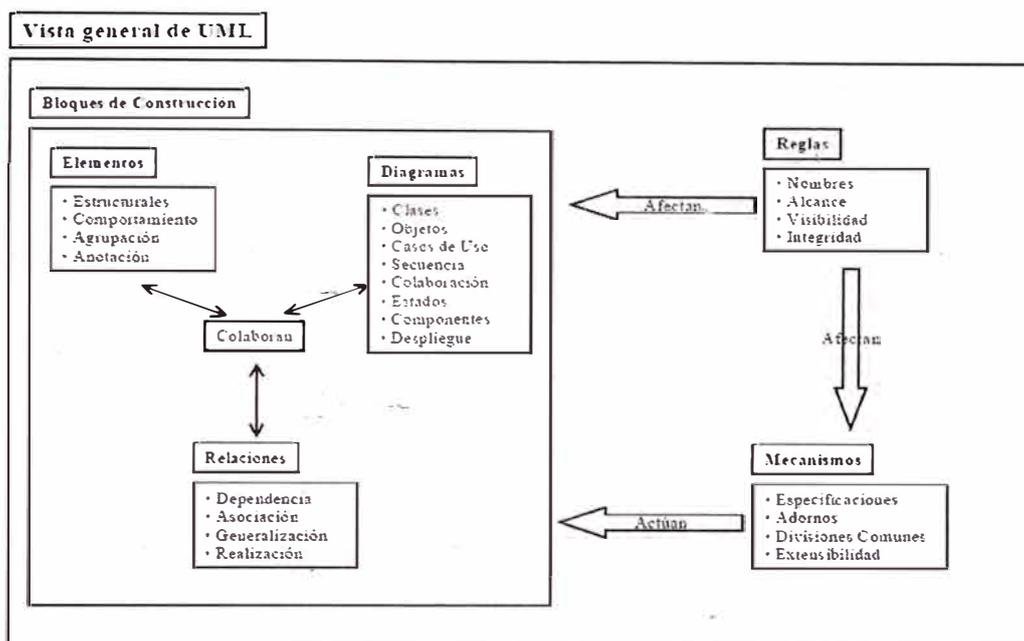


Figura 1.12: Visión general del UML.

El UML se compone de tres elementos básicos: los bloques de construcción, las reglas y algunos mecanismos comunes. Estos elementos interactúan entre sí para dar al UML el carácter de completitud y no ambigüedad ya comentados.

Los bloques de construcción se dividen en tres partes: elementos (abstracciones de primer nivel), relaciones (que unen los elementos entre sí) y los diagramas (que son agrupaciones interesantes de elementos), los cuales comentaremos posteriormente.

Las relaciones se dividen para abarcar las posibles interacciones entre los elementos que se nos pueden presentar a la hora de modelar usando UML. Las reglas dan las pautas a la hora de realizar asociaciones entre elementos para obtener modelos bien formados.

1.5.3. ELEMENTOS DEL UML

Entre los elementos de interés para el presente informe tenemos:

Clase

Elemento estructural, cuyo concepto ya fue abordado anteriormente. Su notación en el UML es mostrada en la Figura 1.13.



Figura 1.13: Notación UML para la clase

Interfaz

Elemento estructural que representa una colección de operaciones que especifican un servicio de una determinada clase (o componente). Una interfaz describe el comportamiento visible externo, mas no la implementación de ese comportamiento. Su notación es mostrada en la Figura 1.14.



Figura 1.14: Notación UML para la interfaz

Colaboración

Elemento tanto estructural como de comportamiento. Define una interacción y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos individuales. Su notación es mostrada en la Figura 1.15.



Figura 1.15: Notación UML para la colaboración

Actor

Un actor es una abstracción de una entidad fuera del *software* (seres humanos, máquinas o hasta mismo otros softwares) que interactúa directamente con el mismo. Su notación es mostrada en la Figura 1.16.



Figura 1.16: Notación UML para el actor.

Casos de Uso

Elemento estructural que describe un conjunto de de secuencias de acciones que un *software* desempeña para producir un resultado observable de valor para un actor específico. Su notación es mostrada en la Figura 1.17.



Figura 1.17: Notación UML para el caso de uso.

Componentes

Elemento estructural que representa típicamente el empaquetamiento físico de diferentes elementos lógicos (clases, interfaces y colaboraciones). Su notación es mostrada en la Figura 1.18.

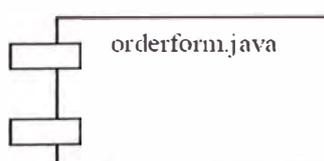


Figura 1.18: Notación UML para el componente.

Interacción

Elemento de comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos. Una interacción involucra mensajes, secuencias de acción y enlaces (conexiones entre objetos). Su notación es mostrada en la Figura 1.19.

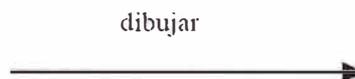


Figura 1.19: Notación UML para la interacción.

Paquete

Elemento de agrupación utilizado para organizar elementos. Los elementos estructurales, de comportamiento, incluso los de agrupación se pueden incluir en un paquete. Su notación es mostrada en la Figura 1.20.



Figura 1.20: Notación UML para el paquete.

Nota

Es un elemento de anotación. Es la parte explicativa de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento del modelo. Su notación es mostrada en la Figura 1.21.

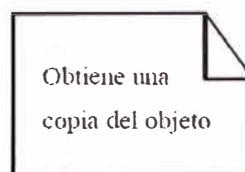


Figura 1.21: Notación UML para las anotaciones.

1.5.4. RELACIONES DEL UML

Existen cuatro tipos de relaciones entre los elementos de un modelo UML: dependencia, generalización, realización y asociación.

Dependencia

Es una relación semántica entre dos elementos en la cual un cambio a un elemento [independiente] puede afectar a la semántica del otro elemento [dependiente]. Su notación es mostrada en la Figura 1.22.



Figura 1.22: Notación UML para la dependencia.

Generalización

Es una relación de especialización / generalización en la cual los objetos del elemento especializado pueden sustituir a los objetos del elemento general. Así, el elemento específico posee todas las características del elemento general, además de las propias. Su notación es mostrada en la Figura 1.23.



Figura 1.23: Notación UML para la generalización.

Realización

Es una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización entre interfaces, clases y componentes que la realizan, y entre los casos de uso y colaboraciones que los realizan. Su notación es mostrada en la Figura 1.24.



Figura 1.24: Notación UML para la realización.

Asociación

Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones que representan alguna asociación entre objetos. La notación para la asociación es mostrada en la Figura 1.25.



Figura 1.25: Notación UML para la asociación.

Adicionalmente a la asociación mostrada se distinguen dos tipos más de asociaciones: agregación y composición. La agregación indica una relación “todo-parte”, es decir, indica que un objeto “todo” es representado por otros objetos que son su “parte”. La composición es una variación más poderosa de agregación. La diferencia es que en la composición la clase parte pertenece apenas a la clase todo, es decir, la clase todo no puede existir sin su parte, ya en la agregación eso sí puede ocurrir. Las notaciones particulares para la agregación y la composición son mostradas en la Figura 1.26

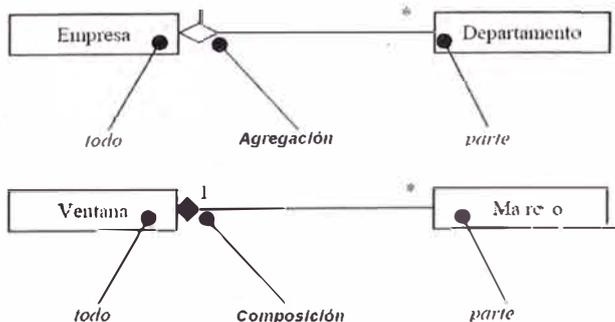


Figura 1.26: Notación UML para la agregación y composición.

1.5.5. DIAGRAMAS DEL UML

Los diagramas se utilizan para representar diferentes perspectivas de un mismo sistema. UML proporciona un amplio conjunto de diagramas que normalmente se utilizan en pequeños subconjuntos para representar las vistas principales (casos de uso, proceso, diseño, implementación y despliegue) de la arquitectura de un sistema.

Entre los diagramas relevantes para el presente informe podemos mencionar: el diagrama de casos de uso, diagrama de clases y diagrama de colaboración. Ejemplos de estos diagramas se mostrarán en el capítulo 3, ya aplicados a la resolución del objetivo principal del presente informe.

Diagrama de Casos de Uso

Es el diagrama que muestra como los actores y los casos de uso se relacionan en un *software*. El propósito de este diagrama es describir los requerimientos funcionales de un *software* de manera consensual entre usuarios y desarrolladores, es decir, cada caso de uso representa una funcionalidad del *software*.

Diagrama de Clases

El diagrama de clases es el principal diagrama dentro del modelado orientado a objetos. Dentro de este diagrama es posible modelar todas las clases y sus respectivas relaciones. También pueden ser modelados en este diagrama interfaces y componentes.

Diagrama de Colaboración.

Es un diagrama que consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Cubren la vista dinámica del sistema. Este diagrama muestra la organización estructural de los objetos que envían y reciben mensajes.

1.6. SISTEMAS DISTRIBUIDOS

1.6.1. DEFINICIÓN

Un sistema distribuido consiste en una colección de computadores autónomos conectados a través de una red y equipados con algún *software* de sistema distribuido [COUL00]. El *software* de sistema distribuido permite que los computadores coordinen sus actividades y compartan los recursos del sistema (*hardware*, *software* y datos). Los usuarios de un sistema distribuido bien proyectado deben percibirlo como siendo un único e integrado sistema de computación, mismo que sea implementado a través de diversos computadores localizados en diferentes regiones.

Antiguamente todo el procesamiento de una empresa era de responsabilidad de sólo una máquina, llamada *Mainframe*. El surgimiento y costo cada vez menor de los *Personal Computer* (PC) con procesadores de alto desempeño, junto con las redes de computadores, hicieron que esta idea quedase obsoleta. Se visualizó que no tenía sentido centralizar todo el procesamiento en apenas una *Mainframe* si era viable técnica y económicamente utilizar varios computadores interconectadas a través de una red, tal y como es mostrado en la Figura 1.27.

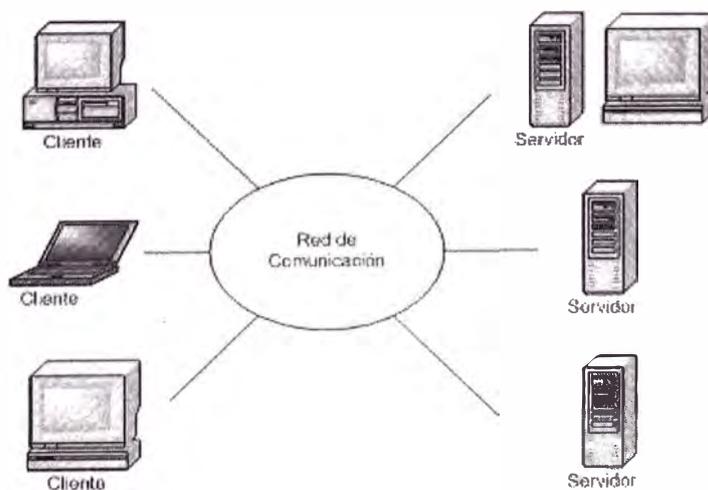


Figura 1.27: Red de computadores

1.6.2. ESTRUCTURA CLIENTE - SERVIDOR

El modelo arquitectural de un sistema define cuales son los diversos componentes del mismo, cómo éstos están organizados y cómo los usuarios tienen acceso a varios recursos del sistema. En el campo de los sistemas distribuidos, algunos modelos ya fueron experimentados, siendo el modelo cliente – servidor el que se afirmó como el patrón para la organización de este tipo de sistemas [TEIX02].

El funcionamiento del modelo cliente – servidor puede ser visualizado en las Figuras 1.27 y 1.28, ocurriendo básicamente de la siguiente manera: existen servidores esparcidos en la red, donde cada uno de ellos es responsable por gerenciar un determinado recurso, y existen clientes que necesitan utilizar tales recursos. Luego el cliente hace una solicitud para el servidor a través de un pedido. El servidor recibe el pedido, lo ejecuta y retorna la repuesta para el cliente que la solicitó. Durante todo el periodo desde la solicitud hasta el recibimiento de la respuesta el cliente permanece bloqueado caracterizando así una llamada síncrona.

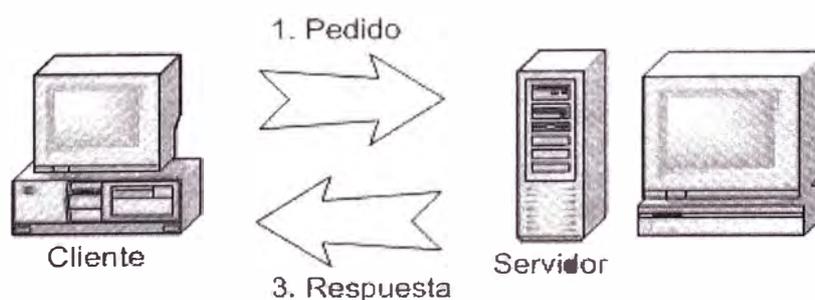


Figura 1.28: Estructura Cliente - Servidor

La utilización de los sistemas distribuidos (específicamente del modelo cliente – servidor) trajo consigo una serie de ventajas más también algunas dificultades (desventajas) adicionales en el proceso de desarrollo de *software*, las cuales son comentadas en el siguiente ítem.

1.6.3. VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS

Entre las ventajas obtenidas de la utilización de sistemas distribuidos podemos mencionar:

- *Compartimiento de recursos* – Un computador puede usar recursos (memoria, impresoras, etc.) que físicamente se encuentran en otro computador perteneciente al sistema distribuido.
- *Velocidad de computación* – Los sistemas distribuidos permiten que un proceso sea ejecutado concurrentemente en varios computadores disminuyendo el tiempo de procesamiento y aumentando la velocidad de computación en comparación a la ejecución del mismo proceso en apenas un computador...
- *Confiabilidad* – Si un computador falla en un sistema distribuido, los computadores restantes pueden continuar operando, dotando al sistema de mayor confiabilidad.
- *Comunicación* – Una gran ventaja de los sistemas distribuidos es la facilidad en el intercambio de informaciones. Esto es posible debido a la interconexión de PCs a través de las redes de comunicación.

Ya con relación a las desventajas, se pueden mencionar:

- *Software* – Los sistemas distribuidos necesitan de *software* diferente de los utilizados por los sistemas centralizados, los cuales son mucho más complejos de ser implementados.
- *Dependencia de la red* – Los sistemas distribuidos son altamente influenciados por el funcionamiento de red, la cual puede saturar o causar otros problemas.
- *Seguridad* – Los sistemas distribuidos necesitan de mecanismos de seguridad mucho más complejos que los sistemas centralizados.

1.7. COMMON OBJECT REQUEST BROKER ARCHITECTURE

1.7.1. DEFINICIÓN

CORBA es una especificación de una arquitectura desarrollada para ambientes distribuidos y heterogéneos que utiliza la tecnología orientada a objetos [COUL00], es decir, esta especificación tiene como principio hacer que aplicaciones distribuidas en una red se comuniquen utilizando los conceptos de la tecnología orientada a objetos, independientemente de la plataforma y/o lenguaje de programación en los que fueron desarrollados.

Esta especificación fue creada por la OMG y se inserta dentro del *Object Management Architecture (OMA)*, también definida por la OMG.

Object Management Group (OMG)

La OMG (organización que ya fue citada en el ítem correspondiente a UML) es una organización internacional de la industria del *software* fundada en mayo de 1989 y soportada por centenas de miembros con el objetivo principal de alcanzar sistemas basados en objetos en ambientes distribuidos y heterogéneos con características de reuso, portabilidad e interoperabilidad [OMG04], características que serán comentadas más adelante.

Object Management Architecture (OMA)

El OMA es el corazón de la especificación CORBA, pues la misma provee una completa arquitectura que es rica y suficientemente flexible para acomodar softwares distribuidos de todas las variedades [OMG04]. La OMA está compuesta de un modelo de objetos y un modelo de referencia. El modelo de objetos es responsable por definir cómo los objetos distribuidos pueden ser descritos a través de ambientes heterogéneos. El modelo de referencia caracteriza la interacción entre esos objetos. A través de estos dos modelos CORBA es capaz de desarrollar sistemas basados en objetos distribuidos en ambientes heterogéneos con características de reuso, portabilidad e interoperabilidad [LANG02].

1.7.2. ESTRUCTURA DE CORBA

CORBA trabaja dentro del modelo cliente – servidor, es decir, un cliente hace un pedido de un servicio, el servidor responsable lo recibe y ejecuta y retorna una respuesta para el cliente. CORBA es considerado un *middleware* porque posibilita la comunicación entre dos aplicaciones separadas y diferentes, tal que él mismo establece la comunicación de un modo transparente, ocultando los detalles complejos de bajo nivel y de implementación, de los usuarios.

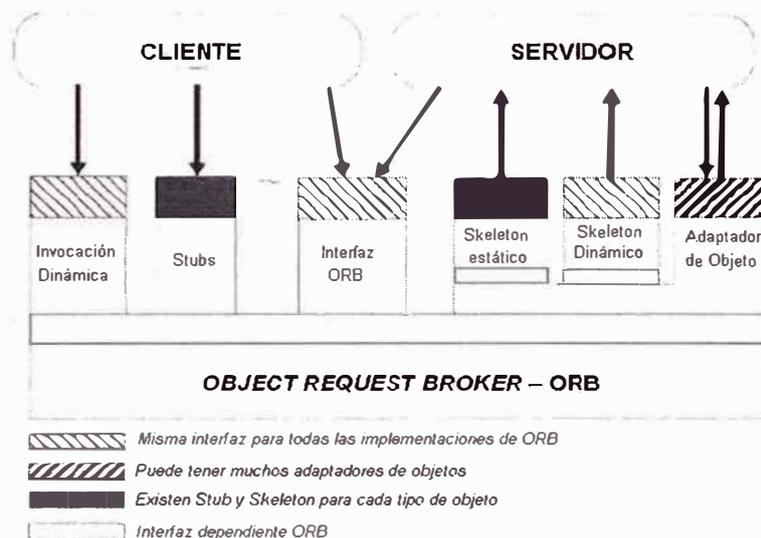


Figura 1.29: Estructura de CORBA

Los softwares (clientes y servidores) que utilizan CORBA precisan acceder o disponibilizar servicios a través de interfaces. CORBA utiliza un lenguaje de definición de interfaces (IDL) para generar archivos con códigos fuentes responsables por la comunicación entre el cliente y el servidor. Estos códigos fuentes son generados (mapeados) para el lenguaje de programación en que será implementada la aplicación distribuida, a través del compilador de interfaces disponibilizado por el fabricante que implementó la especificación CORBA. El archivo generado correspondiente al cliente es denominado “*stub*” y el archivo generado correspondiente al servidor es denominado “*skeleton*”. La utilización de estos archivos puede ser visualizado en la Figura 1.29.

Los archivos stub y skeleton necesitan de un Object Request Broker (ORB) para permitir la comunicación entre el cliente y servidor, tal y como puede ser observado en la Figura 1.29. El ORB es la parte central de la especificación CORBA. Se constituye en la barra por donde circulan las solicitudes a los objetos servidores y las respuestas de los mismos a los clientes. El ORB conecta una aplicación cliente con el objeto que desee utilizar. El cliente no necesita saber si el objeto se encuentra en la misma PC o localizado remotamente en algún lugar de la red. El cliente sólo necesita conocer el nombre del objeto y cómo está estructurada su interfaz.

1.7.3. VENTAJAS DE CORBA

Entre las principales ventajas de la aplicación de CORBA se pueden mencionar:

- **Transparencia** – todas las invocaciones de métodos de objetos serán manoseadas de forma transparente por CORBA. Para el usuario será indiferente si el objeto está localizado remotamente o localmente, como si todas las llamadas ocurriesen localmente.
- **Independencia de plataforma** – CORBA tiene su propio lenguaje para describir las interfaces de los objetos remotos llamada de IDL, donde la misma puede ser compilada independiente de la plataforma o lenguaje de programación. Es a través de este lenguaje que los usuarios definen los métodos que serán disponibles por los objetos remotos.
- **Portabilidad** las aplicaciones CORBA pueden ser fácilmente transferidas de un ORB para otro, desde que los ORB soporten el mismo lenguaje en el cual el objeto CORBA fue originalmente desarrollado. Esto es posible debido a la estandarización de la arquitectura.

- Reuso de *software* – CORBA provee todos los medios para acceder todos los objetos independientemente si están en plataformas o lenguajes diferentes. Esto quiere decir que tales objetos no precisan de ser transferidos (re-escritos o re-compilados), basta apenas ser reaprovechados.
- Integración – Al ser CORBA independiente de la plataforma y del lenguaje de programación, posibilita la integración de componentes de *software* a partir de varios softwares.
- Interoperabilidad – la idea por detrás de la interoperabilidad de CORBA es que los objetos remotos en ejecución se pueden comunicar, mismo que los fabricantes sean diferentes. Esto es posible pues CORBA estandariza su propio protocolo de comunicaciones y su propio lenguaje de definición de interfaces.
- Transparencia de localización – a través del ORB el CORBA es capaz de localizar un objeto independientemente si el mismo está localizado remota o localmente.
- Escalabilidad – CORBA soporta ambientes con un potencial para un gran número de objetos y usuarios.

CAPÍTULO II

SISTEMAS ELÉCTRICOS DE ENERGÍA

2.1. INTRODUCCIÓN

Desde que en 1882 Thomas Alva Edison (1847-1931) instaló el primer Sistema Eléctrico de Energía para vender la electricidad necesaria para la iluminación incandescente de la estación Pearl Street en la ciudad de New York; la energía eléctrica (gracias a la flexibilidad en su generación y transmisión, así como los altos rendimientos de los motores eléctricos) se ha convertido en la forma de energía más utilizada en el mundo.

La producción y entrega de la electricidad a los consumidores finales sólo es posible gracias a una adecuada operación de los SEE; operación que es garantizada mediante diversas simulaciones efectuadas por los ingenieros [de sistemas de potencia] con el fin de prever el comportamiento del SEE en condiciones normales y anormales de operación. Las simulaciones utilizan modelos matemáticos que representan al SEE. Prácticamente todos los modelos son generados aplicando la teoría de circuitos eléctricos y de sistemas de control.

Los propósitos de este capítulo son revisar algunas de las ideas fundamentales de los circuitos eléctricos, establecer la notación que se empleará a lo largo del presente informe, comentar el funcionamiento típico de un SEE e introducir los conceptos relativos a tres de las simulaciones más utilizadas dentro del análisis de los SEE, el “flujo de potencia”, la “estabilidad permanente” y la “estabilidad transitoria”.

2.2. TÉRMINOS BÁSICOS DE ELECTRICIDAD

Antes de discutir las características técnicas en la producción y uso de energía eléctrica, es necesario explicar términos y definiciones presentes entre los ingenieros y personal técnico que trabajan en el área eléctrica.

El término *electricidad* es utilizado para describir un fenómeno físico producto de microscópicas “cargas eléctricas” en reposo o en movimiento. La electricidad puede generarse en dos formas: *Direct Current* (DC) o *Alternating Current* (AC). La diferencia radica en que la electricidad DC es generada a valores que no dependen del tiempo mientras que la AC sí.

El término *frecuencia* (f medido en *Hertz* (Hz)) es utilizado para cuantificar el número de veces que una forma de onda [periódica] se repite en un intervalo de un segundo (s). En el Perú la frecuencia es de 60 Hz.

El término *diferencia de potencial* (ΔV medido en *voltios* (V)), *tensión* o *voltaje* en un sistema eléctrico es el equivalente a la presión en un sistema de flujo de agua o gas. Es formalmente definido como el resultado de la separación de cargas eléctricas (entre dos puntos) asociada con invisibles campos eléctricos. Las *fuentes* [de tensión] son elementos ideales que generan un $\Delta V_{ab} = V_a - V_b$ entre sus terminales a-b.

El término [intensidad de] *corriente* (I medido en *amperios* (A)) es el equivalente al flujo de agua en un sistema de flujo de agua. Es formalmente definido como el resultado del movimiento de las cargas eléctricas (a través del elemento) asociado con invisibles campos magnéticos.

Los *conductores* proveen la ruta para la corriente y son equivalentes a las tuberías en un sistema de flujo de agua. Los conductores pueden ser clasificados principalmente en dos tipos: los elementos que requieren de electricidad para funcionar, llamados de *cargas* (motores, televisores, etc.), y los elementos que sirven principalmente para transportar la electricidad de un punto para otro, llamados de *cables*.

Los términos *resistencia* (R), *inductancia* (L) y *capacitancia* (C), son utilizados para cuantificar propiedades físicas de los conductores que determinan la relación existente entre la tensión y la corriente en los mismos. Un *resistor*, *inductor* y *condensador* son conductores ideales que sólo poseen propiedades de R, de L y de C respectivamente.

El término *impedancia* (Z medida en *Ohms* (Ω)) es utilizado para cuantificar la propiedad de los conductores de impedir el flujo de corriente. Es calculado como la tensión sobre la corriente, siendo una función dependiente de la resistencia, inductancia y capacitancia. Los conductores son generalmente bien representados por el valor de su impedancia. Un *conector* es un cable que no impide el flujo de corriente ($Z=0$). Un *aislador* [ideal] es un elemento que impide completamente el flujo de corriente ($Z=\infty$). Un *swicht* es un elemento lógico que en estado “cerrado” se comporta como un conector, mas en estado “abierto” se comporta como un aislador.

La gran mayoría de elementos eléctricos existentes (focos, motores, etc.) pueden ser modelados como una combinación de las fuentes ideales y los conductores ideales comentados. La relación existente entre la tensión (ΔV) y la corriente (I) en estos elementos es mostrada en la Figura 2.1.

Elemento	Notación	Relación
Fuente AC		$\Delta V(t) = K(t) \quad \forall I(t)$
Resistor		$\Delta V(t) = RI(t)$
Inductor		$\Delta V(t) = L \frac{dI(t)}{dt}$
Condensador		$\Delta V(t) = \frac{1}{C} \int I(t) dt$
Cable ideal		$\Delta V(t) = 0 \quad \forall I(t)$

Figura 2.1: Relación entre ΔV e I para elementos eléctricos ideales.

El término *energía* (E) tiene diferentes unidades y aceptaciones. En la forma eléctrica es la cantidad de potencia entregada (o utilizada) en un intervalo de tiempo. Por ejemplo: un foco de 100 W de potencia consume en un día (24 h) 2400 Wh de energía (o 2.4 kWh). Si el costo de la energía fuese de S/. 0.35 por cada kWh, entonces utilizar un foco de 100 W un día entero costaría S/. 0.84.

El término *potencia* es utilizado para representar la variación del trabajo o energía con respecto al tiempo. En la forma eléctrica es el producto de la tensión por la corriente. Las cargas en algunos casos pueden ser representados por la potencia que necesitan (como en el caso del foco de 100 W.) en lugar de por su impedancia. La potencia total eléctrica (S) está compuesta de dos tipos de potencia: potencia activa y potencia reactiva.

El término *potencia activa* (P medida en Watts (W)) es utilizado para cuantificar la relevancia (fuerza) de una fuente. Un humano saludable debería manejar continuamente una bicicleta para disponibilizar una potencia necesaria para el funcionamiento de 5 focos de 100 W cada uno. Siguiendo los costos presentados recientemente, esto significaría que por 8 horas de trabajo el humano saludable recibiría: $(8 \cdot 5 \cdot 100) \cdot (0.35/1000) = 1.4$ soles; una cantidad ínfima comparada con un salario mínimo diario. Es por eso que la energía eléctrica revolucionó el mundo industrializado al proporcionar trabajo barato a través del uso intensivo del *motor* [eléctrico].

El término *potencia reactiva* (Q medida en VoltAmper reactivo (VAr)) es utilizado para cuantificar un tipo de potencia necesitada por algunos conductores para soportar el proceso de conversión de energía. Esta potencia incrementa la cantidad de corriente que abastece la carga, mas no produce energía útil. Por ejemplo: "Un golfista puede enviar una pelota a un hoyo distanciado 10 m. con varios ángulos de salida. La elevación que la pelota alcance es indiferente mientras sea consiga el objetivo (la pelota en el hoyo). La distancia horizontal vendría a ser el análogo a la potencia activa, mientras que la vertical a la potencia reactiva".

2.3. ANÁLISIS DE CIRCUITOS ELÉCTRICOS

2.3.1. DEFINICIÓN

Un *circuito* [eléctrico] o *red* [eléctrica] es una combinación de fuentes y conductores conectados entre sí con la finalidad de generar, transportar y/o modificar señales eléctricas. Un circuito AC que represente el encendido de una lámpara en un instante $t = 0.1$ s. es mostrado en la Figura 2.2.

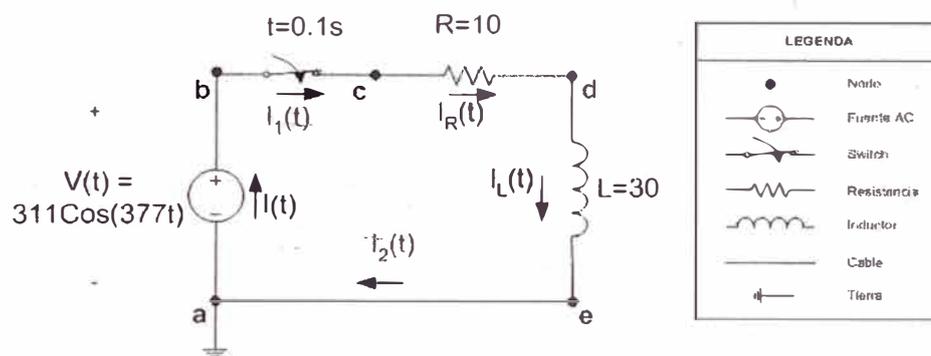


Figura 2.2: Ejemplo de circuito eléctrico AC.

El sistema de encendido de lámparas en nuestros hogares es un sistema dinámico físico. Un sistema puede ser definido como un conjunto de objetos agrupados por alguna interacción o interdependencia, de modo que existan relaciones de causa y efecto en los fenómenos que ocurren con los elementos de esos conjuntos. Un sistema es dinámico cuando algunas magnitudes que caracterizan sus objetos constituyentes varían en el tiempo. Un circuito eléctrico AC es un ejemplo de un sistema dinámico.

Determinar teóricamente la evolución temporal de las magnitudes que caracterizan un sistema puede ser importante en las siguientes situaciones:

- El sistema aún no existe físicamente.
- Se intenta explicar el comportamiento de sistemas ya existentes.
- Las pruebas experimentales son caras o peligrosas.

El objetivo de estos estudios teóricos es intentar prever el futuro de un modo científico. Para hacer esto, es necesario conocer cómo las cosas son y comprender las reglas que gobiernan los cambios que ocurrirán. De la teoría de cálculo se sabe que la variación de una magnitud $x(t)$ en un tiempo continuo t es medida por la derivada $dx(t)/dt$. Es así que el estudio matemático de cambios corresponde al estudio de ecuaciones diferenciales.

El estudio teórico de un sistema dinámico puede ser dividido didácticamente en dos etapas:

- La construcción de un modelo “adecuado”.
- El análisis de este modelo.

Dependiendo de los alcances del estudio que se requiera realizar, el sistema de encendido de lámparas en nuestros hogares puede ser adecuadamente modelado como el circuito eléctrico AC mostrado en la Figura 2.2. El análisis de circuitos se realiza con la finalidad de determinar alguna variable de interés (tensión, corriente, etc.) en algún elemento del circuito para algún instante de tiempo. Este análisis está basado fundamentalmente en las dos leyes de Kirchhoff [BOYL04]. Para comentar estas dos leyes es necesario primero definir los conceptos de *nodo*, *rama* y *mall*.

Un *nodo* es un punto en el circuito donde concurren varios conductores distintos. En el circuito de la Figura 2.2 se puede observar la existencia de 5 nodos (a , b , c , d y e). El nodo a es un nodo especial llamado de *referencia*, al poseer arbitrariamente un potencial eléctrico igual a cero.

Una *rama* es un conjunto de todos los elementos de un circuito comprendidos entre dos nodos consecutivos. Una *mall* es un conjunto de ramas que forman un bucle cerrado tal que no contenga elementos en su interior. En el circuito de la Figura 2.2 para el switch en un estado de “cerrado” se puede observar la existencia de una única mall ($abcdea$).

2.3.2. LEYES DE KIRCHHOFF

Las dos leyes de Kirchhoff provienen de la “ley de la conservación de la energía” y pueden ser resumidas de la siguiente manera:

- La primera ley de Kirchhoff (PLK) o ley de los nodos establece que en cualquier nodo la suma algebraica de las corrientes es igual a cero.
- La segunda ley de Kirchhoff (SLK) o ley de las mallas establece que en cualquier malla la suma algebraica de las diferencias de potencial es igual a cero.

En el circuito mostrado en la Figura 2.2 existe un switch con un estado inicial “abierto” que en el instante $t = 0.1$ s cambia su estado para “cerrado”. Este cambio de estado altera el número de nodos y mallas, originando un cambio en la configuración del circuito. El análisis del circuito se realiza aplicando la PLK y la SLK para cada una de sus configuraciones.

Aplicando la PLK para la configuración inicial ($0 \leq t \leq 0.1$) del circuito obtenemos: $I(t) = 0 = I_1(t) = I_R(t) = I_L(t) = I_2(t)$. La SLK no se puede aplicar en esta configuración ya que no existe ninguna malla.

Aplicando la PLK para la configuración final ($t \geq 0.1$) del circuito obtenemos: $I(t) = I_1(t) = I_R(t) = I_L(t) = I_2(t)$. Aplicando la SLK en esta configuración obtenemos: $V(t) - 0 - \Delta V_R(t) - \Delta V_L(t) - 0 = 0$. De la configuración anterior sabemos que $I(t=0.1) = 0$. Por lo tanto:

$$\begin{aligned} V_{Sen}(wt) - RI(t) - L \frac{dI(t)}{dt} &= 0 & (a) \\ I(0.1) &= 0 & (b) \end{aligned} \quad (2.1)$$

La ecuación diferencial (2.1) representa al modelo matemático correspondiente al circuito AC mostrado en la Figura 2.2 para un tiempo mayor o igual a 0.1 s. Circuitos AC que posean más de una malla son generalmente representados por un sistema de ecuaciones diferenciales.

La aplicación de la PLK y la SLK es fácilmente sistematizada e implementada dentro de un *software*, el cual con el uso (o auxilio) de rutinas de métodos numéricos permite analizar adecuadamente circuitos eléctricos. Los resultados del análisis del circuito mostrado en la Figura 2.2 utilizando el PSpice [CADE06] (*software* comercial para análisis de circuitos eléctricos) para un tiempo de simulación de 0.10 s., es mostrado en la Figura 2.3.

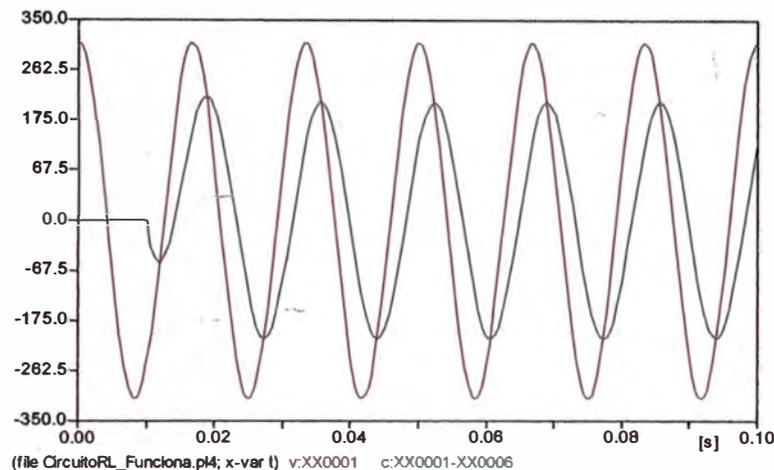


Figura 2.3: Resultados del análisis del circuito de la Figura 2.2.

En la Figura 2.3 puede observarse en color rojo la diferencia de potencial (tensión) existente entre los terminales de la fuente de tensión. Esta tensión tiene una forma de onda sinusoidal, con una amplitud y frecuencia constante. Puede observarse también en color verde la corriente que circula a través de los conductores. Inicialmente esta corriente posee un valor de cero. A partir del instante $t = 0.1$ s. (cierre del switch) la forma de onda de la corriente comienza a cambiar por un tiempo de tiempo denominado de *transitorio*, hasta estabilizarse en una forma sinusoidal con amplitud y frecuencia constante. Ya una vez estabilizada la corriente, es interesante notar que su frecuencia es la misma que la correspondiente a la fuente de tensión; y que existe una diferencia entre los tiempos en que estas dos ondas alcanzan sus valores máximos, es decir existe un *desfasaje* (δ) entre ellas. El valor de $I(t)$ durante el transitorio es denominado respuesta en *régimen transitorio*, mientras que el valor de $I(t)$ para tiempos mayores es denominado de respuesta en *régimen permanente*.

2.3.3. ANÁLISIS FASORIAL

El *análisis fasorial* es un análisis aplicable a circuitos con electricidad generada con una forma de onda sinusoidal de frecuencia constante, en el que las expresiones en el tiempo de las ondas de tensión y corriente son sustituidas por números complejos llamados de *fasores (A)*, de acuerdo a la siguiente equivalencia:

$$\mathbf{A} = A \angle \delta = A (\cos \delta + j \operatorname{Sen} \delta) \approx (\sqrt{2} A) \operatorname{Sen}(\omega t + \delta) \quad (2.2)$$

Donde:

- A Módulo del fasor (Valor eficaz de la onda sinusoidal).
- δ Ángulo del fasor (Desfasaje de la onda sinusoidal).
- j Unidad imaginaria ($j = \sqrt{-1}$).
- f Frecuencia de la onda sinusoidal.
- ω Velocidad angular ($\omega = 2\pi f$) de la onda sinusoidal.

Es importante notar que el derivar o integrar un fasor proporciona otro fasor con su amplitud multiplicada por una constante, y con un incremento o decremento de 90° en su desfasaje respectivamente. Esta característica nos permite redefinir las relaciones para los conductores ideales presentadas originalmente en la Figura 2.1, obteniendo una expresión matemática para sus impedancias (Z), tal y como es mostrado en la Figura 2.4:

Elemento	Notación	Notación Fasorial	Relación Fasorial
Resistor			$\Delta V_R = (R \angle 0^\circ) I = Z_R I$
Inductor			$\Delta V_L = (\omega L \angle 90^\circ) I = Z_L I$
Condensador			$\Delta V_C = \left(\frac{1}{\omega C} \angle -90^\circ\right) I = Z_C I$

Figura 2.4: Relación entre los fasores ΔV e I para conductores ideales.

La representación de los conductores por sus impedancias es de gran utilidad porque uniformiza y proporciona una relación lineal entre los fasores ΔV e I y principalmente porque permite la sustitución de un conjunto de impedancias por una única llamada de equivalente. El cálculo de la impedancia equivalente se consigue con la aplicación sucesiva de equivalencias básicas para la conexión de impedancias en serie y en paralelo. Estas equivalencias son mostradas en la Figura 2.5.

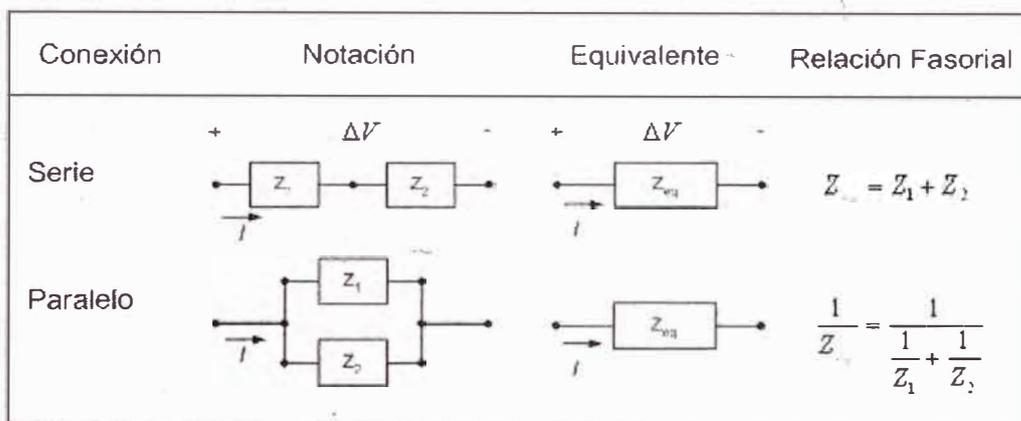


Figura 2.5: Equivalencias para conexiones en serie y paralelo de impedancias.

La *admitancia* (Y medida en Mho) de un conductor es calculado como la inversa de la impedancia. La utilización de admitancias en lugar de impedancias en el análisis fasorial se muestra más atractivo cuando se tienen elementos mayoritariamente conectados en paralelo dentro del circuito. Siendo números complejos tanto la impedancia como la admitancia, poseen parte real e imaginaria con una notación mostrada a continuación:

$$\mathbf{Z}_{km} = Z_{km} \angle \delta = R_{km} + j X_{km} \quad (\text{a})$$

$$\mathbf{Y}_{km} = 1/\mathbf{Z}_{km} = G_{km} + j B_{km} \quad (\text{b})$$

(2.3)

Donde:

- R_{km} Resistencia entre los nodos k y m.
- X_{km} Reactancia entre los nodos k y m.
- G_{km} Conductancia entre los nodos k y m.
- B_{km} Susceptancia entre los nodos k y m.

Las leyes de Kirchhoff siguen siendo válidas dentro del análisis fasorial. Un ejemplo de su aplicación es mostrado a continuación.

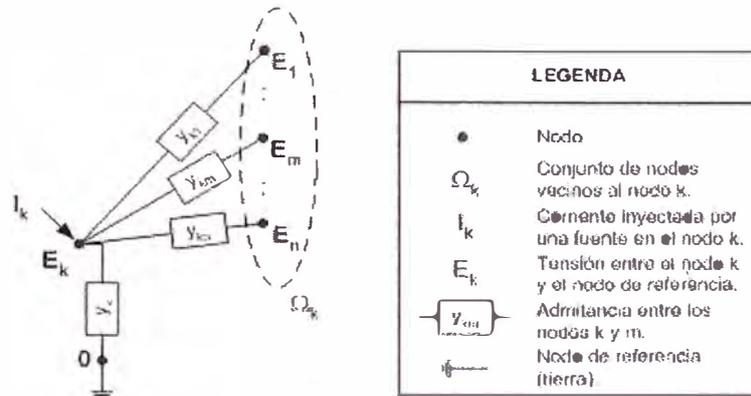


Figura 2.6: Análisis fasorial en un nodo genérico de un circuito AC.

En la Figura 2.6 se muestra un nodo k perteneciente a un circuito de n nodos (n no considera al nodo de referencia). Entre el nodo k y los demás nodos m ($m=1, \dots, n$) existen elementos representados por sus respectivas admitancias y_{km} . Entre el nodo k y el nodo de referencia existe una admitancia equivalente y_k . Entonces aplicando la PLK y SLK, tenemos:

$$I_k = y_k E_k + y_{k1}(E_k - E_1) + \dots + y_{km}(E_k - E_m) + \dots + y_{kn} E_{kn}$$

$$I_k = (y_k + y_{k1} + \dots + y_{km} + \dots + y_{kn}) E_k + (-y_{k1}) E_1 + \dots + (-y_{km}) E_m + \dots + (-y_{kn}) E_n$$

$$I_k = Y_{kk} E_k + Y_{k1} E_1 + \dots + Y_{km} E_m + \dots + Y_{kn} E_n$$

Repetiendo el procedimiento para todos los nodos y agrupando en forma matricial, tenemos:

$$\begin{bmatrix} I_1 \\ \vdots \\ I_k \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & \dots & Y_{1m} & \dots & Y_{1n} \\ \vdots & & & & \vdots \\ Y_{k1} & \dots & Y_{km} & \dots & Y_{kn} \\ \vdots & & & & \vdots \\ Y_{n1} & \dots & Y_{nm} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} E_1 \\ \vdots \\ E_k \\ \vdots \\ E_n \end{bmatrix} \quad \begin{aligned} Y_{km} &= -y_{km} \\ &\dots (2.4) \\ Y_{kk} &= y_k + \sum_{j \in \Omega_k} y_{kj} \end{aligned}$$

Este sistema de ecuaciones algebraicas con variables fasoriales representa al modelo matemático correspondiente al circuito AC en estudio.

En el ítem 2.3.2 se comentó que en un circuito con electricidad generada con una forma de onda sinusoidal de frecuencia constante, las respuestas de tensión y corriente en régimen permanente de los conductores tendrán también una forma de onda sinusoidal con la misma frecuencia mas con diferentes amplitudes y desfases. El análisis fasorial presentado recientemente permite el cálculo de estas respuestas en régimen permanente (amplitudes o módulos y desfases) sin resolver el sistema de ecuaciones diferenciales y resolviendo sólo un sistema de ecuaciones algebraicas. Esto una vez implementado en un *software* se traduce en un ahorro de esfuerzo y tiempo computacional considerable. Sin embargo, si se necesitase de los valores de las magnitudes en régimen transitorio tendría que resolverse el respectivo sistema de ecuaciones diferenciales.

2.3.4. ANÁLISIS EN VALORES POR UNIDAD (PU)

Dentro del análisis de circuitos existen cuatro magnitudes que son de especial interés: tensión, corriente, impedancia y potencia. Si dos de ellas son conocidas, las otras dos pueden ser obtenidas.

El hecho de normalizar el valor de estas magnitudes, o mejor dicho, de representar en los cálculos cada una de ellas como fracción de una magnitud fijada arbitrariamente llamada de magnitud base, reduce el esfuerzo realizado en un análisis fasorial y es llamado de análisis en valores por unidad (pu) [STEV94]. Este análisis se basa en la siguiente relación:

$$Magnitud_{pu} \angle \delta = \frac{Magnitud_{en_{unidad}_{correspondiente}} \angle \delta}{Base_{de_{la}_{magnitud}}} \dots (2.5)$$

Las magnitudes base son números reales, por lo tanto son los módulos de los fasores los que son expresados en pu, mientras que los desfases no son alterados. En lo que resta del presente informe, a menos que se indique lo contrario, las magnitudes se referirán a magnitudes fasoriales expresadas en pu.

2.3.5. SISTEMAS DE CONTROL

El circuito mostrado en la Figura 2.2 representó de una manera muy simplificada el encendido de una lámpara en nuestros hogares en un instante $t = 0.1$ s. Una vez que el *switch* fue cerrado la corriente cambió su valor desde cero hasta un valor $I \angle \alpha$, originando un cambio en las demás magnitudes del circuito; como por ejemplo en la tensión en el nodo d , la cual se estabilizó en un valor $V \angle \phi$. Si la lámpara en el circuito tuviese un valor diferente de inductancia ($L \neq 30$) el valor de la corriente sería diferente de $I \angle \alpha$, por lo que la tensión en el nodo d tendría también un valor diferente de $V \angle \phi$. Las lámparas fueron diseñadas para trabajar en nuestros hogares con una tensión (módulo del fasor) constante de 220 Volts, por lo que es de interés que esa tensión se mantenga constante independientemente del valor de la inductancia de la lámpara conectada. Esto es conseguido con la utilización de sistemas de control dentro de los circuitos eléctricos.

Los sistemas de control fueron relacionados por primera vez por Norbert Wiener [WIEN50] y se aplican en esencia para los organismos vivos, las máquinas y las organizaciones. Un sistema de control está definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado.

El control de temperatura en una habitación es un ejemplo claro y mucho más típico de una aplicación de la ingeniería de control. El objetivo es mantener la temperatura de una habitación en un valor deseado, aunque la apertura de puertas y ventanas y la temperatura en el exterior hagan que la cantidad de calor que pierde la habitación sean variables (disturbios externos). Para alcanzar el objetivo, el sistema de calefacción debe modificarse para compensar esas perturbaciones. Esto se hace a través del termostato, que mide la temperatura actual y la temperatura deseada, y modifica la temperatura del agua del sistema de calefacción para reducir la diferencia entre las dos temperaturas.

El sistema de control dentro de un circuito eléctrico sigue el mismo principio que el control de temperatura recientemente descrito. El modelo matemático correspondiente al circuito AC de la Figura 2.2 fue representado en la ecuación diferencial (2.1). Es importante comentar que esta ecuación diferencial es lineal para la variable dependiente $I(t)$. La forma más general para escribir una ecuación diferencial lineal para una variable $x(t)$ es:

$$a_n(t) \frac{d^n x(t)}{dt^n} + a_{n-1}(t) \frac{d^{n-1} x(t)}{dt^{n-1}} + \dots + a_1(t) \frac{d x(t)}{dt} + a_0(t)x(t) = F(t) \quad \dots (2.6)$$

donde las diversas derivadas $d^i x(t)/dt^i$ aparecen como términos de primer grado (elevados a la primera potencia), y los coeficientes a_i y la entrada $F(t)$ son funciones apenas de la variable independiente t (tiempo). Un sistema que puede ser adecuadamente representado por un modelo lineal es llamado de sistema [dinámico] lineal.

A pesar que en muchas ocasiones la mejor representación de un circuito AC se consigue con un modelo no lineal, el diseño de los diversos sistemas control en un circuito eléctrico es generalmente realizado considerándolo como un sistema lineal. La inclusión de los sistemas de control adiciona ecuaciones diferenciales lineales a las ecuaciones propias del circuito. Las relaciones entre las variables de los sistemas de control [lineales] son representados indistintamente por un sistema de ecuaciones diferenciales lineales o por su correspondiente diagrama de bloques. Un ejemplo de un diagrama de bloques puede ser observado en la Figura 2.7.

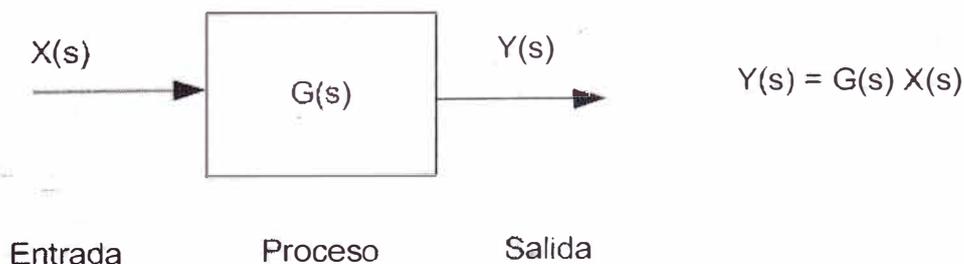


Figura 2.7: Ejemplo de diagrama de bloques para sistemas de control.

En la Figura 2.7 se puede observar que un bloque permite sintetizar las relaciones entre las entradas y salidas de un proceso mediante su modelo matemático. Se puede observar también que la variable independiente no es más el tiempo t y sí la frecuencia compleja o variable de Laplace s . La correspondencia entre una función dependiente del tiempo y dependiente de la variable de Laplace se consigue utilizando una transformación matemática lineal denominada "Transformada de Laplace", definida según la siguiente relación:

$$F(s) = L\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt \quad \dots (2.7)$$

Esta transformada tiene una serie de propiedades que la hacen útil en el análisis de sistemas lineales. Una de las ventajas más significativas radica en que la integración y derivación en el dominio de t se convierten en multiplicación y división en el dominio de s . Esto transforma las ecuaciones diferenciales e integrales en ecuaciones algebraicas, mucho más fáciles de resolver. Otra de las grandes ventajas radica en el hecho que un proceso puede ser representado por una función de la variable s llamada de función de transferencia ($G(s)$), tal que la salida del proceso ($Y(s)$) es igual a la entrada ($X(s)$) multiplicada por la función de transferencia ($G(s)$). Esto puede ser observado en la Figura 2.8.

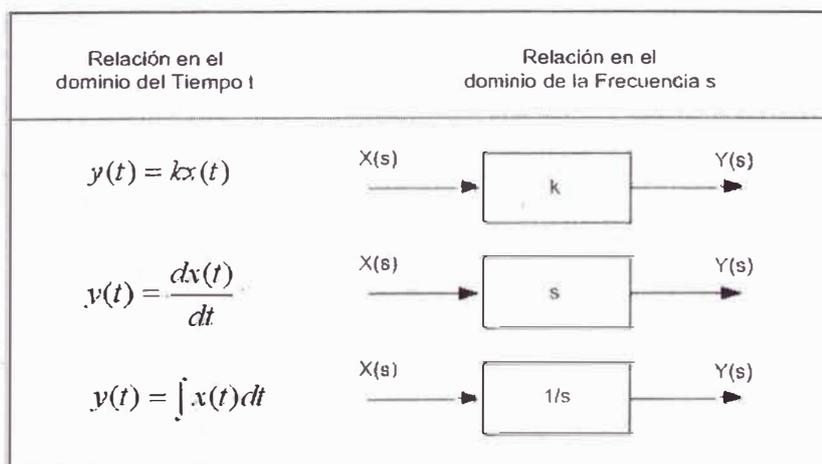


Figura 2.8: Equivalentes entre el dominio del tiempo y de la frecuencia.

El análisis fasorial permite analizar circuitos AC con una manipulación algebraica, mas es sólo válido para el cálculo de las respuestas en régimen permanente. El análisis en el dominio de la frecuencia permite analizar circuitos AC lineales así como sus sistemas de control (controladores) con una manipulación algebraica tanto para el cálculo de las respuestas en régimen transitorio como permanente. Si el circuito AC lineal no representa adecuadamente al sistema físico en análisis, entonces tiene que utilizarse un circuito AC no lineal y por lo tanto tienen que resolverse el respectivo sistema de ecuaciones diferenciales.

Prácticamente todos los sistemas físicos en el mundo son sistemas no lineales. Para determinar el comportamiento dinámico de estos sistemas es necesario especificar valores para sus condiciones iniciales y para los parámetros de las ecuaciones; para luego resolver el respectivo sistema de ecuaciones diferenciales. La solución obtenida tiene una utilidad bastante limitada, ya que es válida únicamente para la situación particular calculada. Realizar esa simulación para todas las posibles combinaciones de los parámetros y condiciones iniciales es un abordaje no muy inteligente. Una formulación más conveniente es la de determinar los parámetros y condiciones iniciales para las cuales el sistema presente un comportamiento dinámico estable.

El término estabilidad será definido y discutido más adelante (ítem 2.6), mas por ahora es conveniente entender un sistema estable como un sistema en que no existen crecimientos interminables en algunas variables que impidan el funcionamiento del sistema. El análisis de estabilidad de un sistema lineal es un tema estudiado en los dos últimos siglos, existiendo criterios para su determinación directamente, tal y como es comentado a continuación.

Una ecuación diferencial de orden "n" como la presentada en la ecuación (2.6) puede ser re-escrita como un sistema de "n" ecuaciones diferenciales de primer orden.

Sea:

$$x_1(t) = x(t); \quad x_2(t) = \frac{dx_1(t)}{dt}; \quad x_3(t) = \frac{dx_2(t)}{dt}; \dots; \quad x_n(t) = \frac{dx_{n-1}(t)}{dt};$$

Entonces

$$\frac{dx_n(t)}{dt} = \frac{F(t)}{a_n(t)} - \frac{a_{n-1}(t)}{a_n(t)} x_n(t) - \dots - \frac{a_2(t)}{a_n(t)} x_2(t) - \frac{a_1(t)}{a_n(t)} x_1(t)$$

Reordenando en una forma matricial tenemos:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ -a_1/a_n & -a_2/a_n & \dots & \dots & F/a_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix}$$

Este sistema tiene la forma $dx(t)/dt = A x$ (el vector x es llamado de vector de estados y la matriz A de matriz de estados). En estos sistemas es posible analizar la estabilidad del sistema analizando solamente la matriz A y siguiendo el primer criterio de estabilidad Lyapunov. Este criterio establece que si los autovalores (complejos) de la matriz A poseen parte real negativa el sistema es estable, si posee al menos un autovalor con parte real positiva el sistema es inestable, y si posee autovalores con parte real igual a cero no es posible definir la estabilidad.

Los controladores en los circuitos eléctricos son diseñados inicialmente considerando modelos lineales de forma de garantizar la estabilidad del sistema y finalmente para regular algunas magnitudes de interés dentro del circuito. La inclusión de controladores origina la inclusión de nuevas variables dentro del vector x , así como un cambio en la matriz A (y un cambio también en sus autovalores) originando un sistema de ecuaciones en el que aún se cumple el primer criterio de estabilidad de Lyapunov, como el mostrado en la ecuación (2.8).

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \quad \dots (2.8)$$

$$y(t) = Cx(t) + Du(t)$$

Donde:

- x Vector de las variables de estado
- u Vector de las variables de control
- y Vector de las variables medidas o controladas
- A Matriz de estados
- B Matriz de entrada.
- C Matriz de salida
- D Matriz de transmisión directa.

Se debe entender como estado de un sistema dinámico al menor conjunto de variables tales que el conocimiento de esas variables en $t=t_0$, juntamente con el conocimiento de la entrada para $t \geq t_0$, determina el comportamiento del sistema para cualquier $t \geq t_0$.

Si bien el primer criterio de estabilidad de Lyapunov nos proporciona información de si el sistema es estable o no, una forma de cuantificar esa estabilidad es con la utilización de los amortecimientos correspondientes a estos autovalores. Mientras mayor sea el amortecimiento, mayor será la estabilidad tal y como es mostrado en la Figura 2.9.

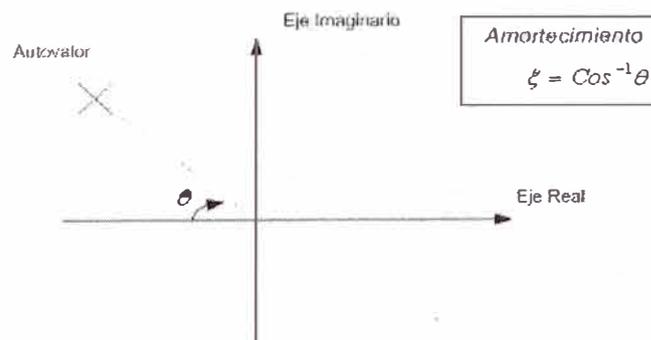


Figura 2.9: Amortecimiento de un autovalor complejo.

El principal interés de haber comentado los circuitos AC con sus respectivos controladores radica en el hecho que prácticamente todos los elementos eléctricos pueden ser modelado como un conjunto de los mismos.

2.4. SISTEMAS ELÉCTRICOS DE ENERGÍA (SEE)

2.4.1. DEFINICIÓN

Un Sistema Eléctrico de Energía (SEE) es un conjunto de elementos que operan en conjunto y de manera coordinada para atender completamente la demanda eléctrica, incurriendo en el menor costo total posible y manteniendo el mejor padrón de calidad posible. Entre los elementos tenemos a los generadores, transformadores, líneas de transmisión, medidores, etc. El padrón de calidad se refiere a que los niveles de tensión y frecuencia se deben encontrar dentro de una faja, que el servicio no debe sufrir interrupciones, que la forma de onda debe ser lo más sinusoidal posible y que el impacto ambiental debe ser mínimo, entre otros. La *demanda* es la cantidad de electricidad requerida por los usuarios, la cual es manipulada con unidades de potencia. En la Figura 2.10 se observa un ejemplo básico de un SEE.

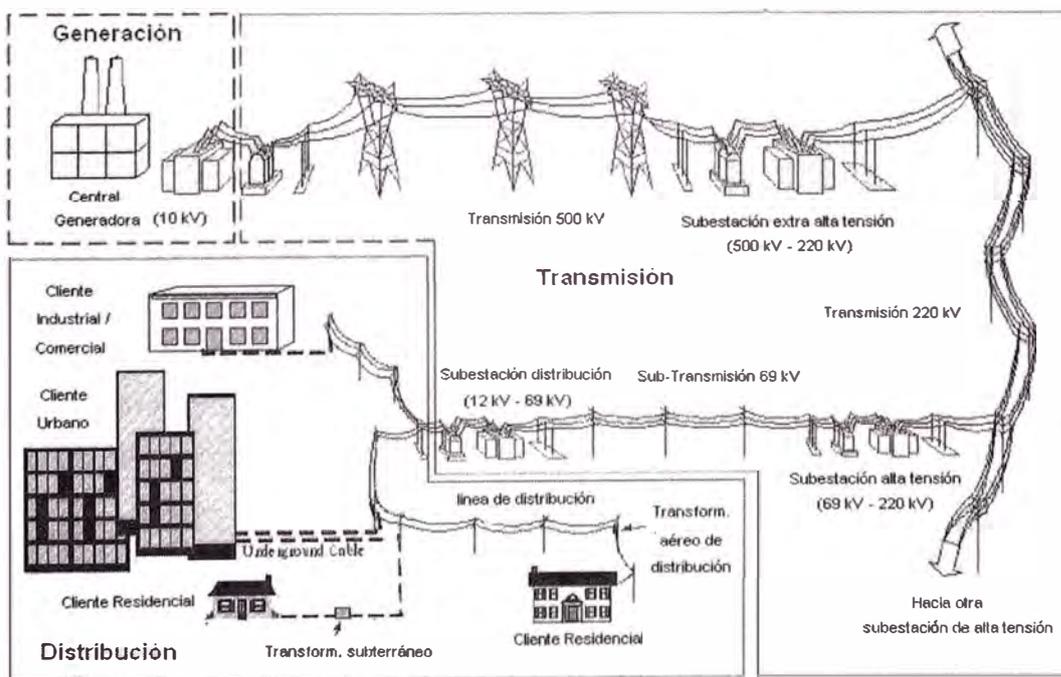


Figura 2.10: Ejemplo básico de un sistema eléctrico de energía.

2.4.2. ESTRUCTURA DE LOS SEE

En la Figura 2.10 se pueden distinguir claramente los tres principales sistemas que conforman un SEE: el sistema de generación, de transmisión y de distribución. Adicionalmente a estos tres sistemas puede mencionarse el sistema de adquisición de datos y supervisión o *Supervisory Control And Data Acquisition* (SCADA).

El sistema de generación está conformado principalmente por los generadores. Los generadores son máquinas eléctricas impulsadas por turbinas que convierten la energía mecánica en eléctrica. Los dos tipos más comunes de turbinas son las turbinas accionadas por la producción de calor originario de carbón, gas, petróleo, etc. (turbinas térmicas); y las turbinas accionadas por el movimiento de grandes cantidades de agua (turbinas hidráulicas). Los generadores térmicos pueden ser construidos lo más cerca posible de los centros de consumo, pero poseen un costo de operación compuesto principalmente por el costo de su materia prima (carbón, gas, etc.). Los generadores hidráulicos tienen que ser construidos en los lugares en que la naturaleza proporciona la materia prima (agua) prácticamente gratis, lugares que se encuentran generalmente lejos de los centros de consumo, por lo que se hace necesario la utilización de un sistema para transmitir la energía eléctrica producida.

El sistema de transmisión está compuesto principalmente por líneas de transmisión y transformadores. Por razones económicas la transmisión es efectuada en niveles de tensión elevados (118 kV, 220 kV) a través de líneas de transmisión. Debido a limitaciones constructivas y de aislamiento, los generadores no pueden operar en tales niveles de tensión. Típicamente los generadores producen tensiones en la faja de 10 kV a 36 kV, necesitando de transformadores, localizados en las subestaciones elevadoras, para elevar la tensión para los niveles de transmisión. Próximo a los centros de carga, los niveles de tensión tienen que ser reducidos nuevamente por razones de seguridad, a través de subestaciones reductoras.

El sistema de distribución está compuesto principalmente por líneas de sub-transmisión y transformadores que trabajan en un nivel de tensión de 10 kV a 220 V -110 V con la finalidad de distribuir la energía eléctrica recibida del sistema de transmisión a todos los usuarios.

La demanda no es una magnitud constante, está continuamente variando. La electricidad AC es relativamente fácil de producir, mas hasta la fecha no se poseen mecanismos para su almacenamiento de forma económicamente viable. Es por eso que el encender una lámpara de 100 W obliga a algún generador (en ese mismo instante) a aumentar su potencia generada en 100 W. Este aumento origina que algunas magnitudes (frecuencia, tensión en los generadores, etc.) también varíen, siendo necesaria la aplicación de algún esquema de control sobre diversos elementos de los SEE, de manera que esas variaciones no perjudiquen el padrón de calidad establecido.

El control de los SEE es realizado en los centros de supervisión y control, también llamados de *Energy Management System* (EMS), los cuales utilizan un sistema SCADA para monitorear y capturar todos los datos necesarios y relevantes de los SEE. Los sistemas SCADA son la interfaz entre el SEE y los EMS. Los EMS son los encargados de procesar todos los datos recibidos efectuando una serie de análisis y estudios que permitan conocer el estado actual de la red, simular la ocurrencia de posibles fallas, y determinar acciones de control para una operación económica y segura.

En el Perú, el sistema de energía eléctrico nacional es llamado de Sistema Interconectado Nacional (SINAC), y tal como se muestra en la Figura 2.11 cubre una extensión geográfica importante en el país (desde Tumbes hasta Tacna) con niveles de tensión de transmisión de 220 kV. y una capacidad instalada de 4757 MW. La operación y control del SINAC es realizada por el EMS denominado Comité de Operación Económica del Sistema Interconectado Nacional (COES SINAC).

MAPA LÍNEAS DE TRANSMISIÓN ELÉCTRICA

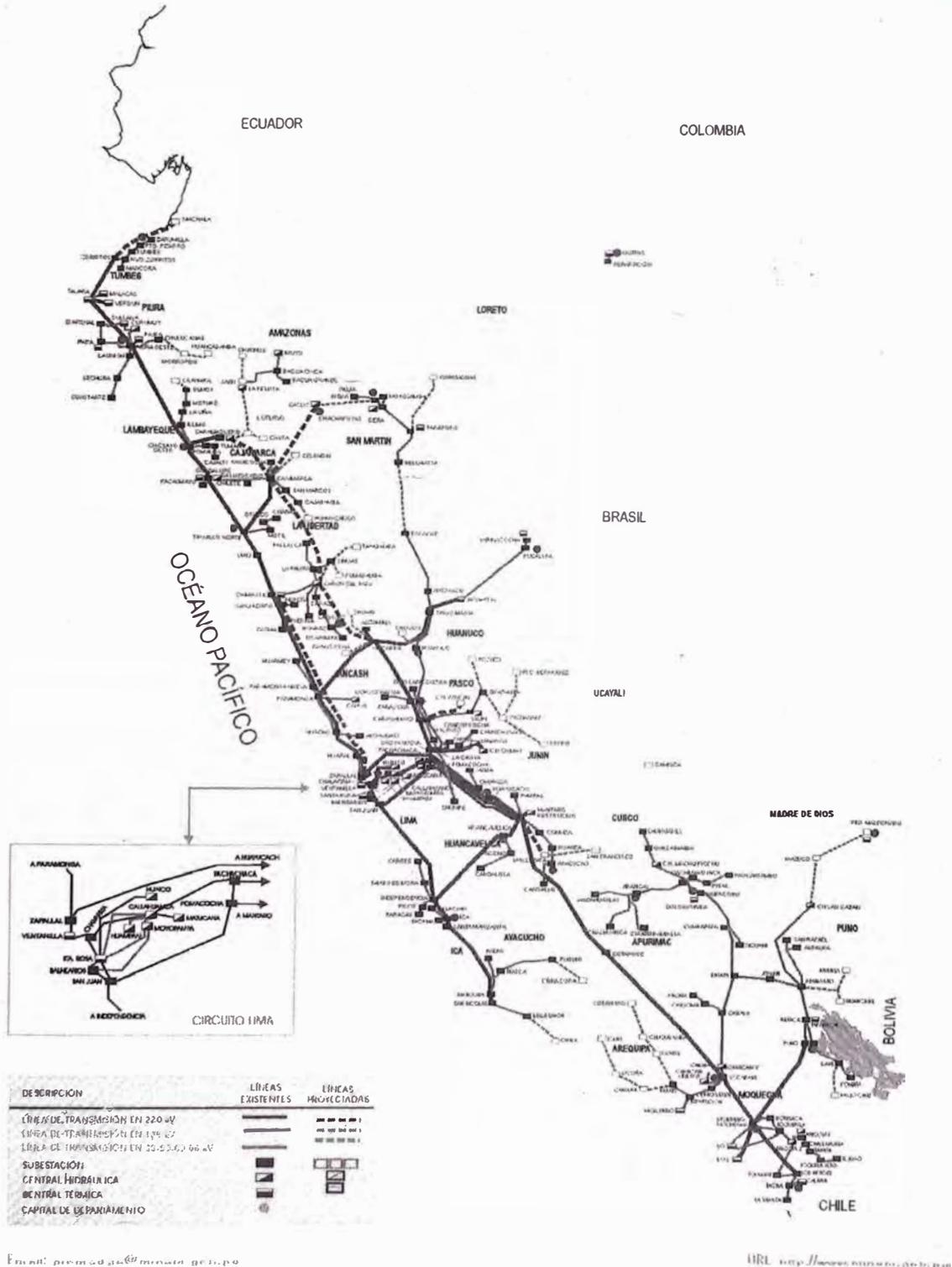


Figura 2.11: Mapa del Sistema Interconectado Nacional (SINAC).

2.4.3. DIAGRAMA UNIFILAR

En la Figura 2.10 se puede observar un aspecto adicional a los ya comentados, mas no menos importante. Los SEE están normalmente compuestos de 3 fases (trifásico), es decir, la energía es transmitida utilizando una configuración de 3 conductores de manera que las formas de onda de la tensión y la corriente varíen simétricamente consiguiendo una potencia total (generada / consumida) con un valor medio constante. Esta característica es importante porque esta configuración es fácil de producir y presenta una eficiencia superior a otras configuraciones.

La unión de los diversos elementos que conforman un SEE puede ser representado mediante la utilización de un diagrama. Cuando cada una de sus fases atiende una parte equitativa de la demanda total, se dice que el SEE se encuentra balanceado. En un sistema trifásico balanceado el comportamiento en cada una de las fases es similar, por lo que raramente es necesario analizar más de una fase. Entonces el diagrama correspondiente puede ser simplificado mostrando sólo una fase del sistema e indicando diversas de sus partes mediante símbolos estándares en lugar de sus circuitos equivalentes. A este diagrama simplificado se le llama *diagrama unifilar* y sirve principalmente para la representación de la topología del sistema y de resultados de sus análisis. En la Figura 2.12 se puede apreciar el diagrama unifilar correspondiente al SEE mostrado en la Figura 2.10.

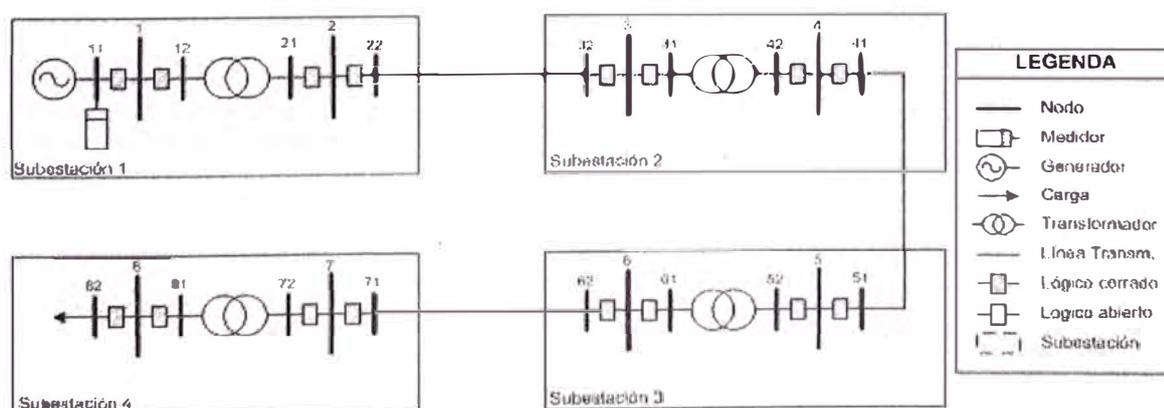


Figura 2.12: Diagrama unifilar del SEE mostrado en la Figura 2.10.

2.4.4. CONFIGURADOR DE RED

Un diagrama unifilar muestra de manera representativa y simplificada la conectividad existente entre los diversos elementos pertenecientes a un SEE, y en algunas ocasiones una configuración de operación de esta conectividad. La conectividad permanece inalterada a menos que algún elemento físico sea adicionado o retirado del sistema. La representación de la conectividad es llamada de *representación física* del SEE.

Una misma conectividad puede presentar diferentes configuraciones de operación dependiendo del estado (abierto o cerrado) de los diversos elementos lógicos. Estos elementos (disyuntores, seccionadores, etc.) son modelados como switches y modifican su estado con el objetivo de aislar o conectar elementos existentes en el sistema. Para motivos de análisis [de circuitos] el retiro de los switches abiertos y la unión de los terminales de un switch cerrado en un solo *nodo resultante* no altera los resultados. Este *nodo resultante* (unión de uno o varios nodos físicos) es llamado de *barra*. La representación de una configuración de la conectividad es llamada de *representación lógica* del SEE, y es obtenido a través de un proceso sistematizado llamado de *configurador de red*. La aplicación de un configurador de red en la conectividad mostrada en la Figura 2.12 es mostrada en la Figura 2.13.

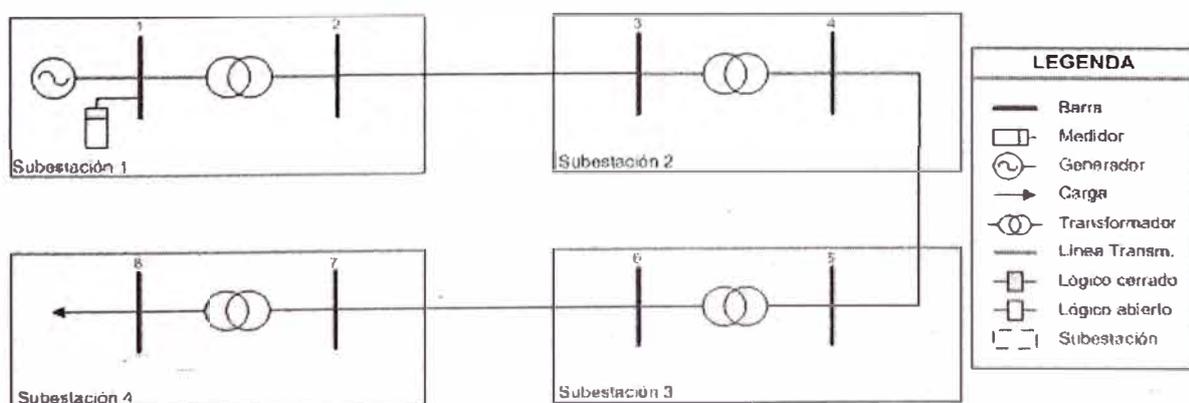


Figura 2.13: Representación lógica del SEE mostrado en la Figura 2.12.

2.4.5. ANÁLISIS DE SEE

Los objetivos de un SEE son alcanzados con un adecuado planeamiento y operación del sistema. El planeamiento de un SEE generalmente es dividido según el horizonte de tiempo considerado y tiene como principal objetivo determinar las adiciones, modificaciones y maniobras de elementos en el sistema para garantizar que una demanda futura sea atendida. Generalmente una primera división del planeamiento es efectuada entre el planeamiento de la expansión y planeamiento de la operación. El planeamiento de la expansión abarca periodos de tiempo 20 años (largo plazo), 10 años (medio plazo) y de 3 años (corto plazo); teniendo como principal objetivo determinar los nuevos elementos que serán incorporados en el SEE. El planeamiento de la operación abarca periodos de tiempo de 1 mes a 1 día; teniendo como principal objetivo determinar como deberán interactuar los diversos elementos existentes en el SEE ante una lista de probables ocurrencias. La operación del SEE es efectuado en tiempo real siguiendo las directrices obtenidas del planeamiento de la operación ante ocurrencias similares a las consideradas, o siguiendo el "buen senso" del operador (algunas veces documentado) ante condiciones completamente diferentes a las consideradas en el planeamiento de la operación.

Un adecuado planeamiento y operación de los SEE exige la realización de una serie de análisis, cada uno de los cuales son efectuados con el uso intensivo de diversos simuladores digitales (softwares). Los softwares utilizados pueden ser agrupados en: [softwares para] análisis energéticos, análisis eléctricos, análisis de hidrología y análisis de previsión de la demanda. Los softwares para análisis eléctricos resuelven modelos matemáticos que simulan el comportamiento conjunto (en régimen transitorio y/o régimen permanente) de los diversos elementos que conforman el SEE ante la presencia de diversos fenómenos, productos de la relación del sistema con su medio ambiente o del irregular funcionamiento de algunos de sus elementos. Entre los principales fenómenos [dinámicos] podemos mencionar los que se muestran en la Figura 2.14.

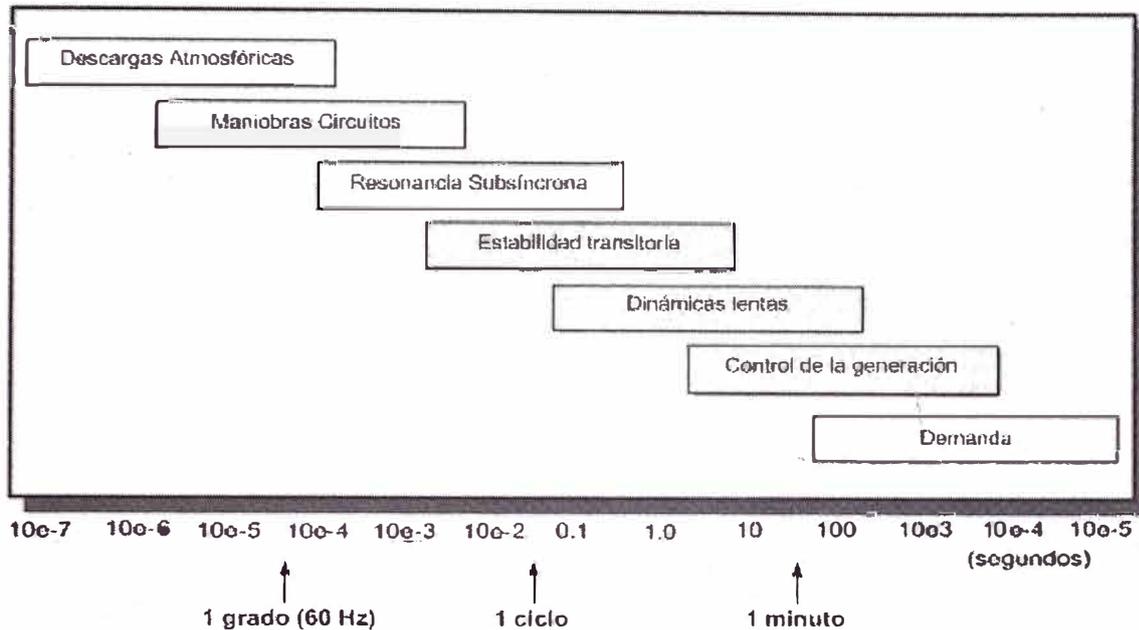


Figura 2.14: Fenómenos dinámicos en SEE.

Los fenómenos mostrados en la Figura 2.14 están agrupados por el tiempo que generalmente demoran para demostrar un efecto considerable en el SEE. Existen fenómenos que se manifiestan en 10^{-7} s., así como algunos que demoran horas para manifestarse. El comportamiento de los elementos es diferente para cada grupo de fenómenos, lo que origina un modelo matemático diferente del SEE, lo que finalmente se traduce en la utilización de un *software* diferente. Históricamente los efectos de estos fenómenos han sido simulados con softwares para análisis de cortocircuito, flujo de carga, flujo de potencia óptimo, análisis armónico, análisis de autovalores, transitorios electromagnéticos y transitorios electromecánicos. Cada uno de estos análisis posee una complejidad que justifica su tratamiento como línea de investigación (o especialidad) dentro de la comunidad científica de energía eléctrica.

Tres de los principales análisis mencionados: flujo de potencia, estabilidad permanente (análisis de autovalores) y estabilidad transitoria (transitorios electromecánicos), son comentados con mayor detalle a continuación.

2.5. FLUJO DE POTENCIA

2.5.1. DEFINICIÓN

La demanda es típicamente representada como la potencia (activa y reactiva) requerida por los diversos usuarios del SEE. Esta potencia requerida es generada en el sistema de generación y fluye a través del sistema de transmisión y/o distribución hasta llegar a los usuarios. La demanda es una magnitud que varía instantáneamente, mas esta variación es lenta, por lo que puede considerarse constante en pequeños intervalos de tiempo (típicamente del orden de minutos). Dentro de cada uno de estos intervalos, las respuestas de régimen transitorio (típicamente del orden de milisegundos) se muestran despreciables predominando sólo las respuestas de régimen permanente. Un análisis de flujo de potencia permite el cálculo de los diversos flujos de potencia en el SEE en régimen permanente.

Formalmente un análisis de flujo de potencia consiste en la determinación o cálculo del estado de operación estático (en régimen permanente) de la red eléctrica [MONT83]. El estado de un sistema para un instante de tiempo está representado por los valores que asumen sus variables de estado en ese instante de tiempo. Las variables de estado son el conjunto mínimo de variables que permite determinar el comportamiento actual y futuro del sistema. Para un análisis de flujo de potencia las variables de estado son las tensiones complejas (ángulo y módulo) en los nodos, las cuales una vez conocidas nos permiten calcular los flujos de potencias a través de los elementos, así como otras magnitudes de interés.

Este análisis inicialmente fue denominado de “flujo de carga”, mas este nombre ya ha sido criticado sugiriéndose el uso de “flujo de potencia” (debido a que la carga no fluye, y la potencia sí), mas su amplia aceptación histórica ha desencadenado el uso indistinto de los dos nombres. El flujo de potencia es el análisis más utilizado y por qué no decir el más importante para los ingenieros de sistemas de potencia.

2.5.2. MODELOS UTILIZADOS

El análisis de flujo de potencia se realiza sobre una representación lógica (producto de un configurador de red) del SEE. En esta representación se distinguen principalmente los nodos o barras, los elementos shunt y los elementos serie.

En la Figura 2.12 se pueden observar ejemplos de elementos shunt y serie. Un elemento shunt es un elemento que sólo está conectado a una barra (se sobre-entiende que el otro terminal del elemento está conectado al nodo de referencia) como por ejemplo: los generadores, las cargas, etc. Un elemento serie es un elemento que está conectado a dos barras como por ejemplo: los transformadores, las líneas de transmisión, etc. Cada uno de estos elementos posee un modelo que generalmente varía entre ser una inyección (o retiro) de potencia externa que controla (o no) alguna variable de la barra, o un conjunto de conductores ideales (R, L, C) en las cuales se cumplen las dos leyes de Kirchhoff. Los modelos de los principales elementos son mostrados en la Figura 2.15.

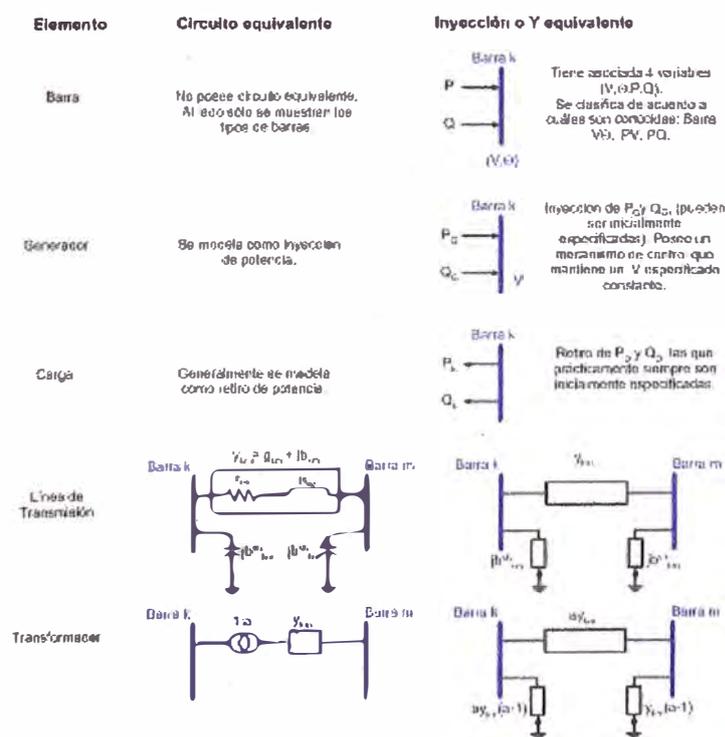


Figura 2.15: Modelos de elementos para análisis de flujo de potencia.

2.5.3. FORMULACIÓN MATEMÁTICA

El análisis de flujo de potencia es formulado aplicando las dos leyes de Kirchhoff al circuito resultante del reemplazo de los elementos eléctricos por sus respectivos modelos. Siguiendo la nomenclatura establecida en la ecuación (2.4) se determinan en los elementos serie las relaciones entre las potencias que fluyen a través de ellos con las tensiones entre sus terminales (barras k y m). Estas relaciones son mostradas en la ecuación (2.9) quedando evidente que si son conocidas las tensiones complejas (V, θ) los diversos flujos también son conocidos.

$$\begin{aligned}
 P_{km} &= (aV_k)^2 g_{km} - aV_k V_m (g_{km} \cos \theta_{km} + b_{km} \operatorname{Sen} \theta_{km}) \\
 Q_{km} &= -(aV_k)^2 (b_{km} + b_{km}^{sh}) - aV_k V_m (g_{km} \operatorname{Sen} \theta_{km} - b_{km} \cos \theta_{km})
 \end{aligned}
 \tag{2.9}$$

Algunas tensiones así como algunas inyecciones de potencia (potencia activa generada; potencia activa / reactiva de la demanda) son especificadas inicialmente al indicar el tipo de barra. El problema de flujo de carga consiste en determinar las tensiones complejas restantes que produzcan en cada barra una suma de flujos igual a las inyecciones de potencia especificadas. Este problema es formulado matemáticamente tal y como mostrado en las ecuaciones (2.10).

$$\begin{aligned}
 P_k^{esp} = P_k(V, \theta) &= V_k \sum_{m \in \{\Omega_k + k\}} V_m (G_{km} \cos \theta_{km} + B_{km} \operatorname{Sen} \theta_{km}) & k = 1, \dots, n_B \\
 Q_k^{esp} = Q_k(V, \theta) &= V_k \sum_{m \in \{\Omega_k + k\}} V_m (G_{km} \operatorname{Sen} \theta_{km} - B_{km} \cos \theta_{km}) & k = 1, \dots, n_B
 \end{aligned}
 \tag{2.10}$$

Donde:

P_k^{esp}	Potencias activa especificada en la barra k .
$P_k(V, \theta)$	Suma de potencias activas P_{km} en la barra k .
Q_k^{esp}	Potencia reactiva especificada en la barra k .
$Q_k(V, \theta)$	Suma de potencias reactivas Q_{km} en la barra k .
n_B	Número de barras del sistema.
Ω_k	Conjunto de barras vecinas a la barra k .
V_k	Módulo de la tensión compleja en la barra k .
θ_{km}	Diferencia de ángulos de la tensión entre las barras k y m .

2.5.4. FLUJO DE POTENCIA NO LINEAL O AC

El sistema de ecuaciones presentado en la ecuación (2.10) es un sistema de ecuaciones no lineal representativo del análisis de flujo de potencia AC. Este sistema puede ser resuelto utilizando [STEV94]: el método de Gauss-Siedel, método de Newton, métodos desacoplados, etc. De entre estos métodos el método de Newton es el que más se ha destacado, siendo el más utilizado dentro de los softwares comerciales.

Método de Newton

La formulación matemática del problema de flujo de potencia mostrada en la ecuación (2.10), puede ser conceptualizada como un sistema de ecuaciones no lineales de la forma:

$$g(\underline{x}) = \begin{bmatrix} \Delta P(V, \theta) \\ \Delta Q(V, \theta) \end{bmatrix}_{1 \times 2n_B} = \begin{bmatrix} P_k^{exp} - P_k(V, \theta) \\ Q_k^{exp} - Q_k(V, \theta) \end{bmatrix}_{1 \times 2n_B} = \underline{0} \dots\dots(2.11)$$

El método de Newton resuelve problemas matemáticos de la forma $g(x)=0$, a través de un proceso iterativo en el que va corrigiendo x de manera que $g(x)$ se aproxime al origen, que finaliza cuando $|g(x)|$ alcanza un valor menor que ε denominada error. Este método utiliza una aproximación de primer orden de la función g que relaciona la variación de g (Δg) con la variación de x (Δx), tal y como mostrado a continuación:

$$g(x + \Delta x) = g(x) + \frac{dg(x)}{dx} \Delta x$$

Suponemos un x inicialmente conocido (por lo tanto $g(x)$ también es conocido) y deseamos calcular un Δx tal que $g(x+\Delta x)$ sea igual a cero. Denominando a la derivada de primer orden ($dg(x)/dx$) como Jacobiana ($J(x)$), para una iteración i tendríamos la siguiente corrección:

$$\Delta x^i = -\frac{g(x^i)}{J(x^i)} = [J(x^i)]^{-1} g(x^i)$$

La aplicación del método de Newton para la resolución del problema de flujo de potencia requiere algunos comentarios previos. Por convención las potencias externas inyectadas poseen un signo positivo y las retiradas un signo negativo, mientras que los flujos de potencia que salen de una barra poseen un signo positivo y los que llegan un signo negativo. Considerando n_α como el número de barras de tipo α (V θ , PV, PQ), la aplicación del método de Newton para resolver el problema de flujo de potencia, produce para cada iteración i el siguiente vector de corrección:

$$\begin{bmatrix} \Delta\theta' \\ \Delta V' \end{bmatrix} = \begin{bmatrix} H(V', \theta') & | & N(V', \theta') \\ M(V', \theta') & | & L(V', \theta') \end{bmatrix}^{-1} \begin{bmatrix} \Delta P(V', \theta') \\ \Delta Q(V', \theta') \end{bmatrix} \quad \left. \begin{array}{l} \} n_{PQ} + n_{PV} \\ \} n_{PQ} \end{array} \right\} \dots (2.12)$$

Donde:

$$\underline{x} = \begin{bmatrix} \theta \\ V \end{bmatrix}; \quad \Delta \underline{x} = \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} \quad \left. \begin{array}{l} \} n_{PQ} + n_{PV} \\ \} n_{PQ} \end{array} \right\}; \quad g(\underline{x}) = \begin{bmatrix} \Delta P(V, \theta) \\ \Delta Q(V, \theta) \end{bmatrix} \quad \left. \begin{array}{l} \} n_{PQ} + n_{PV} \\ \} n_{PQ} \end{array} \right\};$$

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial P(V, \theta)}{\partial \theta} & | & \frac{\partial P(V, \theta)}{\partial V} \\ \frac{\partial Q(V, \theta)}{\partial \theta} & | & \frac{\partial Q(V, \theta)}{\partial V} \end{bmatrix} = \begin{bmatrix} H(V, \theta) & | & N(V, \theta) \\ M(V, \theta) & | & L(V, \theta) \end{bmatrix} \quad \left. \begin{array}{l} \} n_{PQ} + n_{PV} \\ \} n_{PQ} \end{array} \right\}$$

$$H_{kk}(V, \theta) = V_k \sum_{m \in \Omega_k} V_m (-G_{km} \text{Sen} \theta_{km} + B_{km} \text{Cos} \theta_{km})$$

$$H_{km}(V, \theta) = V_k V_m (G_{km} \text{Sen} \theta_{km} - B_{km} \text{Cos} \theta_{km})$$

$$N_{kk}(V, \theta) = 2V_k G_{kk} + \sum_{m \in \Omega_k} V_m (G_{km} \text{Cos} \theta_{km} + B_{km} \text{Sen} \theta_{km})$$

$$N_{km}(V, \theta) = V_k (G_{km} \text{Cos} \theta_{km} + B_{km} \text{Sen} \theta_{km})$$

$$M_{kk}(V, \theta) = V_k \sum_{m \in \Omega_k} V_m (G_{km} \text{Cos} \theta_{km} + B_{km} \text{Sen} \theta_{km})$$

$$M_{km}(V, \theta) = -V_k V_m (G_{km} \text{Cos} \theta_{km} + B_{km} \text{Sen} \theta_{km})$$

$$L_{kk}(V, \theta) = -2V_k B_{kk} + \sum_{m \in \Omega_k} V_m (G_{km} \text{Sen} \theta_{km} - B_{km} \text{Cos} \theta_{km})$$

$$L_{km}(V, \theta) = V_k (G_{km} \text{Sen} \theta_{km} - B_{km} \text{Cos} \theta_{km})$$

La aplicación del método de Newton puede ser resumido como sigue:

- i. Inicio
- ii. $i = 0$ (Contador de generaciones)
- iii. Escoger un estado inicial $x = (V, \theta)$
- iv. Calcular $P(V, \theta)$ y $Q(V, \theta)$ según la ecuaciones (2.10).
- v. Mientras [$\Delta P(V, \theta) > \text{Error}$ o $\Delta Q(V, \theta) > \text{Error}$]
- vi. Si ($i > \text{Limite_Iterac.}$) Entonces GoTo Fin (No existe solución)
- vii. Calcular las correcciones Δx según las ecuaciones (2.12).
- viii. Actualizar el estado ($x \text{ nuevo} = x + \Delta x$)
- ix. Calcular $P(V, \theta)$ y $Q(V, \theta)$ según la ecuaciones (2.10).
- x. $i = i + 1$
- xi. Continua
- xii. Fin

El procedimiento descrito puede ser implementado en un *software*, en el que se comprueba que casi todo el esfuerzo computacional recae en el paso vii. Un sistema de porte real como el SINAC es modelado con poco más de 1000 barras, lo que origina un Jacobiano de dimensiones aproximadas de 2000x2000, el cual en principio debería ser calculado e invertido en cada iteración (paso vii). Softwares comerciales de flujo de potencia calculan Δx sin invertir $J(x)$ y explotan una de las características fundamentales de $J(x)$: su dispersidad. Matrices dispersas son aquellas en la que la mayoría de sus elementos son iguales a cero tal y como es mostrado en la Figura 2.16. Existen técnicas dentro del Algebra Lineal para un tratamiento computacional adecuado de sistemas lineales conformados por matrices y vectores dispersos, los cuales consiguen reducir una simulación para un sistema como el SINAC de horas a segundos.

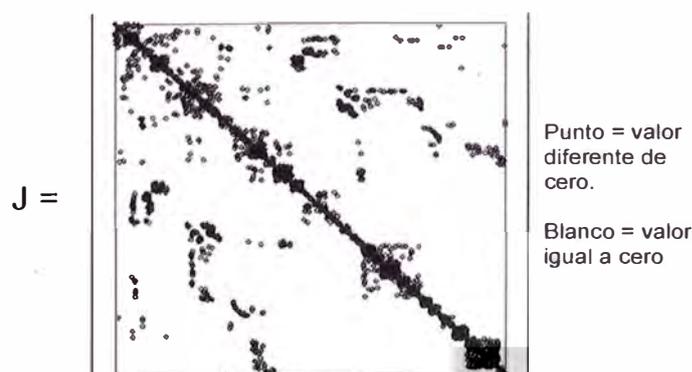


Figura 2.16: Ejemplo de matriz dispersa.

2.5.5. FLUJO DE POTENCIA LINEAL O DC

Bajo ciertas consideraciones el sistema de ecuaciones no lineal presentado en la ecuación (2.10) puede ser tratado como un sistema de ecuaciones lineal. En estos casos se dice que se está trabajando con un análisis de flujo de potencia lineal o DC.

En niveles de tensión elevados se pueden apreciar ciertas características en los SEE tales como: los módulos de las tensiones en las barras son cercanos a la unidad ($V_k = V_m = 1$), la diferencia de ángulos en las barras próximas es cercana a cero ($\text{Sen}\theta_{km} = \theta_{km}$, $\text{Cos}\theta_{km} = 1$), las resistencias en las líneas de transmisión son mucho menores que las reactancias ($r \ll X$) y los transformadores operan con valores de tap cercanos a la unidad ($a=1$). La consideración de estas características en las ecuaciones (2.9) origina las ecuaciones presentadas en (2.13).

$$P_{km} = -b_{km}\theta_{km} \quad Q_{km} = 0 \quad \dots(2.13)$$

Esta ecuación indica que sólo existe un flujo de potencia activa a través del elemento serie, el cual depende *linealmente* de la abertura angular existente entre sus terminales. Como los módulos de las tensiones son conocidos, el problema del flujo de potencia se reduce a la determinación de los ángulos (θ_k) de las tensiones que produzcan en cada barra una suma de flujos de potencia activa igual a las inyecciones activas especificadas (P_k^{esp}). Este problema es formulado matemáticamente tal y como mostrado en el sistema de ecuaciones (2.14).

$$P^{esp} = B\theta \quad \dots(2.14)$$

Este sistema es un sistema de ecuaciones lineal, donde la matriz B es una matriz constante que representa la red (formada por los b_{km} de los elementos serie) siendo altamente dispersa. Es común afirmar en el argot eléctrico que la matriz B representa el modelo DC de la red del SEE.

La solución del sistema de ecuaciones lineales (2.14) puede ser obtenida directamente como:

$$\theta = B^{-1} P^{exp}$$

Esta solución se obtiene con bajo costo computacional y posee una precisión aceptable para los flujos activos en muchas aplicaciones tanto en el planeamiento como en la operación del SEE.

Se debe observar que el modelo DC “no considera” las magnitudes de las tensiones, las potencias reactivas y los taps de los transformadores. Por esta razón, él no puede sustituir por completo los métodos no lineales de flujo de potencia, mas tiene todavía gran utilidad en las fases preliminares de estudios que exigen un análisis de un gran número de casos o que difícilmente pueden ser realizados utilizando métodos convencionales. En fases subsecuentes de los estudios, si fuese necesario el conocimiento de estas magnitudes entonces se debe partir para una solución exacta utilizando alguno de los métodos clásicos de flujo de potencia (Newton, desacoplado, etc.).

2.6. ESTABILIDAD DE SEE

2.6.1. DEFINICIÓN

La estabilidad de SEE es la habilidad del sistema, para dada una *condición de operación* inicial, retornar a un *estado de operación de equilibrio* luego de ser sometido a un *disturbio* físico [KUND04].

Un SEE está en un *estado de operación de equilibrio* si todas las cantidades físicas que se miden (o se calculan) y que describen la *condición de operación* del sistema, se pueden considerar constantes para propósitos de análisis. Si, cuando se está en una condición de operación de equilibrio, ocurre un cambio repentino o una secuencia de cambios en uno o más parámetros del sistema, o en una o más de sus cantidades de operación, se dice que el sistema experimenta un *disturbio*. Un pequeño cambio en la demanda o en la generación son ejemplos de *disturbios pequeños*, ya cambios bruscos en la demanda, pérdida de generadores, fallas en los sistemas de transmisión, etc., son ejemplos de *disturbios grandes*.

Ante la ocurrencia de un disturbio, los dispositivos de control (existentes dentro de diversos elementos) comienzan a actuar intentando restablecer un estado de operación de equilibrio, mas en algunas ocasiones esto sólo es posible mediante la intervención del operador del sistema, y en excepcionales ocasiones simplemente no es posible, ocurriendo lo que comúnmente se conoce como *apagón* o *blackout*. Un *blackout* es el mejor ejemplo de la inestabilidad en un SEE.

El problema de estabilidad de SEE puede ser clasificado según los criterios de la principal variable [de estado] observada, el tipo del disturbio, los dispositivos de control y respectivos periodos de tiempo que deben ser asegurados para una condición operación de equilibrio. Cuando la variable observada es una variable llamada *ángulo del rotor* [del generador], dependiendo de si el disturbio es pequeño o grande, la estabilidad puede ser clasificada como estabilidad permanente o transitoria respectivamente.

2.6.2. MODELOS UTILIZADOS

Los análisis de estabilidad [del ángulo del rotor] consideran el comportamiento del SEE ante disturbios que se manifiestan en tiempos del orden de milisegundos a segundos (tal y como fue mostrado en la Figura 2.14). Para este tipo de disturbios muchos de los modelos correspondientes a los elementos que componen un SEE presentados para los estudios de flujo de potencia continúan válidos. Por ejemplo, las líneas de transmisión y los transformadores mantienen sus modelos; ya las cargas y los generadores necesitan de otros modelos, las primeras porque en muchas ocasiones su representación por una impedancia (o admitancia) constante proporciona resultados más reales, y los segundos porque necesitan de una representación adecuada de sus controladores internos.

Generador

Un generador está compuesto principalmente por una máquina síncrona y diversos controladores. La máquina síncrona es la responsable por el proceso de conversión de energía (de mecánica para eléctrica por ejemplo). Un generador operando adecuadamente en un SEE cumple con varios requisitos (por ejemplo: ante disturbios conseguir mantener constantes la frecuencia de la onda de tensión generada y el módulo de tensión entre sus terminales), los cuales son conseguidos gracias a la actuación de sus controladores, tal y como es mostrado en la Figura 2.17

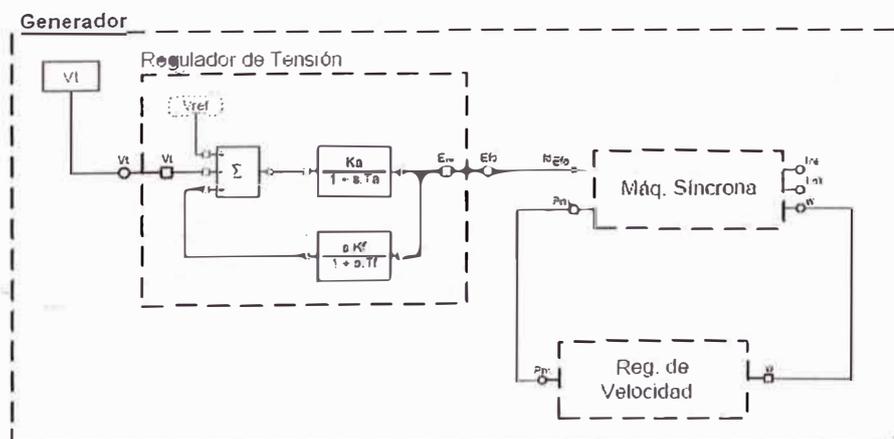


Figura 2.17: Componentes del Generador.

La máquina síncrona puede ser utilizada como motor o generador. Al trabajar como motor la máquina convierte la energía eléctrica como mecánica. Al trabajar como generador AC es impulsada por una turbina y convierte la energía mecánica en eléctrica. Una representación de una máquina síncrona es mostrada en la Figura 2.18. A pesar que explicar los parámetros y detalles del funcionamiento de una máquina escapa de los objetivos del presente informe es importante realizar algunos comentarios al respecto. Una máquina síncrona está compuesta por dos estructuras ferromagnéticas principales: una estacionaria llamada de estator y otra girante llamada de rotor. En un generador AC el rotor es impulsado por una fuente de energía mecánica que generalmente es una turbina de vapor o hidráulica. El rotor posee un devanado por el cual circula una corriente DC, la cual sumada al movimiento del rotor produce campos magnéticos que inducen una tensión en los devanados del estator. Estas tensiones inducidas poseen una forma sinusoidal ($A\text{Sen}(\omega t)$) con una frecuencia (f) que es dependiente de la velocidad (ω) a la que gira la turbina. Este proceso de conversión puede ser modelado como un circuito AC. Existen varios modelos para la máquina síncrona, los cuales varían dependiendo del nivel de detalle deseado. Un ejemplo de los datos necesarios para uno de estos modelos puede ser encontrado en el Apéndice A.

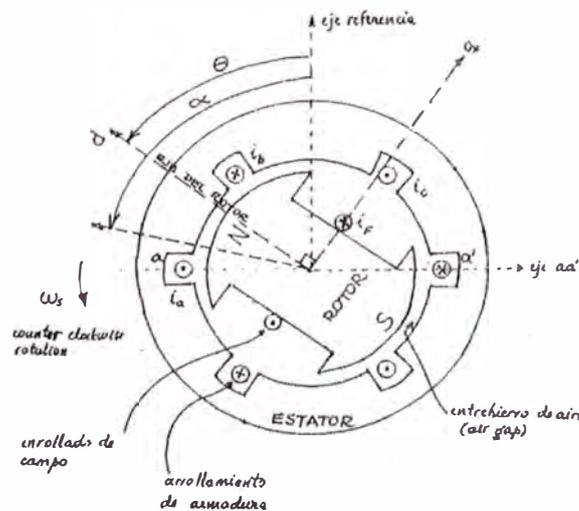


Figura 2.18: Máquina Síncrona [PAUC98].

Entre los principales dispositivos de control existentes dentro de un generador tenemos: el regulador de velocidad o *Governator* (GOV), el regulador de tensión o *Automatic Voltage Regulator* (AVR) y el estabilizador de potencia o *Power System Stabilizer* (PSS). Estos dispositivos son diseñados utilizando la ingeniería de control lineal por lo que su representación es dada utilizando diagramas de bloques.

Un GOV tiene como finalidad controlar la frecuencia en los terminales del generador. La frecuencia es fuertemente dependiente de la velocidad de la turbina. Entonces, caso exista una diferencia entre la frecuencia y un valor de frecuencia de referencia, el GOV modifica la cantidad de agua o vapor que actúan sobre la turbina con la finalidad de cerrar esta diferencia. Este controlador generalmente no es incluido en los estudios de estabilidad del ángulo del rotor debido a que sólo actúa después de varios segundos de haber percibido la diferencia de frecuencias.

Un AVR tiene como finalidad controlar el módulo de la tensión en los terminales del generador. La tensión terminal es fuertemente dependiente de la corriente en el devanado del rotor (corriente de campo). Entonces, caso exista una diferencia entre la tensión Terminal y un valor de tensión de referencia, el AVR modifica la corriente de campo con la finalidad de cerrar esta diferencia. En muchas ocasiones el AVR no consigue que esta diferencia llegue a ser cero, o en todo caso lo consigue en un tiempo relativamente grande (del orden de segundos), producto de amortecimientos pequeños en las respuestas obtenidas. Entonces se hace necesario la inclusión de controladores adicionales.

Un PSS tiene como finalidad mejorar los amortecimientos en las respuestas de tensión obtenidas producto de la actuación del AVR.

En nuestro país todos los generadores poseen máquina síncrona, GOV y AVR, y sólo algunos poseen también PSS. La actuación conjunta de estos controladores es capaz de mejorar la estabilidad en un SEE.

2.6.3. FORMULACIÓN MATEMÁTICA

El análisis de estabilidad [del ángulo del rotor] es formulado aplicando las dos leyes de Kirchhoff al circuito resultante del reemplazo de los elementos eléctricos por sus respectivos modelos. Sobre condiciones de operación normales todas las máquinas funcionan con su rotor girando a la misma velocidad [angular] síncrona ω . Cuando ocurre un disturbio existe un cambio súbito en la potencia eléctrica requerida en los terminales de cada uno de los generadores, lo que origina un cambio en sus velocidades síncrona de acuerdo a la ecuación (2.15):

$$\frac{2H_i}{\omega_s} \frac{d^2 \delta_i}{dt^2} + D_i \frac{d\delta_i}{dt} = P_{mi} - P_{ei} \quad i = 1, \dots, n_g \quad (2.15)$$
$$\dot{\delta}_i = \omega_i - \omega_s$$

Donde:

δ_i	Ángulo del rotor del generador i.
ω_i	Velocidad angular del generador i.
ω_s	Velocidad angular (del sistema) síncrona.
H_i	Constante de Inercia del generador i.
D_i	Constante de amortecimiento del generador i.
n_g	Número de generadores del sistema.
P_{mi}	Potencia mecánica en la turbina del generador i en p.u.
P_{ei}	Potencia eléctrica generada en el generador i en p.u.

El cálculo de la potencia eléctrica depende del modelo de la máquina que se esté utilizando. Los modelos de la máquina varían dependiendo del detalle del estudio que se pretenda realizar. Para fines didácticos utilizaremos a seguir un modelo que represente a las máquinas síncronas por una fuente de tensión sinusoidal constante E_j atrás de una reactancia transitoria en el eje d de la máquina, despreciándose la actuación de los controladores (modelo clásico). Las cargas serán consideradas como impedancias constantes de modo que se consigan impedancias equivalentes entre las barras a las que están conectadas las fuentes de tensión de los generadores. Esto es último es conocido como "reducción a los nodos internos del generador".

Con estas consideraciones la P_{ei} es dada por la ecuación (2.16):

$$P_{ei} = E_i^2 G_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^{n_g} (C_{ij} \text{sen} \delta_{ij} + D_{ij} \text{cos} \delta_{ij}) \quad (2.16)$$

De la ecuación (2.16), C_{ij} y D_{ij} están relacionados con las tensiones en barra y con los elementos de la matriz admitancia de la barra reducida a los nodos internos de los generadores, en la siguiente forma:

$$C_{ij} = E_i E_j B_{ij} \quad ; \quad D_{ij} = E_i E_j G_{ij} \quad (2.17)$$

Sustituyendo $\frac{2H_i}{\omega_s} \equiv M_i$ e $P_i \equiv P_{mi} - E_i^2 G_{ii}$, en la ecuación (2.15), y reordenando se tiene:

$$\begin{aligned} \dot{\delta}_i &= \omega_i - \omega_s \\ \dot{\omega}_i &= \frac{1}{M_i} (P_i - P_{ei}(\delta_1, \dots, \delta_m) - D_i(\omega_i - \omega_s)) \quad i = 1, \dots, n_g \end{aligned} \quad (2.18)$$

El conjunto de ecuaciones formada por (2.18) y (2.16) representa las ecuaciones de estado para un sistema multi-máquina considerando un modelo clásico de los generadores. La consideración de modelos más detallados e inclusión de los controladores sigue el mismo procedimiento que el descrito recientemente y produce similarmente un sistema de ecuaciones algebro diferenciales no lineales del tipo:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.19)$$

En que \mathbf{u} es un vector de entrada que generalmente consiste en las tensiones de la barra, de campo, de estator de los generadores y la potencia mecánica de entrada de cada máquina en el sistema. \mathbf{x} es el vector de variables de estado y contiene variables asociadas a las máquinas síncronas más controladores y algunos otros dispositivos dinámicos de la red eléctrica.

2.6.4. ESTABILIDAD TRANSITORIA

El conjunto de ecuaciones (2.19) es un sistema de ecuaciones álgebra diferenciales representativo del análisis de estabilidad transitoria [del ángulo del rotor] que puede ser resuelto. El término “transitoria” es adicionado para especificar que este sistema de ecuaciones será resuelto para disturbios grandes. La caída de una línea es un ejemplo de un disturbio grande denominado “corto-circuito”. Ante la presencia de un disturbio grande generalmente se hace necesaria la intervención del operador para evitar el colapso del sistema. Es así que se definen dos periodos de tiempo, el primero correspondiente al tiempo en que se encuentra presente el disturbio (estado en falla, generalmente del orden de 0 a 1 s.), y el segundo en el que alguna acción ocasionó la desaparición del disturbio (estado pos-falla, generalmente del orden de 8 s.). Un análisis de estabilidad transitoria resuelve el sistema de ecuaciones (2.19) en los dos periodos de tiempos.

Los métodos que resuelven este sistema de ecuaciones pueden ser divididos en tres grandes grupos [PAUC98]: métodos indirectos, métodos directos y métodos híbridos. De entre estos métodos los métodos indirectos son los que más se han destacado, siendo los más utilizados dentro de los softwares comerciales.

Los métodos indirectos están basados en técnicas de simulación numérica, es decir, utilizan técnicas de integración numérica. Entre las principales técnicas podemos mencionar la técnica trapezoidal y la técnica de Runge-Kutta de cuarto orden. Estas técnicas poseen un paso de integración constante y se muestran eficientes para simulaciones de estabilidad transitoria (del orden de 10 s.); mas se tornan inadecuados para simulaciones de otro tipo de estabilidades (del orden de horas); por lo que actualmente se están propiciando investigaciones en métodos con paso de integración variable. Estos métodos son llamados de indirectos porque la estabilidad es evaluada con la interpretación de los resultados generalmente por la experiencia del operador.

2.6.5. ESTABILIDAD PERMANENTE

Cuando los disturbios son pequeños el sistema de ecuaciones algebrao diferencial no lineal presentado en la ecuación (2.19) puede ser tratado como un sistema de ecuaciones algebrao diferencial lineal de forma que los resultados [de estabilidad] continúen válidos. El encendido de un foco en nuestros hogares es un ejemplo de un disturbio pequeño. En estos casos se dice que ese está trabajando con un análisis de estabilidad permanente [del ángulo del rotor].

La linealización de un SEE produce un sistema de ecuaciones como el presentado anteriormente en la ecuación (2.8) en el que las variables son deltas (pequeñas variaciones) producto de “cualquier” disturbio pequeño, tal y como es presentado en el sistema de ecuaciones (2.20):

$$\begin{aligned}\frac{d}{dt}(\Delta x) &= A\Delta x(t) + B\Delta u(t) \\ \Delta y(t) &= C\Delta x(t) + D\Delta u(t)\end{aligned}\quad \dots (2.20)$$

Donde:

Δx	Vector de las variables de estado
Δu	Vector de las variables de control
Y	Vector de las variables medidas o controladas
A	Matriz de estados
B	Matriz de entrada.
C	Matriz de salida
D	Matriz de transmisión directa.

Entre los métodos que analizan la estabilidad en este sistema de ecuaciones se destacan los basados en análisis de autovalores. La gran ventaja de utilizar estos métodos es que la estabilidad del sistema puede ser determinada sin necesidad de resolver el sistema de ecuaciones, y sólo analizando los autovalores de la matriz de estados.

CAPÍTULO III

BASE COMPUTACIONAL DE SOFTWARE PARA SEE

3.1. INTRODUCCIÓN

La simulación digital de los SEE es vital en los estudios de ingeniería [SELV04] envolviendo desde cálculos simples como los concernientes a un “flujo de potencia”, hasta cálculos complejos como los concernientes al “planeamiento de la expansión”. El incremento constante de la demanda, la tendencia cada vez mayor de interconectar SEE existentes, el desarrollo de nuevos equipos, etc., ocasionan que los estudios sean cada vez más complejos, con un cambio cada vez más constante en los requerimientos, originando que los simuladores utilizados sean constantemente actualizados.

La mayoría de los simuladores fueron inventados en las décadas del 60 y 70, utilizando un paradigma procedural implementados principalmente en Fortran; y prudentemente son utilizados hasta la actualidad al poseer algoritmos sumamente robustos y confiables. Sin embargo, dada la antigüedad de sus metodologías de desarrollo, su mantenimiento se torna cada vez más difícil. La aparición del POO y arquitecturas de sistemas distribuidos abrieron una posibilidad para lidiar con la complejidad inherente del desarrollo y mantenimiento de los diversos tipos de *softwares*, incluyendo a los simuladores digitales de SEE. En este capítulo se presenta una propuesta de una base computacional para SEE que soporte un uso integrado de las metodologías de varios de estos simuladores, utilizando los conceptos de Ingeniería de Software recientemente comentados.

3.2. ESTADO ACTUAL DE LOS SIMULADORES DE SEE

A partir del trabajo pionero de Ward en el año de 1956, en que presenta una implementación de un simulador digital para análisis del flujo de potencia de SEE [WARD56], los simuladores digitales han ido creciendo, logrando un alto nivel de sofisticación y madurez, convirtiéndose algunos en *softwares* comerciales. En la Tabla 3.1 se presenta un resumen de los principales *softwares* existentes para las simulaciones de interés del informe: flujo de potencia (FP), estabilidad permanente (EP) y transitoria (ET).

Tabla 3.1: Principales simuladores de SEE existentes.

Identificación	Empresa / Grupo Investigación				Carácter	Plataforma	Referencia
		FP	EP	ET			
ANAREDE	CEPEL	X			Comercial	Windows	CEPE99
ANATEM	CEPEL			X	Comercial	Windows	CEPE97
ASPEN Power Flow	ASPEN	X			Comercial	Windows	ASPE06
DSA Tools	Powertech	X	X	X	Comercial	Windows	POWE06
EDSA	EDSA	X		X	Comercial	Windows	EDSA06
EST	-	X	X	X	-	Matlab	VOUR04
Eurostag	Tractebel Engineering	X		X	Comercial	Windows	TRAC06
MathEMTP	-			X	-	Matlab	MAHS97
Matpower	-	X			Gratuito	Matlab	ZIMM97
PSAF - FLOW	CYME	X			Comercial	Windows	CYME06
PAT	-	X	X	X	Gratuito	Matlab	SCHO03
Power Station	ETAP	X		X	Comercial	Windows	ETAP06
PSAT	-	X	X	X	Gratuito	Matlab	MILA06
PSCAD/EMTDC	Manitoba HVDC	X		X	Comercial	Windows	MANI06
PST	-	X	X	X	Gratuito	Matlab	CHOW97
PSS/E	PTI Software Solution	X	X	X	Comercial	Windows	PTI06
SPS	-	X	X	X	Gratuito	Matlab	SYBI04
VST	-	X	X	X	Gratuito	Matlab	NWAN02
WinFlu	-	X			Comercial	Windows	PAJA06

Cada una de las simulaciones mencionadas (FP, EP, ET) constituye por sí sola una línea de investigación dentro de la comunidad científica eléctrica. Los avances logrados por ellos, casi siempre de carácter metodológico, son incluidos inicialmente en los *softwares* de carácter gratuito y son principalmente utilizados con fines educativos. Una vez que la metodología consigue un nivel adecuado de aceptación, es incluida en los *softwares* comerciales.

La inclusión de nuevas metodologías en los *softwares* comerciales es realizado generalmente por un (s) integrante de la comunidad científica, por lo que tanto los *softwares* gratuitos, como los comerciales, son sumamente eficientes y confiables; mas comparten una serie de características que necesitan ser revisadas ante los continuos cambios en la regulación actual de los SEE. Entre estas características podemos comentar [AGOS02]:

- La parte del *software* responsable por la simulación propiamente dicha, es desarrollada siguiendo métodos que priorizan la funcionalidad de la simulación de interés (modelado y programación orientada a funciones). Como consecuencia, las estructuras de datos no poseen funcionalidades claras, existiendo sólo para el almacenamiento de datos de los SEE en tiempo de ejecución del *software*. Esto dificulta el soporte de estas estructuras para la inclusión de nuevas simulaciones.
- Los módulos del *software* generalmente son desarrollados por diferentes especialistas, cada uno con una metodología propia de trabajo. Esto dificulta la integración de los diversos módulos, incrementando las inversiones correspondientes al mantenimiento.
- Las modificaciones en una porción específica de código fuente, producen muchas veces efectos colaterales en otros módulos del *software*. Esto origina problemas adicionales a los que motivaron inicialmente el proceso de mantenimiento, obligando muchas veces la re-escritura de grandes porciones de código fuente.

La comunidad científica, conciente de las características recientemente mencionadas; hace algunos años viene proponiendo posibles soluciones. Entre las propuestas, las que más se destacan son las concernientes a la utilización de tecnologías relativamente nuevas (POO, UML, etc.) en el desarrollo de simuladores de SEE. Una revisión bibliográfica al respecto es presentada a continuación.

3.3. REVISIÓN BIBLIOGRÁFICA.

En la literatura especializada, el número de publicaciones que utilicen POO, UML, tecnología de componentes y arquitecturas de sistemas distribuidos, para el desarrollo de simuladores (*softwares*) de SEE es aún pequeño, mas existe una tendencia de alteración significativa de este escenario. Estas publicaciones pueden ser divididas en dos generaciones, las cuales son comentadas a continuación.

3.3.1. PRIMERA GENERACIÓN

La primera generación comenzó a inicios de la década del 90 y buscaba evaluar las potencialidades de la utilización del POO para la realización de una aplicación (simulación digital) en particular. Los trabajos más resaltantes pertenecientes a esta generación son detallados a continuación.

En [NEYE90] se evaluó el potencial de aplicar el POO al problema del flujo de potencia. Se utilizó una estructura para representar tanto la configuración de la red eléctrica, introduciendo el concepto de elementos físicos y conceptuales, así como otra para el problema del flujo de potencia. Los elementos físicos son estáticos, y los conceptuales dinámicos, es decir, cambian sus respectivos estados con la finalidad de determinar la actual estructura topológica del sistema. Los resultados demostraban un desempeño inferior de dos a tres veces comparados a implementaciones equivalentes en Fortran.

En [FLIN92] se propone un modelo de datos que si fuese implementado en una base de datos orientada a objetos podría ser utilizado por diversas aplicaciones, tales como análisis armónicas, flujo de potencia, corto circuito y estabilidad transitoria. Si bien las aplicaciones no son implementadas, lo más relevante de esta publicación es que presenta por primera vez la necesidad de tener una base de datos única que proporcione soporte a la mayor cantidad posible aplicaciones.

En [FOLE93] se presenta una interfaz gráfica para SEE. En esta publicación se propone una estructura jerárquica para representar los diversos elementos gráficos. En [FOLE95] los autores mejoran su propuesta al unirla con los conceptos de elementos físicos y conceptuales presentados en [NEYE90]. La estructura computacional agrupa los dispositivos en subestaciones, imitando la estructura física real de un SEE, y su funcionalidad es comprobada con la implementación de un flujo de potencia.

En [HAVA94] se propone una nueva estructura jerárquica de clases para la representación de los elementos físicos de un SEE, y por primera vez una jerarquía para las simulaciones correspondientes a los SEE considerando operaciones de matrices dispersas. Son descritos problemas de desempeño computacional debido a la utilización de la filosofía purista del POO.

En [ZHOU96] se presenta un aplicativo de flujo de potencia utilizando una estructura orientada a objetos semejante a la propuesta en [HAVA94]. En esta publicación es destacado el buen desempeño computacional del POO. Aunque las razones no son claramente expuestas, son destacables los resultados presentados en la aplicación del POO en la manipulación de matrices dispersas y solución de sistemas lineales de gran dimensión.

En [GAIN96] se presenta un *software* para auxilio en la restauración de SEE. Los autores siguen los criterios presentados por la OMT [RUMB94], siendo el primer trabajo en que se utiliza una notación formal para representar la estructura orientada a objetos propuesta. Son presentados algunos de los diagramas de objetos, flujo de datos y flujo de estados.

En [PHIL96] se propone por primera vez una base computacional orientada a objetos para aplicaciones generales en SEE. La estructura está centrada sobre una base de datos orientada a objetos y sólo utiliza el flujo de carga como aplicativo ejemplo. Son descritos problemas de desempeño computacional al ser comparado con una implementación realizada en C.

En [ZHU97] se propone una base computacional orientada a objetos para sistemas de distribución. Los autores manifiestan que una vez que la estructura computacional es modelada utilizando como base los conceptos reales del sistema, ésta será capaz de soportar una gran cantidad de aplicativos, siendo el primer trabajo que manifiesta modelar el SEE independiente de cualquier aplicativo. La jerarquía presentada utiliza una clase base llamada *Device* (dispositivo o equipo), dividida en tres grandes grupos de elementos: *Shunt* (shunt), *Branch* (ramo) y *Protective* (protección). Todos los elementos son agrupados en clases que representan alimentadores o subestaciones. Este trabajo también muestra un diagrama de clases siguiendo la normativa de la OMT.

En [HAND98] se utiliza el POO para el problema de planeamiento de la transmisión en un ambiente competitivo. La jerarquía presentada utiliza una clase SEE compuesta por agregación de las clases *Load* (carga), *Transmission* (transmisión) y *Generator* (generador), la cual es basada en el número de nodos eléctricos al que los elementos están conectados. Esta jerarquía con pequeñas variantes fue la predominante en esos años. En este trabajo es presentado por primera vez un modelo para el entorno del SEE, llamado de *Market Model* (mercado), mas no son presentados resultados numéricos.

En [FUER98] se presenta un aplicativo orientado a objetos para el problema de flujo de potencia considerando por primera vez elementos de electrónica de potencia FACTs. En este trabajo se muestra también la utilidad de utilizar las librerías STL pertenecientes a C++ en la resolución de sistemas lineales dispersos de gran dimensión. El desempeño computacional presentado es comparable con las implementaciones en Fortran, atribuyendo como razón para estos resultados a las buenas prácticas de programación y no necesariamente a la utilización de la filosofía purista del POO.

En [MANZ99] es presentado un aplicativo para simulación dinámica de los SEE utilizando una estructura similar a la presentada en [HAND98]. En este trabajo son presentados técnicas eficientes para resolver sistemas lineales dispersos de gran dimensión, considerando por primera vez los elementos internos correspondientes al generador. El desempeño computacional obtenido con sistemas de gran dimensión es comparable a los obtenidos con un programa comercial desarrollado en Fortran.

En [ZHU99] se propone la utilización de patrones de diseño para el modelado de ciertos aspectos gráficos y de comportamiento de los elementos de los SEE. A pesar de no presentar ningún resultado numérico, los autores destacan la necesidad de desarrollar una base común orientada a objetos para soportar una amplia variedad de aplicativos de SEE en forma integrada, mas reconocen la dificultad de llegar a una estructura óptima.

En [BRAD99] se propone una estructura jerárquica para simulación de fallas en SEE utilizando los conceptos de elementos físicos y lógicos presentados en [NEYE90], en el que la red eléctrica puede ser vista en varios niveles de agregación. El SEE está formada por varias islas eléctricas, cada una de las cuales son compuestas por líneas y subestaciones, donde cada subestación está compuesta por varios equipos eléctricos. La notación formal utilizada para esta representación no fue presentada.

En [PAND00] [PAND01a] se propone una plataforma de clases para aplicaciones en SEE. Los autores enfocan su plataforma para problemas de régimen permanente (flujo de potencia, corto circuito, procesador de topología y estimación de estados), argumentando que dominios de aplicaciones diferentes requieren modelados diferentes. Un procesador de topología es utilizado para formar la red eléctrica (clase *Network*) que será posteriormente utilizada por los aplicativos. Los aplicativos utilizan una estructura genérica para gerenciar sistemas lineales dispersos de gran dimensión, la cual es presentada en detalle en [PAND01b].

3.3.2. SEGUNDA GENERACIÓN

A finales de la década del 90, los avances de las arquitecturas de *hardware* y consolidación de los compiladores basados en el POO hicieron que la preocupación con la aplicabilidad del POO adquiriese menor relevancia, dando origen a la segunda generación, donde el gran desafío pasó a ser el de encontrar una arquitectura que contenga principalmente una estructura de clases que represente adecuadamente al SEE de manera que sirva como base para una implementación integrada de la mayor cantidad de aplicativos posibles. Los trabajos más resaltantes pertenecientes a esta generación son detallados a continuación.

En [LU02] se presenta una propuesta de un *software* para SEE que agrupa y reutiliza varios aplicativos existentes implementados en lenguajes de programación diferentes (Fortran, C++, Java, etc.), mediante el uso de la tecnología de componentes. La arquitectura propuesta está basada en CORBA, implementándose un servicio por cada simulación requerida y por cada base de datos que necesite ser accesada. La notación formal para esta propuesta, así como resultados numéricos no son presentados.

En [AGOS02] se propone la utilización de patrones de diseño en la propuesta de una base computacional orientada a objetos para aplicaciones generales en SEE. Los autores utilizan una estructura jerárquica para los elementos físicos y otra para las simulaciones. La primera jerarquía utiliza los conceptos de elementos estructurales y elementos de composición. Los elementos estructurales son los elementos físicos que conforman la red eléctrica, los cuales se derivan de una clase base y son divididos en 3 grandes grupos: shunt, ramo y barra. Los elementos de composición son derivados otra clase base y representan el conjunto de elementos que sin pertenecer directamente a la red eléctrica componen un elemento físico. Las facilidades propiciadas por el uso de esta base computacional son comprobadas con la implementación del flujo de potencia y de la estabilidad transitoria.

En [PALM03] se presenta un modelo orientado a objetos para un análisis integrado de energía y transporte en los SEE. Dentro de los SEE son considerados no sólo el subsistema de potencia (red eléctrica), sino también los subsistemas de combustibles y de transporte. El subsistema de potencia es el mismo presentado en [HAND98], el cual ahora se relaciona con los otros dos subsistemas dentro de un entorno cliente-servidor implementado en Java.

En [ABDE04] se presenta la utilización de casos de uso en el desarrollo de componentes reutilizables de *software* para simulación de SEE. Los diversos elementos físicos son modelados mediante una jerarquía de clases entidad, mientras que las simulaciones (algoritmos de solución) son modelados como clases de control. Si bien la jerarquía de las clases entidad no es mostrado, muchos de los componentes diseñados originalmente para la solución del flujo de potencia, son reutilizados en el diseño del flujo de potencia trifásico.

En [SELV04] se presenta la aplicación del POO para tratamiento de sistemas lineales dispersos de gran dimensión. Los autores destacan las ventajas provenientes de este paradigma, principalmente el hecho de no requerir ninguna alteración en el código existente al incluir un nuevo aplicativo. La jerarquía presentada para el tratamiento de la conectividad está basada en dos clases básicas: *Node* (barra) y *Branch* (ramo), de las cuales son derivados los demás elementos que componen el SEE.

En [MEDI04] se presenta la aplicación del POO para el tratamiento de ecuaciones diferenciales ordinarias (EDO), para la simulación en el dominio del tiempo de los SEE. En esta propuesta cada ecuación es descompuesta a través de un conjunto de términos en un esquema de lista encadenada. Una clase conteniendo diversos métodos de integración numérica es presentada, mas no es discutido como es realizado el tratamiento dado a la diferenciación numérica de las ecuaciones.

En [LARS04] se presenta una herramienta educativa para simulación dinámica de SEE, implementada sobre una librería llamada ObjectStab, escrita en el lenguaje Modelica, un lenguaje de modelado de propósito general orientado a objetos. La principal ventaja de esta herramienta es su característica gratuita y de código abierto, lo que permite a los estudiantes (usuarios) realizar las modificaciones que consideren pertinentes.

En [MOHA04] se presenta la aplicación de la tecnología de componentes en el diseño del flujo de potencia. El diseño presentado es completamente reutilizado ante la adición de nuevos elementos (FACTS). La jerarquía de clases de los elementos físicos en el SEE está basada en el número de nodos a los cuales se encuentran conectados, siendo presentada utilizando la notación propuesta en el UML.

En [RODR05] se presenta la aplicación de la tecnología de componentes en la propuesta de una base computacional para aplicaciones generales en SEE. La base está compuesta por 5 componentes dentro de un entorno distribuido: un repositorio de datos, una interfaz gráfica, una jerarquía de clases para los elementos físicos, una jerarquía de clases para los elementos conceptuales, y uno responsable por la comunicación (CORBA). Los aplicativos implementados son el flujo de potencia, la estabilidad transitoria y el flujo de potencia óptimo, los cuales trabajando juntos resuelven una problemática mayor: el análisis de seguridad dinámica.

En [MANZ05] se presenta una base computacional orientada a objetos para aplicaciones generales en SEE. La descripción de la topología de la red es caracterizada para representar los elementos físicos y conceptuales. Un procesador de topología es utilizado para formar la red eléctrica que será posteriormente utilizada por los aplicativos. Uno de los grandes aportes de este trabajo es el grado de generalización para los elementos físicos, que permite que un nuevo elemento sea adicionado al sistema y asimilado automáticamente sin necesidad de efectuar alteraciones en el código fuente.

3.4. REQUERIMIENTOS DEL SISTEMA DE SOFTWARE

El funcionamiento general de un SEE fue descrito en el capítulo 2. A partir de esta descripción se identifican los requerimientos del sistema de *software* (simulador). Estos requerimientos son representados mediante los diagramas de caso de uso, las especificaciones de los casos de uso y las especificaciones de las interfaces del sistema.

En la Figura 3.1 se muestra un Diagrama con los principales Casos de Uso. Se puede observar que sólo se tiene un actor genérico (Ingeniero de Sistemas de Potencia) y 7 Casos de Uso (correspondientes a la creación / edición del SEE, y a la realización de simulaciones en el SEE).

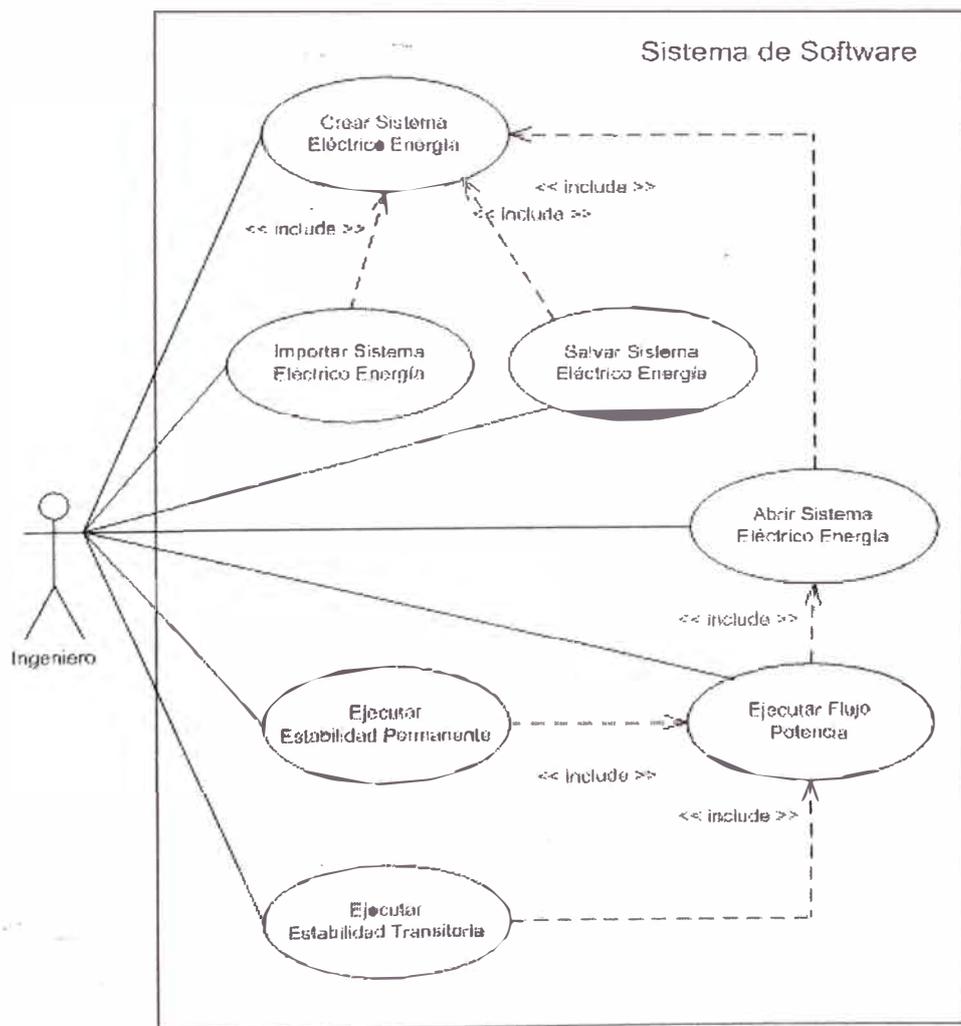


Figura 3.1: Casos de Uso del Sistema de Software

En la Figura 3.2 se muestra la especificación de uno de los Casos de Uso más representativo: “Ejecutar Flujo de Potencia”. El lenguaje utilizado en esta especificación es entendible por un Ingeniero de Sistemas de Potencia, usuario final del sistema de *software*.

CASO DE USO:
Ejecutar Flujo de Potencia
<p>RESUMEN: El Ingeniero calcula el estado actual (tensiones complejas en los nodos y flujos de potencia en las ramas) del SEE.</p> <p>ACTOR:</p> <ul style="list-style-type: none"> • Ingeniero de Sistemas de Potencia. <p>PRE-CONDICIONES:</p> <ul style="list-style-type: none"> • Existe un SEE activo (abierto). <p>DESCRIPCIÓN:</p> <ul style="list-style-type: none"> • El Caso de Uso comienza cuando el Ingeniero solicita el estado actual del SEE activo. • Se cargan los modelos de los elementos. Si existen todos los modelos en la librería de modelos los elementos del SEE activo calculan sus estados iniciales de acuerdo a los valores en sus correspondientes modelos (Procesador de Topología). [Excepción: No existe modelo de algún elemento]. • El SEE es representado por un modelo matemático (sistema de ecuaciones algebraicas no lineales) producto de los modelos de los elementos. • Se resuelve el modelo matemático. [Caso de Uso: Solución de Sistemas Lineales]. [Excepción: No existe solución para el modelo matemático.] • El caso de uso termina cuando la solución del modelo matemático actualiza los estados de los elementos del SEE. <p>POST-CONDICIONES:</p> <ul style="list-style-type: none"> • El estado de los elementos del SEE activo ha sido actualizado. <p>EXCEPCIONES:</p> <ul style="list-style-type: none"> • Excepción: No existe modelo de algún elemento. - El caso de uso termina. • Excepción: No existe solución para el modelo matemático. - El caso de uso termina.

Figura 3.2: Especificación del Caso de Uso: Ejecutar Flujo de Potencia.

En la Figura 3.3 se muestra la especificación de la interfaz principal del sistema de *software*. La interfaz principal permite al usuario abrir o crear un SEE (menú Archivo). Permite también la edición del SEE, ya sea adicionando elementos (barra de elementos) o modificando los datos de los elementos existentes (cajas de dialogo por elemento). La interfaz principal también permite al usuario realizar las simulaciones requeridas (menú Ejecutar), mostrando gráfica y amigablemente los resultados de interés obtenidos.

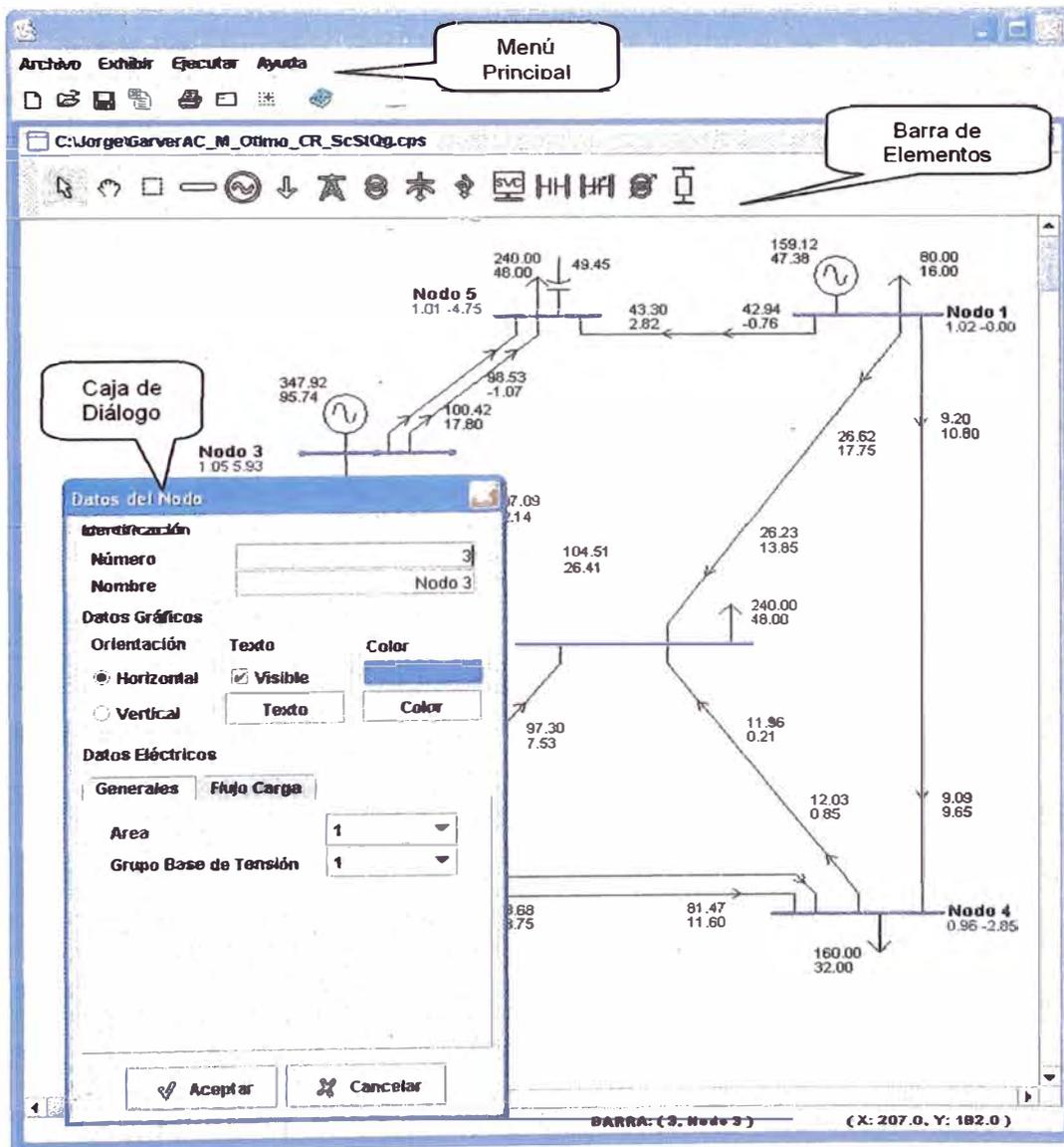


Figura 3.3: Especificación de la Interfaz Principal.

3.5. ANÁLISIS ORIENTADO A OBJETOS

En esta etapa se refinan y estructuran los requerimientos siguiendo los siguientes pasos: definición de la arquitectura del modelo de análisis, análisis de casos de uso, análisis de clases y análisis de paquetes. Un detalle de los pasos seguidos es mostrado a continuación.

3.5.1. ARQUITECTURA DEL MODELO DE ANÁLISIS

La arquitectura del modelo de análisis está definida por la identificación de clases y paquetes obvios, así como por los requerimientos especiales. En la Tabla 3.2 se muestran los principales paquetes identificados, así como sus respectivas clases de análisis.

Tabla 3.2: Lista de Paquetes y Clases de Análisis principales.

Paquete		Clase		
1er. Nivel	2do. Nivel	Nombre	Tipo	Descripción
Datos SEE	SEE	SEE	Entidad	Sistema Eléctrico Energía
		SUBS	Entidad	Elemento físico Subestación
		L_T_	Entidad	Elemento físico Línea de Transmisión
		NODO	Entidad	Elemento físico Nodo eléctrico.
		BARR	Entidad	Elemento Barra eléctrica.
		GERD	Entidad	Elemento físico Generador
		CARG	Entidad	Elemento físico Carga.
		TRAF	Entidad	Elemento físico Transformador.
	LOGI	Entidad	Elemento Lógico.	
	Visualización SEE	WPRIN	Interfaz	Interfaz principal del sistema de software.
		WFLUP	Interfaz	Interfaz para simular Flujo de Carga.
		WESTP	Interfaz	Interfaz para simular Estabilidad Permanente.
	Entrada Salida	WESTT	Interfaz	Interfaz para simular Estabilidad Transitoria.
		ENTR	Entidad	Responsable por la creación, abertura e importación del SEE.
	Análisis Eléctricas	Aplicación	SALID	Entidad
FLUP			Control	Flujo de Potencia
ESTP			Control	Estabilidad Permanente
Matemática		ESTT	Control	Estabilidad Transitoria
		LSOL	Control	Solucionador de Sistemas Lineales.
		VECTOR	Control	Estructura matemática Vector.
		MATRIZ	Control	Estructura matemática Matriz.

En la Figura 3.4 son mostradas las dependencias existentes entre los paquetes identificados. Se puede observar que este modelo está compuesto por 2 paquetes principales (primer nivel). El primer paquete, llamado de "Datos SEE", es responsable principalmente por proporcionar una jerarquía de clases que represente (imite) la estructura física real de un SEE. El segundo paquete, llamado de "Análisis Eléctricas", es responsable principalmente por proporcionar una jerarquía de clases que represente las simulaciones que se pueden realizar sobre el SEE. Este modelo de análisis simplificado ya fue anteriormente presentado en [RODR05].

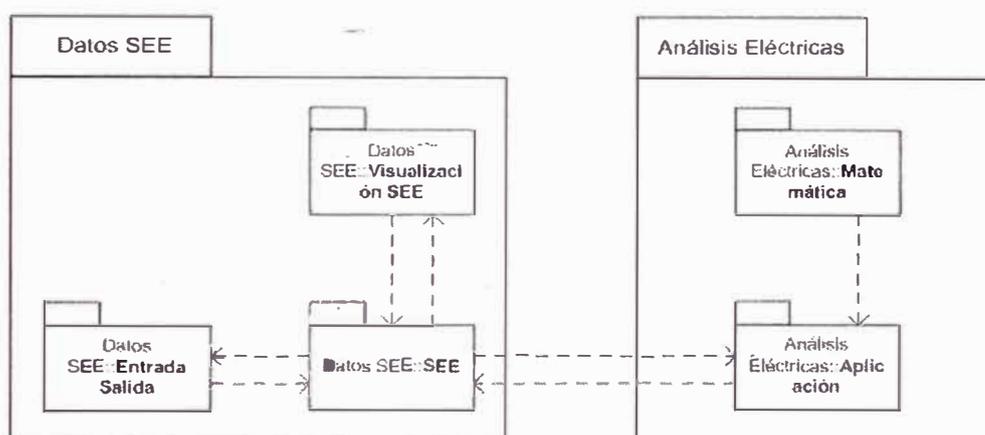


Figura 3.4: Paquetes principales del modelo de análisis.

Paquete Datos SEE

Este paquete está compuesto por 3 paquetes. El primer paquete, llamado de "Entrada Salida", contiene principalmente una jerarquía de clases para el almacenamiento físico (en texto, base de datos, etc.) de los datos. El segundo paquete, llamado de "Visualización SEE", contiene una jerarquía de clases para la visualización y gerenciamiento gráfico de los datos. El tercer paquete, llamado de "SEE", contiene principalmente una jerarquía de clases para la representación de las diversas características físicas de la red eléctrica correspondiente al SEE. En este paquete están considerados tanto los elementos físicos como conceptuales propios de un SEE. Este último paquete es el más importante dentro del paquete "Datos SEE", y es una de las razones de la investigación reportada en el presente documento.

Paquete Análisis Eléctricas

Este paquete está compuesto por 2 paquetes. El primer paquete, llamado de “Matemática”, contiene una jerarquía de clases para la resolución de sistemas lineales. El segundo paquete, llamado de “Aplicación”, contiene una jerarquía de clases que representan las simulaciones que se pueden realizar sobre un SEE. Este paquete necesita de los datos contenidos en el paquete “SEE” para formar el modelo matemático; y necesita también del paquete “Matemática” para solucionar el modelo matemático. El paquete “Aplicación” es el más importante dentro del paquete “Análisis Eléctricas”, y es una de las razones de la investigación reportada en el presente documento.

Dependencia entre los paquetes

Entre las principales relaciones de dependencia visualizadas en la Figura 3.4, podemos mencionar:

- El paquete Aplicación depende del paquete SEE para formar el modelo matemático representativo de la simulación.
- El paquete Aplicación depende del paquete Matemática para resolver el modelo matemático representativo de la simulación.
- El paquete SEE depende del paquete Aplicación para actualizar su estado producto de una simulación.

Requerimientos especiales

Entre los requerimientos especiales identificados se pueden mencionar:

- El objeto correspondiente al SEE debe presentar persistencia.
- Diversas simulaciones podrían ser realizadas simultáneamente (conurrencia) siempre y cuando sólo exista un resultado consolidado.

3.5.2. ANÁLISIS DE CASOS DE USO

El análisis de casos de uso se realiza mostrando como colaboran los objetos de las clases de análisis requeridas para cumplir la funcionalidad del caso de uso. Por motivos didácticos en la Figura 3.5 sólo se muestra el diagrama de colaboración del caso de uso “Ejecutar Flujo de Potencia”, caso de uso presentado en la Figura 3.1 y especificado en la Figura 3.2.

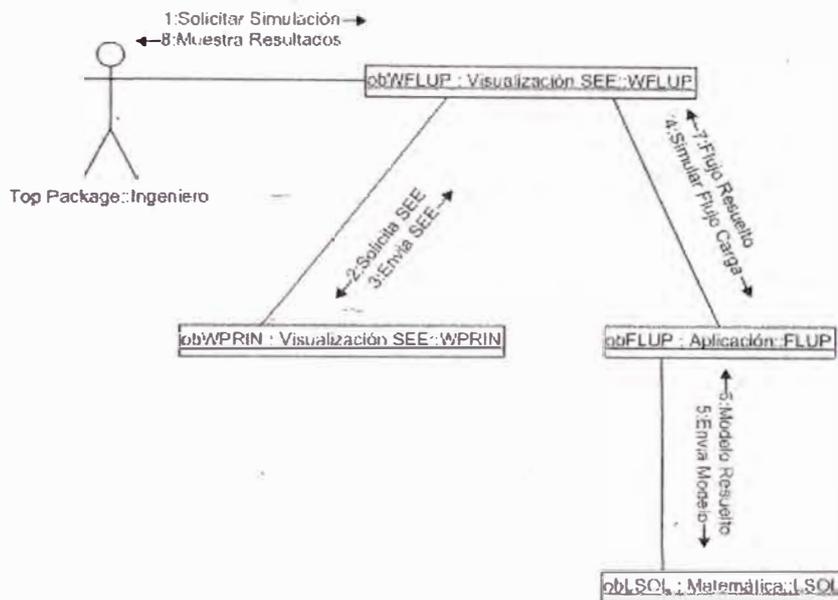


Figura 3.5: Diagrama de colaboración del caso de uso: Ejecutar Flujo Potencia.

En la Figura 3.5 se pueden apreciar las clases de análisis requeridas, y como estas interactúan, para que el caso de uso “Ejecutar Flujo de Potencia” funcione. Se puede observar que el primer mensaje es el llamado del método “Solicitar Simulación” por parte del Ingeniero al objeto obWFLUP (interfaz del flujo de potencia). El segundo mensaje es el llamado del método “Solicita SEE” por parte del objeto obWFLUP al objeto obWPRIN (interfaz principal). Este mensaje es debido a que el objeto obWFLUP necesita poder acceder a los datos correspondientes al SEE en estudio. El tercer mensaje es el llamado del método “Envía SEE” por parte del objeto obWPRIN al objeto obWFLUP. Se puede observar que el intercambio de mensajes continúa hasta llegar al octavo mensaje, en el que objeto obWFLUP muestra los resultados recibidos del objeto obFLUP al ingeniero.

3.5.3. ANÁLISIS DE CLASES Y PAQUETES

El análisis de clases se realiza para identificar las responsabilidades de las clases, los atributos de las clases y las relaciones entre las clases, basados en el rol que juegan en el funcionamiento de los casos de uso. Los diagramas de colaboración realizados para cada escenario de cada uno de los casos de uso presentados permite la creación de un único diagrama de clases de análisis consolidado. En la Figura 3.6 se muestra el diagrama de clases de análisis consolidado, resultantes del análisis de los casos de uso: “Ejecutar Flujo de Potencia”, “Ejecutar Estabilidad Permanente” y “Ejecutar Estabilidad Transitoria”.

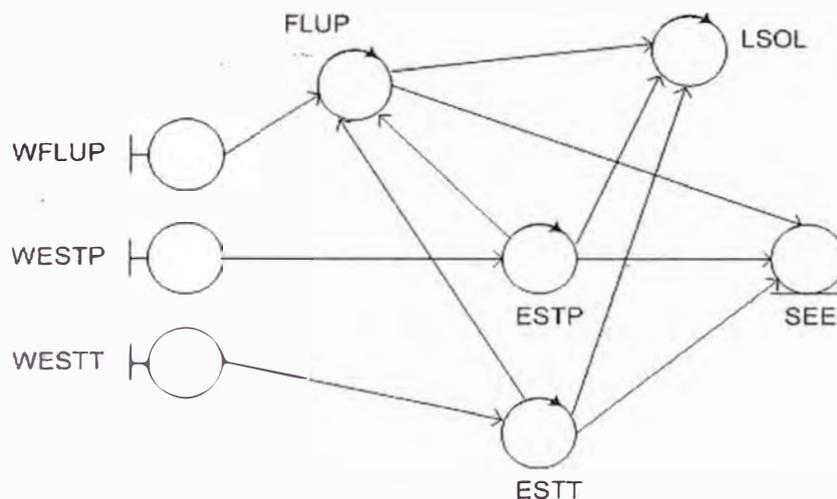


Figura 3.6: Diagrama de clases del Modelo de Análisis.

En la Figura 3.6 se pueden apreciar la existencia de una clase interfaz y una clase control por cada simulación requerida. Estas clases se relacionan con una única clase entidad que representa al SEE. Esta característica es sumamente importante, porque permite que todas las simulaciones posibles (inclusive las que futuramente puedan ser adicionadas) se realicen considerando y operando sobre el mismo SEE.

El análisis de paquetes se realiza para refinar el diagrama de paquetes obtenido inicialmente. En la etapa de análisis el diagrama de paquetes se mantiene tal y como fue mostrado en la Figura 3.4.

3.6. DISEÑO ORIENTADO A OBJETOS

En esta etapa se describe la manera en que se concretarán físicamente los casos de uso a partir de los requerimientos funcionales y no funcionales, siguiendo los siguientes pasos: definición de la arquitectura del diseño, diseño de casos de uso, diseño de clases y diseño de subsistemas. Un detalle de los principales pasos seguidos es mostrado a continuación.

3.6.1. ARQUITECTURA DEL DISEÑO

La arquitectura del diseño está definida principalmente por la identificación de nodos y configuración de la red, por la identificación de subsistemas y sus interfaces, así como por la identificación de clases arquitecturalmente significativas.

Identificación de Nodos y configuración de la red

La configuración de red adoptada para el presente modelo es el de “cliente/servidor”, en el que se distinguen 3 capas: visualización, lógica del negocio y repositorio de datos, donde las capas de visualización y de datos se encuentran de lado del cliente y la capa de negocios del lado del servidor.

Cada una de las posibles simulaciones (flujo de potencia, estabilidad permanente y estabilidad transitoria) es modelada como un servicio. Este tipo de modelado posibilita la existencia de varios servicios en la red correspondientes a una única simulación. Ante la necesidad de conocer los flujos de potencia en el SEE para “ n ” posibles alteraciones en una única configuración de la red, el enfoque descrito permite realizar las “ n ” simulaciones concurrentemente (cada simulación en un servidor diferente), con una posterior consolidación de resultados. Queda claro que la existencia de varios servidores en la red posibilita la interacción o trabajo simultáneo entre ellos en la resolución de problemáticas del tipo SIMD (*Single Instruction Multiple Data*) o MIMD (*Multiple Instruction Multiple Data*), problemáticas cada vez más comunes dentro de los análisis de SEE.

Identificación de subsistemas e interfaces

Los principales subsistemas identificados son los correspondientes a los paquetes principales presentados en la Figura 3.4: Datos SEE y Análisis Eléctricas.

Se debe tener en cuenta también que en las diversas organizaciones (universidades, empresas, etc.) en algunas ocasiones ya existen desarrollos de *softwares* que realizan simulaciones sobre un SEE. Muchos de estos *softwares* existentes son robustos y su funcionalidad ha sido históricamente comprobada mas se encuentran implementadas en lenguajes de programación diversos, predominantemente C++ y Fortran. Caso se posean los códigos fuentes, estos *softwares* pueden ser encapsulados como un subsistema y disponibilizados como un servicio adicional en la red. La incorporación se realizaría con la elección de un *middleware* que soporte estos lenguajes de programación, como por ejemplo el CORBA. Esto permitiría en un futuro agregar nuevos tipo de simulaciones (desarrollados por otros autores) en nuestro sistema de *software*, tal y como es mostrado en la Figura 3.7.

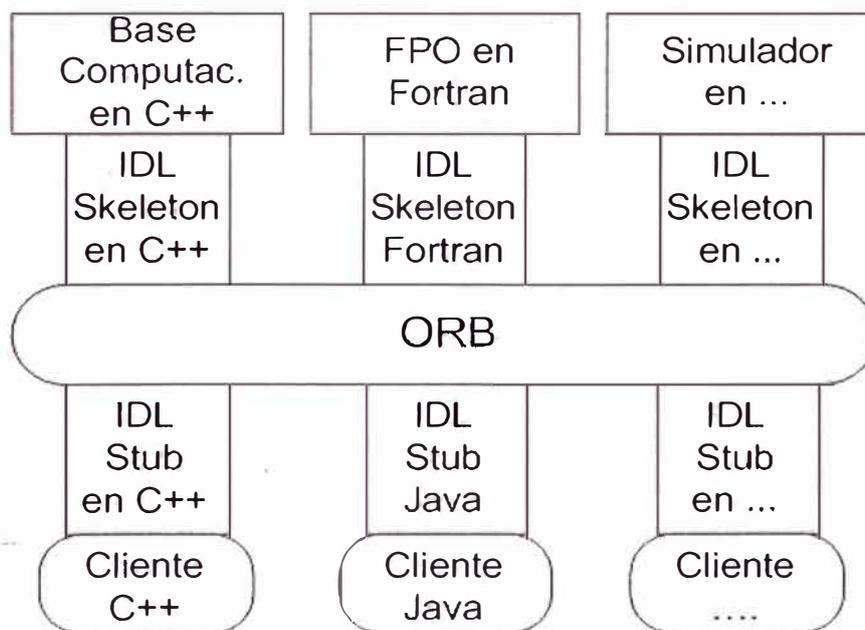


Figura 3.7: Uso de CORBA en la incorporación de softwares antiguos.

Identificación de clases arquitecturalmente significativas

La mayoría de las clases de diseño son identificadas en la etapa de diseño de clases, mas ya se puede vislumbrar que será necesario la inclusión de clases o estructuras para la utilización del CORBA como *middleware*. Estas clases pueden ser empaquetadas en un paquete denominado “Comunicación”, resultando un diagrama de paquetes como el mostrado en la Figura 3.8.

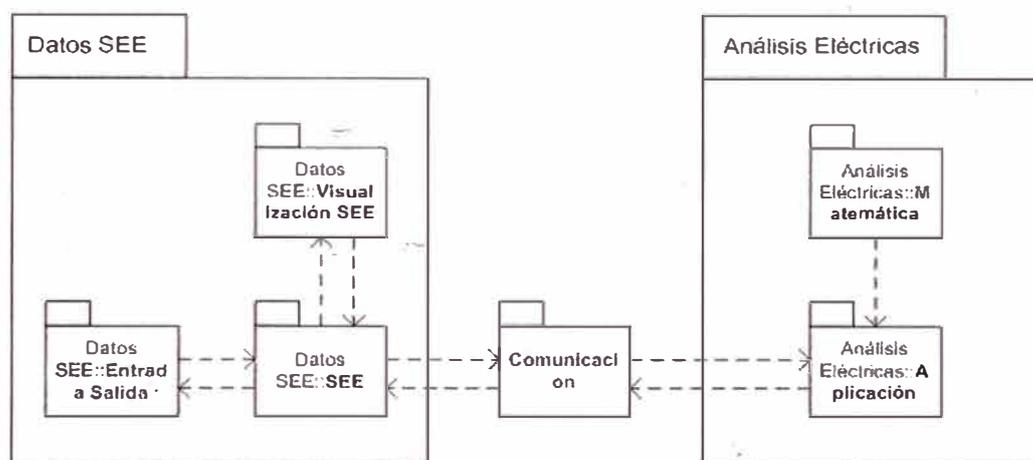


Figura 3.8: Diagrama de paquetes de diseño.

3.6.2. DISEÑO DE CLASES

El diseño de clases es el punto más importante del trabajo presentado en este informe. Es en este punto que se están centrando los esfuerzos de la comunidad científica para la resolución del problema descrito en la Formulación del Problema: “la definición de una estructura de clases que represente adecuadamente al SEE y a las diversas simulaciones que sobre él puedan realizarse”.

En las Figura 3.9 y 3.10 son mostrados los diagramas de clases correspondientes a los paquetes SEE y Aplicación respectivamente. Estos diagramas se encuentran aún en evolución en la tentativa que soporten cada vez un número mayor de simulaciones; siendo revisados y actualizados por el grupo de la COPPE – UFRJ, grupo al cual el autor pertenece.

Representación física

En la Figura 3.9 se pueden observar varias jerarquías existentes dentro del SEE. En un primer nivel jerárquico se presenta a la clase SEE, la cual es creada para agrupar toda la representación física de los elementos [eléctricos] que componen el sistema. Un SEE de grandes dimensiones (como el SINAC por ejemplo) es subdividido en áreas [geográficas] de forma que se simplifique la operación total del sistema. La clase SEE es entonces compuesta por una clase Area que puede contener otras áreas, subestaciones y líneas de transmisión, y que es operada por un EMS. La clase SEE es compuesta también por una clase Empresa, la cual representa a las empresas propietarias de los diversos elementos [eléctricos].

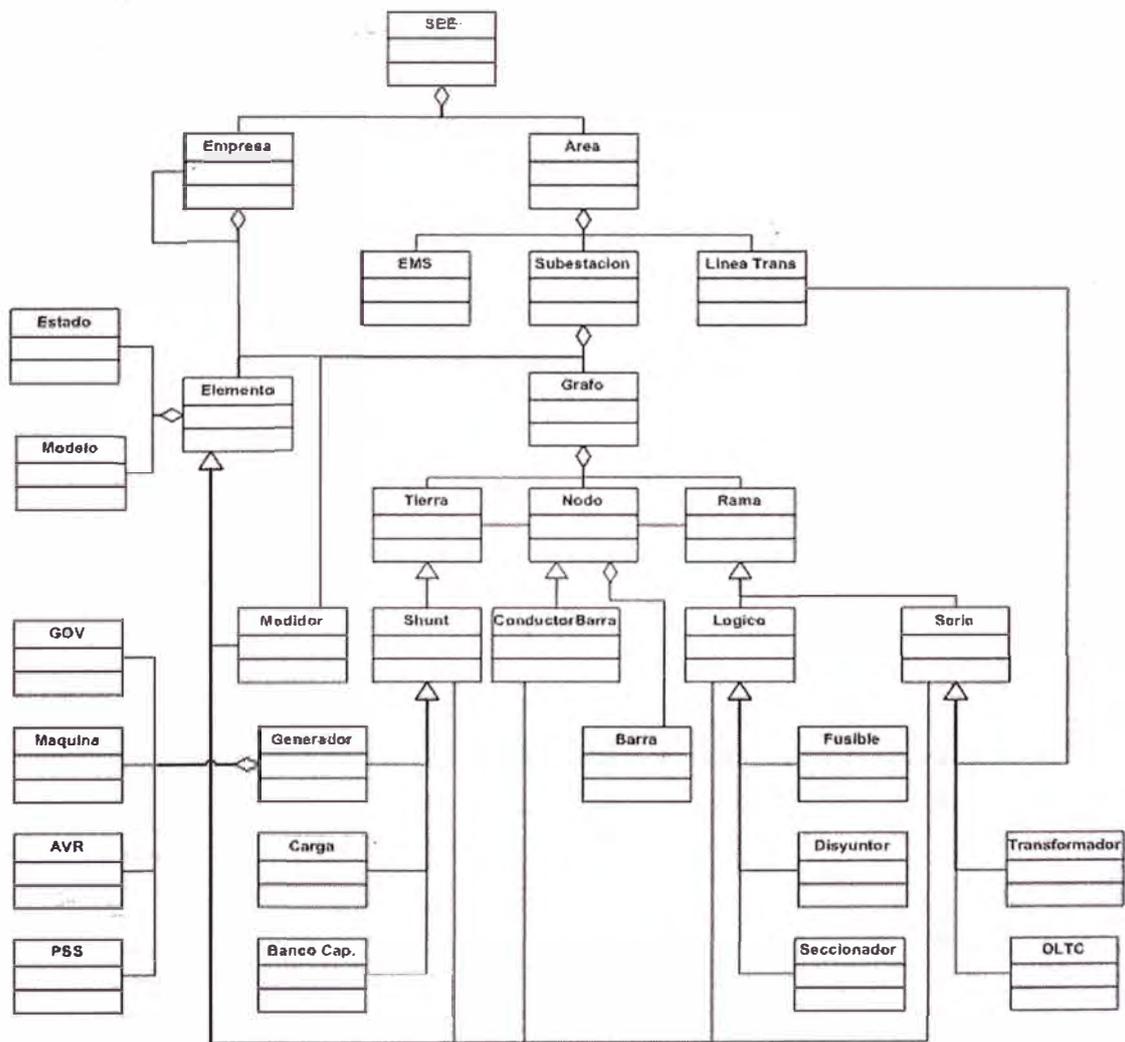


Figura 3.9: Diagrama de Clases del Paquete SEE.

La clase Subestación es compuesta por una clase Grafo y una clase Elemento [eléctrico]. La clase Grafo representa toda la conectividad existente entre los elementos de la red eléctrica, y es compuesta por un grupo de clases elementales: Nodo, Tierra y Rama. La elección de estas clases resulta de la clasificación de los elementos de acuerdo con el número de puntos de conexión. La clase Nodo representa un punto de conexión. Existe un nodo especial denominado de nodo de referencia. La clase Tierra representa a los elementos que se conectan entre un nodo y el nodo de referencia. La clase Rama representa a los elementos que se conectan entre dos nodos. En un SEE pueden existir elementos que posean más de dos puntos de conexión, mas estos elementos pueden ser representados como una combinación de elementos tierra y rama, por lo que no fueron considerados en el presente trabajo.

En un segundo nivel jerárquico se destacan los elementos que componen la subestación. Se pueden distinguir tres tipos principales de elementos: Shunt, Lógico y Serie.

La clase Shunt deriva de la clase Tierra y representa los elementos con apenas un punto de conexión, como por ejemplo: los generadores, las cargas, etc.

La clase Lógico deriva de la clase Rama y representa los elementos que poseen dos puntos de conexión mas que dependiendo de su estado (cerrado o abierto) terminan comportándose como conectores o aisladores ideales independientemente del análisis que se pretenda realizar; como por ejemplo: los disyuntores, los seccionadores, etc.

La clase Serie deriva de la clase Rama y representa los elementos que poseen dos puntos de conexión y poseen una impedancia relacionada; como por ejemplo: las líneas de transmisión, los transformadores, etc. Sería bueno comentar que la mencionada impedancia variará dependiendo del análisis que se pretenda realizar.

La clase Elemento es una clase genérica que representa a cualquier elemento físico que compone un SEE, y que define sus características generales comunes, tales como: identificador, nombre, empresa propietaria, etc.

La clase Medidor representa a los diversos medidores que pueden existir en un SEE, colocados principalmente para el monitoramiento de magnitudes eléctricas e inclusive magnitudes lógicas en los diversos elementos.

La clase ConductorBarra deriva de la clase Nodo con la característica principal de representar a los conductores a partir del cual se ramifican los demás elementos en las subestaciones. Esta clase posee características que las diferencian de un nodo común, ya que posee identificación, nivel de tensión, numeración, etc.

La clase Línea de Transmisión deriva de la clase Serie y representa una clase especial pues a pesar de ser un elemento que posee dos puntos de conexión, estos puntos se encuentran en subestaciones diferentes, por lo que no pertenecen al grafo (s) correspondiente a la subestación, por lo que se le define como un elemento que pertenece a una Área.

Configurador de Red

En el ítem 2.4.4 (configurador de red) fue comentado que una misma descripción física puede presentar diferentes configuraciones de operación dependiendo del estado de sus diversos elementos lógicos. Una vez que estos elementos poseen un estado seleccionado, la configuración es única. Los elementos lógicos poseen dos nodos asociados, cuando poseen un estado de cerrado, estos dos nodos pasan a pertenecer a la misma barra. La clase Barra representa al conjunto de nodos que formen un punto eléctrico común para motivos de análisis. Los objetos correspondientes a la clase Barra deberán ser creados en tiempo de ejecución de una forma que detecten los elementos shunt y serie que se encuentran conectados a ellos.

Descripción Funcional

En la Figura 3.10 se puede observar la jerarquía de clases existente entre los diversos análisis (simulaciones) que se pueden realizar sobre un SEE. Todas las posibles simulaciones poseen métodos comunes como por ejemplo: Recibir SEE, Ejecutar y Actualizar Resultados. Estos métodos son agrupados en una clase genérica denominada APPL o aplicación. Esta clase posee una referencia a la clase LSOL perteneciente al paquete Matematica, ya que independientemente del tipo de simulación a realizar, siempre será necesario resolver un sistema de ecuaciones del tipo "Ax = b".

La clase FLUP deriva de la clase APPL y representa a la simulación de flujo de potencia. Se puede observar que a partir de esta clase pueden ser definidas dos clases: FlujoAC y FlujoDC, las cuales implementan los diferentes métodos que resuelven el problema de flujo de potencia.

La clase ESTP deriva de la clase APPL y representa a la simulación de estabilidad permanente. Por razones técnicas, los diferentes métodos que resuelven este problema no justifican su conceptualización como clases.

La clase ESTT deriva de la clase APPL y representa la simulación de estabilidad transitoria. Se puede observar que a partir de esta clase pueden ser definidas dos clases: FET y SIMa, las cuales implementan los diferentes métodos que resuelven el problema de estabilidad transitoria.

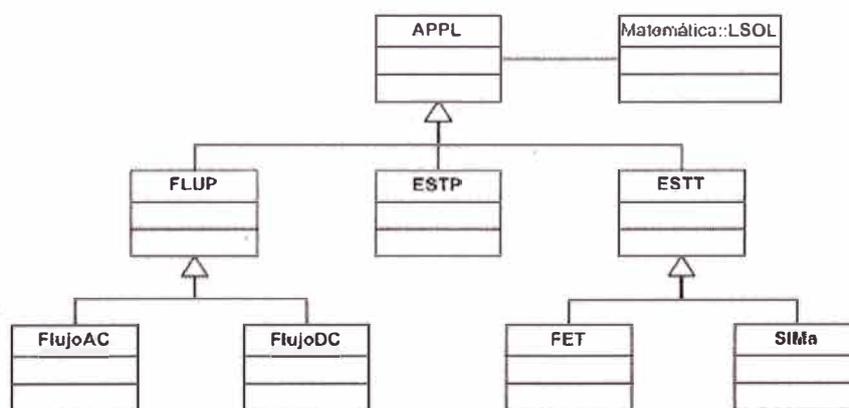


Figura 3.10: Diagrama de Clases del Paquete Aplicación.

3.7. IMPLEMENTACIÓN

Para probar el funcionamiento del modelo recientemente presentado se implementaron dos prototipos.

3.7.1. PRIMER PROTOTIPO IMPLEMENTADO.

El primer prototipo tiene la finalidad de demostrar la validez de la arquitectura cliente – servidor propuesta. Con esta finalidad se decidió que el cliente fuese implementado utilizando el lenguaje de programación Java y que el servidor fuese implementado utilizando el lenguaje de programación C++, de manera que la comunicación entre ellos sea responsabilidad del paquete comunicación, implementado siguiendo la especificación CORBA.

El cliente está conformado por las interfaces que permiten una amigable creación / edición de los principales elementos eléctricos comentados recientemente, así como la petición de la realización de alguna de las simulaciones. El servidor en este prototipo sólo considera la simulación de flujo de potencia, al ser la simulación más representativa dentro de los análisis eléctricos.

3.7.2. SEGUNDO PROTOTIPO IMPLEMENTADO

El segundo prototipo tiene la finalidad de demostrar la flexibilidad de las estructuras de clases presentadas. Este prototipo re-utiliza las clases correspondientes al servidor del primer prototipo adicionando las simulaciones de estabilidad permanente y estabilidad transitoria, mas por simplicidad la arquitectura escogida para las futuras actualizaciones y ampliaciones no sigue la especificación CORBA. Este prototipo fue implementado utilizando el lenguaje de programación C++. Las clases de flujo de potencia tanto en el primer y segundo prototipo son las mismas. La validación de este prototipo, denominado SinergiaSEE, se presenta en el capítulo 5 al comparar los resultados numéricos obtenidos con los resultados de softwares comerciales existentes en el mercado.

CAPÍTULO IV

EXPANSIÓN DE LA TRANSMISIÓN

4.1. INTRODUCCIÓN

El objetivo principal de los diversos SEE es atender la demanda con adecuados niveles de seguridad y calidad, incurriendo en el menor costo posible. Determinar las acciones o inversiones necesarias para que el sistema cumpla con su objetivo principal es llamado de planeamiento de sistemas eléctricos de potencia.

El planeamiento de los SEE envuelve los planeamientos de corto y largo plazo. En el contexto de planeamiento de corto plazo están envueltos el planeamiento de la operación y las medidas correctivas de corto plazo. Considerando la proyección del crecimiento de la demanda, existe un cierto nivel de este crecimiento para el cual el sistema existente no es más capaz de atender adecuadamente la demanda. Es en estas condiciones que existe la necesidad de proceder al planeamiento de largo plazo, incluyéndose en este contexto la construcción de nuevas unidades generadoras y líneas de transmisión con toda la infra-estructura necesaria. Determinar dónde, cuándo y cuántos nuevos equipos deban ser instalados en el SEE, de manera que el sistema abastezca la demanda proyectada, es conocido como el “problema de planeamiento de la expansión del sistema eléctrico”. La expansión del sistema eléctrico es un proceso global normalmente dividido en la expansión de la generación (se calculan las inversiones necesarias para la construcción de nuevas unidades generadoras

considerando fija la red de transmisión), en la expansión de la transmisión (se calculan las inversiones necesarias para la construcción de nuevas líneas / transformadores considerando predeterminado el plan de generación), y en el planeamiento de la distribución [OLIV04].

La formulación de la expansión de la transmisión es un problema complejo de naturaleza combinatoria. En la literatura especializada se pueden encontrar diversas metodologías que permiten sistematizar el proceso de búsqueda y toma de decisión, las cuales básicamente deben resolver problemas de optimización matemáticas de grandes dimensiones, no lineales, no convexos y de carácter combinatorios, complementados con la aplicación de técnicas para mitigar el riesgo económico.

El problema de la expansión de la transmisión es un proceso que normalmente se resuelve en dos etapas. Inicialmente se realiza un planeamiento estático, el cual considera un horizonte de tiempo determinado (por ejemplo 20 años) y se tienen como meta el número de circuitos que deberán ser adicionados y la inversión necesaria al inicio del horizonte de tiempo. Finalmente es realizado un planeamiento multi-etapas, el cual determina no sólo las mejores localizaciones y el tipo de inversión, sino también el tiempo más apropiado para realizar las inversiones. Existe un número de restricciones (por ejemplo, relativas a violaciones de tensiones, flujo de potencia reactiva, límites de estabilidad transitoria, etc.) que no son consideradas en ninguna de las etapas descritas debido principalmente al hecho de utilizar un modelo DC del sistema de transmisión [ESCO04].

En el presente capítulo se presenta inicialmente la formulación clásica (DC) del problema de la expansión de la transmisión. Seguidamente se presenta una breve revisión de las técnicas utilizadas en su solución. Luego se detalla una técnica de optimización combinatoria que será de interés en el presente trabajo: Algoritmos Genéticos (AG). Finalmente se presenta una metodología que resuelve una formulación que considera un modelo AC de la red e incorpora restricciones de estabilidad (permanente y transitoria).

4.2. FORMULACIÓN MATEMÁTICA CLÁSICA O DC

La expansión de la transmisión de energía eléctrica es un problema complejo de gran dimensión donde se tiene que minimizar el costo total de inversión producto de la adición de nuevas ramas, sujeto a restricciones de operación de la red de transmisión.

La operación de la red de transmisión está gobernada por las ecuaciones de flujo de potencia. Históricamente en los problemas de expansión suele utilizarse las ecuaciones del flujo de potencia lineal o DC comentadas en el ítem 2.5.5. Con esta consideración la expansión de la transmisión puede ser formulado como un problema de programación no lineal entero mixto (PNLEM), tal y como es mostrado en la ecuación (4.1).

$$\begin{aligned}
 \text{Min } v = & \sum_{(i,j) \in \Omega} c_{ij} n_{ij} + \sum_i \alpha_i r_i \\
 \text{s.a.} & B(x + \gamma^o) \theta + g + r = d \\
 & (x_{ij} + \gamma_{ij}^o) |\theta_i - \theta_j| \leq (x_{ij} + \gamma_{ij}^o) \phi_{ij}^{\max} \\
 & 0 \leq g \leq g^{\max} \quad \dots (4.1) \\
 & 0 \leq r \leq d \\
 & 0 \leq n_{ij} \leq n_{ij}^{\max}
 \end{aligned}$$

Donde:

c_{ij}	Costo de la adición de una rama igual a la rama (i,j).
n_{ij}	Número de ramas adicionados entre las barras i-j.
Ω	Conjunto de las posibles ramas a adicionar.
α	Parámetro de penalidad asociado con la pérdida de carga.
r	Vector de generadores ficticios o artificiales.
x_{ij}	Nueva susceptancia a ser instalada en el ramo (i,j) $\in \Omega$.
γ_{ij}^o	Susceptancia inicial en el ramo (i,j) $\in \Omega$.
$B(\cdot)$	Matriz de susceptancias.
θ	Ángulos de las tensiones complejas.
ϕ_{ij}^{\max}	Flujo máximo de potencia permitido.
g	Vector de generaciones.
d	Vector de cargas.

En la literatura correspondiente a la expansión de la transmisión se encuentran diferentes modelos matemáticos para la red de transmisión, mas prácticamente todos están basadas en un modelo DC [RIDE06]. A esta formulación la denominaremos como formulación clásica o formulación DC.

Esta formulación representa a un problema de optimización matemática perteneciente al conjunto de problemas llamados “*NP-complete*”, donde la solución óptima cuando son consideradas muchas variables no puede ser obtenida en tiempos razonables de tiempo de computación. Entre las variables podemos distinguir variables continuas, tales como los flujos en los ramos, los niveles de generación y demanda, y variables de inversión enteras como son los ramos candidatos a ser adicionados.

Dentro de la función objetivo puede ser observado el término “ $\sum \alpha r_i$ ” (i = barra de carga) correspondiente a la generación artificial adicionada en las barras de carga, término también llamado de “corte de carga”. Este término no corresponde a la formulación original, mas es adicionado dado que ayuda considerablemente en el proceso de convergencia. La gran ventaja de adicionar este término es que el problema transformado es siempre factible, pues las sobrecargas en el sistema que eventualmente aparecerían en el proceso de resolución, son compensadas por los generadores artificiales colocados en todas las barras de carga.

El hecho de considerar las ecuaciones linealizadas del sistema de transmisión permite que para una determinada propuesta de inversión (n_{ij} conocidas) la formulación sea reducida a un simple problema de programación lineal (PL), cuya finalidad es verificar la viabilidad de la propuesta. Por tanto un algoritmo eficiente debe presentar una solución sin corte de carga ($r=0$) para que esa solución sea equivalente a la formulación original. Ya en el proceso de solución, eventualmente se encuentran soluciones inviables ($r \neq 0$), mas las mismas permiten un mejor desarrollo de los mecanismos de transición y permiten salir más fácilmente de las soluciones óptimas locales [GALL97].

4.3. TÉCNICAS DE OPTIMIZACIÓN UTILIZADAS

De acuerdo al procedimiento seguido en la obtención del plano de expansión, las técnicas pueden ser clasificadas en dos grupos [LATO03]: técnicas basadas en algoritmos de optimización y técnicas basadas en algoritmos heurísticos. Entre tanto, existen técnicas que poseen características de los dos tipos, los cuales son llamados de meta-heurísticas.

4.3.1. TÉCNICAS DE OPTIMIZACIÓN CONVENCIONALES

Las técnicas basadas en algoritmos de optimización trabajan utilizando procedimientos de cálculo que resuelven la formulación matemática del problema. Esto significa resolver la formulación DC comentada como un problema PNLEM. En esta perspectiva la metodología más utilizada fue la técnica de descomposición de Benders [BEND62].

La técnica de descomposición de Benders explora la descomposición natural del problema, en dos partes: un sub-problema de inversión en que se elige un plan de expansión candidato y se calculan los costos de inversión asociados al mismo, y un sub-problema de operación donde es probado el plan de expansión candidato en términos de adecuada atención de la carga. La búsqueda de un óptimo global es realizada a través de una solución iterativa de las soluciones separadas de los sub-problemas de operación e inversión. Aplicaciones de esta técnica en la resolución de la formulación DC comentada pueden ser encontradas en [BINA01] [ROME96].

En [GALL97] se relata que con las metodologías de descomposición fueron obtenidos las soluciones globales en sistemas de pequeña y mediana dimensión, como el caso del sistema Garver (6 barras – 15 ramas) y el sistema Sul Brasileiro (46 barras – 79 ramas), mas para el sistema Nortenordeste Brasileiro (87 barras – 179 ramas) la técnica no obtuvo convergencia debido al número incalculable de alternativas presentes en la última etapa del proceso donde se tiene que resolver un problema de programación entera.

4.3.2. TÉCNICAS DE OPTIMIZACIÓN HEURÍSTICAS

Las técnicas basadas en algoritmos heurísticos trabajan paso a paso, generando, evaluando y seleccionando planes de expansión de la transmisión, realizando búsquedas locales con la guía de lógicas, de reglas empíricas y/o de sensibilidades. Estas reglas son usadas para generar y clasificar las opciones durante la búsqueda. El proceso heurístico es aplicado hasta el algoritmo no ser capaz de encontrar una solución mejor que considere los criterios de evaluación establecidos. Los criterios utilizados incluyen costos, sobrecargas, etc. Este conjunto de técnicas aun puede ser dividido en dos grupos: los algoritmos heurísticos constructivos y los algoritmos de optimización combinatoria.

Técnicas basadas en algoritmos heurísticos constructivos

Los algoritmos heurísticos constructivos se caracterizan como un procedimiento paso a paso que, a partir de la configuración base, adiciona en cada paso uno o más circuitos hasta conseguir una adecuada operación del sistema eléctrico. La diferencia básica entre un algoritmo y otro está en el índice de sensibilidad utilizado, es decir, en cada paso el circuito escogido para ser adicionado es identificado por un índice de sensibilidad pre-establecido. Entre las técnicas pertenecientes a este grupo se pueden mencionar: la técnica de Garver [GARV70], la técnica (método) de mínimo esfuerzo [MONT82] y la técnica de mínimo corte de carga [PERE85]. Estas técnicas generalmente son fáciles de implementar y requieren poco esfuerzo computacional, mas los valores obtenidos quedan generalmente lejos de la respuesta óptima cuando son aplicadas a sistemas de gran dimensión.

Técnicas basadas en algoritmos de optimización combinatoria

Resolver un problema de optimización combinatoria consiste en determinar la mejor o la solución óptima entre un gran número de soluciones alternativas. Se asume para esto que la calidad de una solución es cuantificable y que puede ser comparada con alguna otra solución, además, que el conjunto de soluciones es finito.

Una consecuencia directa de la propiedad de “*NP-complete*” es que la solución óptima no puede ser obtenida en tiempos razonables de computación. Grandes problemas “*NP-complete*” pueden ser resueltos con algoritmos apropiados, estos, por su vez pueden ser seleccionados entre dos opciones, una que obtiene el óptimo global con la desventaja de precisar de tiempos de computación impracticables, y otra que obtiene soluciones sub-óptimas con mucho menor tiempo de computación.

La primera opción constituye la clase de técnicas basadas en algoritmos de optimización (los que ya fueron comentados en el ítem 4.3.1). La segunda opción constituye la clase de algoritmos de aproximación en los cuales se puede distinguir los algoritmos generales y los algoritmos especializados. Los primeros son aplicables a una gran variedad de problemas, mientras que los segundos al emplear información específica limitan su aplicación a un conjunto de problemas restricto.

Algoritmos tales como “*simulated annealing*” (SA), búsqueda tabú (BT), “*greedy randomized adaptive search procedure*” (GRASP) y algoritmos genéticos (AG), son algoritmos generales de alta calidad a pesar que en implementaciones prácticas estos algoritmos se comportan como algoritmos de aproximación.

El SA hace uso de conceptos descritos en la Mecánica Estadística: en el proceso físico el material es calentado a una temperatura elevada para luego ser enfriado de manera lenta de tal forma que la estructura del material al final del proceso quede perfecta [AART89]. La BT nace de las reglas generadas en la inteligencia artificial en la cual se cuenta con un esquema de búsqueda local para explorar el espacio de soluciones aparte del óptimo local [GLOV89]. El GRASP es una técnica interactiva de muestreo aleatoria, donde cada iteración está compuesta de 2 fases, la primera encargada de la construcción de una solución inicial utilizando una función adaptativa aleatoria golosa, y la segunda encargada de aplicar una búsqueda local con la finalidad de mejorar la solución [FEO95].

Los AG (técnicas que comentaremos con más detalle en el ítem 5.4) son técnicas de búsqueda inspiradas en los mecanismos de la evolución y genética naturales que combinan el principio de la sobrevivencia del más apto con intercambios de informaciones genéticas entre individuos para el desarrollo de métodos de optimización simples y robustos [GOLD89].

Aplicaciones de las técnicas descritas en el párrafo anterior en el planeamiento de la expansión de la transmisión pueden ser encontradas en [ROME96] [SILV01] [FARI05] [SILV00]. Aún más en [GALL97] se encuentra una comparación entre tres de las técnicas recientemente descritas: SA, BT y AG cuando son aplicadas al problema de la expansión de la transmisión; concluyendo que ellas, en conjunto con técnicas heurísticas constructivas, pueden formar un equipo que trabajando cooperativamente pueden resolver el problema en estudio. Esta metodología es llamada de "Equipos Asíncronos".

Avances en el área de procesamiento paralelo sumado a la gran variedad de algoritmos apropiados para ser paralelizados, ha originado un creciente interés de los investigadores para resolver problemas de gran dimensión en tiempos computacionales pequeños. En [OLIV94] es presentado una implementación de la metodología de "Equipos Asíncronos" utilizando una versión paralela del SA y AG, los cuales son inicializados con algoritmos heurísticos constructivos y son ejecutados en un ambiente de procesamiento distribuido, siendo satisfactoriamente comprobado su aplicabilidad en sistemas de mediana y gran dimensión.

4.4. ALGORITMOS GENÉTICOS

4.4.1. INTRODUCCIÓN

Los algoritmos evolutivos emulan el proceso de selección natural, proceso que fue el impulsor para el surgimiento de complejas y bien adaptadas estructuras orgánicas. La emulación de estos algoritmos radica en la idea básica que una población de estructuras computacionales (individuos) evoluciona (a través de procesos estocásticos) de forma que exista una mejora en la adecuación promedio al medio ambiente de los individuos de esa población.

El paradigma neo-Darwiniano establece que la historia de la vida envuelve apenas pocos procesos estocásticos que actúan en las (y entre las) poblaciones y especies. Estos procesos son: reproducción, mutación, competición y selección.

La mayoría de las implementaciones actuales de los algoritmos evolutivos se derivan de 5 grandes corrientes independientes, aunque fuertemente relacionadas. Estas corrientes se diferencian en la forma de aplicación de los procesos estocásticos comentados. Entre ellas tenemos a los algoritmos genéticos, la programación evolutiva, las estrategias evolutivas, la programación genética y el *Particle Swarm Optimization* (PSO).

Una de las principales razones de la popularización de los algoritmos evolutivos radica en el hecho de ser el algoritmo de optimización preferido por la naturaleza y de que en muchos casos prácticos de ingeniería no se busca una "solución óptima global", sino lo que se busca realmente es "una mejor solución" que las existentes o conocidas. Es con esta motivación que en este sub-capítulo detallamos una las corrientes pertenecientes a los algoritmos evolutivos: los algoritmos genéticos.

4.4.2. ALGORITMO GENÉTICO SIMPLE (AGS)

De acuerdo con la teoría de selección natural, solamente los individuos más capaces (mejor adaptados al medio) sobreviven, transmitiendo su material genético para sus descendientes. El algoritmo genético simple (AGS) es una técnica de búsqueda inspirada en los mecanismos de la selección natural y genética natural que combina el principio de la sobre-vivencia del más apto con intercambios de informaciones genéticas entre individuos para el desarrollo de métodos de optimización simples y robustos [GOLD89].

En el AGS se posee una población conformada por un conjunto de estructuras (individuos), cada una de las cuales se encuentra codificada como una secuencia de variables binarias (genotipo) que representan una solución potencial del problema (fenotipo). La población es inicializada de manera aleatoria y varía a través de un proceso iterativo, en el que en cada iteración (generación) se aplican los principios de evolución (sobre-vivencia del más apto) y genética (cruzamiento y mutación). La evaluación de la función objetivo correspondiente a cada individuo determina su respectiva aptitud (función *fitness*), y es la única información dependiente del problema necesario para el AGS. La aplicación de los operadores genéticos es de naturaleza probabilística. El proceso iterativo termina cuando se cumple un criterio de parada (objetivo alcanzado, número de generaciones, tiempo, etc.).

La mayor parte de la teoría perteneciente a los AG está basada en el algoritmo genético simple (AGS), el cual fue introducido por Holland en 1962. El AGS funciona bien para problemas de optimización simples y/o de pequeña dimensión en los que se desee maximizar una función objetivo, y es sumamente utilizado con finalidades didácticas.

4.4.3. ALGORITMO CONCEPTUAL

El procedimiento descrito en el sub-ítem anterior se visualiza mejor con la ayuda del siguiente algoritmo conceptual correspondiente al AGS:

```
Inicio
t = 0 (Contador de generaciones)
Crea una población inicial P(t=0)
Evalúa la población inicial P(t=0)
Mientras un criterio de parada no sea alcanzado
    P'(t) = Selección de soluciones de de P(t)
    P(t+1) = Cruzamiento y Mutación de los individuos de P'(t)
    Evalúa P(t+1)
    t = t + 1
Continúa
Fin
```

La evolución de la población de un AGS utilizado en la solución del problema de maximización de la función mostrada en la parte superior izquierda de la Figura 4.1 es mostrada en la parte inferior, para la generación cero (población inicial) en la parte izquierda, así como después de 10 generaciones en la parte derecha.

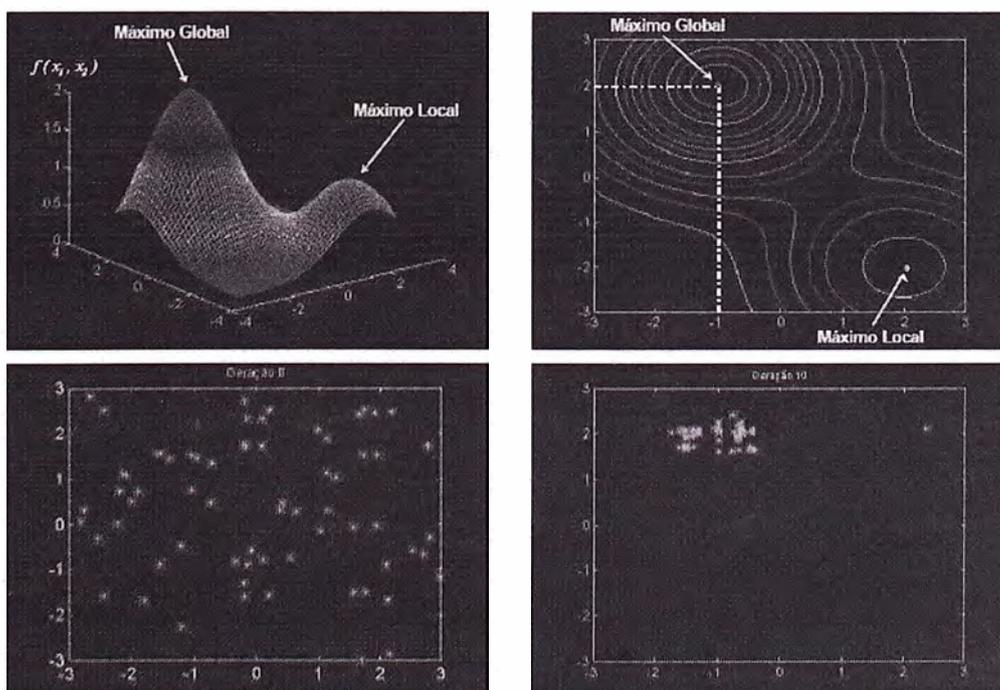


Figura 4.1: Ejemplo de la convergencia del AG a través de las generaciones

4.4.4. ELEMENTOS DEL ALGORITMO GENÉTICO SIMPLE

Codificación

La utilización de la codificación permite una representación compacta de las posibles soluciones del problema. Se asume que una solución potencial de un problema puede ser representada como un conjunto de parámetros. Estos parámetros (llamados de genes) son colocados juntos para formar un *string* de valores llamado cromosoma. El conjunto de cromosomas es llamado individuo, el cual representa una solución potencial del problema a resolver.

El AGS utiliza una codificación binaria para los genes, independientemente de si corresponden a variables enteras, reales o discretas; mas el AG puede ser mejorado utilizando codificaciones enteras, reales o gray [MICH96] dependiendo del problema que se desee resolver.

Un ejemplo de codificación binaria es mostrado a continuación: Una variable entera X_1 que varíe entre 0 y 31 (32 valores posibles $\leq 2^5$) puede ser representado por un gen codificado como una secuencia de 5 bits, en el que [0 0 0 0 0] representa al entero cero y [1 1 1 1 1] representa al entero 31. Una variable discreta X_2 que varíe entre -0.3 y 0.4 con un paso de 0.1 (8 valores posibles $\leq 2^3$), puede ser representado por un gen codificado como una secuencia de 3 bits, en el que [0 0 0] represente al real -0.3 y [1 1 1] represente al real 0.4.

Un problema de optimización en el que las variables sean X_1 y X_2 podría ser resuelto con un AGS en el que cada individuo (en este caso coincidente con el cromosoma) sea representado tal y como es mostrado en al Figura 4.2:

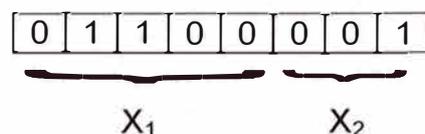


Figura 4.2: Ejemplo de un cromosoma con codificación binaria.

Función Fitness (Evaluación)

Cada problema que se desee resolver con la utilización de un AGS, debe poseer asociado una función *fitness*. Dado un individuo particular, la función *fitness* retorna un número escalar que corresponde a la aptitud del individuo (supuestamente proporcional a su utilidad o habilidad).

El AGS utiliza la función objetivo como función *fitness*, mas el AG a veces puede ser mejorado considerando, además de la función objetivo, varias medidas de desempeño dentro de la definición de la función *fitness*.

Tal y como fue mencionado anteriormente la evaluación de la función objetivo (función *fitness*) es la única información dependiente del problema necesario para el AGS. Esta característica permite al AGS tener una interacción sumamente fácil con aplicaciones específicas, tal y como es mostrado en la Figura 4.3.

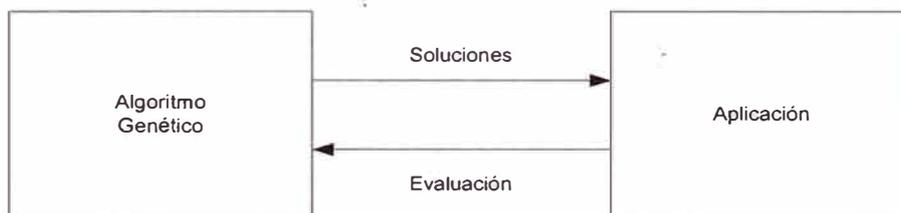


Figura 4.3: Ejemplo de la Interacción entre el AG y una aplicación específica.

Selección

A partir de la población de la generación t , la cual ya se encuentra evaluada, son escogidos los individuos para reproducción (genitores o *mating pool*) a través de un proceso de selección probabilística, que asegura que el número esperado de veces que un individuo es escogido sea proporcional a su desempeño relativo al resto de la población.

Un esquema simple de selección es el proporcional. En este esquema se asocia a un individuo X_i con *fitness* f_i una descendencia proporcional a f_i/f_i^{med} . Se puede apreciar claramente que individuos con *fitness* mayor conseguirán más descendientes. Uno de los inconvenientes de este

esquema es que el número de descendientes puede resultar fraccionario. Una forma de resolver este inconveniente es utilizar la regla de la ruleta. Esta regla divide una ruleta en sectores proporcionales al *fitness* relativo de cada individuo, luego con la ayuda de un generador de número aleatorios se sorteá un sector de la ruleta, procediendo a seleccionar al individuo correspondiente, proceso que es repetido un número de veces igual al número de individuos de la población.

Luego de este proceso en el *mating pool* se poseen copias exactas de los individuos seleccionados de la población de generación *t*.

Cruzamiento

A través del operador de cruzamiento dos individuos seleccionados aleatoriamente del *mating pool* intercambian parte de sus genes, en su forma binaria. El cruzamiento no es usualmente aplicado a todos los pares de individuos seleccionados, este proceso es controlado por un parámetro denominado probabilidad de cruzamiento (P_c). Para cada dos individuos seleccionados es generado aleatoriamente un número entre cero y uno, el cual si es menor que P_c origina el intercambio de porciones del cromosoma en un punto de cruzamiento elegido también aleatoriamente, tal y como es mostrado en la Figura 4.4.

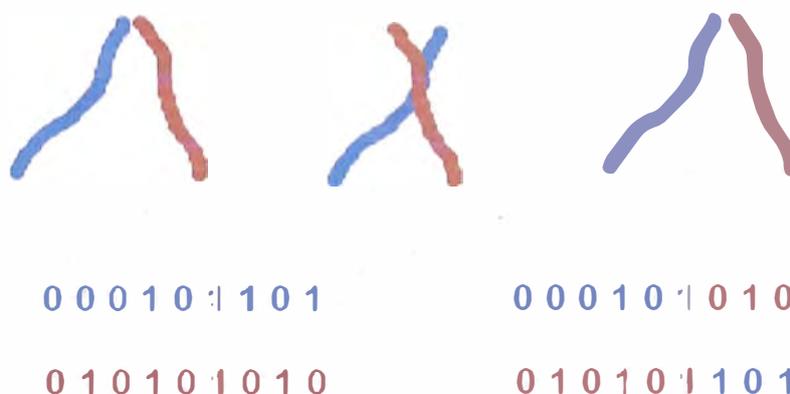


Figura 4.4: Ejemplo del operador cruzamiento

Mutación

Después del cruzamiento un operador de mutación es aplicado de tal forma de explorar aleatoriamente otras trayectorias en el espacio de búsqueda. La mutación consiste en el cambio aleatorio (de 0 a 1, o de 1 a 0) de algunos bits de la población de acuerdo con una probabilidad de mutación (P_m).

El operador mutación cumple un papel secundario, aunque decisivo en el AGS, él entra en acción cuando los procesos de selección y cruzamiento terminan en la homogenización de la población, y en eventualmente pérdida de material genético. La mutación inyecta, de forma aleatoria, nueva información en los individuos a través de alteraciones cromosómicas. Este operador puede ser una herramienta importante contra la convergencia prematura, evitando mínimos locales.

4.4.5. PARÁMETROS DE CONTROL

Para que un AGS pueda funcionar deben ser entregados valores apropiados para un conjunto de parámetros, entre los que se pueden destacar:

- *Tipo de codificación*, generalmente de tipo binaria.
- *Tamaño de la población* (N), cuyo valor típico varía entre 30 y 200.
- *Longitud del cromosoma*, valor dependiente de la cantidad de bits requeridos por todas las variables.
- *Probabilidad o tasa de cruzamiento* (P_c) cuyo valor típico varía entre 0.5 y 1.0.
- *Probabilidad o tasa de mutación* (P_m) cuyo valor típico varía entre 0.001 y 0.01.
- *Criterio de parada*, generalmente el número máximo de generaciones.

4.5. FORMULACIÓN MATEMÁTICA PROPUESTA O AC

La inclusión de restricciones que consideren la dinámica (estabilidad) de los diversos elementos pertenecientes a los SEP, obliga al uso del modelado AC de la red. El uso del modelo AC del sistema de transmisión en el problema de la expansión de la transmisión es incipiente, es decir, prácticamente no existen publicaciones en la literatura especializada en que sea utilizado este modelo, típicamente utilizado en los diversos análisis de la operación de un SEP. Históricamente, la separación de modelos para trabajos de planeamiento de la operación y de planeamiento de la expansión aconteció en la década de 1960 cuando investigadores como L. L. Garver [GARV70] verificaron que no era posible trabajar con el modelo AC en los trabajos de planeamiento de la expansión. A partir de entonces fueron utilizados modelos relajados de representación de la red siendo finalmente el modelo DC considerado como modelo ideal para los trabajos de planeamiento de la expansión de sistemas de transmisión, tendencia que continúa hasta la actualidad [RIDE06].

Actualmente, las grandes desventajas de utilizar el modelo DC en los trabajos de expansión de la transmisión pueden resumirse en:

- Frecuentemente es necesario implementar una fase de refuerzo (adición de líneas) sobre el plan obtenido con el modelo DC para que opere adecuadamente con el modelo AC.
- El problema de localización de fuentes reactivas (parte inherente del proceso de planeamiento de la expansión), debe ser tratado separadamente.
- El plan total de expansión no es óptimo, ya que es fuertemente influenciado por los resultados obtenidos por la expansión inicial obtenida con el modelo DC.

Una formulación matemática que utilice un modelo AC de la red de transmisión no presenta las desventajas mencionadas y permite la inclusión de restricciones que consideren la dinámica de los elementos pertenecientes al SEE. Esta formulación, la cual denominaremos de formulación AC, es definida tomando como base la formulación DC presentada en la ecuación (4.1) y las consideraciones de flujo de potencia no lineal, estabilidad permanente y estabilidad transitoria presentadas en el capítulo 2; tal y como es mostrado en la ecuación (4.2)

$$\begin{aligned}
 \text{Min } v = & \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \\
 \text{s.a.} & V_i \sum_{j \in I} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \text{sen} \theta_{ij}) - P_i = 0 \quad i = 1, \dots, n_b \\
 & V_i \sum_{j \in I} V_j (G_{ij} \text{sen} \theta_{ij} - B_{ij} \cos \theta_{ij}) - Q_i = 0 \quad i = 1, \dots, n_b \\
 & V_i^{\min} \leq V_i \leq V_i^{\max} \quad i = 1, \dots, n_b \quad \dots (4.2) \\
 & 0 \leq MVA_i \leq MVA_i^{\max} \quad i = 1, \dots, n_r \\
 & \text{Min}(\xi_i) \leq 10\% \quad i = 1, \dots, n_p
 \end{aligned}$$

estable transitoriamente

Donde:

c_{ij}	Costo de la adición de una rama igual a la rama i-j.
n_{ij}	Número de ramos adicionados entre as barras i e j.
Ω	Conjunto de las posibles ramas a adicionar.
n_b, n_r	Número de barras y ramos del SEE.
V_i	Módulo de la tensión en la barra i.
G_{ij}	Conductancia final entre las barras i e j.
B_{ij}	Susceptancia final entre las barras i e j.
θ_{ij}	Diferencia entre los ángulos de la tensión de las barras i e j.
P_i	Potencia activa inyectada en la barra i.
Q_i	Potencia reactiva inyectada en la barra i.
V_i^{\min}	Tensión máxima permitida en la barra i.
V_i^{\max}	Tensión mínima permitida en la barra i.
MVA_i	Potencia aparente en el ramo i.
MVA_i^{\max}	Potencia aparente máxima en el ramo i.
ξ_i	Amortecimiento correspondiente al polo i.

Las diferencias entre la presente formulación y la presentada en la ecuación (4.1) se encuentran en las dos primeras restricciones (en que la red de transmisión utiliza un modelado AC), lo cual permite la inclusión de la tercera restricción correspondiente a los límites de tensión (en las barras de interés), y principalmente la inclusión de la quinta y sexta restricción, indicadores de la estabilidad permanente y transitoria respectivamente. Otra diferencia es que en la presente formulación la generación es fija (no es permitido una reprogramación de la generación), por lo que la restricción correspondiente a los límites de generación y demanda no es considerada.

La ecuación (4.2) puede ser reformulada como un problema de optimización sin restricciones, para lo cual las restricciones originales pasan a formar parte de la función objetivo penalizándola, tal como es mostrado en la ecuación (4.3):

$$\text{Min } v = \sum_{ij} c_{ij} n_{ij} + \alpha (I_{fc} + I_{fs} + I_{sc}) \quad \dots(4.3)$$

Donde:

c_{ij}	Costo de la adición de una rama igual a la rama i-j.
n_{ij}	Número de ramos adicionados entre as barras i e j.
α	Parámetro de penalidad.
I_{fc}	Índice que considera las 4 primeras restricciones de (5.2).
I_{fs}	Índice que considera la penúltima restricción de (5.2).
I_{sc}	Índice que considera la última restricción de (5.2).

La ecuación (4.3) aún representa un problema de optimización no lineal entero mixto, donde para cada posible solución los índices establecen que tanto están siendo atendidas sus correspondientes restricciones, siendo cero en el caso que sean atendidas completamente. La ventaja de esta formulación es que cuando es escogida (o determinada) una propuesta de inversión o configuración, la formulación (4.3) se reduce a un problema de cálculo de los índices cuya única finalidad es verificar la viabilidad de la propuesta de inversión, es decir, si la inversión propuesta produce índices iguales a cero.

4.6. METODOLOGÍA AC PROPUESTA

El problema formulado en la ecuación (4.3) puede ser sub-dividido en dos sub-problemas resueltos de modo sucesivo e iterativo tal y como es mostrado en la Figura 4.5.

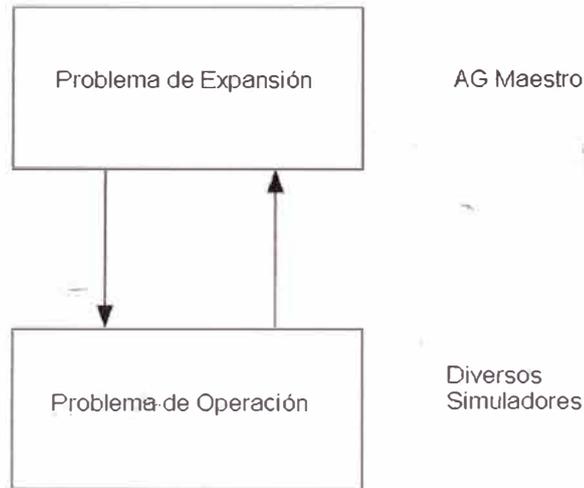


Figura 4.5: Descomposición del problema de la Expansión de la Transmisión.

En el primer sub-problema se decide respecto a las variables de inversión (la cantidad y localización de las nuevas ramas) siendo resuelto con la utilización de un algoritmo genético que llamaremos de maestro. En el segundo sub-problema se determina si con las nuevas ramas adicionadas por el AG maestro es posible operar la red con índices (I_{fc}, I_{fs}, I_{sc}) iguales a cero. Alguno de los índices son difíciles de formular matemáticamente, mas pueden ser calculados como resultado de algunas simulaciones (análisis) eléctricas. El índice I_{fc} puede ser calculado en base a los resultados de una simulación de flujo de potencia o un flujo de potencia óptimo. El I_{fs} puede ser calculado en base a los resultados de una simulación de estabilidad permanente y el I_{sc} puede ser calculado en base a los resultados de una simulación de estabilidad transitoria.

En este sub-capítulo se presentan los detalles concernientes al AG maestro utilizado, y las formas de cálculo de los índices (I_{fc}, I_{fs}, I_{sc}) , así como una adaptación efectuada para mejorar la convergencia en el AG utilizado.

AG maestro

Un AG denominado maestro es el encargado de la determinación de los refuerzos y ampliaciones de los ramos. Considerando que existen N ramos que pueden ser reforzadas o adicionados, que l es el número máximo de refuerzos o adiciones por ramo, y que n es el número mínimo de bits que permiten representar l en base binaria; en la Figura 4.6 se puede observar la representación del cromosoma utilizado. Cada cromosoma contiene N genes, cada uno de los cuales utiliza n bits para representar un número entero. Los n primeros bits (primer gen) del cromosoma indican el número de refuerzos / adiciones iguales al ramo 1 existente que serán construidos; los n siguientes bits (segundo gene) indican el número de refuerzos / adiciones iguales al ramo 2 existente que serán construidos, y así por delante hasta llegar al ramo N .

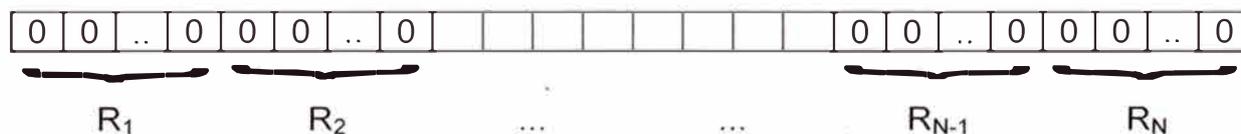


Figura 4.6: Representación del cromosoma – AG maestro.

En la Figura 4.7 se muestra el diagrama de flujo correspondiente a la función *fitness* utilizado por el AG maestro. Se observa que para evaluar la aptitud de cada cromosoma es necesario calcular los índices I_{fc} , I_{fs} e I_{sc} . En esta figura se puede observar también que el cálculo de I_{fs} e I_{sc} depende del cálculo de I_{fc} , y que el cálculo de I_{sc} depende del cálculo de I_{fs} . Esto es debido a que por ejemplo, para atender las restricciones correspondientes a I_{fs} e I_{sc} primero deberán ser atendidas las restricciones correspondientes a I_{fc} . Por lo tanto es necesario que los índices posean límites que representen su relativa relevancia, es decir:

$$I_{sc} \leq I_{fs} \leq I_{fc}$$

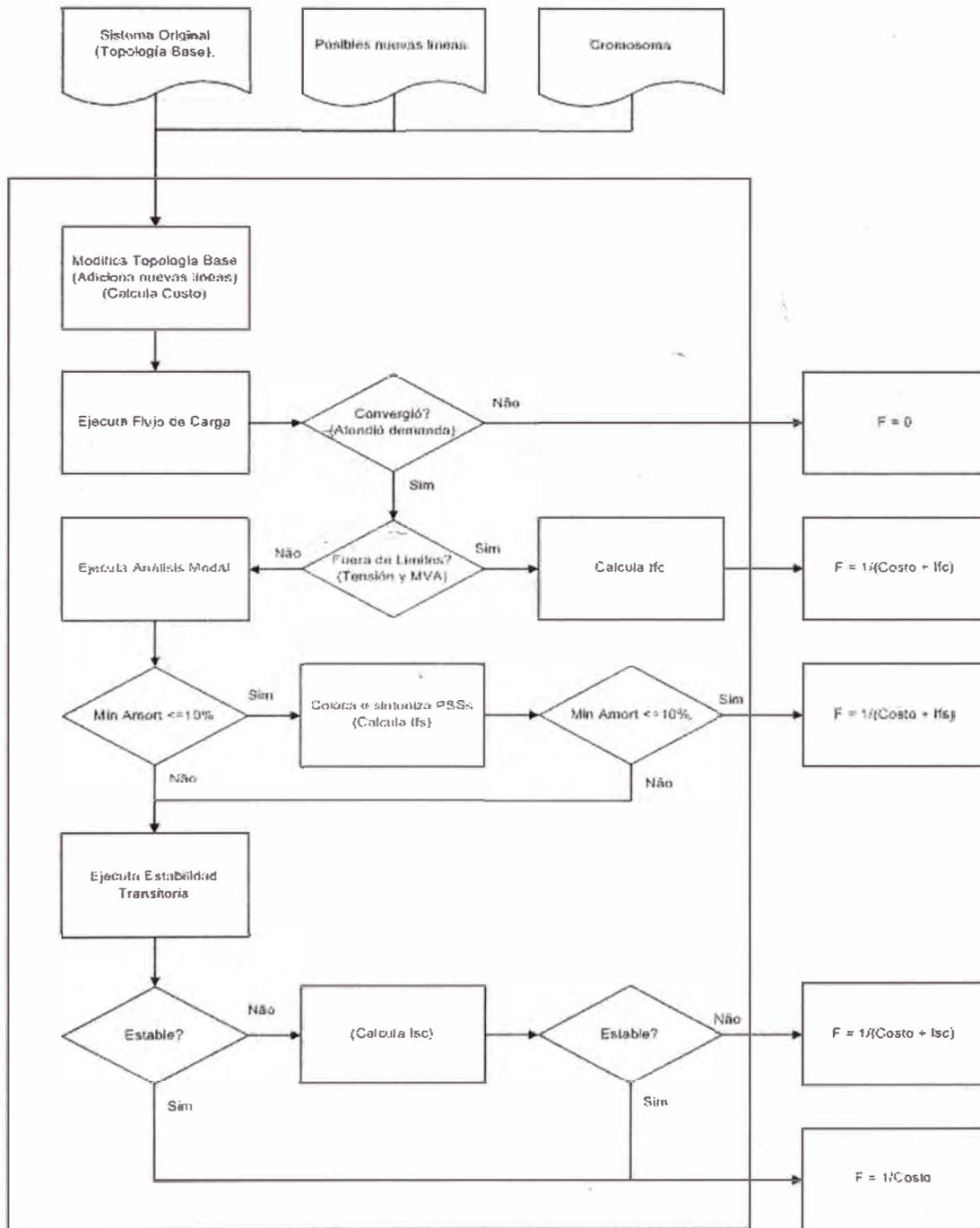


Figura 4.7: Diagrama de flujo de la función *fitness* – AG maestro.

Cálculo de I_{fc} .

Para el cálculo de I_{fc} fueron utilizados los resultados correspondientes al análisis de flujo de potencia, tal e como mostrado a seguir:

$$I_{fc} = \begin{cases} \infty & \text{si el flujo no converge.} \\ 40C_{\max} + 20C_{\max} \left(\frac{\sum_i (S_i - S_i^{\max}) + \sum_j \Delta V_j}{S_{\text{sistema}} + 0.4 * N_b} \right) & \begin{matrix} i \in \{\text{ramos sobrecargados}\} \\ j \in \{\text{barras fuera limites}\} \end{matrix} \\ 0 & \text{si no existe sobrecarga ni sobre-tensiones.} \end{cases}$$

Donde:

C_{\max}	Costo correspondiente a la colocación de todos los refuerzos y adiciones.
S_i	Potencia aparente en el ramo i .
S_i^{\max}	Potencia aparente máxima en el ramo i .
ΔV_j	Desvío de tensión en la barra j .
S_{sistema}	Potencia aparente del sistema.
N_b	Número de barras del sistema.

Cálculo de I_{fs} .

Para el cálculo de I_{fs} fueron utilizados los resultados correspondientes a la estabilidad permanente. Caso el mínimo amortecimiento no sea mayor que 10% son diseñados estabilizadores de potencia (PSS 's) en todos los N_g generadores, siendo asumida una estructura de control clásica con el modelo dinámico consistiendo de una ganancia, un bloque *wash-out* y un bloque de avance-atraso, tal y como es mostrado a seguir:

$$K_i(s) = K_i \frac{T_w s}{1 + T_w s} \left(\frac{1 + \frac{\sqrt{\alpha_i}}{w_i} s}{1 + \frac{1}{w_i \sqrt{\alpha_i}} s} \right)^2 ; i = 1, \dots, N_g$$

El valor de T_w no es considerado crítico, por lo que es considerado constante e igual a 5 segundos. El cálculo de las demás variables: K_i , w_i e α_i ($i=1, \dots, N_g$), es realizado con la utilización de un AG llamado de esclavo.

Entonces el problema de diseño correspondiente al AG esclavo es formulado como el siguiente problema de optimización:

$$\text{Max } F = \sum_{i=1}^{N_g} \left[\sum_{j=1}^{N_p} \xi_j \right]$$

$$\begin{aligned} \text{s.a: } & 0.00 < K_i < 20.0 \\ & 1.25 < w_i < 12.56 \\ & 0.10 < \alpha_i < 10.00 \\ & 10\% < \xi_{\min} \end{aligned}$$

Cada variable es codificada en binario (20 bits), tal y como está representado en la Figura 4.8:

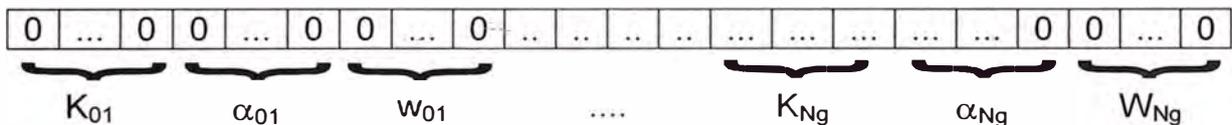


Figura 4.8: Representación del cromosoma – AG esclavo.

La función *fitness* (f) utilizada por el AG esclavo es definida como:

$$f = \begin{cases} 0 & \text{si algún } \zeta < -0.15. \\ 2 & \text{si todos } \zeta > -0.15 \text{ y algún } \zeta < 0.00 \\ 10 & \text{si todos } \zeta > 0.00 \text{ y algún } \zeta < 0.01 \\ 20 & \text{si todos } \zeta > 0.01 \text{ y algún } \zeta < 0.02 \\ 30 & \text{si todos } \zeta > 0.02 \text{ y algún } \zeta < 0.03 \\ 40 & \text{si todos } \zeta > 0.03 \text{ y algún } \zeta < 0.04 \\ 50 & \text{si todos } \zeta > 0.04 \text{ y algún } \zeta < 0.05 \\ 60 & \text{si todos } \zeta > 0.05 \text{ y algún } \zeta < 0.06 \\ 70 & \text{si todos } \zeta > 0.06 \text{ y algún } \zeta < 0.07 \\ 80 & \text{si todos } \zeta > 0.07 \text{ y algún } \zeta < 0.08 \\ 90 & \text{si todos } \zeta > 0.08 \text{ y algún } \zeta < 0.09 \\ 100 & \text{si todos } \zeta > 0.09 \text{ y algún } \zeta < 0.10 \\ 110 & \text{si todos } \zeta > 0.10 \end{cases}$$

O teste de parada del AG esclavo es la obtención de por lo menos un individuo con un *fitness* mayor que 100 al llegar al número máximo de generaciones.

El índice I_{fs} es calculado de acuerdo a la siguiente relación:

$$I_{fs} = \begin{cases} 0 & \text{Si } f > 100. \\ 20C_{\max} + 20C_{\max} \left(\frac{110-f}{110} \right) & \text{Si } f \leq 100 \end{cases}$$

Donde:

C_{\max} Costo correspondiente a la colocación de todos los refuerzos y adiciones.

f *Fitness* del mejor individuo del AG esclavo.

Cálculo de I_{sc} .

Para el cálculo de I_{fs} fueron utilizados los resultados correspondientes a la estabilidad transitoria. El índice I_{sc} es calculado de acuerdo a la siguiente relación:

$$I_{sc} = \begin{cases} 0 & \text{si el sistema es estable.} \\ 20C_{\max} & \text{si el sistema es inestable} \end{cases}$$

Donde:

C_{\max} Costo correspondiente a la colocación de todos los refuerzos y adiciones.

Parámetro α variable

El parámetro α en conjunto con los índices I_{fc} , I_{fs} e I_{sc} penaliza la función objetivo tornando las configuraciones propuestas poco atractivas en el proceso de selección. El parámetro α debe asumir un valor relativamente elevado para garantizar que el proceso pare con una configuración con un índice igual a cero. Mas, en las fases iniciales del proceso, un α grande produce una prematura eliminación de configuraciones que no cumplen las restricciones que eventualmente podrían llevar el proceso para regiones promisorias. Por eso es deseable que en las fases iniciales del proceso las configuraciones que no atienden las restricciones tengan chances de sobrevivencia por algunas generaciones. En el presente trabajo esto es obtenido con un α variable, iniciando con un α menor e incrementando su valor para cada generación, tal y como está mostrado en la Figura 4.9:

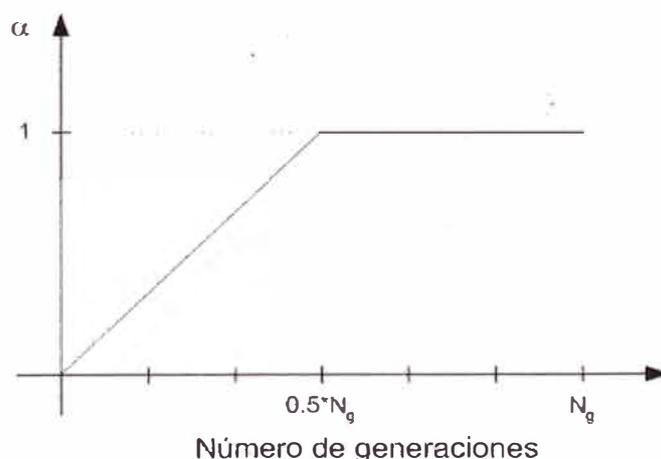


Figura 4.9: Variación del parámetro α a través de las generaciones.

4.7. IMPLEMENTACIÓN DE LA METODOLOGÍA AC EXTENDIENDO LA BASE COMPUTACIONAL PROPUESTA

La metodología propuesta en el ítem 4.6 fue implementada extendiendo y utilizando la base computacional presentada en el capítulo 3. En este sub-capítulo se muestra de manera muy resumida las alteraciones efectuadas en la mencionada base. En la Figura 4.10 se puede observar que la adición de esta nueva simulación origina un nuevo caso de uso “Ejecutar Expansión Transmisión” el cual reutiliza algunos de los casos de uso definidos anteriormente.

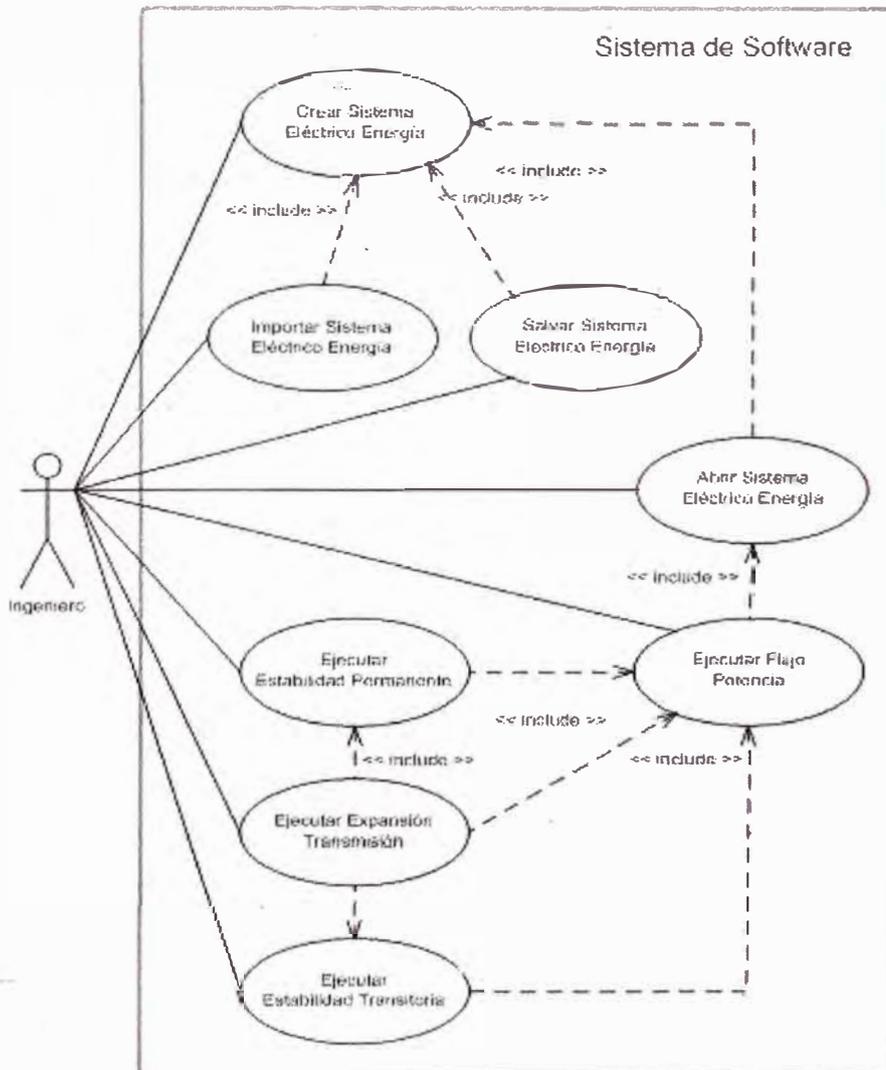


Figura 4.10: Extensión del Diagrama de Casos de Uso.

La inclusión de esta nueva simulación no originó alteraciones en la estructura de clases (Figura 3.9) del paquete SEE presentada en el capítulo 3. Esto es comprensible debido a que el sistema de energía representado por esta estructura de clases fue diseñado de forma de ser lo más independientemente posible de las simulaciones que se necesiten realizar.

La inclusión de esta nueva simulación origina una nueva clase en la estructura de clases (Figura 3.10) del paquete Aplicación, la cual reutiliza muchas de las clases correspondientes a simulaciones existentes. Por ejemplo la clase EXPA_TRAN (correspondiente a la simulación de expansión de la transmisión) posee una relación de agregación con la clase FlujoAC (correspondiente a la simulación de flujo de potencia) debido a que es necesario una simulación de flujo de potencia para el cálculo del índice I_{fc} . Igual razonamiento se utiliza para explicar las relaciones de agregación de la clase EXPA_TRAN con las clases ESTP (correspondiente a la simulación de estabilidad permanente) y SIMa (correspondiente a la simulación de estabilidad transitoria).

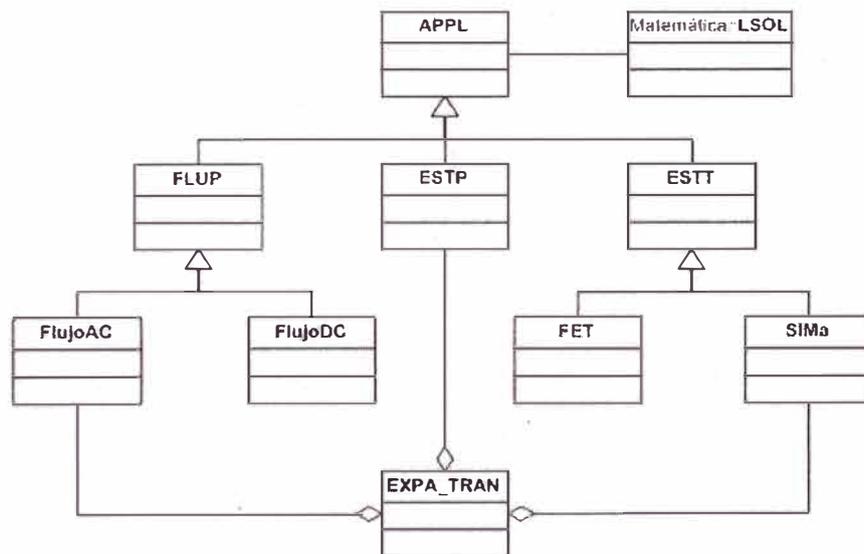


Figura 4.11: Extensión del diagrama de clases.

La utilización de un algoritmo genético originó la inclusión de una estructura de clases dentro del paquete Matematica, el cual puede ser observado en la Figura 4.12.

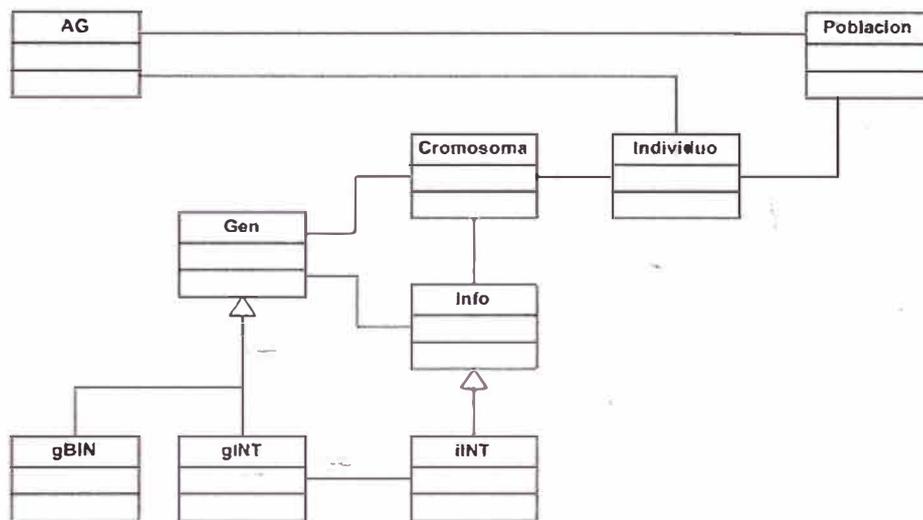


Figura 4.12: Diagrama de clases del AG.

La implementación recientemente descrita fue realizada sobre el segundo prototipo comentado en el capítulo 3, utilizando el lenguaje de programación C++. La validación de este prototipo se presenta en el capítulo 5 al comparar los resultados numéricos obtenidos con algunos resultados conocidos dentro de la literatura especializada.

CAPÍTULO V

ANÁLISIS DE RESULTADOS

5.1. INTRODUCCIÓN

La implementación de las metodologías computacionales presentadas en el capítulo 2 para la resolución de los problemas de flujo de potencia, estabilidad permanente y estabilidad transitoria, se aplica en este capítulo en un sistema eléctrico bastante popular en la literatura especializada, el WECC [ANDE02]. Los resultados obtenidos son comparados con las salidas de 3 simuladores digitales de uso comercial, ANAREDE [CEPE99], PacDyn [CEPE97] y ANATEM [CEPE01]. Estos simuladores comerciales están desarrollados en Fortran, y su principal característica es la resolución del problema de “flujo de potencia”, “estabilidad permanente” y “estabilidad transitoria” respectivamente.

En este capítulo también se presentan los resultados de la implementación del método computacional descrito en el capítulo 4 para la resolución del problema de la “expansión de la transmisión AC” aplicado en un sistema eléctrico cuya solución óptima utilizando el modelo DC es conocida, el sistema Garver [GARV70].

Los respectivos datos de topología; de los modelos, parámetros y controladores de las máquinas pertenecientes a los sistemas eléctricos utilizados en las simulaciones presentadas en este capítulo, pueden ser encontradas en el Apéndice A.

5.2. SISTEMA ELÉCTRICO WECC

El diagrama físico del sistema WECC es mostrado en la Figura 5.1. Se puede observar que este sistema eléctrico está compuesto de 6 subestaciones y 6 líneas de transmisión. Las subestaciones 1, 2 y 3 contienen generadores que se conectan al sistema por medio de transformadores en fase y son responsables por el abastecimiento de la demanda. Las subestaciones 5, 6 y 8 contienen centros de carga (posiblemente subestaciones de distribución) que totalizan una demanda de 315 MW y 115 MVAR.

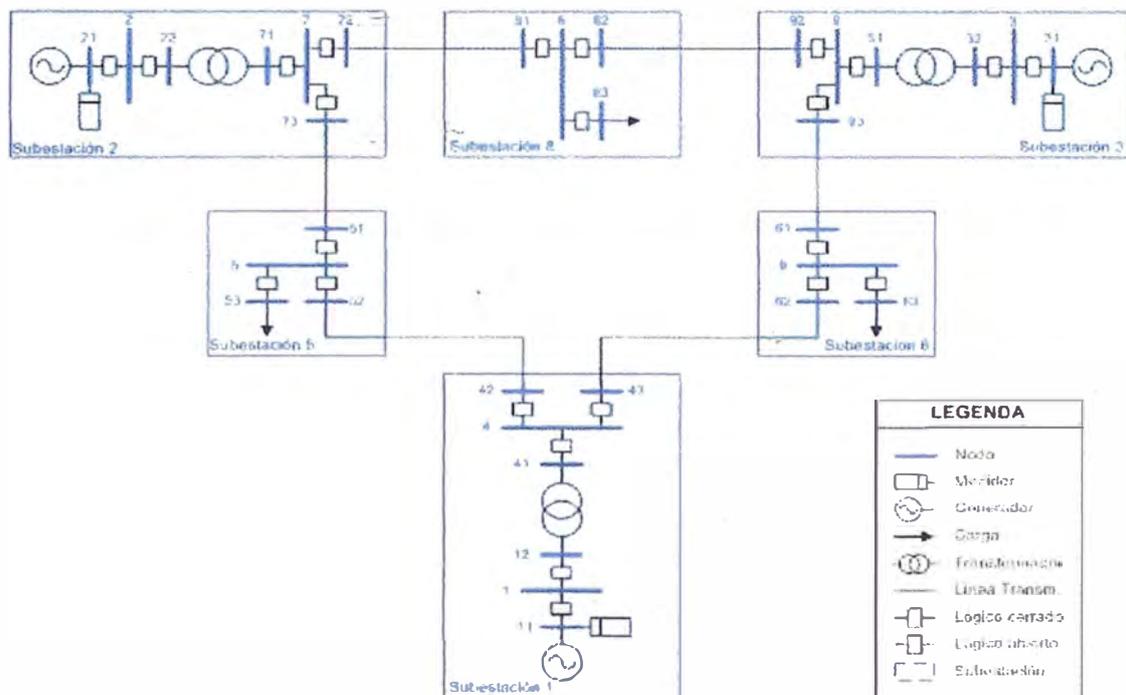


Figura 5.1: Diagrama físico del sistema eléctrico WECC.

Se puede observar también que todos los dispositivos lógicos se encuentran cerrados y que cada generador tiene asociado un medidor. Al encontrarse cerrados los dispositivos lógicos muchos de los nodos eléctricos pasarán a ser una única barra en el diagrama lógico. La presencia de los medidores es resaltada debido a que serán necesarios para almacenar (grabar) algún parámetro correspondiente a los generadores, cuando sean efectuadas las simulaciones para un periodo de tiempo.

5.2.1. FLUJO DE POTENCIA

El estado operativo del sistema WECC fue determinado efectuando la simulación de flujo de potencia utilizando el prototipo computacional SinergiaSEE, considerado errores máximos de 0.01 MW e 0.01 MVar. Los resultados (tensiones complejas de las barras), pueden ser observadas en la Figura 5.2.

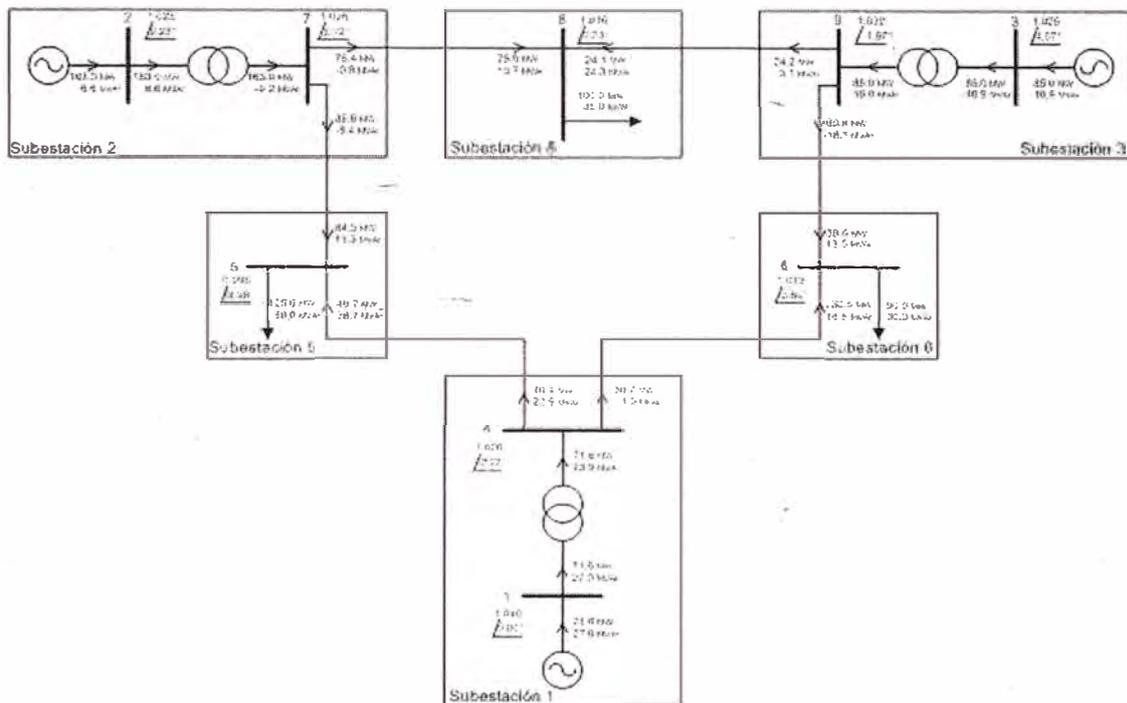


Figura 5.2: Resultado del flujo de carga obtenido con el SinergiaSEE– Sistema WECC.

Estos resultados son comparados con los obtenidos utilizando un programa comercial para cálculo de flujo de carga, ANAREDE. Las respectivas comparaciones son presentadas en las Figuras 5.3 y 5.4.

La comparación de los módulos de las tensiones es efectuada considerando 3 posiciones decimales, y la de los ángulos de las tensiones es efectuada considerando 2 posiciones decimales; debido a que son las mayores precisiones ofrecidas por el reporte del ANAREDE. Observando las Figuras 5.3 y 5.4, puede concluirse que el estado operativo conseguido con el SinergiaSEE es el mismo que el obtenido con el ANAREDE.

Módulo de las tensiones

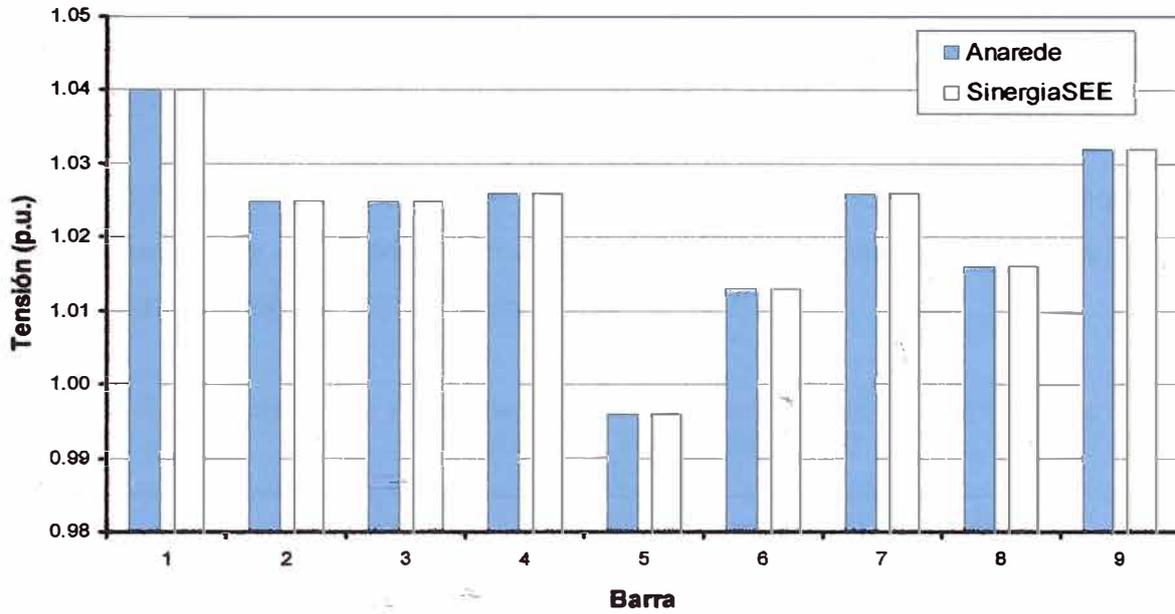


Figura 5.3: Comparación de los módulos de las tensiones obtenidos con el ANAREDE y con el SinergiaSEE - Sistema WECC.

Ángulo de las tensiones

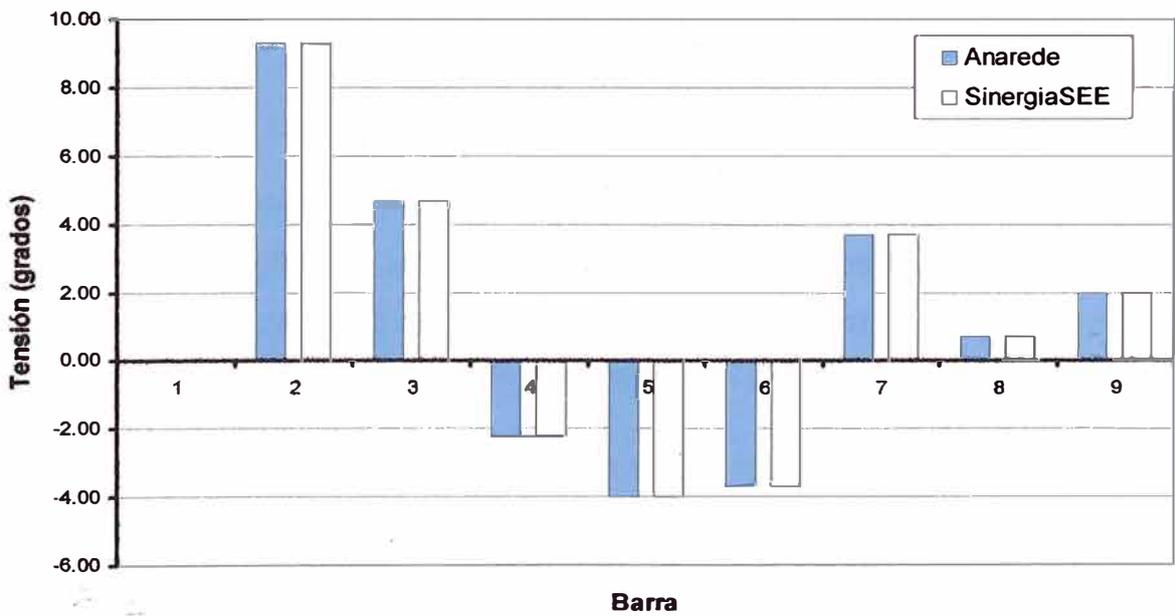


Figura 5.4: Comparación de los ángulos de las tensiones obtenidos con el ANAREDE y con el SinergiaSEE - Sistema WECC.

5.2.2. ESTABILIDAD PERMANENTE

La estabilidad del sistema WECC ante pequeñas perturbaciones fue determinada utilizando el prototipo computacional SinergiaSEE. Se efectuaron 3 simulaciones modificando los modelos de los generadores.

La primera simulación fue efectuada considerando sólo las máquinas correspondientes a cada uno de los generadores. Los autovalores calculados pueden ser observados en la Tabla 5.1.

Tabla 5.1: Autovalores y amortecimiento considerando sólo las máquinas en los generadores – Sistema WECC.

Autovalores			Frecuencia	Amortec.
Real	Imaginario	Módulo	(Hz.)	(%)
-0.3950	8.3785	8.3878	8.3878	4.71
-0.3950	-8.3785	8.3878	8.3878	4.71
-0.8579	11.9013	11.9322	11.9322	7.19
-0.8579	-11.9013	11.9322	11.9322	7.19
-30.7510	0.7712	30.7607	30.7607	99.97
-30.7510	-0.7712	30.7607	30.7607	99.97
-65.7435	0.0000	65.7435	65.7435	100.00
-37.4295	0.0000	37.4295	37.4295	100.00
-33.0775	0.0000	33.0775	33.0775	100.00
-17.9659	0.0000	17.9659	17.9659	100.00
-0.0816	0.0000	0.0816	0.0816	100.00
-0.1778	0.0000	0.1778	0.1778	100.00
-0.1544	0.0000	0.1544	0.1544	100.00
0.0000	0.0000	0.0000	0.0000	10000.00

Estos resultados son comparados con los obtenidos utilizando un programa comercial para el cálculo de estabilidad permanente, PacDyn. Las respectivas comparaciones son presentadas en la Figura 5.5.

En la Figura 5.5 puede observarse que todos los autovalores poseen parte real negativa, por lo tanto el sistema es estable para pequeñas perturbaciones. Las rectas de color rosado y celeste representan un amortecimiento de 5% y 10% respectivamente. Un zoom en la Figura 5.5 es presentada en la Figura 5.6. Observando las Figuras 5.5 y 5.6 pueden validarse los resultados obtenidos por el SinergiaSEE.

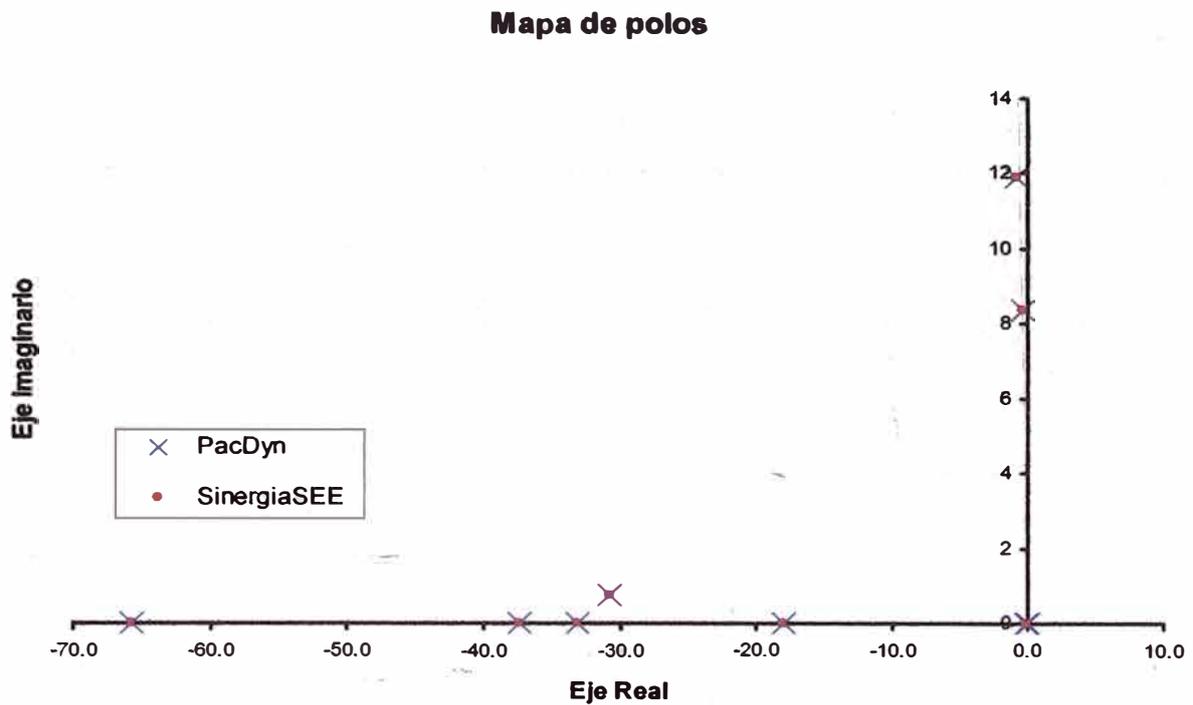


Figura 5.5: Comparación de los autovalores considerando sólo las máquinas en los generadores - Sistema WECC.

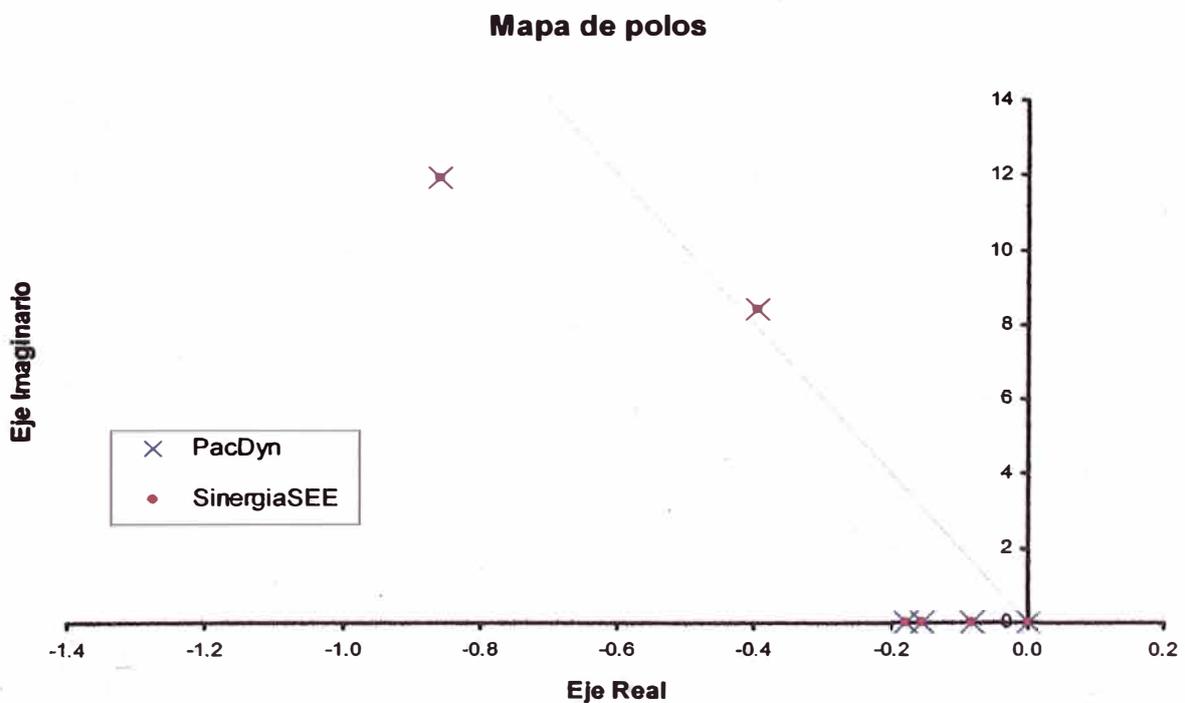


Figura 5.6: Zoom en la comparación de los autovalores considerando sólo las máquinas en los generadores - Sistema WECC.

Una segunda simulación de estabilidad permanente fue efectuada adicionando controladores de tensión (AVRs) en cada uno de los generadores. Los autovalores determinados con el prototipo computacional SinergiaSEE pueden ser observados en la Tabla 5.2.

Tabla 5.2: Autovalores y amortecimiento considerando las máquinas y AVRs en los generadores – Sistema WECC.

Real	Autovalores		Frecuencia (Hz.)	Amortec. (%)
	Imaginario	Módulo		
-0.4403	8.4165	8.4280	8.4280	5.22
-0.4403	-8.4165	8.4280	8.4280	5.22
-0.9070	11.9852	12.0194	12.0194	7.55
-0.9070	-11.9852	12.0194	12.0194	7.55
-32.3947	2.8116	32.5164	32.5164	99.63
-32.3947	-2.8116	32.5164	32.5164	99.63
-18.2142	0.3520	18.2176	18.2176	99.98
-18.2142	-0.3520	18.2176	18.2176	99.98
-65.7384	0.0000	65.7384	65.7384	100.00
-37.7114	0.0000	37.7114	37.7114	100.00
-33.8026	0.0000	33.8026	33.8026	100.00
-15.6447	0.0000	15.6447	15.6447	100.00
-18.8217	0.0000	18.8217	18.8217	100.00
-1.6153	0.0000	1.6153	1.6153	100.00
-0.8576	0.0000	0.8576	0.8576	100.00
-0.5342	0.0000	0.5342	0.5342	100.00
0.0000	0.0000	0.0000	0.0000	10000.00

Estos resultados son comparados con los obtenidos utilizando el PacDyn. Las respectivas comparaciones son presentadas en la Figura 5.7.

En la Figura 5.7 puede observarse que todos los autovalores poseen parte real negativa, por lo tanto el sistema es estable para pequeñas perturbaciones. Las rectas de color rosado y celeste representan un amortecimiento de 5% y 10% respectivamente. Un zoom en la Figura 5.7, es apreciado en la Figura 5.8. Observando las Figuras 5.7 y 5.8 puede concluirse que los autovalores conseguidos con el SinergiaSEE son los mismos que los obtenidos con el PacDyn.

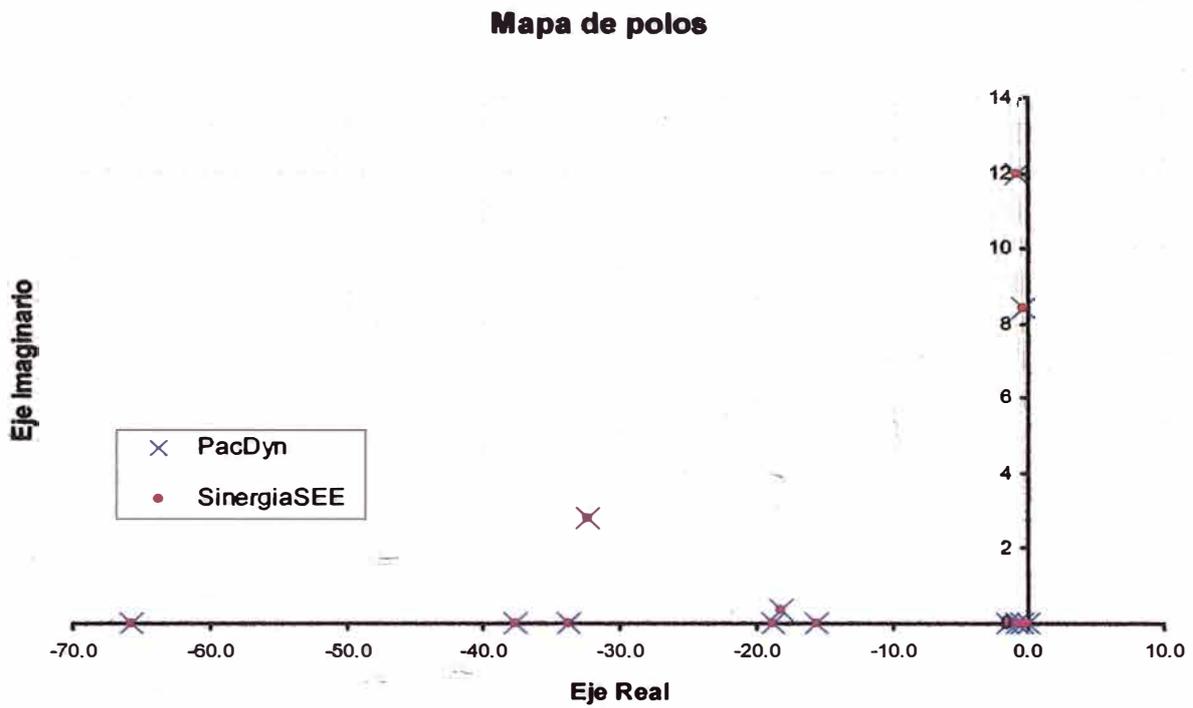


Figura 5.7: Comparación de los autovalores obtenidos considerando las máquinas y AVR en los generadores - Sistema WECC.

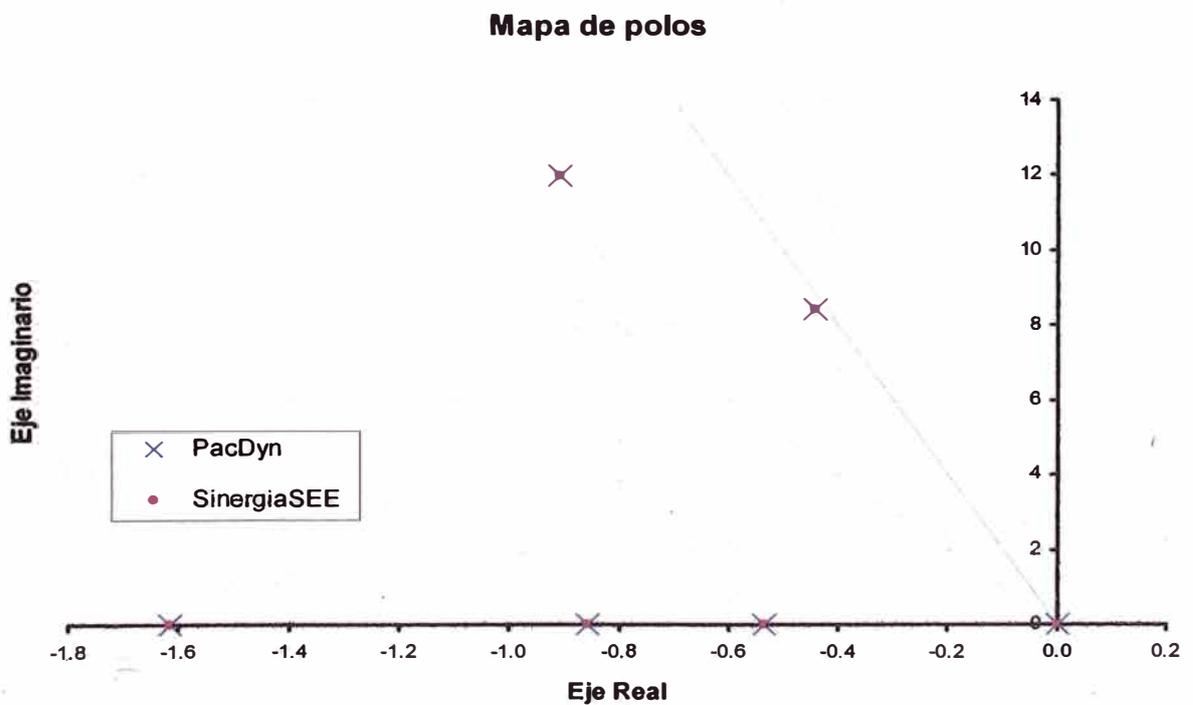


Figura 5.8: Zoom en la comparación de los autovalores obtenidos considerando las máquinas y AVR en los generadores - Sistema WECC.

Una tercera simulación de estabilidad permanente fue efectuada, adicionando controladores de potencia (PSSs) en cada uno de los generadores. Los autovalores determinados con el SinergiaSEE pueden ser observados en la Tabla 5.3.

Tabla 5.3: Autovalores y amortecimiento considerando las máquinas, AVR's y PSSs en los generadores – Sistema WECC.

Real	Autovalores		Frecuencia (Hz.)	Amortec. (%)
	Imaginario	Módulo		
-1.9275	14.5609	14.6880	14.6880	13.12
-1.9275	-14.5609	14.6880	14.6880	13.12
-1.9302	8.8672	9.0749	9.0749	21.27
-1.9302	-8.8672	9.0749	9.0749	21.27
-10.8303	9.9003	14.6735	14.6735	73.81
-10.8303	-9.9003	14.6735	14.6735	73.81
-0.9815	0.6877	1.1984	1.1984	81.90
-0.9815	-0.6877	1.1984	1.1984	81.90
-10.0795	6.3545	11.9154	11.9154	84.59
-10.0795	-6.3545	11.9154	11.9154	84.59
-41.2064	13.9494	43.5035	43.5035	94.72
-41.2064	-13.9494	43.5035	43.5035	94.72
-36.6614	12.0466	38.5899	38.5899	95.00
-36.6614	-12.0466	38.5899	38.5899	95.00
-4.4650	0.7779	4.5323	4.5323	98.52
-4.4650	-0.7779	4.5323	4.5323	98.52
-17.7916	0.4943	17.7985	17.7985	99.96
-17.7916	-0.4943	17.7985	17.7985	99.96
-65.6878	0.0000	65.6878	65.6878	100.00
-37.8362	0.0000	37.8362	37.8362	100.00
-31.1025	0.0000	31.1025	31.1025	100.00
-0.4905	0.0000	0.4905	0.4905	100.00
-1.0354	0.0000	1.0354	1.0354	100.00
-0.2109	0.0000	0.2109	0.2109	100.00
-0.2008	0.0000	0.2008	0.2008	100.00
0.0000	0.0000	0.0000	0.0000	10000.00

Estos resultados son comparados con los obtenidos utilizando el PacDyn. Las respectivas comparaciones son presentadas en la Figura 5.9.

En la Figura 5.9 puede observarse que todos los autovalores poseen parte real negativa, por lo tanto el sistema es estable para pequeñas perturbaciones. Observando las Figuras 5.5 y 5.6 pueden validarse los resultados obtenidos por el SinergiaSEE.

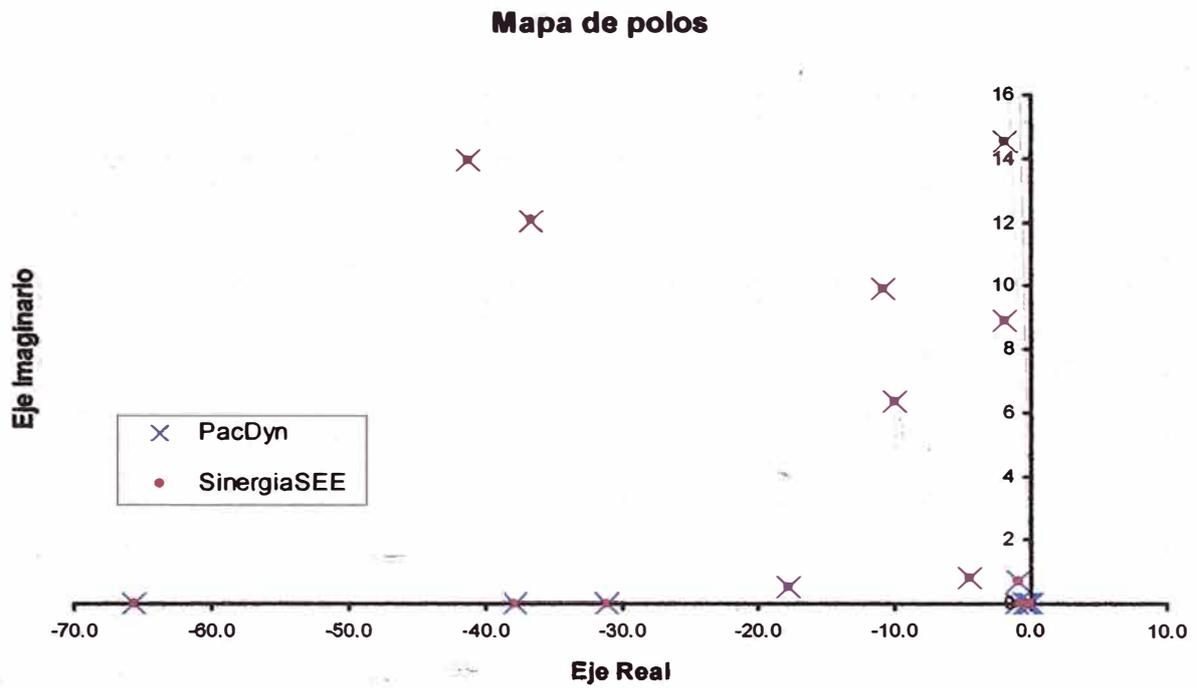


Figura 5.9: Comparación de los autovalores obtenidos considerando las máquinas, AVR y PSSs en los generadores - Sistema WECC.

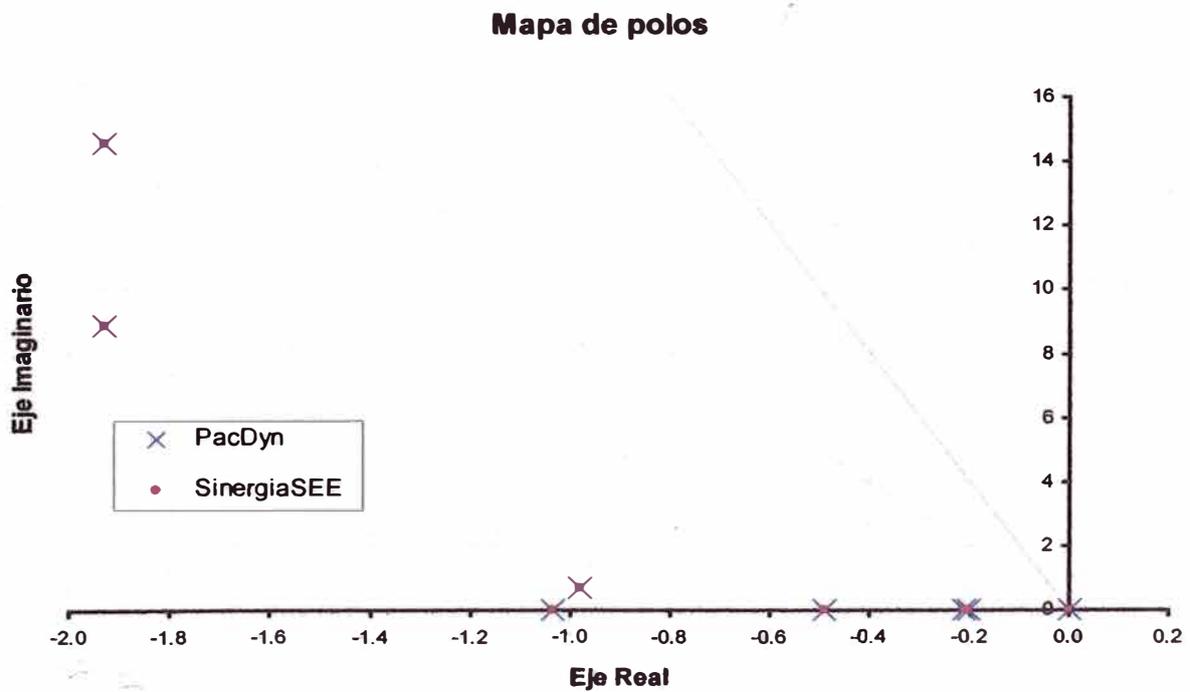


Figura 5.10: Zoom en la comparación de los autovalores obtenidos considerando las máquinas, AVR y PSSs en los generadores - Sistema WECC.

5.2.3. ESTABILIDAD TRANSITORIA

La estabilidad del sistema WECC ante un corto-circuito trifásico en la barra 7 en un tiempo igual de 0.500 s. y removido con la apertura de la línea 5 – 7 en un tiempo igual a 0.583 s. (gran perturbación), fue determinado utilizando el prototipo computacional SinergiaSEE. Se efectuaron 3 simulaciones modificando los modelos de los generadores.

La primera simulación fue efectuada considerando sólo las máquinas correspondientes a cada uno de los generadores. Los resultados (ángulo delta de los generadores) son comparados con los obtenidos utilizando un programa comercial para cálculo de estabilidad transitoria, ANATEM. Las respectivas comparaciones son presentadas en la Figura 5.11.

En la Figura 5.11 puede observarse que el sistema es inestable para la perturbación considerada, así como que los resultados obtenidos con el SinergiaSEE son los mismos que los obtenidos con el ANATEM.

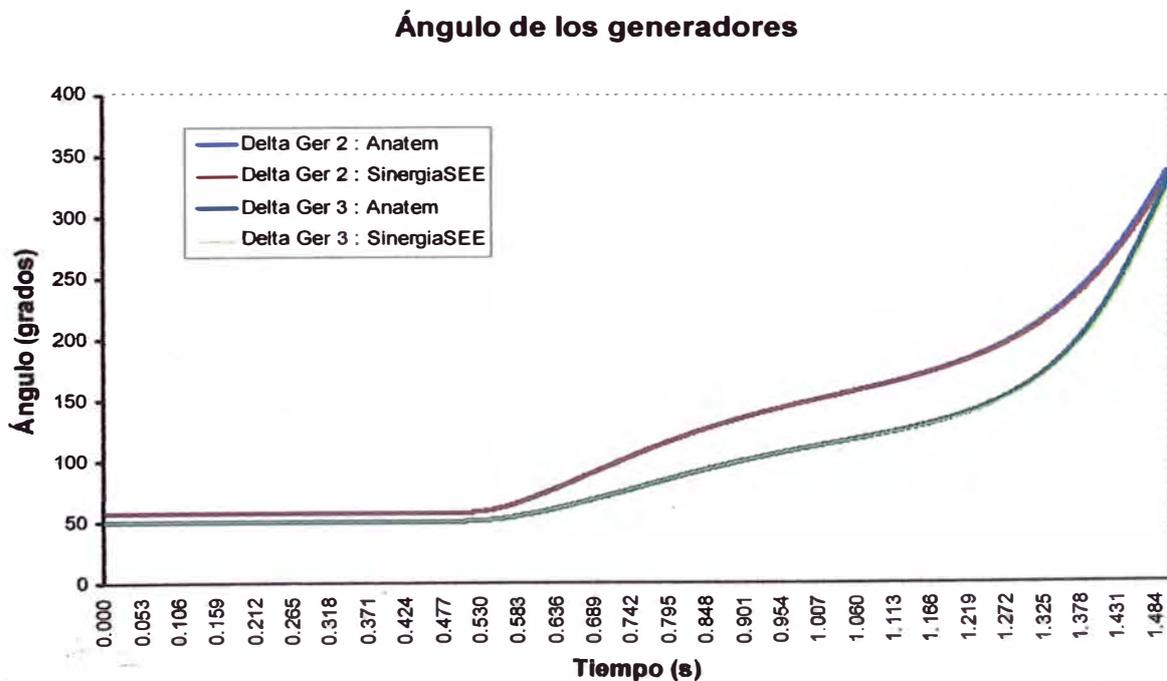


Figura 5.11: Comparación de los ángulos de los rotores de los generadores obtenidos considerando sólo las máquinas - Sistema WECC.

Una segunda simulación de estabilidad transitoria fue efectuada adicionando controladores de tensión (AVRs) en cada uno de los generadores. Los resultados obtenidos con el SinergiaSEE son comparados con los obtenidos con el ANATEM en la Figura 5.12.

En la Figura 5.12 puede observarse que el sistema es estable y poco amortecido para la perturbación considerada, así como que los resultados obtenidos con el SinergiaSEE son los mismos que los obtenidos con el ANATEM.

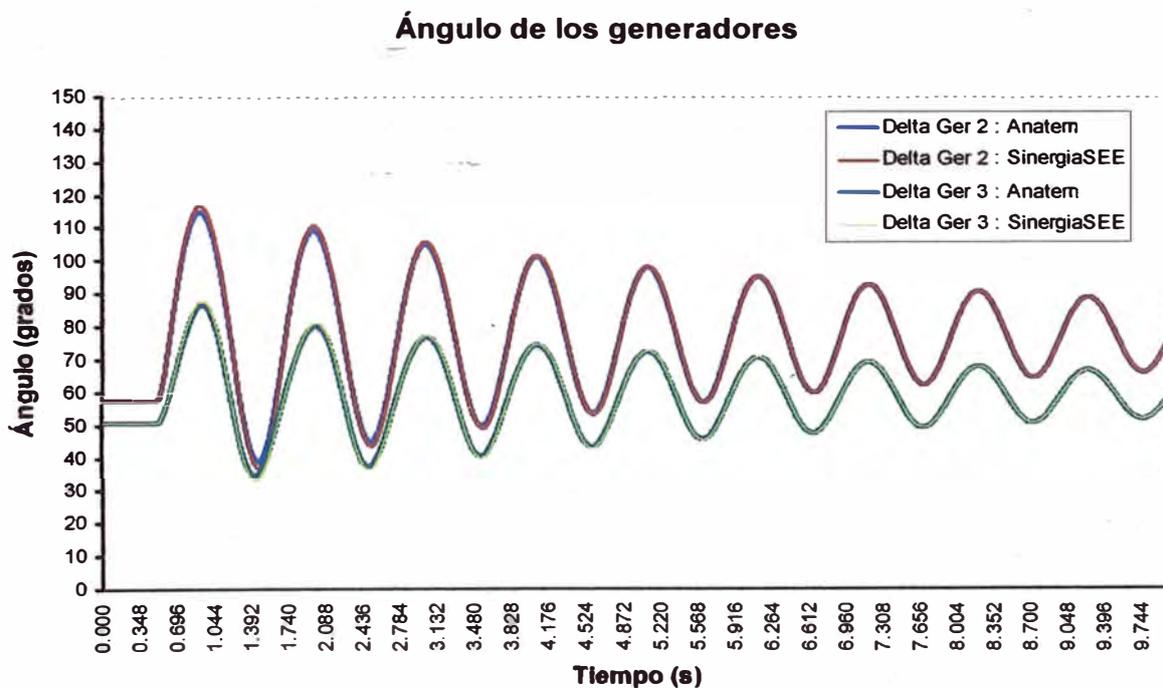


Figura 5.12: Comparación de los ángulos de los rotores de los generadores obtenidos considerando las máquinas y AVR's - Sistema WECC.

Una tercera simulación de estabilidad transitoria fue efectuada adicionando controladores de potencia (PSSs) en cada uno de los generadores. Los resultados obtenidos con el SinergiaSEE son comparados con los obtenidos con el ANATEM en la Figura 5.13.

En la Figura 5.13 puede observarse que el sistema es estable y bien amortecido para la perturbación considerada, así como que los resultados obtenidos con el SinergiaSEE son los mismos que los obtenidos con el ANATEM.

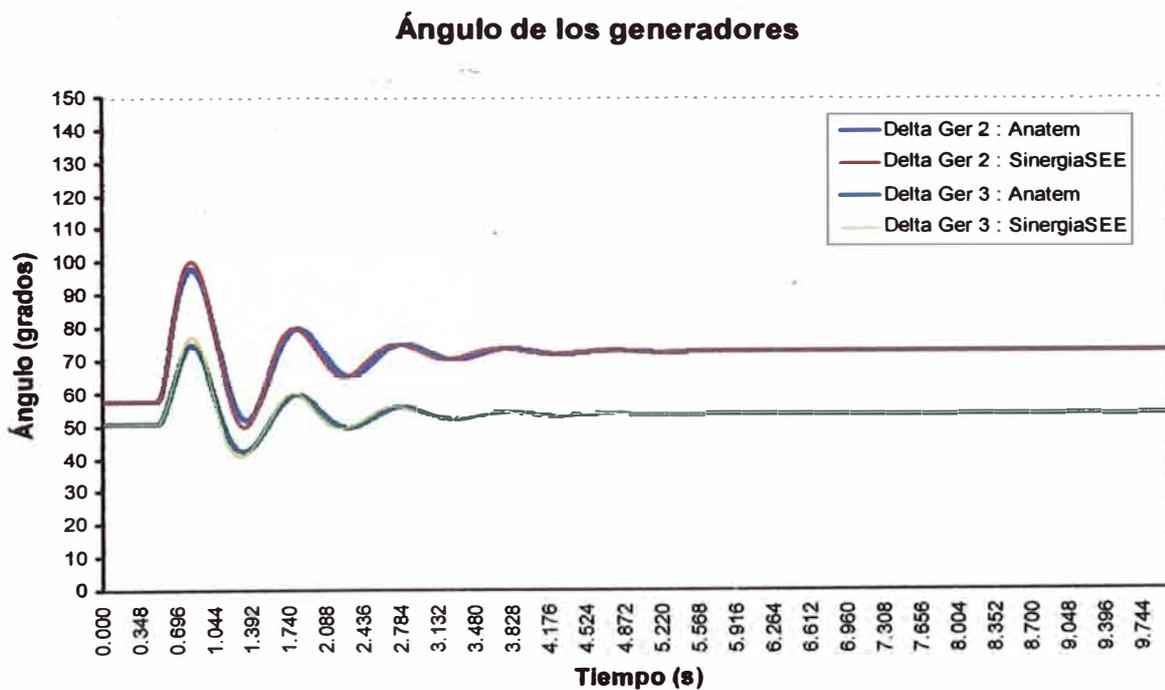


Figura 5.13: Comparación de los ángulos de los rotores de los generadores obtenidos considerando las máquinas, AVR y PSSs - Sistema WECC.

5.3. SISTEMA ELÉCTRICO GARVER

El diagrama físico del sistema Garver es mostrado en la Figura 5.14. Se puede observar que este sistema eléctrico está compuesto de 6 subestaciones, 6 líneas de transmisión (color negro) y 9 rutas para la construcción de futuras líneas de transmisión (color rojo). Las subestaciones 1, 3 y 6 contienen generadores que son responsables por abastecimiento de la demanda. Las subestaciones 1, 2, 3, 4 y 5 contienen centros de carga con demandas proyectadas para un horizonte futuro.

Este SEE admite la construcción de nuevas líneas de transmisión (ramos) paralelas a las ya existentes o en algunas de las rutas previamente establecidas. Las primeras son llamadas de refuerzos, mientras que las segundas son llamadas de adiciones. Se establece que cada uno de los refuerzos y adiciones no debe ser mayor que 8. En la Figura 5.14 puede observarse que existen 6 posibilidades de refuerzos y 9 de adiciones, totalizando 15 posibilidades de construcción de nuevos ramos, cada uno de los cuales varía entre 0 y 8.

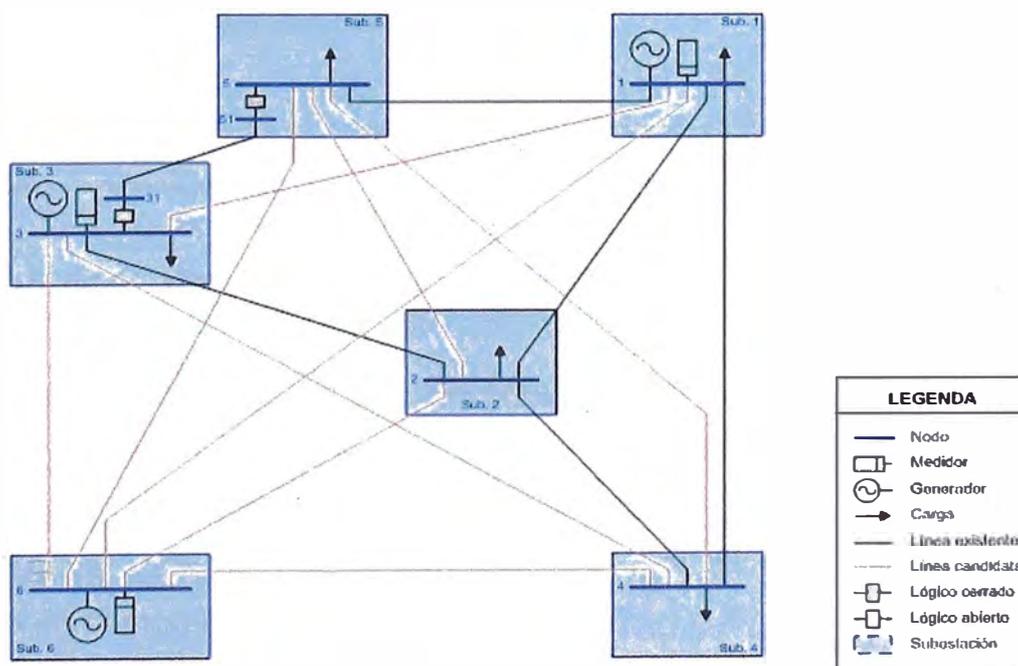


Figura 5.14: Diagrama físico del sistema eléctrico Garver.

La determinación de la configuración del sistema Garver que atienda la demanda futura (expansión de la transmisión) fue determinada utilizando una extensión del prototipo computacional SinergiaSEE, tal y como fue descrito en el capítulo 4. Se efectuaron 2 simulaciones modificando la cantidad de restricciones.

5.3.1. EXPANSIÓN DE LA TRANSMISIÓN DC

La primera simulación fue efectuada considerando sólo las restricciones de carácter estáticas (sobre tensiones en las barras y sobre cargas en los ramos).

En la Figura 5.15 puede observarse la representación del cromosoma correspondiente al algoritmo genético maestro. Cada cromosoma contiene 15 genes, cada uno de los cuales utiliza 4 bits para representar un número entero. Los cuatro primeros bits (el primer gen) del cromosoma indican el número de refuerzos / adiciones iguales al ramo 1 existente que serán construidos, los cuatro bits siguientes (el segundo gen) indican el número de refuerzos / adiciones iguales al ramo 2 existente que serán construidos, y así por delante hasta llegar al ramo 15.

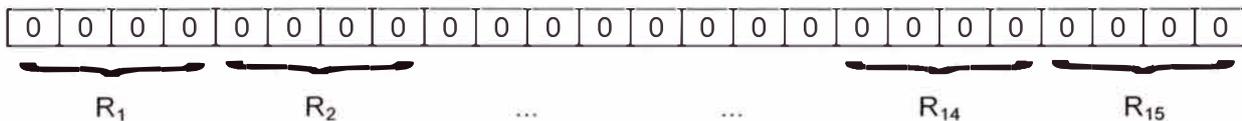


Figura 5.15: Representación del cromosoma correspondiente al algoritmo genético maestro - Sistema Garver.

Una de las formas para conseguir la configuración del sistema con mínimo costo que cumpla las restricciones estipuladas es evaluar todas las posibilidades existentes. En nuestro problema la cantidad de posibilidades existentes es igual a $2^{60} \approx 1\,152 \times 10^{15}$, lo que inviabiliza esta forma de solución en tiempo hábil.

En la Tabla 5.4 pueden observarse los parámetros del algoritmo genético maestro utilizado para las simulaciones presentadas en este subcapítulo.

Tabla 5.4: Parámetros del algoritmo genético maestro - Sistema Garver.

AG parámetros	Valor
Codificación	binaria
Tamaño de la población	100
Longitud del cromosoma	60
Tasa de cruzamiento	0.90
Tasa de mutación	0.10
Número máximo de generaciones	120
Tipo de selección	Ruleta
Tipo de cruzamiento	1 punto
Número de elitismo	1

En la Figura 5.16 pueden observarse los resultados a través del proceso de simulación del algoritmo genético maestro. El tiempo computacional requerido fue de 225.0 segundos (3 minutos 45 segundos).

Convergencia del Algoritmo Genético

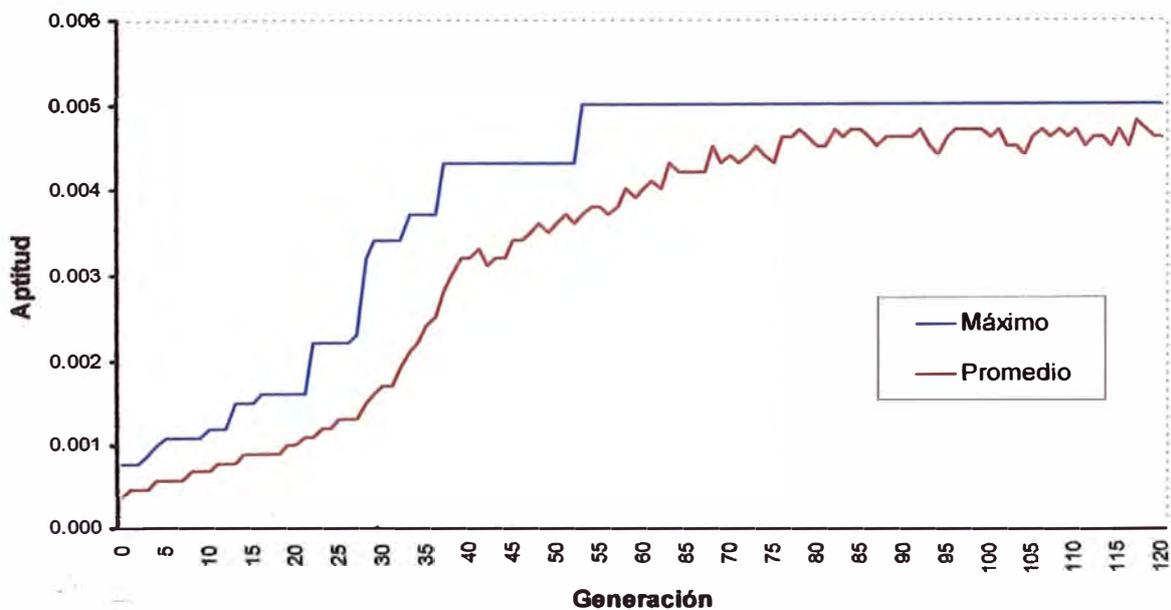


Figura 5.16: Convergencia del algoritmo genético maestro considerando restricciones estáticas – Sistema Garver.

El cromosoma resultado de esta primera simulación, en codificación entera, es mostrado en la Figura 5.17, a la cual le corresponde un costo asociado de 200.0 unidades. Este resultado representa la solución óptima global cuando son considerados sólo las restricciones estáticas. La topología (configuración) correspondiente al cromosoma resultante puede ser observada en la Figura 5.18.

0	0	0	0	0	1	0	0	0	4	0	0	0	2	0
R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₈	R ₁₀	R ₁₁	R ₁₂	R ₁₃	R ₁₄	R ₁₅

Figura 5.17: Cromosoma solución del algoritmo genético maestro considerando restricciones estáticas – Sistema Garver.

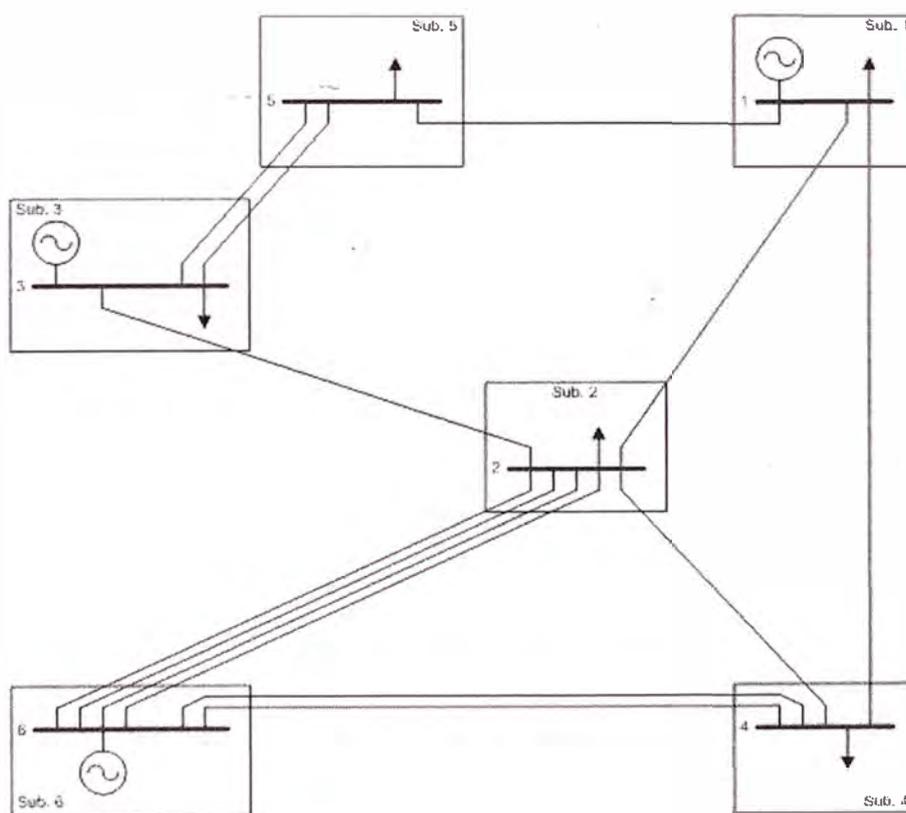


Figura 5.18: Configuración resultante considerando restricciones estáticas – Sistema Garver.

Un análisis de estabilidad permanente realizada con el SinergiaSEE para la configuración resultante es resumido en la Tabla 5.5 y en las Figura 5.19 y 5.20. Se observa que el sistema es estable y que el mínimo amortecimiento obtenido (7.12%) es menor al mínimo requerido (10%).

Tabla 5.5: Autovalores y amortecimiento de la configuración resultante considerando restricciones estáticas – Sistema Garver.

Autovalores			Frecuencia	Amortec.
Real	Imaginario	Módulo	(Hz.)	(%)
-0.5303	7.4263	7.4452	1.1819	7.12%
-0.5303	-7.4263	7.4452	-1.1819	7.12%
-1.7646	9.1685	9.3368	1.4592	18.90%
-1.7646	-9.1685	9.3368	-1.4592	18.90%
-3.9655	9.7449	10.5210	1.5509	37.69%
-3.9655	-9.7449	10.5210	-1.5509	37.69%
-6.3002	5.2194	8.1813	0.8307	77.01%
-6.3002	-5.2194	8.1813	-0.8307	77.01%
-33.3920	1.7705	33.4390	0.2818	99.86%
-33.3920	-1.7705	33.4390	-0.2818	99.86%
-54.4570	0.0000	54.4570	0.0000	100.00%
-45.4370	0.0000	45.4370	0.0000	100.00%
-38.5290	0.0000	38.5290	0.0000	100.00%
-24.6870	0.0000	24.6870	0.0000	100.00%
-14.4270	0.0000	14.4270	0.0000	100.00%
-4.9061	0.0000	4.9061	0.0000	100.00%
0.0000	0.0000	0.0000	0.0000	10000.00%

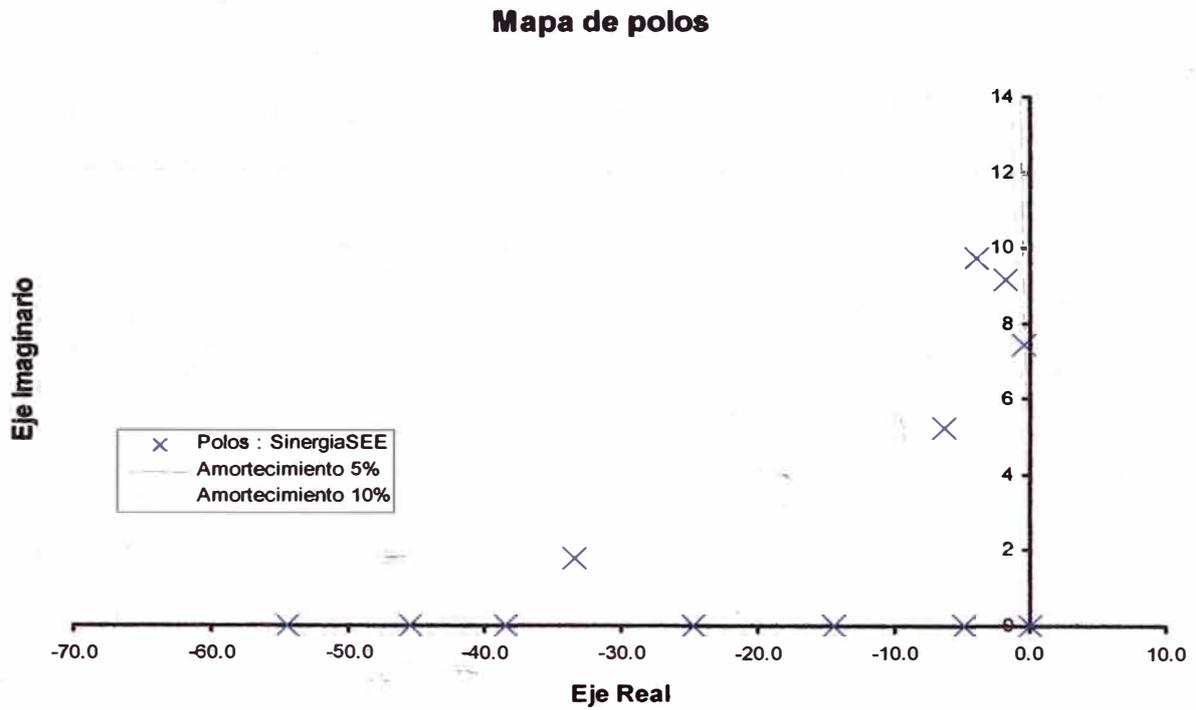


Figura 5.19: Mapa de polos de la configuración resultante considerando restricciones estáticas – Sistema Garver.

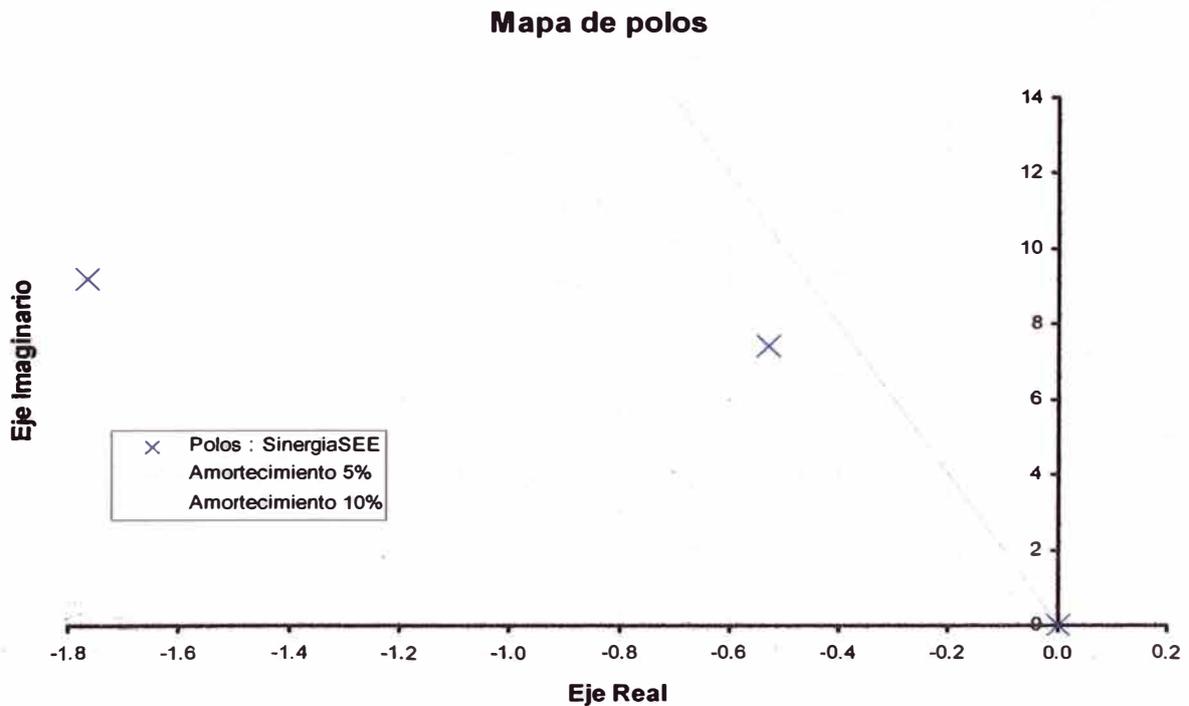


Figura 5.20: Zoom del mapa de polos de la configuración resultante considerando restricciones estáticas – Sistema Garver.

Un análisis de estabilidad transitoria del ángulo del rotor realizada con el SinergiaSEE considerando un corto-circuito trifásico en la barra 5 en un tiempo igual a 0.100 s. y liberada con la abertura del ramo 3-5 en un tiempo igual a 0.200 s., es resumido en la Figura 5.21. El tiempo de simulación considerado fue de 10 s. Se puede observar que el sistema es estable para la contingencia considerada.

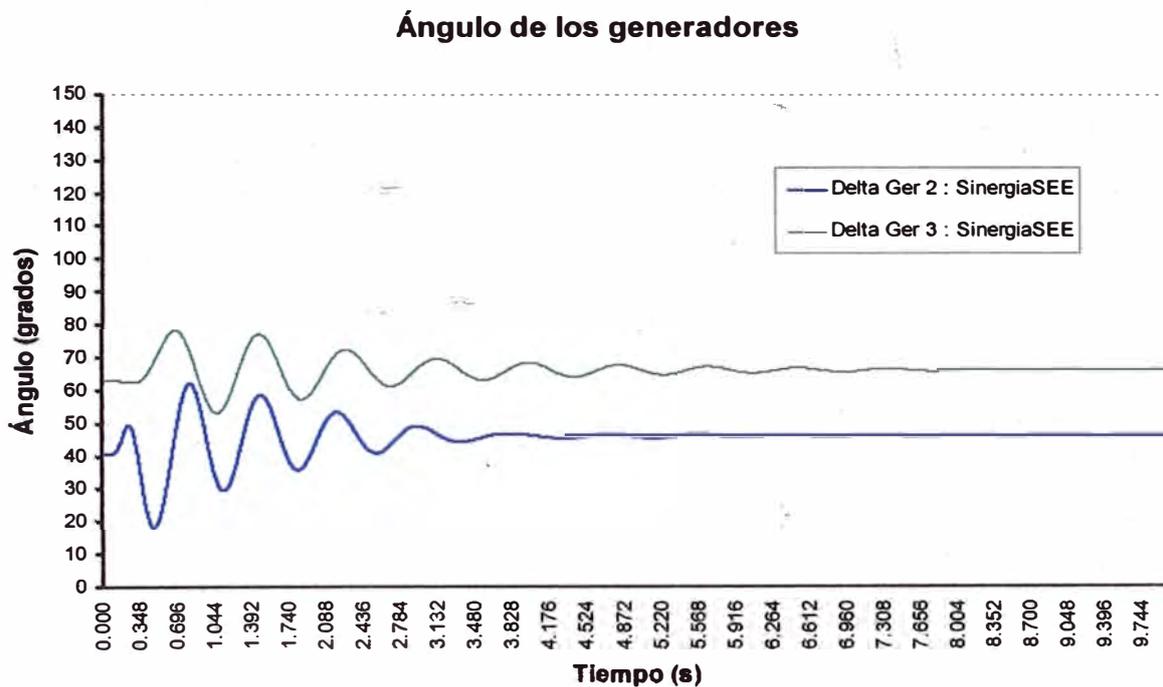


Figura 5.21: Ángulo del rotor de los generadores de la configuración resultante Garver considerando restricciones estáticas.

5.3.2. EXPANSIÓN DE LA TRANSMISIÓN AC

La inclusión de la restricción correspondiente a pequeñas perturbaciones en la formulación inicial permite al algoritmo genético proveer una solución con el mínimo amortecimiento requerido; mas también puede originar problemas de estabilidad para perturbaciones mayores; por lo que es necesario también adicionar la restricción correspondiente a grandes perturbaciones.

Una segunda simulación fue efectuada considerando estas restricciones dinámicas. Los parámetros del algoritmo genético maestro no fueron modificados y los parámetros del algoritmo genético esclavo son mostrado en la Tabla 5.6.

Tabla 5.6: Parámetros del algoritmo genético esclavo - Sistema Garver.

AG parámetros	Valor
Codificación	binaria
Tamaño de la población	60
Longitud del cromosoma	180
Tasa de cruzamiento	0.90
Tasa de mutación	0.10
Número máximo de generaciones	30
Tipo de selección	Ruleta
Tipo de cruzamiento	1 punto
Número de elitismo	1

En la Figura 5.22 se observa la representación del cromosoma utilizado en el algoritmo genético esclavo. Cada cromosoma contiene 9 genes, cada un de los cuales utiliza 20 bits para representar un número real. Los veinte primeros bits (el primer gen) del cromosoma representan el parámetro K del PSS del generador 1, los veinte siguientes bits (el segundo gen) representan el parámetro α del PSS del generador 1, los veinte siguientes bits (el tercer gen) representan el parámetro w del PSS del generador 1, y así por delante hasta llegar al tercer generador.

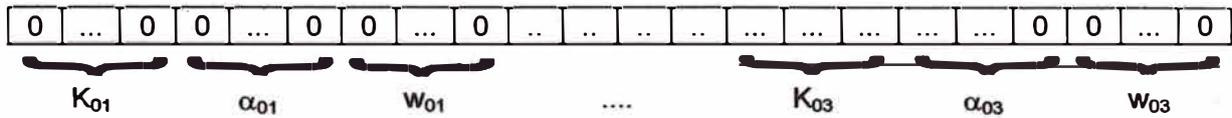


Figura 5.22: Representación del cromosoma correspondiente al algoritmo genético esclavo - Sistema Garver.

En la Figura 5.23 pueden observarse los resultados del proceso de simulación del algoritmo genético maestro correspondiente a la segunda simulación. El tiempo computacional requerido fue de 18 984.0 segundos (5 horas 16 minutos 24 segundos).

Convergencia del Algoritmo Genético

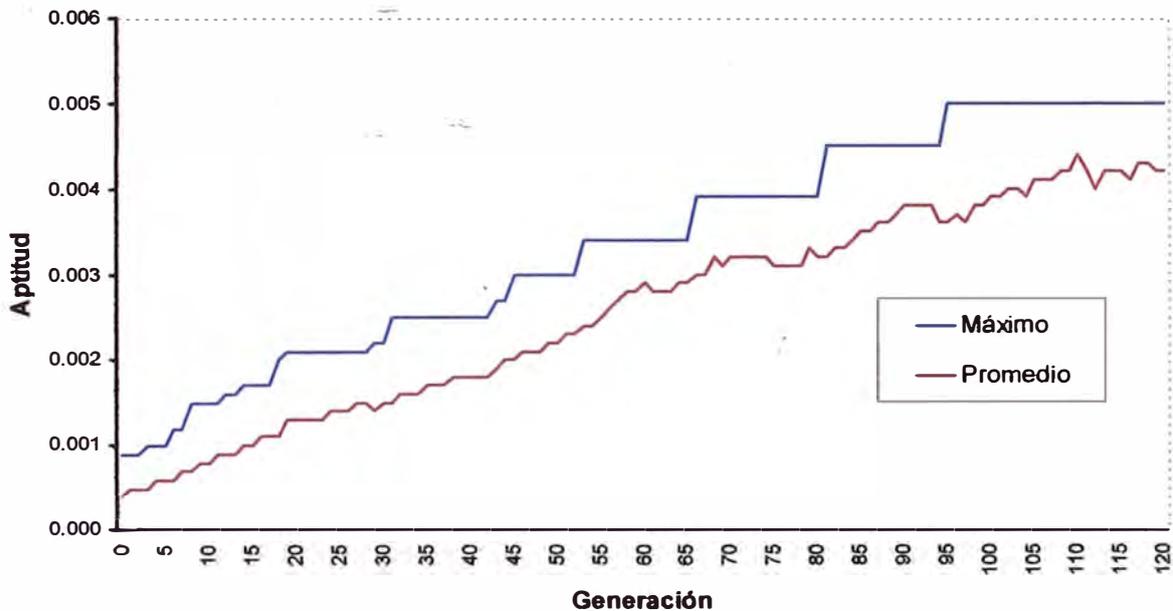


Figura 5.23: Convergencia del algoritmo genético maestro considerando restricciones estáticas e dinámicas - Sistema Garver.

El cromosoma maestro resultado de esta segunda simulación, en codificación entera, es mostrado en la Figura 5.24, a la cual le corresponde un costo asociado de 200.0. La topología correspondiente al cromosoma maestro resultante puede ser observada en la Figura 5.25.

0	0	0	0	0	1	0	0	0	4	0	0	0	2	0
R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₈	R ₁₀	R ₁₁	R ₁₂	R ₁₃	R ₁₄	R ₁₅

Figura 5.24: Cromosoma solución del algoritmo genético maestro considerando restricciones estáticas e dinámicas - Sistema Garver.

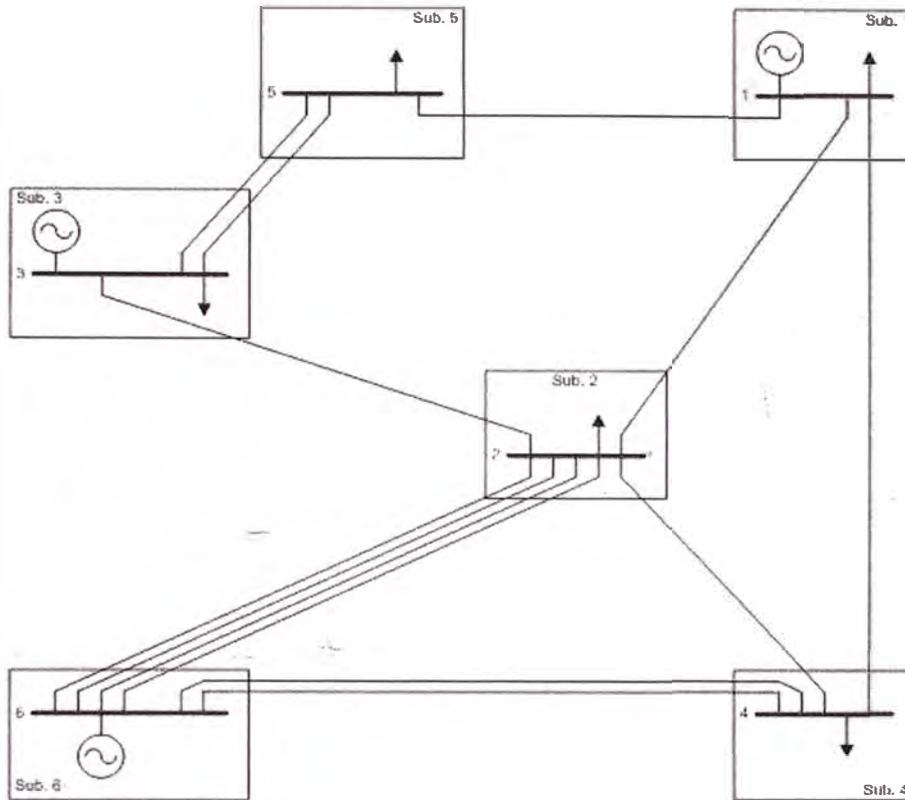


Figura 5.25: Configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver.

A cada cromosoma maestro, en caso de no atender la restricción de mínimo amortecimiento, le corresponde un cromosoma esclavo conteniendo los parámetros de los nuevos PSSs a ser instalados. Debido a que el costo considerado para cada PSS es de cero, la solución encontrada (200.0) aún corresponde a la solución óptima global. Un resumen de los parámetros de los PSSs obtenidos por el AG esclavo para el cromosoma resultado del AG maestro es mostrado en la Tabla 5.7.

Tabla 5.7: Resultados del algoritmo genético esclavo considerando restricciones estáticas e dinámicas - Sistema Garver.

Generador	Barra	K	α	w
1	1	5.6150	5.9191	10.0742
2	3	4.6687	6.8640	11.5314
3	6	28.8590	4.7333	11.3491

Un análisis de estabilidad permanente realizada con el SinergiaSEE para la configuración resultante, es resumida en la Tabla 5.8 y en las Figuras 5.26 y 5.27. Se observa que los autovalores son todos negativos, por lo que el sistema es estable, y que el mínimo amortecimiento existente es de 14.57% lo que sí atiende la restricción de mínimo amortecimiento (10%).

Tabla 5.8: Autovalores y amortecimiento de la configuración resultante considerando restricciones estáticas e dinámicas – Sistema Garver.

Autovalores		Módulo	Frecuencia (Hz.)	Amortec. (%)
Real	Imaginario			
-1.6298	11.0687	11.1880	11.1880	14.5700
-1.6298	-11.0687	11.1880	11.1880	14.5700
-5.1902	21.1112	21.7399	21.7399	23.8700
-5.1902	-21.1112	21.7399	21.7399	23.8700
-4.0299	14.8608	15.3975	15.3975	26.1700
-4.0299	-14.8608	15.3975	15.3975	26.1700
-2.7746	4.8728	5.6074	5.6074	49.4800
-2.7746	-4.8728	5.6074	5.6074	49.4800
-3.8312	5.2300	6.4832	6.4832	59.0900
-3.8312	-5.2300	6.4832	6.4832	59.0900
-52.9952	19.7127	56.5427	56.5427	93.7300
-52.9952	-19.7127	56.5427	56.5427	93.7300
-41.4481	15.0707	44.1029	44.1029	93.9800
-41.4481	-15.0707	44.1029	44.1029	93.9800
-35.7060	11.7723	37.5966	37.5966	94.9700
-35.7060	-11.7723	37.5966	37.5966	94.9700
-54.1885	0.0000	54.1885	54.1885	100.0000
-35.5545	0.0000	35.5545	35.5545	100.0000
-19.6888	0.0000	19.6888	19.6888	100.0000
-13.7591	0.0000	13.7591	13.7591	100.0000
-8.1000	0.0000	8.1000	8.1000	100.0000
-4.8753	0.0000	4.8753	4.8753	100.0000
-1.9939	0.0000	1.9939	1.9939	100.0000
-0.2026	0.0000	0.2026	0.2026	100.0000
-0.2009	0.0000	0.2009	0.2009	100.0000
0.0000	0.0000	0.0000	0.0000	10000.00

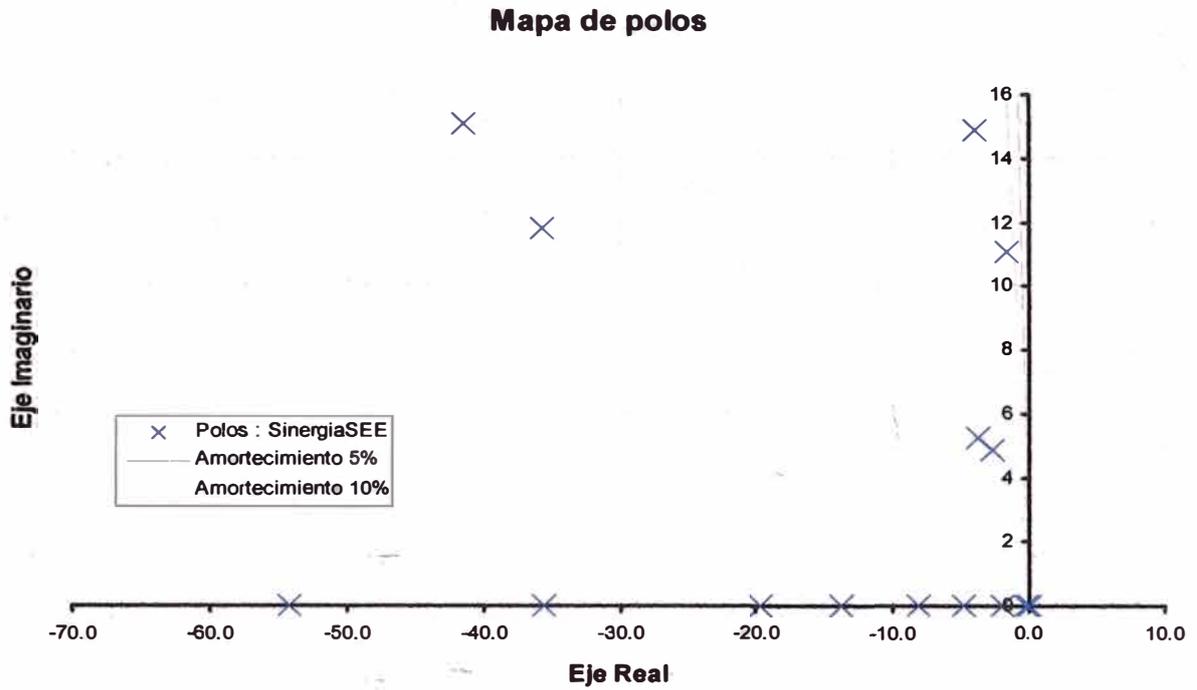


Figura 5.26: Mapa de polos de la configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver.

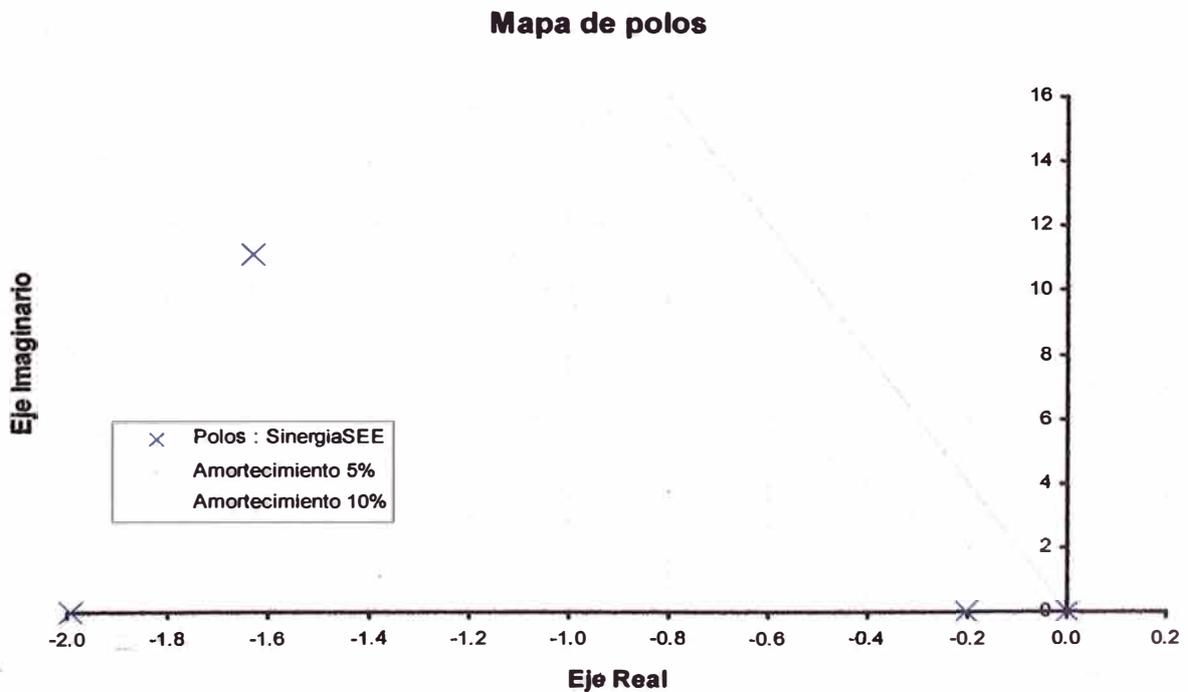


Figura 5.27: Zoom en el mapa de polos de la configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver.

Un análisis de estabilidad de ángulo de rotor (estabilidad transitoria) considerando un corto-circuito trifásico en la barra 5 en un tiempo igual a 0.100 s., la cual es liberada con la abertura del ramo 3-5 en un tiempo igual a 0.200 s. es resumido en la Figura 5.28. El tiempo de simulación considerado fue de 10 s. Se puede observar que el sistema es estable para la contingencia considerada.

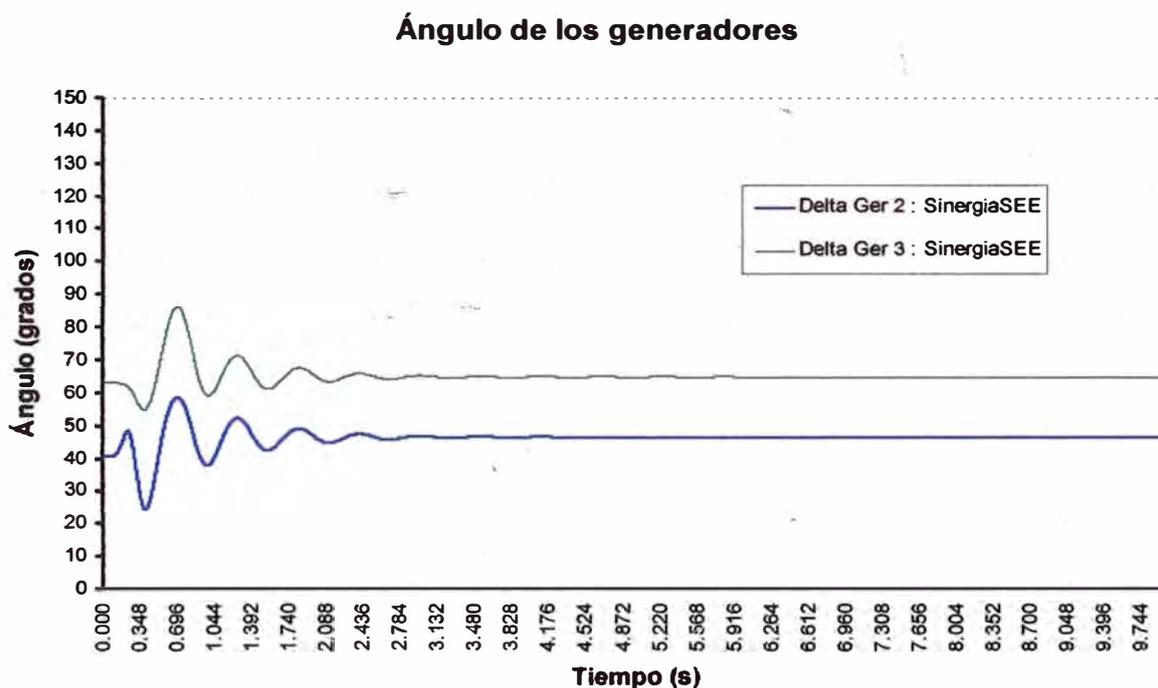


Figura 5.28: Ángulo del rotor de los generadores de la configuración resultante considerando restricciones estáticas e dinámicas – Sistema Garver.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Las principales conclusiones del presente trabajo son las siguientes:

1. La utilización del paradigma orientado a objetos se mostró adecuado para el diseño e implementación de una estructura jerárquica de clases que represente las más diversas características de los SEE, como parte fundamental de una base computacional para desarrollo de *software* para SEE.
2. La base computacional implementada se mostró adecuada para la implementación integrada de métodos computacionales correspondientes a los problemas (especialidades) tradicionales de “flujo de potencia”, “estabilidad permanente” y “estabilidad transitoria”, ampliamente utilizados en la simulación digital de los SEE.
3. La base computacional implementada se mostró lo suficientemente flexible y reutilizable para la implementación de un método computacional correspondiente a un problema mucho más complejo que los denominados tradicionales, la “expansión de la transmisión AC”.
4. La utilización de la especificación CORBA abre la posibilidad que los métodos implementados a partir de una única base computacional, una vez que hayan probado ser lo suficientemente robustos y confiables para sistemas de dimensión real, puedan integrarse con

otros métodos computacionales, históricamente aceptados y utilizados, independientemente de la plataforma y/o lenguaje de programación en la que hayan sido implementado.

RECOMENDACIONES

Diversos trabajos futuros pueden ser definidos a partir de los resultados obtenidos en el presente trabajo:

1. Mejorías del código fuente correspondientes a las herramientas matemáticas, específicamente la adición del cálculo de autovalores críticos, que permitiría el análisis de estabilidad permanente para SEE de dimensión real.
2. Mejorías del código fuente correspondientes a cada uno de los métodos computacionales denominados tradicionales, a fin de robustecerlas lo suficiente para su uso intensivo en SEE de dimensiones reales.
3. Utilización de la base computacional para la implementación de métodos computacionales correspondientes a especialidades tradicionales que no fueron implementadas dentro del presente trabajo.
4. Implementación del componente correspondiente a la interfaz gráfica, que permita la visualización y gerenciamiento gráfico de las diversas características de los SEE.

GLOSARIO DE TÉRMINOS

AC	<i>Alternating Current</i>
AG	Algoritmo genético
AOO	Análisis orientado a objetos.
AVR	<i>Automatic Voltage Regulator</i>
BT	Búsqueda Tabú
CASE	<i>Computer-Aided Software Engineering</i>
CMM	<i>Capability Maturity Model</i>
COES SINAC	Comité de Operación Económica del Sistema Interconectado Nacional.
CORBA	<i>Common Object Request Broker Architecture</i>
DC	<i>Direct Current</i>
DOO	Diseño Orientado a Objetos.
EPS	<i>Electric Power System</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure.</i>
GOV	<i>Governator</i>
IDL	<i>Interface Definition Language</i>
IS	Ingeniería de Software
MIMD	<i>Multiple Instruction Multiple Data</i>
MOO	Modelado orientado a objetos.
OMA	<i>Object Management Architecture</i>
OMG	<i>Object Management Group</i>
OMT	<i>Object Modeling Technique</i>
OO	Orientación a Objetos
OOP	<i>Object-Oriented Paradigm.</i>
OOSE	<i>Object Oriented Software Engineering</i>
ORB	<i>Object Request Broker</i>

PL	Programación lineal.
PLK	Primera ley de Kirchhoff.
PNLEM	Programación no lineal entero mixto.
POO	Paradigma de Orientación a Objetos
PSO	<i>Particle Swarm Optimization</i>
PSS	<i>Power System Stabilizer</i>
RUP	<i>Rational Unified Process</i>
SA	<i>Simulated annealing</i>
SCADA	<i>Supervisory Control And Data Acquisition</i>
SEE	Sistemas Eléctricos de Energía.
SEP	Sistemas Eléctricos de Potencia.
SIG	Sistemas de información de gestión.
SIMD	<i>Single Instruction Multiple Data</i>
SLK	Segunda ley de Kirchhoff.
UML	<i>Unified Modeling Language</i>
WECC	<i>Western Electricity Coordinating Council</i>
WSCC	<i>Western system coordinating council</i>

BIBLIOGRAFÍA

- [AART89] E. Aarts, J. Korst, "Simulated Annealing and Boltzmann Machines", *John Wiley & Sons*, 1989.
- [ABDE04] M. Abdel-Akher, M. Nor, A. H. Abdul Rashid, "Use Case Approach for developing Object Oriented Power System Studies: Object Oriented Three Phase Load Flow", *ICEEC International Conference on Electric, Electronic and Computer Engineering*, pp. 937-940, Sep. 2004.
- [AGOS02] M. N. Agostini, "Nova Filosofia para o projeto de software para sistemas de energia elétrica usando modelagem orientada a objetos", Tese de doutorado, UFSC, Florianópolis, Novembro 2002.
- [ANDE02] P. M. Anderson and A. A. Fouad, "Power System Control and Stability", Wiley-IEEE Press, 2002.
- [ASPE06] ASPEN SOFTWARE FOR ELECTRIC UTILITY: Power Flow, disponible en <http://www.aspeninc.com/> ; accesado en: 05 de diciembre del 2006.
- [BAUE72] F. L. Bauer, "Software Engineering", Information processing, 71, North Holland Publishing Co., Amsterdam, 1972.

- [BEND62] J.F. Benders, "Partitioning Procedures for Solving Mixed-Variables Programming Problems", *Numerische Mathematik*, V. 4, pp. 238-252, 1962.
- [BINA01] S. Binato, M. V. F. Pereira, S. Granville, "A new Benders Decomposition Approach to Solve Power Transmission Network Design Problems", *IEEE Transactions on Power Systems*, Vol. 16, pp. 235-240, May 2001.
- [BOOC97] G. Booch, "Object-Oriented Analysis and Design With Applications"; Addison Wesley, 1998.
- [BOOC98] G. Booch, J. Rumbaugh, I. Jacobson, "The Unified Modeling Language: User Guide", Reading: Addison Wesley, 1998.
- [BOOC99] G. Booch, J. Rumbaugh, I. Jacobson, "The Unified Software Development Process", Reading: Addison Wesley, 1999.
- [BOUZ97] M. Bouzeghoup, G. Gardarin, P. Valduriez; "Object Technology: Concepts and Methods"; Boston International Thomson Computer Press, 1997.
- [BOYL04] R. L. Boylestad, "Introducción a Análisis de Circuitos Eléctricos"; 10 ed., Prentice Hall, 2004.
- [BRAD99] M. E. Bradley, M. J. Bushnell, S. I. Maclean, "Object-Oriented creation of fault sequences for on-line transient stability analysis", *PSCC Power Systems Computation Conference*, Jun 1999.
- [CADE06] CADENCE DESIGN SYSTEM: PSpice Download, disponible en: http://www.cadence.com/products/si_pk_bd/pspice.aspx ; accesado en 05 de diciembre del 2006.

- [CEPE97] CEPEL / PacDyn. Análise e Controle de Estabilidade de Pequeno Sinal V4.1 – May/97. Manual do usuário. Maio 1997.
- [CEPE99] CEPEL / ANAREDE. Programa de Análise de Redes V07-08/99. Manual do usuário. Agosto 1999.
- [CEPE01] CEPEL / ANATEM. Programa de Análise de Transitórios Eletromecânicos V09-12/01. Manual do usuário. Dezembro 2001.
- [CHOW97] J. Chow, "Power System Toolbox Version 2.0: Dynamic Tutorial and Functions", Cherry Tree Scientific Software, RR-5 Colborne, Ontario. 1997.
- [COUL00] G.Coulouris, J. Dollimore and T. Kindberg, "Distributed Systems- Concepts and Design", Addison-Wesley, 3rd Ed., 2000.
- [CYME06] CYME INTERNATIONAL: Power Systems Analysis Framework; disponible en <http://www.cyme.com/software/psaf/> ; accesado en 05 de diciembre del 2006.
- [DEMA79] T. De Marco, "Structured Analysis and System Specification", Prentice Hall, 1979.
- [DOMM68] H. W.Dommel, W.F.Tinney, "Optimal Power Flow Solutions", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-87, No. 10, pp. 1866-1876, Oct. 1968.
- [DOMM69] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transient in Single and Multiphase Networks", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-88, No 4, pp 388 – 399, 1969.

- [EDSA06] EDSA MICROCORPORATION: Power Analytic Solutions; disponible en <http://www.edsa.com/productsServices/index.html> ; accesado en 05 de diciembre del 2006.
- [ETAP06] ETAP SOFTWARE: Power Systems Simulator; disponible en <http://www.etap.com/powersimulator.htm> ; accesado en 05 de diciembre del 2006.
- [ESCO04] Escobar A. H. Gallego R. A. Romero R. "Multistage and Coordinated Planning of the Expansion of Transmission Systems". *IEEE Transactions on Power Systems*. Vol. 19. No. 2. pp. 735-744. May 2004.
- [FARI05] H.Faria Jr., S. Binato, M. G. C. Resende, D. M. Falcão, "Power Transmission Network Design by Greedy Randomized Adaptive Path Relinking", *IEEE Transactions on Power Systems*, Vol. 20, No. 1, Feb. 2005.
- [FEO95] T. A. Feo, M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization*, Vol. 6, pp. 109-133, 1995.
- [FLIN92] D. G. Flinn, R. C. Dugan, "A Database for Diverse Power System Simulation Applications", *IEEE Transactions on Power Systems*, vol. 7, No. 2, pp. 784-790, May. 1992.
- [FOLE93] M. Foley, A. Bose, W. Mitchell, A. Faustini, "An Object Based Graphical User Interface for Power Systems", *IEEE Transactions on Power Systems*, Vol. 8, No. 1, pp. 97-104, Nov. 93.
- [FOLE95] M. Foley, A. Bose, "Object-Oriented on-line Network Analysis", *IEEE Transactions on Power Systems*, Vol. 10, No. 1, pp. 125-132, Feb. 1995.

- [FUER98] C. R. Fuerte-Esquivel, E. Acha, S. G. Tan, J. J. Rico, "Efficient Object Oriented Power Systems Software for the Analysis of Large-Scale Networks containing FACTS-Controlled Branches", *IEEE Transactions on Power Systems*, Vol. 13, No. 2, pp. 464-472, May. 1998.
- [GAIN96] Z. L. Gaing, C. N. Lu, B. S. Chang, C. L. Cheng, "An Object-Oriented Approach for Implementing Power System Restoration Package", *IEEE Transactions on Power Systems*, Vol. 11, No. 1, pp. 483-489, Feb. 1996.
- [GALL97] R.A. Gallego, "Planejamento a Longo Prazo de Sistemas de Transmissão Usando Técnicas de Otimização Combinatorial", *Tese de doutorado*, UNICAMP, São Paulo, 1997.
- [GANN89] G. Gannet, "Handbook of Algorithms and Data Structures", Addison Wesley 2 ed., 1989.
- [GARV70] L. L. Garver, "Transmission Network Estimation Using Linear Programming", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 11, pp. 1688-1697, Nov. 1970.
- [GLOV89] F. Glover, "Tabu Search" – Part I, *ORSA Journal Computing*, V. 1, N. 3, Summer 1989.
- [GOLD89] D. E. Goldberg, "Genetic Algorithms in Search. Optimization and Machine Learning", *Addison-Wesley Text*, 1989.
- [HAND98] E. Handschin, M. Heine, D. Konig, T. Nikodem, T. Seibt, R. Palma, "Object-Oriented Software Engineering for Transmission Planning in Open Access Schemes", *IEEE Transactions on Power Systems*, Vol. 13, No.1, pp. 94-100, Feb. 1998.
- [HAVA94] B. Havavik, A. T. Holen, "Power System Modelling and Sparse Matrix Operations using Object-Oriented Programming", *IEEE*

Transactions on Power Systems, Vol. 9, No. 2, pp. 1045-1051, May 1994.

- [IEEE90] Computer Society of the IEEE, "IEEE Std 610.12-1990: IEEE Standard glossary of software engineering terminology", Computer Society Press, 1990.
- [ILIC98] D. M. Ilic, F. D. Galiana, L. H. Fink, "Power System Restructuring Engineering and Economics", Kluwer International Series in Engineering and Computer Science, 1998.
- [KUND04] P. Kundur, "Definition and classification of power system stability IEEE / CIGRE joint task force on stability terms and definitions", *IEEE Transactions on Power Systems*, Vol. 19, No. 3, pp.1387-1401, Aug. 2004.
- [LATO03] G. Latorre, R. D. Cruz, J. M. Areiza, A. Villegas, "Classification of Publications and Models on Transmission Expansion Planning", *IEEE Transactions on Power Systems*, Vol. 18, No. 2, May 2003.
- [LANG02] U. Lang, R. Schreiner, "Developing Secure Distributed System with CORBA", Artech House Computing Library, Boston, London, 2002.
- [LARS04] M. Larsson, "ObjectStab – An educational tool for power systems stability studies", *IEEE Transactions on Power Systems*, Vol. 19, No. 1, pp. 56-63, Feb. 2004.
- [LCE92] Decreto ley DL25844, Ley de Concesiones Eléctricas, Lima, Perú, 19 de noviembre de 1992.
- [LU02] F. Y. Lu, S. Chen, "Using Component Technology in Power System Simulations", Power Engineering Society Winter Meeting, vol. 1. pp: 684-689, Jan 2002.

- [MANI06] Welcome to PSCAD, disponible en http://pscad.com/site.pl?cmd=display_page&display_page=500, accesado en 05 de diciembre del 2006.
- [MANZ99] A. Manzoni, A. S. Silva, I. C. Decker, "Power Systems Dynamics Simulation using Object-Oriented Programming", *IEEE Transactions on Power Systems*, Vol. 14, No. 1, pp. 249-255, Feb. 1999.
- [MANZ05] A. Manzoni, "Desenvolvimento de um Sistema Computacional orientado a objetos para Sistemas Elétricos de Potência: Aplicação a Simulação Rápida e Análise da Estabilidade de Tensão", *Tese de doutorado*, COPPE/UFRJ, Rio de Janeiro, Março 2005.
- [MAHS97] J. Mahseredjian, F. Alvarado, "Creating an Electromagnetic Transient Program in Matlab: MatEMTP", *IEEE Transactions on Power Delivery*, Vol. 12, No. 1, pp. 380-388, 1997.
- [MEDI04] A. Medina, A. Ramos-Paz, R. Mora-Juarez, C. R. Fuerte-Esquivel, "Object oriented programming techniques applied to conventional and fast time domain algorithms for the steady state solution in nonlinear electric networks", *IEEE Power Engineering Society General Meeting*, Vol. 1, pp. 342-346, Jun. 2004.
- [MICH96] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer, 1996.
- [MILA06] DR FEDERICO MILANO: Power Systems Analysis Toolbox, disponible en <http://www.power.uwaterloo.ca/~fmilano/psat.htm>, accesado en 05 de diciembre del 2006.
- [MILL72] H. D. Mills, Technical Report FSC7 1-60-12 IBM Corp.. Federal Systems Divisions, Gaithersburg, Maryland, 1972.

- [MOHA04] K. Mohamed Nor, H. Mokhlis, T. A. Gani, "Reusability Techniques in Load-Flow Analysis Computer Program", *IEEE Transactions on Power Systems*, Vol. 19, No. 4, pp. 1754-1762, Nov. 2004.
- [MONT82] A. Monticelli, A. Santos Jr. M. V. F. Pereira, S. Cunha, J. G. Praça, B. Park, "Interactive Transmission Network Planning Using a Least-Effort Criterion", *IEEE Trans. Power App. Systems*, Vol. 101, No. 10, pp. 3919-3925, Oct. 1982.
- [MONT83] A. Monticelli, "Fluxo de Carga em Redes de Energia Elétrica", Edgard Blucher, São Paulo, 1983.
- [NEYE90] A. F. Neyer, F. F. Wu, K. Mhof, "Object-Oriented Programming for Flexible Software: Example of a Load Flow", *IEEE Transactions on Power Systems*, Vol. 5, No. 3, pp. 689-696, Aug. 1990.
- [NWAN02] C. Nwankpa, "Voltage Stability Toolbox version 2", Center for Electric Power Engineering, Drexel University, 2002.
- [OMG04] OBJECT MANAGEMENT GROUP: Document Access Page disponible en <http://www.omg.org/technology/documents/>, accesado en 05 de diciembre del 2006.
- [OLIV04] Oliveira S. "Metas-heurísticas aplicadas ao Planejamento da Expansão da Transmissão de Energia Elétrica em Ambiente de Processamento Distribuído", *Tese de doutorado*, UNICAMP. Campinas. 2004.
- [PALM03] R. B. Palma, "Object oriented platform for an integrated analysis of energy and transportation networks", *IEEE Transactions on Power Systems*, Vol. 18, No. 3, pp. 1062-1069, Aug 2003.

- [PAJA02] P. Pajan, V. L. Paucar, "Fuzzy power flow: considerations and application to the planning and operation of a real power system", *Proceedings on Power System Technology, PowerCon 2002*, Vol. 1, pp. 433-437, Oct. 2002.
- [PAJA06] P. Paján, "Análisis de riesgo y su impacto en el desarrollo de los sistemas de transmisión", *Tesis de Maestría*, UNI, Lima, 2006.
- [PAND00] S. Pandit, S. A. Soman, S. A. Khaparde, "Object-Oriented Design for Power System Applications", *IEEE Computer Applications in Power*, pp. 43-47, Oct. 2000.
- [PAND01a] S. Pandit, S. A. Soman, S. A. Khaparde, "Object-Oriented Network Topology Processor", *IEEE Computer Applications in Power*, pp. 42-46, Apr. 2001.
- [PAND01b] S. Pandit, S. A. Soman, S. A. Khaparde, "Design of Generic Direct Sparse Linear System Solver in C++ for Power System Analysis", *IEEE Transactions on Power Systems*, Vol. 16, No. 4, pp. 647-652, Nov. 2001.
- [PAUC98] V. L. Paucar, "Métodos computacionais baseados na função de energia para análise direta da estabilidade transitória de sistemas de energia elétrica", *Tese de doutorado*, UNICAMP, São Paulo, 1998.
- [PERE85] M. V. F. Pereira, L.M.V.G. Pinto, "Application of Sensitive Analysis of Load Supplying Capability to Interactive Transmission Expansion Planning", *IEEE Trans. Power App. Systems*, Vol. PAS-104, No. 2, Feb. 1985.
- [PHIL96] N. B. P. Phillips, J. O. Gann, M. R. Irving, "The Simian Architecture – An Object Orientated Framework for Integrated Power System Modeling Analysis and Control", *Power System*

Control and Management, Conference Publication No. 421, IEE, pp. 148-153, Apr. 1996.

- [POWE06] POWERTECH: Dynamic Security Analysis Power Tools, disponible en: <http://www.dsapowertools.com/> ; accesado en 05 de diciembre del 2006.
- [PRES02] R. S. Pressman, "Ingeniería del software: Un enfoque práctico", 5ta edición, Mc Graw Hill, Madrid, 2002.
- [PTI06] SIEMENS PTI: Power System Simulator for Engineering, disponible en <http://www.pti-us.com/pti/software/psse/index.cfm> , accesado en 05 de diciembre del 2006.
- [RIDE06] M. J. Rider, "Planejamento da expansão de sistemas de transmissão usando os modelos CC – CA e técnicas de programação não - linear", *Tese de doutorado*, UNICAMP, Campinas, 2006.
- [RLCE93] Decreto Supremo DS009-93-EM, Reglamento de la Ley de Concesiones Eléctricas, Lima, Perú, 25 de Febrero de 1993.
- [ROME96] R. Romero, R. A. Gallego, A. Monticelli, "Transmission System Expansion Planning by Simulated Annealing", *IEEE Transactions on Power Systems*, Vol. 11, pp. 364-369, Feb. 1996.
- [RODR05] J. I. Rodriguez, "Integração de metodologias para avaliação do custo da segurança dinâmica em mercados elétricos", *Tese de Mestrado*, UFMA, São Luis, Fevereiro 2005.
- [RUMB94] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, "Modelagem e Projetos baseados em Objetos", Ed. Campus, 1994.

- [SCHW70] F. C. Schweppe, "Power System State Estimation, Part III: Implementation", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, pp. 130-135, 1970.
- [SCHO03] K. Schoder, A. Feliachi, A. Hasanovic, "A power analysis toolbox for Matlab / Simulink", *IEEE Transactions on Power Systems*, Vol. 18, No. 1, pp. 42-47, 2003.
- [SELV04] M. P. Selvan, K. S. Swarup, "Large Scale Power System Computations with Object-Oriented Design", *IEEE Power Engineering Society General Meeting*, Vol. 1, pp. 330-335, Jun. 2004.
- [SILV00] E.L. da Silva, H. A. Gil, J. M. Areiza, "Transmission Network Expansion Planning under an Improved Genetic Algorithm", *IEEE Transactions on Power Systems*, Vol. 15, pp. 1168-1175, Aug. 2000.
- [SILV01] E. L. Silva, J. M. Areiza, G. C. de Oliveira, S. Binato, "Transmission Network Expansion Planning under a Tabu Search Approach", *IEEE Transactions on Power Systems*, Vol. 16, pp. 62-68, Feb. 2001.
- [SILV05] I. de J. Silva, M. J. Rider, R. Romero, A. V. Garcia, C. A. Murari, "Transmission network expansion planning with security constraints", *IEE Proceedings Generation Transmission Distribution*, Vol. 152, No. 6, Nov. 2005.
- [STEV94] W. Stevenson, "Power System Analysis", McGraw-Hill, 1994.
- [STOT74] B. Stott, O. Alsac, "Fast Decoupled Load Flow", *IEEE Transactions on Power Apparatus and Systems*, PAS-93, pp. 859-869, 1974.

- [SYBI04] G. Sybille, "SimPowerSystems User's guide version 4", published under sublicense from Hydro Quebec and The MathWorks Inc, 2004.
- [TAYL97] A. D. Taylor, "Object Technology A Management Guide", 2nd ed., Addison Wesley, 1997.
- [TEIX02] M. M. Teixeira, "Desenvolvimento de Aplicações Distribuídas usando Arquitetura CORBA", Notas didáticas, No. 59, ICMC-USP, Nov. 2002.
- [TRAC06] TRACTEBEL: Eurostag in short, disponible en <http://www.eurostag.be/frameset.htm>, accesado en 05 de diciembre del 2006.
- [VOUR04] C. D. Vournas, E. G. Potamianakis, C. Moors, T. V. Cutsem, "An educational Tool for Power System Control and Stability", *IEEE Transactions on Power Systems*, Vol. 19, No. 1, pp. 48-55, 2004.
- [WARD56] J. Ward, H. W. Hale, "Digital Computer Solution of Power Flow Problems"; *AIEE Transactions Part III Power Apparatus and Systems*, 75 : 398 – 404.
- [WIEN50] N. Wiener, "*The Human Use of Human Beings: Cybernetics and Society*", 1950.
- [WIKI06] PORTADA – WIKIPEDIA, la enciclopedia libre; disponible en <http://es.wikipedia.org/wiki/Portada>, accesado en 05 de diciembre del 2006.
- [YOUR89] E. N. Yourdon, "Modern Structured Analysis", Prentice Hall, 1990.

- [ZHOU96] E. Z. Zhou, "Object-oriented Programming, C++ and Power System Simulation", *IEEE Transactions on Power Systems*, Vol 11, No. 1, pp. 206-215, Feb. 1996.
- [ZHU97] J. Zhu, D. Lubkeman, "Object-Oriented Development of Software Systems for Power System Simulations", *IEEE Transactions on Power Systems*, Vol. 12, No. 2, pp. 1002-1007, May 1997.
- [ZHU99] J. Zhu, P. Jossman, "Application of Design Patterns for Object-Oriented Modeling of Power Systems", *IEEE Transactions on Power Systems*, Vol. 14, No. 2, pp. 532-537, May 1999.
- [ZIMM97] R. D. Zimmerman, D. Gan, "Matpower, Documentation for version 2", Power System Engineering Research Center, Cornell University, 1997.

ANEXOS

A. DATOS DE LOS SISTEMAS DE PRUEBA

A.1. NOMENCLATURA

Datos de las barras

N_b	Número de la barra.
<i>Nombre</i>	Nombre de la barra.
T	Tipo de la barra (0 = PQ. 1 = PV. 2 = V θ).
V_m	Módulo de la tensión (p.u.).
V_a	Ángulo de la tensión (graus).
P_G	Generación de potencia activa (MW).
Q_G	Generación de potencia reactiva (MVAR).
P_D	Potencia activa de la carga o demanda (MW).
Q_D	Potencia reactiva de la carga o demanda (MVAR).
g_s	Conductancia shunt (MW para $V = 1.0$ p.u.).
b_s	Susceptancia shunt (MVAR para $V = 1.0$ p.u.).

Datos de los ramos

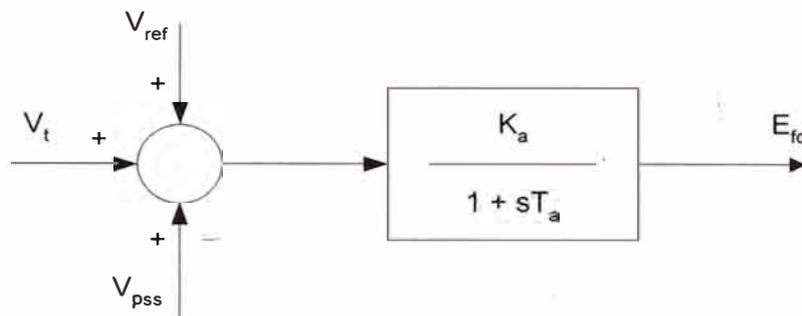
N_r	Número de la rama.
N_l	Número de la barra de salida.
N_F	Número da la barra de llegada.
N_C	Número de circuitos existentes.
r	Resistencia (p.u.).
x	Reactancia (p.u.).
b	Susceptancia total de la línea (p.u.).
Tap	Tap del transformador (p.u.).
ϕ	Ángulo de desfasaje del transformador en fase.
P_{Max}	Máxima capacidad de transferencia de la potencia aparente (MVA).
<i>Costo</i>	Costo de la construcción del ramo.

Datos de los generadores

N_b	Número de la barra de la máquina.
H	Constante de inercia (p.u.).
D	Constante de amortecimiento (p.u.).
S	Potencia base (MVA).
R	Resistencia (% p.u.).
X_d	Reactancia síncrona eje-d (% p.u.).

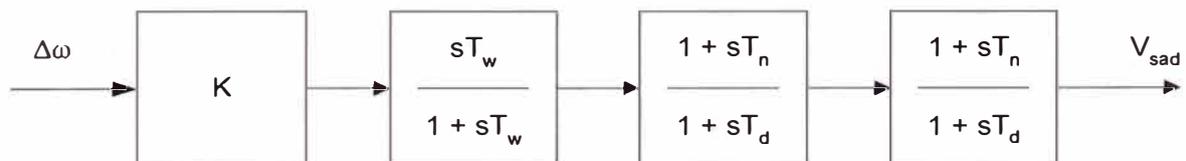
X_q	Reactancia síncrona eje-q (% p.u.).
X_{ld}	Reactancia transitoria eje-d (% p.u.).
X_{lld}	Reactancia sub-transitoria eje-d (% p.u.).
X_{llq}	Reactancia sub-transitoria eje-q (% p.u.).
T_{ld0}	Constante de tiempo circuito abierto eje-d (s)
T_{lld0}	Constante de tiempo sub-transitoria circuito abierto eje-d (s).
T_{llq0}	Constante de tiempo sub-transitoria circuito abierto eje-q (s).

Datos de los reguladores de tensión



N_b	Número de la barra de la máquina.
K_a	Ganancia del regulador de tensión (p.u./p.u.).
T_a	Constante de tiempo del regulador de tensión (s).
V_{ref}	Señal de referencia (p.u.).
V_t	Señal de entrada del regulador de tensión (p.u.).
V_{pss}	Señal do estabilizador de potencia (p.u.).
E_{fd}	Tensión de campo de la máquina (p.u.).

Datos de los estabilizadores de potencia



N_b	Número de la barra de la máquina.
K	Ganancia del estabilizador de potencia (p.u./p.u.).
T_w	Constante de tiempo del bloque wash-out (s).
T_n	Constante de tiempo del numerador en el bloque lead-leag (s).
T_d	Constante de tiempo del denominador en el bloque lead-leag (s).
$\Delta\omega$	Desvío de la velocidad angular de la máquina (p.u.).
V_{pss}	Señal estabilizador aplicado en el regulador de tensión (p.u.).

A.2. SISTEMA WECC

Datos de las barras

N_b	Nombre	T	V_m	V_a	P_G	Q_G	P_D	Q_D	g_s	b_s
1	Barra 1	2	1.040	0.00	0.00	0.00				
2	Barra 2	1	1.025	0.00	163.0	0.00				
3	Barra 3	1	1.025	0.00	85.0	0.00				
4	Barra 4	0	1.000	0.00						
5	Barra 5	0	1.000	0.00			125.0	50.0		
6	Barra 6	0	1.000	0.00			90.0	30.0		
7	Barra 7	0	1.000	0.00						
8	Barra 8	0	1.000	0.00			100.0	35.0		
9	Barra 9	0	1.000	0.00						

Datos de las ramas

N_r	N_l	N_F	N_C	r	x	b	Tap	ϕ	P_{Max}	Cost
1	1	4	1	0.0000	0.0576	0.00	1.000			
2	2	7	1	0.0000	0.0625	0.00	1.000			
3	3	9	1	0.0000	0.0586	0.00	1.000			
4	4	5	1	0.0100	0.0850	17.60				
5	4	6	1	0.0170	0.0920	15.80				
6	5	7	1	0.0320	0.1610	30.60				
7	6	9	1	0.0390	0.1700	35.80				
8	7	8	1	0.0085	0.0720	14.90				
9	8	9	1	0.0119	0.1008	20.90				

Datos de los generadores

N_b	H	D	S	u	R	X_d	X_q	X_{ld}	X_{lld}	X_{llq}	T_{ldo}	T_{lldo}	T_{llqo}
1	23.65	0.00	100.0	1	0.00	14.60	9.69	6.08	4.864	4.864	8.96	0.030	0.060
2	6.40	0.00	100.0	1	0.00	89.58	86.45	11.98	9.584	9.584	6.00	0.033	0.078
3	3.01	0.00	100.0	1	0.00	131.3	125.8	18.13	14.5	14.5	5.89	0.033	0.070

Datos de los reguladores de tensión

N_b	K_a	T_a
1	10.000	0.050
2	10.000	0.050
3	10.000	0.050

Datos de los estabilizadores de potencia

N_b	K_a	T_w	T_n	T_d
1	14.2823	5.0	0.4449	0.2183
2	71.7941	5.0	0.2440	0.0355
3	93.0477	5.0	0.1881	0.0459

A.3. SISTEMA GARVER

Datos de las barras

N_b	Nombre	T	V_m	V_a	P_G	Q_G	P_D	Q_D	g_s	b_s
1	Barra 1	2	1.000	0.00	50.0	0.00	80.0	0.00	0.00	0.00
2	Barra 2	0	1.000	0.00			240.0	0.00	0.00	0.00
3	Barra 3	1	1.000	0.00	165.0	0.00	40.0	0.00	0.00	0.00
4	Barra 4	0	1.000	0.00			160.0	0.00	0.00	0.00
5	Barra 5	0	1.000	0.00			240.0	0.00	0.00	0.00
6	Barra 6	1	1.000	0.00	545.0	0.00	0.00	0.00	0.00	0.00

Datos de las ramas

N_r	N_l	N_F	N_C	r	x	b	Tap	ϕ	P_{Max}	Cost
1	1	2	1	0.00	0.40	0.00			100.0	40.0
2	1	4	1	0.00	0.60	0.00			80.0	60.0
3	1	5	1	0.00	0.20	0.00			100.0	20.0
4	2	3	1	0.00	0.20	0.00			100.0	20.0
5	2	4	1	0.00	0.40	0.00			100.0	40.0
6	3	5	1	0.00	0.20	0.00			100.0	20.0
7	1	3	0	0.00	0.38	0.00			100.0	38.0
8	1	6	0	0.00	0.68	0.00			70.0	68.0
9	2	5	0	0.00	0.31	0.00			100.0	31.0
10	2	6	0	0.00	0.30	0.00			100.0	30.0
11	3	4	0	0.00	0.59	0.00			82.0	59.0
12	3	6	0	0.00	0.48	0.00			100.0	48.0
13	4	5	0	0.00	0.63	0.00			75.0	63.0
14	4	6	0	0.00	0.30	0.00			100.0	30.0
15	5	6	0	0.00	0.61	0.00			78.0	61.0

Datos de los generadores

N_b	H	D	S	u	R	X_d	X_q	X_{ld}	X_{lld}	X_{llq}	T_{ld0}	T_{lld0}	T_{llq0}
1	3.394	0.00	231.6	1	0.00	93.0	69.0	30.2	24.5	24.5	8.00	0.030	0.060
3	3.302	0.00	192.0	1	0.00	165.1	159.0	23.2	17.1	17.1	5.90	0.033	0.078
6	4.768	0.00	525.0	1	0.00	122.0	116.0	17.4	13.4	13.4	8.97	0.033	0.07

Datos de los reguladores de tensión

N_b	K_a	T_a
1	98.515	0.1158
3	71.490	0.0306
6	45.751	0.0442

B. ÍNDICE DE TABLAS

Tabla 1.1:	Metodologías orientadas a objetos.....	33
Tabla 3.1:	Principales simuladores de SEE existentes.	92
Tabla 3.2:	Lista de Paquetes y Clases de Análisis principales.....	104
Tabla 5.1:	Autovalores y amortecimiento considerando sólo las máquinas en los generadores – Sistema WECC.	149
Tabla 5.2:	Autovalores y amortecimiento considerando las máquinas y AVR's en los generadores – Sistema WECC.....	151
Tabla 5.3:	Autovalores y amortecimiento considerando las máquinas, AVR's y PSS's en los generadores – Sistema WECC.....	153
Tabla 5.4:	Parámetros del algoritmo genético maestro - Sistema Garver. .	160
Tabla 5.5:	Autovalores y amortecimiento de la configuración resultante considerando restricciones estáticas – Sistema Garver.....	162
Tabla 5.6:	Parámetros del algoritmo genético esclavo - Sistema Garver...	165
Tabla 5.7:	Resultados del algoritmo genético esclavo considerando restricciones estáticas e dinámicas - Sistema Garver.....	167
Tabla 5.8:	Autovalores y amortecimiento de la configuración resultante considerando restricciones estáticas e dinámicas – Sistema Garver.	168

C. ÍNDICE DE FIGURAS

Figura 1.1: Curva de fallas del hardware.....	16
Figura 1.2: Curva de fallas real e idealizada del software	16
Figura 1.3: Capas de la ingeniería de software.....	19
Figura 1.4: Modelo lineal secuencial.	22
Figura 1.5: Modelo de construcción de prototipos.....	23
Figura 1.6: Modelo basado en componentes.	24
Figura 1.7: Ejemplo de abstracción.....	27
Figura 1.8: Una clase y sus instancias.....	28
Figura 1.9: Ejemplo de encapsulamiento.....	29
Figura 1.10: Ejemplo de herencia.....	30
Figura 1.11: Modelo de proceso orientado a objetos.....	31
Figura 1.12: Visión general del UML.....	35
Figura 1.13: Notación UML para la clase	36
Figura 1.14: Notación UML para la interfaz	36
Figura 1.15: Notación UML para la colaboración.....	36
Figura 1.16: Notación UML para el actor.....	37
Figura 1.17: Notación UML para el caso de uso.....	37
Figura 1.18: Notación UML para el componente.....	37
Figura 1.19: Notación UML para la interacción.....	38
Figura 1.20: Notación UML para el paquete.....	38
Figura 1.21: Notación UML para las anotaciones.....	38
Figura 1.22: Notación UML para la dependencia.....	39
Figura 1.23: Notación UML para la generalización.....	39

Figura 1.24: Notación UML para la realización.....	40
Figura 1.25: Notación UML para la asociación.....	40
Figura 1.26: Notación UML para la agregación y composición.	40
Figura 1.27: Red de computadores.....	42
Figura 1.28: Estructura Cliente - Servidor.....	43
Figura 1.29: Estructura de CORBA.....	46
Figura 2.1: Relación entre ΔV e I para elementos eléctricos ideales.....	51
Figura 2.2: Ejemplo de circuito eléctrico AC.....	53
Figura 2.3: Resultados del análisis del circuito de la Figura 2.2.....	56
Figura 2.4: Relación entre los fasores ΔV e I para conductores ideales.....	57
Figura 2.5: Equivalencias para conexiones en serie y paralelo de impedancias.....	58
Figura 2.6: Análisis fasorial en un nodo genérico de un circuito AC.....	59
Figura 2.7: Ejemplo de diagrama de bloques para sistemas de control.	62
Figura 2.8: Equivalentes entre el dominio del tiempo y de la frecuencia.....	63
Figura 2.9: Amortecimiento de un autovalor complejo.....	66
Figura 2.10: Ejemplo básico de un sistema eléctrico de energía.	67
Figura 2.11: Mapa del Sistema Interconectado Nacional (SINAC).....	70
Figura 2.12: Diagrama unifilar del SEE mostrado en la Figura 2.10.....	71
Figura 2.13: Representación lógica del SEE mostrado en la Figura 2.12.	72
Figura 2.14: Fenómenos dinámicos en SEE.	74
Figura 2.15: Modelos de elementos para análisis de flujo de potencia.	76
Figura 2.16: Ejemplo de matriz dispersa.	80
Figura 2.17: Componentes del Generador.	84
Figura 2.18: Máquina Síncrona [PAUC98].	85
Figura 3.1: Casos de Uso del Sistema de Software.....	101
Figura 3.2: Especificación del Caso de Uso: Ejecutar Flujo de Potencia. ...	102
Figura 3.3: Especificación de la Interfaz Principal.....	103
Figura 3.4: Paquetes principales del modelo de análisis.....	105
Figura 3.5: Diagrama de colaboración del caso de uso: Ejecutar Flujo Potencia.	107

Figura 3.6:	Diagrama de clases del Modelo de Análisis.	108
Figura 3.7:	Uso de CORBA en la incorporación de softwares antiguos.	110
Figura 3.8:	Diagrama de paquetes de diseño.....	111
Figura 3.9:	Diagrama de Clases del Paquete SEE.....	112
Figura 3.10:	Diagrama de Clases del Paquete Aplicación.....	115
Figura 4.1:	Ejemplo de la convergencia del AG a través de las generaciones.....	127
Figura 4.2:	Ejemplo de un cromosoma con codificación binaria.....	128
Figura 4.3:	Ejemplo de la Interacción entre el AG y una aplicación específica.....	129
Figura 4.4:	Ejemplo del operador cruzamiento.....	130
Figura 4.5:	Descomposición del problema de la Expansión de la Transmisión.....	135
Figura 4.6:	Representación del cromosoma – AG maestro.....	136
Figura 4.7:	Diagrama de flujo de la función <i>fitness</i> – AG maestro.....	137
Figura 4.8:	Representación del cromosoma – AG esclavo.....	139
Figura 4.9:	Variación del parámetro α a través de las generaciones.....	141
Figura 4.10:	Extensión del Diagrama de Casos de Uso.....	142
Figura 4.11:	Extensión del diagrama de clases.....	143
Figura 4.12:	Diagrama de clases del AG.....	144
Figura 5.1:	Diagrama físico del sistema eléctrico WECC.....	146
Figura 5.2:	Resultado del flujo de carga obtenido con el SinergiaSEE– Sistema WECC.....	147
Figura 5.3:	Comparación de los módulos de las tensiones obtenidos con el ANAREDE y con el SinergiaSEE - Sistema WECC.....	148
Figura 5.4:	Comparación de los ángulos de las tensiones obtenidos con el ANAREDE y con el SinergiaSEE - Sistema WECC.....	148
Figura 5.5:	Comparación de los autovalores considerando sólo las máquinas en los generadores - Sistema WECC.....	150
Figura 5.6:	Zoom en la comparación de los autovalores considerando sólo las máquinas en los generadores - Sistema WECC.....	150

Figura 5.7: Comparación de los autovalores obtenidos considerando las máquinas y AVR's en los generadores - Sistema WECC.	152
Figura 5.8: Zoom en la comparación de los autovalores obtenidos considerando las máquinas y AVR's en los generadores - Sistema WECC.	152
Figura 5.9: Comparación de los autovalores obtenidos considerando las máquinas, AVR's y PSS's en los generadores - Sistema WECC.	154
Figura 5.10: Zoom en la comparación de los autovalores obtenidos considerando las máquinas, AVR's y PSS's en los generadores - Sistema WECC.	154
Figura 5.11: Comparación de los ángulos de los rotores de los generadores obtenidos considerando sólo las máquinas - Sistema WECC.	155
Figura 5.12: Comparación de los ángulos de los rotores de los generadores obtenidos considerando las máquinas y AVR's - Sistema WECC.	156
Figura 5.13: Comparación de los ángulos de los rotores de los generadores obtenidos considerando las máquinas, AVR's y PSS's - Sistema WECC.	157
Figura 5.14: Diagrama físico del sistema eléctrico Garver.	158
Figura 5.15: Representación del cromosoma correspondiente al algoritmo genético maestro - Sistema Garver.	159
Figura 5.16: Convergencia del algoritmo genético maestro considerando restricciones estáticas – Sistema Garver.	160
Figura 5.17: Cromosoma solución del algoritmo genético maestro considerando restricciones estáticas – Sistema Garver.	161
Figura 5.18: Configuración resultante considerando restricciones estáticas – Sistema Garver.	161
Figura 5.19: Mapa de polos de la configuración resultante considerando restricciones estáticas – Sistema Garver.	163

Figura 5.20: Zoom del mapa de polos de la configuración resultante considerando restricciones estáticas – Sistema Garver..... 163

Figura 5.21: Ángulo del rotor de los generadores de la configuración resultante Garver considerando restricciones estáticas..... 164

Figura 5.22: Representación del cromosoma correspondiente al algoritmo genético esclavo - Sistema Garver. 166

Figura 5.23: Convergencia del algoritmo genético maestro considerando restricciones estáticas e dinámicas - Sistema Garver..... 166

Figura 5.24: Cromosoma solución del algoritmo genético maestro considerando restricciones estáticas e dinámicas - Sistema Garver. 166

Figura 5.25: Configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver..... 167

Figura 5.26: Mapa de polos de la configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver..... 169

Figura 5.27: Zoom en el mapa de polos de la configuración resultante considerando restricciones estáticas e dinámicas - Sistema Garver..... 169

Figura 5.28: Ángulo del rotor de los generadores de la configuración resultante considerando restricciones estáticas e dinámicas – Sistema Garver. 170