

UNIVERSIDAD AUSTRAL DE CHILE
SEDE PUERTO MONTT
ESCUELA DE INGENIERIA EN COMPUTACION



SISTEMA DE ANÁLISIS DE VARIABLES ELÉCTRICAS CON
REDES DE NEURONAS ARTIFICIALES

Seminario de Titulación
para optar
al título de Ingeniero en Computación

PROFESOR PATROCINANTE:
Sra. Mónica Gallardo Vargas

RICARDO ANDRÉS RUBILAR SANHUEZA

PUERTO MONTT – CHILE

2008



Universidad Austral de Chile
Escuela de Ingeniería en Computación

Los Pinos s/n, Balneario Pelluco
Sede Puerto Montt
Casilla 1327 · Fono: 56 65 260990
Fax: 56 65 277156
Email: ecomputa@uach.cl
www.uach.cl

Puerto Montt, 24 de septiembre de 2008

COMUNICACIÓN INTERNA N° 202/08

DE : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA DE INGENIERIA EN COMPUTACION

A : Dr. Renato Westermeier H. – **VICERRECTOR SEDE PUERTO MONTT**
Mag. César Pino Soto – **COORDINADOR ACADEMICO SEDE PUERTO MONTT**
Sra. Cristina Barriga – **REGISTRO ACADEMICO**
Sra. Alba Vásquez - **ENCARGADA DE TITULACIÓN SEDE PUERTO MONTT**

C.c : Sr. Ricardo Rubilar Sanhueza ✓✓
Sra. Mónica Gallardo Vargas
Sr. Moisés Coronado Delgado
Sra. Sandra Ruiz Aguilar

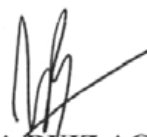
MOTIVO:

Informar a usted, las calificaciones obtenidas por el alumno de Ingeniería en Computación **Sr. Ricardo Andrés Rubilar Sanhueza** Rut 13.968.910-0, en su informe de Titulación "*Sistema de Análisis de Variables Eléctricas con Redes de Neuronas Artificiales*"

Prof. Mónica Gallardo Vargas	6.5
Prof. Moisés Coronado Delgado	6.5
Prof. Sandra Ruiz Aguilar	6.2

Promedio Seminario	6.40
---------------------------	-------------

Sin otro particular, le saluda atentamente,


SANDRA RUIZ AGUILAR
DIRECTORA



SRA/mva

PUERTO MONTT, 24-sept-2008

De : Sra. Mónica Gallardo Vargas
PROFESORA PATROCINANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno **Ricardo Andrés Rubilar Sanhueza** en su Seminario de Titulación "Sistema de Análisis de Variables Eléctricas con Redes de Neuronas Artificiales":

NOTA:

6,5

JUSTIFICACION:

Solución innovadora a un problema actual.
Bien desarrollado el tema.

OTRAS OBSERVACIONES:

Proyecto debería convertirse en apli-
cación final y cuantificar su eficiencia.


Sra. Mónica Gallardo Vargas
PROFESORA PATROCINANTE

PUERTO MONTT, 24-87-2008

De : Sr. Moisés Coronado Delgado
PROFESOR INFORMANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno **Ricardo Andrés Rubilar Sanhueza** en su Seminario de Titulación "Sistema de Análisis de Variables Eléctricas con Redes de Neuronas Artificiales":

NOTA: 6,5

JUSTIFICACION:

→ buen tema, completo de contenido y explicación.
→ ¿Por qué no fue más?
→ con los equipos de línea poco se completó.

OTRAS OBSERVACIONES:

→ problemas con algebra, mayor es: 14 y 10

Sr. Moisés Coronado Delgado
PROFESOR INFORMANTE

PUERTO MONTT, 24 sept 2008

De : Sra. Sandra Ruiz Aguilar
PROFESORA INFORMANTE

A : Sra. Sandra Ruiz Aguilar
DIRECTORA ESCUELA INGENIERÍA EN COMPUTACIÓN

MOTIVO:

Informar a Usted la calificación obtenida por el alumno **Ricardo Andrés Rubilar Sanhuesa** en su Seminario de Titulación "Sistema de Análisis de Variables Eléctricas con Redes de Neuronas Artificiales":

NOTA: 6.2

JUSTIFICACION:

Sistema innovador.

Relativamente complejo de entender.

OTRAS OBSERVACIONES:


Sra. Sandra Ruiz Aguilar
PROFESORA INFORMANTE

INDICE

SINTESIS EN CASTELLANO

SINTESIS EN INGLÉS

1	INTRODUCCIÓN.....	1
2	OBJETIVOS.....	5
2.1	Objetivo General	5
2.2	Objetivos específicos	5
3	PLANTEAMIENTO DEL PROBLEMA.....	6
3.1	Antecedentes	6
3.1.1	Definición del problema.....	6
3.1.2	Identificación de esfuerzos anteriores para resolverlo	10
3.1.3	Solución propuesta.....	11
3.1.4	Equipo de trabajo	14
3.2	Justificación.....	15
3.2.1	Situación sin proyecto	15
3.2.2	Situación con proyecto	15
3.3	Delimitación.....	16
4	METODOLOGÍA.....	17
4.1	Desarrollo de Software.....	17
4.2	Implementación de Red Neuronal.....	18
5	RECURSOS	20
5.1	Equipos	20
5.2	Software	21
5.2.1	Herramientas de desarrollo	21
5.2.2	Usuarios	22
6	Inicio del Proyecto	23
6.1	Investigación previa.....	23
6.1.1	Teoría eléctrica respecto de las variables analizadas	23

6.1.2	Normativa eléctrica Chilena Vigente (NCH 2003-04)	31
6.1.3	Equipo de monitoreo eléctrico DM-II	32
6.1.4	Sistemas de reconocimiento de patrones.	33
6.1.5	Redes Neuronales.....	34
6.1.6	Generalidades de las Redes Neuronales.....	35
6.2	Preparación del entorno de desarrollo.	40
6.3	Modelado Inicial	41
7	Elaboración.....	42
7.1	Definición de requerimientos.....	42
7.1.1	Modelo del Dominio.....	42
7.1.2	Glosario de términos	43
7.1.3	Diagramas de Casos de Uso del Sistema.....	47
7.1.4	Especificación de los Casos de Uso	48
7.2	Diagrama de Clases de Base de Objetos	55
7.3	Detalle de Clases y relaciones	58
8	Construcción.....	65
8.1	Definición de Arquitectura	65
8.2	Análisis estadístico de la medición y pre condiciones de búsqueda de anomalías.....	66
8.3	Redes Neuronales utilizadas en el análisis	67
8.3.1	Red Neuronal 1. Sugerencias de análisis.	68
8.3.2	Red Neuronal 2. Detección y clasificación de anomalías.....	72
8.3.3	Sub Redes especializadas	77
9	Interfaz de Usuario	78
10	Pruebas.....	84
11	Conclusiones y Recomendaciones	88
12	Bibliografía	90
13	ANEXOS	92

TABLAS

Tabla 1. Equipo de trabajo.....	14
Tabla 2. Características técnicas equipo DM-II.....	33
Tabla 3. Detalle de clases.....	58
Tabla 4. Detalle de asociaciones.	59
Tabla 5. Detalle de atributos clase Cliente.....	60
Tabla 6. Detalle de atributos clase Consultor	61
Tabla 7. Detalle de atributos clase Auditoria.....	62
Tabla 8. Detalle de atributos clase Medición	63
Tabla 9. Detalle de atributos clase Registro	64
Tabla 10. Arreglos de entrada y salida de red neuronal 1.	69
Tabla 11 Configuraciones de Red Neuronal 1.	70
Tabla 12. Error en la iteración número 1.000 del proceso de entrenamiento de la red neuronal 1.....	72
Tabla 13. Arreglos de Entrada y salida Red Neuronal 2.	73
Tabla 14. Configuraciones Red Neuronal 2.	74
Tabla 15. Error en entrenamiento de distintas configuraciones de la red neuronal 2.....	75
Tabla 16. Error en la iteración número 1.000 del proceso de entrenamiento de la red neuronal 2.....	76

FIGURAS

Figura 1. Instalación típica de equipo de captura de variables eléctricas.	7
Figura 2. Proceso de análisis de los datos eléctricos capturados.....	9
Figura 3. Ciclo de vida de AUP.....	18
Figura 4. Definición de la tensión eléctrica.	24
Figura 5. Definición de corriente eléctrica.....	24
Figura 6. Definición de potencia.....	25
Figura 7. Ecuación de la potencia en función de la tensión y la corriente.....	26
Figura 8. Ecuación y representación grafica de tensión y la corriente con desfase.	27
Figura 9. Funciones de la tensión y la corriente en régimen sinusoidal.....	27
Figura 10. Ecuación de la potencia en régimen sinusoidal	28
Figura 11. Ecuación de la potencia media.	29
Figura 12. Ecuación de potencia aparente.	31
Figura 13.Red neuronal genérica.....	35
Figura 14. Red neuronal Artificial.....	36
Figura 15. Comparación de resultados obtenidos con Matlab y C#.....	41
Figura 16. Modelo del Dominio	43
Figura 17. Casos de uso del Sistema	48
Figura 18. Diagrama de clases de Base de Objetos.....	55
Figura 19. Diagrama de clases Red Neuronal	57
Figura 20. Capas de Arquitectura de la Aplicación.	65

Figura 21. Funciones de activación del perceptron multicapa.	67
Figura 22. Error en entrenamiento de distintas configuraciones de la red neuronal 1.	71
Figura 23. Interfaz de Usuario – Ingreso/Edición de Cliente.....	79
Figura 24. Interfaz de Usuario –Creación de Auditoría.....	80
Figura 25. Interfaz de Usuario – Ingreso/validación patrones de análisis.	81
Figura 26 Interfaz de Usuario – Proceso entrenamiento Redes Neuronales....	82
Figura 27. Interfaz de Usuario – Generación/Visualización de Reportes.....	83
Figura 28. Reporte de Visual Studio para pruebas de unidad	86
Figura 29. Implementación de Test Clase MedicionHelper.....	95
Figura 30. Implementación de análisis de sobre carga.....	96
Figura 31. Implementación método de aprendizaje Back Propagation.....	97
Figura 32. Implementación cálculo de salida de cada neurona	98
Figura 33. Implementación clase Network.....	100

SINTESIS EN CASTELLANO

Los requerimientos de esta implementación nacen de las actividades realizadas por la Empresa Eficiencia Energética Integral (EEI) que, en la ejecución de sus actividades de consultoría, captura la información eléctrica de sus clientes mediante la utilización de un equipo de medición autónomo.

La herramienta de software consiste en una aplicación entrenada para la detección de fallos de acuerdo a la experiencia aportada por los consultores de EEI, para lo cual se utilizaron redes neuronales de perceptrones distribuidos en múltiples capas y entrenados mediante el procedimiento de propagación del error hacia atrás (Back Propagation). Además, según los diagnósticos obtenidos, la aplicación cuenta con una base de conocimientos capaz de asignar a cada hallazgo una posible solución, lo que provee a los consultores de una aproximación del diagnóstico lo que incide en una reducción de las horas hombre (HH) de ejecución de los análisis y un tiempo de respuesta reducido para el cumplimiento de los requerimientos de los clientes.

Como patrón de este desarrollo se adoptó la metodología Ágil, la cual provee en sus distintas etapas entregas parcializadas incrementales, lo que permitió ajustar las características de la herramienta a los requerimientos de los usuarios, que en primera instancia no contaban con una clara definición.

La arquitectura del sistema se estructuró en tres capas definidas: Capa de Acceso a Datos, Capa de Negocios e Interfaz de Usuario, que para su implementación requirieron de las herramientas Microsoft .Net, Db4o.

Los objetivos propuestos en la definición inicial del proyecto fueron cubiertos a cabalidad, obteniendo como producto final una herramienta capacitada para el diagnóstico de fallos en el ámbito eléctrico y con la capacidad de adaptarse a nuevas condiciones de falla identificadas en los procesos de asesoría realizados por EEI y su continua investigación de las nuevas tecnologías incorporadas al mercado eléctrico.

SINTESIS EN INGLÉS

Later it is expose the development of a software tool for the analysis of electrical variables, placed in the Qualification Seminar of the thesis's student. This development consists in the implementation of an anomaly's diagnosis system on electrical facilities of low tension, by the identification of patterns failures using artificial neurons networks.

The requirements of this implementation born of the activities done by the Company Integral Energetic Efficiency (EEI in Spanish) that, in the execution of its consultancy activities, captures the electrical information of its clients by the utilization of an autonomous measurement equipment.

The software tool consists of an application trained for the detection of failures based in the experience contributed by the consultants of EEI, being used for this, neurons networks of perceptrons distributed in multiple layers and trained by the procedure of mistake backward propagation (Back Propagation). In addition, according to the obtained diagnoses, the application relies on a base of knowledge capable of assign to every find a possible solution, which provides the consultants of an approximation of the diagnosis wich produce a reduction of the execution man-hours (HH) in the analyses and a reduce time of response for the fulfillment of the requirements of the clients.

As a pattern of this development it was adopted the Agile Methodology, which provides in its different stages incremental partialized deliveries, which allowed to fit the characteristics of the tool to the user's requirements, who in first instance didn't have a clear definition.

The architecture of the system was constructed in three defined caps: Cap of Access to Data, Business Cap and User's Interface, which for its implementation was required of the Microsoft .Net, Db4o tools.

The aims proposed in the initial definition of the project were totally covered, obtaining as final product a tool qualified for the diagnosis of failures in the electrical field and with the capability to adapt itself to new conditions of fault identified in the advising processes done by EEI and its continuous investigation of the new technologies incorporated into the electrical market.

1 INTRODUCCIÓN

El crecimiento económico se vincula a un mayor consumo de energía y por ende a una mayor demanda de recursos para la generación de ésta.

Este fenómeno ha ido variando en los países industrializados registrando una drástica disminución en la relación crecimiento versus recursos energéticos.

Este desacoplamiento es el resultado de la aplicación y fomento de políticas de uso eficiente de las energías disponibles. La aplicación de éstas, han logrado rebajas del orden de 1,5 puntos porcentuales en promedio anual en la intensidad energética.

Sin embargo, mientras en los países desarrollados la Eficiencia Energética constituye una parte central de la matriz energética, en Chile, el consumo de energía ha crecido a la par con la economía, lo cual sobrelleva enormes esfuerzos de inversión para la satisfacción de esta demanda y genera efectos ambientales locales y globales ligados inevitablemente a la generación eléctrica así como al consumo de combustibles.

Conocedor el gobierno de esta problemática, ha decidido crear políticas para fomentar buenas prácticas a nivel industrial, por medio de la realización de Auditorías en Eficiencia Energéticas, las cuales contarán con subvención del

estado y cuya finalidad es crear conciencia con respecto al tema para lograr revertir y controlar el crecimiento de la demanda energética del país.

Dentro de este marco, la empresa Eficiencia Energética Integral Ltda. (EEI) ha decidido integrarse a este plan país y realizar Consultorías en materias de gestión y uso de energías, incluyendo dentro de sus servicios los análisis eléctricos mediante la utilización de un equipo capturador de variables eléctricas marca Amprobe, modelo DM-II, con la finalidad de orientar a las empresas en el emprendimiento de actividades tendientes a mejorar la eficiencia en el uso de la energía en sus procesos industriales, las cuales son formalizadas mediante la entrega de un informe con las sugerencias mediante las cuales el cliente podrá mejorar su desempeño energético.

Las variables eléctricas obtenidas por el capturador, deben ser analizadas por los ingenieros de EEI, proceso que requiere de un tiempo excesivo lo cual aumenta los costos en la realización de las auditorias, debido al mayor número de Horas Hombre (HH) dedicados exclusivamente al análisis de los datos.

Además de los requerimientos de HH dedicadas a los análisis, es un factor clave para el análisis, la experiencia de quien revisa los datos, ya que esta condición le permite identificar anomalías que perjudican el desempeño energético de la empresa, lo cual incide en los costos de producción de

diversas maneras y no sólo por el desperdicio de energía que es el factor más evidente.

La incidencia en los costos del alto número de HH dedicados al análisis, junto con la condición de vulnerabilidad por la dependencia directa del conocimiento práctico de cada uno de los ingenieros de EEI, motivan la necesidad de contar con una herramienta automatizada de análisis y que permita además plasmar el conocimiento adquirido en cada uno de los análisis en una base de datos que permita identificar patrones de fallo y orientar al ingeniero en las posibles soluciones, basados en las experiencias obtenidas de los análisis realizados con anterioridad.

Para el desarrollo de esta herramienta automatizada de análisis, se ha seleccionado como pauta del desarrollo los conceptos expuestos en la Metodología Ágil, ya que se acopla a las condiciones de baja definición inicial del problema y por la orientación de la misma hacia el cumplimiento a cabalidad de los requerimientos del usuario por medio de una participación activa del mismo en el proceso de desarrollo.

Los contenidos a continuación expuestos, detallan el proceso de implementación de la herramienta y pueden ser resumidos de la siguiente manera:

- En el capítulo 2 se describe el problema abordado y las definiciones iniciales del proyecto a desarrollar.
- En el capítulo 3 se establecen objetivos del proyecto.
- En el capítulo 4 se describe la metodología a utilizar para el desarrollo del software y la implementación de las redes neuronales.
- En el capítulo 5 se describen los recursos utilizados en el proyecto.
- En el capítulo 6 se describe la etapa de inicio del proyecto y los contenidos básicos asociados al desarrollo de la aplicación.
- En el capítulo 7 se detallan los modelos y definiciones correspondientes a la etapa de Elaboración.
- En el capítulo 8 se expone el proceso de construcción de la herramienta.
- En el capítulo 9 se exhiben algunas de las implementaciones de la Interfaz de usuario.
- En el capítulo 10 se listan las pruebas realizadas en el proceso de desarrollo de la aplicación.
- En el capítulo 11 se indican las conclusiones y recomendaciones.
- En el capítulo 12 se detalla la bibliografía utilizada
- En el capítulo 13 se exhiben los anexos.

2 OBJETIVOS

2.1 Objetivo General

Desarrollo de una herramienta de análisis de variables eléctricas capaz de clasificar mediante redes de neuronas artificiales condiciones anómalas según patrones previamente definidos

2.2 Objetivos específicos

- Reducir las Horas Hombre requeridas para el análisis de los datos eléctricos capturados.
- Mantener una base de patrones de falla y las soluciones asociadas a los a mismos.
- Determinar el cumplimiento normativo de la energía eléctrica analizada
- Detectar problemas eléctricos derivados del deficiente suministro eléctrico
- Detectar problemas eléctricos derivados del deficiente estado de las instalaciones
- De acuerdo a la asociación de variables y la experiencia de los profesionales de la empresa, determinar causas y sugerir soluciones a las anomalías detectadas.

3 PLANTEAMIENTO DEL PROBLEMA

3.1 Antecedentes

3.1.1 Definición del problema

En la matriz energética de las empresas, la electricidad constituye unos de los elementos con mayor impacto en la estructura de costos, por lo que controlar y gestionar este recurso se convierte en una condición indispensable para la competitividad de las empresas.

EEl, empresa certificada por el Instituto Nacional de Normalización (INN) para la realización de auditorías en Eficiencia Energética, cuenta con el equipo Amprobe DM- II, diseñado para monitorear las diferentes variables derivadas del consumo eléctrico en las empresas, de manera confiable y con el suficiente detalle como para tomar decisiones en cuanto a la manera en que se consume y la identificación de posibles ineficiencias derivadas de la configuración o diseño de los sistemas de potencia presentes en las diversas líneas de procesos utilizadas en las industrias.

A continuación se muestra la configuración tipo de los sistemas eléctricos en las industrias y cómo es instalado el equipo para la captura de los datos.

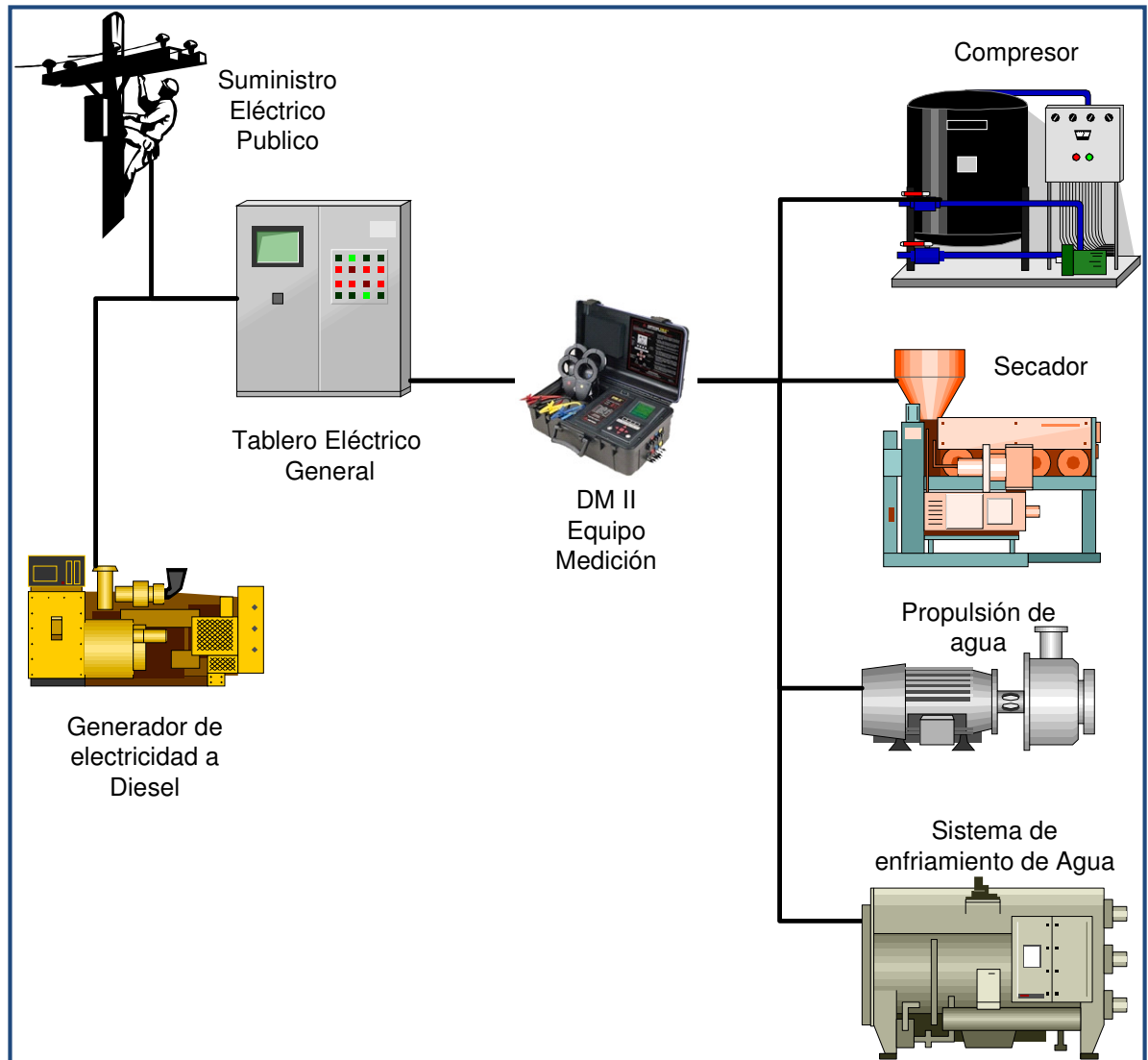


Figura 1. Instalación típica de equipo de captura de variables eléctricas.

Las características descritas anteriormente derivan en la captura simultánea por parte del equipo de 26 variables eléctricas, las cuales son asociadas a la fecha y hora en que fueron capturadas.

Para visualización de la información capturada, la aplicación provista por el fabricante del equipo (Download Suite 2.3.0.5), permite desplegar en formato tabla, cada uno de los registros, ordenados cronológicamente.

Para el posterior análisis de la información, el personal de EEI exporta los datos a una Hoja de Cálculo, mediante la cual realiza una serie de análisis estadísticos que permiten tener una primera aproximación respecto a posibles anomalías presentes en la instalación eléctrica analizada.

Si son evidenciados por los métodos estadísticos ciertos patrones de posibles anomalías, el ingeniero a cargo del análisis procede a una revisión minuciosa de cada uno de los registros que en promedio son 6.000, por lo que esta tarea requiere un número de HH elevado, lo que encarece el costo final de las auditorías realizadas por EEI.

A continuación se grafica la cronología del análisis de los datos:

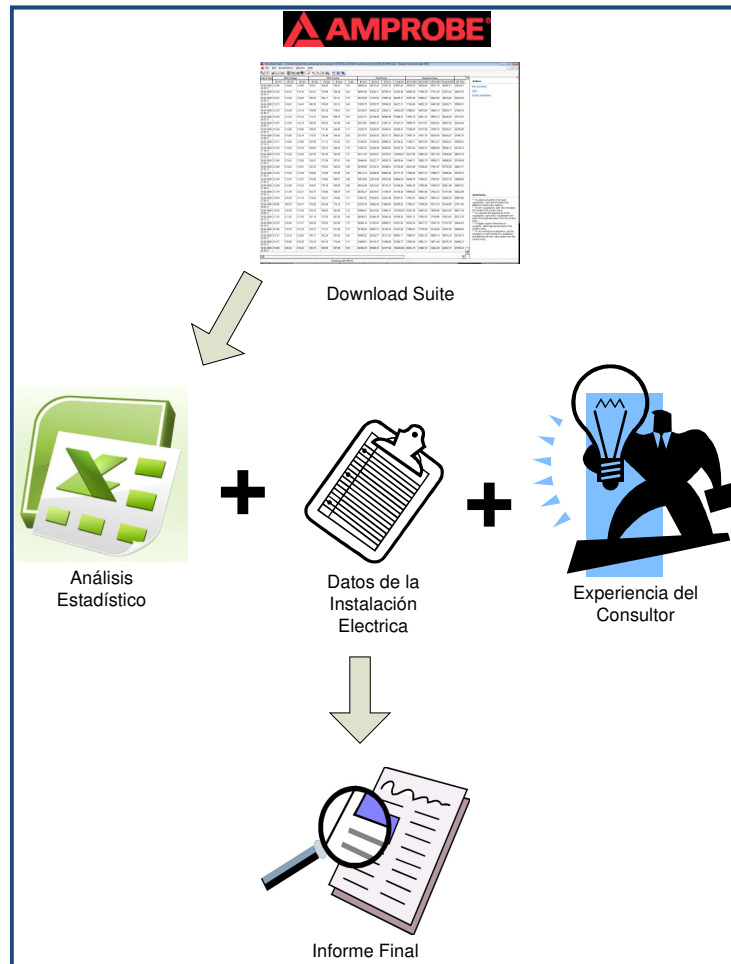


Figura 2. Proceso de análisis de los datos eléctricos capturados.

Sobre los datos capturados el especialista logra determinar las siguientes anomalías:

- Variación de Voltaje
- Errónea secuencia de las Fases

- Descompensación de cargas entre fases
- Incumplimiento normativo de Factor de Potencia
- Sobrecarga de líneas
- Sobre Demanda

El proceso de revisión de los registros, así como las sugerencias de posibles causas y soluciones de las anomalías detectadas son entregadas en un documento llamado “Informe Final”, y actualmente no cuenta con una centralización parametrizada, por lo que este conocimiento se obtiene de los ingenieros encargados del análisis, lo que los convierte en un factor crítico dentro de la consultora.

3.1.2 Identificación de esfuerzos anteriores para resolverlo

Como aproximación de solución se podría considerar la hoja de cálculo que utiliza el Ingeniero a cargo del análisis de la información, la cual es implementada con funciones estadísticas básicas que le dan una primera aproximación a posibles problemas presentes en los datos analizados.

Adicional a esto no se registran algún otro intento por solucionar esta situación.

3.1.3 Solución propuesta

Analizando la situación anteriormente expuesta, se propone lo siguiente:

“Desarrollo de una herramienta de análisis de variables eléctricas capturadas en instalaciones industriales”

Para el desarrollo de este proyecto se considera la definición de los siguientes aspectos:

3.1.3.1 Arquitectura

Se desarrollará una aplicación en el cliente, debido a la condición de utilización de un laptop por parte del Ingeniero y que las labores son desarrolladas en sitios alejados y no se cuenta con conexión a internet en todos los casos como para desarrollar una aplicación web.

Para la actualización de la red neuronal encargada de la detección de patrones de fallo, se contará con un servidor en el cual se mantendrá la última versión actualizada, que podrá ser descargada por cada una de las aplicaciones cliente.

3.1.3.2 Funciones

Para la realización de los análisis, la aplicación deberá contar con las siguientes características:

- a. Importación de los datos desde un archivo Excel hacia la aplicación.
- b. Normalización de los datos y almacenamiento en formato de la aplicación.
- c. Análisis de cumplimiento normativo de calidad de suministro.

Para la detección de incumplimiento normativo se utilizan los criterios definidos en la normativa eléctrica vigente. Se realiza un análisis estadístico el cual incluye la determinación de mínimos, máximos, media y desviación estándar por cada una de las variables capturadas, con la finalidad de identificar inicialmente posibles anomalías. Una vez detectadas las primeras evidencias de posibles fallas, se procesa cada uno de los registros mediante 8 tipos de análisis, los que son aplicados a sugerencia de la red neuronal entrenada para estos fines, según lo evidenciado inicialmente por la información estadística. Los análisis a realizar son los que a continuación se detallan:

- Desbalance de Tensión
- Desbalance de Corriente
- Secuencia de Fases
- Sobre Carga
- Normativo
- Corriente Reactiva

- Corriente de Neutro
 - Índice de uso de la potencia contratada.
- d. Detección de problemas eléctricos derivados del deficiente suministro eléctrico.

Cuando la calidad de la energía entregada por el proveedor no cumple con los requerimientos técnicos, se evidencia esta condición en la **tensión** presente en la red, que en condiciones deficientes provoca ciertas anomalías en los equipos, las cuales serán identificadas por la aplicación mediante la comparación de los índices de anomalías de cada uno de los registros. Estos índices de anomalía estarán determinados por los análisis ejecutados según las inferencias realizadas por el previo proceso de los índices estadísticos y las sugerencias aportadas por la Red Neuronal entrenada para estos fines.

- e. Detectar problemas eléctricos derivados del deficiente estado de las instalaciones.

Así como las condiciones del suministro afectan el comportamiento de los consumos, el deficiente estado de las instalaciones y la configuración de los equipos pueden producir diversas condiciones no deseadas dentro de la empresa, lo cual se pretende detectar también mediante el análisis del comportamiento de la **corriente** y la **energía**, así como la correlación

que existe entre éstos. Se utilizará una red neuronal capacitada para identificar las fallas y según la clasificación que realice de las mismas, indicará las sugerencias para la corrección de dichas anomalías.

- f. Entregar información detallada de los resultados obtenidos en el proceso de análisis.

3.1.4 Equipo de trabajo

Para el desarrollo de este proyecto se contará con el siguiente personal

Nombre	Institución	Cargo
Ricardo Toledo	EEI Ltda	Gerente Técnico
Edgardo Badilla	EEI Ltda	Gerente Proyectos
Ricardo Rubilar	EEI Ltda	Enc. Proyecto

Tabla 1. Equipo de trabajo.

Los componentes de este equipo de trabajo, evidencian un gran potencial, al ser profesionales de distintas áreas que están vinculadas directamente con el tipo de proyecto que se desea desarrollar.

3.2 Justificación

3.2.1 Situación sin proyecto

La situación actual presenta una serie de falencias que pudieran forzar a cometer errores en los análisis realizados por el personal de EEI. Dentro de estas falencias podemos mencionar que los criterios de evaluación no se encuentran normados y sólo se basan en la experiencia de los evaluadores.

El volumen de información requerido para realizar cada uno de los análisis dificulta, en ocasiones, la correcta identificación de las fallas, por lo que se han registrado problemas con algunos de los análisis realizados anteriormente en cuanto a la acertividad de los diagnósticos.

El factor humano sigue siendo crítico ya que los parámetros de análisis de la información son basados en el profesional que lo realiza y no existe un registro centralizado donde se disponga de esta información.

3.2.2 Situación con proyecto

El desarrollo de este proyecto solucionaría las deficiencias que presenta el procedimiento de análisis eléctrico que tiene la empresa, aportando las siguientes funcionalidades:

- Sistema de análisis capaz de detectar anomalías, determinar posibles causas y sugerir soluciones para las mismas.
- Reducción del tiempo de análisis de los sistemas auditados.
- Estandarización en los criterios de análisis de la información capturada.
- Automatización de procesos anticipando aumento en la demanda de análisis, según lo proyectado por la empresa en su análisis de mercado.

3.3 Delimitación

Este proyecto no considera la interfaz de transmisión de datos desde los capturadores a la aplicación, debido a que cada uno de ellos cuenta con aplicaciones propietarias de los mismos.

Cada uno de los análisis quedará limitado a las características ofrecidas por el capturador de datos.

La implementación de la solución constituye un prototipo de funcionamiento en forma local.

4 METODOLOGÍA

4.1 Desarrollo de Software

Considerando la definición del problema, se ha seleccionado la metodología de desarrollo AUP (Agile Unified Process), por las siguientes razones:

No existe un sistema predecesor. Si bien el cliente tiene una clara conciencia de la necesidad, los requisitos finales están poco definidos, por lo tanto, la metodología debe aportar características dinámicas con respecto a la definición de los requerimientos como lo indica el manifiesto Ágil respecto a la capacidad de responder a los cambios sobre el plan original.

El carácter iterativo de esta metodología, permite la realización de ajustes en el diseño, característica no provista por otras metodologías de desarrollo secuencial.

Minimización temprana de altos riesgos, sean estos de tipo técnicos, de requisitos, objetivos y utilización, así como entregas parciales funcionales más rápidas.

Provee de retroalimentación temprana, e integración de los usuarios al equipo de desarrollo, comprometiendo su interés en el proceso y la utilización posterior del mismo.

El desarrollo iterativo con cuenta con menor tiempo entre iteraciones manteniendo el mismo esquema de RUP, como se aprecia en el siguiente diagrama de tareas.

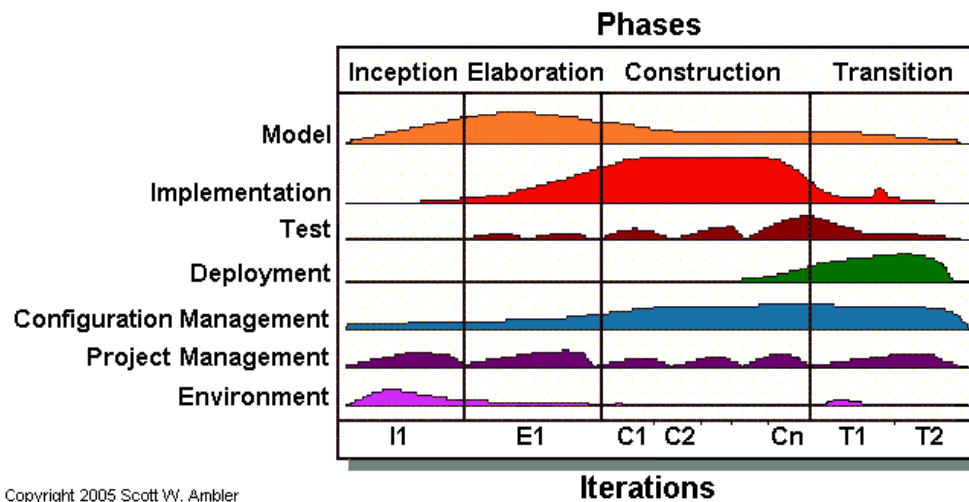


Figura 3. Ciclo de vida de AUP.

4.2 Implementación de Red Neuronal

Cuando se aborda un problema utilizando el Perceptron Multicapa, existen dos pasos base para la implementación de la arquitectura de la red, según lo definido en [Isasi2005], los cuales se detallan a continuación:

- Determinación de la función de activación.

Su elección depende del recorrido deseado, no influyendo significativamente en la capacidad de resolución del problema.

- Definición de la cantidad capas y neuronas por capa.

La definición de neuronas en la capa de entrada, así como en la capa de salida viene dada por las variables que definen el problema. Para la determinación de la arquitectura se parte con el criterio del diseñador, apoyándose en procedimientos de prueba y error, hasta encontrar la arquitectura adecuada para resolver el problema.

Si bien el número de capas ocultas puede influir en el comportamiento de la red, la capacidad de generalización de la red permite al perceptron multicapa resolver de manera adecuada el mismo problema con distintas arquitecturas de red.

5 RECURSOS

5.1 Equipos

Desarrollo

HP Pavilion dv 2200

Procesador Intel Core Duo

2Gb de memoria en RAM

WiFi

Clientes

Para este caso se implementará para un cliente que cuenta con un equipo portátil de las siguientes características:

HP Pavilion dv 2622

Procesador Intel Core Duo

1Gb de memoria en RAM

WiFi

5.2 **Software**

5.2.1 Herramientas de desarrollo

Microsoft Windows Vista

Sistema operativo que provee las plataformas base para la ejecución de las herramientas productivas y de desarrollo de este proyecto.

Visual Studio 2008

Herramienta Visual que permitirá el diseño de las aplicaciones, mediante la utilización del lenguaje de programación C# y elementos para formularios para Windows.

DB4o

Base de objetos que dará soporte al almacenamiento de datos y su posterior manejo.

Power Designer

Herramienta para el diseño de diagramas requeridos en la metodología seleccionada.

Fireworks MX 2004

Aplicaciones orientadas al diseño gráfico de elementos a utilizar en la interfaz de usuario.

5.2.2 Usuarios

Sistema Operativo Windows Vista.

6 INICIO DEL PROYECTO

Este proyecto nace de la necesidad de llevar a un nivel superior los análisis realizados por EEI Ltda, pudiendo contar con una cantidad de patrones de fallos incremental que le permita realizar diagnósticos acertados y prestar un servicio de primer nivel a sus clientes.

6.1 Investigación previa

6.1.1 Teoría eléctrica respecto de las variables analizadas

Para determinar cuál es la implicancia de cada una de las variables eléctricas analizadas, se detallarán a continuación los fundamentos teóricos detrás de cada una de ellas. Estos conceptos están basados en las definiciones obtenidas de [Nilsson2005].

6.1.1.1 Tensión y corriente.

El concepto de carga eléctrica es la base para describir todos los fenómenos y efectos eléctricos pueden atribuirse a la separación de las cargas y al movimiento de las mismas.

En teoría de circuitos, la separación de las cargas crea una fuerza eléctrica (tensión), mientras que el movimiento de una carga crea un fluido eléctrico (corriente).

Los conceptos de tensión y corriente son útiles desde el punto de vista del ingeniero porque pueden expresarse en forma cuantitativa. Cuando se separan cargas positivas y negativas, se gasta energía en el proceso. La tensión es la energía por unidad de carga creada por la separación. Su expresión en forma diferencial se describe a continuación.

$$v = \frac{dw}{dq}$$

Donde v= tensión en voltios
 w=energía en julios
 q=carga en culombios

Figura 4. Definición de la tensión eléctrica.

Los efectos eléctricos causados por las cargas en movimiento dependen de la tasa de flujo de la carga. La tasa de flujo de la carga se conoce como corriente eléctrica, que se expresa como

$$i = \frac{dq}{dt}$$

Donde i= corriente en amperios
 q=carga en culombios
 t=tiempo en segundos

Figura 5. Definición de corriente eléctrica.

6.1.1.2 Potencia y energía

Los cálculos de potencia son también importantes en el análisis de circuitos. Una de las razones es que, aunque la tensión y la corriente son variables útiles en el análisis y diseño de sistemas eléctricos, la salida útil del sistema es a menudo de naturaleza no eléctrica, y esta salida puede expresarse convenientemente en términos de potencia y energía. Otra razón es que todos los dispositivos prácticos tienen limitaciones en cuanto a la cantidad de potencia que pueden manejar. Por tanto, en el proceso de diseño, los cálculos de tensiones y corrientes no son suficientes por sí mismos.

Ahora se establecerá la relación entre potencia y la energía con la tensión y la corriente. Matemáticamente, la energía por unidad de tiempo se expresa en forma derivada, de la manera siguiente:

$$p = \frac{dw}{dt}$$

Donde p= potencia en vatios
 W=energía en julios
 t=tiempo en segundos

Figura 6. Definición de potencia.

La potencia asociada con el flujo de carga se sigue directamente de la definición de la tensión y de la corriente, según las ecuaciones detalladas en las figuras 3 y 4 y es:

$$p = \frac{dw}{dt} = \left(\frac{dw}{dq}\right)\left(\frac{dq}{dt}\right) = vi$$

Figura 7. Ecuación de la potencia en función de la tensión y la corriente

La Figura 7 muestra que la potencia asociada con un elemento de circuito básico es, simplemente el producto de la corriente que atraviesa el elemento por la tensión que cae en él.

6.1.1.3 Fuentes sinusoidales

Una fuente de tensión sinusoidal produce una tensión que varía sinusoidalmente con el tiempo. Una fuente de corriente sinusoidal produce una corriente que varía sinusoidalmente con el tiempo. A la hora de analizar las funciones sinusoidales, vamos a utilizar una fuente de tensión, pero los resultados también deben ser aplicados a fuentes de corriente.

Se expresará una función sinusoidal mediante la función seno o la función coseno. Aunque las dos resultan adecuadas, no usaremos ambas formas funcionales simultáneamente. Para el análisis, vamos a utilizar la función coseno, de modo que se describirá una tensión sinusoidal de la siguiente forma:

$$v = V_m \cos(\omega t + \varphi)$$

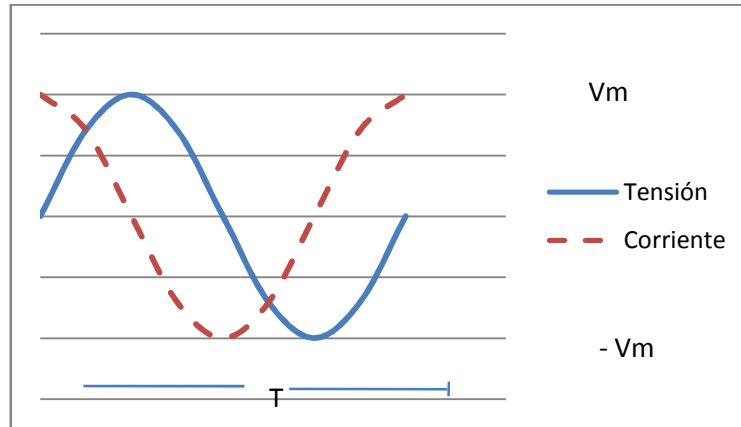


Figura 8. Ecuación y representación grafica de tensión y la corriente con desfase.

6.1.1.4 Potencia Instantánea

La potencia Instantánea se mide en vatios cuando la tensión esta en voltios y la corriente en amperios. En primer lugar, se detallan las expresiones correspondientes para cada término:

$$v = V_m \cos(\omega t + \theta_v)$$

$$i = I_m \cos(\omega t + \theta_i)$$

Figura 9. Funciones de la tensión y la corriente en régimen sinusoidal.

En las ecuaciones, θ_v es el ángulo de fase de la tensión y θ_i es el ángulo de fase de la corriente. Al operar en régimen sinusoidal se puede elegir cualquier instante conveniente como tiempo cero. Los diseños de ingeniería de transmisión de grandes cantidades de potencia, utilizan como origen de tiempo un instante en que la corriente esté pasando por un máximo positivo. Este sistema de referencia requiere un desplazamiento de la tensión y de la corriente según un ángulo θ_i .

6.1.1.5 Potencia Media y Reactiva

Según lo descrito en la figura 8, podemos reescribir la ecuación como sigue:

$p = P + P \cos 2\omega t + Q \sin 2\omega t$	
donde	
Potencia Media (real)	$P = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i)$
Potencia Reactiva	$Q = \frac{V_m I_m}{2} \sin(\theta_v - \theta_i)$

Figura 10. Ecuación de la potencia en régimen sinusoidal

P se denomina potencia media y Q se denomina potencia reactiva. La potencia también se llama, en ocasiones, potencia real, porque describe una potencia

que es transformada en un circuito de energía eléctrica a energía no eléctrica dentro de un circuito. Aunque los dos términos son intercambiables, se utilizará principalmente el término potencia media.

La potencia media asociada con las señales sinusoidales es el promedio de la potencia instantánea a lo largo de un periodo, lo que se representa a continuación:

$$P = \frac{1}{T} \int_{t_0}^{t_0+T} p \, dt$$

Figura 11. Ecuación de la potencia media.

En la ecuación expresada, T es el periodo de la función sinusoidal. Los límites de la ecuación implican que se puede iniciar el proceso de integración en cualquier instante t_0 que resulte conveniente, pero se debe terminar la integración exactamente un periodo después.

6.1.1.6 Factor de potencia

El ángulo $\theta_v - \theta_i$ juega un papel importante en el cálculo tanto de la potencia media como de la potencia reactiva, y se lo denomina como ángulo de factor de potencia. El coseno de este ángulo se denomina factor de potencia (abreviado fp) y el seno de este ángulo se denomina factor reactivo (abreviado fr).

Conocer el valor del factor de potencia no permite determinar el valor del ángulo del factor de potencia, porque $\cos(\theta_v - \theta_i) = \cos(\theta_i - \theta_v)$. Para describir completamente este ángulo, se utilizarán las fases descriptivas Factor de Potencia en Retardo y Factor de Potencia en adelanto. Un factor de potencia en retardo implica que la corriente está retrasada con respecto a la tensión, lo que quiere decir que se tiene una carga inductiva. Un factor de potencia en adelanto implica que la corriente está adelantada con respecto a la tensión, por lo que se tendrá una carga capacitiva. Tanto el factor de potencia como el factor reactivo son magnitudes que facilitan la descripción de las cargas eléctricas.

6.1.1.7 Potencia Compleja

La potencia compleja es la suma compleja de la potencia real y de la potencia reactiva. Desde el punto de vista de las dimensiones, la potencia compleja tiene las mismas que la potencia media o la potencia reactiva. Sin embargo, para distinguir la potencia compleja de los otros dos tipos de potencia, se utilizara como unidades los voltio-amperios (VA). Así, usaremos los voltio-amperios para la potencia compleja, lo vatios para la potencia media y los vars para la potencia reactiva.

La magnitud de la potencia compleja se denomina potencia aparente, lo cual se grafica en la siguiente fórmula:

$$|S| = \sqrt{P^2 + Q^2}$$

Figura 12. Ecuación de potencia aparente.

La potencia aparente, como la potencia compleja, se mide en voltio-amperios. El requisito de consumo de potencia aparente, o de voltio-amperios, para un dispositivo diseñado para convertir energía eléctrica en energía no eléctrica es más importante que el requisito de potencia media. Aunque la potencia media representa la salida útil del dispositivo conversor de energía, la potencia aparente representa la capacidad en voltio-amperios requerida para suministrar la potencia media.

Muchos aparatos útiles y la mayoría de las cargas industriales funcionan con un factor de potencia en retardo. El factor de potencia de estas cargas se corrige en ocasiones añadiendo un condensador al propio dispositivo o conectando condensadores entre las dos líneas que alimentan la carga.

6.1.2 Normativa eléctrica Chilena Vigente (NCH 2003-04)

Para los efectos de estimación de cumplimiento normativo, se utilizará la normativa eléctrica vigente NCH 2003-04, la cual en el artículo 243° define lo siguiente:

“La norma técnica fijará las magnitudes de la tensión nominal en 50 Hz. El proveedor del servicio deberá indicar explícitamente, a cada usuario, la tensión en el punto de conexión entre ambos, en adelante punto de conexión.

Las variaciones u holguras permitidas de la tensión nominal en el punto de conexión, serán las siguientes:

En Baja Tensión (BT): Excluyendo periodos de interrupciones de suministro, el valor estadístico de la tensión medido de acuerdo a la norma técnica correspondiente, deberá estar dentro el rango de -7.5% y 7.5% durante el 95% del tiempo de cualquiera semana del año o de siete días consecutivos de medición y registro.”

6.1.3 Equipo de monitoreo eléctrico DM-II

El equipo disponible para la captura de los datos eléctricos es el DM-II de la marca Amprobe. Este equipo cuenta con la capacidad de capturar información de circuitos de 1 a tres fases y sus registros son en formato TRUE RMS.

Las variables capturadas por el equipo, cumplen con los requerimientos técnicos detallados a continuación:

Entradas	
Voltaje	Tres canales para fases mas un común, soporte hasta 1000 V.
Corriente	Cuatro canales (I1, I2, I3, I4) DM-CT: 1 a 1000 A $\pm$ 1% de lectura
Rangos	AC Voltaje (True RMS) 5 a 600 Vrms AC Corriente (True RMS) 1 a 1000 Arms
Exactitud	Voltaje $\pm$ 1% en lectura Corriente $\pm$ 1% en lectura

Mediciones

	True RMS Voltaje y corriente
	En potencia Aparente (VA), Reactiva (VAR), Real (W) y Factor de potencia
	Energía kWh y demanda (kW)
Modos de grabación	Intervalo discreto y continuo
	1segundo ,5 segundos,15 segundos ,30 segundos,
	1 minuto, 5 minutos, 15 minutos y 30 minutos.

Tabla 2. Características técnicas equipo DM-II.

Para la descarga de los datos se realiza mediante una interfaz RS-232 con la aplicación Download Suite, de Amprobe.

6.1.4 Sistemas de reconocimiento de patrones.

Uno de los desafíos más grandes de nuestra era ha sido la construcción de sistemas inteligentes, a lo que la Inteligencia Artificial (IA) ha surgido como área de estudio del diseño e implementación de los mismos.

Podemos diferenciar dos áreas de trabajo bastantes definidas en la IA según [Isasi2005], la primera se conoce como IA simbólica, la cual define el problema a resolver y diseña el sistema capaz de resolverlo siguiendo esquemas fijados por la disciplina, como lo hacen por ejemplo los Sistemas Expertos, que utilizan

mecanismos de inferencia parecidos a que utilizamos al razonar para posteriormente sacar conclusiones.

La segunda área de la IA es la conocida como Subsimbólica, en la cual se realizan diseños genéricos de alto nivel que cuentan con la capacidad de adaptarse y construir por sí mismo una solución al problema. Dentro de este campo encontramos las Redes Neuronales. El objetivo de las Redes Neuronales es llegar a diseñar máquinas con elementos neuronales de procesamiento paralelo, con la capacidad de adaptarse a las condiciones del problema.

Dadas las condiciones del problema abordado y lo anteriormente expuesto, las Redes Neuronales cumplen con la capacidad de adaptarse a las condiciones del problema, que inicialmente no cuentan con una clara definición, por lo que las capacidades dinámicas de esta herramienta de reconocimiento de patrones cumple con los requerimientos de la solución propuesta.

6.1.5 Redes Neuronales

Bajo el nombre genérico de “Red Neuronal” se engloban diferentes estructuras de procesamiento paralelo distribuido, algunas de ellas biológicamente inspiradas, en las que el número de elementos simples de cómputo no lineal se interconectan en forma más o menos densa. Estas estructuras son de utilidad en reconocimiento de patrones, modelado y filtrado no lineal de señales, entre

muchas otras aplicaciones. En general, las redes presentan algún tipo de plasticidad en su estructura que permite sintonizar su operación a fin de optimizar una figura de mérito, como por ejemplo alguna medida de distancia al comportamiento deseado. De esta forma, la red puede aprender a realizar una tarea u operación. Esta característica es de gran utilidad allí donde el conocimiento del problema a resolver es impreciso o variante en el tiempo.

6.1.6 Generalidades de las Redes Neuronales

Según lo expuesto en [Sierra2006], la red es un operador no lineal formado por un conjunto de elementos simples de procesamiento o neuronas, que se conectan entre si y con el medio externo a través de conexiones o sinapsis, que poseen un peso asociado, así como se expone en el siguiente diagrama.

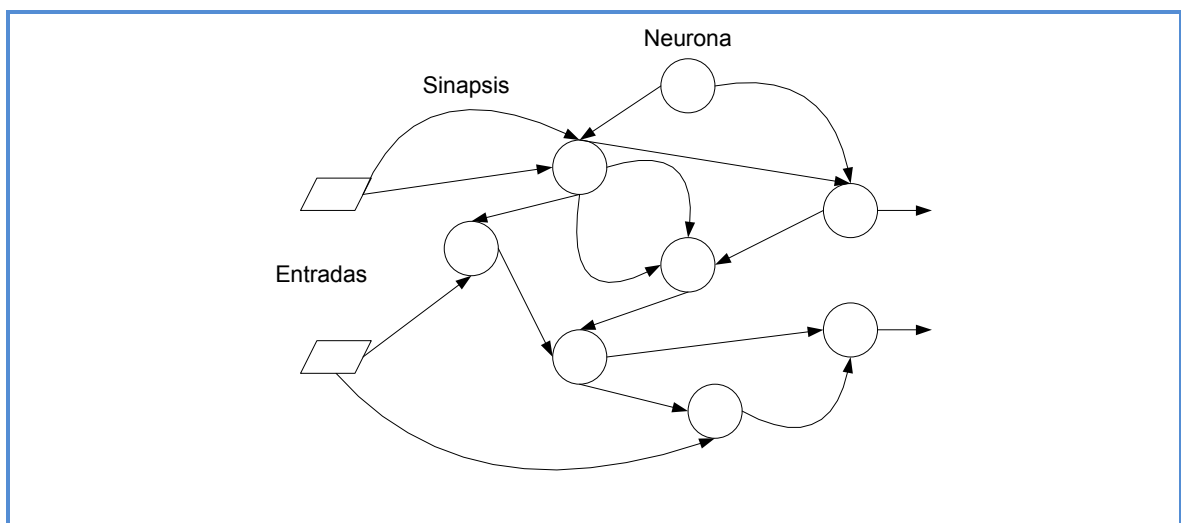


Figura 13.Red neuronal genérica

La siguiente figura ilustra el modelo habitualmente utilizado para una neurona artificial. Esencialmente, la operación de la neurona involucra el cálculo de una función de entrada h , a partir de las señales que ingresan a la misma, y la posterior aplicación de una función de activación g , en general no lineal. En algunos casos, existe una entrada de umbral (offset), que puede ser siempre asociada a una entrada adicional de valor 1 y peso igual al umbral. La salida de cada neurona dependerá, entonces, de sus señales de entrada, del peso asociado a cada entrada, y de las características de las funciones de entrada y activación.

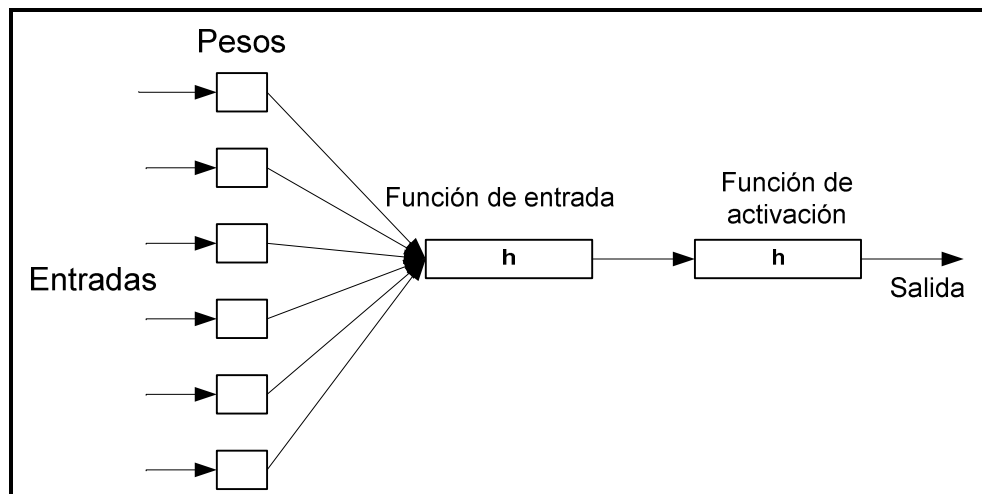


Figura 14. Red neuronal Artificial

En la mayoría de los casos, los elementos plásticos de la red son los pesos de las conexiones, y el mecanismo utilizado para adaptar esos pesos se conoce como algoritmo de entrenamiento o aprendizaje. Los elementos citados

permiten clasificar los diferentes tipos de redes encontradas en la literatura según los siguientes aspectos:

- Número y disposición de las neuronas.

En una red de tipo feedforward las neuronas están organizadas en capas, y no hay conexiones recurrentes o realimentación de la salida sobre la entrada o entre neuronas internas. Existen, entonces, capas directamente conectadas a la entrada del sistema, capas directamente conectadas a la salida, y capas ocultas o intermedias. En una red recurrente, por el contrario, existe realimentación entre neuronas, lo que determina la aparición de una dinámica más compleja.

- No-linealidad presente en cada neurona.

La salida de cada neurona es, en general, una función no lineal de la función de entrada. Si bien pueden utilizarse salidas lineales, se verá en las siguientes secciones que su capacidad de modelado es moderada (básicamente la red es un combinador o clasificador lineal). La no linealidad puede ser suave (esto es, con derivada continua con respecto a su entrada) o abrupta. Las características de la no linealidad tienen directa relación con el algoritmo de aprendizaje a ser utilizado en la fase de entrenamiento.

- Red de interconexiones.

Los valores de los pesos de la red de interconexiones imponen la influencia que la salida de una neurona tiene sobre las otras. Esta influencia puede ser excitatoria, cuando tiende a aumentar el valor de la activación, o inhibitoria, cuando tiende a disminuirlo. En otras redes, como las de base radial, los pesos especifican las coordenadas donde la no-linealidad de cada neurona está centrada.

- Algoritmo de entrenamiento.

En los algoritmos supervisados, existe un tutor o salida deseada que especifica que valor debería tener la salida de la red para cada posible entrada dentro de un conjunto de datos de entrenamiento. El error entre la salida actual de la red y la deseada es utilizada por el algoritmo para sintonizar el valor de los pesos de la red. En los algoritmos no supervisados, por el contrario, no existe un tutor y el objetivo del algoritmo de entrenamiento es detectar alguna regularidad, estructura o característica en particular de los datos, de forma que la red se convierte en un detector de características (feature detector). Los algoritmos de entrenamiento son, generalmente procesos de optimización, a menudo heurísticos, y tales sufren de problemas de convergencia y existencia de mínimos locales en la superficie de búsqueda.

- Comportamiento estático y dinámico

Una vez entrenadas, las redes que no poseen memoria se comportan como combinadores no lineales. Por el contrario, la existencia de elementos capaces de almacenar valores de estado pasado de la red la transforman en un sistema dinámico no lineal, lo que amplía significativamente sus capacidades de moldeado y agrega complejidad a su comportamiento y tratamiento matemático.

- Generalización

Una red neuronal es capaz de generalizar cuando existe una estrecha relación entre la salida deseada y la salida calculada por la red para muestras o ejemplos que no han sido usados en el entrenamiento, entendiendo por éste el proceso de ajuste de los pesos sinápticos de la red en base a minimizar una determinada función de error o maximizar el éxito en la decisión de ésta mediante un algoritmo de aprendizaje concreto.

Por esta razón, se debe disponer de un conjunto de datos, ejemplos o patrones para ajustar los pesos de la red a un determinado algoritmo de aprendizaje (conjunto de entrenamiento), así como de un conjunto de datos, ejemplos o patrones que la red nunca ha utilizado (conjunto de generalización). La generalización es la cualidad que poseen las redes neuronales artificiales para extraer información de los datos presentados

en su entrenamiento y extender ese conocimiento más allá de esas situaciones conocidas.

6.2 Preparación del entorno de desarrollo.

Con los antecedentes estudiados del problema, se procedió a buscar las herramientas de software que más se acoplaran con los requerimientos del proyecto.

Como base para el desarrollo se eligió Visual Studio 2008 .NET, realizando las implementaciones en C# y para el almacenamiento de los datos se optó por Db4o ya que se acopla perfectamente al desarrollo orientado a objetos.

Para la implementación de la red neuronal fueron seleccionadas las clases definidas en el proyecto Aforge [Kirirlov2005]. Como proceso de verificación del correcto funcionamiento de esta implementación, se compararon los resultados de las implementaciones de redes de similares características en Matlab y en C#. Se procedió a entrenar y probar ambas con un set de datos en común y demostraron el siguiente comportamiento con 30 patrones y 10 pruebas, obteniendo un margen de error no mayor al 3%.

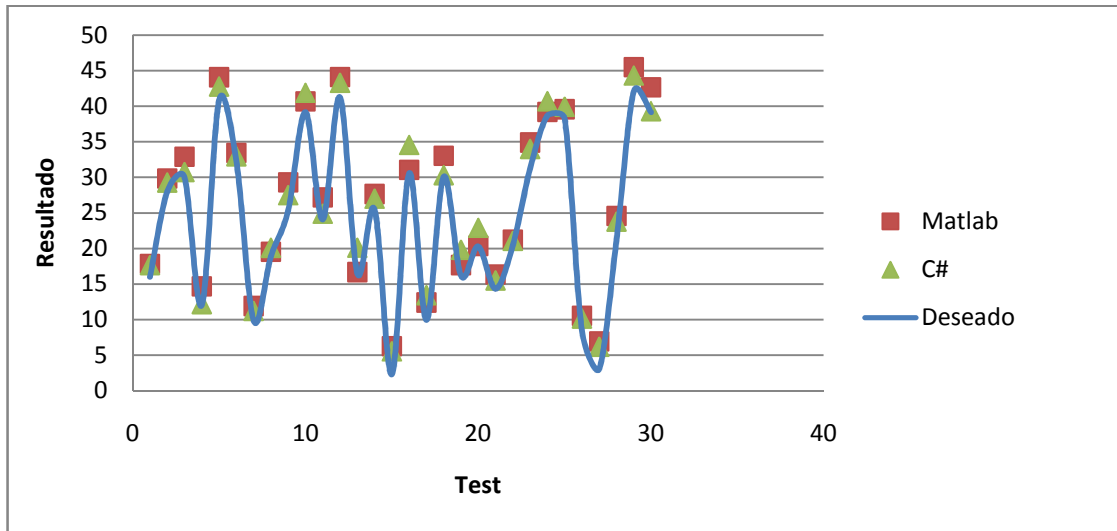


Figura 15. Comparación de resultados obtenidos con Matlab y C#.

6.3 Modelado Inicial

En el modelado inicial se desarrollaron las primeras aproximaciones del diagrama de casos de uso, así como se utilizaron diagramas de flujo de datos para la identificación de las reglas de negocio. Se identificaron preliminarmente los requerimientos técnicos de la aplicación, así como se incluyeron las primeras especificaciones en el Glosario de Términos.

Los diagramas derivados de este modelado inicial fueron mantenidos durante las etapas siguientes, por lo que su exposición se realiza en los capítulos posteriores.

7 ELABORACIÓN

En esta etapa, se busca tener las definiciones de arquitectura del sistema, diagrama de clases de la base de objetos junto con el diagrama de casos de uso del sistema. Se desarrollan también los prototipos de importación de datos desde el archivo Excel así como su almacenamiento en la base de objetos DbTemp.

7.1 Definición de requerimientos

7.1.1 Modelo del Dominio

A continuación se expone el Modelo del Dominio de la solución a implementar, como una primera representación de los requerimientos de acuerdo a lo expuesto en [Ambler2005]. Este diagrama es el resultado del estudio de requerimientos realizado en la etapa de inicio según lo indicado por la metodología.

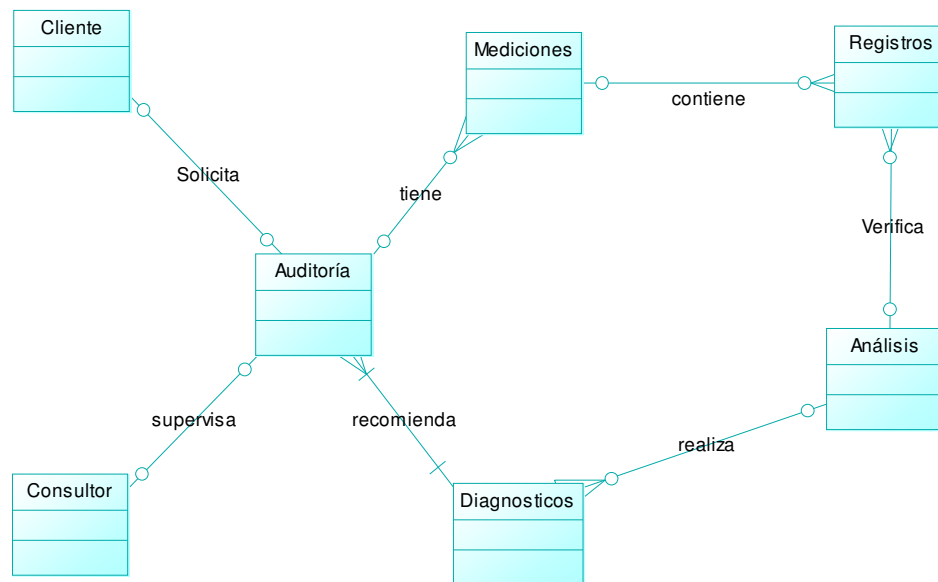


Figura 16. Modelo del Dominio

7.1.2 Glosario de términos

A continuación se describen los términos utilizados en el desarrollo del presente proyecto. Este diccionario de datos y nomenclatura es definido para contar con una definición estandarizada de los términos empleados para definir los requerimientos del sistema.

Los términos se detallan alfabéticamente a continuación:

Amper: Unidad de medida para la corriente.

Análisis estadístico: Consiste en el análisis detallado de cada una de las variables eléctricas medidas, con la finalidad de obtener los valores máximos,

mínimos, promedio y la desviación estándar por cada una de las variables capturadas.

Anomalía: Registro derivado de los análisis realizados por el sistema, el cual es almacenado en cuanto a la magnitud de desviación respecto a la condición normal de funcionamiento.

Auditoria: Consiste en el proceso de verificación del funcionamiento de un sistema eléctrico definido, para lo cual se hacen mediciones de los parámetros eléctricos para ser posteriormente analizados con la finalidad de encontrar anomalías.

Banco de Condensadores: Sistema eléctrico para la compensación de energía reactiva presente en la red, con la finalidad de optimizar el uso de la energía eléctrica y eliminar multas derivadas por este concepto, según lo indica la normativa eléctrica vigente.

Cliente: Persona natural o jurídica que cuenta con instalaciones eléctricas del tipo residencial o industrial, que requiere los servicios de la empresa para solucionar inconvenientes con sus sistemas eléctricos, así como optimizar el uso de los mismos.

Condiciones de Borde: Detalle de las características técnicas presentes en la instalación eléctrica analizada.

Consultor: Ingeniero certificado para la realización de consultorías en el área de eficiencia energética según los estándares definidos por el INN.

Corriente de neutro: Análisis que busca identificar el momento en el cual la corriente registrada en el neutro supera al 10% de la corriente presente en las fases, lo que constituye un riesgo eléctrico.

Desbalance de carga: Identificación del instante en el cual el cociente entre las fases supera el 30%.

Desbalance de tensión: Identificación del instante en el cual el cociente entre las fases supera al 2%.

Diagnóstico: Resultado del análisis realizado por el sistema en el cual se indican posibles causas de las anomalías registradas, así como las soluciones para las mismas.

Factor de potencia: Cociente entre la potencia y la energía reactiva.

Fecha de certificación: Fecha en la que el consultor recibió su última certificación por parte del INN.

Generación en Hora Punta (GHP): Generación de energía eléctrica por medio de equipos electrógenos, con la finalidad de abaratar los costos derivados del

consumo de energía en la red pública en horas determinadas como de consumo en punta.

Intervalo: Unidad de tiempo en la cual son capturados los registros.

Malla Baja Tensión (BT): Estructura de cobre enterrada en la tierra que cumple la función de proteger los equipos eléctricos conectados a la red de baja tensión.

Malla Media Tensión (MT): Estructura de cobre enterrada en la tierra que cumple la función de proteger los equipos conectados a la red de media tensión.

Medición: Conjunto de registros capturados en una red eléctrica, en el cual se cuantifican las variables como tensión, corriente, potencia y factor de potencia.

Potencia: Unidad de medida eléctrica con la que se mide el trabajo generado por la energía eléctrica. Su unidad de medida es el Watt.

Registro: Conjunto de datos que contiene las variables eléctricas registradas en un momento definido.

Secuencia de Fases: Análisis que identifica la condición de mala secuencia de fases en la instalación analizada.

Sobre Carga: Condición en la cual se ingresan consumos superiores al 40% del promedio registrado en los últimos cinco registros capturados.

Sugerencia de análisis: Resultado del proceso de la información estadística en la red neuronal entrenada para determinar los análisis necesarios según condiciones predefinidas.

Suministro: Proveedor de la energía eléctrica. Este puede ser público (ej. Saesa, Crell) o privado en el caso que cuente con generación propia ya sea por equipos electrógeno, turbina hidráulica, generador eólico entre otros.

Volt: Unidad para medir la tensión eléctrica.

7.1.3 Diagramas de Casos de Uso del Sistema.

Se elaboró el siguiente diagrama de casos de uso de Sistema de Análisis Eléctrico, como recurso inicial para la identificación de las clases componentes del sistema, así como sus actores.

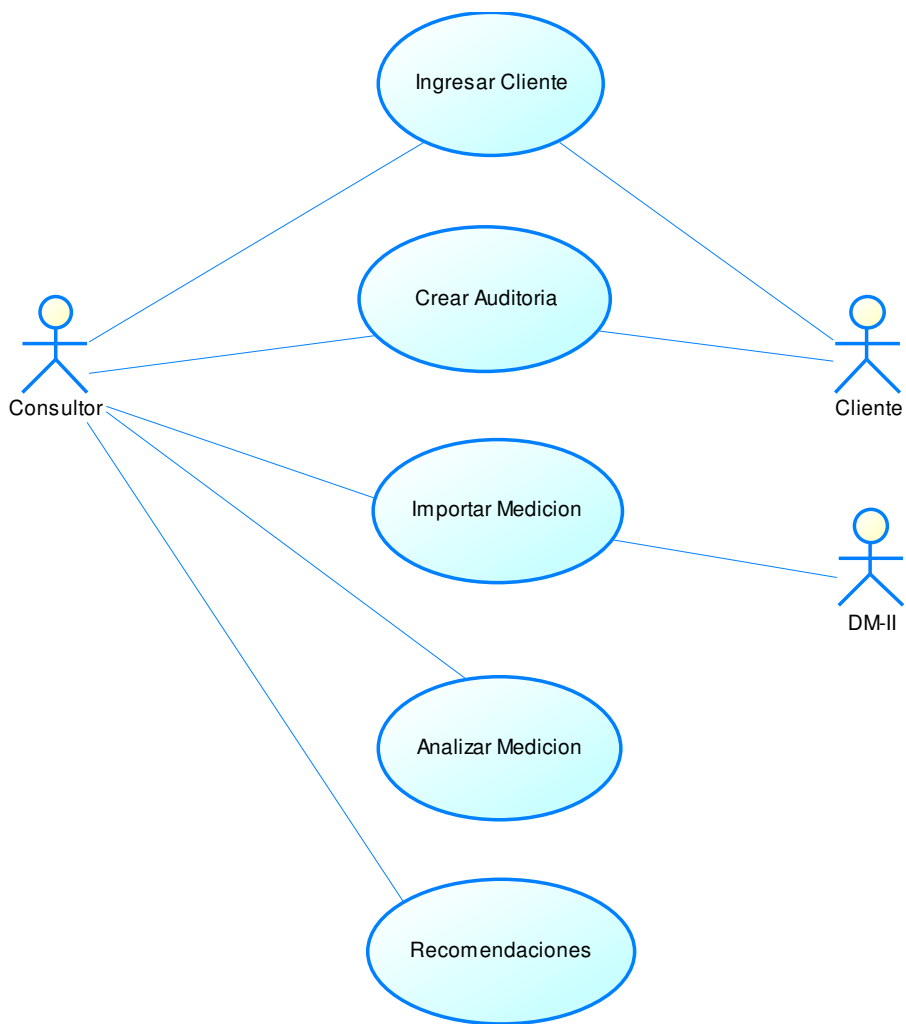


Figura 17. Casos de uso del Sistema

7.1.4 Especificación de los Casos de Uso

Para la especificación detallada de los requerimientos del sistema se han realizado las especificaciones de los casos de uso para Importar Medición,

Analizar Medición y Recomendaciones, basándose en las sugerencias de análisis detallados expuestas en [Ambler2005].

Caso de Uso: Importar Medición

Actores: Consultor

Descripción: Los datos capturados por el equipo DM-II y posteriormente exportados a un archivo Excel, son importados a la aplicación de análisis y almacenados en la base de objetos según el formato definido en la clase Registro.

Pre Condiciones: Datos exportados desde la aplicación propietaria a archivo Excel en formato 2003.

Flujo Normal: El consultor debe seleccionar el archivo desde donde se importarán los datos. El sistema extrae los datos y verifica si el formato del archivo corresponde a variables eléctricas capturadas con el equipo DM-II. Posteriormente normaliza los datos para ingresarlos a una base de objetos temporal la cual se utilizará para el posterior análisis de los datos. Los datos son almacenados con la clase Registro, la cual consta de la fecha de la captura y los campos asociados a cada fase en las variables: Tensión, corriente, potencia, energía reactiva, energía aparente y factor de potencia.

Flujo Alternativo:

1. El archivo ingresado no corresponde a un archivo de Excel por lo cual el sistema le indicará al consultor que debe seleccionar un nuevo archivo de datos.
2. El archivo de Excel no cumple con el formato de las mediciones del equipo, por lo que la aplicación indicará al consultor que debe seleccionar un nuevo archivo con el formato de medición válido.

Post Condiciones: Base de objetos con los registros a analizar.

Caso de Uso: Analizar medición.

Actores: Consultor

Descripción: Consiste en el procesamiento de la información importada a la base de objetos, con la finalidad de reconocer patrones de anomalías e identificar adecuadamente los análisis requeridos para la correcta identificación de las fallas capturadas.

Precondiciones: Datos importados y normalizados e información de condiciones de borde ingresada.

Flujo Normal: Se realiza el análisis estadístico de la medición, en el cual se obtiene los mínimos, máximos, media y desviación estándar de cada una de las variables capturadas por el equipo. Posteriormente se ejecuta el proceso que determina las precondiciones que constituyen los parámetros de entrada a la red neuronal, la que a su vez indica cuales son los análisis requeridos. Los análisis indicados serán ejecutados en cada uno de los registros.

Una vez entregado el detalle de los análisis sugeridos por la red neuronal, el consultor debe validar lo sugerido por el sistema o agregar otro análisis no sugerido e indica el inicio de los procedimientos de análisis.

Flujo alternativo:

1. La primera red neuronal sugiere no realizar ningún análisis detallado debido a que la instalación analizada no presenta condiciones anómalas, ante lo cual se cierra el análisis en este punto.
2. El consultor estima que se debe incluir algún/os de los análisis no sugeridos por la aplicación. Los nuevos análisis indicados son almacenados como nuevo patrón de fallo para entrenar nuevamente la red neuronal incluyendo el nuevo patrón registrado.
3. Se ejecuta el análisis de **Desbalance de Tensión**.
Se buscan los registros en los cuales las diferencias entre los valores capturados para la tensión sean mayores al grado de tolerancia indicado

como parámetro de la función. Como regla general y como indica la documentación técnica este nivel debe ser inferior al 2%.

4. Se ejecuta el análisis Desbalance de Corriente.

Mediante análisis de los registros se buscan los registros en los cuales la diferencia entre las corrientes registradas sea superior al valor ingresado como parámetro de la función. Se utiliza como desbalance base un 30% de variación entre las fases.

5. Se ejecuta el análisis Secuencia de Fases.

Las fases deben encontrarse en una correcta secuencia para el normal funcionamiento de los equipos. Esta condición es determinada por el sentido de las corrientes correspondientes a cada una de ellas. Al momento de registrar una corriente con valor negativo, esto indica que la secuencia de fase es incorrecta.

6. Se ejecuta el análisis Sobre Carga.

El ingreso de grandes cargas al sistema eléctrico provoca una serie de variaciones que pueden afectar el desempeño de los otros equipos presentes en la red, por lo que se registran las condiciones en que se identifican ingresos de carga al sistema por sobre el 30% del valor de la media registrada en las últimas 5 mediciones capturadas.

7. Se ejecuta el análisis de Cumplimiento Normativo.

Según los estándares definidos en la normativa eléctrica se verifica la condición de cumplimiento de dicha norma técnica.

8. Se ejecuta el análisis de Energía Reactiva.

Un valor negativo en la corriente reactiva indica problemas en la configuración del banco de condensadores presente en la instalación, por lo que se registra esta condición para lograr asociar su ocurrencia con otras condiciones anómalas presentes en la red.

9. Se ejecuta el análisis de Potencia Contratada.

Se determinan los casos en los cuales la potencia requerida por el sistema se encuentra por sobre el 70% de la potencia contratada.

Post Condiciones:

Detalle de las anomalías encontradas en cada uno de los registros analizados.

Validación de las sugerencias de análisis realizado por el sistema.

Caso de Uso: Mejoras

Actores: Consultor

Descripción: Toma las anomalías detectadas en los registros y las analiza para la detección de patrones predefinidos de fallas identificados mediante la utilización de una red neuronal entrenada bajo supervisión. Según los patrones identificados realiza las indicaciones relativas a la anomalía y su solución.

Precondiciones: Análisis de anomalías según las sugerencias basadas en análisis estadístico junto con la validación de las mismas por parte del consultor y condiciones pre definidas en la red neuronal encargada de realizar las sugerencias.

Flujo Normal: Se analizan los registros de anomalías y de acuerdo a los patrones predefinidos de anomalías eléctricas utilizados para el entrenamiento de la red neuronal, se identifican y categorizan de acuerdo a su índice de recurrencia. Una vez procesados todos los registros, se categorizan las anomalías encontradas y se procede a realizar las sugerencias de mejora al sistema eléctrico analizado.

Flujo Alternativo:

1. No se identifican patrones de falla, por lo que se sugiere realizar una nueva medición y realizar el proceso de análisis nuevamente, para descartar errores en la captura de los datos.
2. Según las anomalías detectadas, se utilizan sub redes especializadas para cada tipo de anomalía con la finalidad de determinar con mayor detalle, las causas de la falla detectada y orientar de mejor manera al consultor en su posible solución.

Post Condiciones:

Detalle de las recomendaciones realizadas por el sistema.

7.2 Diagrama de Clases de Base de Objetos

De acuerdo a los requerimientos obtenidos, se identificaron las siguientes clases y sus relaciones:

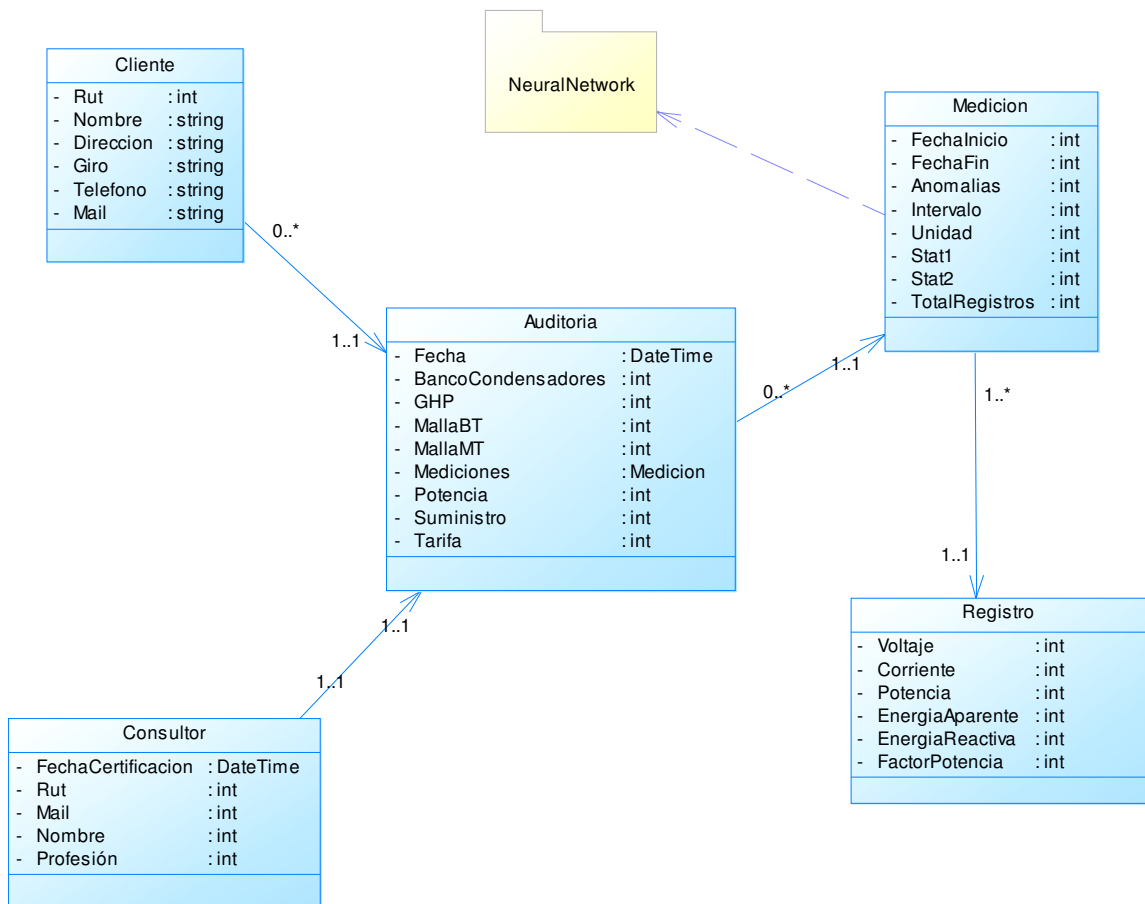


Figura 18. Diagrama de clases de Base de Objetos.

Para la implementación de la red neural, se establecieron las clases siguiendo los patrones de implementación definidos en [Kirirlov2005], la cual provee la estructura necesaria para la construcción de red neuronal con sus capas y nodos correspondientes.

Se implementaron las funciones de activación sigmoidal y sigmoidal bipolar para la realización de las pruebas de comportamiento de la red utilizando ambas configuraciones en el proceso de construcción.

Para el proceso de aprendizaje supervisado se implementó la clase Back Propagation, la cual contiene la implementación del método de propagación de error hacia atrás.

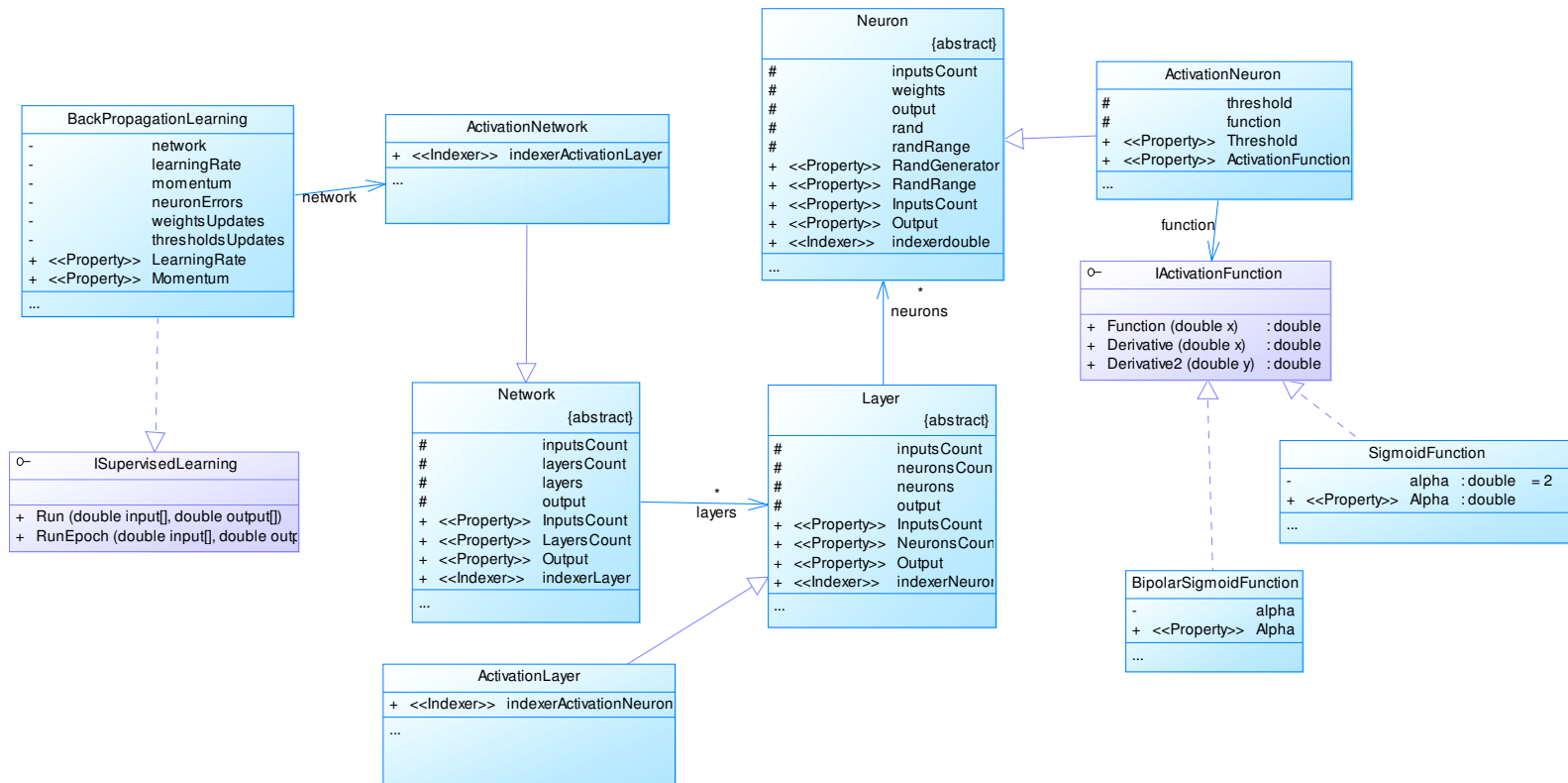


Figura 19. Diagrama de clases Red Neuronal

7.3 Detalle de Clases y relaciones

A continuación se detallan las clases del sistema, así como también sus atributos y relaciones.

Nombre Clase	Descripción
Cliente	Detalle de quien contrata los servicios de análisis
Consultor	Personal de EEI que está encargado del ejecutarla auditoría solicitada por el cliente
Auditoria	Proceso mediante el cual se realizan mediciones a las instalaciones eléctricas de los clientes con la finalidad de determinar la existencia de anomalías.
Medición	Conjunto de registros capturados en forma secuencia en intervalos de tiempo fijo
Registro	Conjunto de variables eléctrica capturadas en un instante determinado y que representa el comportamiento eléctrico de la instalación en un momento determinado

Tabla 3. Detalle de clases.

Nombre Asociación	Descripción	Multiplicidad
Solicita Auditoria	El cliente solicita la ejecución de una auditoria, con lo cual se crea una nueva instancia de la clase auditoria y se le asigna al cliente	0..*
Consultor Encargado	Al momento de crear una clase auditoria esta debe tener un consultor encargado de la ejecución de la misma	1..*
Mediciones Realizadas	Proceso mediante el cual se realizan mediciones a las instalaciones eléctricas de los clientes con la finalidad de determinar la existencia de anomalías.	0..*
Registros capturados	Los registros importados desde el archivo Excel pertenecen a una medición.	1..1

Tabla 4. Detalle de asociaciones.

Nombre Atributo	Tipo	Descripción
Rut	entero	Rol Único Tributario del cliente
Nombre	alfanumérico	Nombre o Razón Social
Dirección	alfanumérico	Dirección comercial del cliente
Giro	alfanumérico	Actividad económica desarrollada por el cliente
Teléfono	alfanumérico	Numero teléfono de contacto
Mail	alfanumérico	Correo electrónico de contacto con el cliente

Tabla 5. Detalle de atributos clase Cliente

Nombre Atributo	Tipo	Descripción
Rut	entero	Rol Único Nacional del consultor
Nombre	alfanumérico	Nombre del consultor
Fecha Certificación	fecha	Fecha del último certificado obtenido por el consultor, emitido por el INN
Profesión	alfanumérico	Título del consultor

Tabla 6. Detalle de atributos clase Consultor

Nombre Atributo	Tipo	Descripción
Fecha	fecha	Fecha de inicio de la auditoria
Banco de Condensadores	entero	Potencia del banco de condensadores, si este no existe el valor es 0
GHP	booleano	Indica si existe generación propia en las horas tipificadas como de punta
Malla BT	entero	Capacidad de la malla de protección para los consumos en baja tensión, si esta no existe el valor es 0
Malla MT	entero	Capacidad de la malla de protección para los consumos en media tensión, si esta no existe el valor es 0
Mediciones	Medición	Conjunto de mediciones pertenecientes a una auditoria
Potencia	entero	Potencia contratada al suministro público de energía
Suministro	alfanumérico	Indica si la energía es provista por un proveedor publico o generada en forma particular
Tarifa	alfanumérico	Indica la tarifa contratada al proveedor de energía público.

Tabla 7. Detalle de atributos clase Auditoria

Nombre Atributo	Tipo	Descripción
Fecha Inicio	fecha	Fecha en la que se inicio la captura de los datos eléctricos
Fecha Fin	fecha	Fecha en que se captura el último registro
Anomalías	double	Detalle de anomalías por cada uno de los registros correspondientes a la medición
Intervalo	entero	Unidad numérica del intervalo en que son capturados los datos
Unidad	alfanumérico	Indica si el intervalo corresponde a minutos o segundos
Stat1	registro	Arreglo en donde se guardan las mediciones en donde se registran los mínimos y máximos de las variables analizadas
Stat2	double	Arreglo en donde se guarda el promedio y la desviación estándar de cada una de las variables capturadas.
Total Registros	entero	Número total de registros capturados

Tabla 8. Detalle de atributos clase Medición

Nombre Atributo	Tipo	Descripción
Fecha	fecha	Instante en el que son capturadas las variables eléctricas
Voltaje	double	Medición de la tensión en cada una de las fases
Corriente	double	Medición de la corriente en cada una de las fases
Potencia	double	Medición de la potencia en cada una de las fases
Energía Aparente	double	Medición de la energía aparente en cada una de las fases
Energía Reactiva	double	Medición de la energía reactiva en cada una de las fases
Factor de Potencia	double	Medición del factor de potencia de cada una de las fases

Tabla 9. Detalle de atributos clase Registro

8 CONSTRUCCIÓN

8.1 Definición de Arquitectura

Se utilizo como base para el diseño de la arquitectura de la aplicación lo planteado en [Lhotka2005] y se ha definido una arquitectura compuesta por tres capas, las cuales se detallan en el siguiente diagrama:

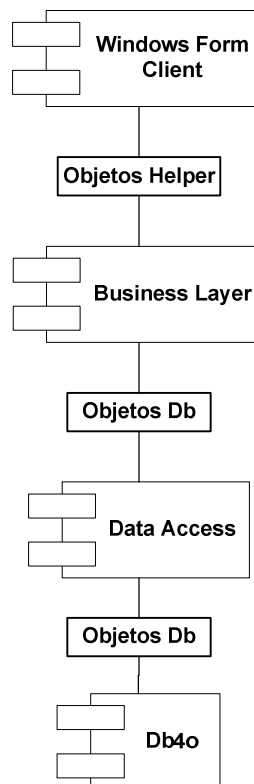


Figura 20. Capas de Arquitectura de la Aplicación.

Cabe destacar que entre las capas solo existe traspaso de objetos, esto gracias a la implementación de todas las capas con tecnologías orientadas a Objetos.

8.2 Análisis estadístico de la medición y pre condiciones de búsqueda de anomalías.

Se utilizó como base del desarrollo el procedimiento seguido por el consultor, el cual mediante el análisis de la información estadística básica, determinaba primeramente las posibles anomalías, tras lo cual, procedía a revisar detalladamente los 6.000 registros capturados, considerando las 24 variables para indicar la presencia de una anomalía puntual, en un momento determinado por la fecha y hora del registro analizado.

En la aplicación se implementó un procedimiento que, posterior a la importación de los datos, se procesan éstos y se determinan los valores: máximos, mínimos, media aritmética y desviación estándar por cada una de las variables analizadas. Se utilizó este método como una forma de descartar información que no tenga relevancia en el análisis posterior asistido por las redes neuronales, ya que como se indica en [Isasi2004], se debe identificar claramente la información mediante la ejecución de un análisis previo, las variables que aportarán información a la red y cuáles no.

8.3 Redes Neuronales utilizadas en el análisis

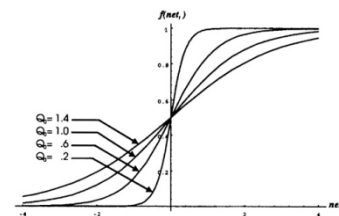
Para la implementación de las redes neuronal se selecciono como estructura base, una red de perceptrones con multicapas, basada en la teoría expuesta en [Isasi2004], con la implementación del algoritmo de retroalimentación hacia atrás o Back Propagation, en el cual el error registrado en la capa de salida es propagado hacia atrás para la corrección de los pesos correspondientes a las neuronas de cada una de las capas ocultas.

Como función de activación se utilizaron el sigmoide y el sigmoide bipolar, cuya diferencia está en el recorrido de cada una. Ambas cumplen con la condición de ser continuamente diferenciable [Sierra2006].

Para cada caso se implementaron pruebas con cada una de ellas.

Función Sigmoidal

$$f_1(x) = \frac{1}{1 + e^{-x}}$$



Función Sigmoidal Bipolar

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

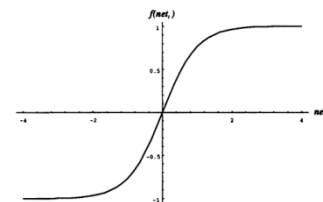


Figura 21. Funciones de activación del perceptron multicapa.

Ambas son funciones crecientes con dos niveles de saturación: el máximo, que proporciona la salida 1, y el mínimo, salida 0 para la función sigmoideal y salida -1 para el sigmoide bipolar.

8.3.1 Red Neuronal 1. Sugerencias de análisis.

Se definió una red neuronal con la capacidad de asistir al consultor en la decisión de los análisis a ejecutar sobre la información capturada.

Los análisis son realizados de acuerdo a los resultados aportados por el análisis estadístico y que busca medir solamente las condiciones que registren mayores anomalías, a fin de no perjudicar el correcto reconocimiento de los patrones de fallas.

Se creó un set de patrones de entrenamiento de un total de 1.000 pares de arreglos de entrada y salida. Los parámetros de entrada están constituidos por un arreglo de 30 variables, que cumplen con las condiciones observadas sobre el análisis estadístico realizado a la medición y el arreglo de salida determina los análisis detallados que se deben ejecutar.

In	Medida
0	Dsv sobre 5% AV
1	Dsv sobre 5% BV
2	Dsv sobre 5% CV
3	Si Max fuera de rango AV
4	Si Max fuera de rango BV
5	Si Max fuera de rango CV
6	Si Min fuera de rango AV
7	Si Min fuera de rango BV
8	Si Min fuera de rango CV
9	Desbalance Volt> 3% A/B
10	Desbalance Volt> 3% A/C
11	Desbalance Volt> 3% B/C
12	Desbalance Volt> 3% B/C
13	Desbalance I> 30% A/C
14	Desbalance I> 30% B/C
15	Si Max/(Dsv+AVG) >40% AA
16	Si Max/(Dsv+AVG) >40% BA
17	Si Max/(Dsv+AVG) >40% CA
18	Si Avg > 70% Pot Cont TW
19	Si Max> Pot Cont TW
20	Si AVAR<0
21	Si BVAR<0
22	Si CVAR<0
23	Si AVG <0,93 APF
24	Si AVG <0,93 BPF
25	Si AVG <0,93 CPF
26	Si AVG-DSV<0,93 APF
27	Si AVG-DSV<0,93 BPF
28	Si AVG-DSV<0,93 CPF
29	SI NA>10% AA,BA,CA

Out	Analisis
0	Desbalance Tension
1	Desbalance Corriente
2	Secuencia de Fases
3	Sobre Carga
4	Normativo
5	Corriente Reactiva
6	Corriente Neutro
7	Potencia Contratada

Tabla 10. Arreglos de entrada y salida de red neuronal 1.

Para determinar la optima configuración de la red neuronal, así como el numero de neuronas en la capa oculta, se utilizaron 5 configuraciones distintas, utilizando el método de ensayo- error definido en [Madan2005] .

Para todas las configuraciones se utilizaron los siguientes parámetros en común:

Learning rate: 0.1
Momentum: 0.1
Iteraciones: 1.000
Set de entrenamiento: 1.000

Se precedió a entrenar cada una de las configuraciones, obteniendo el error del proceso de entrenamiento expuesto a continuación:

Configuración	1	2	3	4	5
Neuronas en la Capa Oculta	16	60	19	10	15
Función de Activación	Sigmoide	Sigmoide	Sigmoide	Sigmoide Bipolar	Sigmoide Bipolar

Tabla 11 Configuraciones de Red Neuronal 1.

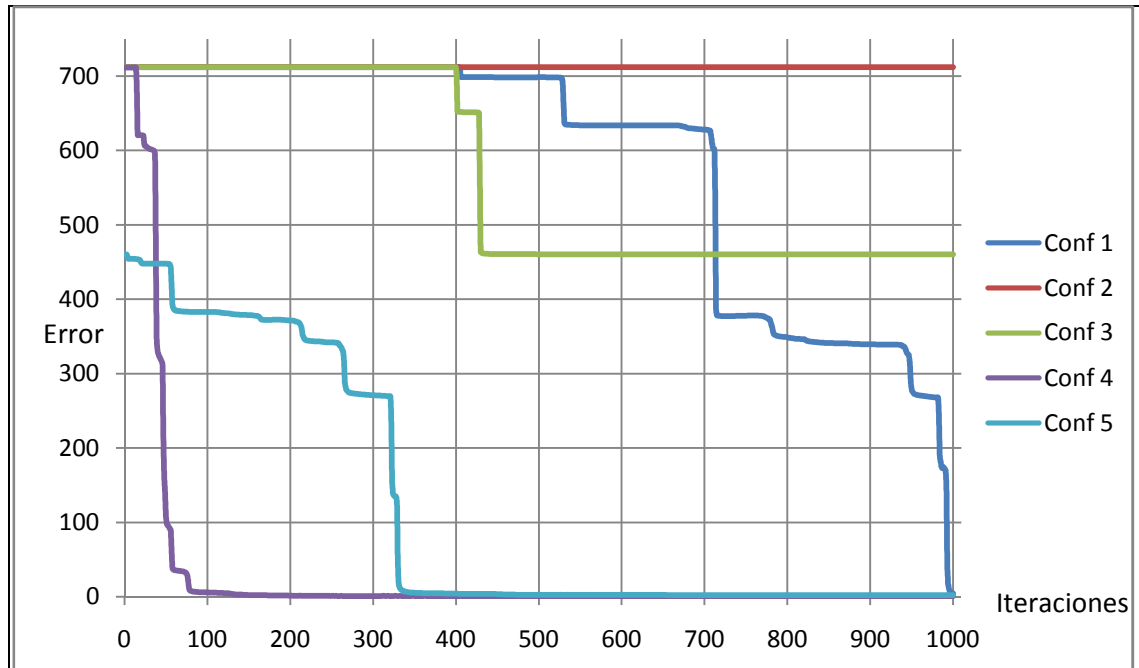


Figura 22. Error en entrenamiento de distintas configuraciones de la red neuronal 1.

Se realizaron tres procesos independientes de entrenamiento con cada una de las configuraciones para asegurar el error obtenido posterior a la ejecución de cada uno de los procesos.

El error final registrado para cada una de las configuraciones fue el siguiente:

Conf 1	Conf 2	Conf 3	Conf 4	Conf 5
4,765449	712,00	460,3868	0,558455	2,111243

Tabla 12. Error en la iteración número 1.000 del proceso de entrenamiento de la red neuronal 1.

Por lo demostrado en el entrenamiento de cada una de las configuraciones, se ha seleccionado la configuración número 4, como la más adecuada para las características del problema.

8.3.2 Red Neuronal 2. Detección y clasificación de anomalías.

Se definió una red neuronal que contará con la capacidad de identificar y clasificar anomalías presentes en las instalaciones en donde fueron capturados los registros analizados.

La necesidad de esta implementación, y objetivo principal de este seminario, está dada por la complejidad del análisis debido a la combinación de componentes en cada uno de los diversos tipos de fallos así como la nula estandarización en los procedimientos de identificación de los patrones de anomalías.

Para el requerimiento anteriormente presentado, se ha determinado la utilización de una red neuronal de perceptrones con multicapas, de acuerdo a la descripción inicialmente realizada.

Se creó un set de patrones de entrenamiento de un total de 175 pares de arreglos de entrada y salida. Los parámetros de entrada están constituidos por un arreglo de 10 variables, que representan las anomalías detectadas por los análisis previamente ejecutados y el diagnostico constituye un arreglo de 10 variables donde se indica cual sería el origen de la anomalía presentada en la entrada de la red.

	Anomalia		Diagnostico
0	Desbalance de Tensión	0	Banco de Condensadores
1	Desbalance de Corriente	1	Malla Baja Tension
2	Secuencia de Fase	2	Malla Media Tensión
3	Normativo Tensión	3	Gestión de Consumos
4	Normativo Factor de Potencia	4	Calidad de Suministro
5	Corriente de Neutro	5	Distribución de Cargas
6	SobreCarga	6	Secuencia de Fases
7	Sobre Potencia Contratada	7	Calidad de Neutro
8	Energía Reactiva	8	Problemas de Generación
9	Generación en Horario Punta	9	Potencia Contratada

Tabla 13. Arreglos de Entrada y salida Red Neuronal 2.

Para la búsqueda de la mejor configuración de la red neuronal de acuerdo a las características del problema, se probaron 5 configuraciones distintas, variando

la cantidad de neuronas en la capa oculta, así como la función de activación utilizada.

Parámetros comunes para todas las configuraciones:

Learning rate: 0.1

Momentum: 0.1

Iteraciones: 1.000

Set de entrenamiento: 175

Configuración	1	2	3	4	5
Neuronas en la Capa Oculta	10	60	19	10	15
Función de Activación	Sigmoide Bipolar	Sigmoide Bipolar	Sigmoide	Sigmoide Bipolar	Sigmoide

Tabla 14. Configuraciones Red Neuronal 2.

El error obtenido en el proceso de entrenamiento de cada una de las configuraciones se detalla en el siguiente gráfico:

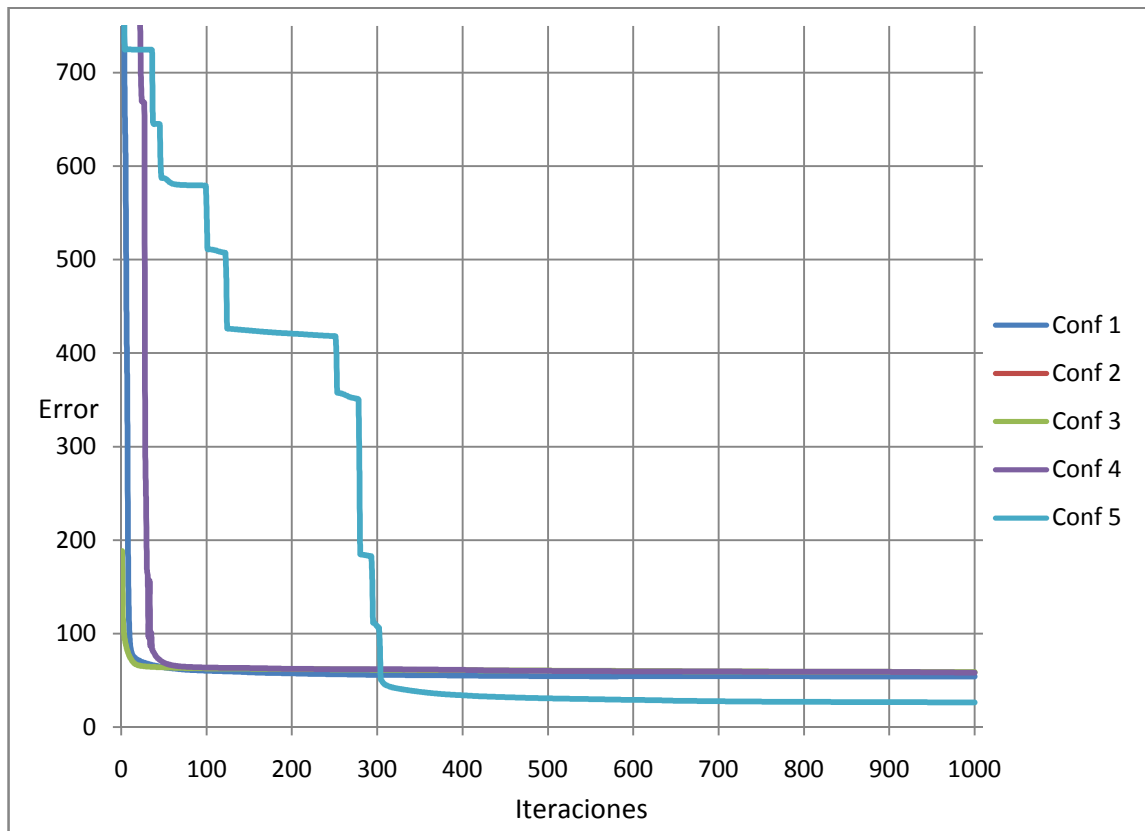


Tabla 15. Error en entrenamiento de distintas configuraciones de la red neuronal 2.

Se realizaron tres procesos independientes de entrenamiento con cada una de las configuraciones para asegurar el error obtenido posterior a la ejecución de cada uno de los procesos.

El error final registrado para cada una de las configuraciones fue el siguiente:

Conf 1	Conf 2	Conf 3	Conf 4	Conf 5
5,422242	81,04997	5,894396	5,817642	2,642332

Tabla 16. Error en la iteración número 1.000 del proceso de entrenamiento de la red neuronal 2.

Según el análisis expuesto, la configuración de red neuronal que más se adecúa a las condiciones del problema es la número 5.

8.3.3 Sub Redes especializadas

Se configuraron adicionalmente dos redes para los diagnósticos asociados con el Banco de Condensadores y Gestión de Consumos.

Estas redes cuentan con la capacidad de realizar 4 sub diagnósticos para cada una de las aéreas señaladas haciendo un segundo análisis de las entradas de la Red Neuronal 2.

Se adoptó esta modalidad de implementación debido a los diversos factores asociados a cada uno de los diagnósticos que, de acuerdo a las pre condiciones dadas, implican resultados diversos.

Se implementaron ambas sub redes con funciones de activación sigmoidales y los patrones de entrenamiento consistieron en el sub conjunto de patrones extraídos de la muestra principal utilizada para el entrenamiento de la Red Neuronal 2.

9 **INTERFAZ DE USUARIO**

Se diseñó la interfaz de usuario de acuerdo a los requerimientos presentados por los consultores, poniendo especial énfasis en la disposición de la información y la validación de la información ingresada.

Como guía del desarrollo de la interfaz, se utilizó como base las definiciones de [Ambler2005], de las cuales a continuación se exponen las principales:

La Estructura: Las interfaces se deben organizar de acuerdo al propósito de las mismas utilizando modelos claros y consistentes así como familiares para el usuario.

La Simplicidad: El diseño debe ser simple, tareas comunes de simple ejecución, manteniendo la comunicación con el usuario en su propio lenguaje. Se utilizaron los colores institucionales para el diseño de las interfaces.

La Retroalimentación: Se debe mantener informados a los usuarios respecto de cambios de estado o condición, sin caer en redundancia de información.

La Reusabilidad: El diseño debe reusar componente internos y externos, manteniendo la consistencia en toda la aplicación, reduciendo así la necesidad del cliente de recordar o repensar el comportamiento de la interfaz.

Durante el proceso de elaboración se realizaron tres prototipos de interfaces, los cuales fueron validados por los usuarios y a continuación se exponen algunas capturas de los formularios definitivos del sistema.

The screenshot shows a web application window titled 'Sistema de Análisis Eléctrico - [SetClienteFr]'. The navigation menu includes 'Cliente', 'Auditoria', 'Medición', 'Server', and 'Test'. The main content area is divided into two sections:

Selección del Cliente

This section contains a form with the following fields:

- Rut: 13968910
- Nombre: Cliente 1
- Dirección: Direccion Cliente 1
- Telefono: 257755
- Giro: Produccion
- Mail: contacto@cliente1.cl

Below the form are three buttons: 'Ver Lista', 'Ingresar', and 'Editar'.

Listado de Auditorias

This section displays a table with the following data:

	Fecha	Potencia	Tarifa	Mediciones
▶	27-08-2008	120	AT 4.1	0
*				

Below the table are two buttons: 'Nueva' and 'Editar'.

At the bottom of the main content area are two buttons: 'Siguiente' and 'Cancelar'.

The footer of the application shows 'Sistema de Analisis Eléctrico - Stand By', 'Cliente', and 'Auditoria'.

Figura 23. Interfaz de Usuario – Ingreso/Edición de Cliente

Sistema de Análisis Eléctrico

Cliente Auditoría Medición Server Test

Detalle de Auditoría

Fecha Creación: 27-08-2008 Tipo Suministro: Público

Tarifa Contratada: AT 4.1 Potencia: 150 kW

Banco de Condensadores: ☒ Potencia: 200 kVA ☒ Generación Hora Punta

Malla Baja Tensión: ☒ Potencia: 100 Malla Media Tension: ☒ Potencia: 150

Mediciones

	F Inicio	F Fin	Registros	Intervalo	Unidad
▶	15-02-2008	19-02-2008	1000	1	Segundos
	20-02-2008	22-02-2008	2320	1	Segundos
*					

Importar

Guardar Cancelar

Sistema de Analisis Eléctrico - Stand By Cliente Auditoría

Figura 24. Interfaz de Usuario –Creación de Auditoría

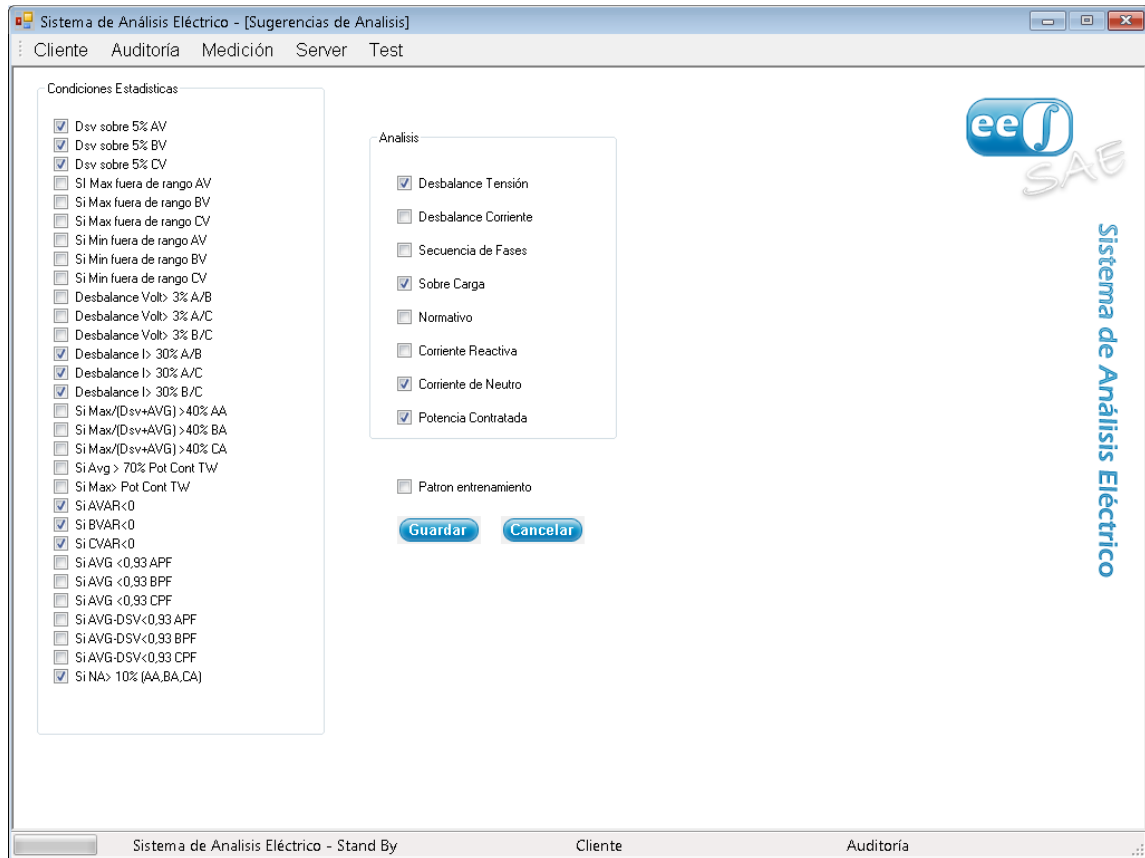


Figura 25. Interfaz de Usuario – Ingreso/validación patrones de análisis.

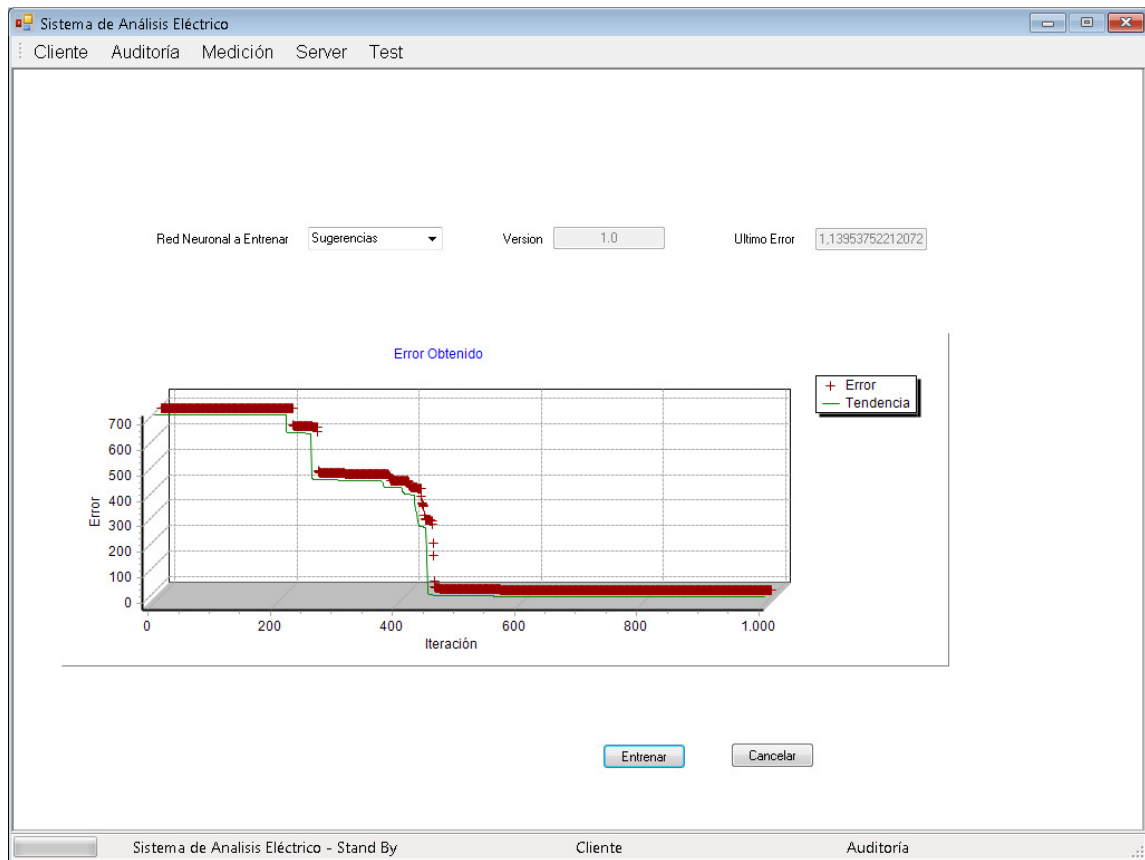


Figura 26 Interfaz de Usuario – Proceso entrenamiento Redes Neuronales

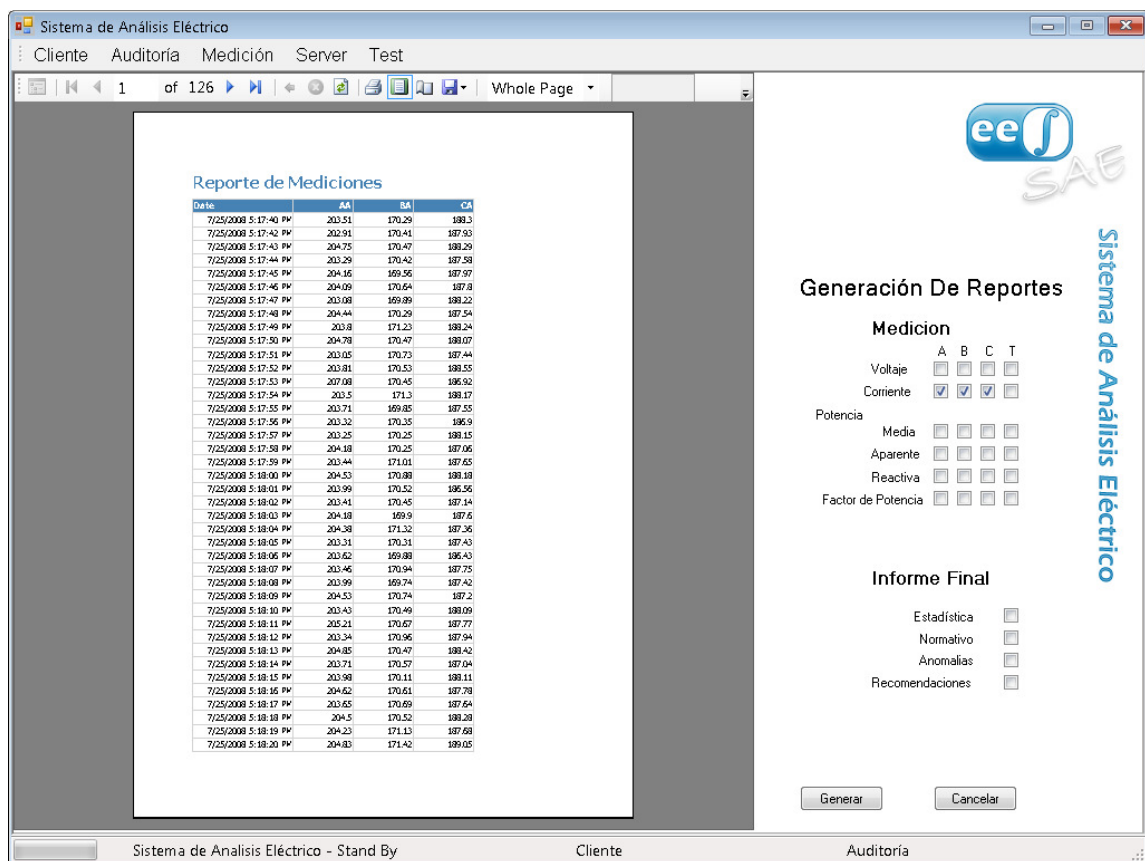


Figura 27. Interfaz de Usuario – Generación/Visualización de Reportes

10 PRUEBAS

Según la definición inicial del Plan de Pruebas, se desarrollaron las siguientes actividades en cada una de las respectivas etapas del desarrollo:

Inicio:

Según la definición de requerimientos realizada en el modelo del dominio, se verificó el cumplimiento de las condiciones técnicas predefinidas y se validaron los requerimientos como los básicos de la primera iteración según lo definido en la metodología.

Elaboración:

Se implementó un prototipo de la función de importación de los datos desde la hoja Excel para validar las condiciones técnicas de Db4o como motor de Base de Objetos de la Aplicación.

Se comprobó la persistencia de los objetos al realizar inserciones y recuperaciones de las estructuras de las redes neuronales definidas en la aplicación, las cuales se desarrollaron en completa conformidad.

Construcción:

En esta etapa se utilizó la IDE de Visual Studio 2008 .Net que incorpora los elementos necesarios para la ejecución y seguimiento de las pruebas de unidad.

A continuación se expone el resultado de la ejecución del test de unidad de la clase MediciónHelper, correspondiente a la capa de negocios de la aplicación.

Debido a la dependencia de datos, se creó un set de datos para cada una de las clases testeadas, el cual fue manejado de manera independiente a los utilizados en las pruebas posteriores correspondientes a la fase de transición.

Test Results			
BLtests Run Debug			
Test run warning Results: 34/34 passed; Item(s) checked: 0			
Result	Test Name	Project	Error Message
Passed	InicioTest	SAETest	
Passed	MedicionHelperConstructorTest	SAETest	
Passed	GuardaAnomaliasTest	SAETest	
Passed	SobrePotConTest	SAETest	
Passed	EstadisticoTest	SAETest	
Passed	GetAllTest	SAETest	
Passed	FinTest	SAETest	
Passed	FinTest	SAETest	
Passed	Stat1Test	SAETest	
Passed	DesbalanceTensionTest	SAETest	
Passed	MedicionDbConstructorTest	SAETest	
Passed	NumRegTest	SAETest	
Passed	SecuenciaFasesTest	SAETest	
Passed	NormativoTest	SAETest	
Passed	TotalTest	SAETest	
Passed	EnergiaReacTest	SAETest	
Passed	InicioTest	SAETest	
Passed	MedicionHelperConstructorTest	SAETest	
Passed	UnidadTest	SAETest	
Passed	UnidadTest	SAETest	
Passed	GuardaAnomaliasTest	SAETest	
Passed	AnomaliasTest	SAETest	
Passed	AnEjectTest	SAETest	
Passed	IntervaloTest	SAETest	
Passed	IntervaloTest	SAETest	
Passed	NumRegFuncTest	SAETest	
Passed	SobreCargaTest	SAETest	
Passed	DesbalanceCargasTest	SAETest	
Passed	fechalTest	SAETest	
Passed	TotalTest	SAETest	
Passed	MedicionesTest	SAETest	
Passed	SetAnomaliasTest	SAETest	
Passed	Stat2Test	SAETest	
Passed	CorrientesNeutroTest	SAETest	

Figura 28. Reporte de Visual Studio para pruebas de unidad

Transición:

Se validó en conjunto con los usuarios, el cumplimiento de los requisitos, así como también se ejecutaron test de integración, en los cuales se evidenció el correcto funcionamiento del prototipo.

Para las pruebas del funcionamiento de las redes neuronales, se utilizó un set de pruebas distinto al utilizado en el acondicionamiento inicial de las mismas para evitar el sobre entrenamiento de la red y descartar la existencia de mínimos locales. Estas pruebas se realizaron siguiendo los parámetros definidos en [Kröse1996].

11 CONCLUSIONES Y RECOMENDACIONES

Dados los lineamientos iniciales del problema presentado por la empresa Eficiencia Energética Integral, el alumno consideró enfrentar el desarrollo por dos áreas principales: El estudio del peso específico de la información a procesar y su relación con las anomalías eléctricas posibles de identificar por el reconocimiento de patrones, así como las diversas condiciones eléctricas presentes en los distintos ambientes industriales.

Este análisis previo, junto a las características aportadas por la metodología respecto a la inclusión del cliente en las tareas propias del desarrollo del software, permitieron al alumno lograr un cabal cumplimiento de los objetivos planteados.

Las redes neuronales aportan un gran potencial a las características dinámicas de herramienta final, ya que provee una solución cercana a las condiciones de las actividades desarrolladas por EEI, que por ser una empresa pionera en la región y pertenecer al selecto grupo de empresas certificadas por el Instituto Nacional de Normalización, aporta día a día al estudio de la Eficiencia Energética en la Industria Nacional.

Este proyecto también ha permitido a EEI, optimizar el uso de sus recursos, acortando el tiempo de desarrollo de sus auditorías y permitiéndole dar una mejor atención a sus clientes por medio del desarrollo in-situ de los informes

finales de sus auditorías, condición que previo a la implementación del Software de Análisis Eléctrico, era realizada en los días posteriores a la captura de la información.

Como recomendación general al proyecto, el alumno ha identificado las siguientes:

Dadas la características del prototipo desarrollado, queda la posibilidad de implementar la interfaz de usuario en formato Web, y con el desarrollo de algunas aplicaciones de gestión energética, que posibilitaría al cliente auto gestionar energéticamente su empresa, siguiendo los lineamientos entregados por EEI.

Identificando la tecnología disponible en el mercado respecto a equipos de monitoreo eléctrico, cabe señalar como nueva área de desarrollo, la implementación de un sistema de detección de anomalías eléctricas On-line, el cual permitiría anticipar de acuerdo a los patrones previamente definidos posibles condiciones críticas y anticipar y programar actividades de mantención en equipamiento de vital importancia en los diversos procesos productivos.

12 **BIBLIOGRAFÍA**

- [Isasi2004] Isasi Viñuela, P; Galván León, I.M. Redes de Neuronas Artificiales, Un enfoque Practico. Pearson. Primera Edición. 2004.
- [Deitel2006] Deitel, H.M; Deitel, P.J. Visual C# 2005, How to Program. Pearson. Second Edition. 2006
- [Larman2003] Larman, C. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson. Segunda Edición.2003.
- [Nilsson2005] Nilsson,J.W; Riedel,S.A. Circuitos Eléctricos. Pearson. Séptima Edición. 2005.
- [Ambler2003] Ambler, Scott W. Agile DatabaseTechniques: Effective Strategies for the Agile Software Developer. Wiley Publishing, Inc. 2003
- [Ambler2005] Ambler, Scott W. The Agile Unified Process (AUP).

Disponible en
<http://www.ambysoft.com/unifiedprocess/agileUP.html>,

Septiembre 2005.

- [Madan2003] Madan, Gupta. Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. Wiley Inc. 2003
- [Kirillov2008] Kirillov, Andrew .AForge Project.
- Disponible en
- <http://code.google.com/p/aforge/>
- Agosto 2008.
- [Kröse1996] Kröse, Ben. An Introduction to Neural Networks. University of Amsterdam. Eighth Edition. 1996.

13 ANEXOS

```
public class MedicionHelperTest
{
    MedicionHelper Medicion = new MedicionHelper();
    private TestContext testContextInstance;
    public TestContext TestContext
    {
        get
        {
            return testContextInstance;
        }
        set
        {
            testContextInstance = value;
        }
    }

    #region Additional test attributes

    [TestInitialize()]
    public void MyTestInitialize()
    {
        Medicion.Load(@"C:\SW\SAETest\Test Data\DATOS Test.xls");
    }

    [TestCleanup()]
    public void MyTestCleanup()
    {
        Medicion = new MedicionHelper();
    }
    #endregion

    ///A test for Unidad
    [TestMethod()]
    public void UnidadTest()
    {
        MedicionHelper target = Medicion;
        string actual;
        actual = target.Unidad;
        Assert.AreEqual(target.Unidad, actual);
    }

    ///A test for Total
    [TestMethod()]
    public void TotalTest()
    {
        MedicionHelper target = Medicion;
        int actual;
```

```

        actual = target.Total;
        Assert.AreEqual(target.Total, actual);
    }

    ///A test for Mediciones
    [TestMethod()]
    public void MedicionesTest()
    {
        MedicionHelper target = Medicion;
        List<RegistroDb> actual;
        actual = target.Mediciones;
        Assert.AreEqual(actual, target.Mediciones);
    }

    ///A test for Intervalo
    [TestMethod()]
    public void IntervaloTest()
    {
        MedicionHelper target = Medicion;
        int actual;
        actual = target.Intervalo;
        Assert.AreEqual(actual, target.Intervalo);
    }

    ///A test for Inicio
    [TestMethod()]
    public void InicioTest()
    {
        MedicionHelper target = Medicion;
        DateTime actual;
        actual = target.Inicio;
        Assert.AreEqual(target.Inicio, actual);
    }

    ///A test for Fin
    [TestMethod()]
    public void FinTest()
    {
        MedicionHelper target = Medicion;
        DateTime actual;
        actual = target.Fin;
        Assert.AreEqual(actual, target.Fin);
    }

    ///A test for SobrePotCon
    [TestMethod()]
    public void SobrePotConTest()
    {
        MedicionHelper target = Medicion;
        MedicionHelper Actual;
        int PotCont = 0;
        Actual = Medicion;
        Actual.SobrePotCon(PotCont);
        target.SobrePotCon(PotCont);
        Assert.AreEqual(Actual, target);
    }

```

```

}

///A test for SobreCarga
[TestMethod()]
public void SobreCargaTest()
{
    MedicionHelper target = Medicion;
    Medicion.SobreCarga();
    target.SobreCarga();
    Assert.AreEqual(target, Medicion);
}

///A test for SecuenciaFases
[TestMethod()]
public void SecuenciaFasesTest()
{
    MedicionHelper target = Medicion;
    target.SecuenciaFases();
    Medicion.SecuenciaFases();
    Assert.AreEqual(target, Medicion);
}

///A test for Normativo
[TestMethod()]
public void NormativoTest()
{
    MedicionHelper target = Medicion;
    target.Normativo();
    Medicion.Normativo();
    Assert.AreEqual(target, Medicion);
}

///A test for GuardaAnomalias
[TestMethod()]
public void GuardaAnomaliasTest()
{
    MedicionHelper target = Medicion;
    target.GuardaAnomalias();
    Medicion.GuardaAnomalias();
    Assert.AreEqual(target, Medicion);
}

///A test for Estadistico
[TestMethod()]
public void EstadisticoTest()
{
    MedicionHelper target = Medicion;
    target.Estadistico();
    Medicion.Estadistico();
    Assert.AreEqual(target, Medicion);
}

///A test for EnergiaReac
[TestMethod()]
public void EnergiaReacTest()

```

```

{
    MedicionHelper target = Medicion;
    target.EnergiaReac();
    Medicion.EnergiaReac();
    Assert.AreEqual(target, Medicion);
}
///A test for DesbalanceTension
[TestMethod()]
public void DesbalanceTensionTest()
{
    MedicionHelper target = Medicion;
    double dt = 0.03;
    target.DesbalanceTension(dt);
    Medicion.DesbalanceTension(dt);
    Assert.AreEqual(target, Medicion);
}
///A test for DesbalanceCargas
[TestMethod()]
public void DesbalanceCargasTest()
{
    MedicionHelper target = Medicion;
    double dc = 0.3;
    target.DesbalanceCargas(dc);
    Medicion.DesbalanceCargas(dc);
    Assert.AreEqual(target, Medicion);
}
///A test for CorrientesNeutro
[TestMethod()]
public void CorrientesNeutroTest()
{
    MedicionHelper target = Medicion;
    target.CorrientesNeutro();
    Medicion.CorrientesNeutro();
    Assert.AreEqual(target, Medicion);
}
[TestMethod()]
public void MedicionHelperConstructorTest1()
{
    MedicionHelper target = new MedicionHelper();
    Medicion = new MedicionHelper();
    Assert.AreEqual(target, Medicion);
}
}
}

```

Figura 29. Implementación de Test Clase MedicionHelper

```

public void SobreCarga()
{
    double l1 = 0, l2 = 0, l3 = 0, l4 = 0, l5 = 0, al5 = 0, a7=0;
    int IndAnom = 0;
    double des1 = 0, des2 = 0, des3 = 0;
    double[] PASS= new double[7];
    bool flag = false;
    List<RegistroDb> col = MedicionDb.GetAll();
    Medicion.SetAnomalias();

    foreach (RegistroDb i in col)
    {

        al5 = (l1 + l2 + l3 + l4 + l5) / 5;
        if ((al5 + (Medicion.Stat2[1][10] / 2)) < i.TW & al5>0)
        {
            a7 = 1-(al5/i.TW);
            PASS = Medicion.GetAnom(IndAnom);
            PASS[6] = a7;
            Medicion.SetAnom(IndAnom, PASS);
        }

        l1 = i.TW;
        l2 = l1;
        l3 = l2;
        l4 = l3;
        l5 = l4;
        IndAnom++;
    }
}

```

Figura 30. Implementación de análisis de sobre carga.

```

public BackPropagationLearning( ActivationNetwork network )
{
    this.network = network;

    // Creación de error y arreglo delta
    neuronErrors = new double[network.LayersCount][];
    weightsUpdates = new double[network.LayersCount][][];
    thresholdsUpdates = new double[network.LayersCount][];

    // Se inicializa el error y los arreglos delta en cada capa
    for ( int i = 0, n = network.LayersCount; i < n; i++ )
    {
        Layer layer = network[i];

        neuronErrors[i] = new double[layer.NeuronsCount];
        weightsUpdates[i] = new double[layer.NeuronsCount][];
        thresholdsUpdates[i] = new double[layer.NeuronsCount];

        // para cada neurona
        for ( int j = 0; j < layer.NeuronsCount; j++ )
        {
            weightsUpdates[i][j] = new double[layer.InputsCount];
        }
    }
}

```

Figura 31. Implementación método de aprendizaje Back Propagation

```

public override double Compute( double[] input )
{
    // Verificación de tamaño de input
    if ( input.Length != inputsCount )
        throw new ArgumentException( );

    // Valor inicial de la suma
    double sum = 0.0;

    // Calculo de la suma de pesos por entradas
    for ( int i = 0; i < inputsCount; i++ )
    {
        sum += weights[i] * input[i];
    }
    sum += threshold;

    return ( output = function.Function( sum ) );
}

```

Figura 32. Implementación cálculo de salida de cada neurona


```

public abstract class Network
{

    protected string name;
    protected double version;

        protected int    inputsCount;
        protected int    layersCount;
        protected Layer[] layers;
        protected double[] output;

    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    public double Version
    {
        get { return version; }
        set { version = value; }
    }

        public int InputsCount
        {
            get { return inputsCount; }
        }

        public int LayersCount
        {
            get { return layersCount; }
        }

        public double[] Output
        {
            get { return output; }
        }

        public Layer this[int index]
        {
            get { return layers[index]; }
        }

        protected Network( int inputsCount, int layersCount )
        {
            this.inputsCount = Math.Max( 1, inputsCount );
            this.layersCount = Math.Max( 1, layersCount );
            // creación de colección de capas
            layers = new Layer[this.layersCount];
        }
    }

```

```

public virtual double[] Compute( double[] input )
{
    output = input;

    // Calculo por cada capa
    foreach ( Layer layer in layers )
    {
        output = layer.Compute( output );
    }

    return output;
}

public virtual void Randomize( )
{
    foreach ( Layer layer in layers )
    {
        layer.Randomize();
    }
}
}

```

Figura 33. Implementación clase Network