

**UNIVERSIDAD CATÓLICA SANTA MARÍA FACULTAD DE
CIENCIAS FISICAS Y FORMALES PROGRAMA
PROFESIONAL DE INGENIERIA DE SISTEMAS**



**DETECCION DE ARMONICOS EN CORRIENTE
ELECTRICA BASADA EN REDES NEURONALES**

Tesis presentado por los bachilleres:

FONSECA NUÑEZ CESAR ALEJANDRO

ZAMUDIO LA ROSA JEAN CARLO

Para optar el Título profesional de:

INGENIERO DE SISTEMAS

Arequipa- Perú
2014

Presentación

Sra. Directora del Programa Profesional de Ingeniería de Sistemas

Sres. Miembros del Jurado Examinador de Tesis

De conformidad con las disposiciones del reglamento de Grados y Títulos del Programa Profesional de Ingeniería de Sistemas, remitimos a vuestra consideración el estudio de investigación titulado “Detección de Armónicos en Corriente Eléctrica Basado en Redes Neuronales”, el mismo que al ser aprobado nos permitirá optar el título profesional de Ingeniería de Sistemas.

Arequipa, 01 de Setiembre del 2014

César Alejandro Fonseca Núñez

Jean Carlo Zamudio La Rosa

Agradecimientos

Agradezco a mi familia, a mis amigos y en especial a mi madre, por todos sus consejos los cuales me han ayudado en toda mi vida, a Nadia Gamero por apoyarnos en todo momento, a Lily Leyva por siempre darnos ánimos hasta el final y a Pamela Núñez, por recordarme siempre terminar lo que empecé.

“La vida no se mide en años, sino en la cantidad de experiencias vividas.”

César Alejandro Fonseca Núñez

A mi abuela Paula, que en donde esté, también estará muy feliz por mí.

A mi madre, por estar siempre conmigo en las buenas y malas hasta el día de hoy y siempre.

Alejandro, un amigo incondicional y que me hizo la vida imposible desde que lo conocí, gracias por tu apoyo hermano.

A todos los amigos y compañeros que de una o de otra forma me apoyaron con este proyecto.

"No confundas el tener menos con ser menos, tener más no significa ser más; tampoco confundas lo que posees con quien eres" - Noah benShea

Jean Carlo Zamudio La Rosa

RESUMEN

Las industrias intentan acortar gastos de producción para sus productos con la intención de generar más ganancias, pero uno de los mayores gastos está dentro del pago de energía eléctrica y mantenimiento de los equipos que usan, adicionalmente puede ocurrir una mala instalación de algún equipo o mala entrega de energía por parte de la empresa proveedora de la misma, este tipo de problemas acarrea la falla en los equipos conectados, el deterioro de los mismos, reducción del tiempo de vida o un consumo adicional de energía eléctrica a los requeridos por la empresa.

Esta tesis fue desarrollada en colaboración con la empresa I + T S.C.R.L. y como parte del proyecto Swap Smart, del cual se enfocó a la detección de armónicos de corriente eléctrica usando Redes Neuronales Artificiales, y financiado por el Fondo para la Innovación, Ciencia y Tecnología (FINCyT), contando con la colaboración del Centro Interdisciplinario de Investigación e Innovación de la Universidad Católica de Santa María.

La tesis muestra los resultados de las investigaciones realizadas para determinar los armónicos en la corriente eléctrica utilizando Redes Neuronales Artificiales (RNA) aplicando los métodos de Feedforward - Backpropagation por medio de un generador de señales eléctricas desarrollado en C# (C Sharp). Para esto se estudiaron las causas que producen los armónicos de corriente, cuáles son sus implicancias en el trabajo diario y los filtros para atenuar dichos armónicos. Para la generación de armónicos, se implementó un emisor de señales eléctricas mediante software, desarrollando también en C# (C Sharp) de forma que se obtenga data sin procesar y lo más real posible, para así realizar los análisis correspondientes, simulando errores en la señal eléctrica que se dan en la realidad, para luego detectarlos y procesar esta data.

Así mismo, luego de las pruebas realizadas, se determinó que el mejor método para la detección de armónicos aplicando Redes Neuronales Artificiales es Feedforward –

Backpropagation con un entrenamiento supervisado y así poder manejar los datos de entrada y salida para obtener un mejor resultado.

Esta investigación se sustenta en el mejor método para determinar dichos armónicos incluyendo la velocidad de procesamiento y entrenar esta red para determinar eficientemente los armónicos en la corriente. Al usar el modelo de red Feedforward estamos usando un modelo multicapa, al tener dos o más capas mejora la memorización y la interpolación de puntos.



ABSTRACT

Industries try to shorten production costs for its products with intent to generate more profits, but one of the biggest expenses is within the payment of electricity and maintenance of the equipment they use, further improper installation can occur in any computer or bad energy delivery by the supplier thereof, such problems entails the connected equipment failure, deterioration thereof, reduced lifetime or additional power consumption to the energy required by the company.

This thesis was developed in collaboration with the company I + T SCRL and as part of their project Swap Smart, which he focused on the detection of harmonic electric current using Artificial Neural Networks, and funded by the Fund for Innovation, Science and Technology (FINCyT), with the collaboration of the Interdisciplinary Center research and Innovation at the Catholic University of Santa Maria.

This thesis shows the results of investigations to determine the harmonics in the electrical current through the use of Artificial Neural Networks (ANN) using the methods of Feedforward - Backpropagation through a generator of electrical signals in C# (C Sharp). For this we studied the causes of the current harmonics, what are the implications in the daily work and filters to attenuate these harmonics. For the generation of harmonics, implemented a transmitter of electrical signals by software, also developed in C# (C Sharp) so as to obtain data raw and real as possible, in order to perform tests for simulating errors in the signal power that occur in reality and then process this data.

Also it was determined that the best method for the detection of harmonics using Artificial Neural Networks is Feedforward – Backpropagation with supervised training in order to handle the input and output to get a better result.

This research is based on the best method for determining these harmonics including the processing speed and be able to train the network to efficiently determine the current

harmonics. Using Feedforward network model we are using a multilayer model, having two or more layers improves memory and interpolation of points.



INTRODUCCIÓN

En el Capítulo I veremos porque se decide hacer esta investigación, los problemas que las empresas poseen con respecto a este tema, las investigaciones y discusiones de otros autores con respecto al mismo problema a tratar.

En el capítulo II se detallará los modelos de Red Neuronal Artificial, los alcances y limitaciones de las mismas, los Armónicos de corriente eléctrica, las fórmulas que se usaron para poder simular tanto las ondas de corriente como los armónicos de corriente respectivos.

En este capítulo III se implementará el modelo propuesto, el cual constará de la creación de un software para emular las señales eléctricas de corriente, a la vez poder representar en una gráfica estos datos obtenidos, este software también podrá generar y mostrar los armónicos de corriente eléctrica, estos datos obtenidos serán exportados a la red neuronal implementada en MatLab para el aprendizaje y/o entrenamiento correspondiente y mostrar los resultados de los mismos, estos últimos serán interpretados y comparados con los datos de I+T.

En el capítulo IV se mostrarán los resultados de las pruebas realizadas tanto con el simulador de señales como con la red neuronal artificial, las comparaciones entre los distintos armónicos de corriente y también el resultado del procesamiento de las señales eléctricas y las comparaciones visuales entre los distintos resultados.

ÍNDICE GENERAL

RESUMEN	III
ABSTRACT.....	V
INTRODUCCIÓN	VII
CAPÍTULO I: PLANTEAMIENTO TEÓRICO	16
1. EL PROBLEMA.....	16
1.1. <i>Título descriptivo.</i>	16
1.2. <i>Descripción del problema.</i>	16
2. OBJETIVOS.....	18
2.1. <i>Objetivo general.</i>	18
2.2. <i>Objetivos específicos.</i>	18
3. SOLUCIÓN PROPUESTA.	18
3.1. <i>Justificación del problema.</i>	18
3.2. <i>Alcances y limitaciones.</i>	19
4. ANTECEDENTES INVESTIGATIVOS.....	19
5. DISCUSIÓN SOBRE LOS AUTORES.	22
CAPÍTULO II: MARCO TEÓRICO	24
1. MARCO TEÓRICO.....	24
1.1. <i>Redes neuronales.</i>	24
1.2. <i>Modelo de neurona.</i>	26
1.3. <i>Topologías neuronales.</i>	29
2. PERCEPTRÓN MULTICAPA.....	30
2.1. <i>El combinador lineal adaptativo.</i>	30
2.2. <i>El perceptrón.</i>	31
2.3. <i>El perceptrón multicapa.</i>	35
2.4. <i>Capacidades y limitaciones del perceptrón multicapa.</i>	37
3. ¿QUÉ SON LOS ARMÓNICOS?	39
3.1. <i>Origen de los armónicos.</i>	42

3.2. <i>Contenido normal de armónicos.</i>	43
4. HERRAMIENTA MATLAB.....	43
4.1. <i>Introducción.</i>	43
4.2. <i>Características principales.</i>	44
4.3. <i>Herramientas de desarrollo.</i>	45
4.4. <i>Diseño de interfaces graficas de usuario.</i>	46
4.5. <i>Lenguaje de programación para el desarrollo del software.</i>	47
CAPÍTULO III: MODELO PROPUESTO.....	52
1. DESCRIPCIÓN DEL CAPÍTULO.....	52
2. ETAPAS DEL MODELO PROPUESTO.....	52
2.1. <i>Generador de señales.</i>	53
2.2. <i>Diagramas UML del Generador de Señales.</i>	55
2.3. <i>Diseño de Pantallas.</i>	62
2.4. <i>Entrenamiento supervisado.</i>	68
2.5. <i>Perceptrón multicapa.</i>	75
CAPÍTULO IV: RESULTADOS	89
1. GENERADOR DE SEÑALES.....	89
1.1. <i>Señal generada.</i>	89
2. RED NEURONAL.	104
A. <i>Armónico de orden 3</i>	105
B. <i>Armónico de orden 5</i>	106
C. <i>Armónico de orden 7</i>	107
D. <i>Armónico de orden 9</i>	108
E. <i>Armónico de orden 11</i>	109
CONCLUSIONES	110
RECOMENDACIONES	112
BIBLIOGRAFÍA	113
GLOSARIO DE TÉRMINOS.....	115

ÍNDICE DE DIAGRAMAS

<i>Diagrama 1: Flujograma del generador de señales. (a) Fuente: Elaboración Propia</i>	<i>55</i>
<i>Diagrama 2: Flujograma del generador de señales. (b) Fuente: Elaboración Propia</i>	<i>56</i>
<i>Diagrama 3: Casos de Uso. Fuente: Elaboración Propia</i>	<i>57</i>
<i>Diagrama 4: Diagrama de Clases. Fuente: Elaboración Propia</i>	<i>58</i>
<i>Diagrama 5: Casos de Estado. Fuente: Elaboración Propia</i>	<i>59</i>
<i>Diagrama 6: Casos de Componentes. Fuente: Elaboración Propia</i>	<i>60</i>
<i>Diagrama 7: Casos de Secuencias. Fuente: Elaboración Propia</i>	<i>61</i>



ÍNDICE DE FIGURAS

<i>Figura 1: Patrones de entrada y respuesta esperada Fuente: [HGL, 2002]</i>	<i>31</i>
<i>Figura 2: Perceptrón y no-linealidades. Fuente: [HGL, 2002]</i>	<i>32</i>
<i>Figura 3: Línea de separación en el espacio de los patrones. Fuente: [HGL, 2002]</i>	<i>33</i>
<i>Figura 4: Funciones linealmente separables. Fuente: [HGL, 2002]</i>	<i>35</i>
<i>Figura 5: Señal fundamental más señal armónica. Fuente: [LRA]</i>	<i>40</i>
<i>Figura 6: Tipos de frecuencias de armónicos. Fuente: [LRA]</i>	<i>40</i>
<i>Figura 7: Sumatoria de un armónico a la señal fundamental. Fuente: [LRA]</i>	<i>42</i>
<i>Figura 8: Imagen referencial de MatLab. Fuente: [MATHWORKS]</i>	<i>45</i>
<i>Figura 9: Un M-Lint Código Verificador de informe que incluya recomendaciones para hacer el código más rápido y más fácil de mantener. Fuente:</i>	
<i>http://www.mathworks.cn/cmsimages/ml_m-lintcodechecker_wl_7157.gif</i>	<i>46</i>
<i>Figura 10: GUÍA diseño de un análisis wavelet GUI (arriba), junto con la interfaz completo (abajo). Fuente: [MATHWORKS]</i>	<i>47</i>
<i>Figura 11: Ranking Top 20 de TIOBE – Julio 2013. Fuente:</i>	
<i>http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html</i>	<i>50</i>
<i>Figura 12: Histórico del Lenguaje C#: Rating Alto (desde el 2001): 8.763% (Posición 3 en Enero 2012). Fuente: http://www.tiobe.com/index.php/paperinfo/tpci/C_.html</i>	<i>51</i>
<i>Figura 13: Pantalla Principal del Generador de Señales. Fuente: Elaboración Propia</i>	<i>62</i>
<i>Figura 14: Proceso de creación de datos. Fuente: Elaboración Propia</i>	<i>63</i>
<i>Figura 15: Generación de armónicos terminada. Fuente: Elaboración Propia</i>	<i>63</i>
<i>Figura 16: Gráfica generada de acuerdo a los datos ingresados. Fuente: Elaboración Propia.</i>	<i>64</i>
<i>Figura 17: Modificación de onda generada. Fuente: Elaboración Propia.</i>	<i>65</i>
<i>Figura 18: Data generada para el quinto armónico. Fuente: Elaboración Propia.</i>	<i>65</i>
<i>Figura 19: Gráfica del 5to armónico. Fuente: Elaboración Propia</i>	<i>66</i>
<i>Figura 20: Gráfica del 5to armónico. Fuente: Elaboración Propia</i>	<i>66</i>
<i>Figura 21: Guardado de archivos. Fuente: Elaboración Propia.</i>	<i>67</i>

<i>Figura 22: Comprobación de los archivos generados con la data correspondiente. Fuente: Elaboración Propia.</i>	<i>68</i>
<i>Figura 23: Sección de los archivos generados con los datos y sus diferencias. Fuente: Elaboración Propia.</i>	<i>68</i>
<i>Figura 24: Esquema de una Neurona Artificial. Fuente: [GEP, 2002]</i>	<i>70</i>
<i>Figura 25: Funciones de Activación. Fuente: http://joel-redesneuronalesartificiales.blogspot.com/2008/07/redes-neuronales-artificiales.html</i>	<i>71</i>
<i>Figura 26: RNA usando la Función de Activación tansig. Fuente: Elaboración Propia</i>	<i>72</i>
<i>Figura 27: RNA usando la Función de Activación logsig. Fuente: Elaboración Propia</i>	<i>72</i>
<i>Figura 28: RNA usando la Función de Activación purelin. Fuente: Elaboración Propia ..</i>	<i>72</i>
<i>Figura 29: RNA usando la Función de Activación hardlim. Fuente: Elaboración Propia .</i>	<i>73</i>
<i>Figura 30: RNA usando la Función de Activación hardlims. Fuente: Elaboración Propia</i>	<i>73</i>
<i>Figura 31: RNA usando la Función de Activación poslim. Fuente: Elaboración Propia ...</i>	<i>73</i>
<i>Figura 32: RNA usando la Función de Activación satlim. Fuente: Elaboración Propia</i>	<i>74</i>
<i>Figura 33: RNA usando la Función de Activación satlims. Fuente: Elaboración Propia...</i>	<i>74</i>
<i>Figura 34: RNA usando la Función de Activación radbas. Fuente: elaboración Propia....</i>	<i>74</i>
<i>Figura 35: Estructura de un Perceptrón Multicapa. Fuente: http://www.cs.us.es/~fsancho/?e=72</i>	<i>76</i>
<i>Figura 36: Error al utilizar la RNA con más de una capa. Fuente: elaboración Propia....</i>	<i>77</i>
<i>Figura 37: Prueba realizada con 10 neuronas. Fuente: elaboración Propia.....</i>	<i>78</i>
<i>Figura 38: Prueba realizada con 20 neuronas. Fuente: elaboración Propia.....</i>	<i>78</i>
<i>Figura 39: Prueba realizada con 30 neuronas. Fuente: Elaboración propia</i>	<i>79</i>
<i>Figura 40: Prueba realizada con 40 neuronas, no muestra un cambio significativo. Fuente: Elaboración propia.....</i>	<i>79</i>
<i>Figura 41: Entrenamiento con 300 iteraciones. Fuente: Elaboración propia.....</i>	<i>80</i>
<i>Figura 42: Señal procesada con 300 iteraciones. Fuente: Elaboración propia.....</i>	<i>81</i>
<i>Figura 43: Entrenamiento con 100 iteraciones. Fuente: Elaboración propia.....</i>	<i>82</i>
<i>Figura 44: Señal procesada con 100 iteraciones. Fuente: Elaboración propia.....</i>	<i>83</i>
<i>Figura 45: Pantalla Principal. Fuente: Elaboración propia</i>	<i>84</i>
<i>Figura 46: Pantalla de carga de datos. Fuente: Elaboración propia.....</i>	<i>85</i>

<i>Figura 47: Pantalla de señal fundamental y armónico. Fuente: Elaboración propia.....</i>	<i>85</i>
<i>Figura 48: Pantalla de entrenamiento de la RNA. Fuente: Elaboración propia.....</i>	<i>86</i>
<i>Figura 49: Pantalla luego del entrenamiento de la RNA. Fuente: Elaboración propia.....</i>	<i>87</i>
<i>Figura 50: Pantalla al graficar. Fuente: Elaboración propia.....</i>	<i>87</i>
<i>Figura 51: Pantalla al graficar. Fuente: Elaboración propia.....</i>	<i>88</i>
<i>Figura 52: Generando datos con valores por defecto de señal armónica fundamental. Fuente: Elaboración Propia.....</i>	<i>90</i>
<i>Figura 53: Grafica de la señal fundamental generada con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>91</i>
<i>Figura 54: Generando datos para el armónico de orden tres. Fuente: Elaboración propia</i>	<i>92</i>
<i>Figura 55: Gráfica del armónico de orden tres generada con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>92</i>
<i>Figura 56: Gráfica del armónico de orden tres sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>93</i>
<i>Figura 57: Generando datos para el armónico de orden cinco. Fuente: Elaboración propia</i>	<i>94</i>
<i>Figura 58; Gráfica del armónico de orden cinco generado con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>94</i>
<i>Figura 59: Gráfica del armónico de orden cinco sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>95</i>
<i>Figura 60: Generando datos para el armónico de orden siete. Fuente: Elaboración propia</i>	<i>96</i>
<i>Figura 61: Gráfica del armónico de orden siete generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda. Fuente: Elaboración propia.....</i>	<i>96</i>
<i>Figura 62: Gráfica del armónico de orden siete generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda. Fuente: Elaboración propia.....</i>	<i>97</i>
<i>Figura 63: Gráfica del armónico de orden siete sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia.....</i>	<i>97</i>

<i>Figura 64: Generando datos para el armónico de orden nueve. Fuente: Elaboración propia.....</i>	<i>99</i>
<i>Figura 65: Gráfica del armónico de orden nueve generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda. Fuente: Elaboración propia</i>	<i>99</i>
<i>Figura 66: Gráfica del armónico de orden nueve generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda. Fuente: Elaboración propia</i>	<i>100</i>
<i>Figura 67: Gráfica del armónico de orden nueve sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia</i>	<i>100</i>
<i>Figura 68: Generando datos para el armónico de orden once. Fuente: Elaboración propia</i>	<i>102</i>
<i>Figura 69: Gráfica del armónico de orden once generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda. Fuente: Elaboración propia</i>	<i>102</i>
<i>Figura 70: Gráfica del armónico de orden once generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda. Fuente: Elaboración propia</i>	<i>103</i>
<i>Figura 71: Gráfica del armónico de orden once sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia</i>	<i>103</i>
<i>Figura 72: Armónico de orden 3. Fuente: Elaboración propia</i>	<i>105</i>
<i>Figura 73: Armónico de orden 5. Fuente: Elaboración propia</i>	<i>106</i>
<i>Figura 74: Armónico de orden 7. Fuente: Elaboración propia</i>	<i>107</i>
<i>Figura 75: Armónico de orden 9. Fuente: Elaboración propia</i>	<i>108</i>
<i>Figura 76: Armónico de orden 11. Fuente: Elaboración propia</i>	<i>109</i>

CAPÍTULO I: PLANTEAMIENTO TEÓRICO

1. El problema.

1.1. Título descriptivo.

“Detección de Armónicos en Corriente Eléctrica Basado en Redes Neuronales”

1.2. Descripción del problema.

1.2.1. Definición.

Las industrias intentan acortar gastos de producción para sus productos con la intención de generar más ganancias, pero uno de los mayores gastos está dentro del pago de energía eléctrica y mantenimiento de los equipos que usan, adicionalmente puede ocurrir una mala instalación de algún equipo o mala entrega de energía por parte de la empresa proveedora de la misma, este tipo de problemas acarrea la falla en los equipos conectados, el deterioro

de los mismos, reducción del tiempo de vida o un consumo adicional de energía eléctrica a los requeridos por la empresa.

Aunque los equipos eléctricos y electrónicos son la principal fuente de distorsión de la corriente, estos también pueden ser una víctima.

La distorsión de la corriente interacciona con la impedancia del sistema, creando distorsión de voltaje, esta misma puede interferir con la operación del suministro de energía causando retardos de tiempo, restablecimientos del equipo y daños al suministro de energía.

La distorsión armónica afecta a la carga y más a los que necesite que la tensión sea de la forma más pura posible como son los procesos electrónicos, sistemas de comunicación, etc., estas cargas generan problemas en los conductores eléctricos, calentamientos (no aprovechan toda la sección creando el efecto corona), interruptores diferenciales, disparos intempestivos, reactancias inductivas, fusibles, motores, transformadores, conductores eléctricos de medida tipo inducción (lecturas erróneas).

1.2.2. Área científica.

- Área específica: Inteligencia Artificial.
- Línea o Tópico: Redes Neuronales Artificiales.

1.2.3. Tipo y nivel de investigación.

- Tipo de investigación: Tecnológico.
- Nivel de investigación: Descriptivo, Comparativo y Exploratorio.

2. Objetivos.

2.1. Objetivo general.

Aplicar Redes Neuronales Artificiales (RNA) para la detección de armónicos en corriente eléctrica.

2.2. Objetivos específicos.

- Proponer un modelo para la detección y simulación de corriente eléctrica.
- Investigar los problemas originados en la calidad de la energía eléctrica.
- Identificar los armónicos y las frecuencias en las que oscilan.
- Identificar el algoritmo adecuado para implementar en la red neuronal.
- Identificar el algoritmo adecuado para el entrenamiento de la red neuronal.
- Determinar el método apropiado para el análisis de armónicos.
- Desarrollar un simulador para la generación de señales eléctricas.

3. Solución propuesta.

3.1. Justificación del problema.

El poder determinar los armónicos de corriente eléctrica a través de una RNA en base a datos obtenidos por un simulador de flujo de corriente, el cual arroja datos de señal eléctrica normal y las sumas de las señales armónicas con el fin de detectar y/o aprender un patrón de estos mismos en

forma intuitiva y rápida, acortando el tiempo de procesamiento de la información.

3.2. Alcances y limitaciones.

Una limitación fue que la empresa al trabajar con sus propios equipos, nos proporcionó una data procesada, por lo cual se optó por desarrollar un simulador de señales para la generación de datos de corriente eléctrica, que nos proporcione la información en bruto.

4. Antecedentes investigativos.

La búsqueda de información nos proporcionó los siguientes artículos relacionados:

a) “Determinación del Contenido Armónico de Corriente Basado en Redes Neuronales Artificiales para Cargas No Lineales Monofásicas”

Autor: Claudinor Francisco Do Nascimento

Institución: Universidad de Sao Paulo

Año: 2007

Resumen: Se enfoca en el análisis de armónicos utilizando equipos especializados para la recolección de los datos y así poder entrenar la red neuronal artificial, esto permite analizar y anticipar los problemas causados por los armónicos, la RNA usada es el PERCEPTRON MULTICAPA aplicado al MATHLAB.

El algoritmo Levenberg – Marquardt o conocido como backpropagation, es usado para el entrenamiento de la RNA, se utiliza este algoritmo dado que permite un aprendizaje más rápido y da la ventaja de poder hacer dicho entrenamiento en modo OFFLINE y ONLINE.

b) “Fuentes de Distorsión Armónica”

Autor: FaraGauss System S.A. (México) constituida el 25 de Octubre de 1996.

Institución: FaraGauss System S.A.

Año: 2000

Resumen: Faragauss System es una empresa dedicada a la investigación y desarrollo de ciencia y tecnología en el área de electromagnetismo, enfocándose a la reingeniería en los campos de la electricidad y la electrónica.

La información obtenida es sobre calidad de energía y problemas relacionados con corriente eléctrica, de los cuales nos enfocamos a la parte de armónicos, causas de las mismas y como afectan a los equipos eléctricos y electrónicos.

c) “Generalidades sobre Armónicos y su influencia en Sistemas de Distribución de Energía”

Autor: Ernesto Noriega Stefanova

Institución: Empresa Eléctrica Matanzas – Cuba

Año: 2002

Resumen: La información obtenida trata sobre teoría de armónicos de corriente, como hallar las ondas senoidales, aplicación de la transformada de Fourier, y la generación de las gráficas de las curvas de onda.

d) “Predicción de Corrientes Armónicas en Ventanas de Carga Residenciales, Mediante Modelos Neuronales Artificiales”

Autores: Víctor A. Barrera Núñez, Jorge Andrés Cormane Angarita, Gabriel Ordoñez Plata

Institución: Universidad Industrial de Santander

Año: 2004

Resumen: En la actualidad, la mayoría de las cargas residenciales conectadas a los sistemas de distribución de energía eléctrica son no lineales, es decir, la corriente contiene componentes armónicas al igual que la onda de tensión. Estos equipos inyectan armónicos a la red de alimentación eléctrica en el punto de conexión común. Para el análisis de la no linealidad presente en estas cargas, es necesario predecir las corrientes armónicas generadas por este tipo de equipos o grupos de carga. En este artículo se presentan modelos de comportamiento armónico de equipos eléctricos residenciales no lineales utilizando redes neuronales artificiales. Los equipos analizados tienen como característica principal que su sistema de suministro de energía eléctrica está basado en rectificadores monofásicos de onda completa con filtro capacitivo.

5. Discusión sobre los autores.

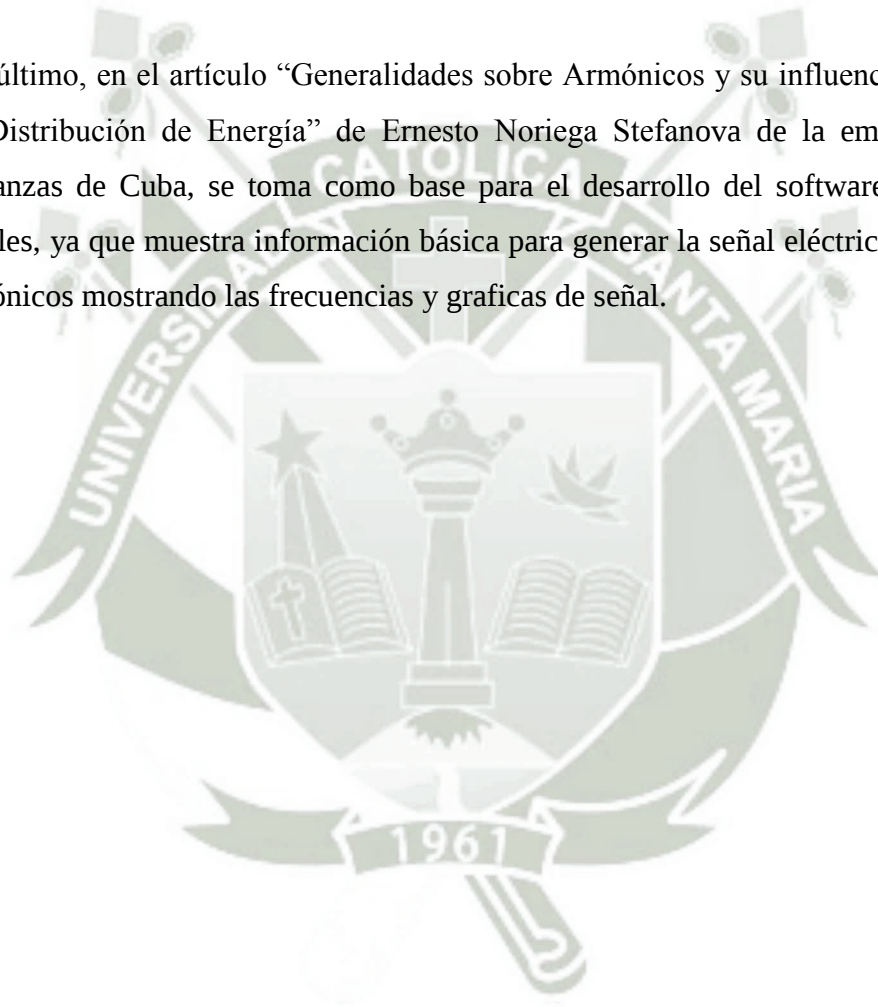
La lectura sobre los artículos, papers y tesis descritas anteriormente nos dan la base para iniciar el desarrollo de esta investigación; la tesis de “Determinación del Contenido Armónico de Corriente Basado en Redes Neuronales Artificiales para Cargas No Lineales Monofásicas” del autor Claudinor Francisco Do Nascimento, utiliza equipos de medición y recolección de datos, de energía eléctrica, en tiempo real para luego procesarlas y analizarlas y así detectar los armónicos de corriente eléctrica. Nosotros al no poseer los equipos de recolección de muestras de corriente eléctrica, se decidió implementar un generador de señales para poder hacer las pruebas correspondientes, también utiliza entrenamiento supervisado para la Red Neuronal Artificial, y nosotros también optamos por usar el mismo tipo de entrenamiento, ya que se requiere de un muestreo inicial para que así, pueda determinar los armónicos correspondientes.

Por otro lado, la empresa FaraGauss System S.A. nos dan conocimiento de que las fuentes de distorsión armónica, aplica a varios equipos, desde motores hasta alumbrado, ellos se enfocan en aminorar o eliminar el tercer armónico de corriente, con la utilización de equipos especiales que ellos proveen. Esta información nos dio el conocimiento de que no solo debemos enfocarnos en un solo armónico, y es por eso que el generador de señales provee hasta el 11avo armónico, siendo este último no muy tomando en cuenta en el campo de la detección de armónicos de corriente.

En el paper “Predicción de Corrientes Armónicas en Ventanas de Carga Residenciales, Mediante Modelos Neuronales Artificiales” de los autores Víctor A. Barrera Núñez, Jorge Andrés Cormane Angarita, Gabriel Ordoñez Plata de la Universidad de Santander, muestran la obtención y medición de armónicos de corriente mediante una simulación, esta simulación está realizada con equipos eléctricos y electrónicos, obteniendo frecuencias que corresponderían si se hiciera una medición a la red eléctrica doméstica, la Red Neuronal Artificial que describen, cuenta con 4 entradas y 10 salidas, ingresando como datos las magnitudes y los ángulos de los armónicos, y mostrando

como resultado la magnitud del componente fundamental expresado como porcentaje, nosotros tomamos en cuenta sus frecuencias de simulación con equipos eléctricos y electrónicos y las utilizamos para nuestro propio simulador de señales, agregando la opción de poder modificar la amplitud de onda. Nuestra Red Neuronal es multicapa de una entrada y una salida, ya que nuestro resultado se muestra de forma gráfica para hacer una comparación visual de los distintos resultados de los diferentes armónicos que pueda haber, haciendo la detección más sencilla.

Por último, en el artículo “Generalidades sobre Armónicos y su influencia en Sistemas de Distribución de Energía” de Ernesto Noriega Stefanova de la empresa eléctrica Matanzas de Cuba, se toma como base para el desarrollo del software generador de señales, ya que muestra información básica para generar la señal eléctrica y la teoría de armónicos mostrando las frecuencias y graficas de señal.



CAPÍTULO II: MARCO TEÓRICO

1. Marco teórico.

1.1. Redes neuronales.

Según [HS, 1999], una red neuronal puede verse como una máquina diseñada originalmente para modernizar la forma en que el sistema nervioso de un ser vivo realiza una determinada tarea. Para lograr este objetivo, una red neuronal está formada por un conjunto de unidades de procesamiento interconectadas llamadas neuronas.

Cada neurona recibe como entrada un conjunto de señales discretas o continuas, las pondera e integra, y transmite el resultado a las neuronas conectadas a ella. Cada conexión entre dos neuronas tiene una determinada importancia asociada denominada peso sináptico o, simplemente, peso. En los pesos se suele guardar la mayor parte del conocimiento que la red neuronal tiene sobre la tarea en cuestión. El proceso mediante el cual se ajustan estos pesos para lograr un determinado objetivo se denomina aprendizaje o entrenamiento y el procedimiento concreto utilizado para ello se conoce como algoritmo de aprendizaje o algoritmo de entrenamiento. El ajuste de pesos es la principal forma de aprendizaje de las redes neuronales, aunque hay otras formas posibles.

Según [POJA, 2002] en su artículo menciona que, McCulloch y Pitts (1943) se considera como el punto de arranque de la investigación en redes neuronales; este trabajo introdujo también la teoría de autómatas finitos como modelo computacional. McCulloch y Pitts propusieron un modelo simplificado de la actividad nerviosa real en el que cada neurona de una red neuronal podía activarse o desactivarse en función de lo que hicieran las neuronas conectadas a ella. Debido a que una neurona solo podía estar activada o desactivada, la capacidad computacional de la red completa se define en términos del conjunto de predicados lógicos que es capaz de computar. En este artículo ya aparecen redes neuronales recurrentes, a las que se denomina redes con ciclos.

Unos años después, Kleene (1956) reformulo algunos de estos resultados e introdujo una notación más compacta y general. En su artículo se define el concepto de expresión regular (allí llamado evento regular), tan importante para la teoría de lenguajes actual y sus aplicaciones. A partir de ahí, el campo de las redes neuronales y el de la teoría de lenguajes comienzan a tomar caminos separados.

De hecho, el segundo acaparó una creciente atención desde entonces hasta nuestros días, mientras que el primero quedó prácticamente olvidado a raíz del trabajo de Minsky y Papert (1969). Salvo algunos trabajos aislados (Werbos 1974), las redes neuronales quedan relegadas a un segundo plano hasta la llegada de los ochenta.

Las redes neuronales destacan por su estructura fácilmente paralelizadle y por su elevada capacidad de generalización (capacidad de producir salidas correctas para entradas no vistas durante el entrenamiento).

Otras propiedades interesantes son:

- ✓ No linealidad.- Una red neuronal puede ser lineal o no lineal. Esta última característica es muy importante, especialmente si se intenta modernizar sistemas generados mediante pautas no lineales.

- ✓ Adaptabilidad.- Las redes neuronales son capaces de reajustar sus pesos para adaptarse a cambios en el entorno. Esto es especialmente útil cuando el entorno que suministra los datos de entrada es no estacionario, es decir, algunas de sus propiedades varían con el tiempo.
- ✓ Tolerancia ante fallos.- Una red neuronal es tolerante ante fallos en el sentido de que los posibles fallos operacionales en partes de la red solo afectan débilmente al rendimiento de esta. Esta propiedad es debida a la naturaleza distribuida de la información almacenada o procesada en la red neuronal.

1.2. Modelo de neurona.

En el modelo más habitual de neurona se identifican cinco elementos básicos para la j -ésima neurona de una red de tiempo discreto:

- Un conjunto de n señales de entrada, $z_i[t]$, $i = 1. \dots n$, que suministran a la neurona los datos del entorno; estos datos pueden ser externos a la red neuronal, pertenecientes a la salida de otras neuronas de la red, o bien correspondientes a la salida anterior de la propia neurona.
- Un conjunto de sinapsis, caracterizada cada una por un peso propio W_{ji} , $i = 1. \dots n$. El peso W_{ji} está asociado a la sinapsis que conecta la unidad i -ésima con la neurona j -ésima.
- Un sesgo W_j cuya presencia aumenta la capacidad de procesamiento de la neurona y que eleva o reduce la entrada a la neurona, según sea su valor positivo o negativo.
- Un sumador o integrador que suma las señales de entrada, ponderadas con sus respectivos pesos, y el sesgo.

- Una función de activación g que suele limitar la amplitud de la salida de la neurona.

Utilizando la notación definida anteriormente podemos describir la operación de una neurona mediante la ecuación que determina su activación en el instante $t + 1$:

$$z_j[t + 1] = g \left(\sum_{i=1}^n W_{ji} z_i[t] + W_j \right)$$

Es habitual, y así se hará normalmente en este trabajo, considerar el sesgo como un peso más de la red y no distinguirlo del resto de pesos sinápticos.

Por tanto, mientras no se diga lo contrario, el termino pesos se refiere indistintamente tanto a W_{ji} como a W_j .

La función de activación es la que define finalmente la salida de la neurona.

Las funciones de activación más utilizadas habitualmente son las siguientes:

- ✓ Función identidad. Tiene la forma $g_I(x) = x$ y se utiliza cuando no se desea acotar la salida de la neurona.
- ✓ Función escalón. Adopta la forma:

$$g_E(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Y proporciona una salida con dos posibles valores. Es habitual encontrársela con el nombre de función de Heaviside.

- ✓ Función logística. Las funciones sigmoideas son un conjunto de funciones crecientes, monótonas y acotadas que provocan una transformación no lineal de su argumento. Una de las más utilizadas es la función logística definida por:

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

La función logística está acotada entre 0 y 1.

- ✓ Función tangente hiperbólica. Otra función sigmoidea es la tangente hiperbólica:

$$g_T(x) = \tanh(x)$$

En este caso, la función está acotada entre -1 y 1. La función logística y la tangente hiperbólica se relacionan mediante la ecuación:

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

Función de base radial. Las más habituales son funciones gaussianas no monótonas del tipo

$$g_B(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Donde σ define la anchura. La función alcanza su valor máximo cuando la entrada es cero.

1.3. Topologías neuronales.

La forma en que se interconectan las neuronas de una red neuronal define un grafo dirigido. Si este grafo es cíclico, la red se denomina red neuronal hacia adelante (en inglés, feedforward network) y en el caso de que posea ciclos, se denomina red neuronal recurrente. En el primer grupo están, entre otros, los perceptrones, las máquinas de vectores soporte y las redes de funciones de base radial (Haykin 1999).

En el caso de las redes recurrentes, los ciclos existentes tienen un profundo impacto en la capacidad de aprendizaje de la red y las hacen especialmente indicadas para el procesamiento de secuencias temporales; son estas últimas, las redes recurrentes, las que se usan en esta tesis.

2. Perceptrón multicapa.

De entre los diferentes modelos de RNA [HGL, 2002] que existen, en esta tesis se ha utilizado el llamado perceptrón multicapa (PMC), por lo que en el presente capítulo se describe dicha red neuronal artificial.

2.1. El combinador lineal adaptativo.

El combinador lineal adaptativo o “neurona artificial” consiste en un elemento cuya salida es una combinación lineal de sus entradas (figura 1). En cada instante k , la salida lineal se calcula como suma ponderada de las entradas $X_k = [X_0, X_{1k}, X_{2k}, \dots, X_{nk}]^T$, es decir, la salida es igual al producto escalar $S_k = X_k^T W_k$. El conjunto de coeficientes $W_k = [W_{0k}, W_{1k}, W_{2k}, \dots, W_{nk}]^T$ se denomina vector de pesos. Las componentes del vector de entrada, pueden ser valores analógicos continuos o valores binarios. Los pesos son variables reales que pueden tomar valores tanto positivos como negativos.

Durante el proceso de entrenamiento (aprendizaje), se presentan al combinador lineal los patrones de entrada X_k , y las respuestas deseadas correspondientes d_k .

El algoritmo de aprendizaje ajusta los pesos de forma que las salidas se aproximen tanto como sea posible a los valores deseados. Un método muy empleado para adaptar los pesos es el algoritmo de mínimos cuadrados ($\alpha - LMS$, “Least Mean Square”) también denominado regla delta de Widrow-Hoff.

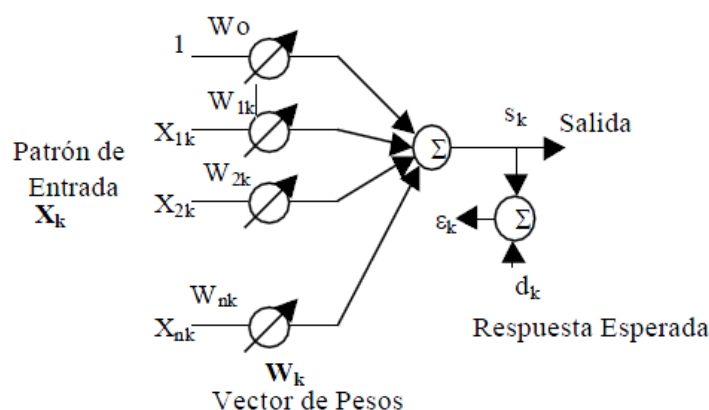


Figura 1: Patrones de entrada y respuesta esperada
Fuente: [HGL, 2002]

Este algoritmo minimiza la suma de los cuadrados de los errores lineales del conjunto de entrenamiento. El error lineal ε_k se define como la diferencia entre la respuesta deseada y la salida en el instante k , o sea $\varepsilon_k = d_k - s_k$. Esta señal de error es necesaria para adaptar los pesos.

2.2. El perceptrón.

El elemento constitutivo básico utilizado en muchas redes neuronales, es el perceptrón. Consiste en un combinador lineal adaptativo seguido de un elemento no lineal. En esa figura el error lineal es ε_k y el error en la salida no lineal (Figura 2) es ε^*k , siendo sus expresiones:

$$\varepsilon_k = d_k - s_k, \quad \varepsilon^*k = d_k - y_k$$

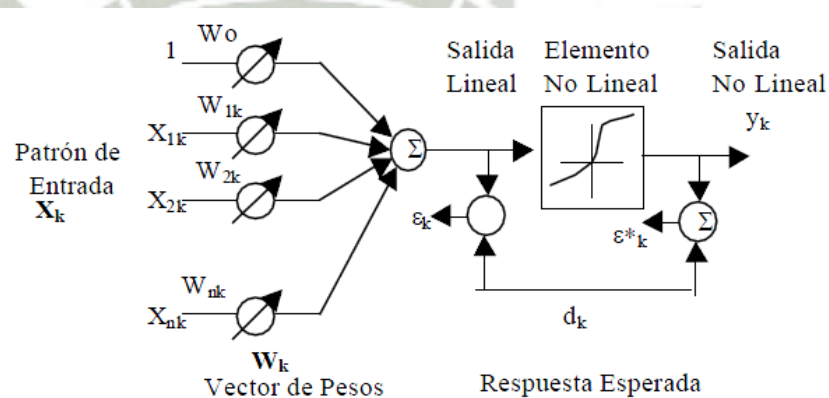
Existen tres tipos comunes de no linealidades: los cuantizadores (función signo), los elementos con umbrales de saturación y las no linealidades de tipo sigmoidal. El peso w_0 , conectado a una entrada constante $x_0 = 1$, controla el nivel de umbral del elemento no lineal.

Separación Lineal: Con n entradas binarias y una salida, un perceptrón como el de la es capaz, tan solo, de realizar un subconjunto de todas las posibles funciones lógicas: las funciones linealmente separables. El perceptrón separa el espacio de patrones de entrada en dos categorías, dependiendo dicha separación de los valores de los pesos. Para un perceptrón de dos entradas, la condición umbral crítica se produce cuando la salida lineal s se iguala a cero:

$$S = x_1w_1 + x_2w_2 + w_0 = 0$$

Por tanto:

$$x_2 = w_1/w_2 \cdot x_1 - w_0/w_2$$



Tipos de no linealidades

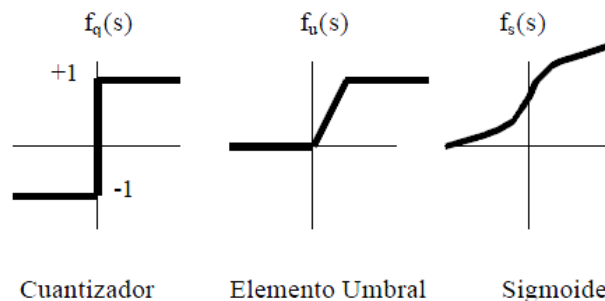


Figura 2: Perceptrón y no-linealidades.
Fuente: [HGL, 2002]

En la Figura 2, se muestra esta relación lineal para un perceptrón con un elemento no lineal de tipo signo. Los tres pesos determinan la pendiente, y la ordenada en el origen, y los semiplanos, a ambos lados de la línea de separación, corresponden a salidas positivas y negativas. Para un perceptrón con tres entradas, la frontera de separación es un plano; con más de tres entradas la frontera es un hiperplano. Si el peso de polarización es cero, el hiperplano es homogéneo, es decir, pasará por el origen del espacio de patrones de entrada. Los patrones de entrada binarios se clasifican como sigue: $(+1, +1) +1$, $(+1, -1) +1$, $(-1, -1) +1$, y $(-1, +1) -1$. Esta función constituye un ejemplo de función linealmente separable.

Un ejemplo de función que no es linealmente separable es la función or exclusiva (xor); no existe ninguna línea recta que pueda lograr la separación de los patrones de entrada, de forma que un simple perceptrón no puede realizar la función XOR.

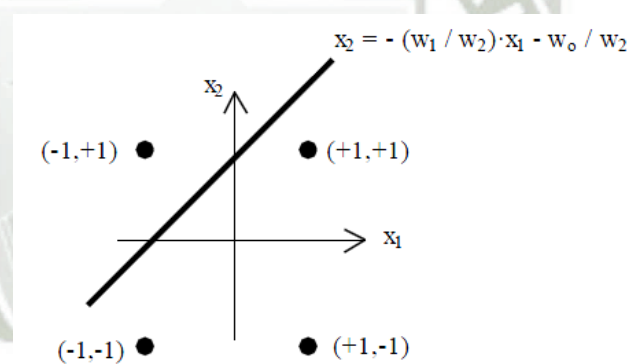


Figura 3: Línea de separación en el espacio de los patrones.
Fuente: [HGL, 2002]

Los pesos de las conexiones y el umbral (w_0) del perceptrón pueden adaptarse utilizando diversos algoritmos. El algoritmo de convergencia del perceptrón, o regla del perceptrón, debido a Rosenblatt, utiliza el error de cuantización, definido como la diferencia entre la respuesta deseada y la salida del elemento no lineal, $e^*k = d_k - s_k$. Los pesos sólo se adaptan cuando el error es distinto de cero. Si la salida es diferente de la deseada, se efectúa la adaptación de los pesos, de forma que

$$w_{k+1} = w_k + \eta(d_k - y_k)x_k$$

Rosenblatt demostró que si el espacio de patrones de entrada es linealmente separable en dos clases, la regla del perceptrón converge, y sitúa el hiperplano de decisión entre ambas clases.

Entre las diversas variantes propuestas a la regla del perceptrón, la más extendida introduce un segundo término en la ecuación de actualización de los pesos, de forma que:

$$w_{k+1} = w_k + \varepsilon^* k \eta(d_k - y_k)x_k + \mu(w_k - w_{k-1})$$

Al término dado por $\mu(w_k - w_{k-1})$, se le denomina término de momento. Intuitivamente, si el cambio de los pesos en la iteración previa fue grande, al añadir una fracción de esta cantidad al peso actual, se acelerará el proceso de descenso del algoritmo hacia el punto de convergencia.

También puede emplearse el algoritmo $\alpha - LMS$ o regla delta para el entrenamiento del perceptrón, sustituyendo en las ecuaciones, el error de cuantización $\varepsilon^* k$ por el error lineal ε_k . No existe, sin embargo, una garantía de convergencia para este algoritmo, si bien en el caso de clases no separables linealmente, suele conducir a soluciones con menor error. Por el contrario, el algoritmo del perceptrón continúa indefinidamente si los patrones no son linealmente separables, y no produce, necesariamente, soluciones con errores pequeños, aún en el caso de que existan.

2.3. El perceptrón multicapa.

Minsky y Papert (1969), realizaron un estudio detallado de los tipos de representaciones posibles con el perceptrón, y demostraron que, en un gran número de casos, estos tipos de redes son incapaces de resolver los problemas de clasificación. Por otro lado, también demostraron que un perceptrón constituido por varias capas, puede realizar cualquier aplicación desde las capas de entrada a las de salida. El inconveniente que se tenía en estos primeros estadios del manejo del perceptrón, era el desconocimiento de un algoritmo de entrenamiento eficiente para redes con más de una capa.

El problema de la separación lineal limita la capacidad de representación del perceptrón, ya que, la probabilidad de que una función lógica sea linealmente separable es muy pequeña, incluso para un número moderado de variables. Por esta razón el perceptrón de una sola capa oculta está limitado, en la práctica, a problemas muy sencillos.

n	Nº Funciones	Funciones linealmente separables
1	4	4
2	16	14
3	256	104
4	65.536	1.882
5	$4.3 \cdot 10^9$	94.572
6	$1.8 \cdot 10^{19}$	5.028.134

Figura 4: Funciones linealmente separables.

Fuente: [HGL, 2002]

El perceptrón multicapa es una red a cíclica de propagación directa con una o más capas de nodos entre las entradas y las salidas. Estas capas adicionales contienen nodos denominados ocultos, ya que no son visibles directamente ni desde las entradas, ni desde las salidas.

La capacidad de computación del perceptrón multicapa procede de las funciones de activación no lineales empleadas en los nodos. La salida de una capa lineal consiste en el producto del vector de entrada por el vector de pesos entrantes en esa capa, es decir, $y_1 = W_1 X_1$. Si a continuación se conecta en cascada otra capa también lineal, cuyas entradas sean las salidas de la capa anterior, la salida de esta segunda capa será $y_2 = W_2 y_1 = W_2 W_1 x_1 = W x_1$. Por tanto, una red de dos (o más) capas lineales, equivale a otra de una sola capa, cuya matriz de pesos sea igual al producto de las dos (o más) matrices de pesos de partida. Como se ha visto en el apartado anterior, las redes de una sola capa están muy limitadas en su capacidad de computación; por tanto, las funciones de activación no lineales son imprescindibles para extender la capacidad de las redes.

Las posibilidades del perceptrón multicapa pueden considerarse desde tres perspectivas diferentes. La primera está relacionada con su capacidad para realizar expresiones booleanas, la segunda, con su capacidad para dividir el espacio de patrones en los problemas de clasificación, y la tercera con su capacidad para realizar transformaciones no lineales en problemas de aproximaciones funcionales.

Un perceptrón puede considerarse como una unidad básica de lógica binaria que es capaz de realizar todas las funciones booleanas excepto las funciones xor y xnor. Conectando en cascada varios perceptrones de forma adecuada, pueden llevarse a cabo expresiones booleanas arbitrarias. En el campo de las redes neuronales el interés se centra no tanto en realizar expresiones booleanas conocidas como en aprender dichas expresiones partiendo de ejemplos de entradas/salidas. Dado que son suficientes dos niveles de puertas lógicas para expresar funciones booleanas

arbitrarias, no son necesarias más que dos capas de perceptrones para, potencialmente, poder aprender dichas funciones.

También ha sido demostrado que un perceptrón de dos capas con funciones de activación de tipo sigmoideal, es suficiente para realizar un mapa no lineal continuo arbitrario. A pesar de estos resultados, para algunos problemas, una red de tres capas puede requerir una cantidad mucho menor de nodos y pesos que una red con sólo una capa oculta.

2.4. Capacidades y limitaciones del perceptrón multicapa.

El PMC es capaz de realizar mapas no lineales arbitrarios; además, dado un conjunto de ejemplos, el algoritmo de retropropagación es capaz de aprender a representar dichos mapas en los puntos de ejemplo. Existen sin embargo otros aspectos de tipo práctico que deben considerarse. Uno de dichos aspectos es la complejidad del aprendizaje, esto es, la posibilidad de aprender el mapa deseado en una cantidad razonable de tiempo.

Se debe considerar también la capacidad de la red para generalizar, es decir, la habilidad para producir respuestas correctas para patrones fuera del conjunto de entrenamiento. En los siguientes párrafos se consideran estas y otras cuestiones.

En primer lugar, los resultados teóricos indican que el PMC es capaz de representar mapas no lineales arbitrarios, pero esto sólo es cierto si el tamaño de la red puede crecer arbitrariamente. Normalmente, no se conoce cuál es el tamaño adecuado de la red para un problema determinado, si bien no suelen ser necesarias más de tres capas (dos capas ocultas). Seleccionar el número de nodos en cada capa no es tampoco una tarea trivial, y requiere un conocimiento específico del problema, a pesar de que se han establecido ciertas reglas para clases de problemas específicos.

Otra cuestión fundamental es la elección de los parámetros para el algoritmo de aprendizaje. Como se ha visto en el apartado anterior, uno de los parámetros clave es el paso de aprendizaje, cuyo valor óptimo depende del problema en consideración. No obstante, las mayores tasas de convergencia suelen lograrse para valores grandes tanto de h (siempre que el aumento en dicho valor no produzca inestabilidades) como el momento m . La elección de los pesos iniciales es también importante, ya que la función de error para el PMC puede tener varios mínimos locales, y el punto final de convergencia depende de las condiciones iniciales. Típicamente, dicho problema se resuelve aplicando el algoritmo de retropropagación varias veces con diferentes pesos iniciales hasta que se encuentra una solución adecuada.

En caso de que sea posible determinar el tamaño óptimo de la red y los parámetros de aprendizaje, el encontrar los pesos de la red que minimizan el error, sigue siendo un problema inherentemente difícil. El problema de encontrar el conjunto de pesos que representa el mapa buscado de forma correcta para un conjunto de entrenamiento, Esto implica que si el problema es muy grande no es probable que sea resuelto en un tiempo razonable.

Por otro lado, a medida que se entrena la red, los pesos pueden tomar valores muy grandes, lo que puede provocar que la mayoría de las neuronas operen en sus regiones de saturación, donde las derivadas de la función de activación son muy pequeñas. Dado que el error retropropagado para el aprendizaje es proporcional a esta derivada, el proceso de entrenamiento puede detenerse virtualmente. Este fenómeno se conoce con el nombre de parálisis de la red.

Una última consideración respecto al PMC es que normalmente son necesarios una gran cantidad de patrones de entrenamiento para que la solución aprendida generalice bien nuevos datos. Parece que el número de patrones de entrenamiento necesarios para una buena generalización es linealmente proporcional al número de

pesos (aproximadamente unas diez veces mayor). Dependiendo de las dimensiones del problema y el tamaño de la red, el número de pesos puede ser muy grande, lo que puede hacer aumentar en gran medida el tiempo necesario para aprender la solución.

A pesar de las limitaciones descritas en los párrafos procedentes, el perceptrón multicapa ha sido aplicado con éxito a cierto número de problemas reales. Estos incluyen procesamiento de señales sonar, control, reconocimiento de caracteres manuscritos, medicina, finanzas, etc. El perceptrón multicapa y sus extensiones proporcionan una potente herramienta para la resolución de problemas de procesamiento de información compleja.

3. ¿Qué son los armónicos?

Según [LRA], los armónicos son distorsiones de las ondas sinusoidales de tensión y/o corriente de los sistemas eléctricos, debido al uso de cargas con impedancia no lineal, a materiales ferromagnéticos, y en general al uso de equipos que necesiten realizar conmutaciones en su operación normal. La aparición de corrientes y/o tensiones armónicas en el sistema eléctrico crea problemas tales como, el aumento de pérdidas de potencia activa, sobretensiones en los condensadores, errores de medición, mal funcionamiento de protecciones, daño en los aislamientos, deterioro de dieléctricos, disminución de la vida útil de los equipos, entre otros.

En un sistema de potencia eléctrica, los aparatos y equipos que se conectan a él, tanto por la propia empresa como por los clientes, están diseñados para operar a 50 o 60 ciclos, con una tensión y corriente sinusoidal. Por diferentes razones, se puede presentar un flujo eléctrico a otras frecuencias de 50 o 60 ciclos sobre algunas partes del sistema de potencia o dentro de la instalación de un usuario. La forma de onda existente está compuesta por un número de ondas sinusoidales de diferentes frecuencias, incluyendo una referida a la frecuencia fundamental. En la figura se observa la descomposición de

una onda distorsionada en una onda sinusoidal a la frecuencia fundamental (60 Hz) más una onda de frecuencia distinta. El término componente armónico o simplemente armónico, se refiere a cualquiera de las componentes sinusoidales mencionadas previamente, la cual es múltiplo de la fundamental. La amplitud de los armónicos es generalmente expresada en por ciento de la fundamental.

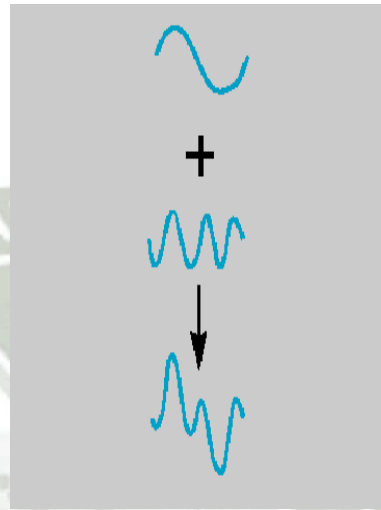


Figura 5: Señal fundamental más señal armónica.
Fuente: [LRA]

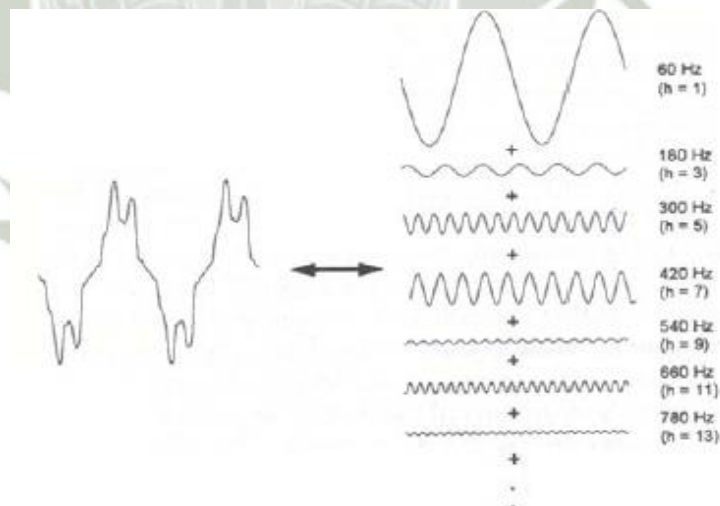


Figura 6: Tipos de frecuencias de armónicos.
Fuente: [LRA]

Los armónicos se definen habitualmente con los dos datos más importantes que les caracterizan, que son:

- su amplitud: hace referencia al valor de la tensión o intensidad del armónico,
- su orden: hace referencia al valor de su frecuencia referido a la fundamental (60 Hz). Así, un armónico de orden 3 tiene una frecuencia 3 veces superior a la fundamental, es decir $3 * 60 \text{ Hz} = 180 \text{ Hz}$.

El orden el armónico, también referido como el rango del armónico, es la razón entre la frecuencia de un armónico f_n y la frecuencia del fundamental (60 Hz).

$$n = \frac{f_n}{f_1} \quad (\text{Por principio, la fundamental } f_1 \text{ tiene rango 1})$$

Cualquier onda no senoidal puede ser representada como la suma de ondas senoidales (armónicos) teniendo en cuenta que su frecuencia corresponde a un múltiplo de la frecuencia fundamental (en el caso de la red = 50 o 60Hz), según la relación:

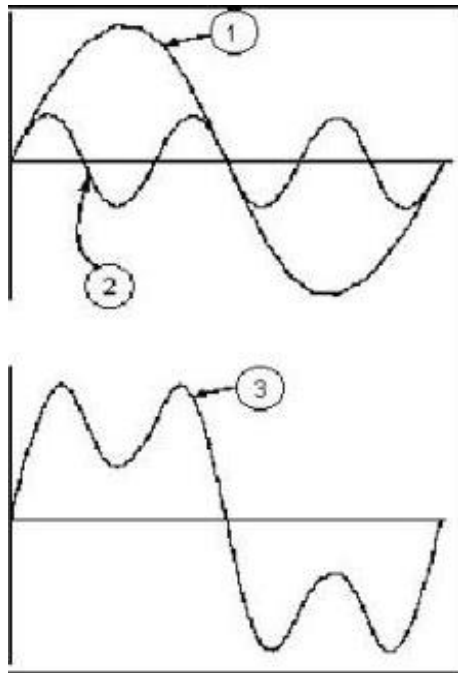
$$v(t) = V_0 + \sum_{k=1}^{\infty} V_k \sin(\omega_k t + \varphi_k)$$

Dónde:

V_0 = Valor medio de $v(t)$ (onda en estudio)

V_1 = Amplitud de la fundamental de $v(t)$

V_k = Amplitud del armónico de orden k de $v(t)$



Leyenda:

Fundamental.

Tercer armónico.

Onda distorsionada.

Figura 7: Sumatoria de un armónico a la señal fundamental. Fuente: [LRA]

3.1. Origen de los armónicos.

En general, los armónicos son producidos por cargas no lineales, lo cual significa que su impedancia no es constante (está en función de la tensión). Estas cargas no lineales a pesar de ser alimentadas con una tensión sinusoidal adsorben una intensidad no sinusoidal, pudiendo estar la corriente desfasada un ángulo φ respecto a la tensión.

Existen dos categorías generadoras de armónicos. La primera es simplemente las cargas no lineales en las que la corriente que fluye por ellas no es proporcional a la tensión. Como resultado de esto, cuando se aplica una onda sinusoidal de una sola frecuencia, la corriente resultante no es de una sola frecuencia. Transformadores, reguladores y otros equipos conectados al sistema pueden presentar un comportamiento de carga no lineal y ciertos tipos de bancos de transformadores

multifase conectados en estrella-estrella con cargas desbalanceadas o con problemas en su puesta a tierra. Diodos, elementos semiconductores y transformadores que se saturan son ejemplos de equipos generadores de armónicos, estos elementos se encuentran en muchos aparatos eléctricos modernos. Invariablemente esta categoría de elementos generadores de armónicos, lo harán siempre que estén energizados con una tensión alterna. Estas son las fuentes originales de armónicos que se generan sobre el sistema de potencia. El segundo tipo de elementos que pueden generar armónicos son aquellos que tienen una impedancia dependiente de la frecuencia.

3.2. Contenido normal de armónicos.

Los armónicos crean problemas sólo cuando interfieren con la operación propia del equipo, incrementando los niveles de corriente a un valor de saturación o sobrecalentamiento del equipo o cuando causan otros problemas similares. También incrementan las pérdidas eléctricas y los esfuerzos térmicos y eléctricos sobre los equipos. Los armónicos lo que generalmente originan son daños al equipo por sobrecalentamiento de devanados y en los circuitos eléctricos, esta es una acción que destruye los equipos por una pérdida de vida acelerada, los daños se pueden presentar pero no son reconocidos que fueron originados por armónicos. El nivel de armónicos presente puede estar justamente abajo del nivel que pueden causar problemas, incrementar este valor límite puede presentarse en cualquier momento y pasar a un valor donde no se pueden tolerar.

4. Herramienta Matlab.

4.1. Introducción.

MATLAB[®] [MATHWORKS] es una técnica de computación del lenguaje de alto nivel y un entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico. El uso del producto MATLAB, puede

resolver problemas de computación técnica más rápida que con los lenguajes de programación tradicionales, tales como C, C++ y Fortran.

Se puede utilizar MATLAB en una amplia gama de aplicaciones, incluyendo la imagen y el procesamiento de señales, comunicaciones, diseño de control, de prueba y medición, modelado y análisis financiero y biología computacional. Add-on cajas de herramientas (colecciones de efectos especiales-las funciones de MATLAB, disponible por separado) amplían el entorno de MATLAB para resolver clases particulares de problemas en estas áreas de aplicación.

MATLAB proporciona una serie de funciones para documentar y compartir su trabajo. Puede integrar su código de MATLAB con otros lenguajes y aplicaciones, y distribuir sus algoritmos de MATLAB y aplicaciones.

4.2. Características principales.

- ✓ Lenguaje de alto nivel para la computación técnica.
- ✓ Entorno de desarrollo para la gestión de código, archivos y datos.
- ✓ Herramientas interactivas para la exploración iterativa, el diseño y la resolución de problemas.
- ✓ Funciones matemáticas para álgebra lineal, estadísticas, análisis de Fourier, filtrado, optimización e integración numérica.
- ✓ 2-D y las funciones de gráficos en 3-D para visualizar datos.
- ✓ Herramientas para la construcción de interfaces gráficas de usuario.

-
- The screenshot displays the MATLAB R2022b environment with the following components:
- Top Toolbar:** Includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. Below these are icons for New Script, Find Files, Import Data, Save Workspace, Open Variable, Run and Time, Analyze Code, Simulink Library, Preferences, Set Path, Help, and Request Support.
 - Editor Window:** Shows a script named 'FilterData.m' with the following code:


```
1 % Signal Parameters
2 N = 10; % Order
3 fcl = 290; % First cutoff frequency
4 fcu = 310; % Second cutoff frequency
5
6 % Filter and Plot the Signal
7 Hd = createfilter(fcl,fcu,'fs');
8 y = filter(Hd,x);
9
10 filterplot(t,y,x)
11
12 % Amplitude Histogram
```
 - Command Window:** Shows the execution of the script:


```
>> fname = 'Signal.mat';
>> [x,fs,bits] = wavread(fname);
>> t = (0:length(x)-1)/fs';
>> dt = 1/fs;
>> npts = 512;
f1 >> spectrogram(x,t,fs)
```
 - Workspace:** Lists variables: Name, Value, Min, Max.

Name	Value	Min	Max
bits	16	16	16
dt	2.3676e-05	2.3676e-05	2.3676e-05
fcl	290	290	290
fcu	310	310	310
fname	'Signal.mat'		
 - Plots:**
 - Filtered Signal:** A line plot showing the filtered signal (blue) over time (s) from 0.61 to 0.66. The amplitude ranges from -0.01 to 0.01.
 - Raw Signal:** A line plot showing the raw signal (green) over time (s) from 0.61 to 0.66. The amplitude ranges from -1 to 1.
 - Unfiltered Signal Spectrogram:** A 3D surface plot showing the spectrogram of the raw signal. The axes are Time (0 to 2 seconds), Frequency (0 to 10 Hz), and Amplitude (0 to 15). A peak is labeled with coordinates: X: 5254, Y: 11.89, Z: 69.00.

4.3. Herramientas de desarrollo.

MATLAB incluye herramientas de desarrollo que le ayudarán a implementar su algoritmo eficiente. Estos incluyen los siguientes:

- 45

- ✓ **MATLAB Profiler** - Se registra el tiempo de ejecución de cada línea de código.
- ✓ **Directorio de Informes** - Analizar todos los archivos en un directorio y un informe sobre la eficiencia del código, las diferencias de archivos, dependencias de los archivos, y la cobertura de código.

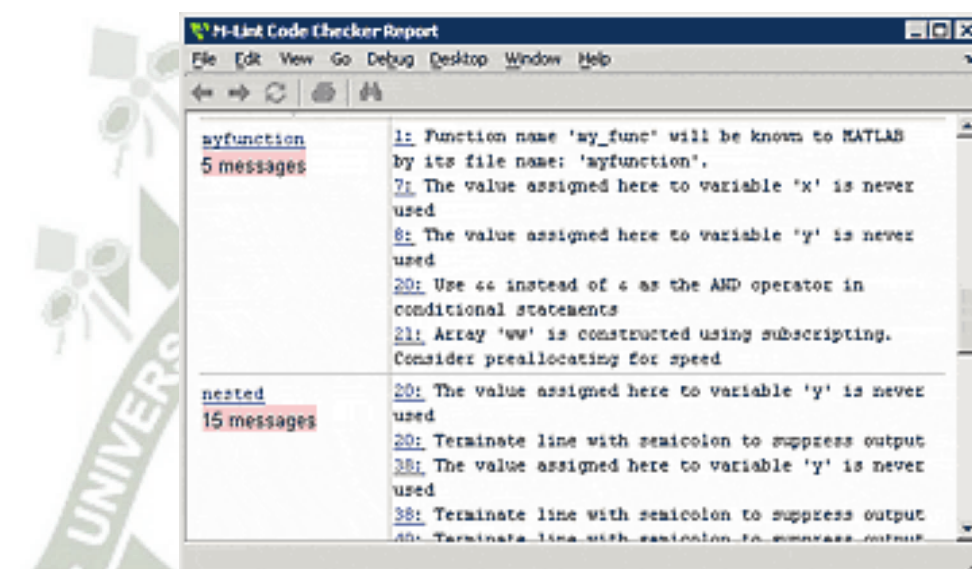


Figura 9: Un M-Lint Código Verificador de informe que incluya recomendaciones para hacer el código más rápido y más fácil de mantener. Fuente: http://www.mathworks.cn/cmsimages/ml_m-lintcodechecker_wl_7157.gif

4.4. Diseño de interfaces graficas de usuario.

Se puede utilizar la herramienta interactiva GUIDE (Graphical User Interface Development Environment) para diseño y edición de interfaces de usuario. GUÍA permite incluir cuadros de lista, menús desplegables, botones, botones de radio, y reguladores, así como parcelas de MATLAB y controles ActiveX. Como alternativa, puede crear interfaces gráficas de usuario mediante programación con funciones de MATLAB.

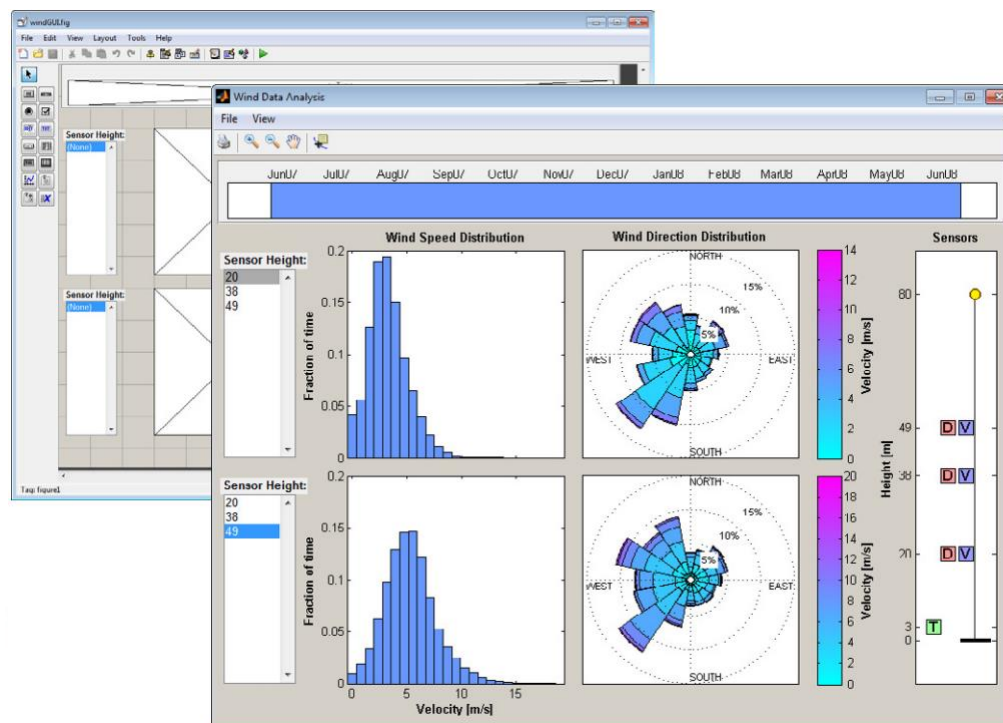


Figura 10: GUÍA diseño de un análisis wavelet GUI (arriba), junto con la interfaz completo (abajo).
Fuente: [MATHWORKS]

4.5. Lenguaje de programación para el desarrollo del software.

4.5.1. C# (C Sharp)

C# o C Sharp es un lenguaje de programación que está incluido en la Plataforma .NET y corre en el Lenguaje Común en Tiempo de Ejecución (CLR, Common Language Runtime). El primer lenguaje en importancia para el CLR es C#, mucho de lo que soporta la Plataforma .NET está escrito en C#. C# intenta ser el lenguaje base para escribir aplicaciones .NET.

C# deriva de C y C++, es moderno, simple y enteramente orientado a objetos, simplifica y moderniza a C++ en las áreas de clases, namespaces, sobrecarga de métodos y manejo de excepciones. Se eliminó la complejidad de C++ para hacerlo más fácil de utilizar y menos propenso a errores.

Algunas características de C# son:

- ✓ C# provee el beneficio de un ambiente elegante y unificado.
- ✓ No maneja apuntadores, para emular la función de los apuntadores se utiliza delegates el cual provee las bases para el .NET event model.
- ✓ Por default trabaja con código administrado.
- ✓ La Plataforma .NET provee un colector de basura que es responsable de administrar la memoria en los programas C#.
- ✓ El manejo de errores está basado en excepciones.
- ✓ No existen funciones globales, variables o constantes. Todo deber ser encapsulado dentro de la clase, como un miembro de la instancia (accesible vía una instancia de clase) o un miembro estático (vía el tipo).
- ✓ Soporta los modificadores de acceso private, protected, public y agrega un cuarto modificador internal.
- ✓ Permite acceder a diferentes APIs a través de .NET Common Language Specification, el cual define el estándar de interoperabilidad entre lenguajes que se adhieran a este estándar.
- ✓ La Plataforma .NET provee un acceso transparente a COM.
- ✓ Soporta OLE.

- ✓ Permite la interoperabilidad con APIs al estilo C y DLLs, esta característica para acceder a APIs nativas es llamada Platform Invocation Services (PInvoke)

Por las características presentadas anteriormente, se eligió usar este lenguaje de programación que a la vez está muy difundido a nivel mundial como da cuenta TIOBE en su ranking de lenguajes de programación de mayor popularidad que está basado en los skills que dan varios ingenieros alrededor del mundo, cursos dados de este lenguaje y una tercera parte proveniente de los vendedores de software. La siguiente grafica muestra el ranking de TIOBE actualizado hasta Junio para el 2014.



Jun 2014	Jun 2013	Change	Programming Language	Ratings	Change
1	1		C	16.191%	-1.62%
2	2		Java	16.113%	-0.54%
3	3		Objective-C	10.934%	+0.58%
4	4		C++	6.425%	-2.39%
5	6	⬆	C#	3.944%	-1.84%
6	7	⬆	(Visual) Basic	3.736%	-0.61%
7	5	⬇	PHP	2.848%	-3.14%
8	8		Python	2.710%	-1.47%
9	10	⬆	JavaScript	2.000%	+0.35%
10	12	⬆	Visual Basic .NET	1.914%	+0.85%
11	13	⬆	Transact-SQL	1.898%	+0.99%
12	9	⬇	Perl	1.480%	-0.79%
13	11	⬇	Ruby	1.392%	-0.09%
14	42	⬆	ActionScript	1.327%	+1.13%
15	43	⬆	F#	1.058%	+0.89%
16	14	⬇	Lisp	0.843%	-0.04%
17	18	⬆	Delphi/Object Pascal	0.833%	+0.23%
18	15	⬇	Pascal	0.808%	+0.03%
19	20	⬆	MATLAB	0.774%	+0.21%
20	23	⬆	Assembly	0.771%	+0.27%

Figura 11: Ranking Top 20 de TIOBE – Julio 2013.
Fuente: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Nótese que el índice de TIOBE no es sobre el mejor lenguaje de programación, su cálculo se realiza usando las máquinas de búsqueda más populares como son Google, Bing, Yahoo!, Wikipedia, YouTube y Baidu.

Este índice puede ser usado para tomar una decisión estratégica sobre cual lenguaje de programación puede ser adoptado para diseñar un nuevo software.

A continuación se muestra una gráfica del crecimiento histórico del C# entre el 2002 y el 2014.

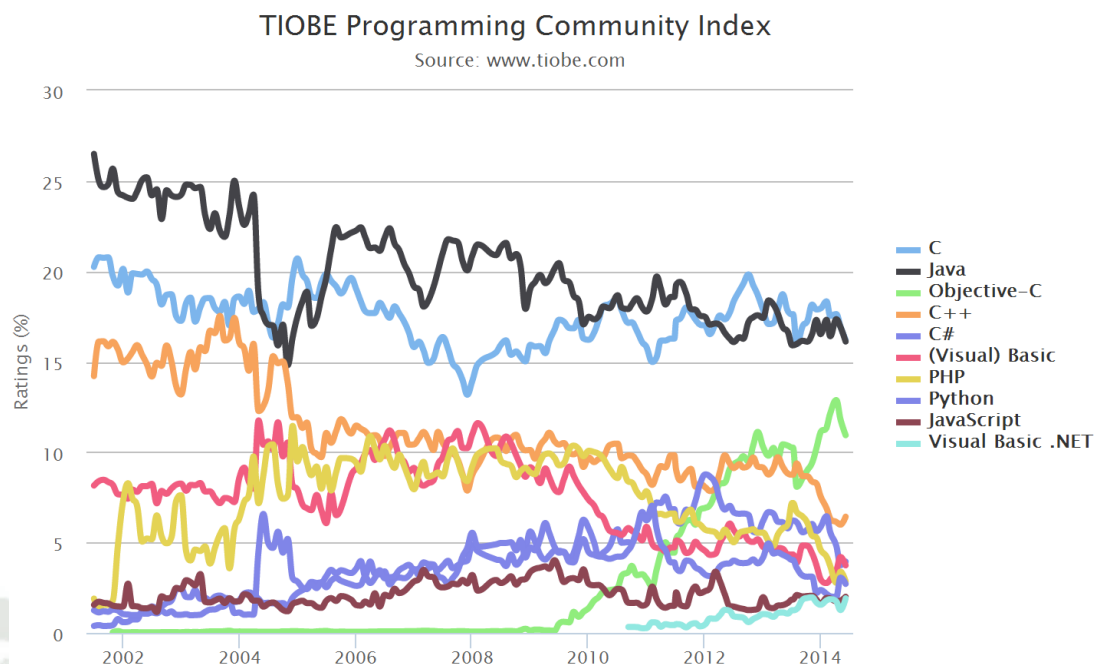


Figura 12: Histórico del Lenguaje C#: Rating Alto (desde el 2001): 8.763% (Posición 3 en Enero 2012).
Fuente: http://www.tiobe.com/index.php/paperinfo/tpci/C_.html

CAPÍTULO III: MODELO PROPUESTO

1. Descripción del capítulo.

En este capítulo se implementará el modelo propuesto, el cual constará de la creación de un software para emular las señales eléctricas de corriente, a la vez poder representar en una gráfica estos datos obtenidos, este software también podrá generar y mostrar los armónicos de corriente eléctrica, estos datos obtenidos serán exportados a la red neuronal implementada en MatLab para el aprendizaje y/o entrenamiento correspondiente y mostrar los resultados de los mismos, estos últimos serán interpretados y comparados con los datos de I+T.

2. Etapas del modelo propuesto.

Este modelo está compuesto por 3 etapas:

- ✓ Generar los datos de la señal deseada.
- ✓ Entrenar la red con la señal fundamental y la señal armónica.
- ✓ Interpretar el resultado de la señal.

2.1. Generador de señales.

Descripción: Este generador de señales eléctricas está basado en las formulas encontradas en la investigación previa al desarrollo de esta tesis, asignándole los valores necesarios para recrear los datos necesarios, en base a estos poder escoger entre los tipos de armónicos comunes (3ro, 5to, 7mo y 9no), generar la suma del armónico a la señal fundamental, poder graficar el resultado en pantalla y poder exportar los datos en un archivo de texto el cual será leído por la red neuronal implementada en MatLab.

El formulario contará con el adicional de cálculo de tiempo que tarda el proceso de generación de los datos, estos datos serán variables dependiendo del equipo de cómputo con el que se trabaje, los datos estarán visibles en pantalla en todo momento.

El formulario en el que se muestra la gráfica en base a los datos calculados, tendrá la funcionalidad de modificar el ancho y la amplitud de onda, estos cambios se mostrarán en pantalla luego de la selección, servirán para poder observar con mayor detenimiento la onda graficada con los datos obtenidos previamente.

- Desarrollo:

Para la creación del generador de señales se usa como base la fórmula matemática de la onda senoide, la fórmula matemática es:

$$A.\text{sen}(\omega x + \varphi)$$

Esta misma fórmula aplicada a la corriente eléctrica en función del tiempo se define como:

$$a(t) = A_0.\text{sin}(\omega t + \beta)$$

Dónde:

- A_0 : es la amplitud en voltios o amperios (también llamado valor máximo o valor pico).
- ω : es la pulsación en radianes/segundo.
- t : es el tiempo en segundos.
- β : es el ángulo de fase inicial en radianes.

La velocidad angular (ω) se puede expresar en forma matemática dentro de la fórmula de la siguiente manera:

$$a(t) = A_0 \cdot \sin(2\pi ft + \beta)$$

Donde f es la frecuencia en hercios (Hz) y equivale a la inversa del periodo $f = \frac{1}{T}$.

Los valores más usados en la distribución son de 50 Hz y 60 Hz. En nuestro caso, se utiliza el valor de 60 Hz correspondiente a las características de corriente dentro de nuestro país.

Teniendo la formula definida se pasa a desarrollar el software para el generador comenzando a generar los puntos a partir de la fórmula aplicada a corriente alterna.

2.2. Diagramas UML del Generador de Señales.

2.2.1. Diagrama de Flujo

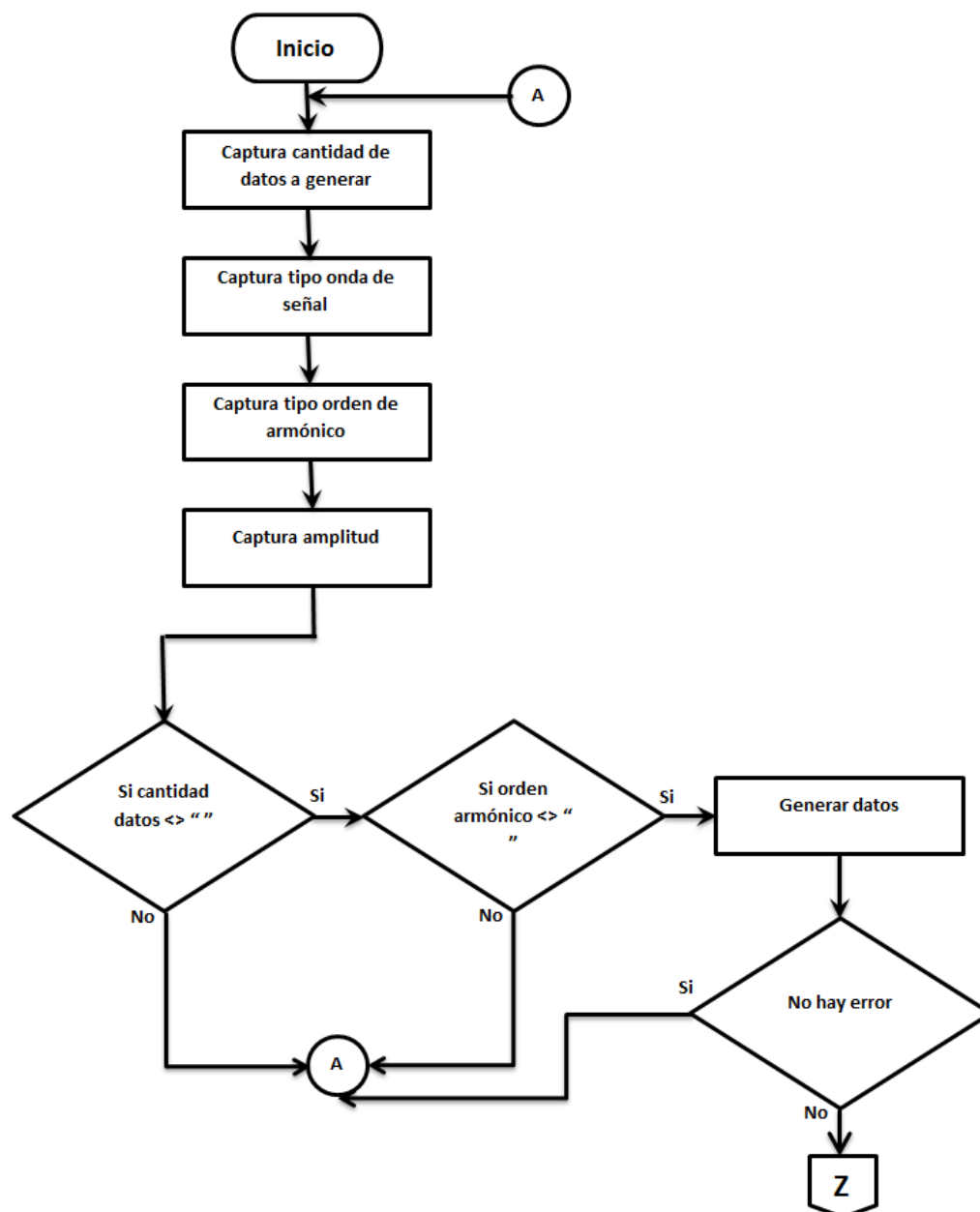


Diagrama 1: Flujograma del generador de señales. (a)
Fuente: Elaboración Propia

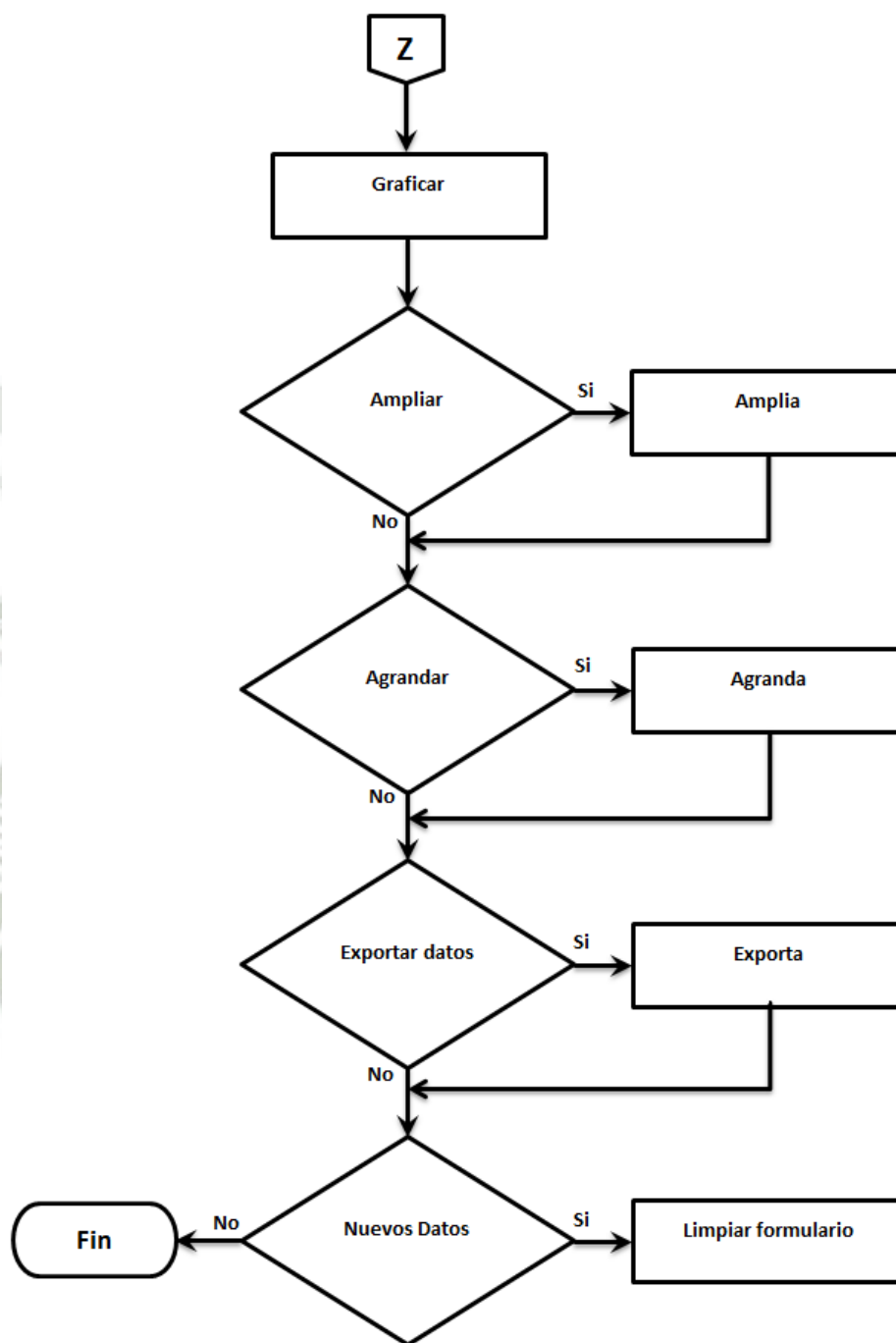


Diagrama 2: Flujograma del generador de señales. (b)
Fuente: Elaboración Propia

2.2.2. Diagrama de Casos de Uso.

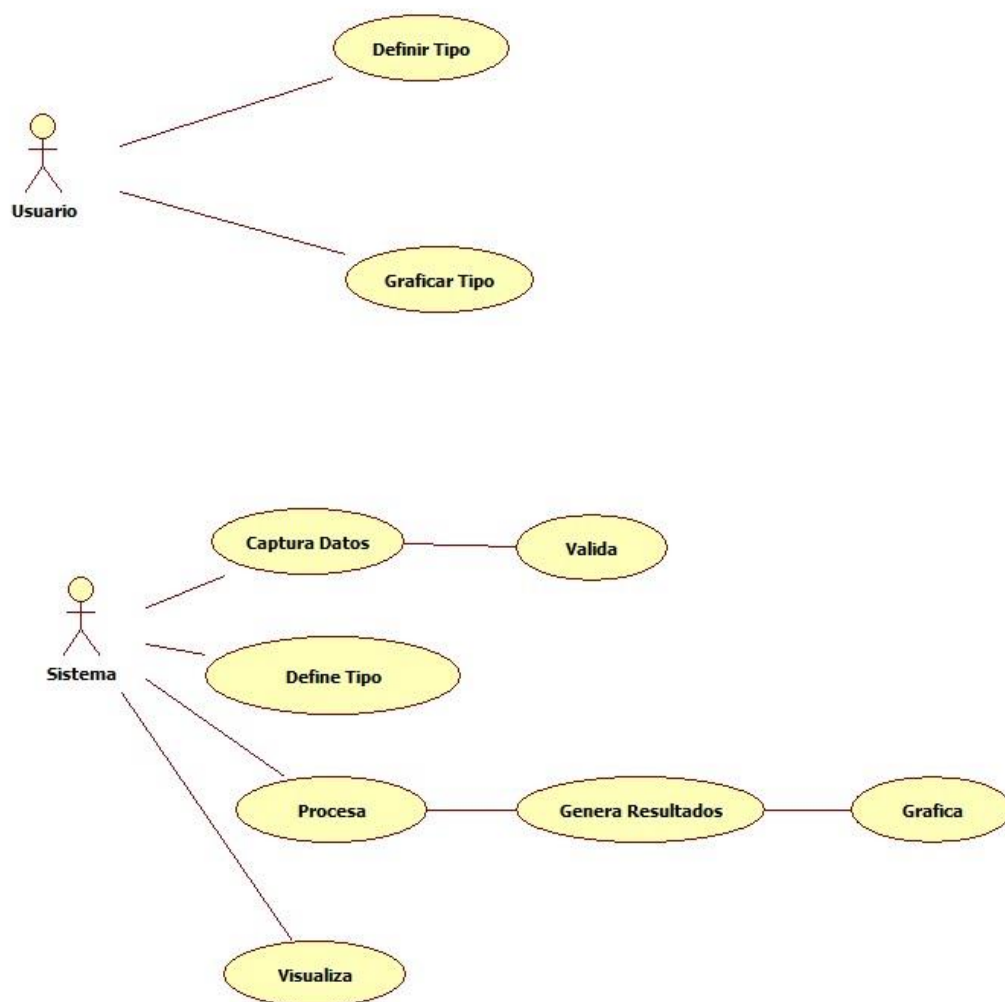


Diagrama 3: Casos de Uso.
Fuente: Elaboración Propia

2.2.3. Diagrama de Clases.

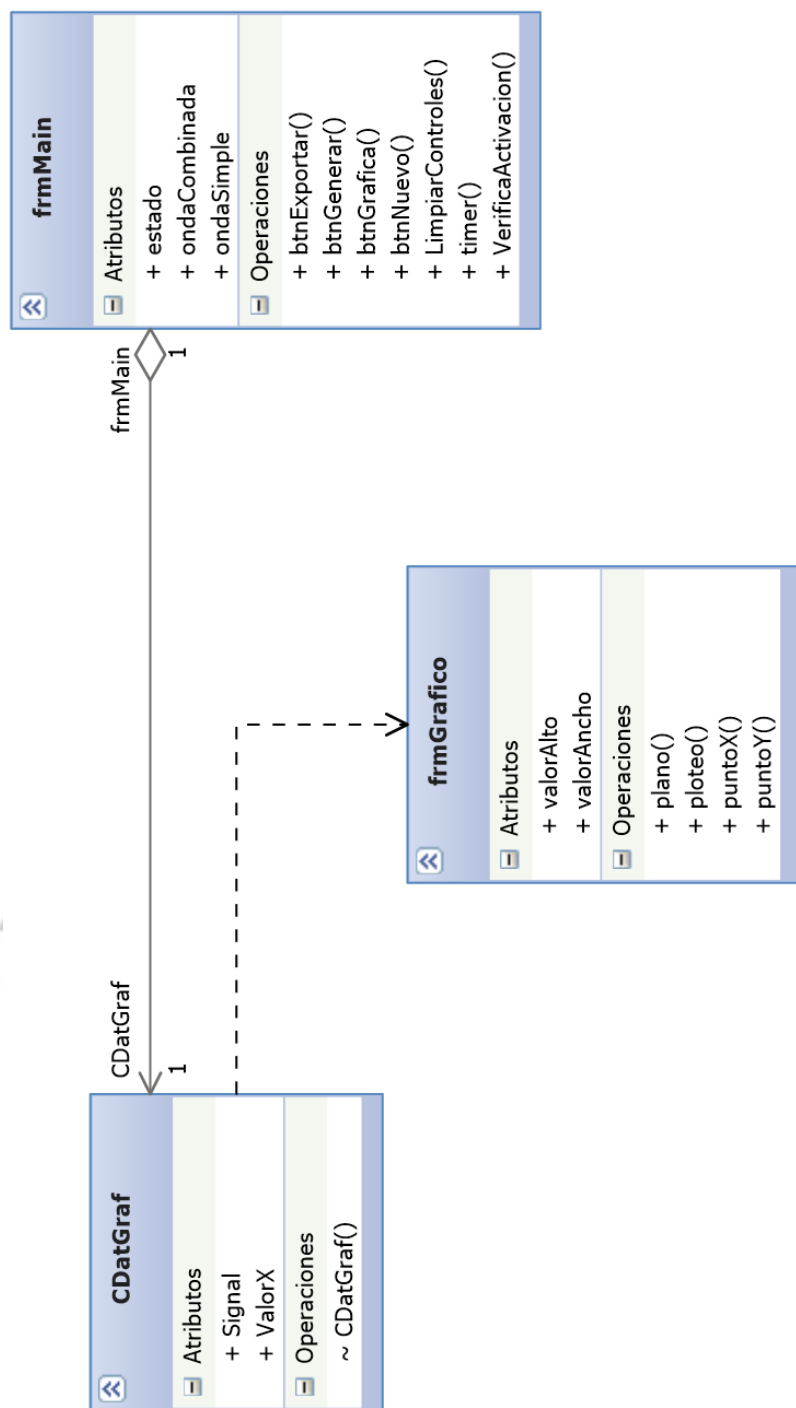


Diagrama 4: Diagrama de Clases.
Fuente: Elaboración Propia

2.2.4. Diagrama de Estado.

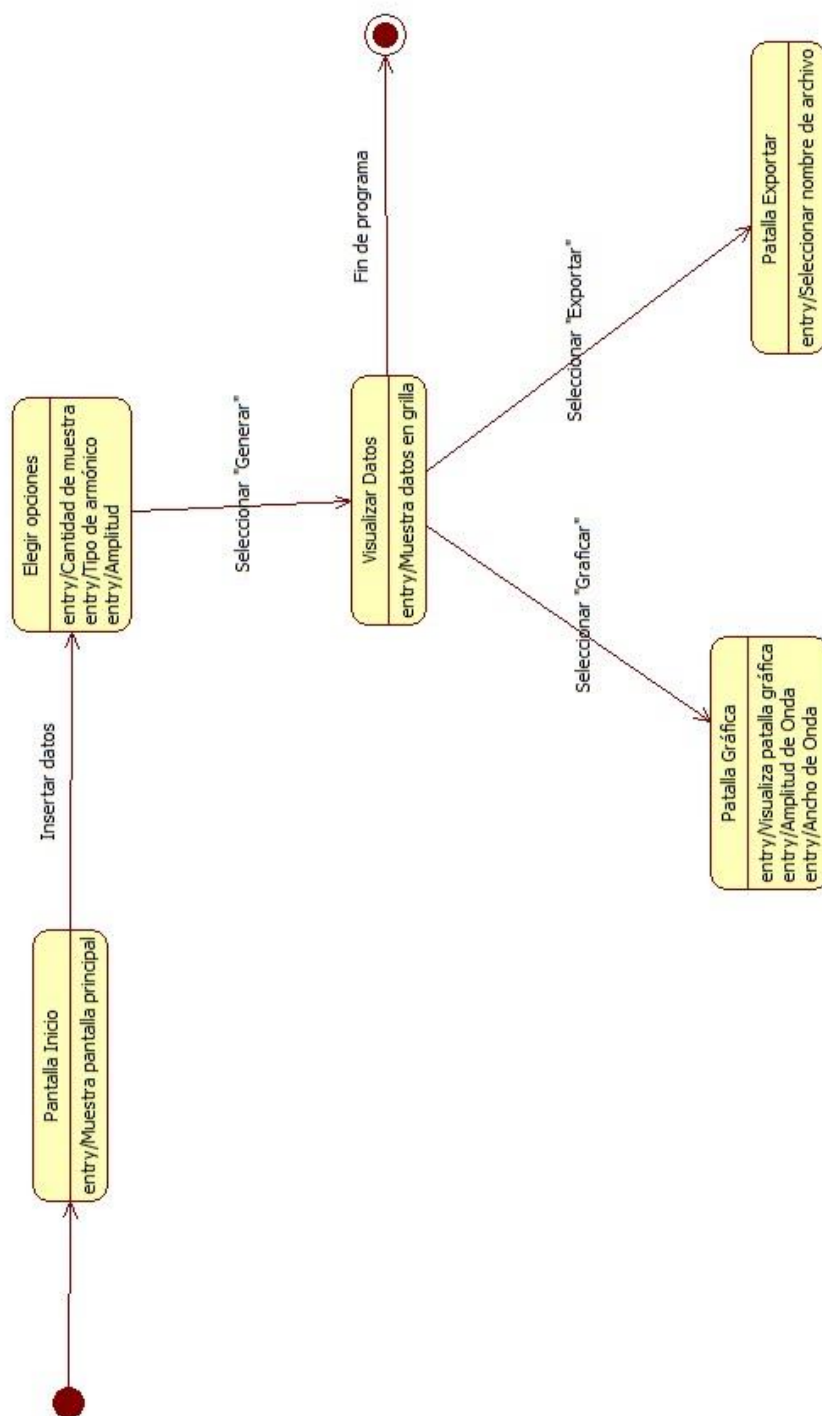


Diagrama 5: Casos de Estado.
Fuente: Elaboración Propia

2.2.5. Diagrama de componentes.

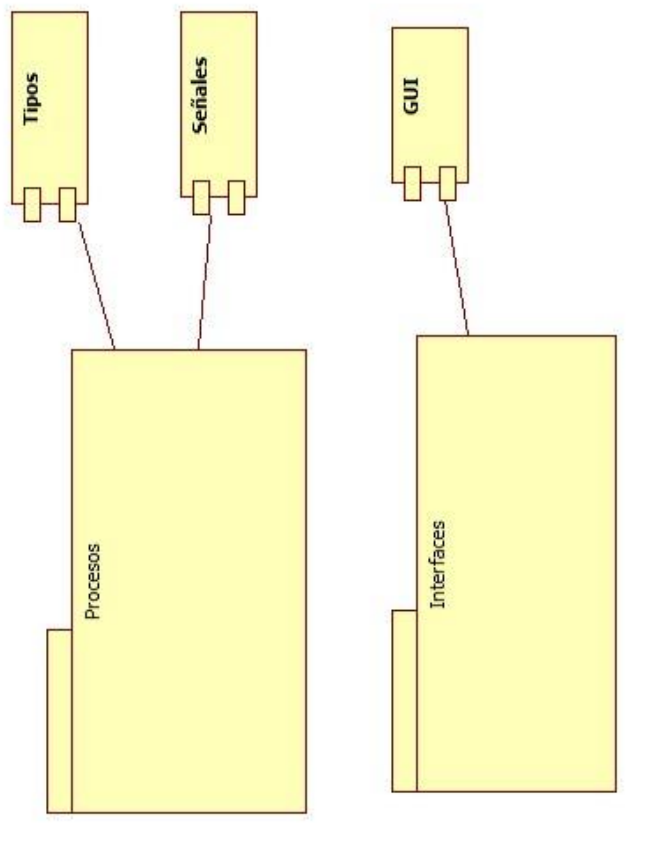


Diagrama 6: Casos de Componentes.
Fuente: Elaboración Propia

2.2.6. Diagrama de Secuencia.

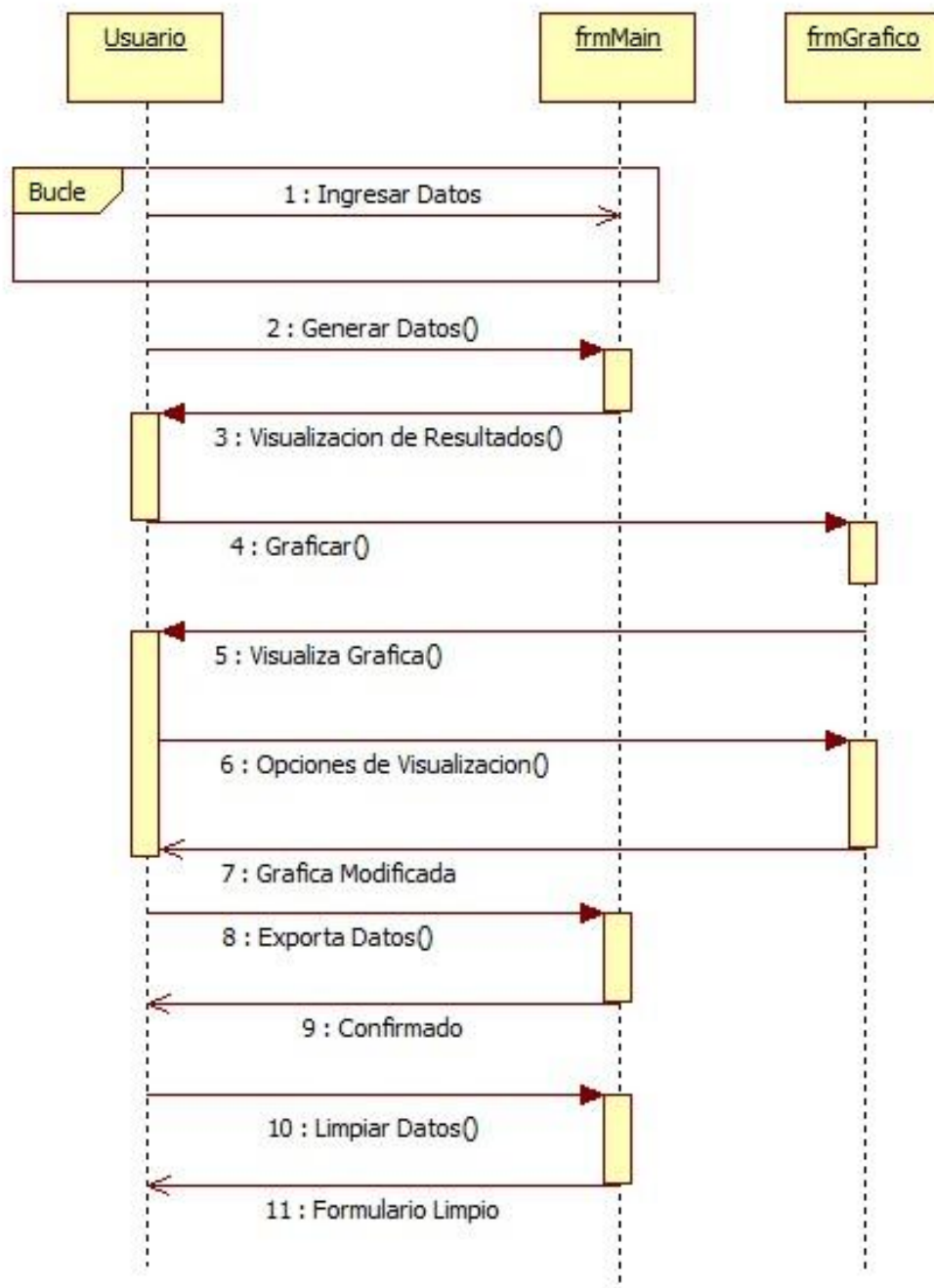


Diagrama 7: Casos de Secuencias.
Fuente: Elaboración Propia

2.3. Diseño de Pantallas.

A continuación se presenta el diseño de pantallas de dicho software detallando su funcionamiento.

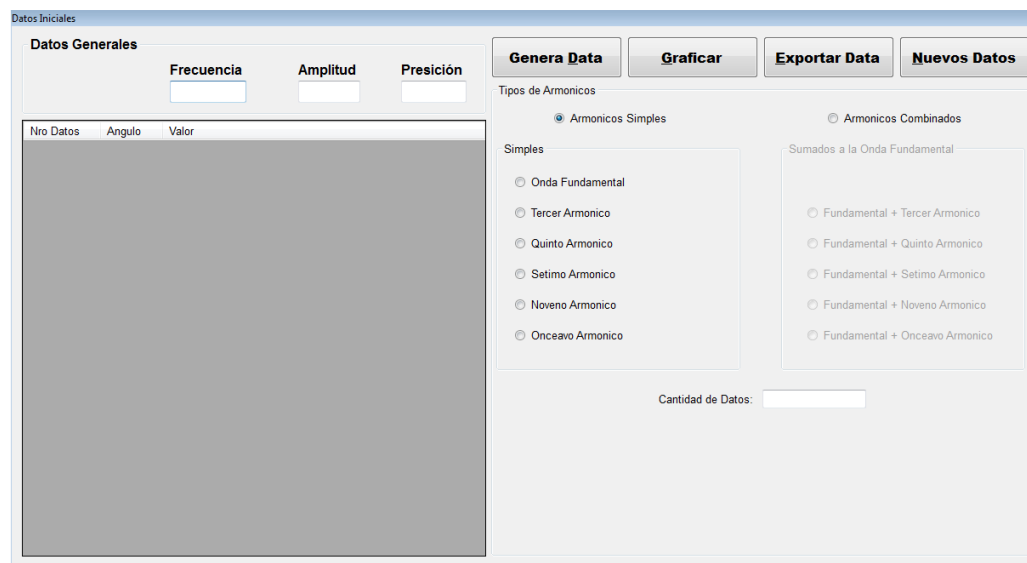
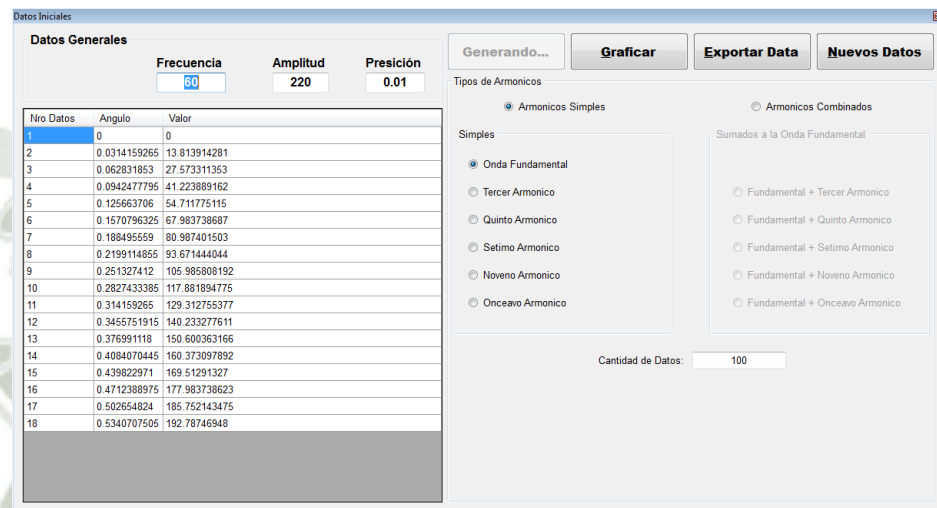


Figura 13: Pantalla Principal del Generador de Señales.
Fuente: Elaboración Propia

En la figura anterior se muestra la pantalla principal de este generador, se observa la siguiente distribución:

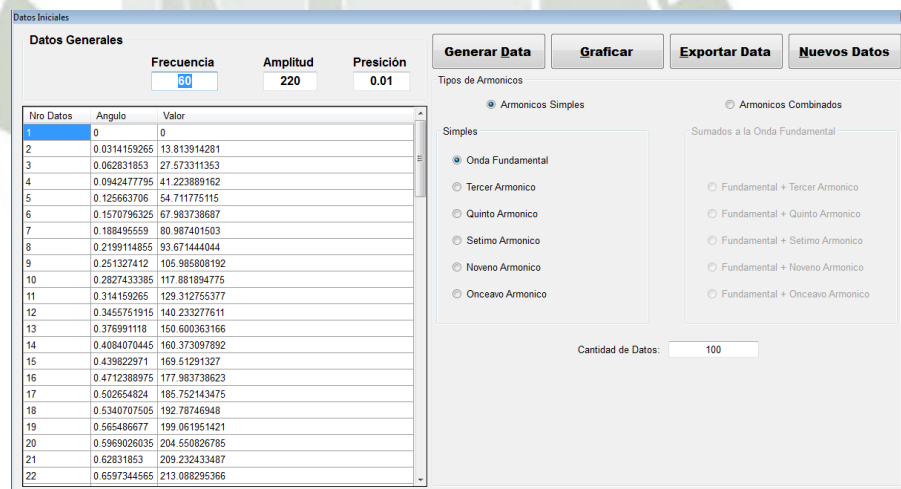
- Botón Generar Data: este botón generará una data de acuerdo a las especificaciones dadas en las cajas de texto ubicadas al lado izquierdo (Frecuencia, Amplitud y Precisión), también se observa en el lado derecho una opción para poder escoger qué onda generará el software, puede ser armónicos simples o armónicos combinados, es decir, este último es la suma de la señal fundamental y su respectivo armónico, esta opción ya aparece seleccionada por defecto, depende al tipo de armónico que se elija se habilitará las secciones de armónicos simples y/o combinados, esta opción tiene que ser seleccionada por el usuario en forma obligatoria, al

mismo tiempo el usuario deberá escoger la cantidad de datos que se desea generar, si no se le colocara ninguna información en las cajas de texto de frecuencia, amplitud y precisión, el software generara una data por defecto, generando una onda como la siguiente:



Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.813914281
3	0.062831853	27.573311353
4	0.0942477795	41.223889162
5	0.125663706	54.711775115
6	0.1570796325	67.983738687
7	0.188495559	80.987401503
8	0.2199114855	93.671444044
9	0.251327412	105.985808192
10	0.2827433385	117.881894775
11	0.314159265	129.312755377
12	0.3455751915	140.233277611
13	0.376991118	150.600363166
14	0.4084070445	160.373097892
15	0.439822971	169.51291327
16	0.4712388975	177.983738623
17	0.502654824	185.752143475
18	0.5340707505	192.78746948

Figura 14: Proceso de creación de datos.
Fuente: Elaboración Propia



Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.813914281
3	0.062831853	27.573311353
4	0.0942477795	41.223889162
5	0.125663706	54.711775115
6	0.1570796325	67.983738687
7	0.188495559	80.987401503
8	0.2199114855	93.671444044
9	0.251327412	105.985808192
10	0.2827433385	117.881894775
11	0.314159265	129.312755377
12	0.3455751915	140.233277611
13	0.376991118	150.600363166
14	0.4084070445	160.373097892
15	0.439822971	169.51291327
16	0.4712388975	177.983738623
17	0.502654824	185.752143475
18	0.5340707505	192.78746948
19	0.565486677	199.061951421
20	0.5969026035	204.550826785
21	0.62831853	209.232433487
22	0.6597344565	213.088295366

Figura 15: Generación de armónicos terminada.
Fuente: Elaboración Propia

- Como se observa en la figura anterior, la data que se genera por defecto está dada por los elementos ubicados en las cajas de texto como son: Frecuencia, Amplitud y Precisión.
- Botón Graficar: este botón abrirá una nueva ventana que mostrara la gráfica de acuerdo a los datos generados, en este paso solo ha creado una onda fundamental.

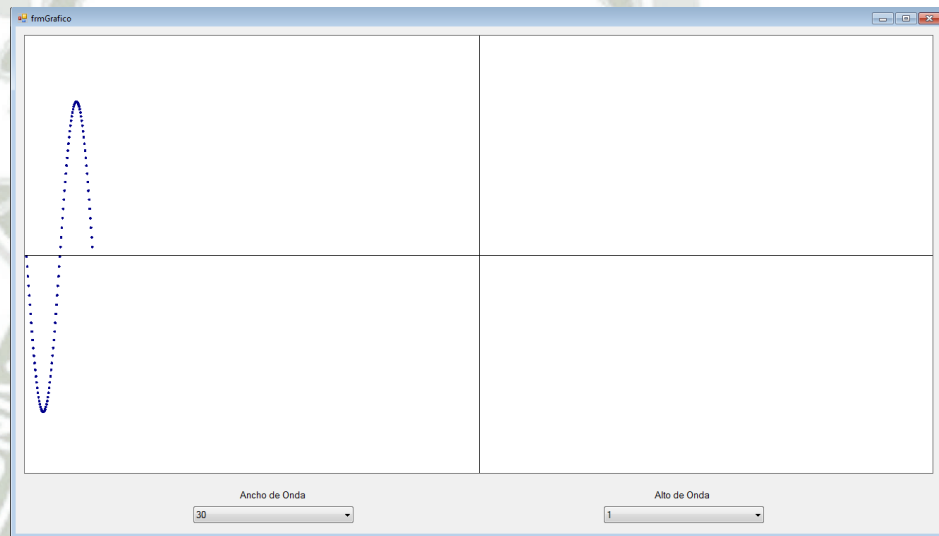


Figura 16: Gráfica generada de acuerdo a los datos ingresados.
Fuente: Elaboración Propia.

- Se observa en la nueva ventana la gráfica creada a partir de los datos generados, si bien las ondas están juntas, las opciones debajo de la gráfica son una ayuda para ampliar o reducir el ancho de la onda al igual que la amplitud de la misma.

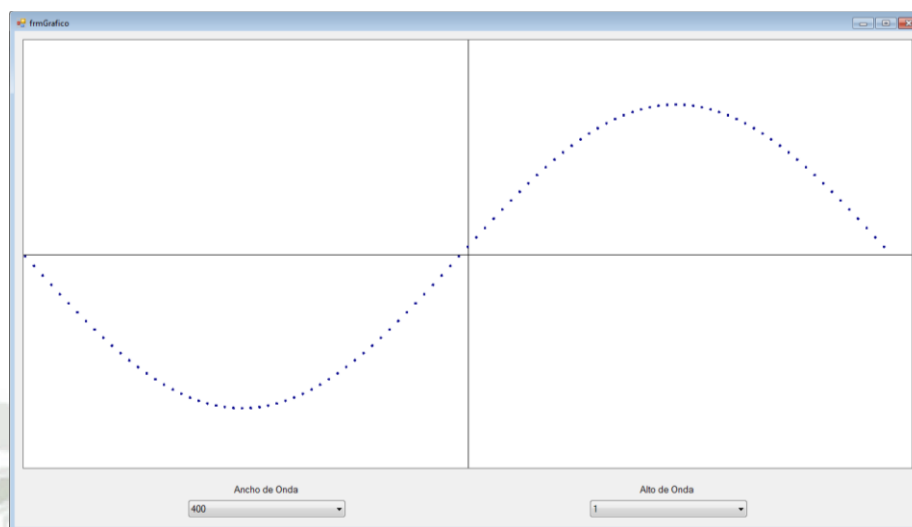


Figura 17: Modificación de onda generada.

Fuente: Elaboración Propia.

- Regresamos a la pantalla anterior y generamos una nueva señal pero esta vez con la sumatoria del 5to armónico, la siguiente figura muestra la data generada:

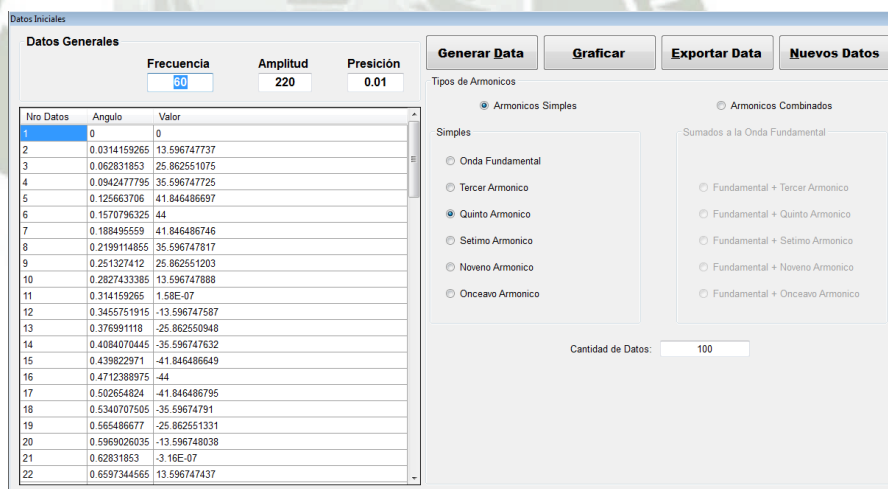


Figura 18: Data generada para el quinto armónico.

Fuente: Elaboración Propia.

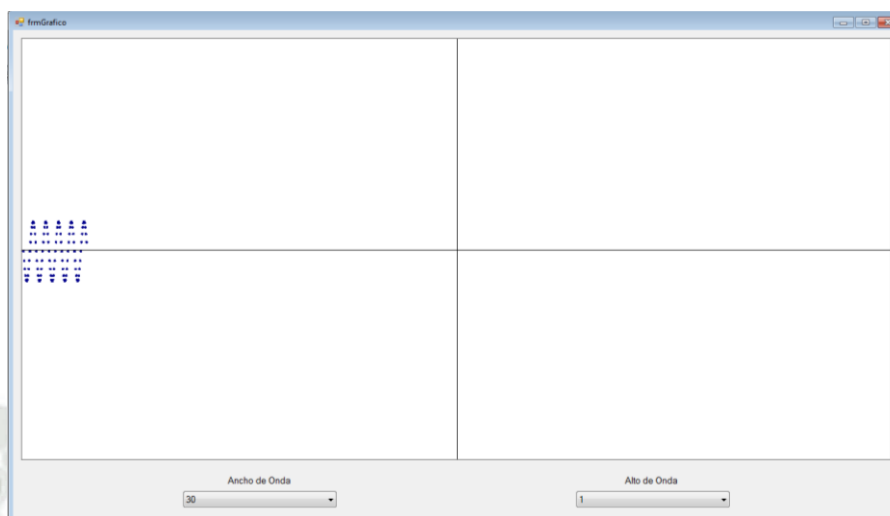


Figura 19: Gráfica del 5to armónico.
Fuente: Elaboración Propia.

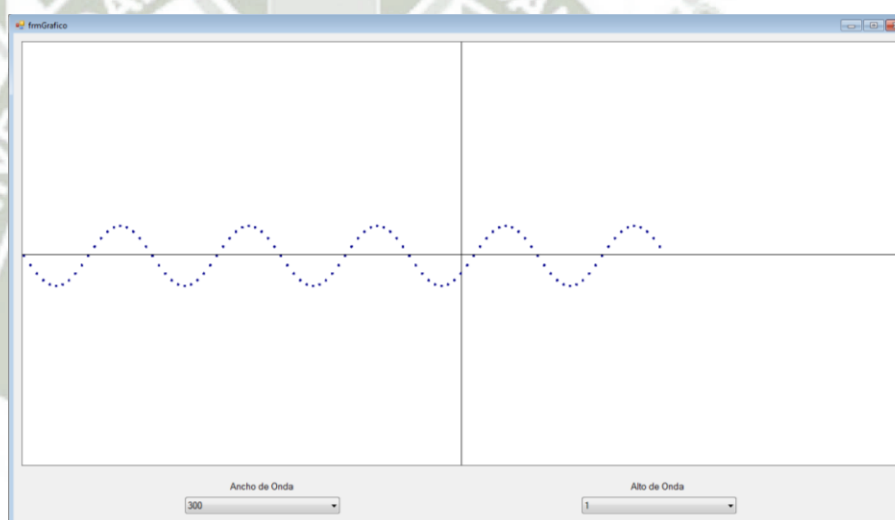


Figura 20: Gráfica del 5to armónico.
Fuente: Elaboración Propia.

- Botón Guardar: este botón almacenara la data generada en dos archivos TXT (archivo de texto) con la data generada, cada archivo contendrá los datos pero en formatos distintos, es decir, uno será almacenado con el uso de coma para la separación del valor entero y los decimales; el otro archivo será almacenado con el uso del punto.

- Estos archivos se almacenarán en una carpeta que generara el programa llamada “Data Armónicos”, estos archivos tendrán como nombres la fecha y hora actual en la que fueron creados y adicionalmente tendrán como parte del nombre de archivo las palabras “PUNTO” o “COMA” para poder diferenciarlos.
- Se tomó esta medida debido a que el Matlab trabaja de acuerdo a la configuración del punto flotante especificado al momento de la instalación del Sistema operativo. A continuación se muestra dichas características.

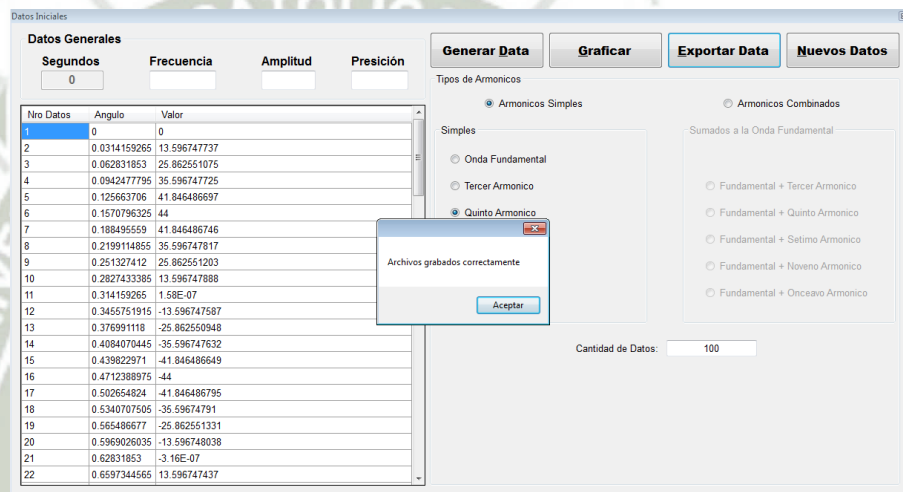


Figura 21: Guardado de archivos.
Fuente: Elaboración Propia.

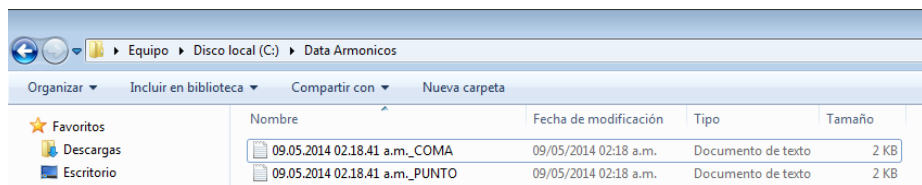


Figura 22: Comprobación de los archivos generados con la data correspondiente.
Fuente: Elaboración Propia.

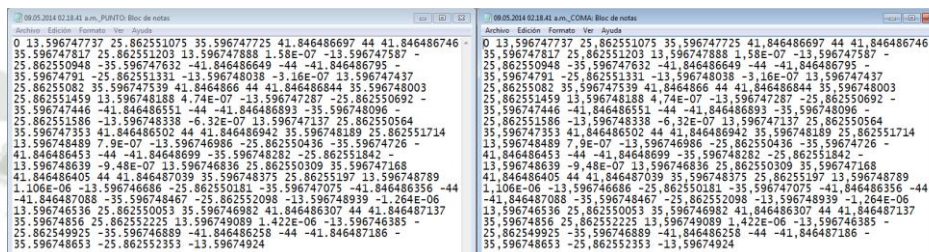


Figura 23: Sección de los archivos generados con los datos y sus diferencias.
Fuente: Elaboración Propia.

2.4. Entrenamiento supervisado.

- **Descripción:** Se decidió usar el entrenamiento supervisado para la Red Neuronal Artificial debido a los armónicos a analizar (3ro, 5to, 7mo y 9no) ya que necesitan una confirmación visual de los resultados obtenidos luego de procesar la data obtenida del generador de señales.

El método de entrenamiento supervisado propiamente para la red, es el método de backpropagation, el cual consiste en presentar un conjunto de pares de datos, representando la entrada, y la salida deseada para esta entrada, para así minimizar el error entre la salida deseada y la actual.

Se utiliza la estructura del perceptrón multicapa, esto ayuda a un mayor análisis y procesamiento de los datos y utilizando el método de entrenamiento backpropagation, descrito anteriormente, se acopla al problema que se quiere resolver.

○ Desarrollo:

En este tipo de RNA se emplea un vector de estímulos de entrada y se predice el valor de una variable de salida. Al mismo tiempo se dispone del valor real de esta salida, de modo que la red se va adaptando a las salidas deseadas a partir de los errores que se van cometiendo entre predicción y salida.

Usaremos el Perceptrón Multicapa, estas redes permiten caracterizar las típicas relaciones entre conjuntos de variables que se dan en condiciones normales de funcionamiento de equipos o procesos. Se utilizan para estimar una salida, que se compara con el valor real de la misma. Las anomalías se detectan cuando se produce una desviación significativa entre el valor real y el estimado.

Al llegar a la neurona [GEP, 2002] todos los datos de las informaciones, ponderados por sus pesos, son sumadas. Esto es lo que se denomina la Regla de Propagación. La fórmula más habitual es la simple suma de todas las entradas ponderadas que toma el nombre de Potencial sináptico. Es decir, el resultado de la regla de propagación elegida es denominado el potencial sináptico.

$$H_i(t) = \sum_{j=1}^N X_j \times W_j$$

Formula del Potencial sináptico

Siendo:

- $H_i(t)$ el potencial sináptico de la neurona i en el momento t .

- X_i la entrada de datos procedentes de la fuente de información j .
- W_j el peso sináptico asociado a la entrada W_j .

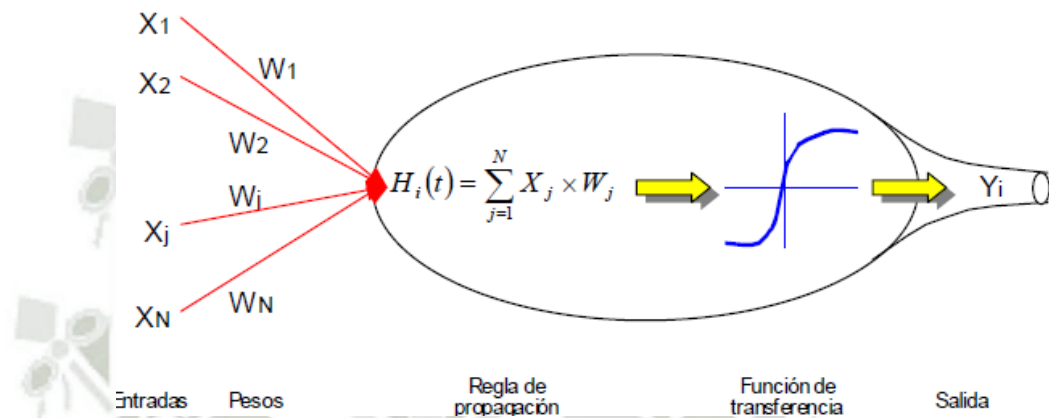


Figura 24: Esquema de una Neurona Artificial.
Fuente: [GEP, 2002]

Cuando el resultado de la regla de propagación supera un cierto número, denominado umbral, entonces la neurona se activa y el número resultante de la regla de propagación se “introduce” en una función denominada Función de activación.

Existen diferentes funciones de activación, tras una serie de pruebas se definió que la mejor función de activación es la Función Lineal.

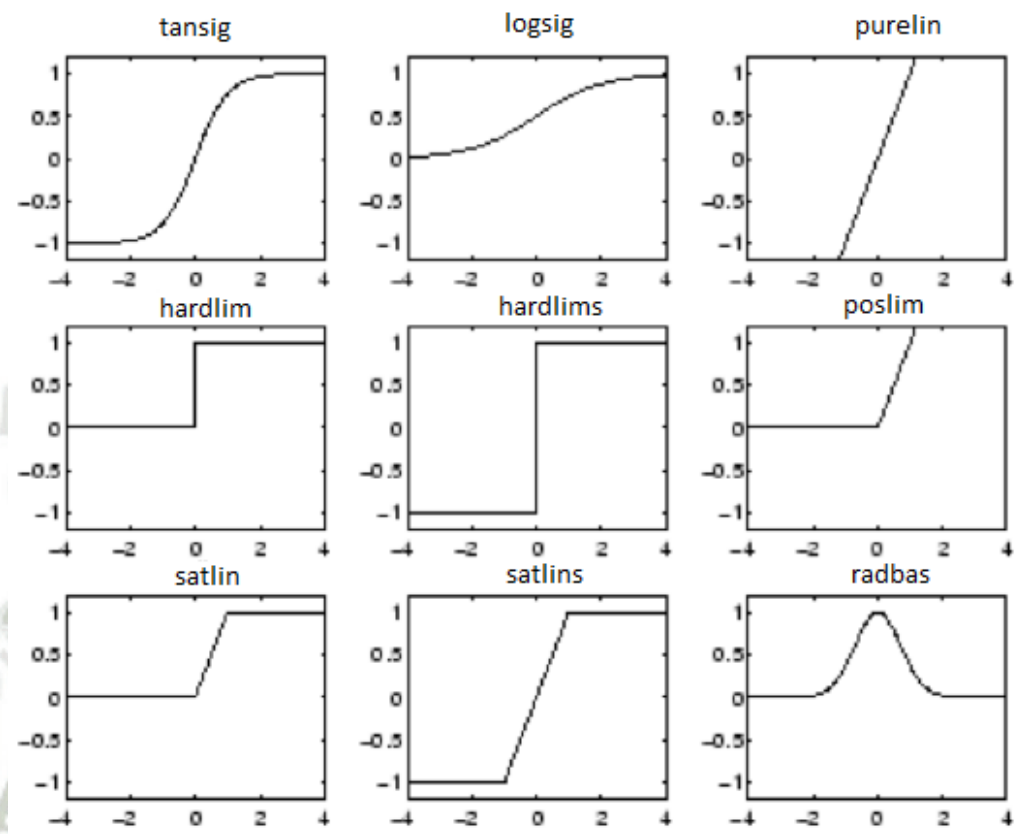


Figura 25: Funciones de Activación.

Fuente: <http://joel-redesneuronalesartificiales.blogspot.com/2008/07/redes-neuronales-artificiales.html>

A continuación se muestra las pruebas con cada una de las funciones para la data de salida.

- **Descripción:** Se utilizó 2000 datos para cada una de las pruebas, hasta hallar la función de activación que mejor se adecuara a nuestro modelo, ya que con una cantidad mayor de datos, no hay diferencia para este punto.

Se optó por utilizar la función purelin o función lineal, ya que los puntos de coincidencia se dan en los picos de la onda, los cuales son los que se pretenden analizar.

- Desarrollo:

La función de Activación Tangente Sigmoidal Hiperbólica (tansig)

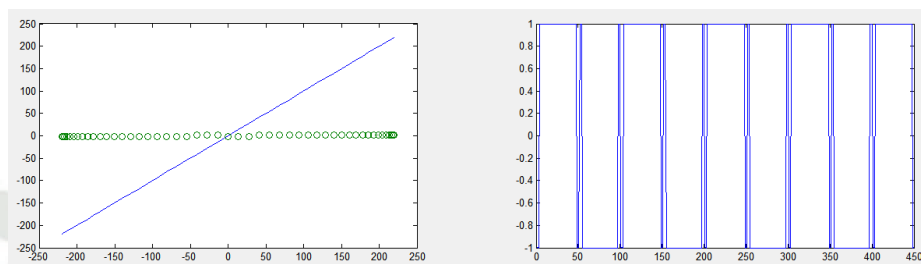


Figura 26: RNA usando la Función de Activación tansig.
Fuente: Elaboración Propia

La función de Activación Sigmoidal Logarítmico (logsig)

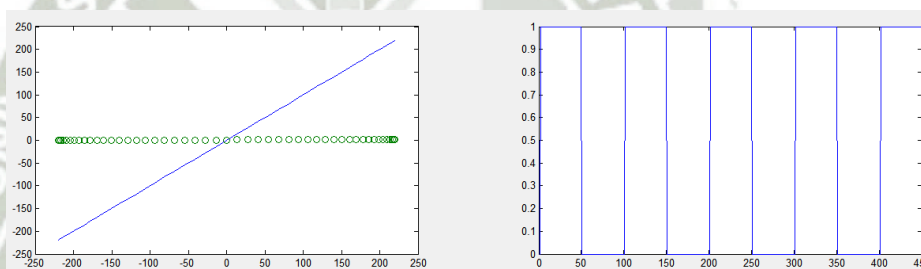


Figura 27: RNA usando la Función de Activación logsig.
Fuente: Elaboración Propia

La función de Activación Lineal (purelin)

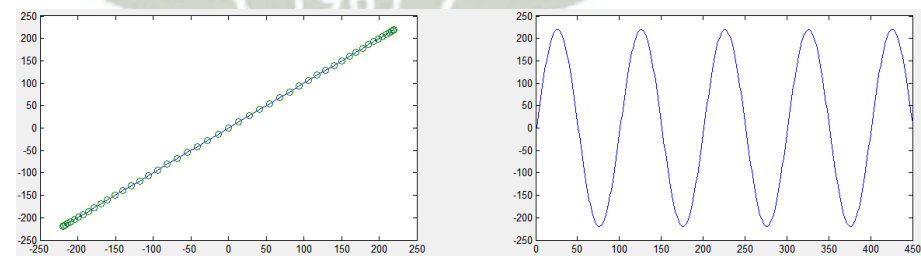


Figura 28: RNA usando la Función de Activación purelin.
Fuente: Elaboración Propia

La función de Activación de Limitador Fuerte (hardlim)

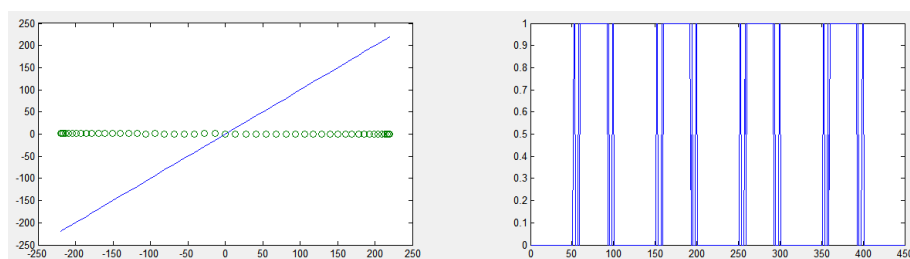


Figura 29: RNA usando la Función de Activación hardlim.
Fuente: Elaboración Propia

La función de Activación de Limitador Fuerte Simétrico (hardlims)

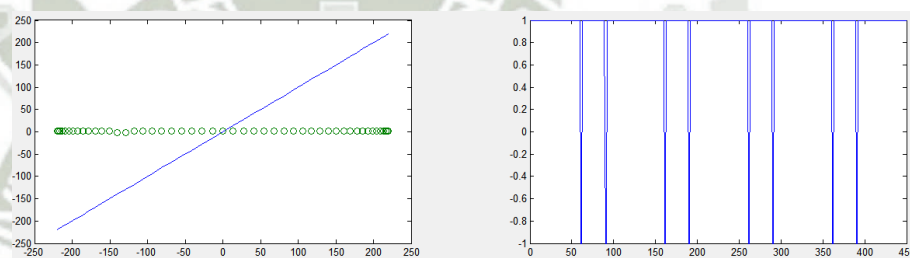


Figura 30: RNA usando la Función de Activación hardlims.
Fuente: Elaboración Propia

La función de Activación Lineal Positiva (poslim)

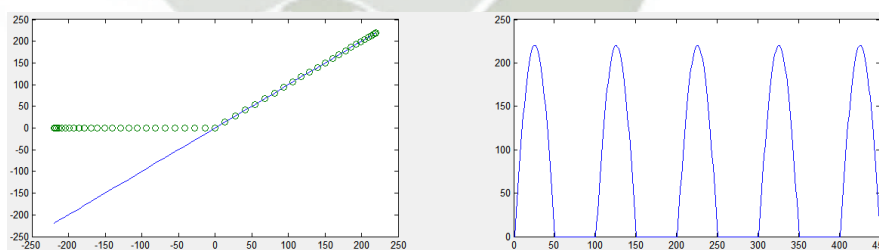


Figura 31: RNA usando la Función de Activación poslim.
Fuente: Elaboración Propia

La función de Activación Lineal Saturado (satlim)

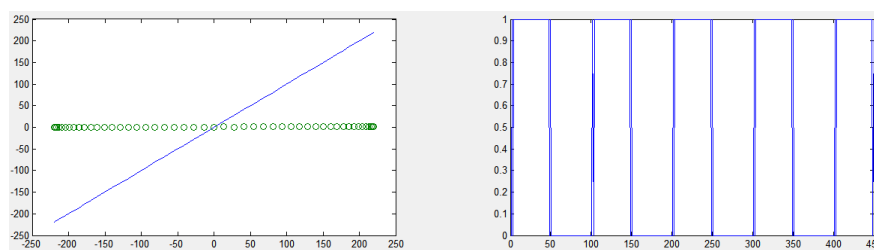


Figura 32: RNA usando la Función de Activación satlim.
Fuente: Elaboración Propia

La función de Activación Lineal Saturado Simétrico (satlims)

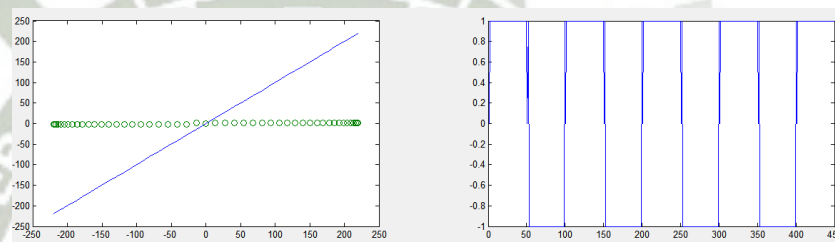


Figura 33: RNA usando la Función de Activación satlims. Fuente: Elaboración Propia

La función de Activación Gaussiana (radbas)

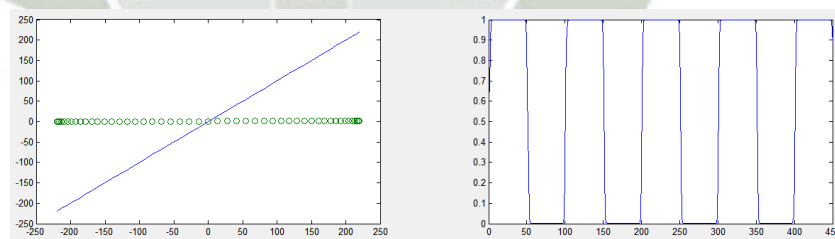


Figura 34: RNA usando la Función de Activación radbas.
Fuente: elaboración Propia

2.5. Perceptrón multicapa.

Una de las arquitecturas más utilizadas en la resolución de problemas. Esto es debido, fundamentalmente, a su capacidad como aproximador universal, así como su fácil uso y aplicabilidad. Estas redes han sido aplicadas con éxito para la resolución de una gran variedad de problemas como clasificación, modelización, control discreto, etc. La arquitectura del perceptrón multicapa se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas está formada por un conjunto de neuronas y se distinguen tres tipos de capas diferentes: la capa de entrada, las capas ocultas y la capa de salida. Las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones que proceden del exterior y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos. Las conexiones entre las neuronas llevan asociado un número real, llamado peso de la conexión.

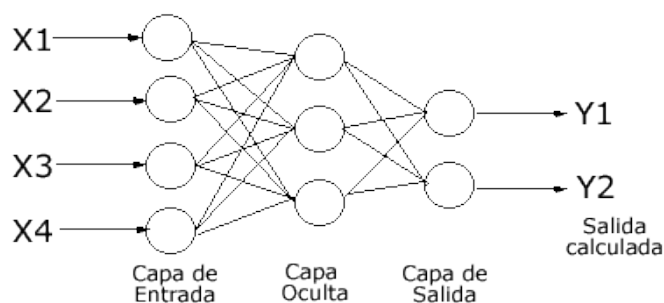


Figura 35: Estructura de un Perceptrón Multicapa.

Fuente: <http://www.cs.us.es/~fsancho/?e=72>

El perceptrón multicapa es de aprendizaje supervisado y utiliza el algoritmo de backpropagation. Este algoritmo está basado en la regla de aprendizaje por corrección de error, considerada como una generalización del algoritmo de los mínimos cuadrados. Su operación consta de dos fases, una directa y una inversa o de retroceso. En la fase directa, se traduce el patrón de actividad en la capa de entrada de la red (vector de entrada), que recorre todas las capas siguientes. Se obtiene la respuesta predicha de la red en la capa de salida. En esta fase los pesos sinápticos de la red permanecen fijos. En la fase inversa, los pesos sinápticos son ajustados de acuerdo con la regla de corrección del error cometido, entre la respuesta predicha y real. Esta regla, conocida como método de Levenberg-Marquardt, minimiza el cuadrado de las diferencias entre la respuesta o salida deseada y la salida real de la red.

Las pruebas realizadas mostraron que al usar una Red Neuronal Artificial multicapa, genera un error, y no es factible utilizarla para este caso, pero si es factible utilizar el entrenamiento con una sola capa.

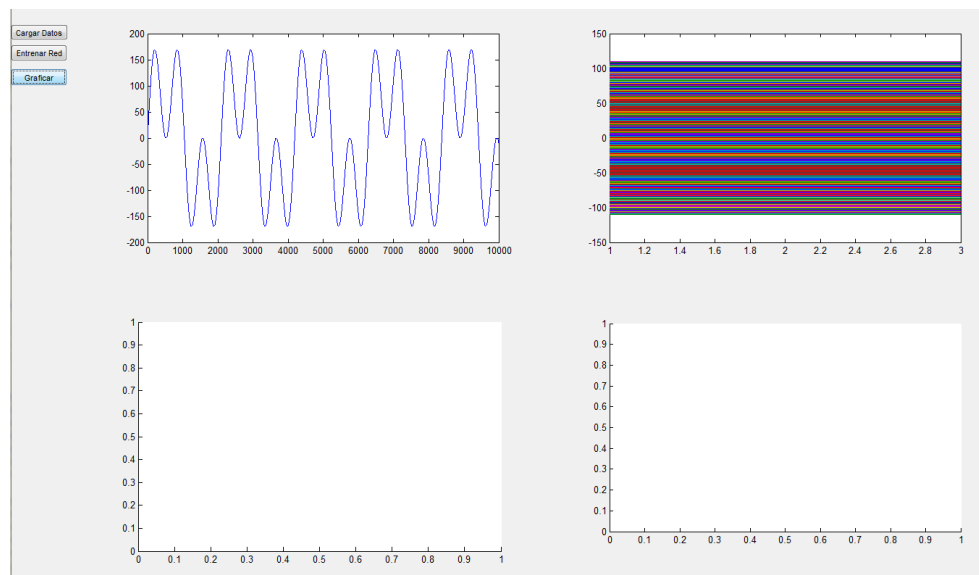


Figura 36: Error al utilizar la RNA con más de una capa.
Fuente: elaboración Propia

La RNA al tener mayor cantidad de capas posee un filtrado mejor de la data, se probó que con 30 neuronas en adelante no muestra un cambio significativo en el resultado, por lo que se decidió emplear esta cantidad de neuronas para las pruebas.

La primera prueba se realizó con 10 neuronas, y mostro un cambio en la gráfica pero con ciertas irregularidades, que podría confundir al momento de la lectura para detectar los armónicos deseados.

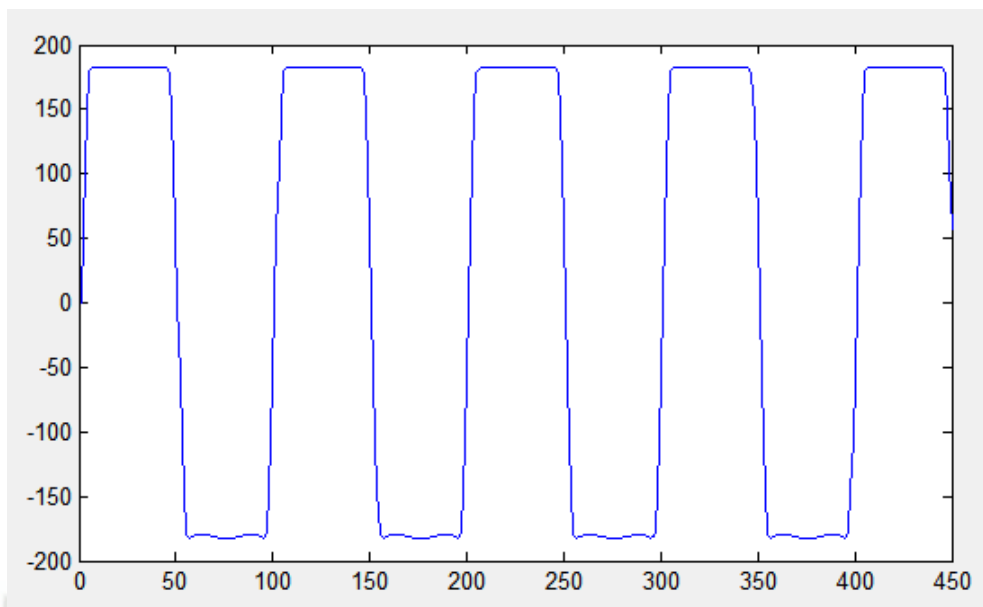


Figura 37: Prueba realizada con 10 neuronas.
Fuente: elaboración Propia

La segunda prueba se realizó con 20 neuronas, y se puede notar que la señal es más limpia pero aun presenta ciertas irregularidades en la señal.

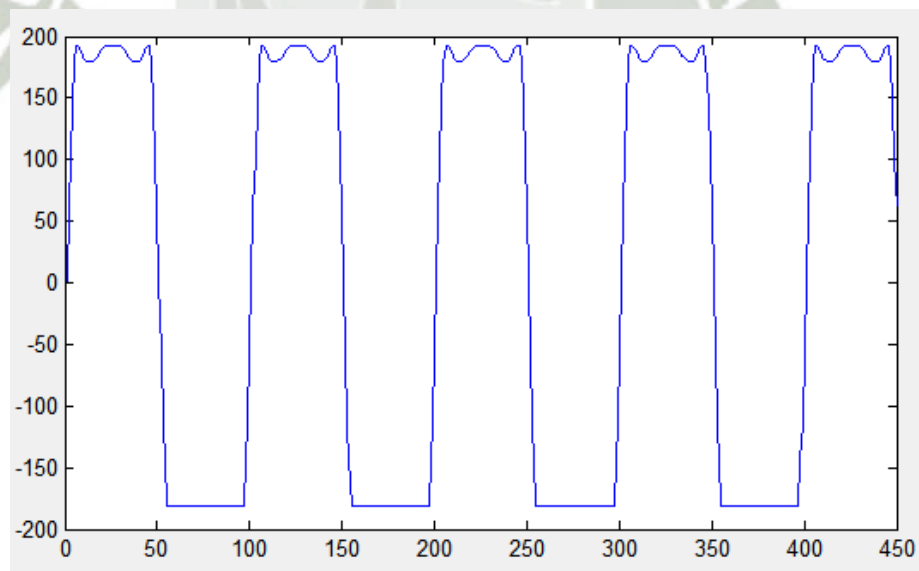


Figura 38: Prueba realizada con 20 neuronas.
Fuente: elaboración Propia

La tercera prueba se realizó con 30 neuronas y la señal se mostró más limpia, permitiendo una mejor lectura de la señal.

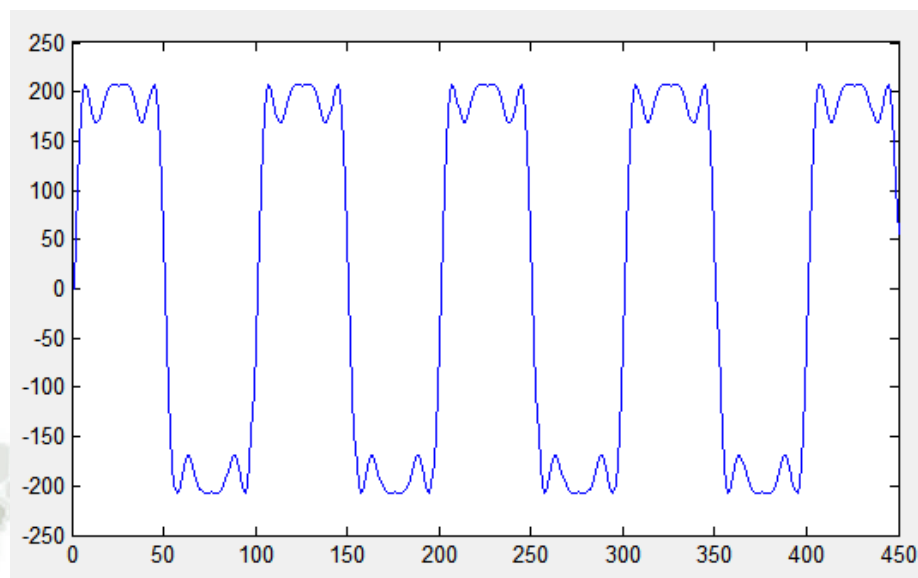


Figura 39: Prueba realizada con 30 neuronas.
Fuente: Elaboración propia

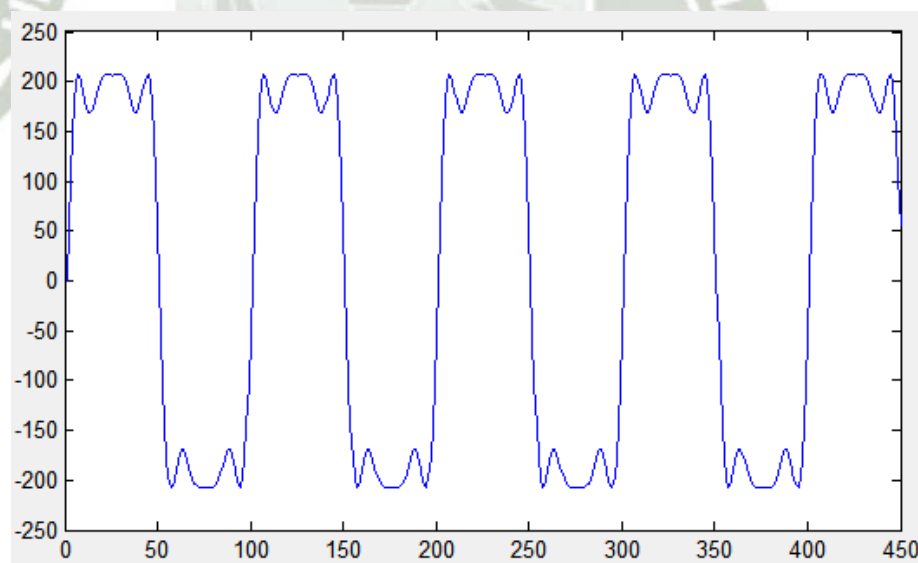


Figura 40: Prueba realizada con 40 neuronas, no muestra un cambio significativo.
Fuente: Elaboración propia

El entrenamiento de la red se realizó con 300 iteraciones, por las investigaciones realizadas se encontró que no hay una fórmula para hallar el número de iteraciones necesarias para poder entrenar la red, y que esto se determina mediante pruebas de ensayo y error.

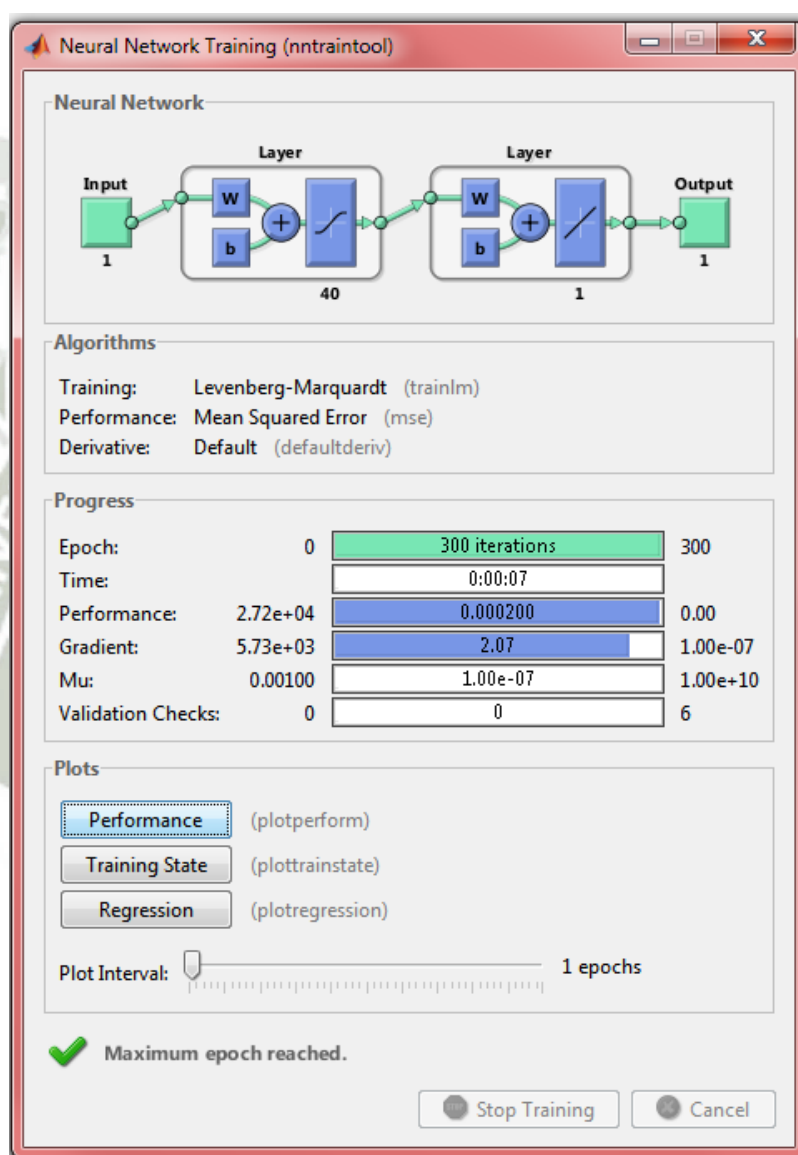


Figura 41: Entrenamiento con 300 iteraciones.
Fuente: Elaboración propia

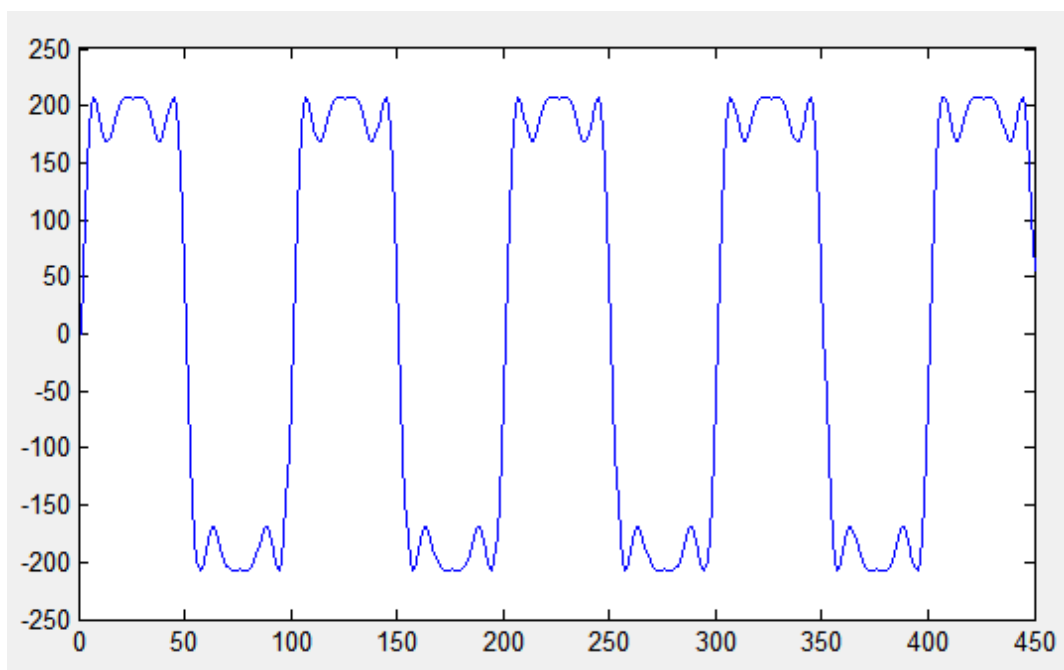


Figura 42: Señal procesada con 300 iteraciones.
Fuente: Elaboración propia

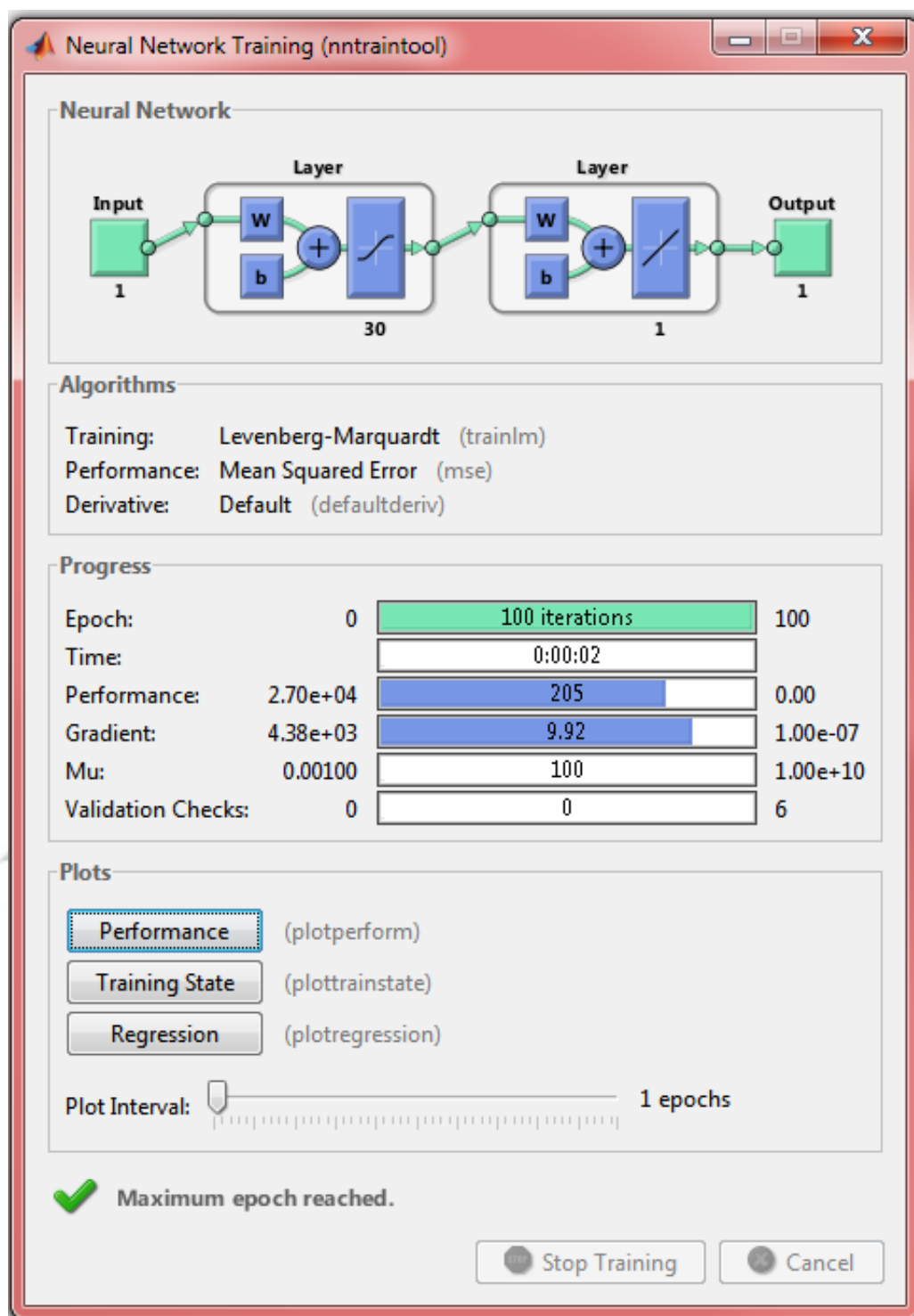


Figura 43: Entrenamiento con 100 iteraciones.
Fuente: Elaboración propia

Con 100 iteraciones la señal procesada presenta una ligera irregularidad en la señal, para una mejor lectura, se decidió usar 300 iteraciones.

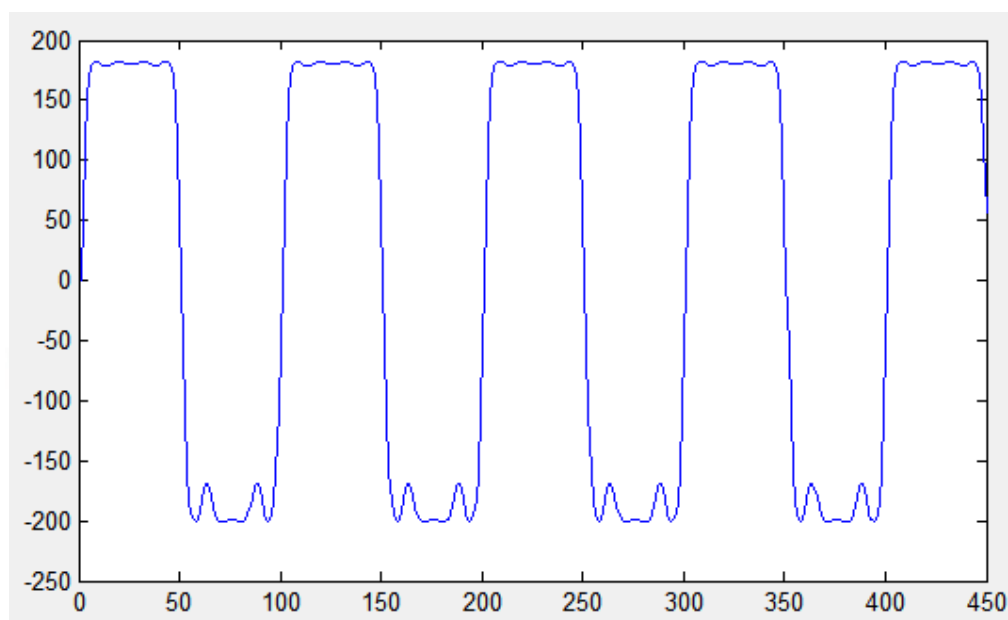


Figura 44: Señal procesada con 100 iteraciones.
Fuente: Elaboración propia

A continuación se presenta el diseño de pantallas de dicho software detallando su funcionamiento.

En la pantalla principal se da click en el botón “Cargar Datos”

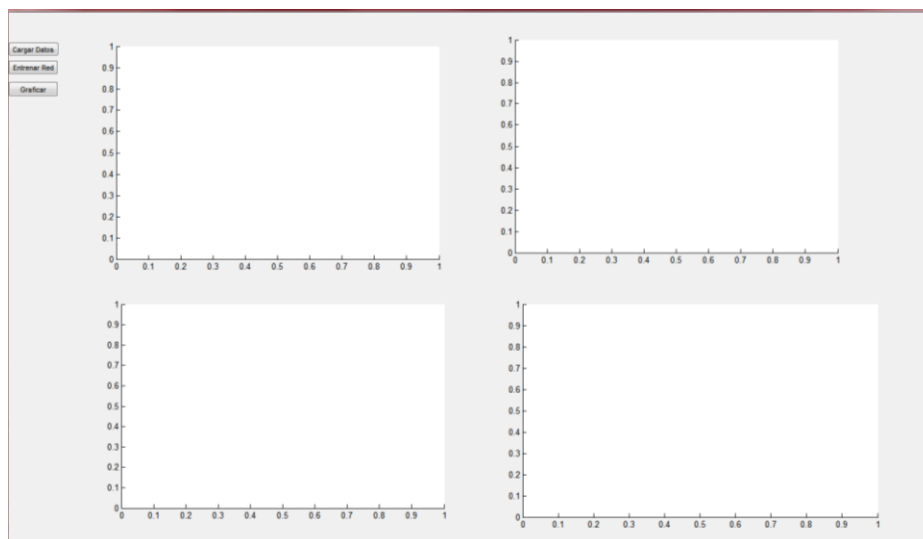
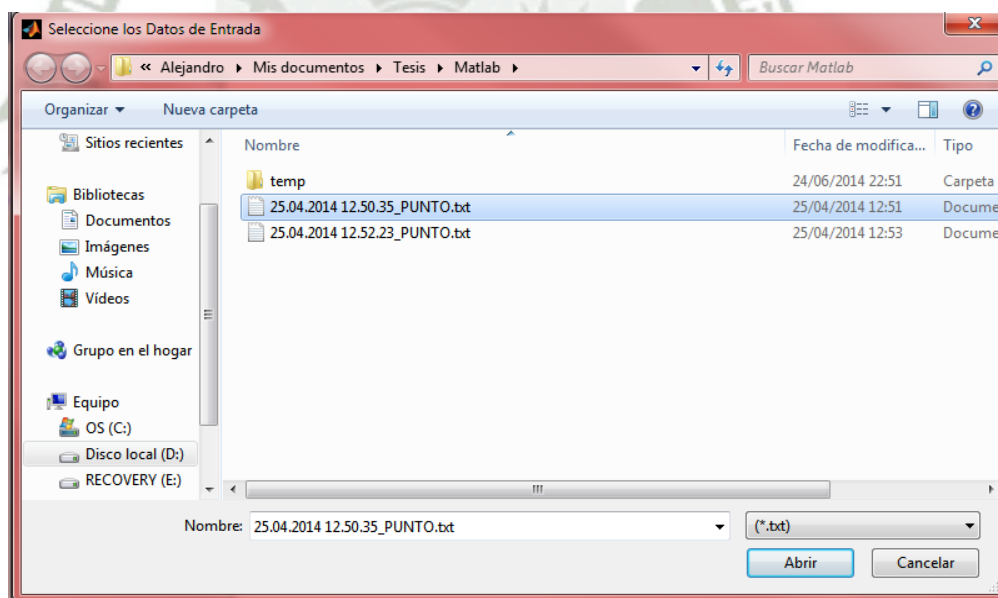


Figura 45: Pantalla Principal.
Fuente: Elaboración propia

Al dar click en el botón “Cargar Datos” se abre una ventana para la carga de la señal fundamental (Figura 46 superior) y luego para la del Armónico (Figura 46 inferior)



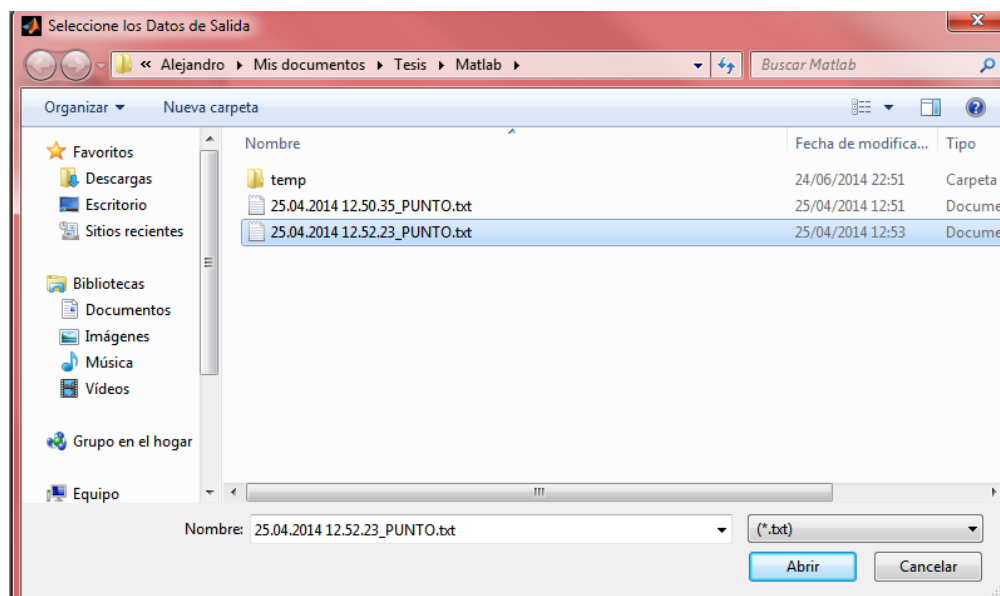


Figura 46: Pantalla de carga de datos.
Fuente: Elaboración propia

Una vez cargado los dos archivos correspondientes, se pueden visualizar gráficamente como son. Y se procede a dar click en el botón “Entrenar Red”

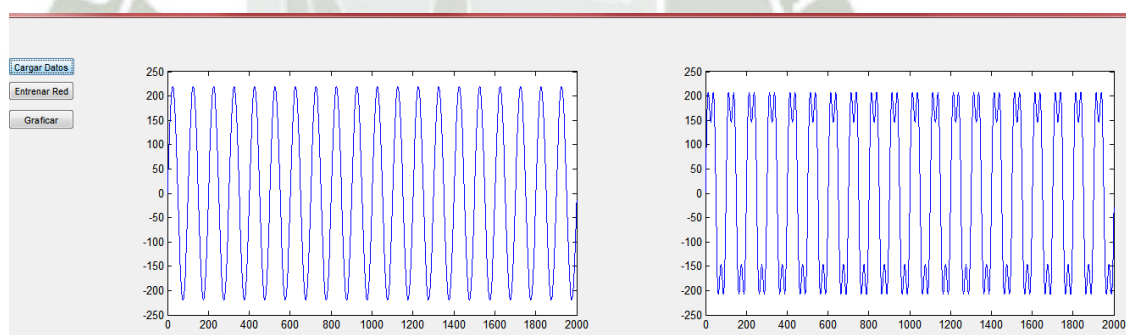


Figura 47: Pantalla de señal fundamental y armónico.
Fuente: Elaboración propia

El entrenamiento de la red se hace de manera automática, por el Matlab, realiza 300 iteraciones de entrenamiento.

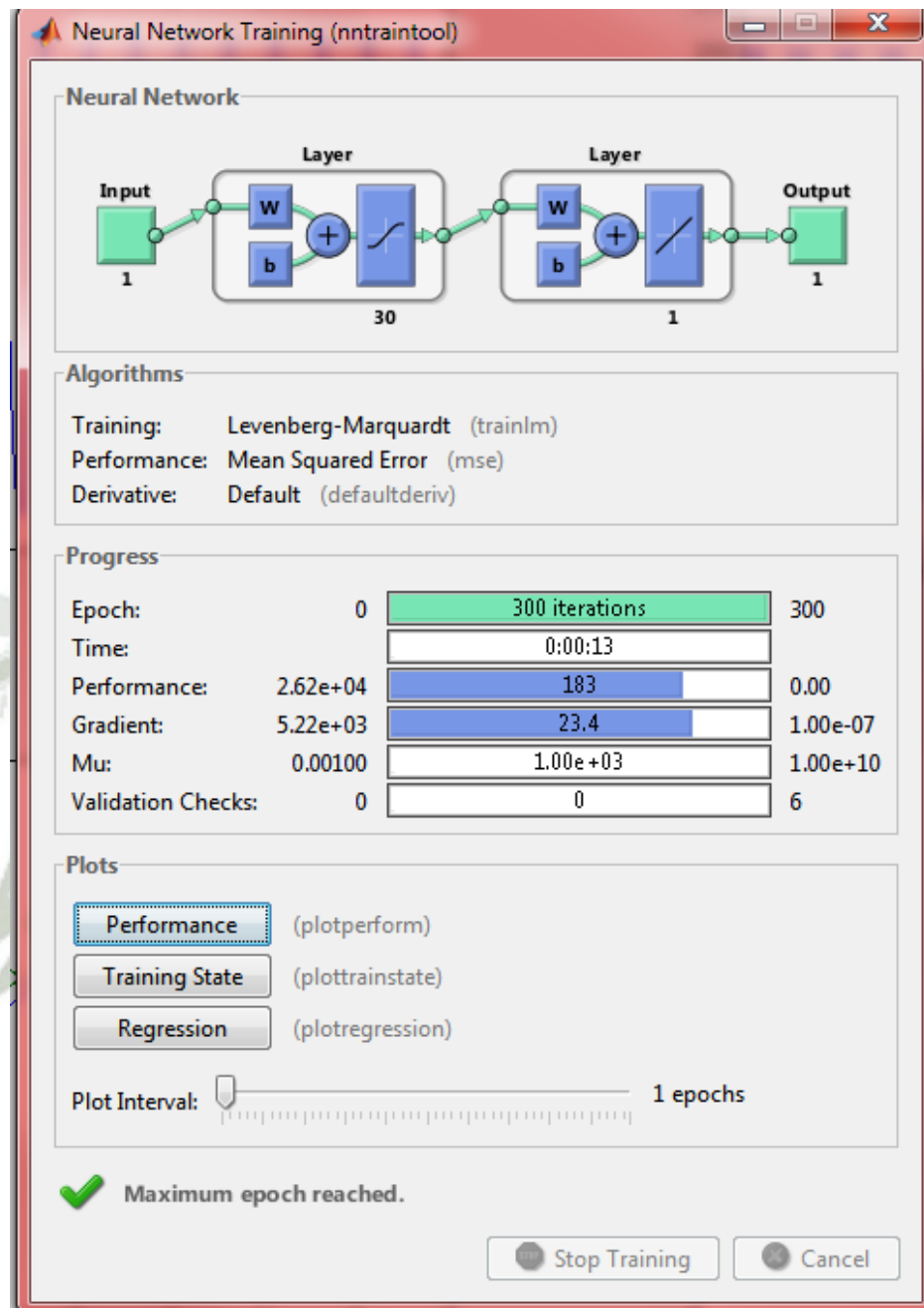


Figura 48: Pantalla de entrenamiento de la RNA.
Fuente: Elaboración propia

Una vez terminado el entrenamiento regresamos a la pantalla principal, donde podemos ver, como la onda de tercer armónico (para este ejemplo que se usó) se

sobrepone a la onda de la señal fundamental. Luego procedemos a dar click en el botón “Graficar”.

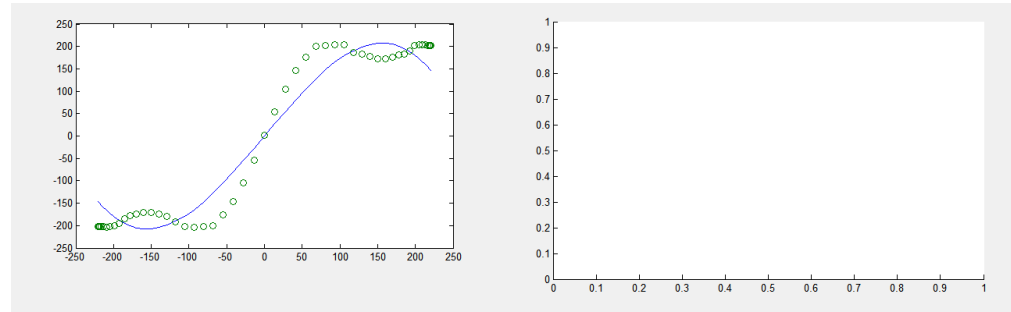


Figura 49: Pantalla luego del entrenamiento de la RNA.
Fuente: Elaboración propia

Al graficar podemos ver que se obtiene un resultado, cuando hay muchos datos, la imagen no se logra distinguir bien, por lo que podemos poner el cursor del mouse en la gráfica para hacer zoom y visualizar de una mejor manera.

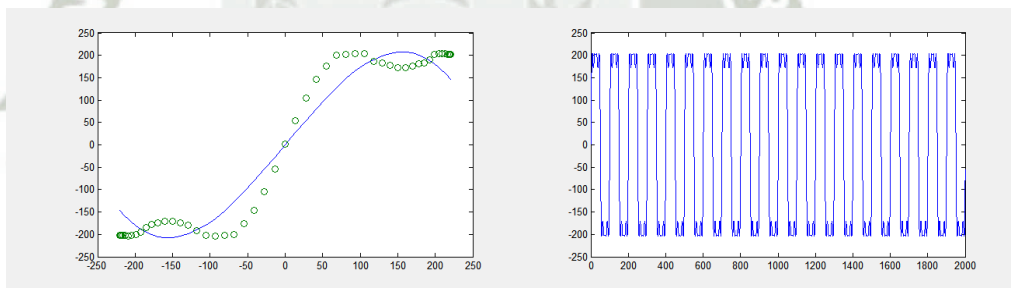


Figura 50: Pantalla al graficar.
Fuente: Elaboración propia

Al hacer zoom a la gráfica donde se cargó los datos del armónico, y de graficar la red una vez entrenada, podemos diferenciar, de manera visual las diferencias que hay, y esto es diferente para los distintos tipos de armónicos.

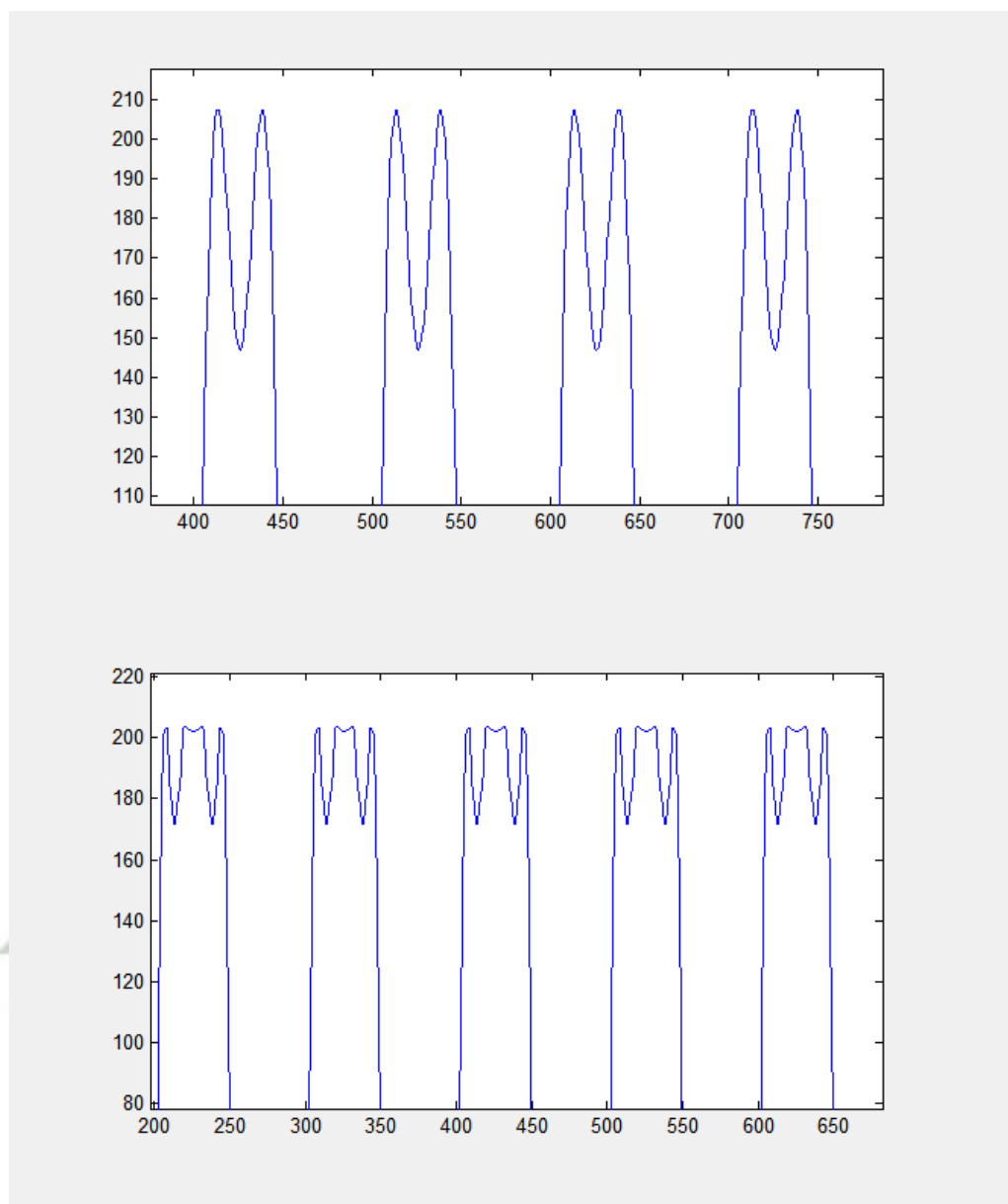


Figura 51: Pantalla al graficar.
Fuente: Elaboración propia

CAPÍTULO IV: RESULTADOS

1. Generador de señales.

Los resultados fueron corroborados por los ingenieros eléctricos y de sistemas expertos del proyecto Swap Smart.

Se realizaron pruebas de señal generada por dicho generador, teniendo como resultado lo siguiente:

1.1. Señal generada.

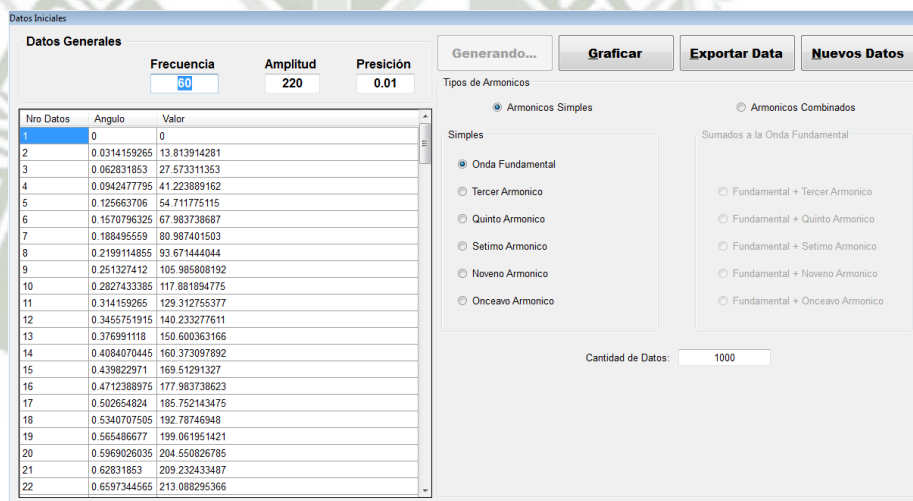
Las primeras pruebas se realizaron para mostrar las señales armónicas solas sin la suma de la señal fundamental, después de comprobar las gráficas se procedió a generar datos para la suma de la señal fundamental con sus respectivos armónicos, se generó esta señal con una cantidad de 1000 datos, se presenta a continuación el proceso de creación de dicha señal y las señales combinadas.

La amplitud de todas las señales en las gráficas fueron en base a 220 voltios de acuerdo a la formula general de corriente eléctrica.

a. Señal fundamental.

En la Figura 52 se observa la generación de datos para la señal fundamental que luego servirá para poder graficar dichos datos y mostrarse como esta en la Figura 53, aquí se observan los puntos generados en forma de señal senoidal ideal, es decir, se observa la señal de corriente eléctrica en su forma original.

Esta será la base para luego poder sumar a esta señal, los diferentes tipos de armónicos y ver sus diferencias en forma gráfica ya que se mostrará la imagen del armónico y la imagen del armónico sumado a la señal fundamental.



Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.813914281
3	0.062831853	27.573311353
4	0.0942477795	41.223889162
5	0.125663706	54.711775115
6	0.1570796325	67.983738687
7	0.188495559	80.987401503
8	0.2199114855	93.671444044
9	0.251327412	105.985808192
10	0.2827433385	117.881894775
11	0.314159265	129.312755377
12	0.3455751915	140.23277611
13	0.376991118	150.600363166
14	0.4084070445	160.373097892
15	0.439822971	169.51291327
16	0.4712388975	177.983738623
17	0.502654824	185.752143475
18	0.5340707505	192.78746948
19	0.565486677	199.061951421
20	0.5969026035	204.550826785
21	0.62831853	209.232433487
22	0.6597344565	213.088295366

Figura 52: Generando datos con valores por defecto de señal armónica fundamental.
Fuente: Elaboración Propia

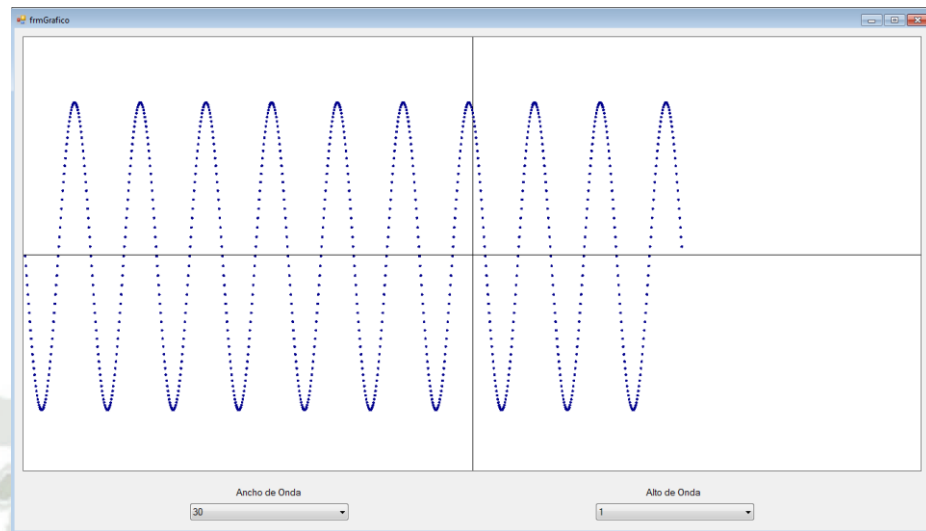


Figura 53: Gráfica de la señal fundamental generada con los datos obtenidos anteriormente.
Fuente: Elaboración propia

b. 3° armónico.

La Figura 54 muestra la generación de datos para el tercer armónico, en la Figura 55 se muestra la gráfica del tercer armónico, se puede ver la diferencia en el ancho de la onda, esto se debe a que la fórmula original varia ya que a la amplitud (220v) se le multiplica $\frac{1}{3}$, por ese motivo la Figura 55 muestra una gráfica pequeña en comparación con la Figura 53 que es la señal fundamental.

En la figura 56 se genera la sumatoria del tercer armónico mas la señal fundamental, como se puede observar, la amplitud de la Figura 56 es igual que el de la señal fundamental, solo varia la cresta de la onda, adaptándose a la forma del armónico hallado anteriormente.

Datos Iniciales

Datos Generales

Frecuencia: Amplitud: Presición:

Generando... Graficar Exportar Data Nuevos Datos

Tipos de Armonicos

☒ Armonicos Simples ☐ Armonicos Combinados

Sumados a la Onda Fundamental

Simple

☐ Onda Fundamental

☒ Tercer Armonico

☐ Quinto Armonico

☐ Setimo Armonico

☐ Noveno Armonico

☐ Onceavo Armonico

☐ Fundamental + Tercer Armonico

☐ Fundamental + Quinto Armonico

☐ Fundamental + Setimo Armonico

☐ Fundamental + Noveno Armonico

☐ Fundamental + Onceavo Armonico

Cantidad de Datos:

Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.741296387
3	0.062831853	26.995800501
4	0.0942477795	39.293964925
5	0.125663706	50.200121055
6	0.1570796325	59.327912874
7	0.188495559	66.353983807
8	0.2199114855	71.029431789
9	0.251327412	73.188626743
10	0.2827433385	72.755078114
11	0.314159265	69.744144577
12	0.3455751915	64.262489954
13	0.376991118	56.504304591
14	0.4084070445	46.744426073
15	0.439822971	35.328602961
16	0.4712388975	22.661246479
17	0.502654824	9.191104045
18	0.5340707505	-4.604637831
19	0.565486677	-18.237258117
20	0.5969026035	-31.223814443
21	0.62831853	-43.104251579
22	0.6597344565	-53.457699117

Figura 54: Generando datos para el armónico de orden tres.
Fuente: Elaboración propia

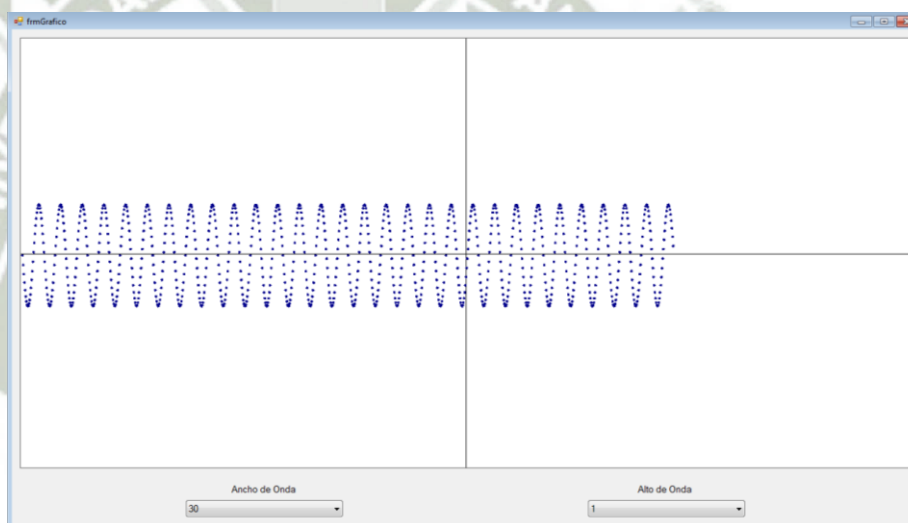


Figura 55: Gráfica del armónico de orden tres generada con los datos obtenidos anteriormente.
Fuente: Elaboración propia

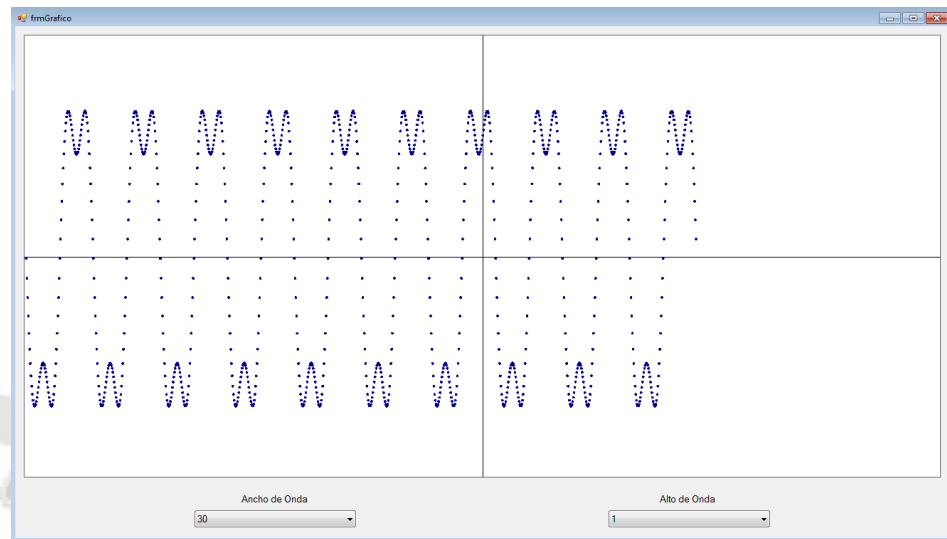


Figura 56: Gráfica del armónico de orden tres sumado con la señal fundamental generado con los datos obtenidos anteriormente. Fuente: Elaboración propia

c. 5° armónico.

Como en el punto anterior, la Figura 57 muestra la generación de datos para el quinto armónico y en la Figura 58 se puede observar su grafica de acuerdo a los datos hallados anteriormente.

En la Figura 58 se puede observar que la amplitud de la imagen es aún más pequeña que la generada con el tercer armónico, de la misma forma, esto sucede ya que se multiplica el valor de amplitud y velocidad angular por $1/5$, esto hace que la amplitud se reduzca más.

La Figura 59 muestra la sumatoria de la señal fundamental con el quinto armónico, se puede observar que al igual que el punto anterior, la imagen conserva su amplitud original (220v) pero las crestas se acomodan de acuerdo a la imagen generada del quinto armónico.

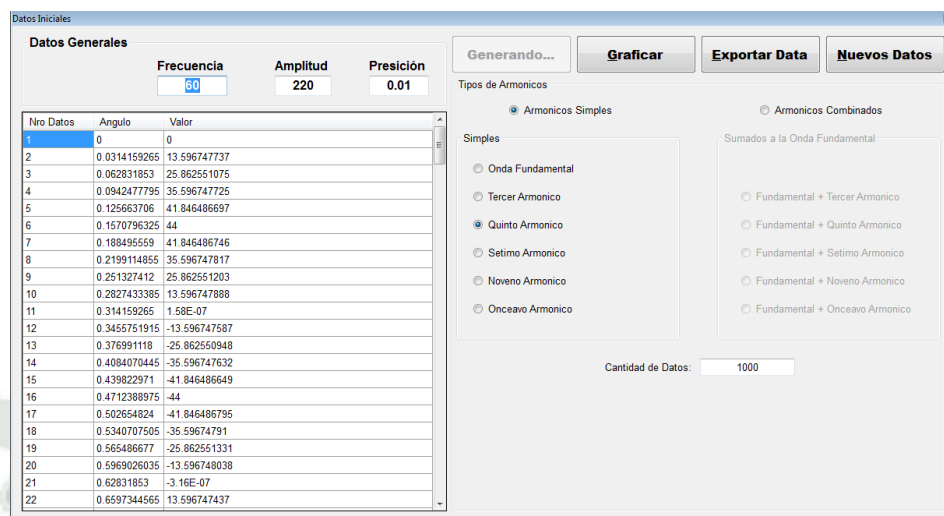


Figura 57: Generando datos para el armónico de orden cinco.
Fuente: Elaboración propia

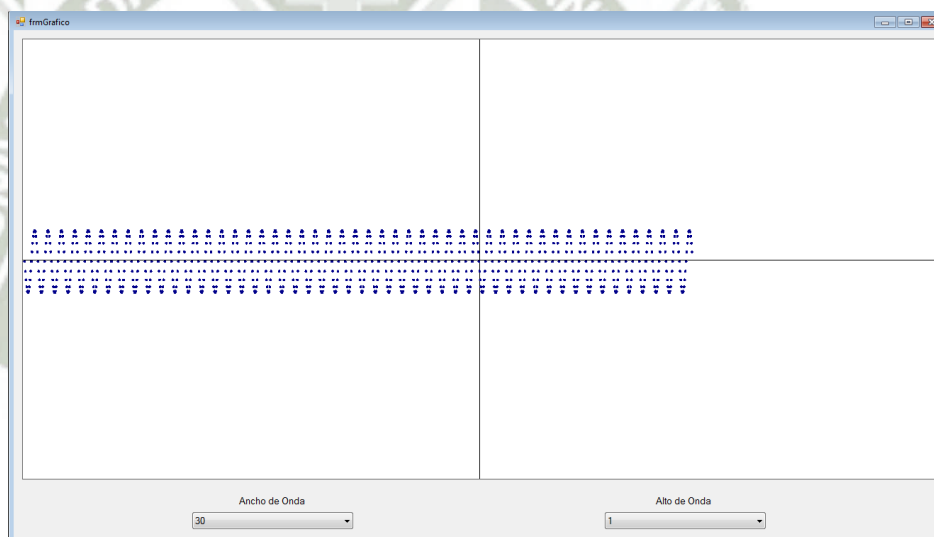


Figura 58; Gráfica del armónico de orden cinco generado con los datos obtenidos anteriormente.
Fuente: Elaboración propia



Figura 59: Gráfica del armónico de orden cinco sumado con la señal fundamental generado con los datos obtenidos anteriormente.
Fuente: Elaboración propia

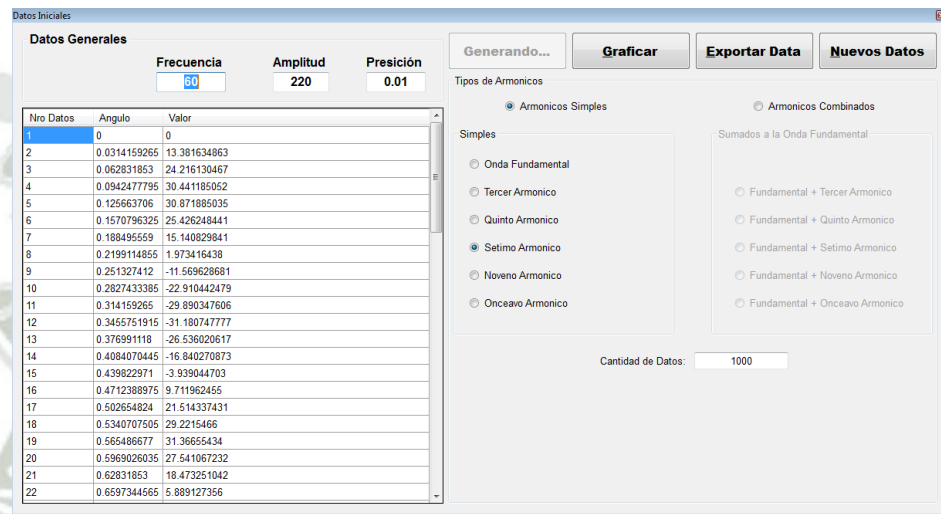
d. 7° armónico

La Figura 60 muestra la generación de datos para el séptimo armónico y en la Figura 61 se puede observar su grafica de acuerdo a los datos hallados anteriormente.

En la Figura 61, como ya se observó anteriormente la amplitud de la imagen es aún más pequeña que la generada con el quinto armónico, esto es porque se multiplica el valor de amplitud y velocidad angular por $1/7$, esto hace que la amplitud reduzca su tamaño en una séptima parte.

Como la imagen está muy pequeña, se hace uso de las opciones incluidas en la ventana de la gráfica, en la Figura 62 se muestra la imagen pero mostrando a más detalle la onda generada, mantiene su amplitud pero su ancho varia, esto lo hacemos para distinguir mejor la onda y hacer las comparaciones.

La Figura 63, como en los puntos anteriores, muestra la sumatoria de la señal fundamental con el séptimo armónico, se puede observar que al igual que el punto anterior, la imagen conserva su amplitud original (220v) pero las crestas se acomodan de acuerdo a la imagen generada del quinto armónico.



Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.381634863
3	0.062831853	24.216130467
4	0.0942477795	30.441185052
5	0.125663706	30.871885035
6	0.1570796325	25.426248441
7	0.188495559	15.140829841
8	0.2199114855	1.973416438
9	0.251327412	-11.569628681
10	0.2827433385	-22.910442479
11	0.314159265	-29.890347606
12	0.3455751915	-31.180747777
13	0.376991118	-26.536020617
14	0.4084070445	-16.840270873
15	0.439822971	-3.939044703
16	0.4712388975	9.711962455
17	0.502654824	21.514337431
18	0.5340707505	29.2215466
19	0.565486677	31.36655434
20	0.5969026035	27.541067232
21	0.62831853	18.473251042
22	0.6597344565	5.889127356

Figura 60: Generando datos para el armónico de orden siete.
Fuente: Elaboración propia

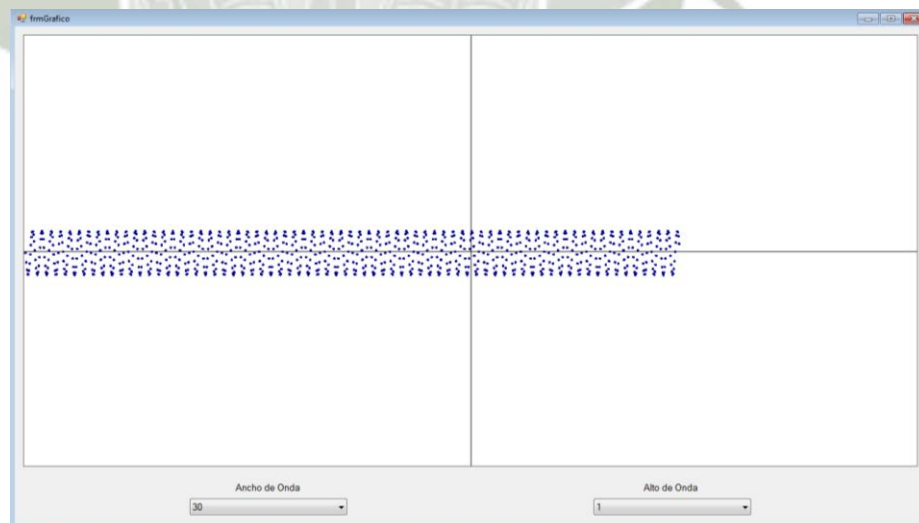


Figura 61: Gráfica del armónico de orden siete generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda.
Fuente: Elaboración propia

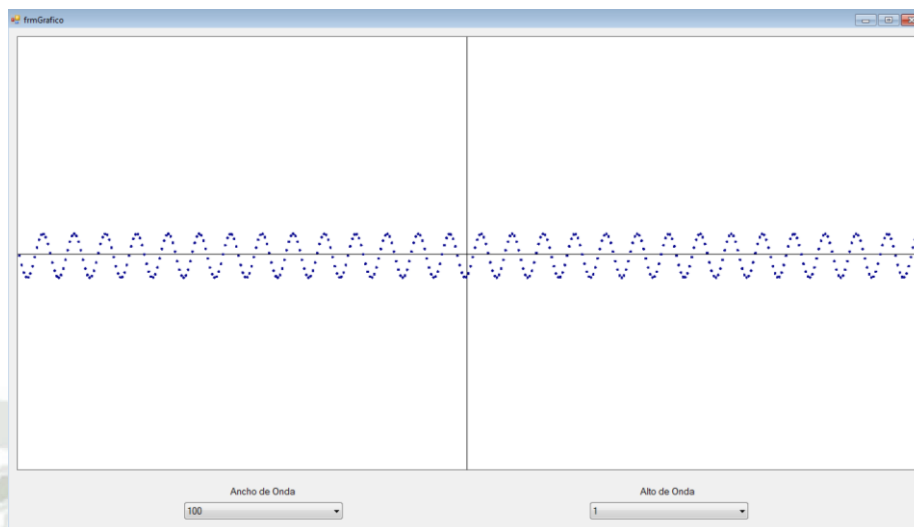


Figura 62: Gráfica del armónico de orden siete generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda.
Fuente: Elaboración propia

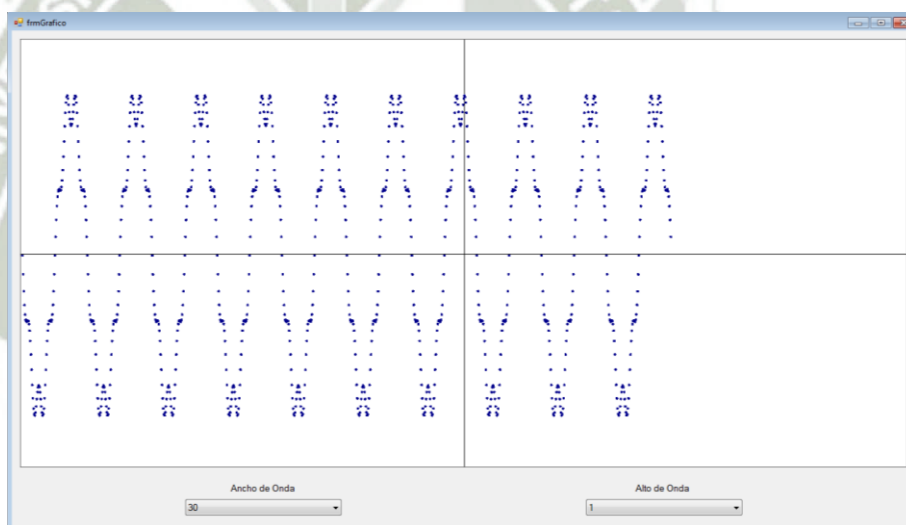


Figura 63: Gráfica del armónico de orden siete sumado con la señal fundamental generado con los datos obtenidos anteriormente.
Fuente: Elaboración propia

e. 9° armónico

La Figura 64 muestra la generación de datos para el noveno armónico y en la Figura 65 se puede observar su grafica de acuerdo a los datos hallados anteriormente.

En la Figura 66, como se mencionó anteriormente la amplitud de la imagen es aún más pequeña que la generada con el séptimo armónico, esto es porque se multiplica el valor de amplitud y velocidad angular por $1/9$, esto hace que la amplitud reduzca nueve veces su tamaño.

Como en el punto anterior, la imagen está muy pequeña, se hace uso de las opciones incluidas en la ventana de la gráfica, en la Figura 66 se muestra la imagen pero mostrando a más detalle la onda generada, mantiene su amplitud pero su ancho varia, esto lo hacemos para distinguir mejor la onda y hacer las comparaciones.

La Figura 67, como en los puntos anteriores, muestra la sumatoria de la señal fundamental con el armónico de orden nueve, se puede observar que al igual que el punto anterior, la imagen conserva su amplitud original (220v) pero las crestas se acomodan de acuerdo a la imagen generada del armónico de orden nueve.

Si se comparan las imágenes de la Figura 63 y la Figura 67, se ve cierta similitud, esto sucede porque, a más pequeña la imagen de un armónico de un orden mayor, las diferencias serán menores.

Datos Iniciales

Datos Generales

Frecuencia: Amplitud: Presición:

Generando... Graficar Exportar Data Nuevos Datos

Tipos de Armonicos

☒ Armonicos Simples ☐ Armonicos Combinados

Simples

☐ Onda Fundamental

☐ Tercer Armonico

☐ Quinto Armonico

☐ Setimo Armonico

☒ Noveno Armonico

☐ Onceavo Armonico

Sumados a la Onda Fundamental

☐ Fundamental + Tercer Armonico

☐ Fundamental + Quinto Armonico

☐ Fundamental + Setimo Armonico

☐ Fundamental + Noveno Armonico

☐ Fundamental + Onceavo Armonico

Cantidad de Datos:

Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	13.097988308
3	0.062831853	22.117994602
4	0.0942477795	24.251692705
5	0.125663706	18.834768197
6	0.1570796325	7.553748826
7	0.188495559	-6.079086039
8	0.2199114855	-17.819233039
9	0.251327412	-24.011466105
10	0.2827433385	-22.727869707
11	0.314159265	-14.368084073
12	0.3455751915	-1.53487954
13	0.376991118	11.776200756
14	0.4084070445	21.420829858
15	0.439822971	24.396208931
16	0.4712388975	19.775971113
17	0.502654824	8.998600412
18	0.5340707505	-4.580431871
19	0.565486677	-16.73373493
20	0.5969026035	-23.676477197
21	0.62831853	-23.248048274
22	0.6597344565	-15.581475561

Figura 64: Generando datos para el armónico de orden nueve.
Fuente: Elaboración propia

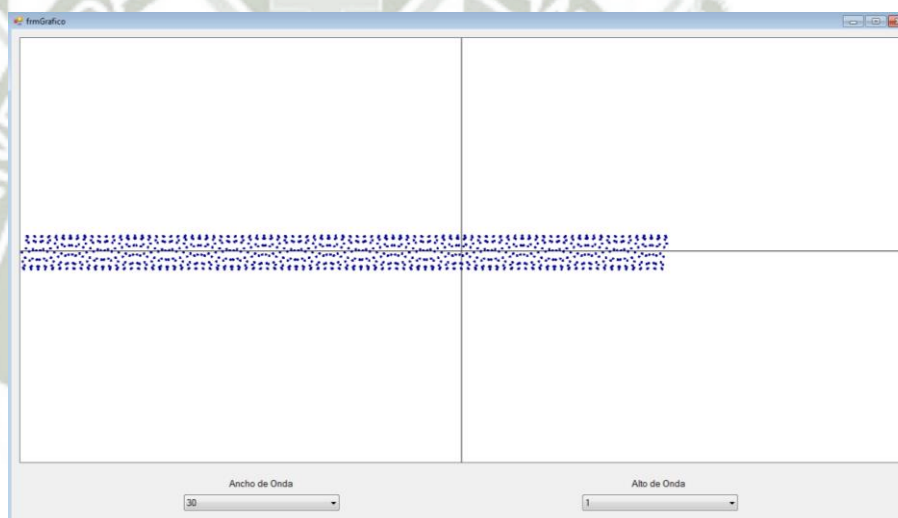


Figura 65: Gráfica del armónico de orden nueve generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda.
Fuente: Elaboración propia

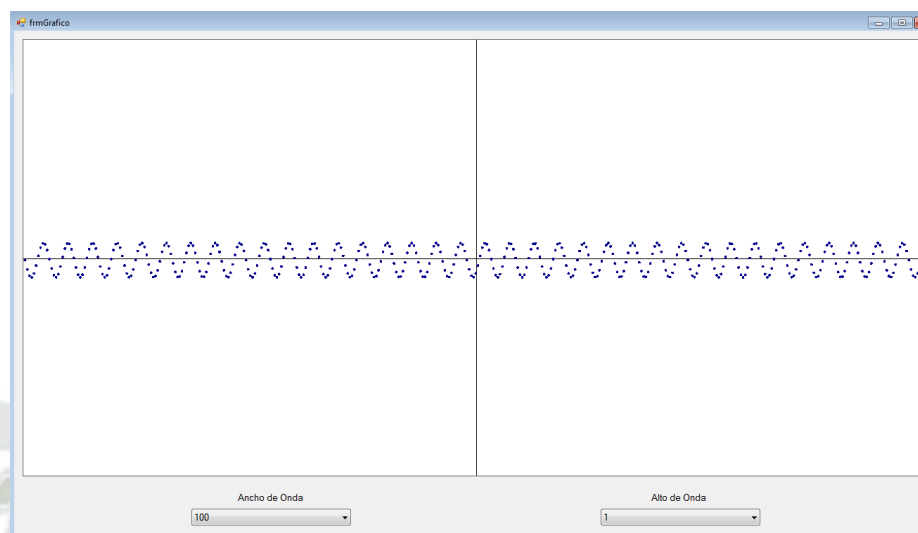


Figura 66: Gráfica del armónico de orden nueve generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda.
Fuente: Elaboración propia

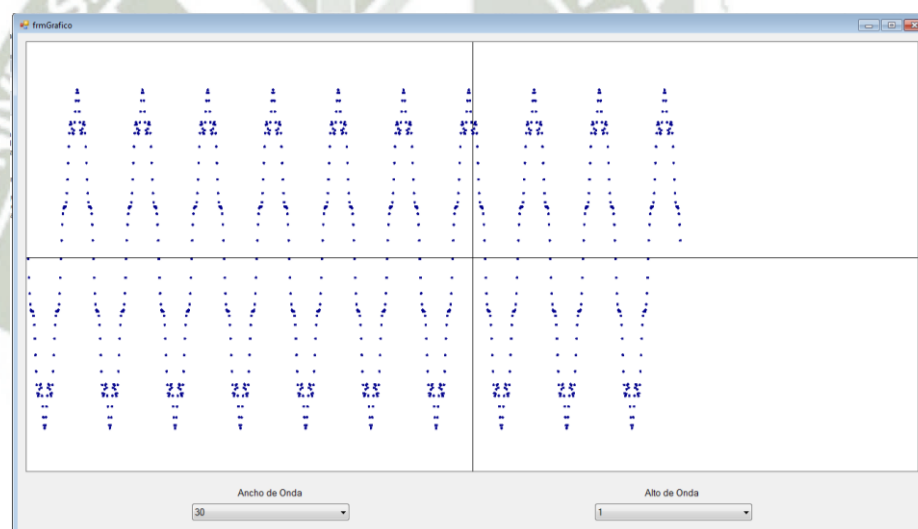


Figura 67: Gráfica del armónico de orden nueve sumado con la señal fundamental generado con los datos obtenidos anteriormente.
Fuente: Elaboración propia

f. 11° armónico

La Figura 68 muestra la generación de datos para el armónico de orden once y en la Figura 69 se puede observar su grafica de acuerdo a los datos hallados anteriormente.

En la Figura 69, como se mencionó anteriormente la amplitud de la imagen es aún más pequeña que la generada con el armónico de orden nueve y en este caso, las ondas generadas están estrechamente separadas, esto es porque se multiplica el valor de amplitud y velocidad angular por $1/11$, esto hace que la amplitud reduzca once veces su tamaño.

Como en el punto anterior, la imagen está muy pequeña y ahora ya casi no se diferencian las ondas generadas, se hace uso de las opciones incluidas en la ventana de la gráfica, en la Figura 70 se muestra la imagen pero mostrando a más detalle la onda generada, mantiene su amplitud pero su ancho varia, esto lo hacemos para distinguir mejor la onda y hacer las comparaciones.

La Figura 71, como en los puntos anteriores, muestra la sumatoria de la señal fundamental con el armónico de orden once, se puede observar que al igual que el punto anterior, la imagen conserva su amplitud original (220v) pero las crestas se acomodan de acuerdo a la imagen generada del armónico de orden nueve.

Como se mencionó anteriormente, si se comparan las imágenes de la Figura 67 y la Figura 71, se ve cierta similitud, esto sucede porque, a más pequeña la imagen de un armónico de un orden mayor, las diferencias serán menores.

Datos Iniciales

Datos Generales

Frecuencia: 60 Amplitud: 220 Presición: 0.01

Generando... Graficar Exportar Data Nuevos Datos

Tipos de Armonicos

☒ Armonicos Simples ☐ Armonicos Combinados

Simples

☐ Onda Fundamental

☐ Tercer Armonico

☐ Quinto Armonico

☐ Setimo Armonico

☐ Noveno Armonico

☒ Onceavo Armonico

Sumados a la Onda Fundamental

☐ Fundamental + Tercer Armonico

☐ Fundamental + Quinto Armonico

☐ Fundamental + Setimo Armonico

☐ Fundamental + Noveno Armonico

☐ Fundamental + Onceavo Armonico

Cantidad de Datos: 1000

Nro Datos	Angulo	Valor
1	0	0
2	0.0314159265	12.748479783
3	0.062831853	19.645745009
4	0.0942477795	17.526133624
5	0.125663706	7.362491112
6	0.1570796325	-6.180339912
7	0.188495559	-16.896558459
8	0.2199114855	-19.84229404
9	0.251327412	-13.690942211
10	0.2827433385	-1.255810532
11	0.314159265	11.755704918
12	0.3455751915	19.371663179
13	0.376991118	18.09654113
14	0.4084070445	8.515586017
15	0.439822971	-4.973797529
16	0.4712388975	-16.180339748
17	0.502654824	-19.960534584
18	0.5340707505	-14.579372732
19	0.565486677	-2.506664953
20	0.5969026035	10.716535646
21	0.62831853	19.021130228
22	0.6597344565	18.59552984

Figura 68: Generando datos para el armónico de orden once.
Fuente: Elaboración propia

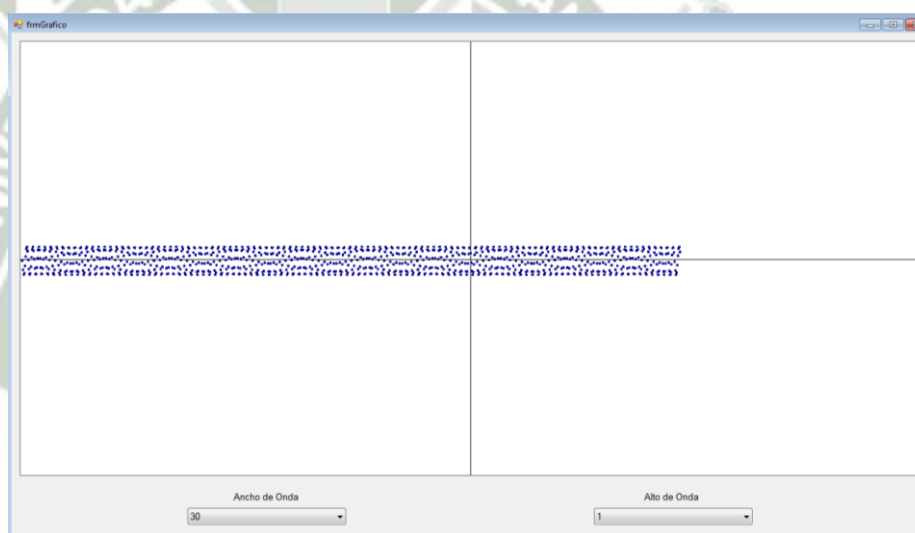


Figura 69: Gráfica del armónico de orden once generado con los datos obtenidos anteriormente, imagen sin modificar el ancho de onda.
Fuente: Elaboración propia

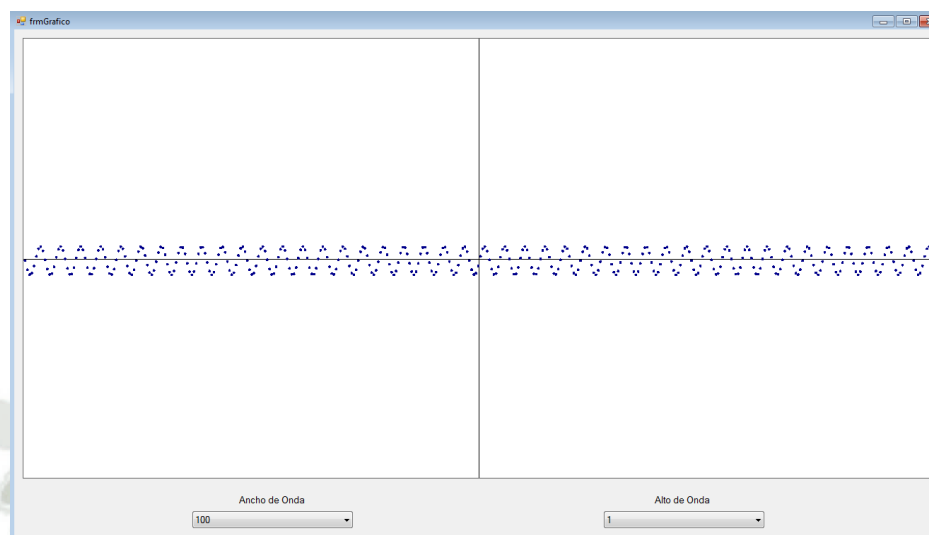


Figura 70: Gráfica del armónico de orden once generado con los datos obtenidos anteriormente, imagen ampliada con la opción de ancho de onda para facilitar la visión de la onda.
Fuente: Elaboración propia

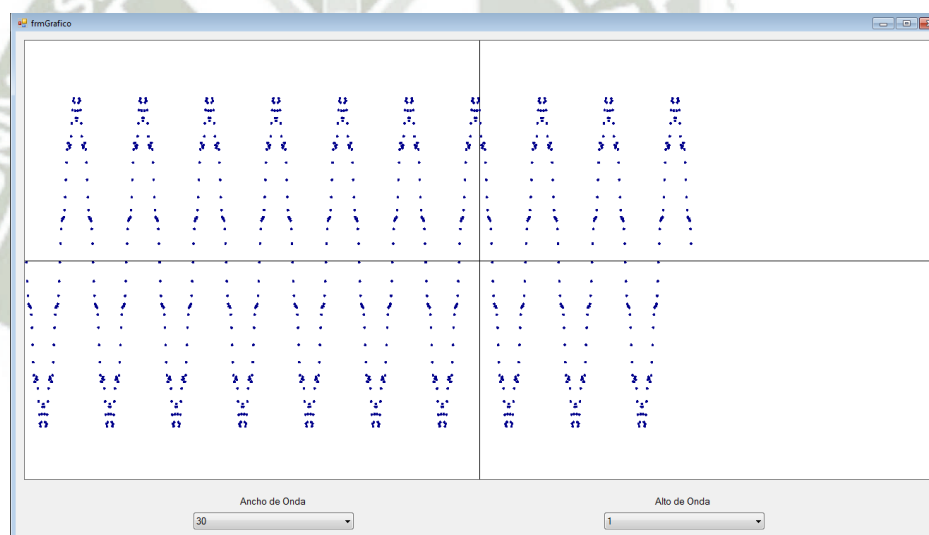


Figura 71: Gráfica del armónico de orden once sumado con la señal fundamental generado con los datos obtenidos anteriormente.
Fuente: Elaboración propia

2. Red neuronal.

Los resultados fueron corroborados por los expertos del proyecto Swap Smart.

Las pruebas se realizaron para el armónico 3, 5, 7 y 9, mostrando en cada uno una imagen diferente, por la cual se va a poder diferenciar de forma gráfica el orden de los armónicos.

Se puede diferenciar por el tipo de grafica que muestra la red, toda señal procesada, que muestre alguno de estos patrones, tanto en el pico positivo o negativo pertenece al armónico del orden que se muestra en cada gráfica, respectiva.



A. Armónico de orden 3

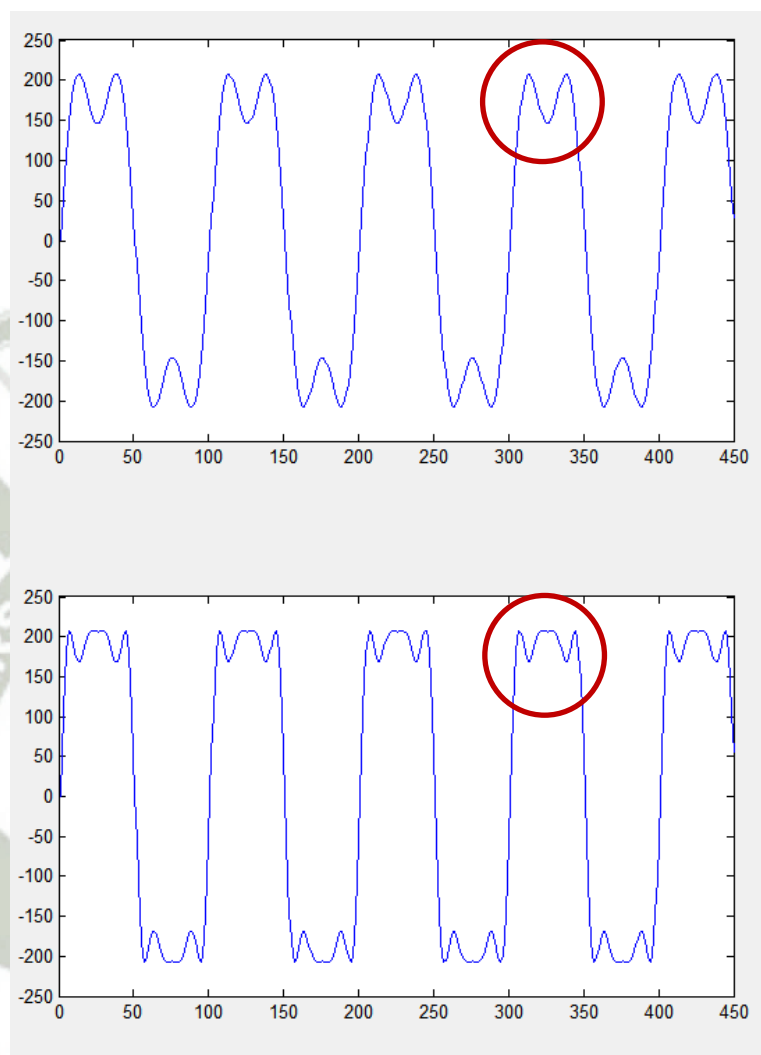


Figura 72: Armónico de orden 3.
Fuente: Elaboración propia

En la Figura 72 podemos ver al Armónico de orden 3 (Grafico superior de la imagen) y el resultado de la señal procesada por la Red Neuronal Artificial (Grafico inferior de la imagen). Podemos saber que el Armónico es de orden 3 por el cambio en las gráficas (Circulo de color rojo) entre las dos graficas de la figura 72.

B. Armónico de orden 5

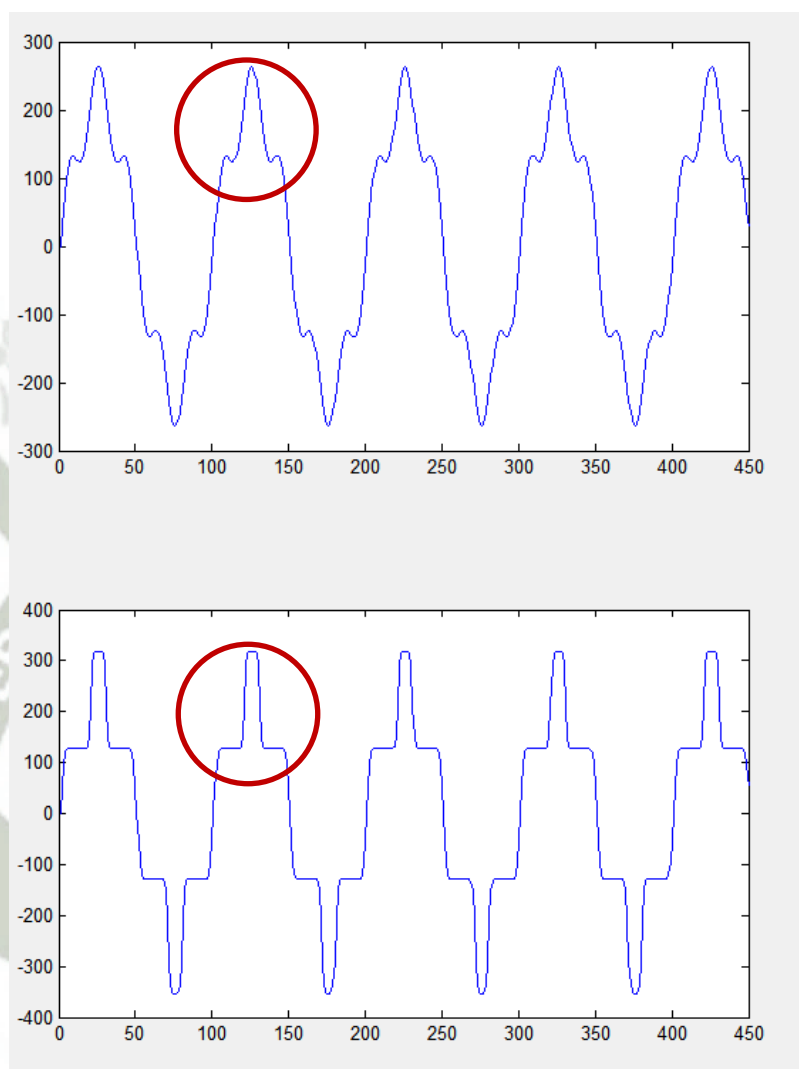


Figura 73: Armónico de orden 5.
Fuente: Elaboración propia

En la Figura 73 podemos ver al Armónico de orden 5 (Grafico superior de la imagen) y el resultado de la señal procesada por la Red Neuronal Artificial (Grafico inferior de la imagen). Podemos saber que el Armónico es de orden 5 por el cambio en las gráficas (Circulo de color rojo) entre las dos graficas de la figura 73.

C. Armónico de orden 7

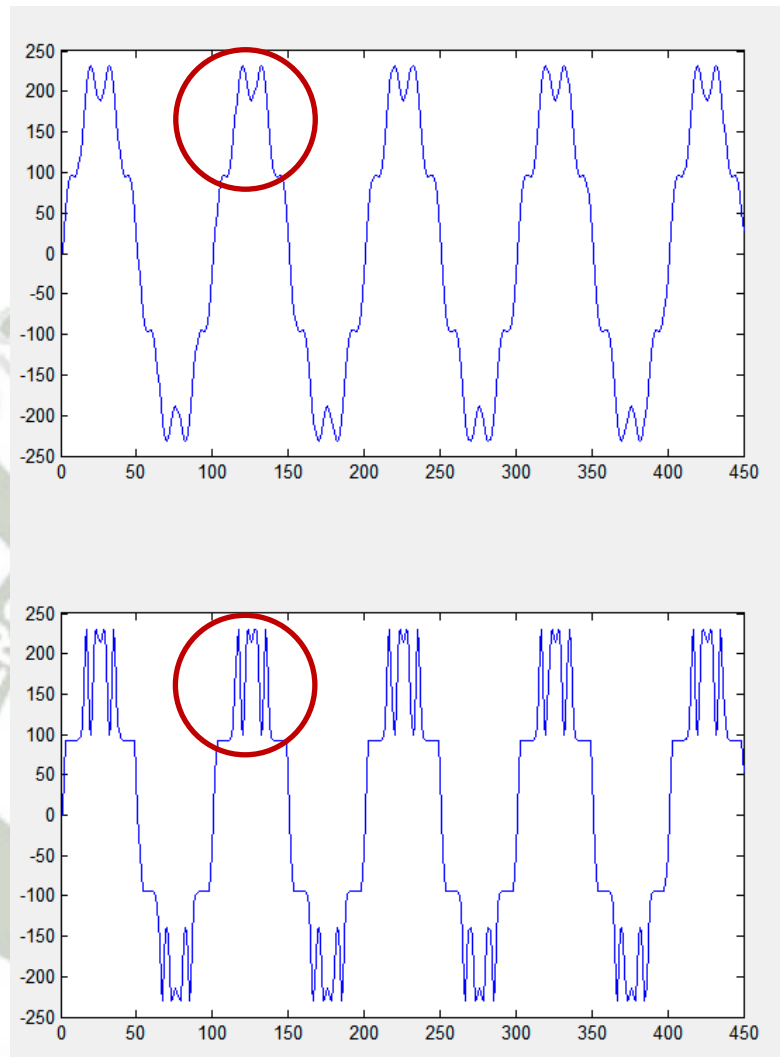


Figura 74: Armónico de orden 7.
Fuente: Elaboración propia

En la Figura 74 podemos ver al Armónico de orden 7 (Grafico superior de la imagen) y el resultado de la señal procesada por la Red Neuronal Artificial (Grafico inferior de la imagen). Podemos saber que el Armónico es de orden 7 por el cambio en las gráficas (Circulo de color rojo) entre las dos graficas de la figura 74.

D. Armónico de orden 9

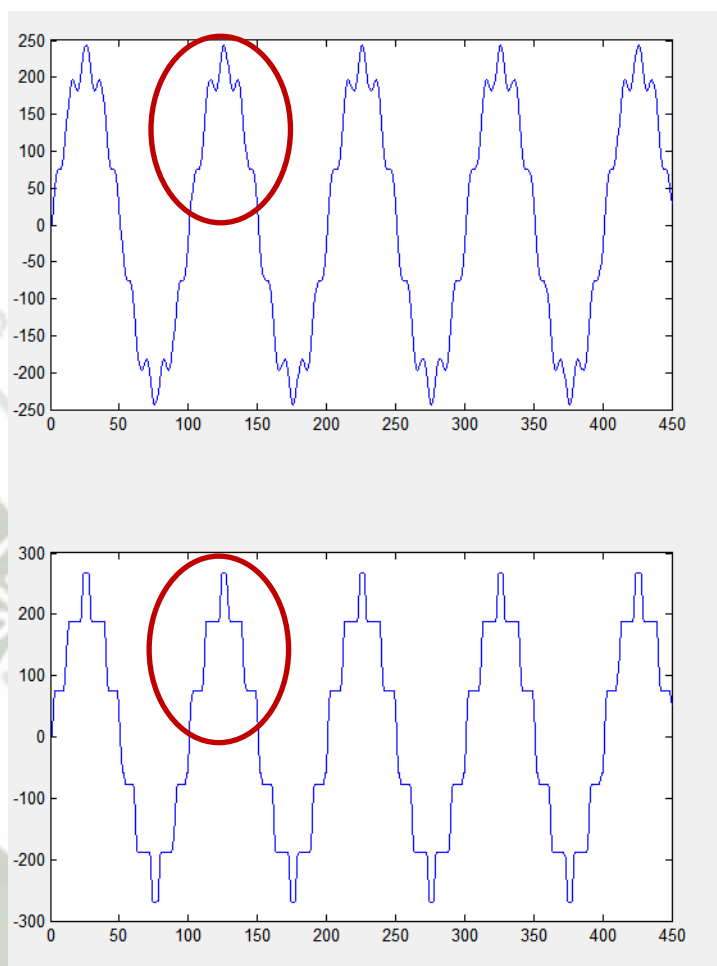


Figura 75: Armónico de orden 9.
Fuente: Elaboración propia

En la Figura 75 podemos ver al Armónico de orden 9 (Grafico superior de la imagen) y el resultado de la señal procesada por la Red Neuronal Artificial (Grafico inferior de la imagen). Podemos saber que el Armónico es de orden 9 por el cambio en las gráficas (Circulo de color rojo) entre las dos graficas de la figura 75.

E. Armónico de orden 11

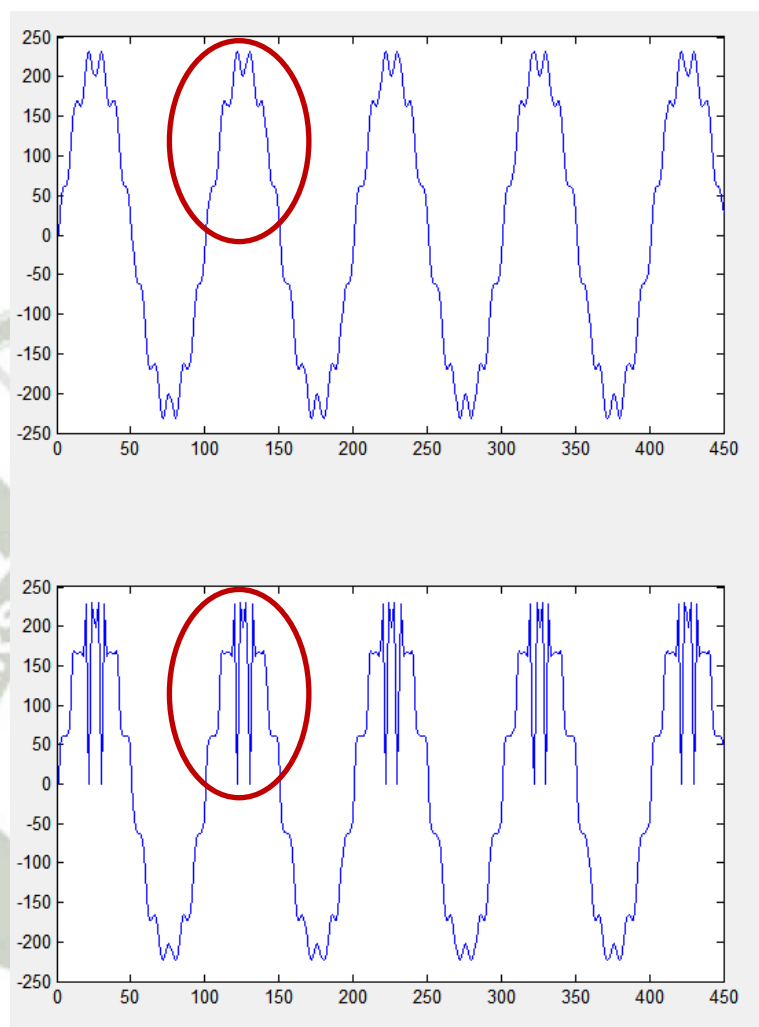


Figura 76: Armónico de orden 11.
Fuente: Elaboración propia

En la Figura 76 podemos ver al Armónico de orden 11 (Grafico superior de la imagen) y el resultado de la señal procesada por la Red Neuronal Artificial (Grafico inferior de la imagen). Podemos saber que el Armónico es de orden 11 por el cambio en las gráficas (Circulo de color rojo) entre las dos graficas de la figura 76.

CONCLUSIONES

1. Por los entrenamientos de la señal fundamental con respecto a los armónicos de corriente eléctrica, podemos detectar visualmente una señal armónica mediante una gráfica después de haber aplicado los métodos para su detección.
2. Para el entrenamiento de la RNA, se usó el perceptrón multicapa con backpropagation, ya que reduce tiempo de procesamiento y al utilizar más de una capa hace un análisis más óptimo.
3. Los armónicos más frecuentes que afectan a las industrias son los armónicos de rango impar, como, 3ro (198 Hz), 5to (330 Hz), 7mo (462 Hz), 9no (594 Hz) y 11avo (726 Hz).
4. Se determinó que la función de activación permite rangos positivos y negativos, y es adaptable al tipo de datos generados por el software.
5. Usamos el método Feedforward ya que la salida es conocida, el armónico de corriente que queremos hallar.
6. Al no haber data real, no se pudo determinar un resultado exacto por que se usó data simulada, pero el prototipo debería funcionar con data real.
7. La investigación determinó que las empresas no mantienen una calidad de energía eléctrica adecuada debido a la falta de medidas de prevención en la instalación de nuevos equipos, ya que estos tienden a producir armónicos de corriente eléctrica y degradan la esta señal, perjudicando a futuro los equipos eléctricos en general.
8. Se logró recrear señales de armónicos de corriente intentando llegar lo más cercano a un caso real.

9. El software diseñado se puede aplicar en forma didáctica a los campos de electrónica y mecánica eléctrica para el estudio de señales.
10. Debido a que la configuración de algunos sistemas operativos están determinados en distintos idiomas, se tuvo el problema del punto y/o coma decimal, por esa razón el software guarda la data generada en dos archivos tanto con punto y coma decimal.
11. El tratamiento de señales en general tiene un trabajo arduo, el proyecto presentó una forma idealizada y compacta de recrear, procesar y obtener data para aplicarlos a la red neuronal.



RECOMENDACIONES

1. Debido que, MatLab tiene las funciones de RNA internas no es posible obtener el valor de algunas variables, sería recomendable desarrollar una RNA y así obtener y/o generar variables para poder mostrar un resultado determinativo y no solo visual como se logró en este trabajo.
2. Se recomienda investigar acerca del uso de memoria en el proceso de algoritmos matemáticos, para mejorar el rendimiento de memoria de un equipo de cómputo en al momento de generar las señales.
3. Se recomienda aplicar un algoritmo para generar la señal fundamental sumada a cualquier armónico en forma aleatoria daría como resultado cálculos más cercanos a lo real al momento de procesarse en la red neuronal.
4. Para la generación de una señal de corriente eléctrica trifásica es posible que se pueda modificar y/o adicionarle un módulo para que genere este tipo de señales.
5. En una futura investigación de desarrollo de software con programación paralela se podría aplicar estos métodos de programación para el cálculo de las señales, aprovechar las ventajas de los procesadores actuales que existen en el mercado.
6. Se recomienda desarrollar una librería específica para la manipulación y generación de la data y centralizar el uso en un solo entorno, de forma que esta librería sea incluida en el conjunto de librerías de MatLab.

BIBLIOGRAFÍA

- [HS, 1999] Haykin, Simon. “Neural networks: a comprehensive foundation”, 2ª ed., Prentice-Hall, New Yersey (EEUU).

- [POJA, 2002] Pérez Ortiz, Juan Antonio. “Modelos Predictivos Basados en Redes Neuronales Recurrentes De Tiempo Discreto”, España: Universidad de Alicante.
<http://rua.ua.es/dspace/bitstream/10045/3826/1/P%C3%A9rez%20Ortiz,%20Juan%20Antonio.pdf>

- [HGL, 2002] Hontoria García, Leocadio “Generación de series sintéticas de radiación solar combinando herramientas estadísticas y redes neuronales”, España: Universidad de Jaén.
<http://www4.ujaen.es/~hontoria/PagPerLHG1/PPLHontoriaInvestigacion/PPLHontoriaInvestigacion.htm>

- [LRA] De La Rosa, Amín. Disponible en:
<http://www.todomonografias.com/electronica-y-electricidad/armonicos/>

- [MATHWORKS,2014] Disponible en:
<http://www.mathworks.es/products/datasheets/pdf/matlab.pdf>

- [ER,] Escobar, Rolando. Disponible en:
<http://rolandoescobar.wordpress.com/2011/02/24/manuales-net-manual-c/>

- [GEP, 2002] García Estévez, Pablo “Aplicaciones de las Redes Neuronales en las Finanzas”, España: Facultad de Ciencias Económicas y Empresariales.
<http://eprints.ucm.es/6767/1/0205.pdf>

- [DNCF, 2007] Do Nascimento, Claudinor Francisco, “Determinação do conteúdo harmônico de corrente baseada em redes neurais artificiais para cargas não-lineares monofásicas”, Brasil: Universidad de Sao Paulo
- [FARAGAUSS SYSTEM] Disponible en: http://www.faragauss-international.com/articulos/fuente_distoarmonic.htm, consultado en julio del 2014
- [NSE, 2002] Noriega Stefanova, Ernesto, “Generalidades sobre Armónicos y su influencia en Sistemas de Distribución de Energía” Cuba: Empresa Eléctrica Matanzas. Disponible en: <http://www.monografias.com/trabajos21/armonicos/armonicos.shtml>



GLOSARIO DE TÉRMINOS

- Red Neuronal Artificial (RNA): Red compuesta de muchas unidades de procesamiento simple que están comunicadas a través de conexiones que usualmente transportan información.
- Armónico de corriente: Distorsiones de las ondas sinusoidales de tensión y/o corriente de los sistemas eléctricos.
- Distorsión armónica: Cuando el voltaje o la corriente de un sistema eléctrico tienen deformaciones con respecto a la forma de onda senoidal.
- Efecto Corona: Fenómeno eléctrico que se produce en los conductores de las líneas de alta tensión y se manifiesta en forma de halo luminoso a su alrededor.
- Interruptores diferenciales: Dispositivo electromecánico que se coloca en las instalaciones eléctricas con el fin de proteger a las personas de las derivaciones causadas por faltas de aislamiento entre los conductores activos y tierra o masa de los aparatos.

- Reactancias inductivas: Es la oposición o resistencia que ofrecen al flujo de la corriente por un circuito eléctrico cerrado las bobinas o enrollados hechos con alambre de cobre.
- Oscilación de frecuencias: Número de ciclos por segundo, o hercios (Hz),
- Perceptrón multicapa: Es una RNA formada por múltiples capas, esto permite resolver problemas que no son linealmente separables.
- Backpropagation: Algoritmo de aprendizaje supervisado que se usa para entrenar RNA.
- Transformada de Fourier: Es el espectro de frecuencias de una función.
- Onda de tensión: Es la diferencia de potencial entre dos puntos graficado en forma de onda.
- Rectificadores: Permite convertir la corriente alterna en corriente continua.
- Monofásicos: Alimentación en corriente alterna por una fase de la red eléctrica.

- **Peso sináptico:** O también llamado peso, son las conexiones entre neuronas.
- **Hiperplano:** Un hiperplano es un plano corriente; divide el espacio en dos mitades.
- **Computables:** Calculable.
- **Umbrales de saturación:** Es la máxima concentración o intensidad del estímulo que puede ser captada.
- **Función Sigmoidal:** Función matemática, gráficamente tiene la forma de una “S”.
- **Peso de polarización:** Valor constante de entrada en un RNA.
- **Perceptrón converge:** Perceptrón con un numero finita de iteraciones, si la data es linealmente separable.
- **PMC:** Perceptrón Multi Capa
- **Impedancia no lineal:** Es una magnitud que establece la relación entre la

tensión y la intensidad de corriente.

- Potencia activa: Es la potencia disipada por las cargas resistivas.
- Dieléctricos: Material mal conductor de electricidad.
- Frecuencia fundamental: Es la frecuencia más baja del espectro de frecuencias.

