

Construcción de un prototipo de software para el procesamiento y reconocimiento de las marchas básicas de un caballo en los ejes (X, Y, Z), con el fin de generar modelos de movimientos aproximados

Andrés Felipe Camacho Barragán & Diego Andrés Carranza Rivera

Universidad de Cundinamarca
Facultad de Ingeniería Programa
Ingeniería de Sistemas,
noviembre 2017

**Construcción de un prototipo de software para el procesamiento y ii
reconocimiento de las marchas básicas de un caballo en los ejes (X, Y, Z), con
el fin de generar modelos de movimientos aproximados**

Andrés Felipe Camacho Barragán & Diego Andrés Carranza Rivera

Director: Ing. Jaime Eduardo Andrade Ramírez

Grupo de investigación de sistemas y tecnología de
Facatativá (**GISTFA**)

Universidad de Cundinamarca
Facultad de Ingeniería Programa
Ingeniería de Sistemas,
noviembre 2017

Mi padre, mi madre y mi hermana, son las personas de las que más he aprendido y que nunca me han negado su apoyo, a ellos mi infinito agradecimiento por estar siempre a mi lado en este largo proceso de aprendizaje que es la vida.

Andrés Camacho

Agradezco, a mis padres que me apoyaron en cada fase de mi proceso educativo y al docente Alexander Espinosa por la disposición y colaboración en algunas de las etapas del Proyecto.

Andrés Carranza

Yo, Andrés Felipe Camacho Barragán con célula de identidad No. 1073243255 y con cód. 461213207 estudiante del programa de Ingeniería de Sistemas de la universidad de Cundinamarca, declaro que: El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al Director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

Firma: Andrés Camacho B.

Compromiso de Autor

v

Yo, Diego Andrés Carranza Rivera con célula de identidad No. 1016069879 y con cód. 461213208 estudiante del programa de Ingeniería de Sistemas de la universidad de Cundinamarca, declaro que: El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al Director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

Firma: _____



Actualmente el proyecto de investigación titulado “**Diseño de un simulador de tiro montado en la escuela de carabineros Alfonso López Pumarejo**” de la escuela de carabineros Alfonso López Pumarejo carece de una herramienta tecnológica que aporte en la generación de modelos de movimientos aproximados de los movimientos o marchas básicas equinas (paso, trote y galope) que sirven para la construcción de un simulador de tiro montado, facilitando la instrucción de los jinetes y reduciendo los accidentes durante las clases de equitación. El presente trabajo de investigación tuvo como fin la construcción de un prototipo de hardware y software que mediante una interfaz permita la captura de las marchas básicas de un semoviente equino. Adjunto en el prototipo de software se diseñaron dos interfaces para el entrenamiento de una red neuronal artificial que reconozca los movimientos capturados en el dominio del tiempo y en el dominio de la frecuencia. Para llevar a cabo la construcción del sistema, se siguió una metodología, que comenzó con el análisis de requerimientos globales, el diseño de hardware y software, la construcción del prototipo, la integración del sistema y la validación del prototipo. El resultado de la investigación fue en un prototipo de hardware y software que captura un número de muestras de un movimiento por medio de los sensores seleccionados y dos redes neuronales artificiales que reconocen los movimientos capturados bajo un cierto porcentaje de error; así como la visualización de los datos mediante gráficas. Por medio de un robot equino de pequeña escala con tres secuencias de movimientos programadas, se procedió a evaluar el prototipo, capturando un número de muestras de cada secuencia, que posteriormente sirvieron para el entrenamiento de las redes neuronales, siendo la red del dominio de la frecuencia la que mejor reconoce los movimientos o secuencias capturadas.

Palabras clave: RNA, Acelerómetros, Matlab, Prototipo, metodología

vii

Índice General

viii

Lista de tablas	ix
Lista de figuras.....	xi
Lista de Anexos.....	xiii
Introducción	xiv
Informe De Investigación	17
Estado del Arte.....	17
Línea de Investigación	20
Planteamiento del Problema	20
Objetivo General.....	21
Objetivos Específicos.....	21
Metodología	23
Análisis de requerimientos globales.	25
Modelo de Prototipos.....	29
Validación del prototipo..	49
Impacto del Proyecto	55
Poblacional en el ámbito policial y académico.....	55
En Innovaciones tecnológicas.....	56
Marcos de Referencia	57
Marco Histórico	57
Marco Teórico.....	62
Marco Legal	73
Documentación del Software	78
Plan de Proyecto	78
Estimación de recursos	78
Determinación de requerimientos	79
Especificación del Sistema	85
Definición del Sistema.....	85
Especificación del Diseño.....	87
Diagramas de Casos de Uso.....	87
Descripción de Casos de Uso.....	89
Diagramas de Secuencia:	91
Diagrama de Colaboración	94
Diagramas de Actividades	97
Diagrama de Despliegue:.....	100
Diseño de los Casos de Prueba	101
Resultados	108
Conclusiones y Recomendaciones	117
Lista de referencias	121
Anexos	128

Tabla 1 Comparación de los acelerómetros	26
Tabla 2 Comparación de los módulos de Radio Frecuencia.....	27
Tabla 3 Costo Total Aproximado de los Componentes Físicos.....	29
Tabla 4. Sesiones de Captura	50
Tabla 5. Sesiones de entrenamiento en el tiempo basados en las capturas 1 y 3.....	52
Tabla 6 Sesiones de entrenamiento en el tiempo basados en la captura 2.....	53
Tabla 7. Sesiones de entrenamiento en frecuencia basadas en las capturas 1 y 3	54
Tabla 8. Sesiones de entrenamiento en frecuencia basadas en la captura 2	55
Tabla 9. Cronograma y descripción de actividades	78
Tabla 10. Recursos utilizados	79
Tabla 11. Requerimiento Funcional 1.....	79
Tabla 12 Requerimiento Funcional 2.....	80
Tabla 13 Requerimiento Funcional 3.....	80
Tabla 14 Requerimiento Funcional 4.....	80
Tabla 15 Requerimiento Funcional 5.....	81
Tabla 16 Requerimiento Funcional 6.....	81
Tabla 17 Requerimiento Funcional 7.....	81
Tabla 18 Requerimiento Funcional 8.....	81
Tabla 19 Requerimiento Funcional 9.....	82
Tabla 20 Requerimiento Funcional 10.....	82
Tabla 21 Requerimiento Funcional 11.....	82
Tabla 22 Requerimiento Funcional 12.....	83
Tabla 23 Requerimiento Funcional 13.....	83
Tabla 24 Requerimiento Funcional 14.....	83
Tabla 25 Requerimiento NO Funcional 1	84
Tabla 26 Definición del sistema	86
Tabla 27 Definición – Casos de uso	89
Tabla 28 Diagrama de secuencia - Definiciones.....	94
Tabla 29 Diagrama de despliegue - Elemento	100
Tabla 30 Diseño de casos de prueba 1	101
Tabla 31 Diseño de casos de prueba 2	101
Tabla 32 Diseño de casos de prueba 3	102
Tabla 33 Diseño de casos de prueba 4	102
Tabla 36 Diseño de casos de prueba 7	104
Tabla 37 Diseño de casos de prueba 8	104
Tabla 38 Diseño de casos de prueba 9	105
Tabla 39 Diseño de casos de prueba 11	105
Tabla 40 Diseño de casos de prueba 11	106
Tabla 41 Diseño de casos de prueba 12	106
Tabla 42 Diseño de casos de prueba 13	107
Tabla 43 Porcentaje de error promedio de las sesiones de entrenamiento en el tiempo.....	110
Tabla 44 Porcentaje de error promedio de las sesiones de entrenamiento en frecuencia	110

Tabla 45 Error promedio (%) del entrenamiento en el tiempo basados en la sesión de captura 3 x	111
Tabla 46 Error promedio (%) del entrenamiento en frecuencia basado en la sesión de captura 3	111
Tabla 47 Promedio de ciclos de la red en el tiempo basados en la sesión de captura 3	111
Tabla 48 Promedio de ciclos de la red en frecuencia basados en la sesión de captura 3.....	111

Figura 1 Metodología general del sistema	24
Figura 2 Modelo de Prototipos	25
Figura 3 Conexión de los dispositivos	33
Figura 4 Flujo de datos entre dispositivos	33
Figura 5 Flujo de uso de la ventana principal	35
Figura 6 Flujo general de uso de la segunda ventana	37
Figura 7 Flujo general de uso de la tercera ventana.....	38
Figura 8 Entrada de la RNA en el dominio del tiempo.....	39
Figura 9 Arquitectura de la RNA.....	41
Figura 10 Flujo de entrenamiento de la RNA.....	42
Figura 11 Montaje Xbee sobre protoboard	43
Figura 12 Baquelita módulo transmisor.....	43
Figura 13 Baquelita módulo receptor	44
Figura 14 Módulo transmisor.....	44
Figura 15 Módulo receptor	45
Figura 16 Acelerómetros con extensiones	45
Figura 17 Ventana principal.....	46
Figura 18 Segunda ventana.....	47
Figura 19 Tercera ventana	47
Figura 20 Robot Equino.....	49
Figura 21 Prototipo completo	50
Figura 22 Acelerómetro con tres ejes	63
Figura 23 Acelerómetro triaxial MMA7455.....	64
Figura 24 Módulo de radio frecuencia Xbee PRO S1	65
Figura 25 Neurona biológica	67
Figura 26 Neurona artificial.....	68
Figura 27 Arquitectura red neuronal artificial	69
Figura 28 Nodo y red con propagación hacia atrás.....	69
Figura 29 Proceso backpropagation.....	71
Figura 30 Arduino Mega 2560.....	72
Figura 31 Casos de uso - Captura de datos	88
Figura 32 Casos de uso - RNA – Tiempo	88
Figura 33 Casos de uso - RNA – Frecuencia.....	89
Figura 34 Diagrama de secuencia - Captura de datos.....	92
Figura 35 Diagrama de secuencia - Entrenamiento RNA – Tiempo	92
Figura 36 Diagrama de secuencia - Entrenamiento RNA – Frecuencia	93
Figura 37 Diagrama de secuencia - Comprobación RNA	93
Figura 38 Diagrama de colaboración - Captura de datos.....	95
Figura 39 Diagrama de colaboración - Entrenamiento RNA tiempo	95
Figura 40 Diagrama de colaboración -Entrenamiento RNA frecuencia.....	96
Figura 41 Diagrama de colaboración - Comprobación RNA	96
Figura 42 Diagrama de actividades - Captura de datos	97
Figura 43 Diagrama de actividades - Entrenamiento y Comprobación RNA – Tiempo	98

Figura 44 Diagrama de actividades - Entrenamiento y Comprobación RNA - Frecuencia.....	99xii
Figura 45 Diagrama de despliegue	100
Figura 46 número de neuronas vs promedio de ciclos en el tiempo - sesión de captura 3	112
Figura 47 tasa de aprendizaje vs promedio de ciclos en el tiempo-sesión de captura 3	113
Figura 48 número de neuronas vs promedio de ciclos en la frecuencia-sesión de captura 3	114
Figura 49 tasa de aprendizaje vs promedio de ciclos en la frecuencia-sesión de captura 3	115

Lista de Anexos

<u>Anexo A Primer prototipo de software desarrollado</u>	<u>128</u>
<u>Anexo B Segundo prototipo de software desarrollado.....</u>	<u>129</u>
<u>Anexo C Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 1.....</u>	<u>130</u>
<u>Anexo D Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 2.....</u>	<u>131</u>
<u>Anexo E Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 3.....</u>	<u>132</u>
<u>Anexo F Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 1.....</u>	<u>133</u>
<u>Anexo G Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 2.....</u>	<u>134</u>
<u>Anexo H Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 3.....</u>	<u>135</u>

Introducción

El presente trabajo de investigación forma parte de uno de los objetivos del proyecto titulado “**Diseño de un simulador de tiro montado en la escuela de carabineros Alfonso López Pumarejo**” realizado por la Universidad de Cundinamarca conjunto con la escuela de carabineros Alfonso López Pumarejo, que tiene como principal objetivo la construcción de un simulador de tiro montado que permita la instrucción de los jinetes en las clases de equitación reduciendo el número de accidentes; para la construcción de dicho simulador es necesario basarse en movimientos reales de equinos en sus diferentes marchas básicas (paso, trote y galope), debido a esto, para lograr la construcción del simulador, esta investigación se hace con el fin de contribuir con el desarrollo de una herramienta que ayude en la generación de modelos de movimientos de las marchas básicas equinas.

A lo largo del documento se encontrarán diferentes capítulos como el de los marcos de referencia que sostienen los fundamentos teóricos y legales de la investigación, el informe de investigación que tiene en cuenta los trabajos realizados de otras investigaciones similares con el tema tratado, así como la metodología usada para dar solución a los objetivos planteados, la especificación del diseño del software desarrollado, los resultados de la investigación y las conclusiones.

El software propuesto captura un número de muestras de un determinado movimiento por medio de diferentes sensores y permite visualizarlos en el dominio del tiempo y la

frecuencia mediante una interfaz de usuario que también realiza el reconocimiento de diferentes movimientos por medio de redes neuronales artificiales.

Sobre las bases de las ideas expuestas, el desarrollo del prototipo de software se inició con el análisis de requerimientos, con el fin de seleccionar los dispositivos de hardware que más se adecuan a la finalidad del proyecto teniendo en cuenta los costos, formas de comunicación, facilidad de adquisición, el rendimiento de los mismos y también un lenguaje de programación que permitiera una fácil adquisición y manipulación de los datos entregados por los dispositivos; posteriormente el desarrollo de hardware y software se hizo con base en el modelo de prototipos que parte de la identificación de requerimientos, el diseño y construcción rápida del prototipo, la evaluación por parte del cliente, refinamiento de los requerimientos y nuevas iteraciones que parten del diseño y terminan en el producto final; por último se hace la validación del prototipo.

Durante el diseño y construcción de hardware y software se determinaron las interfaces de comunicación de los dispositivos seleccionados, el modo de conexión y el flujo de traspaso de información entre los mismos; posteriormente de acuerdo con la información enviada, se definieron las interfaces de usuario del programa desarrollados para el tratamiento de los datos y el reconocimiento de los movimientos capturados por los sensores.

Para la validación del prototipo tanto de hardware como de software se hicieron tres sesiones de capturas diferentes desde la interfaz de usuario, a diferentes distancias y con diferente número de muestras; para posteriormente sobre cada una de las sesiones de capturas efectuadas, realizar el entrenamiento de la red neuronal con diferentes parámetros, como el número de neuronas, la tasa de aprendizaje y el porcentaje de los datos capturados durante la sesión, tanto en el dominio del tiempo como en el dominio de la frecuencia.

Informe De Investigación

Estado del Arte

Específicamente se presentan adelantos referidos a la detección de movimientos de equinos que son de gran ayuda a la hora de tomar decisiones orientadas hacia las técnicas y procedimientos en la construcción del simulador: Pfau, Stubbs, Kaiser, Brown, & Clayton, (2012) presentan un estudio del efecto de la velocidad de trote en equinos y la simetría del movimiento en círculos sobre una superficie blanda.

Starke, Witte, May, & Pfau, (2012) Muestran la exactitud y precisión de los tiempos de contacto de las extremidades posteriores de un equino montado sobre la pelvis y determinados a partir de unidades de medidas inerciales.

S. D. Starke, Willems, Head, May, & Pfau, (2012) Hace un estudio de flexión próxima en las extremidades posteriores en el equino: efecto sobre la simetría del movimiento y las implicaciones para definir su estado de salud.

Olsen, Haubro Andersen, & Pfau, (2012) Presenta exactitud y precisión de la detección equina con sensores inerciales montados en el tronco de un equino y caminando.

Sandra D. Starke, Willems, May, & Pfau, (2012) presenta el estado de la cabeza vertical y adaptaciones de movimientos del tronco en caballos con los sonidos del trote en una superficie dura.

WARNER, KOCH, & PFAU, (2010) utiliza sensores inerciales para la evaluación del movimiento hacia atrás en caballos durante la marcha en terreno.

WALKER, WILSON, & PFAU, (2010) realiza la comparación de cálculos de índices de simetría cinemática y los efectos del trote en círculo hacia la derecha.

Valentin, Davis, Wilson, & Pfau, (2010) muestra el efecto de la carga en el lomo de un equino - un estudio preliminar.

T. Pfau, Spence, Starke, Ferrari, & Wilson, (2009) muestra que el estilo de conducción moderna mejora los tiempos de las carreras de caballos.

S. D Starke, Robilliard, Weller, Wilson, & Pfau, (2009) presenta la clasificación de las marchas según los aires del caballo: un enfoque multidimensional.

Church, Walker, Wilson, & Pfau, (2009) presenta la evaluación del análisis discriminante basado en índices de simetría dorso ventral para cuantificar la cojera en extremidad posterior durante las marchas del equino en tierra.

PARKES, WELLER, GROTH, MAY, & PFAU, (2009) evidencia el desarrollo de conocimientos “dominio restringido” en el reconocimiento de características de movimientos asimétricos de cojera en el miembro posterior del caballo.

Thilo Pfau, Ferrari, Parsons, & Wilson, (2008) comparte la técnica de segmentación basada en la zancada usando modelos ocultos de markov aplicada al movimiento del tronco en equinos mediante sensores inerciales.

Parsons, Pfau, Ferrari, & Wilson, (2008) estudia la locomoción de un caballo pura sangre al galope de alta velocidad: El efecto de la inclinación en el centro del movimiento de masas y la fluctuación de la energía mecánica.

PFAU et al., (2007) evalúa la cojera leve del miembro posterior equino durante la locomoción terrestre usando análisis de discriminantes lineales de los datos de sensores inerciales.

T. Pfau, (2006) muestra el centro del movimiento de masas y la fluctuación de la energía mecánica durante la locomoción al galope en el caballo de carrera pura sangre.

T. Pfau, (2005) comparte un método para derivar datos de desplazamiento equino durante el movimiento cíclico utilizando un sensor inercial.

Línea de Investigación

Software, Sistemas Emergentes, Nuevas Tecnologías

Planteamiento del Problema

La Escuela Nacional de Carabineros Alfonso López Pumarejo, tiene como misión formar y capacitar integralmente el talento humano de la policía nacional de Colombia, a través del proyecto educativo institucional con el propósito de desarrollar competencias para el servicio de policía.

Al momento de realizar las clases e instrucciones de equitación para el programa académico Técnico Profesional en Seguridad Rural montada (curso de carabinero), se han presentado una serie de accidentes que han lesionado en algunas ocasiones de forma grave a los jinetes y/o a los semovientes equinos.

En la instrucción de equitación, se imparte la asignatura Tiro montado, allí se imparten las diferentes actividades de la clase, desde a pie firme hasta los ejercicios de disparar pistola Sig Sauer SP 2020 y Fusil Galil 5.56 sobre el caballo. la práctica y ejecución de estos ejercicios sin una previa instrucción puede incurrir en accidentes y en algunas ocasiones graves como fracturas, que generan ausentismo laboral y en algunas ocasiones lesiones de gran severidad para el semoviente equino.

La construcción de un simulador a escala que permita el entrenamiento del jinete de forma segura carece de un modelo de software que permita reconocer y modelar los movimientos aproximados de las marchas equinas básicas (Paso, Trote, Galope) y así evitar gastos innecesarios a la hora de construir el simulador.

¿Es posible la construcción de un prototipo de software que detecte los movimientos o marchas básicas de un caballo y sirva como referencia para la construcción de un modelo a escala?

Objetivo General

Hacer un aplicativo en software que realice el procesamiento automático de los datos de posición del caballo, generando resultados significativos del ejercicio y que presente una interfaz de usuario para visualizar o ilustrar los datos del movimiento del caballo según los requerimientos espaciales a que haya lugar.

Objetivos Específicos

- Realizar un estado del arte de módulos de captura y registro en equinos que se han desarrollado o ya se comercializan, con el fin de verificar sus características más relevantes, estándares de implementación, funcionamiento y niveles de satisfacción en el caso de ser usados para la instrucción y/o capacitación de personas.

- Identificar y caracterizar el sensor más apropiado para tal fin, teniendo en cuenta: costos, comercialización, precisión, exactitud, linealidad, señal generada, tipo de comunicación.
- Implementar un sistema electrónico portátil y fácil de usar que se adapte adecuadamente al caballo y permita capturar los movimientos más representativos del mismo, utilizando varios sensores en diferentes marchas, a cualquier velocidad y garantizar el envío de los datos en forma inalámbrica y en “tiempo real” a una computadora, de tal forma, que se considere el mínimo de variables y poder ajustar la metodología y procedimiento de captura.
- Establecer los sistemas de acondicionamiento, filtrado y conversión de la(s) señal(es) entregada (s) por el (los) sensor(es).
- Definir el sistema digital micro procesado para la captura y almacenamiento de los datos, referentes a los patrones del movimiento en coordenadas x,y,z para cada uno de los sensores.
- Definir e implementar el sistema de comunicación inalámbrica que permita enviar los datos capturados del movimiento desde el caballo hasta una estación móvil de trabajo (equipo de cómputo) en forma rápida y fiable.

- Garantizar la alimentación (energía) constante de los instrumentos y equipos electrónicos instalados en el caballo, según los parámetros eléctricos de los componentes.
- Integrar todos los componentes de hardware y adaptarlos según la anatomía del semoviente.
- Desarrollar una interfaz que integre la identificación de los pasos básicos equinos mediante Inteligencia artificial (RNA) y permita controlar las secuencias de movimiento del robot equino según la marcha seleccionada con sus respectivos gráficos en el tiempo y la frecuencia.

Metodología

El presente trabajo de investigación se sitúa dentro de la línea de investigación **Software, Sistemas emergentes y Nuevas tecnologías** del programa Ingeniería de Sistemas de la Universidad de Cundinamarca, donde se diseña y se construye un prototipo de hardware y software, que captura, procesa y reconoce las secuencias de los movimientos de un robot a pequeña escala que simula los movimientos equinos.

Recolectar información y modelar todo el sistema que conforma el prototipo de software no es posible solo desde la exploración sino también desde la observación, por lo que se considera una investigación mixta que permita conseguir un enfoque más extenso y

profundo del fenómeno, para que la percepción de este sea integral, completa y holística. (Hernandez Sampieri, Fernandez Collado, & Baptista Lucio, 2010).

La investigación se realizó en tres etapas o fases principales como se muestra en la figura 1 estas fase son: el análisis de requerimientos globales donde se analizan las herramientas y dispositivos con que funcionará el sistema, la fase de aplicación del modelo de prototipos tal como se muestra en la figura 2, esta fase consta de seis sub-fases que son la identificación de requerimientos donde se detalla cuáles deben ser las características del prototipo, el diseño de software y hardware en la cual se analizan como integrar todas las herramientas, la construcción del prototipo donde se lleva a cabo el montaje y la construcción de los dispositivos así como el desarrollo del software, la evaluación del cliente, en la que el cliente evalúa el resultado de la construcción, el refinamiento del prototipo para establecer nuevos requerimientos o aclarar los existentes y el producto terminado donde el cliente acepta todos los cambios presentados durante la evaluación; por último se encuentra la tercera y última fase de validación del prototipo donde se prueba el funcionamiento de todo el sistema en general.

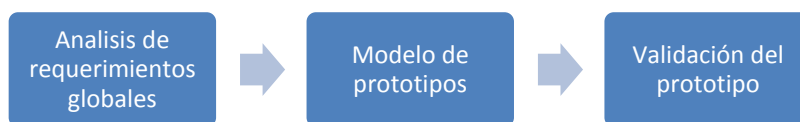


Figura 1 Metodología general del sistema

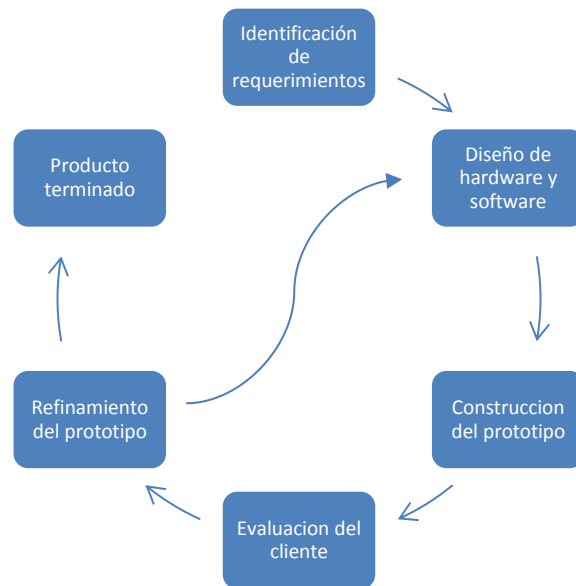


Figura 2 Modelo de Prototipos

A continuación, se explicará cada una de las fases desarrolladas:

Análisis de requerimientos globales. En esta fase se definieron todos los dispositivos y herramientas que integrarían el sistema con base en lecturas de otros trabajos de investigación y las características de los mismos, a continuación, se detallan los dispositivos seleccionados y el criterio de selección:

1. Acelerómetro MMA7455 para la captura de muestras de movimientos:

La selección del dispositivo se hizo basada en una comparación con otro acelerómetro como se muestra en la tabla 1:

Acelerómetros	MMA7361	MMA7455
Consumo de Corriente	400 uA	400 – 490 uA
Voltaje de operación	2.2 – 3.6 V	2.4 – 3-6 V
Sensibilidad	1,5g – 6g (mV/g)	2g,4g,8g (g/count)
Tipo de salida	Análoga	Digital (I2C, SPI)
Calibración	Auto test para calibración	Registro para calibración
Robustez	Diseño robusto, alta amortiguación de impactos (1000g)	Diseño robusto, alta amortiguación de impactos (5000g)
Precio Aproximado	\$19.000	\$16.500

Tabla 1 *Comparación de los acelerómetros*

Una vez hecha la comparación se selecciona el MMA7455 con base en las siguientes explicaciones

- El manejo de la potencia (que el sistema tenga autonomía por periodos prologados de tiempo “varias horas”).
- La comunicación y la alimentación desde el sensor hasta el sistema de control (I2C o SPI) y que permitiera alimentarse desde el módulo de control.
- Los rangos de sensibilidad (2G, 4G y 8G) sean seleccionables.
- Los datos entregados por el sensor estén en formato digital.
- El dispositivo sea comercial y viable económicamente.
- Tamaño reducido y fácil de manipular para poderlos instalar sobre la piel de un caballo.

2. Dos Xbee PRO S1 para la transmisión y recepción de datos de forma inalámbrica:

La selección del dispositivo se hizo basada en una comparación con tres módulos de transmisión de datos como se muestra en la tabla 2:

Módulo transmisión	Wifi esp8266	Xbee pro s1	Bluetooth HC06
Distancia Aproximada (Máxima)	Variable	1500 metros línea de vista	5-10 metros, máximo 25 metros línea de vista
Consumo de corriente en Trasmisión	170-300mA	140mA	40mA
Costo	\$20.000	\$150.000	\$15.000
Velocidad de Trasmisión	11,54,300 Mbps	250 Kbps	1000 Kbps

Tabla 2 *Comparación de los módulos de Radio Frecuencia*

- Permite comunicación punto a punto.
- Capacidad de transmisión de datos a distancias considerables.
- El dispositivo sea comercial y viable económicamente.
- Tamaño reducido, facilidad de manipulación.
- Velocidad de transmisión de los datos.

Una vez seleccionados los acelerómetros y los Xbee se seleccionaron los módulos de control, para el Xbee transmisor se optó por un **Arduino MEGA 2560 R3** y para el Xbee receptor se escogió el **Arduino UNO R3**.

3. El Arduino MEGA 2560 se seleccionó bajo los siguientes criterios:

- Soporte para la comunicación I2C y SPI de los acelerómetros
 - Soporte para comunicación serial con el Xbee
 - Gran número de entradas y salidas digitales para la integración con el Xbee y los acelerómetros.
 - Comercialmente conseguible y viable económicamente.
- 4. El Arduino UNO R3 se seleccionó porque soporta la comunicación serial con el Xbee y es viable económicamente.**
- 5. Entorno a las redes neuronales artificiales se optó por una red backpropagation debido a que esta es capaz de auto-adaptar los pesos de las células o neuronas de las capas ocultas para establecer una relación entre un conjunto de entradas y sus salidas correspondientes. (Hilera & José Martínez Hernando, 1995)**
- 6. Por último, se escogió MATLAB como herramienta de desarrollo de software atendiendo a estas consideraciones:**
- Fácil comunicación con los módulos de procesamiento.
 - Fácil manipulación de sus datos mediante el lenguaje de programación.
 - Velocidad de cálculo en operaciones matriciales y vectoriales

- Integración de algoritmos matemáticos para pasar del dominio del tiempo al dominio de la frecuencia (fft).

Adicionalmente se consideran otros elementos secundarios que se tienen en cuenta para hacer parte del sistema; el costo total aproximado de todos los componentes requeridos se refleja en la tabla 3:

Dispositivo	Valor
Arduino Mega 2560 + caja acrílica	\$50.000
Arduino UNO R3 + caja acrílica	\$35.000
Xbee Pro x2	\$300.000
Acelerómetro MMA7455 x2	\$33.000
Circuito Impreso x84cm2	\$30.240
Cables ide x50cm	\$5.000
Batería iones de litio Samsung 2600 mAh	\$70.000
Total Aproximado	\$523.000

Tabla 3 *Costo Total Aproximado de los Componentes Físicos*

Modelo de Prototipos. Bajo este modelo se establecen seis sub-fases para alcanzar la construcción del prototipo, algunas de estas fases suelen repetirse en varias ocasiones con el fin de ayudar al cliente a definir los requerimientos que no tuvo en cuenta al comienzo del desarrollo y a redefinir los requerimientos iniciales. A continuación, se detalla el proceso realizado en cada una de estas sub-fases con base en todos los requerimientos aprobados e implementados sobre el prototipo final.

Identificación de requerimientos. En una reunión previamente establecida con el cliente, se recogen los requisitos iniciales que debe llevar el sistema tanto en la parte de software como la de hardware. Los requerimientos mencionados por el cliente fueron:

Captura de datos. El usuario tendrá la opción de guardar el número de muestras permitidas por el software las veces que él lo desee. El sistema entrenará la RNA de mejor manera si las muestras de entrenamiento son bastantes.

Almacenamiento de Datos. El sistema ofrecerá al usuario la posibilidad de guardar la información obtenida por los sensores en la ubicación que desee. Los datos almacenados solo podrán ser los que se visualice en la tabla de datos de los sensores.

Entrenar RNA. El sistema permitirá al usuario realizar el entrenamiento con las muestras que él lo desee. Los datos almacenados solo podrán ser los que se visualice en la tabla de datos de los sensores.

Visualización de capturas. El sistema mostrará la captura de datos en la tabla de datos de los sensores o cualquier marcha que ya haya sido almacenada previamente.

Graficar Marcha. El sistema permitirá graficar los datos que se encuentren actualmente en la tabla de datos de los sensores.

Iniciar Transmisión. El sistema dará acceso para el envío habilitando la opción por medio de un botón.

Terminar Transmisión. El sistema tendrá la opción de terminar la transmisión una vez los datos hayan sido almacenados desactivando la transmisión con un botón.

Carga de Datos. El sistema tendrá la opción de cargar las marchas almacenadas anteriormente para visualizarlas o usarlas para el entrenamiento de la RNA. El sistema también tendrá la opción de cargar la los pesos de entrenamiento de la RNA en caso de que la RNA ya haya sido entrenada previamente.

Graficar error. El sistema podrá graficar el error de la RNA con respecto a las muestras de entrenamiento para que las visualice el usuario.

Almacenar pesos. El sistema tendrá la opción de almacenar la información de la RNA para que no haya la necesidad de volver a entrenarla.

Aplicar Umbral. El sistema tendrá la opción de aplicar un umbral dado por el usuario a una muestra o varias siempre y cuando compartan la información de sensor y eje.

Graficar dominio de la Frecuencia. El sistema podrá graficar la información de las marchas aplicando la FFT y el umbral (de haberse aplicado esta opción esta última).

Comprobar RNA. El sistema tendrá la opción de hacerle un testing a la RNA usando marchas que hayan sido guardadas anteriormente, que hayan sido usadas o no para el entrenamiento de la misma con la finalidad de comprobar que la RNA haya sido entrenada de forma correcta.

Diseño de Hardware y Software. Partiendo del diseño de hardware, se analizó la forma en la cual iban a ir conectados los dispositivos; se empezó con el módulo de transmisión de datos, que está compuesto por los acelerómetros, un Xbee Pro S1 y el Arduino Mega 2560. considerando una conexión mediante el puerto serial del Arduino Mega y el puerto serial del Xbee PRO S1, posteriormente se hizo la integración entre el Arduino Mega y los acelerómetros mediante la interfaz SPI, ya que esta permite una forma más eficiente de manipular de los acelerómetros por medio de las salidas digitales del Arduino. No se vio viable el uso de la interfaz I2C debido a que la selección del dispositivo del cual se desean recibir datos se hace mediante una dirección que identifica cada dispositivo, por eso al tener dos acelerómetros iguales, las direcciones de acceso al sensor también eran iguales siendo imposible posible acceder a los datos de manera correcta, adjunto a esto los acelerómetros seleccionados no permiten cambiar su dirección.

Para el módulo de recepción se eligió el mismo diseño que se definió para el módulo transmisor, se planteó una conexión entre el Xbee PRO S1 y el Arduino UNO mediante el

uso del puerto serial, pero en este caso no existe una conexión con los acelerómetros; en cambio este módulo de recepción iría conectado a una computadora que tuviera la herramienta de desarrollo MATLAB. La figura 3 representa la conexión de todos los dispositivos.

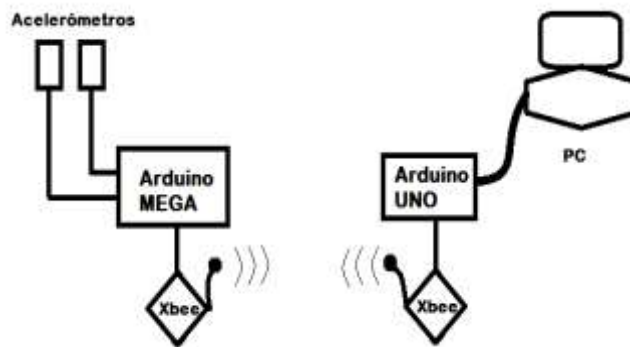


Figura 3 Conexión de los dispositivos

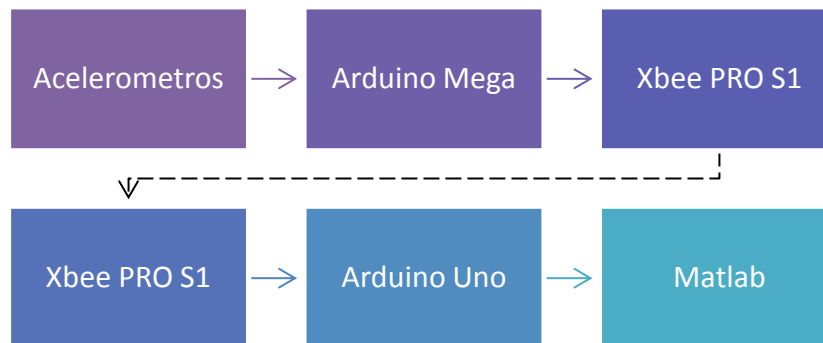


Figura 4 Flujo de datos entre dispositivos

El diseño de software parte del envío de datos desde los sensores, los datos enviados son los valores medidos por cada uno de los acelerómetros en cada uno de sus ejes separados por comas; estos se toman cada cincuenta milisegundos, es decir se envían veinte

muestras por segundo de un determinado movimiento; la información generada se manipulará desde la herramienta MATLAB. La figura 4 muestra el orden del flujo de datos entre los dispositivos.

Finalmente se trabajó en el diseño de la interfaz de usuario en MATLAB donde planteo la construcción de tres ventanas principales, en cada una de estas ventanas las funciones se separarían por paneles de acuerdo con sus funciones. La primera ventana o la ventana principal contiene tres paneles y la barra de herramientas:

Panel de transmisión de datos. Da la orden al dispositivo transmisor de cuando enviar o dejar de enviar los datos al dispositivo receptor.

Panel de captura de muestras. Donde el usuario establece el número de muestras que desea capturar, en un rango de 100 a 2000 muestras, dicho número de muestras debe ser múltiplo de 100.

Panel de visualización de datos. Una vez capturadas las muestras el usuario podrá seleccionar el movimiento o marcha que se capturo y ver los valores de cada eje del sensor mediante una tabla o una gráfica en el tiempo.

La barra de herramientas. Contiene un enlace a las otras dos ventanas que serán las de los módulos de las redes neuronales artificiales (RNAs), así como una función que permite guardar los datos de movimientos en archivos que sirvan para pasárselos como parámetros a las redes neuronales.

El flujo general de uso de la ventana principal se muestra en la figura 5.

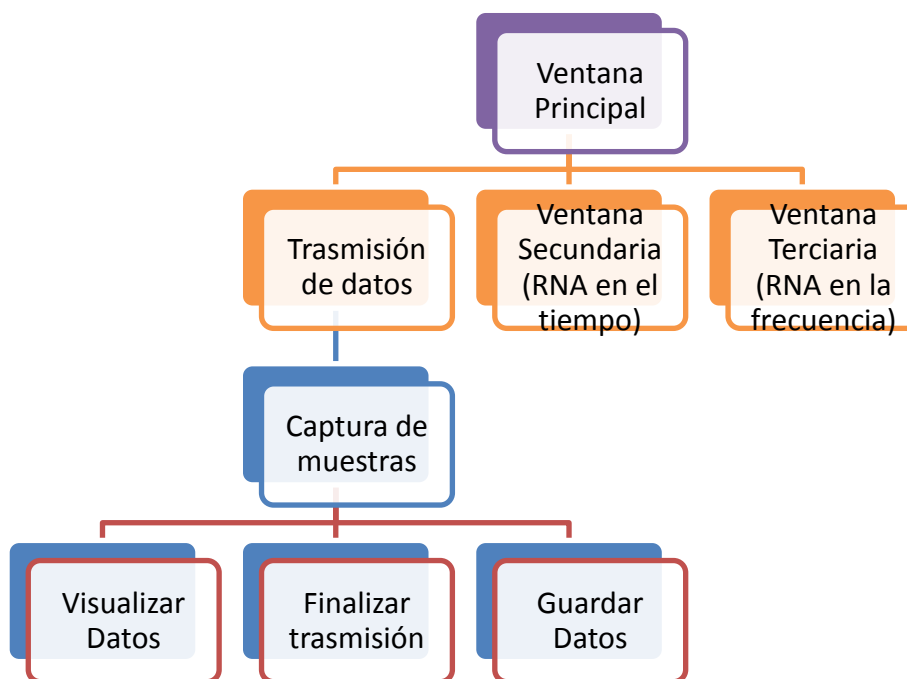


Figura 5 Flujo de uso de la ventana principal

La segunda ventana corresponde al módulo de la red neuronal artificial que entrena los datos en el tiempo; contiene cuatro paneles y una barra de herramientas.

Barra de herramientas. Desde esta barra se esperan cargar los datos de entrenamiento y los datos de prueba de la red neuronal, así como guardar y cargar los pesos de la red en archivos; dichos pesos son los parámetros de la red que realizarán el reconocimiento de los movimientos o marchas una vez entrenada la RNA. El archivo que se generará al guardar, tendrá la extensión “. **mat**” y contiene los pesos de la capa oculta y la capa de salida en dos matrices distintas. También se define que los datos que se cargan para el entrenamiento de la RNA deben ser distintos a los datos que se cargan para prueba, esto con el fin de validar el funcionamiento de la red.

Panel de datos de entrenamiento. Permite ver al usuario el número de ejemplos de entrenamiento que se cargaron para distintos movimientos o marchas, estos ejemplos o datos de entrenamiento son las muestras con las que aprenderá la red neuronal artificial. Cada 100 muestras capturadas desde el panel de captura de datos de la primera ventana corresponderá a un ejemplo de entrenamiento.

Panel de entrenamiento de la RNA. Permite el entrenamiento de la RNA además de la configuración los parámetros de la red neuronal, como el número de células o neuronas de la capa oculta y la tasa de aprendizaje y permite visualizar el error de la red en determinada época (ciclo o iteración).

Panel de prueba del entrenamiento. Una vez entrenada la RNA y cargados los datos de prueba, valida el reconocimiento de las marchas.

Panel de arquitectura de la RNA. Este panel mostrará al usuario la arquitectura o topología de la red neuronal artificial con la cual se decidió entrenar.

El flujo general de uso de la segunda ventana se muestra en la figura 6.

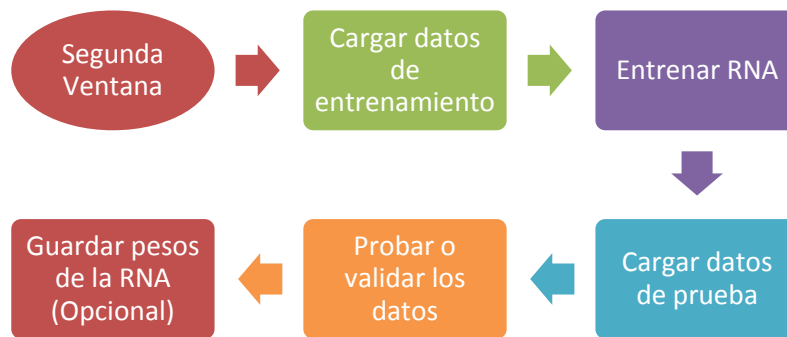


Figura 6 Flujo general de uso de la segunda ventana

La tercera ventana corresponde al módulo de la red neuronal artificial que entrena los datos en la frecuencia; el diseño propuesto para esta es el mismo diseño que se definió anteriormente para la segunda ventana, pero en este caso consta de un panel extra que mostrará mediante graficas los movimientos o marchas cargadas en el dominio de la frecuencia, además de conceder la opción de aplicar un umbral o filtro a los datos antes de pasarlos por la red neuronal con el fin de eliminar muestras con ruido. En el caso de esta interfaz independientemente del número de muestras que contengan los archivos cargados

para entrenamiento, se considera un ejemplo de entrenamiento cada archivo y no cada 100 muestras capturadas como se hacía en la red neuronal en el dominio del tiempo.

El flujo general de uso de la tercera ventana se muestra en la figura 7.

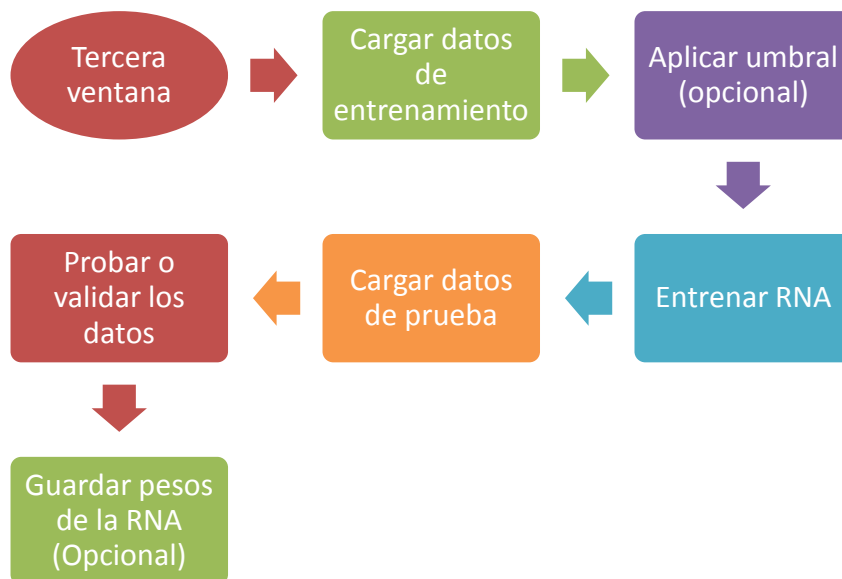


Figura 7 Flujo general de uso de la tercera ventana

Dentro del diseño de software se consideró un aspecto muy importante que fue las redes neuronales artificiales y se plantea una red tipo backpropagation de 3 capas.

Para la red neuronal en el tiempo se definió 600 entradas. El número de entradas elegido es con base a la marcha básica más lenta de un equino que es el paso, pues se pudo determinar mediante observación que cinco segundos es tiempo suficiente para capturar un ejemplo completo de la marcha. Lo anteriormente expuesto referente a la frecuencia con que se envían los datos indica que si se toman veinte muestras por segundo, para representar

cinco segundos se necesitan 100 muestras arrojadas por los sensores en cada uno de sus ejes (X, Y, Z), al tener únicamente dos sensores triaxiales, habrán 100 muestras de cada eje para un total de 600 muestras como se observa en la figura 8.

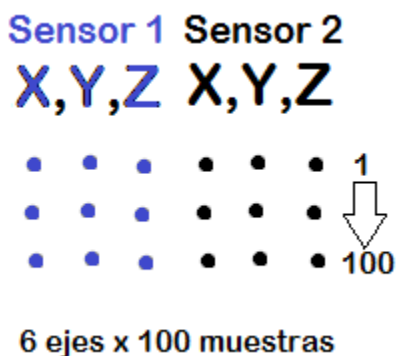


Figura 8 Entrada de la RNA en el dominio del tiempo

Para la red neuronal que entrena los datos en la frecuencia, se definieron 1542 entradas, debido a que sin importar el número de muestras que se hayan tomado de la marcha en el dominio del tiempo, estas marchas se convierten al dominio de la frecuencia a 257 muestras por eje para un total de 1542 muestras.

Con el fin de hacer el prototipo de software más dinámico el número de neuronas de la capa oculta se deja a elección de la persona que vaya a ser el usuario. En cuanto a las neuronas de salida se estableció usar el mínimo número de neuronas posibles para representar las marchas a reconocer ya que entre más neuronas se tengan, más costoso computacionalmente se volverá el entrenamiento de la red; es por eso que para la capa de salida se seleccionaron solamente dos neuronas, pues son suficientes para representar el

resultado de la red mediante salidas binarias. Se sigue la siguiente fórmula para calcular el mínimo número de células que se necesitan con base en los patrones a reconocer:

n = Número de patrones a reconocer.

$n=3$ (Paso, trote, galope)

N = Cantidad mínima de neuronas.

$R = \log_2 n = 1,58$

Si R es un número entero $N=R$; en caso contrario N es igual al siguiente número entero mayor que R , en este caso $N=2$.

Las salidas para las marchas quedaron de la siguiente manera:

- $\{1\ 1\}$ si la red reconoce la marcha del paso.
- $\{0\ 1\}$ si la red reconoce la marcha del trote.
- $\{1\ 0\}$ si la red reconoce la marcha del galope.
- $\{0\ 0\}$ o un valor diferente si no se reconoce ninguna marcha

En el caso de la red diseñada cualquier valor mayor o igual a 0,8 que arroje la red se toma como 1 y cualquier valor menor o igual a 0,2 se toma como 0.

La topología de las redes neuronales se observa en la figura 9 donde i es la cantidad de entradas; $i=600$ para la red en el dominio del tiempo, $i=1542$ para la red en el dominio de la frecuencia y n es el número de neuronas de la capa oculta ingresado por el usuario.

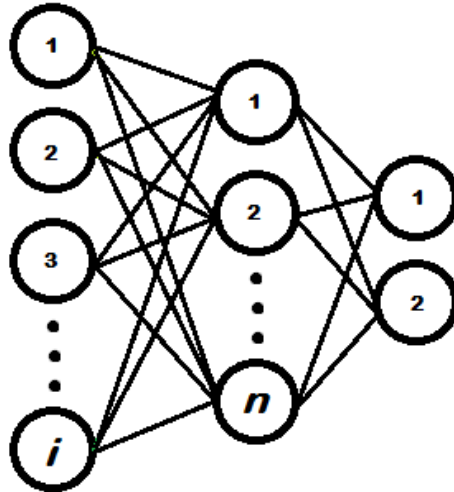


Figura 9 Arquitectura de la RNA

El entrenamiento de la red empieza con la inicialización aleatoria de los pesos, estos hacen sinapsis con las entradas de la red y producen la salida de la red, se calcula el error de todos los ejemplos de entrenamiento y se acumula, si el error acumulado es menor que 0,01 se finaliza el entrenamiento, en caso contrario, se modifican los pesos y se repite el proceso desde la sinapsis. La figura 10 muestra el flujo del entrenamiento donde **n es el número de ejemplos de entrenamiento.**

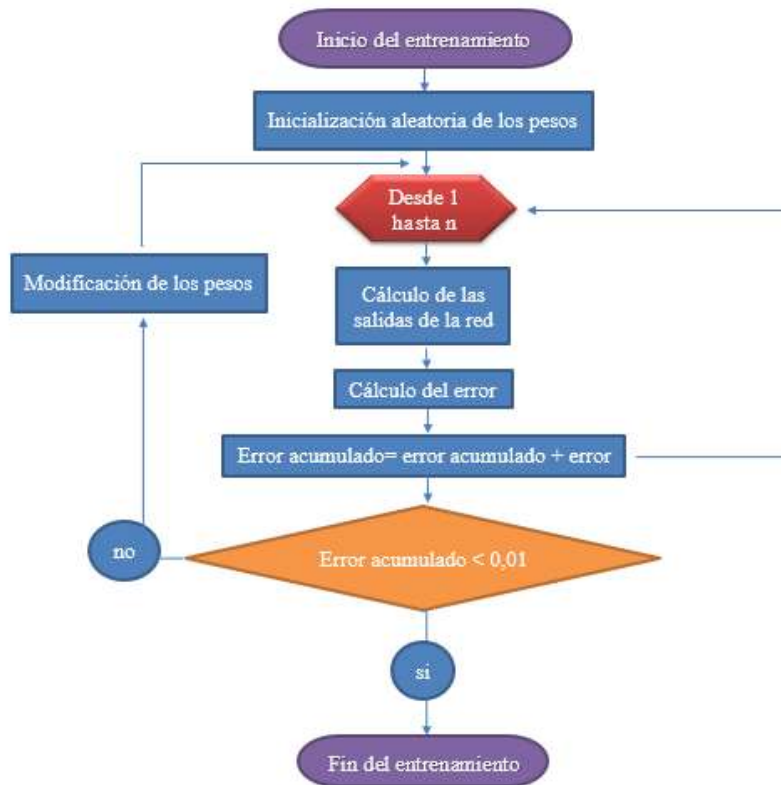


Figura 10 Flujo de entrenamiento de la RNA

Construcción del prototipo. Durante esta fase se construyó todo el sistema empezando por la parte de hardware; el montaje realizado se hizo inicialmente sobre protoboards con el fin de verificar todas las conexiones y una vez probado el funcionamiento de los dispositivos se procedió a traspasar las conexiones sobre una baquelita. Las figuras 11 a la 16 muestran algunos resultados de la construcción.

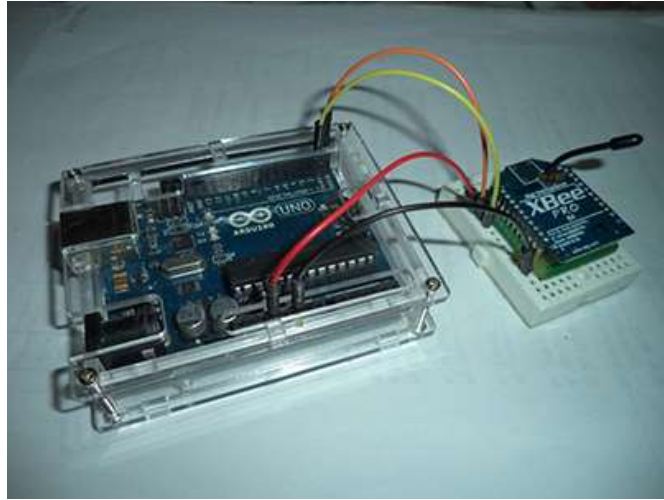


Figura 11 Montaje Xbee sobre protoboard

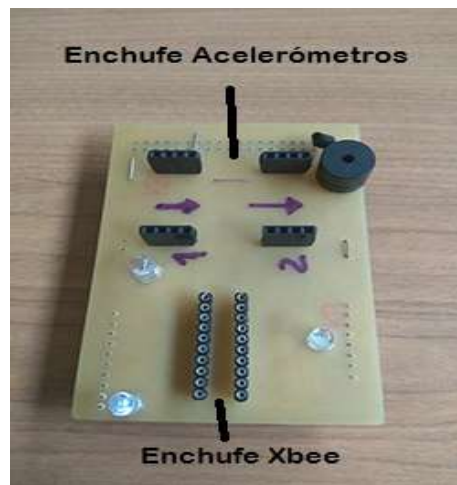


Figura 12 Baquelita módulo transmisor

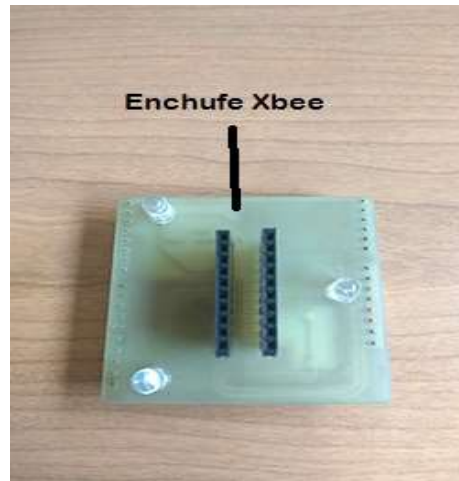


Figura 13 Baquelita módulo receptor



Figura 14 Módulo transmisor



Figura 15 Módulo receptor



Figura 16 Acelerómetros con extensiones

El módulo transmisor mostrado en la figura 14 fue programado para tomar los datos de los sensores accediendo a cada uno de los registros de los ejes en donde se guardan los valores medidos y se envían al Xbee que se encarga enviarlos al módulo receptor. En el

módulo receptor, el Xbee recibe los datos y los envía al Arduino UNO que posteriormente los envía a Matlab.

En la parte de software se fueron construyendo, probando e implementando varias funciones usando programación estructurada y haciendo uso de algunas de las funciones integradas de MATLAB como por ejemplo la fft (fast Fourier transform) que permite pasar del dominio del tiempo al dominio de la frecuencia para poder hacer un análisis frecuencial de las marchas capturadas. Tal como se estipuló en la fase de diseño el prototipo de software se constituye de tres ventanas como se muestra en las figuras 17 a 19.



Figura 17 Ventana principal

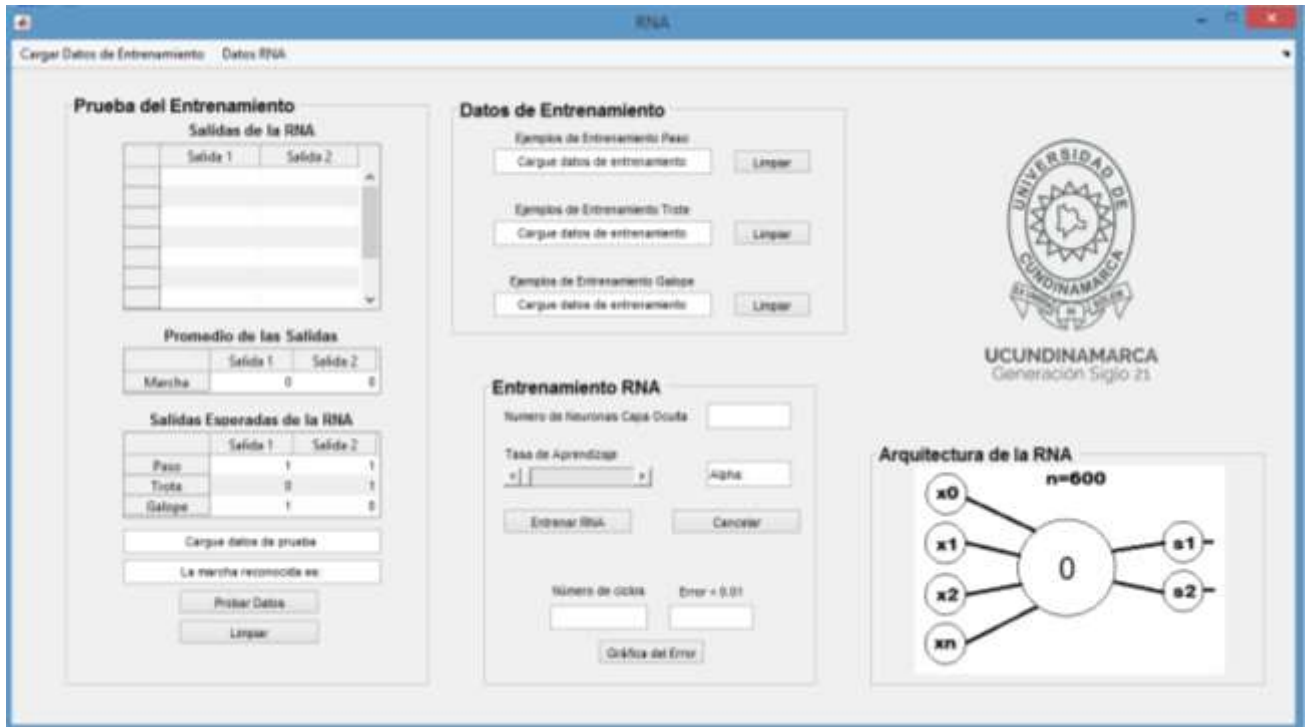


Figura 18 Segunda ventana

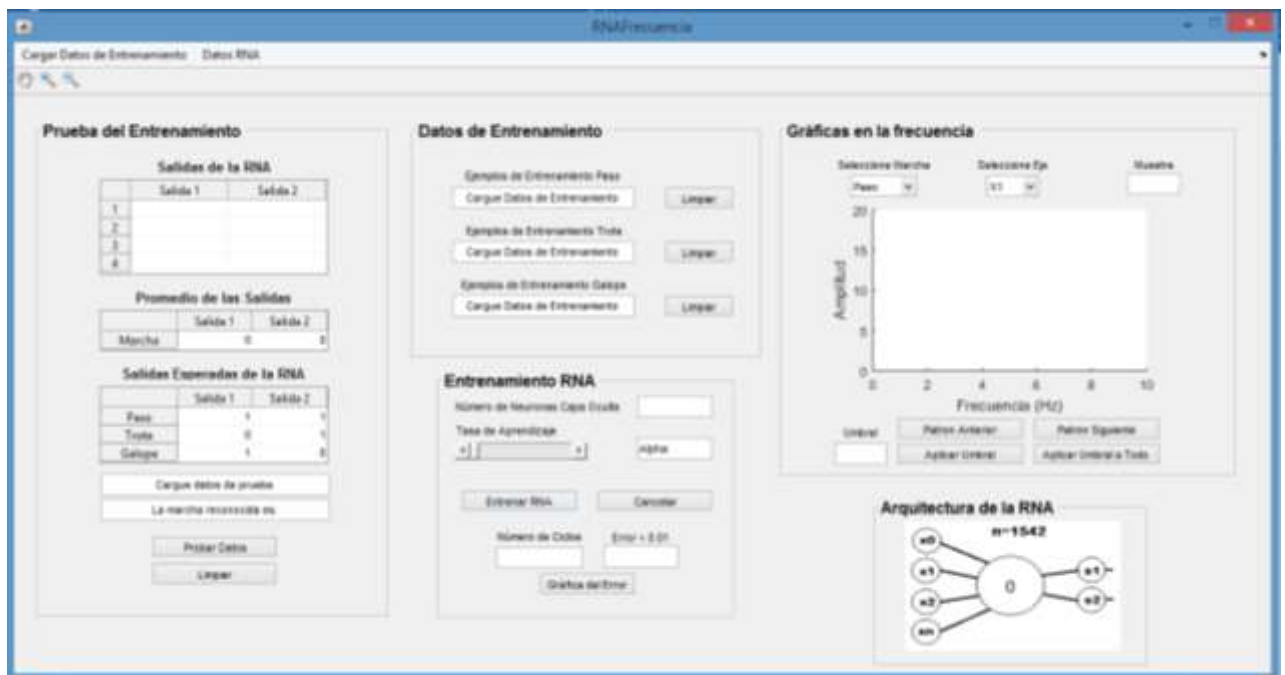


Figura 19 Tercera ventana

Evaluación del cliente. El cliente hace una evaluación del prototipo de software y hardware interactuando con todo el sistema.

Refinamiento del prototipo. Durante esta fase el cliente después de evaluar el sistema hace observaciones, correcciones y define nuevos requerimientos que no tuvo en cuenta al concretar los requerimientos iniciales, consecutivamente se vuelve a pasar a la fase de diseño de hardware y software. En caso de que no haya más observaciones, correcciones o requerimientos se pasa a la última fase.

Producto terminado. El cliente está de acuerdo con todo el desarrollo.

Durante toda la fase de aplicación del modelo de prototipos se hicieron varias reuniones con el cliente con el fin de redefinir los requerimientos, las fases de refinamiento de requerimiento, el diseño y la implementación de hardware y software que se acabaron de presentar nacieron de varias revisiones con el cliente. En los anexos A y B se muestra los resultados de los prototipos de software diseñados y desarrollados de forma rápida durante esta fase; el anexo A muestra el primer prototipo entregado al cliente donde se muestra los paneles que tienen que ver con la transmisión de datos y el panel de la red neuronal con los códigos de salidas esperadas por la red, el anexo B muestra la segunda entrega realizada al cliente donde se visualiza una nueva ventana que es la de la red neuronal en el dominio del tiempo que fue separada de la ventana que contiene los paneles transmisión de datos, la tercera entrega realizada corresponden a las ventanas mostradas en las figuras 17 a la 19.

Validación del prototipo. En esta fase del proyecto se procedió a validar toda la construcción del prototipo de hardware y software mediante pruebas que se hicieron sobre un robot equino de pequeña escala, sobre este robot se programaron unas secuencias aproximadas de las marchas paso trote y galope teniendo en cuenta los movimientos de un equino real; subsiguientemente se procedió a conectar los acelerómetros al lomo del robot con el fin de realizar una captura de los datos de dichas secuencias programadas como se muestra en la figura 20 y 21.

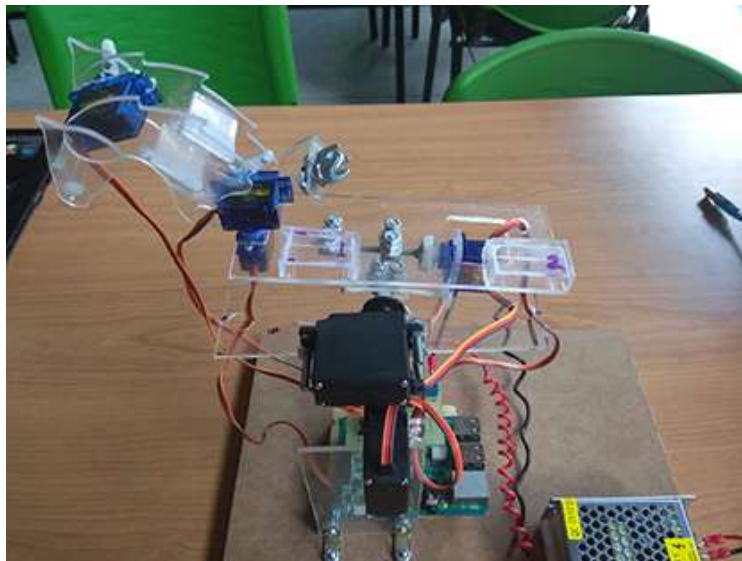


Figura 20 Robot Equino

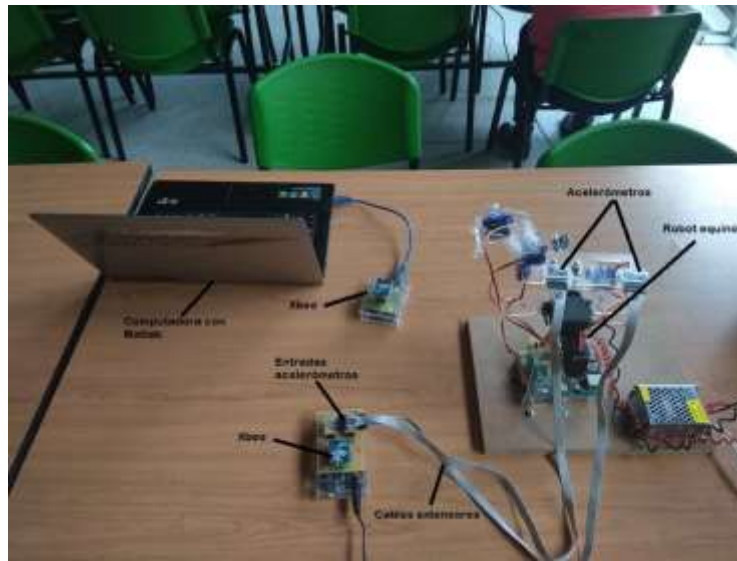


Figura 21 Prototipo completo

Se hicieron tres sesiones de captura de datos distintas, en cada sesión de captura se programaron distintas marchas para el paso, el trote y el galope en el robot y se capturaron distinto número de muestras, así como también se hicieron a distintas distancias entre el módulo transmisor y el módulo receptor, la Tabla 4 presenta la información referente a las sesiones de captura.

Tabla 4. *Sesiones de Captura*

Sesiones	Nº muestras capturadas por marcha	Distancia Aproximada (metros)
Primera sesión	10000	1,5
Segunda sesión	8000	8
Tercera sesión	20000	25

Una vez validado el módulo de captura de datos, con las muestras capturadas se procedió a validar las redes neuronales, para esto se comenzó con la red neuronal en el dominio del tiempo entrenándola con diferente número de neuronas en la capa oculta, diferente porcentaje de las muestras capturadas y diferente tasa de aprendizaje como se muestra en las siguientes tablas 5 y 6:

Tabla 5. Sesiones de entrenamiento en el tiempo basados en las capturas 1 y 3

Porcentaje de muestras capturadas de cada marcha (%)	Numero de neuronas en la capa oculta	Tasa de aprendizaje
20	30	0.1
		0.5
		1
	60	0.1
		0.5
		1
	100	0.1
		0.5
		1
60	30	0.1
		0.5
		1
	60	0.1
		0.5
		1
	100	0.1
		0.5
		1

Tabla 6 *Sesiones de entrenamiento en el tiempo basados en la captura 2*

Porcentaje de muestras capturadas de cada marcha (%)	Numero de neuronas en la capa oculta	Tasa de aprendizaje
25	30	0.1
		0.5
		1
	60	0.1
		0.5
		1
	100	0.1
		0.5
		1
62.5	30	0.1
		0.5
		1
	60	0.1
		0.5
		1
	100	0.1
		0.5
		1

Las tablas 7 y 8 muestra cómo se realizó la validación para la red neuronal en el dominio de la frecuencia:

Tabla 7. *Sesiones de entrenamiento en frecuencia basadas en las capturas 1 y 3*

Porcentaje de muestras capturadas de cada marcha (%)	Numero de neuronas en la capa oculta	Tasa de aprendizaje
20	5	0.1
		0.5
		1
	15	0.1
		0.5
		1
	30	0.1
		0.5
		1
60	5	0.1
		0.5
		1
	15	0.1
		0.5
		1
	30	0.1
		0.5
		1

Tabla 8. *Sesiones de entrenamiento en frecuencia basadas en la captura 2*

Porcentaje de muestras capturadas de cada marcha (%)	Numero de neuronas en la capa oculta	Tasa de aprendizaje
25	5	0.1
		0.5
		1
	15	0.1
		0.5
		1
	30	0.1
		0.5
		1
62.5	5	0.1
		0.5
		1
	15	0.1
		0.5
		1
	30	0.1
		0.5
		1

Impacto del Proyecto

Poblacional en el ámbito policial y académico. El software de control de un simulador de tiro montado se presenta como una alternativa de gran valor para mejorar y hacer eficiente el proceso de enseñanza-aprendizaje en las prácticas de tiro montado con el personal policial, además de incentivar la productividad investigativa de estudiantes, directores, jurados y profesores de las instituciones vinculados al proyecto y en especial fomentar la cultura investigativa del grupo ESCAR y GITSFA de la UDEC, ya que se demuestra técnicamente como se pueden recoger y aproximar los movimientos del equino de una manera apropiada, haciendo uso de técnicas electrónicas, mecánicas y de programación en software.

En Innovaciones tecnológicas. Posibilidad de aplicación de los resultados a través de:

Modelo de utilidad: Del prototipo automatizado del software de control de un simulador de tiro montado, junto con la captura y registro de los movimientos del semoviente en el cual se incorporan diseños de hardware y firmware que permiten reducir costos en la producción del sistema.

Derechos de autor sobre hardware, firmware y software elaborado para la simulación e implementación de la automatización del simulador en general.

Marcos de Referencia

Marco Histórico

A continuación, se presentan algunos software's que presentan parte de la evolución de programas que capturan y analizan movimiento.

Son programas especializados que permiten por medio de una videograbación analizar el movimiento de cualquier destreza deportiva. Los estudios biomecánicos pueden hacerse por observación directa o por indirecta, en el primer caso el método solo requiere de una videograbación y una serie de pasos para descomponer la destreza y poder analizarla, mientras que en el segundo caso, existen en el mercado distintos software para el procesamiento y análisis de datos de mediano y alto costo que permiten mediante de un video o videograbación de la ejecución hacer un estudio más objetivo y específico del comportamiento mecánico del cuerpo.

No obstante, cada software tiene particularidades en el uso y además la capacidad para aportar resultados en variables biomecánicas, tal es el caso de los software libre que sus funciones y usos son básicas permitiendo solo medir variables cinemáticas en 2D, mientras que las funciones de software propietarios son más completas, permitiendo medir variables tanto cinemáticas como dinámicas en 2D y 3D. Como se mencionó anteriormente, para poder utilizar estos software es necesario un video para evaluaciones en 2D y dos o más videos para estudios en 3D, cada video debe pasar por un proceso llamado digitalización, que no es más que el hallazgo y marcado de puntos sobre cada imagen de vídeo de un modelo preestablecido con el fin de adquirir las

coordenadas de los puntos anatómicos en dos y hasta en tres dimensiones (x;y e x;y;z, respectivamente) especificados en la imagen del video.

Sin embargo, existen variedades de software que no requieren de la digitalización de la secuencia del movimiento completamente y ofrecen cálculos que se pueden hacer instantáneamente y otros que permiten hacer una captura automática de las coordenadas de los puntos anatómicos tan solo con la ayuda de marcadores reflectantes.

Dentro del software gratuito que se pueden adquirir por las páginas web se sus autores tenemos:

1. Abiomo: Software desarrollado por los profesores Antonio Hernández y Gustavo Velazco de la Universidad de los Andes, es una herramienta sencilla que permite capturar las coordenadas de los puntos del modelo anatómico con el uso “Paint” por lo que se debe descomponer previamente el video en fotogramas. Una vez obtenidas las coordenadas en x e y se procede a vaciar la información dentro del software “Abiomo” y finalmente obtener resultados cinemáticos como centro de gravedad, distancias, velocidades y ángulos de posición y proyección. Además permite capturar gráficos como el ciclograma de la ejecución del movimiento.
2. Kinovea: Este es otro software libre disponibles en la web, presenta diferentes herramientas para captura de video de una y dos cámaras a la vez, también tiene la opción de subir un video guardado en la pc y editarlo dentro del mismo programa, entre las diferentes herramientas que presenta hace que su uso sea muy productivo ya que no requiere de digitalización por lo que los análisis son directos sobre la o

las imágenes que se quieren analizar lo que hace que el proceso sea muy expedito, sin embargo este software permite medir solo variables básicas como desplazamientos, trayectorias, tiempo, velocidades y ángulos.

3. SkillSpector: Es otro de los software libre, pero que a diferencia de los anteriores solo se presenta en el idioma inglés. Este programa permite análisis en 2D y 3D mediante la digitalización semi automática y permite calcular variables cinemáticas tanto lineales como angulares. Para los estudios en 3D se utilizan solo dos videos que pueden ser tomados directamente desde el programa u obtenerlos del disco duro de la computadora, entre otras herramientas más.

Por otro lado, encontramos aquellos software con licencia paga como lo es el Hu-man, éste se puede decir que dentro de los software con licencia paga es de bajo costo.

¿Qué hace el Hu-man? - Es un software que fue diseñado por el profesor Tom Duck de la escuela de kinesiología de la Universidad de York, Canadá. El autor desarrolló dicho programa bajo un enfoque de enseñanza - aprendizaje con el cual se podría aprender enseñando y viceversa, por lo que se basó en dos cursos independientes para aprender por un lado a utilizar todas las herramientas del programa y por el otro obtener la información para analizar la mecánica del movimiento durante la ejecución de destrezas físicas y deportivas mediante un video grabado en formato "avi". El programa cuenta con una gran disponibilidad de herramientas para hacer estudios en 2D y 3D tanto en variables cinemáticas como dinámicas.

Este fue creado con un enfoque didáctico por lo que principalmente se utiliza para obtener resultados biomecánicos, específicamente en destrezas deportivas, no obstante

no solo se limita al deporte sino que también tiene aplicaciones en el ámbito clínico y ergonómico, por ende la información obtenida en el software sirve para; 1) prevenir lesiones en el aparato locomotor, 2) mejorar la técnica deportiva y 3) mejorar la calidad de vida.

Este programa brinda la posibilidad de realizar digitalizaciones manuales y automáticas o cálculos manuales instantáneos para poder obtener resultados preliminares si así se quisiera. En cuanto a las digitalizaciones manuales permite crear un modelo anatómico propio o preestablecido en el sistema y de allí comenzar la digitalización fotograma por fotograma hasta completar toda la ejecución de la destreza (Katusca Vielma, 2014).

Otro software que también se puede encontrar es:

El software de análisis de movimiento TEMA (Track Eye Motion Analysis) de IMAGE SYSTEMS es un sistema completo para el análisis avanzado de imágenes de alta velocidad. Es capaz de manejar y analizar grandes cantidades de datos procedentes de cinematografía, vídeo digital u otro sensor de alta velocidad. Dispone de algoritmos específicos de seguimiento o tracking de acuerdo a los estándares actuales (para dianas de tipo QUAD, circulares, MXT etc.), así como de módulos específicos para el análisis de airbags (tracking del contorno, cálculo de áreas y puntos extremos etc.), así como análisis avanzados en 3D y 6D.

Las aplicaciones son numerosas: el software TEMA está presente en nuestro país, a través de Alava Ingenieros, en laboratorios de crash, universidades, centros de biomecánica, instalaciones militares e industria del automóvil (ÁLAVA INGENIEROS ©, 2017).

Sus principales características son:

- Lectura de imágenes de cualquier modelo de cámara de alta velocidad digital existente en el mercado.
- Interpolación de puntos ocultos.
- Lectura de película de cinematografía y vídeo analógico digitalizado.
- Definición libre de sistemas de coordenadas.
- Número ilimitado de puntos de seguimiento (tracking).
- Conversión de formato Bayer a AVI Algoritmos específicos: Correlación, Circular, Centro de gravedad, MXT opcional.
- Escalado y corrección estática y dinámica de profundidad.
- Algoritmo exclusivo QUAD de extraordinaria precisión.
- Filtrados CFC/FIR.
- Puntos virtuales y visualización de contornos.
- Interfaz gráfica y de texto para la generación de informes.
- Módulos específicos de airbag, 3D y 6D opcionales.
- Importación/exportación de AVI, TIFF, MPEG y otros.
- Lectura automática de códigos de tiempos opcional.
- Re-muestreo de tiempos
- Importación/exportación de datos de/a DIAdem y otras herramientas estándar (Excel etc.).
- Cálculo de posición, distancia, ángulo, velocidad y aceleración

Marco Teórico

Acelerómetro. Dentro de la instrumentación usada para detectar los cambios de velocidad en un intervalo de tiempo existe un sensor denominado acelerómetro. Según Cañada et al., (2015) los acelerómetros “son pequeños instrumentos electrónicos que miden la magnitud de los cambios de la aceleración del centro de masas del cuerpo durante el movimiento”, cuyo resultado o el valor entregado es una unidad denominada “counts”; el nivel de aceleración respecto al movimiento se registra en cada uno de los tres ejes de referencia. Comercialmente se pueden encontrar varios tipos de acelerómetros como los uniaxiales que registran la aceleración en un solo eje que suele ser el vertical, biaxiales, que miden la aceleración en dos ejes y los triaxiales que registran la aceleración en 3 ejes por separado, así como un valor ponderado de estos.

Acelerómetro MMA 7455. El acelerómetro MMA 7455 es un sensor micro mecanizado de baja potencia de salida digital que es capaz de medir la aceleración a lo largo de sus ejes (X, Y, Z). Este ofrece múltiples modos de configuración entre los cuales se encuentra la selección para múltiples rangos de sensibilidad. (Parallax, 2009)

Tiene un regulador de voltaje que lo puede hacer operar con un voltaje en un rango de 2.5 VDC a 5.5 VDC y posee dos interfaces de comunicación que son I2C (Circuito interintegrado) y SPI (Interfaz Periférica en Serie) (Parallax, 2009).

Según Parallax (2009) este sensor es un sistema micro electromecánico formado por materiales semiconductores, que se puede visualizar como una pequeña masa suspendida en el centro del dispositivo por diminutos resortes, esto hace que cuando una aceleración se le aplique al dispositivo, la masa se desvía a lo largo de uno o más ejes dentro del dispositivo y los circuitos integrados miden constantemente la cantidad de deflexión a lo largo de los ejes y los convierte en datos de aceleración. La figura 22 muestra cómo sería el interior de un acelerómetro.



Figura 22 Acelerómetro con tres ejes Fuente: (Sheldon, 2014)

Es posible encontrar que el sensor convierte la aceleración detectada en un número que puede ser fácilmente leído por un microcontrolador externo, este número generado se denomina counts y depende del modo de sensibilidad seleccionado. El número de counts es determinado por el número de bits especificados para los datos; puede ser 8 bits (256 counts) y 10 bits (1024) counts. (Parallax, 2009). La figura 23 muestra el acelerómetro MMA7455.

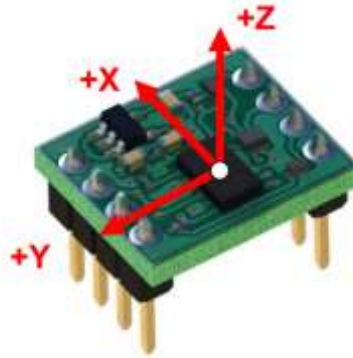


Figura 23 Acelerómetro triaxial MMA7455 Fuente: (Parallax,2009)

Xbee PRO S1. De acuerdo con la marca fabricante, el xbee es un módulo de radiofrecuencia diseñado para cumplir con el estándar IEEE 802.15.4 que opera dentro de la banda de 2.4Ghz, tiene un alcance en línea de vista de 1335 metros, pero para interior en edificaciones es de 43 metros esto mediante el uso de una antena tipo whip. (Digi, n.d.).

Los módulos de radiofrecuencia se comunican mediante una interfaz a través del puerto serial que puede ser un traductor de voltaje de algún dispositivo a través del estándar RS-232 o una interfaz USB como también un transmisor-receptor asíncrono universal (UART). (Digi, s.f.) La figura 24 muestra un Xbee PRO S1.



Figura 24 Módulo de radio frecuencia Xbee PRO SI

Redes Neuronales Artificiales. Las redes neuronales artificiales son una forma de emular ciertas características del ser humano, como la capacidad de reconocer o memorizar hechos; se puede observar que estos problemas no se pueden solucionar con un algoritmo corriente ya que la solución para ese tipo de problemas se debe tener un rasgo en común que es la experiencia, es por eso que una manera de dar una solución aproximada es mediante el diseño de un sistema que permita emular estas características humanas. (Matich, 2001)

Por eso una red neuronal es tan solo un modelo artificial del cerebro humano, construido de forma simple, debido a que el cerebro es el sistema por el cual se aprende y se obtiene experiencia y aprendizaje, el termino aprendizaje da a entender que cualquier problema relacionado con este, inicialmente puede resolverse solo hasta cuando se haya obtenido información acerca del problema, se puede decir que la red neuronal:

- Consta de unidades de procesamiento que intercambian datos o información.

- Se utiliza para reconocer patrones, incluyendo imágenes, manuscritos y secuencias de tiempo (por ejemplo: tendencias financieras).
- Tienen capacidad de aprender y mejorar su funcionamiento. (Matich, 2001)

La unidad fundamental o pieza básica de una red neuronal es la neurona; las neuronas se organizan en capas y varias de ellas conforman la red neuronal, y esta red a su vez puede tener millones de neuronas; la red es definida como un subsistema con funcionalidad propia, y todo un conjunto de subsistemas conforma el sistema global que para el caso de las redes biológicas es el sistema nervioso. Para el caso de la construcción de redes neuronales artificiales, se establece una estructura análoga; el componente básico es la neurona artificial, que pueden estar organizadas mediante capas y varias capas de estas constituyen toda la red, de igual modo como en las redes biológicas, un conjunto de redes neuronales artificiales junto con interfaces de entrada y salida y algunos módulos extra constituyen un sistema global. (Martín del Brio & Serrano Cinca, 1995)

La neurona como procesador elemental está constituida por unos elementos básicos como:

- Un conjunto de entradas y pesos sinápticos

- Una regla de propagación
- Una función de activación que proporciona una salida. (Martín del Brio & Serrano Cinca, 1995)

Para Hilera & José Martínez Hernando (1995), una neurona biológica posee cualidades que no se encuentran en otras células vivas como lo es la capacidad de comunicarse, esta se compone de un cuerpo celular, un conjunto de ramas cortas denominadas dendritas y una rama principal denominada axón; las dendritas y el cuerpo celular reciben señales de entrada que se combinan e integran y producen una señal de salida que es transmitida mediante el axón y recibida por otras neuronas. La figura 25 muestra el esquema de una neurona biológica.

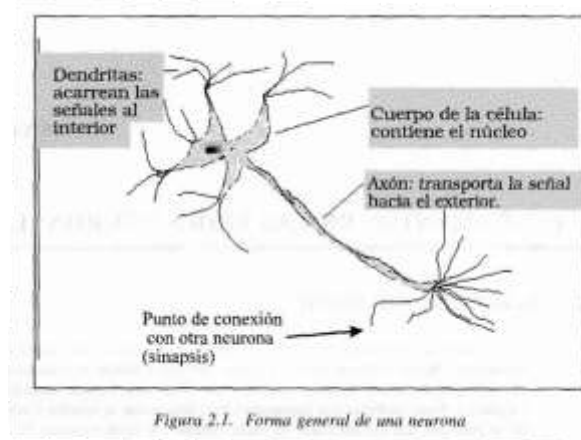


Figura 25 Neurona biológica **Fuente:** Redes neuronales artificiales: fundamentos, modelos y aplicaciones. (Hilera & José Martínez Hernando, 1995)

Una neurona artificial procura replicar las características de una neurona biológica cuyas propiedades son un estado de activación (a) asociado a la neurona, una función de

salida (f) que convierte dicho estado de activación de una señal de salida (y) que posteriormente se modifica cuando hace sinapsis con el peso de otras neuronas de la red; todas las señales modificadas que entran a una neurona se convierten en una entrada total (net). (Hilera & José Martínez Hernando, 1995). La figura 26 muestra el esquema de una neurona artificial.

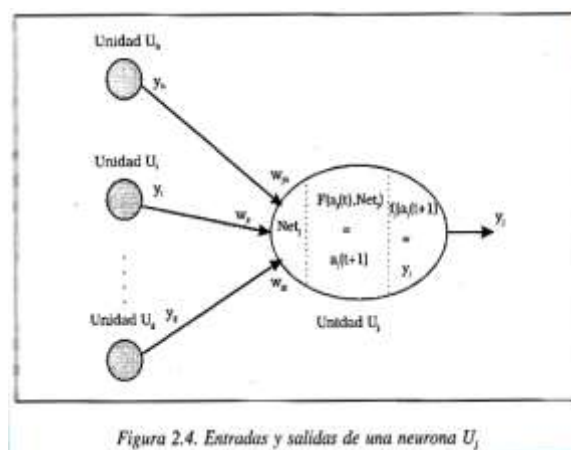


Figura 2.4. Entradas y salidas de una neurona U_j

Figura 26 Neurona artificial. **Fuente:** *Redes neuronales artificiales: fundamentos, modelos y aplicaciones.* (Hilera & José Martínez Hernando, 1995)

Hilera & José Martínez Hernando (1995) también afirman que una red se forma con un determinado número de neuronas formando diferentes niveles de capas, diferenciando tres tipos de capas, la capa de entrada que es la encargada de recibir la información externa a la red, las capas ocultas que pueden estar un rango entre cero o más capas y no tienen nada que ver con el exterior de la red y la capa de salida que arroja la información al exterior. En la figura 27 se muestra la arquitectura o topología de una red neuronal artificial.

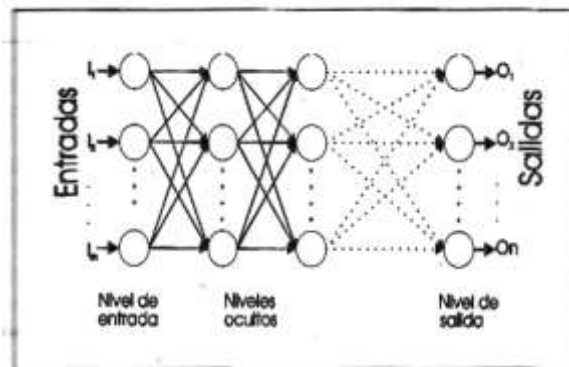


Figura 2.17. Estructura de una red multinivel con todas las conexiones hacia adelante

Figura 27 Arquitectura red neuronal artificial **Fuente:** Redes neuronales artificiales: fundamentos, modelos y aplicaciones. (Hilera & José Martínez Hernando, 1995)

Cuando las neuronas de una red no es entrada de ninguna de otra neurona del mismo nivel o de niveles anteriores se refiere a una red de propagación hacia adelante o en caso contrario cuando una neurona se conecta a neuronas de una capa anterior, de la misma capa o a ella misma, la red es de propagación hacia atrás. (Hilera & José Martínez Hernando, 1995). En la figura 28 se muestra una red con propagación hacia atrás.

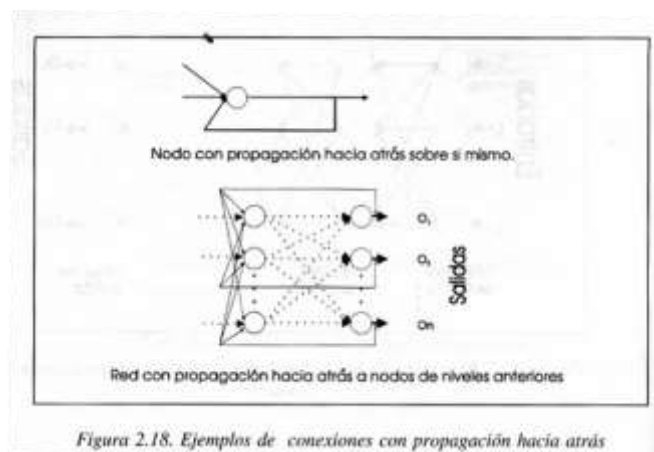


Figura 2.18. Ejemplos de conexiones con propagación hacia atrás

Figura 28 Nodo y red con propagación hacia atrás **Fuente:** Redes neuronales artificiales: fundamentos, modelos y aplicaciones. (Hilera & José Martínez Hernando, 1995)

La arquitectura o topología de una red neuronal artificial tiene que ver en cómo están organizadas las neuronas para formar las diferentes capas de la misma; si se realiza una clasificación de la red en términos topológicos se pueden encontrar redes con un solo nivel o capa, o redes multicapa que constan de dos más capas. (Hilera & José Martínez Hernando, 1995)

Backpropagation. Existe un algoritmo para el entrenamiento de la redes neuronales artificiales de más de dos capas con un modelo conexionista hacia adelante conocido como backpropagation o propagación hacia atrás; el algoritmo se compone de dos partes, en donde la primera parte existe un grupo de entradas que se le presenta a una red neuronal inicializada con pesos aleatorios para que genere una salida, en la cual las salidas de la red se comparan con las salidas esperadas para calcular la diferencia entre ellas o el error; este proceso se repite para el número de ejemplos con el que se vaya a entrenar la red con el objetivo de ir acumulando el error de todos los ejemplos de entrenamiento. (Rairán-Antolines, Chiquiza-Quiroga, & Parra-Pachón, 2012)

En la segunda parte, se parte con base en el error acumulado y se empiezan a cambiar los pesos de las conexiones en la capa de salida con fin de disminuir el error, lo cual también se hace para los pesos de las neuronas de las demás capas. Al finalizar las dos partes del algoritmo se dice que se ha terminado una época; este procedimiento se repite las épocas que sea necesario hasta que el error se encuentre en un rango aceptable. (Rairán-Antolines et al., 2012). La figura 29 muestra el proceso de una red backpropagation.

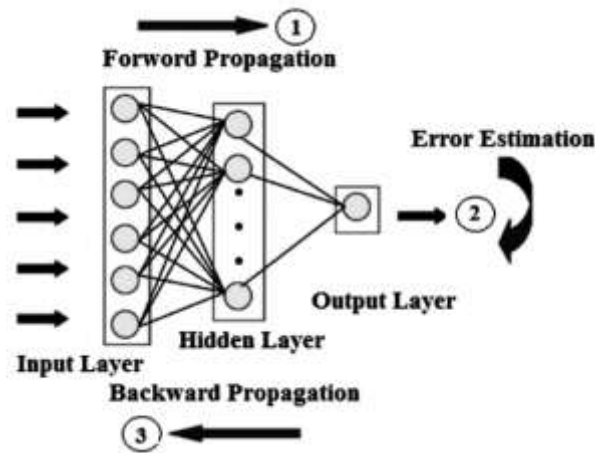


Figura 29 Proceso backpropagation **Fuente:** *The Prediction of Undersaturated Crude Oil Viscosity: An Artificial Neural Network and Fuzzy Model Approach* (Abedini, Esfandyari, Nezhadmoghadam, & Rahmanian, 2012)

Arduino. Siendo una plataforma electrónica de código abierto que está basada en la facilidad de uso la board arduino está diseñada para leer entradas de diferentes sensores o actuadores y convertirlas en salidas, esto mediante el envío de instrucciones al controlador sobre la board.

Para hacer uso de la placa se ofrecen varias herramientas de apoyo como el IDE multiplataforma basado en processing y un lenguaje de programación basado en wiring, además de un conjunto de librerías que permiten una fácil conexión a diferentes dispositivos, sensores, módulos, actuadores. (Arduino, 2017)

Arduino Mega 2560. Existen ordenadores de placa reducida como el Arduino Mega revisión número 3, que son diseñados para proyectos que requieren una mayor capacidad

de cálculo y/o almacenamiento y se suelen usar en proyectos de impresoras 3d o de robótica. Este tipo de arduino entre sus características más destacables cuenta con cincuenta y cuatro entradas y/o salidas digitales de las cuales quince pueden ser usadas como salidas pwm, dieciséis entradas análogas, cuatro puertos seriales, un cristal oscilador de dieciséis mega Hertz, además de usar diferentes tipos de interfaz de comunicación como SPI e I2C. (Arduino, 2017). La figura 30 muestra un arduino mega 2560 en una caja de acrílico.



Figura 30 Arduino Mega 2560

Matlab. Como lenguaje de alto nivel y entorno interactivo Matlab es usado por millones de ingenieros y científicos en el mundo. Matlab es la abreviación de “matrix laboratory” que en español es laboratorio de matrices. El lenguaje basado en matrices es una forma fácil de expresar matemáticas computacionales. (Matlab, 2016)

Se diferencia de otros lenguajes de programación que generalmente trabajan con un número a la vez, mientras que este opera matrices enteras y vectores; entre el funcionamiento fundamental del lenguaje abarca operaciones básicas como la creación de variables, indexación de vectores, aritmética y tipos de datos. (Matlab, 2016)

Entre las funciones matemáticas, provee distintos métodos de computación numérica y las principales funciones son librerías optimizadas por procesador para el rápido cálculo de operaciones con vectores y matrices. (Matlab, 2016)

Marco Legal

De acuerdo a la constitución, una de las leyes que aprueban la investigación en el área informática, es el artículo 361, en el que a continuación se muestra algunas anotaciones importantes de dicho artículo.

ARTICULO 361. (Modificado por el art 2º, Acto Legislativo 005 de 2011). Los ingresos del Sistema General de Regalías se destinarán al financiamiento de proyectos para el desarrollo social, económico y ambiental de las entidades territoriales; al ahorro para su pasivo pensional; para inversiones físicas en educación, para inversiones en ciencia, tecnología e innovación; para la generación de ahorro público; para la fiscalización de la exploración y explotación de los yacimientos y conocimiento y

cartografía geológica del subsuelo; y para aumentar la competitividad general de la economía buscando mejorar las condiciones sociales de la población.

Para efectos de cumplir con los objetivos y fines del Sistema General de Regalías, créanse los Fondos de Ciencia, Tecnología e Innovación; de Desarrollo Regional; de Compensación Regional; y de Ahorro y Estabilización.

La diferencia entre el total de los ingresos del Sistema General de Regalías y los recursos destinados al ahorro pensional territorial, al Fondo de Ciencia, Tecnología e Innovación, al Fondo de Desarrollo Regional, al Fondo de Compensación Regional, así como a los que se refiere el inciso 2° del presente artículo se destinará al Fondo de Ahorro y Estabilización.

Los Fondos de Ciencia, Tecnología e Innovación y de Desarrollo Regional tendrán como finalidad la financiación de proyectos regionales acordados entre las entidades territoriales y el Gobierno Nacional.

Los programas y/o proyectos en ciencia tecnología e innovación de los departamentos, municipios y distritos que se financiarán con los recursos del Fondo de Ciencia, Tecnología e Innovación, se definirán por un órgano colegiado de administración y decisión, en el cual tendrán asiento el Gobierno Nacional, representado por tres (3) Ministros o sus delegados, un (1) representante del Organismo Nacional de Planeación y un (1) representante del Organismo Nacional encargado del manejo de la política pública de ciencia y tecnología e innovación, quien además ejercerá la Secretaría Técnica, un (1) Gobernador por cada una de las instancias de planeación regional a que se refiere el inciso siguiente del presente artículo; cuatro (4)

representantes de las universidades públicas y dos (2) representantes de universidades privadas. Así mismo, los recursos de este Fondo de Ciencia, Tecnología e Innovación, se distribuirán en la misma proporción en que se distribuyan a los departamentos, los recursos de los Fondos de Compensación Regional y de Desarrollo Regional. En ningún caso los recursos de este fondo podrán financiar gasto corriente.

Los proyectos de impacto regional de los departamentos, municipios y distritos que se financiarán con los recursos de los Fondos de Desarrollo y Compensación Regional se definirán a través de ejercicios de planeación regional por órganos colegiados de administración y decisión donde tengan asiento cuatro (4) Ministros o sus delegados y un (1) representante del Organismo Nacional de Planeación, los gobernadores respectivos o sus delegados y un número representativo de alcaldes. (Colombia, 1991)

Por ultimo tenemos algunos decretos a fines correspondientes al área de investigación:

- Decreto 1510 del 2013

ART. 78. Contratación de Bienes y Servicios en el Sector Defensa, la Dirección Nacional de Inteligencia y la Unidad Nacional de Protección que necesiten reserva para su adquisición. Las entidades estatales no están obligadas a publicar los documentos del proceso para adquirir bienes y servicios en el Sector Defensa, la Dirección Nacional de Inteligencia y la Unidad Nacional de Protección que requieren reserva. En estos procesos de contratación la adquisición debe hacerse en condiciones de mercado sin que sea necesario recibir varias ofertas.

3. Redes, sistemas de información y comunicaciones, incluyendo hardware y software, servicios y accesorios, infraestructura para la ciberdefensa y ciberseguridad

informática y física, incluyendo la consultoría, el diseño, las metodologías de análisis, implementación y configuración, requeridos para el Sector Defensa y la Dirección Nacional de Inteligencia (Gobierno Nacional de la República de Colombia, 2013).

- Decreto 730 de 5 de mayo de 2017

Por el cual se dictan medidas asociadas al sector de Tecnologías de la Información y las Comunicaciones, en el marco del Estado de Emergencia Económica, Social y Ecológica decretado mediante Decreto 601 de 2017.(Mintic, 2017b)

- Decreto 1414 de 25 de agosto de 2017

Por el cual se modifica la estructura del Ministerio de Tecnologías de la Información y las Comunicaciones y se dictan otras disposiciones.(Mintic, 2017a)

- Decreto 1413 de 2017

Por el cual se adiciona el título 17 a la parte 2 del libro 2 del Decreto Único Reglamentario del sector de Tecnologías de la Información y las Comunicaciones, Decreto 1078 de 2015, para reglamentarse parcialmente el capítulo IV del título III de la Ley 1437 de 2011 y el artículo 45 de la Ley 1753 de 2015 estableciendo lineamientos generales en el uso y operación de los servicios ciudadanos digitales.(Tecnolog, Informaci, & Decrelj, 2017)

DERECHOS DE AUTOR

El software “Udecquino” se encuentra en trámite de registro en la Dirección Nacional de Derechos de Autor (DNDA), con número de registro 1-2017-89739, como tipo de

obra en registro de soporte lógico – software desde el día 17 de Octubre del 2017.
(Gobierno de Colombia, 2017).

Documentación del Software

Plan de Proyecto

En la Tabla 9 se relacionan las actividades planeadas para la realización de la presente investigación con sus respectivos responsables, además se especifica la duración de las tareas y los hitos en el desarrollo del proyecto.

Tabla 9. *Cronograma y descripción de actividades*

Actividad	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Responsable de la Actividad
Aprobación Minr008	X						CIT
Creación del artículo IEEE					X		Andrés Camacho / Andrés Carranza
Libro del proyecto de grado	X	X	X	X	X		Andrés Camacho / Andrés Carranza
Recolección de datos X Y Z		X	X				Andrés Camacho / Andrés Carranza
Creación del modelo de comunicación para los movimientos		X	X				Andrés Camacho / Andrés Carranza
Creación de la interfaz de control del software			X	X			Andrés Camacho / Andrés Carranza
Pruebas				X	X		Andrés Camacho / Andrés Carranza
Presentación Final						X	Andrés Camacho / Andrés Carranza

Estimación de recursos

En la Tabla 10 se hace un resumen por rubros de los recursos utilizados en el presente proyecto y se detalla la cantidad de dinero en pesos colombianos solicitado a la Universidad de Cundinamarca y/o a otras entidades.

Tabla 10. *Recursos utilizados*

Rubros	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
		UDEC	Otras Entidades	
PERSONAL	\$0	N/A	N/A	\$0
EQUIPOS	\$0	N/A	N/A	\$0
MATERIALES E INSUMOS	\$0	N/A	N/A	\$0
SERVICIOS TECNOLOGICOS	\$0	N/A	N/A	\$0
VIAJES	\$0	N/A	N/A	\$0
OTROS	\$0	N/A	N/A	\$0
TOTALES	\$0	N/A	N/A	\$0

Determinación de requerimientos

Requerimientos funcionales. De las tablas 11 a la 24 se muestran los requerimientos funcionales del sistema.

Tabla 11. *Requerimiento Funcional 1*

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Captura de Datos.
Descripción del requerimiento:	El sistema tendrá las opciones de capturar los datos correspondientes a las marchas básicas equinas a petición del Usuario.

Tabla 12 *Requerimiento Funcional 2*

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Almacenamiento de los Datos.
Descripción del requerimiento:	Los datos que se visualicen en la tabla de datos de los sensores podrán ser almacenados en la memoria del computador a petición del usuario.

Tabla 13 *Requerimiento Funcional 3*

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Entrenar RNA.
Descripción del requerimiento:	El Usuario podrá entrenar la RNA a voluntad siempre y cuando los datos requeridos ya están cargados en el sistema.

Tabla 14 *Requerimiento Funcional 4*

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Visualización Capturas.
Descripción del requerimiento:	El sistema visualizará por defecto los datos de la última captura, y podrá visualizar capturas de los datos ya almacenados a petición del usuario.

Tabla 15 *Requerimiento Funcional 5*

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Graficar Marchas.
Descripción del requerimiento:	El usuario podrá graficar los datos que desee con tal de que dicha información se visualice en la tabla de datos de los sensores.

Tabla 16 *Requerimiento Funcional 6*

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Iniciar transmisión.
Descripción del requerimiento:	El usuario podrá habilitar esta opción para permitir el paso de información de los sensores.

Tabla 17 *Requerimiento Funcional 7*

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Terminar Transmisión.
Descripción del requerimiento:	El usuario podrá usar esta opción una vez haya terminado de capturar los datos necesarios.

Tabla 18 *Requerimiento Funcional 8*

Identificación del requerimiento:	RNF08
Nombre del Requerimiento:	Cargar Datos.
Descripción del requerimiento:	El sistema permitirá que el usuario cargue las marchas o pesos de la RNA que hayan sido guardadas previamente.

Tabla 19 *Requerimiento Funcional 9*

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Aplicar FFT
Descripción del requerimiento:	Una vez los datos hayan sido cargados, el sistema le aplicará de cálculo de la transformada rápida de Fourier a dichos datos, pero solo cuando los datos se carguen en la interface de entrenamiento con dominio a la frecuencia.

Tabla 20 *Requerimiento Funcional 10*

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Graficar Error.
Descripción del requerimiento:	El usuario visualizará una gráfica del error con respecto a los ciclos de entrenamiento y valor de dicho error una vez la RNA haya sido entrenada.

Tabla 21 *Requerimiento Funcional 11*

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Almacenar Pesos.
Descripción del requerimiento:	El usuario tendrá la posibilidad de almacenar los pesos de la RNA una vez esta haya sido entrenada, para no tener que volver a entrenarla.

Tabla 22 *Requerimiento Funcional 12*

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Aplicar Umbral.
Descripción del requerimiento:	El usuario podrá aplicar umbral dado por él mismo a una muestra o todas las que correspondan a las mismas características (Sensor y eje).

Tabla 23 *Requerimiento Funcional 13*

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Graficar Dominio de la Frecuencia.
Descripción del requerimiento:	El usuario podrá visualizar las gráficas de las marchas almacenadas que previamente se calcularon con dominio en la frecuencia (FFT) de cada eje por cada sensor.

Tabla 24 *Requerimiento Funcional 14*

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Comprobación RNA.
Descripción del requerimiento:	El usuario podrá comprobar el entrenamiento de la RNA, cargando una marcha que puede o no haberse usado para el entrenamiento.

Requerimientos No Funcionales. En la tabla 25 se muestra el requerimiento NO funcional.

Tabla 25 *Requerimiento NO Funcional 1*

Identificación del requerimiento:	RNF1
Nombre del Requerimiento:	Identificación de la Institución
Descripción del requerimiento:	En el aplicativo se visualizará imágenes alusivas a la Universidad de Cundinamarca.

Especificación del Sistema

En esta sección se especifica el propósito y las características fundamentales del sistema que se desarrolla en la presente investigación.

Definición del Sistema

Es un sistema que recopila y procesa información de datos dados por los acelerómetros mma7455 los cuales están adaptados a un robot equino de 6 grados de libertad, usados para el entrenamiento y comprobación de una Red Neuronal Artificial cuya función es el procesamiento y reconocimiento de las marchas de un equino en los ejes (X, Y, Z). En la Tabla 26 se muestra la definición del sistema.

Tabla 26 *Definición del sistema*

P(ropietarios)	Universidad de Cundinamarca.
A(ctores)	Usuarios autorizados por el Docente Jaime Eduardo Andrade Ramírez.
C(lientes)	Escuela de carabineros “Alfonzo López Pumarejo”.
R(estricciones o marco regulatorio del sistema)	El aplicativo está desarrollado en el programa MATLAB y solo puede ser ejecutado a través del mismo por computadoras de escritorio y computadores portátiles. Se usará el lenguaje de programación de dicho programa (.m). Se recomienda tener instalada una versión de MATLAB del 2015B. El equipo donde se instale el sistema debe ser capaz de procesar los datos a alta velocidad.
W(eltanchauung, marco conceptual o declaración de intenciones del sistema)	El aplicativo se denomina UDECQUINO®, su intención es la recopilación de datos de un robot equino con el propósito de detectar patrones de movimiento por medio del uso de redes neuronales artificiales. El sistema almacenará los datos de los sensores para usarlos en el entrenamiento de la RNA a través del aplicativo.

Especificación del Diseño

En este capítulo se encuentra la arquitectura del aplicativo “UDECCUINO”, la arquitectura está conformada por varios modelos UML 2.0 que se clasifican en tres grupos definidos como (Reinoso, 1999):

- Dimensión Estática: En este grupo se encuentra el modelo relacionado a la estructura del software.
- Dimensión Dinámica: En este grupo se encuentran los modelos relacionados a la interacción de objetos en el software.
- Dimensión Funcional: En este grupo se encuentran los modelos relacionados al comportamiento del software.

Sin embargo, solo se incluirán los diagramas que aplican al proyecto, por ejemplo, el aplicativo no usa una base de datos por lo que no se realiza un diagrama de entidad relación ni un diagrama de clases ya que se desarrolló mediante programación estructurada.

Dimensión Estática

Diagramas de Casos de Uso

El Diagrama de Casos de uso contiene un actor que intervienen en el software y las funcionalidades que realiza cada uno de ellos.

- A. Tenemos las opciones para la captura de los datos (Figura 31).

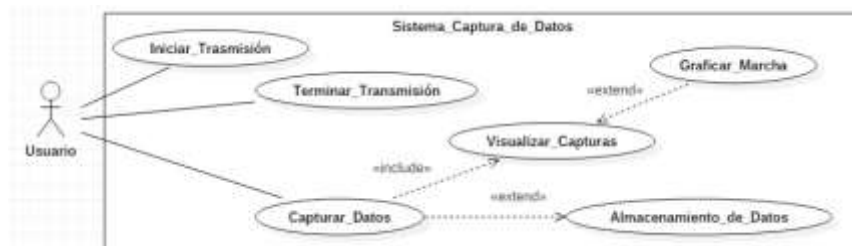


Figura 31 Casos de uso - Captura de datos

- B. Tenemos la opción para el entrenamiento de la RNA con datos con dominio en el tiempo (Figura 32).

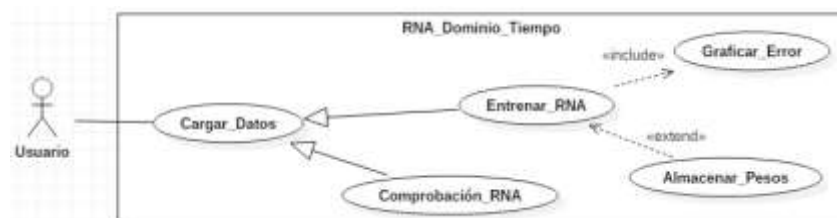


Figura 32 Casos de uso - RNA – Tiempo

- C. Tenemos la opción para el entrenamiento de la RNA con datos con dominio en la frecuencia (Figura 33).

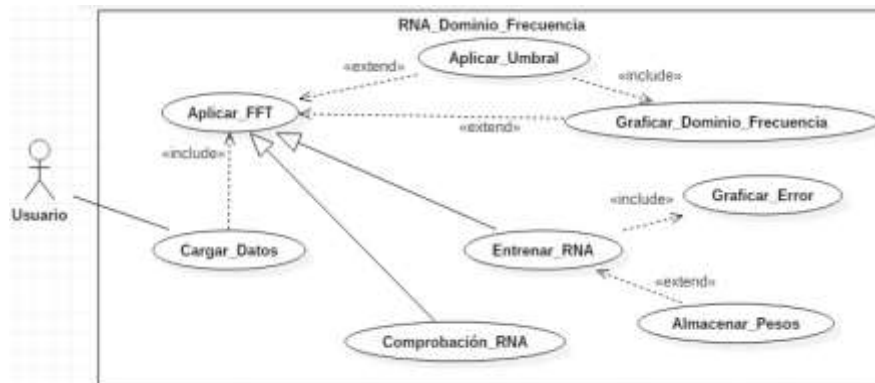


Figura 33 Casos de uso - RNA – Frecuencia

Descripción de Casos de Uso

En la Tabla 27 se describen las funcionalidades mostradas en el diagrama de casos de uso.

Tabla 27 Definición – Casos de uso

CASOS DE USO	DESCRIPCIÓN
Iniciar_Transmisión	Autorización por parte del Usuario para permitir que el Arduino envíe la información de los sensores al equipo.
Terminar_Transmisión	Detención por parte del Usuario para negar el paso de información del Arduino.
Captura_de_Datos	El sistema tendrá las opciones de capturar los datos correspondientes a las marchas básicas equinas (Paso, Trote y Galope) a petición del Usuario

Visualizar_Capturas	El sistema visualizará los datos tanto de la captura de datos como la de los datos ya almacenados a petición del usuario.
Graficar_Marcha	El sistema ofrecerá al usuario la opción de graficar las marchas mientras que dicha información se visualice en la tabla de datos de los sensores.
Almacenamiento_de_Datos	El sistema tendrá la opción de almacenar los datos que se visualicen en la tabla de datos de los sensores en la memoria del equipo a petición del usuario.
Cargar_Datos	El sistema tendrá la opción de cargar los datos que él mismo haya guardado.
Aplicar_FFT	Una vez los datos hayan sido cargados, se le aplicara la transformada rápida de Fourier a dichos datos, pero solo cuando los datos se carguen en la interface de entrenamiento con dominio en la frecuencia.
Aplicar_Umbral	El sistema permitirá al usuario modificar una muestra o todas las que correspondan a las mismas características (Sensor y eje) aplicando un umbral.
Graficar_Dominio_Frecuencia	El usuario podrá visualizar las gráficas de las marchas almacenadas que previamente se calcularon con dominio en la frecuencia (FFT) de cada eje por cada sensor.

Entrenar_RNA	El sistema tendrá la opción para entrenar la RNA a voluntad si los datos requeridos ya están cargados en el sistema.
Graficar_Error	El usuario visualizará una gráfica del error con respecto a los ciclos de entrenamiento una vez la RNA haya sido entrenada.
Almacenar_Pesos	El usuario tendrá la posibilidad de almacenar los Pesos de la RNA una vez esta haya sido entrenada, para no tener que volver a entrenarla.
Comprobación_RNA	El usuario podrá comprobar el entrenamiento de la RNA, cargando una marcha que puede o no haberse usado para el entrenamiento.

Dimensión Dinámica

Diagramas de Secuencia:

A continuación, en las figuras 34 a la 37 se muestran los diagramas de secuencia utilizados.

A. Captura de datos:

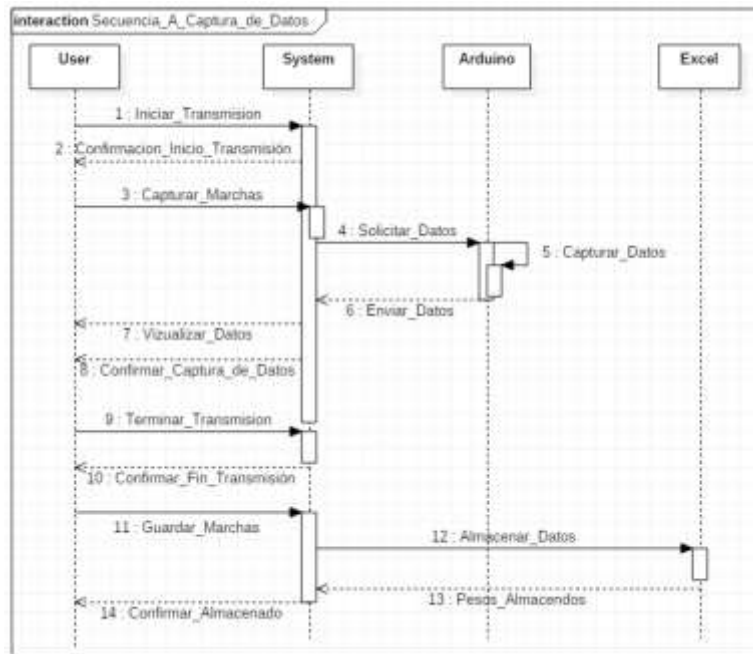


Figura 34 Diagrama de secuencia - Captura de datos

B. Entrenamiento con dominio en el tiempo:

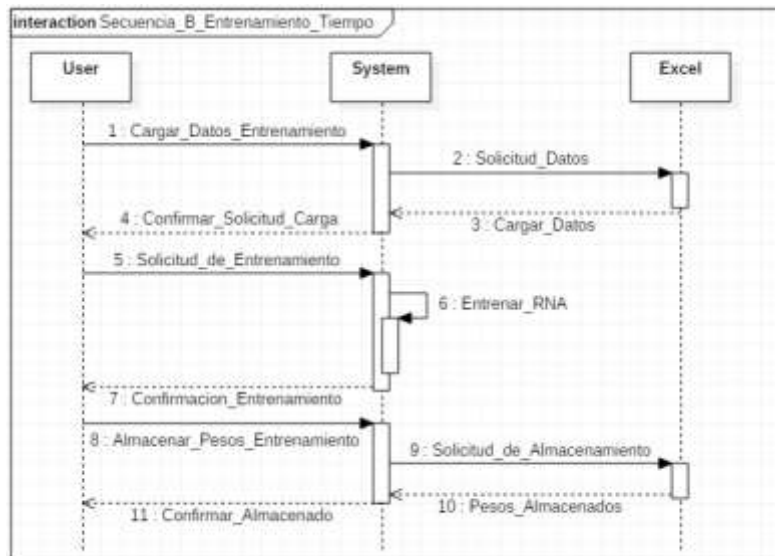


Figura 35 Diagrama de secuencia - Entrenamiento RNA – Tiempo

C. Entrenamiento con dominio en la Frecuencia:

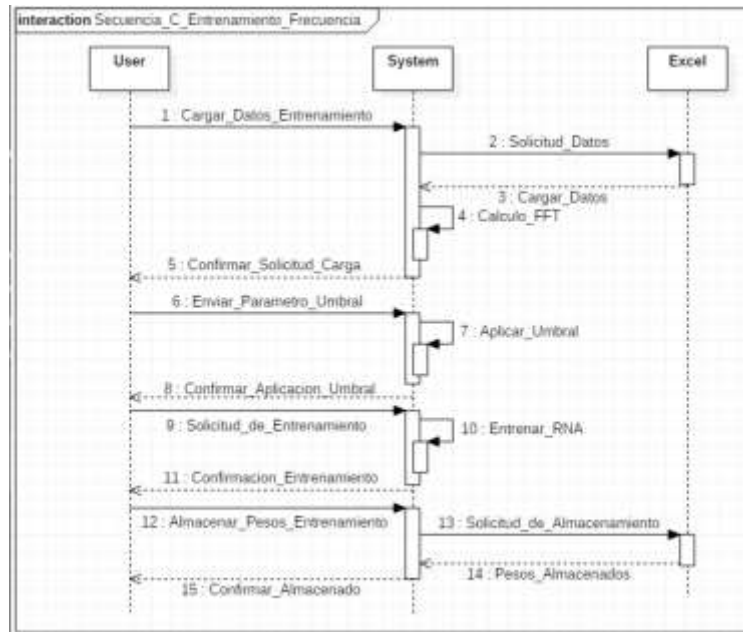


Figura 36 Diagrama de secuencia - Entrenamiento RNA – Frecuencia

D. Comprobación RNA:

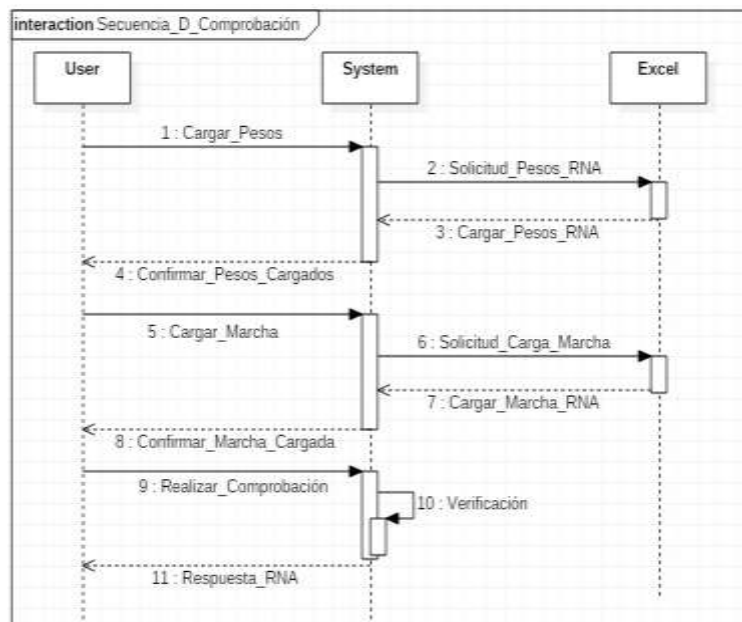


Figura 37 Diagrama de secuencia - Comprobación RNA

Descripción de los diagramas de secuencia.

En la Tabla 28 se describen los objetos diagrama de secuencia.

Tabla 28 *Diagrama de secuencia - Definiciones*

Diagrama de secuencia	Descripción
User	Es el administrador del aplicativo UDECQUINO®
System	Es el aplicativo en sí, la interface que recibe las solicitudes del usuario.
Arduino	Micro controlador que captura, encapsula y envía los datos al sistema (System)
Excel	Archivos en los cuales se almacena información (.mat) tal como datos enviados por el Arduino, o resultado de los cálculos de la RNA.

Diagrama de Colaboración

Muestran la interacción basada en los diagramas de secuencia como se muestra en las figuras 38 a la 41.

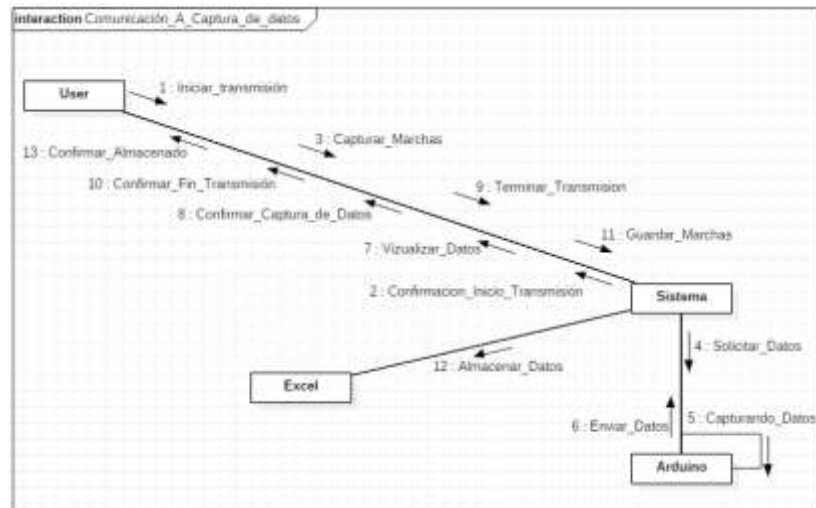


Figura 38 Diagrama de colaboración - Captura de datos

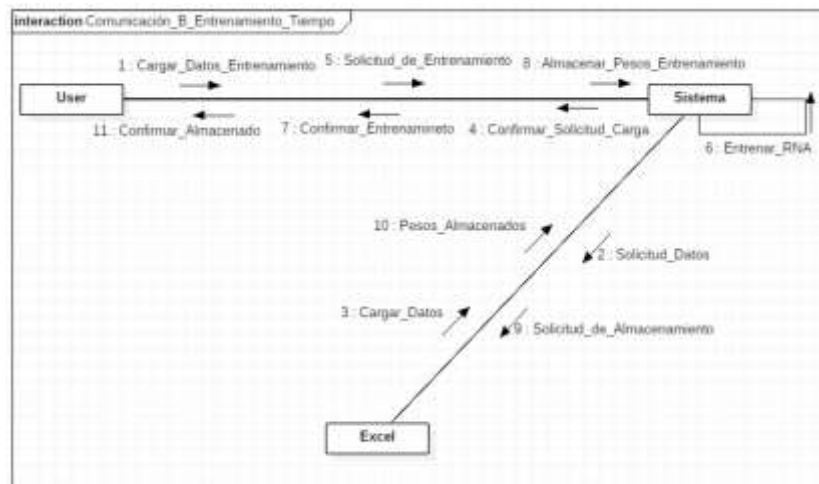


Figura 39 Diagrama de colaboración - Entrenamiento RNA tiempo

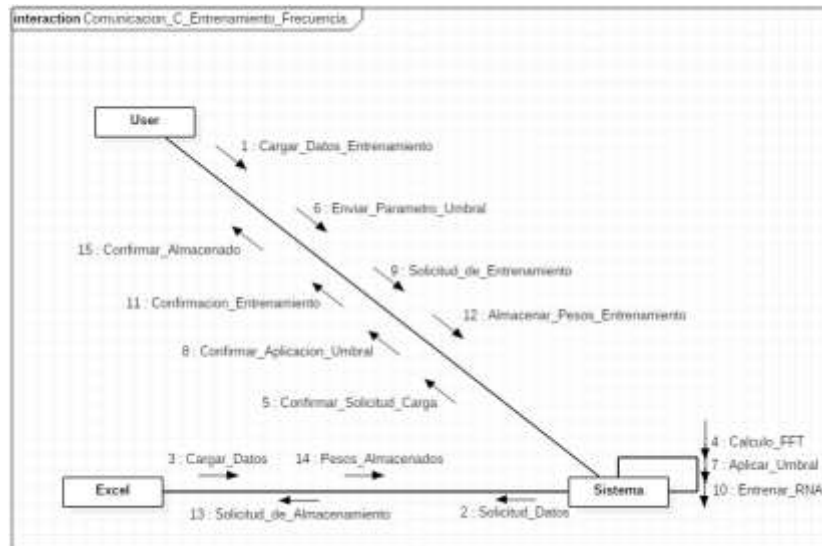


Figura 40 Diagrama de colaboración -Entrenamiento RNA frecuencia

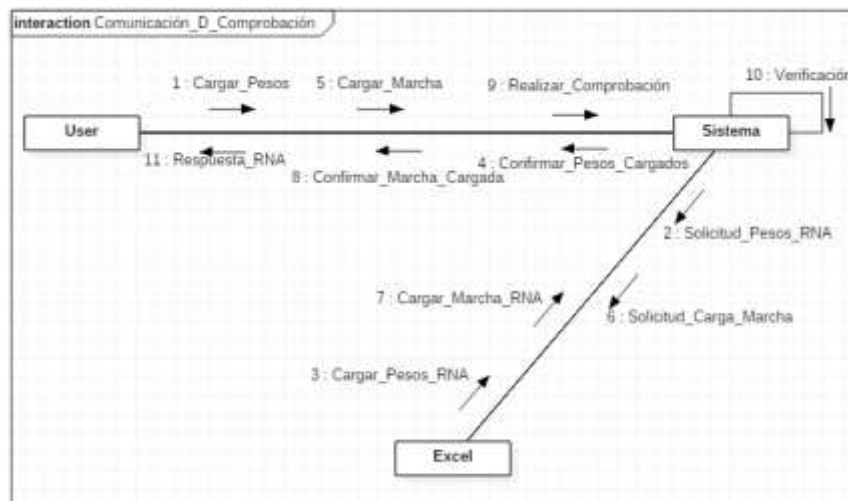


Figura 41 Diagrama de colaboración - Comprobación RNA

Dimensión Funcional

Diagramas de Actividades

De la figura 42 a la 44 se muestran los diagramas de actividades definidos en el proyecto.

A. Captura de datos:

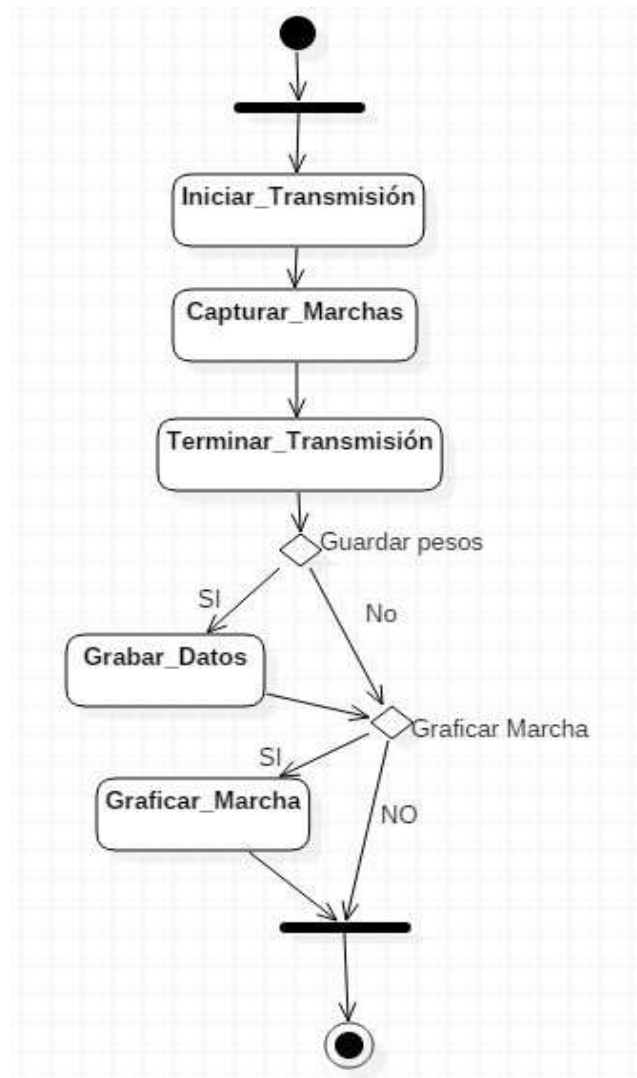


Figura 42 Diagrama de actividades - Captura de datos

B. Entrenamiento en dominio en el Tiempo y comprobación:

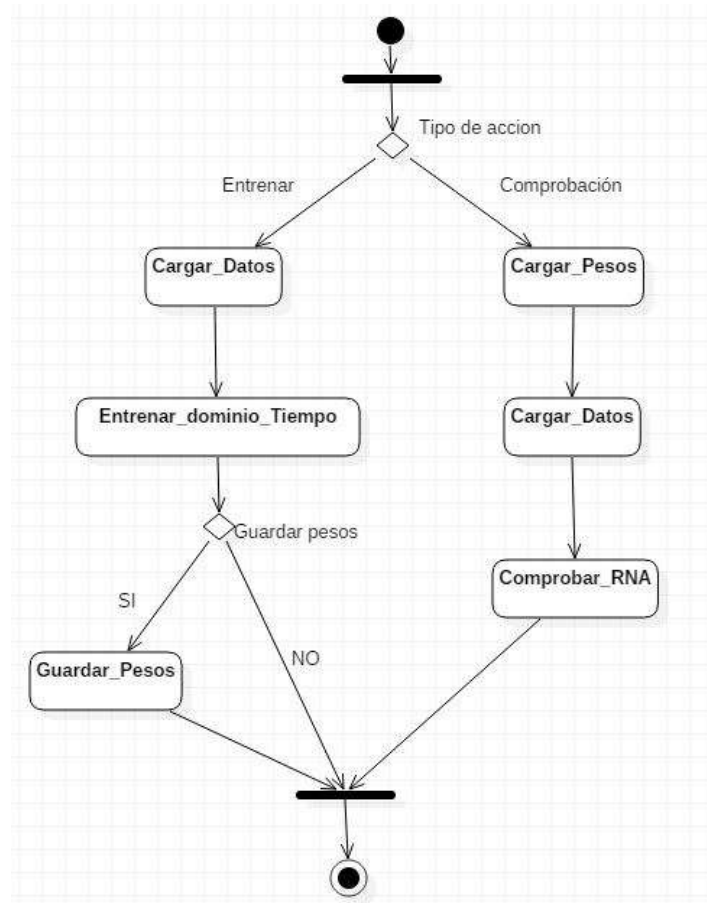


Figura 43 Diagrama de actividades - Entrenamiento y Comprobación RNA – Tiempo

C. Entrenamiento en el dominio de la frecuencia y comprobación:

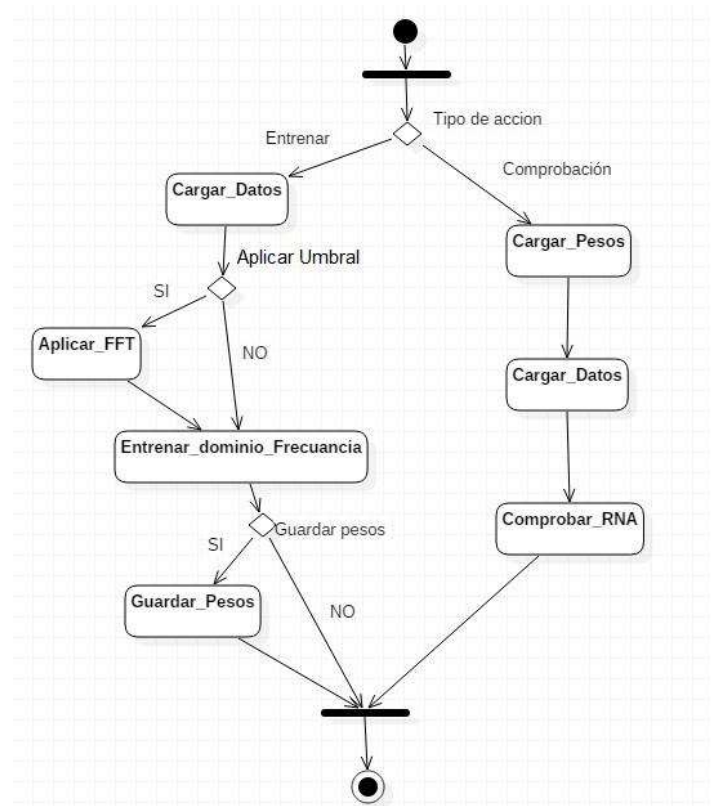


Figura 44 Diagrama de actividades - Entrenamiento y Comprobación RNA - Frecuencia

Dimensión organizacional:

Diagrama de Despliegue:

Muestra la configuración de instalación recomendada de componentes de hardware y software para que el aplicativo funcione correctamente como se muestra en la figura 45.

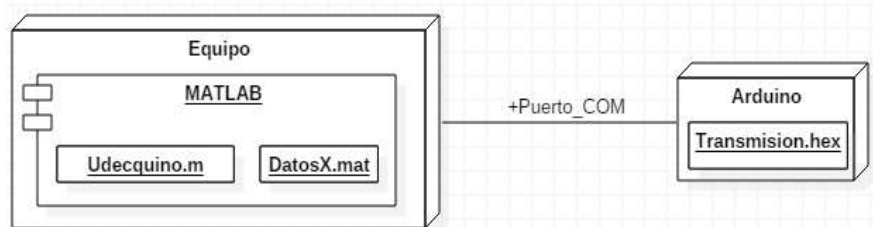


Figura 45 Diagrama de despliegue

En la Tabla 29 se describen todos los elementos incluidos en el diagrama de despliegue.

Tabla 29 Diagrama de despliegue - Elemento

ELEMENTO	Descripción
Equipo	Es el equipo en el que el aplicativo está instalado, requiere el software MATLAB, en el cual se ejecuta el aplicativo
Arduino	Es el micro controlador que captura, encapsula y envía la información.

Diseño de los Casos de Prueba

A continuación, de la tabla 30 a la 42, se especifican los casos de prueba que se plantean para el aplicativo UDECQUINO.

Tabla 30 *Diseño de casos de prueba 1*

Descripción	Iniciar transmisión
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Iniciar”
Datos de Prueba	- Valores dados por los sensores
Resultados esperados	Los sensores inician a transmitir la información al aplicativo.
Resultados obtenidos	El sistema permite controlar el ciclo de trabajo de los sensores al iniciar la transmisión de datos, el cambio es visible por medio de la iluminación del LED de color verde (El cual indica la transmisión de datos).

Tabla 31 *Diseño de casos de prueba 2*

Descripción	Terminar Transmisión
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Terminar”
Datos de Prueba	- Valores datos por los sensores
Resultados esperados	Los sensores no transmiten información al aplicativo.
Resultados obtenidos	El sistema permite controlar el ciclo de trabajo de los sensores terminando la transmisión de datos, el cambio es visible por que el LED de color verde deja de iluminar (Lo cual indica que la transmisión cesó).

Tabla 32 *Diseño de casos de prueba 3*

Descripción	Captura de datos
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Iniciar” - Pulsar uno de los botones “Paso”, “Trote” o “Galope”
Datos de Prueba	<ul style="list-style-type: none"> - Valores dados por los sensores - Numero de muestras
Resultados esperados	Si el número de muestras ingresados es correcto (parámetros), el sistema inicia a tomar las muestras.
Resultados obtenidos	El sistema muestra una barra de progreso de captura de muestras, de lo contrario muestra una ventana expresando el error ocurrido.

Tabla 33 *Diseño de casos de prueba 4*

Descripción	Visualizar Capturas
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Iniciar” - Pulsar uno de los botones “Paso”, “Trote” o “Galope” - Visualizar datos
Datos de Prueba	Valores ubicados en la tabla de “Panel de Visualización de Datos”
Resultados esperados	Si las muestras no presentaron errores en la captura, dichos datos se muestran en la tabla ya mencionada
Resultados obtenidos	El usuario puede visualizar las muestras que haya decidido capturar

Tabla 34 Diseño de casos de prueba 5

Descripción	Graficar marcha
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Iniciar” - Pulsar uno de los botones “Paso”, “Trote” o “Galope” - Graficar marcha o - Cargar alguna muestra que este almacenada
Datos de Prueba	- Valores visualizados en el “Panel de Visualización de Datos”
Resultados esperados	Se visualiza los datos que hayan capturado los sensores en una muestra específica
Resultados obtenidos	El sistema muestra una gráfica de los datos capturados por cada sensor en cada eje (6 graficas).

Tabla 35 Diseño de casos de prueba 6

Descripción	Almacenamiento de datos
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Pulsar el botón “Iniciar” - Pulsar uno de los botones “Paso”, “Trote” o “Galope” - Almacenar marcha
Datos de Prueba	- Valores visualizados en el “Panel de Visualización de Datos”
Resultados esperados	Se podrá almacenar los datos capturados en la memoria del equipo
Resultados obtenidos	El usuario puede almacenar los datos capturados donde él lo deseé

Tabla 36 *Diseño de casos de prueba 7*

Descripción	Cargar datos
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos
Datos de Prueba	- Valores almacenados con datos dados por los sensores
Resultados esperados	Se podrá subir al aplicativo marchas que hayan sido capturadas anteriormente
Resultados obtenidos	El sistema permite al usuario cargar archivos que contengan los datos referentes a las marchas

Tabla 37 *Diseño de casos de prueba 8*

Descripción	Aplicar FFT
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos
Datos de Prueba	- Valores almacenados con datos dados por los sensores
Resultados esperados	Los datos se les aplica la transformada rápida de Fourier
Resultados obtenidos	El aplicativo realiza los cálculos de la transformada rápida de Fourier una vez que el usuario haya cargado las marchas capturadas

Tabla 38 *Diseño de casos de prueba 9*

Descripción	Aplicar Umbral
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Aplicar FFT - Aplicar umbral
Datos de Prueba	<ul style="list-style-type: none"> - Valores subidos en el aplicativo con las muestras capturadas - Valor del Umbral
Resultados esperados	Si el usuario lo desea podrá aplicar un umbral a la muestra que visualice
Resultados obtenidos	El sistema permite al usuario indicar un umbral y aplicarlo a una muestra o a todas las que compartan sus características (Sensor, Eje)

Tabla 39 *Diseño de casos de prueba 11*

Descripción	Entrenar RNA
Procedimiento	Dominio en el Tiempo <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos Dominio en la Frecuencia <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Aplicar FFT - Aplicar umbral
Datos de Prueba	Dominio en el Tiempo <ul style="list-style-type: none"> - Valores subidos en el aplicativo con las muestras capturadas Dominio en la Frecuencia <ul style="list-style-type: none"> - Valores subidos en el aplicativo con las muestras capturadas aplicando la FFT con su respectivo proceso de aplicación del umbral
Resultados esperados	Si el usuario lo desea podrá iniciar el entrenamiento de la RNA
Resultados obtenidos	El sistema permite al usuario controlar el entrenamiento de la RNA siempre que haya indicado con datos de entrenamiento

Tabla 40 *Diseño de casos de prueba 11*

Descripción	Graficar Error
Procedimiento	Dominio en el Tiempo <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Entrenar RNA Dominio en la Frecuencia <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Aplicar FFT - Aplicar umbral - Entrenar RNA
Datos de Prueba	<ul style="list-style-type: none"> - Valores subidos en el aplicativo con las muestras capturadas - Valor del Umbral
Resultados esperados	Si el usuario lo desea podrá visualizar una gráfica con el valor del error de la RNA
Resultados obtenidos	El sistema permite al usuario visualizar una gráfica del error con respecto a los ciclos del proceso una vez la RNA termine de entrenar

Tabla 41 *Diseño de casos de prueba 12*

Descripción	Almacenar pesos
Procedimiento	Dominio en el Tiempo <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Entrenar RNA Dominio en la Frecuencia <ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos - Aplicar FFT - Aplicar umbral - Entrenar RNA
Datos de Prueba	<ul style="list-style-type: none"> - Valores del cálculo de la RNA entrenada
Resultados esperados	Si el usuario lo desea podrá almacenar los datos del entrenamiento de la RNA
Resultados obtenidos	El sistema permite al usuario almacenar en el equipo los datos resultantes del cálculo de la RNA (pesos)

Tabla 42 *Diseño de casos de prueba 13*

Descripción	Comprobación RNA
Procedimiento	<ul style="list-style-type: none"> - Ingresar al aplicativo - Cargar datos
Datos de Prueba	<ul style="list-style-type: none"> - Valores subidos en el aplicativo con las muestras capturadas - Valores del cálculo de la RNA
Resultados esperados	El usuario podrá comprobar el uso de la RNA cargando datos para el reconocimiento e los mismos
Resultados obtenidos	El sistema permite al usuario cargar al aplicativo muestras que pueden o no haber sido usadas para el entrenamiento, para saber si dichas muestras las reconoce la RNA

Resultados

De acuerdo con el proceso desarrollado en el presente proyecto y descrito en la metodología, se presentan los siguientes resultados, dando a conocer inicialmente la aplicación desarrollada junto con el prototipo de hardware, así mismo la información obtenida en la fase de validación.

Prototipo de software Udecquino

El prototipo de software Udecquino que fue creado bajo la herramienta de desarrollo MATLAB es una aplicación de escritorio que de acuerdo con la forma en que se programó, permite a los usuarios del sistema, por una parte, realizar la captura de muestras de tres diferentes secuencias de movimientos y visualizarlas haciendo uso de tablas, que muestran los valores registrados por los acelerómetros en el número de muestras seleccionadas por el usuario y su respectiva gráfica en el dominio del tiempo.

Udecquino es una herramienta que permite guardar los datos de las marchas capturadas para posteriormente ser utilizadas como entradas de una red neuronal, así como la posibilidad de cargarlos para que el usuario pueda realizar estudios sobre los movimientos.

Con respecto a los módulos redes neuronales, el aplicativo permite el entrenamiento de las redes para el reconocimiento de las marchas (paso, trote, galope). En primera medida da la opción de cargar las muestras de las marchas capturadas para ubicarlos como entradas de la red neuronal tanto en el dominio del tiempo como en el dominio de la frecuencia,

además de esto, los módulos brindan la posibilidad al usuario de ajustar algunos parámetros de la red neuronal artificial como lo son el número de neuronas de la capa oculta y la tasa de aprendizaje, como también la validación de la red.

Prototipo de hardware

El prototipo de hardware construido bajo la metodología mencionada tiene como resultado dos módulos que permiten la transmisión y la recepción de los datos. El primer módulo que es el de transmisión, está compuesto por dos acelerómetros conectados a cables extensores y una baquelita que tiene tres sockets o enchufes, los dos enchufes similares son para conectar los cables extensores y el restante es para la conexión del Xbee PRO S1, del mismo modo esta baquelita puede ser conectada al Arduino Mega 2560. El segundo módulo es el de recepción y se compone de una baquelita con un socket para el Xbee PRO S1 y va acoplada al Arduino UNO.

Fase de Validación

Los módulos construidos permiten alimentar el sistema por medio de un cable usb o cualquier cargador que tenga una salida de voltaje soportada por el arduino.

Según lo descrito en la metodología se realizaron tres sesiones de capturas de datos a diferentes distancias y con distinto número de muestras. Las tablas 43 a la 48 que se

muestran a continuación brindan información de los resultados obtenidos en los entrenamientos en los dominios del tiempo y la frecuencia:

Tabla 43 *Porcentaje de error promedio de las sesiones de entrenamiento en el tiempo*

Porcentaje de error promedio basado en la captura 1: Distancia aproximada 1,5m	Porcentaje de error promedio basado en la captura 2: Distancia aproximada 8m	Porcentaje de error promedio basado en la captura 3: Distancia aproximada 25m
16%	24%	14%

Porcentaje de error promedio del entrenamiento de la red neuronal en el tiempo con los datos de las diferentes sesiones de captura

Tabla 44 *Porcentaje de error promedio de las sesiones de entrenamiento en frecuencia*

Porcentaje de error promedio basado en la captura 1: Distancia aproximada 1,5m	Porcentaje de error promedio basado en la captura 2: Distancia aproximada 8m	Porcentaje de error promedio basado en la captura 3: Distancia aproximada 25m
1%	8%	7%

Promedio de error del entrenamiento de la red neuronal en la frecuencia con los datos de las diferentes sesiones de captura

Tomando como referencia los datos anteriormente relacionados, se muestra la información obtenida de los diferentes tipos de entrenamiento basados en la sesión de captura 3 teniendo en cuenta que, en esta, el porcentaje de error se mantuvo constante respecto a las otras dos sesiones de entrenamiento:

Tabla 45 Error promedio (%) del entrenamiento en el tiempo basados en la sesión de captura 3

20% de los datos	60% de los datos
22%	6%

Porcentaje de error promedio de la red neuronal en el tiempo, con el 20% y el 60% de los datos entrenados basados en la sesión de captura 3

Tabla 46 Error promedio (%) del entrenamiento en frecuencia basado en la sesión de captura 3

20% de los datos	60% de los datos
9%	5%

Porcentaje de error promedio de la red neuronal en la frecuencia, con el 20% y el 60% de los datos entrenados basados en la sesión de captura 3

Tabla 47 Promedio de ciclos de la red en el tiempo basados en la sesión de captura 3

Número de neuronas	20% de los datos:	60% de los datos:
	Promedio total de ciclos	Promedio total de ciclos
	3790	6235
30	4217	7556
60	3840	5992
100	3313	5158

Promedio de ciclos realizados durante el entrenamiento de la red neuronal en el tiempo basado en la sesión de captura 3 con el 20% y el 60% de los datos cada uno con 30, 60 y 100 neuronas

Tabla 48 Promedio de ciclos de la red en frecuencia basados en la sesión de captura 3

Número de neuronas	20% de los datos:	60% de los datos:
	Promedio total de ciclos	Promedio total de ciclos
	1119	1212
5	1837	1913
15	915	966
30	605	758

Promedio de ciclos realizados durante el entrenamiento de la red neuronal en la frecuencia basado en la sesión de captura 3 con el 20% y el 60% de los datos cada uno con 5, 15 y 30 neuronas

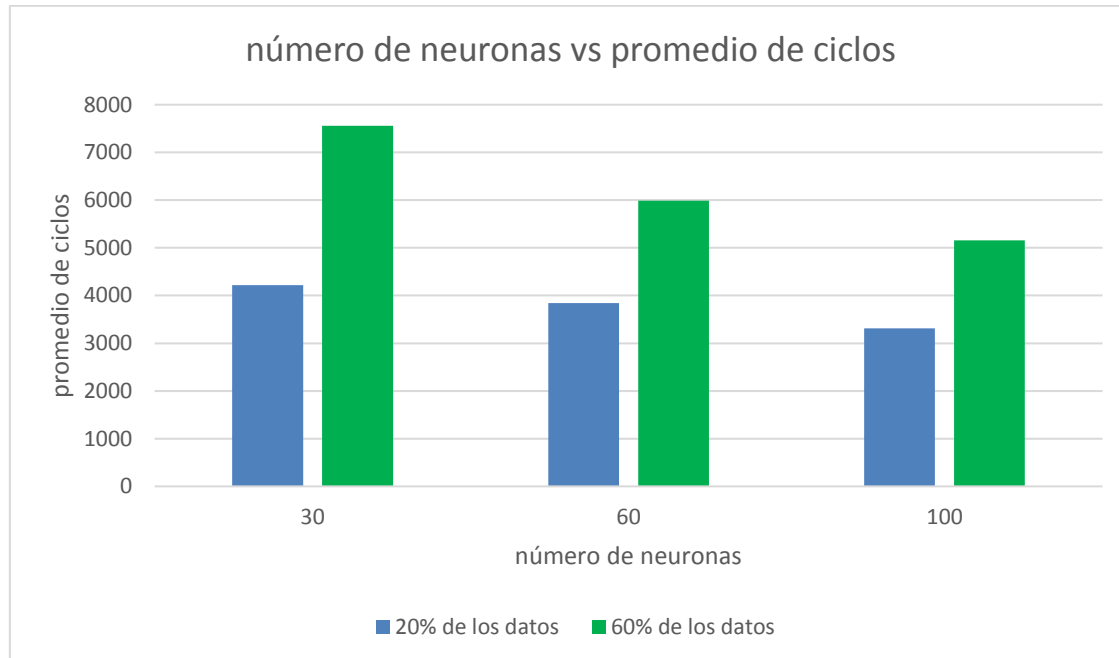


Figura 46 número de neuronas vs promedio de ciclos en el tiempo - sesión de captura 3

En la Figura 46 podemos observar el promedio de ciclos de entrenamientos usados por la red en el tiempo para converger, respecto al número de neuronas y cierto porcentaje de los datos como se analizará a continuación:

Para el 20% de los datos la red neuronal en el tiempo uso un promedio de 4217 ciclos con 30 neuronas, 3840 ciclos con 60 neuronas y 3313 ciclos con 100 neuronas.

Para el 60% de los datos la red neuronal en el tiempo uso un promedio de 7556 ciclos con 30 neuronas, 5992 ciclos con 60 neuronas y 5158 ciclos con 100 neuronas.

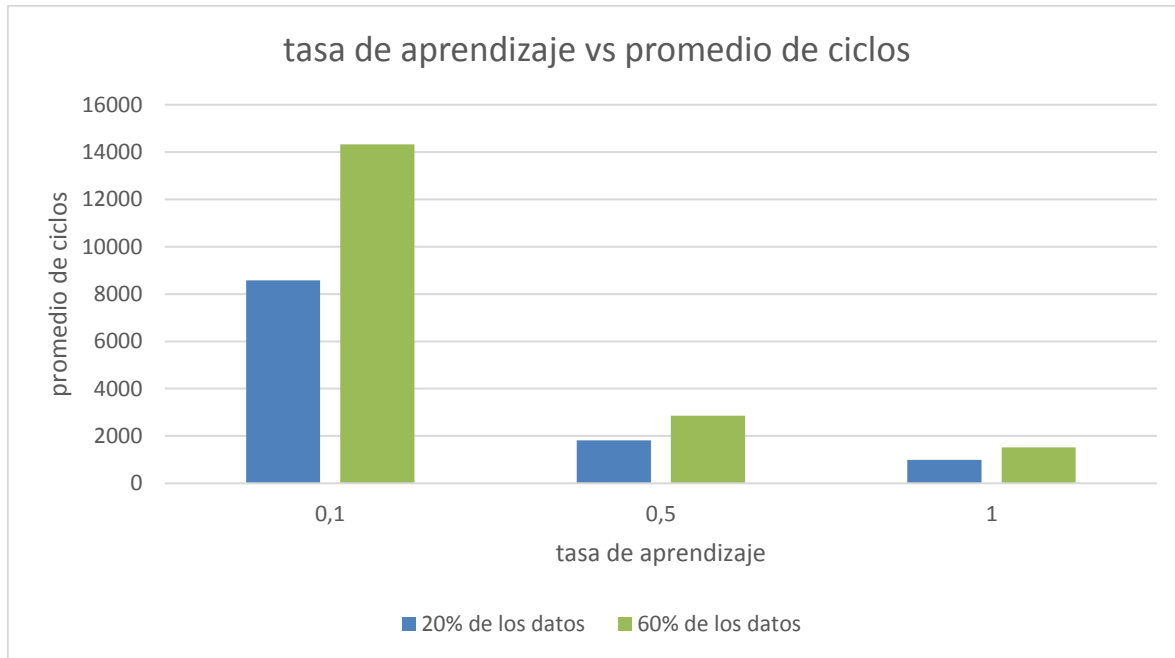


Figura 47 tasa de aprendizaje vs promedio de ciclos en el tiempo-sesión de captura 3

En la figura 47 podemos observar el promedio de ciclos de entrenamientos usados por la red en el tiempo para converger, respecto al valor de la tasa de aprendizaje y cierto porcentaje de los datos como se analizará a continuación:

Para el 20% de los datos la red neuronal en el tiempo uso para su entrenamiento un promedio de 8571 ciclos con una tasa de aprendizaje de 0,1, 1811 ciclos con una tasa de aprendizaje de 0,5 y 988 ciclos con una tasa de aprendizaje de 1.

Para el 60% de los datos la red neuronal en el tiempo uso para su entrenamiento un promedio de 14326 ciclos con una tasa de aprendizaje de 0,1, 2857 ciclos con una tasa de aprendizaje de 0,5 y 1523 ciclos con una tasa de aprendizaje de 1.

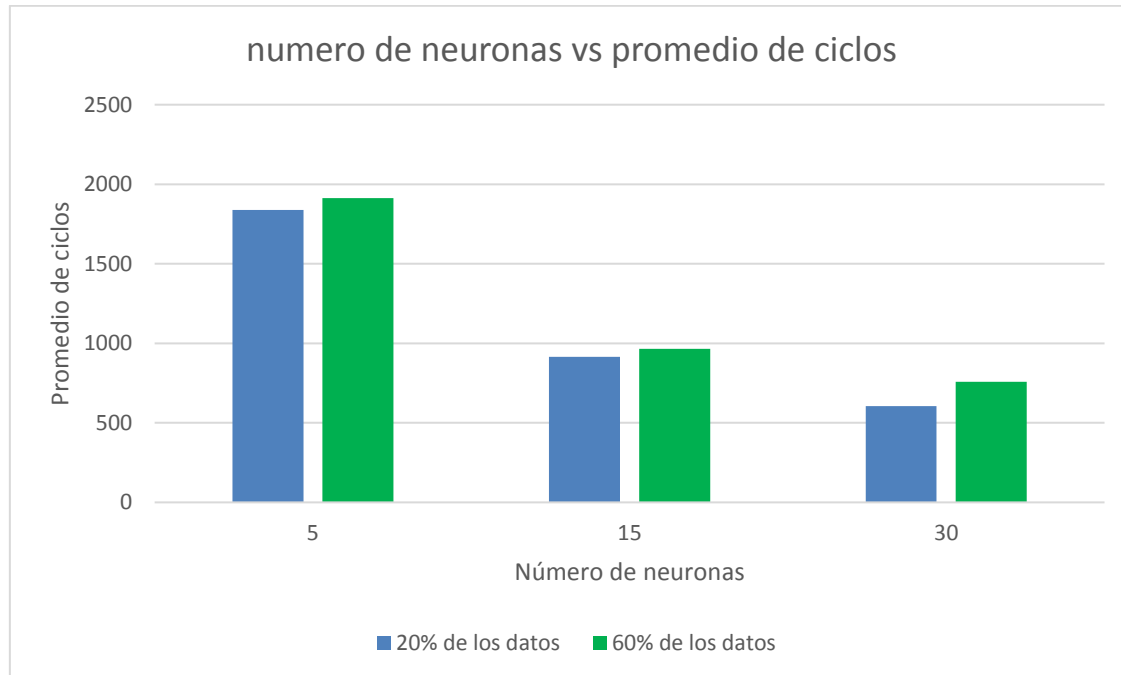


Figura 48 número de neuronas vs promedio de ciclos en la frecuencia-sesión de captura 3

En la figura 48 podemos observar el promedio de ciclos de entrenamientos usados por la red en la frecuencia para converger, respecto al número de neuronas y cierto porcentaje de los datos como se analizará a continuación:

Para el 20% de los datos la red neuronal en la frecuencia uso un promedio de 1837 ciclos con 5 neuronas, 916 ciclos con 15 neuronas y 605 ciclos con 100 neuronas.

Para el 60% de los datos la red neuronal en la frecuencia uso un promedio de 1914 ciclos con 5 neuronas, 967 ciclos con 15 neuronas y 758 ciclos con 30 neuronas.

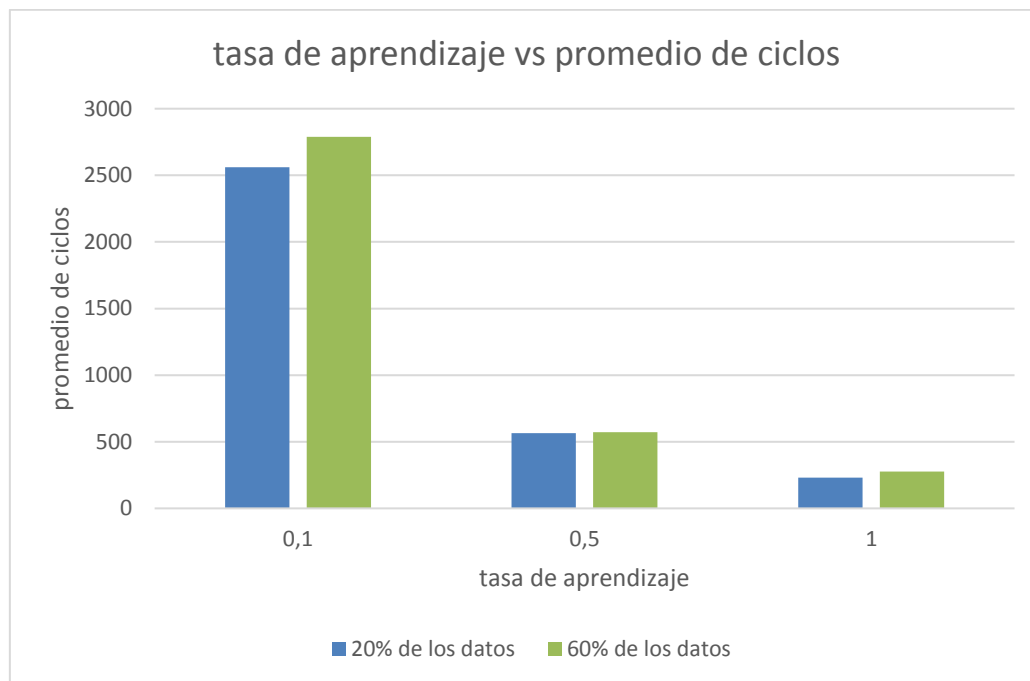


Figura 49 tasa de aprendizaje vs promedio de ciclos en la frecuencia-sesión de captura 3

En la figura 49 podemos observar el promedio de ciclos de entrenamientos usados por la red en la frecuencia para converger, respecto al valor de la tasa de aprendizaje y cierto porcentaje de los datos como se analizará a continuación:

Para el 20% de los datos la red neuronal en la frecuencia uso para su entrenamiento un promedio de 2561 ciclos con una tasa de aprendizaje de 0,1, 566 ciclos con una tasa de aprendizaje de 0,5 y 232 ciclos con una tasa de aprendizaje de 1.

Para el 60% de los datos la red neuronal en la frecuencia uso para su entrenamiento un promedio de 2790 ciclos con una tasa de aprendizaje de 0,1, 571 ciclos con una tasa de aprendizaje de 0,5 y 276 ciclos con una tasa de aprendizaje de 1.

La información relacionada anteriormente proviene de las tablas de error que se encuentran en los anexos C al H.

Conclusiones y Recomendaciones

Con base en la metodología implementada y los resultados obtenidos se definen las siguientes conclusiones:

La fase de modelos de prototipos usada en la metodología permitió el desarrollo conjunto de hardware - software y de igual manera fue muy útil para redefinir y establecer los requerimientos que el cliente no había tenido en cuenta.

El prototipo de hardware generado se compone de un módulo de transmisión robusto que permite ser manipulado con facilidad al tener dos cables extensores flexibles para los acelerómetros, evitando las desconexiones durante las pruebas en la fase de validación.

El funcionamiento del software Udecquino cumplió con los parámetros definidos por el cliente para cada una de las tres ventanas desarrolladas.

Los archivos generados por el software son consistentes puesto que los datos que se encuentran en estos están completos y ordenados

De la fase de validación se puede concluir que:

La captura de los datos realizada en las tres sesiones se realizó con éxito y la información de las marchas capturadas sirvió como parámetros de entrada para el entrenamiento de la red neuronal tanto en el tiempo como en la frecuencia.

Durante el entrenamiento de la red neuronal en el tiempo, las distancias definidas para cada sesión de captura aparentemente no afectó el porcentaje de error de la red y se puede observar que la sesión de captura 3 a pesar de que fue a la que mayor distancia se realizó, tuvo un menor porcentaje de error con un 14%, pero para la red neuronal en la frecuencia se encuentra que la distancia definida si afecta el porcentaje de error donde la sesión de captura 1 que fue a la que menor distancia se hizo, tuvo el menor porcentaje con un 1%.

El error en distancias superiores a 1,5 metros es casi igual con una distancia aproximada de 8 metros que con una distancia aproximada de 25 metros.

A pesar de que la sesión de captura 2 se hizo a menor distancia que la sesión 3, se observó que el número de muestras capturado también influye en el error de la red, pues fue en esta sesión donde se tomaron menos muestras que en las demás con un total de 8000 muestras. La sesión de captura 2 tiene un error más alto que las demás sesiones tanto en la red neuronal del tiempo como en la red neuronal de la frecuencia.

Tomando como referencia la sesión 3 se pudo determinar que:

Entrenar las redes neuronales con el 60% de los datos produce un error mucho menor que con el 20%, puesto que para la red en el tiempo el promedio de error con el 60% de los datos fue un 16% menos y en la red de la frecuencia un 4% menos. Como lo muestran las tablas 45 y 46.

Aumentar la cantidad de neuronas para el entrenamiento de las redes neuronales reduce el número de ciclos que usa la red para entrenarse. El número de ciclos es menor cuando se entrena con el 20% de los datos que con el 60% y con la misma cantidad de neuronas. Como se evidenció en las tablas 47 y 48.

Aumentar la tasa de aprendizaje hasta un valor de 1 en el entrenamiento de las redes neuronales, así como el número de neuronas, redujo notablemente el número el promedio de ciclos de entrenamiento, Como se mostró en las figuras 46 a la 49.

En general la red neuronal artificial en el dominio de la frecuencia siempre tuvo un menor porcentaje de error que la red en el tiempo incluso con el 20% de los datos capturados para el entrenamiento.

A continuación, se detallan una serie de recomendaciones que pueden tenerse en cuenta para mejorar la operabilidad del sistema:

Incluir como un módulo adicional en el prototipo de software la opción de poder aplicar el filtro de Kalman para la eliminación de ruido que pueda ser generado durante la captura de datos aun cuando las redes neuronales sean buenas en la discriminación de este.

Cambiar la sensibilidad de los sensores solo en el caso de que las marchas o movimientos a reconocer lo requieran.

Se recomienda implementar en el prototipo de software un elemento desplegable que permita al usuario poder seleccionar el puerto COM al que se encuentre conectado el Arduino ya que el puerto COM seleccionado por defecto para el aplicativo es el COM6.

Implementar un método que permita determinar la carga restante de una batería cuando el dispositivo esté conectado a una.

Lista de referencias

Abedini, R., Esfandyari, M., Nezhadmoghadam, A., & Rahmanian, B. (2012). The Prediction of Undersaturated Crude Oil Viscosity: An Artificial Neural Network and Fuzzy Model Approach. *Petroleum Science and Technology - PET SCI TECHNOL*, 30, 2008–2021.

ÁLAVA INGENIEROS ©. (2017). Software de análisis de movimiento. Retrieved from <http://www.grupoalava.com/ingenieros/productos/tecnologias-de-vision-e-imagen/software-de-imagen/software-de-analisis-de-movimiento/>

Arduino. (2017). No Title. Retrieved from <https://www.arduino.cc>

Cañada, F. C., Torres-Luque, G., López-Fernández, I., Santos-Lozano, A., Garatachea, N., & Carnero, E. Á. (2015). Actividad física y acelerometría; orientaciones metodológicas, recomendaciones y patrones. *Nutricion Hospitalaria*, 31(1), 115–125. <https://doi.org/10.3305/nh.2015.31.1.7450>

Church, E. E., Walker, A. M., Wilson, A. M., & Pfau, T. (2009). Evaluation of discriminant analysis based on dorsoventral symmetry indices to quantify hindlimb lameness during over ground locomotion in the horse. *Equine Veterinary Journal*, 41(3), 304—308. <https://doi.org/10.2746/042516409x397352>

Colombia, R. de. (1991). CONSTITUCIÓN POLÍTICA DE COLOMBIA 1991.

Retrieved from <http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=4125>

Digi. (n.d.). XBee/XBee-PRO S1 802.15.4 (Legacy) RF Modules User Guide.

Gobierno de Colombia. (2017). Obras literarias - Derecho de Autor. Retrieved from

<http://www.derechodeautor.gov.co/web/guest/obras-literarias>

Gobierno Nacional de la República de Colombia. (2013). Decreto 1510 de 2013.

Retrieved November 2, 2017, from

<https://sintesis.colombiacompra.gov.co/normatividad/documento/14049>

Hernandez Sampieri, R., Fernandez Collado, C., & Baptista Lucio, M. del P. (2010).

Metodología de la investigación. Metodología de la investigación. <https://doi.org/>-

ISBN 978-92-75-32913-9

Hilera, J., & José Martínez Hernando, V. (1995). *Redes neuronales artificiales :*

fundamentos, modelos y aplicaciones / José Ramón Hilera González, Victor José

Martínez Hernando. SERBIULA (sistema Librum 2.0).

Katusca Vielma. (2014). Biomecanica: Softwares para el análisis de datos. Retrieved

from <http://inforbiomeca.blogspot.com.co/2014/10/software-para-el-analisis-de->

datos.html

Martín del Brio, B., & Serrano Cinca, C. (1995). Fundamentos De Las Redes Neuronales

Artificiales: Hardware Y Software. *Scire: Representación Y Organización Del*

Conocimiento, 1, 103–125. Retrieved from

<http://ibersid.eu/ojs/index.php/scire/article/view/1036/1018>

Matich, D. J. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia, 55*.

Retrieved from [ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-](ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf)

[Neuronales/Libros/matich-redesneuronales.pdf](ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf)

Matlab. (2016). Matlab Documentation. Retrieved from <https://es.mathworks.com>

Mintic. (2017a). Decreto 1414 25 AGO 2017. Retrieved from

[http://es.presidencia.gov.co/normativa/normativa/DECRETO 1414 DEL 25 DE](http://es.presidencia.gov.co/normativa/normativa/DECRETO%201414%20DEL%2025%20DE%20AGOSTO%20DE%202017.pdf)

[AGOSTO DE 2017.pdf](http://es.presidencia.gov.co/normativa/normativa/DECRETO%201414%20DEL%2025%20DE%20AGOSTO%20DE%202017.pdf)

Mintic. (2017b). Decreto 730 2017. Retrieved from

[http://es.presidencia.gov.co/normativa/normativa/DECRETO 730 DEL 05 DE](http://es.presidencia.gov.co/normativa/normativa/DECRETO%20730%20DEL%2005%20DE%20MAYO%20DE%202017.pdf)

[MAYO DE 2017.pdf](http://es.presidencia.gov.co/normativa/normativa/DECRETO%20730%20DEL%2005%20DE%20MAYO%20DE%202017.pdf)

Olsen, E., Haubro Andersen, P., & Pfau, T. (2012). Accuracy and Precision of Equine

Gait Event Detection during Walking with Limb and Trunk Mounted Inertial Sensors. *Sensors*, 12(12), 8145–8156. <https://doi.org/10.3390/s120608145>

Parallax. (2009). MMA7455 3-Axis Accelerometer Module. Retrieved from <https://www.parallax.com/sites/default/files/downloads/28526-MMA7455-3-Axis-Accelerometer-Documentation-v1.1.pdf>

PARKES, R. S. V, WELLER, R., GROTH, A. M., MAY, S., & PFAU, T. (2009). Evidence of the development of “domain-restricted” expertise in the recognition of asymmetric motion characteristics of hindlimb lameness in the horse. *Equine Veterinary Journal*, 41(2), 112–117. <https://doi.org/10.2746/042516408X343000>

Parsons, K. J., Pfau, T., Ferrari, M., & Wilson, A. M. (2008). High-speed gallop locomotion in the Thoroughbred racehorse. II. The effect of incline on centre of mass movement and mechanical energy fluctuation. *Journal of Experimental Biology*, 211(6), 945–956. <https://doi.org/10.1242/jeb.006692>

Pfau, T. (2005). A method for deriving displacement data during cyclical movement using an inertial sensor. *Journal of Experimental Biology*, 208(13), 2503–2514. <https://doi.org/10.1242/jeb.01658>

Pfau, T. (2006). Centre of mass movement and mechanical energy fluctuation during

gallop locomotion in the Thoroughbred racehorse. *Journal of Experimental Biology*, 209(19), 3742–3757. <https://doi.org/10.1242/jeb.02439>

Pfau, T., Ferrari, M., Parsons, K., & Wilson, A. (2008). A hidden Markov model-based stride segmentation technique applied to equine inertial sensor trunk movement data. *Journal of Biomechanics*, 41(1), 216–220. <https://doi.org/10.1016/j.jbiomech.2007.08.004>

PFAU, T., ROBILLIARD, J. J., WELLER, R., JESPERS, K., ELIASHAR, E., & WILSON, A. M. (2007). Assessment of mild hindlimb lameness during over ground locomotion using linear discriminant analysis of inertial sensor data. *Equine Veterinary Journal*, 39(5), 407–413. <https://doi.org/10.2746/042516407X185719>

Pfau, T., Spence, A., Starke, S., Ferrari, M., & Wilson, A. (2009). Modern Riding Style Improves Horse Racing Times. *Science*, 325(5938), 289–289. <https://doi.org/10.1126/science.1174605>

Pfau, T., Stubbs, N. C., Kaiser, L. J., Brown, L. E. A., & Clayton, H. M. (2012). Effect of trotting speed and circle radius on movement symmetry in horses during lunging on a soft surface. *American Journal of Veterinary Research*, 73(12), 1890–1899. <https://doi.org/10.2460/ajvr.73.12.1890>

- Rairán-Antolines, J. D., Chiquiza-Quiroga, D. F., & Parra-Pachón, M. Á. (2012). Implementación de neurocontroladores en línea. Tres configuraciones, tres plantas. *Ingeniería Y Universidad*, 16(1), 163–182.
- Reinoso, G. B. (1999). Modelado Orientado a Objetos : Una Evaluación Crítica. *Revista Facultad de Ingeniería*. Retrieved from <http://eii.pucv.cl/pers/gbustos/PDF/Evalua.PDF>
- Sheldon. (2014). La física tras el acelerómetro. Retrieved from <http://atomosybits.com/la-fisica-tras-el-acelerometro/>
- Starke, S. D., Robilliard, J. J., Weller, R., Wilson, A. M., & Pfau, T. (2009). Walk-run classification of symmetrical gaits in the horse: a multidimensional approach. *Journal of The Royal Society Interface*, 6(33), 335–342. <https://doi.org/10.1098/rsif.2008.0238>
- Starke, S. D., Willems, E., Head, M., May, S. A., & Pfau, T. (2012). Proximal hindlimb flexion in the horse: Effect on movement symmetry and implications for defining soundness. *Equine Veterinary Journal*, 44(6), 657–663. <https://doi.org/10.1111/j.2042-3306.2012.00576.x>
- Starke, S. D., Willems, E., May, S. A., & Pfau, T. (2012). Vertical head and trunk

movement adaptations of sound horses trotting in a circle on a hard surface. *The Veterinary Journal*, 193(1), 73–80. <https://doi.org/10.1016/j.tvjl.2011.10.019>

Starke, S. D., Witte, T. H., May, S. A., & Pfau, T. (2012). Accuracy and precision of hind limb foot contact timings of horses determined using a pelvis-mounted inertial measurement unit. *Journal of Biomechanics*, 45(8), 1522–1528. <https://doi.org/10.1016/j.jbiomech.2012.03.014>

Tecnolog, M. D. E., Informaci, A. S. D. E. L. A., & Decrelj, L. A. S. C. (2017). 5a 2017, 1–22.

Valentin, S., Davis, R., Wilson, A., & Pfau, T. (2010). The effect of Loading on the equine spine - A preliminary study. *Wiener Tierärztliche Monatsschrift*, 97(3–4), 74–80.

WALKER, A. M., WILSON, A. M., & PFAU, T. (2010). Comparison of kinematic symmetry index calculations and the effects of straight and circular trotting. *Equine Veterinary Journal*, 42, 482–487. <https://doi.org/10.1111/j.2042-3306.2010.00195.x>

WARNER, S. M., KOCH, T. O., & PFAU, T. (2010). Inertial sensors for assessment of back movement in horses during locomotion over ground. *Equine Veterinary Journal*, 42, 417–424. <https://doi.org/10.1111/j.2042-3306.2010.00200.x>

Anexos

reconocimiento

Datos Red Neuronal

Captura de Movimientos

Captura Movimiento 1 Captura Movimiento 2

Captura Movimiento 3

Captura Movimiento Test

Trasmision de datos

Iniciar Terminar

Red Neuronal

Numero Neuronas Capa Oculta

Error

Entrenar RNA

Codigos de Salida

	Salida 1	Salida 2
Movimiento 1	1	1
Movimiento 2	0	1
Movimiento 3	1	0

S1 S2

1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0

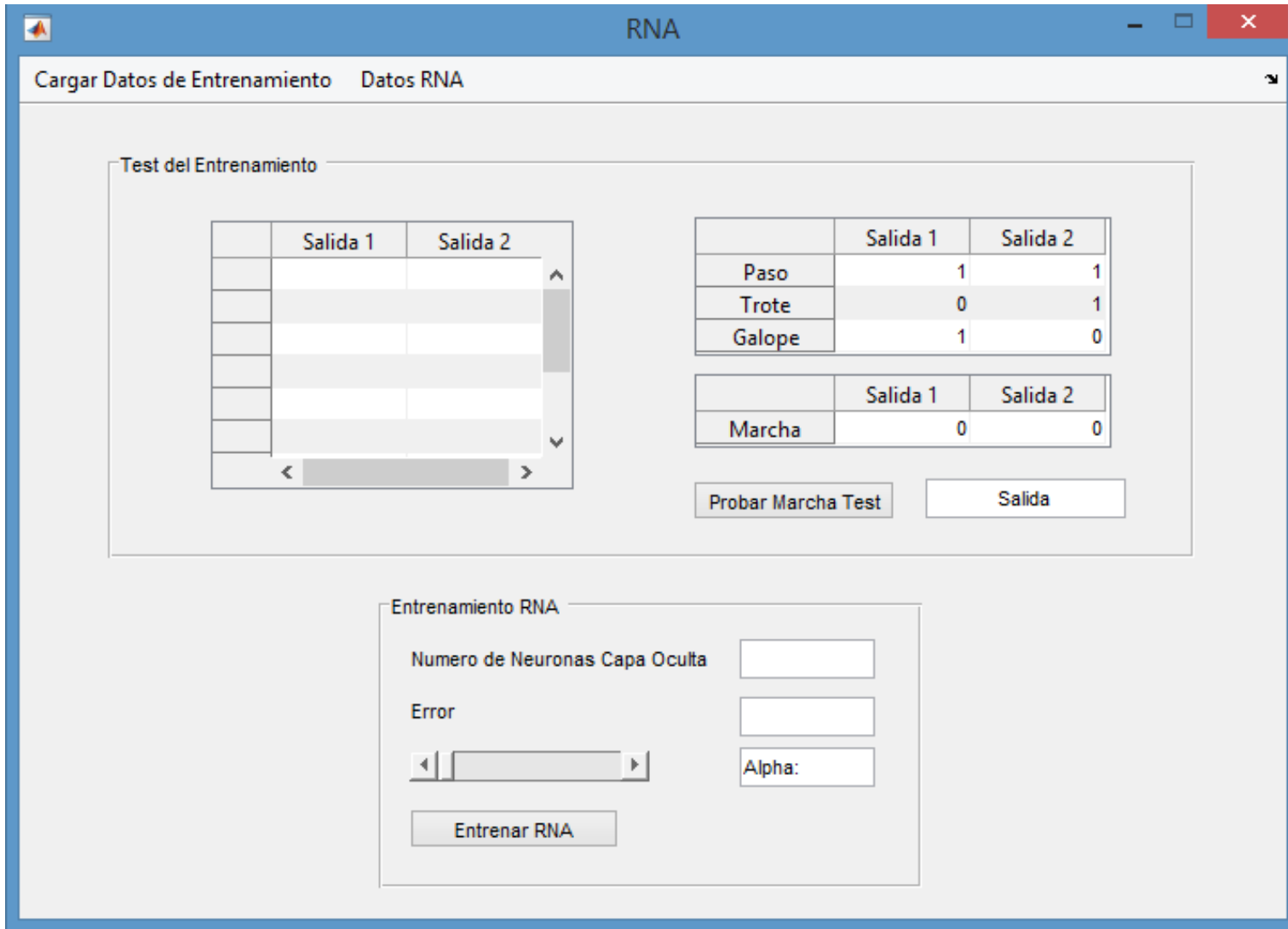
Probar Movimiento Test

Movimiento 1

	x1	y1	z1	x2	y2	z2
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0

Graficar

Anexo A Primer prototipo de software desarrollado



Anexo B Segundo prototipo de software desarrollado

Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 1									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
20	30	0,1	5198	0,4558	0,0463	0,2003	0,0074	0,4648	0,6058
		0,5	1085	0,4422	0,0211	0,2436	0,0167	0,4337	0,5854
		1	613	0,4668	0,014	0,1585	0,0079	0,3423	0,5872
	60	0,1	4473	0,506	0,0263	0,2288	0,0104	0,4711	0,6016
		0,5	928	0,3718	0,0382	0,22	0,0078	0,4206	0,572
		1	544	0,4693	0,0217	0,1812	0,0071	0,33	0,5579
	100	0,1	3996	0,376	0,0523	0,2119	0,0093	0,4968	0,6035
		0,5	898	0,4946	0,0205	0,2408	0,0146	0,3608	0,5717
		1	505	0,3415	0,0318	0,2333	0,0075	0,339	0,5501
60	30	0,1	10158	0,0748	0,0223	0,0407	0,0041	0,0478	0,1347
		0,5	1862	0,1152	0,0102	0,0318	0,0034	0,0854	0,1762
		1	1049	0,1063	0,0098	0,0281	0,0033	0,0258	0,1309
	60	0,1	7865	0,043	0,0103	0,0572	0,003	0,062	0,1573
		0,5	1508	0,0949	0,0112	0,0471	0,0033	0,0435	0,1206
		1	927	0,0428	0,0103	0,0214	0,0025	0,0338	0,1104
	100	0,1	7476	0,0582	0,0176	0,0328	0,0033	0,0409	0,0977
		0,5	1464	0,0731	0,0139	0,049	0,0035	0,0383	0,0992
		1	809	0,0796	0,0056	0,0342	0,0023	0,0196	0,0582

Anexo C Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 1

Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 2									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
25	30	0,1	6554	0,1172	0,0311	0,6067	0,2102	0,2025	0,592
		0,5	1404	0,0791	0,0282	0,579	0,2	0,1989	0,5507
		1	894	0,0828	0,0296	0,5392	0,1364	0,1824	0,5713
	60	0,1	5829	0,1055	0,0346	0,5935	0,1701	0,232	0,5912
		0,5	1186	0,0704	0,0236	0,5209	0,1306	0,1706	0,5976
		1	720	0,0495	0,0221	0,5393	0,1201	0,1674	0,547
	100	0,1	5261	0,1248	0,0328	0,5818	0,2379	0,2345	0,635
		0,5	1124	0,0539	0,0367	0,5657	0,1583	0,1928	0,6054
		1	648	0,0503	0,0122	0,517	0,1094	0,1852	0,5545
62,5	30	0,1	11996	0,1189	0,0151	0,5919	0,2637	0,1643	0,6898
		0,5	2429	0,0202	0,0061	0,387	0,0874	0,0936	0,6203
		1	1267	0,0184	0,0054	0,365	0,0546	0,115	0,6004
	60	0,1	9840	0,0628	0,0162	0,4688	0,1449	0,1386	0,6156
		0,5	2090	0,0349	0,0085	0,3927	0,1291	0,0914	0,5792
		1	1179	0,0244	0,0055	0,311	0,0744	0,0924	0,6117
	100	0,1	9487	0,0538	0,0082	0,5657	0,1385	0,1155	0,5751
		0,5	2010	0,0259	0,0072	0,3948	0,0793	0,2007	0,5782
		1	1141	0,032	0,0037	0,3422	0,0796	0,0884	0,5354

Anexo D Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 2

Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 3									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
20	30	0,1	9484	0,3151	0,0194	0,5132	0,171	0,353	0,6277
		0,5	2115	0,1147	0,0097	0,4146	0,1275	0,1729	0,5622
		1	1052	0,0694	0,0078	0,3877	0,0679	0,1109	0,3185
	60	0,1	8778	0,1805	0,022	0,4638	0,1505	0,1883	0,4638
		0,5	1777	0,0916	0,0204	0,438	0,1071	0,1065	0,4119
		1	965	0,0526	0,0068	0,3927	0,0727	0,1004	0,3597
	100	0,1	7451	0,205	0,0258	0,4412	0,166	0,2293	0,5373
		0,5	1540	0,1386	0,0103	0,424	0,131	0,144	0,4316
		1	948	0,0871	0,0096	0,4044	0,1168	0,0798	0,2898
60	30	0,1	17730	0,0311	0,0048	0,175	0,017	0,0646	0,1903
		0,5	3242	0,0144	0,0044	0,1246	0,0063	0,0842	0,132
		1	1697	0,0129	0,0032	0,112	0,0077	0,048	0,0901
	60	0,1	13651	0,0166	0,0027	0,1363	0,0149	0,0652	0,1681
		0,5	2868	0,0227	0,0032	0,1079	0,0104	0,055	0,1397
		1	1456	0,0119	0,0024	0,1682	0,0082	0,0646	0,1172
	100	0,1	11597	0,0182	0,0034	0,2125	0,016	0,0718	0,1503
		0,5	2462	0,015	0,0028	0,1536	0,0159	0,0513	0,1244
		1	1416	0,0075	0,0023	0,1235	0,0172	0,0766	0,1315

Anexo E Entrenamiento de la red en el tiempo con los datos de la sesión de Captura 3

Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 1									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
20	5	0,1	3734	0,0086	0,0056	0,0095	0,0003	0,0005	0,0083
		0,5	709	0,007	0,0049	0,0095	0,0015	0,0022	0,0078
		1	345	0,0055	0,0063	0,0085	0,0014	0,0027	0,0094
	15	0,1	1846	0,0073	0,0089	0,0093	0,0011	0,001	0,0117
		0,5	326	0,01	0,0063	0,0077	0,0021	0,0015	0,0097
		1	168	0,0058	0,0069	0,0119	0,0035	0,0024	0,0087
	30	0,1	1191	0,0104	0,0032	0,0116	0,0021	0,0023	0,008
		0,5	227	0,0069	0,0059	0,0111	0,002	0,0024	0,0109
		1	115	0,0101	0,0061	0,0138	0,002	0,0022	0,0115
60	5	0,1	3580	0,0065	0,0048	0,0085	0,0014	0,0014	0,0079
		0,5	967	0,0042	0,0075	0,0073	0,0006	0,0019	0,0093
		1	406	0,005	0,0063	0,0051	0,0017	0,0019	0,0116
	15	0,1	1732	0,008	0,006	0,0094	0,0015	0,0019	0,0087
		0,5	385	0,0049	0,0044	0,0086	0,002	0,003	0,0072
		1	200	0,0078	0,0081	0,0076	0,0006	0,0008	0,0085
	30	0,1	1305	0,0104	0,0053	0,0072	0,0016	0,0011	0,0074
		0,5	232	0,0086	0,0051	0,0088	0,0013	0,001	0,0097
		1	114	0,0053	0,0063	0,0062	0,0014	0,0022	0,009

Anexo F Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 1

Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 2									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
25	5	0,1	3860	0,0378	0,0105	0,0214	0,0025	0,0548	0,3433
		0,5	910	0,0273	0,0116	0,0126	0,0022	0,2188	0,3365
		1	466	0,0405	0,0049	0,0201	0,0016	0,0779	0,2726
	15	0,1	2172	0,0342	0,0077	0,0392	0,0041	0,0624	0,2702
		0,5	490	0,0183	0,0103	0,0346	0,0035	0,1234	0,3372
		1	214	0,0303	0,0058	0,0582	0,0037	0,0469	0,3025
	30	0,1	1610	0,0325	0,0095	0,0672	0,0049	0,0856	0,3066
		0,5	352	0,0387	0,0085	0,0408	0,0055	0,0506	0,1779
		1	166	0,052	0,0074	0,0233	0,0051	0,0548	0,2431
62,5	5	0,1	4614	0,0057	0,0058	0,0133	0,0047	0,0546	0,486
		0,5	1015	0,0077	0,0033	0,0213	0,0051	0,1368	0,3325
		1	421	0,0097	0,0069	0,0092	0,0009	0,0205	0,3753
	15	0,1	2470	0,0042	0,0059	0,0259	0,0046	0,0213	0,3722
		0,5	497	0,0055	0,0065	0,0184	0,0017	0,012	0,5247
		1	280	0,0069	0,008	0,0393	0,0033	0,0074	0,4018
	30	0,1	1998	0,0153	0,0035	0,0349	0,0019	0,0698	0,2696
		0,5	363	0,0089	0,0071	0,0342	0,002	0,0192	0,2876
		1	177	0,0121	0,0061	0,0116	0,0009	0,0772	0,6485

Anexo G Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 2

Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 3									
Porcentaje datos de entrenamiento	Numero de neuronas	Tasa de aprendizaje	ciclos realizados	Salidas esperadas					
				Paso		Trote		Galope	
				1	1	0	1	1	0
				Error respecto a las salidas esperadas					
25	5	0,1	4116	0,0051	0,0048	0,515	0,0889	0,0033	0,0202
		0,5	1019	0,0043	0,0056	0,3594	0,0603	0,0019	0,0243
		1	377	0,007	0,0054	0,4262	0,0322	0,0023	0,0189
	15	0,1	2165	0,0068	0,0086	0,5138	0,0376	0,0014	0,0453
		0,5	398	0,0076	0,0058	0,4057	0,0216	0,0042	0,0576
		1	184	0,0068	0,0057	0,4692	0,095	0,0022	0,0232
	30	0,1	1401	0,0067	0,0057	0,4477	0,0124	0,0036	0,0602
		0,5	280	0,0088	0,0038	0,6529	0,03	0,0048	0,0552
		1	135	0,0065	0,0064	0,3878	0,0447	0,0099	0,0643
62,5	5	0,1	4413	0,0035	0,0076	0,0088	0,0015	0,0539	0,1238
		0,5	903	0,0061	0,0065	0,0099	1E-04	0,011	0,2012
		1	425	0,0076	0,0058	0,0091	0,0016	0,0297	0,2173
	15	0,1	2211	0,0081	0,0047	0,0127	0,0017	0,1006	0,2327
		0,5	469	0,005	0,0057	0,0117	0,0016	0,1154	0,1944
		1	217	0,0071	0,0034	0,011	0,0028	0,0721	0,1825
	30	0,1	1745	0,0078	0,0037	0,0225	0,0022	0,1672	0,2204
		0,5	343	0,0074	0,0036	0,0208	0,0019	0,1894	0,2026
		1	186	0,0084	0,0039	0,0158	0,0034	0,1484	0,2111

Anexo H Entrenamiento de la red en la frecuencia con los datos de la sesión de Captura 3