



**UNIVERSIDAD MICHOACANA DE SAN
NICOLÁS DE HIDALGO**

FACULTAD DE INGENIERIA ELECTRICA

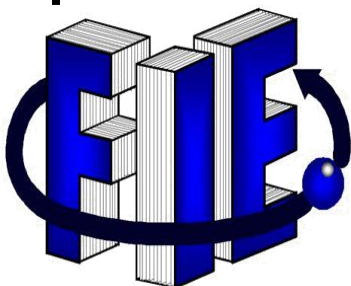
**INSTRUMENTACIÓN DE UN DISPOSITIVO
DE CALENTAMIENTO TEXTIL DE USO
DOMÉSTICO**

TESIS

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN ELECTRÓNICA

PRESENTA
JOEL EMIR CAMILO TORRES

DIRECTOR DE TESIS
Dr. Roberto Tapia Sánchez



Morelia Michoacán, Diciembre 2018

DEDICATORIA

A Dios por permitirme vida y salud para lograr alcanzar este objetivo en mi vida.

A mis familiares, en especial a mis amados padres, los cuales siempre me han brindado su confianza y el apoyo necesario para alcanzar esta gran meta.

A mi novia quien siempre ha estado presente en los momentos más difíciles durante esta estadía y por brindarme siempre su cariño y confianza.

A mis amigos con los que compartí la gran experiencia de poder cursar una carrera universitaria, haciendo mi estadía en ésta, mucho más agradable.

AGRADECIMIENTOS

A Dios por todo lo que me ha dado.

A la Universidad Michoacana de San Nicolás de Hidalgo por permitirme realizar y concluir mis estudios dentro de ella, y a la Facultad de Ingeniería Eléctrica.

A mis padres por todo su apoyo durante mi estadía en la universidad.

A todos los maestros que me dieron clases, en especial al Doctor Roberto Tapia Sánchez, por asesorarme durante la elaboración del presente proyecto.

Resumen

En la presente tesis, se presenta primeramente una recopilación de la evolución que ha tenido a lo largo de la historia la plancha de ropa.

Después, las características que presenta el dispositivo son descritas a detalle, con el objetivo de identificar cuáles son susceptibles a poder ser instrumentadas.

Se realizan principalmente tres mejoras con la ayuda de la instrumentación:

En primer lugar, el desarrollo de un sistema de seguridad capaz de evitar la mayor cantidad de accidentes que se puedan ocasionar durante cualquier tipo de descuido (sensor táctil y sistema de levantamiento).

En segundo lugar, hacer una conversión de un proceso mecánico a uno electrónico, ya que este aparato básicamente cuenta en su mayoría con sistemas que funcionan mecánicamente (bomba de agua).

En tercer lugar, la creación de un sistema capaz de hacer un reconocimiento textil, con la finalidad de que esta pueda decidir que temperatura es la más adecuada para el proceso de planchado (identificación del tipo de tela).

Palabras clave

Sistema, Arduino, Sensores, Matlab, Redes Neuronales.

Abstract

In the present thesis, it's present first a compilation of the evolution that the clothing iron has had throughout history.

Then, the characteristics presented by the device are described in detail, with the aim of identifying which are susceptible to being able to be instrumented.

Three improvements are mainly made with the help of instrumentation:

First, the development of a security system capable of avoiding the greatest number of accidents that may occur during any type of oversight (touch sensor and survey system).

Second, make a conversion from a mechanical process to an electronic one, since this device basically has mostly systems that work mechanically (water pump).

Third, the creation of a system capable of making a textile recognition, so that it can decide which temperature is the most suitable for the ironing process (identification of the type of fabric).

Keywords

System, Arduino, Sensors, MATLAB, Neural Networks.

DEDICATORIA.....	i
AGRADECIMIENTOS.....	ii
Resumen	iii
Palabras clave	iii
Abstract.....	iv
Keywords.....	iv
Lista de figuras	ix
Lista de tablas	xi
Lista de Códigos	xii
CAPITULO 1	1
1. Introducción.....	1
1.1 Definición del problema	1
1.2 Objetivos.....	1
1.2.1 Objetivos particulares.....	1
1.3 Justificación	2
1.4 Antecedentes	2
1.5 Metodología.....	4
1.6 Aportaciones	4
1.7 Descripción de capítulos	4
CAPITULO 2	6
2. Descripción de sistema	6
2.1 Introducción	6
2.2 Antecedentes	6
2.2.1 Origen y evolución de la plancha [1] [2].....	6
2.2.2 Historia de la plancha eléctrica.....	7
2.2.3 La primera plancha de vapor	7
2.2.4 La plancha sin cable	8
2.2.5 Origen de la tabla de planchar	8
2.2.6 Clasificación de todas las planchas	8
2.2.6.1 Plancha simple.....	9
La plancha simple funciona sólo con calor y presión.....	9
2.2.6.2 Planchas de vapor.....	10

2.2.6.3	Centros de planchado	11
2.2.6.4	Plancha vertical	11
2.3	Descripción del sistema	12
2.3.1	Partes que componen la plancha.....	13
2.3.1.1	Cubierta superior	13
2.3.1.2	Contenedor de agua.....	13
2.3.1.3	Mica de señalización	13
2.3.1.4	Asa.....	14
2.3.1.5	Tornillos sujetadores	14
2.3.1.6	Base sujetadora de cable.	14
2.3.1.7	Botón de rocío	14
2.3.1.8	Base Intermedia.....	14
2.3.1.9	Cable conductor.....	15
2.3.1.10	Cabe polar rojo	15
2.3.1.11	Cable polar azul.....	15
2.3.1.12	Cable dipolar amarillo.....	15
2.3.1.13	Piloto	15
2.3.1.14	Termostato.....	15
2.3.1.15	Bimetálicos.....	16
2.3.1.16	Manuales	16
2.3.1.17	Automáticos	16
2.4	Conclusiones	16
Capítulo 3	17
3.	Hardware Del Dispositivo	17
3.1	Introducción	17
3.2	Placa de desarrollo	17
3.2.1	Arduino.....	17
3.2.1.1	Arduino Mega	18
3.2.1.2	Arduino Uno.....	19
3.2.1.3	Arduino Nano	20
3.2.1.4	Arduino Mini.....	21
3.2.2	Raspberry.....	23

3.2.2.1	Hardware	23
3.2.2.2	Procesador	23
3.2.2.3	RAM.....	24
3.2.2.4	Periféricos.....	24
3.2.2.5	Video	25
3.2.2.6	Raspberry Pi 3 Modelo B+.....	25
3.2.2.7	Raspberry Pi 3 Modelo A+	26
3.2.2.8	La Raspberry Pi Zero	27
3.2.3	Selección.....	28
3.3	Sensor de posición	29
3.3.1	sensor de inclinación - AT407.....	29
3.3.2	KEYES KY-017	30
3.3.3	MPU-9250	31
3.4	Sistema de levantamiento	32
3.5	Sistema táctil.....	34
3.6	Sistema de bombeo	35
3.6.1	Bomba.....	35
3.7	Detección del nivel de líquido	36
3.8	Cámara	37
3.8.1	OV7670	38
3.8.2	LS-Y201	39
3.8.3	Mini cámara espía con gatillo para foto o video	40
3.9	Conclusiones	41
Capítulo 4	42
4.	Software Desarrollado	42
4.1	Introducción	42
4.2	Algoritmo de posición	42
4.3	Algoritmo del sistema táctil.....	43
4.4	Algoritmo del sistema de levantamiento	44
4.5	Algoritmo para el sistema de bombeo	45
4.6	Algoritmo de reconocimiento textil.....	48
4.6.1	Principios	54

4.6.2	Redes neuronales	55
4.6.3	Obtención de la base de datos.....	56
4.6.3.1	Agrupación de datos.....	58
4.6.3.2	Asignación de identificador	60
4.6.4	Ingreso de base de datos a una RN (descriptores y clases)	60
4.6.5	Matriz de Confusión y eficiencia.....	62
4.6.6	Función Net	63
4.6.7	Conversión a código C	65
4.6.8	Implementación en Arduino	68
4.7	Conclusiones	70
Capítulo 5	71
5.	Conclusiones Y Trabajos Futuros	71
5.1	Conclusiones	71
5.2	Trabajos futuros	72
Bibliografía.....	73

Lista de figuras

<i>figura 1. 1 Diagrama de bloques.</i>	4
<i>figura 2. 1 Plancha simple.</i>	9
<i>figura 2. 2 Plancha de vapor.</i>	10
<i>figura 2. 3 Centro de planchado.</i>	11
<i>figura 2. 4 Plancha vertical.</i>	12
<i>figura 3. 1 Arduino Mega 2560.</i>	18
<i>figura 3. 2 Arduino Uno.</i>	19
<i>figura 3. 3 Arduino Nano</i>	20
<i>figura 3. 4 Arduino Mini.</i>	21
<i>figura 3. 5 Raspberry Pi B+</i>	26
<i>figura 3. 6 Raspberry Pi modelo A+</i>	27
<i>figura 3. 7 Raspberry Pi Zero W.</i>	28
<i>figura 3. 8 Raspberry Pi Zero</i>	28
<i>figura 3. 10 Forma de trabajo del sensor de inclinación AT407</i>	30
<i>figura 3. 9 Sensor de inclinación AT407.</i>	30
<i>figura 3. 11 Modulo KEYES KY-017</i>	31
<i>figura 3. 12 Sensor MPU-9250.</i>	32
<i>figura 3. 13 Servomotor.</i>	34
<i>figura 3. 14 Bosquejo del funcionamiento del sensor táctil.</i>	35
<i>figura 3. 15 Bomba de agua sumergible.</i>	36
<i>figura 3. 16 Sensor de nivel de líquido On / Off.</i>	37
<i>figura 3. 17 Cámara OV7670.</i>	39
<i>figura 3. 18 Cámara LS-Y201</i>	40
<i>figura 3. 19 Mini cámara espía con gatillo para foto o video.</i>	41
<i>Figura 4. 1 Creación de una RNA</i>	49
<i>Figura 4. 2 Importación de los datos de las flores de iris.</i>	49
<i>Figura 4. 3 Valores de entrenamiento de la RNA.</i>	50
<i>Figura 4. 4 Neuronas de nuestra RNA</i>	50
<i>Figura 4. 5 Entrenamiento de la RNA</i>	51
<i>Figura 4. 6 Ventana de Neural Network Training.</i>	51
<i>Figura 4. 7 Matriz de confusión para los datos de las flores de iris.</i>	52
<i>Figura 4. 8 Esquema de una RNA</i>	56
<i>Figura 4. 9 Matriz de confusión de ejemplo</i>	62
<i>Figura 4. 10 Matriz de confusión de la base de datos de las telas.</i>	63
<i>Figura 4. 11 Función net</i>	64
<i>Figura 4. 12 Uso de la función net.</i>	64
<i>Figura 4. 13 Generación de la función Func_Net.</i>	65
<i>Figura 4. 14 Comprobación de los resultados de ambas funciones.</i>	66
<i>Figura 4. 15 Ingreso de la función a convertir a código C.</i>	66
<i>Figura 4. 16 Especificaciones de entradas de la función Func_Net.</i>	67

<i>Figura 4. 17 Selección de código a convertir listo para ser generado.</i>	<i>67</i>
<i>Figura 4. 18 Archivos .c y .h generados por Matlab Coder.</i>	<i>68</i>

Lista de tablas

<i>Tabla 1. 1 Evolución de la plancha de ropa</i>	<i>3</i>
<i>Tabla 3. 1 Especificaciones del Arduino Mega.....</i>	<i>18</i>
<i>Tabla 3. 2 Especificaciones del Arduino Uno.</i>	<i>19</i>
<i>Tabla 3. 3 Especificaciones del Arduino Nano.</i>	<i>20</i>
<i>Tabla 3. 4 Especificaciones del Arduino Mini.....</i>	<i>21</i>
<i>Tabla 3. 5 Especificaciones de las cuatro placas de desarrollo Arduino</i>	<i>22</i>
<i>Tabla 3. 6 Especificaciones de la Raspberry Pi B+.....</i>	<i>26</i>
<i>Tabla 3. 7 Especificaciones de la Raspberry modelo A+</i>	<i>27</i>
<i>Tabla 3. 8 Especificaciones de las Raspberry Pi Zero y Zero W.....</i>	<i>28</i>
<i>Tabla 3. 9 Especificaciones del sensor AT407</i>	<i>30</i>
<i>Tabla 3. 10 Especificaciones del módulo KEYES KY-017.....</i>	<i>30</i>
<i>Tabla 3. 11 Especificaciones del sensor MPU-9250.....</i>	<i>32</i>
<i>Tabla 3. 12 Especificaciones del Servomotor MG995.</i>	<i>33</i>
<i>Tabla 3. 13 Especificaciones de la bomba de agua sumergible.....</i>	<i>36</i>
<i>Tabla 3. 14 Especificaciones del sensor de nivel de líquido On/Off</i>	<i>37</i>
<i>Tabla 3. 15 Especificaciones de la Cámara OV7670</i>	<i>38</i>
<i>Tabla 3. 16 Especificaciones de la cámara LS-Y201.....</i>	<i>39</i>
<i>Tabla 3. 17 Especificaciones de la mini cámara espía con gatillo para foto o video.</i>	<i>40</i>
<i>Tabla 4. 1 Telas utilizadas para la generación la base de datos.....</i>	<i>58</i>
<i>Tabla 4. 2 Esquema general de la base de datos.....</i>	<i>59</i>
<i>Tabla 4. 3 Esquema general de la base de datos con clases</i>	<i>60</i>

Lista de Códigos

<i>Código 4. 1 Algoritmo de posición utilizado para el sistema.</i>	43
<i>Código 4. 2 Algoritmo del sensor táctil.</i>	44
<i>Código 4. 3 Algoritmo del sensor de levantamiento.</i>	45
<i>Código 4. 4 Algoritmo del sensor de bombeo.</i>	46
<i>Código 4. 5 Algoritmo del funcionamiento vertical.</i>	46
<i>Código 4. 6 Algoritmo del funcionamiento horizontal.</i>	47
<i>Código 4. 7 Algoritmo de la red iris.</i>	53
<i>Código 4. 8 Algoritmo de la red iris.</i>	53
<i>Código 4. 9 Algoritmo de la red iris.</i>	54
<i>Código 4. 10 Algoritmo para la obtención de la base de datos.</i>	57
<i>Código 4. 11 Algoritmo para la obtención de la base de datos.</i>	58
<i>Código 4. 12 Algoritmo con nuestros descriptores y clases en él.</i>	61
<i>Código 4. 13 Código generado por Matlab Coder en C.</i>	69
<i>Código 4. 14 Código editado para el funcionamiento de nuestro sistema.</i>	69

CAPITULO 1

1. Introducción

1.1 Definición del problema

Con el avance de la tecnología se han ido implementado nuevas maneras de hacer la vida del hombre mucho más cómoda, un claro ejemplo de esto son los aparatos eléctricos y electrónicos que realizan un trabajo que podría ser esencialmente pesado o difícil para el hombre.

Existen también aparatos que a pesar de que no han sido actualizados recientemente en el ámbito tecnológico, brindan una gran ayuda para el hombre, como lo es una plancha de ropa. Este aparato en la actualidad cumple muy bien con su función a realizar, que es la de quitar arrugas a las prendas de vestir, no obstante, podría ser mejorada para brindar una mayor comodidad al usuario a la hora de ser utilizada.

1.2 Objetivos

Realizar la instrumentación de una plancha de ropa con la finalidad de que ésta tenga una mayor facilidad de uso.

1.2.1 Objetivos particulares

Identificar la posición horizontal del instrumento para poder realizar una o varias acciones.

Identificar la posición vertical del instrumento para poder realizar una o varias acciones.

Reemplazar los botones mecánicos por sensores táctiles

Diseño del sistema de seguridad

1.3 Justificación

Sin duda alguna las nuevas tecnologías hacen mucho más placentera la vida del hombre, es por eso por lo que muchos aparatos eléctricos y electrónicos son actualizados conforme se van descubriendo nuevas y mejores formas de realizar un trabajo, pero desgraciadamente aun con las mejoras, siguen existiendo (aunque en menor probabilidad) los accidentes con estos aparatos.

Además, estas nuevas mejoras en los aparatos generalmente siempre tienden a aumentar su precio con cada mejora que se les realiza.

Es por esto, por lo que al realizar la instrumentación de una plancha de ropa se tratara de evitar el mayor riesgo posible de accidentes relacionados con quemaduras de tela y aun mayor con incendios causados por descuidos al haber dejado el aparato encendido y sin ningún tipo de supervisión. Así también pretenderá hacer mucho más eficiente una plancha de ropa con la posibilidad de que pueda ser accesible económicamente hablando.

1.4 Antecedentes

Sabemos que una plancha de ropa es un instrumento necesario a la hora de estar presentable ante la sociedad, sin embargo, este aparato no siempre ha sido como lo conocemos hoy en día. La evolución de la plancha de ropa se ha venido dando a través de lo largo de la historia, más específicamente desde el siglo IV en Grecia y China.

Su evolución fue sin duda alguna conforme las nuevas tecnologías se iban mejorando, puesto que las planchas comenzaron siendo objetos pesados de gran tamaño o de forma cilíndrica para poder alisar las prendas.

Posteriormente, empezaron a ser de metal para ser calentadas en las brasas y de esta manera llevarlas a la ropa y quitar las arrugas. Seguidamente con el avance de la tecnología la plancha también se fue actualizando, como contar con un pequeño depósito para contener el carbón o gas.

Actualmente una mejora que hasta en nuestros tiempos existe es la utilización de la energía eléctrica, la cual fue utilizada para generar un arco eléctrico que, aunque no se aprovechó de la manera correcta, hoy en día su uso es la manera más rápida y eficiente con la que este aparato funciona.

Como se menciona anteriormente, el uso de la electricidad en una plancha de ropa moderna es la mejor manera de hacer que esta cumpla con su función de una manera rápida, eficiente y limpia, pues no tiene problemas de peso, de que manche las prendas o de un tiempo extremadamente largo para calentar.

En la Tabla 1.1 se aprecia la evolución de la plancha de ropa a través del tiempo.

Tabla 1. 1 Evolución de la plancha de ropa

	SIGLO IV	Plancha que se pasaba sobre las brasas para marcar los pliegues de la ropa de lino
	SIGLO XV	Alisador de madera europea. Se utilizaba en frio ya que la goma vegetal utilizada como almidón no permitía utilizar el calor
	SIGLO XIX	Plancha con un funcionamiento a base de gas.
	SIGLO XVII	Plancha hueca en la cual se introducían trozos de madera ardiendo.
	AÑO 1924	Plancha eléctrica
	AÑO 1926	Plancha de vapor.
	AÑO 2017	Plancha eléctrica echa de acero, aluminio y adherente.

1.5 Metodología

Se realizará una investigación previa para conocer los diferentes tipos de elementos electrónicos (sensores, software, hardware, etc.) que son necesarios para el funcionamiento correcto del dispositivo.

Una vez se cuente con ellos se procederá a realizar las pruebas necesarias con cada uno de los elementos para poder determinar su correcto funcionamiento y de esta manera poder explotar todo su potencial.

Se utilizará cada una de las diferentes propiedades de cada elemento electrónico, con el fin de agrupar todas y cada una de ellas en el dispositivo final, teniendo como resultado un producto mucho más eficiente y con características que facilitaran más su uso a comparación de los productos actuales en el mercado.

La Figura 1.1 muestra el diagrama a bloques que resume la metodología descrita anteriormente.



figura 1. 1 Diagrama de bloques.

1.6 Aportaciones

Se implementará una mejora en un aparato ya existente, con la finalidad de que esta pueda ser utilizada de una manera más fácil.

1.7 Descripción de capítulos

En el Capítulo 1 se muestra una pequeña introducción sobre el beneficio que se tiene gracias a las nuevas tecnologías, se define del problema a resolver que en realidad no es que exista uno como tal, sino que se trata de realizar una mejora sobre un instrumento muy utilizado en la vida cotidiana del hombre. Así también se

puede encontrar los antecedentes de tal instrumento que presenta como ha sido la evolución de este a través del tiempo.

En el Capítulo 2 se hace un breve resumen de la historia que a tenido la plancha de ropa. También se hace una pequeña explicación de las partes que la componen y el funcionamiento que estas tiene.

En el Capítulo 3 se presenta una recopilación de los dispositivos y sensores que se utilizan en el sistema, para que este pueda funcionar de la mejor manera posible. También se muestran sus especificaciones de cada uno de ellos.

En el Capítulo 4 se muestran los códigos que hacen funcionar a cada uno de los elementos con los que cuenta el dispositivo, y que serán incorporados al sistema. También se hace referencia a cómo crear una Red Neuronal Artificial (RNA) en la plataforma de Matlab, así como el proceso de convertir un código Matlab a lenguaje C.

En el Capítulo 5 aparecen las conclusiones y observaciones obtenidas de la implementación de este proyecto, además de mencionar los trabajos futuros que aran que este mismo proyecto mejore significativamente.

CAPITULO 2

2. Descripción de sistema

2.1 Introducción

Para poder comprender para qué y cómo funciona un aparato electrodoméstico es necesario saber de dónde proviene y cuáles fueron las causas que llevaron a su aparición. Es por eso que en este capítulo se desarrolla una investigación de sus características y antecedentes, con el fin de que se comprenda el funcionamiento de una plancha para ropa, así como cada uno de los componentes que la integran.

2.2 Antecedentes

La plancha es uno de los aparatos domésticos presente en la mayoría de los hogares, pues es de gran ayuda a la hora de querer tener una prenda sin arrugas y presentar una buena imagen personal.

2.2.1 Origen y evolución de la plancha [1] [2]

Este aparato electrodoméstico tiene sus inicios en el siglo IV. Los griegos usaban una barra de hierro cilíndrica calentada, similar a un rodillo de amasar, que se pasaba sobre las ropas de lino para marcar los pliegues.

Un par de siglos después, los romanos planchaban la ropa y plisaban los vestidos con una especie de mazo, que más que planchar martilleaba las arrugas haciendo que el planchado se convirtiera en tarea muy pesada sólo apta para esclavos.

Los chinos del siglo IV alisaban la seda con unos recipientes de latón de tamaño mediano provistos de un mango. En su interior se introducían trozos de madera aromática ardiendo, para que su suave aroma impregnase los tejidos y las ropas. Pese a ser una plancha antigua, era muy avanzada para la época.

La primera plancha europea fue el alisador de madera, vidrio o mármol que hasta el siglo XV se utilizó también en frío, ya que la goma vegetal utilizada como almidón no permitía usar el calor.

A aquellos modelos, les sucedieron unas planchas que se calentaban mediante agua hirviendo, gas e incluso alcohol. También aparecieron las planchas de carbón, que alcanzaron bastante popularidad.

Cuando en el siglo XIX se puso de moda el gas de alumbrado algunos pensaron trasladar aquella posibilidad a la plancha, y lo hicieron, naciendo así la plancha de gas. Pero en estos artefactos se producían fácilmente escapes y con ello explosiones que aconsejaban llevar la ropa arrugada o perder la vida en el intento.

2.2.2 Historia de la plancha eléctrica

La primera plancha eléctrica: la idea se le ocurrió a un norteamericano de Nueva Jersey llamado Henry W. Seely en 1882, inventor de la plancha eléctrica [1].

Este artilugio disponía de un arco voltaico abierto que no pudo ser aprovechada por dos razones importantes: en las casas no había conexión a la red eléctrica, y no existía aún un termostato regulable que evitara que las prendas se quemaran, por lo que se llamó al invento de Henry “el invento de los tontos”, y aunque la idea era excelente, su aplicación no sería posible.

En 1897 un camarero de Minneapolis, Charles Carpenter, dotó a la plancha eléctrica de Henry Seely de una resistencia espiral, solucionando algunos de los problemas que el artilugio presentaba [1].

2.2.3 La primera plancha de vapor

Para quienes no podían contar con el beneficio de la electricidad se crearon en 1926 las primeras planchas de vapor a casi el doble de precio, porque tenían una ventaja: no chamuscaban la ropa debido a que rodeaban la prenda de una humedad uniforme.

El inventor de la plancha de vapor fue una pequeña empresa neoyorquina de limpieza química llamada Eldec Company, la cual desarrolló el invento y poco a poco lo fue perfeccionando [1].

2.2.4 La plancha sin cable

En 1978 la firma Lyon & Brandfield Limited lanzó la plancha sin cable y la plancha sin pilas, que funcionaba con gas, mediante recambios de butano o benzina incorporado que le daba una autonomía de cinco horas.

La plancha sin cables, invento de los ingleses Howard O. Freckleton y John Stanley Bird, había sido concebida para utilizar en lugares a los que no llegaba la red eléctrica, para las acampadas y el mundo del caravaning, por ejemplo [1].

2.2.5 Origen de la tabla de planchar

La tabla de planchar ya había aparecido en el siglo XIX: armatostes plegables que a veces se convertían en una trampa para los dedos de las amas de casa. Eran tablas pesadas e incómodas y requerían un espacio excesivo para su almacenamiento: casi era preferible utilizar la mesa de la cocina.

Atentos a esa posibilidad, el inventor Bernard Theeten obtuvo en 1986 la Medalla de Oro del Salón de Inventores de Bruselas por su tabla de planchar reversible, con la patente número 4779365. Concebida para ser colocada sobre la mesa, sobre cuya superficie se elevaba hasta unos doce centímetros [1].

2.2.6 Clasificación de todas las planchas

Características de las planchas de ropa

Una plancha es un aparato electrodoméstico empleado para eliminar las arrugas de la ropa y prendas de vestir por medio de la aplicación de calor, presión y generalmente también vapor.

Además de calor es necesario aplicar presión a las prendas, por lo que las planchas deben tener cierto peso que evite esfuerzo al usuario. Pero tampoco debe pesar tanto que dificulte su manejo.

Actualmente existen modelos muy completos de planchas para ropa que incluyen emisión de vapor, planchado vertical, planchado en seco y otras funciones como selección y ajuste de temperatura, lo que nos permite ajustarla al material

correspondiente. Casi todas las planchas actuales son capaces de alisar incluso tejidos difíciles, como el lino o los tejidos sintéticos [3].

Se pueden distinguir varios tipos de planchas:

2.2.6.1 Plancha simple

La plancha simple funciona sólo con calor y presión. La plancha eléctrica: utiliza calor generado por una resistencia a partir de la corriente eléctrica. Las amas de casa todavía no la podían utilizar ya que no existía la conexión a la red eléctrica y no se había inventado aun el termostato.

El calor se producía en una resistencia colocada en el interior de la plancha que con el paso de la corriente eléctrica se calentaba por el efecto Joule. Esto consiste en que la circulación de corriente eléctrica por la resistencia desprende más o menos cantidad de calor dependiendo de tres factores: el valor del cuadrado de la intensidad, la resistencia y el tiempo de funcionamiento del aparato eléctrico [4].

En la Figura 2.1 se puede apreciar una plancha simple tal como se menciona anteriormente.



figura 2. 1 Plancha simple.

2.2.6.2 Planchas de vapor

Las planchas de vapor funcionan con calor, presión y vapor.

Cuenta con un generador independiente de vapor que hace de la tarea del planchado sea algo más rápido y fácil; el vapor es expulsado por lo general mediante pequeños orificios en la superficie metálica de planchado, mediante la cual se va dosificando el vapor hacia la prenda.

Las planchas a vapor emiten dos tipos de vapor: vapor continuo y super vapor.

El vapor continuo es un vapor de emisión constante que mantiene siempre su intensidad, independiente si el depósito de agua se encuentra lleno o vacío. La medida ideal de emisión de vapor continuo es de alrededor de 20 o 30 gramos por minuto. Estas contienen un elemento de calefacción eléctrico de alambre de nicromo (níquel-cromo) que es envuelto en un revestimiento termorresistente y aislado eléctricamente, y colocado en la base de la plancha. En la Figura 1.2 se puede apreciar una plancha de vapor que en la actualidad son unas de las más vendidas en el mercado. La Figura 2.2 muestra una imagen de este tipo de plancha.



figura 2. 2 Plancha de vapor.

La corriente eléctrica va al elemento de calefacción a través de un termostato que detiene el flujo de la corriente cuando la plancha está bastante caliente, y la enciende cuando está muy fría. Un circuito controlado por un termostato calienta la plancha de vapor [4].

2.2.6.3 Centros de planchado

Los centros de planchado pueden ser considerados como: las planchas de vapor avanzadas que incorporan prestaciones adicionales. Los centros de planchado son similares a las planchas, pero disponen de un calderín de agua independiente. Son más caros que las planchas, y tardan más tiempo en estar listos para su uso. Sin embargo, pesan menos que las planchas llenas de agua y el chorro de vapor dura más tiempo. Por este motivo pueden compensar a aquellos usuarios que planchan grandes cantidades de ropa.

Un centro de planchado (véase Figura 1.3), es un conjunto formado por una plancha de ropa y un generador-depósito de vapor independiente. Al tener un depósito de agua separado de la unidad de planchado son capaces de generar más vapor que una plancha de ropa convencional. Tal cantidad de vapor facilita la tarea del planchado. Los centros de planchado tardan más en calentarse que una plancha convencional y su precio es más elevado [5] [6].



figura 2. 3 Centro de planchado.

2.2.6.4 Plancha vertical

Una plancha vertical permite planchar la ropa sin que haya contacto directo de la plancha con el tejido. Es decir, se puede planchar cualquier tejido sin tener que utilizar la tabla de planchar, solamente teniendo colgada la prenda y utilizando la

fuerza del vapor. El vapor consigue penetrar en los tejidos y relajar las fibras, que quedan automáticamente planchadas. La Figura 2.4 muestra este tipo de planchas.



figura 2. 4 Plancha vertical.

No hay que ver el planchado vertical como sustituto del planchado tradicional, sino más bien como un complemento. Si la prenda está muy arrugada o es de un tejido “rebelde” necesitarás utilizar tu plancha tradicional como toda la vida [7].

2.3 Descripción del sistema

La plancha de ropa está formada por una placa metálica térmica a la que se le llama suela. También cuenta con un mango, generalmente ergonómico y aislante del calor, para sujetarla mientras se pasa por la prenda para alisarla.

La plancha de vapor cuenta con un depósito para agua, que al calentarse sale en forma de vapor por las perforaciones de la suela de la plancha. Esto facilita el planchado, además de que protege la ropa contra quemaduras. También puede contar con la función de ráfaga de vapor, que consiste en dejar salir súbitamente una gran cantidad de vapor con tan solo oprimir un botón. Esto permite planchar prendas delicadas únicamente con el vapor, sin necesidad de tocarlas con la plancha.

La mayoría de las planchas modernas disponen de mecanismos de seguridad como el apagado automático si se deja inmóvil demasiado tiempo posada en posición horizontal, o si se deja encendida en el soporte de la tabla de planchar, así como sistema antigoteo automático y función de descalcificación automática.

Muchos modelos también ofrecen recubrimientos cerámicos o antiadherentes para mejorar su deslizamiento, y evitar daños y que el tejido se quemara o se pegue, brindando así un impecable acabado y un mejor cuidado de la ropa [8].

2.3.1 Partes que componen la plancha

2.3.1.1 Cubierta superior

La cubierta superior es de plástico, y tiene una forma de semicurvatura a los lados y una punta al final, la parte posterior es recta. Tiene un mango superior que sirve como agarradera, en ambos tiene un espacio alargado en forma de curva en donde se puede apreciar el contenedor de agua.

En general la función de la cubierta superior es cubrir y proteger la base intermedia, el contenedor de agua, el circuito eléctrico y la plataforma secundaria en dado caso de golpes o en caso de algún fluido de por medio.

2.3.1.2 Contenedor de agua

El contenedor de agua es de plástico, en color gris grafito con una transparencia del 80%, se encuentra dentro de la cubierta superior y su forma es semejante a la cubierta superior.

El contenedor sirve para almacenar agua, la cual es introducida a través de el orificio frontal. La finalidad del contenedor es, almacenar agua y ser fuente de distribución de la misma.

2.3.1.3 Mica de señalización

La mica de señalización es un plástico luminiscente. Este plástico solo funciona cuando el piloto este encendido, es decir cambia la intensidad de color, esto es debido a que el piloto es un foco y este emite luz.

Por lo tanto, si la luz es roja y destellante implica que está encendida la plancha; la mica únicamente nos dice si la plancha esta encendida o no a través del tono de luz que presente.

2.3.1.4 Asa

La Asa está hecha a base de plástico, en la parte frontal de la curva se ubica una tapa que abre/cierra de manera vertical, cuya función es bloquear o permitir el paso de agua del exterior hacia el contenedor.

La finalidad del asa es cubrir el espacio existente en el mango de la cubierta superior, además de cubrir las 2 extracciones del contenedor del agua.

2.3.1.5 Tornillos sujetadores

La función de los tornillos sujetadores, es sujetar bajo presión y unir con precisión las piezas según estas tengan los orificios de pijas correspondientes.

2.3.1.6 Base sujetadora de cable.

Esta base se localiza en la parte posterior de la cubierta superior, la finalidad de esta base, es presionar al cable alimentador de corriente para que este no se mueva y a la larga no genere una discontinuidad de corriente.

2.3.1.7 Botón de rocío

El botón de rocío trabaja bajo un principio hidráulico, es decir, al presionar el botón permite el paso del agua desde el contenedor hasta la suela de acero, es decir al exterior; cuando se deja de presionar el botón el resorte vuelve a su forma original, esto genera una obstrucción al paso y regreso del agua.

2.3.1.8 Base Intermedia

La base intermedia, es una base que se encuentra en la parte media de la plancha, se ubica debajo de la cubierta superior y del contenedor de agua, así mismo está por encima de la plataforma metálica secundaria.

La finalidad de toda esta base es cubrir las plataformas metálicas y a su vez separar el contenedor del agua de las plataformas y así evitar el derretimiento del contenedor.

2.3.1.9 Cable conductor

El cable conductor, son filamentos de cobre forrados por un plástico aislante de electricidad, en la parte final del cable esta una clavija, cubierta de un plástico aislante con 2 varillas metálicas que sirven como. El cable sirve como conductor y alimentador de corriente.

2.3.1.10 Cabe polar rojo

Son filamentos de aluminio forrados por un plástico texturizado de color rojo, este cable es de polaridad negativa, y su función es regresar la corriente que pasa por el circuito a la fuente que la emite.

2.3.1.11 Cable polar azul

Son filamentos de aluminio forrados por un plástico texturizado azul, que sirve como aislante de electricidad, su función es recibir la carga directa y distribuirla por el circuito.

2.3.1.12 Cable dipolar amarillo

Son filamentos de cobre forrados por plástico texturizados de color amarillo que es un aislante eléctrico. Al ser dipolar, habrá tanto polaridad positiva como polaridad negativa. La función de estos cables es transmitir desde la resistencia hasta el piloto la corriente.

2.3.1.13 Piloto

El piloto, es un pequeño foco de cristal de bajo voltaje, tiende a trabajar bajo corriente en serie. El objetivo del piloto es encender si es que el termóstato está trabajando, la intensidad de la luz variara de acuerdo a la intensidad de temperatura que se necesite [9].

2.3.1.14 Termostato

Un termostato es el componente de un sistema de control simple que abre o cierra un circuito eléctrico en función de la temperatura.

2.3.1.15 Bimetálicos

El bimetálico, consiste en dos láminas de metal unidas con diferente coeficiente de dilatación térmico. Cuando la temperatura cambia, la lámina cambia de forma automáticamente, actuando sobre unos contactos que cierran un circuito eléctrico.

Pueden estar normalmente abiertos o normalmente cerrados, cambiando su estado cuando la temperatura alcanza el nivel para el que están preparados.

2.3.1.16 Manuales

Son los que requieren intervención humana para regresar a su estado inicial, como los termostatos de seguridad que realizan una función en caso de que la temperatura alcance niveles peligrosos.

2.3.1.17 Automáticos

Regresan a su estado inicial sin necesidad de intervención humana. Actúan de una forma totalmente automática, de ahí su aplicación actual en gran parte de los hogares [10].

2.4 Conclusiones

En el presente capítulo se presentaron los diferentes tipos de planchas que existen en el mercado, además de las características y componentes que estas tienen y el funcionamiento de cada uno de ellos. Por lo que se puede identificar cada una de las partes que podrían ser susceptibles a ser modificadas, para tener un mejor funcionamiento o agregar más características a estos.

En este caso se podría agregar un sensor que detecte la posición del aparato con la finalidad de tener un sistema de apagado automático en determinadas condiciones, como cuando este quede inmóvil por un largo periodo de tiempo, logrando evitar accidentes. Además del termostato, el cual puedan ser ajustado automáticamente, dependiendo del tipo de tela que se vaya a planchar, logrando así, hacer este proceso totalmente automático.

Capítulo 3

3. Hardware Del Dispositivo

3.1 Introducción

En este capítulo se describen los diferentes elementos desarrollados que componen al sistema. Así también como los medios en los cuales se desarrollan. Para lo siguiente se describirá cada uno de estos elementos y la función con la que deberá cumplir en cada momento.

3.2 Placa de desarrollo

Para la placa de desarrollo en que se realiza tanto la implementación de hardware como la creación del software se toma en cuenta la plataforma de Arduino y Raspberry Pi como las posibles opciones a ser utilizadas. Las cuales son en la actualidad unas de las placas de desarrollo más populares debido a su fácil uso y adquisición.

3.2.1 Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas y convertirla en una salida: activar un motor, encender un LED y publicar algo en línea. Puede indicársele a la tarjeta qué debe hacer, enviando un conjunto de instrucciones al microcontrolador en la tarjeta. Arduino tiene una gama completa de productos oficiales incluyendo tableros, módulos, shields y kits [11].

A continuación, se muestra algunas de las placas de desarrollo de Arduino, así como módulos que son una versión más pequeña de los tableros clásicos.

3.2.1.1 Arduino Mega

El Arduino Mega es un tablero de microcontrolador basado en el ATmega 2560 (véase Figura 3.1). Cuenta con 54 pines de entrada / salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP, y un botón de reinicio [12], las especificaciones se muestran en la Tabla 3.1.

Tabla 3. 1 Especificaciones del Arduino Mega.

Arduino Mega 2560	
Microcontrolador	ATmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pernos digitales de E / S	54 (de los cuales 15 proporcionan salida PWM)
Clavijas de entrada analógica	16
Corriente DC por Pin de E / S	20 mA
Corriente DC para 3.3V Pin	50 mA
Memoria flash	256 KB de los cuales 8 KB utilizados por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52
Anchura	53.3 mm
Peso	37 g



figura 3. 1 Arduino Mega 2560

3.2.1.2 Arduino Uno

Arduino Uno, es un tablero para microcontroladores basado en el ATmega328P (véase Figura 3.2). Tiene 14 pines de entrada / salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio [13]. Las características se muestran en la Tabla 3.2.

Tabla 3. 2 Especificaciones del Arduino Uno.

Arduino Uno	
Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pernos digitales de E / S	14 (de los cuales 6 proporcionan salida PWM)
PWM Digital I / O Pins	6
Clavijas de entrada analógica	6
Corriente DC por Pin de E / S	20 mA
Corriente DC para 3.3V Pin	50 mA
Memoria flash	32 KB (ATmega328P) de los cuales 0.5 KB utilizados por el cargador de arranque
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	68.6 mm
Anchura	53.4 mm
Peso	25 g



figura 3. 2 Arduino Uno.

3.2.1.3 Arduino Nano

El Arduino Nano es un tablero pequeño, completo y fácil de utilizar basado en el ATmega328P (véase Figura 3.3). Tiene más o menos la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente [14]. Las características se muestran en la Tabla 3.3.

Tabla 3. 3 Especificaciones del Arduino Nano.

Arduino Nano	
Microcontrolador	ATmega328
Arquitectura	AVR
Tensión de funcionamiento	5 V
Memoria flash	32 KB de los cuales 2 KB utilizados por el gestor de arranque
SRAM	2 KB
Velocidad de reloj	16 MHz
Analog IN Pins	8
EEPROM	1 KB
Corriente DC por los pernos de E / S	40 mA (pernos de E / S)
Voltaje de entrada	7-12 V
Pernos digitales de E / S	22 (6 de los cuales son PWM)
Salida PWM	6
El consumo de energía	19 mA
Tamaño de PCB	18 x 45 mm
Peso	7 g
Código de producto	A000005

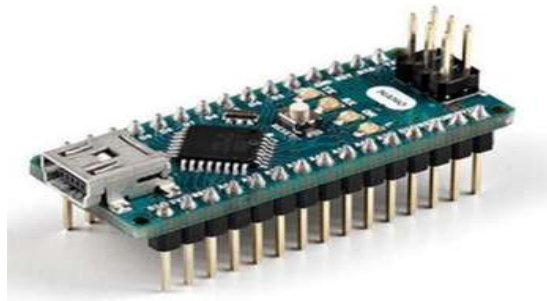


figura 3. 3 Arduino Nano

3.2.1.4 Arduino Mini

El Arduino Mini es una versión muy compacta del Arduino Nano sin una conexión USB a serie integrada (véase Figura 3.4). Es una pequeña placa para microcontroladores originalmente basada en el ATmega168, pero que ahora se suministra con la 328. Diseñada para usarse en las tablas de pruebas (Protoboards) y cuando el espacio es escaso [15]. Las características se muestran en la Tabla 3.4.

Tabla 3. 4 Especificaciones del Arduino Mini

Arduino Mini	
Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada	7-9 V
Pernos digitales de E / S	14 (de los cuales 6 proporcionan salida PWM)
Clavijas de entrada analógica	8 (de los cuales 4 están divididos en pasadores)
Corriente DC por Pin de E / S	40 mA
Memoria flash	32 KB (de los cuales 2 KB utilizados por el gestor de arranque)
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz
Longitud	30 mm
Anchura	18 mm

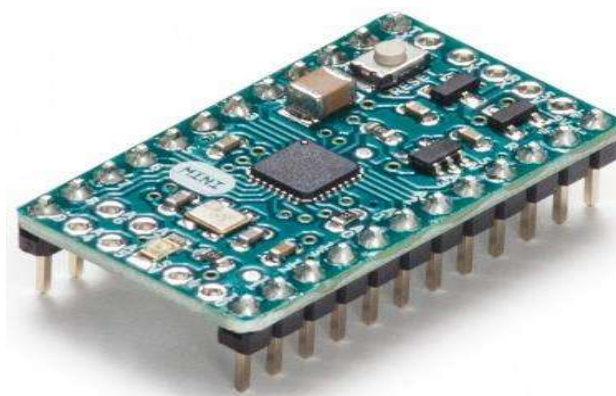


figura 3. 4 Arduino Mini

En la Tabla 3.5 se muestra una comparación de las características entre estas cuatro distintas placas de Arduino, las cuales se consideraron para la implementación de las mejoras a realizar al dispositivo.

Tabla 3. 5 Especificaciones de las cuatro placas de desarrollo Arduino

Nombre	Procesador	Voltaje de entrada	Velocidad de la CPU	Analog In / Out	IO / PWM digital	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
<u>Mega</u> <u>2560</u>	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
<u>Uno</u>	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
<u>Mini</u>	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
<u>Nano</u>	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1

3.2.2 Raspberry

La Raspberry Pi es una serie de pequeñas computadoras de una placa. Varias generaciones de Raspberry Pi han sido lanzadas. Todos los modelos cuentan con un sistema Broadcom en un chip (SoC) con una unidad de procesamiento central (CPU) compatible con ARM integrada y una unidad de procesamiento de gráficos en chip (GPU).

La velocidad del procesador varía de 700 MHz a 1.4 GHz para el Pi 3 Modelo B +; La memoria integrada oscila entre 256 MB y 1 GB de RAM.

3.2.2.1 Hardware

El hardware de Raspberry Pi ha evolucionado a través de varias versiones que presentan variaciones en la capacidad de memoria y la compatibilidad con dispositivos periféricos.

Los modelos A, A+ y Pi Zero son similares a los modelos B y B+, pero carecen de los componentes de concentrador Ethernet y USB. El adaptador de Ethernet está conectado internamente a un puerto USB adicional. En el Modelo A, A + y el Pi Zero, el puerto USB está conectado directamente al sistema en un chip (SoC). En los modelos Pi 1 modelo B+ y posteriores, el chip USB / Ethernet contiene un concentrador USB de cinco puertos, de los cuales hay cuatro puertos disponibles, mientras que el modelo Pi 1 B solo proporciona dos. En el Pi Zero, el puerto USB también está conectado directamente al SoC, pero utiliza un puerto micro USB (OTG). [16]

3.2.2.2 Procesador

La Raspberry Pi 2B utiliza un procesador ARM Cortex-A7 de cuatro bits y 900 MHz de cuatro núcleos.

El Broadcom BCM2835 SoC utilizado en la primera generación de Raspberry Pi incluye un procesador ARM11 76JZF-S de 700 MHz, unidad de procesamiento de gráficos VideoCore IV (GPU), y RAM. Tiene un caché de nivel 1 (L1) de 16 KB y un caché de nivel 2 (L2) de 128 KB. El caché de nivel 2 es utilizado principalmente por la GPU. El SoC se apila debajo del chip RAM, por lo que solo su borde es visible. El 1176JZ (F) -S es la misma CPU utilizada en el iPhone original, aunque a una velocidad de reloj más alta, y se aparearon con una GPU mucho más rápida.

El modelo V1.1 anterior de Raspberry Pi 2 usaba un Broadcom BCM2836 SoC con un procesador ARM Cortex-A7 de 900 MHz de 32 bits y cuatro núcleos, con

256 KB de caché L2 compartida. El Raspberry Pi 2 V1.2 se actualizó a un Broadcom BCM2837 SoC con un procesador ARM Cortex-A53 de 1,2 GHz de 64 bits y cuatro núcleos, el mismo SoC que se usa en el Raspberry Pi 3, pero no está bloqueado (de forma predeterminada) a la misma velocidad de reloj de la CPU de 900 MHz que la V1.1. El SoC BCM2836 ya no está en producción a finales de 2016.

La Raspberry Pi 3+ usa un SoC Broadcom BCM2837B0 con un procesador ARM Cortex-A53 de 1,4 GHz de 64 bits y cuatro núcleos, con 512 KB de caché L2 compartida.

3.2.2.3 RAM

En las placas beta más antiguas del Modelo B, 128 MB se asignaron de forma predeterminada a la GPU, dejando 128 MB para la CPU. En la primera versión de 256 MB del Modelo B (y del Modelo A), fueron posibles tres divisiones diferentes.

La división predeterminada fue 192 MB (RAM para CPU), que debería ser suficiente para la decodificación de video 1080p independiente, o para 3D simple, pero probablemente no para ambos juntos. 224 MB fue solo para Linux, con solo un framebuffer de 1080p, y es probable que falle para cualquier video o 3D. 128 MB fue para 3D pesado, posiblemente también con decodificación de video (por ejemplo, XBMC).

La Raspberry Pi 2 y la Raspberry Pi 3 tienen 1 GB de RAM. La Raspberry Pi Zero y Zero W tienen 512 MB de RAM.

3.2.2.4 Periféricos

Las placas Modelo 2B incorporan cuatro puertos USB para conectar periféricos.

La Raspberry Pi puede operarse con cualquier teclado y mouse de computadora USB genérico. También se puede usar con almacenamiento USB, convertidores de USB a MIDI y prácticamente cualquier otro dispositivo / componente con capacidades USB.

Se pueden conectar otros periféricos a través de varios pines y conectores en la superficie de la Raspberry Pi.

3.2.2.5 Video

La primera Raspberry Pi 1 Modelo A, con un puerto HDMI y un puerto de video compuesto RCA estándar para pantallas antiguas

El controlador de video puede generar resoluciones de televisión modernas estándar, como HD y Full HD, y resoluciones de monitor más altas o más bajas, así como resoluciones de TV CRT estándar NTSC o PAL más antiguas. Tal como se envió (es decir, sin overclocking personalizado), puede admitir las siguientes resoluciones: 640 × 350 EGA ; 640 × 480 VGA ; 800 × 600 SVGA ; 1024 × 768 XGA ; HDTV 1280 × 720 720p ; Variante WXGA 1280 × 768 ; 1280 × 800 variante WXGA ; 1280 × 1024 SXGA ; Variante WXGA de 1366 × 768 ; 1400 × 1050 SXGA + ; 1600 × 1200 UXGA ; 1680 × 1050 WXGA + ; 1920 × 1080 HDTV 1080p ; 1920 × 1200 WUXGA [16] .

3.2.2.6 Raspberry Pi 3 Modelo B+

La Raspberry Pi 3 Model B+ es el último producto de la gama Raspberry Pi 3, que cuenta con un procesador de cuatro núcleos de 64 bits que funciona a 1,4 GHz, doble banda de 2.4GHz y 5GHz LAN inalámbrica, Bluetooth 4.2 / BLE, Ethernet más rápida y PoE capacidad a través de un PoE HAT separado (véase Figura 3.5).

La LAN inalámbrica de banda dual viene con una certificación de cumplimiento modular, lo que permite que la placa se diseñe en productos finales con pruebas de cumplimiento de LAN inalámbrica significativamente reducidas, lo que mejora el costo y el tiempo de comercialización [17]. Las especificaciones de la Raspberry Pi 3 Model B + se muestran en la Tabla 3.6.

Tabla 3. 6 Especificaciones de la Raspberry Pi B+

Raspberry Pi 3 Model B +	
Procesador:	Broadcom BCM2837B0, CorteC-A53 SoC de 64 bits a 1,4 GHz
Memoria:	SDRAM LPDDR2 de 1 GB
Conectividad	LAN inalámbrica IEEE 802.11.b / g / n / ac de 2.4GHz y 5GHz, Bluetooth 4.2, BLE Ethernet Gigabit sobre USB 2.0 (rendimiento máximo de 300 Mbps) 4 puertos USB 2.0
Acceso:	Cabecera GPIO de 40 pines extendida
Sonido del video:	1 x tamaño completo HDMI Puerto de visualización DSI MIPI Puerto de cámara MIPI CSI Salida estéreo de 4 polos y puerto de video compuesto
Multimedia:	H.264, decodificación MPEG-4 (1080p30); Codificación H.264 (1080p30); OpenGL ES 1.1, gráficos 2.0
Soporte de tarjeta SD:	Formato micro SD para cargar el sistema operativo y el almacenamiento de datos.
Potencia de entrada:	5V / 2.5A DC a través de conector micro USB 5V DC vía cabecera GPIO Power over Ethernet (PoE): habilitado (requiere PoE HAT por separado)
Ambiente:	Temperatura de funcionamiento, 0–50 ° C.



figura 3. 5 Raspberry Pi B+

3.2.2.7 Raspberry Pi 3 Modelo A+

La Raspberry Pi 3 Modelo A+ (Figura 3.6) es el último producto de la gama Raspberry Pi 3. Al igual que el Raspberry Pi 3 Modelo B+, cuenta con un procesador de cuatro núcleos de 64 bits que funciona a 1.4 GHz, banda inalámbrica de banda dual 2.4 GHz y 5 GHz y Bluetooth 4.2 / BLE.

La LAN inalámbrica de banda dual viene con una certificación de cumplimiento modular, lo que permite que la placa se diseñe en productos finales con una reducción significativa de las pruebas de cumplimiento de LAN inalámbrica, mejorando tanto el costo como el tiempo de comercialización [18]. Las especificaciones de la Raspberry Pi 3 Modelo A+ se muestran en la Tabla 3.7.

Tabla 3. 7 Especificaciones de la Raspberry modelo A+

Raspberry Pi 3 Modelo A +	
Procesador:	Broadcom BCM2837B0, CorteC-A53 SoC de 64 bits a 1,4 GHz
Memoria:	512MB LPDDR2 SDRAM
Conectividad	LAN inalámbrica de 2,4 GHz y 5 GHz IEE 802.11.b / g / n / ac, Bluetooth 4.2 / BLE
Acceso:	Cabecera GPIO de 40 pines extendida
Sonido del video:	1 x tamaño completo HDMI Puerto de visualización DSI MIPI Puerto de cámara MIPI CSI Salida estéreo de 4 polos y puerto de video compuesto
Multimedia:	H.264, decodificación MPEG-4 (1080p30); Codificación H.264 (1080p30); OpenGL ES 1.1, gráficos 2.0
Soporte de tarjeta SD:	Formato micro SD para cargar el sistema operativo y el almacenamiento de datos
Potencia de entrada:	5 V / 2.5 A DC a través de conector micro USB 5 V DC a través de cabecera GPIO
Ambiente:	Ambiente: Temperatura de funcionamiento, 0–50 ° C

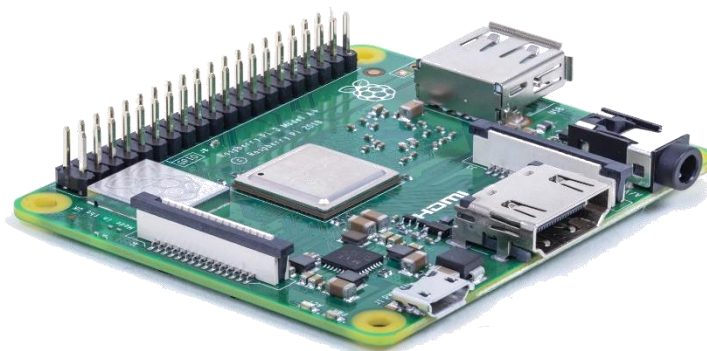


figura 3. 6 Raspberry Pi modelo A+

3.2.2.8 La Raspberry Pi Zero

La Raspberry Pi Zero W (Figura 3.7) extiende la familia Pi Zero. El Pi Zero W tiene toda la funcionalidad del Pi Zero (Figura 3.8) original [19]. Las especificaciones de ambas Raspberry se muestran en la Tabla 3.8.

Tabla 3. 8 Especificaciones de las Raspberry Pi Zero y Zero W

Características	Raspberry Pi Zero W	Raspberry Pi Zero
LAN inalámbrica 802.11 b / g / n	Si	No
Bluetooth 4.1	Si	No
Bluetooth de baja energía (BLE)	Si	No
1 GHz, CPU de un solo núcleo	Si	Si
512 MB de RAM	Si	Si
Mini puertos HDMI y USB On-The-Go	Si	Si
Alimentación micro USB	Si	Si
Cabezal de 40 pines compatible con HAT	Si	Si

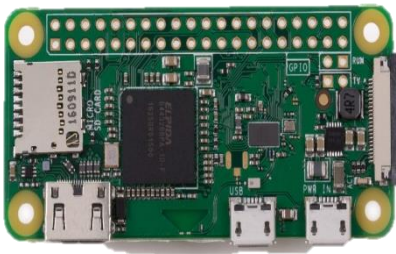


figura 3. 7 Raspberry Pi Zero W



figura 3. 8 Raspberry Pi Zero

3.2.3 Selección

Para hacer la selección de la placa de desarrollo en la cual se realizará tanto la implementación del hardware como la creación del software, se contempla que esta sea capaz de funcionar con gran cantidad de hardware, así también como que su lenguaje de programación se sencillo y fácil de utilizar. Además, se requiere que esta placa de desarrollo sea del menor tamaño posible debido a que se encontrara dentro de una plancha de ropa. Otra característica que se contempla es que este sea de fácil acceso al público, así como de un costo accesible. Otra característica a tener en cuenta es que esta placa de desarrollo sea de una respuesta rápida. Pues

se requiere que los procesos que se implementaran en esta se hagan de la manera más rápida posible.

Teniendo en cuenta todas estas características necesarias para desarrollar el sistema, además de la información encontrada en internet y las posibilidades con las que se cuenta actualmente como son: la ubicación y el acceso a cada una de estas placas de desarrollo, se ha optado por hacer la implementación en un sistema de placa Arduino.

En específico, se desarrolla el sistema en una placa de desarrollo Arduino Nano, la cual es un módulo mucho más pequeño que un Arduino Mega o un Arduino Uno. Esta selección se hace de acuerdo las características necesarias para el sistema como, la de tener un tamaño reducido y un costo relativamente bajo, además de que su entorno de programación es muy simple y claro.

3.3 Sensor de posición

El sensor de posición está pensado para poder detectar en cualquier momento la posición del sistema. Esto, como se mencionó en capítulos anteriores, para poder detectar si el sistema está de suela, pie o se encuentra volteado, lo cual para la primera opción podría tener un alto riesgo si se encuentra sin supervisión de una persona.

Existen diversos tipos de sensores de posición, por lo que también se tomó en cuenta cuál de ellos era el mejor para el sistema.

3.3.1 sensor de inclinación - AT407

Este interruptor de inclinación básico AT407 (Figura 3.9) se puede usar fácilmente para detectar la orientación. Dentro de su encapsulado hay una bola que hace contacto con los pasadores cuando la caja está en posición vertical. Inclinando el sensor, las bolas no se tocan. Por otro lado, la Figura 3.10 muestra las diferentes posiciones que puede tener el sensor [20] [21]. Las especificaciones del sensor de inclinación se muestran en la Tabla 3.9.

Tabla 3. 9 Especificaciones del sensor AT407

AT407	
Clasificación eléctrica	<6mA 24VDC
Vida eléctrica	> 50,000 ciclos
Resistencia de contacto	1Ω
Temperatura de soldadura:	250 °C 3 segundos
Temperatura ambiente:	0 °C ~ 100 °C
Recipiente:	Cobre chapado en oro.
Electrodo:	Cobre chapado en oro.
Rolling Ball	Acero inoxidable chapado en oro
Pieza aislada:	Nylon 46 (94V-0)



figura 3. 10 Sensor de inclinación AT407

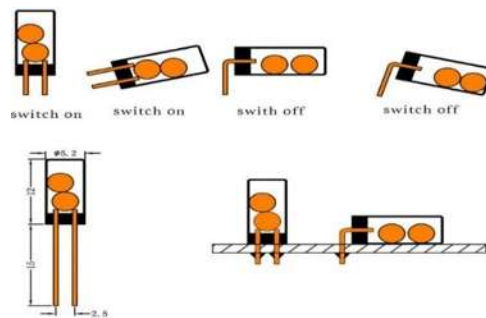


figura 3. 9 Forma de trabajo del sensor de inclinación AT407

3.3.2 KEYES KY-017

El Keyes Mercury Switch Module es un módulo de detección de movimiento simple y efectivo (véase Figura 3.11). Cuando algún vehículo o robot autónomo está inclinado, la bola de mercurio dentro del tubo se moverá para completar un circuito eléctrico que, a su vez, enviará una señal al sistema de control Arduino [22]. Las especificaciones del sensor de inclinación se muestran en la Tabla 3.10.

Tabla 3. 10 Especificaciones del módulo KEYES KY-017

Especificaciones	
Material	FR4
Longitud:	30 mm
Ancho	15 mm
Altura	10 mm
Peso	1.50 g
Voltaje de funcionamiento	5 vcc

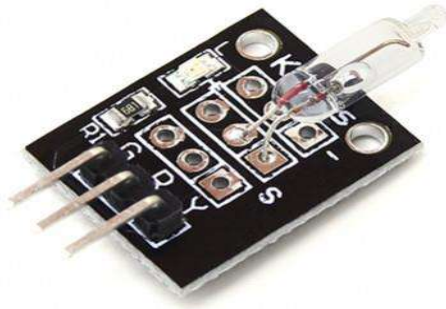


figura 3. 11 Modulo KEYES KY-017

3.3.3 MPU-9250

Los sensores MPU-9150 y MPU-9250 (véase Figura 3.12) son IMU de 9DOF, una nueva generación fabricados por Invensense. Ambos dispositivos forman parte de la misma familia de componentes, siendo el MPU-9250 el modelo actual y recomendado por el fabricante.

Internamente, el MPU-9250 incorpora en un mismo integrado una IMU MPU-6500, que a su vez está formado por: un acelerómetro de 3DOF, un giroscopio de 3DOF, y un magnetómetro AK8963 de 3DOF.

Por su parte, el MPU-9150 está formado por una IMU MPU-6050 y un magnetómetro AK8975. La comunicación en ambos modelos puede realizarse tanto por bus SPI como por bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de bajo voltaje entre 2.4 a 3.6V.

Dispone de conversores analógicos digitales (ADC) de 16bits. El rango del acelerómetro puede ser ajustado a $\pm 2g$, $\pm 4g$, $\pm 8g$, y $\pm 16g$, el del giroscopio a ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$, y el del magnetómetro hasta $\pm 4800\mu\text{T}$ [23] [24]. Las especificaciones del sensor de inclinación se muestran en la Tabla 3.11.

Tabla 3. 11 Especificaciones del sensor MPU-9250

Características del giroscopio
Sensores de frecuencia angular (giroscopios) de eje X, Y y Z de salida digital con un rango de escala completa programable por el usuario de ± 250 , ± 500 , ± 1000 y ± 2000 ° / seg y ADC de 16 bits integrados
Filtro de paso bajo programable digitalmente
Corriente de funcionamiento del giroscopio: 3.2mA
Modo de reposo actual: 8 μ A
Factor de escala de sensibilidad calibrado en fábrica
Auto test

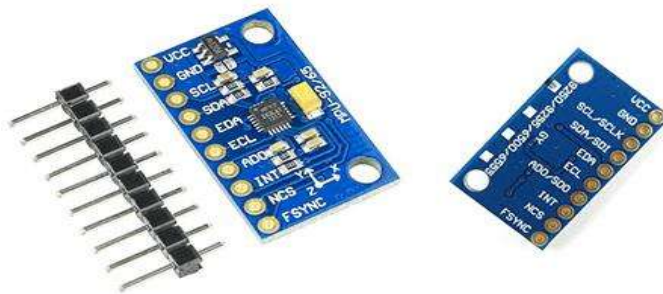


figura 3. 12 Sensor MPU-9250

3.4 Sistema de levantamiento

Para tener un elemento más de seguridad se plantea implementar un mecanismo que sea capaz de levantar el sistema de tal manera de que éste, cuando se encuentre sin supervisión de una persona, pueda levantarse por sí solo, logrando que la suela caliente se despegue de la superficie y de esta manera evitar un accidente grave como podría ser la quemadura de la tela o comenzar un incendio en el hogar. Para este caso en especial se toma en cuenta la utilización de servomotores, los cuales cuentan con un sistema de engranaje haciendo que estos puedan ejercer mayor potencia con un mínimo de esfuerzo además de que su velocidad de movimiento es controlada.

Un servomotor es un motor eléctrico al que se le puede controlar tanto la velocidad, como la posición del eje que gira (también llamada dirección del eje o giro del rotor). Los servomotores no giran su eje 360° (aunque ahora hay algunos que si lo permiten), como los motores normales, solo giran 180° hacia la izquierda

o hacia la derecha (ida y retorno). En la Figura 3.13 se puede observar uno de estos servomotores.

Un servomotor es un sistema compuesto por los siguientes componentes:

Un motor eléctrico: es el encargado de generar el movimiento, a través de su eje.

Un sistema de regulación: formado por engranajes, que actúan sobre el motor para regular su velocidad y el par. Mediante estos engranajes, normalmente ruedas dentadas, podemos aumentar la velocidad y el par o disminuirlas.

Un sistema de control o sensor: circuito electrónico que controla el movimiento del motor mediante el envío de pulsos eléctricos.

Un potenciómetro: conectado al eje central del motor que permite saber en todo momento el ángulo en el que se encuentra el eje del motor [25] [26]. En la Tabla 3.12 se muestran las especificaciones de este servomotor.

Tabla 3. 12 Especificaciones del Servomotor MG995.

Modelo MG995	
Tensión de funcionamiento	4.8 - 7.2 V
Velocidad de operación a 4.8V	0.20seg / 60 °
Velocidad de funcionamiento a 6,6 V	0.16 seg / 60 °
Par Torque a 4.8V	10 kg-cm
Par de torsión a 6,6 V	12 kg-cm
Temperatura de funcionamiento (° C)	-30 a 60
Ancho de banda muerta	1 µs
Tipo de engranaje	Metal
Grado de Rotación	180
Longitud (mm)	40.5
Ancho (mm)	20
Altura (mm)	44
Descripción del cable	
Rojo	Positivo
Marrón	Negativo
Naranja	Señal



figura 3. 13 Servomotor

3.5 Sistema táctil

Este sistema está pensado al igual que los anteriores para evitar accidentes donde la ropa pueda resultar dañada o en casos más graves donde está a causa de un calor excesivo pueda provocar el inicio de un incendio en el hogar. Este sensor táctil estará siempre censando que algo en él lo esté tocando, de lo contrario se activaran el sistema de levantamiento el cual no dejara que la suela del sistema pegue en la superficie, logrando así que el calor no siga llegando a la tela una vez que este ya ha sido dejado de tocar.

Además de este sensor existirán 2 botones táctiles más, los cuales serán descritos en el siguiente apartado (Sistema de bombeo).

Los sensores capacitivos son un tipo de sensor eléctrico. Los sensores capacitivos (KAS) reaccionan ante metales, y no metales que al aproximarse a la superficie activa sobrepasan una determinada capacidad. La distancia de conexión respecto a un determinado material es tanto mayor cuanto más elevada sea su constante dieléctrica.

Es un dispositivo que presenta un comportamiento similar a un pulsador, pero puede ser activado con poca o ninguna presión. Este tipo de sensor táctil basa su funcionamiento en la medición de la variación de la capacitancia. La placa sensora y el cuerpo humano actúan como condensador y, por tanto, forman un sistema que almacena una carga eléctrica. Al reducir la distancia la capacitancia aumenta y el sistema almacena una carga superior. Esta acumulación de carga puede ser detectada en la placa sensora y generar una señal digital cuando supere un cierto valor. En la Figura 3.14 se puede observar un bosquejo del funcionamiento de este

tipo de sensor. Esta señal de disparo puede a ser vez ser capturada con una entrada digital de Arduino [27] [28].

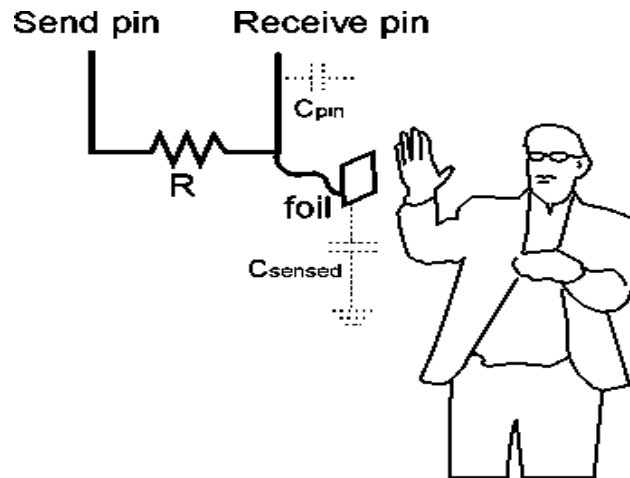


figura 3. 14 Bosquejo del funcionamiento del sensor táctil.

3.6 Sistema de bombeo

Con este sistema de bombeo se pretende hacer que el agua pueda ir desde el contenedor de agua hasta la suela, generando de esta manera el vapor que ayudara a que las telas sean más fáciles de desarrugar, además también se contara con un botón de roció el cual servirá para como su nombre lo indica rociar agua sobre la ropa y hacer más fácil la eliminación de arrugas.

Con la implementación de este sistema se logrará eliminar por completo los botones mecánicos sustituyéndolos por un botón táctil. Se contará con dos botones, uno como se mencionó anteriormente, para generar el vapor que saldrá por la suela, y otro para activar el rociador que se encontrara por la parte de enfrente.

El fin de este sistema es pasar lo más posible los sistemas mecánicos que se usan habitualmente a sistemas electrónicos.

3.6.1 Bomba

Debido a que se pretende que el sistema sea lo más pequeño posible logrando de esta manera una buena imagen comercial, además de tener el menor peso posible se considera la utilización de una bomba de agua sumergible debido a que esta estará en contacto directo en todo momento con el agua [29].

En la Tabla 3.13 se muestran las especificaciones de la bomba de agua sumergible que se usara en el sistema. A demás en la Figura 3.15 se muestra una ilustración de esta misma.

Tabla 3. 13 Especificaciones de la bomba de agua sumergible.

Mini Bomba de Agua Sumergible 240 L/H
Rotor magnético permanente, sin escobillas.
Adopta el eje de cerámica del alto rendimiento.
El eje está encerrado con sellado estático, no dinámico, lo que puede evitar problemas de fugas.
Diseño anfibia.
Instalación sumergible y totalmente impermeable.
Bajo consumo.
Bajo ruido (menos de 35db).
Material de la bomba: ABS.
Fluidos: Agua, aceite, gasolina, ácido y solución alcalina.
Temperatura de trabajo: 0 ~ 75 °C.
Consumo de energía: 4.2W.
Voltaje clasificado: 12V DC.
Corriente nominal máxima: 350mA.
Caudal máximo: 4L / MIN (1.06G / MIN).



figura 3. 15 Bomba de agua sumergible.

3.7 Detección del nivel de liquido

El sistema de nivel de líquido que se implementa está diseñado de tal manera que se pueda apreciar mediante una señal visual (en este caso un led) que el agua en el depósito está a punto de acabarse, y de esta manera poder realizar nuevamente un llenado del tanque de agua sin tener sorpresas de que esta se acabe justo cuando más se necesite.

El sensor de agua está pensado básicamente debido a que existen planchas de ropa que no tienen alguna forma de ver el nivel de agua actual con el que cuenta, puesto que es necesario que el sistema de roció de agua como la generación de vapor dejen de funcionar para saber que esta se ha acabado.

De esta manera se podrá saber cuándo hay que volver a llenar el contenedor de agua antes de que esta se agote. Con este sensor se podrá conocer si el nivel de agua a pasado cierto nivel, además el sensor posee aislamiento eléctrico a prueba de agua.

La implementación se logra colocando un primer sensor en la parte superior del contenedor de agua para detectar que el agua a llegado a ese nivel una vez que el sensor se haya cerrado. Un segundo sensor estará colocado justo antes de que el agua se termine de modo de que este al desactivarse emita una alerta visual que indicara que el nivel de agua en el contenedor es demasiado bajo y está cerca de terminarse. En la Tabla 3.14 se muestran las especificaciones de el sensor de nivel de líquido y la Figura 3.16 muestra una imagen de este.

Tabla 3. 14 Especificaciones del sensor de nivel de líquido On/Off

Sensor de Nivel de líquido On/Off	
Voltaje Máximo	100V DC/AC
Corriente Máxima	0.5A
Carga Máxima	10W
Cable	30 cm



figura 3. 16 Sensor de nivel de líquido On / Off.

3.8 Cámara

El sistema de la cámara está diseñado para poder tomar fotografías de la tela que se está planchando, en ese momento para posteriormente ser analizada y detectar de qué tipo de tela se trata. Todos esto, para poder determinar cuál es la temperatura adecuada para que la tela sea planchada.

Existen gran diversidad de cámaras en el mercado por lo que para la selección de esta se requiere que sea de un tamaño pequeño y de una resolución relativamente buena, pues se necesita que la imagen sea lo más clara posible para poder obtener a partir de ella buenos parámetros que son los que nos servirán para conocer el tipo de tela que se está planchando.

Algunas de las cámaras más asequibles para Arduino son:

3.8.1 OV7670

LA OV7670 es un sensor de imagen CMOS de bajo voltaje que proporciona la funcionalidad completa de una cámara VGA de un solo chip y un procesador de imágenes en un paquete de espacio reducido (Figura 3.17). El OV7670 / OV7171 proporciona imágenes de 8 bits de cuadro completo, submuestreadas o en ventanas en una amplia gama de formatos, controlados a través de la interfaz del Bus de control de cámara en serie (SCCB) [30]. Las especificaciones de esta cámara se muestran en la Tabla 3.15.

Tabla 3. 15 Especificaciones de la Cámara OV7670

Especificaciones
Tamaño óptico 1/6 pulgada
Resolución 640 x 480 VGA
Regulador a bordo, solo se necesita un suministro de 3.3 V
Montado con lentes de alta calidad F1.8 / 6mm
Alta sensibilidad para el funcionamiento con poca luz
Baja tensión de funcionamiento para aplicaciones portátiles integradas
Interfaz SCCB estándar compatible con la interfaz I2C
Admite tamaños de imagen: VGA, CIF y cualquier tamaño que se reduzca de CIF a 40 x 30
Funciones de control de imagen automáticas que incluyen:
Control de exposición (AEC), Control automático de ganancia (AGC), Balance de blancos automático (AWB), Automático
Filtro de banda (ABF) y calibración automática de nivel de negro (ABLCL)
Controles de calidad de imagen que incluyen saturación de color, matiz, gamma, nitidez (realce de bordes) y anti-blooming
ISP incluye reducción de ruido y corrección de defectos.
Soporta modo estroboscópico LED y flash



figura 3. 17 Cámara OV7670

3.8.2 LS-Y201

LS-Y201 es el módulo de cámara de puerto serie de nueva generación de LinkSprite (Véase Figura 3.18). Puede capturar imágenes de alta resolución utilizando el puerto serie. LS-Y201 es un diseño modular que genera imágenes JPEG a través de UART y se puede integrar fácilmente en el diseño existente [31]. Las especificaciones de esta cámara se muestran en la Tabla 3.16.

Tabla 3. 16 Especificaciones de la cámara LS-Y201.

Especificación
Resolución VGA / QVGA / 160 * 120.
Soporte de captura de JPEG desde el puerto serie.
La velocidad de transmisión predeterminada del puerto serie es 38400.
Fuente de alimentación DC 3.3V o 5V.
Tamaño 32 mm x 32 mm.
Consumo de corriente: 80-100mA.
APLICACIONES
Diferentes sistemas de captura de imágenes.
Monitoreo ambiental
Monitoreo de la industria
Equipo medico
Vídeo llamada
Seguridad
GPS basado en vehículo



figura 3. 18 Cámara LS-Y201

3.8.3 Mini cámara espía con gatillo para foto o video

Esta mini cámara espía es más pequeña que una cámara miniatura, con una resolución lo suficientemente alta. Es un módulo de resolución (video de 480p y foto de 1280x720), con una placa de controlador de aproximadamente 1 pulgada cuadrada, con un soporte de tarjeta microSD. La mini cámara se muestra en la Figura 3.19. El módulo de imagen tiene aproximadamente el tamaño de la cámara de un teléfono celular (el cuerpo mide solo 6,2 x 6,2 mm) y tiene una parte posterior adherida, por lo que es fácil de montar en el timbre de la puerta o detrás del ojo de un oso de peluche. Hay un LED de alimentación y un LED de 'actividad' que le permite saber qué está haciendo. En términos de imágenes fijas, la cámara es capaz de imágenes estáticas de 1280x720 píxeles y video de 640x480 [32]. Las especificaciones de esta mini cámara se muestran en la Tabla 3.17.

Tabla 3. 17 Especificaciones de la mini cámara espía con gatillo para foto o video.

Mini cámara espía	
Formato de foto	JPEG
Resolución de la foto	1280 × 720
Formato de video	AVI
Resolución de video	640x480
Soporte máximo MicroSD	32G
Voltaje de alimentación	3.7V ~ 5V, se puede conectar directamente a una batería de litio
Corriente en espera	80mA
Corriente de funcionamiento	110mA



figura 3. 19 Mini cámara espía con gatillo para foto o video.

3.9 Conclusiones

Debido a la gran cantidad de sensores y sistemas desarrollados en la actualidad es posible hacer este tipo de avances casi con cualquier dispositivo.

En este capítulo se puede apreciar la gran cantidad de sensores y módulos existentes en el mercado que son compatibles con la placa de desarrollo de Arduino. Es debido a esto, que en la actualidad se puede modificar casi cualquier aparato (si no es que todos) en uno mucho mejor y con características que pudieran ser casi personalizadas por cualquiera.

Es importante seleccionar bien los componentes que se utilizaran en el sistema, pues de ellos dependerá que éste sea lo suficientemente bueno como para salir en algún momento a la venta. Además, de que deberá tener una buena estética en la que los módulos y sensores queden ser ordenados de la mejor manera posible. Si bien, algunos de estos módulos y sensores son demasiados grandes y con características poco satisfactorias, no se puede dudar que en un futuro estas serán de un tamaño mucho menor y con características que superen por mucho a las actuales.

Algunas de las etapas en las que se utilizan estos módulos y sensores no han sido elaboradas por lo que se mencionara de ello en el apartado de proyectos futuros. No obstante, se han hecho pruebas que dan certeza de que estos pueden funcionar de manera adecuada una vez implementadas con todo lo necesario para funcionar.

Capítulo 4

4. Software Desarrollado

4.1 Introducción

En el presente capítulo se presentan los códigos desarrollados para cada uno de los sensores utilizados en el sistema.

Debido a que se ha optado por la utilización de una de las placas de desarrollo de Arduino, la mayor parte de la programación se desarrolla en su propio entorno de programación (el IDE de Arduino).

Algunos de los códigos utilizados para el correcto funcionamiento de módulos y sensores, ya han sido implementados por varias personas y son de uso público pudiendo hacer uso de ellos, una de las páginas más populares es GitHub [41], la cual es una plataforma de desarrollo inspirada en tu forma de trabajar. Desde el código abierto hasta el negocio, puede alojar y revisar códigos, administrar proyectos y crear software.

4.2 Algoritmo de posición

Este algoritmo de posición como se menciona en el capítulo anterior, sirve para poder detectar en qué posición se encuentra el sistema, y de esta manera poder actuar de una o de otra forma según convenga a la posición en la que se encuentra.

El Código 4.1, mostrado a continuación, es una parte del código total que se ha extraído de GitHub, y modificado para conveniencia del sistema propuesto en esta tesis, cumpliendo con las características deseadas.

Como primera prueba para comprobar el funcionamiento tanto del código, como del dispositivo MPU, se implementa el Código 4.1. Obteniendo resultados satisfactorios [33].

```
if (roll > -30 && roll < 30 && pitch > -30 && pitch < 30)
{
  digitalWrite (LED2, HIGH);
  delay (100);
}
else {
  digitalWrite (LED2, LOW);
  delay (100);
}
```

Código 4. 1 Algoritmo de posición utilizado para el sistema.

4.3 Algoritmo del sistema táctil

Para el desarrollo de este código se hace uso de las librerías de Arduino, las cuales son de mucha ayuda a la hora de escribir un código, pues están ayudando a que los códigos generados sean de un tamaño mucho más reducido y entendible para la mayoría del público.

Este primer código (Código 4.2) es una prueba para ver el funcionamiento del sensor táctil por sí solo, de esta manera se puede tener un entendimiento mucho mejor de cómo funciona dicho sensor, al igual que las librerías utilizadas en ella.

En el código se pone un valor fijo el cual será el que se tendrá que superar a la hora de que el sensor táctil sea tocado. Posteriormente está la declaración del número del pin que se usará y su modo de operación, en este caso será una salida. Posteriormente se pasa a la función “void loop” la cual tiene como característica principal que siempre se estará ejecutando. Dentro de esta función se encuentra otra función que leerá el valor actual del sensor táctil para almacenarlo en una variable, que después será comparada con el límite y decir si se prende o no el led designado [34].


```

#include <CapacitiveSensor.h>
CapacitiveSensor capSensor = CapacitiveSensor (4, 2);

int limite = 500;
const int ledPin = 6;

void setup ()
{
  Serial.begin(9600);
  pinMode (ledPin, OUTPUT);
}

void loop ()
{
  long sensorValue = capSensor.capacitiveSensor(30);
  Serial.println(sensorValue);

  if (sensorValue > limite)
  {
    digitalWrite (ledPin, HIGH);
  }
  else
  {
    digitalWrite (ledPin, LOW);
  }
  //delay (100);
}

```

Código 4. 2 Algoritmo del sensor táctil.

4.4 Algoritmo del sistema de levantamiento

Para este caso al igual que los demás se realiza una primera prueba con el servo motor, esto también para familiarizarse un poco con el sistema de funcionamiento de éste, y poder hacer posteriormente la modificación en el código en el que ya se ha implementado el sensor de posición y el sensor táctil.

En el siguiente código (Código 4.3) se muestra el funcionamiento del servo motor: en el cual primero se crea el objeto “myservo” y posteriormente en la función “void loop” se hacen algunos retrasos de tiempos para que el servomotor tenga tiempo de poder posicionarse en la posición que le corresponde según se le ha indicado. Debido a que los servomotores no están exactamente bien calibrados es necesario realizar pruebas de posiciones logrando de esta manera encontrar la más adecuada para los requerimientos que se ocupen [35].

```

#include <Servo.h>
Servo myservo;

void setup (){
  Servo myservo;
}

void loop () {
  delay (100);
  myservo.attach (2); //activamos el servo con este pin (2)
  myservo.write (0); //posición de 0 grados
  delay (500);
  myservo.detach (); //desactivamos el servo

  delay (100);
  myservo.attach (2); //activamos el servo con este pin (2)
  myservo.write (120); //posición de 80 grados (90 grados)
  delay (500);
  myservo.detach (); //desactivamos el servo
  delay(100);
}

```

Código 4. 3 Algoritmo del sensor de levantamiento.

De esta manera se contará ya con un sistema de seguridad relativamente satisfactorio a comparación de muchas planchas de ropa actuales en el mercado.

4.5 Algoritmo para el sistema de bombeo

El sistema de bombeo está pensado para realizar dos funciones específicas. La primera es la de generar vapor saliente por la suela, la cual como se menciona el capítulo anterior servirá, para la eliminación de las arrugas de la ropa de una manera más fácil.

La segunda función es la de realizar un disparo de roció, el cual saldrá por la parte frontal de la plancha de ropa. De igual manera como se menciona en el capítulo anterior, servirá para humedecer la ropa logrando que las arrugas sean más fáciles de eliminar.

Generalmente el uso de un sistema de bombeo implementado en Arduino es muy fácil pues constaría de un circuito en serie que se serraría cada vez que se requiera de un flujo de agua.

Como siempre la elaboración de un código en el cual se pueda familiarizarse con el funcionamiento de este sistema nunca está de más.

En el Código 4.4 se puede apreciar cómo se enciende y se apaga la bomba durante intervalos de un segundo [36]:

```
const int pin = 9;

void setup () {
  Serial.begin(9600);    //iniciar puerto serie
  pinMode (pin, OUTPUT); //definir pin como salida
}

void loop (){
  digitalWrite (pin, HIGH);    // Bomba Prendida
  delay (1000);                // esperar un segundo
  digitalWrite (pin, LOW);    // Bomba Apagada
  delay (1000);                // esperar un segundo
}
```

Código 4. 4 Algoritmo del sensor de bombeo.

De esta manera ya se cuenta tanto como con un sensor de posición, un sensor de táctil, un sistema de levantamiento, y un sistema que bombeo incorporados dentro del sistema que se está desarrollando.

En el Código 4.5 se muestra una de las partes más importantes del código que se empleara en el sistema. En este caso servirá para mantener prendido o apagado el sistema en posición vertical. De esta manera nuestro sistema solo se encontrara encendido en posición vertical siempre y cuando esté siendo tocado, de otra manera procederá a apagarse.

```
if (roll > -30 && roll < 30 && pitch > -30 && pitch < 30 && sensorValue3 > limite) // VERTICAL
{
  digitalWrite(LED1, HIGH);
}
else {
  digitalWrite(LED1 , LOW);
}
```

Código 4. 5 Algoritmo del funcionamiento vertical.

En el Código 4.6 se muestra otra de las partes más importantes del código que se empleara para el sistema. En este caso este código servirá para mantener prendido o apagado el sistema en posición horizontal siempre que se encuentre siendo tocado o no, teniendo de esta manera un sistema de seguridad una vez que el sistema se encuentre sin supervisión humana. Además, en esta sección se agregan los demás códigos mostrando anteriormente, estos códigos han sido escritos de tal manera que el sistema funcione correctamente.

```
if (roll > 60 && roll < 120 && pitch > -50 && pitch < 50 && sensorValue3 > limite) //HORIZONTAL
{

  if (sensorValue1 > limite) // BOMBA 1
  {
    digitalWrite(BOMBA1, HIGH);
  }
  else {
    digitalWrite(BOMBA1, LOW);
  }

  if (sensorValue2 > limite) // BOMBA 2
  {
    digitalWrite(BOMBA2, HIGH);
  }
  else {
    digitalWrite(BOMBA2, LOW);
  }

  digitalWrite(LED2, HIGH);
  delay(100); // levantamiento OFF
  myservo.attach (2); //activamos el servo con este pin (2)
  myservo.write (0); //posicion de 0 grados
  delay(500);
  myservo.detach (); //desactivamos el servo
}
else {
  digitalWrite(LED2, LOW);
  delay(100); // levantamiento ON
  myservo.attach (2); //activamos el servo con este pin (2)
  myservo.write (80); //posicion de 80 grados (90 grados)
  delay(500);
  myservo.detach (); //desactivamos el servo
  delay(100);
}
```

Código 4. 6 Algoritmo del funcionamiento horizontal

4.6 Algoritmo de reconocimiento textil

Para realizar el reconocimiento del tipo de tela de manera automática, con el objetivo de poder ajustar la temperatura, se propone la utilización de una red neuronal para hacer esta tarea. Este reconocimiento estará siendo utilizado en conjunto con la cámara seleccionada anteriormente, la cual será la encargada de estar tomando fotografías de la tela a planchar. Esta última acción la tendrá que realizar la plancha cada vez que esta se encuentra posicionada de manera horizontal y presenta un movimiento.

Este código es generado con ayuda de la herramienta computacional Matlab [42], la cual contienen un toolbox donde se puede encontrar esta opción. De tal manera que mediante una serie de datos de entrada y salida puede ser entrenada (la red neuronal) para que después pueda por si sola reconocer datos distintos nuevos y clasificarlos de manera adecuada según sea su clase.

Para poder ver más claro cómo funciona esta herramienta, Matlab toma en cuenta datos reales, de los cuales se presenta a continuación un ejemplo.

El conjunto de datos de la flor de iris es un conjunto de datos multivariante introducido por Ronald Fisher en su artículo de 1936. Estos datos son los que se usaran para la creación de un ejemplo de una red neuronal creada con la ayuda del toolbox de Matlab, ya que estos datos pueden ser cargados automáticamente desde la misma aplicación para crear la red neuronal.

Para la creación de esta red neuronal se utiliza la app de Matlab llamada “Neural Net Pattern Recognition”, desde la cual se pueden cargar los datos que se requieren como se muestra en la Figura 4.1.

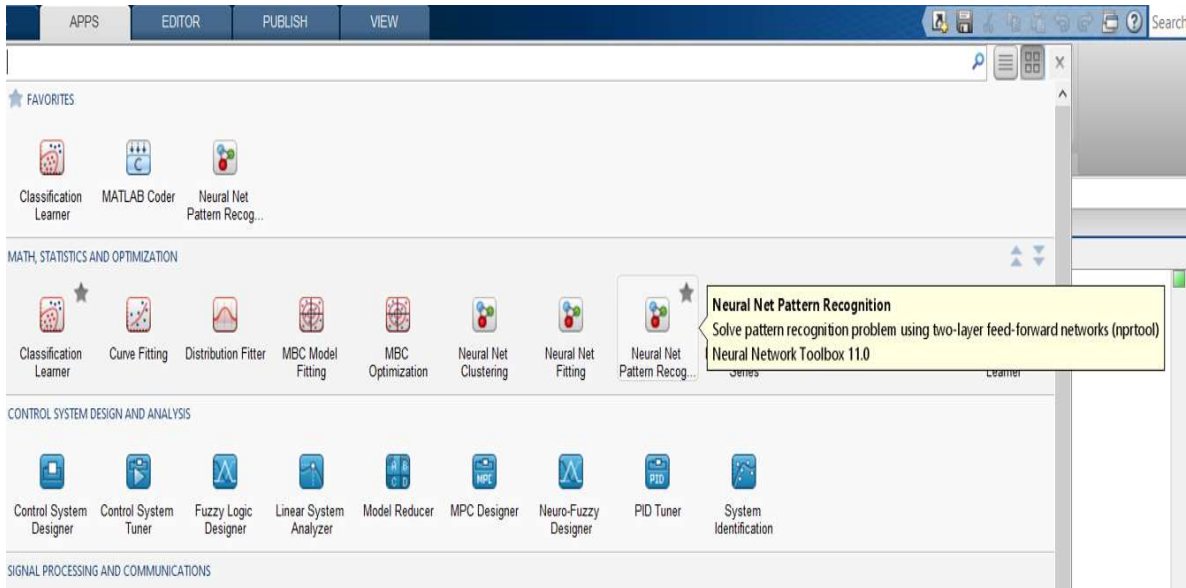


Figura 4. 1 Creación de una RNA

Después de abrir la app, pueden cargarse los datos de las flores iris desde esa misma ventana. Este paso se muestra en la Figura 4.2.

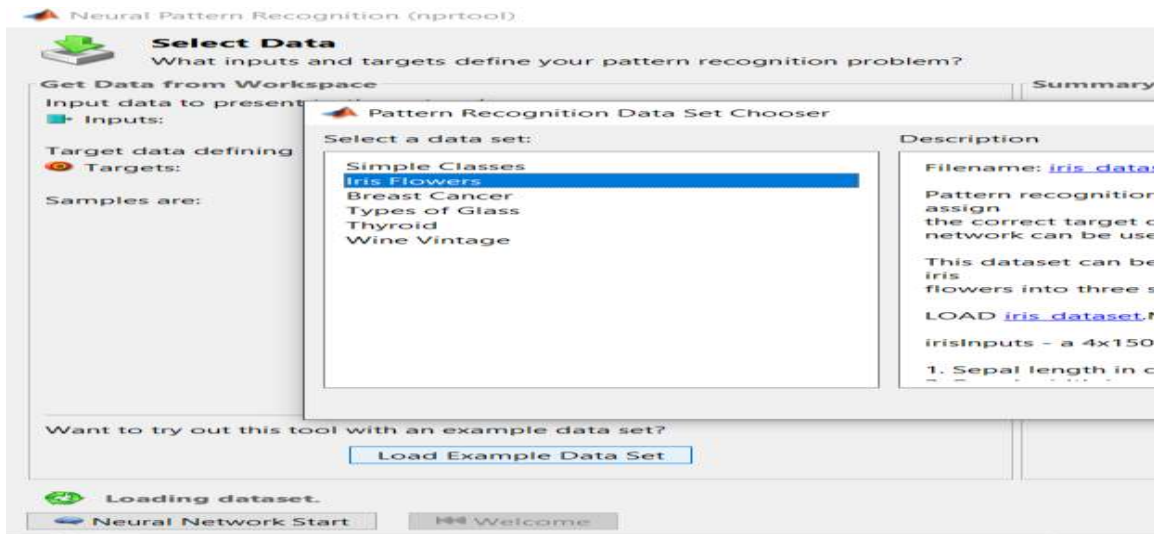


Figura 4. 2 Importación de los datos de las flores de iris.

Una vez que se han importado los datos se continua con la validación y datos de prueba. Estos datos pueden cambiarse si así se desea. Este proceso se muestra en la Figura 4.3.



Figura 4.3 Valores de entrenamiento de la RNA

Terminado este proceso se eligen la cantidad de neuronas que tendrá la red. Este proceso se muestra en la Figura 4.4.

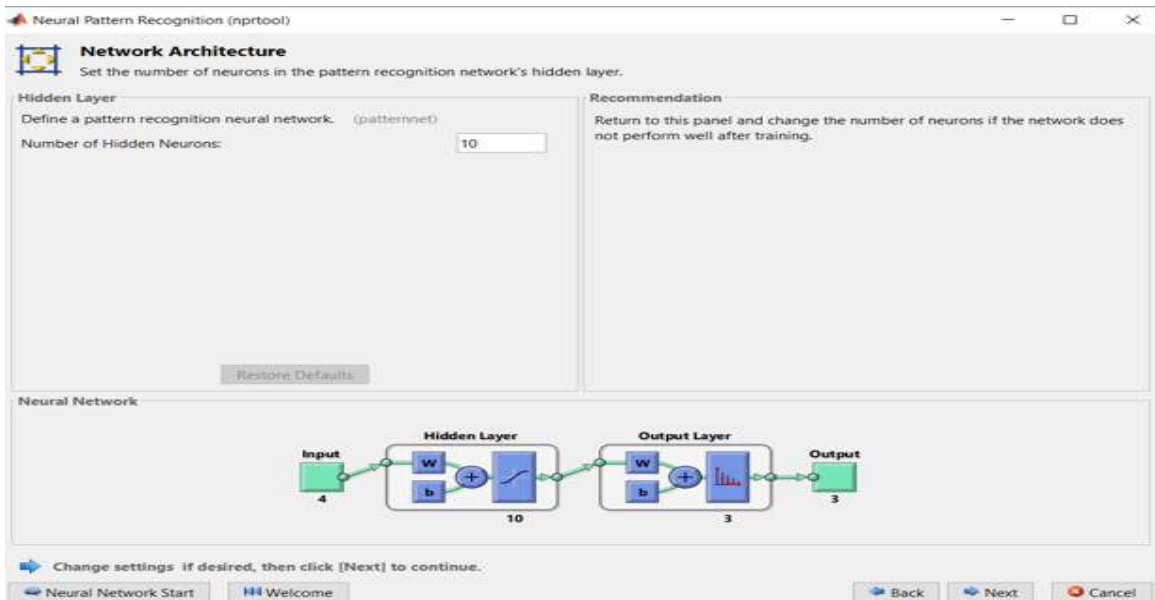


Figura 4.4 Neuronas de nuestra RNA

A continuación, se procede a entrenar la RNA con todas las características que se han configurado. Este proceso se muestra en la Figura 4.5.

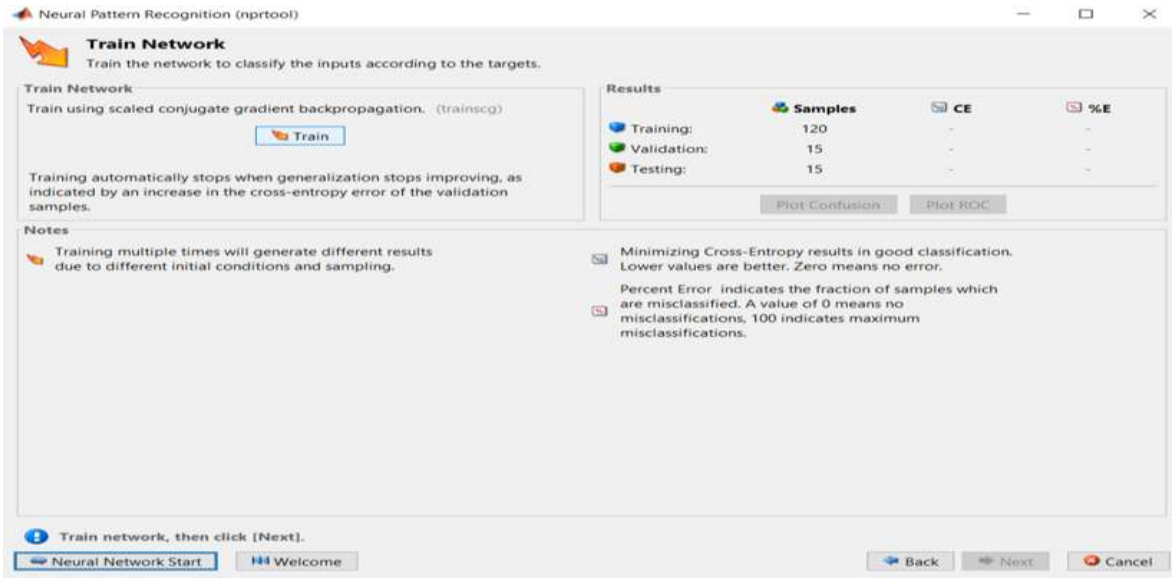


Figura 4. 5 Entrenamiento de la RNA

Una vez entrenada la red se crea una ventana como la mostrada en la Figura 4.6, donde pueden ser encontradas las características generadas de la matriz de confusión.

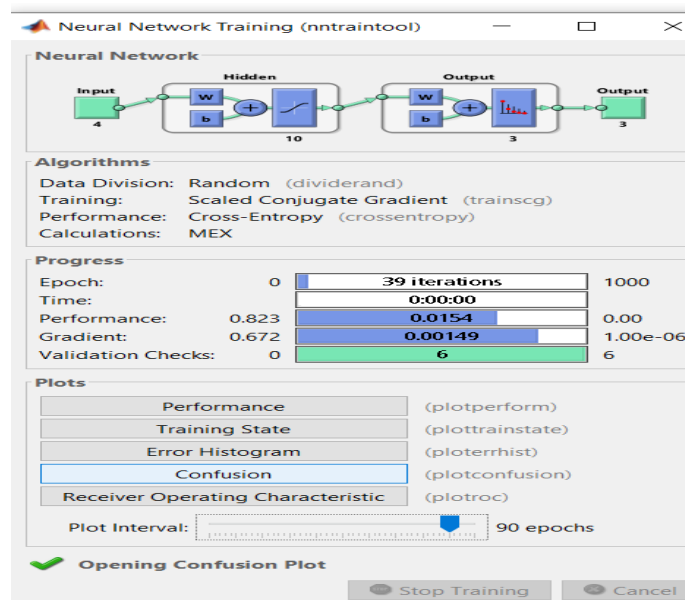


Figura 4. 6 Ventana de Neural Network Training.

En la Figura 4.7 se muestra la matriz de confusión creada para los datos de las flores de iris. Esta matriz muestra lo bien que se han clasificado todos los datos.

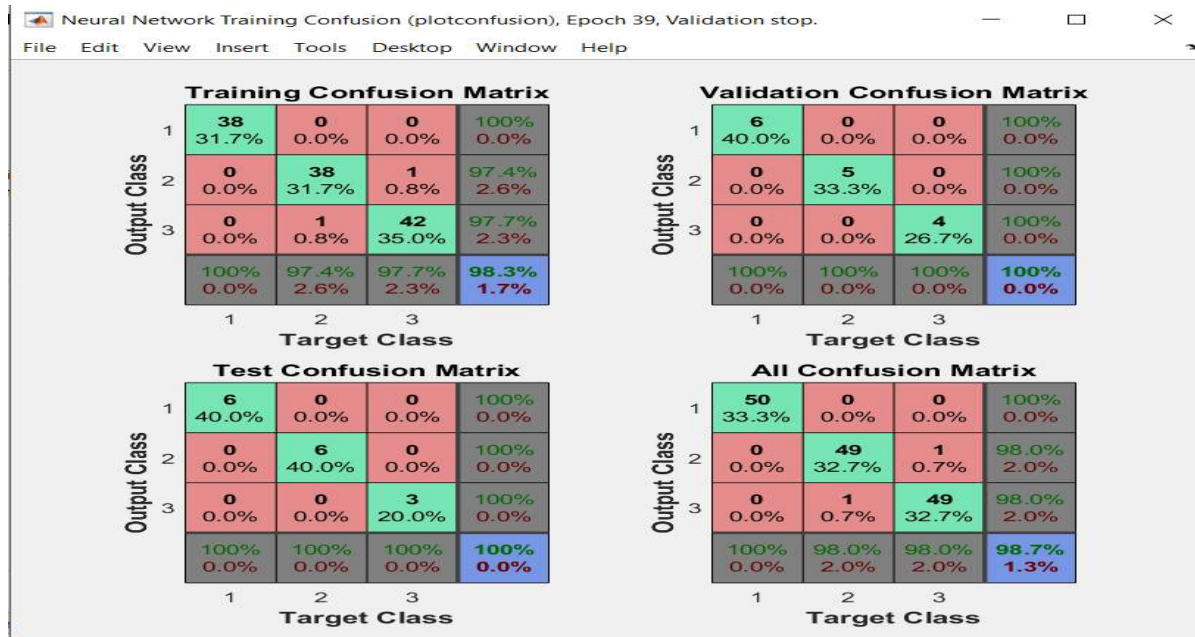


Figura 4. 7 Matriz de confusión para los datos de las flores de iris.

La matriz de confusión indica como a salido cada etapa del proceso de creación de la red neuronal. Para este caso en específico se puede ver que en el entrenamiento ha fallado en dos ocasiones: una donde identifica a una versicolor (planta) como virginica y otra ocasión donde clasifica a una virginica como a una versicolor, como se aprecia en Training Confusión Matrix de la Figura 4.7 (elementos fuera de la diagonal). En las demás matrices se puede observar que todas las clases han sido bien clasificadas, esto se puede ver más clara mente en el porcentaje del recuadro azul de la parte inferior derecha. Finalmente, en la matriz de confusión total la red neuronal clasificara un 98.7% bien los datos de este tipo de flores.

Para la generación del código se prosigue en la ventana de Neural Pattern Recognition, donde se puede obtener directamente una función que recibirá los nuevos datos a evaluar para poder ser clasificados. Además de la obtención de esta función y prosiguiendo con la ventana se puede obtener un Script donde se puede entrenar y probar una nueva red con diferentes parámetros de entrada y salida.

Es aquí donde a partir de este ejemplo y la generación de este Script se puede entrenar la red neuronal que se necesita para el reconocimiento textil. El script generado para la red neuronal de las flores de iris se muestra en el Código 4.7, 4.8 y 4.9. El cual es utilizado para la generación de la red neuronal que identificara la tela.

```

% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
%
% This script assumes these variables are defined:
%
% irisInputs - input data.
% irisTargets - target data.

x = irisInputs;
t = irisTargets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize, trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

```

Código 4. 7 Algoritmo de la red iris.

```

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'crossentropy'; % Cross-Entropy

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

```

Código 4. 8 Algoritmo de la red iris.

```

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end

```

Código 4. 9 Algoritmo de la red iris.

4.6.1 Principios

Las redes neuronales (también conocidas como sistemas conexionistas) son un modelo computacional basado en un gran conjunto de unidades neuronales simples (neuronas artificiales) de forma aproximadamente análoga al comportamiento observado en los axones de las neuronas en los cerebros biológicos. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida [37].

Niveles o capas de neuronas

La distribución de neuronas dentro de una red se realiza formando niveles o capas de un número de neuronas determinado, donde se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas) [37], [38].

Se pueden distinguir tres tipos de capas:

Capa de Entrada: Es la capa que recibe directamente la información proveniente de las fuentes externas de la red.

Capa Oculta: Son internas a la red y no tiene contacto directo con el exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de la capa oculta pueden estar interconectadas de diferentes maneras, lo que determina, junto con su número, las distintas tipologías de redes neuronales.

Capa de Salida: Transfieren información de la red hacia el exterior.

4.6.2 Redes neuronales

Una red neuronal es un modelo de computación cuya estructura de capas se asemeja a la estructura interconectada de las neuronas en el cerebro, con capas de nodos conectados. Una red neuronal puede aprender de los datos, de manera que se puede entrenar para que reconozca patrones, clasifique datos y pronostique eventos futuros.

Las redes neuronales descomponen las entradas en capas de abstracción. Se pueden entrenar con muchos ejemplos para que reconozcan patrones de voz o en imágenes, por ejemplo, igual que el cerebro humano. Su comportamiento está definido por la forma en que se conectan sus elementos individuales, así como por la importancia (o ponderación) de dichas conexiones. Estas ponderaciones se ajustan automáticamente durante el entrenamiento de acuerdo con una regla de aprendizaje especificada hasta que la red neuronal lleva a cabo la tarea deseada correctamente.

Las redes neuronales resultan especialmente adecuadas para llevar a cabo el reconocimiento de patrones a fin de identificar y clasificar objetos o señales en sistemas de voz, visión y control. También se pueden emplear para el modelado y la predicción de series temporales.

Una red neuronal combina diversas capas de procesamiento y utiliza elementos simples que operan en paralelo, y están inspiradas en los sistemas nerviosos biológicos. Consta de una capa de entrada, una o varias capas ocultas y una capa de salida. Las capas están interconectadas mediante nodos, o neuronas; cada capa utiliza la salida de la capa anterior como entrada, en la Figura 4.8 se puede apreciar un esquema de una RNA.

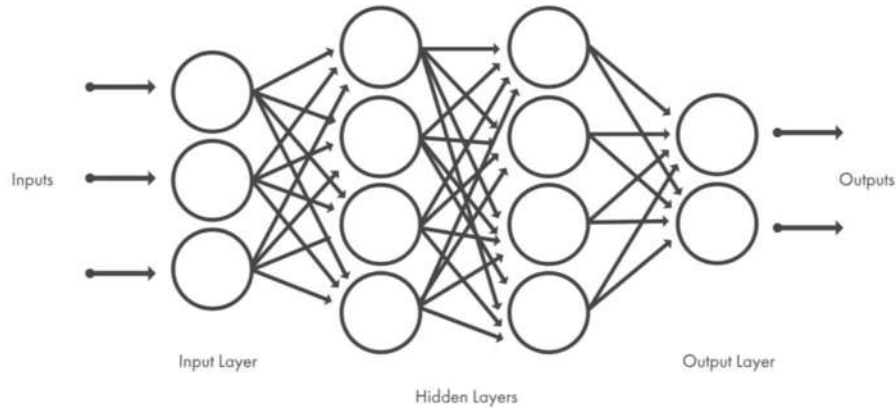


Figura 4. 8 Esquema de una RNA

Aprendizaje supervisado

Las redes neuronales supervisadas se entrenan para producir las salidas deseadas como respuesta a entradas de muestra, por lo que resultan idóneas para modelar y controlar sistemas dinámicos, clasificar datos con ruido y predecir eventos futuros.

Aprendizaje no supervisado

Las redes neuronales no supervisadas se entrenan permitiendo que la red neuronal se autoajuste continuamente a las nuevas entradas. Se emplean para inferir información a partir de conjuntos de datos que constan de datos de entrada sin respuestas etiquetadas. Se pueden utilizar para descubrir distribuciones naturales, categorías y relaciones entre categorías en los datos [39].

4.6.3 Obtención de la base de datos

Para la obtención de la base de datos se realiza la captura de imágenes mediante una cámara de un teléfono inteligente, para posteriormente ser pasadas a una computadora donde se ejecutará un script realizado en Matlab. Este script contiene un código el cual consiste en convertir cada una de estas fotos color blanco y negro, pues en este caso el color de la tela no se considera como un parámetro característico que ayude a la identificación. En segundo lugar, se divide la fotografía en cuatro partes, para posteriormente aplicar a cada una de estas partes la transformada de Fourier. Una vez que sea aplicada la transformada de Fourier a esa parte de la imagen, se procede a obtener distintos parámetros estadísticos, los cuales son: media, desviación estándar, sesgo y curtosis.

Este mismo código se encarga de agrupar los datos en una matriz vacía, la cual se llena conforme se van leyendo y aplicando la transformada de Fourier a cada una de las fotografías. En la Código 4.10 y 4.11, se muestra una parte del código de este script.

```

Fotos = dir('C:\Users\CAMILO\Documents\MATLAB\FOTO_PRUEBA\*.JPEG');

i = 1;

for k = 1:length(Fotos)
    archivo = Fotos(k).name;
    nombre='C:\Users\CAMILO\Documents\MATLAB\FOTO_PRUEBA\';
    Foto = imread(strcat(nombre,archivo));

    Orig_byn=rgb2gray(Foto);

    [RE, CO, ~] = size(Orig_byn);

    if i ==1
        BASE_4_PARTES = [];
        BASE_RE=0;
        i=2;
    end

    %-----PARTE A DE LA FOTO-----

    A= Orig_byn(1:RE/2 , 1:CO/2);
    A_RE = zeros(RE/2, RE/2);
    A_CO = zeros(CO/2, CO/2);

    for u = 0 : ((RE/2) - 1)
        for x = 0 : ((RE/2) - 1)
            A_RE(u+1, x+1) = exp(-2 * pi * 1i / (RE/2) * x * u);
        end
    end

    for v = 0 : ((CO/2) - 1)
        for y = 0 : ((CO/2) - 1)
            A_CO(y+1, v+1) = exp(-2 * pi * 1i / (CO/2) * y * v);
        end
    end

    FA = A_RE * im2double(A) * A_CO;
    F1 = FA.*255;

```

Código 4. 10 Algoritmo para la obtención de la base de datos.

```

BASE_RE = BASE_RE + 1;
BASE_CO = 1;

Media_F1 = median(F1);
Media_F1 = median(Media_F1);
BASE_4_PARTES) = Media_F1;    BASE_CO = BASE_CO + 1;

Des_Est_F1 = std(F1);
Des_Est_F1 = std(Des_Est_F1);
BASE_4_PARTES(BASE_RE,BASE_CO) = Des_Est_F1;    BASE_CO = BASE_CO + 1;

Sesgo_F1 = skewness(F1);
Sesgo_F1 = skewness(Sesgo_F1);
BASE_4_PARTES(BASE_RE,BASE_CO) = Sesgo_F1;    BASE_CO = BASE_CO + 1;

Kurtosis_F1 = kurtosis(F1);
Kurtosis_F1 = kurtosis(Kurtosis_F1);
BASE_4_PARTES(BASE_RE,BASE_CO) = Kurtosis_F1;    BASE_CO = BASE_CO + 1;

```

Código 4. 11 Algoritmo para la obtención de la base de datos.

4.6.3.1 Agrupación de datos

Debido a que se está realizando una identificación, se cuenta para esta prueba con 4 distintos tipos de tela, las cuales son: Algodón, Lino, Gabardina y Seda.

Estas telas han sido seleccionadas de modo que entre ellas existe un diferente patrón de cosido, además de que son telas que se usan día a día por un gran número de personas. Las telas se muestran en la Tabla 4.1.

Tabla 4. 1 Telas utilizadas para la generación la base de datos.

ALGODON	POLIÉSTER	GABARDINA	MEZCLIYA
			

Debido a que el código puede leer todas las fotografías que se encuentran en una carpeta, se colocan específicamente todas las fotografías de una misma tela en una carpeta con el nombre de la tela, de esta manera se puede tener más en claro el proceso que está realizando el script.

Además, debido a esto se obtienen en la base de datos todas las características agrupadas según sea la carpeta que se elija primero. En la base de datos se llenó de la siguiente manera: primero se seleccionó la carpeta con las fotografías de algodón, la segunda carpeta fue la que contenía las fotografías de poliéster, en tercer lugar, la carpeta de gabardina y por último la carpeta de mezclilla.

Cabe mencionar que cada carpeta contiene 100 fotografías de cada tipo de tela. Por lo que al dividir está en 4 partes se estarían generando 400, de modo que la base de datos tendría dimensiones de 1600x4.

La Tabla 4.2 muestra un esquema general de cómo está conformada la base de datos.

Tabla 4. 2 Esquema general de la base de datos.

...	MEDIA	DEVIACION ESTANDAR	SESGO	KURTOSIS
1	ALGODÓN			
.				
.				
.				
400	POLIESTER			
401				
.				
.				
800	GABARDINA			
801				
.				
.				
1200	MESCLILLA			
1201				
.				
.				
1600				

4.6.3.2 Asignación de identificador

Para saber a qué tipo de clase pertenece cada grupo de datos se asigna a la matriz una segunda columna, esta columna corresponde a la clase de cada tela, el 1 está representando a el algodón, el 2 al poliéster, el 3 a la gabardina y el 4 a la mezclilla.

De tal manera que la red pueda identificar qué descriptores corresponden a que clase. La base de datos que se obtiene se muestra en la Tabla 4.3.

Tabla 4. 3 Esquema general de la base de datos con clases

...	MEDIA	DEVIACION ESTANDAR	SESGO	KURTOSIS	CLASE
1	ALGODÓN				1
.					.
.					.
.					.
400	POLIESTER				1
401					2
.					.
.					.
800	GABARDINA				2
801					3
.					.
.					.
1200	MESCLILLA				3
1201					4
.					.
.					.
1600					4

4.6.4 Ingreso de base de datos a una RN (descriptores y clases)

Contando con el script que creara y entrenara la red neuronal además de la base de datos obtenida a partir de todas las fotografías, se procede a generar la red neuronal que detectara la tela que se está planchando.

Debido a que ya se cuenta con el código creado por la red anterior (Código 4.10 y 4.11) solo es necesario cambiar los datos de entrada y salida, es decir poner

los descriptores y clases obtenidas a partir de las fotografías en lugar de los descriptores y clases de la flor iris. Debido a que las clases se encuentran en la misma matriz (en la que se encuentran los descriptores), es necesario definir los límites de las columnas que serán tomadas para cada uno de ellos. En el Código 4.12 se muestra el código con nuestros descriptores y clases en él.

```

x = BASE_4_PARTES(:,1:4);
t = BASE_4_PARTES(:,5);
t = full(ind2vec(t));

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [30 30];
net = patternnet(hiddenLayerSize, trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% Choose a Performance Function
net.performFcn = 'crossentropy'; % Cross-Entropy

% Test the Network before training
net = configure(net,x,t);
y = net(x);
tind = vec2ind(t);
yind = vec2ind(y);

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y);
valPerformance = perform(net,valTargets,y);
testPerformance = perform(net,testTargets,y);

```

Código 4. 12 Algoritmo con nuestros descriptores y clases en él.

4.6.5 Matriz de Confusión y eficiencia

En el campo de la inteligencia artificial una matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

En la matriz ejemplo que aparece en la Figura 4.9, de 8 gatos reales, el sistema predijo que tres eran perros, y de seis perros predijo que uno era un conejo y dos eran gatos. A partir de la matriz se puede ver que el sistema tiene problemas distinguiendo entre gatos y perros, pero que puede distinguir razonablemente bien entre conejos y otros animales [40].

		Valor Predicho		
		Gato	Perro	Conejo
Valor Real	Gato	5	3	0
	Perro	2	3	1
	Conejo	0	2	11

Figura 4. 9 Matriz de confusión de ejemplo

En este caso, una vez que esta creada la red neuronal con los descriptores y clases se puede ver la matriz de confusión, esta indicará que tan bien se han clasificado todos los datos. Y de esta manera poder predecir de qué tipo de tela es la que se esté planchado. La matriz de confusión generada por la red neuronal se muestra en Figura 4.10.



Figura 4. 10 Matriz de confusión de la base de datos de las telas.

Debido a que la matriz de confusión presenta un 81.4% de probabilidades de poder identificar la tela correctamente se considera que es una probabilidad muy baja, por lo que será ineficiente.

4.6.6 Función Net

En el momento en el que se crea la red neuronal, el código genera un archivo en el workspace de Matlab con nombre "net", esta es una función donde se encuentra la red neuronal y puede ser usada como una función, donde las entradas serán los predictores que se obtengan de nuevas imágenes. Esta función se puede observar en la Figura 4.11.

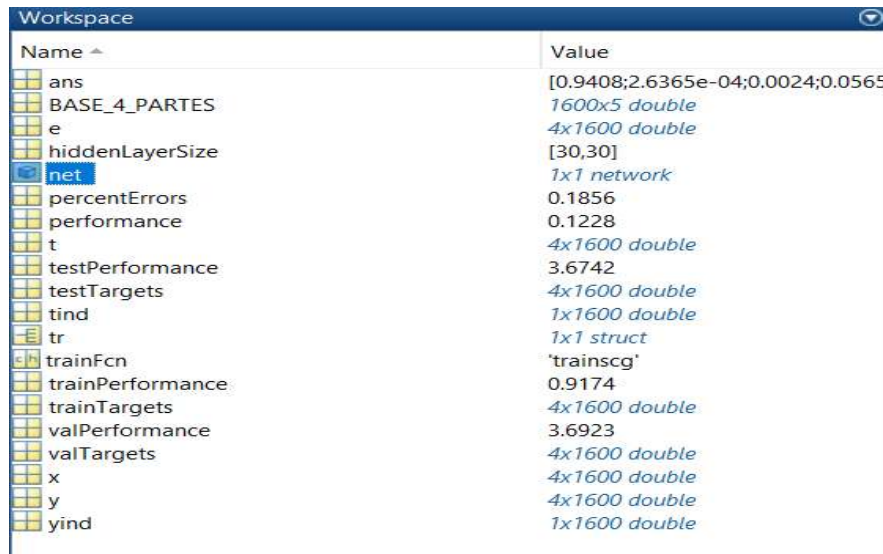


Figura 4. 11 Función net

En la Figura 4.12 se aprecia un claro ejemplo de su uso. Además, se observa que para la prueba realizada de algodón, esta genera una probabilidad del 94% de decir que es algodón.

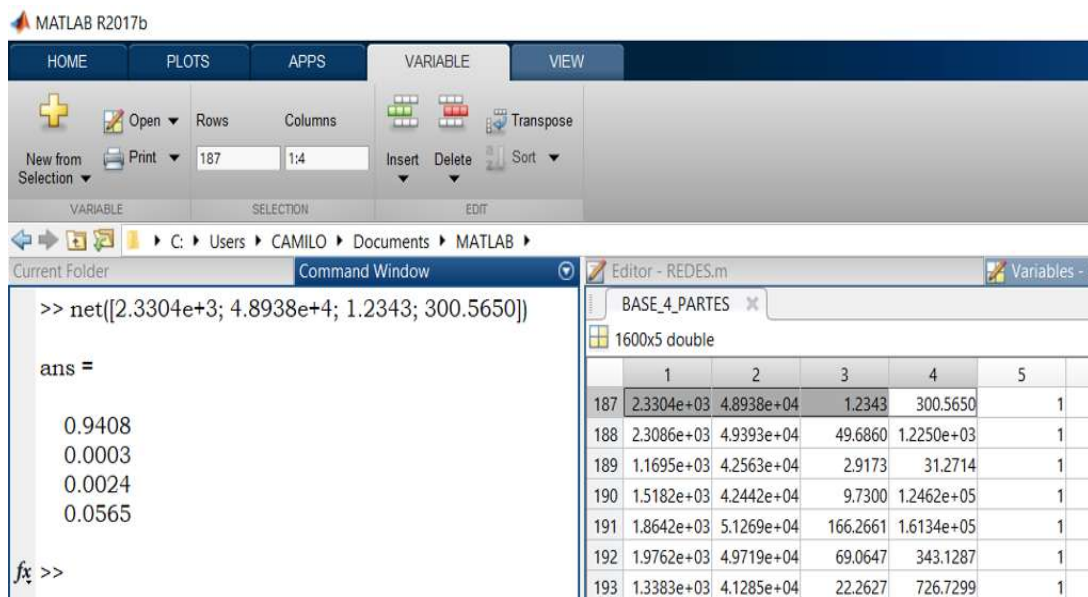
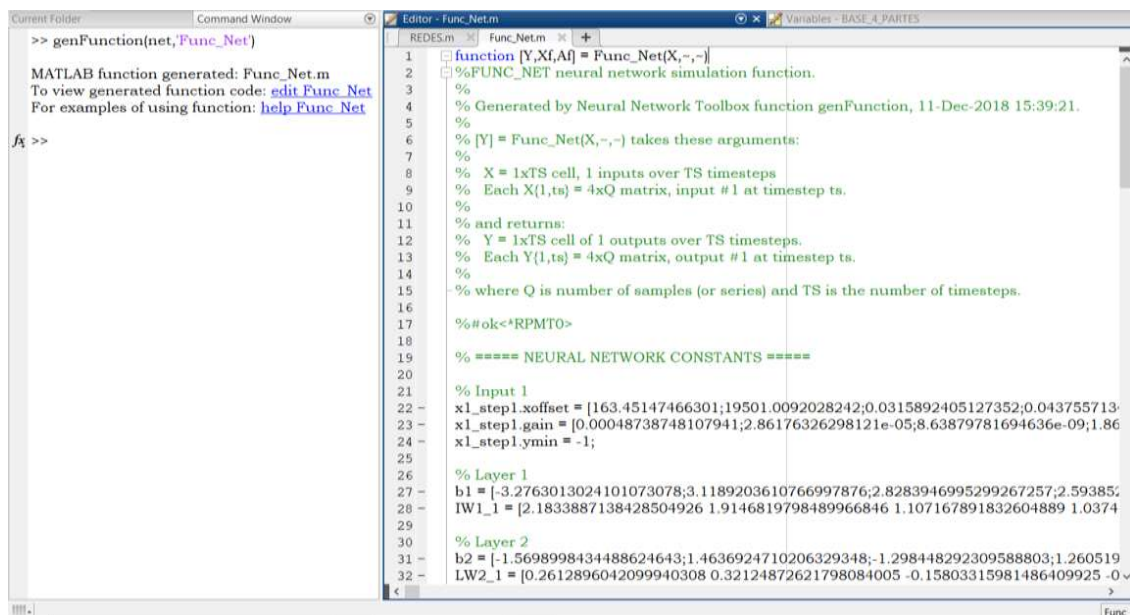


Figura 4. 12 Uso de la función net.

4.6.7 Conversión a código C

Debido a que se requiere que esta función este en un código C para poder ser leída por el Arduino, lo primero que se realiza, es usar la función “genFunction(net,pathname)” de Matlab, la cual genera una función Matlab independiente completa para simular una red neuronal que incluye todos los ajustes, valores de peso y sesgo, funciones de módulo y cálculos en un archivo.

Una vez que se utiliza la función “genFunction(net,pathname)” se genera un archivo con el nombre que en ese momento se propuso. Para este caso el nombre fue “Func_Net”. En la Figura 4.13 se muestra la utilización de esta función, así como el parte del código generado por esta.



The screenshot shows the MATLAB environment with the Command Window on the left and the Editor on the right. The Command Window displays the command `>> genFunction(net,'Func_Net')` and the resulting message: "MATLAB function generated: Func_Net.m. To view generated function code: [edit Func_Net](#). For examples of using function: [help Func_Net](#)". The Editor window shows the generated code for `Func_Net.m`, which is a neural network simulation function. The code includes comments and numerical values for weights and biases.

```
1 function [Y,Xf,Af] = Func_Net(X,-,-)
2 %FUNC_NET neural network simulation function.
3 %
4 % Generated by Neural Network Toolbox function genFunction, 11-Dec-2018 15:39:21.
5 %
6 % [Y] = Func_Net(X,-,-) takes these arguments:
7 %
8 % X = 1xTS cell, 1 inputs over TS timesteps
9 % Each X(1,ts) = 4xQ matrix, input #1 at timestep ts.
10 %
11 % and returns:
12 % Y = 1xTS cell of 1 outputs over TS timesteps.
13 % Each Y(1,ts) = 4xQ matrix, output #1 at timestep ts.
14 %
15 % where Q is number of samples (or series) and TS is the number of timesteps.
16
17 %#ok<^RPMTO>
18
19 %==== NEURAL NETWORK CONSTANTS =====
20
21 % Input 1
22 x1_step1.xoffset = [163.45147466301;19501.0092028242;0.0315892405127352;0.043755713-
23 x1_step1.gain = [0.00048738748107941;2.86176326298121e-05;8.63879781694636e-09;1.86
24 x1_step1.ymin = -1;
25
26 % Layer 1
27 b1 = [-3.2763013024101073078;3.1189203610766997876;2.8283946995299267257;2.59385-
28 IW1_1 = [2.1833887138428504926 1.9146819798489966846 1.107167891832604889 1.0374
29
30 % Layer 2
31 b2 = [-1.5698998434488624643;1.4636924710206329348;-1.298448292309588803;1.260519-
32 LW2_1 = [0.2612896042099940308 0.32124872621798084005 -0.15803315981486409925 -0.0
```

Figura 4.13 Generación de la función Func_Net

Se corrobora que la función creada también tenga las mismas características que la “net”, por lo que al ser evaluada con los mismos parámetros de entrada genera las mismas salidas. De modo que esta función “Func_Net” está funcionando correctamente y de la misma manera que la “net”.

En la Figura 4.14 se presenta la evaluación de ambas funciones, presentando los mismos resultados.

The screenshot shows two windows from the MATLAB IDE. On the left is the Command Window, and on the right is the Editor window.

Command Window:

```
>> net([2.3304e+3; 4.8938e+4; 1.2343; 300.5650])
ans =
    0.9408
    0.0003
    0.0024
    0.0565

>> X = [2.3304e+3; 4.8938e+4; 1.2343; 300.5650];
>> Func_Net(X)
ans =
    0.9408
    0.0003
    0.0024
    0.0565

fx >>
```

Editor - Func_Net.m:

```
1 function [Y] = Func_Net(X)
2 %FUNC_NET neural network simulation f
3 %
4 % Generated by Neural Network Toolbox f
5 %
6 % [Y] = Func_Net(X,-,-) takes these argun
7 %
8 % X = 1xTS cell, 1 inputs over TS timeste
9 % Each X{1,ts} = 4xQ matrix, input #1 at
10 %
11 % and returns:
12 % Y = 1xTS cell of 1 outputs over TS time
13 % Each Y{1,ts} = 4xQ matrix, output #1 at
14 %
15 % where Q is number of samples (or series
16
17 %#ok<*>RPMT0>
18
19 % ===== NEURAL NETWORK CONSTANTS
20
21 % Input 1
22 x1_step1.xoffset = [163.45147466301;1950
```

Figura 4. 14 Comprobación de los resultados de ambas funciones.

Una vez que la función “Func_Net” está funcionando correctamente se procede a generar este mismo código a un código C, esto con la finalidad de convertir todos aquellos comandos que no pueden ser entendidos por el IDE de Arduino.

Para realizar esta tarea se utiliza otra app de Matlab, esta es la “Matlab Coder” que es capaz de genera códigos C y C ++ a partir de códigos Matlab para una variedad de plataformas de hardware, desde sistemas de escritorio hasta hardware integrado.

En las Figura 4.15 se muestra que primeramente se selecciona la función que deseamos convertir de código de Matlab a código C.



Figura 4. 15 Ingreso de la función a convertir a código C.

Posteriormente en la Figura 4.16 se especifican las entradas y las dimensiones que estas tendrán.

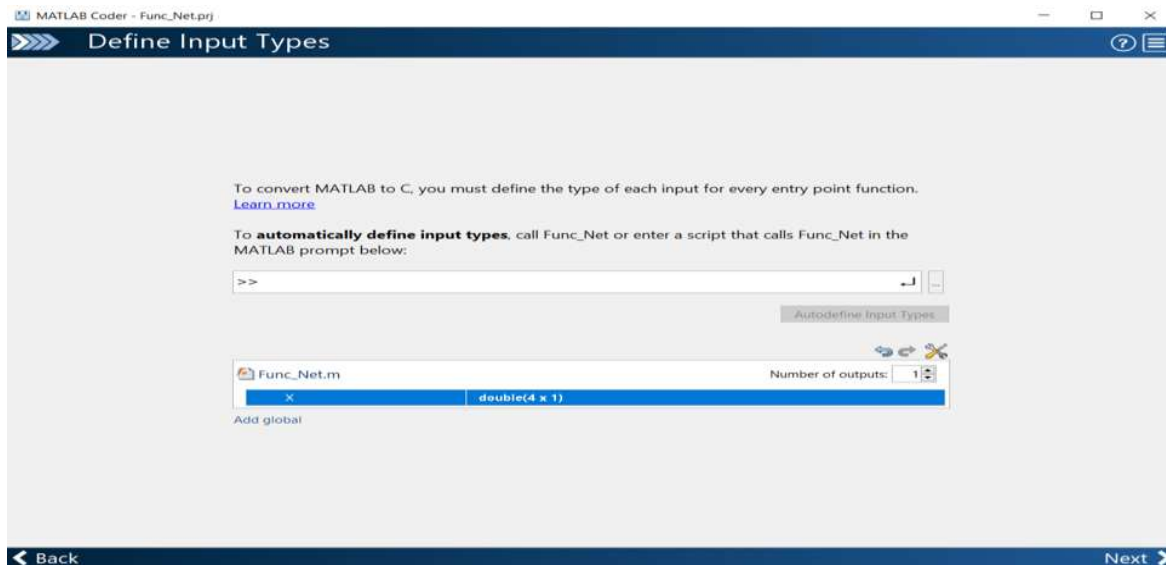


Figura 4. 16 Especificaciones de entradas de la función Func_Net.

Una vez terminado el proceso anterior, se realiza la generación del código seleccionado. En este caso la conversión se hace a código C como se muestra en la Figura 4.17.

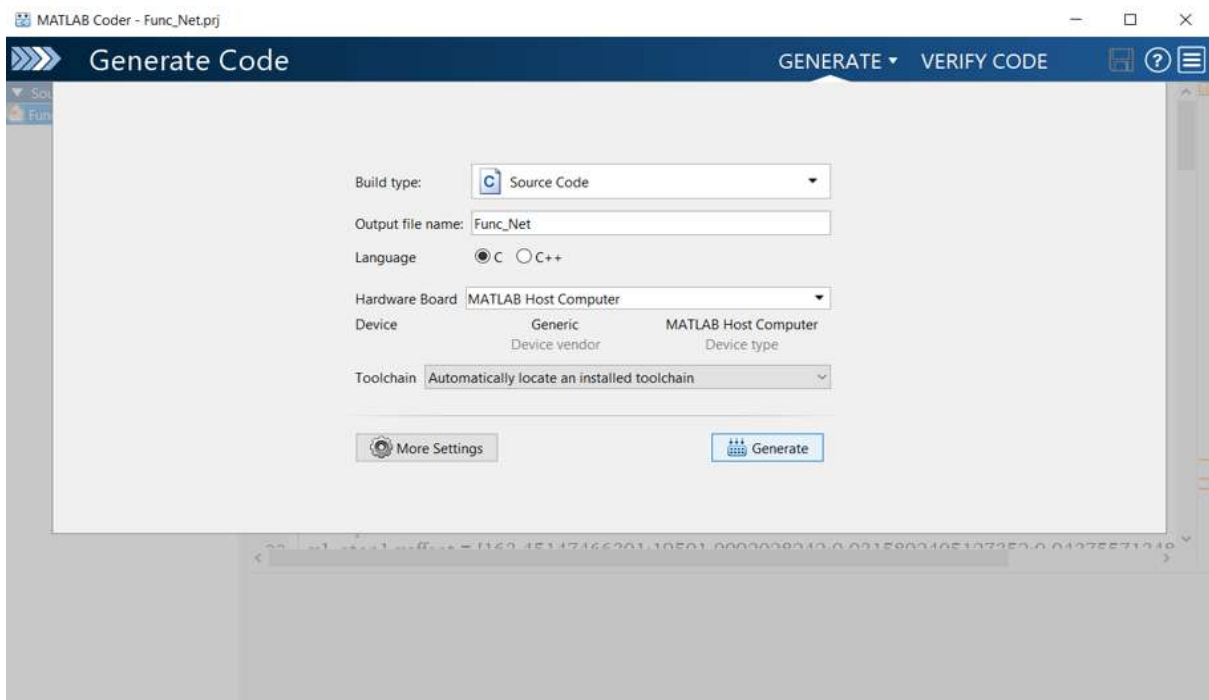


Figura 4. 17 Selección de código a convertir listo para ser generado.

Una vez que la conversión ha sido exitosa se generan los correspondientes archivos con extensión .c y .h los cuales contienen en su interior el código de la función “Func_Net” pero ahora en el lenguaje de programación que ha sido seccionado previamente. En la Figura 4.18 se aprecian dichos archivos generados, así también como parte del código que contiene.

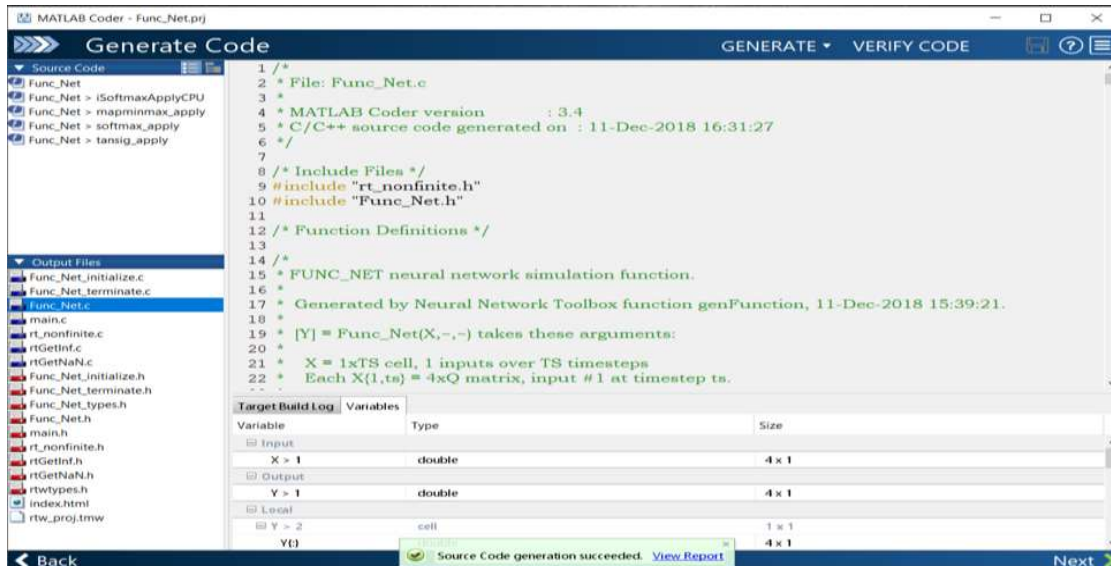


Figura 4. 18 Archivos .c y .h generados por Matlab Coder.

4.6.8 Implementación en Arduino

Una vez que se cuenta con el código en C, generado por la app de Matlab, se implementa este código en el IDE de Arduino. Con el fin de comprobar que de igual manera que se generó la función “Func_Net”, este pueda también dar los mismos resultados obtenidos antes de su conversión.

El código, aunque ya se encuentra en C es necesario adaptarlo de tal manera que se pueda tener un concepto claro del proceso que se está realizando, pues el código generado cuenta con algunas diferencias con respecto al original.

En el Código 4.13 se muestra una parte del código generado por la Matlab Coder, en el IDE de Arduino debido a que este es demasiado extenso para poder ser mostrado.

```

/* Include Files */
#include "rt_nonfinite.h"
#include "Func_Net.h"

void Func_Net(const double X[4], double Y[4])
{
    int k;
    double numerator[4];
    static const double dv0[4] = { 163.45147466301, 19501.0092028242,
        0.0315892405127352, 0.0437557134898461 };

    static const double dv1[4] = { 0.00048738748107941, 2.86176326298121E-5,
        8.63879781694636E-9, 1.86422497765597E-8 };

    double Xp1[4];
    double denominator;
    int ix;
    double y[30];
    static const double a[30] = { -3.2763013024101073, 3.1189203610767,
        2.8283946995299267, 2.593852183977154, -2.1471783108366522,
        -2.1385763215076743, -1.5108854258944708, -1.4519147438857616,
        -1.4967928597140006, -1.097889426314451, -1.0044692799102746,
        -0.853568299988867, -0.46670252508590759, -0.1346250953916612,
        -0.054554152736467479, 0.12941940164017648, -0.59148935863006014,
        0.7188471955821546, -0.6301390826538581, -0.82769960073822579,

```

Código 4. 13 Código generado por Matlab Coder en C.

A este código es necesario editarlo de tal manera que se adecue a las necesidades que se requieren. El código editado se muestra a continuación en el Código 4.14.

```

#include "rt_nonfinite.h"
#include "rt_nonfinite.c"
#include "Func_Net.h"

void setup() {
    Serial.begin(9600);
}

void loop() {

    Serial.println("en la funcion LOOP");
    double valor1 = 1;
    double valor2 = 2;
    double valor3 = 3;
    double valor4 = 4;

    double Z = Func_Net(valor1,valor2,valor3,valor4);
    delay(2000);
}

double Func_Net(double valor1,double valor2,double valor3,double valor4)
{
    Serial.println("en la funcion NET");
    const double Mis4Entradas[4]= {valor1,valor2,valor3,valor4};
    double Mis4Salidas[4];

```

Código 4. 14 Código editado para el funcionamiento de nuestro sistema.

4.7 Conclusiones

En el presente capítulo se puede hacer mención de que la mayoría de los códigos desarrollados son relativamente fáciles de elaborar, esto debido a que las placas de desarrollo como las de Arduino son de la preferencia de muchos desarrolladores de código, pues la mayoría de los sensores son fáciles de leer o de hacer funcionar, esto también debido a que Arduino tiene una gran facilidad y capacidad para la implementación de códigos, además de que cuenta con una gran cantidad de librerías que son de mucha utilidad a la hora de estar censando cualquier tipo de dispositivo.

Esto no significa que la programación siempre sea fácil, pues en este caso se presentaron dificultades con el código que se generó en Matlab. Teniendo esto en cuenta, es necesario tener un conocimiento básico de programación, pues son estos con los que se ha podido solucionar dicho problema. Además de que también se han presentado más problemas al tratar de desarrollar el código de la red neuronal en el IDE de Arduino ya que el código en C generado por Matlab no era idéntico al de origen.

Capítulo 5

5. Conclusiones Y Trabajos Futuros

5.1 Conclusiones

Sin duda alguna con la realización de este proyecto se puede apreciar el avance que se presenta en todos los ámbitos, pues se deja ver que no solo la plancha de ropa ha venido cambiando poco a poco desde hace mucho tiempo, pues al pensar un poco sobre los demás objetos de la vida diaria, como lo puede ser un vehículo de transporte, éste también ha sufrido cambios significativos.

Con la implementación del hardware se puede hacer notar que, puede ser utilizado en una amplia gama de sistemas, con el objetivo de lograr automatizar algún proceso. Para el caso de la presente tesis, dicho hardware fue utilizado para lograr hacer un poco más 'inteligente' un aparato electrodoméstico tradicional.

Es por eso que los microcontroladores toman una gran importancia en la elaboración de este tipo de sistemas donde se plantea la utilización de uno o varios sensores, pues son capaces de realizar tareas repetitivas que serán usadas de la mejor manera para acoplarlas a la necesidad que se tenga.

Debido a toda la implementación que se ha desarrollado para el sistema propuesto, se puede decir que ahora será una plancha de ropa mucho más segura y con la capacidad de poder detectar automáticamente el tipo de tela, lo cual ara que se convierta de cierta manera en conjunto con todos los demás sensores en una plancha inteligente. Es también por esto que en la actualidad a los celulares se les suele decir smartphone, pues cuentan con diversos sensores que actúan de tal manera como si en realidad este fuera de cierta manera inteligente.

5.2 Trabajos futuros

Para los trabajos futuros es de gran interés para el desarrollador de esta tesis la implementación completa de este sistema.

Además de desarrollar un empaque totalmente nuevo y de acuerdo a las necesidades que se requiera.

Desarrollar un reconocimiento textil que sea más eficiente, logrando obtener un porcentaje por encima de 98% o alcanzar el 100%

Uno de los grandes avances en la tecnología es la carga inalámbrica, pudiendo ser este principio aplicado a un modelo posterior para eliminar el cable de alimentación de la plancha.

Apagado automático, esta función aún no se encuentra en el sistema, pero sería de gran ayuda al igual que para la seguridad también lo sería para el ahorro de energía eléctrica.

Bibliografía

- [1] <http://www.curiosfera.com/historia-de-la-plancha/>
- [2] <https://es.slideshare.net/julianaldana777/linea-del-tiempo-de-la-plancha>
- [3] <http://www.guiaspracticas.com/planchas-de-ropa>
- [4] <http://yesukama.blogspot.com/2009/05/funcionamiento-de-la-plancha-la.html>
- [5] https://es.wikipedia.org/wiki/Plancha_de_ropa#Centro_de_planchado
- [6] https://es.wikipedia.org/wiki/Centro_de_planchado
- [7] <https://centrosdeplanchado.com/plancha-vertical>
- [8] <http://www.guiaspracticas.com/planchas-de-ropa>
- [9] “análisis de la plancha, clasificación de sistemas y subsistemas”
- [10] <https://es.wikipedia.org/wiki/Termostato>
- [11] <https://www.arduino.cc/en/Guide/Introduction>
- [12] <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- [13] <https://store.arduino.cc/usa/arduino-uno-rev3>
- [14] <https://store.arduino.cc/usa/arduino-nano>
- [15] <https://store.arduino.cc/usa/arduino-mini-05>
- [16] https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-grandmax_brose_2012-21
- [17] <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>
- [18] https://www.raspberrypi.org/app/uploads/2018/11/Raspberry_Pi_3A_product_brief.pdf
- [19] <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
- [20] <https://www.littlebird.com.au/tilt-sensor-at407#description>
- [21] <https://www.sparkfun.com/datasheets/Sensors/Ball-Rolling%20Switch%20AT.pdf>
- [22] https://hobbyking.com/en_us/keyes-ky-017-mercury-switch-module-for-arduino.html?store=en_us

- [23] <https://store.invensense.com/datasheets/invensense/MPU9250REV1.0.pdf>
- [24] <https://www.luisllamas.es/usar-arduino-con-los-imu-de-9dof-mpu-9150-y-mpu-9250/>
- [25] <http://www.areatecnologia.com/electricidad/servomotor.html>
- [26] <https://robu.in/product/towerpro-mg995-metal-gear-servo-motor/>
- [27] https://es.wikipedia.org/wiki/Sensor_capacitivo
- [28] <https://www.luisllamas.es/interruptor-touchless-con-arduino-y-sensor-capacitivo/>
- [29] <http://www.electronicapty.com/tienda/modulos-y-sensores-para-arduino/mini-bomba-de-agua-para-arduinios-336-detail>
- [30] <http://www.arducam.com/products/camera-breakout-board/0-3mp-ov7670/>
- [31] <https://www.sparkfun.com/datasheets/Sensors/Imaging/1274419957.pdf>
- [32] <https://www.adafruit.com/product/3202>
- [33] <https://github.com/TKJElectronics/KalmanFilter>
- [34] <https://www.rinconingenieril.es/sensor-capacitivo/>
- [35] <https://www.arduino.cc/en/Reference/Servo>
- [36] <https://www.luisllamas.es/bomba-de-agua-con-arduino/>
- [37] https://es.wikipedia.org/wiki/Red_neuronal_artificial
- [38] http://members.tripod.com/jesus_alfonso_lopez/RnaIntro2.html
- [39] <https://la.mathworks.com/discovery/neural-network.html>
- [40] https://es.wikipedia.org/wiki/Matriz_de_confusi%C3%B3n
- [41] <https://github.com/>
- [42] <https://la.mathworks.com/products/matlab.html>