

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Sistema inalámbrico para la medición de tiempos de reacción,  
por medio de un estímulo visual**

Trabajo de graduación presentado por Marcel André Velásquez Guzmán  
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2020







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Sistema inalámbrico para la medición de tiempos de reacción,  
por medio de un estímulo visual**

Trabajo de graduación presentado por Marcel André Velásquez Guzmán  
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2020



Vo.Bo.:



(f) \_\_\_\_\_  
MAEB. Pablo Daniel Mazariegos de la Cerda

Tribunal Examinador:



(f) \_\_\_\_\_  
MAEB. Pablo Daniel Mazariegos de la Cerda



(f) \_\_\_\_\_  
MSc. Carlos Alberto Esquit Hernandez



(f) \_\_\_\_\_  
MSc. José Eduardo Morales Espinoza

Fecha de aprobación: Guatemala, 17 de junio de 2020.



Cuando comencé mis estudios en la Universidad del Valle y me inscribí en la carrera Ingeniería Electrónica, nunca imaginé que iban a ser los 5 años más significativos de mi vida. Estudiar esta carrera fue una de las mejores experiencias de mi vida. Experimenté una gran cantidad de buenos momentos y una gran cantidad de malos momentos también. Mientras escribo esta capítulo de mi trabajo de graduación, pienso en todo lo que pase al lado de mis amigos y mis catedráticos. Y no puedo estar más agradecido con toda estas personas que llegué a conocer o coincidir en alguno momento. Porque todas estas personas dejaron aunque sea un granito en este rejol de arena. Y al igual que ellos me dejaron algo de que recordar, espero que yo también les haya dejado algo de mi que quede para siempre en sus recuerdos.

Primero quiero agradecer a todos mis catedráticos y la Universidad del Valle por darme todas las herramientas para poder completar mi Trabajo de Graduación y guiarme en el camino del éxito.

También quiero agradecer a mis compañeros y amigos. Sin ellos sé que estos 5 años hubieran sido los más difíciles de mi vida. Gracias al grupo de Electrónicos por todas esas risas en la cafetería, por esas salidas de viernes para despejarnos de la Universidad y por esos almuerzo en donde nuestra preocupación máxima era buscar una mesa adentro para poder almorzar tranquilos y sin calor. De este grupo de amigos, quiero agradecerle a dos personas en específico: Pablo Muñoz y Josue cifuentes, porque desde que comenzaron estos 5 años siempre estuvimos los tres apoyándonos y a pesar de que existieron malos momentos entre nosotros, siempre seguimos siendo el tridente.

Y por último, quiero agradecer a mi familia. A mi mamá, Olga Guzmán, por siempre apoyarme. A mi papá, Eugener Velásquez, por siempre guiarme en el camino del bien. A mi Hermano, Eugene Velásquez, por ser una ejemplo a seguir. Y a mi novia Waleska Girón, que estuvo en todos los buenos y malos momentos en estos 5 años porque nunca dejó de creer en mi. Los amo a todos.

Espero haber dejado, a todas las personas que conocí en este trayecto, un poco de la felicidad que siempre entrego. Los quiero.



<b>Prefacio</b>	<b>v</b>
<b>Lista de figuras</b>	<b>XIII</b>
<b>Lista de cuadros</b>	<b>XV</b>
<b>Resumen</b>	<b>XVII</b>
<b>Abstract</b>	<b>XIX</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
<b>3. Justificación</b>	<b>5</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo general . . . . .	7
4.2. Objetivos específicos . . . . .	7
<b>5. Alcance</b>	<b>9</b>
<b>6. Marco teórico</b>	<b>11</b>
6.1. Tiempo de reacción . . . . .	11
6.1.1. ¿Qué es el tiempo de reacción? . . . . .	11
6.1.2. ¿Cómo se mide el tiempo de reacción? . . . . .	12
6.1.3. ¿Cómo mejorar el tiempo de reacción? . . . . .	12
6.2. Radiofrecuencia . . . . .	12
6.2.1. ¿Qué es radiofrecuencia? . . . . .	12
6.2.2. Aplicación en las comunicaciones . . . . .	12
6.2.3. Radiofrecuencia RFID . . . . .	13
6.3. Protocolo de comunicación . . . . .	13
6.3.1. ¿Qué es un protocolo? . . . . .	13
6.3.2. Protocolos orientados a conexión . . . . .	13

6.3.3. Implementación de un protocolo . . . . .	13
6.4. Comunicación SPI . . . . .	13
6.5. Modulación por ancho de pulso . . . . .	15
6.6. Servidor web . . . . .	15
6.6.1. ¿Qué es un servidor web? . . . . .	15
<b>7. Selección de equipo electrónico</b>	<b>17</b>
7.1. Sensor ultrasónico JSN-SR04T-2.0 . . . . .	17
7.2. Módulo de radiofrecuencia RFM69HC 433MHz . . . . .	17
7.3. Feather 32u4 RFM69HCW . . . . .	18
7.4. NeoPixel Diffused 5mm . . . . .	19
<b>8. Definición del protocolo de comunicación</b>	<b>21</b>
8.1. Significado de cada Byte . . . . .	21
8.1.1. ID Dpositivo . . . . .	21
8.1.2. Color . . . . .	22
8.1.3. Test . . . . .	22
8.1.4. ON/OFF . . . . .	23
8.1.5. Deporte . . . . .	23
8.1.6. Entrenamiento . . . . .	23
8.1.7. Sensor . . . . .	23
8.1.8. N/A . . . . .	23
<b>9. Comunicación por radiofrecuencia entre el dispositivo esclavo y maestro</b>	<b>25</b>
9.1. Configuración del módulo RFM69HCW en la Raspberry Pi 3 . . . . .	25
9.2. Programación dispositivo maestro . . . . .	26
9.2.1. Programa FuncionesVariables.py . . . . .	26
9.2.2. Programa RFM69Rasp.py . . . . .	27
9.3. Programación dispositivo esclavo . . . . .	28
<b>10. Diseño electrónico del dispositivo esclavo</b>	<b>31</b>
10.1. Librería Feather32u4 . . . . .	31
10.2. Esquemático utilizado para el diseño de la placa principal . . . . .	31
10.3. Diseño del PCB para la placa principal . . . . .	33
10.4. Esquemático utilizado para el diseño de la placa secundaria . . . . .	33
10.5. Diseño del PCB para la placa secundaria . . . . .	34
<b>11. Diseño mecánico del dispositivo esclavo</b>	<b>35</b>
11.1. Pieza inferior del diseño mecánico del dispositivo esclavo . . . . .	36
11.2. Pieza soporte para la placa principal . . . . .	36
11.3. Pieza intermedia del diseño mecánico del dispositivo esclavo . . . . .	36
11.4. Pieza soporte para la placa secundaria . . . . .	37
11.5. Pieza superior del diseño mecánico del dispositivo esclavo . . . . .	38
11.6. Pieza superior del diseño mecánico del dispositivo esclavo . . . . .	39

<b>12. Implementación del dispositivo para la medición del tiempo de reacción de un deportista</b>	<b>41</b>
12.1. Conexión Raspberry Pi 3 y RFM69HCW . . . . .	41
12.2. Pruebas de comunicación RFM69HCW+Raspberry Pi 3 y Feather 32u4 RFM69HCW	42
12.2.1. Envío de datos utilizando protocolo de comunicación . . . . .	43
12.3. Utilización de los sensores capacitivo y ultrasónico en el Feather 32u4 RFM69HCW	45
12.3.1. Pruebas con el sensor ultrasónico JSN-SR04T-2.0 en el Feather 32u4 RFM69HCW . . . . .	45
12.3.2. Pruebas con el sensor capacitivo en el Feather 32u4 RFM69HCW . . .	46
12.4. Pruebas para las rutinas diseñadas para los deportes: softball, basquetball y box . . . . .	47
12.5. Construcción del dispositivo Esclavo . . . . .	51
<b>13. Aplicación web</b>	<b>53</b>
13.1. Diseño de página web . . . . .	53
13.1.1. Página login y sign-up . . . . .	53
13.1.2. Página menú principal . . . . .	54
13.1.3. Página entrenos . . . . .	55
13.2. Servidor web en el dispositivo maestro . . . . .	55
13.2.1. Instalación de Django . . . . .	56
13.2.2. Instalación de PostgreSQL base de datos . . . . .	56
13.3. Raspberry Pi 3 como punto de acceso inalámbrico . . . . .	57
13.4. Implementación de rutinas en página web . . . . .	61
<b>14. Conclusiones</b>	<b>63</b>
<b>15. Recomendaciones</b>	<b>65</b>
<b>16. Bibliografía</b>	<b>67</b>
<b>17. Anexos</b>	<b>69</b>
17.1. Piezas del diseño mecánico . . . . .	69
17.2. Conexiones . . . . .	72
17.3. Impresión placa usando la fresadora . . . . .	74



---

## Lista de figuras

---

1.	Diagrama de funcionamiento de la comunicación SPI [6]	14
2.	PWM con distintos ciclos de trabajo [7]	15
3.	Servidor web [8]	15
4.	Sensor ultrasónico JSN-SR04T-2.0	18
5.	RFM69HCW	18
6.	Feather 32u4 RFM69HCW	19
7.	NeoPixel Diffused 5mm	19
8.	Software de configuración de la Raspberry Pi 3	26
9.	Menú de entrenos del dispositivo maestro	28
10.	Esquemático del componente electrónico Feather32u4 RFM69HCW	32
11.	Diseño del circuito impreso del componente electrónico Feather32u4 RFM69HCW	32
12.	Esquemático dispositivo esclavo placa principal	32
13.	Diseño PCB placa principal	33
14.	Esquemático dispositivo esclavo placa secundaria	34
15.	Diseño PCB placa secundaria	34
16.	Modelado 3D completo del diseño mecánico	35
17.	Pieza inferior del diseño mecánico del dispositivo esclavo	36
18.	Soporte inferior para la placa principal del diseño mecánico del dispositivo esclavo	37
19.	Parte intermedia del diseño mecánico del dispositivo esclavo	37
20.	Soporte superior para la placa secundaria del diseño mecánico del dispositivo esclavo	38
21.	Rueda para ensamblaje de la placa	38
22.	Rueda para ensamblaje de la placa	39
23.	Modelo de la placa principal	39
24.	Modelo de la placa secundaria	40
25.	Conexión virtual de la Raspberry Pi 3 y del módulo RFM69HCW	42
26.	Primera prueba de comunicación con el dispositivo esclavo desde la Raspberry Pi 3	43

27.	Primera prueba de comunicación con el dispositivo maestro desde el Feather32u4 RFM69HCW . . . . .	43
28.	Programación para la recepción de datos del dispositivo maestro . . . . .	44
29.	Datos recibidos en el Feather 32u4 RFM69HCW . . . . .	45
30.	Protocolo utilizado en la prueba unidireccional de comunicación . . . . .	45
31.	Programación para la medición de distancia de un objeto utilizando el sensor JSN-SR04T-2.0 . . . . .	46
32.	Resultados de la medición de distancia utilizando el sensor JSN-SR04T-2.0 . . . . .	46
33.	Programación para el sensor capacitivo . . . . .	47
34.	Resultados de las pruebas utilizando el sensor capacitivo . . . . .	47
35.	Prueba entrenamiento 1 basquetball . . . . .	48
36.	Prueba entrenamiento 2 basquetball . . . . .	48
37.	Prueba entrenamiento 3 basquetball . . . . .	49
38.	Prueba entrenamiento 1 box . . . . .	49
39.	Prueba entrenamiento 2 box . . . . .	50
40.	Prueba entrenamiento 3 box . . . . .	50
41.	Prueba entrenamiento 1 softball . . . . .	51
42.	Prueba entrenamiento 2 softball . . . . .	51
43.	Prueba entrenamiento 3 softball . . . . .	51
44.	Pieza sensores del dispositivo esclavo . . . . .	52
45.	Construcción dispositivo esclavo . . . . .	52
46.	Dispositivo esclavo ensamblado . . . . .	52
47.	Página web . . . . .	54
48.	Sing-up de la página web . . . . .	54
49.	Menú principal página web . . . . .	55
50.	Entrenos página web . . . . .	55
51.	Configuración dhcpcd . . . . .	57
52.	Configuración IP estática . . . . .	58
53.	Configuración hostapd . . . . .	59
54.	Configuración dnsmasq . . . . .	60
56.	Conexión WiFi . . . . .	60
55.	Conexión red WiFi . . . . .	61
57.	Carpeta Training . . . . .	61
58.	Función de entreno en archivo Views.py . . . . .	62
59.	Corte láser parte inferior . . . . .	69
60.	Corte láser parte en medio . . . . .	70
61.	Corte láser parte superior . . . . .	70
62.	Soporte placa principal . . . . .	71
63.	Soporte 1 placa secundaria . . . . .	71
64.	Soporte 2 placa secundaria . . . . .	72
65.	Conexión física de la Raspberry Pi 3 y del módulo RFM69HCW . . . . .	72
66.	Montaje del sensor ultrasónico en el Feather 32u4 RFM69HCW . . . . .	73
67.	Montaje del sensor capacitivo en el Feather 32u4 RFM69HCW . . . . .	73
68.	Placa principal vista 1 . . . . .	74
69.	Placa principal vista 2 . . . . .	74
70.	Placa secundaria vista 1 . . . . .	75

71. Placa secundaria vista 2 . . . . .	75
--	----



---

Lista de cuadros

---

1.	Primer Byte del protocolo . . . . .	21
2.	Segundo Byte del protocolo . . . . .	21



El proyecto plantea el mejoramiento del desarrollo y fabricación del hardware y software de un conjunto de dispositivos electrónicos para obtener la medición de tiempos de reacción de un deportista ante un estímulo visual. Se tiene como objetivo la implementación de estos dispositivos en los entrenamientos de alto rendimiento para atleta que desarrollan sus habilidades en los deportes tales como: Basketball, Box y Softball.

Para lograr obtener la medición del tiempo de reacción de un atleta, se construyó un conjunto de dispositivos inalámbricos. Estos dispositivos al ser activados crean una luz que indica que el tiempo comenzó a correr. El atleta interactúa con el dispositivo para detener el tiempo, ya sea tocándolo o pasando su mano o objeto por encima de este. El dispositivo manda el tiempo hacia un dispositivo principal que obtiene los tiempos y activa los dispositivos de interacción.

El conjunto de dispositivos inalámbricos consiste en desarrollar un módulo esclavo que se comunica por medio de radiofrecuencia a un dispositivo maestro. El trabajo del dispositivo maestro es modular todo el conjunto de dispositivos esclavos para que estos puedan interactuar con el atleta. El dispositivo esclavo contiene dos sensores para la desactivación del mismo. El primero es un sensor ultrasónico que detecta objetos hasta una distancia de 30 cm. El segundo es un sensor capacitivo que funciona cuando el atleta presiona una zona metálica que se encuentra en la parte superior del dispositivo. Se tienen 4 NeoPixels, LEDs RGBs programables, que funcionan para crear la luz o alarma visual que indica el inicio de toma de tiempo. El módulo esclavo envía los tiempos obtenidos al dispositivo maestro y éste los presenta visualmente en un monitor.

Se usa un Feather 32u4 como el controlador de todas las funciones del dispositivo esclavo. El Feather 32u4 trae un módulo de radiofrecuencia incluido, el módulo RFM69HCW, que se comunica con módulo idéntico conectado al dispositivo maestro en una banda de 433 MHz. Este módulo tiene cobertura de 30 metros. Y como dispositivo maestro se utiliza una Raspberry Pi 3 Model B.



The project proposes the improvement of the development and manufacture of the hardware and software of a set of electronic devices to obtain the measurement of reaction times of an athlete before a visual stimulus. The objective is the implementation of these devices in training of high performance for athletes who develop their skills in sports such as: Basketball, Box and Softball.

In order to obtain an athlete's reaction time measurement, a set of wireless devices was built. These devices, when activated, create a light that indicates that the time has begun. The athlete interacts with the device to stop time, either by touching it or by passing his hand or object over it. The device sends the time to a main device that obtains the times and activates the interaction devices.

The set of wireless devices consists in developing a slave module that communicates via radio frequency to a master device. The work of the master device is modular for the entire set of slave devices so that they can interact with the athlete. The slave device contains two sensors to deactivate it. The first is an ultrasonic sensor that detects objects up to a distance of 300 cm. The second is a capacitive sensor that works when the athlete presses a metal zone that is located on the top of the device. There are 4 NeoPixels, LED programmable RGBs, which have to create the light or visual alarm that indicates the start of time taking. The slave module sends the necessary times to the master device and it presents them visually on a monitor.

A Feather 32u4 is used as the controller of all functions of the slave device. The Feather 32u4 comes with an included radio frequency module, the RFM69HCW module, which communicates with an identical module connected to the master device in a 433 MHz band. This module has 30 meter coverage. And as a master device a Raspberry Pi 3 Model B is used.



# CAPÍTULO 1

---

## Introducción

---

El trabajo de graduación trata sobre la medición de tiempos de reacción de un deportista ante un estímulo visual. Este proyecto tiene como fin brindar ayuda a los deportistas del comité olímpico de Guatemala, haciendo que los deportistas mejoren sus técnicas utilizando un dispositivo desarrollado en la Universidad del Valle de Guatemala. El dispositivo que se está desarrollando funciona con componentes electrónicos que realizan un estímulo visual, para que el deportista interactúe con este y así este desactive el dispositivo que está creando dicho estímulo. El objetivo general del trabajo de graduación consiste en mejorar dicho dispositivo. Para mejorarlo se plantearon objetivos específicos, pero esto se mencionará más adelante.

La meta del proyecto es brindar ayuda a todos los deportistas guatemaltecos que quieren mejorar en su disciplina. Es importante dado que en Guatemala existen deportistas que tienen mucho potencial en su disciplina, pero como no se tiene maquinaria o dispositivos que ayuden a estos deportistas, no se logran desarrollar completamente y esto causa que no logren destacar internacionalmente. Además, no solamente se quiere brindar un dispositivo que se pueda utilizar durante los entrenos, si no, que puedan darles datos específicos de sus tiempos durante los entrenamientos. Así, ellos pueden mejorar y plantearse objetivos para su desarrollo como atletas.

La motivación que se tiene detrás del proyecto es poder implementar todo lo aprendido sobre la Ingeniería Electrónica y con esto poder dar una ayuda a los deportista de Guatemala que quieren sobresalir internacionalmente. Y lograr que los conocimientos obtenidos durante el tiempo de aprendizaje sobre la ingeniería Electrónica puedan ayudar a las personas en Guatemala y les sirva como inspiración.



Actualmente la Universidad del Valle de Guatemala cuenta con un dispositivo inalámbrico para la medición de tiempos de reacción de atletas, bajo un estímulo visual [1]. Este funciona como un dispositivo esclavo, que se conecta con un dispositivo maestro. El maestro envía una señal para que se enciendan las luces LEDs del dispositivo esclavo. Al momento que el esclavo detecta que se apagan las luces LEDs, envía una respuesta, esta respuesta es el tiempo en que transcurrió desde que se prendieron los LEDs hasta que se apagaron. El dispositivo se apaga de dos maneras, pasando por encima la mano, pie o cualquier objeto y también se puede apagar si se toca el centro del dispositivo. El dispositivo maestro se comunica con una página Web y dicha página recibe los datos del tiempo de reacción y los muestra en pantalla. Para que los datos del esclavo lleguen al dispositivo maestro se utiliza un protocolo de comunicación de radio frecuencia (RF) entre un microcontrolador, el dispositivo esclavo y el dispositivo maestro. El microcontrolador sirve como mediador entre la Raspberry Pi Zero W, que es el dispositivo maestro, y el dispositivo esclavo. La Raspberry Pi Zero W, funciona como un punto de acceso y como cliente, pero no los dos al mismo tiempo. Este cambia de modo de funcionamiento cada cierto tiempo para que se pueda comunicar con la página Web y con el microcontrolador. El dispositivo esclavo tiene un encapsulado que está hecho con una impresora 3D. Este encapsulado es a prueba de agua, por lo que su estructura es totalmente cerrada. El esclavo tiene una placa la cual se encarga de enviar las respuestas, tiempo en que se tarda el deportista en reaccionar, y encender o apagar los LEDs cuando se le indica.



---

### Justificación

---

Los deportistas tienen como objetivo mejorar su rendimiento para destacar en su disciplina y sobre otros deportistas. Generalmente hay deportistas que se estancan en su rendimiento debido a que sus reacciones no son totalmente rápidas, comparadas a otros competidores. El enfoque del dispositivo creado en la Universidad del Valle es mejorar este tiempo de reacción haciendo que exista una mejor conexión entre el cerebro y el cuerpo, por lo que en cada entrenamiento se trabaja mucho la memoria muscular. Al hacer esto el deportista logra obtener la habilidad de realizar movimientos sin la necesidad de estar pensando. Este dispositivo que mide el tiempo de reacción de un deportista se puede enfocar en muchos deportes. Algunos ejemplos pueden ser: natación, box, béisbol, básquetbol y entre muchos más. La ventaja de estos dispositivos es que se pueden adaptar a las necesidades de los deportistas. Supongamos que un boxeador no logra conectar con exactitud sus golpes y tarda mucho en observar cuando el contrincario baja su guardia.

Como futuro Ingeniero Electrónico, busco obtener la mayor cantidad de conocimientos posibles, no solamente en el área enfocada a celulares, computadoras, consolas de videojuegos, etc. También es importante que tenga un conocimiento en todas las áreas que puede cubrir la Electrónica, por ejemplo los deportes. El dispositivo que se está desarrollando en la Universidad del Valle es de suma importancia para los deportistas, porque ayudará a mejorar exponencialmente el rendimiento de los deportistas. Es por esto que me interesa mucho mejorar el dispositivo. Es importante brindarle apoyo a los deportistas Guatemaltecos y hacer que tengan un reconocimiento, no solamente nacional, sino también internacional. Para un atleta tener una reacción sumamente rápida es de gran importancia y beneficio. Es por eso que me interesa mejorar la comunicación del dispositivo maestro y dispositivo esclavo, porque de esta manera se obtendrán en menor tiempo posible los datos del deportista. Además, los deportistas también necesitan entrenar en distintos lugares con diferentes situaciones climáticas, por ejemplo: Bajo el sol, en lugares fríos, lugares húmedos, etc. Es por esto que quiero mejorar los sensores infrarrojos, para que los deportistas puedan desarrollar sus disciplinas en cualquier lugar sin que sea afectado por el ambiente. Estos aportes funcionarán para hacer que los deportistas de Guatemala puedan destacar. Este trabajo me permitirá poder crear una conexión más grande entre los deportistas y la tecnología,

haciendo que un futuro pueda crear dispositivos que logren dar una ayuda a los deportistas, no solamente en su rendimiento, si no en competencias y eventos importantes tales como torneos internacionales.

### 4.1. Objetivo general

Mejorar el sistema del dispositivo inalámbrico para la medición de tiempo de reacción de un deportista ante un estímulo visual.

### 4.2. Objetivos específicos

- Determinar una alternativa de receptor para que el dispositivo funcione en áreas en donde se encuentre interferencias a los sensores infrarrojos por la luz solar.
- Implementar una nueva alternativa de comunicación entre el dispositivo esclavo y la Raspberry Pi 3.
- Diseñar rutinas de entrenamiento para deportes específicos como, Basquetbol, Box y Softbol.
- Modificar la página web para tener una interfaz más amigable al deportista e implementar las rutinas de entrenamiento



El alcance de este trabajo es desarrollar y aplicar las herramientas de hardware y software para el mejoramiento de un dispositivo esclavo y uno maestro, que trabajan en conjunto para obtener el tiempo de reacción de un estímulo visual. El trabajo incluye la implementación de un módulo de comunicación de radiofrecuencia para la interacción de los dispositivos esclavos y el dispositivo maestro. Además, contiene la selección y desarrollo de dos sensores. El primero, un sensor ultrasónico que detecta objetos a una distancia máxima de 30 centímetros. El segundo, un sensor capacitivo que detecta la presencia de un atleta por medio del contacto de la mano en una zona metálica.

Los dispositivos esclavos tienen un sistema mecánico desarrollado para protección de los dispositivos electrónicos. El trabajo también incluye la construcción de una placa de circuito impreso para los dispositivos esclavos y el desarrollo y aplicación de un estímulo visual generado por el dispositivo esclavo.



## 6.1. Tiempo de reacción

### 6.1.1. ¿Qué es el tiempo de reacción?

El tiempo de reacción o de respuesta es la cantidad de tiempo que transcurre desde que se percibe un estímulo hasta que se genera una respuesta consecuente. El tiempo de reacción depende de tres factores: Percepción, Procesamiento y respuesta.[2]

- Percepción: Consta de ver, oír o sentir el estímulo. Por ejemplo, en una competencia de natación, cuando suena la campana o alarma de salida. Aquí el sonido es el estímulo y nuestro oído percibe este estímulo.
- Procesamiento: Se basa en comprender y concentrarse en el estímulo. Cuando se escucha el sonido de salida, el atleta comprende el estímulo y sabe que en ese momento se debe de salir.
- Respuesta: Es la agilidad motora que se tiene para actuar ante el estímulo. Cuando el atleta ya percibió el estímulo, su respuesta, en el caso de la alarma, es saltar la piscina y comenzar a nadar.

Estos tres factores son importantes para entender bien el tiempo de reacción. Es importante mencionar que también existen tipos de estímulo.

- Simple: Existe una única respuesta a un único estímulo.
- De elección: Existen distintas respuestas a distintos estímulos.
- De selección: Existen distintos estímulos, pero solo se responde a uno de ellos.

El tiempo de reacción es importante en nuestra vida, no solamente en los deportistas. Si se tiene un buen tiempo de reacción nos volvemos ágiles y eficientes al momento de responder ante situaciones o estímulos. Esto puede presentarse en cualquier momento: Cuando manejamos y evitamos tener un accidente, cuando estamos respondiendo una serie de preguntas en un tiempo corto, etc. Y así existen muchas situaciones en donde ponemos en práctica nuestro tiempo de reacción.

### **6.1.2. ¿Cómo se mide el tiempo de reacción?**

Medir el tiempo de reacción es importante, porque con esto tenemos un dato que nos indica qué tan rápido estamos procesando los estímulos y que tan rápido respondemos a ellos. Para medir el tiempo de reacción existen una infinidad de pruebas. Hay pruebas con estímulos visuales, pruebas con estímulos auditivos, pruebas con estímulos sensitivos, etc. Una forma de medir el tiempo de reacción ante un estímulo visual, es tomar el tiempo desde que comienza el estímulo visual, encender una luz, por ejemplo, hasta que la persona realiza una acción ante dicho estímulo, apagar la luz que se encendió.

### **6.1.3. ¿Cómo mejorar el tiempo de reacción?**

Para mejorar el tiempo de reacción, existen ejercicios que fortalecen nuestro músculo más importante, el cerebro. Al entrenar nuestro cerebro, podemos mejorar la conexión del cerebro con los músculos del cuerpo, para obtener respuestas rápidas y eficientes. Además, cuando entrenamos mucho nuestro cerebro, comenzamos a obtener experiencia, haciendo que se pueda hacer una predicción o encontrar un patrón. Esto hace que se pueda reaccionar antes que comience o detecte un estímulo, haciendo que el tiempo de reacción disminuya.

## **6.2. Radiofrecuencia**

### **6.2.1. ¿Qué es radiofrecuencia?**

Se le dice radiofrecuencia a la parte con menos energía de un espectro electromagnético. El espectro electromagnético se crea cuando se genera una corriente a través de un conductor. Este espectro se transmite por medio de ondas y se puede recibir por medio de una antena, que sería nuestro dispositivo receptor. [3]

### **6.2.2. Aplicación en las comunicaciones**

La radiofrecuencia se aplica en todos los medios de comunicación inalámbricos. Un ejemplo básico es la señal de la radio de nuestro automóvil o también la comunicación entre los Walkie Talkies. Aunque, no solamente se utiliza en estos casos. La comunicación entre nuestros dispositivos inteligentes como los teléfonos móviles, consolas de videojuegos, etc.

### **6.2.3. Radiofrecuencia RFID**

La radiofrecuencia RFID, Identificación por Radiofrecuencia, se refiere a las frecuencias eléctricas que mandan información a un receptor. Un ejemplo actual es el pago con las tarjetas de crédito. Ahora podemos pagar nuestros gastos con la tarjeta de crédito sin la necesidad de insertar la tarjeta en el dispositivo de cobro, simplemente por medio de RFID se acerca la tarjeta al dispositivo sin tocarlo y este realiza el cobro.

## **6.3. Protocolo de comunicación**

### **6.3.1. ¿Qué es un protocolo?**

Un protocolo es un método que consiste en un conjunto de reglas y procedimientos que deben seguirse estrictamente para el envío y la recepción de datos por medio de una red [4]. Existen protocolos que se especializan en la transferencia de archivos o en la detectar errores y ver el estado de transmisión.

### **6.3.2. Protocolos orientados a conexión**

Estos protocolos controlan la transmisión de datos mientras existe una comunicación entre dos dispositivos. En estos protocolos un dispositivo debe de ser el emisor y otro el receptor. El emisor solamente envía datos y el receptor se encarga de recibirlos y mandar un mensaje de respuesta indicando que se recibieron todos los datos.

### **6.3.3. Implementación de un protocolo**

Como se sabe, un protocolo solo se preocupa de cómo se deben de comunicar y que series de pasos deben de seguir los dispositivos de comunicación para transmitir datos entre ellos. Por lo que implementar un protocolo solo consta de definir el conjunto de reglas y la secuencia de datos que van intercambiar.

## **6.4. Comunicación SPI**

La interfaz serial periférica SPI, por sus siglas en inglés Serial Peripheral Interface, es un protocolo de comunicación serial síncrono. Este protocolo permite la comunicación de un sistema con uno o varios dispositivos a través de un bus de comunicación compartido [5]. La comunicación SPI es utilizada generalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

La transmisión y sincronización de datos se realiza por medio de 4 señales:

- SCLK (Clock): Es el pulso que marca la sincronía entre los dispositivos. Con cada pulso de este reloj, se puede leer o enviar un bit.
- MOSI (Master Output Slave Input): Es la salida de datos del maestro y entrada de datos del esclavo.
- MISO (Master Input Slave Output): Es la entrada de datos del maestro y la salida de datos del esclavo.
- SS/Select: Es para seleccionar un esclavo o para activar al esclavo.

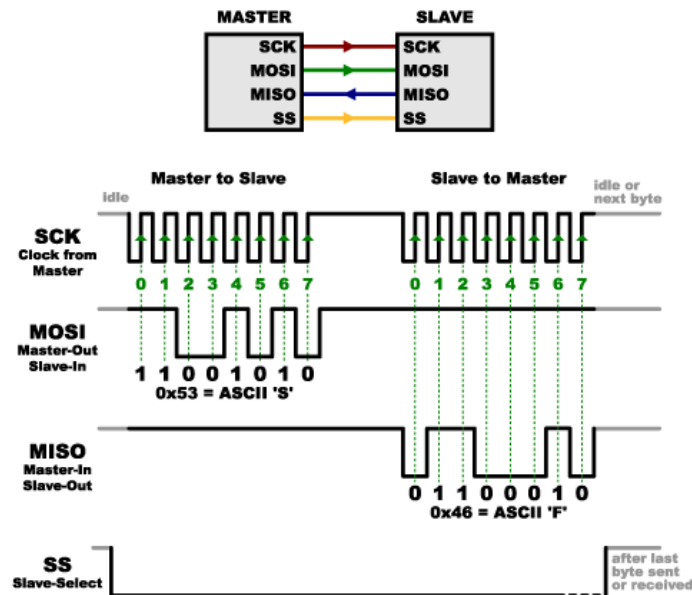


Figura 1: Diagrama de funcionamiento de la comunicación SPI [6]

El SPI funciona con comunicación Full Duplex, es decir, que puede mantener una comunicación bidireccional, enviando y recibiendo datos de forma simultánea. Existen 4 modos de funcionamiento para la comunicación entre el maestro y el esclavo.

- Modo 1: Modo en donde el reloj se mantiene en un 0 lógico y la información se envía en cada flanco de reloj positivo.
- Modo 2: Modo en donde el reloj se mantiene en un 0 lógico y la información se envía en cada flanco de reloj negativo.
- Modo 3: Modo en donde el reloj se mantiene en un 1 lógico y la información se envía en cada flanco de reloj positivo.
- Modo 4: Modo en donde el reloj se mantiene en un 1 lógico y la información se envía en cada flanco de reloj negativo.

## 6.5. Modulación por ancho de pulso

La modulación por ancho de pulso PWM, por sus siglas en inglés Pulse-width modulation, es un método que modifica el ciclo de trabajo de una señal periódica digital, sinusoidal o cuadrada, para producir un aproximado de una señal analógica. Este método es comúnmente usado para reducir o controlar la intensidad de un LED o para variar la velocidad de un motor. [7]

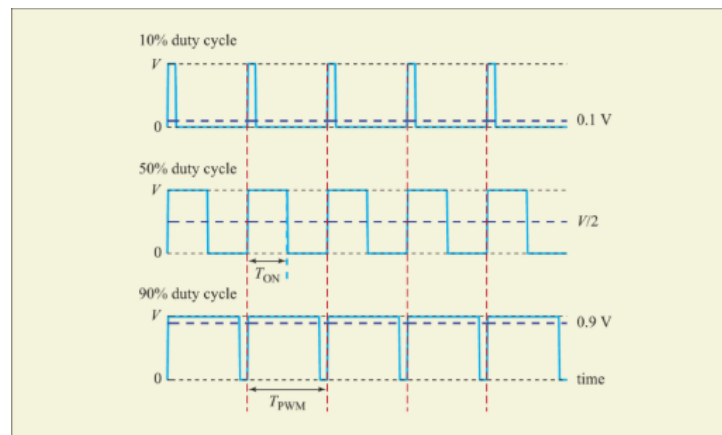


Figura 2: PWM con distintos ciclos de trabajo [7]

## 6.6. Servidor web

### 6.6.1. ¿Qué es un servidor web?

Un servidor web, es un software que utiliza un protocolo para generar una conexión entre hardware y software [8]. Esta conexión puede ser unidireccional o bidireccional. Existen servidores estáticos y dinámicos. Los servidores estáticos, son servidores que no contienen una base de datos y los servidores dinámicos, contienen una o varias bases de datos en donde almacenan información del cliente o información que puede ser utilizada en las páginas web.

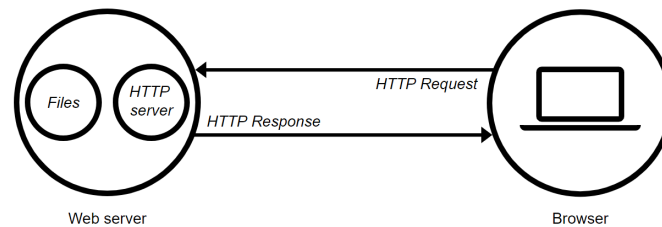


Figura 3: Servidor web [8]



---

## Selección de equipo electrónico

---

La selección del equipo electrónico es importante para el desarrollo del dispositivo esclavo y para el desarrollo del dispositivo maestro. Como se sabe la selección de equipo electrónico dará una orientación sobre el desarrollo del software del conjunto de dispositivos para la medición del tiempo de reacción y para el desarrollo de software del dispositivo maestro. Además, esto aportará una idea para el diseño mecánico del conjunto de dispositivos y la fabricación del circuito impreso.

### 7.1. Sensor ultrasónico JSN-SR04T-2.0

Es un sensor ultrasónico que puede detectar distancias desde 20 cm a 600 cm. Su voltaje de operación es de 3.0V a 5.0V. Su temperatura de operación es entre los 20°C a 70°C. Tiene 4 pines: Trigger, Echo, Power y GND. Los pines Echo y Trigger funcionan para la salida y entrada de las ondas de sonido que genera el sensor (Figura 4). Se seleccionó este sensor ya que es efectivo para la medición de distancia y transmite datos totalmente acertados. Además, puede utilizarse en cualquier clima, bajo el sol, bajo la lluvia, etc. Y sigue teniendo un funcionamiento efectivo. Este sensor tiene muchas funcionalidades, se utiliza generalmente en la parte trasera de los carros como indicador de distancia de un objeto.

### 7.2. Módulo de radiofrecuencia RFM69HC 433MHz

Este módulo de radiofrecuencia es un transceptor, esto nos indica que este componente electrónico es capaz de recibir y enviar información en una frecuencia de 433 MHz. El componente electrónico RFM69HCW mantiene una comunicación SPI con el microcontrolador al que se encuentre conectado y tiene una capacidad de potencia de 10mW. Cuando



Figura 4: Sensor ultrasónico JSN-SR04T-2.0

este dispositivo está recibiendo información de otro transceptor, demanda aproximadamente 30mA de corriente y cuando el dispositivo se encuentra enviando información, demanda entre 50mA a 150 mA de corriente. Puede comunicarse con otros transceptores en un rango de 500 metros aproximadamente, dependiendo de los obstáculos que se encuentren en el medio de comunicación (paredes, puertas, carros, etc). Aunque, si se conecta una antena amplificadora de señal, el RFM69HCW puede llegar a transmitir/recibir información en un rango de 5 kilómetros [9]. Es por estas razones por la que se decidió utilizar este dispositivo como medio de comunicación entre el conjunto de dispositivos esclavos y el dispositivo maestro.



Figura 5: RFM69HCW [9]

### 7.3. Feather 32u4 RFM69HCW

El Feather 32u4 RFM69HCW es un microcontrolador que tiene ya incluido un módulo de radiofrecuencia RFM69HCW que trabaja a 433 MHz, Figura 6 . Este microcontrolador tiene incluido una entrada micro-USB para ser programado y una entrada para conectar una batería de litio. Tiene un procesador ATmega32u4 que funciona a 8MHz con un voltaje de 3.3V. El feather cuenta con 10 entradas analógicas y 20 pines digitales. Adicionalmente, se puede conectar al microcontrolador una antena amplificadora de señal para obtener comunicación en una mayor rango de distancia. El Feather tiene la capacidad de cargar la batería que se esté conectando al dispositivo, siempre y cuando el microcontrolador se encuentre conectado a un computador o un tomacorriente que le brinde energía. El Feather 323u4 RFM69HCW será el cerebro del dispositivo esclavo.

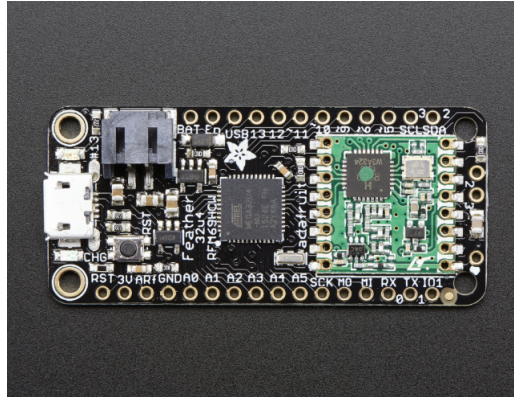


Figura 6: Feather 32u4 RFM69HCW [10]

#### 7.4. NeoPixel Diffused 5mm

Los NeoPixels, Figura 7, son LEDs RGBs programables que modifican el control de iluminación utilizando un PWM para tener colores de 24 bits. Su voltaje de funcionamiento es de 3.3V - 5V. Su estructura es de 4 pines. El primer pin es para la entrada de datos de la señal PWM, el segundo pin es para la fuente de voltaje, el tercer pin para la conexión a tierra y por último, el 4 pin funciona como salida de datos del PWM. La idea del último pin es que se puedan conectar muchos NeoPixels en serie y por medio de programación, se pueda programar con un solo pulso PWM una cantidad infinita de neopixels, haciendo que cada NeoPixel pueda ser utilizado para iluminar un color diferente en una sola línea de código.

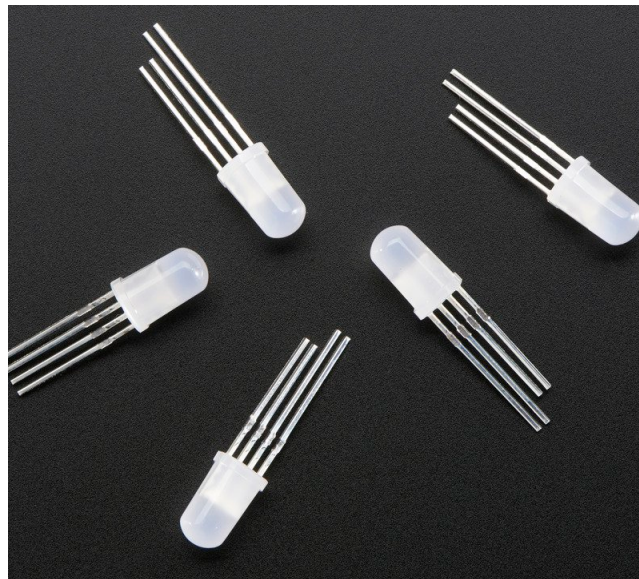


Figura 7: NeoPixel Diffused 5mm [11]



---

## Definición del protocolo de comunicación

---

La definición del protocolo de comunicación es importante para comunicar la información del dispositivo maestro al conjunto de dispositivos esclavos. El protocolo definido funcionara para enviar los datos necesarios para la configuración de los esclavos y estos comiencen a funcionar según la información recibida. El protocolo se definió en dos byte: El primer Byte se muestra en el Cuadro 1 y el segundo Byte se muestra en el Cuadro 2

### 8.1. Significado de cada Byte

bit 1	bit2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8
ID Dispositivo	ID Dispositivo	ID Dispositivo	Color	Color	Color	Test	ON/OFF

Cuadro 1: Primer Byte del protocolo de comunicación

bit 1	bit2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8
Deporte	Deporte	Entrenamiento	Entrenamiento	Entrenamiento	Entrenamiento	Sensor	N/A

Cuadro 2: Segundo Byte del protocolo de comunicación

El significado de cada bit del primer byte y segundo byte se describen a continuación:

#### 8.1.1. ID Dipositivo

Este está conformado por 3 bits, los primeros 3 bits del primer byte, y representa que dispositivo es al que se está comunicando. Se utilizaron 3 bits porque así se pueden te-

ner 7 dispositivos esclavos que esten funcionando al mismo tiempo. Las combinaciones y su significado son las siguientes:

- 000: Esta combinación no se utiliza para ningún dispositivo esclavo.
- 001: Esta combinación representa al dispositivo esclavo número 1.
- 010: Esta combinación representa al dispositivo esclavo número 2.
- 011: Esta combinación representa al dispositivo esclavo número 3.
- 100: Esta combinación representa al dispositivo esclavo número 4.
- 101: Esta combinación representa al dispositivo esclavo número 5.
- 110: Esta combinación representa al dispositivo esclavo número 6.
- 111: Esta combinación representa al dispositivo esclavo número 7.

### 8.1.2. Color

Al igual que el anterior, este también está conformado por 3 bits del primer byte y funciona para seleccionar el color de los Neopixels del dispositivo esclavo. Los bits que se usan son del bit número 4 al bit número 6. El funcionamiento aquí es simple, cada bit representa un color. Rojo es el bit 4, Verde es el bit 5 y azul es el bit 6. Con estos 3 bits se pueden hacer 8 combinaciones de colores. Las combinaciones de colores son las siguientes:

- 000: Esta combinación representa al color negro, pero no se utiliza.
- 001: Esta combinación representa al color azul.
- 010: Esta combinación representa al color verde.
- 011: Esta combinación representa al color aqua.
- 100: Esta combinación representa al color rojo.
- 101: Esta combinación representa al color fucsia.
- 110: Esta combinación representa al color amarillo.
- 111: Esta combinación representa al color blanco.

### 8.1.3. Test

El test funciona para comprobar que el esclavo funciona y no tiene ningún problema. Este solo utiliza 1 bit del primer byte, el bit número 7. Si el bit es 1, nos indica que se hará un test para ver que todo funciona y si se encuentra en 0, no se hará ninguna prueba.

#### **8.1.4. ON/OFF**

Este se representa por el bit 8 del primer byte y es el indicador para decir que si el dispositivo se encontrará activado o apagado. Si el bit está en 1, nos indica que está activado el dispositivo y si se encuentra en 0 el dispositivo está apagado.

#### **8.1.5. Deporte**

Los deportes están representados por los primeros 2 bits del segundo byte, los bits número 1 y 2. La combinación de los bits nos permiten tener 4 deportes diferentes, es por eso que se escogieron solamente dos bits. En este caso solamente se están usando 3 combinaciones, dado que se está enfocando en solamente 3 deportes. .

- 00: Esta combinación no se utiliza para ningún deporte.
- 01: Esta combinación representa al deporte Basquetball.
- 10: Esta combinación representa al deporte Box.
- 11: Esta combinación representa al deporte Softball.

#### **8.1.6. Entrenamiento**

Los entrenamientos están representados por 4 bits del segundo byte, desde el bit número 3 al bit número 6. Con estos bits se pueden representar 16 entrenos en total. Por lo que cada deporte tiene 16 entrenos distintos que puede realizar un atleta.

#### **8.1.7. Sensor**

Para los sensores se utilizó un solo bit, el bit 7 del segundo byte. En este bit, el número 1 representa que se utilizará el sensor capacitivo para el entreno y el número 0 representa que se utilizará el sensor ultrasónico.

#### **8.1.8. N/A**

Este último bit del segundo byte no se utiliza en el protocolo.



---

## Comunicación por radiofrecuencia entre el dispositivo esclavo y maestro

---

La comunicación por radiofrecuencia entre el dispositivo maestro y el conjunto de dispositivos esclavos permite la transferencia de datos entre ellos. La información que se transmite a cada uno de los dispositivos esclavos, se realiza por medio de Software. En la programación del Software se encuentra la información que se transmite y la recepción de los datos de cada uno de los esclavos.

### 9.1. Configuración del módulo RFM69HCW en la Raspberry Pi 3

Para la comunicación del dispositivo maestro, Raspberry Pi 3, se conectó el dispositivo RFM69HCW de manera física. El maestro se programó en Python 3, este programa ya viene instalado en el sistema operativo de la Raspberry Pi 3. La librería Open Source: `rfm69`, permite la comunicación entre dispositivos de radiofrecuencia RFM69HCW que se encuentren en el mismo canal de comunicación y la misma frecuencia. Esta librería puede utilizarse en Python 3 y permite la programación del módulo mencionado con anterioridad.

Para el desarrollo del Software es importante realizar unas configuraciones a la Raspberry PI 3. Esto se debe de realizar para lograr tener una comunicación SPI con el módulo de radiofrecuencia y la Raspberry Pi 3.

1. Primero, se debe abrir la terminal de la Raspberry Pi 3 y se debe de ingresar el comando: `sudo raspi-config`.
2. Luego se desplegará una pantalla, Figura 8, en donde estarán las configuraciones de la Raspberry Pi.

3. Por último, se selecciona “Interface options” y se habilitan las opciones de “SPI” y “Remote GPIO”

Con estas configuraciones se puede comenzar a desarrollar el programa en Python 3 para el uso del RFM69HCW.

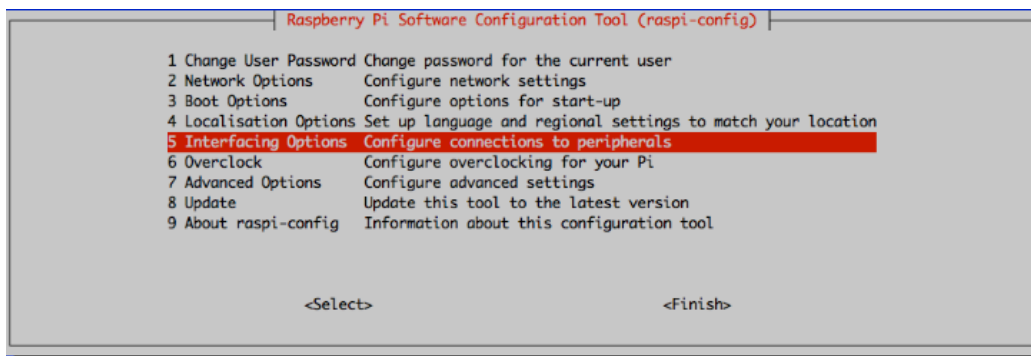


Figura 8: Software de configuración de la Raspberry Pi 3

## 9.2. Programación dispositivo maestro

La programación del dispositivo maestro se basa en la comunicación con el dispositivo esclavo y en los entrenos que realizará cada deportista. La programación esta compuesta de dos archivos: “FuncionesVariables.py” y “RFM69Rasp.py”. El primero contiene todas las funciones para la comunicación y entrenamientos. El segundo tiene la función de un menú principal para los deportistas o entrenadores y estos puedan elegir qué deporte practicar y que entrenamientos realizar.

### 9.2.1. Programa FuncionesVariables.py

Este programa contiene todas las funciones de comunicación por RF y las funciones para los entrenos. Las librerías que se utilizaron fueron: RFM69 para la comunicación y datetime, time para los delays en el programa.

Las funciones que se utilizaron en el programa y su explicación son las siguientes:

- IniciarRadio(nodoM,nodoE): La función IniciarRadio(nodoM,nodoE), tiene 2 variables de entrada, nodoM y nodoE. nodoM es una variable para indicar que dispositivo es el que envía mensajes y nodoE es para indicar en qué red se estará comunicando. La función solamente inicia la comunicación entre los dispositivos.
- transmitir(nodo,mensaje,radio): La función contiene 3 parámetros de entrada para su funcionamiento:
  1. Nodo: Indica a qué dispositivo se envían los datos.
  2. Mensaje: Indica que información se está enviando al dispositivo esclavo.

3. Radio: Indica en qué canal se está comunicando el maestro con el esclavo.

La función transmite datos según los 3 parámetros que se están indicando.

- recibir(radio): Esta función tiene como parámetro radio, que indica en qué canal se está comunicando el maestro con el esclavo. La función, con el uso del parámetro radio, mantiene al módulo RFM69HCW esperando a que reciba información del dispositivo esclavo que se esta comunicando por medio del canal indicado por el parámetro.
- BBe1() BBe2(veces) BBe3() : Estas tres funciones, son los entrenos para el deporte de Basquetball. Cada uno es distinto del otro, pero tienen características similares en cuanto a los sensores que utilizan. El primer entreno, BBe1(), utiliza solo 2 dispositivos esclavos y funciona con el sensor capacitivo. El segundo entreno, BBe2(veces), utiliza 3 dispositivos y funciona igual que el anterior, con el sensor capacitivo. En esta rutina se puede escoger cuantas veces se quiere realizar el entreno. Por último, BBe3(), funciona con 3 dispositivos y combina el uso de los dos sensores, sensor capacitivo y ultrasónico, para tener una dinámica distinta al entrenar.
- entrenamientosBB(entreno): Esta función realiza alguna de las tres funciones anteriores para el deporte Basquetball, según el entreno que se escoja realizar.
- Be1() Be2() Be3(): Estas tres funciones, son los entrenos para el deporte de Box. En esta ocasión, los entrenos solo cuentan con el uso del sensor ultrasónico. El entreno Be1(), es el primero y funciona con 3 dispositivos esclavos. El entreno Be2(), funciona con 4 dispositivos y estos por un tiempo iluminan de un color y en otro tiempo iluminan con otro color. Por último, el entreno Be3() es un combinación de las dos anteriores. El propósito de este entreno es que cuando ya domine con excelencia los dos entrenos anteriores, este sea una combinación y sea de mayor dificultad.
- entrenamientosB(entreno): Esta función realiza alguna de las tres funciones anteriores para el deporte Box, según el entreno que se escoja realizar.
- SBe1() SBe2() SBe3(): Al igual que las funciones para los entrenos de Box, los entrenos de Softball solo utilizan el sensor ultrasónico. El primer entreno, SBe1(), solo utiliza un dispositivo esclavo y este se activa cada cierto tiempo 10 veces. Para el segundo entreno, SBe2(), se utiliza 1 dispositivo esclavo y este ilumina con 2 colores distintos. Se hizo esto para que el entreno no sea solamente para un deportista, si no para entrenar en parejas. Por último, el entreno SBe3(), utiliza 2 dispositivos y este maneja dos colores distintos, para que el entrenamiento sea en parejas y más dinámico.
- entrenamientosSB(entreno): Esta función realiza alguna de las tres funciones anteriores para el deporte Softball, según el entreno que se escoja realizar.

### 9.2.2. Programa RFM69Rasp.py

En este programa se tiene el menú de opciones para la selección del deporte y entrenamiento que se llevará a cabo. Además, se importó el archivo: FuncionesVariables.py para tener todas las funciones que se definieron dentro de ella. El menú se ve de la siguiente manera, Figura 9

```
>>> %Run RFM69Rasp.py
Bienvenido al programa de medicion de tiempo de reaccion de un deportista
Por favor, ingrese los datos que se le piden para comenzar su entreno
Que deporte entrenara?
1. Basquetball 2. Box 3. Softbal
Ingrese el numero del deporte que entrenara
1
Que entrenamiento quiere hacer?
1. Entreno numero 1, 2. Entreno numero 2 3. Entreno numero 3
ingrese el numero del entrenamiento que realizara
```

Figura 9: Menú de entrenos del dispositivo maestro

### 9.3. Programación dispositivo esclavo

Para la programación del dispositivo esclavo se utilizó la plataforma Arduino IDE. La programación es muy básica. El dispositivo esclavo está programado para recibir información del dispositivo maestro. Cuando el dispositivo esclavo es desactivado, este transmite el tiempo de reacción del atleta al maestro. Las librerías para el desarrollo del Software son Open Source. Cada función que se creo, se explica más adelante.

Librerías para la recepción, transmisión y procesamiento de datos:

1. RFM69.h
2. RFM69-ATC.h
3. ArduinoJson.h”

Librerías para configuración y funcionamiento de los sensores y NeoPixels:

1. Adafruit-NeoPixel.h
2. Ultrasonic.h
3. CapacitiveSensor.h

Las funciones que se utilizaron y sus explicaciones son las siguiente:

- `Resetear()`: En esta función se restablecen las variables a los valores inicializados en el programa.
- `SetVariablesDatos()`: Esta función asigna los valores de cada variable dentro del programa. Los valores que se asignan dependen de la información que se obtiene del protocolo de comunicación. Según los valores de cada variable, se establece una configuración única para el dispositivo esclavo y establece también el funcionamiento de este.

- `escuchar()`: En esta función es donde se recibe el mensaje del dispositivo maestro. Al momento de ser recibido el mensaje del maestro, la información se almacena en una variable. Se utiliza la función `SetVariablesDatos()` para la asignación de datos. Luego de la lectura de información, se pasa a la verificación para saber si el dispositivo se encontrará activado o apagado. Si el caso es que se encuentra apagado, el dispositivo se quedará esperando otro mensaje del dispositivo maestro. Si es el caso contrario, el dispositivo se activará, este empezará a tomar el tiempo y activará los NeoPixels. Al hacer esto, estará esperando que el dispositivo se desactive por medio de alguno de los dos sensores, según su configuración, y detendrá el tiempo. Por último, envía el tiempo de reacción del dispositivo utilizando la función `apagar()`.
- `apagar()`: `Apagar()` funciona para indicar que el dispositivo fue desactivado. Entonces, se envía un mensaje al dispositivo maestros con el tiempo de reacción del deportista. Por último, se reinicia una variable para que el dispositivo vuelva a la función `escuchar()` y espere a que le envíen un mensaje.
- `ultraSonico()`: Esta función hace que el sensor ultrasónico se mantenga leyendo distancias continuamente. Dentro de la función se tiene establecido un rango lectura. Si la distancia que se está leyendo continuamente, se mantiene dentro de este rango. Quiere decir que se detectó un objeto y por lo tanto el dispositivo se debe de apagar.
- `capacitivo()`: Esta función es igual de simple que, `ultraSonico()`. Al presionar el metal del dispositivo, se envía una señal al Feather 32u4 y este nos indica que se apagó el dispositivo. La variable que indica que el dispositivo se apagó, se activa.



---

## Diseño electrónico del dispositivo esclavo

---

El diseño electrónico del dispositivo esclavo son dos placas de circuito impreso. La placa principal que contiene la conexión del Feather con el módulo de ultrasónico y la placa secundaria que con tiene la conexión de los LEDs y los sensores. Las dos placas están conectadas entre ellas, por medio de Headers o módulos de conexión. El diseño de las placas de los circuitos impresos se trabajaron en la plataforma Altium Designer 17.1.

### 10.1. Librería Feather32u4

Altium Designer 17.1 exige que para la creación de una placa de circuito impreso, se deben de tener todos los componentes de forma digital dentro de la plataforma. El Feather 32u4 RFM69HCW no tiene un diseño dentro de Altium Designer 17.1. Por lo tanto se realizó una librería para modelar el componente electrónico dentro de la plataforma de diseño. La Figura 10 muestra el esquemático de la librería del Feather 32u4 RFM69HCW y la Figura 11 muestra el diseño del circuito impreso del componente electrónico.

### 10.2. Esquemático utilizado para el diseño de la placa principal

El esquemático se puede observar en la Figura 12. Este esquemático muestra las conexiones que se realizaron para el funcionamiento del dispositivo. El esquemático está conformado por el módulo del sensor ultrasónico, Headers de 2x1, que funcionan para la conexión con la placa secundaria para el sensor capacitivo y Headers de 3x1 para la conexión de los Neopixels en la placa secundaria.

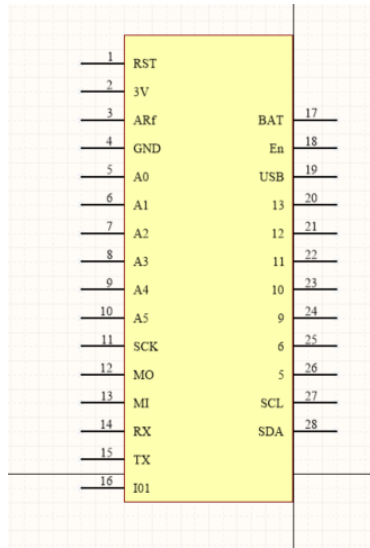


Figura 10: Esquemático del componente electrónico Feather32u4 RFM69HCW

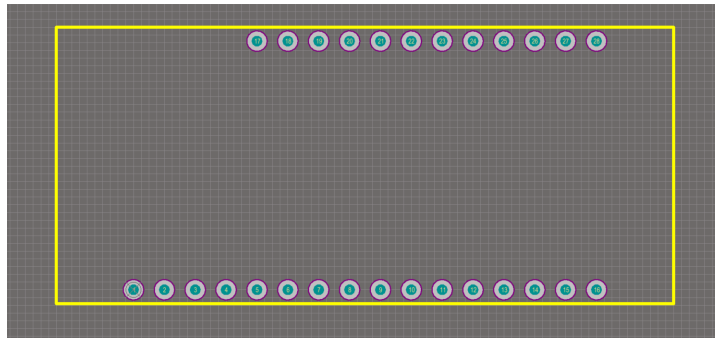


Figura 11: Diseño del circuito impreso del componente electrónico Feather32u4 RFM69HCW

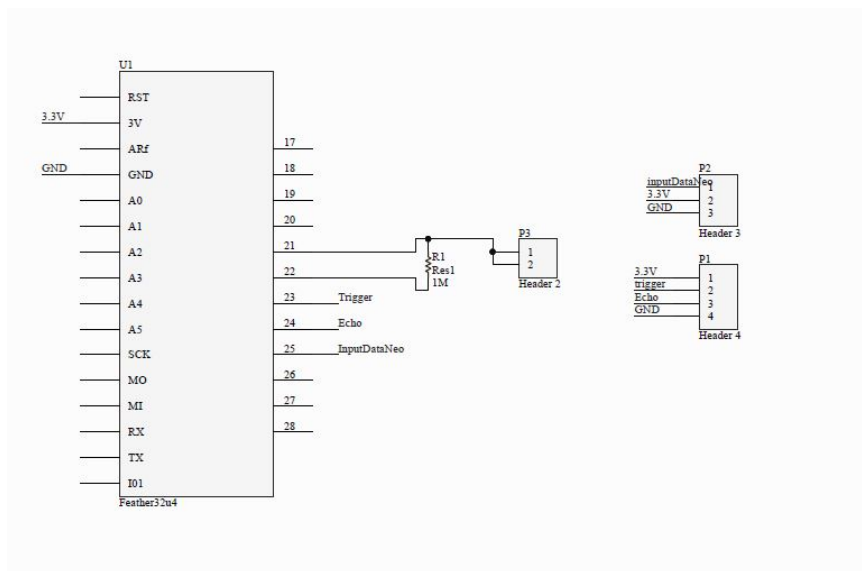


Figura 12: Esquemático dispositivo esclavo placa principal

### 10.3. Diseño del PCB para la placa principal

La placa principal utilizada, Figura 13, se hizo con un diseño circular, para tener una placa más compacta y se acoplara al diseño mecánico del dispositivo. La placa es muy pequeña, tiene radio de 4.4 cm, en donde se encuentran todos los componentes explicados con anterioridad. Esta placa tiene conexiones solo en una capa, la capa de abajo, y tiene un orificio en donde pasa el cable para la conexión de la batería al Feather 32u4 y 4 orificios para que se pueda sujetar con una de las piezas del diseño mecánico. Adicional a los agujeros, tiene otras perforaciones con forma de media luna, para que también se pueda sujetar la placa con otra de las piezas del diseño mecánico.

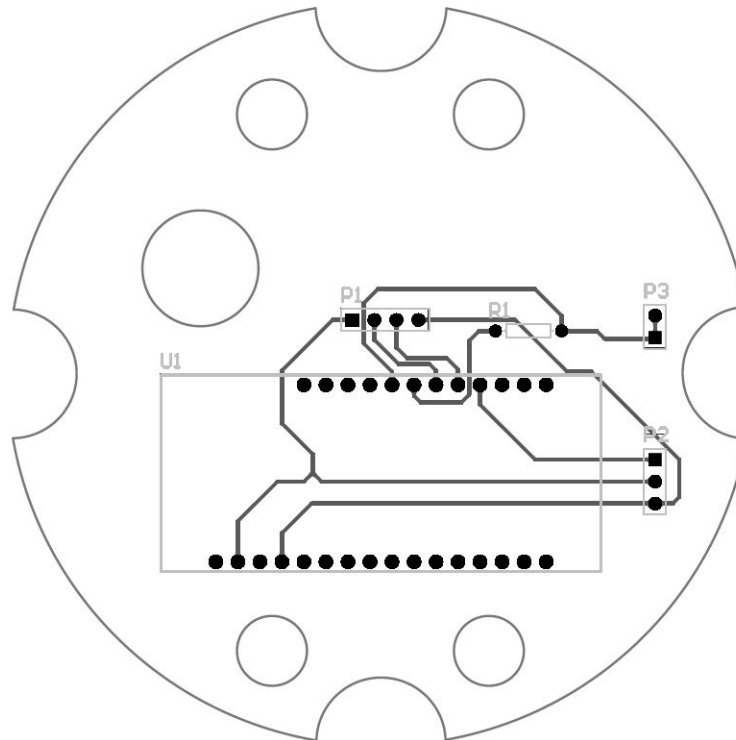


Figura 13: Diseño PCB placa principal

### 10.4. Esquemático utilizado para el diseño de la placa secundaria

El esquemático se puede observar en la Figura 14. Este esquemático muestra las conexiones que se realizaron para el funcionamiento de los NeoPixels y sensores. El esquemático esta conformado por los 4 NeoPixels utilizados, 1 Header de 3x1 para la conexión con la placa principal y 3 Headers de 2x1 que funcionan para conexión de la placa principal con los sensores.

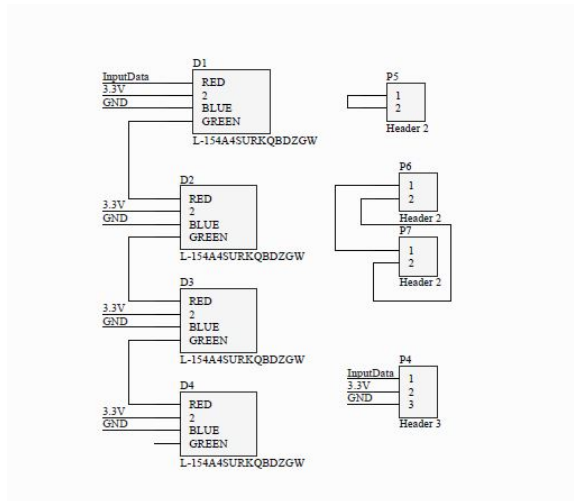


Figura 14: Esquemático dispositivo esclavo placa secundaria

## 10.5. Diseño del PCB para la placa secundaria

Al igual que la placa principal, la placa secundaria se hizo con un diseño circular, Figura 15, para tener una placa más compacta y se acoplara al diseño mecánico del dispositivo. La placa es muy pequeña, tiene radio de 4.3 cm, en donde se encuentran todos los componentes explicados con anterioridad. Esta placa tiene conexiones solo en la capa de abajo y tiene 4 agujeros que funcionan para poder sujetar la placa con una de las piezas del diseño mecánico y tiene 4 perforaciones en forma de media luna para poder sujetarse con los tornillos que se utilizaran en el ensamblaje final. Además de los 4 agujeros, tiene un agujero en medio que funciona para que atravesase el sensor ultrasónico.

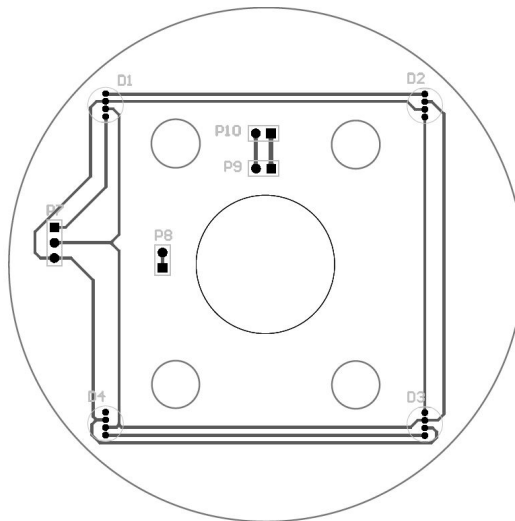


Figura 15: Diseño PCB placa secundaria

---

## Diseño mecánico del dispositivo esclavo

---

El diseño mecánico del dispositivo se definió desde un principio para que tuviera una forma circular y así tener un dispositivo más compacto e interactivo. El diseño mecánico se dividió en varias piezas. Para la realización de este diseño, se utilizó la plataforma AutoDesk Inventor Professional 2017. Para la fabricación del diseño mecánico se utilizó una cortadora láser. Como material principal se utilizó una plancha de grosor de 3.2 mm. Y el diseño está hecho por capas, es decir que el diseño está conformado por varias piezas iguales. Adicional a las piezas modeladas en Inventor, también se modelaron las dos placas para tener una idea de cómo iría quedando el diseño final, agregando a las placas los modelos 3D de los componentes. En la Figura 16, se puede observar como se ve el modelado en la plataforma de Inventor.

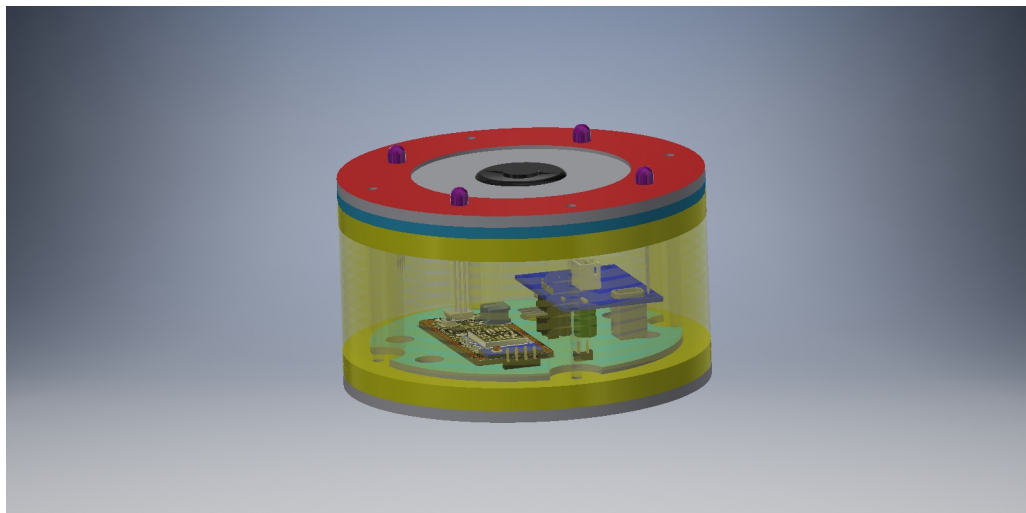


Figura 16: Modelado 3D completo del diseño mecánico

### 11.1. Pieza inferior del diseño mecánico del dispositivo esclavo

El diseño de la pieza inferior del diseño mecánico, Figura 17, es muy básica. Este modelado es un simple disco. Que tiene como propósito guardar la batería que dará energía al Feather 32u4 RFM69HCW. El diseño de esta pieza tiene 4 agujeros en la orilla. Los agujeros funcionan para que se pueda utilizar un tornillo y se ensamble esta pieza al diseño completo.

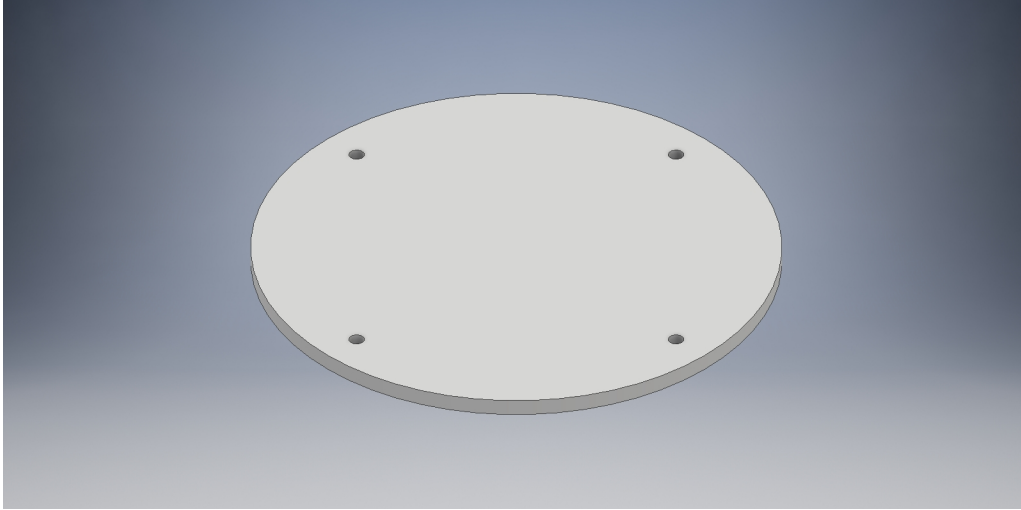


Figura 17: Pieza inferior del diseño mecánico del dispositivo esclavo

### 11.2. Pieza soporte para la placa principal

Esta pieza, Figura 18, se diseñó para que funcione como soporte y sujetador de la placa principal, para que al momento que se haga el ensamblaje junto a la placa principal, esta no se esté moviendo de su posición y que no cause ningún daño a la placa o componentes en ella.

Este soporte tiene forma circular y tiene una especie de cruz que funciona como el soporte de la placa y que se pueda atar la placa en la pieza. Además, tiene sus 4 agujeros para que se pueda ensamblar con los tornillos y las demás piezas.

### 11.3. Pieza intermedia del diseño mecánico del dispositivo esclavo

La pieza intermedia del diseño mecánico, Figura 19, es un disco hueco que tiene agujeros en su borde. Se trabajó de esta manera para poder aprovechar lo más que se puede el espacio y así poder cablear las placas por el interior del dispositivo. Esta pieza es la base de todo el diseño y los 4 agujeros que tiene, como todas las piezas en general, para poder atornillar la pieza y unirla al ensamblaje final. Se utilizaron 14 de estas piezas para el diseño mecánico.

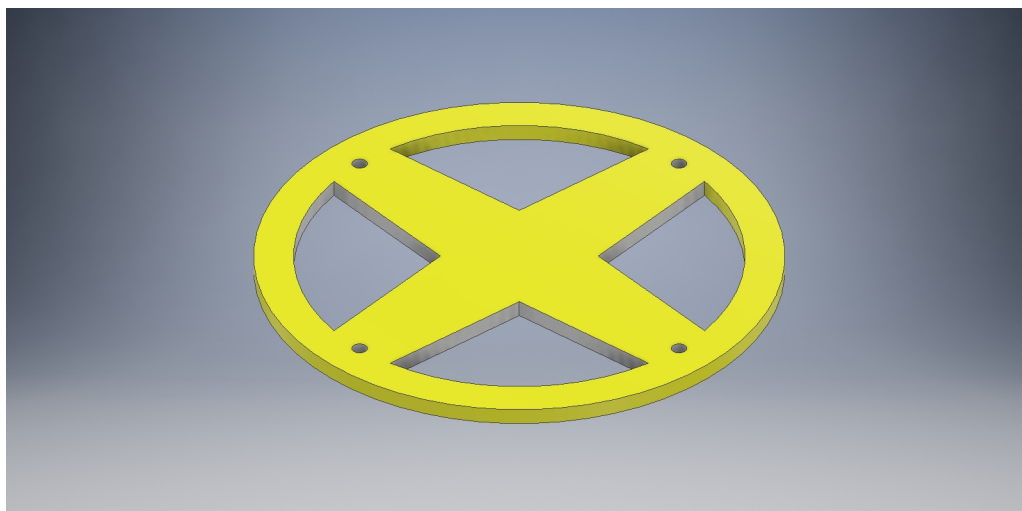


Figura 18: Soporte inferior para la placa principal del diseño mecánico del dispositivo esclavo

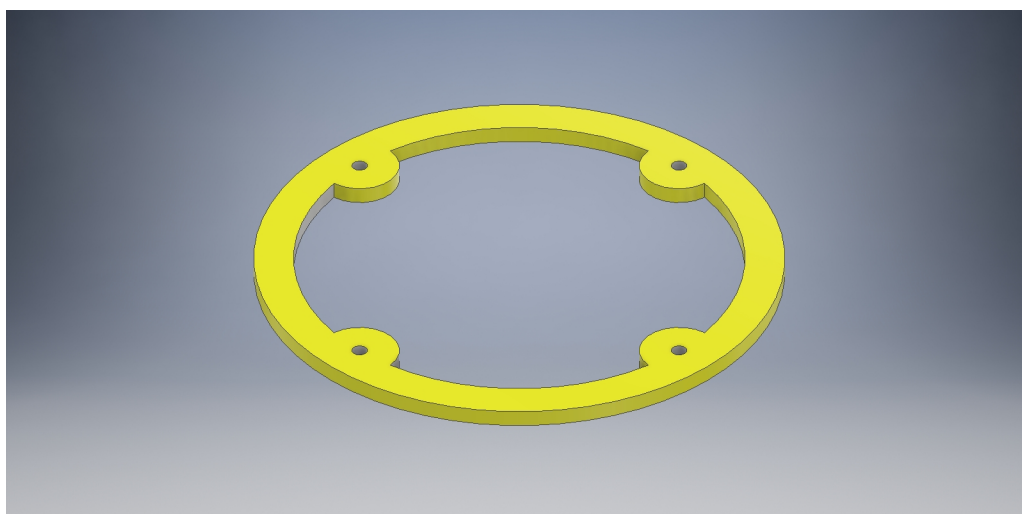


Figura 19: Parte intermedia del diseño mecánico del dispositivo esclavo

#### 11.4. Pieza soporte para la placa secundaria

Al igual que la placa principal, a la placa secundaria se le diseño un soporte, Figura 20, para poder sostenerla y que no se moviera en el diseño mecánico. Debido al tamaño de la placa, que es más pequeña de radio que el diseño mecánico, también se hizo un modelo de una rueda que hace que se ajuste la placa secundaria al diseño mecánico y se pueda sostener también con los tornillos del ensamblaje. Como se observa en la Figura 21, se puede ver los agujeros que complementan los de la placa principal, para poder ensamblarla placa.

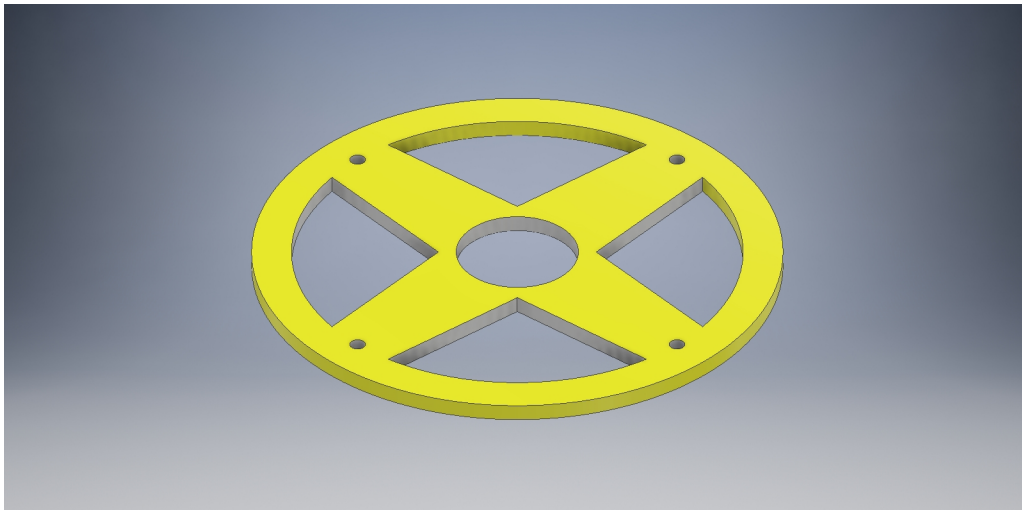


Figura 20: Soporte superior para la placa secundaria del diseño mecánico del dispositivo esclavo

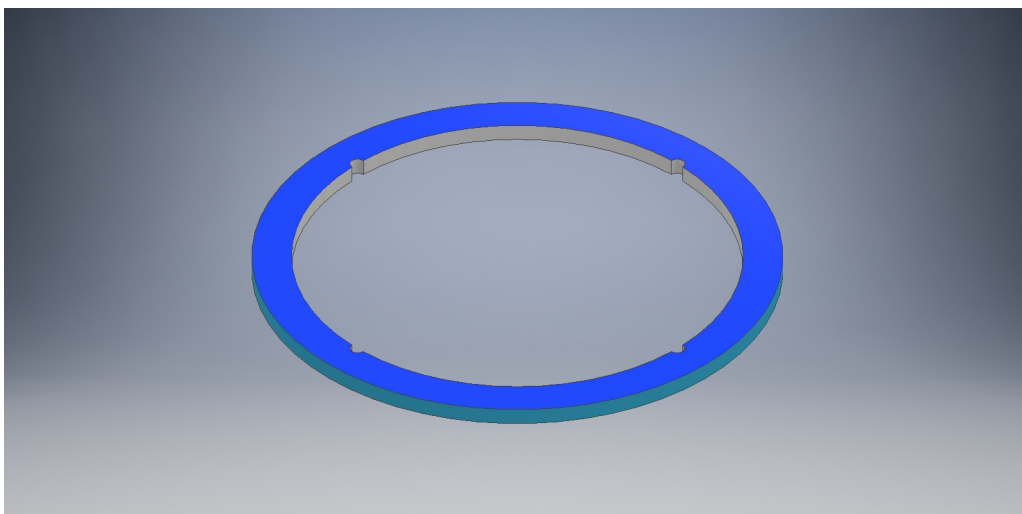


Figura 21: Rueda para ensamblaje de la placa

### 11.5. Pieza superior del diseño mecánico del dispositivo esclavo

La parte superior del diseño mecánico, Figura 22, es un disco que cuenta con varios agujeros. 4 de estos agujeros, los más externos, funcionan para poder atornillar la pieza con el resto del diseño. Los otro 4 agujeros funcionan para colocar los 4 NeoPixels. El último agujero sirve para que pase el sensor ultrasónico y se pueda conectar con la placa secundaria.

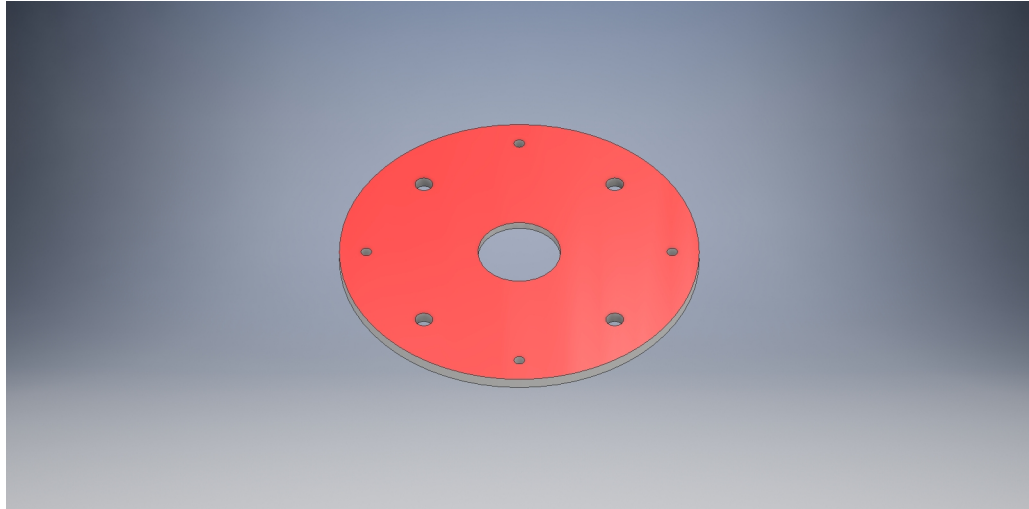


Figura 22: Parte superior del diseño mecánico del dispositivo esclavo

### 11.6. Pieza superior del diseño mecánico del dispositivo esclavo

Además de realizar los diseños de cada pieza del diseño mecánico, también se diseñaron las dos placas, para poder tener una perspectiva completa de como se estaría viendo el dispositivo esclavo completo. En la Figura 23 y Figura 24 se pueden ver las dos placas.

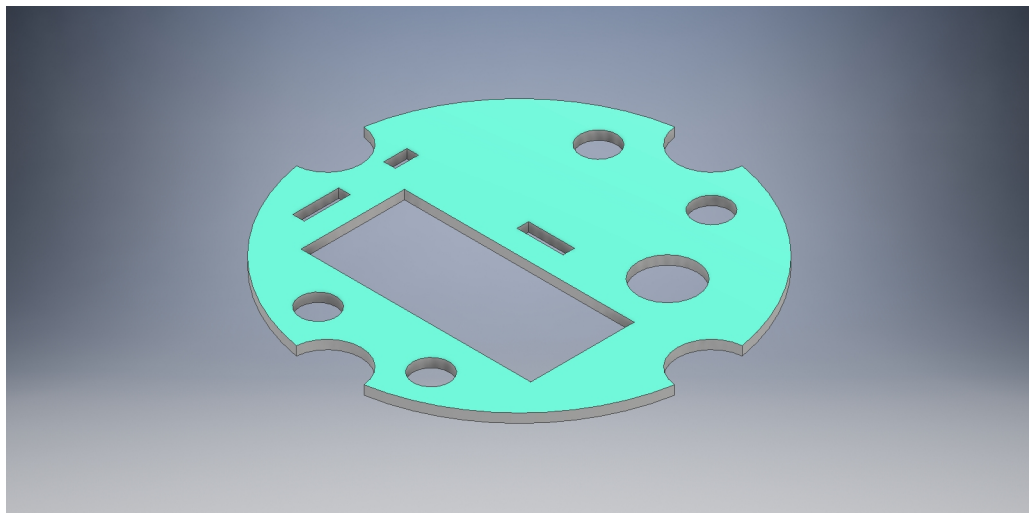


Figura 23: Modelo de la placa principal

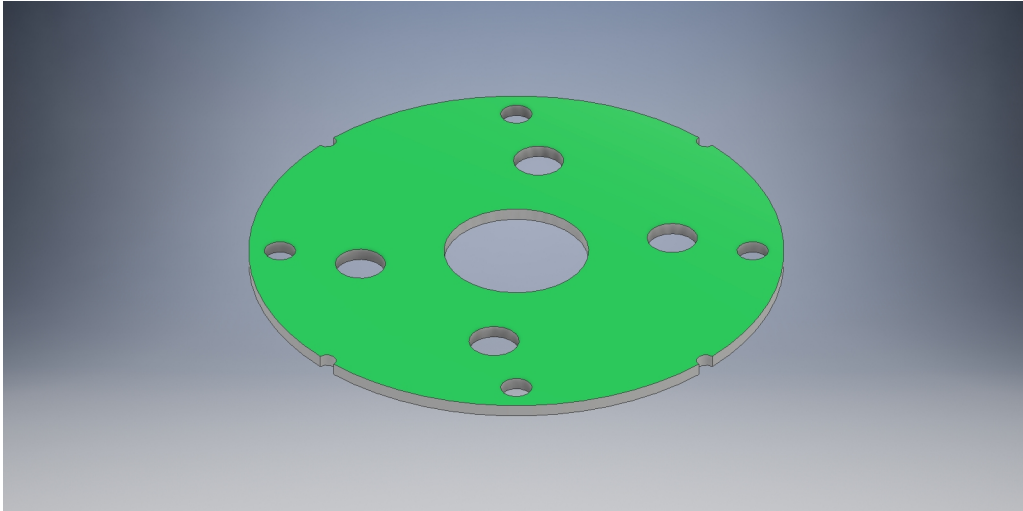


Figura 24: Modelo de la placa secundaria

---

### Implementación del dispositivo para la medición del tiempo de reacción de un deportista

---

Para llevar a cabo la implementación del dispositivo para la medición del tiempo de reacción de un deportista, se debe realizar varias pruebas con los equipos electrónicos seleccionados y con los diseños electrónicos y mecánicos.

Las primeras pruebas fueron las de la comunicación por radiofrecuencia. Es decir, la comunicación entre la Raspberry Pi 3 conectada con el módulo RFM69HCW y el Feather 32u4 RFM69HCW. Estas pruebas eran primordiales, debido a que si no se tenía una comunicación entre los dispositivos, no se podría desarrollar un dispositivo para la medición del tiempo de reacción de un deportista.

#### 12.1. Conexión Raspberry Pi 3 y RFM69HCW

Para la conexión del maestro se utilizaron los siguientes pines:

- +3.3V: Para alimentar con voltaje el RFM69HCW.
- GPI 10 (MOSI): Para la conexión SPI del MOSI.
- GPIO 9 (MISO): Para la conexión SPI del MISO.
- SLCK: Para la sincronía de los dispositivos.
- GPI 5: Para la conexión al Reset del modulo RFM69HCW.
- Ground: Para aterrizar a tierra el RFM69HCW.

- GPI 24: Para la conexión del G0 del módulo RFM69HCW.
- GPI 8: Para la conexión del Chip Select del RFM69HCW.

Se puede observar en la Figura 25, la conexión de la Raspberry Pi 3 y RFM69HCW.

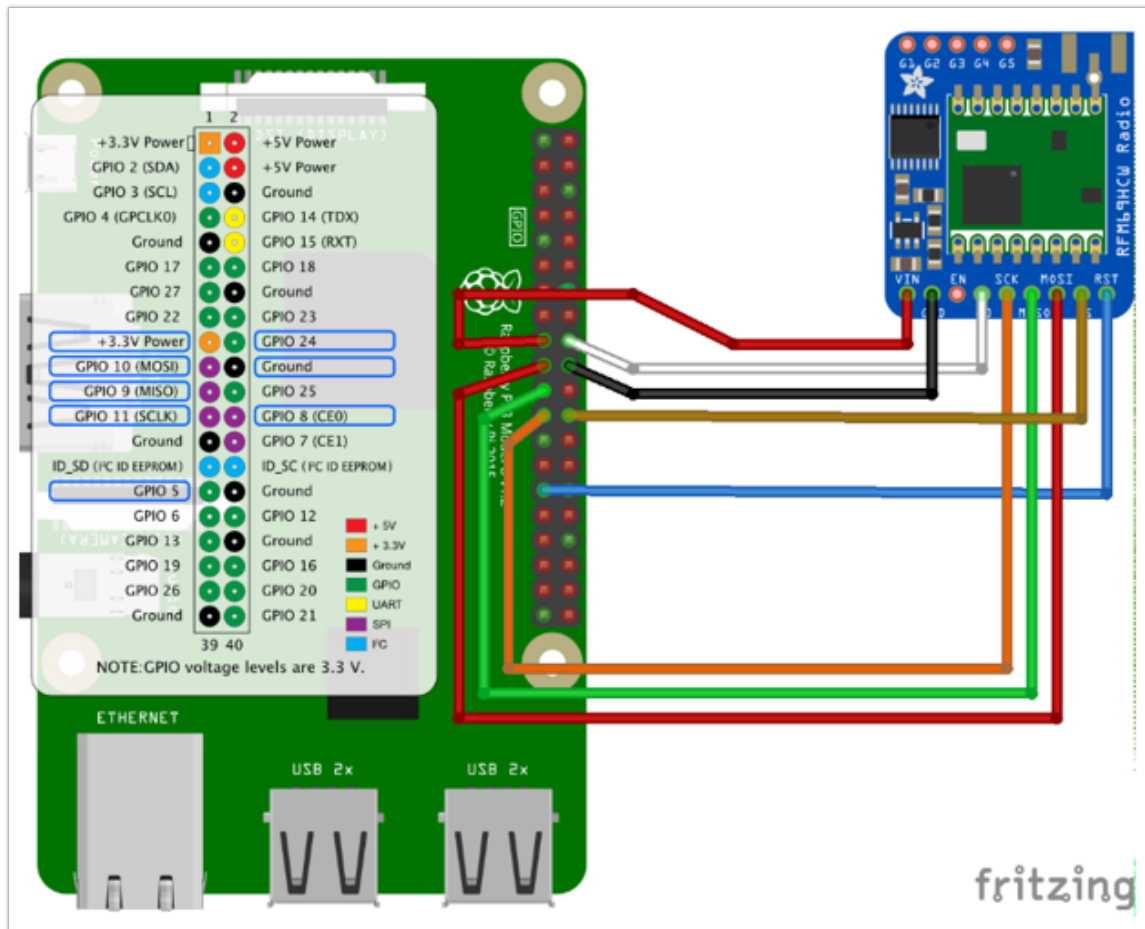


Figura 25: Conexión virtual de la Raspberry Pi 3 y del módulo RFM69HCW

## 12.2. Pruebas de comunicación RFM69HCW+Raspberry Pi 3 y Feather 32u4 RFM69HCW

En estas pruebas se logró tener una conexión completa con el maestro y el esclavo. Primero se hizo una prueba en donde se tenía una comunicación entre los dos dispositivos y estos se enviaban mensajes cada cierto tiempo. En la Figura 26 se muestra la prueba de comunicación vista desde el dispositivo maestro. Y en la Figura 27, se observa la prueba de comunicación vista desde el dispositivo esclavo.

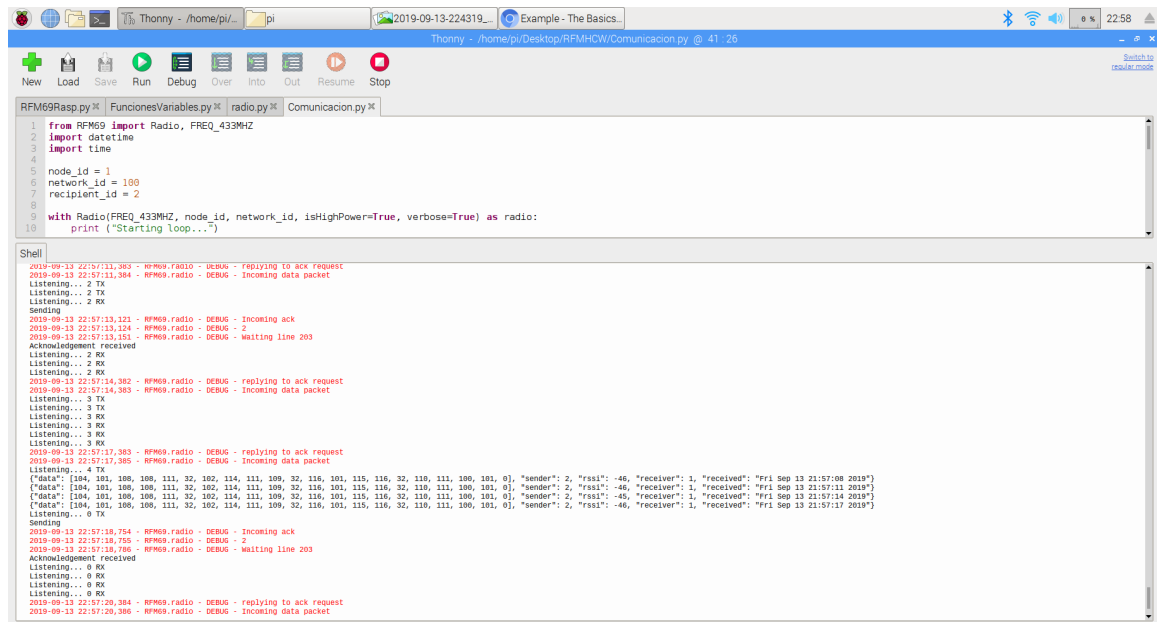


Figura 26: Primera prueba de comunicación con el dispositivo esclavo desde la Raspberry Pi 3



Figura 27: Primera prueba de comunicación con el dispositivo maestro desde el Feather32u4 RFM69HCW

### 12.2.1. Envío de datos utilizando protocolo de comunicación

Como se observa con anterioridad se obtuvo una comunicación exitosa. Dado a que se logró tener una buena comunicación, se decidió utilizar el protocolo de comunicación definido. Una vez hecho esto se modificó la programación del dispositivo esclavo. Esto se hizo para poder mostrar los resultados de los datos que se enviaban desde el dispositivo maestro. Cabe mencionar que en esta ocasión solamente se hizo una comunicación unidireccional, solo se

enviaron datos desde el dispositivo maestro. En la Figura 28, se muestra el código utilizado para recibir los datos del dispositivo maestro.

```
void loop() {
  if (rf69_manager.available())
  {
    String mensaje = leerMensaje(rf69_manager);
    AP = mensaje.charAt(7);
    Test = mensaje.charAt(6);
    if (AP=="0"){

    }
    if (AP=="1"){
      dato = "";
      if(Test=="1"){
        Serial.println(mensaje.charAt(6));
        colorWipe(pixels.Color(255, 0, 0), 50); // Red
        colorWipe(pixels.Color( 0, 255, 0), 50); // Green
        colorWipe(pixels.Color( 0, 0, 255), 50); // Blue
        colorWipe(pixels.Color( 255, 255, 255), 50); // Blue
      }
      for (int i=0; i<=15; i++){
        dato = dato + mensaje.charAt(i);
        if (i == 2){
          if (dato == "001")
            ID = 1;

          if (dato == "010")
            ID = 2;

          if (dato == "011")
            ID = 3;
          dato="";
        }
        if(i == 5){
          if(dato == "001"){
            r = 0;
            g = 0;
            b = 255;
          }

          if(dato == "010"){
            r = 0;
            g = 255;
            b = 0;
          }

          if(dato == "011"){
```

Figura 28: Programación para la recepción de datos del dispositivo maestro

Los datos que se obtuvieron se muestran en la Figura 29. Aquí observamos que los datos recibidos coinciden con los datos del protocolo establecido para esta prueba. El protocolo se muestra en la Figura 30.

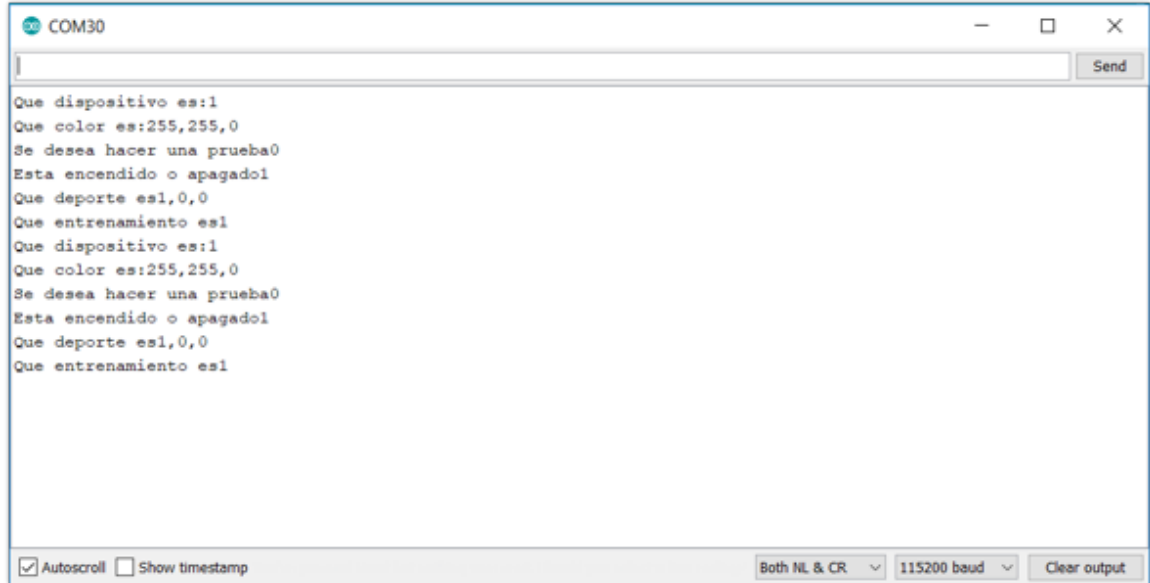


Figura 29: Datos recibidos en el Feather 32u4 RFM69HCW

1 Byte					2 Byte										
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
ID Dispositivo					Color		Test	ON/OFF	Deporte			Entrenamiento		N/A	N/A

Figura 30: Protocolo utilizado en la prueba unidireccional de comunicación

Como se obtuvieron pruebas de comunicación totalmente exitosas. Se comenzó a trabajar en la configuración y programación de los dos sensores, capacitivo y ultrasónico.

### 12.3. Utilización de los sensores capacitivo y ultrasónico en el Feather 32u4 RFM69HCW

Se comenzó haciendo pruebas con el sensor ultrasónico, debido a que este es el sensor más complicado de utilizar.

#### 12.3.1. Pruebas con el sensor ultrasónico JSN-SR04T-2.0 en el Feather 32u4 RFM69HCW

Para estas pruebas, se utilizó la librería ultrasonic.h. Esta librería permite elegir los pines que se utilizan para enviar las señales del Echo y Trigger. La programación se observa en la Figura 31. La programación es muy fácil. La librería utilizada para el sensor, permite obtener el dato de la distancia del objeto más cercano. Según esto se hizo la programación en el dispositivo esclavo. Esta prueba, que fue la única que se hizo utilizando solamente el sensor ultrasónico JSN-SR04T-2.0 y el Feather 32u4 RFM69HCW, se basó en la detección de la distancia de un objeto. El programa básicamente entraba a un ciclo y detectaba un objeto próximo cada cierto tiempo. Los resultados de las distancias se pueden observar en la Figura 32.

```

#include <Ultrasonic.h>

Ultrasonic ultrasonic(10, 9);
int distance;

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

void loop() {
  distance = ultrasonic.read();

  Serial.print("Distance in CM: ");
  Serial.println(distance);
  delay(1000);
}

```

Figura 31: Programación para la medición de distancia de un objeto utilizando el sensor JSN-SR04T-2.0



Figura 32: Resultados de la medición de distancia utilizando el sensor JSN-SR04T-2.0

Ya que se obtuvo un buen resultado en la utilización del sensor ultrasónico. Se comenzó a realizar pruebas con el sensor capacitivo.

### 12.3.2. Pruebas con el sensor capacitivo en el Feather 32u4 RFM69HCW

Para el sensor capacitivo no se utilizó ningún componente electrónico. Se utilizó una librería que proporciona Arduino, CapacitiveSensor.h. El único componente electrónico que se utilizó fue una resistencia de 1 Megaohm que estaba conectada en paralelo con dos pines

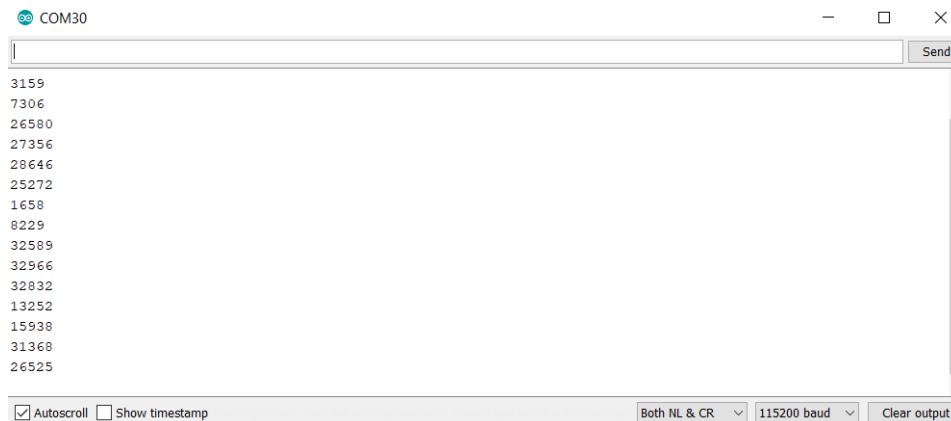
del Feather 32u4 RFM69HCW. La programación para las pruebas con este sensor se muestra en la Figura 50. Para esta prueba, se usó solamente la resistencia mencionada y un cable que funcionaba para la conexión con la parte metálica, en este caso se utilizó aluminio. La programación consiste en que cada vez que se presiona la parte metálica conectada, se recibe una señal. Si esta señal es mayor a 1000, quiere decir que la parte metálica tuvo contacto con algún objeto o parte del cuerpo (Mano, dedos, pie, etc). Los resultados se muestran en la Figura 34.

```
#include <CapacitiveSensor.h>

CapacitiveSensor  cs = CapacitiveSensor(11,12);
void setup()
{
  cs.set_CS_Autocal_Millis(0xFFFFFFFF);
  Serial.begin(9600);
}

void loop()
{
  long sensor1 = cs.capacitiveSensor(50);
  if(sensor1 >= 1000)
  {
    Serial.println(sensor1); |
  }
}
```

Figura 33: Programación para el sensor capacitivo



```
COM30
3159
7306
26580
27356
28646
25272
1658
8229
32589
32966
32832
13252
15938
31368
26525
```

Figura 34: Resultados de las pruebas utilizando el sensor capacitivo

## 12.4. Pruebas para las rutinas diseñadas para los deportes: softball, basketball y box

Las pruebas de los entrenamientos son cruciales debido a que estos ayudaran a los deportistas en sus entrenos. Se hicieron tres rutinas para cada deporte, es decir nueve rutinas en total entre los tres deportes. Las rutinas diseñadas funcionan con 1 o más dispositivos

esclavos. Debido a que solo se tenía un Feather dispositivo esclavo con la implementación del diseño mecánico, Figura 71 y un Feather 32u4 RFM69HCW sin ningún componente electrónico conectado, además de su módulo de radiofrecuencia. Se decidió modificar el código del dispositivo 2 para simular un tiempo de reacción y que existieran dos dispositivos que estuvieran comunicándose con el dispositivo maestro.

En las pruebas de basquetball se están utilizando los dos sensores, sensor ultrasónico y sensor capacitivo. Se decidió hacer esto porque los atletas de este deporte, necesitan poder robar el balón de los contrincantes y también necesitan ser rápidos al momento de realizar un bloqueo. Estas dos razones hicieron que se tomara la decisión de utilizar los dos sensores en los entrenos de este deporte.

### Pruebas para los entrenamientos para el deporte Basquetball:

```

1 Programa Principal
2 12/8/2019
3 Marcel Velazquez 15534
4 Programa Principal para la comunicacion con
5 los dispositivos esclavos
6
7
8 #Importar el programa FuncionesVariables, aqui se encuentran todas las funciones para el funcionamiento del proyecto
9 import FuncionesVariables as fv
10
11 print("Bienvenido al programa de medicion de tiempo de reaccion de un deportista")
12 print("Por favor, ingresa los datos que se le piden para comenzar su entreno")
13 deporte = int(input("Que deporte entrenara?n 1. Basquetball 2. Box 3. Softball\n Ingrese el numero del deporte que entrenara\n"))
14 entrenamiento = int(input("Que entrenamiento quiere hacer?n 1. Entreno numero 1. 2. Entreno numero 2 3. Entreno numero 3\n Ingrese el numero del entrenamiento que realizara\n"))
15 if (deporte==1):
16     fv.entrenamientosS8(entrenamiento)
17
18 if (deporte==2):
19     fv.entrenamientosS8(entrenamiento)
20
21 if (deporte==3):
22     fv.entrenamientosS8(entrenamiento)
23
24
25
Shell
>>> !run FuncionesVariables.py
>>> !run RFM69Rasp.py
Bienvenido al programa de medicion de tiempo de reaccion de un deportista
Por favor, ingresa los datos que se le piden para comenzar su entreno
Que deporte entrenara?
1. Basquetball 2. Box 3. Softball
Ingrese el numero del deporte que entrenara
1
Que entrenamiento quiere hacer?
1. Entreno numero 1. 2. Entreno numero 2 3. Entreno numero 3
Ingrese el numero del entrenamiento que realizara
1
~/usr/local/lib/python3.6/dist-packages/RFM69/radio.py:79: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
GPIO.setwarnings(False, GPIO.OUT)
Acknowledgment received del dispositivo 21
[0x 00, 00, 00]
Dato del tiempo esclavo 1
1:55
>>>

```

Figura 35: Prueba entrenamiento 1 basquetball

```

1 ---
2 Programa Principal
3 12/8/2019
Shell
>>> !run RFM69Rasp.py
Bienvenido al programa de medicion de tiempo de reaccion de un deportista
Por favor, ingresa los datos que se le piden para comenzar su entreno
Que deporte entrenara?
1. Basquetball 2. Box 3. Softball
Ingrese el numero del deporte que entrenara
1
Que entrenamiento quiere hacer?
1. Entreno numero 1. 2. Entreno numero 2 3. Entreno numero 3
Ingrese el numero del entrenamiento que realizara
2
Cuantas veces quiere realizar el entreno numero 2?
1
~/usr/local/lib/python3.6/dist-packages/RFM69/radio.py:79: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
GPIO.setwarnings(False, GPIO.OUT)
Acknowledgment received del dispositivo 21
[0x 00, 00, 00]
Dato del tiempo esclavo 1
1:52
Acknowledgment received del dispositivo 22
[0x 00, 07, 05, 00, 00, 01]
Dato del tiempo esclavo 2
2:00.13
Matriz
Acknowledgment received del dispositivo 22
[0x 00, 07, 05, 00, 00, 01]
Dato del tiempo esclavo 2
2:00.4
Acknowledgment received del dispositivo 22
[0x 00, 00, 00, 00, 00, 01]
Dato del tiempo esclavo 2
2:00.13
Acknowledgment received del dispositivo 21
[0x 00, 00, 00]
Dato del tiempo esclavo 1
0:48
Tiempo con el dispositivo 1:
0:48
Tiempo con el dispositivo 2:
[2:00.71, 2:00.4, 2:00.13] Segundos
Tiempo con el dispositivo 1:
[] Segundos
>>>

```

Figura 36: Prueba entrenamiento 2 basquetball

```

RFM69Rasp.py X FuncionesVariables.py X
1  radio=initiar(RF69,MODELO,MODO)
2  mensaje1="001101010100100"
3  mensaje2="001100001000110"
4  transmitir(FRONT, mensaje1, radio1)
5  transmitir(FRONT, mensaje2, radio1)
6  transmitir(FRONT, mensaje1, radio1)

Shell
Thorny: ~
>>> !Run FuncionesVariables.py
>>> !Run RFM69Rasp.py
Atención! el programa de medición de tiempo de reacción de un deportista
por favor, ingrese los datos que se le piden para comenzar su entreno
Que deporte entrenará?
1. Basquetball 2. Box 3. Softbal
Ingrese el número del deporte que entrenará
2
Que entrenamiento quiere hacer?
1. Entreno numero 1, 2. Entreno numero 2 3. Entreno numero 3
Ingrese el número del entrenamiento que realizará
1
/usr/local/lib/python3.5/dist-packages/RFM69/radio.py:79: RuntimeWarning: This channel is already in use, continuing anyway. Use gpio.setwarnings(False) to disable warnings.
  Acknowledgment received del dispositivo 21
[07, 56, 01]
Acknowledgment received del dispositivo 22
[01, 53, 50, 48, 49, 49, 49]
Acknowledgment received del dispositivo 21
[49, 50, 50, 47]
Acknowledgment received del dispositivo 22
[01, 50, 50, 50, 50, 48, 49]
Acknowledgment received del dispositivo 22
[01, 51, 50, 50, 50, 49, 49]
Datos del tiempo esclavo 2
0:00:00
RFM69
Acknowledgment received del dispositivo 22
[01, 50, 50, 48, 50, 49]
Datos del tiempo esclavo 2
0:00:00
Final
Tiempo con el dispositivo 1:
[0.00, 1.47] Segundos
Tiempo con el dispositivo 2:
[3508.01, 3508.07, 3508.56, 3508.32] Segundos
Tiempo con el dispositivo 3:
[0] Segundos
>>>
>>> !Run FuncionesVariables.py

```

Figura 37: Prueba entrenamiento 3 basquetball

En las pruebas de box se utilizó solamente un sensor, el sensor ultrasónico. Los boxeadores no necesitan golpear el dispositivo, además de que se pueden dañar al ser golpeados con mucha fuerza, ellos solo necesitan reaccionar al estímulo visual y desactivarlos sin la necesidad de tocarlos. Los entrenos de este deporte se basaron en la rapidez de golpe del boxeador. El enfoque de los entrenos se basó en simular que el contrincante bajo su guardia. La simulación es la activación del estímulo visual del dispositivo esclavo. Como se creó una oportunidad de golpe, el boxeador la aprovechará y golpeará al contrincante, por lo que apagar el dispositivo simula el golpe del boxeador.

Pruebas para los entrenamientos para el deporte box:

```

FuncionesVariables.py X RFM69Rasp.py X
1  ...
2  Programa Principal
3  12/16/2019
.....

Shell
Thorny: ~
>>> !Run FuncionesVariables.py
>>> !Run RFM69Rasp.py
Atención! el programa de medición de tiempo de reacción de un deportista
por favor, ingrese los datos que se le piden para comenzar su entreno
Que deporte entrenará?
1. Basquetball 2. Box 3. Softbal
Ingrese el número del deporte que entrenará
2
Que entrenamiento quiere hacer?
1. Entreno numero 1, 2. Entreno numero 2 3. Entreno numero 3
Ingrese el número del entrenamiento que realizará
1
/usr/local/lib/python3.5/dist-packages/RFM69/radio.py:79: RuntimeWarning: This channel is already in use, continuing anyway. Use gpio.setwarnings(False) to disable warnings.
  Acknowledgment received del dispositivo 22
[49, 50, 48, 50, 52, 50, 49]
RFM69
Acknowledgment received del dispositivo 21
[01, 49, 50, 47]
Acknowledgment received del dispositivo 22
[49, 50, 48, 54, 52, 48, 49]
Acknowledgment received del dispositivo 22
[49, 50, 48, 56, 49, 53, 49]
RFM69
Acknowledgment received del dispositivo 21
[01, 49, 50, 47]
Acknowledgment received del dispositivo 22
[49, 50, 49, 50, 50, 48, 49]
Acknowledgment received del dispositivo 22
[01, 50, 49, 51, 51, 50, 49]
RFM69
Acknowledgment received del dispositivo 21
[01, 50, 49]
Acknowledgment received del dispositivo 22
[49, 50, 49, 50, 50, 51, 49]
Tiempo con el dispositivo 1:
[1.01, 1.07, 0.99] Segundos
Tiempo con el dispositivo 2:
[1202.42, 1206.4, 1208.15, 1212.2, 1213.93, 1217.00] Segundos
Tiempo con el dispositivo 3:
[0] Segundos
>>>
>>> !

```

Figura 38: Prueba entrenamiento 1 box

```

1 ***
2 Programa Principal
3 12/05/2019
4 Marcel Velasquez 15534
5
Shell
>>> #Run RFM69Rasp.py
Ejecutando el programa de medicion de tiempo de reaccion de un deportista
Por favor, ingrese los datos que se le piden para comenzar su entreno
Que deporte entrenaras?
1. Baseball 2. Softball
ingrese el numero del deporte que entrenara
2
Que entrenamiento quiere hacer?
1. Entreno numero 1, 2. Entreno numero 2 3. Entreno numero 3
ingrese el numero del entrenamiento que realizara
2
RFM69
RFM69:local/120/serial/5.0141/packages/RFM69/radio.py:78: RuntimeWarning: This channel is already in use, continuing anyway. Use gpio.setwarnings(False) to disable warnings.
GPIO.setmode(GPIO.BOARD)
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 57, 61]
RFM69
No Acknowledgment del dispositivo 21
[0, 56, 62]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 63]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 64, 61]
Acknowledgment received del dispositivo 21
[0, 56, 57, 58, 59, 60]
RFM69
No Acknowledgment del dispositivo 21
[0, 56, 65, 62]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 66]
RFM69
No Acknowledgment del dispositivo 21
[0, 56, 67]
>>>

```

Figura 39: Prueba entrenamiento 2 box

```

[0, 56, 57, 58, 59, 60]
[0, 56, 57, 58, 59, 60]
RFM69
Acknowledgment received del dispositivo 21
[0, 56, 61]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
RFM69
Acknowledgment received del dispositivo 22
[0, 56, 48, 49, 50, 51]
Acknowledgment received del dispositivo 21
[0, 56, 62]
RFM69
No Acknowledgment del dispositivo 21
[0, 56, 63]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 64, 61]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
RFM69
No Acknowledgment del dispositivo 21
[0, 56, 65]
Acknowledgment received del dispositivo 22
[0, 56, 57, 58, 59, 60]
Acknowledgment received del dispositivo 21
[0, 56, 66, 63]
Acknowledgment received del dispositivo 22
[0, 56, 48, 49, 50, 51, 67]
RFM69
Acknowledgment received del dispositivo 21
[0, 56, 68]
Acknowledgment received del dispositivo 22
[0, 56, 48, 49, 50, 51, 69]
[0, 56, 48, 49, 50, 51, 69]
Tiempo con el dispositivo 1:
[1.25, 1.4, 0.83, 0.86, 0.96, 1.38, 0.69, 0.69] Segundos
Tiempo con el dispositivo 2:
[3283.28, 3289.81, 3294.64, 3296.65, 3300.15, 3307.21, 3310.69, 3314.15] Segundos
Tiempo con el dispositivo 3:
[] Segundos
Tiempo con el dispositivo 4:
[] Segundos
>>>

```

Figura 40: Prueba entrenamiento 3 box

En las pruebas de softball se utilizó solo el sensor ultrasónico, al igual que en box. En estas pruebas se tiene como máximo 2 dispositivos esclavos. Se hizo esto debido a que los entrenos se basaron mucho en la forma de bateo del atleta. Los entrenos diseñados se basan solamente en pasar el bate por encima del dispositivo. Los atletas reaccionan ante el estímulo visual y su reacción es pasar el bate por encima.

Prueba para los entrenamientos para el deporte softball:



se encontraban los sensores y los LEDs, como se puede observar en la Figura 44.

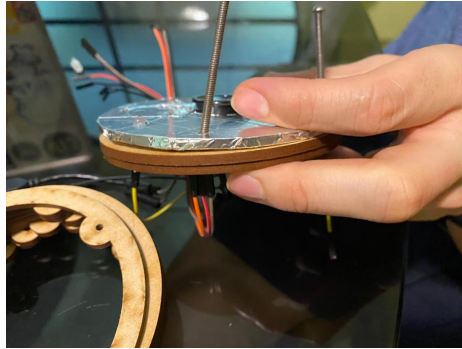


Figura 44: Pieza sensores del dispositivo esclavo

Luego se fueron colocando el resto de las piezas una por una y conectando las placas entre ellas y ajustándolas al diseño mecánico. En la Figura 45, se puede observar como se están conectando las piezas y también se puede observar que la placa de los sensores ya se encuentra ensamblada. En la Figura 46, se puede observar como se ve el dispositivo ensamblado.

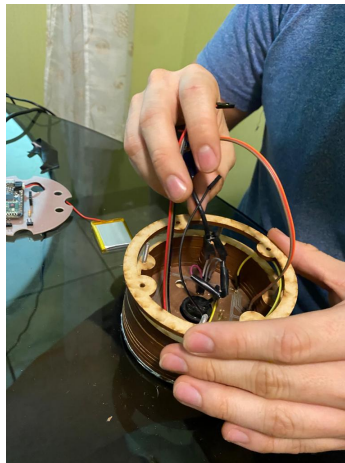


Figura 45: Construcción dispositivo esclavo



Figura 46: Dispositivo esclavo ensamblado

Para utilizar el servidor y la página web en el dispositivo maestro, se siguieron los pasos de instalación de paquetes y librerías que se muestran en la siguiente guía [12]. La instalación de estos paquetes y librerías, permite que el dispositivo maestro funcione como un servidor y como un punto de acceso, es decir una red WiFi. Que el servidor funcione como una red WiFi, nos permite conectarnos desde cualquier dispositivo electrónico que tenga instalado una aplicación web y poder utilizar la página web creada utilizando el servidor de Django.

### 13.1. Diseño de página web

Para lograr un ambiente amigable para el usuario, se debe de utilizar una página web que sea fácil de usar para el usuario. Para el diseño de la página web se utilizó HTML y CSS. CSS permite que el diseño de una página web sea más fácil de realizar y de modificar. Se utilizó como software Visual Studio Code, para crear y diseñar las páginas web. El primer diseño que se hizo fue la página de Login y Sign-Up.

#### 13.1.1. Página login y sign-up

La página web de Login y Sign-Up, se tiene un mismo diseño, la diferencia se encuentra en los campos que el usuario debe llenar. La página web de Sign-Up es la página para registrarse y crear un usuario por primera vez. La página web de Login funciona para iniciar sesión con un usuario ya creado y registrado con anterioridad. Al iniciar sesión, la página web se dirige al menú principal.

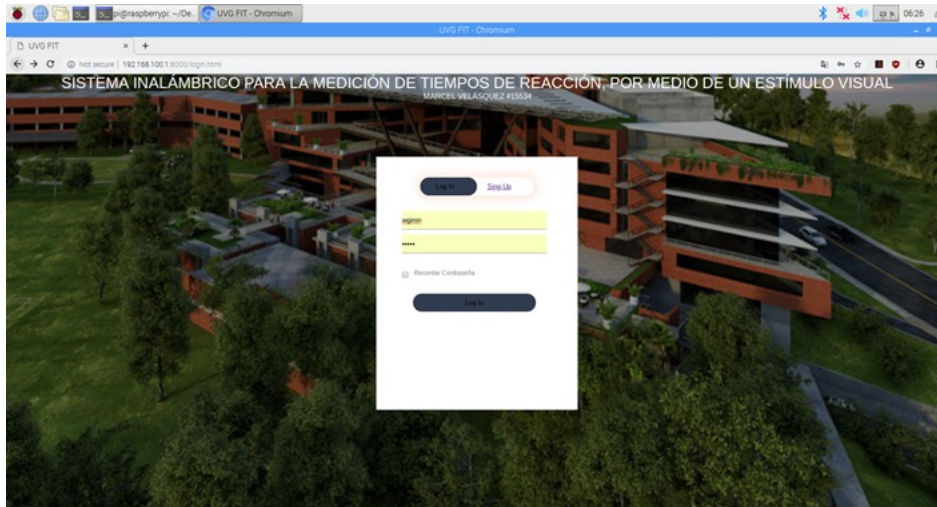


Figura 47: Página web

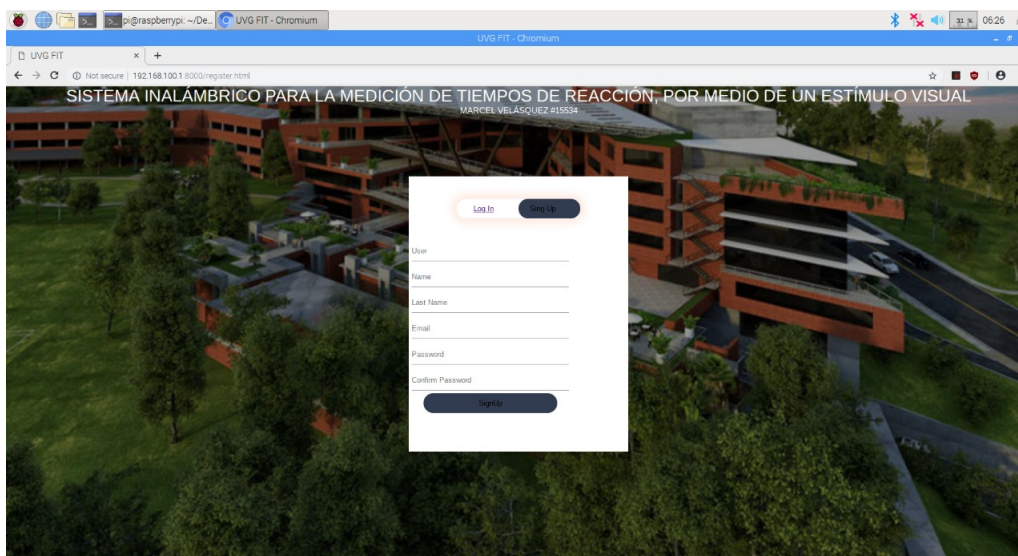


Figura 48: Sing-up de la página web

### 13.1.2. Página menú principal

El diseño de la página web del menú principal, consiste en varias pestañas rectangulares en el centro de la página. Estas pestañas se diseñaron para que cuando sean presionadas se extiendan y pueda observarse todas las opciones. Por ejemplo en la Figura 49, se observa que al presionar la pestaña se puede ver las opciones de entrenamiento que existen. Al seleccionar algunas de las opciones de los deportes, se dirige a una nueva página web con un diseño general para todos los entrenos.

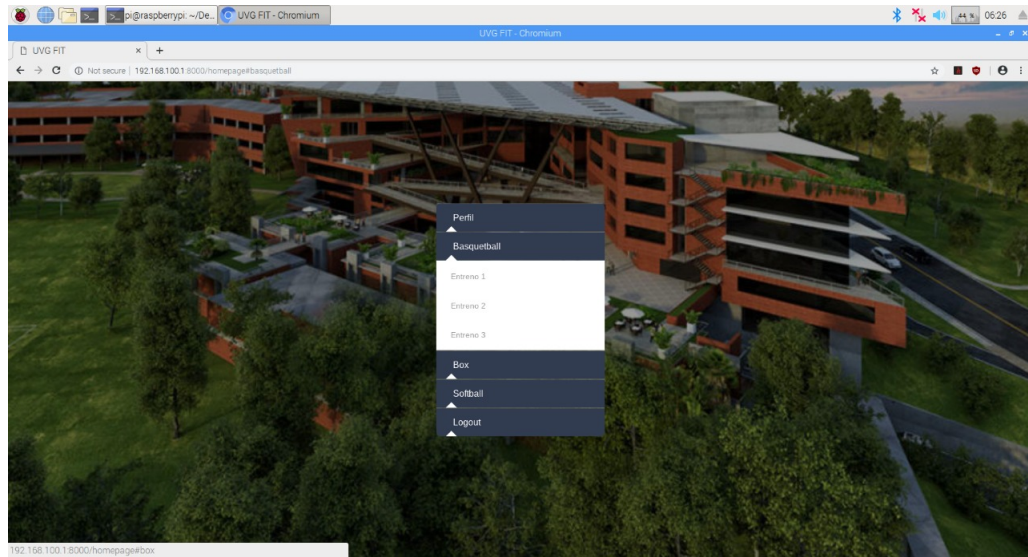


Figura 49: Menú principal página web

### 13.1.3. Página entrenos

En esta página web es un diseño general para todos los entrenos disponibles. Se le puso una tabla en donde se muestra cuanto tiempo hizo el deportista en apagar cada uno de los dispositivos utilizados en el entreno.

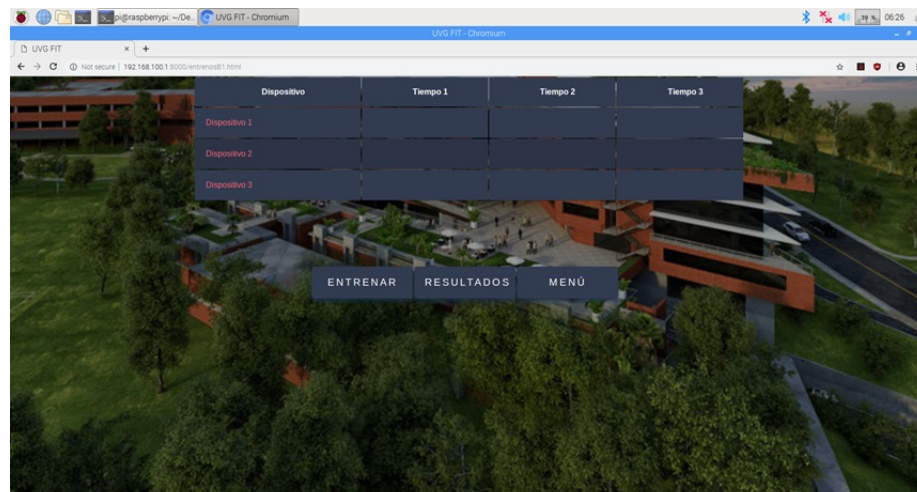


Figura 50: Entrenos página web

## 13.2. Servidor web en el dispositivo maestro

Para la instalación del servidor en la Raspberry Pi 3, como se mencionó antes, se utilizó como referencia la siguiente tesis [12]. El servidor funciona con Python 3.6 y con Django. Para la instalación del servidor web, se realizaron los siguientes pasos:

### 13.2.1. Instalación de Django

Para instalar Django solo se debe de realizar los siguientes dos comandos en la terminal de la Raspberry Pi.

```
$ pip3 install django
$ pip3 install psycopg2
```

Si falla alguno de estos comandos falla, es recomendable hacer un update del sistema, utilizando estos dos comandos:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

### 13.2.2. Instalación de PostgreSQL base de datos

Para guardar la información de los deportistas se debe de tener una base de datos, que también funcionará para almacenar los datos de los entrenamientos realizados por los deportistas. Para la instalación se debe de hacer lo siguiente:

```
$ pyenv deactivate
$ sudo apt-get install libpq-dev python-dev
$ sudo apt-get install postgresql postgresql-9.5
$ sudo -u postgres psql
```

Una vez se instala la base de datos, se debe de configurar y crear en la aplicación web y para lograr esto se utiliza lo siguiente:

Ingresamos a la base da datos y creamos un usuario. Se utilizará el usuario por defecto Pi. Después de ingresar estos comandos, nos pedirá ingresar una contraseña, esta contraseña es importante guardarla porque funcionara ingresar a la plataforma de la base de datos por si en algún momento queremos modificarla. Después nos hará una serie de preguntas, se debe de contestar N,Y,Y.

```
$ sudo su postgres
$ createuser pi -P --interactive
```

Por último podemos crear la base de datos, con los comandos mostrados abajo. Es posible que para empezar a utilizar la base de datos, se deba de realizar algunas migraciones. Estas migraciones funcionan para crear los modelos de la base de datos.

```
$ psql
> CREATE DATABASE training_app
```

Los comando para las migraciones son:

```
$ python3 manage.py makemigrations
$ python3 manage.py migrate
```

### 13.3. Raspberry Pi 3 como punto de acceso inalámbrico

La Raspberry Pi debe de funcionar como un punto de acceso para que desde cualquier dispositivo se pueda conectar a la red y pueda ingresar a la página web . Los pasos para hacer a la Raspberry Pi una punto de acceso so los siguientes:

Primero instalamos los paquetes dnsmasq y Hostapd:

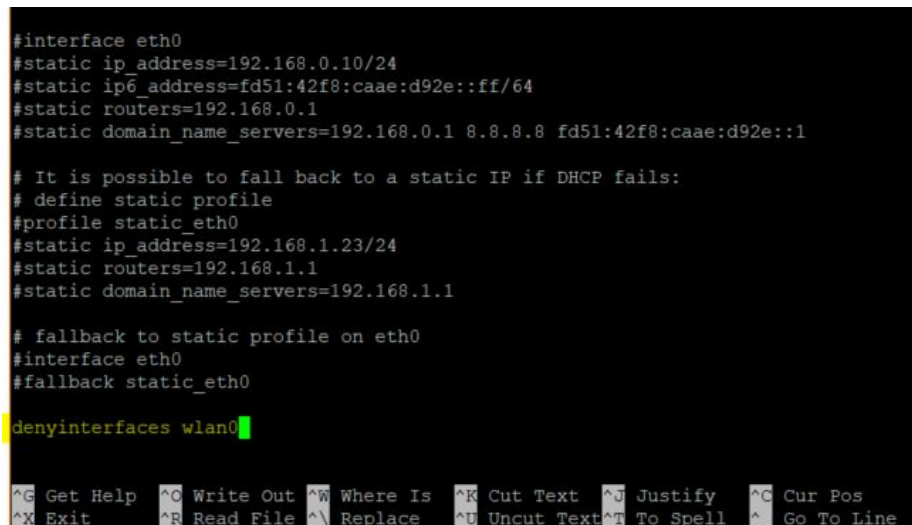
```
$ sudo apt-get -y install hostapd dnsmasq
```

El paquete de Hostapd funciona para que se pueda utilizar el Wifi radio de la Raspberry Pi como un punto de acceso y el dnsmasq funciona para la combinación de DHCP y el servidor DNS, es decir, maneja las direcciones IP y traduce los nombres de dominio a direcciones IP.

La configuración de la red está controlada por el dhcpd. Por lo tanto debemos de ignorar la interfaz inalámbrica, wlan0, y configurar una IP estática.

Entonces para esto, editamos el siguiente archivo, dhcpd. Y hasta el final del documento escribimos denyinterfaces wlan0. En la Figura 51, se puede observar como se debe de ver el archivo. Guardamos el archivo con CTRL+X y luego presionamos Y.

```
$ sudo nano /etc/dhcpd.conf
denyinterfaces wlan0
```



```
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

denyinterfaces wlan0
```

Figura 51: Configuración dhcpd

Ahora debemos de indicar la dirección de la IP estática para interfaz WiFi. Abrimos la interfaz con el comando de abajo y al final del documento debemos de agregar lo siguiente. El documento debe de verse como se muestra en la Figura 52. Las IPs pueden variar, no necesariamente deben de ponerse esas.

```
$ sudo nano /etc/network/interfaces
```

Lines a agregar:

```

auto eth0
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.5.1
    netmask 255.255.255.0
    network 192.168.5.0
    broadcast 192.168.5.255

```

```

auto eth0
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.5.1
    netmask 255.255.255.0
    network 192.168.5.0
    broadcast 192.168.5.255

```

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos  
^X Exit   ^R Read File   ^\ Replace   ^U Uncut Text   ^T To Spell   ^\_ Go To Line

Figura 52: Configuración IP estática

Después de indicar la IP estática, se debe de configurar el Hostapd. El Hostapd se configura para transmitir un SSID y permitir conexiones WiFi en un canal determinado. Entonces ingresamos el comando de abajo y las líneas que se muestran abajo para configurar. El comando para ingresar al archivo, creara un nuevo archivo porque este no existe hasta este punto. El SSID es nombre de la red, el Channel el canal en donde se estará transmitiendo la red, se puede cambiar entre un numero del 1 al 11 y por último el wpapassphrase es la contraseña de la red. Estos datos se pueden cambiar cuando sea. Como siempre, al final guardamos el archivo.

```

$ sudo nano /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=FIT_UVG
hw_mode=g
channel=6
ieee80211n=1
wmm_enabled=1
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
macaddr_acl=0
auth_algs=1

```

```

ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=UVGFIT123
rsn_pairwise=CCMP

```

Al terminar esto, pasamos a indicar al Hostapd donde se encuentra la configuración. Para esto debemos de abrir el archivo hostapd e ingresamos la siguiente linea:

```

$ sudo nano /etc/default/hostapd
DAEMON_CONF="/etc/hostapd/hostapd.conf"

```

El archivo debe de ver como se muestra en la Figura 53. Y por último guardamos el archivo.

```

# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"
#
# Additional daemon options to be appended to hostapd command:-
#   -d show more debug messages (-dd for even more)
#   -K include key data in debug messages
#   -t include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#

```

Figura 53: Configuración hostapd

El último paso es configurar el Dnsmasq. Este nos ayuda a asignar una dirección IP a los nuevos dispositivos que se conectan a la red de la raspberry pi. Ingresamos los siguientes comandos para configurar el archivo:

```

$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.bak
$ sudo nano /etc/dnsmasq.conf

```

Una vez hacemos esto, se crea el archivo e ingresamos los datos que se muestran abajo. La IP 192.168.5.1 esta reservada directamente para la Raspberry Pi, entonces eso quiere decir que ninguna IP entre 192.168.5.2 - 192.168.5.9 y entre 192.168.5.201 - 192.168.5.24 se puede utilizar para los dispositivos que se conecten. por eso el rango de IPs se tiene de 192.168.5.100 - 192.168.5.200. El archivo debe de verse como se muestran en la Figura 54.

```

interface=wlan0
listen-address=192.168.5.1
bind-interfaces
server=8.8.8.8
domain-needed

```

```
bogus-priv
dhcp-range = 192.168.5.100, 192.168.5.200, 24h
```

```
interface=wlan0
listen-address=192.168.5.1
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=192.168.5.100,192.168.5.200,24h
```

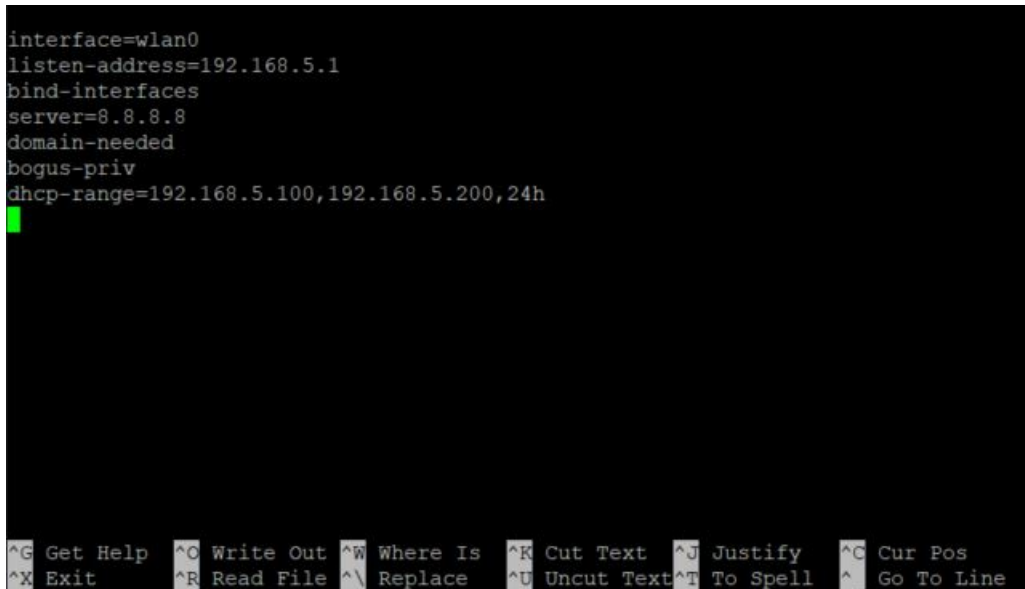


Figura 54: Configuración dnsmasq

Ahora por último, reiniciamos nuestra Raspberry Pi, ya sea por medio de la terminal o por la forma interactiva del sistema operativo.

Una vez se reinicie la raspberry Pi, podremos encontrar nuestra red creada desde la Raspberry Pi como se mira en la Figura 55 y la Figura 56.

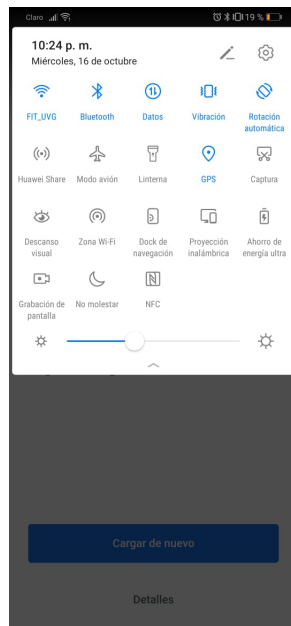


Figura 56: Conexión WiFi

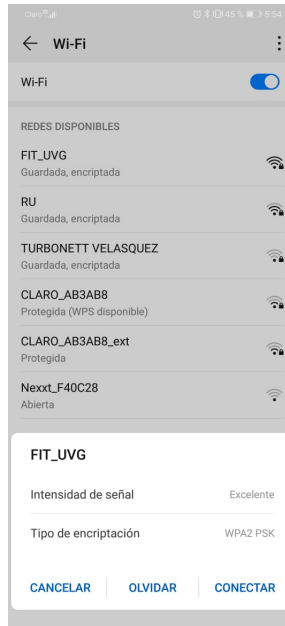


Figura 55: Conexión red WiFi

## 13.4. Implementación de rutinas en página web

Para implementar las rutinas en la página web, no se necesita hacer mucho. Esto porque nuestro servidor funciona con Python. Lo único que se debe de hacer es mover nuestra librería del modulo de radio frecuencia a la carpeta de Training de nuestro servidor. Y por ultimo solo ponemos nuestra función en alguna de las funciones del archivo Views.py, para que cuando se utilice la página de entreno, al tocar el botón de entrenar, se inicié la función y empiece a correr el programa para entrenar.

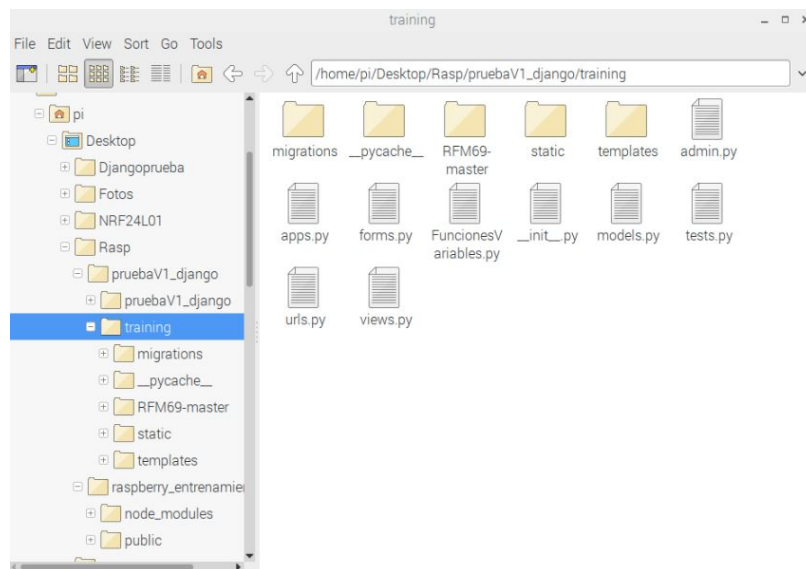


Figura 57: Carpeta Training

```
forms.py x views.py x
118 def prueba(request):
119     return render(request, 'training/login.html')
120
121 @login_required(login_url='login/')
122 def entreno(request):
123     BBeI()
124     return render(request, 'training/homepage.html')
125
126 @login_required(login_url='login/')
127 def games(request):
128     games_list = PresetTrainingSession.objects.order_by('id')
129     return render(request, 'training/games.html', {'games_list': games_list})
130
131 @login_required(login_url='login/')
132 def instrucciones(request):
133     games_list = PresetTrainingSession.objects.order_by('id')
134     return render(request, 'training/instrucciones.html', {'games_list': games_list})
135
136 @login_required(login_url='login/')
137 def capture_mode(request):
138     return render(request, 'training/modo_captura.html')
139
140 @login_required(login_url='login/')
141 def configuration(request):
142     return render(request, 'training/config.html')
143
144 #entrenos basketball views
145 @login_required(login_url='login/')
146 def entrenoIB(request):
147     BBeI()
148     return render(request, 'trainina/entrenosB1.html')
```

Figura 58: Función de entreno en archivo Views.py

1. Se encontró que el sensor JSN-SR04T-2.0 funciona como una alternativa de receptor que no se ve afectado por la luz solar y que se puede utilizar en ambientes en donde se salpique o moje el dispositivo de tiempo de reacción.
2. Se logró encontrar con el modulo RFM69HCW una alternativa de comunicación entre el dispositivo esclavo y la Raspberry Pi 3. Haciendo que se eliminara por completo el PIC16F887, que se tenía de por medio de los dos dispositivos, en la versión anterior.
3. Se cumplió con el objetivo de crear rutinas para los deportes basquetball, box y softball. Las rutinas diseñadas funcionan con el conjunto de dispositivos esclavos y el dispositivo maestro.
4. Se logró modificar la pagina web, cambiando totalmente su estilo y forma de interacción con el usuario. Al igual que se logró implementar las rutinas creadas en Python a la página web para que el usuario pueda interactuar solamente con la página de entrenos.



## CAPÍTULO 15

---

### Recomendaciones

---

- Hubo algunos problemas para unir el programa del módulo de radio frecuencia con la aplicación web. Por lo que se aconseja probar con otro servidor que funciona similar a Djagno. Por ejemplo Flask, que también funciona con Python y es bastante completo para la creación de aplicaciones Web.



- 
- [1] J. R. C. Boch, «Diseño de dispositivo inalámbrico para la medición de tiempos de reacción de atletas ante un estímulo visual dado para entrenamientos en diferentes deportes», Tesis doct., Universidad del Valle de Guatemala, 2018.
  - [2] CogniFit, *Tiempo de respuesta*, Accedido por última vez el 10 de junio de 2019, 2016. dirección: <https://www.cognifit.com/es/habilidad-cognitiva/tiempo-de-respuesta>.
  - [3] CCM, *Protocolo de comunicación*, Accedido por última vez el 4 de junio de 2019, 2016. dirección: <https://es.ccm.net/contents/275-protocolo-de-comunicacion>.
  - [4] U. I. de Valencia., *Radiofrecuencia: ¿Qué es y cuáles son sus aplicaciones?*, Accedido por última vez el 5 de junio de 2019, Actualizado 21 de Marzo de 2018. dirección: <https://es.ccm.net/contents/275-protocolo-de-comunicacion>.
  - [5] S. Salas, «Todo sobre sistemas embebidos», en *Arquitectura, programación y diseño de aplicaciones prácticas con el PIC18F*, Universidad Peruana de Ciencias Aplicadas, 2017, págs. 110-113.
  - [6] Sparkfun, *Serial Peripheral Interface (SPI)*, Accedido por última vez el 24 de septiembre de 2019, dirección: <https://es.ccm.net/contents/275-protocolo-de-comunicacion>.
  - [7] R. Kuc, «The Digital Information Age», en *An Introduction to Electrical Engineering*, School of Engineering Applied Science Yale University, 2014, págs. 110-113.
  - [8] M. contributors, *Servidor web*, Accedido por última vez el 8 de mayo de 2020, 2017. dirección: [https://developer.mozilla.org/es/docs/Learn/Common\\_questions/Que\\_es\\_un\\_servidor\\_WEB](https://developer.mozilla.org/es/docs/Learn/Common_questions/Que_es_un_servidor_WEB).
  - [9] H. M. CO, *RFM69HCW ISM TRANSCEIVER MODULE v1.1*, Accedido por última vez el 15 de marzo de 2019, dirección: <https://cdn-shop.adafruit.com/product-files/3076/RFM69HCW-V1.1.pdf>.

- [10] A. L. System, *Adafruit Feather 32u4 Radio with RFM69HCW Module*, Accedido por última vez el 10 de marzo de 2019, dirección: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-32u4-radio-with-rfm69hcw-module.pdf?timestamp=1567794225>.
- [11] —, *Adafruit NeoPixel Überguide*, Accedido por última vez el 11 de marzo de 2019, dirección: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-neopixel-uberguide.pdf?timestamp=1568593423>.
- [12] B. J. B. Cifuentes, «Diseño e implementación de aplicación web orientada a la medición de tiempo de reacción en el deporte», Tesis doct., Universidad del Valle de Guatemala, 2018.

### 17.1. Piezas del diseño mecánico



Figura 59: Corte láser parte inferior



Figura 60: Corte láser parte en medio



Figura 61: Corte láser parte superior



Figura 62: Soporte placa principal



Figura 63: Soporte 1 placa secundaria



Figura 64: Soporte 2 placa secundaria

## 17.2. Conexiones



Figura 65: Conexión física de la Raspberry Pi 3 y del módulo RFM69HCW

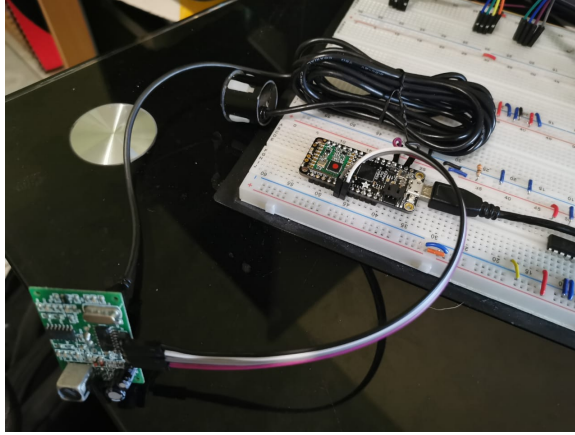


Figura 66: Montaje del sensor ultrasónico en el Feather 32u4 RFM69HCW

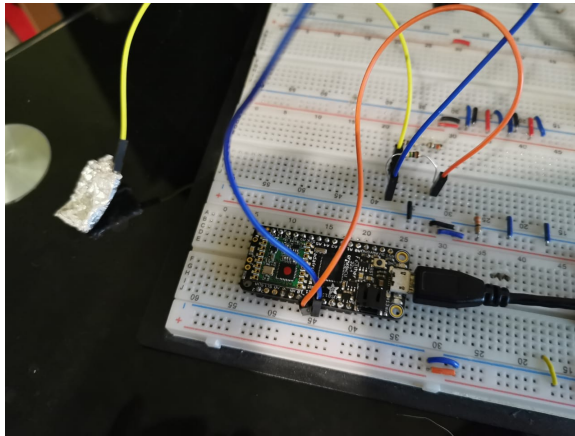


Figura 67: Montaje del sensor capacitivo en el Feather 32u4 RFM69HCW

### 17.3. Impresión placa usando la fresadora



Figura 68: Placa principal vista 1



Figura 69: Placa principal vista 2



Figura 70: Placa secundaria vista 1

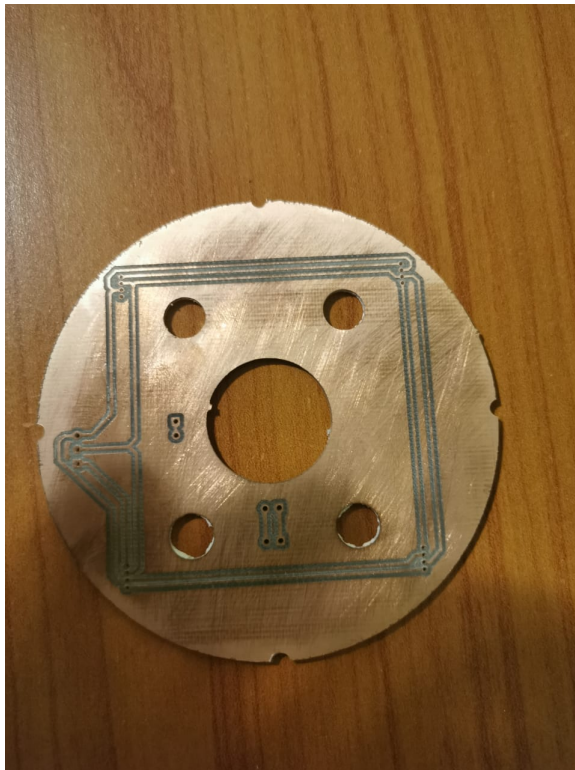


Figura 71: Placa secundaria vista 2

