

# IE PADEL TRAINER

**Jesús García Arranz**

Máster de Ingeniería de Telecomunicación  
Electrónica

**Aleix López Antón**

**Carlos Monzo Sánchez**

13-Junio-2019

Copyright © 2019 Jesús García Arranz.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>IE PADEL TRAINER</i>
<b>Nombre del autor:</b>	<i>Jesús García Arranz</i>
<b>Nombre del consultor/a:</b>	<i>Aleix López Antón</i>
<b>Nombre del PRA:</b>	<i>Carlos Monzo Sánchez</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2019
<b>Titulación::</b>	<i>Máster de Ingeniería de Telecomunicación</i>
<b>Área del Trabajo Final:</b>	<i>Electrónica</i>
<b>Idioma del trabajo:</b>	<i>Español Castellano</i>
<b>Palabras clave</b>	<i>Baliza, triangulación, entrenador</i>
<p><b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>La finalidad es obtener una máquina que cree el entrenamiento técnico y físico.</p> <p>Del estudio previo se extrae que son posibles los siguientes hitos:</p> <ol style="list-style-type: none"> <li>1.- Monitorizar temperatura, humedad ambiental</li> <li>2.- Monitorizar SpO2, pulsaciones en tiempo real</li> <li>3.- Posicionar una máquina respecto a 3 referencias.</li> <li>4.- Desplazar máquinas sin colisionar..</li> </ol> <p>La tecnificación del deporte llega al profesional y usuario amateur.</p> <p>Escuelas deportivas compran tecnología y máquinas que ayudan a profesores y preparadores.</p> <p>El deporte está muy especializado, entrenar al atleta, difiere del futbolista, el estudio se centra en el pádel, popularizado en España en 2001, y practicado en América y Europa.</p> <p>2 parejas de 2 jugadores compiten en una pista de 20m x 10m ancho, separados por red, estando "cerradas" 4 caras. Se golpea con palas, el objetivo es lograr 2 botes en campo contrario.</p> <p>La técnica enseña a golpear, y los físicos mejoran capacidad y resistencia.</p> <p>La técnica requiere lanzamientos repetitivos, efectos proporcionados a la bola mecánicamente.</p> <p>El físico recoge datos biométricos, analizados en tiempo real y ajusta tareas en función del momento.</p> <p>Desarrollar una máquina implica seguir la metodología:</p> <ol style="list-style-type: none"> <li>1.- Mecánica desplazamiento.</li> <li>2.- Mecanismo lanzador bolas.</li> <li>3.- Sensores.</li> <li>4.- Actuadores.</li> </ol>	

5.- Autolocalización.

6.- Monitorización biométrica.

7.- Análisis del juego.

8.- Inteligencia, respuestas.

9.- Integrarlo.

Conclusión, la miniaturización biométrica, balizas de bajo consumo, integración BLE y Wifi con microprocesadores de calidad permiten desarrollar un entrenador de pádel electrónico, de bajo coste, utilizado por profesionales y principiantes para mejorar su técnica y físico.

### **Abstract (in English, 250 words or less):**

The purpose is to obtain a machine that creates technical and physical training.

From the previous study it is extracted that the following milestones are possible:

1.- Monitor temperature, environmental humidity

2.- Monitor SpO<sub>2</sub>, pulsations in real time

3.- Position a machine with respect to 3 references.

4.- Move machines without colliding.

The technification of sport reaches the professional and amateur user.

Sports academies buy technology and machines that help teachers and trainers.

The sport is very specialized, training the athlete, differs from the soccer player, this study focuses on the paddle, popularized in Spain in 2001, paddle is practiced in America and Europe.

2 pairs of 2 players compete on a track of 20m x 10m wide, separated by network. The ball is hit with the racket, the objective is to get the point.

The technique teaches hitting, and physical trainer improve ability and endurance.

The technique requires repetitive throws, effects provided to the ball mechanically.

The trainer collects biometric data, analyzed in real time adjusts tasks according to the moment.

Developing a machine implies following the methodology:

1.- Mechanics of movement

2.- Ball throwing mechanism.

3.- Sensors.

4.- Actuators.

5.- Autolocation.

6.- Biometric monitoring.

7.- Analysis of the game.

8.- Intelligence, answers.

9.- Integrate it.

Conclusion, biometric miniaturization, low energy beacons, BLE integration and Wi-Fi with quality microprocessors allow the development of an electronic low-cost padel entrainer used by professionals and beginners to improve their technique and physic conditions.

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
1.5 Breve resumen de productos obtenidos .....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Estado de la técnica .....	6
2.1 Estado del arte .....	6
2.2 Conceptos básicos del posicionamiento con balizas BLE .....	7
2.3 Sensores corporales BLE .....	10
2.4 Intel Edison Sondas.....	14
2.5 Sistemas de desplazamiento autónomo.....	17
2.6 Sistemas de impulsión de bolas .....	18
2.7 Análisis del pádel .....	20
3. Prototipo de entrenador personal inteligente.....	23
3.0 Diseño físico.....	23
3.1 Diseño mecánico.....	24
3.2 Diseño electrónico.....	34
3.3 Posicionamiento con balizas BLE .....	48
3.4 Desarrollo de Inteligencia .....	57
4.-Interfase con usuario.....	69
4.1 Servidor WEB.....	69
4.2 NODE.JS.....	70
4.3 JAVA .....	70
5.- Integración (implementación técnica).....	71
5.0 Intel Edison Puesta en marcha y programación .....	71
5.1 Comunicación Sensores Deportista .....	74
5.2 Comunicación Sensores Internos.....	77
5.3 Posicionamiento de la máquina.....	81
5.4 Interfase de Usuario .....	86
6.- Construcción física.....	87
6.1 Implementación física: chasis.....	88
6.2 Motores de desplazamiento .....	90
6.3 Mecanismo de impulsión de bolas .....	93
6.4 Fuente de alimentación .....	98
6.5 Red Sensorial.....	99
6.5 Red Actuadores.....	100
6.7 Instalación del controlador.....	101
6.8 Herramientas auxiliares.....	101
6.9 Valoración Económica del trabajo .....	102
7. Conclusiones.....	105
8. Glosario .....	107
9. Bibliografía y recursos Web.....	110
9.1.- Bibliografía .....	110
9.2.- Recursos Web.....	112
10. Anexos .....	113
10.1 Anexo I Pseudocódigo.....	114
10.2 Anexo II Programa principal de control del Intel edison.....	121

10.3 Anexo III	Servidor nodejs .....	125
10.4 Anexo IV	Ibeacon y cálculo de posición .....	127
10.5 Anexo V	Programa cliente.....	130
10.6 Anexo VI	GNU Free License .....	133

## Lista de figuras

Figura 1: Sistema ILS.....	7
Figura 2: Trilateración.....	8
Figura 3: Canales BLE.....	10
Figura 4: Sensor BLE, despiece.....	11
Figura 5: Wearable Esquema conexión Corporal.....	12
Figura 6: Monitor deportivo.....	13
Figura 7: Arduino Controlador.....	14
Figura 8: Microordenador.....	14
Figura 9: Microordenador Intel Edison.....	15
Figura 10: Placa Kit Expansión Arduino.....	15
Figura 11: Sensor HC-SR04.....	16
Figura 12: Sensor Ritmo Cardíaco BLE.....	16
Figura 13: Ibeacon Sensor.....	16
Figura 14: NASA, Curiosity Rover.....	17
Figura 15: Pirate 4WD Robot.....	18
Figura 16: Lanzador por Resorte elástico.....	19
Figura 17: Lanzador por gas.....	19
Figura 18: Lanzador Aire Comprimido.....	19
Figura 19: Lanzabolos Comercial.....	20
Figura 20: Impulsor Discos Eléctricos.....	20
Figura 21: Imagen 21; Pádel Pista de Juego, dimensiones.....	21
Figura 22: Entrenamiento Pádel.....	21
Figura 23: Fases Diseño Prototipo.....	23
Figura 24: Diseño Físico.....	23
Figura 25: Diseño Componentes.....	23
Figura 26: Diseño mecánico.....	24
Figura 27: Desplazamiento Diseño.....	25
Figura 28: Rueda Motriz.....	26
Figura 29: Ruedas Equilibrio.....	26
Figura 30: Motor brushless.....	27
Figura 31: Hub Motor Wheel.....	28
Figura 32: Lanzador ONECONCEPT.....	29
Figura 33: Motor D.C. RS 795:.....	31
Figura 34: Tipos Efectos Bola.....	31
Figura 35: Fuente alimentación.....	32
Figura 36: Diseño electrónico.....	34
Figura 37: Intel Edison Bus.....	35
Figura 38: Placa Expansión Intel Arduino.....	36
Figura 39: Arduino Pin Out Diagram.....	36
Figura 40: Programación Fases de Implementación.....	38
Figura 41: Diseño sensorial.....	38
Figura 42: Red Sensorial.....	39
Figura 43: Sonda BLE Ritmo Cardíaco.....	40
Figura 44: Sónar animal.....	41
Figura 45: HC-SR04.....	42
Figura 46: Ultrasonic Timing Chart.....	43
Figura 47: JSR- SR04 Cabeza Sonda.....	43
Figura 48: Directividad Echo.....	44
Figura 49: Ibeacon BLE.....	45
Figura 50: NRF5 Esquema Básico.....	47
Figura 51: Puntos de corte 2 circunferencias.....	49
Figura 52: Cirunferencias sin corte.....	50

Figura 53: Ejes radicales .....	51
Figura 54: Driver WS55-220.....	52
Figura 55: Driver BLDC .....	54
Figura 56: uP, Driver, Motor.....	54
Figura 57: Driver Motor D.C. Lanzabolas .....	56
Figura 58: IA .....	57
Figura 59: Arduino Software.....	58
Figura 60: Movimiento .....	59
Figura 61: Rodar .....	59
Figura 62: Girar.....	61
Figura 63: Rueda Loca .....	63
Figura 64: Tracción .....	64
Figura 65: Control Maestro-Esclavo .....	64
Figura 66: Orientación, brújula .....	65
Figura 67: Distribución Ibeacons .....	66
Figura 68: Intensidad, tipos.....	67
Figura 69: Intensidad Efectos.....	67
Figura 70: Servidor Web, código node js.....	69
Figura 71: Servidor Web Plataformas .....	69
Figura 73: Ejemplo de servidor .....	70
Figura 72: Node.JS Logo .....	70
Figura 74: Ensamblando Chip y Placa base .....	71
Figura 75: Edison, instalando bajo Windows.....	72
Figura 76: IDE Arduino .....	73
Figura 77: Terminal PuTTY.....	73
Figura 78: Programas acceso a Edison.....	74
Figura 79: Sondas BLE Personal Trainer.....	74
Figura 80: Heart Rate Setup.....	76
Figura 81: 10us Echo .....	77
Figura 82: Conexión Edison Sondas SR04 .....	78
Figura 83: Edison Programando sensor distancia .....	79
Figura 84: LM36 PINOUT.....	80
Figura 85: Sonda Temperatura Conexión .....	80
Figura 86: Posicionamiento por RSSI, sondas .....	81
Figura 87: RDL51822 Detección Sondas.....	82
Figura 88: Interfase Usuario, navegador multiplataforma .....	86
Figura 89: Dimensiones Chásis.....	87
Figura 90: Policarbonato Compacto.....	88
Figura 91: Montaje Fijaciones.....	88
Figura 92: Montaje ruedas.....	89
Figura 93: Chásis montado.....	89
Figura 94: Esquema eléctrico Desplazamiento .....	90
Figura 95: Check Desplazamiento PWM.....	90
Figura 96: Desplazamiento Driver nomenclatura .....	91
Figura 97: Lanzamiento Drivers .....	93
Figura 98: Lanzador Bajo coste .....	94
Figura 99: Lanzador despiece .....	94
Figura 100: Lanzador mecanismo discos .....	95
Figura 101: Check Motores Lanzamiento .....	96
Figura 102: Filtro Paso Bajo .....	97
Figura 103: Fuentes alimentación .....	98
Figura 104: Personal Trainer terminado.....	99
Figura 105: Detalle Controlador, Panel Control, conexiones .....	100
Figura 106: Herramientas auxiliares .....	101



## Lista de Tablas

Tabla 1: BLE características .....	11
Tabla 2: Características físicas Personal Trainer .....	25
Tabla 3: Hub Motor Wheel, características .....	28
Tabla 4: Lanzador características Mínimas .....	29
Tabla 5: Características Unidad Elegida .....	30
Tabla 6: Características del motor seleccionado RS 795: .....	30
Tabla 7: Características Fuente Principal.....	32
Tabla 8: Fuente 18650-10SP2 Características.....	33
Tabla 9: Fuente Alimentación Secundaria (Controlador) .....	33
Tabla 10: XP4000 Características .....	33
Tabla 11: Requisitos del uProcesador. ....	34
Tabla 12: Características Intel Edison: .....	35
Tabla 13: Asignación Pines Placa Kit Arduino.....	37
Tabla 14: Sonda BLE Ritmo Cardíaco, características .....	40
Tabla 15: Características sensores de distancia.....	41
Tabla 16: HC-SR04 características técnicas .....	42
Tabla 17: JSN-SR04T características técnicas .....	44
Tabla 18: Características principales BLE .....	45
Tabla 19: NRF5 Características .....	46
Tabla 20: UUID descriptor.....	46
Tabla 21: Parámetros de Potencia nRF 51822 .....	46
Tabla 22: Driver Motores desplazamiento, requisitos.....	53
Tabla 23: 15A 500W DC12V BLDC Driver Board with Hall.....	53
Tabla 24: Driver Motores desplazamiento, requisitos.....	55
Tabla 25: E15A1AA700381DC Driver Board with Display .....	56
Tabla 26: Giro .....	63
Tabla 27: Movimiento rectilíneo (Adelante, Atrás) .....	63
Tabla 28: RSSI Experimental 3 Ibeacons .....	82
Tabla 29: Valoración Económica Prototipo .....	102
Tabla 30: Valoración económica Producción Real .....	103
Tabla 31: Estudio y análisis de Mercado .....	104

## Lista de Pseudocódigos

Pseudocódigo 1: Programa Principal .....	68
Pseudocódigo 2: Subrutina Distancia.....	114
Pseudocódigo 3: Subrutina Automático.....	114
Pseudocódigo 4: Subrutina Lanza .....	115
Pseudocódigo 5: Subrutina Retrocede.....	115
Pseudocódigo 6: Subrutina Avanza.....	116
Pseudocódigo 7: Subrutina Temperatura .....	116
Pseudocódigo 8: Subrutina Ritmo Cardíaco HRM .....	117
Pseudocódigo 9: Programa Posición .....	117
Pseudocódigo 10: Subrutina Giros .....	118
Pseudocódigo 11: Subrutina Condiciones.....	119
Pseudocódigo 12: Subrutina comunicación .....	120
Pseudocódigo 13: Programa Servidor + Comunicación.....	120

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

La actual tecnificación del deporte no llega sólo a los profesionales sino que gracias a la globalización cualquier hito tecnológico es aplicado de forma inmediata al usuario en general. Hoy en día la gente ya no sale a correr sino a hacer running, registrando los datos por geolocalización para conocer su trayectoria, los tiempos empleados y de ahí su velocidad, esfuerzo....

De igual forma se preparan ciclistas aficionados con máquinas de rodillo con sensores de esfuerzo.

Del estudio de los datos anteriores se pueden obtener conclusiones de mejora.

Por otro lado existen sensores biométricos que pueden indicar la temperatura corporal, ritmo cardíaco..

Existen escuelas profesionales deportivas e incluso centros de alto rendimiento donde la tecnología y las máquinas ayudan a los profesores y preparadores físicos.

Y si dando un paso más en la aplicación de esta nueva tecnología se pudiera **obtener una máquina que no ayude al preparador sino que sea ella la que prepare, la que cree el entrenamiento técnico y físico.**

De estudios analizados se puede extraer que ya son posibles los siguientes hitos:

- 1.- monitorizar temperatura, humedad y presión ambiental
- 2.- Monitorizar temperatura, SpO2, Ritmo cardíaco y presión arterial en tiempo real
- 3.- Conocer la posición de una máquina respecto a un sistema con 3 referencias
- 4.- Hacer que una máquina se desplace sin colisionar dentro de un recinto.

El pádel es un deporte que se popularizó en España a principios del siglo XXI procedente de Argentina y que hoy en día cuenta se practica además en toda Iberoamérica, Costa Oeste de Estados Unidos, Italia, Portugal, Suecia..

2 parejas de 2 jugadores se enfrentan entre sí en una superficie rectangular (pista, cancha) de 20m x 10m ancho, con 2 campos de 10x 10m separados por una red de 1m de altura, estando la pista "cerrada" a 4 caras con diferentes alturas. Se juega con palas sintéticas que golpean 1 bola y el objetivo es lograr 2 botes en campo contrario.

Los preparadores técnicos de pádel enseñan a los jugadores a golpear de derecha, de revés, de volea, saques, restos, bandejas, víboras, smash (remate) y globos.

Mientras los preparadores físicos ponen a punto la capacidad pulmonar, la resistencia y el tono, la flexibilidad y los reflejos del deportista, de acuerdo a las aptitudes del sujeto preparado.

En la actualidad sólo existen máquinas "tontas", básicamente carritos con ruedas que un entrenador desplaza hasta una posición fija, y se dedican a impulsar bolas hasta que se las para.

El humano realiza el análisis del deportista, toma todas las decisiones y proporciona los desplamientos del carrito de bolas.

La relevancia pasa por la moda de este deporte, se juega con 6 años y con 60, no hace falta ser deportista de élite, pero sí una pequeña base técnica y entrenar.

Se busca realizar una máquina inteligente que reemplace al entrenador del pádel.

## 1.2 Objetivos del Trabajo

Este trabajo pretende alcanzar los objetivos siguientes:

- a.- Estudiar la viabilidad técnica de obtener una máquina que entrene pádel en lugar de un humano, para ello se la dotará de sentidos, inteligencia y capacidad de respuesta.
- b.- Construir un prototipo a escala real que permita verificar experimentalmente y no en simulador que las hipótesis de partida son correctas, y que es posible crear la máquina.
- c.- Estudiar la viabilidad económica del prototipo.
- d.- Analizar la viabilidad empresarial para producir las máquinas y vender en el mercado en sustitución de los actuales modelos “tontos”.

La tarea de entrenamiento técnico mediante una máquina requiere que ésta sea capaz de lanzar de forma repetitiva, variando la velocidad, altura, dirección y creando efectos (liftado, cortado,...) que pueden proporcionarse a una bola mediante diferentes procedimientos mecánicos.

La tarea de entrenamiento físico requiere recoger los datos biométricos del deportista, analizarlos en tiempo real y crear tareas de lanzamiento que fuercen al deportista a realizar esfuerzos (desplazamientos, saltos, golpeo) de diferente intensidad en función del momento del entrenamiento y de sus constantes vitales en ese instante.

Los objetivos generales se desglosarán en los siguientes parciales:

- 1.- Diseñar un sistema de desplazamiento eléctrico.
- 2.- Seleccionar e incorporar un sistema básico de lanzamiento de bolas.
- 3.- Diseñar una red sensorial para la máquina.
- 4.- Diseñar un sistema de actuadores que gobierne los motores de desplazamiento e impulsión
- 5.- Diseñar un sistema de autolocalización de la máquina.
- 6.- Diseñar un sistema de monitorización biométrica del sujeto entrenado.
- 7.- Diseñar un sistema de inteligencia autónoma, que analice los datos y ejecute respuestas.
- 8.- Análisis básico de un entrenamiento técnico y físico del pádel.
- 9.- Integrarlo todo en una solución única que actúe como un auténtico entrenador de pádel.

## 1.3 Enfoque y método seguido

Se va a desarrollar un nuevo producto rompiendo con la tendencia actual en el mercado que desarrolla y vende carritos con lanzador de bolas.

Lo que se pretende es algo absolutamente nuevo, que sondee variables del entorno, analice datos en tiempo real y de respuestas en función de ellas.

Que sea capaz de desplazarse de forma autónoma (y permita modo manual) pero siempre motorizado sin tracción humana.

La única técnica que se va a aprovechar para acortar la complejidad del proyecto es el lanzamiento, la impulsión en sí. Se justifica porque los últimos 20 años la industria y otros trabajos universitarios han focalizado el esfuerzo únicamente en el lanzamiento por lo tanto está suficientemente estudiado y evolucionado como para integrarlo de forma natural en este trabajo y dedicar el análisis al resto de funciones, mucho más actuales e innovadoras.

Parece esta estrategia la idónea puesto que la documentación sobre los diferentes campos es extensa, la disponibilidad física es asequible, y el objetivo de integrarlo todo es factible por el conocimiento actual de toda esta tecnología.

El gran reto es ver si es posible integrarlo y su viabilidad técnica y comercial y por ello lo mejor es obtener un prototipo real en lugar de simulaciones informáticas, que podrían resultar igual de caras y menos prácticas en las conclusiones obtenidas.

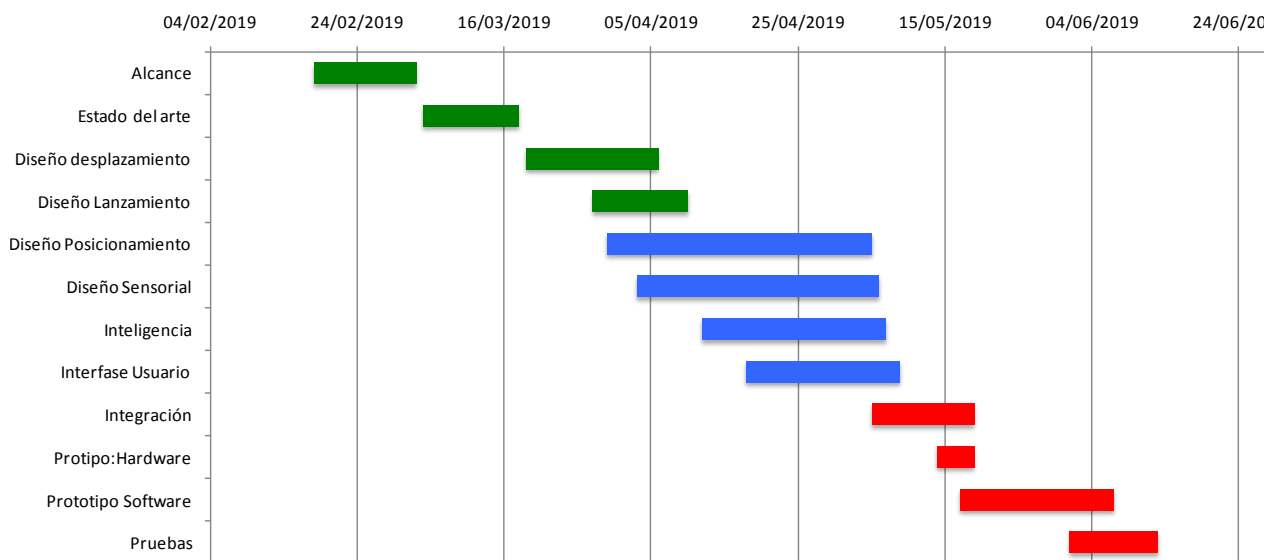
## 1.4 Planificación del Trabajo

Como recursos principales se utilizará abundante documentación bibliográfica, gran cantidad de dedicación medida en tiempo y esfuerzo, ordenadores y software, y todos los componentes físicos necesarios para construir el prototipo, así como un laboratorio de electrónica y mecánica completamente equipado.

Como el conocimiento y el tiempo son recursos escasos se ha intentado optimizar el tiempo y seguir la planificación temporal establecida en las PECs.

Así en la PEC2 se definió y concretó el estado de la técnica y del arte, para en la PEC3 entregar en la memoria completamente desarrollado el diseño en general, la interfase de usuario, la integración y la construcción del prototipo.

Diagrama de Gantt del TFM



Nombre de la tarea	Fecha de inicio	Fecha final	Duración (días)
Alcance	18/02/2019	04/03/2019	14
Estado del arte	05/03/2019	18/03/2019	13
Diseño desplazamiento	19/03/2019	06/04/2019	18
Diseño Lanzamiento	28/03/2019	10/04/2019	13
Diseño Posicionamiento	30/03/2019	05/05/2019	36
Diseño Sensorial	03/04/2019	06/05/2019	33
Inteligencia	12/04/2019	07/05/2019	25
Interfase Usuario	18/04/2019	09/05/2019	21
Integración	05/05/2019	19/05/2019	14
Protipo:Hardware	14/05/2019	19/05/2019	5
Prototipo Software	17/05/2019	07/06/2019	21
Pruebas	01/06/2019	13/06/2019	12

## 1.5 Breve resumen de productos obtenidos

- 1.- Diseño de un sistema de desplazamiento
- 2.- Selección de un sistema de lanzamiento existente en el mercado.
- 3.- Diseño de un sistema de posicionamiento autónomo
- 4.- Red sensorial externa
- 5.- Red sensorial interna
- 6.- Inteligencia, análisis
- 7.- Sistema de actuadores
- 8.- Interfase con usuario multiplataforma
- 9.- Integración de los puntos anteriores
- 10.- Construcción de prototipo.

## 1.6 Breve descripción de los otros capítulos de la memoria

### Capítulo 2 Estado de la técnica

Versa sobre el estado de la técnica actual, su evolución pasada y la tendencia futura que muestra.

Se presentan los conceptos básicos y definiciones de diferentes elementos simples y complejos que se utilizan en el proyecto.

Sirve de base para el planteamiento y la justificación del Trabajo.

### Capítulo 3.- Prototipo de pádel trainer

Abarca toda la fase de diseño de la máquina, desde la parte más física hasta la lógica pasando por las eléctricas, electrónicas, sensoriales, actuadores y análisis (inteligencia).

Desarrolla el porqué, la necesidad y características mínimas de cada recurso.

Determina lo que se puede utilizar y lo que no, y las justificaciones.

Además selecciona las mismas realimentando cíclicamente con las anteriores y las siguientes etapas de diseño, pues existe interacción e interdependencia de todas ellas.

### Capítulo 4.- Interfase con usuario

Diseño de la aplicación que mostrará los datos y recogerá parámetros de uso por parte del propietario de la máquina, que no de la persona entrenada (salvo que ambas sean la misma).

Debe permitir que el operador humano pueda conectarse desde cualquier plataforma actual, smartphone, tableta, portátil, PC, videoconsola..., a la máquina para poder interactuar con ella cuando sea necesario.

El programa mediante una interfase gráfica amable, mostrará diferentes parámetros en tiempo real como las pulsaciones, la temperatura, la posición y permitirá parar la máquina, iniciar programas inteligentes (automáticos) y además control manual de operador.

### Capítulo 5.- Integración (Implementación técnica)

Modelo el prototipo del entrenador personal, basándonos en Procesador Intel Edison que cumple todas las características especificadas en la etapa de diseño y debe comunicarse con las sondas externas e internas, siguiendo la secuencia lógica de la misma, empezando por la parte física del Intel, su puesta en marcha, su preparación, y su programación.

## Capítulo 6- Construcción física del pádel trainer

Aborda todo el proceso de ensamblaje de piezas físicas y la programación del procesador así como pequeñas pruebas y test de respuesta y funcionamiento del protipo

Viabilidad técnica y económica

Estudio de mercado y análisis de datos obtenidos al elaborar este trabajo que permite emitir conclusiones sobre la viabilidad a nivel técnico del prototipo y a nivel económico empresarial de producto real.

## Capítulo 7.- Conclusiones

Recoge un resumen de ideas finales obtenidas en base a las experiencias e impresiones obtenidas durante la elaboración del TFM.

Viabilidad del proyecto, mejoras futuras, problemas surgidos, soluciones, posibles alternativas y valoración del estudio.

## Capítulo 8.- Glosario

Palabras clave y acrónimos utilizados y relevantes

## Capítulo 9.- Bibliografía

Cita de libros, artículos y Webs visitadas para realizar el TFM

## Capítulo 10.- Anexos

Incluye Código, Pseudocódigo, tablas de características OEM, licencia de uso.

## 2. Estado de la técnica

### 2.1 Estado del arte

El entorno social y cultural actual es un medio cambiante, en constante evolución, una sociedad de consumo rápido, con una necesidad de alta densidad de información, instantánea, hábitos de vida asociados al imparable desarrollo tecnológico, donde conceptos como la obsolescencia programada forman parte de una nomenclatura conocida por el usuario final, muy lejos de ser un término técnico o estrictamente científico.

Los teléfonos inteligentes son capaces de leer, editar, crear cualquier aplicación antes sólo disponible para ordenadores, e incluso superan a éstos en el número de aplicaciones y usuarios disponibles, y mediante los mismos, los humanos estamos "on line", y permanentemente conectados a una red global.

Pero los avances técnicos se producen también en medios de desplazamiento antes con tracción humana y que ahora incorporan ayuda eléctrica, bicicletas asistidas, patines eléctricos y "hoverboards" inundan las calles, carriles bici y aceras.

De forma consciente o sin darnos cuenta la necesidad de tener esta tecnología en todo momento a llevado a que la incorporemos también cuando hacemos deporte, otrora un tiempo dedicado a "desconectar del mundo", reparar el cuerpo y sanar la mente y sin embargo hoy en día nadie sale a correr, sino a hacer running y para ello es preciso llevar un medidor de ritmo cardiaco, de temperatura y seguir mediante el GPS el recorrido efectuado, incluyendo los "tiempos" logrado, los pasos dados y las Calorías quemadas.

Dependiendo de la mentalidad propia de país se desarrollan diferentes apuestas por el futuro tecnológico que como ya vemos es paralelo al desarrollo social, Japón por ejemplo ha apostado por el desarrollo de robots humanoides que serán destinados a cubrir casi cualquier necesidad de una población que es hoy en día la más envejecida del mundo.[1]

España tiene una población que en 15 años será la más longeva y se colocará a la cabeza, las peculiaridades mediterráneas hacen que la gente disfrute del deporte a cualquier edad, supeditados como es normal a las limitaciones propias. La fisioterapia en España ha adquirido gran importancia y reconocimiento a la par que los preparadores físicos y los masajistas, cada uno especializado en un campo.

¿Es posible también que la tecnología ayude a la preparación y la reparación de nuestro físico?

El presente trabajo pretende utilizar tecnología de desplazamiento similar a la de los hoverboards, controladores de bajo coste al alcance de cualquier persona, y técnicas de posicionamiento sencillas para obtener un preparador personal, un entrenador que en este caso dedicaremos al deporte del pádel, de amplia práctica en los últimos años.



## 2.2 Conceptos básicos del posicionamiento con balizas BLE

Las **balizas** son comúnmente aceptadas como marcas o señales que se colocan para advertir zonas peligrosas. [2]

La RAE la define como "Señal fija o móvil que se pone de marca para indicar lugares peligrosos o para orientación del tráfico marítimo, aéreo y terrestre".

Son comunes las balizas ópticas que advierten de un peligro, como una altura elevada que no se distingue visualmente en determinadas condiciones atmosféricas o de noche, y también señales electromagnéticas como las usadas en navegación aérea.

Según AESA, Asociación Española de Seguridad Aérea, una baliza es un "Objeto **señalizador** fijo o móvil, expuesto sobre el nivel del terreno para indicar un obstáculo y/o lugares peligrosos, orientación de la aeronave o trazar un límite".

El uso de radio balizas en navegación aérea y marítima consiste en una estación de radio equipada con una antena no direccional, da direcciones vía radio. Se pueden entender como sistemas de **Orientación**, llegando a emplearse como localizadores junto con brújulas de radio.

Existen radiobalizas de emergencia, transmisores de radio activados en caso de emergencia permite localizar personas o naves enviando señales con datos recogidos por satélites COSPAS-SARSAT que **calcula la posición** y activan la emergencia. Pueden ser portátiles.

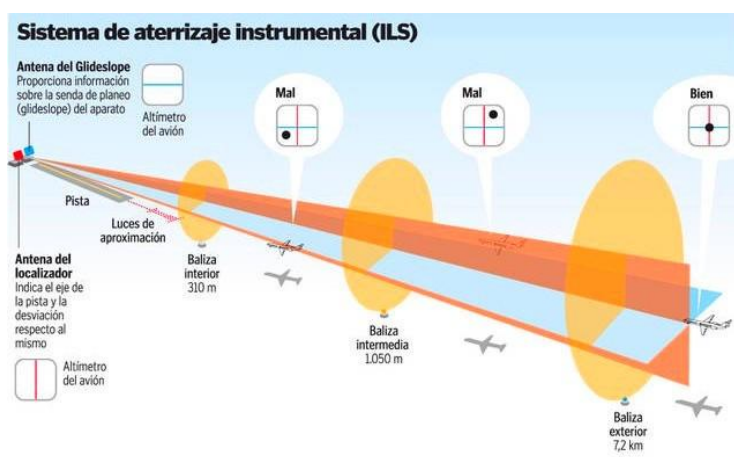


Figura 1: Sistema ILS

La navegación aérea utiliza diferentes **sistemas de posicionamiento** para ayudar al aterrizaje y despegue de aeronaves, sin duda el más completo es el ILS, este sistema tiene 3 balizas complementarias: OM (Outer Marker), la más alejada de la pista de aterrizaje que manda código morse dos rayas por segundo. Cuando la aeronave la sobrevuela el piloto escuchará dos tonos largos, y se enciende el testigo de baliza exterior, de color azul. La MM (Middle Marker) señala la aproximación al mínimo enviando una señal en código Morse punto raya que en cabina enciende el testigo de baliza ámbar. IM (Inner Marker) umbral de manda señal Morse de dos puntos y en recepción se ilumina el testigo de baliza blanco. Gracias a este complemento, el sistema ILS permite aterrizar en condiciones meteorológicas adversas.

Pero la navegación aérea o la marítima no son las únicas que utilizan balizas, los más modernos medios de transporte terrestre también las emplean, por ejemplo ADIF (ASFA Vía Digital) las define como "Elemento del sistema instalado en la vía, utilizado para la transmisión de información puntual de la vía al tren".

En esta última se añade un concepto muy interesante, **transmisión de información**.

Así pues hemos visto que los propósitos de una baliza han ido evolucionando desde advertir, orientar, posicionar e informar.

En este contexto Apple decide hace una década desarrollar la idea del **ibeacon** de la que NO es propietaria pero su presencia global a través de los dispositivos iPhone y iPad ha permitido su difusión a nivel mundial en los últimos 5 años.

Su idea, que sirvan de guía para la gente en estadios, museos y otros entornos de interior donde la tecnología GPS presenta problemas de cobertura.

Hoy en día están disponibles también para dispositivos Android.

Ibeacon es una baliza capaz de transmitir información vía electromagnética a través del estándar BLE, en la banda de 2.4Ghz, permite velocidades de 1Mbps y conexión cifrada AES 128 bits pero la idea fundamental es LE, el bajo consumo, lo que minimiza las fuentes de energía y así el tamaño de estos dispositivos.

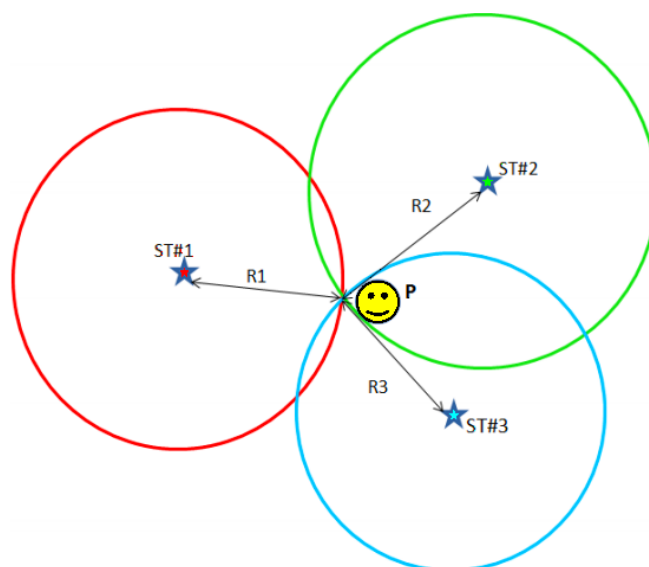
Ibeacon permite enviar UUID, Identificador Único Universal, además de la energía transmitida y ha añadido datos de temperatura, movimiento, acelerómetros, presión, y otras señales biométricas.

**UUID** 16 bytes (128 bits) tiene 36 caracteres, 5 grupos de 32 dígitos hexadecimales separados por un guión: **550e8400-e29b-41d4-a716-446655440000**

Se puede usar como un identificador específico para identificar el mismo objeto en diferentes contextos.

Ideales para usar en IPS, **sistema de posicionamiento en interiores** pues son capaces de enviar notificaciones push a dispositivos compatibles.

Un Ibeacon puede detectar otros ibeacons en su vecindario, es decir próximos a ellos porque detecta las tramas que otras balizas están emitiendo. A partir de ahí se pueden lanzar eventos (aplicaciones), adquirir información y procesarla, o determinar posicionamiento.



**Figura 2: Trilateración**

El posicionamiento en los ibeacons se basa en **TriLateración**.

Necesitamos conocer las coordenadas de 3 puntos "fijos" que se usaran como sistema de

referencia y la distancia del objetivo a cada uno de los 3 puntos.

La distancia se determinará a partir de la transmisión del RSSI, indicativo de fuerza de la señal recibida, que básicamente nos dice con qué fuerza (nivel de energía) se ha enviado la señal, y como nuestro sistema detecta con qué "fuerza" la hemos recibido, podemos con la diferencia de fuerzas determinar a qué distancia se encuentra.

De esta forma analizando la señal de 3 ibeacons (usados como referencia de coordenadas) podremos determinar la posición relativa del receptor respecto a ellos.

Con 1 baliza tendremos una distancia, con 2 balizas 2 posibles puntos respecto a ellas (corte de 2 circunferencias) y con 3 una posición única.

Hay técnicas que permitirían usar el método de **Ángulo de llegada** (AoA) que podría ser más preciso puesto que una dirección de la señal de radio es menos afectada por obstáculos que el RSSI pero habría que usar antenas más avanzadas, mayor gasto energético y algoritmos más complejos.

La precisión mediante RSSI es de 1m para distancias de hasta 30m.

Una variante de ibeacon es **eddystone** que fue lanzado por google en 2015. Básicamente sigue los mismos principios pero introduce más versatilidad en sus configuraciones, direccionamiento, capacidad de integración de sensores y de transmisión de datos. Así Eddystone puede mandar datos de temperatura, humedad, aceleración, nivel de batería.

UID, ID único es similar al UUID, con 2 partes el nombre de espacio y la Instancia.

URL, un identificador de recursos uniforme, comprimido para enlazar a la WEB.

TLM para envío de telemetría y EID que aporta medidas de seguridad adicionales.

Utilizaremos un sistema de coordenadas cartesianas bidimensional.

## 2.3 Sensores corporales BLE

### 2.3.1 BLE [3]

BLE es un protocolo denominado Bluetooth Low Energy BLE que fue adoptado por Bluetooth SIG a partir del proyecto Wibree de Nokia.

Hace muy sencillo establecer comunicación entre pequeños sensores y aplicaciones multiplataforma (fijas o móviles) y sus sistemas operativos sin necesidad de certificaciones MFI en el caso del iOS de Apple.

Es compatible con Android +4.3, iOS +5, Windows +8, Linux BlueZ +4.93.....

**Generic Access Profile (GAP)** maneja las conexiones, proporciona visibilidad y permite o deniega que 2 dispositivos interactúen. Asigna roles, siendo periférico un pequeño sensor y siendo central un dispositivo con mayor capacidad de proceso (teléfono inteligente, pc,..)

La transmisión básica se denomina **Advertising Data Payload**, se realiza continuamente como Broadcast, unidireccionalmente, a intervalos regulares por el periférico para que el central detecte su presencia. El central puede buscar mediante "**Scan Response Payload**" dispositivos e información extra (nombre, características...)

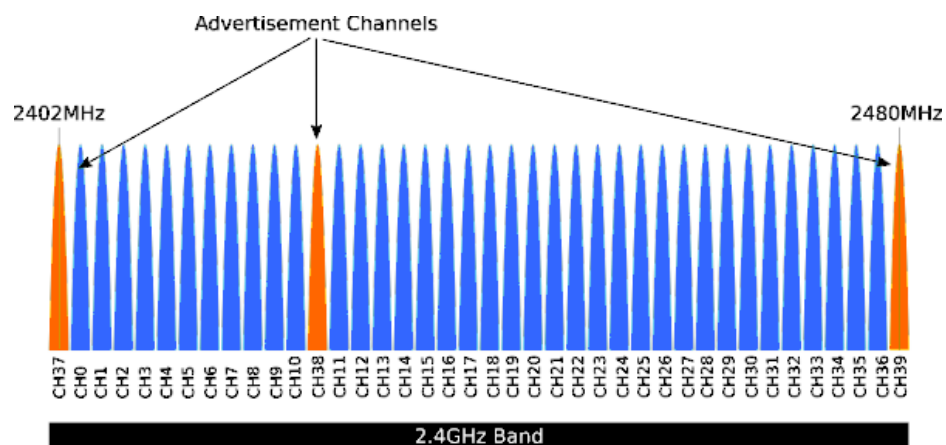


Figura 3: Canales BLE

Una vez que se establece **conexión** entre 1 periférico y un solo central se usa el **modo GATT (Generic Attribute Profile)**, bidireccional y exclusiva (para el periférico), que mediante protocolo ATT define como se comunicarán los 2 dispositivos usando una tabla e identificadores de 16 bits, modelo **Cliente** (Central Maestro) /**Servidor** (Periférico Esclavo).

El periférico NO tiene un perfil propio sino que existe una colección de Servicios predefinida por fabricantes o especificada por el Bluetooth SIG, que se pueden combinar en el periférico.

Cada Servicio posee un ID único (UUID) de 16 o 128bits, y divide los datos en entidades lógicas que poseen una o varias "características", o pedazos de datos de un único tipo. Cada una de ellas tiene 16 o 128 bits y tienen su propio UUID.

Distribución de audio avanzado (A2DP), Control remoto de audio y vídeo (AVRCP), Básico de imagen (BIP); Básico de impresión (BPP), Telefonía inalámbrica (CTP), Red de marcado

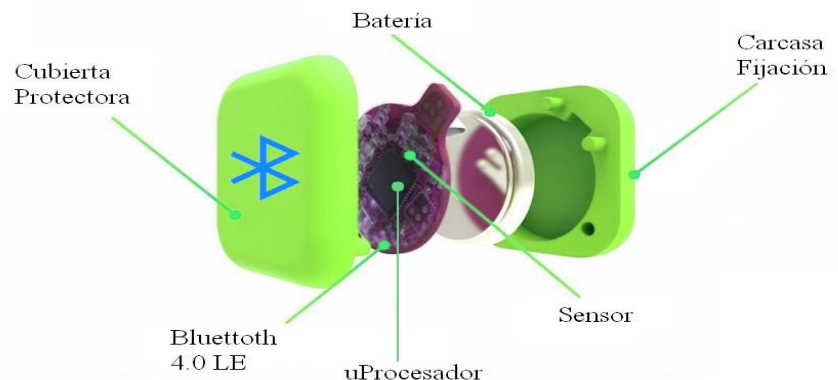
(DUNP), Fax (FAX), Transferencia de archivos (FTP), Acceso genérico (GAP), Genérico de intercambio de objetos (GOEP), Sustitución de cable de copia impresa (HCRP), Manos libres (HFP), Dispositivo de interfaz humana (HID), Auricular (HSP), Intercomunicador (IP), Acceso LAN (LAP), Objeto push (OPP), Redes de área personal (PAN), Acceso SIM (SAP), Aplicación de descubrimiento de servicio (SDAP), Sincronización (SP) con administrador de información personal (abreviado PIM), Puerto de serie (SPP).

### 2.3.2 INTRODUCCION AL SENSOR BLE [4]

Gracias al bajo consumo de los dispositivos BLE se ha hecho posible la miniaturización del sistema de captación (sensor), transductor, fuente de alimentación y sistema de envío, generalizándose a muy diversas áreas con innumerables aplicaciones ya sean civiles como los campos de cultivo, áreas de incendios, drones, robótica, domótica o militares como localización, seguimiento, detección...

Los Dispositivos sensoriales BLE están de forma general compuestos de las siguientes partes:

- 1.- Sensor
- 2.- Convertidor A/D
- 3.- Microcontrolador
- 4.- Batería
- 5.- Transceptor



**Figura 4: Sensor BLE, despiece**

La técnica y la tecnología aplicada a este tipo de dispositivos pretende salvaguardar siempre el principio de bajo consumo y el uso de bandas de frecuencias libres, en este caso la asignada pertenece al UHF en el rango de los 2.4Ghz, banda Mundial sin licencia ISM ( Industrial, Scientific and Medical). [4]

El nº máximo de canales es 79 y cada país puede restringirlo, así en España es de 23 canales.

Características principales	BLE
Banda	2.4 GHz
Modulación	GFSK
Velocidad	1.0 Mbps
Rango máximo	100 m
Consumo Corriente máx	15 mA
Consumo en Espera	< 2 uA
Topología de Red	Dispersión

**Tabla 1: BLE caracterísiticas**

### 2.3.3 LOS WEARABLES [6]

Pero las aplicaciones que más han evolucionado son sin duda los relacionados directamente con el cuerpo humano, la medicina y el deporte. En este sentido hay que apuntar que el catalizador ha sido la expansión de los smartphones, capaces de comunicarse con los sensores BLE y de soportar aplicaciones que permiten monitorizar los pasos dados, las pulsaciones, la temperatura y la humedad, la tensión, la aceleración e incluso el intercambio de oxígeno y ritmo respiratorio.

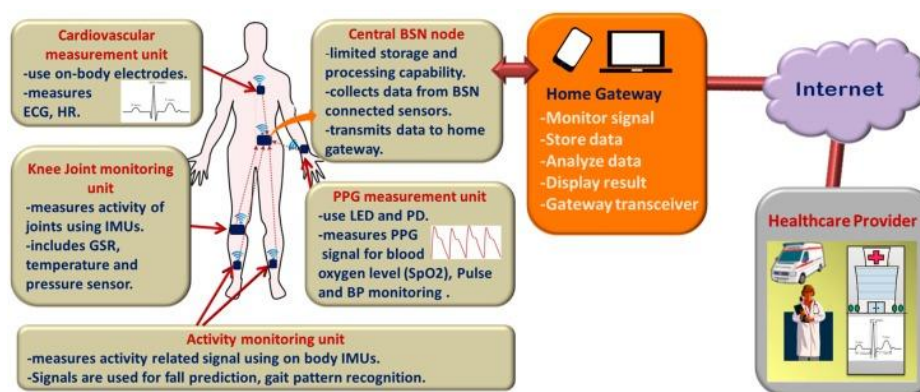


Figura 5: Wearable Esquema conexión Corporal

Los sensores deportivos llamados wearables (que los puedes vestir) no sólo se han desarrollado tecnológicamente, prestaciones, capacidad, compatibilidad sino que hoy forman parte de la misma moda siendo un complemento imprescindible como lo es una gafa de sol, un bolso...

Tienen características específicas respecto a otros sensores BLE, en gran medida especificadas por normativa y legislación por su uso en humanos:

Ergonómicos (ligeros, cómodos, fáciles de poner y quitar, difíciles de perder).

Médicos , Materiales no tóxicos, hipoalergénicos, no intrusivos ni invasivos.

Químicos, químicamente inertes.

Moda, diseños desde muy llamativos a absolutamente invisibles, pero diseño, diseño y diseño.

### 2.3.4 CONTEXTO SOCIAL

Hace 30 años no había teléfonos inteligentes ni wearables y uno salía a correr con un chándal y unas deportivas que también usaba para ir a jugar al fútbol, al tenis o al baloncesto.

Hoy en día no se corre, se hace footing o running y hay calzado deportivo para cada una de esas modalidades, botas de diferentes suelas según el fútbol se practique en hierba natural o artificial, en campo de tierra o en pista interior (indoor) pero ninguno tiene que ver con las zapatillas de jugar al tenis que cuentan con variantes para agarrar en arcilla, en hierba o en pista dura.

Los cambios pues no son sólo tecnológicos sino también sociales y ahora no hace falta pertenecer a un club deportivo para hacer ejercicio e conjunto, basta un grupo de whatsapp al que se sube la ruta que has hecho, el paso por Km, las calorías, los pasos.....

Y es que estar a la última hoy en día es un concepto global, vestir + conocimiento + salud (healthy life) + tecnología + notoriedad.

Y un sensor BLE es tecnología sí, pero es moda, vía mensajería electrónica es notoriedad y aporta el conocimiento sobre uno mismo para vigilar permanentemente (monitorizar) la salud.

Monitorizar la salud hace apenas una década se realizaba en deportistas de élite como los ciclistas a los que se les medía el "esfuerzo" e indicaba si podían subirlo un poquito más, hoy en día se pone al alcance de cualquier usuario.

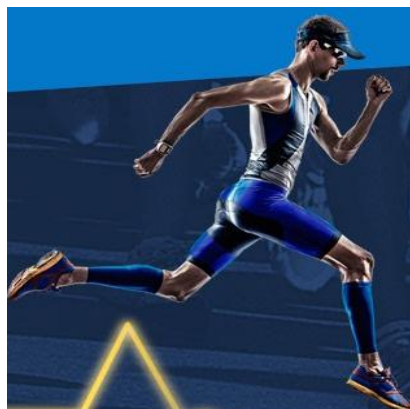


Figura 6: Monitor deportivo

El último factor no contado hasta ahora de los sensores BLE, su precio, y es que la producción es ahora a nivel mundial, todo está conectado, no se lanza un producto para una región sino para un "sector específico pero a escala mundial", de forma que la miniaturización, la compatibilidad y la producción masiva permite unos costes muy bajos lo que unido a la alta competencia en el mercado productor proporciona unos precios de venta asequibles (salvo en casos claros de diferenciación de marca, pero esto sería objeto de otro análisis y de otra especialidad de TFM).

Un apunte social final, año 2019 lo importante no es tenerlo sino **Mostrarlo**.

### 2.3.5 EVOLUCION [7]

Los sensores BLE corporales están rápidamente evolucionando gracias a los desarrollos de nuevos materiales aplicables a la industria textil, al cada vez menor consumo de energía de los dispositivos, y al desarrollo de técnicas de captación de energía limpia externa (sol, microondas, movimiento, temperatura corporal...)

En pocos años los tejidos inteligentes incorporarán electrodos secos, tantos como deseemos capaces de ser un captador continuo de todos los parámetros corporales ya indicados, temperatura, presión, ritmo cardíaco y respiratorio, humedad, SPO2 a los que sin duda añadiremos los neuronales, en ese contexto los sensores corporales ahora individuales serán "redes sensoriales" y los tejidos se auto adaptarán a las condiciones que el individuo necesite, aportando frío o calor, cambiando de color, protección UV, antibacterianos, con memoria de forma, o regulando la distribución de aromas, medicamentos, etc.

Hoy en día ya se estudian y clasifican:

- I.- Pasivo: Independiente del entorno exterior ,no modifica su característica (sensor).
- II.- Activo: Actúan específicamente sobre un agente exterior (sensor-actuador).
- III.-Muy activo: Auto adapta sus características bajo estímulos externos.



## 2.4 Intel Edison Sondas

### 2.4.1 MICROCONTROLADORES

Los microcontroladores [8] son circuitos integrados que se pueden programar para que Unidad Central de Procesamiento ejecute órdenes que guardan en una memoria, para manejar periféricos de entrada y salida.

Las frecuencias de reloj suele ser proporcionales al consumo de potencia, parámetro que caracteriza a estos dispositivos. Antes de poder funcionar debe grabarse información en su EEPROM para que pueda arrancar.

Existe una gran variedad en el mercado tanto para aplicaciones industriales y profesionales como para estudio, formación, investigación y desarrollo.

Para facilitar su programación e interconexión con otros dispositivos se han creado las plataformas, que basándose en un microcontrolador específico pero de propósito general, ofrecen placas con todos los elementos necesarios, reguladores de voltaje de alimentación, puertos de comunicaciones, Kits de desarrollo, placas de expansión y todo con un software de desarrollo sencillo de usar, programar e implementar.



**Figura 7: Arduino Controlador**

Existen varias plataformas que se han popularizado en los entornos académicos, NANODE, Libelium Waspote, siendo sin duda ARDUINO la más polular. Existen miles de proyectos compartidos a nivel mundial de Arduino, su bajo coste alrededor de 10€, gran disponibilidad de escudos y placas de ampliación, compatibilidad de sensores y actuadores y la sencillez de programar su entorno IDE le han llevado a ser la opción favorita de los desarrollos electrónicos.[9]

### 2.4.2 MICROORDENADORES [10]

Son dispositivos que ofrecen las características de un ordenador pero con unas dimensiones y consumos muy reducidas. Disponen de procesadores, memorias, puertos USB y varios puertos de conectividad, Sistema Operativo y aplicaciones para programación.



**Figura 8: Microordenador**

Se crearon para fomentar la enseñanza informática y el interés por la programación en las comunidades educativas primarias, colegios, institutos. Como ocurre con los



microcontroladores disponen de plataformas de desarrollo asociadas y un bajo coste ( 35€ ) que es accesible a cualquier usuario. BeagleBone [11] y Raspberry Pi son los más populares.

#### 2.4.3 INTEL EDISON [12]

La potente compañía INTEL vio hace 5 años como se había quedado al margen del desarrollo de microordenadores y microcontroladores en el ámbito educativo, los futuros (y actuales) usuarios y desarrolladores de aplicaciones informáticas y electrónicas, y decidió actuar.

Cogió lo mejor de los 2 mundos, microordenadores y microcontroladores y desarrolló primero un dispositivo denominado Intel Galileo para unos años más tarde presentar su Intel Edison.



Figura 9: Microordenador Intel Edison

Intel Edison es el proyecto que incluye el microordenador más el Kit de desarrollo Arduino así como la plataforma soporte.

Ofrece compatibilidad con las características de ARDUINO, pero incluyendo un ordenador del tamaño de un sello de correos que bajo S.O. Linux permite el uso de lenguajes de programación muy variados desde C++, Javascript, Python, Jnode,... Lo que le convierten en una plataforma muy versátil de desarrollo de electrónica y programación, con conexiones WiFi, BLE, USB todo ya incluido, además de lectores de memorias en el Kit de Expansión Arduino.



Figura 10: Placa Kit Expansión Arduino

Otra de las ventajas que ofreció Intel es crear su dispositivo plenamente integrado en IOT, Internet Of The Things. IOT es una plataforma para conectar personas, objetos y entornos a través de las redes ya existentes bajo **una arquitectura será orientada a eventos** aunando lo físico a lo digital. IOT facilita la Transformación Digital de una empresa.

**Por ello he elegido este dispositivo como cerebro del entrenador personal en este TFM.**

#### 2.4.4 SONDAS COMPATIBLES [13]

Dado que el Edison ha sido desarrollado para ser la evolución de la plataforma Arduino (puede ser incluso programado con el IDE Arduino) muchos de los sensores que trabajaban con Arduino siguen haciéndolo bajo Edison, y además se han añadido muchas de las sondas BLE y WiFi.

Existen más de 400 dispositivos compatibles con Edison e IOT entre los que hay medidores biométricos, controladores de motores paso a paso, servos, relés, actuadores, sondas de presión, gas, temperatura, humedad, magnetismo.....

Para el proyecto actual utilizaremos desde sensores tan simples y documentados como el detector de proximidad HC-SR04, hasta Medidores de Ritmo cardiaco bajo BLE como el Polar H7, pasando por sondas "Eddystone BLE" dotados de sondas de temperatura y humedad.

**HC-SR04 [14]** es un sensor que por medio de ultrasonidos puede generar medidas de distancia. Su rango va desde los 20 a los 4000mm sin contacto físico con los objetos. Incluye transmisor y un receptor de ultrasonidos (40 KHz) y un circuito de control. Trabaja aplicando la fórmula: Distancia = (Tiempo a nivel alto  $\times$  velocidad del sonido (340m/S)) / 2.



Figura 11: Sensor HC-SR04

**Polar H7** es una sonda BLE capaz de detectar el ritmo cardiaco y enviarlo vía BTL (smart). Su autonomía es de 150h, con un alcance superior a los 10m.



Figura 12: Sensor Ritmo Cardiaco BLE

**Eddystone /Ibeacon [15]** con sensores de temperatura. Enviarán datos atmosféricos y proporcionarán sistema de referencia para posicionamiento. (Genérico)



Figura 13: Ibeacon Sensor

## 2.5 Sistemas de desplazamiento autónomo

### 2.5.1 DEFINICION

Un sistema de desplazamiento autónomo es aquel que permita moverse sin ser tripulado por la acción directa del ser humano. Pueden ser sistema terrestres, aéreos, acuáticos, mixtos.

En los sistemas terrestres la mayor parte de las fabricantes de coches están investigando conjuntamente con organismos oficiales diferentes sistemas de guiado mediante GPS combinado con sistemas sensoriales, como Tesla.

De forma paralela empresas que nada tienen que ver con ese sector como Apple y Google invierten en el desarrollo de soluciones de vehículos comerciales, aún prototipos.

En Agricultura hace ya tiempo que se emplean sistema como RHEA [16] (Robot Fleets for Highly Effective Agricyulture and Forestry Management), sin dejar al lado otros sistema que guían por GPS el recorrido de un tractor para arar en una parcela delimitada.



**Figura 14:** NASA, Curiosity Rover

La NASA ha abierto parte del conocimiento de su Curiosity Rover [17] a dominio público con los diseños y programación de un prototipo a escala de ese vehículo.

Hoy en día es posible encontrar pequeños aspiradores redondos que son capaces de limpiar un cuarto sin mayor necesidad de participación humana.

### 2.5.2 FUNDAMENTOS

Lo primero que necesitamos es que el vehículo pueda ponerse en marcha por sí mismo, basta para ello dotarlo de una secuencia de arranque que encienda los motores, y el vehículo se desplazará. Hasta aquí no hay diferencia con un vehículo tripulado.

Lo segundo es tener una ruta hacia la que dirigirse, puede estar programada, ser aleatoria, o buscada a través de bancos de datos de coordenadas.

El hito, tercer fundamento es que el vehículo detecte, reconozca y salve obstáculos, es decir que disponga de elementos que identifiquen una obstrucción en su trayectoria, que pueda cuantificar qué tipo de dificultad supone y decida que alternativa le permita continuar su viaje.

### 2.5.3 TECNOLOGIA [18]

El hito, tercer fundamento es que el vehículo detecte, reconozca y salve obstáculos, es decir Detectar un obstáculo no debe ser un problema para modernas sondas basadas en láser tipo LIDAR (Light Detection And Ranging), ultrasonidos, geoposicionamiento, sondas de ultrasonidos, cámaras de visión artificial...

Lo más importante es sin duda qué hacer con esa información que los diferentes sensores recojan puesto que su análisis es en tiempo real y también las respuestas a ese análisis. Eso pasa por dotar de "inteligencia" al dispositivo lo cual nos lleva a introducir el uso de procesadores que gobiernen al vehículo.

Pero la inteligencia introduce otro apartado, qué ocurre conmigo mismo, cómo estoy yo, esa especie de consciencia. No sólo debe preocuparme el objetivo que es llegar, sino si estoy listo para hacerlo, tengo alguna avería, el nivel de energía para lograrlo es correcto...

Los vehículos capaces de circular por vías públicas deben además ser "predictivos" es decir evaluar las posibles trayectorias que otros vehículos y peatones van a realizar y anticipar su respuesta ante ellos.

Por ello estos vehículos incorporan sensores que evalúan el estado del vehículo en sí mismo.

### 2.5.4 SELECCION

Los vehículos capaces de circular por vías públicas deben además ser "predictivos" es decir

Para este proyecto se han evaluado proyectos más modestos basados en Kits de motores, chasis y microcontroladores que con juegos de sensores básicos permiten construir vehículos con autonomía en entornos No hostiles como pueda ser una pista de pádel.



Figura 15: Pirate 4WD Robot

Uno de los más interesantes es el PIRATE 4WD ROBOT [19] basado en Arduino, que puede ser adaptado para el actual proyecto bajo INTEL EDISON.

Estos prototipos ofrecen diferentes variantes modulares que nos van a permitir reconfigurar dinámicamente para optimizar la solución a medida que se desarrolla el proyecto.

## 2.6 Sistemas de impulsión de bolas

### 2.6.1 LANZABOLAS

Lanzar una pelota no es nada complicado para un humano, simplemente se puede arrojar con la mano donde la daremos velocidad, altura, profundidad e incluso efectos. Es una manera mecánica utilizando nuestro propio cuerpo.

Para dar mayor velocidad se emplean accesorios de golpeo como son las raquetas y las palas que además nos permiten proteger las extremidades.

Cuando pensamos en crear un sistema que impulse bolas de forma natural lo hacemos en sistemas que catapulten las bolas, bien por **resortes elásticos** linealmente a modo de ballesta o



Figura 16: Lanzador por Resorte elástico

de forma angular como las catapultas.

Sin embargo esos sistemas no permiten de una manera sencilla automatizar los procesos, ni crear máquinas con control automático con ese tipo de sistemas.

Los siguientes pasos son fijarnos en las armas de lanzamiento, por ejemplo los **cañones [20]**, que si pueden ser automatizadas y accionadas mediante dispositivos electrónicos, Sin embargo su peligrosidad no justifica su uso para fines de entrenamiento deportivo.

Hay desarrollos en USA de ese tipo de lanzadores para entrenar baseball basados en gases, incluso en propano, que automáticamente descartamos para nuestro actual proyecto. Trabajan provocando mini explosiones que hacen que la bola salga despedida.



Figura 17: Lanzador por gas

Otras variantes factibles incluso con algún prototipo comercializado está basado en el **aire comprimido**, que llena poco a poco mediante un compresor un calderín, cuyo contenido es liberado de forma abrupta sobre un tubo donde se aloja la bola, que es literalmente despedida. Se puede controlar por medios electromecánicos el resorte de apertura por lo que sí es posible automatizar ese tipo de maquinaria.



Figura 18: Lanzador Aire Comprimido

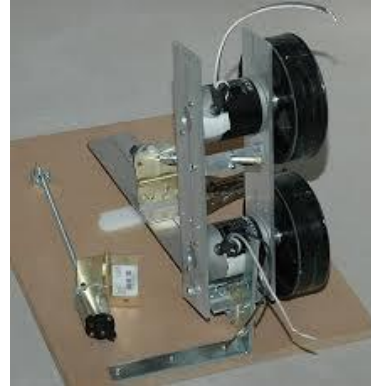
Sin embargo este tipo de máquinas son muy voluminosas debido al tamaño de los calderines , y además son muy pesadas y por tanto difíciles de mover, no siendo un sistema factible para nuestro actual proyecto de entrenador personal que debe ser autónomo y poderse desplazar fácilmente.



Por último está los diseños basados en **motores eléctricos [21]** que mueven ruedas separadas por una distancia inferior al diámetro de la bola, girando ambas en sentido contrario y por las que al pasar la bola, transmitirán a ésta su velocidad lineal, saliendo despedida (lanzada). La mayor parte de los lanza bolas comerciales existentes en el mercado para tenis, baseball, tenis de mesa y pádel están basadas en este tipo de mecanismos, incluso en lanza pelotas para mascotas.



**Figura 19: Lanzabolas Comercial**



**Figura 20: Impulsor Discos Eléctricos**

Por ello en el actual Trabajo de Fin de Máster me he decidido a utilizar el lanzador siguiendo este modelo usando uno ó dos motores independientes, mediante 2 ruedas lo que permitirá dar efectos a la bola variando simplemente la relación de velocidades entre ambas.

Además la variación estará gobernada por dispositivos de control electrónicos que seguirán las órdenes del controlador principal. Como mejora de los disponibles en el mercado se pretende que el lanzador pueda variar el azimut, la inclinación y la posición.

## 2.7 Análisis del pádel

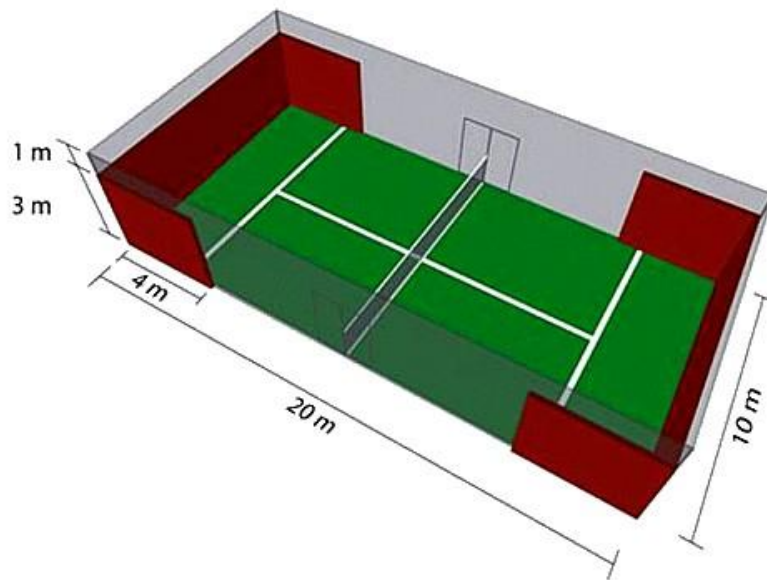
### 2.7.1 QUÉ ES EL PADEL [22]

Es un deporte nacido en Argentina 1960, criado en España y exportado a países de América, Europa y África, cada año su popularidad aumenta y su expansión se multiplica a nivel global.

Se juega con bolas similares a las usadas en el tenis a las que se golpea con una pala como si estuviéramos en la playa, normalmente en parejas (2 contra 2) dentro de una pista o cancha cerrada, delimitadas por paredes que llegan a formar parte de la pista de juego.

#### LA PISTA [23]:

Es un rectángulo de 20 x 10 ,divido en 2 cuadrados de 10 x 10, uno para cada pareja, separados por la red de unos 90cm de alta. A6,95 m de la red están las líneas de servicio, divididas en 2 áreas simétricas o áreas de saque delimitadas por una línea perpendicular, que se prolongará 20 cm más allá de la línea de servicio. El grosor de las líneas será de 5 cm.



**Figura 21: Imagen 21; Pádel Pista de Juego, dimensiones**

La altura libre mínima será de 6,0m, las paredes del fondo tienen 4m de altura, 3 de ellos para rebote y 1 metro de reja. Las paredes laterales tienen en su primeros 4m una altura de 3,0m y el resto es de reja.

El suelo debe permitir que la bola bote en vertical más de un 80% y los coeficientes de fricción debe estar entre 0,4 y 0,8.

### **2.7.2 OBJETIVOS Y GOLPES BASICOS [24]**

El objetivo del juego es ganar 2sets antes que la pareja contraria. Cada set se compone de juegos. Para adjudicarse un set se necesita ganar 6 juegos con diferencia de al menos 2. Cada juego se gana con 4 puntos con diferencia de al menos 2. El conteo es 15, 30, 40, Juego.

Punto se gana cuando la bola bota en el campo oponente 2 o más veces antes de que puedan devolverla, o cuando el oponente no consiga devolver la bola a nuestro campo.

Los golpes básicos son el saque, el resto, la derecha, el revés, bolea de derecha y de revés, las bandejas y víboras, salidas de pared, el globo, la dejada y el remate.

### **2.7.3 ENTRENAMIENTO [25]**



**Figura 22: Entrenamiento Pádel**

**Los entrenamientos sirven para que los deportistas aprendan a mejorar en su deporte y se puede hacer preparando la técnica o la forma física.**

En Pádel entrenar la técnica, se logra mediante la repetición de un golpe una y otra vez hasta lograr que la preparación, el movimiento, la distancia y altura de golpeo y la finalización sean ejecutadas de forma correcta. Además se entrena el movimiento por la pista.

El entrenamiento físico sirve para mejorar la preparación del deportista de forma que su rendimiento corporal permita que el jugador ejecute los golpes durante el juego como lo haría en un entrenamiento técnico.

#### **2.7.4 PERSONAL TRAINING ENTRENADOR PERSONAL**

Se encargará de sustituir al entrenador para que el deportista pueda practicar en cualquier momento, pista sin necesidad de presencia humana.

Debe servir para mejorar la técnica la táctica y el físico.



### 3. Prototipo de entrenador personal inteligente

El primer prototipo de esta máquina se va realizar siguiendo con el esquema elaborado en los entregables e índice del actual Trabajo de Fin de Máster.

Realizaremos de este modo un estudio, definición, descripción y especificaciones de cada uno de los componentes del entrenador personal inteligente, yendo desde la parte física a la lógica.

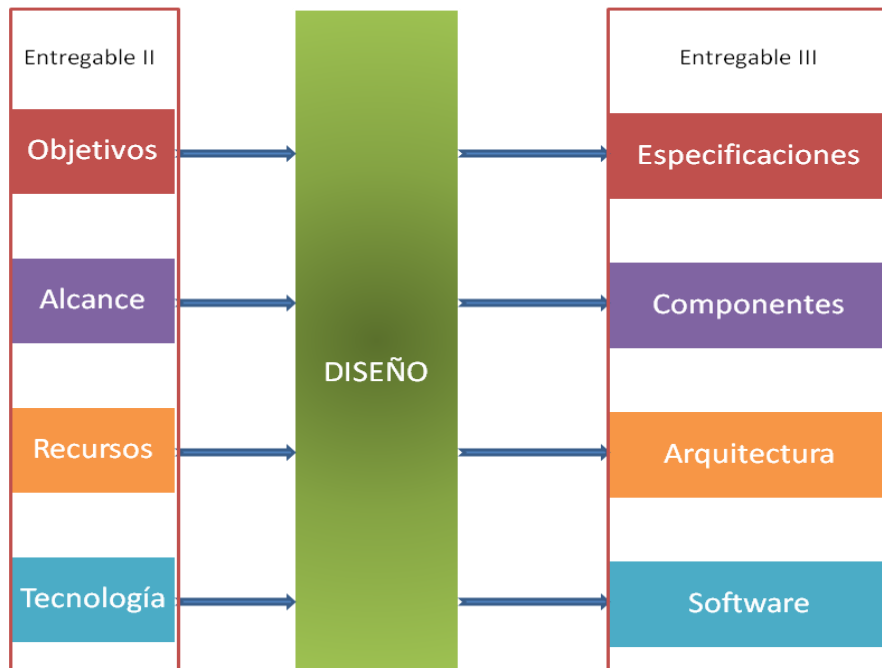


Figura 23: Fases Diseño Prototipo

#### 3.0 Diseño físico

La máquina debe ser estable, robusta, ligera y eficaz.

Tiene que ser capaz de tomar decisiones en función del análisis de datos obtenidos a través de sensores, que pueden ser de nivel de potencia, proximidad, distancia, temperatura, pulsos...

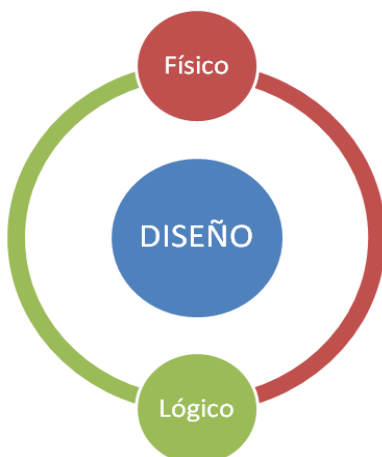


Figura 25: Diseño Componentes



Figura 24: Diseño Físico

El chasis debe ser lo suficientemente robusto para alojar en él los diferentes mecanismos de movimiento y propulsión y a la vez ligero para no sobrecargar la potencia necesaria y con ello aumentar tamaño y el consumo de motores lo que a su vez implica mayores drivers, baterías...

Hay que llegar a un compromiso entre resistencia mecánica y masa, lo que nos puede incitar a pensar en materiales modernos, pero además necesitamos un segundo compromiso entre diseño y precio. Estamos buscando una máquina que dentro de las prestaciones ya comentadas debe estar al alcance de cualquier bolsillo, no se está desarrollando para que lo compre un club de pádel, sino cada uno de los practicantes de forma individual.

El volumen debe permitir alojar sin problema los diferentes componentes mecánicos, electrónicos y eléctricos y unos pequeños volúmenes de seguridad para amortiguar impactos bien de elementos externos o causados por fallos del sistema.

### 3.1 Diseño mecánico

Se divide en 2 etapas, en la primera de ellas definiremos las necesidades específicas de la parte mecánica necesarias para permitir el desplazamiento del conjunto.

Posteriormente definiremos los elementos necesarios para lograr que el sistema lance las bolas.



Figura 26: Diseño mecánico

El ángulo lateral de lanzamiento lo asignaremos como tarea de la unidad de desplazamiento

<b>Largo máximo:</b>	<b>50 cm</b>
<b>Ancho máximo:</b>	<b>20 cm</b>
<b>Alto máximo:</b>	80cm
<b>Peso mínimo:</b>	<b>5 Kg</b>
<b>Peso máximo:</b>	20 Kg
<b>Materiales:</b>	<b>Ligeros y robustos</b>
<b>Detectores Distancia:</b>	0 - 500 cm
<b>Detectores Potencia señal:</b>	<b>-80dBm, 0dBm</b>
<b>Detectores Temperatura:</b>	-15°, +50°C
<b>Detectores humedad relativa:</b>	<b>0, 100 %</b>
<b>Detectores Ritmo cardiaco:</b>	25-200 latidos/minuto
<b>Resistencia impactos:</b>	<b>Alta</b>
<b>Distancia Lanzamiento</b>	0 - 10m
<b>Velocidad máxima:</b>	<b>10 Km/hora</b>
<b>Wifi:</b>	11
<b>Bluetooth:</b>	<b>4.1 BLE</b>
<b>Coste conjunto:</b>	< 450€

**Tabla 2: Características físicas Personal Trainer**

### **3.1.1 DISEÑO DESPLAZAMIENTO**

De los 3 elementos principales para conseguir desplazar el equipo, Ruedas, Motor y Drivers, sólo las ruedas pueden considerarse puramente mecánicos pues los motores serán eléctricos, e igual sus drivers correspondientes. Sin embargo motivado por la elección del tipo de rueda-motor voy a incluir en esta sección de diseño mecánico los motores, dejando los drivers como actuadores dependientes del controlador principal.



**Figura 27: Desplazamiento Diseño**

Para evitar que el sistema con solo 2 ruedas pueda perder el equilibrio, se usarán otras 2 ruedas auxiliares, sin embargo se dispondrán en un eje perpendicular a las motrices, y será ruedas locas, capaces de girar en torno a sí mismas (rotar) mediante un sistema de rodamientos que conectan la parte de fijación al chasis con la rueda en sí.

#### **RUEDAS**

El desplazamiento de una máquina lanzabolos tradicional se realiza dotándola de un sistema de ruedas sin acoplar a ningún tipo de transmisión, por lo que no son capaces de desplazarse ni de forma autónoma ni en modo remoto. El entrenador humano desplaza la máquina empujándola hasta el lugar de la pista que considera oportuno para realizar cada ejercicio, y las ruedas facilitan ese movimiento. Desde esta óptica el sistema no deja de ser un carrito sobre el que se coloca una máquina capaz de propulsar bolas.

Hay una gran variedad de ruedas en el mercado, con diferentes características:

- a.- Tipo de material.
- b.- Tamaño.
- c.- Capacidad de carga.
- d.- Tipo de sujeción al eje.
- e.- Entorno de uso (Tipo de suelo)
- f.- Tipo de acabado, rugosidad.

La elección de las ruedas se realiza teniendo en cuenta que la superficie en la que se va a utilizar que en todo caso será plana y regular, descartando las antiguas pistas de hierba natural pues no son de entrenamiento sino de competición.

Las pistas más comunes son de césped artificial, las más lentas y que más favorecen la amortiguación de botes, desplazamientos. Para mejorar la tracción se emplea arena.

Pistas de resina sintética, resinas de caucho , más rápidas, usadas en torneos.

Pistas de hormigón poroso para superficies exteriores. Bastante abrasivas

Pistas de cemento, muy rápidas y abrasivas.

### **Ruedas motrices:**

Debido a las dimensiones de cada lado de la cancha de juego, 10m x 10m la velocidad de desplazamiento no debe ser alta, y es más importante la colocación. Por ello las ruedas deberían cumplir las características siguientes:

Tamaño: 6,5"

Material: ABS + Aleación Aluminio

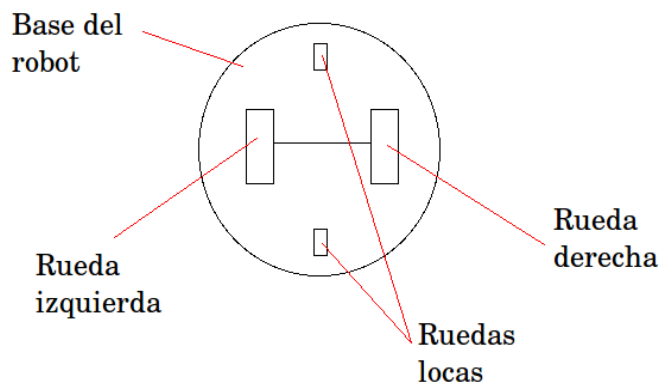
Neumático: Caucho vulcanizado, superficie rugosa y canales de evacuación.



**Figura 28: Rueda Motriz**

### **Ruedas de equilibrio:**

Para mantener el equilibrio correcto del entrenador, vamos a disponer de 2 ruedas motrices y de 2 ruedas de equilibrio que estarán en ejes perpendiculares entre sí, como más adelante se justificará. Estas ruedas de equilibrio pueden ser simplemente para apoyar en ciertas situaciones, por las irregularidades ya comentadas de las diferentes superficies de juego. Serán ruedas más sencillas, de bajo coste y que puedan girar libremente sin oponer resistencia al movimiento que marquen las ruedas principales.



**Figura 29: Ruedas Equilibrio**

## MOTORES

Para que nuestro personal trainer pueda desplazarse es necesario dotarlo de motores. Dentro de la amplia variedad disponible en el mercado he seleccionado motores eléctricos porque son más versátiles, más limpios, más sencillos de controlar y gobernar.

Premisas para la selección y la elección de los motores:

- a.- Deben ser eléctricos
- b.- Alimentados desde un sistema de baterías de tensión continua de 36v. (autónomo)
- c.- Gobernables y regulables por controlador (driver).
- d.- Independientes entre sí
- e.- Deben permitir desplazamiento hacia delante, hacia atrás y giros a izquierda y derecha.
- f.- Todas las maniobras de desplazamiento deben poderse realizar con solo 2 motores
- g.- Cada motor debe ser capaz de desplazar hasta 40kg de masa (peso).
- h.- La velocidad máxima será de 10Km/hora (Por seguridad)
- i.- Bajo consumo. (Potencias No superiores a 350w)
- j.- Estandarizados, sencillos de conseguir.
- k.- Fáciles de reemplazar
- l.- Coste reducido

Se podrían usar motores con escobillas para corriente continua, o motores paso a paso, sin embargo la selección idónea es la de motores sin escobillas (Brushless).



**Figura 30: Motor brushless**

En el mercado actual existe una gran variedad de motores brushless, caracterizados por su potencia, velocidad de giro (rpm) por su par (torque), dimensiones, tensiones de alimentación, señales de control....

Para nuestro TFM he escogido ruedas con motor integrado "HUB MOTOR WHEEL", como las que se usan en patines eléctricos y hoverboards porque cumplen todas las premisas de diseño.

En particular se han elegido sendas ruedas con las características siguientes :

<b>Material: aleación</b>	<b>+ ABS</b>
<b>Tamaño:</b>	17x5,7 cm
<b>Diámetro exterior de la rueda:</b>	<b>6,5 pulgadas</b>
<b>Cable longitud:</b>	26 cm
<b>Potencia del Motor</b>	<b>350 w, sin escobillas</b>
<b>Cadena</b>	libre de progresión
<b>Sistema de frenado</b>	<b>eléctrico</b>
<b>Voltaje nominal</b>	36 V
<b>Velocidad nominal</b>	<b>800 rpm</b>
<b>Velocidad máxima</b>	15-18 km/h (según la decisión de peso del usuario)
<b>Peso de carga</b>	<b>240 lb (108 kg)</b>

**Tabla 3: Hub Motor Wheel, características**

Estos motores disponen a su vez de sensores de efecto Hall:

Determinan la posición del motor a través de la variación del campo magnético.

A continuación se muestra una fotografía que muestra la rueda con motor integrado.



**Figura 31: Hub Motor Wheel**

Hay que destacar que una de las principales diferencias entre la elección de los motores de movimiento e impulsión es el número de RPM que vamos a necesitar. Mientras que en los de movimiento no queremos superar las 800 RPM, en los de lanzamiento buscaremos valores de 20.000 RPM lo que sin duda nos hace optar por diferentes características de diseño.

### 3.1.2 DISEÑO LANZAMIENTO

Son muchos los artilugios y artefactos que se han diseñado para lanzar bolas, mediante diferentes técnicas y aplicaciones ingeniosas de las diferentes tecnologías existentes.

Para el actual propósito del TFM, el entrenador personal de pádel, la parte del lanzamiento no es la fundamental pues se trata simplemente de echar bolas y que sea el jugador quien haga los gestos y ejecute los movimientos para mejorar su técnica y ejercitar su físico. El objetivo de este trabajo es que haya una capacidad de análisis del estado del jugador y que la máquina pueda optar por ejercitar diferentes movimientos en función de las variables del mismo.

Las características que vamos a pedir a nuestro lanzador serán:

<b>Fuente de energía:</b>	<b>Eléctrica</b>
<b>Tensión alimentación:</b>	0-36 v (dc)
<b>Velocidad bolas:</b>	<b>0-20km/h</b>
<b>Distancia lanzamiento:</b>	0-10m (fondo de pista)
<b>Capacidad cargador bolas:</b>	<b>1-3 bolas</b>
<b>Controlador:</b>	Adaptable a uProcesador
<b>Peso:</b>	<b>0-20 kg</b>
<b>Medidas máximas:</b>	40 x 40 x 40 (cm)
<b>Giro incorporado:</b>	<b>No</b>
<b>Elevación regulable:</b>	No

**Tabla 4: Lanzador características Mínimas**

Para este proyecto he seleccionado un producto comercial de bajo coste existente en el mercado. El fabricante es ONECONCEPT, una empresa alemana sita en Berlín y que se dedica a producir su modelo 10030841, una pequeña unidad capaz de propulsa bolas a distancias superiores a los 10 metros.



**Figura 32: Lanzador ONECONCEPT**

Este dispositivo cuenta con una pequeña zona de carga de bolas, una salida de expulsión de las mismas, una unidad de aceleración formada por 2 motoresdc que giran de forma independiente sendas ruedas dentadas separadas una distancia ligeramente inferior al diámetro de las bolas, a través de esa zona haremos pasar las pelotas que adquirirán cantidad de movimiento. Posee también una unidad de control interna y la posibilidad de variar la velocidad según un selector (pulsador). Cuenta además con unos Leds de control.

<b>Fuente de energía:</b>	<b>Eléctrica</b>
<b>Tensión alimentación:</b>	9 v (dc)
<b>Velocidad bolas:</b>	<b>0, 10, 15, 20km/h</b>
<b>Distancia de lanzamiento:</b>	0-10m
<b>Capacidad cargador bolas:</b>	<b>3 bolas</b>
<b>Controlador:</b>	Propio, se adaptará a uP.
<b>Peso:</b>	<b>1,2 Kg</b>
<b>Medidas:</b>	23 x 26 x 34 (cm)
<b>Giro incorporado:</b>	<b>No</b>
<b>Elevación regulable:</b>	No

**Tabla 5: Características Unidad Elegida**

El reto de diseño será en la parte electrónica y lógica porque estas unidades de bajo coste no vienen desarrolladas para ser gobernadas por un controlador exterior. Por ello se estudiará de forma somera su red interna de comunicación y se adaptará para ser gobernada por la unidad de control general del proyecto.

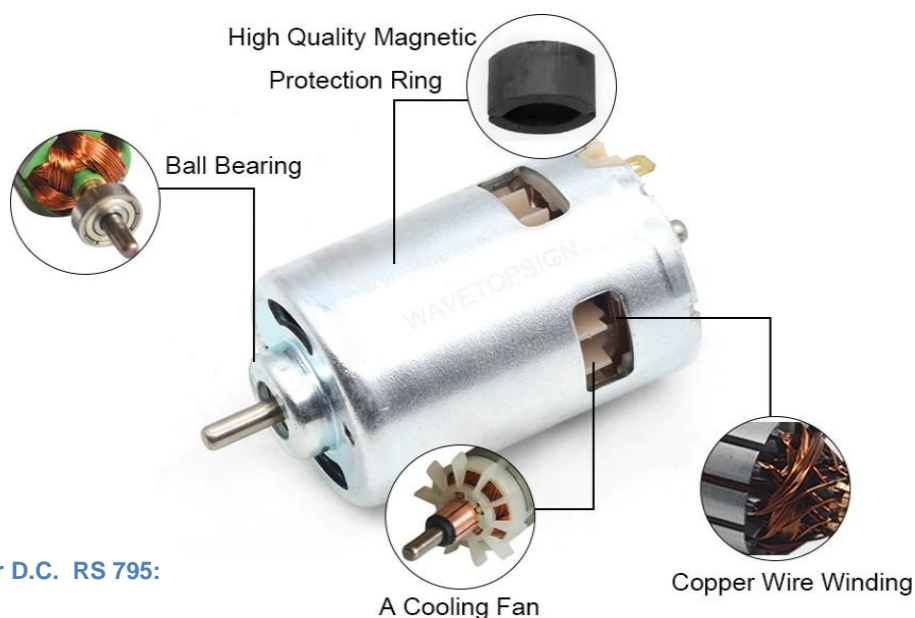
Desde el punto de vista físico del lanzamiento podemos aumentar la distancia adquirida por la bola si incrementamos la velocidad de salida de ésta, y eso pasa por tener más revoluciones por minuto y mayor torque en el motor o bien modificar el radio de giro de la rueda que aplicará la energía a la bola. Como no hay mayor espacio pues es una unidad comercial, no es viable para este proyecto modificar el radio de giro, por lo tanto necesitaremos un motor con mayores RPM al que además debemos gobernar desde el microprocesador del sistema de inteligencia del entrenador personal (mediante el uso de un driver).

<b>Fuente de energía:</b>	<b>Eléctrica</b>
<b>Tensión alimentación:</b>	24 v (dc)
<b>Velocidad Giro:</b>	0-20.000 RPM
<b>Par máximo:</b>	6 kg/cm 0.58N.m
<b>Corriente carga</b>	2,5 A (máxima)
<b>Controlador:</b>	Externo
<b>Peso:</b>	375 gr
<b>Medidas:</b>	7 x 4.2 x 4.2 (cm)+ 1 (Eje)

**Tabla 6: Características del motor seleccionado RS 795:**



El motor seleccionado es:



**Figura 33: Motor D.C. RS 795:**

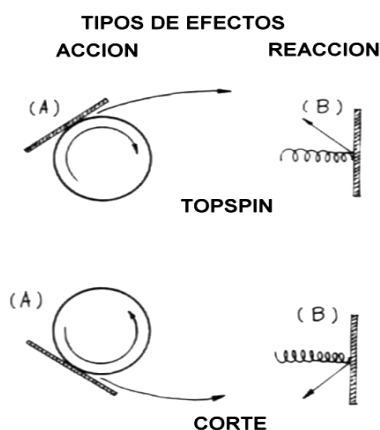
Es un motor dc de alto torque dentro de su reducido tamaño, y con una velocidad de giro de hasta 20.000RPM que regularemos mediante un driver apropiado a sus características.

Se van a usar 2 motores uno para cada motor de lanzamiento, de esta forma, ayudados por unos drivers reguladores, podremos variar de forma independiente la velocidad de giro de cada uno, y conseguiremos dar a la bola diferentes "efectos".

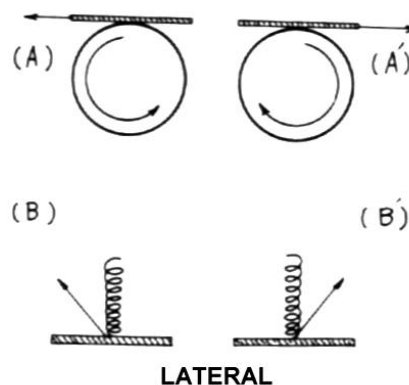
Los efectos en todos los juegos con balones y bolas esféricas sirven para cambiar la forma en que la pelota se comportará cuando toque suelo o sea golpeada por el rival y además para ir cambiando de trayectoria durante su viaje.

Cuando 2 motores giran a la misma velocidad exacta (en sentido contrario) y hacemos pasar la bola, ésta saldrá impulsada con una velocidad pero en teoría sin ningún efecto, esto se traduce a salir sin "spin", sin rotar ni girar sobre su propio eje, y previsiblemente cuando la bola bote, seguirá la misma dirección que traía de origen.

Sin embargo cuando la bola va con spin y dependiendo de en que eje (ejes), obtendremos efectos como liftado (top spin) [39], cortado, y se podrán entrenar dejadas, globos, bolas altas.....



**Figura 34: Tipos Efectos Bola**



### 3.1.3 DISEÑO ELÉCTRICO ( ALIMENTACIÓN )

Hemos definido en nuestro proyecto es que la unidad debe ser autosuficiente en su movimiento y toma de decisiones, por ello un objetivo de diseño primordial es, la autonomía del conjunto. No podemos tener por la pista de juego cables de conexión a la red eléctrica pues supondrían peligro por tropiezo de la propia máquina y porque no está garantizado que una pista de pádel vaya a tener "enchufes" eléctricos.

Es inviable el uso de paneles solares puesto que la unidad se puede utilizar en pistas Indoor y no tenemos garantizada la incidencia de luz natural.

Hay que descartar aunque parezca obvio el uso de generadores basados en combustibles, porque no se permite su uso por contaminación atmosférica y sonora.

Por todo ello en el diseño del equipo se van a utilizar baterías recargables, unas de 36v con alta capacidad de carga para alimentar los motores y el propulsor, y otra para el controlador.

La razón de utilizar 2 fuentes diferentes es evitar 2 efectos:

- a.- La pérdida de control del dispositivo si falla la fuente principal por consumo de motores
- b.- Las posibles interferencias generadas por picos de tensión de los drivers



Figura 35: Fuente alimentación

A.- Las características que debe cumplir la fuente principal:

<b>Tensión trabajo:</b>	36v (dc)
<b>Capacidad:</b>	> 3Ah
<b>Celdas:</b>	Ión Litio
<b>Tiempo recarga:</b>	< 2horas
<b>Ciclos uso:</b>	> 1.000
<b>Dimensiones:</b>	<150 x 80 x 100 mm (volumen disponible)
<b>Peso:</b>	< 5Kg
<b>Certificación:</b>	SGS de seguridad en la carga

Tabla 7: Características Fuente Principal

Para cumplir con estas especificaciones de diseño he seleccionado el modelo 18650-10SP2.

<b>Tensión trabajo:</b>	36v (dc)
<b>Capacidad:</b>	3.6Ah
<b>Celdas:</b>	Ión Litio
<b>Tiempo recarga:</b>	1.5horas
<b>Ciclos uso:</b>	1.500
<b>Dimensiones:</b>	135 x 55 x 85 mm
<b>Peso:</b>	1 Kg
<b>Certificación:</b>	Corte tensión > 42v en carga
<b>Marcado CE</b>	Sí

**Tabla 8: Fuente 18650-10SP2 Características**

B.- Las características que debe cumplir la fuente secundaria (Controlador):

<b>Tensión trabajo (Vout):</b>	7-15v (dc)
<b>Corriente nominal:</b>	> 500 mA
<b>Capacidad:</b>	> 3.5 Ah
<b>Celdas:</b>	Ión Litio
<b>Tiempo recarga:</b>	< 2horas
<b>Ciclos uso:</b>	> 1.000
<b>Dimensiones:</b>	<100 x 40 x 80 mm
<b>Peso:</b>	< 1Kg

**Tabla 9: Fuente Alimentación Secundaria (Controlador)**

Se ha seleccionado el modelo XP4000 (Energizer)

<b>Tensión trabajo (Vout):</b>	8.4 (dc)
<b>Corriente nominal:</b>	1.500 mA
<b>Capacidad:</b>	4.0 Ah (8.4 v)
<b>Celdas:</b>	Ión Litio
<b>Tiempo recarga:</b>	1hora
<b>Ciclos uso:</b>	> 1.500
<b>Dimensiones:</b>	80 x 35 x 10 mm
<b>Peso:</b>	< 1Kg

**Tabla 10: XP4000 Características**

La máquina va a tener un regulador cargador de baterías que permita que cuando una de las baterías se descargue, podamos volver a llenarla de energía, pero sin exceder la carga porque podría dañar las baterías y poner en riesgo el equipo entero.

Por último el sistema dispondrá de un alimentador externo que sea capaz de suministrar corriente continua al regulador interno.

## 3.2 Diseño electrónico

El diseño electrónico engloba para este proyecto 3 apartados fundamentales, el microprocesador, los sensores y los actuadores.

La prioridad en la selección se centra en el procesador que condicionará la lógica de control del sistema. Después debemos seleccionar los sensores que proporcionarán las señales de entrada y los actuadores que ejecutarán las acciones para que todo el conjunto realice las funciones de entrenador personal..



Figura 36: Diseño electrónico

A continuación voy a resumir las características más importantes que este proyecto requiere a la hora de definir un microprocesador que tenga todas las capacidades necesarias.

<b>Conectividad Bluetooth:</b>	BLE
<b>Conectividad Wi Fi:</b>	802.1 a/b
<b>Controlador:</b>	Arduino
<b>Salida PWM:</b>	> 2
<b>Entradas analógicas:</b>	Necesarias
<b>Entradas Digitales:</b>	Necesarias
<b>Almacenamiento:</b>	Interno
<b>Unidad de escritura:</b>	Opcional
<b>S.O.:</b>	Linux
<b>Lenguajes Programación:</b>	C++, Html, Node, Java, Python
<b>Alimentación:</b>	< 36v
<b>Conexión PC:</b>	USB
<b>V in</b>	0, 5v
<b>V out</b>	0, 5 v

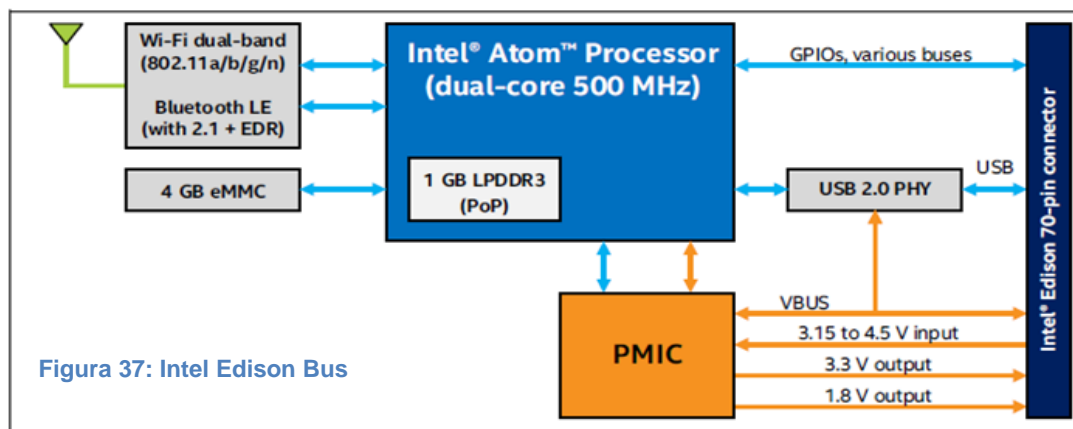
Tabla 11: Requisitos del uProcesador.

Para la elección se han evaluado y probado dispositivos tales como Arduino UNO, Arduino Mega 2560, Raspberry Pi, PcDuino, Intel Galileo, el procesador seleccionado es **Intel Edison**.

### 3.2.1 EL PROCESADOR; INTEL EDISON [12]

Desarrollado por INTEL es un microprocesador de propósito general que integra WiFi y BLE para desarrollar un microordenador de pequeño tamaño y reducido coste.

Tiene disponible un Kit de expansión compatible con el controlador Arduino, compatible con esa plataforma añadiendo S.O. Linux sobre el que se pueden ejecutar programas realizados en C++, Javascript, Python, Jnode,...



El procesador Intel Edison, como el resto de procesadores del mercado necesita de una placa base, soporte del propio dispositivo, y dispone de 2 placas principales del mismo fabricante Intel. Por un lado el BreakOut, que es un dispositivo pequeño y económico que optimiza su uso como ordenador de bolsillo. Las características propias del procesador son:

<b>Procesador:</b>	Dual Core IA-32 @ 500 MHz, 32-bit Intel® Atom™ Processor
<b>RAM:</b>	1 GB LPDDR3 POP memory (2 channel 32 bits @ 800 MT/sec)
<b>Almacenamiento Interno:</b>	4 GB eMMC (v4.51 spec)
<b>Power:</b>	TI SNB9024 power management IC
<b>Wireless</b>	Dual-band (2.4 and 5 GHz) IEEE 802.11a/b/g/n
<b>Bluetooth:</b>	BT 4.0 + 2.1 EDR
<b>Antena</b>	Dual-band onboard chip antenna or u.FL for external antenna
<b>Conectores:</b>	70-pin Hirose DF40 Series
<b>Tamaño:</b>	35.5 × 25.0 × 3.9 mm máximo
<b>Alimentación Entrada:</b>	<b>3.15 - 4.5 v</b>
<b>Voltaje Salida Pin Out:</b>	<b>1.8 – 3.3v</b>
<b>Puertos Salida PWM:</b>	4
<b>E/S (I/O):</b>	40 propósito general GPIO se pueden configurar como:
<b>SD card:</b>	1 interface
<b>UART:</b>	2 controladores (uno control total de flujos, uno control datos)
<b>I2C:</b>	2 controlador
<b>SPI:</b>	1 controlador con 2 chip selects
<b>I2S:</b>	1 controlador
<b>GPIO:</b>	Adicional 14 (4 con PWM)
<b>USB 2.0:</b>	1 OTG controller
<b>Clocks:</b>	19.2 MHz, 32 kHz

Tabla 12: Características Intel Edison:

Pero Intel pone además una segunda opción de placa soporte del chip, el Kit de expansión compatible “Arduino”, de mayor tamaño y más caro, pero que proporciona 3 características adicionales esenciales para este trabajo:



Figura 38: Placa Expansión Intel Arduino

Posibilidad de trabajar como Controlador de puertos simultáneamente al procesado.

Alimentación Entrada: **7 - 18 V**

Señales lógicas nivel: **0 - 5v**

Por ello he elegido este dispositivo junto con el Kit como cerebro del entrenador personal.

El esquema como controlador gracias a este Kit, es similar al Arduino UNO.

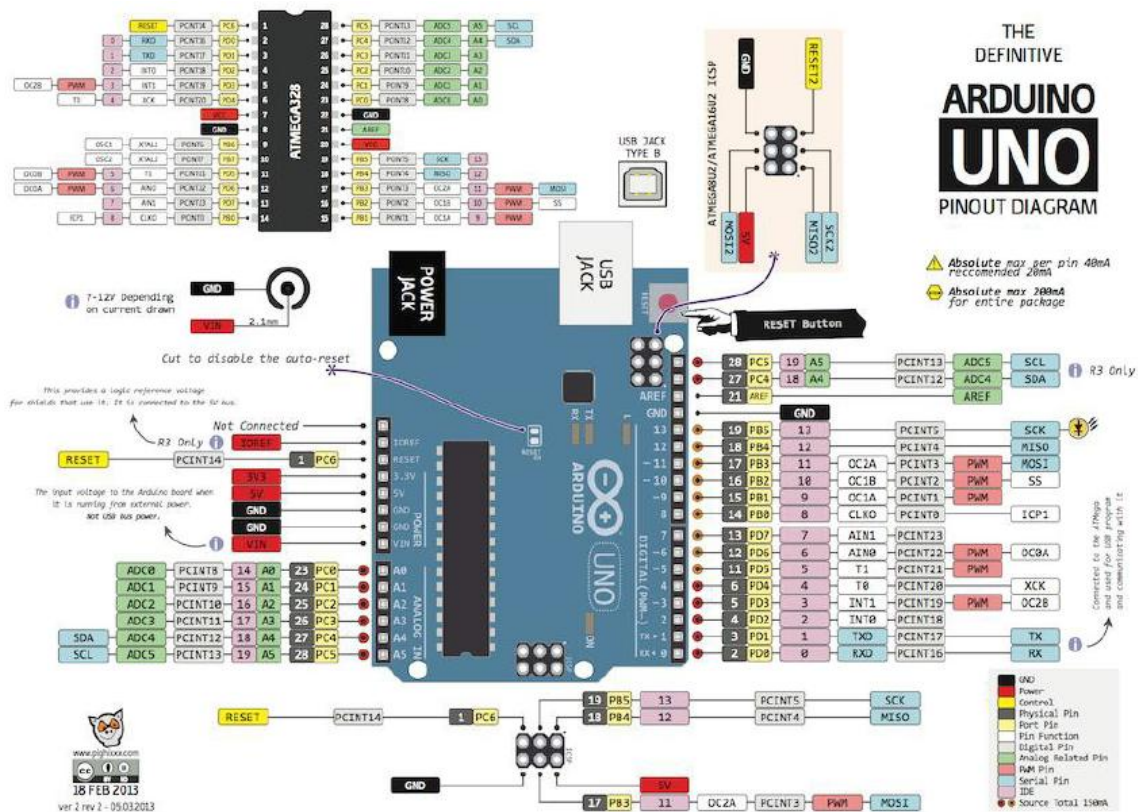


Figura 39: Arduino Pin Out Diagram



Header	Arduino pin	Signal function
Power	N/C	Notconnected
Power	IOREF	Shield I/O reference voltage (select 3.3 or 5 V via jumper on board)
Power	RESET	Shield reset (programmable via software or manual push button)
Power	3.3 V	System 3.3 V output
Power	5 V	System 5 V output
Power	GND	Ground
Power	GND	Ground
Power	VIN	System input power (7 to 15 V)
Analog	A0	Analog input or digital I/O
Analog	A1	Analog input or digital I/O
Analog	A2	Analog input or digital I/O
Analog	A3	Analog input or digital I/O
Analog	A4 / SDA	Analog input, digital I/O, or I2C data (also connected to digital header)
Analog	A5 / SCL	Analog input, digital I/O, or I2C data (also connected to digital header)
Digital	SCL	I2C clock
Digital	SDA	I2C data
Digital	AREF	ADC reference voltage (select AREF or IOREF via jumper J8 on board)
Digital	GND	Ground
Digital	13 / SCK	Digital I/O, or SPI clock
Digital	12 / MISO	Digital I/O, or SPI receive data
Digital	~11 / MOSI	Digital I/O, SPI send data, or PWM (configured with PWM swizzler)
Digital	~10	Digital I/O, SPI signal select, or PWM (configured with PWM swizzler)
Digital	~9	Digital I/O, PWM (configured with PWM swizzler)
Digital	8	Digital I/O
Digital	7	Digital I/O
Digital	~6	Digital I/O, PWM (configured with PWM swizzler)
Digital	~5	Digital I/O, PWM (configured with PWM swizzler)
Digital	4	Digital I/O
Digital	~3	Digital I/O, PWM (configured with PWM swizzler)
Digital	2	Digital I/O
Digital	1 / TX □ □	Digital I/O
Digital	0 / RX □ □	Digital I/O
ICSP	MISO	SPI receive data (connected to digital pin 12)
ICSP	5V	System 5 V output
ICSP	SCK	SPI clock (connected to digital pin 13)
ICSP	MOSI	SPI send data (connected to digital pin 11)
ICSP	RESET	Shield reset (programmable via software or manual push button)

Tabla 13: Asignación Pines Placa Kit Arduino

## Configuración:

El Procesador Intel Edison y el Kit de expansión compatible Edison se adquieren por separado.

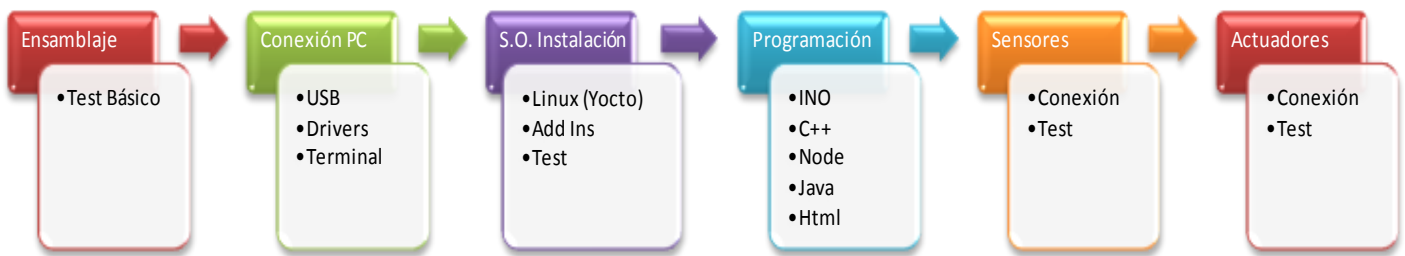


Figura 40: Programación Fases de Implementación

Para poder utilizar este procesador como ordenador y como controlador se establecen las siguientes etapas que deberán seguirse en la fase de implementación.

### 3.2.2 DISEÑO SENSORIAL

La red de sensores debe permitir que la máquina adquiriera un conjunto de datos preestablecidos que después serán pasados a un microprocesador para su análisis y resultado de ello se ejecutarán unas instrucciones que darán lugar a las acciones de entrenamiento, objetivo final del dispositivo.

Las características comunes a cualquier sensor son su rango de medida, resolución, linealidad, precisión, sensibilidad, offset y rapidez de respuesta:

Rango: intervalo de la magnitud física a medir donde el sensor trabaja.

Resolución: Es la variación más pequeña de magnitud física medida que es capaz de apreciar.

Offset: Es la salida del sensor cuando su entrada es 0.

Linealidad: Diferencial máximo entre respuesta teórica lineal y la real.

Precisión: Diferencial máximo entre la salida real y uno teórico para una entrada definida.

Rapidez de respuesta: Tiempo inherente al sensor para seguir variaciones en la entrada.

Derivas: Magnitud diferente a la medida, que modifican la salida. (temperatura, humedad...)

Sensibilidad: Relación entre variación en la salida al variar 1 ud de entrada.

Los sensores deben ser fiables dentro de un rango de diseño, y además fáciles de adquirir, de sustituir (mantenimiento) y económicos y estas son 3 características fundamentales para seleccionar sensores lo más estandarizados posible.

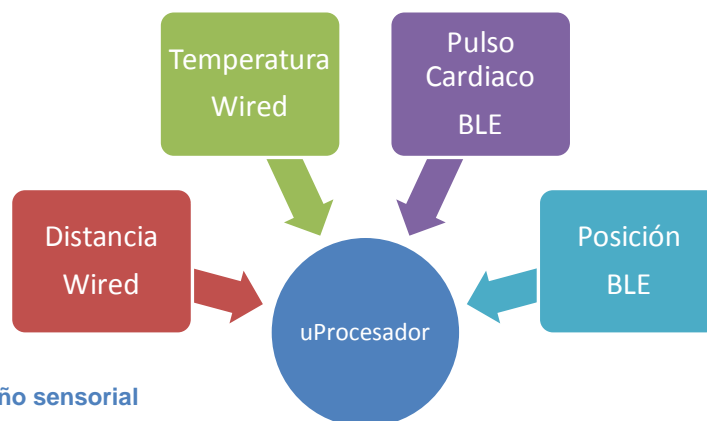


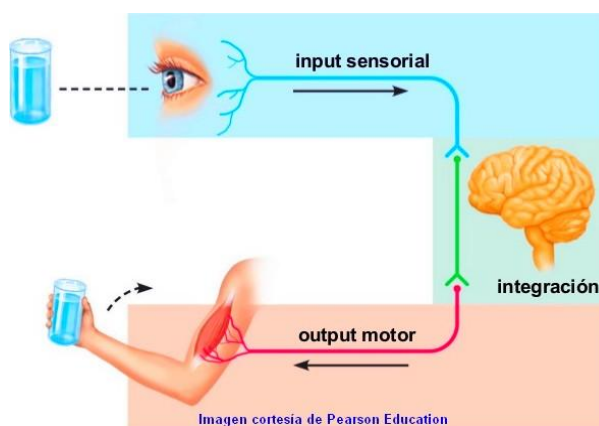
Figura 41: Diseño sensorial



Otra de las características fundamentales es que sean 100% compatibles con los puertos de adquisición de datos de su controlador para no complicar en exceso las interfaces, los protocolos y en definitiva el diseño.

Una de las premisas fundamentales que requeriremos de la unidad de control es que sea lo más versátil posible en cuanto a las comunicaciones, puertos de entrada y salida, interfaces y protocolos que proporcione, al igual que con los niveles de señal que sea capaz de admitir. Todo ello nos proporcionará grandes ventajas a la hora del diseño sensorial, una de ellas la libertad de escoger entre una amplio rango de dispositivos disponibles.

Como se comentó en el apartado 3.0 , las dimensiones del conjunto completo van a ser muy limitadas y por ello la "red" sensorial debe cubrir distancias muy pequeñas ( <1.0m ) en cuanto a su parte cableada (wired) y distancias medias (<10m ) en su accesos inalámbricos (Wired Free y BLuetooth).



**Figura 42: Red Sensorial**

Imagen cortesía de Pearson Education

Los sensores pueden ser analógicos o digitales en función de la respuesta de salida. Los analógicos proporcionan un valor de tensión o corriente proporcional a la magnitud medida, es decir en función de los valores de entrada se modifica esa tensión o corriente de salida.

Los digitales van a proporcionar señales binarias formadas por secuencias de 1 y 0 que aportan dato completo al procesador que los recibe. Además de la variable que representa la magnitud a medir son capaces de enviar y recibir información adicional sobre por ejemplo su estado, el nombre de variable medida, identificación de que sensor es quien lo envía.....

La miniaturización de los sensores digitales junto con su bajo consumo y su gran capacidad de conectividad les hace ideales por el grado de autonomía que proporcionan.

Al utilizar el Kit Arduino podemos emplear los sensores tanto analógicos como digitales, que trabajaban con Arduino, bajo Edison, añadiendo nuevas sondas BLE y WiFi.

He seleccionado sensores estándar, simples y documentados como el detector de proximidad HC-SR04 y el JSN-ST04, Medidores de Ritmo cardiaco bajo BLE, y sondas "I-beacon BLE".

Las aplicaciones de estos sensores son innumerables, pero en el actual proyecto los utilizaremos para que la máquina sea capaz de autopoicionarse respecto a unas coordenadas fijas (las balizas) y también el sistema tanga capacidad para detectar obstáculos y evitar colisiones.

Ambos sistemas, el de posicionamiento y el de detección de colisión son independientes, pero podemos mediante la inteligencia de programación hacer que los dos sean complementarios, siendo compatibles y en un momento dado pudiendo reemplazar temporalmente las funciones del otro.

### 3.2.2.1 SONTA BLE DE RITMO CARDIACO

El entrenador personal va a ser capaz de medir en tiempo real las pulsaciones del deportista entrenado. Con ese dato podrá determinar como variar la rutina de entrenamiento que srá función de variables adicionales como edad, sexo, capacidad física (nivel) de la persona entrenada, condiciones ambientales, horario y tipo de intensidad seleccionada.

También será capaz de alertar al usuario a través de la interfase WEB en tiempo real de la situación de carga cardiaca, e incluso alertar de posibles problemas y detener el entrenamiento.

Las características que vamos a exigir a éste "wearable", es que sea cómodo de llevar, que tenga una fiabilidad, una precisión y una exactitud suficiente y que su consumo sea mínimo.

Existen varias técnicas de medición del látido, el ECG, la Fotoplestímoqrafía.[26]

La Fotoplestímoqrafía mide la variación del volumen de sangre en la capa microvascular del tejido. esta técnica utiliza sondas acopladas a las puntas de los dedos y No parece adecuado para un deportista con una pala entre ambas manos. Por eso la tecnología de medida será de tipo ECG, electrocardiograma y basada en sensores BLE, y en cuanto al formato físico, los formados por correas sujetas al pecho puesto que se adaptan muy bien a este deporte que exige movimientos de las extremidades superiores e inferior y giros de cintura.

El alcance de cobertura debe ser de unos 30m, situación que sólo se alcanzará cuando el deportista esté situado en el extremo opuesto de la diagonal del campo a la máquina.



**Figura 43: Sonta BLE Ritmo Cardíaco**

Polar H7, emítie con un protocolo propio del fabricante (GymLink 5KHz) [27] con el que se puede vincular a sus relojes y pulseras, preo tiene otro emisor estándar incorporado basado en BLE que permite emparejar la sonda con dispositivos BLE bajo por ejemplo Android.

Se selecciona uno compatible, más económico, que cumple las especificaciones requeridas.

Resolución	> 10 bits
Ancho Banda (pulsaciones)	1-150 Hz
Frecuencia Muestreo	500 muestras /segundo
<b>Peso:</b>	52 g
<b>Colocación:</b>	Pecho
<b>Sensor Tipo:</b>	Pulsaciones de Electrodo ECG / Derivación
<b>Conectividad:</b>	Bluetooth / 5Khz
Cobertura:	30m
<b>Apto Deportes:</b>	Sí
Batería Voltaje	3v (botón)
Duración Batería:	150 horas (en uso)

**Tabla 14: Sonta BLE Ritmo Cardíaco, características**

### 3.2.2.2 SENSORES DE DISTANCIA

Estos sensores son externos a la unidad de control y deben conectarse a ella mediante buses, cables o tecnologías inalámbricas. Se basan en proporcionar una señal eléctrica que es función de una variable física, en este caso la distancia.

En el mercado existen diferentes tecnologías de medición, que van a determinar el rango de medición y también la precisión.

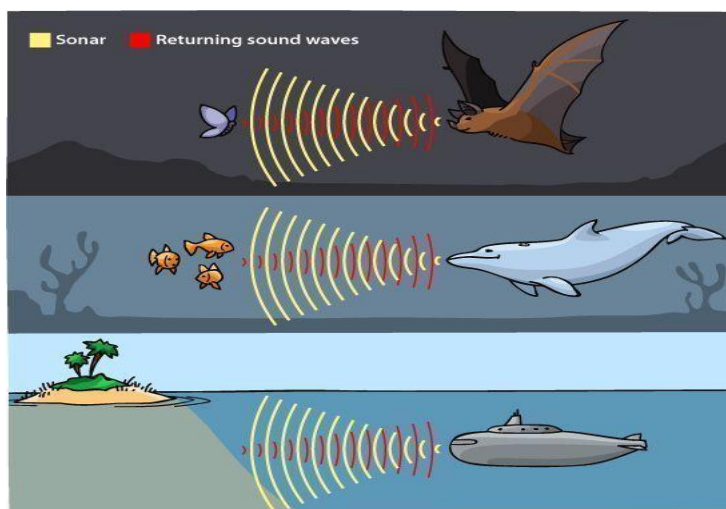
Lo primero es definir qué rango de distancia necesitamos medir y con qué precisión y a partir de ahí vamos a elegir la tecnología precisa de entre las ofertadas en el mercado.

<b>Comunicaciones</b>	Soportadas por el uProcesador
<b>Protocolos:</b>	<b>Soportados por el uProcesador</b>
<b>Tensión alimentación:</b>	+5v dc
<b>Consumo (I) :</b>	< 30mA
<b>Señales:</b>	0 - 5v dc
<b>Rango de medida:</b>	<b>0, 400cm</b>
<b>Precisión:</b>	+2cm
<b>Ángulo de apertura mínimo:</b>	<b>10°</b>
<b>Transmisión:</b>	Cableada (wired)
<b>Dimensiones:</b>	< 50 x 50 x 50 mm
<b>Tecnología:</b>	Ultrasonidos 40Khz

**Tabla 15: Características sensores de distancia**

He seleccionado ultrasonidos frente a otras disponibles como láser, inductivos, capacitivos, LVDT, encoders lineales.....

Para cumplir con nuestras especificaciones de diseño vamos a usar 2 sondas diferentes:



**Figura 44: Sonar animal**

La sonda básica HC-SR04 y una más avanzada y versátil como es la **Sonda JSN-SR04T** evolución de la primera y que nos permitirá ser usada tanto para la detección de obstáculos como para el posicionamiento del entrenador tal y como se expone en el apartado posicionamiento.

#### a.- SENSOR DE DISTANCIA ULTRASONICO HC-SR04

Este sensor suele utilizarse para prácticas de aprendizaje del uso de sensores ultrasónicos y también en dispositivos como coches rudimentarios de control remotos, lo que no lo invalida para poder incluirlo en el actual proyecto.

Si bien en el mercado disponemos de sondas algo más avanzadas, su capacidad de medir distancias cortas inferiores a los 20cm nos va a dar una versatilidad para detectar objetos dentro de la pista con los que no se cuenta (obstáculos) y evitar así colisiones no deseadas.

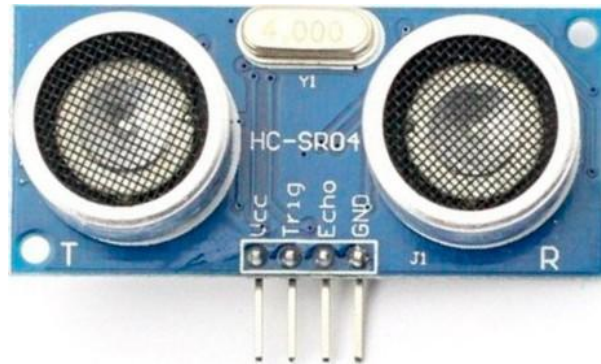


Figura 45: HC-SR04

Como podemos observar en la figura superior el dispositivo integra una sonda de emisión independiente de otra para recepción así como un pequeño circuito de control de señales.

<b>Tensión de trabajo:</b>	<b>DC 5 V</b>
<b>Corriente de trabajo:</b>	15mA
<b>Frecuencia de Trabajo:</b>	<b>40 KHz</b>
<b>Max Rango</b>	4m
<b>Min Rango:</b>	<b>2cm</b>
<b>Precisión</b>	+/-1cm
<b>Angulo de medida:</b>	<b>15 °</b>
<b>Disparo señal entrada_</b>	10uS TTL pulso
<b>Dimensiones:</b>	<b>45*20*15mm</b>

Tabla 16: HC-SR04 características técnicas

Este modelo de sonda se puede usar para detectar objetos situados en un cono muy cerrado, estrecho y por ello buscaremos una segunda sonda que nos permita detectar tanto los límites de la pista como objetos ubicados fuera del alcance de la HC-SR04.

#### Funcionamiento:

a.- Para interrogar al módulo el controlador deberá enviar un pulso de 5v(high, nivel alto) y T >= 10us al pin Trig (trigger).

b.- El sensor enviará una secuencia de 8 pulsos a 40 KHz (Inaudible para el oído humano) y esperará hasta detectar su vuelta (eco).

c.- Si la señal vuelve dentro del rango de tiempo máximo del dispositivo, el tiempo en que la señal se mantuvo a nivel alto será el tiempo en que la señal recorrió el camino de ida hasta el objeto que la reflejó + el tiempo de vuelta hasta el receptor.

Aquí se hace la primera suposición de diseño, que es pensar que ambas trayectorias han sido idénticas para simplificar los cálculos.

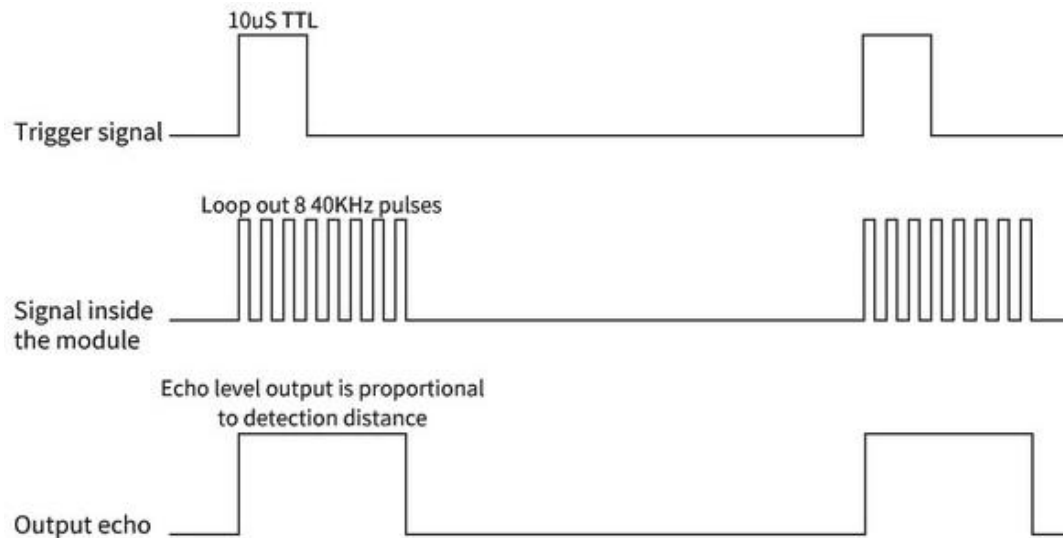


Figura 46: Ultrasonic Timing Chart

De esta manera con una simple fórmula podemos determinar la distancia:

$$d = \frac{T \cdot V_{sonido}}{2}$$

Consideraremos una velocidad de sonido en el aire de 340 m/s.

Existen sondas con un funcionamiento similar que nos permiten características físicas más versátiles, permitiendo integrar en una única cabeza todo el transductor.



Figura 47: JSR- SR04 Cabeza Sonda

## Sonda JSN-SR04T

Sonda por ultrasonidos con una operativa de funcionamiento similar a la HC-SR04, tiene como principales diferencias:

- a.- Su alcance, desde 20cm hasta 600cm.
- b.- La sonda en sí está separada del circuito de control por lo que permite ubicarla alejada del mismo haciéndola más versátil.
- c.- Es resistente al agua por lo que puede usarse en aplicaciones exteriores.

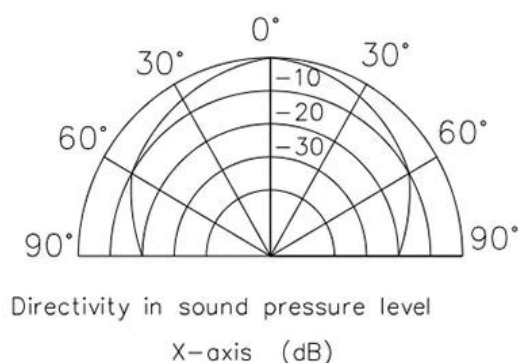
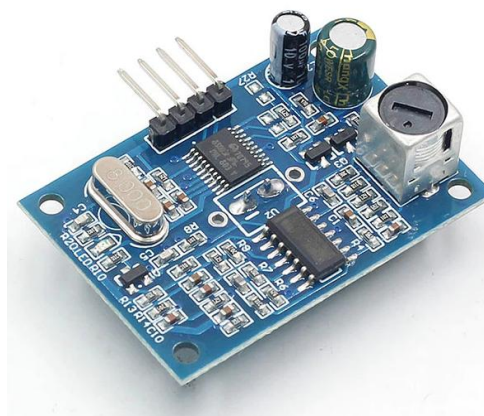


Figura 48: Directividad Echo



Cumple todos los requisitos exigidos por diseño y se complementa con la sonda anterior en cuanto al rango de distancias de trabajo, por lo que es muy interesante combinar ambas.

<b>Tensión de trabajo:</b>	DC 3.0-5.5 V
<b>Corriente de trabajo:</b>	< 8mA
<b>Frecuencia de Trabajo:</b>	40 KHz
<b>Max Rango</b>	600cm
<b>Mín Rango:</b>	20cm
<b>Resolución</b>	1mm
<b>Angulo de medida:</b>	75 °
<b>Disparo señal entrada_</b>	10uS TTL pulso
<b>Dimensiones:</b>	42*29*12mm
<b>Temperatura Trabajo</b>	-20°C , +70°C

Tabla 17: JSN-SR04T características técnicas

### 3.2.2.4 SENSOR DE TEMPERATURA AMBIENTAL

Utilizaremos la sonda LM36 muy barata y sencilla, fácil de conectar y una precisión válida para los requerimientos de este proyecto, usa sólo 1 salida analógica.

Alimentación 2.7V - 5.5V R

Rango es de 0V a -50°C y de 1.75V a 125°

Temperatura °C = 100\*(Lectura de Voltaje)-50

### 3.2.2.3 SENSORES POSICIÓN BLE

BLE es un protocolo denominado Bluetooth Low Energy BLE, son dispositivos inalámbricos de bajo consumo, compuestos básicamente de un sistema de captación (sensor), transductor, fuente de alimentación y sistema de envío, aplicándose en este caso mediante lectura del nivel de señal que llega al receptor, a determinar la distancia relativa receptor emisor, y mediante técnicas de trilateración poder despejar la posición frente a un sistema de coordenadas definido.

- 1.- Sensor
- 2.- Convertidor A/D
- 3.- Microcontrolador
- 4.- Batería
- 5.- Transceptor



Figura 49: Ibeacon BLE

Cada uno de los dispositivos usará bandas de frecuencias libres UHF en el rango de los 2.4Ghz, banda Mundial sin licencia ISM [4], de entre los 23 canales permitidos en España, 79 globales.

Banda	2.4 GHz
Modulación	GFSK
Velocidad	1.0 Mbps
Rango máximo	100 m
Consumo Corriente máx	15 mA
Consumo en Espera	< 4uA
Topología de Red	Dispersión

Tabla 18: Características principales BLE

Para el actual proyecto he seleccionado una sonda BLE con chip NRF5, compatible con las especificaciones i-Beacon de Apple. Necesitamos 2 dispositivos para ayudados por los detectores de distancia, hayar la posición del entrenador en pista.



Las características principales de este sensor son su bajo consumo y cobertura.

<b>Chip:</b>	<b>NRF51822</b>
<b>Banda</b>	2.4 GHz, BLE
<b>Tensión Alimentación</b>	<b>2.0-3.3V</b>
<b>Corriente estado durmiente</b>	3.1uA
<b>Corriente radiodifusión</b>	<b>4mA</b>
<b>Corriente media</b>	15.1uA
<b>Intervalo radiodifusión</b>	<b>2s</b>
<b>Rango cobertura máximo</b>	100 m
<b>Compatible:</b>	Apple's iBeacon
<b>Android:</b>	4.3 o superior

**Tabla 19: NRF5 Características**

#### **Aplicaciones típicas:**

Wearables, accesorio móvil, Beacons, periférico de PC, Proximidad/Alerta sensor, Smart Home, sensor deportivo, fitness y salud, domótica, mayor/minor.

Qué servicio nos va a interesar para nuestra aplicación, uno básico de estos sensores, el Servicio UUID y sus atributos de lectura / escritura, que podemos ver en la tabla.

<b>Descripción</b>	<b>UUID</b>	<b>Atributos</b>	<b>Longitud</b>
<b>Móvil &gt;sensor</b>	00001805-494c-4f47-4943-544543480000	Notificación lectura	20 (máx)
<b>Sensor móvil</b>	00001805-494c-4f47-4943-544543480000	Escritura	20 (máx)

**Tabla 20: UUID descriptor**

Podemos ajustar la potencia de emisión mediante los siguientes parámetros:

<b>Potencia</b>	<b>Parámetro</b>	<b>1m RSSI</b>
<b>0 dBm</b>	0x00	-54
<b>4 dBm</b>	0x04	-50
<b>-30 dBm</b>	0xD8	-89
<b>-20 dBm</b>	0xEC	-77
<b>-16 dBm</b>	0xF0	-72
<b>-12 dBm</b>	0xF4	-65
<b>-8 dBm</b>	0xF8	-62
<b>-4 dBm</b>	0xFC	-58

**Tabla 21: Parámetros de Potencia nRF 51822**

Ajustaremos los parámetros de envío en el emisor, encada uno de ellos (2 unidades) que formarán un sistema de coordenadas relativos a la línea que los une y la perpendicular a uno de ellos. El receptor de la señal va a medir la potencia de señal recibida, la comparará con la emitida y de esta manera conocerá las pérdidas, y éstas determinan la distancia entre ambos puntos, emisor y receptor.

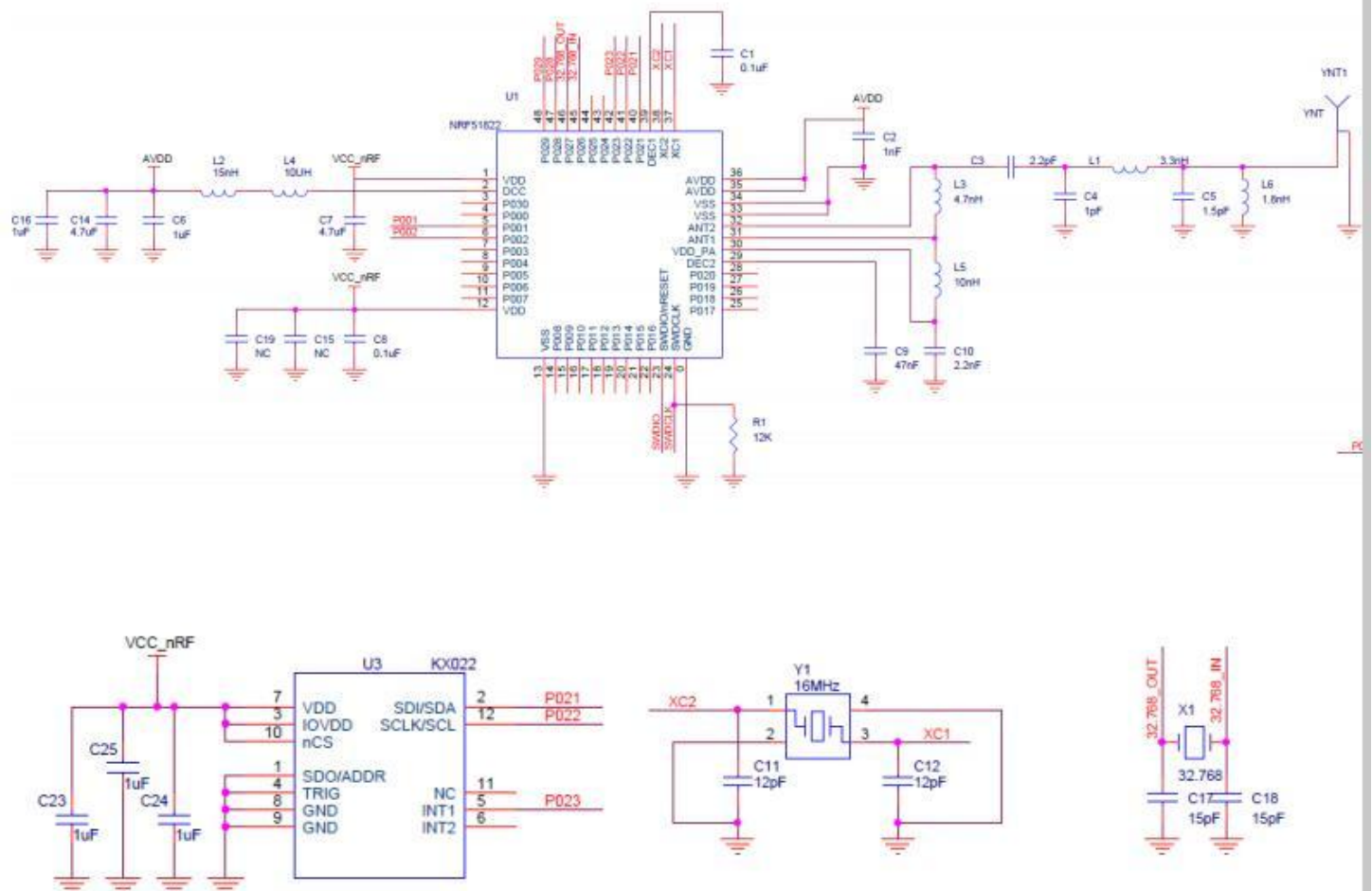


Partimos de la premisa de que ambos emisores van a estar fijos, uno en el punto 0,0 y el otro en el punto 0,10 y el entrenador como receptor, dadas las posiciones de la pista de pádel y que sólo se va a desplazar en 1 de los campos de juego, sólo podrá estar en 1 cuadrante, en este caso (0-10, 0-10), es decir en el punto mínimo (0,0) y en el máximo (10,10).

Las máximas distancias a cubrir son de Raíz  $(10^2 + 10^2) = 10 \text{ raíz } 2 = 14,1\text{m}$

Por todo ello se justifica más que de sobra la elección del NRF5 preparado para cubrir distancias mayores tanto indoor como exteriores.

El esquema básico de este dispositivo es el siguiente.



### 3.3 Posicionamiento con balizas BLE

Ibeacon es capaz de transmitir información con muy bajo consumo, a través del estándar BLE, en la banda de 2.4Ghz, permite velocidades de 1Mbps y conexión cifrada AES 128 bits.

Ibeacon permite con una cadencia programable enviar UUID, Identificador Único Universal, la energía transmitida, datos ambientales del entorno, acelerómetros y señales biométricas.

**UUID** 16 bytes (128 bits) tiene 36 caracteres, 5 grupos de 32 dígitos hexadecimales separados por un guión: **550e8400-e29b-41d4-a716-446655440000**

Se puede usar como un identificador específico para identificar el mismo objeto en diferentes contextos.

Como idea original el posicionamiento en los ibeacons se basa en **TriLateración**.

Conociendo las coordenadas de 3 puntos "fijos" que se usaran como sistema de referencia y la distancia del objetivo a cada uno de los 3 puntos, tenemos determinada la posición.

La distancia se determinará a partir de la transmisión del RSSI, indicativo de fuerza de la señal recibida, mediante fórmulas matemáticas o tablas determinaremos a qué distancia se encuentra.

Para ello debemos aceptar las siguientes asunciones:

- 1.- La radiación será omnidireccional.
- 2.- El patrón de radiación de cada antena será esférico.
- 3.- Los nodos emisores estarán en el mismo plano que el nodo receptor.

Con estas 3 suposiciones podemos aproximar que los puntos que reciben el mismo nivel de señal forman circunferencias de centro la baliza y radio proporcional al nivel de señal recibido.

Con 1 baliza tendremos una distancia, con 2 balizas 2 posibles puntos respecto a ellas (corte de 2 circunferencias) y con 3 una posición única.

La precisión mediante RSSI es de 1m para distancias de hasta 30m.

Utilizaremos un sistema de coordenadas cartesianas bidimensional.

**¿Necesitamos 3 puntos de referencia, es decir 3 ibeacons en nuestra semipista de pádel?**

Como ya se ha explicado en el Capítulo II, la pista de pádel es un rectángulo de 20 x 10 dividido en 2 cuadrados de 10 x 10 separados entre sí por una red.

Cada rectángulo está cerrado completamente en su perímetro ya sea por cristales, hormigón o un sistema enrejado, además de la red ya mencionada contra el campo contrario.

Sabemos que un entrenador de pádel, como pasa en tenis, pinpong,..., se encuentra siempre en uno solo de los semicampos y no pasa al campo contrario en ningún momento.

Por lo tanto tenemos como recinto un cuadrado de 10 x 10m.

Si definimos la esquina exterior izquierda como centro de coordenadas (por ejemplo), con valores (0,0) es decir ( $x=0$ ,  $y=0$ ) y siendo la superficie de juego plana no contemplamos eje z, y definimos la esquina inferior derecha como (10, 0), y sabiendo que los ejes en dirección y son perpendiculares podemos ver que geoméricamente sólo va a haber 1 punto de corte en ese semiplano, y por lo tanto NO necesitamos el tercer punto para la trilateración.

### ¿Cómo determinar ese punto de corte?

Planteamos la ecuación general de 1 circunferencia conocido su centro y su radio.

$$(x-x_0)^2 + (y-y_0)^2 = r^2$$

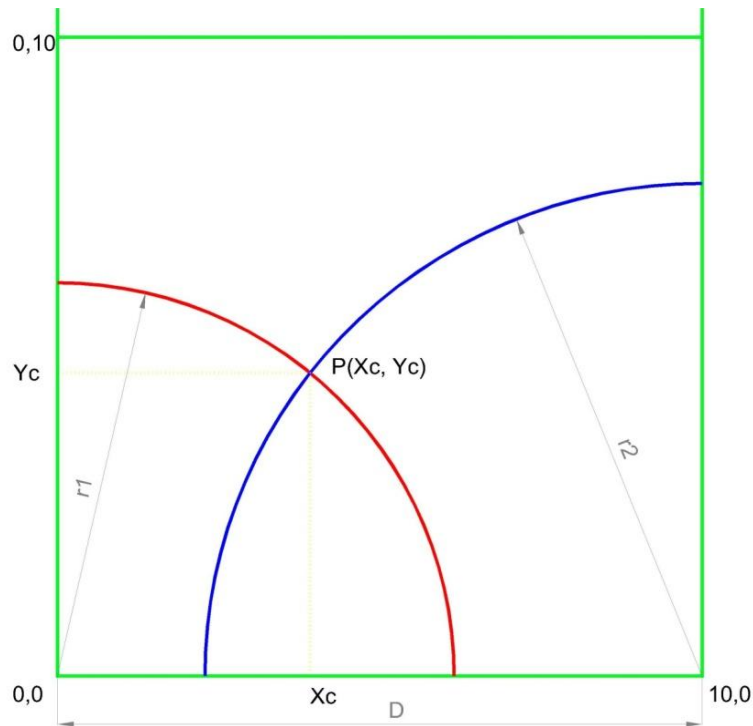
El radio r en nuestro caso lo obtendremos de los valores de RSSI de los Ibeacons.

Para el ibeacon situado en el origen de coordenadas la ecuación será:  $(x_1 = 0, y_1 = 0)$

$$x^2 + y^2 - r_1^2 = 0$$

Para el ibeacon situado en la esquina derecha la ecuación será;  $(x_2 = D = 10, y_2 = 0)$

$$(x-D)^2 + y^2 - r_2^2 = 0$$



**Figura 51: Puntos de corte 2 circunferencias**

El punto de corte de 2 circunferencias puesto que  $X_{c1}=X_{c2}$  e  $Y_{c1}=Y_{c2}$  se determina igualando las ecuaciones, siendo  $X_{c1}=X_{c2}$  e  $Y_{c1}=Y_{c2}$  las coordenadas X e Y de la primera y de la segunda circunferencia donde se van a cortar.

$$x^2 + y^2 - r_1^2 = (x-D)^2 + y^2 - r_2^2$$

$$x^2 + y^2 - r_1^2 = (x^2 - 2x \cdot D + D^2) + y^2 - r_2^2$$

$$x^2 - x^2 + y^2 - y^2 = -2x \cdot D + r_1^2 + D^2 - r_2^2$$

$$2x \cdot D = r_1^2 - r_2^2 + D^2$$

$$X_c = \frac{r_1^2 - r_2^2 + D^2}{2 \cdot D} ; Y_c = \pm \sqrt{r_1^2 - x_c^2}$$

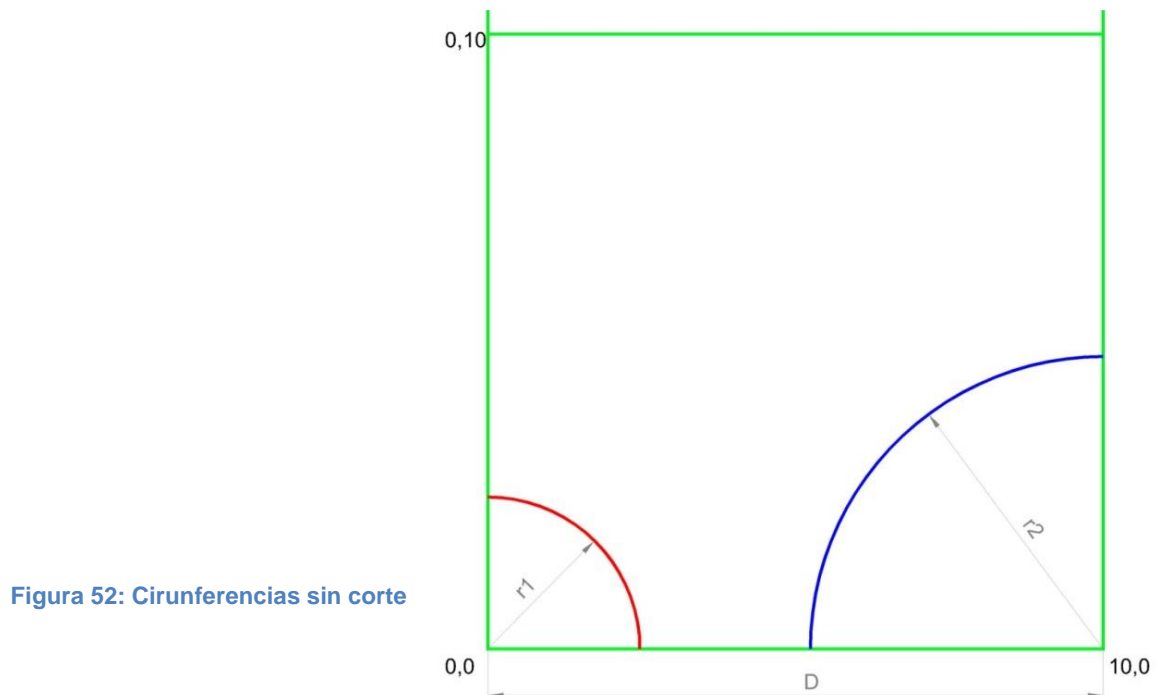
**Fórmula**

La distancia D es fija, en este caso es la distancia eje horizontal entre ambos ibeacons = 10m.

El resultado de la raíz daría valores en eje y positivo y en eje y negativo, y para esa indeterminación se usaría el tercer Ibeacon, sin embargo al partir de que sólo se opera en el primer cuadrante (+, +), los valores de  $Y_c < 0$  no son tenidos en cuenta, ahorrando 1 Ibeacon.

¿Qué ocurre con las indeterminaciones si los valores obtenidos no dan puntos de corte?

Ocurrirá cuando  $X_c > r_1$  o cuando  $X_c > r_2$



Pueden servirnos de guía dos documentos, por un lado el Trabajo de Fin de Carrera "Sistema de posicionamiento móvil para interiores vía WIFI" [31], que se basa por otro lado en la tesis doctoral "Sistema de Localización inalámbrica en entornos de interior mediante el uso del retardo temporal y el nivel de potencia de las señales de radiofrecuencia".

Nos indican que para 2 dimensiones (x,y) la técnica de multilateración se aproxima por el método para determinar la intersección de N circunferencias ( $N \geq 3$ ) con centro en la posición de los "nodos satélites o balizas" y radio la distancia estimada del "nodo cliente" a cada nodo satélite.

La resolución del sistema de corte de N circunferencias se puede reducir a hallar los cortes de sus "ejes radicales", teniendo  $n(n-1)/2$  ecuaciones de ejes.

La ventaja de éste método es que aunque no haya puntos de corte, sí que existirán ejes radicales, con  $N=3$  obtendremos 3 ejes radicales con sólo 2 ejes radicales no paralelos ya obtendríamos el punto de corte buscado.

Si planteamos todas las ecuaciones de los ejes radicales y lo llevamos a su determinante, para que el sistema sea compatible determinado todos los ejes deberían cortarse en un solo punto, pero en la práctica no será así, y obtendremos varios puntos. Según la tesis de Bahillo [33] la solución óptima es el punto que minimiza la suma de distancias al cuadrado a cada uno de los ejes radicales.

Aplicando para de esa teoría a nuestro caso de 2 dimensiones necesitaríamos  $N=3$  (3 ibeacons) y obtendríamos 3 ejes radicales  $(n \cdot (n-1) / 2 = 3 \cdot 2 / 2;$

## Obtención de 3 ejes radicales:

### Eje radical 1-2

Se puede obtener el eje radical como la recta perpendicular a la línea que une los centros de las circunferencias y que pasa por el punto  $E_{12}$  ( $E_{12x}$ ,  $E_{12x}$ ) (Punto de corte del eje radical con el eje X) de esa recta, que en este caso particular es el propio eje de abscisas (eje x) y cuyas coordenadas son del tipo ( $E_{12x}$ , 0).

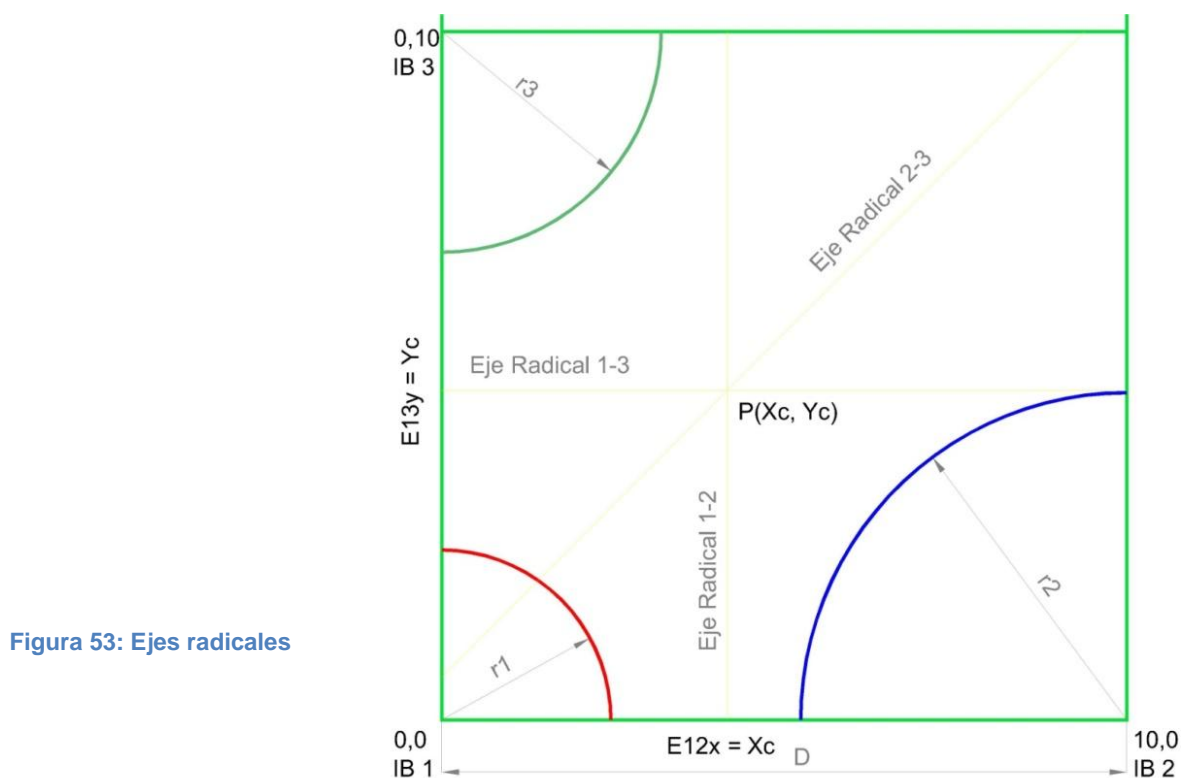


Figura 53: Ejes radicales

Por tanto obtendremos el eje radical de la forma siguiente:

Recta  $1 \cdot X + 0 \cdot Y = E_{12x}$  Por tanto recta  $X = E_{12x}$

### Eje radical 1-3 o Cómo obtener la coordenada Y.

Si utilizáramos una tercera radiobaliza (ibeacon) y la situásemos en el punto (0,10) tendríamos ese tercer punto contra la circunferencia con centro en (0,10) y radio  $R_{ssi}$ .

El eje radical de esa circunferencia respecto a la del Ibeacon (0,0) de forma análoga al anterior:

Recta  $0 \cdot X + 1 \cdot Y = E_{13y}$  Por tanto recta  $Y = E_{13y}$

La solución al problema, sería el punto de corte de ambos ejes, que será:  $P_{c1213}(E_{12x}, E_{13y})$

**Eje radical 2-3;** hay un tercer el formado entre los ibeacons 2 y 3, que se cortará con los 2 anteriores en el mismo punto de corte  $P_{c1223}(E_{12x}, E_{13y}) = P_{c1323}$

D; Distancia entre centros de las circunferencias (En este caso  $D=10$ )

$$E_{12x} = \frac{r_1^2 - r_2^2 + D^2}{2 \cdot D}; E_{13y} = \frac{r_1^2 - r_3^2 + D^2}{2 \cdot D}; D=10 \quad \text{Fórmula}$$

Se aplican las fórmulas y se obtienen los valores buscados siendo esos valores  $r_1$ ,  $r_2$ ,  $r_3$  proporcionales a los datos de RSSI de los 3 Ibeacons.

### 3.2.3 ACTUADORES

#### 3.2.3.1 Drivers para motores desplazamiento

Para transformar las señales para el gobierno de los motores procedentes de la unidad de control principal debemos utilizar "drivers para motor". Los drivers son circuitos interfaz entre la Unidad de Control principal y los propios motores, y sirven para proporcionar arranque, velocidad, freno y sentido de giro. Son capaces de administrar la intensidad y potencia a entregar en cada momento al motor en función de la carga y velocidad que deben entregar.

Existen comercialmente en el mercado múltiples tipos de drivers, especializados en cada tipo singular de motor, para corriente continua con escobillas, tipo paso a paso y para motores sin escobillas (brushless).

Este tipo de driver proporciona una señal de tensión continua p.e 10v y a través de un



Figura 54: Driver WS55-220

potenciómetro se puede regular la velocidad de giro con la tensión en el cursor.

La idea es sustituir el potenciómetro por señales PWM del controlador principal.

Normalmente el uprocesador principal es capaz de controlar este tipo de motores enviando señales de salida de tipo PWM y recibiendo a través de sus puertos de entrada las señales de frecuencia de giro (RPM) así como las procedentes de los sensores Hall.

Por ello el uProcesador tiene 2 funciones respecto al motor, Ordenarle y controlarle.

Sin embargo necesitan una interfase que haga de etapa de potencia que alimente a los motores, y para ello se utilizan los drivers. En este caso hemos decidido usar este dirver inteligente

Los drivers BLDC actúan como interfase entre el microprocesador principal y el motor, de forma que liberan al primero de la tarea de controlar al motor,dedicándose el uProcesador simplemente a Ordenarle. Con ello liberamos gran cantidad de recursos del Procesador a nivel de proceso y de puertos de entrada y salida.

Además estos drivers evitan que lleguen al procesador sobrecargas debido a corrientes de alimentación de los motores, cumpliendo el driver una función de "protección del procesador".

Los drivers deberán ser para motores sin escobillas y cumplir los siguientes requisitos:

<b>a.- Tipo de Motor:</b>	<b>BLDC motores eléctricos sin escobillas</b>
<b>b.- Tensiones alimentación:</b>	<b>36v dc</b>
<b>c.- Control Sensor Interno:</b>	<b>Sensores de efecto Hall</b>
<b>d.- Potencia máxima:</b>	<b>350 W por motor</b>
<b>e.- Salida de control:</b>	<b>Frecuencia de giro (RPM )</b>
<b>f.- Sentido de giro:</b>	<b>Ambos sentidos, inversión</b>
<b>g.- Velocidad del motor:</b>	<b>Regulable</b>
<b>h.- Freno motor:</b>	<b>Sí.</b>
<b>i.- V Señales Entrada:</b>	<b>0, 5v</b>

**Tabla 22: Driver Motores desplazamiento, requisitos**

Para cumplir estas características se ha seleccionado el driver

15A 500W DC12V-36V Brushless Motor Speed Controller BLDC Driver Board with Hall

<b>Tensión de operaciones:</b>	<b>DC 12V-36V</b>
<b>Corriente de operaciones:</b>	<b>&lt;=15A</b>
<b>Potencia máxima Driver:</b>	<b>&lt;=500W</b>
<b>Tensión Velocidad:</b>	<b>0.1V-5V</b>
<b>Temperatura de trabajo:</b>	<b>-40-85°C</b>
<b>Humedad:</b>	<b>10% - 90%RH</b>
<b>Protección bloqueo motor:</b>	<b>Sí</b>
<b>Control Interno Habilitado:</b>	<b>Sí, Sensores Hall</b>
<b>Protección sobrecarga:</b>	<b>Sí</b>
<b>Control Dirección:</b>	<b>Sí, ambos sentidos</b>
<b>Señal control velocidad</b>	<b>Sí, RPM</b>
<b>Tamaño:</b>	<b>6.3x4.2cm</b>

**Tabla 23: 15A 500W DC12V BLDC Driver Board with Hall**



Se necesita 1 controlador por cada motor, por lo tanto 2 drivers controladores para realizar el personal trainer.

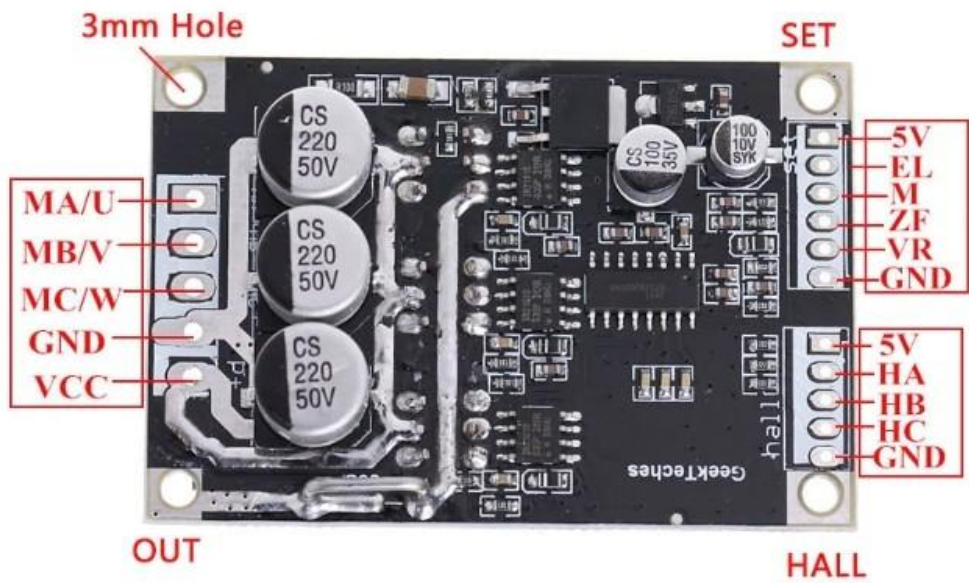


Figura 55: Driver BLDC

Este driver de motor BLDC tiene 3 bancos de señales:  
MA/U, MB/V, MC/W, GND, VCC que gobiernan el motor en sí.  
5V, HA, HB, HC, GND para recoger las señales de los sensores de efecto Hall en el motor.  
5V, EL, M, ZF, VR, GND diferentes señales de entrada / salida.

El esquema básico de control será el siguiente:

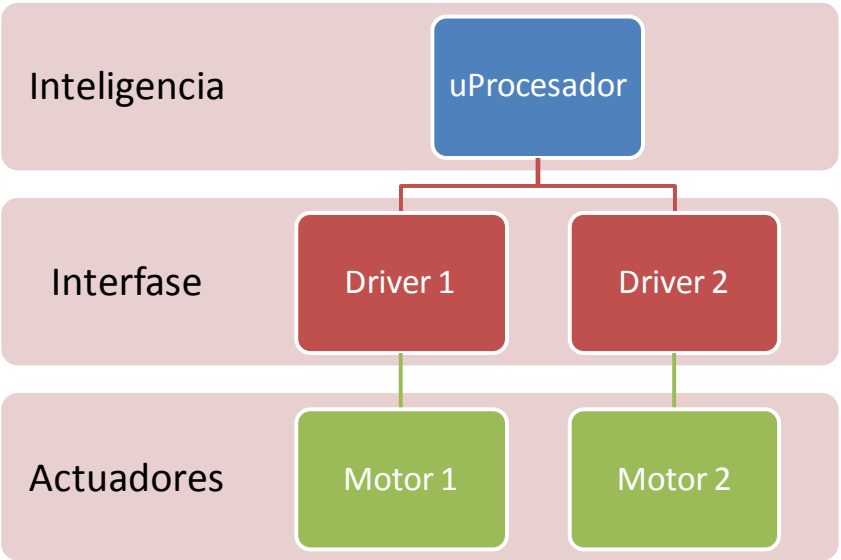


Figura 56: uP, Driver, Motor

Será también de aplicación en el control de los motores de lanzamiento pese a que los motores serán diferentes y por lo tanto también lo serán los drivers, pero el esquema genérico es válido.



### 3.2.3.2 Drivers para motores Lanzamiento

Con idéntica base a lo comentado en el punto anterior para desplazamiento sobre los drivers de motores se procede al diseño del driver y del motor en este punto para el desplazamiento.

¿Qué buscamos para poder lanzar una bola?

Se puede realizar de forma eléctrica con 1, 2 o 3 motores según el tipo de complejidad que necesitemos pedirle al sistema.

Lo principal ahora es la velocidad lineal que el sistema precise dar a la bola a la salida del lanzador y que estará directamente relacionada con la velocidad de giro del motor y también con el radio de aplicación de ese giro.

En este caso los motores más simples que pueden entregarnos altas velocidades de giro y un par motor (mediante buen torque) aceptable son los motores dc con escobillas. Este tipo de motor no posee en general entradas de control, sensores hall.....

¿Cómo regular su velocidad?

Se realiza modificando la tensión y corrientes aportadas al motor y nuevamente nos ayudaremos de un driver, pero en este caso el apropiado será para corriente continua.

Como los del punto anterior este tipo de driver proporciona una señal de tensión continua p.e 10v y a través de un potenciómetro se puede regular la velocidad de giro con la tensión en el cursor, y vamos a cambiar su potenciómetro por señales PWM del controlador principal.

El uprocesador principal controlará este motor enviando señales de salida de tipo PWM y recibiendo a través de sus puertos de entrada las señales de frecuencia de giro (RPM).

Por ello el uProcesador tiene 2 funciones respecto al motor, Ordenarle y controlarle.

El driver impedirá sobrecargas debido a corrientes de alimentación de los motores, cumpliendo el driver una función de "protección del procesador".

El driver deberá ser para motor con escobillas y cumplir los siguientes requisitos:

<b>a.- Tipo de Motor:</b>	motores eléctricos dc con escobillas
<b>b.- Tensiones alimentación:</b>	<b>36v dc</b>
<b>c.- Control Sensor Interno:</b>	No habilitado
<b>d.- Potencia máxima:</b>	<b>60 W por motor</b>
<b>e.- Salida de control:</b>	Frecuencia de giro (RPM )
<b>f.- Sentido de giro:</b>	<b>1 sentido</b>
<b>g.- Velocidad del motor:</b>	Regulable
<b>h.- Freno motor:</b>	No
<b>i.- V Señales Entrada:</b>	0, 5v

**Tabla 24: Driver Motores desplazamiento, requisitos**

Existen en el mercado una gran variedad de drivers para este tipo de motores, buscaremos los más adecuados en función de su relación prestaciones-calidad frente a precio. Unas características complementarias son display de % de RPM y pulsador de marcha-paro físico para poder detenerlos en caso de fallo del sistema de control electrónico principal.

Para cumplir estas características se ha seleccionado el driver modelo E15A1AA700381 PWM DC Motor Speed Controller del fabricante OOTDTY

Tensión de operaciones:	DC 6V-60V
Corriente de operaciones:	0<=30A, < 20A régimen continuo
Potencia máxima Driver:	<=1.200W
Tensión Velocidad:	0-10V
Temperatura de trabajo:	-40-85° C
Humedad:	10% - 90%RH
Protección bloqueo motor:	Sí
Control InternoHabilitado:	No
Protección sobrecarga:	Sí
Control Dirección:	Sí, ambos sentidos
Señal control velocidad	Sí, RPM
Tamaño:	9.5 x 6.0cm

Tabla 25: E15A1AA700381DC Driver Board with Display

Se necesita 1 controlador por cada motor, por lo tanto 1driverpara el lanzador.

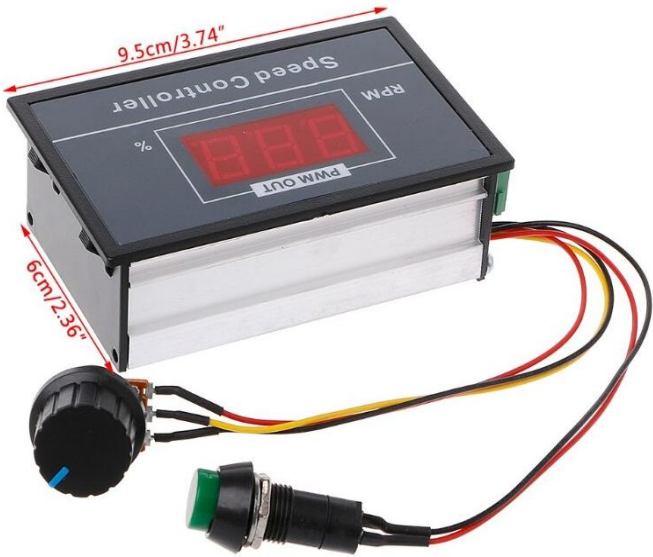


Figura 57: Driver Motor D.C. Lanzabolas

Este driver de motor DC tiene 2 bancos de señales:  
GND, VCC que gobiernan el motor en sí.  
10V, EL, M, ZF, VR, GND diferentes señales de entrada / salida.

### 3.4 Desarrollo de Inteligencia

Una vez que se ha realizado el diseño físico, el eléctrico y el electrónico es necesario dotar al dispositivo de un sistema capaz de reaccionar a los estímulos de entrada que recibirá a través de los sensores externos (análisis del juego) e internos (como se encuentra la máquina) elaborando respuestas.

Cualquier autómatas es capaz de realizar operaciones prefijadas y que en función de las variables de entrada, de salidas anteriores y de variables internas serán elaboradas y ejecutadas por el automatismo. Cuando las tareas son más complejas, se analizan más datos empieza a hablarse de inteligencia.

Entre las acepciones que la RAE [28] acepta como inteligencia tenemos:

1. f. Capacidad de entender o comprender.
- 2. f. Capacidad de resolver problemas.**
3. f. Conocimiento, comprensión, acto de entender.
4. f. Sentido en que se puede tomar una sentencia, un dicho o una expresión.
- 5. f. Habilidad, destreza y experiencia.**
6. f. Trato y correspondencia secreta de dos o más personas o naciones entre sí.
7. f. Sustancia puramente espiritual.

Por lo tanto sí podemos hablar de inteligencia en una máquina capaz de resolver un problema.

Un matiz para asemejar la máquina a lo humano es la variabilidad, es decir que no siempre se responda de la misma manera ante el mismo planteamiento. Esto en las máquinas se puede conseguir mediante algoritmos que introduzcan cierto grado de aleatoriedad.

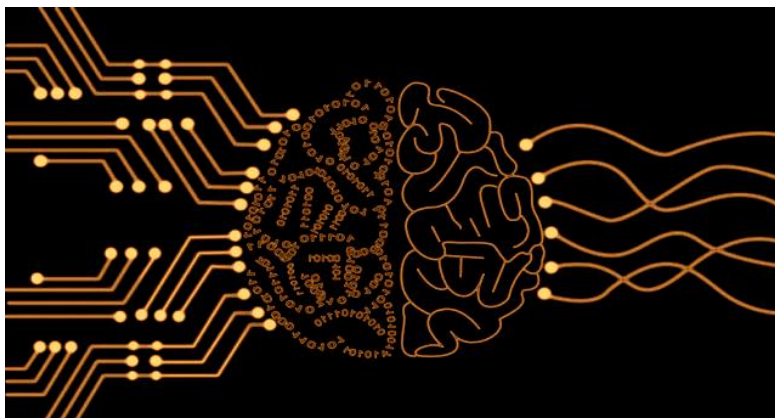


Figura 58: IA

El Procesador elegido Intel Edison es capaz de funcionar en modo multiusuario y multitarea, esta última característica la vamos a aprovechar para ejecutar simultáneamente varios programas que permitan simular el comportamiento de un humano (a pequeño nivel).

Por un lado se ejecutará un programa cíclico que permita que la máquina esté viva y realice sus funciones vitales mínimas. (En un humano puede ser el gobierno del corazón, los pulmones).

Por otro lado un programa basado en algo distinto y que permita a la máquina recibir estímulos del mundo exterior (los sentidos humanos, vista, oído,.....)

El cerebro debe elaborar respuestas y través de las redes neuronales enviarlas para ejecutar respuestas (movimiento, impulsión)..

Y por último lo que nos más identifica al ser humano es la capacidad de comunicarse con su entorno y esto se realizará a través de la interfase de usuario.

Selección de software para inteligencia del entrenador personal.

#### 1.- Funciones vitales:

Se va a usar un programa que se ejecutará nada más iniciarse la máquina de forma automática. El sistema operativo lo cargará como un proceso en segundo plano y dotará de esos mínimos de operatividad interna de la máquina. Además la dejará lista para comunicarse con otros programas.

Este programa se elaborará bajo entorno wiring [29], que es el lenguaje de Arduino, basado en el proyecto open source "Processing". [30]

Wiring es una plataforma, desarrollada en Java al igual que Processing, pero añadiendo un compilador GCC para C/C++.

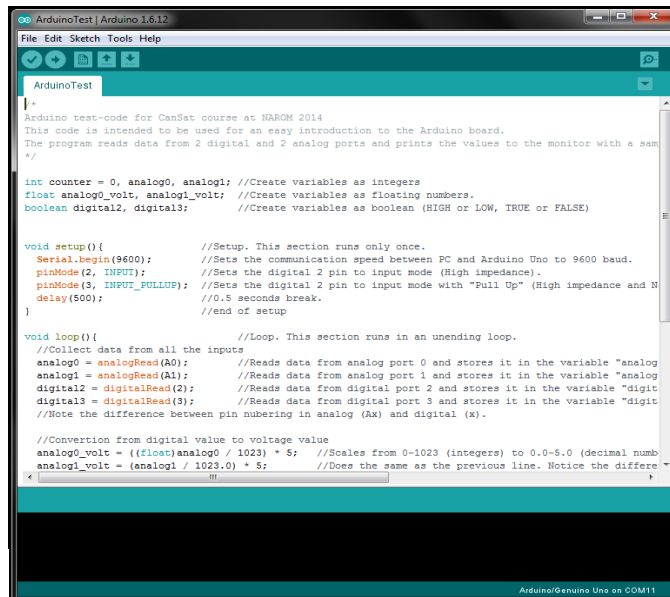


Figura 59: Arduino Software

En realidad el entorno real de programación de arduino recoge características de ambos a los que añade propias pues Arduino está orientado a operaciones de entradas y salidas, puertos.

Es decir a la parte física y electrónica, por ello Edison añade la parte inteligente.

El programa se elaborará como un Sketch de arduino y será compilado y subido al edison donde se ejecutará nada más iniciarse la máquina.

#### 2.- Estímulos exteriores, preparación de respuestas:

Se van a realizar a través de un programa realizado bajo Java [33] y Nodejs [34] y que se llamará desde el propio inteledison, bajo el nombre app.js.

Este programa se comunica internamente con el programa de funciones vitales del punto 1.

#### 3.- Respuestas neuronales:

Al igual que ocurre en el cuerpo humano de la interacción de los 2 programas anteriores, cerebro y cerebelo, se enviarán respuestas a través de la red neuronal a los actuadores, las ruedas (piernas) y el lanzador (brazos) pero también a los controles de energía (baja batería...)

#### ¿Cuáles son las principales respuestas que debe realizar la máquina?

Pues debe entrenar al deportista, mediante el lanzamiento de bolas.

Por lo tanto la inteligencia estará en saber dónde se encuentra, analizar las señales que llegan y traducirlo en acciones.

Por ello el desarrollo de inteligencia estará plasmado en calcular y mandar ejecutar:

Movimiento, la orientación, el Calibrado automático, el lanzamiento.....

### 3.4.1 MOVIMIENTO

Es la acción para cambiar de lugar, bastará con accionar los motores en cualquiera de los sentidos, durante un intervalo de tiempo  $t$ , para que una vez transcurrido el entrenador personal se encuentre un sitio diferente al inicial.

Por lo tanto hay 3 datos fundamentales, posición inicial  $s_1$ , posición final  $s_2$  y tiempo  $t$ .

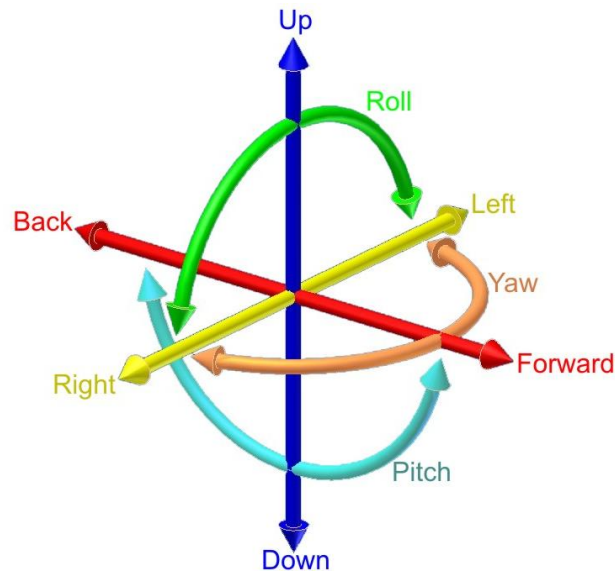


Figura 60: Movimiento

Tendremos los grados de libertad de movimiento [35] restringidos, a 2 ejes ( $z=0$ ).

Si consideramos la descomposición en cada uno de los ejes ( $x$ ,  $y$ ), y encada uno de esos componentes el movimiento más simple, un rectilíneo uniforme, estará relacionado así:

$s_2 = s_1 + v \cdot t$ , donde  $v$  es una variable, la velocidad lineal.

Es evidente que tendremos movimientos más complejos, que pasarán por aceleraciones positivas o negativas,  $s_2 = s_1 + v \cdot t + \frac{1}{2} a \cdot t^2$

**Fórmula**

Como tenemos 2 ruedas motrices diferentes, podemos conseguir diferentes trayectorias incluso curvas mediante combinaciones de sus movimientos (velocidades angulares) mientras se tracciona.

Es muy importante el concepto de rodar sin deslizar [35]

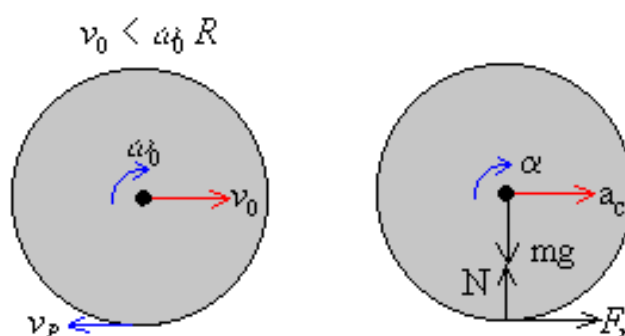


Figura 61: Rodar

### **Justificación, tipos de direccionamiento:**

Se ha consultado el TFG "Control de un Robot con ruedas de giro limitado" [35] basada en la tesis "Analytical Configuration of Wheeled Robotic Locomotion" [36].

En ambas se postula que hay 4 tipos de direccionamiento para un robot con ruedas:

- 1.- Articulado (Articulated steering) se gira parte del chasis.
- 2.- Coordinado (Coordinated steering), Se gira mecánicamente el eje de ambas ruedas
- 3.- Independiente (Independent), cada rueda se dirige individualmente, alta maniobrabilidad.
- 4.- Por derrape (Skid steering), diferente velocidad en las rueda de cada lado. El eje No rota.

En un giro se crea un ICR, Centro Instantáneo de Rotación en torno al que se desarrolla una trayectoria con la curvatura buscada.

Las hipótesis más simplistas se basan en considerar 2 hipótesis de partida; las ruedas no derrapan y las ruedas no deslizan, el movimiento se basa en rozamiento puro.

La máquina utiliza **Geometría diferencial, un solo eje**, por ello podemos definir un ICR válido bajo las hipótesis de no derrape y no deslizamiento, que pertenece a un punto del eje.

### **Movimientos básicos:**

- 1.- Hacia adelante constante:

Los 2 motores giran en el mismo sentido (horario visto desde la derecha), a las mismas revoluciones que se mantienen constantes.

- 2.- Hacia atrás constante: (marcha atrás)

Los 2 motores giran en el mismo sentido (antihorario visto desde la derecha), a las mismas revoluciones que se mantienen constantes.

- 3.- Acelerar hacia adelante:

Los 2 motores giran en el mismo sentido (horario visto desde la derecha), a las mismas revoluciones que se incrementan durante el periodo de aceleración.

- 4.- Acelerar hacia atrás:

Los 2 motores giran en el mismo sentido (antihorario visto desde la derecha), a las mismas revoluciones que se incrementan durante el periodo de aceleración.

- 5.- Frenar:

Los 2 motores giran en el mismo sentido, a las mismas revoluciones que se decrementan durante el periodo de frenado.

- 6.- Rotación izquierda

Con el dispositivo parado los motores giran en diferente sentido (el izquierdo antihorario (retroceso), el derecho horario (avance) vistos desde la derecha), revoluciones que primero se incrementan y luego se frenan durante el periodo de rotación.

#### 7.- Rotación derecha

Con el dispositivo parado los motores giran en diferente sentido (el izquierdo horario (avance), el derecho antihorario (retroceso) vistos desde la derecha), revoluciones que primero se incrementan y luego se frenan durante el periodo de rotación.

#### 8.- Giro izquierda

Con el dispositivo en movimiento, si estamos yendo hacia adelante los motores giran en el sentido (horario (avance) vistos desde la derecha), pero a diferentes velocidades, siendo mayor la de la rueda derecha. El radio de giro va a depender de la diferencia de velocidades.

Si el dispositivo está marcha atrás, el sentido de giro es antihorario visto desde la derecha.

#### 9.- Giro derecha

Con el dispositivo en movimiento, si estamos yendo hacia adelante los motores giran en el sentido (horario (avance) vistos desde la derecha), pero a diferentes velocidades, siendo mayor la de la rueda izquierda. El radio de giro va a depender de la diferencia de velocidades.

Si el dispositivo está marcha atrás, el sentido de giro es antihorario visto desde la derecha.

#### 10.- Paro.

Se frenan ambas ruedas hasta que la velocidad de giro de ambas sea nula.

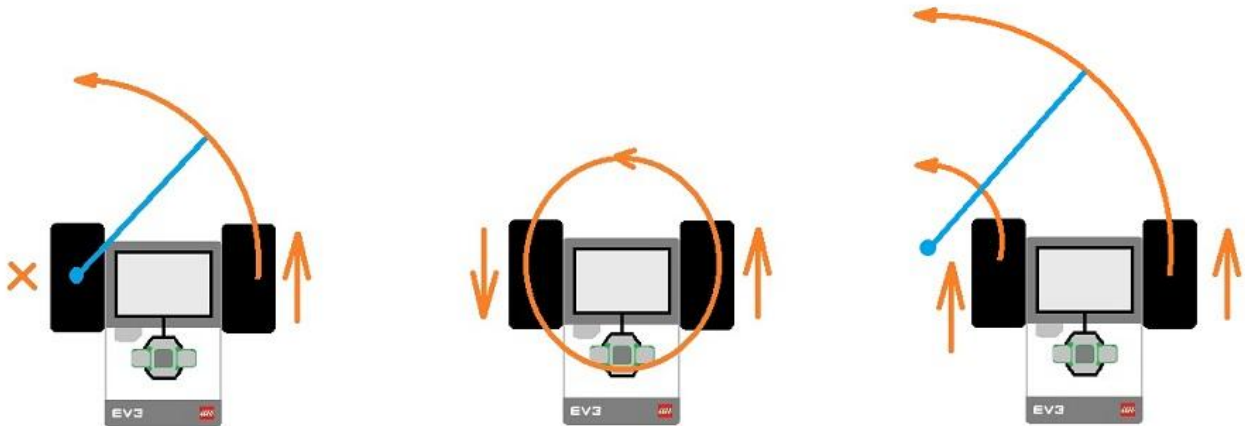


Figura 62: Girar

Los movimientos básicos que se van a realizar para simplificar la programación son:

Adelante, Atrás, Rotación derecha, Rotación izquierda, Paro.

#### Adelante, Atrás

**Para recorrer la trayectoria recta de longitud S**

$$S = n^{\circ} \text{ Vueltas rueda} \cdot 2 \cdot \text{Pi} \cdot r = n \cdot \text{Pi} \cdot \varnothing$$

Por lo tanto para una trayectoria recta, el n° de vueltas será:

$$n = \pi \cdot \frac{S}{\varnothing}$$

Fórmula

## Rotación (Giros)

La rotación no tiene porqué ser de vuelta completa, sino que será de un ángulo 0-360°.

Se considerarán movimientos rectilíneos uniformes para simplificar el problema, y se hará contando vueltas de cada una de las ruedas.

Es sabido que una trayectoria en giro puro se puede aproximar por una circunferencia caracterizada por un punto de centro P y por un radio R. El desarrollo de la misma será

$sR(m) = 2 \cdot \pi \cdot R$ , esta es la distancia a recorrer para obtener un giro completo.

$sR(m) = 2 \pi \cdot \text{"ancho"}$ . Ambas ruedas tendrán el mismo radio (r) y en cada giro su desarrollo:

$sR(m) = 2 \cdot \pi \cdot r$ . De esta forma podemos aproximar cuantas vueltas de rueda necesitamos para realizar un giro completo de la máquina.

$sR = sR \cdot n^\circ \text{ Vueltas rueda}$ ,  $2 \cdot \pi \cdot R = 2 \cdot \pi \cdot r \cdot n$  por tanto se deduce que:  
 $n^\circ \text{ vueltas} = R/r$ , se analizan 2 casos:

a.- Sólo gira una de las ruedas mientras la contraria permanece fija;

Simplificamos como circunferencia con centro en la rueda contraria que permanecería fija, mientras que la que gira será la exterior y deberá realizar todo el desplazamiento. Si la separación entre ruedas es "L eje" (m) tendremos:

$n^\circ \text{ vueltas} = R/r$ ,  $= L/r$  siendo  $R=L$

b.- Ambas ruedas giran con idéntica velocidad pero en sentido contrario;

Simplificamos como circunferencia con centro en el punto medio del eje, y radio el semieje:

$n^\circ \text{ vueltas} = R/r$ ,  $= L/2 / r$  siendo  $R= L/2$  Por tanto  $n^\circ \text{ Vueltas} = R/2 \cdot r$

Vamos a expresarlo en función del ángulo de rotación deseado ( $\alpha$ ):

1 Vuelta completa son 360°, para conseguir giros menores: Factor = ángulo / 360°

De este modo tendremos:

$n^\circ \text{ vueltas} = \frac{\alpha}{360^\circ} \cdot \frac{L}{2 \cdot r}$ ;  $2 \cdot r = \emptyset$  (Diámetro Rueda) y  $\alpha = \text{ángulo de giro deseado}$

$$n = \frac{\alpha}{360^\circ} \cdot \frac{L}{\emptyset}$$

Fórmula



Resumen hipótesis que se aplicarán en este modelo de movimiento:

<b>Direccionamiento</b>	SkidSteering, Diferencial
<b>Fricción</b>	Pura
<b>Derrapaje</b>	No permitido (hipótesisI )
<b>Deslizamiento</b>	No permitido (hipótesis II)
<b>Nº Ruedas en Giro</b>	2 (sentidos contrarios de giro)
<b>Separación entre ruedas</b>	L
<b>Radio de Giro</b>	L/2
<b>Diámetro ruedas Ø</b>	6,5"
<b>Nº vueltas por rueda ( giro<math>\alpha^\circ</math> )</b>	$\frac{\alpha}{360^\circ} \cdot \frac{L}{\varnothing}$

Tabla 26: Giro

<b>Direccionamiento</b>	SkidSteering, Diferencial
<b>Fricción</b>	Pura
<b>Derrapaje</b>	No permitido (hipótesisI )
<b>Deslizamiento</b>	No permitido (hipótesis II)
<b>Nº Ruedas en Giro</b>	2 (mismo sentido de giro)
<b>Separación entre ruedas</b>	L
<b>Distancia a recorrer</b>	S
<b>Diámetro ruedas Ø</b>	6,5"
<b>Nº vueltas por rueda</b>	$\pi \cdot \frac{S}{\varnothing}$

Tabla 27: Movimiento rectilíneo (Adelante, Atrás)

### 3.4.2 EQUILIBRIO

Para estudiar el equilibrio sobre un sistema con sólo 1 eje y 2 ruedas se ha estudiado el TFM de Jorge Ferrer Gómez , "Diseño e Implementación de un Robot Autobalanceado" [37].

Tras el análisis de ese trabajo..... son necesarios acelerómetros, inclinómetros..... que no se justifican en el actual proyecto porque pueden sustituirse por una simple "rueda loca".



Figura 63: Rueda Loca

El concepto de rueda loca (casterwheel) es usar una rueda sin motor, que es capaz de girar libremente en todas las direcciones y en un modelo teórico ideal sin alterar el movimiento provocado por las ruedas motrices (hipótesis de trabajo que vamos a asumir)  
Gracias a esta rueda evitaremos cabeceos en los cambios de dirección y lograremos una buena estabilidad con un bajo coste, premisa principal de este TFM.

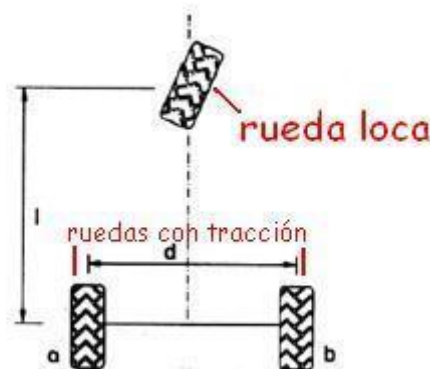


Figura 64: Tracción

Colocaremos 2 ruedas locas alineadas en un eje que será perpendicular al eje de tracción, con la ventaja de dar solamente apoyo sin modificar las ecuaciones de giro y desplazamiento.

### 3.4.3 CALIBRACIÓN

Como propone Jorge Ferrer, es imposible tener idéntica velocidad en 2 motores diferentes gobernados por un sistema PWM, debido a pequeñas diferencias de señal, de los drivers, de fabricación inherentes al propio motor, por lo que debemos igualar las desviaciones para controlar esa diferencia de velocidad. Un método sencillo, que usaremos es un control proporcional maestro-esclavo.

Suponemos un motor 1 como maestro, el segundo como esclavo y medimos ambas velocidades, tomando como referencia V1 y haciendo que V2 deba converger hacia V1.

Se logra mediante la resta (el error) entre ambas:

$$\text{Error}_v = 100 - \frac{v_1}{v_2} \cdot 100$$

Fórmula

Existen métodos más complejos y precisos pero exceden la necesidad del presente trabajo.

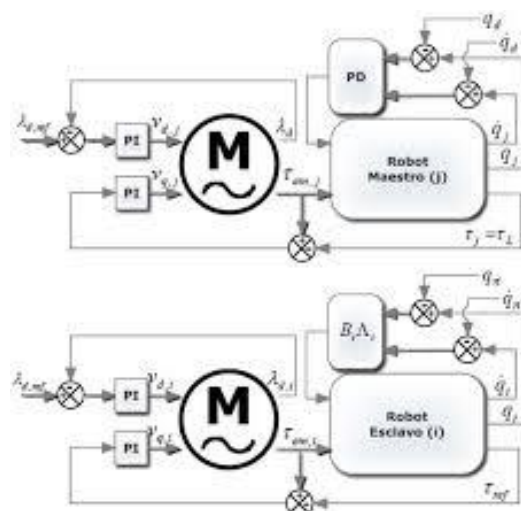


Figura 65: Control Maestro-Esclavo

### 3.4.4 ORIENTACIÓN

Dado que nuestro dispositivo va a estar dotado de unos sensores para detectar obstáculos, que son capaces de medir distancias a los mismos, y como el perímetro de nuestra área está cerrado, podemos usar esas sondas y los datos que proporcionan para obtener las distancias a las paredes y mediante substracción de 2 medidas comprobar la dirección respecto a las paredes. Para completar la orientación nos ayudaremos de la energía recibida en las señale de los ebeacons.

Avance perpendicular a las paredes, lo obtenemos cuando la distancia de 2 sondas paralelas, hasta la pared sea la misma, es decir su resta iguala a cero y con el movimiento hacia delante o atrás, la diferencia entre ambas distancias sea cero.



Figura 66: Orientación, brújula

Para determinar respecto a qué pared estamos avanzando seguiremos los criterios siguientes:

- 1.- Pared 1, fondo pista, coordenadas (0,0) - (10, 0);  
Vamos hacia ella cuando el Nivel de energía de las sondas 1 y 2 se incrementa y mientras se decrementa el nivel respecto de la sonda 3.
- 2.- Pared 2, Lateral izquierdo, coordenadas (0,0) - (0,10)  
Vamos hacia ella cuando el Nivel de energía de las sondas 1 y 3 se incrementa y mientras se decrementa el nivel respecto de la sonda 2.
- 3.- Pared 3, Lateral derecho, coordenadas (10,0) - (10,10)  
Vamos hacia ella cuando el Nivel de energía de las sondas 1 y 3 se decrementa y mientras se incrementa el nivel respecto de la sonda 2.
- 4.- Pared 4, red, coordenadas (0,10) - (10,10)  
Vamos hacia ella cuando el Nivel de energía de las sondas 1 y 2 se decrementa y mientras se incrementa el nivel respecto de la sonda 3
- 5.- Cuando el nivel de energía de la sonda 1 sea similar al de la sonda 2, estará avanzando por el centro de la pista en sentido subida bajada
- 6.- Cuando el nivel de energía de la sonda 1 sea similar al de la sonda 3 estará moviéndose a media pista en dirección izquierda-derecha (el sentido depende de los puntos 2, 3)

7.- Cuando el nivel de las 3 sondas sea el mismo, estará en el centro geométrico de la pista, cruce de las 2 diagonales de la semipista.

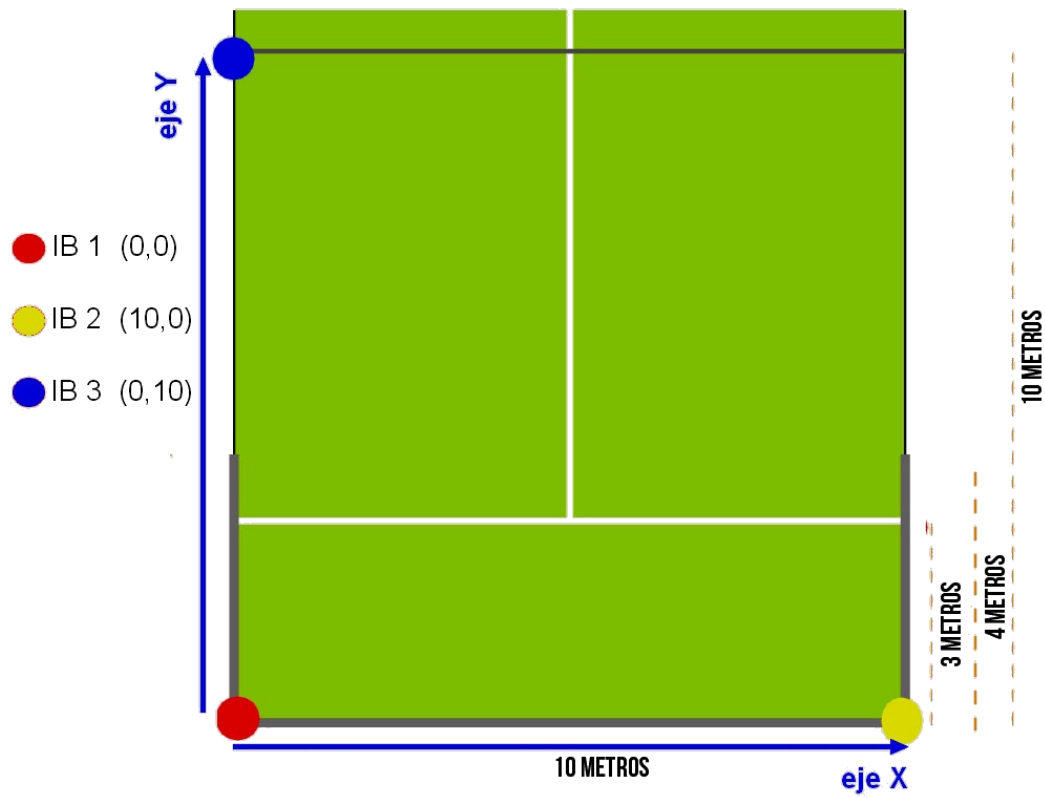


Figura 67: Distribución Ibeacons

### 3.4.5 INTENSIDAD DEL ENTRENAMIENTO

La intensidad del ejercicio puede depender de las variables del entorno como la temperatura, humedad, la hora del día, la época del año y de parámetros como la duración del entrenamiento, y de las condiciones del deportista como la edad, el sexo, y del tipo de preparación que se busca. Además, depende de sus constantes vitales entre las que se encuentra la frecuencia cardíaca.

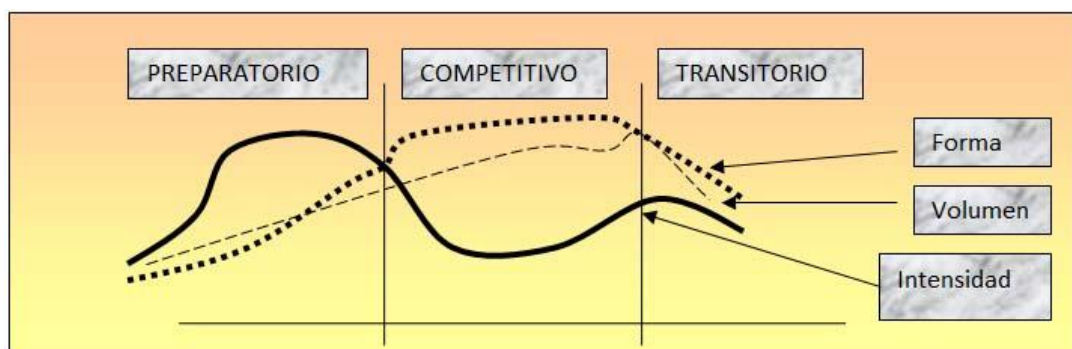


Figura 68: Intensidad, tipos

La frecuencia cardíaca será monitorizada por una sonda BLE portada por el deportista a través de un cinturón ajustable que contiene electrodos y que irá colocada en el pecho. Esta cinta BLE transmite cada 5s información sobre las pulsaciones del entrenado.

No hace falta estar todo el tiempo monitorizando las constantes, pero sí se pueden leer antes de ejecutar el siguiente ejercicio, o a intervalos ya sean regulares o aleatorios para comprobar que la persona se encuentre por debajo de unos niveles de seguridad.

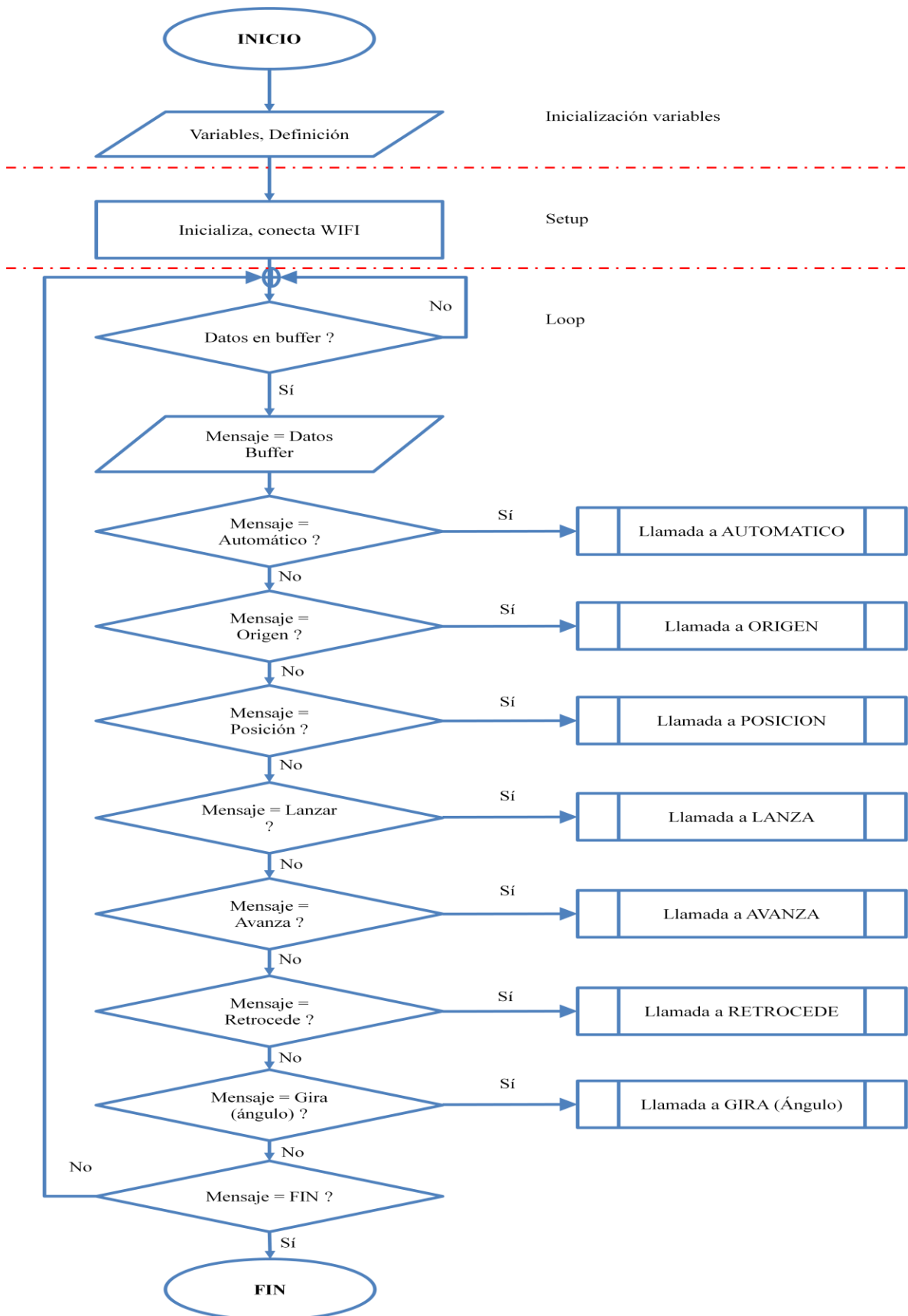
Intensidad	Duración	Efectos
<b>Máxima</b> 90% - 100%	0 - 2 min	Aumenta la velocidad la potencia y la fuerza <small>*Si no eres un deportista de élite ten cuidado al llegar a esta zona de entrenamiento</small>
<b>Intensa</b> 80% - 90%	2 - 10 min	Mejora la resistencia anaeróbica e incrementa la fuerza y resistencia
<b>Moderada</b> 70% - 80%	10 - 40 min	Mejora la resistencia aeróbica y la forma física en general
<b>Suave</b> 60% - 70%	40 - 80 min	Quema grasas y mejora la resistencia aeróbica
<b>Muy suave</b> 50% - 60%	20 - 40 min	Acelera la recuperación tras un ejercicio intenso

Figura 69: Intensidad Efectos

En caso de superarlos el ejercicio se suspenderá automáticamente y se puede enviar una señal de alerta a los dispositivos conectados, o un del entrenador a través de un buzzer (zumbador). Cuando el deportista se encuentre cercano a los niveles de seguridad, el entrenamiento debe suavizarse, con golpes más lentos y menor cadencia. Por el contrario cuando el jugador se encuentre muy por debajo de esos niveles, entonces la máquina deberá incrementar la intensidad del entrenamiento.

## Pseudocódigo de Programación para la Inteligencia (Consultar anexo I)

Este será el programa principal que gobierna la máquina eléctrica y mecánicamente.



Pseudocódigo 1: Programa Principal

## 4.-Interfase con usuario

La comunicación con el usuario (exterior) y también la que realice con su propio entorno (interior) se realizará a través de la interfase de usuario y estará basado en:

- a.- Un servidor WEB.
- b.- Node.js
- c.- Java
- d.- HTML, página Web

### 4.1 Servidor WEB

a.-Un servidor WEB, creado en el propio Intel Edison.

```
const http =require('http');
const hostname ='127.0.0.1';
const port =3000;
const server =http.createServer((req, res)=>{
  res.statusCode=200;
  res.setHeader('Content-Type','text/plain');
  res.end('Hola Mundo\n');
});
server.listen(port,hostname,()=>{
  console.log(`El servidor se está ejecutando en http://${hostname}:${port}/`);
});
```

Figura 70: Servidor Web, código node js

Intel Edison trae un servidor Web por defecto que realmente no hace nada, simplemente uno se puede conectar vía web en la dirección `http://intel_edison_IP:80`.

Siendo `intel_edison_IP` una dirección interna de red , por lo tanto Privada yNo pública, del tipo 192.168.1.35 y el puerto por defecto, el 80.

Se inhabilitará este servidor y se creará uno específico para la aplicación que se alojará en el propio Intel Edison en su memoria de almacenamiento.

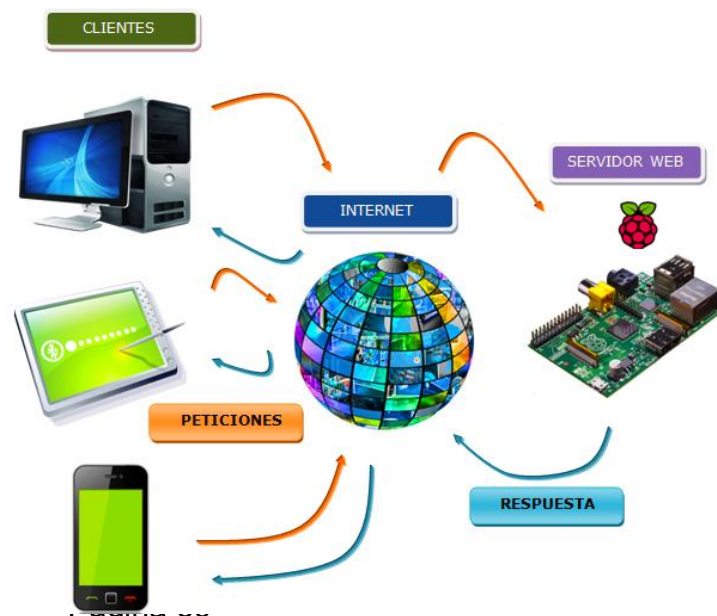


Figura 71: Servidor Web Plataformas

El usuario podrá acceder a este servidor desde cualquier dispositivo que tenga un navegador compatible, que soporte HTML, JAVA, ya sea desde PC, portátil, tableta, teléfono inteligente,..., y desde los diferentes sistemas operativos de estos aparatos, Android, Windows, Linux,.....

## 4.2 NODE.JS

Este entorno en tiempo de ejecución, es de código abierto y multiplataforma, basado en el lenguaje de programación ECMAScript para el Servidor, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Se usa como herramienta de programas de red altamente escalables, y lo desarrolló Ryan Dahl en 2009 y después por Joyent.



Figura 72: Node.JS Logo

El código en node.js se ejecuta en el propio servidor, por ello utiliza Intel Edison que es capaz de procesar en paralelo múltiples accesos de usuarios, aunque no es imprescindible.

Para este proyecto, el uso de Node.js proporcionará un juego de instrucciones compatibles con el entorno java para poder enviar y recibir datos a otras aplicaciones que operen con node.js

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 1337;
http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Figura 73: Ejemplo de servidor

## 4.3 JAVA

Se utilizará este conjunto de utilidades y herramientas para desarrollar código que permita la comunicación entre la máquina y el usuario de la misma, pero también entre procesos internos de la propia máquina, e incluso su interacción con el servidor WEB:

Para ello se lanzará un programa desarrollado en java, alojado en el servidor WEB que se ejecutará bajo el nombre app.js

Además habrá una página web asociada, que estará alojada en el servidor en el propio Intel Edison pero que cada usuario ejecutará en su propio dispositivo y navegador.



## 5.- Integración (implementación técnica)

Se va a construir un modelo prototipo del entrenador personal, basándonos en Procesador Intel Edison que cumple todas las características especificadas en la etapa de diseño y debe comunicarse con las sondas externas e internas, siguiendo la secuencia lógica de la misma, empezando por la parte física del Intel, su puesta en marcha, su preparación, y su programación.

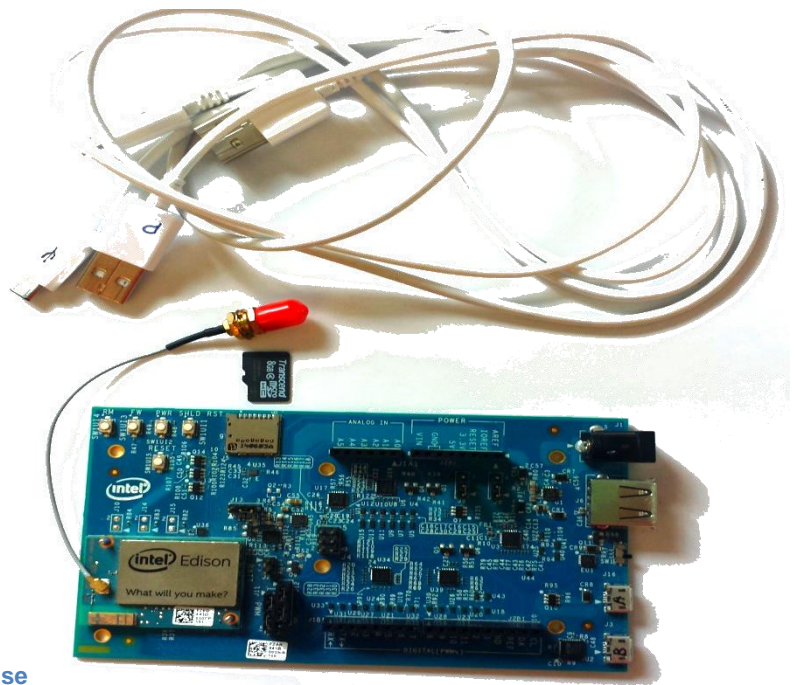
### 5.0 Intel Edison Puesta en marcha y programación

#### PASO 1

Lo primero es sacar el uProcesador de su caja, retirar todos los precintos y protecciones y comprobar visualmente que no ha recibido golpes o daños en transporte.

En segundo lugar se realiza idéntica operación con el tablero del Kit de arduino para Edison.

En tercer lugar se inserta el Intel Edison (uProcesador) en su placa Madre (Kit Arduino).



**Figura 74: Ensamblando Chip y Placa base**

#### PASO 2

Para poder encenderlo necesitamos una fuente de alimentación externa, y para poder conectarlo precisamos de algún cable de tipo USB (al menos la primera vez).

Aunque no es imprescindible si se recomienda el uso de una tarjeta de memoria adicional de tipo Micro SD

Como fuente de alimentación es posible cualquier entrada USB incluyendo la de los propios Pcs, por ello en la foto podemos observar que vamos a emplear 2 cables USB - uUSB.

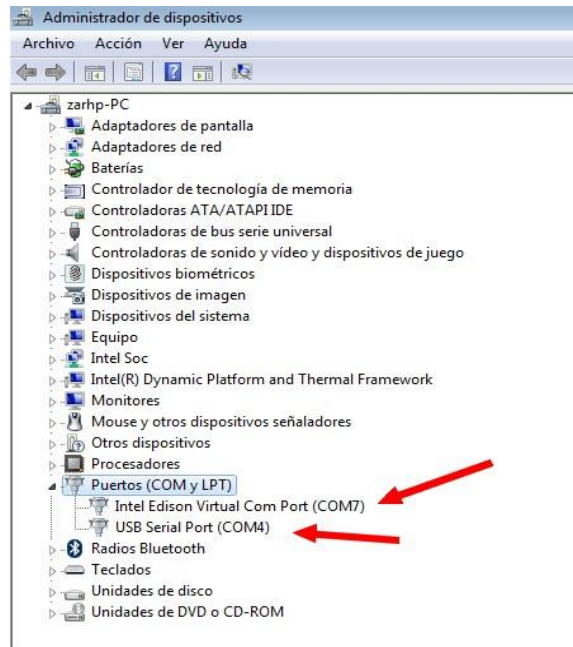
Una vez conectado al PC (la primera vez al menos) procederemos a encender el equipo informático.

### PASO 3

Lo primero que hará el sistema será detectar nuevo Hardware y nos pedirá que instalemos sus drivers. Bajo S.O. Windows, se instalarán varios drivers y además se habilitarán 2 puertos nuevos de comunicación del sistema, por ejemplo COM4 y COM6.

1 Puerto es de comunicación del propio Intel Edison, el otro es para su "monitor serie".

a continuación habrá que instalar el Sistema Operativo del uProcesador.



**Figura 75: Edison, instalando bajo Windows**

El sistema operativo es un núcleo Linux compilado bajo el nombre de Yocto que está dimensionado con herramientas para este procesador. Para realizar esa operación basta con descargarse en el PC la imagen más actual del núcleo y grabarla en una unidad virtual que los drivers han permitido crear al S.O, como si fuera un pendrive (p.e. F: ).

Al reiniciar Intel Edison en él se instalará el nuevo núcleo y ya está listo para ser configurado.

### PASO 4

Ya es posible acceder y programar Intel Edison como si fuera un simple Arduino, para ello hay que descargar, instalar y ejecutar el programa "Arduino IDE" desde la página web: <https://www.arduino.cc/>

La versión más reciente en la fecha de publicación de este trabajo es "ARDUINO 1.8.9" con versiones disponibles para Windows, MAC y Linux.

Al instalar el software se puede configurar la placa Intel Edison y además numerosas librerías extras a las que por defecto ya incluye el programa.

Desde este IDE de Arduino podemos crear Sketch, que son programas en un lenguaje propio, similar al C++. Poseen una estructura particular propia con el formato siguiente:

- 1.- Comentarios y Definición de variables
- 2.- Setup void setup() { // }

### 3.- Bucle de ejecución `void loop() { // put your main code here, to run repeatedly: }`

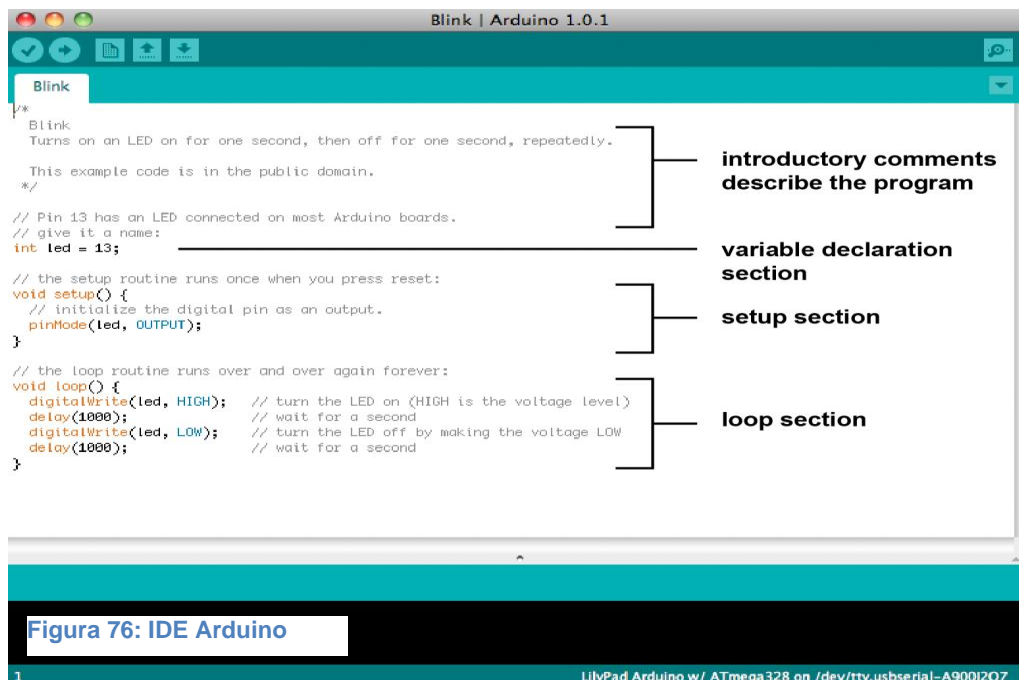


Figura 76: IDE Arduino

Desde la misma pantalla y a través de un menu se puedes compilar el programa y además subirlo (grabarlo) en el Intel Edison, que lo ejecutará cada vez que encienda o reinicie el Intel.

Así opera un Arduino de las diferentes familias, pero Edison puede ir más allá.

#### PASO 5

Conectarse al Edison vía Puerto Serie. Utilizaremos programas auxiliares para conexión por los puertos serie del ordenador. Uno muy simple bajo windows es el PuTTY programa gratuito, de pequeño tamaño pero que permite comunicarse al Intel a través del puerto Serie.

Una vez configurada la velocidad de caracteres y el nombre del puerto serie, se conecta en modo terminal.

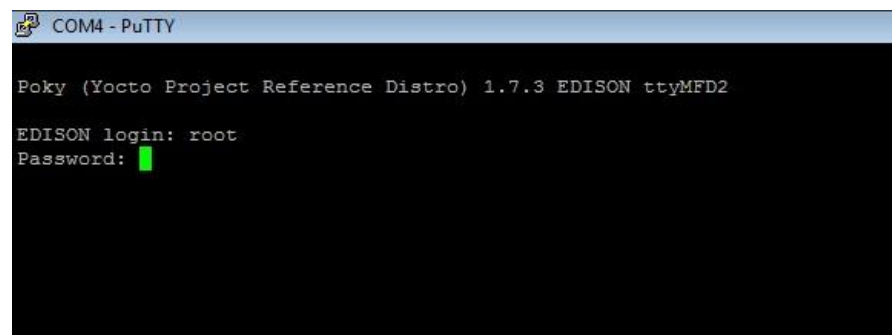


Figura 77: Terminal PuTTY

Solicita login (root por defecto) y un password que habrá que cambiar.

Luego se configura la red y el nombre mediante: `configure_edison`.

Ya está operativo EDISON como un ordenador con un S.O. Linux.

## PASO 6

Es muy recomendable para acelerar la carga y edición de programas el empleo de la herramienta gratuita WinSPC que permite que nos conectemos a EDISON como si fuera otra unidad de windows operando de forma rápida y pudiendo abrir varias sesiones. Puede accederse configurando el WIFI o a través de cable mediante protocolo SFTP.

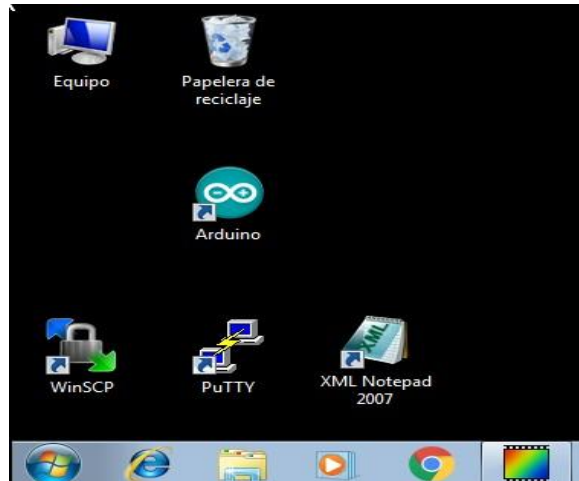


Figura 78: Programas acceso a Edison

## 5.1 Comunicación Sensores Deportista

Para establecer esta comunicación se necesitan las propias sondas, en este caso el Heart Rate Monitor y además configurar convenientemente el Intel EDISON.

La sonda es la seleccionada en el proceso de diseño Heart Rate Monitor BLE ZE749300.

Se ve en la fotografía junto a las sondas BLE de posición (Ibeacon).



Figura 79: Sondas BLE Personal Trainer

Para establecer una comunicación con el deportista dado que la sonda es BLE, debe utilizarse el servicio de BLE que Intel EDISON tiene y la antena y módulo BLE (Broadcom BCM 43340) que el propio chip trae incorporado.

Mediante un proyecto denominado BlueZ se puede comunicar fácilmente con la sonda:

En línea de comandos ejecutar las instrucciones siguientes:

```
systemctl stop bluetooth  
systemctl startbluetooth  
rfkilllist  
rfkillunblock bluetooth
```

Utiliza uno de losperfiles que trae este paquete de herramientas por defecto, el HRP, Heart RateProfile. Internamente está basado en Python que es otro de los lenguajes que soporta el EDISON, que va a decodificar automáticamente las notificaciones que envía la sonda HR.

La secuencia de comandos a utilizar es:

```
bluetoothctl  
[bluetooth] agent DisplayYesNo  
[bluetooth] default-agent  
[bluetooth] scan on  
Devuelve : Device E8:4C:CC:3A:30:4B  HRM_Sensor 3B2FA1C  
[bluetooth] trust E8:4C:CC:3A:30:4B  
[bluetooth] pair E8:4C:CC:3A:30:4B  
[bluetooth] connect E8:4C:CC:3A:30:4B  
[bluetooth] info E8:4C:CC:3A:30:4B
```

El programa informará de todos los identificadores de la sonda, batería,...

HEART RATE ATTRIBUTE

000180d-0000-1000-8000-00805f9b34fb

000180d-0000-1000-8000-00805f9b34fb

```
EDISON login: root
Password:
root@EDISON:~# cd node1
root@EDISON:~/node1# cd ..
root@EDISON:~# systemctl stop bluetooth
root@EDISON:~# systemctl start bluetooth
root@EDISON:~# rfkill unblock bluetooth
root@EDISON:~# bluetoothctl
[NEW] Controller 98:4F:EE:03:8A:8A EDISON [default]
[NEW] Device 20:91:48:C7:75:48 iTrack
[NEW] Device D8:90:E8:FD:7D:0C J (Galaxy S4)
[NEW] Device E8:4C:CC:3A:30:4B HRM_Sensor
[NEW] Device 10:02:B5:27:15:B3 ZARHP-PC
[bluetooth]# agent DisplayYesNo
Agent registered
[bluetooth]# default-agent
Default agent request successful

[bluetooth]# trust E8:4C:CC:3A:30:4B
Changing E8:4C:CC:3A:30:4B trust succeeded

[bluetooth]# info E8:4C:CC:3A:30:4B
Device E8:4C:CC:3A:30:4B
    Name: HRM_Sensor
    Alias: HRM_Sensor
    Appearance: 0x0341
    Paired: no
    Trusted: yes
    Blocked: no
    Connected: no
    LegacyPairing: no
    UUID: Generic Access Profile (00001800-0000-1000-8000-00805f9b34fb)
    UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
    UUID: Device Information (0000180a-0000-1000-8000-00805f9b34fb)
    UUID: Heart Rate (0000180d-0000-1000-8000-00805f9b34fb)
    UUID: Battery Service (0000180f-0000-1000-8000-00805f9b34fb)
[bluetooth]#
```

Figura 80: Heart Rate Setup

Se Dispone de un script para leer los valores de heartrate, se llama test-heartrate. desde el directorio /usr/lib/bluez/test > ./test-heartrate -b E8:4C:CC:3A:30:4B

Responde con:

Value 63 (que son las pulsaciones por minuto que envía la sonda)  
Contact 1  
Interval 724

Esto se repite con la cadencia de unos 5s donde la información se va actualizando. Por supuesto esta información se puede meter en un fichero, o intercambiar con otros procesos.



## 5.2 Comunicación Sensores Internos

2 Sensores de distancia más 1 sensor de temperatura.

Los sensores internos más importantes serán los que recogen los datos de ambas sondas de distancia, la sonda de corto alcance HC-SR04 y de medio alcance la **Sonda JSN-SR04T**.

Su diagrama es similar y también su modo de funcionamiento

### Ultrasonic timing chart:

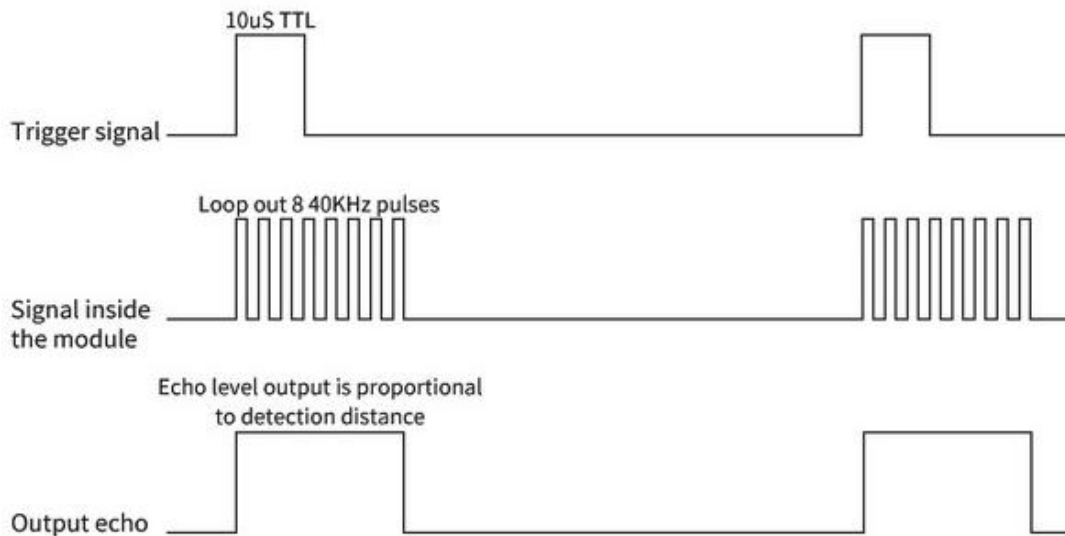


Figura 81: 10us Echo

De esta manera con una simple fórmula podemos determinar la distancia:

$$d = \frac{T \cdot V_{sonido}}{2}$$

Consideraremos una velocidad de sonido en el aire de 340 m/s.

Sus pines se llaman en ambos casos del mismo modo, marcamos los pines de conexión a Edison

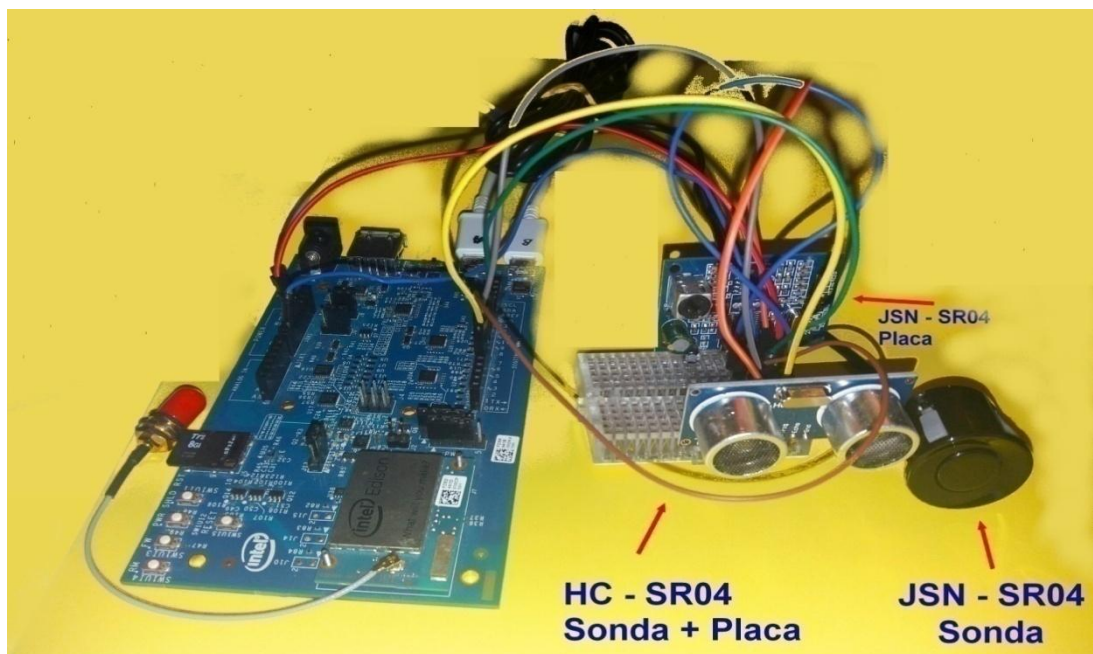
### HC - SR04

Vcc; +5v	-----	Vcc Edison 5V
Gnd; 0v	-----	GND EDISON 0V
Trigger; Rx	-----	D5
Echo; Tx	-----	D6

### JSN - SR04

Vcc; +5v	-----	Vcc Edison 5V
Gnd; 0v	-----	GND EDISON 0V
Trigger; Rx	-----	D7
Echo; Tx	-----	D8

A continuación se plasma una foto del montaje empleado con las 2 sondas antes de montarlo.



**Figura 82: Conexión Edison Sondas SR04**

Ahora la programación (ARDUINO IDE)

```
//SKETCH EDISON 2 SONDAS DISTANCIA MEDICION
```

```
//Jesús García Arranz, TFM
```

```
//Variables definición
```

```
const int EchoPin1 = 5;
```

```
const int TriggerPin1 = 6;
```

```
const int EchoPin2 = 7;
```

```
const int TriggerPin2 = 8;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(TriggerPin1, OUTPUT);
```

```
  pinMode(EchoPin1, INPUT);
```

```
  pinMode(TriggerPin2, OUTPUT);
```

```
  pinMode(EchoPin2, INPUT);
```

```
}
```

```
void loop() {
```

```
  int cm1 = ping(TriggerPin1, EchoPin1);
```

```
  Serial.print("Distancia Sonda HC: ");
```

```
  Serial.println(cm1);
```

```
  delay (100);
```

```
  int cm2 = ping(TriggerPin2, EchoPin2);
```

```
  Serial.print("Distancia Sonda JSN: ");
```

```
  Serial.println(cm2);
```

```
  delay(1000);
```

```
}
```

```
int ping(int TriggerPin, int EchoPin) {
```

```
  long duration, distanceCm;
```



```

digitalWrite(TriigerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
delayMicroseconds(4);
digitalWrite(TriigerPin, HIGH); //generamos Trigger (disparo) de 10us
delayMicroseconds(10);
digitalWrite(TriigerPin, LOW);
duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos, en microsegundos
distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
return distanceCm;
}

```

Finalmente ver en pantalla los resultados de las 2 sondas girando al mismo tiempo hacia paredes, pudiendo observar el Sketch transferido correctamente al EDISON.

The screenshot shows the Arduino IDE interface. The main window displays the 'distancia\_jga' sketch, which includes pin definitions, setup, loop, and a ping function. The serial monitor (COM7) shows the output of the program, displaying distance measurements for two sensors (JSN and HC) in centimeters. The output shows alternating readings for both sensors, with values ranging from 24 to 317 cm. The status bar at the bottom indicates 'Transfer complete'.

```

const int EchoPin1 = 5;
const int TriggerPin1 = 6;
const int EchoPin2 = 7;
const int TriggerPin2 = 8;

//const int LedPin = 13;

void setup() {
  Serial.begin(9600);
  //pinMode(LedPin, OUTPUT);
  pinMode(TriggerPin1, OUTPUT);
  pinMode(EchoPin1, INPUT);
  pinMode(TriggerPin2, OUTPUT);
  pinMode(EchoPin2, INPUT);
}

void loop() {
  int cm1 = ping(TriggerPin1, EchoPin1);
  Serial.print("Distancia Sonda HC: ");
  Serial.println(cm1);
  delay(100);
  int cm2 = ping(TriggerPin2, EchoPin2);
  Serial.print("Distancia Sonda JSN: ");
  Serial.println(cm2);
  delay(1000);
}

int ping(int TriggerPin, int EchoPin) {
  long duration, distanceCm;

  digitalWrite(TriggerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds(4);
  digitalWrite(TriggerPin, HIGH); //generamos Trigger (disparo) de 10us
  delayMicroseconds(10);
  digitalWrite(TriggerPin, LOW);

  duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos, en microsegundos

  distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
  return distanceCm;
}

```

Serial Monitor Output (COM7):

```

Distancia Sonda JSN: 408
Distancia Sonda HC: 26
Distancia Sonda JSN: 27
Distancia Sonda HC: 27
Distancia Sonda JSN: 25
Distancia Sonda HC: 25
Distancia Sonda JSN: 24
Distancia Sonda HC: 26
Distancia Sonda JSN: 24
Distancia Sonda HC: 25
Distancia Sonda JSN: 24
Distancia Sonda HC: 25
Distancia Sonda JSN: 24
Distancia Sonda HC: 25
Distancia Sonda JSN: 24
Distancia Sonda HC: 317
Distancia Sonda JSN: 317
Distancia Sonda HC: 318
Distancia Sonda JSN: 316
Distancia Sonda HC: 317
Distancia Sonda JSN: 316

```

Transfer complete

**Figura 83: Edison Programando sensor distancia**

Ambas sondas tienen resultados muy similares tanto a corta distancia 24, 25 cm como a larga distancia 317, 316cm y que la diferencia de 1cm, es fácil de prever porque se mantiene constante, por tanto es un offset fácil de eliminar restando o añadiendo +-1cm.

La aplicación en el programa principal se realizará convirtiendo parte de este programa en una función a la que el main (loop) llamará cuando convenga.

## Sensor de Temperatura ambiental

Como se explica en diseño se va a utilizar un LM36.

Una sonda muy simple y barata que usa una entrada analógica y mide la temperatura ambiente.

Alimentación, de 2.7V a 5.5V

Rango Temperaturas; 0V a -50°C y de 1.75V a 125°

Voltaje Operativo: 2.7V a 5.5V DC

Rango de Medición: -50°C a 125°C

Salida: Analógica

Fórmula de la temperatura en milivoltios es: Temperatura °C = 100\*(Lectura de Voltaje)-50

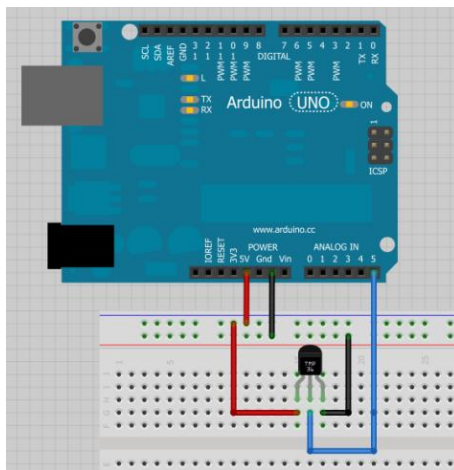
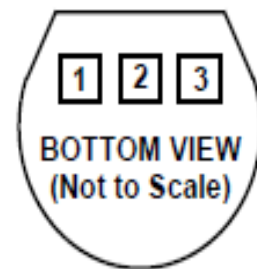


Figura 85: Sonda Temperatura Conexión



PIN 1, +V<sub>S</sub>; PIN 2, V<sub>OUT</sub>; PIN 3, GND

Figura 84: LM36 PINOUT

Su código será:

```
// Declaracion de variables globales
float T;
int pinLM36 = 0;

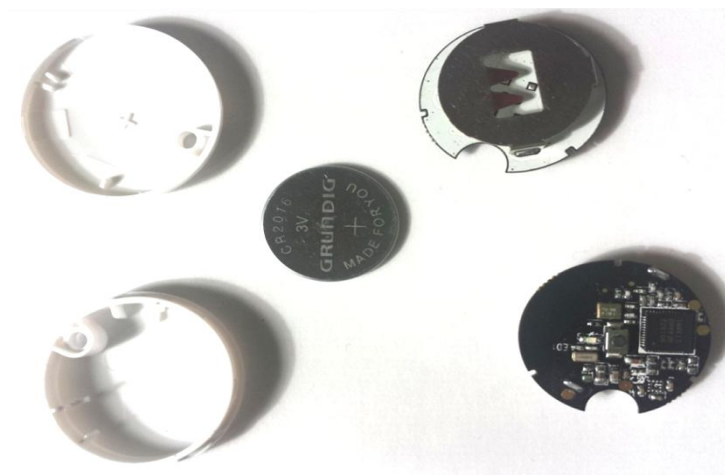
//Valor del sensor (0 a 1023)
//Pin de entrada del sensor (A0)

void setup() {
    Serial.begin(9600);
    // Configuramos el puerto serial a 9600 bps
}

void loop() {
    T = analogRead(pinLM36); // analogRead lee el sensor ( 0 a 1023 )
    T = (5.0 * T * 100.0)/1024.0; // Calculamos la temperatura con la fórmula
    Serial.print(T);           / // Envia el dato al puerto serial
    Serial.print("\n");        // Salto de línea
    delay(1000);               // Esperamos un tiempo para repetir el loop
}
```

### 5.3 Posicionamiento de la máquina

En parte de diseño quedó definido que se hallaría el resultado de posición como el punto de intersección de los ejes radicales formados por las 3 circunferencias imaginarias que tienen centro en cada sonda ibeacon y radio  $r_1, r_2, r_3$  determinado por el RSSI que capta EDISON.



**Figura 86: Posicionamiento por RSSI, sondas**

Tomamos los ejes radicales 1-2 y 1-3, la solución al problema, sería el punto de corte de ambos ejes, que será:  $P_{C1213} (E_{12x}, E_{13y})$

D; Distancia entre centros de las circunferencias (En este caso  $D=10$ )

$$E_{12x} = \frac{r_1^2 - r_2^2 + D^2}{2 \cdot D}; \quad E_{13y} = \frac{r_1^2 - r_3^2 + D^2}{2 \cdot D}; \quad D = 10$$

Pero ¿cómo determinrs esos radios  $r_1, r_2, r_3$ ?, pues transformando valores RSSI en valores de distancia y para ello previamente deben realizarse mediciones con el propio EDISON de las señales recibidas de las sondas ibeacon.

Tomar 18 mediciones de las sondas ibeacon en 3 puntos distintos de la pista pero siempre con la misma distancia hasta el ibeacon.

En el primer metro hay un rango más amplio de señal y por ello mejor definición, por lo que se toman muestras más finas, es decir mayor número, a distancias de 0, 0,25, 0,5 y 1,0m. A partir de ahí los intervalos serán de 1,0m.

#### PASO 1

Con los resultados obtenidos en la tabla (valores medios) realizaremos una conversión directa desde nivel de señal SRRI (dB) hasta distancia (m). Obteniendo así  $r_1, r_2, r_3$ .

#### PASO 2

Aplicando las fórmulas  $E_{12x} = \frac{r_1^2 - r_2^2 + D^2}{2 \cdot D}; \quad E_{13y} = \frac{r_1^2 - r_3^2 + D^2}{2 \cdot D}; \quad D = 10$

Escribir datos obtenidos por Intel edison al ejecutar la aplicación.....

Los datos obtenidos se recogen en una tabla. Se realizaron con un móvil con S.O. Android de la marca Samsung S4 con Bluetooth y la aplicación "IbeaconDetector".



Figura 87: RDL51822 Detección Sondas

Distancia ( m )	RSSI móvil ( dB )	RSSI 1 Edison (dB)	RSSI 2 Edison (dB)	RSSI 3 Edison (dB)	media edison (dB)	Desviación
0,00	34	36	37	35	36	1,0
0,25	50	51	48	49	49	1,5
0,50	55	53	52	53	53	0,6
1,00	60	58	57	57	57	0,6
2,00	67	63	64	64	64	0,6
3,00	73	69	68	68	68	0,6
4,00	75	72	73	72	72	0,6
5,00	78	76	77	75	76	1,0
6,00	82	80	79	78	79	1,0
7,00	84	81	82	83	82	1,0
8,00	86	85	84	86	85	1,0
9,00	89	87	87	88	87	0,6
10,00	91	88	88	89	88	0,6
11,00	93	89	89	89	89	0,0
12,00	95	91	90	91	91	0,6
13,00	96	93	91	92	92	1,0
14,00	97	95	94	94	94	0,6
14,50	98	96	96	95	96	0,6

Tabla 28: RSSI Experimental 3 Ibeacons

El código empleado (IDE ARDUINO)

//SKETCH EDISON 3 SONDAS POSICIÓN

//Jesús García Arranz, TFM

**//Variables definición**

int r1 = 0;

int r2 = 0;

int r3 = 0;

**void setup() {**

**}**

**void loop() {**

int r1 = traduce(rssi1);

delay (100);

int r2 = traduce(rssi2);

delay (100);

int r3 = traduce(rssi3);

delay(1000);

**}**

**int traduce (int rssi) {**

int radio;

switch(var){

case 34, 35, 36, 37,38,39,40, 41, 42, 43, 44, 45

radio= 0;

break;

case 46, 47, 48, 49

radio =0,25;

break;

case 50, 51, 52, 53, 54

radio =0,5;

break;

case 55, 56, 57, 58, 59, 60

radio =1;

break;

case 61, 62, 63, 64, 65

radio =2;

break;

case 66, 67, 68, 69

radio =3;

break;

case 70, 71, 72, 73

radio =4;

break;

case 74, 75, 76, 77

radio =5;

```

    break;
case 78, 79, 80
    radio =6;
    break;
case 81, 82, 83
    radio =7;
    break;
case 84, 85, 86
    radio =8;
    break;
case 87
    radio =9;
    break;
case 88
    radio =10;
    break;
case 89, 90
    radio =11;
    break;
case 91
    radio =12;
    break;
case 92, 93
    radio =13;
    break;
case 94, 95, 96
    radio =14;
    break;
default:
    //opcional si nada coincide se ejecuta esto
    break;
}
return radio;
}
*****

```

```

// Obtenidos los 3 radios r1, r2, r3 hallamos Posición (X, Y)
Posición X =  $(r_1^2 + 100 - r_2^2) / 20$ 
Posición Y =  $(r_1^2 + 100 - r_3^2) / 20$ 

```

```

//SKETCH EDISON POSICION X e Y
//Jesús García Arranz, TFM
//Variables definición

```

```

int r1 = 0;
int r2 = 0;
int r3 = 0;
void setup() {

```

```

}
void loop() {
    x = posicion(r1, r2);
    delay (100);
    y= posicion(r1, r3);
    delay(1000);
}

int posicion (int v1, v2) {
    int coordenada;
    coordenada =  $(v_1^2 + 100 - v_2^2) / 20$ ;

    return coordenada;
}

```

**Para ver el código en Bleno de detección de sonda, consultar el anexo IV**

## 5.4 Interfase de Usuario

Para la interfase de usuario necesitamos tener creado y estar activo un servidor web.

También necesitamos un código que intercambien mensajes con la página HTML. Además el código debe intercambiar mensajes con el ARDUINO IDE (HTML + JAVA + NODE.JS)

Ver ANEXO III + ANEXO V



Figura 88: Interfase Usuario, navegador multiplataforma

La interfase será accesible desde cualquier plataforma con navegador compatible con HTML, simplemente indicando la dirección del servidor y su puerto (8080).



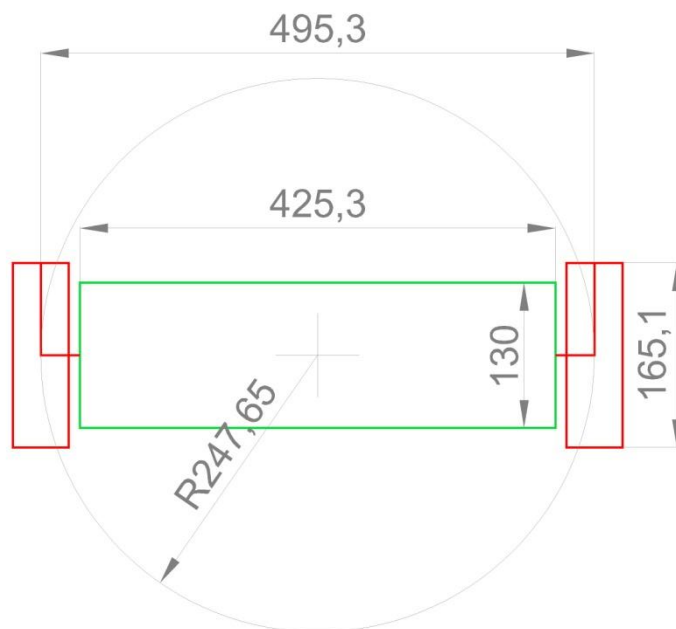
## 6.- Construcción física

Se va a construir un modelo prototipo del entrenador personal, basándonos en todas las características especificadas en la etapa de diseño siguiendo la secuencia lógica de la misma, empezando por la parte física desde su chasis, sus elementos motores, alimentación, drivers, lanzador, etapas de control, detectores.

Se ha elegido como material para el chasis un policarbonato compacto de 6mm, transparente.

El material se compra en planchas de diferentes medidas, espesores y tamaños.

Qué cantidad es necesaria para el chasis, pues para su ancho sobre los 130mm, y para su largo, vamos a hacer que el eje centro de rueda, centro de rueda tenga 49,53cm = 495,3mm.



**Figura 89: Dimensiones Chasis**

Este número no es aleatorio, sino basado en proporciones, se va a justificar su elección:

Si se hace Rotar, Girar sobre su eje de simetría al entrenador personal, un número entero de vueltas de rueda, una rueda girando en sentido horario y la otra antihorario vistas ambas desde el mismo lateral, vuelve a posición de partida, habrá un giro completo de  $360^\circ = 1$  vuelta.

Las ruedas a utilizar miden 6,5", eso es su diámetro, correspondiendo su desarrollo a:

$$n^\circ \text{vueltas} = \frac{\alpha}{360^\circ} \cdot \frac{L}{2 \cdot r} ; \quad 2 \cdot r = \emptyset \text{ ( Diámetro Rueda) } \text{ y } \alpha = \text{ángulo de giro deseado}$$

$$n = \frac{\alpha}{360^\circ} \cdot \frac{L}{\emptyset} \text{ si } \alpha = 360^\circ \text{ (1 vuelta) tendremos } n = L/\emptyset, \text{ por ello } L = \emptyset \cdot n$$

$$L = \emptyset \cdot n = 6,5" \cdot 3 = 19,5" \text{ Si lo expresamos en cm, sabiendo que } 1" = 2,54\text{cm}$$

$$L = 19,5" \cdot 2,54 = 49,53 \text{ cm (495,3 mm)}$$

De este modo 1 giro completo de entrenador personal equivale a 3 vueltas completas de rueda.

## 6.1 Implementación física: chasis

La medida del chasis será algo menor porque las ruedas tienen 5cm de ancho, con un punto de apoyo en el centro (2,5cm) y además el eje sobresale 1cm del propio chasis, y como tenemos 2 ruedas, por simetría:  $L_{chasis} = 495,3 - 25 - 10 - 25 - 10 = 425,3 \text{ mm}$ ,  $L_{chasis} = 425,3 \text{ mm}$

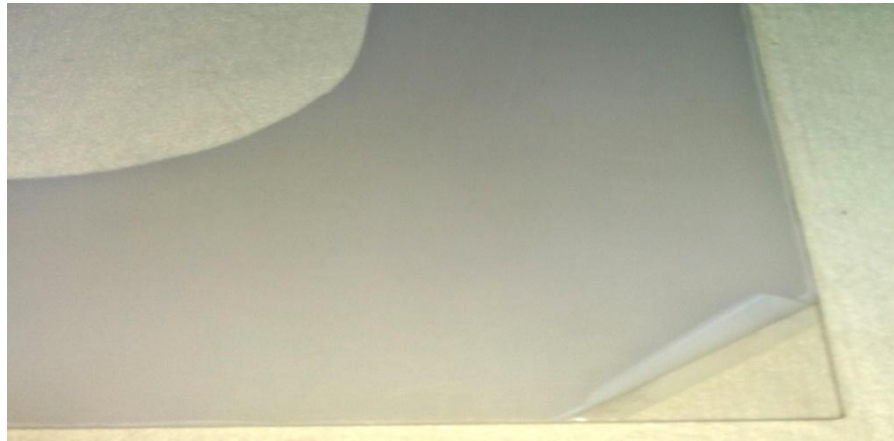


Figura 90: Policarbonato Compacto

1 Vuelta de rueda conseguiremos 120° de ángulo.

Sacar piezas de 425,3 x 120mm perforando para poder alojar las ruedas y sus fijaciones. Una vez que hemos sacado el chasis vamos a presentar los tornillos y fijaciones para ambos ejes de las ruedas. Estos ejes vienen con una parte plana y otra semicircular, aprovechando ambas para hacer una cama con buen asiento donde alojarlo de forma estable contra el chasis.



Figura 91: Montaje Fijaciones

Para sujetar las 4 ruedas se utilizan diferentes tornillos, tuercas, arandelas, se abren orificios pasantes para colocar los ejes de las ruedas motrices, y los espárragos de las ruedas locas, además del cableado de alimentación/control de los motores.

Colocar 4 tornillos con sus correspondientes tuercas y arandelas en cada una de las ruedas y mediante prisioneros de acero quedarán fijados correctamente.

Las ruedas son como se especificó en diseño de 6,5" es decir 16,51cm de diámetro. Podemos ver como del propio eje salen hasta 8 cables de colores, que corresponden a las 3 fases de energía, 3 sensores internos Hall y Vcc, Gnd.



**Figura 92: Montaje ruedas**

Vista de cómo queda ensamblado el nuevo conjunto chasis ruedas del entrenador personal.



**Figura 93: Chasis montado**

La batería principal se coloca en el centro y pegada al chasis para lograr que el centro de gravedad esté coincidente con el centro geométrico de la máquina y lo más bajo posible.

## 6.2 Motores de desplazamiento

El esquema de conexión entre el driver, la fuente de alimentación y el motor-rueda, está basado en el siguiente esquema de montaje

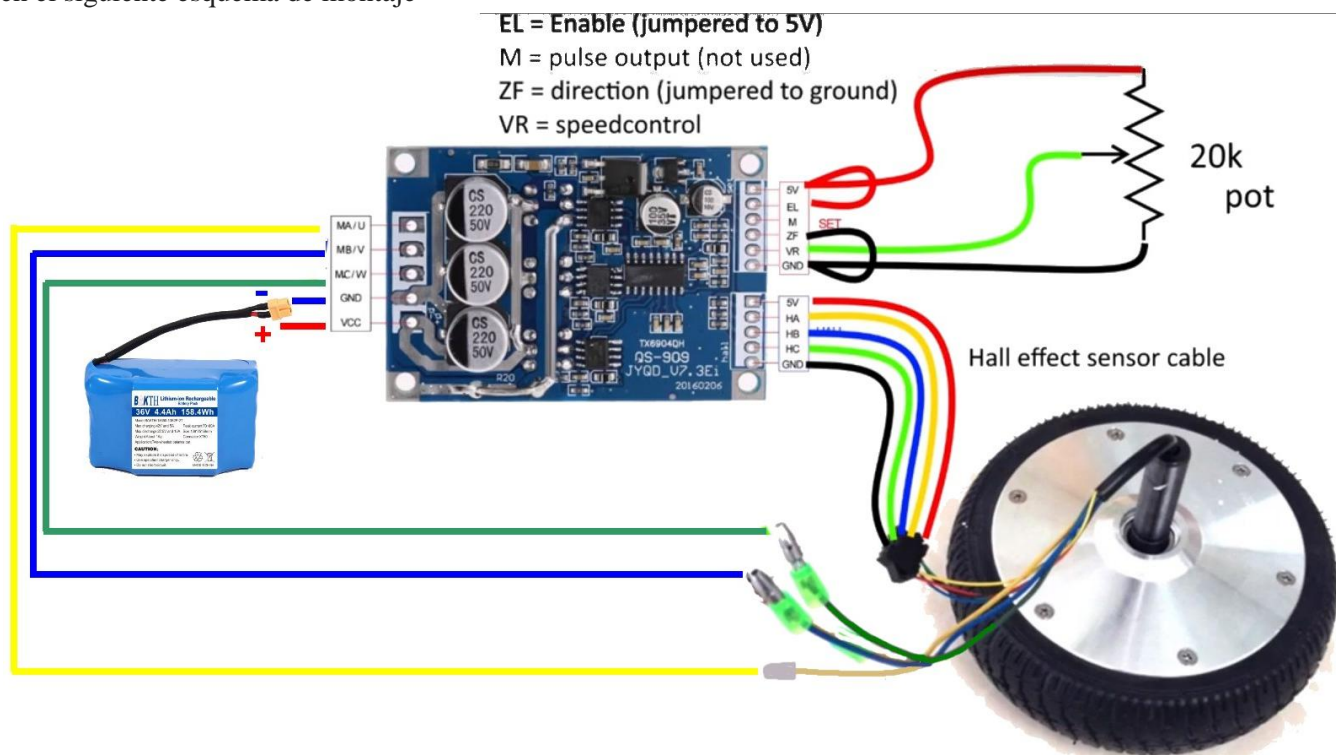


Figura 94: Esquema eléctrico Desplazamiento

Las ruedas con sus motores integrados se han probado antes de unirlos al chasis de forma individual gracias al potenciómetro para verificar que tanto la rueda como el driver funcionaban correctamente, y que respondían a la regulación manual mediante el movimiento del cursor.

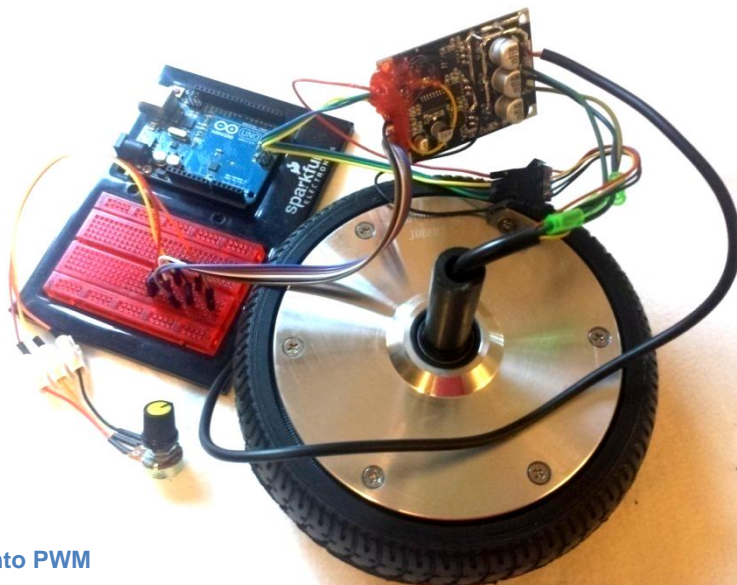
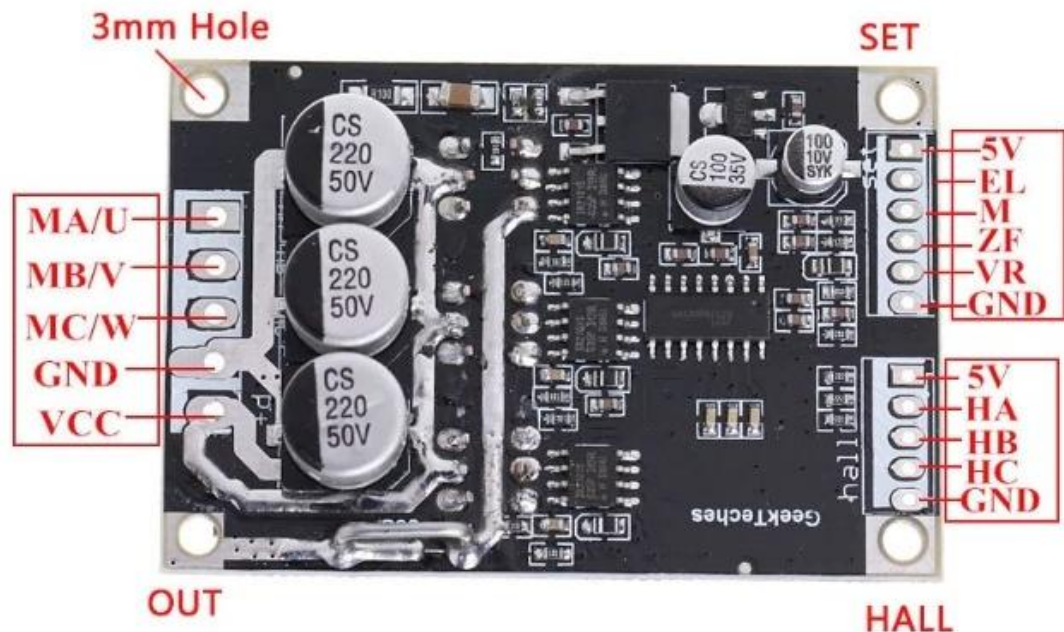


Figura 95: Check Desplazamiento PWM

Además se ha chequeado su respuesta a impulsos PWM generados por un arduino UNO.



Las conexiones se basan en este esquema y quedan como sigue:



**Figura 96: Desplazamiento Driver nomenclatura**

Los pines MA, MB, MC son salida de señales, se conectan a los cables del motor de la rueda del mismo nombre, se corresponden con las fases U, V, W del motor.

GND, VCC se conectan a la fuente principal de alimentación.

HA, HB, HC son señales de entrada se conectan y provienen de los sensores Hall del motor, con el mismo nombre y las salidas 5V y GND al mismo bus de la rueda para alimentar a los sensores Hall.

Los pines de control son:

EL, para habilitar, deshabilitar y se puede usar mediante un interruptor conectado a +5V.

M, es de salida e indica las vueltas, cada pulso suyo es una vuelta del motor. (entrada Edison)

ZF señal de entrada e indica el sentido de giro, se conecta al Intel Edison

VR señal de entrada, gobierna realmente la velocidad del motor, se conecta a EDISON PWM

Por lo tanto las conexiones al EDISON por cada Driver son 1 señales de salida Digital + 1señal salida PWM de EDISON y una señal de entrada al EDISON (3 señales (Pines) en total por cada rueda), por lo tanto 6 señales totales (2 Salidas Digitales + 2 Salidas PWM + 2 Entradas).

En cuanto al código, lo creamos en lenguaje ARDUINO IDE, en el siguiente sketch;

```
// DEFINICION DE VARIABLES
// ASIGNACION DE PINES      Edison Pines PWM = 3, 5, 6, 9
constint analogInPin = A0;    // Entrada analógica función del potenciómetro velocidad RPM
constint analogOutPin1 = 9;    // Salida PWM ataca al Driver1, rueda izquierda
constint analogOutPin2 = 6;    // Salida PWM ataca al Driver2, rueda derecha
intsensorValue = 0;           // almacena el valor de entrada leído
intoutputValue = 0;           // Variable de Salida PWM

voidsetup() {
    // Inicia Puerto Serie para datos al monitor (comprobar código)
    Serial.begin(9600);
}

voidloop() {
    // Lee Sensor, Potenciómetro
    sensorValue = analogRead(analogInPin);
    // Mapea el valor de entrada de 0 - 1023, a valores de 0 - 255
    outputValue = map(sensorValue, 0, 1023, 0, 255);
    // Escribe nuevo valor PWM a la salida
    analogWrite(analogOutPin1, outputValue);
    analogWrite(analogOutPin2, outputValue); // Ambas ruedas girarán a igual RPM

    // Escribe datos en el puerto serie (Monitor)
    Serial.print("sensor = ");
    Serial.print(sensorValue);
    Serial.print("\t output = ");
    Serial.println(outputValue);
    // Pausa de 2 ms
    delay(2);
}
```

Este código sirve para controlar estos motores BLDC de las ruedas, pero también para controlar los motores DC de los lanzabolas.

La explicación es que realmente las señales PWM no van a atacar a los motores directamente sino a sus drivers, controladores y traductores de señal, que transformarán estas señales PWM del uProcesador en las apropiadas a cada motor.

Necesita pues 4 señales de salida PWM, 1 para cada 1 de los drivers de estos 4 motores (2 ruedas BLDC) y 2 motores DC 795 para el lanzador.

No es un problema, en la fase de diseño se vio que el Intel EDISON tiene 4 salidas PWM.

### 6.3 Mecanismo de impulsión de bolas

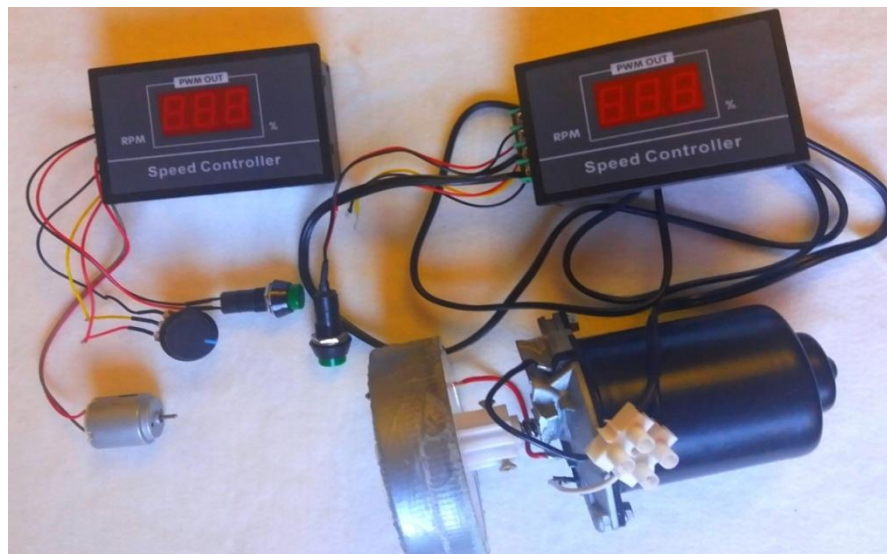
Este dispositivo como se definió en la fase de diseño, está formado mecánicamente por 2 discos separados una distancia algo inferior al diámetro de la bola, para que ésta pase a través de ellos, deformándose al hacerlo pero recuperando su forma original al salir despedida. No se pone en peligro la integridad de las bolas puesto que son diseñadas para deformarse por ejemplo al ser golpeadas por una pala (pádel) o raqueta (tenis), como ocurre con los balones de fútbol cuando son pateados, digamos que hay una deformación plástica reversible.

Cada uno de esos discos está unido por un eje a un motor que dará vueltas a una velocidad variable y regulable de entre 1 a 20.000 rpm.

Ambos pueden girar a velocidades diferentes para conseguir "efectos" de golpeo.

Para gobernarlos el uProcesador se ayuda de sendos drivers, en este caso dotados de display donde se podrá visualizar en tanto por ciento las RPM a las que gira el motor.

El primer paso ha sido comprobar que los drivers funcionaban, y para ello se conectaron a diferentes motores y de diferentes voltajes y se les reguló a través de potenciómetro.



**Figura 97: Lanzamiento Drivers**

Superadas estas pruebas preliminares de los drivers, se les liberó del potenciómetro manual y se conectaron a un microcontrolador Arduino UNO donde el propio controlador se debía encargar de subir y bajar las velocidades de ambos motores mediante el uso de PWM. En este caso el problema se complicó y no se ajustaba a lo esperado, con velocidades aleatorias y extremas de casi el 0% al 100% muy brusco sin valores intermedios.

Los ciclos de PWM se visualizaron a través de un osciloscopio HAMEG de doble Traza para comprobar el problema y ajustar parámetros como el dutycycle..... La solución adoptada fue simple pero costó encontrarla. Hubo que usar un FPB (Filtro Paso Bajo) se explica al final de este capítulo.

A continuación se implementa la parte física del lanzador, que está basado como se comentó en la fase de diseño en un aparato comercial del fabricante ONECONCEPT, modelo 10030841, una pequeña unidad capaz de propulsa bolas a distancias superiores a los 10 metros. Se desempaqueta el aparato que muestra el aspecto siguiente:



**Figura 98: Lanzador Bajo coste**

Ahora se procede a desarmar la máquina puesto que lo único que nos interesa es la parte mecánica del propio lanzador.

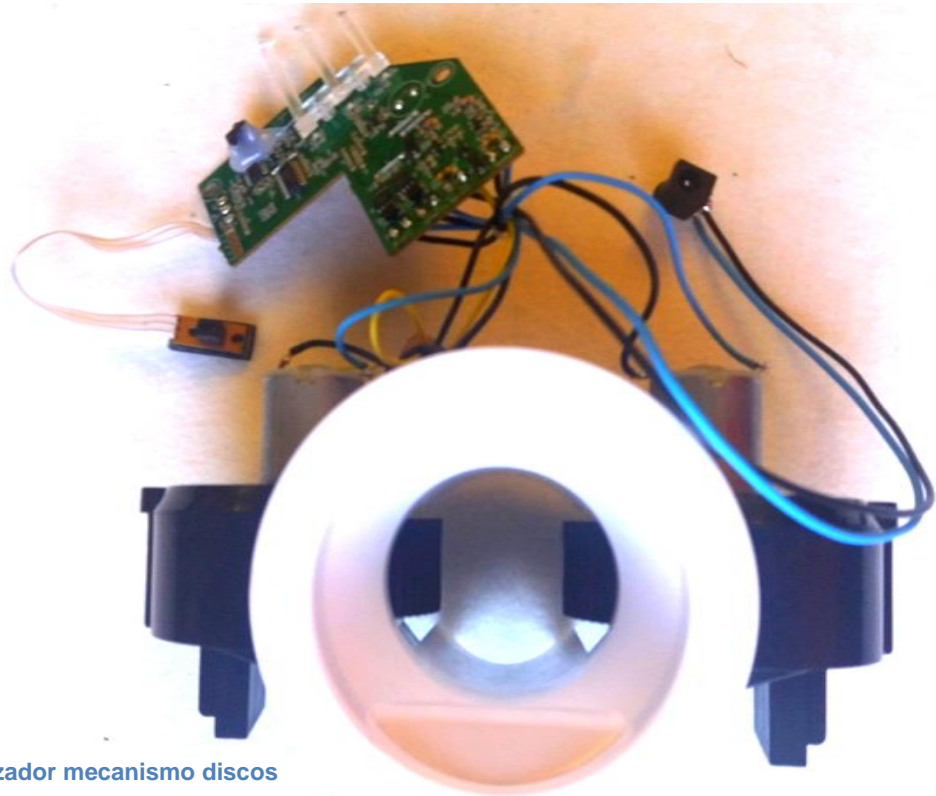
La imagen muestra todas las piezas que lo conforman, desglosadas y que no se van a usar para nuestra máquina entrenador personal Inteligente. En la parte inferior su base de apoyo, lateralmente las paredes naranjas que forman el cuerpo, el mecanismo interno de caída de las bolas así como la tapa del portapilas, pulsadores plásticos y tornillería, además de elementos accesorios de pequeñas dimensiones.



**Figura 99: Lanzador despiece**



En la foto inferior se puede observar el mecanismo electrónico además del mecanismo físico de lanzamiento en sí. No nos interesa el dispositivo electrónico propio al aparato pues es muy limitado por las propias características técnicas de la máquina.



**Figura 100: Lanzador mecanismo discos**

El mecanismo original funciona sólo con voltaje de hasta 9v lo que limita la velocidad de giro y por lo tanto la de la propia bola. Por ello se sustituirán los motores por los que se indicaron en la etapa de diseño, el RS795 de 24v u hasta 20.000 RPM.

Además se desecha toda la placa electrónica, basada en la detección de la bola por un pequeño sensor que en ese momento envía una señal a su chip, al que éste responde poniendo en marcha ambos motores a la misma velocidad de giro, lo que no permite realizar efectos ni ajustes de calibración individual ni conjunta.

Además mediante un pulsador sólo es posible realizar las operaciones de;

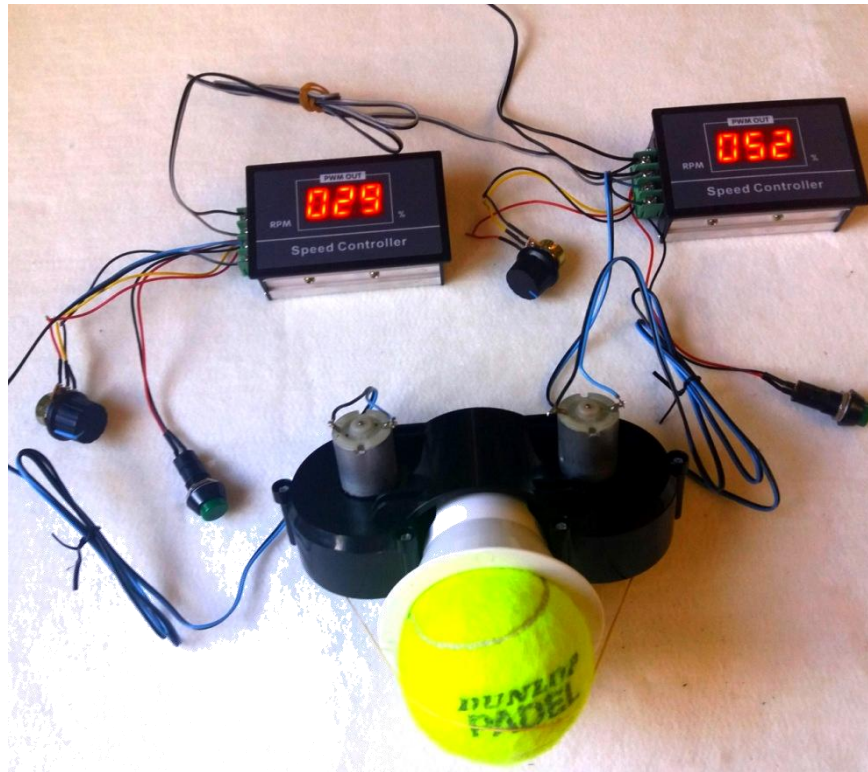
- 1, puesta en marcha
- 2 velocidad lenta
- 3 velocidad media
- 4 velocidad alta
- 5 parada.

Se repite de forma cíclica y tampoco nos interesa, por ello se desecha por completo.

Enlazando nuestros controladores ya probados como se hizo en el paso primero, con los nuevos motores colocados en el dispositivo de lanzamiento, se unen mediante cables nuevos a los controladores de velocidad de motor de corriente continua,

Antes de montar definitivamente en el lanzador, probamos que todo el sistema enciende y funciona correctamente, y que ahora sí es posible variar a voluntad la velocidad de giro de cada motor de forma independiente.

La ventaja de este modelo es que está construido en material muy ligero pero a la vez muy resistente y que podremos alzarlo a una altura relativamente distante de la base sin que desestabilicemos el sistema en su conjunto.



**Figura 101: Check Motores Lanzamiento**

En cuanto al código que ataca los motores DC de los lanzabolas es idéntico al escrito para controlar motores BLDC de las ruedas.

La explicación es que realmente las señales PWM no van a atacar a los motores directamente sino a sus drivers, controladores y traductores de señal, que transformarán estas señales PWM del uProcesador en las apropiadas a cada motor.

Necesitamos pues 4 señales de salida PWM, 1 para cada 1 de los drivers de estos 4 motores (2 ruedas BLDC) y 2 motores DC 795 para el lanzador.

En cuanto al código, lo creamos en lenguaje ARDUINO IDE, en el siguiente sketch;

```
// DEFINICION DE VARIABLES
// ASIGNACION DE PINES
const int analogInPin = A0;    // Entrada analógica función del potenciómetro varía velocidad
                                RPM
const int analogOutPin3 = 3;    // Salida PWM ataca al Driver3 Lanzabolas Motor Izquierdo
const int analogOutPin4 = 5;    // Salida PWM ataca al Driver4 Lanzabolas Motor Derecho
int sensorValue = 0;           // almacena el valor de entrada leído
int outputValue = 0;           // Variable de Salida PWM

void setup() {
```

```

        Serial.begin(9600);    // Inicializo Puerto Serie para datos al monitor (comprobar
código)
    }

    voidloop() {
        // Leemos Sensor, Potenciómetro
        sensorValue = analogRead(analogInPin);
        // Mapeo entrada de 0 - 1023, a valores de 0 -
255
        outputValue = map(sensorValue, 0, 1023, 0, 255);

        analogWrite(analogOutPin3, outputValue);    // Escribo nuevo valor PWM a la salida
        analogWrite(analogOutPin4, outputValue);    // en este ejemplo ambas ruedas girarán a
igual RPM
        Serial.print("sensor = ");                  // Escribo datos en el puerto serie
(Monitor)
        Serial.print(sensorValue);
        Serial.print("\t output = ");
        Serial.println(outputValue);
        delay(2);                                    // Pausa de 2 ms
    }

```

## FILTRO PASO BAJO

La señal PWM que emite el EDISON no gobernaba correctamente el driver. Después de muchas pruebas y gracias al uso del osciloscopio se llegó a la conclusión de que la señal PWM generada no servía para alimentar al control del driver y que éste necesitaba una señal de tensión continua (no una PWM). Este extremo se verificó usando fuentes continuas (diferentes baterías y pilas) y se corroboró. ¿Cómo convertir la señal PWM a señal VCC equivalente?

La forma más sencilla es usar un Filtro Baso bajo de primer orden, de esta forma tenemos señal correcta.

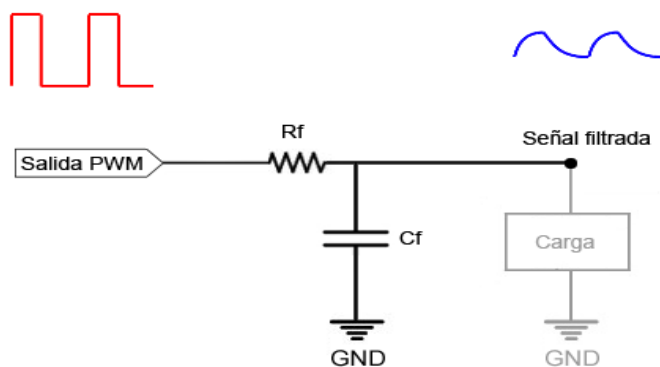


Figura 102: Filtro Paso Bajo

Los valores calculados  $R_f = 10K$  y  $C_f = 1\mu F$ .

Se debe a que la frecuencia está en torno a 500Hz,  $T = 2 \cdot 10^{-3}s$ , con  $V = +5v$  y Duty Cycle=50%.  $RC = 0,01s$  y un tiempo de respuesta de 0,023s.

## 6.4 Fuente de alimentación

Tal y como se especificó en diseño, se utilizarán 2 fuentes de alimentación por separado. Hay que tener la precaución de que las GND (tensión 0) de ambas fuentes estén unidas para que esa sea la tensión de referencia 0 de todo el conjunto (trabajaremos en lógica positiva).



**Figura 103: Fuentes alimentación**

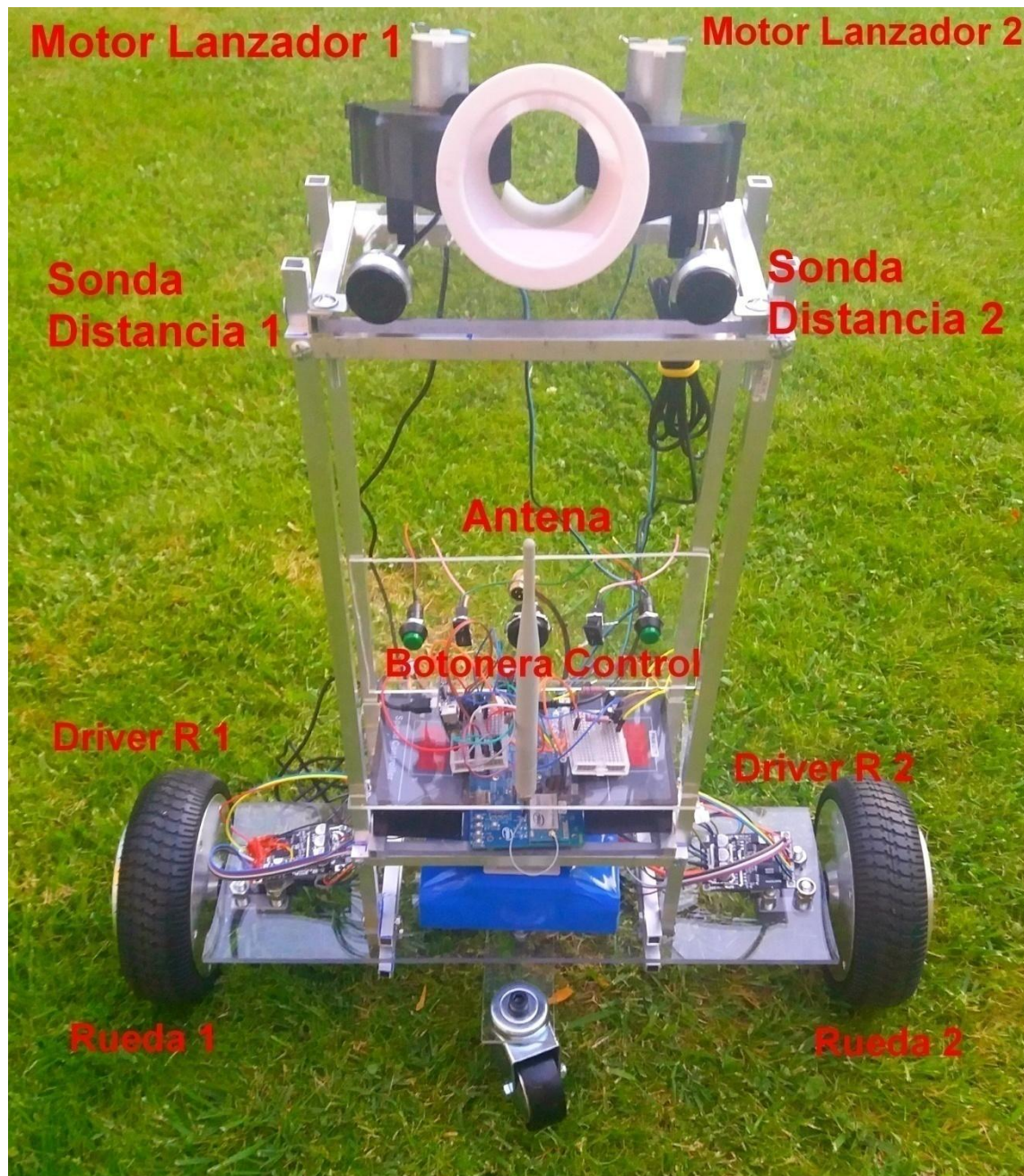
Se van a instalar sobre la base más baja del sistema de forma que el peso esté centrado en el entrenador y a la menor altura posible para dejar estable el sistema.

Todo el sistema será recargado por una fuente externa al sistema capaz de suministrar 42v y 1,5A de corriente. Se prevé una carga de en torno a 2 horas y una autonomía del sistema de más de 4 horas reales.



## 6.5 Red Sensorial

Tal y como se especificó en diseño, los sensores se comunicarán por BLE a través de la antena integrada en el propio EDISON



**Figura 104: Personal Trainer terminado**

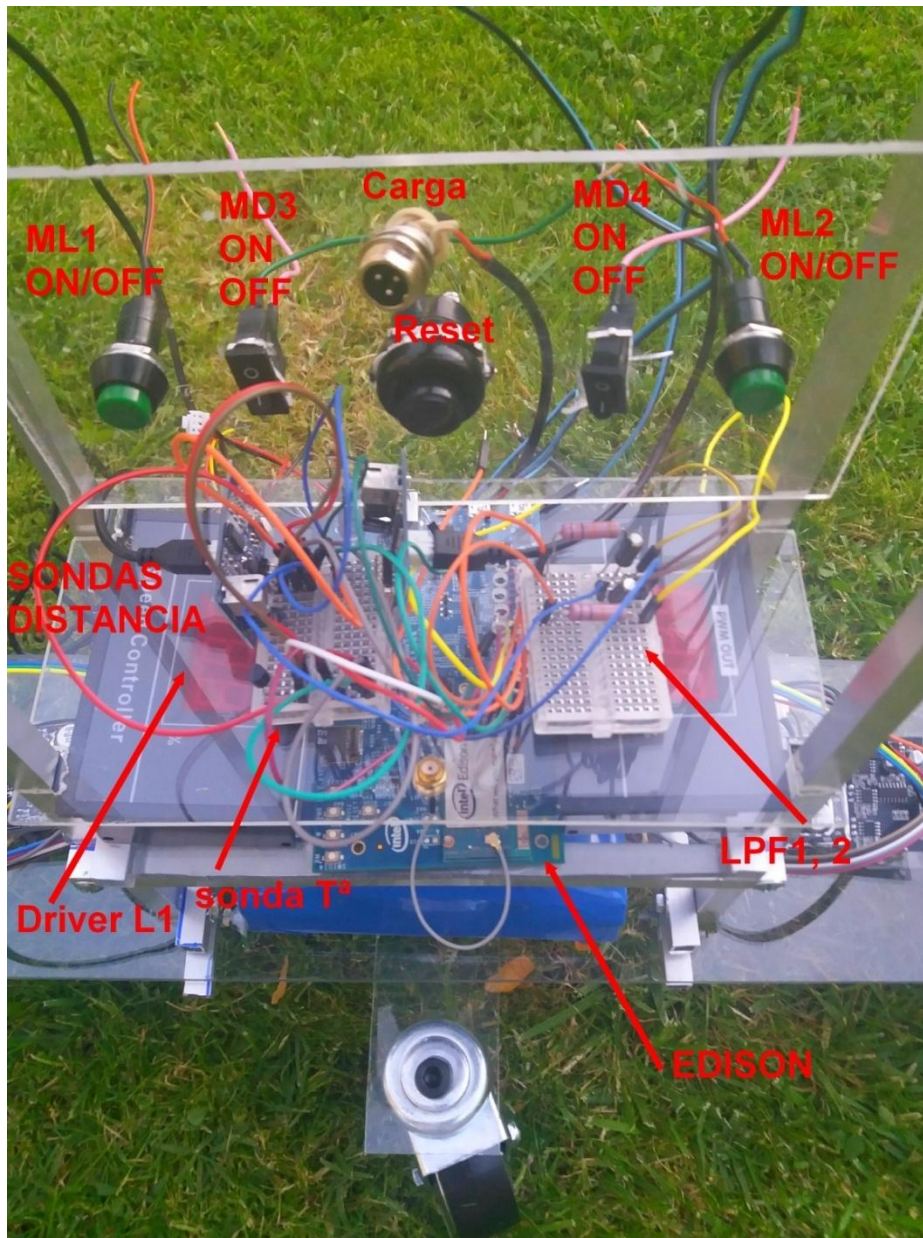
Las sondas de posición se colocarán en la parte delantera del entrenador, en ambos extremos y paralelas entre sí. Antes de su funcionamiento real, se realizarán pruebas de calibración como se explica en el apartado correspondiente de diseño.

Las señales de los motores también pueden sensarse a través del Intel EDISON.



## 6.5 Red Actuadores

Los motores estarán situados en la parte más baja en el caso de las ruedas, y en la parte más elevada en el caso del lanzador.



**Figura 105: Detalle Controlador, Panel Control, conexiones**

Sus drivers estarán en lugares intermedios, cercanos al uProcesador para evitar problemas de pérdidas de señal o errores indeseados.

## 6.7 Instalación del controlador

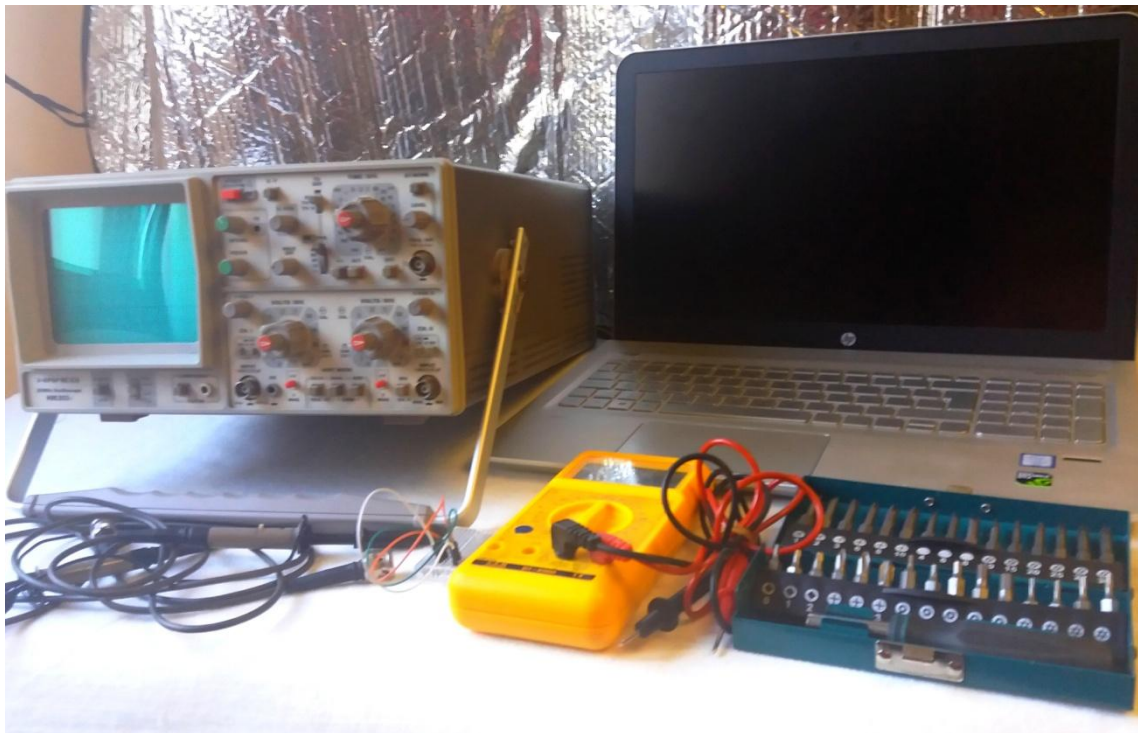
El Intel EDISON será ubicado a una altura intermedia, por encima de las baterías y por debajo de la unidad lanzadora. Estará alimentado directamente por una batería auxiliar a la principal para evitar interferencias y problemas derivados de la puesta en marcha y parada de los motores.

## 6.8 Herramientas auxiliares

Para la parte electrónica ha sido necesario el empleo de ordenadores para poder programar el Intel Edison y realizar pruebas de las diferentes sondas.

También se ha testado la interfase de usuario utilizando tablets y smartphones.

Las partes eléctricas se han testado con multímetros (polímetros digitales) y para el análisis de las señales se empleó un osciloscopio HAMEG.



**Figura 106: Herramientas auxiliares**

Además se han usado herramientas manuales y eléctricas para construir las diferentes partes del Entrenador, tales como sierras de vaivén, destornilladores, taladros eléctricos, soldadores, pistolas de pegado, desoldadores y tornillería variada, .....

## 6.9 Valoración Económica del trabajo

Se ha realizado un doble estudio con el fin de obtener los datos necesarios para poder realizar de forma objetiva un análisis sobre la viabilidad del producto.

El primer análisis recoge datos sobre el prototipo, lo acontecido en el propio TFM, sus costes en materiales y sus costes en desarrollo y montaje. Es por lo tanto una valoración tanto a nivel de coste físico, como de desarrollo de este TFM.

El segundo estudia la viabilidad de una producción real de estos equipos, de las lecciones aprendidas (se verá más adelante) se extraen elementos que pueden obtenerse con costes menores, ya no se tiene en cuenta el coste de desarrollo, sino el de fabricación, se adjunta el importe de cada una de las partidas de materiales,

A.- Análisis de costes del prototipo elaborado para este TFM.

Material	ud	Precio/ud (€)	Coste (€)
Procesador Inte Edison	1	57	57
Board Kit Arduino Intel	1	63	63
Máquina Lanzador	1	36	36
Motores 24v cc	2	9	18
Rueda Hub Motor	2	43	86
Controlador Ruedas Brushless	2	9	18
Controlador Motor cc	2	8	16
Sonda Ibeacon	3	5	15
Pulsómetro banda com H7 polar	1	17	17
Sonda detector JSN SR-04	2	6	12
Sonda LM 36 temperatura	1	2	2
Polycarbonato 500 x 500 x 6mm	1	45	45
Perfiles aluminio 10x 10 x1000	4	2	8
Tornillería variada	1	3	3
Conectores, cableado, Kit	1	18	18
Batería 36v 4.000mAh	1	36	36
batería Respaldo procesador	1	12	12
Cargador Baterías	1	24	24
			0
<b>SubTotal:</b>			<b>486</b>
descripción	hora	Precio/h (€)	Coste (€)
Desarrollo Proyecto, producto	480	25	12000
Construcción Prototipo	32	25	800
<b>Total:</b>			<b>13.286</b>

Tabla 29: Valoración Económica Prototipo



De la tabla de datos se desprende que para el prototipo hay un fuerte componente económico de costes de desarrollo, 12.000€ frente a los 486€ de materiales, y que el montaje (con muchas operaciones de desmontaje, modificación de piezas, nuevo montaje, pruebas..) asciende a 800€. No hay un elemento que sea demasiado gravoso para el proyecto, la parte más cara es el ordenador, es decir el propio Intel Edison añadiendo su placa base, ambos en total 120€, que No supone una barrera económica para su ejecución.

#### B.- Estudio de viabilidad para la producción:

En primer lugar para este apartado se han tenido en cuenta las lecciones aprendidas, hay componentes que pueden comprarse como un Kit en lugar de manera individual y supone menor coste por tanto reducción del precio del producto .

Material	ud	Precio/ud (€)	Coste (€)
Procesador Inte Edison	1	57	57
Board Kit Arduino Intel	1	63	63
<b>Placa lanzador + 2 discos</b>	<b>1</b>	<b>16</b>	<b>16</b>
Motores 24v cc	2	9	18
Rueda Hub Motor	2	<b>hoverboard</b>	<b>60</b>
Batería 36v 4.000mAh	1		
Cargador Baterías	1	<b>hoverboard</b>	
Controlador Ruedas Brushless	2	9	18
Controlador Motor cc	2	8	16
Sonda Ibeacon	3	5	15
Pulsómetro banda com H7 polar	1	17	17
Sonda detector JSN SR-04	2	6	12
Sonda LM 36 temperatura	1	2	2
Polycarbonato 500 x 500 x 6mm	1	45	45
Perfiles aluminio 10x 10 x1000	4	2	8
Tornillería variada	1	3	3
Conectores, cableado, Kit	1	18	18
Batería Respaldo procesador	1	12	12
			0
<b>SubTotal:</b>			<b>380</b>
descripción	hora	Precio/h (€)	Coste (€)
Ensamblado componentes	7	25	175
Programación Pruebas	1	25	25
<b>Total:</b>			<b>580</b>

**Tabla 30: Valoración económica Producción Real**

Se ve como se reduce el coste de material hasta los 380€/unidad

En segundo lugar ya no hay costes de desarrollo y se reduce notablemente el tiempo de montaje.

Por último se incorpora un pequeño estudio de mercado para analizar la posibilidad real de venta de equipos y así la viabilidad del proyecto. El estudio de mercado se ha realizado para el entorno del pádel en España y Argentina, los 2 países donde está más implantado este deporte, y además se incorpora al tenis en España y en Francia porque "El entrenador personal" es absolutamente válido y compatible con la preparación física, técnica y táctica de ese deporte.

Mercado	nº	% compra	nº unidad	total unidades	coste/ud	Venta/ud	Subtotal	Beneficio
Jugadores Federados	68.873	5	1	344	580	1.000	344.365	144.633
Practicantes amateurs	2.000.000	3	1	6.000	580	850	5.100.000	1.620.000
<b>Clubes pádel</b>	<b>1.242</b>	<b>25</b>	<b>4</b>	<b>124</b>	<b>580</b>	<b>1.200</b>	<b>149.040</b>	<b>77.004</b>
<b>España Pádel</b>							<b>5.593.405</b>	<b>1.841.637</b>
Jugadores Federados	25.000	2	1	50	680	1.000	50.000	16.000
Practicantes amateurs	2.500.000	2	1	5.000	680	850	4.250.000	850.000
<b>Clubes pádel</b>	<b>600</b>	<b>10</b>	<b>4</b>	<b>24</b>	<b>680</b>	<b>1.200</b>	<b>28.800</b>	<b>12.480</b>
<b>Argentina Pádel</b>							<b>4.328.800</b>	<b>878.480</b>
Jugadores Federados	80.000	5	1	400	580	1.000	400.000	168.000
Practicantes amateurs	1.000.000	3	1	3.000	580	850	2.550.000	810.000
<b>Clubes tenis</b>	<b>500</b>	<b>25</b>	<b>4</b>	<b>50</b>	<b>580</b>	<b>1.200</b>	<b>60.000</b>	<b>31.000</b>
<b>España tenis</b>							<b>3.010.000</b>	<b>1.009.000</b>
Jugadores Federados	1.100.000	2	1	2.200	580	1.000	2.200.000	924.000
Practicantes amateurs	2.500.000	2	1	5.000	580	850	4.250.000	1.350.000
<b>Clubes tenis</b>	<b>400</b>	<b>10</b>	<b>4</b>	<b>16</b>	<b>580</b>	<b>1.200</b>	<b>19.200</b>	<b>9.920</b>
<b>Francia Tenis</b>							<b>6.469.200</b>	<b>2.283.920</b>
<b>Total:</b>	<b>5 años de expansión</b>			<b>22.209</b>			<b>38.802.810</b>	<b>12.026.075</b>

**Tabla 31: Estudio y análisis de Mercado**

La tabla refleja el número de jugadores federados en pádel y tenis en los diferentes países, así como el número de clubs, además de la estimación de practicantes habituales de cada uno.

Las unidades en Argentina llevan un sobrecoste de 100€ debido al transporte marítimo o aéreo.

Por cada posible comprador se ha estimado un porcentaje de adquisición probable, siendo el menor para los amateurs (2-3%) y mayor para los clubes pues su coste es menor al de un entrenador real, se presume que un 25% de los clubes pueden optar por la compra de equipos.

Cuántos equipos comprará cada potencial cliente, pues 1ud un aficionado o profesional para uso particular, y hasta 4ud un club pues de media tienen mayor número de pistas de prácticas.

Los datos se estiman como ventas en al menos 5 años, son totales en esos 5 años, no anuales.

22.000 equipos en 5 años significa producir 200ud al día por lo que si se necesitan 8 horas (1 turno completo) para 1ud, al menos 200 operarios. El Know How, el aprendizaje y la práctica en el montaje llevarán a una reducción real del 50% de tiempo, por ello 100 operarios lo cubrirán.

El coste de 100 operarios es de 1.600.000€ /año, en 5 años son 8.000.000€ sigue habiendo un beneficio bruto de en torno a los 4.000.000€ para ese periodo, es decir 800.000€/año.

De ello habrá que reducir los gastos fijos y otros puesto de trabajo.

Por todo ellos un informe de rentabilidad real corresponderá a una empresa especializada, pero independientemente de su "atractivo como inversión", el proyecto es viable.

## 7. Conclusiones

A.- Lecciones y conclusiones principales del TFM.

- 1.- El proyecto es viable
- 2.- El diseño se puede mejorar, hay que tener en cuenta que se ha construido un primer prototipo.
- 3.- Los elementos de control se pueden modificar y ampliar para dotarlo de mayor inteligencia.
- 4.- Los ibeacons No tienen el alcance que indican los fabricantes (teórico). Los valores reales quedan siempre por debajo, aunque sí cumplen con el rango de 14m máximo del proyecto.
- 5.- El RSSI recibido no es lineal con la distancia. Por ejemplo en el primer metro el rango de nivel de señal recibida varía de 35dB hasta los 50dB, es decir 15dB en un metro, buena precisión, pero a medida que nos alejamos, apenas tenemos 1dB de margen entre los 12 y los 13m de distancia.
- 6.- Las sondas aún siendo todas del mismo fabricante y habiéndose calibrado con su programa (una apk), ofrecen valores diferentes de RSSI situadas en los mismos puntos respecto a Edison. Por lo tanto se debe personalizar las medias de referencia tomadas para cada una y tratar de forma independiente de las otras para mejorar la precisión en el posicionamiento. Las sondas de distancia ofrecen una precisión de 10mm, por lo tanto mucho mejor que lo que obtenemos vía Ibeacon (1m de precisión). Si bien en este proyecto 1m nos puede valer, ya que tenemos las sondas de distancia para detectar obstáculos, se pueden incorporar para auxiliar el posicionamiento.
- 7.- No es necesario el uso de varias sondas de detección de distancia. 1 sola colocada sobre un servo puede girar simulando un radar militar y escanear todo lo situado alrededor. En este TFM se han usado 4 puertos PWM para los motores, por lo tanto un servo que ocuparía un 5 PWM, no tiene lugar pues Edison maneja sólo 4 señales PWM.
- 8.- Es más económico adquirir unidades comerciales tipo Hoverboard que comprar por piezas.
- 9.- La autonomía del equipo es algo inferior a la esperada, sobre 1,5h. Los consumos instantáneos en los movimientos pueden explicar el diferencial.

B.- Logros:

- 1.- Se ha logrado diseñar el prototipo de forma teórica
- 2.- Se ha conseguido construir el prototipo físicamente
- 3.- Se ha conseguido implementar el Servidor Web sobre el Edison
- 4.- Existe comunicación e integración entre todos los interfases y programas en ejecución.
- 5.- Las sondas de distancia, de temperatura y de posición están operativas y funcionando.
- 6.- Los elementos auxiliares de carga funcionan
- 7.- El panel de control realiza sus funciones de manera correcta.
- 8.- La comunicación BLE y la WiFi funcionan e interoperan simultáneamente.

#### C.- Análisis de seguimiento, planificación y metodología.

- 1.- Durante la realización de las diferentes fases se ha seguido el método de investigación seleccionado, así como los de ejecución planificados. Ha habido en algunos casos que hacer correcciones en un proceso continuo de realimentación mientras se ha desarrollado este TFM.
- 2.- Cuando ha sido necesario se ha añadido nuevas aportaciones de otros TFM y tesis doctorales, también trabajos de fin de grado que se han considerado pertinentes y que podían cubrir parcialmente necesidades de desarrollo del presente trabajo.
- 3.- A nivel de posicionamiento se ha terniso que realizar una labor crítica con la fuente de datos, y aplicarla de diferente manera a este proyecto.
- 4.- Además se han introducido nuevos esquemas de programación.

#### D.- Lineas pendientes:

- 1.- Adaptar el presente lanzador, la parte física extraída de unidad comercial, para que se puedan introducir bolas de pádel o tenis.
- 2.- Se puede incorporar un alimentador automático de bolas. Esto se puede resolver medinate un pick up (acumulador/recogedor de bolas) de hasta 17ud.
- 3.- Se puede diseñar una rutina de autorecogida de bolas aprovechando las 2 sondas de distancia
- 4.- Se puede conectar fácilmente o implementar IOT.
- 5.- Se podría dotar de conexión 5G compatible en NGN con WiFi actual.
- 6.- Es sencillo añadir una cámara al Edison y poder grabar en tiempo real el entrenamiento.
- 7.- Añador un interfase con rutinas personalizables para cada usuario.
- 8.- Añadir y desarrollar Inteligencia Artifical al personal Trainer
- 9.- Línea de autocalibrado de posición respecto al sistema de coordenadas (ibeacon) gracias a las sondas de distancia.
- 10.- Se puede añadir que elEdison actué como servidor WiF (Router) en lugar e ser un cliente.
- 11.- Realizar un sistema de protección frente a impactos de la propia unidad.

## 8. Glosario

### A

A2DP	Advanced Audio Distribution Profile (BLE)	Perfil BLE Audio
ADIF	Administrador de Infraestructuras Ferroviarias	Empresa
AES	Advanced Encryption Standard	Esquema cifrado por Bloques
AESA	Asociación Española de Seguridad Aérea	Asociación
AoA	Ángulo de Llegada	Método determinar dirección
ASFA	Anuncio de Señales y Frenado Automático	Sistema de Alarma automático
AVRCP	Audio/Video Remote Control Profile (BLE)	Perfil BLE Video

### B

BLDC	Brushless Direct Current	Motor
BLE	Bluetooth Low Energy	Tecnología de red de Área Personal

### C

C++	Lenguaje C incrementado	Lengua Programación
COSPAS	Cosmicheskaya Sistema Poiska Avariynih Sudov	Programa Internacional Búsqueda
CTP	Cordless Telephony Profile (BLE)	Perfil BLE telefonía inalámbrica

### D

DC	Direct Current	Motor
----	----------------	-------

### E

ECG	Electrocardiograma	Abreviatura de electrocardiograma
EEPROM	Electrically Erasable Programmable Read-Only Memory	Memoria de almacenamiento, tipo
EID	Eddystone Ephemeral Identifier	Perfil Eddystone
eMMC	Embedded Multimedia Card	Memoria de almacenamiento, tipo

### F

FTP	File Transfer Protocol	Protocolo
-----	------------------------	-----------

### G

GAP	Generic Access Profile (BLE)	Perfil BLE
GATT	Generic Attribute Profile (BLE)	Perfil BLE
GCC	GNU Compiler Collection	Grupo de compiladores de lenguaje C
GFSK	Gaussian Frequency Shift Keying	Clase de Modulación
GND	Ground	Abreviatura
GOEP	Generic Object Exchange Profile (BLE)	Perfil BLE
GPS	Global Position System	Systema de Posicionamiento global

## H

HID	Human Interface Device	Interface de usuario
HR	Heart Rate	abreviatura
HTML	HyperText Markup Language	Conjunto de lenguajes páginas Web

## I

ICR	Centro Instantáneo de Rotación	Punto respecto al cual el movimiento es rotación pura
IEEE	Institute of Electrical and Electronics Engineers	Asociación mundial de Ingenieros
ILS	Instrument Landing System	Sistema de aterrizaje y aproximación de aeronaves
IM	Inner Marker	abreviatura
iOS	Iphone Operative System	Sistema operativo para dispositivos iPhone
IOT	Internet Of The Things	Concepto, interconexión de dispositivos con Internet
IP	Internet Protocol	Protocolo
IPS	Sistema de Posicionamiento de Interiores	Sistema
ISM	Industrial Scientific and Medical	Bandas de radio

## L

LIDAR	Lght Detection And Ranging	Dispositivo para determinar distancias
LPF	Low Pass Filter	Filtro paso bajo

## M

MAC	Macintosh	abreviatura marca
MAC	Medium Access Control	Conjunto de protocolos y mecanismos
MFI	Made For Iphone	abreviatura
MM	Middle Marker	abreviatura

## N

NASA	National Aeronautics and Space Administration	Agencia Estadounidense del espacio
------	---	------------------------------------

## O

OM	Outer Marker	acrónimo
OM	Outer Marker	acrónimo
OTG	USB On-The-Go	Extensión de norma USB

## P

PAN	Personal Area Network	Redes de área personal
PWM	Pulse-width modulation	Tipo de Modulación por ancho del pulso

## R

RAE	Real Academia de la Lengua Española	Institución cultural
RC	Resistencia Condensador	acrónimo (electricidad, electrónica)
RH	Relative Humidity	acrónimo (física y química)
RHEA	Robot Fleets for Highly Effective Agryculture	Programa para el campo y los bosques
RPM	Revoluciones Por Minuto	Acrónimo
RSSI	Received Signal Strength Indicator	Escala de Referencia , mide potencia

## S

SARSAT	Search And Rescue Satellite-Aided Tracking	Programa Internacional Búsqueda y Salvamento
SFTP	Secure File Transfer Protocol	Protocolo de comunicación
SIM	subscriber identity module	Módulo de Identificación de abonado
SPP	Serial Port Profile (BLE)	Perfil BLE

## T

TFM	Trabajo Fin de Máster	Acrónimo Universitario
TLM	Telemetría, frame de Eddystone	Acrónimo (Eddystone)

## U

UHF	Ultra Low Frequency	Bandas de radio
UID	Identificador Único Universal	Número de 128 dígitos
URL	Identificador de Recursos Uniforme	Secuencia de caracteres formato estandarizado
USB	Universal Serial Bus	Bus de comunicación estándar más protocolos
UUID	Identificador Único Universal	Número de 128 dígitos

## 9. Bibliografía y recursos Web

### 9.1.- Bibliografía

- [1] [https://www.abc.es/sociedad/abci-japon-confia-tercera-generacion-robots-para-cuidar-ancianos-201709302021\\_noticia.html](https://www.abc.es/sociedad/abci-japon-confia-tercera-generacion-robots-para-cuidar-ancianos-201709302021_noticia.html)
- [2] Sistema de posicionamiento móvil para interiores vía WIFI. Jesús Piedra Vega. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/53021/7/pfi45js390pTFC0716memoria.pdf>
- [3] Bluetooth 4.0 Low Energy: Análisis de las prestaciones y aplicaciones para la automoción, TFG; Yassir Akhayad, UPC
- [4] Bluetooth BLE: el conocido desconocido 24 mayo, 2017 por Jesús Macías.  
<https://es.scribd.com/document/362986798/Bluetooth-BLE-El-Conocido-Desconocido>
- [5] ISM, [https://es.wikipedia.org/wiki/Banda\\_ISM](https://es.wikipedia.org/wiki/Banda_ISM)
- [6] "Wearable Sensors for Remote Health Monitoring", Sumit Majumder, Tapas Mondal .  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5298703/>
- [7] " Los tejidos inteligentes y el desarrollo tecnológico de la industria textil", Javier R. Sánchez  
<http://www.tecnicaindustrial.es/tifrontal/a-687-los-tejidos-inteligentes-desarrollo-tecnologico-industria-textil.aspx>
- [8] Microcontrolador  
<https://es.wikipedia.org/wiki/Microcontrolador>
- [9] Cuatro alternativas a Arduino: BeagleBone, Raspberry Pi, Nanode y Waspnote  
<https://blogthinkbig.com/4-alternativas-arduino-beaglebone-raspberrypi-nanode-waspnote>
- [10] Microordenador, raspberry, " [https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)",  
[https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)
- [11] "Desarrollo de un controlador domótico basado en el sistema Beaglebone Black", J López  
<http://bibing.us.es/proyectos/abreproy/12427/fichero/PFC-2427-LOPEZ.pdf>
- [12] Get Started with the Intel® Edison Development Board on Windows\* 64-bit  
<https://software.intel.com/es-es/get-started-edison-windows>
- [13] Sensores compatibles,  
<http://upm.mraa.io/findSensor.html>
- [14] Sensor HC-SR04,  
<https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>
- [15] "La tecnología beacons: una revolución en alza para la experiencia de usuario y las estrategias de marketing"



<https://www.uoc.edu/portal/es/news/actualitat/2016/099-beacons.html>

[16] ROBOT FLEETS FOR HIGHLY EFFECTIVE AGRICULTURE AND FORESTRY MANAGEMENT

<https://cordis.europa.eu/project/rcn/95055/factsheet/en>

[17] Curiosity Rover

<https://es.wikipedia.org/wiki/Curiosity>

[18] "Coches autónomos, el estado del arte con Alex Barredo", (Luis del Valle)

<https://programarfacil.com/podcast/coche-autonomo-estado-del-arte/>

[19] "Robot con Arduino, montando el Pirate 4WD paso a paso"

<https://programarfacil.com/blog/arduino-blog/robot-con-arduino-pirate-4wd-dfrobot/>

[20] " How to Build a tennis ball cannon"

<http://ballchoices.blogspot.com/2017/04/how-to-build-tennis-ball-cannon.html>

[21] <https://www.forodeltenis.com/materiales-y-equipamiento/proyecto-maquina-lanza-pelotas-de-tenis>

[22] Pádel, Wikipedia

<https://es.wikipedia.org/wiki/Pádel>

[23] "REGLAMENTO DE JUEGO DEL PÁDEL FEDERACIÓN INTERNACIONAL PÁDEL."

[https://www.padelfederacion.es/refs/docs/REGLAMENTO\\_DE\\_JUEGO\\_FIP\\_2017.pdf](https://www.padelfederacion.es/refs/docs/REGLAMENTO_DE_JUEGO_FIP_2017.pdf)

[24] " Golpes básicos en pádel". Irene Madroño -10/11/2018

<https://www.empadelados.com/golpes-basicos-padel/>

[25] <https://padeladdict.com/por-que-es-bueno-entrenar-al-padel/>

[26] "Diseño, desarrollo y test de un prototipo de pulsera para adquirir el electrocardiograma y la onda de pulso". Proyecto Fin de Carrera, Estudios: Ingeniería Electrónica, Año: 2016

Autor: Alberto Barea Cañizares, UPC

<https://upcommons.upc.edu/bitstream/handle/2117/98951/PFC.pdf?sequence=1&isAlloved=y>

[27] Recurso WEB, sondas ECG. "https://gouforit.com/polar-h7-pulsometro-opinion-y-analisis"

[28] RAE, Real Academia Española de la Lengua. (Inteligencia)

<http://lema.rae.es/drae2001/srv/search?id=2qlbggrBDXX2HVuWUL7>

[29] Wiring, <http://wiring.org.co/>

[30] Processing, <https://processing.org/>

[31] "Sistema de posicionamiento móvil para interiores vía WIFI", TFC de Jesús Piedra Vega, presentado en la Universidad Oberta de Catalunya (UOC) en Junio de 2016.

[32] "Sistema de Localización inalámbrica en entornos de interior mediante el uso del retardo temporal y el nivel de potencia de las señales de radiofrecuencia".

A.B. Martínez, Tesis doctoral 2009.

[33] Java, término designa el software y sus componentes: Java RuntimeEnvironment (JRE), Java Virtual Machine (JVM) y plugin.<https://www.java.com/es/>

[34] NodeJs, es un entorno de ejecución para JavaScript, orientado a eventos asíncronos.

<https://nodejs.org/es/>

[35] "Equilibrio entre el movimiento de traslación y el movimiento de rotación (I)".

[http://www.sc.ehu.es/sbweb/fisica/solido/mov\\_general/equilibrio/equilibrio.htm](http://www.sc.ehu.es/sbweb/fisica/solido/mov_general/equilibrio/equilibrio.htm)

[36] "Control de un Robot con ruedas de giro limitado"

TFG, David Ribera Corullón, ETSII, Universidad Politécnica de Madrid.

[37] "DISEÑO E IMPLEMENTACIÓN DE UN ROBOT AUTOBALANCEADO"

TFM Jorge Ferrer Gómez, Escuela Técnica Superior de Ingeniería del Diseño, Valencia

[39] "La técnica en el tenis", artículo Web.

<http://www.padeltour.es/gotenis/noticias/guias-del-tenis/los-efectos-en-el-tenis.aspx>

## 9.2.- Recursos Web

Visitas entre Febrero y Junio del 2019

Arduino

<https://www.arduino.cc/>

BLENO / NOBLE

<https://github.com/noble/bleno>

Edison

<https://software.intel.com/es-es/iot/hardware/discontinued>

<https://software.intel.com/es-es/get-started-edison-windows>

NODE.JS

<https://nodejs.org/es/>

Putty

<https://www.putty.org/>

Wikipedia

<https://es.wikipedia.org/wiki/Wikipedia:Portada>

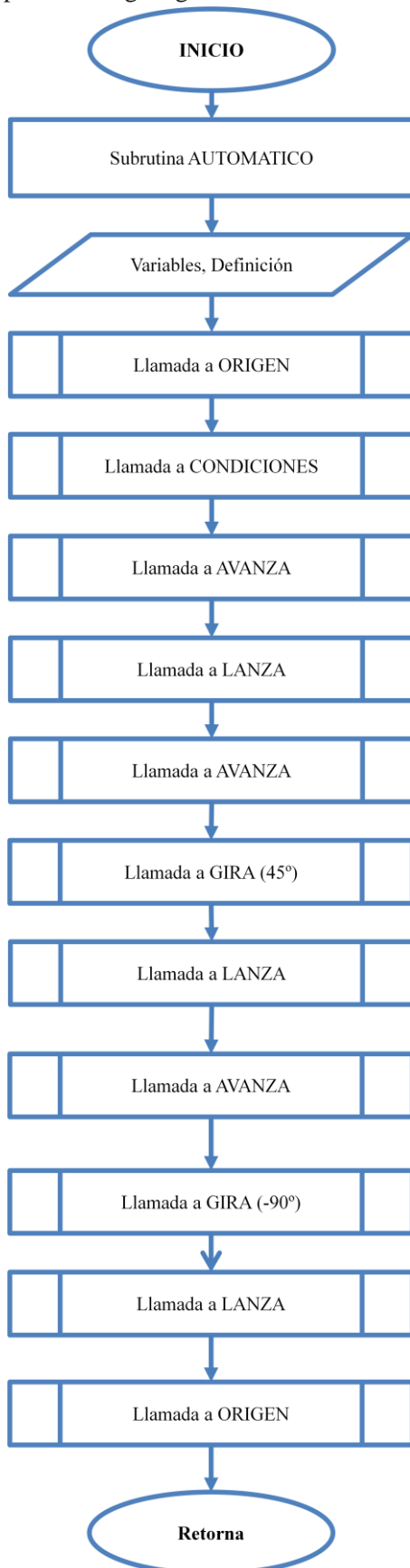
WinSCP

<https://winscp.net/eng/download.php>

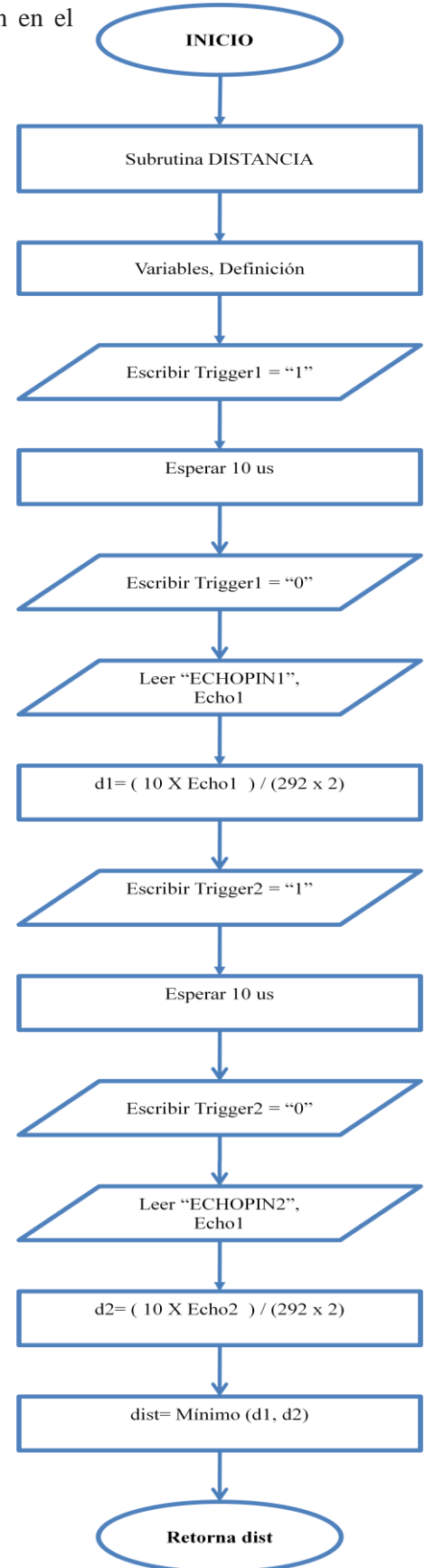
## 10. Anexos

## 10.1 Anexo I Pseudocódigo

La subrutina "AUTOMÁTICO", y la de Distancia se basarán en el pseudocódigo siguiente:

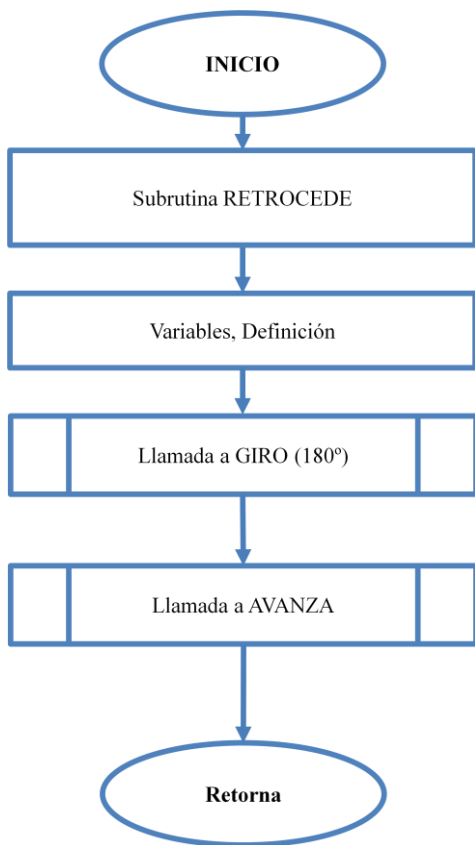


Pseudocódigo 3: Subrutina Automático

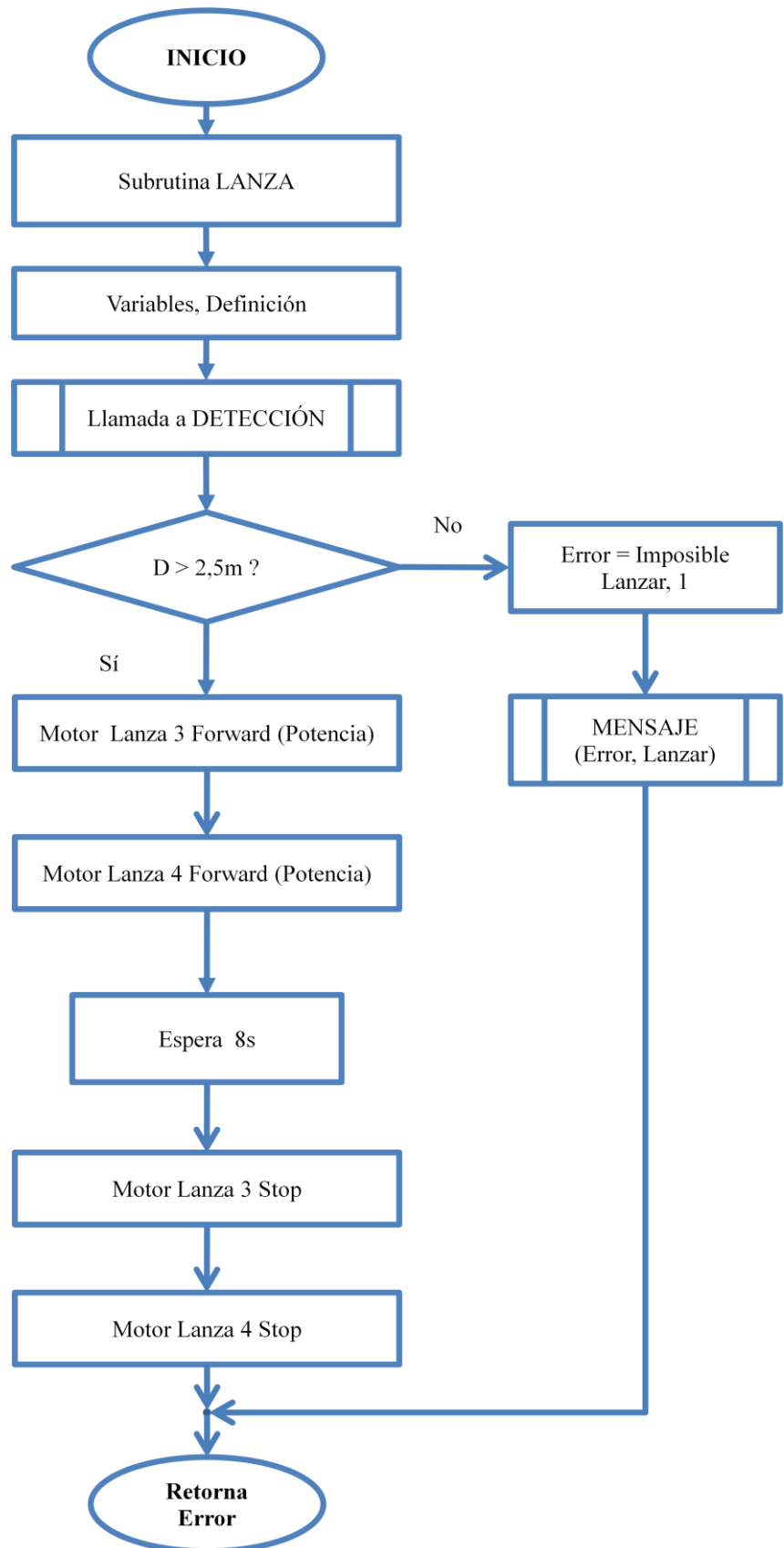


Pseudocódigo 2: Subrutina Distancia

## Las subrutinas Retrocede y Lanza

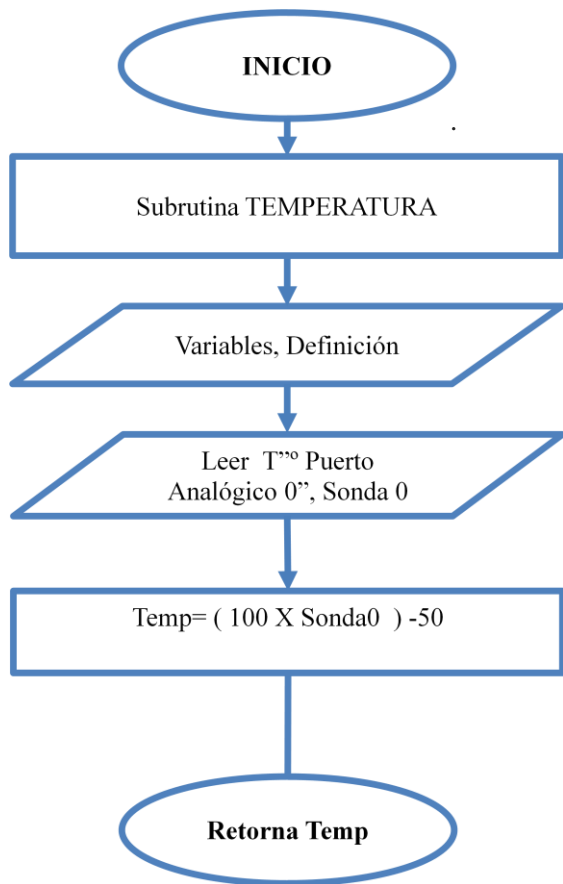


Pseudocódigo 5: Subrutina Retrocede

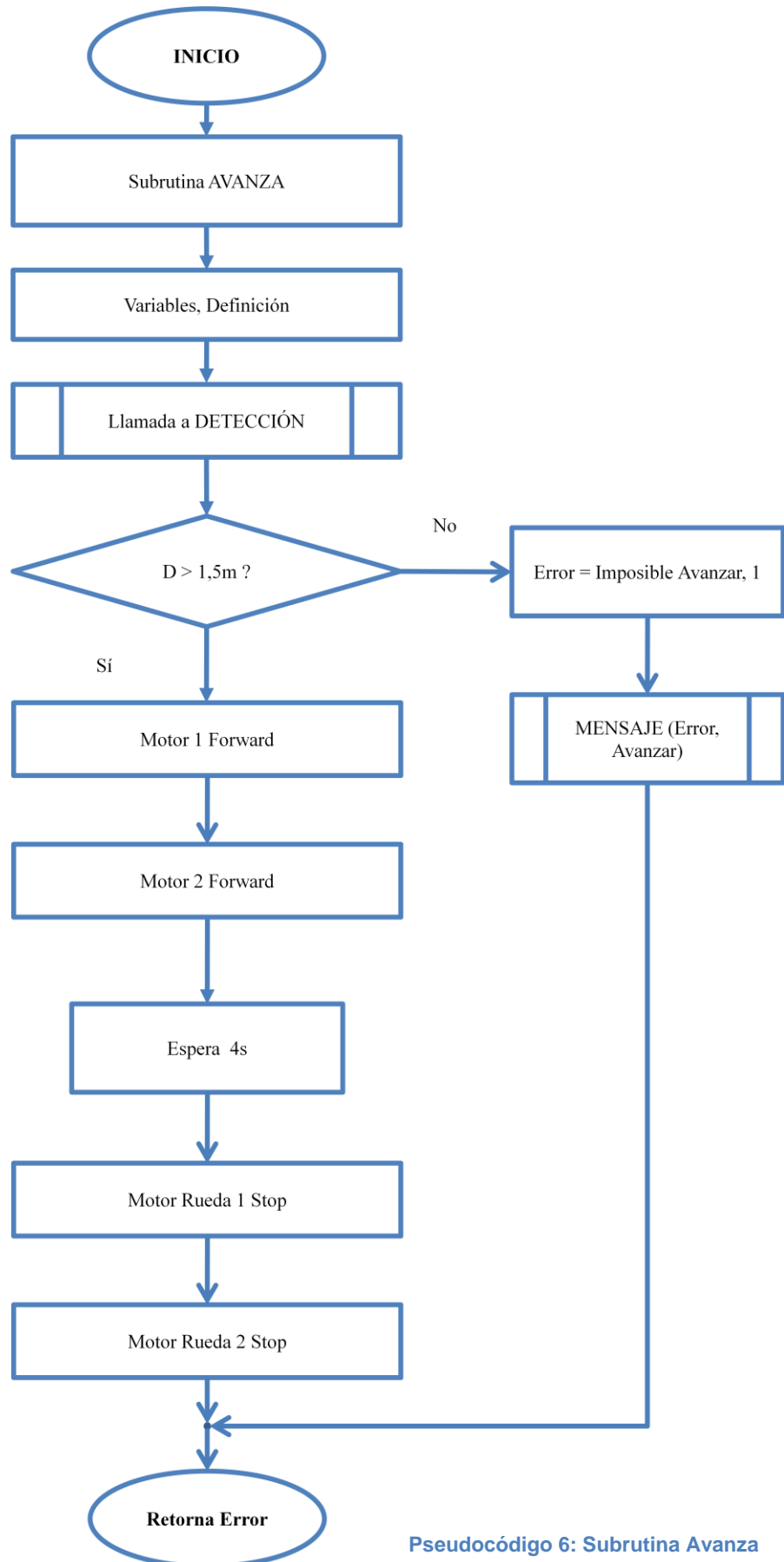


Pseudocódigo 4: Subrutina Lanza

Las subrutinas Temperatura y Avanza

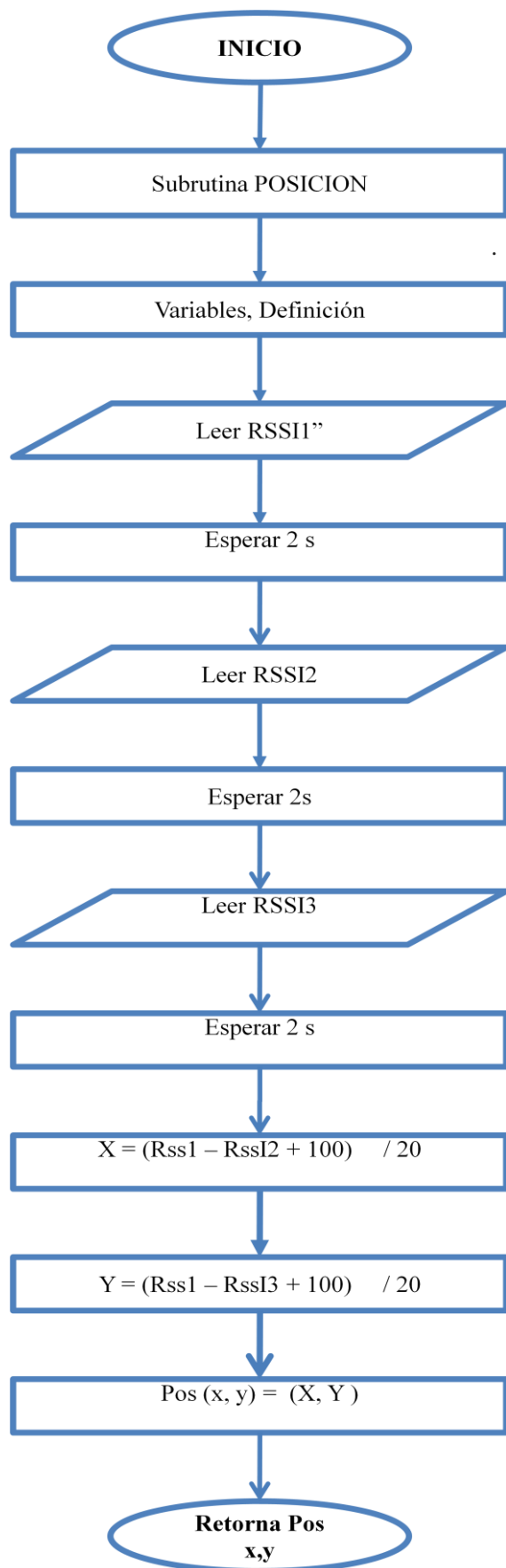


Pseudocódigo 7: Subrutina Temperatura

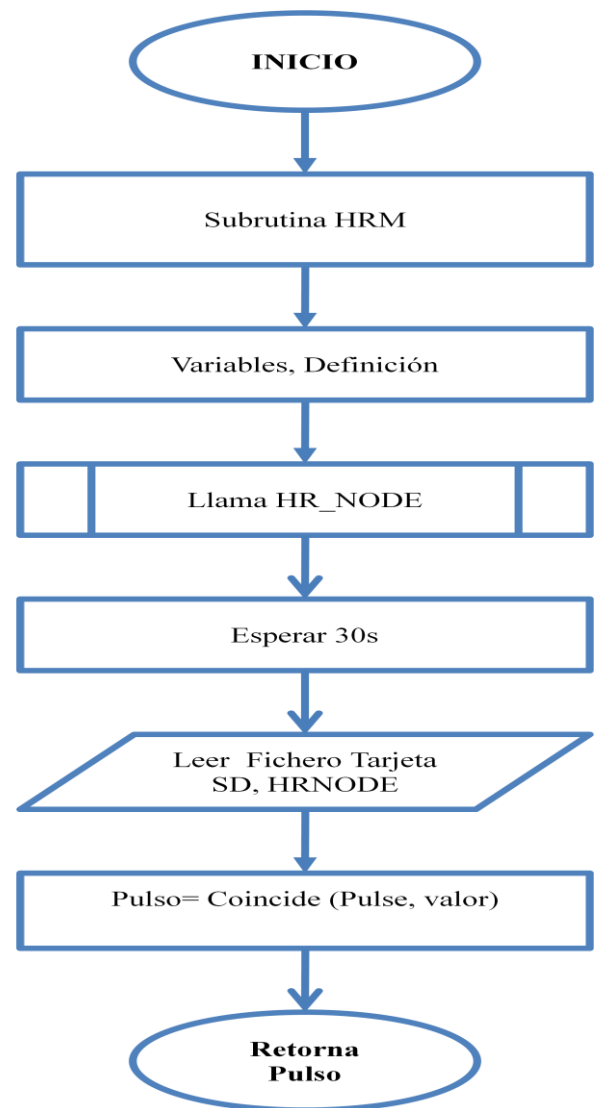


Pseudocódigo 6: Subrutina Avanza

Las de Posición y Pulso, Heart Rate

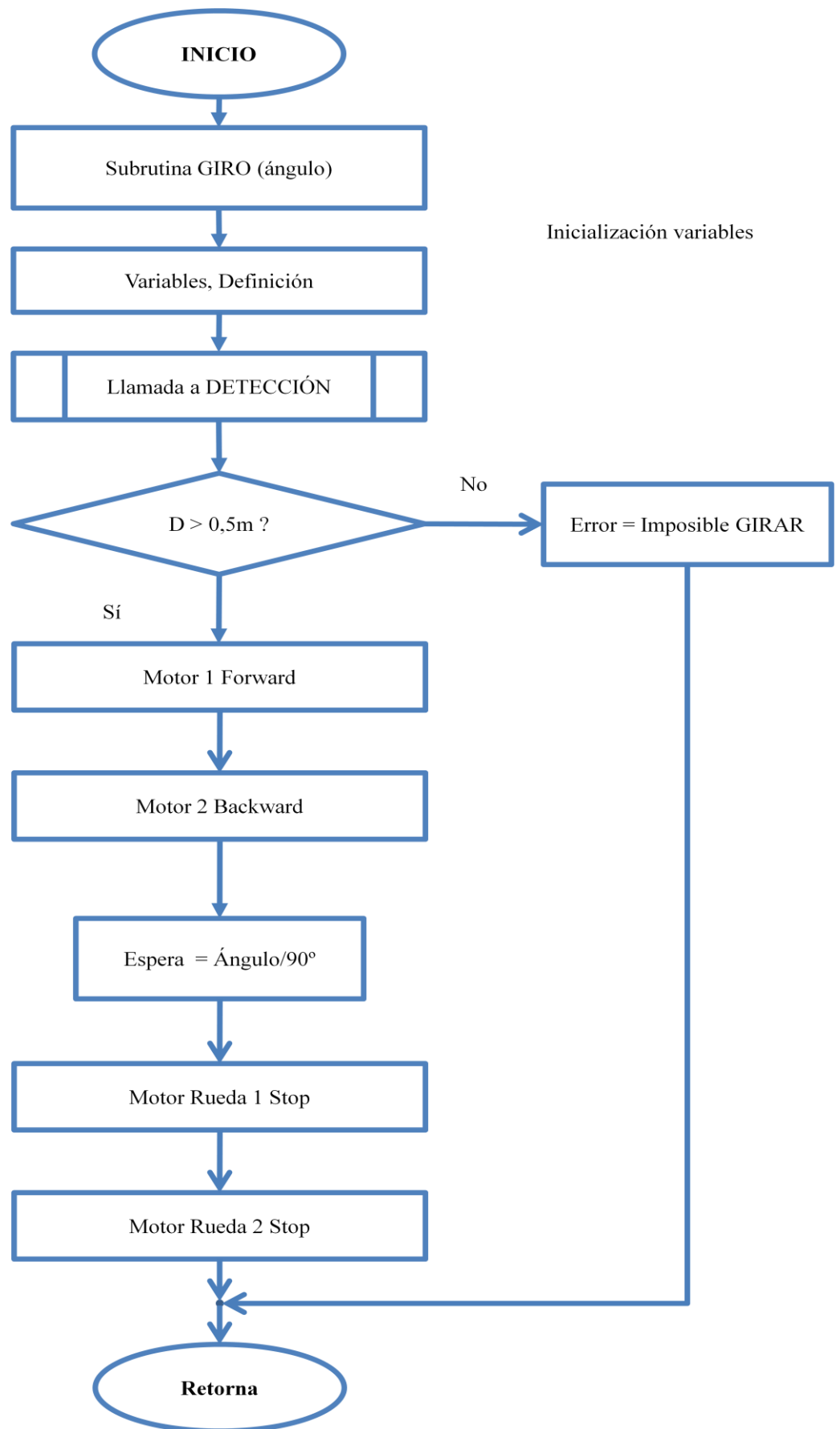


Pseudocódigo 9: Programa Posición



Pseudocódigo 8: Subrutina Ritmo Cardíaco HRM

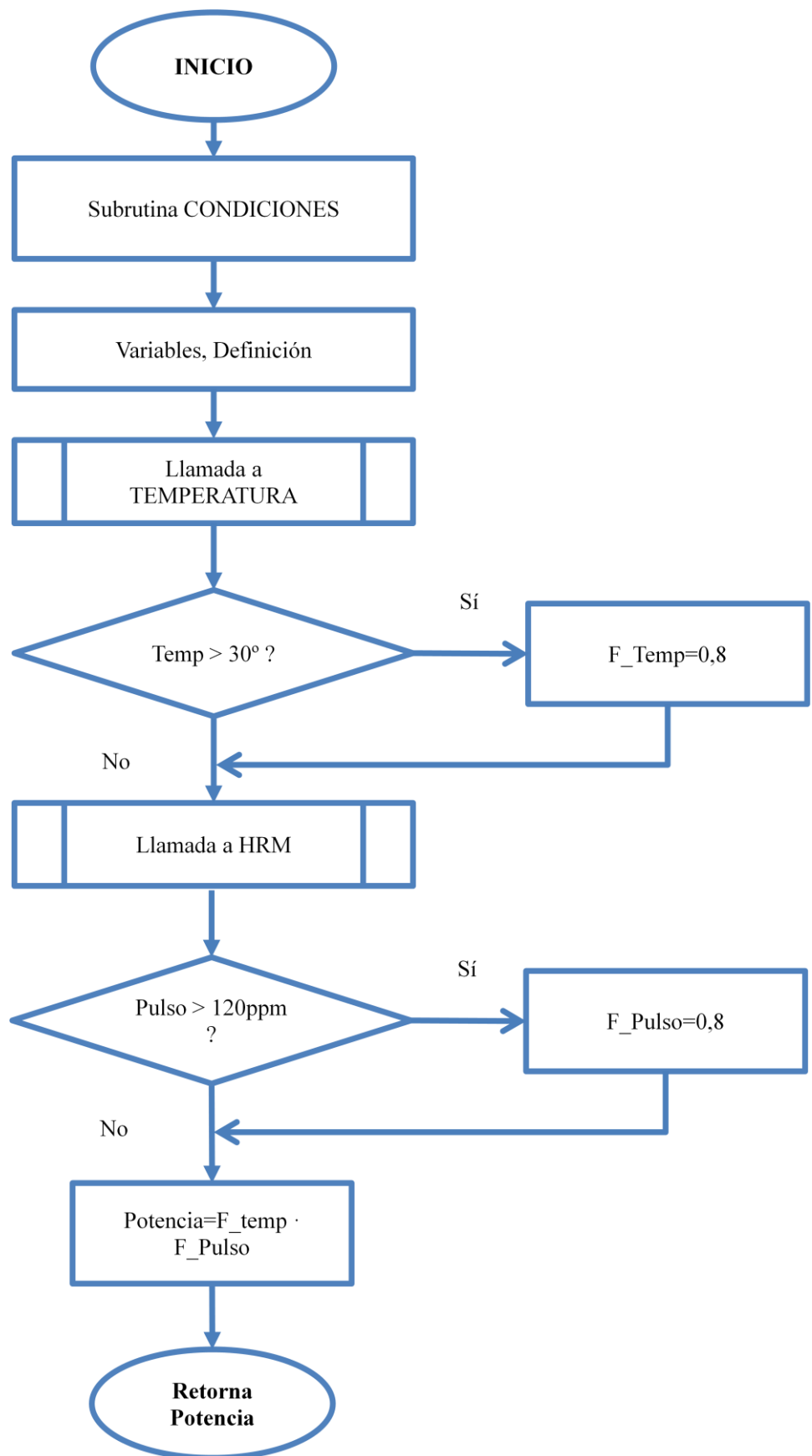
La subrutina de Giro;



Pseudocódigo 10: Subrutina Giros

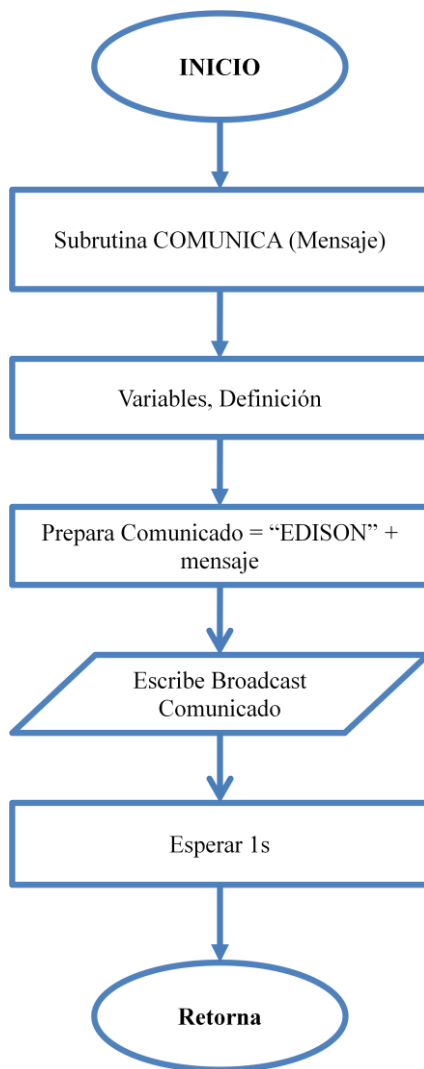


La subrutina de condiciones (Potencia retornada)

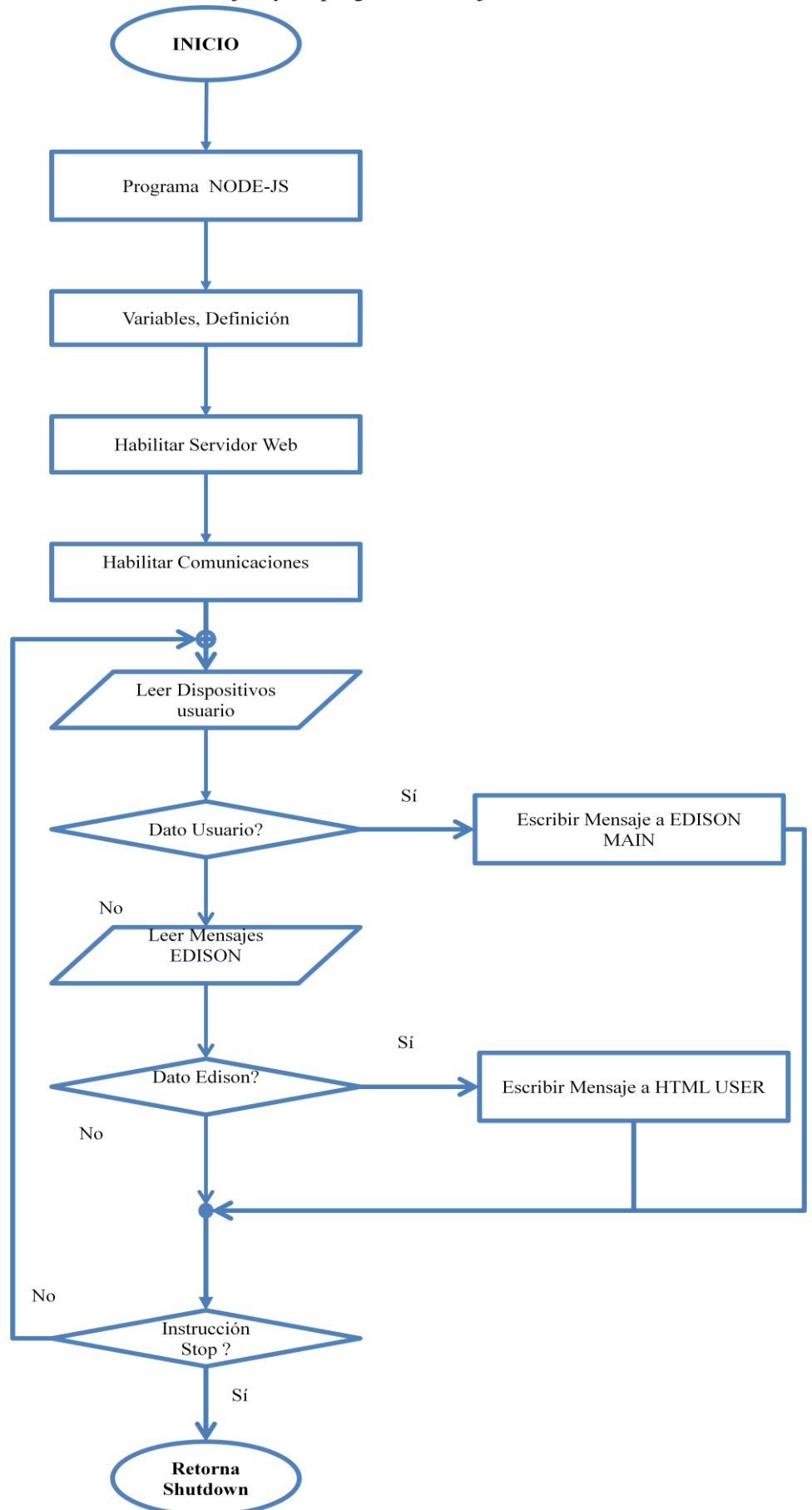


Pseudocódigo 11: Subrutina Condiciones

Por último las subrutinas de comunicación de mensajes, y de programa Node.js Web Server



Pseudocódigo 12: Subrutina comunicación



Pseudocódigo 13: Programa Servidor + Comunicación

## 10.2 Anexo II Programa principal de control del Intel edison

```
/"Entrenador Personal Inteligente"
// Programa IDE ARDUINO, Programa 1/4
// TFM Jesus Garcia UOC Ramon Llull
// IDE ARDUINO

#define ESP32
#include <SocketIOClient.h>
#include <string.h>
SocketIOClient client;
char* ssid = "A";
const char* password = "holahola";
char b[2];
char host[] = "127.0.0.1";
int port = 8080;
extern String RID;
extern String Rname;
extern String Rcontent;
int pinLM35 = 0; // Variable del pin de entrada del sensor (A0)
int pinli=3, pinld=5, pinri=6, pinrd=9, dri=7, drd=8, pri=12, prd=13; // Motores
int casos[10]={0,45, 0, -45, 90, 0, -90, 135, 180, -135}; // Giros
const int EchoPin1 = 2; // Distancia Variables
const int TriggerPin1 = 4;
const int EchoPin2 = 10;
const int TriggerPin2 = 11;
const int pld = 18;
const int pli = 19;
void setup() {
    // Distancia 2 sondas
    pinMode(TriggerPin1, OUTPUT);
    pinMode(EchoPin1, INPUT);
    pinMode(TriggerPin2, OUTPUT);
    pinMode(EchoPin2, INPUT);
    // Motores lanzador apago=255
    pinMode(dri, OUTPUT);
    pinMode(pri, OUTPUT);
    pinMode(drd, OUTPUT);
    pinMode(prd, OUTPUT);
    lanzaonoff();
    digitalWrite(prd, LOW);
    digitalWrite(pri, LOW);
    analogWrite(pinld, 255);
    analogWrite(pinli, 255);
    // WiFi
    Serial.begin(115200);
    delay(10);
    Serial.print("Conectando a : ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
}
```

```

if (!client.connect(host, port)) {
    Serial.println("Imposible conectar");
    return;
}
}

void loop() {
    int aa=0;
    int potencia=100; // valores de entre 0-100
    int tiempo=1000; //1 s, se modifica en función de giro
    if (client.monitor())
    {
        aa= RID.toInt();
        Serial.println(RID);
        switch (aa) {
            case 2: // Avanza
                Serial.print("Me ha llegado 2 desde html ");
                tiempo=740*3; //(740ms equivale a 1m) al 100% de potencia
                avanza(potencia, tiempo);
                client.send("ARDUINO", "message", b);
                break;
            case 5: // LANZA

                lanza();
                client.send("ARDUINO", "message", b);
                break;
            case 10: // Temperatura
                temperatura();
                client.send("TE", "message", b);
                break;
            case 11: // auto
                automatica();
                client.send("TE", "message", b);
                break;
            case 1: case 3: case 4: case 6: case 7: case 8: case 9: // auto
                giro(casos[aa]);
                client.send("ARDUINO", "message", "Giro ok");
                break;
            default:
                ;
                client.send("ARDUINO", "message", "posicion ok");
                break;
        }
    }
}
}

```

// Declaracion de variables globales

```

void temperatura() {
    String st1;
    float tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)
    tempC = analogRead(pinLM35); //Leo sensor
    tempC = (5.0 * tempC * 100.0)/1024.0; // Calculamos la temperatura con la fórmula
    int tempe= (int) tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)
    st1=String(tempe);
}

```

```

    st1.toCharArray(b,3);
}

void giro(int angulo) {
    int intervalo=1000;
    digitalWrite(prd, HIGH);
    digitalWrite(pri, HIGH);
    Serial.println(angulo);
    if (angulo<0){
        digitalWrite(drd, HIGH); //deben tener igual nivel porque al estar montadas al contrario
        giran al revés
        digitalWrite(dri, HIGH);
    }
    else{
        digitalWrite(drd, LOW);
        digitalWrite(dri, LOW);
    }
    intervalo=abs(12*angulo);
    analogWrite(pinrd, 100);
    analogWrite(pinri, 100);
    delay(intervalo);

    analogWrite(pinrd, 0);
    analogWrite(pinri, 0);
    digitalWrite(prd, LOW);
    digitalWrite(pri, LOW);
    Serial.println("angulo");
    Serial.println(intervalo);
}

void lanza() {
    distancia();
    int intervalo=5000;
    lanzaonoff();
    analogWrite(pinld, 140);
    analogWrite(pinli, 140);
    delay(intervalo);
    analogWrite(pinld, 255);
    analogWrite(pinli, 255);
    lanzaonoff();
    Serial.println("lanza");
}

void avanza(int potencia, int tiempo) {
    distancia();
    int intervalo=3000;
    int velocidad= (int) (potencia * 100/100); //el 100 del num debe ser 255
    velocidad=200;
    digitalWrite(dri, HIGH); //Deben estar cambiadas una alta, otra baja, pues están montadas al
    revés
    digitalWrite(drd, LOW);
    analogWrite(pinrd, velocidad);
    analogWrite(pinri, velocidad);
    digitalWrite(prd, HIGH);
    digitalWrite(pri, HIGH);
    delay(intervalo);
    analogWrite(pinrd, 0);

```

```

    analogWrite(pinri, 0);
    digitalWrite(prd, LOW);
    digitalWrite(pri, LOW);
    Serial.println("Avanza");
}
void distancia() {
    int cm1 = ping(TriigerPin1, EchoPin1);
    Serial.print("Distancia Sonda HC: ");
    Serial.println(cm1);
    delay (100);
    int cm2 = ping(TriigerPin2, EchoPin2);
    Serial.print("Distancia Sonda JSN: ");
    Serial.println(cm2);
    delay(1000);
}
int ping(int TriggerPin, int EchoPin) {
    long duration, distanceCm;
    digitalWrite(TriggerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
    delayMicroseconds(4);
    digitalWrite(TriggerPin, HIGH); //generamos Trigger (disparo) de 10us
    delayMicroseconds(10);
    digitalWrite(TriggerPin, LOW);
    duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos, en microsegundos
    distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
    return distanceCm;
}

void lanzaonoff() {
    pinMode(pld, OUTPUT); //enciendo motor lanzador derecho
    pinMode(pli, OUTPUT);
    digitalWrite(pld, LOW);
    digitalWrite(pli, LOW);
    delay(50);
    pinMode(pld, INPUT);
    pinMode(pli, INPUT);
}

void automatica() {
    avanza(80,740);
    lanza();
    avanza(80,740);
    giro(casos[1]);
    lanza();
    giro(casos[6]);
    lanza();
    giro(casos[1]);
    avanza(80,740);
    lanza();
    giro(casos[8]);
    avanza(80,740);
    avanza(80,740);
    avanza(80,740);
    giro(casos[8]);
}

```

### 10.3 Anexo III Servidor nodejs

```
//"Entrenador Personal Inteligente"
// Programa Servidor HTML, Programa 2/4
// TFM Jesus Garcia UOC Ramon Llull
// Nodejs
var util = require('util');
var express = require('express')
var app = express();
app.use(express.static(__dirname + '/public'))
var server = require('http').createServer(app);
var io = require('socket.io')(server);
var fs = require('fs');
var system= require('child_process').exec;
var http = require('http');
var Edisonstatus="On";
var a=2500, x, y;
var tout;
server.listen(8080);
io.on('connection', function (socket) {
  console.log("Connected");
  socket.on('ARDUINO', function (data) {
    //io.sockets.emit('mensaje', "desde Arduino");
    a=data.toString()
    socket.broadcast.emit('mensaje', data.message)
    console.log(data.message);
  });
  socket.on('HR', function(data){
    a=parseInt(data)
    a=a*2
    socket.emit('HR', a)
    console.log('HR',a)
  });
  socket.on('TE', function(data){
    a=parseInt(data.message)
    socket.broadcast.emit('TE', a)
    console.log('TE',a)
  });
  socket.on('GP', function(data){
    fs.readFile('x.txt', 'utf8', (error, datos) => {
      x=Math.floor(datos);
      console.log('GxP',x)
    });
    fs.readFile('y.txt', 'utf8', (error, datos) => {
      y=Math.floor(datos);
      console.log('GPY',y);
    });
    tout=setTimeout(gpxy, 1000)
  });
});
```

```

});
socket.on('Edison_State', function(data){
    var Edisonstatus=data.toString()
    io.sockets.emit(Edisonstatus, 'EDISON');
    console.log('edison',Edisonstatus)
});
function gpxy()
{
    var xy= 100*x +y;
    socket.emit('GP',xy);
    var yx;
    yx= xy.toString();
    io.sockets.emit(yx, 'EDISON');
    console.log('GPxy',yx);
    clearTimeout(tout)
}
});

```



## 10.4 Anexo IV Ibeacon y cálculo de posición

```
// "Entrenador Personal Inteligente"
// Programa sondas app (Noble), Programa 3/4
// TFM Jesus Garcia UOC Ramon Llull
// Noble

var noble = require('noble');
var uuid= ["00001803494c4f474943544543480000"];
const fs = require('fs');
var id5;
var id2 = "c31316fa136a"; // id of device we want to discover
var id3 = "ff46c6d3a66f";
var id1 = "c7d637c4d7cc";
var id4= "e84ccc3a304b";
var contax=0, contay=0, contaz=0;
var peri1;
var xx=0;
var yy=0;
var zz=0;
var rs=[53, 57,64,68,72,76,79,82,85,87,88,89,91,92,94]
noble.on('stateChange', function(state) {
  console.log('on -> stateChange: ' + state);
  if (state === 'poweredOn') {
    noble.startScanning(uuid, true);
  } else {
    noble.stopScanning();
  }
});
noble.on('discover', function(peripheral) {
  id5=peripheral.id;
  //console.log('on -> discover: ' + peripheral.id + peripheral.rssi);
  noble.stopScanning();

  if (id5==id1){
    peri1=peripheral;
    console.log('RSSI-x: ' + peripheral.rssi );
    xx=peripheral.rssi;
    contax=contax+1;
  }
  if (id5==id2){
    yy=peripheral.rssi;
    console.log('RSSI-y ' + peripheral.rssi );
    contay=contay+1;
  }
  if (id5==id3){
    zz=peripheral.rssi;
    console.log('RSSI-z ' + peripheral.rssi );
    contaz=contaz+1;
  }
  if (id5==id4){
    peripheral.connect(function(error){
```

```

var serviceUUID = ["180d"];
var characteristicUUID = ["2a37"]; // 2a37 characteristic heart rate
peripheral.discoverSomeServicesAndCharacteristics(serviceUUID, characteristicUUID, function(error,
services, characteristics){
  characteristics[0].notify(true, function(error){
    characteristics[0].on('data', function(data, isNotification){
      fs.writeFile("./h.txt", data[1], function (err) {
        if (err) {
          return console.log(err);
        }
      });
    });
    console.log('data is: ' + data[1]);
    //peripheral.disconnect();
    tout=setInterval(restart, 4000);
  });
});
});
});
});
}
if (contax<1 || contay<1 || contaz<1) {

  noble.startScanning(uuid, false)
  //console.log('Enciendo escanner');
  }else //if (conta==2){
  {
    xx=convierte(xx);
    yy=convierte(yy);
    zz=convierte(zz);
    var x= (xx*xx - yy*yy +100)/20;
    fs.writeFile("./x.txt", x, function (err) {
      if (err) {
        return console.log(err);
      }
    });
    var y= (xx*xx - zz*zz +100)/20;
    fs.writeFile("./y.txt", y, function (err) {
      if (err) {
        return console.log(err);
      }
    });
    noble.startScanning(["180d"]);
    tout=setInterval(restart, 4000);
  }
});
function restart()
{
  var fs1 = require('fs'),
  spawn = require('child_process').spawn,
  out = fs1.openSync('./out.log', 'a'),
  err = fs1.openSync('./out.log', 'a');
  spawn('node', ['scan1.js'], {
    stdio: [ 'ignore', out, err ], // piping stdout and stderr to out.log
    detached: true
  }

```

```

        }).unref();
        process.exit(0);
    }
function convierte( rss)
{
    var ds=0;
    rss=-rss;
    var ii=0;
    while (ii<=14){
        if (rss>=rs[ii]){
            ds=ii+1;
        }
        ii++;
    }
    return ds;
}
//*****

```

## 10.5 Anexo V Programa cliente

```
<!doctype html>
<!-- //"Entrenador Personal Inteligente"
// Programa Cliente HTML, Programa 4/4
// TFM Jesus Garcia UOC Ramon Llull
-->
<html>
<head>
  <script src='jquery-1.7.2.min.js'></script>
  <script src='socket.io-1.3.4.js'></script>
  <script>
    //var socket = io.connect('//192.168.1.50:8080');
    var socket = io.connect('//192.168.43.64:8080')
    var a="On"
    var reta=2500
    socket.on('mensaje', function(data){
      console.log('mensaje:recibido:', data)
      a=data.toString()
      MSG(a)
    })
    socket.on('HR', function(data){
      reta=parseInt(data)
      MSGHR(reta)
      console.log('HR', data)
    })
    socket.on('TE', function(data){
      reta=parseInt(data)
      MSGTE(reta)
      console.log('TE', data)
    })
    socket.on('GP', function(data){
      a=data.toString()
      MSGGP(a)
      console.log('GP', data)
    })

    socket.on('NN', function(data){
      a=data.toString()
      MSG(a)
      console.log('NN', data)
    })
    socket.on('SS', function(data){
      a=data.toString()
      MSG(a)
      console.log('SS', data)
    })
  </script>
</head>

<body>
<table>
  <tr>
    <td>
      <input type = "image" src="heart.jpg" id="heart" width="150" height="150"
      onclick = 'HR()'>
    </td>
    <td>
      <input type = "image" src="termo.jpg" id="termo" width="150" height="150"
      onclick = 'NO(10)'>
    </td>
  </tr>
</table>
</body>
</html>
```

```

                </td>
                <td>
                    <input type = "image" src="gpsxy.jpg" id="gpsxy" width="150" height="150"
onclick = 'GP()'>
                </td>
            </tr>
            <tr>
                <td>
                    <input type="text" id="textoh" value="60" size="7" style="font-size:25px"/>
                </td>
                <td>
                    <input type="text" id="textot" value="20" size="7" style="font-size:25px"/>
                </td>
                <td>
                    <input type="text" id="textog" value="0,0" size="7" style="font-size:25px"/>
                </td>
            </tr>
            <tr>
                <td>
                    <input type = "image" src="auto.jpg" id="auto" width="150" height="150"
onclick = 'AU()'>
                </td>
                <td>
                    <input type="text" id="textom" value="Lanzabola" size="7" style="font-
size:25px"/>
                </td>
                <td>
                    <input type = "image" src="stop.jpg" id="stop" width="150" height="150"
onclick = 'ST()'>
                </td>
            </tr>
            <tr>
                <td>
                    <input type = "image" src="NO.jpg" id="NOID" width="150" height="150"
onclick = 'NO(1)'>
                </td>
                <td>
                    <input type = "image" src="N.jpg" id="NID" width="150" height="150"
onclick = 'NO(2)'>
                </td>
                <td>
                    <input type = "image" src="NE.jpg" id="NEID" width="150" height="150"
onclick = 'NO(3)'>
                </td>
            </tr>
            <tr>
                <td>
                    <input type = "image" src="O.jpg" id="OID" width="150" height="150"
onclick = 'NO(4)'>
                </td>
                <td>
                    <input type = "image" src="LANZA.jpg" id="LAID" width="150"
height="150" onclick = 'NO(5)'>
                </td>
                <td>
                    <input type = "image" src="E.jpg" id="EID" width="150" height="150"
onclick = 'NO(6)'>
                </td>
            </tr>

```

```

        <tr>
            <td>
                <input type = "image" src="SO.jpg" id="SOID" width="150" height="150"
onclick = 'NO(7)'>
            </td>
            <td>
                <input type = "image" src="S.jpg" id="SID" width="150" height="150"
onclick = 'NO(8)'>
            </td>
            <td>
                <input type = "image" src="SE.jpg" id="SEID" width="150" height="150"
onclick = 'NO(9)'>
            </td>
        </tr>
    </table>
    <script>
        function HR(){
            socket.emit('HR', '60')
            console.log('Envio', 'HR')
        }

        function GP(){
            socket.emit('GP', '0,0')
            console.log('Envio', 'GP')
        }

        function AU(){
            socket.emit('Edison_State', '11')
            console.log('Envio', 'AU')
        }

        function ST(){
            socket.emit('Edison_State', 'ST')
            console.log('Envio', 'ST')
        }

        function NO(a){
            var aa=a
            socket.emit('Edison_State', a)
            console.log('Envio', a)
        }

        function MSG(r){
            var rr=r
            document.getElementById('textom').value=rr
        }

        function MSGHR(r){
            var rr=r
            document.getElementById('textoh').value=rr
        }

        function MSGTE(r){
            var rr=r
            document.getElementById('textot').value=rr
        }

        function MSGGP(r){
            var rr=r
            document.getElementById('textog').value=rr
        }
    </script>
</body>
</html>

```

## 10.6 Anexo VI GNU Free License

### **GNU Free Documentation License**

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### **0. PREAMBLE**

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### **1. APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.



## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this

License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and

not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## **11. RELICENSING**

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the

document and put the following copyright and license notices just after the title page:

*Copyright (C) YEAR YOUR NAME.*

*Permission is granted to copy, distribute and/or modify this document*

*under the terms of the GNU Free Documentation License, Version 1.3*

*or any later version published by the Free Software Foundation;*

*with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.*

*A copy of the license is included in the section entitled "GNU*

*Free Documentation License".*

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

*with the Invariant Sections being LIST THEIR TITLES, with the*

*Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.*

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.