

INSTITUTO SUPERIOR TECNOLÓGICO

VIDA NUEVA



CARRERA:

TECNOLOGIA EN ELECTROMECHANICA

TEMA DEL PROYECTO DE APLICACIÓN PRÁCTICA:

DESARROLLO DE UN SISTEMA PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ IMPLEMENTADOS EN UNA CABEZA ROBÓTICA

AUTOR:

CORTEZ PEÑAFIEL MICHAEL ANDRES

TUTOR:

ING. TOCA TRUJILLO LUIS EDISON

FECHA:

MARZO 2020

QUITO – ECUADOR

CESIÓN DE DERECHOS DE AUTOR

Yo, **CORTEZ PEÑAFIEL MICHAEL ANDRES** portador/a de la cedula de ciudadanía **172708085 – 3**, facultado/a de la carrera **TECNOLOGIA EN ELECTROMECAÁNICA**, autor/a de esta obra certifico y proveo al **Instituto Superior Tecnológico Vida Nueva**, usar plenamente el contenido plasmado en este escrito con el tema **“DESARROLLO DE UN SISTEMA PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ IMPLEMENTADOS EN UNA CABEZA ROBOTICA”**, con el objeto de aportar y promover la lectura e investigación, autorizando la publicación de mi trabajo de titulación en la colección digital del repositorio institucional bajo la licencia de Creative Commons: Atribución-NoComercial-SinDerivadas.

En la ciudad de Quito, del mes de Marzo de 2020.

CORTEZ PEÑAFIEL MICHAEL ANDRES

C.I.: 172708085-3

CERTIFICACIÓN

En mi calidad de Tutor del Proyecto: **“DESARROLLO DE UN SISTEMA PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ IMPLEMENTADOS EN UNA CABEZA ROBOTICA”** en la ciudad de Quito, presentado por el ciudadano **CORTEZ PEÑAFIEL MICHAEL ANDRES**, para optar por el título de Tecnólogo en **ELECTROMECHANICA**, certifico, que dicho proyecto ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe

En la ciudad de Quito, del mes de Marzo de 2020.

TUTOR: ING. TOCA TRUJILLO LUIS EDISON

C.I.: 171988925-3

APROBACIÓN DEL JURADO EXAMINADOR O TRIBUNAL

Los miembros del tribunal examinador aprueban el informe del Proyecto de Aplicación práctica, sobre el tema: **“DESARROLLO DE UN SISTEMA PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ IMPLEMENTADOS EN UNA CABEZA ROBOTICA”** en la ciudad de Quito.

Del estudiante: **CORTEZ PEÑAFIEL MICHAEL ANDRES**

De la Carrera, **TECNOLOGÍA EN ELECTROMECHANICA**

Para constancia firman:

C.I.:

C.I.:

C.I.:

C.I.:

DECLARACIÓN DE AUTENTICIDAD

Yo, **CORTEZ PEÑAFIEL MICHAEL ANDRÉS**, estudiante del **Instituto Superior Tecnológico Vida Nueva**, portador/a de la cedula de ciudadanía **172708085 – 3**, declaro que he realizado este trabajo de titulación tomando en consideración citas bibliográficas que se nombran en texto.

El Instituto Superior Tecnológico “Vida Nueva” puede utilizar este trabajo de titulación como una ayuda bibliográfica.

En la ciudad de Quito, del mes de Marzo de 2020.

CORTEZ PEÑAFIEL MICHAEL ANDRES

C.I.: 172708085-3

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	I
CERTIFICACIÓN	II
APROBACIÓN DEL JURADO EXAMINADOR O TRIBUNAL.....	III
DECLARACIÓN DE AUTENTICIDAD.....	IV
INDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS	X
ÍNDICE DE ANEXOS	XI
1. INTRODUCCIÓN.....	1
2. ANTECEDENTES	2
3. OBJETIVOS	4
3.1. Objetivo General.....	4
3.2. Objetivos específicos.....	4
4. DESARROLLO.....	5
4.1. Marco Teórico – Conceptual	5
4.2. La Robótica.....	5
4.3. Historia de la robótica.....	5
4.4. Robot	8
4.4.1. Definiciones de Robot:	9
4.5. Robot social	10
4.6. Inteligencia artificial.....	13
4.7. Reconocimiento por Voz.....	16
4.7.1. Problemas Del Análisis Y Reconocimiento De Voz	18
4.8. Hardware	19
4.8.1. Impresión 3D	20
4.8.2. Módulo EasyVr 3.0.....	21
4.8.2.1. Características del Easy VR 3	22

4.8.2.2.	Easy VR 3 Como Placa de Desarrollo.....	23
4.8.2.3.	Procedimiento Independiente del Módulo Easy VR 3	24
4.8.2.4.	Especificaciones Técnicas	25
4.8.2.5.	Interfaz de Comunicación.....	27
4.8.2.6.	Protocolo de comunicación	27
4.8.3.	Arduino Uno	28
4.8.3.1.	Especificaciones Técnicas	29
4.8.4.	Arduino Nano	30
4.8.4.1.	Especificaciones Técnicas	30
4.8.5.	Servomotores	31
4.8.5.1.	Estructura interna.....	31
4.8.5.2.	Terminales	32
4.8.6.	Servomotor MG996R	32
4.8.6.1.	Especificaciones Técnicas	34
4.9.	Software.....	35
4.9.1.	Software Arduino IDE.....	35
4.9.2.	Software EasyVR Commander.....	36
4.9.3.	Software Quick Synthesis 5.....	37
4.9.4.	Software Adobe Audition.....	37
5.	PROCEDIMIENTO – METODOLOGÍA.....	38
5.1.	Diseño.....	38
5.1.1.	Fuente variable 12V.....	38
5.1.1.1.	Elaboración De Circuito Impreso	38
5.1.2.	Amplificador de audio 20W	40
5.1.2.1.	Elaboración del Circuito Impreso.....	41
5.1.3.	Conexión Easy VR Shield 3 y Arduino Uno.....	41
5.1.3.1.	Características Easy VR Shield 3	42

5.1.3.2.	Configuración de Puente	45
5.1.4.	Conexión de Arduino Uno con Servomotores.....	46
5.2.	Construcción.....	48
5.2.1.	Ensamblaje de Servomotores en la Cabeza Robótica.....	48
5.2.2.	Ensamblaje de Módulos.....	48
5.2.3.	Instalación de Parlantes 10W	48
5.3.	Implementación	48
5.3.1.	Sintetización De Sonidos De Voz	48
5.3.2.	Carga De Tabla De Sonidos	50
5.3.3.	Importación De Tabla De Sonido Al Software EasyVr. Commander.....	53
5.3.4.	Configuración Del Reconocimiento De Voz.....	55
5.3.5.	Generación De Código IDE en el EasyVr Commander	58
5.3.6.	Programación en Arduino IDE	59
5.3.6.1.	Programación del Reconocimiento De Voz	59
5.3.6.2.	Programación del movimiento de los servomotores (Código IDE)	64
6.	CONCLUSIONES.....	66
7.	REFERENCIAS	67
7.1.	BIBLIOGRAFÍA	67
8.	ANEXOS	71
8.1.	Circuitos.....	71
8.2.	Servomotores.....	73
8.3.	Módulos.....	73

INDICE DE FIGURAS

Figura 1 El Flautista y El Pato Creado por Jacques Vaucason.	6
Figura 2 Muñeca Mecanica Creado por Henri Maillardet.	7
Figura 3 R.U.R (Rossum´s Universal Robots) Obra Creada por Karel Capek.	8
Figura 4 Evolución De Los Robots.	9
Figura 5 Robot De Servicio.....	10
Figura 6 Maggie	11
Figura 7 Partner Robot.....	11
Figura 8 K-LU-K.....	12
Figura 9 Hensar	12
Figura 10 Ars Manga	14
Figura 11 Modulo De Reconocimiento De Voz EasyVr. 3	21
Figura 12 Easy VR 3 Como Placa de Desarrollo Fuente: (VeeR, 2016)	23
Figura 13 Administrador de Dispositivos Fuente: Propia.....	24
Figura 14 Easy VR 3 Especificaciones Técnicas	25
Figura 15 Protocolo de comunicación.....	27
Figura 16 Arduino UNO	28
Figura 17 Arduino Nano	30
Figura 18 Servomotor MG996R	33
Figura 19 Arduino IDE Ventana Principal.....	35
Figura 20 EasyVR Commander Ventana Principal.....	36
Figura 21 Sensory Quick Synthesis 5 Ventana Principal.....	37
Figura 22 Adobe Audition Ventana Principal.....	37
Figura 23 Ventana Principal Proteus.....	39
Figura 24 Ventana Paquete de Dispositivos.....	39

Figura 25 Ventana Principal De Ares	40
Figura 26 Easy VR 3	41
Figura 27 Easy VR Shield 3.....	42
Figura 28 Easy Vr Shield 3 Especificaciones Técnicas	44
Figura 29 Conexión Servomotor Arduino.....	47
Figura 30 Librería Servo	47
Figura 31 Nuevo Documento Adobe Audition	49
Figura 32 Formato de Audio	49
Figura 33 Ventana de Grabación.....	50
Figura 34 Nuevo Documento QuickSyntesis	51
Figura 35 Carga de Tabla de Sonidos	52
Figura 36 Easy Commander Carga Tabla De sonidos	53
Figura 37 Selección de archivo de QuickSyntesis	54
Figura 38 Tabla de Sonidos.....	55
Figura 39 Modo PC Easy VR Shield 3	55
Figura 40 Add Command.....	56
Figura 41 Train Command.....	56
Figura 42 Group 1 Configuración de Comandos de Voz.....	57
Figura 43 Generación de Código IDE.....	58
Figura 44 Código de Easy Commander en Arduino IDE	58

ÍNDICE DE TABLAS

Tabla 1 Acontecimientos destacados (Impresora 3D)	20
Tabla 2 Características Técnicas Easy VR 3.....	26
Tabla 3 Características Técnicas Arduino UNO	29
Tabla 4 Características Técnicas Arduino Nano	30
Tabla 5 Características Servomotor MG996R	34
Tabla 6 Características Técnicas Easy VR Shield 3	44

ÍNDICE DE ANEXOS

Anexo 1 Proteus 8 Ventana Principal Circuito Eléctrico Fuente Variable 12V	71
Anexo 2 Proteus 8 ISIS Ventana De Diseño Circuito Impreso Fuente Variable 12V	71
Anexo 3 Circuito Impreso	72
Anexo 4 Proteus 8 Ventana Principal Circuito Eléctrico Amplificador de 20W	72
Anexo 5 Proteus 8 ISIS Ventana Principal Circuito Eléctrico Amplificador de 20W	72
Anexo 6 Circuito Impreso Amplificador de 20W	73
Anexo 7 Cabeza Robótica Servomotores	73
Anexo 8 Arduino y Easy VR Shield 3	73
Anexo 9 Ensamblaje de Módulos	74
Anexo 10 Cabeza Robótica	74

1. INTRODUCCIÓN

Con el paso del tiempo el hombre ha intentado explicar y entender cómo pensamos, puesto que se encuentra inmerso en una de las revoluciones más grandes y trascendentes de la historia, llamado revolución tecnológica, donde la ciencia de la robótica combina ampliamente tareas, como la mecánica, la electrónica, la informática y la inteligencia artificial, por lo cual la interacción entre el humano y el robot, es una de las características más importantes y una de las tareas más difíciles de alcanzar, debido a que la mente humana no reacciona emocionalmente frente a objetos artificiales.

En la actualidad hablar de inteligencia artificial es adentrarse en un mundo de información finita que requiere de datos pre-programados o pre-concebidos, que recrean reacciones a estímulos de los robots de hoy en día, de tal forma que alrededor de todo el mundo, obtienen diversas disciplinas, conocimientos teóricos y prácticos, que permiten automatizar sistemas basados en estructuras mecánicas, dotados con un determinado grado de inteligencia, que están destinados a la producción industrial.

En el presente proyecto de aplicación práctica se desarrolla un sistema de reconocimiento de comandos de voz, utilizando el módulo Easy VR 3, por ende, se efectuará el sistema electrónico y la programación en diversos Softwares, por lo tanto, permitirá a una cabeza robótica interactuar, dentro del Instituto Superior Tecnológico Vida Nueva, en el cual se desarrollará la comunicación y aprendizaje práctico, así fomentando el desarrollo de nuevas tecnologías basándose en la ciencia de la robótica. Finalmente, por todo lo mencionado anteriormente el presente trabajo de investigación práctica podrá plantear y distinguir con claridad prácticas didácticas para los estudiantes de la institución.

2. ANTECEDENTES

Origen de la palabra robot, procede de la palabra checa robota, que significa 'trabajo obligatorio' fue empleado por primera vez en la obra teatral R.U.R. (Robots Universales de Rossum), estrenada en enero de 1921 en Praga por el novelista y dramaturgo checo Karel Capek. La obra fue un éxito inmediato y pronto se estrenó en multitud de teatros por toda Europa y Estados Unidos. En ella, el gerente de una fábrica construía unos seres al absoluto servicio del hombre, que realizaban todas las tareas físicas mientras los humanos se dedicaban al ocio permanente. Cuando el gerente de la fábrica decide construir robots más perfectos que experimentaran felicidad y dolor, hasta que finalmente los robots se rebelan contra los hombres y destruyen al ser humano, a excepción de sus creadores. (CETI, 2015)

El mundo de la robótica parecía ser un mundo de ciencia ficción hace pocos años, sin embargo, los desarrollos tecnológicos han avanzado rápidamente. En unos pocos años los robots se involucraran en nuestras vidas cotidianas como asistentes médicos y asistentes educativos, recepcionistas y mascotas. (Sandoval, 2012)

La robótica es una de las ramas de la tecnología que ha dado pasos cada día más grandes, tanto como a nivel industrial o al servicio de la comunidad. En la actualidad existen ya robots asistentes, algunos todavía en estudio o que están en desarrollo. La inteligencia artificial es un proceso continuo en la robótica donde se introduce indudablemente la educación como área estratégica para la humanidad, por lo cual, a lo largo de la vida, el individuo va aprendiendo constantemente, en diferentes formas y al mismo tiempo va enseñando y compartiendo la experiencia adquirida por procesos de retroalimentación dirigido al campo tecnológico.

(Anonimo, 2018) afirma: “El reconocimiento de voz es un tipo de inteligencia artificial que trata de establecer una comunicación entre el hombre y los ordenadores o dispositivos inteligentes, a través del lenguaje humano.” Actualmente se están dando sólidos pasos en esta dirección, muchos laboratorios alrededor del mundo llevan a cabo investigación interdisciplinaria para desarrollar robots que sean útiles y seguros en la vida de las personas tanto al servicio comunitario o industrial.

3. OBJETIVOS

3.1. Objetivo General

Desarrollar un sistema electrónico que permita el reconocimiento de comandos de voz de una cabeza robótica interactiva.

3.2. Objetivos específicos

- Investigar el proceso de reconocimiento de voz en el módulo Easy VR 3.
- Desarrollar el sistema de reconocimiento de voz que permita el correcto funcionamiento de la cabeza robótica.
- Ensamblar el módulo de reconocimiento de voz en la cabeza robótica.
- Realizar pruebas de funcionamiento para descartar errores y corregir la programación.

4. DESARROLLO

4.1. Marco Teórico – Conceptual

4.2. La Robótica

La robótica como hoy en día la conocemos, sus orígenes se basan hace miles de años, en este caso se introducirá en hechos registrados a través de la historia, fundamentalmente se aclarará que antiguamente los robots eran conocidos con el nombre de autómatas, y que la robótica no era reconocida como ciencia, cabe destacar que la palabra robot surgió hace mucho después del origen de los autómatas. (Rivera, 2016) afirma “El termino robótica (robotics), se le atribuye al escritor americano de origen ruso Isaac Asimov (1920 - 1992) y fue utilizada por primera vez en una historia corta titulada Runaround publicada en 1942”. Desde el principio de los tiempos, el hombre ha deseado crear vida artificial y se ha empeñado en dar vida a seres artificiales que se incluyan en su morada, seres que realicen sus tareas repetitivas, tareas pesadas o difíciles de efectuar por la capacidad de un ser humano. (Molina, 2015)

“La robótica es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots.” (Dimate, 2014)

(Gonzales, 2014), señala que la robótica combina disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física. Otras áreas sustancialmente importantes en robótica son el álgebra, los autómatas programables, la animatrónica y las máquinas de estados.

4.3. Historia de la robótica

Por siglos el ser humano ha construido máquinas que imiten las partes del cuerpo humano. Los antiguos egipcios unieron brazos mecánicos a las estatuas de sus dioses. Estos brazos

fueron operados por sacerdotes, quienes clamaban que el movimiento de estos era inspiración de sus dioses. Los griegos construyeron estatuas que operaban con sistemas hidráulicas, los cuales se utilizaban para fascinar a los adoradores de los templos.

Durante los siglos XVII y XVIII en Europa fueron construidos muñecos mecánicos muy ingeniosos que tenían algunas características de robots.

Jacques de Vaucanson construyó varios músicos de tamaño humano a mediados del siglo XVIII. Esencialmente se trataba de robots mecánicos diseñados para un propósito específico: la diversión. (Figura 1)



Figura 1 El Flautista y El Pato Creado por Jacques Vaucason.

Fuente: (Emaze, 2011)

En 1805, Henri Maillardert construyó una muñeca mecánica que era capaz de hacer dibujos (Figura 2). Una serie de levas se utilizaban como “El Programa” para el dispositivo en el proceso de escribir y dibujar. Éstas creaciones mecánicas de forma humana deben considerarse como inversiones aisladas que reflejan el genio de hombres que se anticiparon a su época.



Figura 2 Muñeca Mecanica Creado por Henri Maillardet.

Fuente: (Sanchez, 2007)

Hubo otras invenciones mecánicas durante la revolución industrial, creadas por mentes de igual genio, muchas de las cuales estaban dirigidas al sector de la producción textil. Entre ellas se puede citar la hiladora giratoria de Hargreaves (1770), la hiladora mecánica de Crompton (1779), el telar mecánico de Cartwright (1785), el telar de Jacquard (1801), y otros.

El desarrollo en la tecnología, donde se incluyen las poderosas computadoras electrónicas, los actuadores de control retroalimentados, transmisión de potencia a través de engranes, y la tecnología en sensores han contribuido a flexibilizar los mecanismos autómatas para desempeñar tareas dentro de la industria. Son varios los factores que intervienen para que se desarrollaran los primeros robots en la década de los 50's. La investigación en inteligencia artificial desarrolló maneras de emular el procesamiento de información humana con computadoras electrónicas e inventó una variedad de mecanismos para probar sus teorías.

No obstante, las limitaciones de las máquinas robóticas actuales, el concepto popular de un robot es que tiene una apariencia humana y que actúa como tal. Este concepto humanoide ha sido inspirado y estimulado por varias narraciones de ciencia ficción.

Una obra checoslovaca publicada en 1921 por Karel Capek, denominada Rossum's Universal Robots (Figura 3), dio lugar al término robot. La palabra checa "Robota" significa servidumbre o trabajador forzado, y cuando se tradujo al inglés se convirtió en el término robot. (Molina, 2015)



Figura 3 R.U.R (Rossum's Universal Robots) Obra Creada por Karel Capek.

Fuente: (Billy, 2011)

4.4. Robot

Cuando se intenta definir el termino robot normalmente se lo define como una mquina inteligente, que detecta o se mueve a su alrededor, con un comportamiento humanoide, como se especificó anteriormente en la historia de la robótica, la palabra robot proviene de la palabra checa "Robota" que se refiere al trabajo realizado de manera forzada, el uso de la palabra robot se lo debemos al escritor checo Karel Capek (1890 - 1938).

(Ocaña, 2015) señala que una definición más exacta del termino robot podría ser, Autómata programable capaz de realizar determinadas operaciones de manera autónoma, en especial tareas repetitivas, precisas o peligrosas. (pag.2)

4.4.1. Definiciones de Robot:

- Es una máquina.
- Se ha fabricado para funcionar de manera automática. Es decir, por si sola, sin intervención humana.
- Su comportamiento obedece a un programa informático. Por lo tanto, cambiando el programa se puede modificar su funcionamiento.
- Las aplicaciones más importantes de los robots son hacer tareas que se realicen una y otra vez (repetitiva), o en las que es necesario no cometer ningún fallo (precisas), o en las que se corre riesgo para la integridad física de quien las realiza (peligrosas).

Determinar el termino robot es un tema de controversia por la razón que influye la industria, el cine, la televisión, internet y la literatura de ciencia ficción entre otras cosas, posteriormente comparando las definiciones anteriores se muestra los distintos tipos de robot actuales en la figura 4 y figura 5.

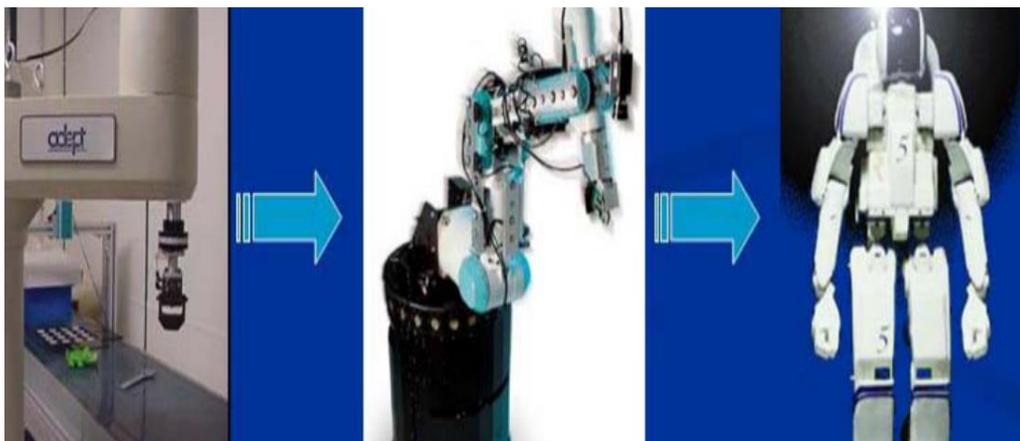


Figura 4 Evolución De Los Robots.

Fuente: (Sanz, 2006)



Figura 5 Robot De Servicio

Fuente: (Sanz, 2006)

4.5. Robot social

La interacción es la principal diferencia entre un robot industrial y un robot social, de tal manera que la comunicación interactiva entre el humano y el robot es una de las características más importantes y tiene un grado de dificultad para poderla desarrollar, debido a que la mente humana no reacciona emocionalmente frente a objetos artificiales. (Takayuki, K. y Hiroshi, I., 2012)

Según el sitio web MEGATENDENCIAS (Moriello, 2008), define que un robot social “es aquel que interactúa y se comunica con las personas (de forma sencilla y agradable) siguiendo comportamientos, patrones y normas sociales”.

(Mejia, M & Nuñez, J, 2016) señalan que, a finales de 1990, en el Instituto Tecnológico de Massachusetts se desarrolló un robot antropomórfico expresivo llamado Kismet, criatura robótica expresiva con modalidades perceptivas y motoras adaptadas a los canales naturales de comunicación humana, posteriormente entre los robots sociales más importantes y destacados se pueden mencionar a:

- **Maggie**, un robot desarrollado por el Robotics Lab de la Universidad Carlos III Madrid en España, diseñado para estudiar la interacción Humano – Robot, la inteligencia de tal robot es capaz de reconocer voz, caras y sentir cuando alguien lo toca. (Figura 6)



Figura 6 Maggie

Fuente: (Universidad Carlos III de Madrid, 2019)

- **Partner Robot**, fabricado por Toyota, que está habilitado con la amabilidad e inteligencia para ayudar a las personas, desde la atención médica y en el hogar. (Figura7)



Figura 7 Partner Robot

Fuente: (Toyota, 2015)

Cabe considerar que, en el Ecuador, retrospectivamente se han desarrollado trabajos relacionados con la robótica social, tales como:

- **K-Lu-K**, un rostro robótico de apariencia humana que ejecuta gestos y emociones de tal forma que un hombre común. (Figura 8)



Figura 8 K-LU-K

Fuente: (Ruales, 2015)

- **Hensar**, robot capaz de simular las emociones expresadas por un rostro humano, el cual cuenta con distintas presentaciones como reconocimiento de voz, comunicación inalámbrica, reproducción de formato MP3, que en su combinación brinda un amplio rango de movimientos y realismo. (Figura 9)



Figura 9 Hensar

Fuente: (Astudillo, López, & Sotomayor, 2014)

4.6. Inteligencia artificial

Definir el concepto de Inteligencia Artificial (IA) no es nada fácil, sumamente en general sería el estudio de la informática centrándose en el desarrollo de software o una maquina con inteligencia humana, en donde se incluye la deducción, el razonamiento lógico, la capacidad de manipular y mover objetos, el objetivo a largo plazo es lograr la creatividad, la inteligencia social y la inteligencia general (a nivel Humano).

(Garcia, 2012) argumenta que la Inteligencia Artificial es quizás unas de las disciplinas que más despiertan la imaginación de los que oyen hablar de ella por primera vez. Tal vez debido a que la ciencia ficción se ha encargado de crear ese halo de misterio que la envuelve e invita a soñar con máquinas capaces de razonar, como el maravillo HAL 900 (Heuristically Programmed Algorithmic Computer) computador algorítmico programado heurísticamente, imaginado por Kubrick.

Aunque a veces no sea demasiado evidente, la Inteligencia Artificial (IA) está cada vez más en nuestro día a día, por ejemplo, cada vez que usamos un celular para tomar fotografías o gravar un video, un algoritmo de IA está identificando la imagen para poder ser enfocarla. Por extraño que pueda parecer, lo cierto es que no hay consenso entre los científicos e ingenieros sobre lo que es la Inteligencia Artificial, y mucho menos a una definición exacta y concisa que permite dirimir, que programas son o no inteligentes.

En la actualidad la Inteligencia Artificial (IA) es una de las tecnologías más importantes para el 2019, según Gartner Inc. (empresa dedicada a la consultoría e investigación de las tecnologías de la información), de todos modos la IA es una área de la ciencia de la computación que tiene como objetivo desarrollar los fundamento teóricos y prácticos para el desarrollo de sistemas de computación que presentan características inteligentes, al mismo

tiempo muchos problemas que ocurren en la industria, servicios y entretenimiento corresponden a problemas inteligentes, y su solución se hace cada vez más indispensable.

“Entre los diversos tipos de sistemas inteligentes se puede señalar a los juegos hombre-máquina, los sistemas basados en el conocimiento, los sistemas de optimización y machine-learning (aprendizaje automático)”. (WordPress, 2017)

De acuerdo con el libro sobre la Inteligencia Artificial (Dr. Ponce, 2010), señala que primero se debe plantear la siguiente pregunta “¿Qué es la inteligencia artificial?” sin duda, se trata de una pregunta difícil cuya respuesta aún no ha sido resuelta totalmente, la cual sigue desconcertando tanto a los biólogos como a los psicólogos y filósofos de nuestra época. Se podría considerar que unos de los primeros pasos hacia la IA fueron dados hace mucho tiempo por Aristóteles (348-322 a.C.), cuando se dispuso a explicar y codificar ciertos tipos de razonamiento deductivo que él llamó silogismos. Otro intento sería el de Ramón Llull (d.C 1235-1316), místico y poeta catalán, quien construyó un conjunto de ruedas llamado Ars Magna, el cual se suponía que iba a ser una máquina capaz de responder todas las preguntas.

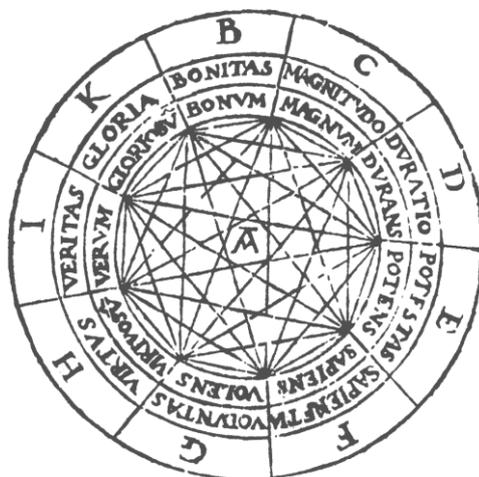


Figura 10 Ars Manga

Fuente: (Ramon, 2012)

Formalmente la IA se inicia en 1956 cuando se acuñó el término, este estudio es una de las disciplinas más antiguas, desde hace más de 2000 años, los filósofos se han esforzado por comprender como se ve, se aprende, se recuerda y se razona, así como la manera en que esas actividades deberían realizarse, de cualquier manera se podría comenzar por destacar algunas propiedades generales que presenta la inteligencia humana, como por ejemplo la habilidad de enfrentar nuevas situaciones, la habilidad de resolver problemas, de responder preguntas o elaborar planes. Desde sus inicios, el hombre representó el mundo real mediante símbolos, los cuales constituyen la base del lenguaje humano, aunque en otro sentido se considera que los procesos simbólicos son una característica esencial de la IA.

Finalmente, a partir de lo más fundamental o importante expuesto anteriormente es posible formular una definición más aproximada al objetivo del estudio planteado:

La Inteligencia Artificial es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de contextos informáticos que implican la deducción, el razonamiento, la presentación del conocimiento, la planificación, o en tal caso la resolución de problemas.

Los temas fundamentales de la Inteligencia Artificial, se componen de varias áreas de estudio, las más comunes e importantes son:

- Búsqueda de soluciones
- Sistemas expertos
- Procesamientos de lenguaje natural

- Robótica
- Aprendizaje de las maquinas
- Lógica
- Incertidumbre y “lógica difusa”

4.7. Reconocimiento por Voz

Desde hace mucho tiempo la ciencia ficción ha presentado diferentes películas y series, donde se muestra al hombre y maquina desarrollando diferentes actividades que superan las capacidades extremas de un ser humano, algunos de los escenarios planteados resultan ser acertados, pero en otras ocasiones están muy lejos de volverse realidad, lo que sí es un hecho es que algunas Inteligencias Artificiales tales como los asistentes de voz que compiten, para convertirse en la mejor IA del reconocimiento de voz, así como:

- **Siri:** el asistente inteligente, funciona a base del procesamiento de lenguaje natural.
- **Amazon Echo:** Asisten te voz que te ayuda a reproducir música, realizar llamadas o en tal caso configurar alarmas y recordatorios.
- **Cortana:** asistente virtual de Microsoft cuyo objetivo es facilitarte recordatorios con momentos, lugares o personas.

El reconocimiento de voz es una tecnología biométrica que utiliza comandos de voz de un individuo para lograr su identificación, según (Barrios, Lopez, Mendieta, Benavides, & Yessica, 2018) argumenta que, un sistema de reconocimiento de voz no es más que la capacidad que posee un ordenador, de convertir las palabras de la voz humana a un código binario comprensible por la computadora. Los primeros sistemas creados estaban basados en conversaciones de tipo discretas, en las cuales se utilizaba un lenguaje puntuado por pausas y con un límite de palabras que no excedía de las 50. Años después se comienza a trabajar e un sistema de conversaciones continuas que no requería que el usuario realizara pausa entre las palabras, esto debido a que se emplearon técnicas para minimizar la diferencia en la velocidad del habla.

En un principio su utilidad fue destinada al sector de la salud, cuya funcionera la de reemplazar las tradicionales transcripciones médicas, además de este sector, otros sectores también se beneficiaron con dicha tecnología, entre los cuales se puede mencionar en el sector de comercio, artículos como computadoras, celulares y automóviles inteligentes o en otro caso en los hogares (domótica).

Un sistema de reconocimiento de voz es el encargado de establecer la comunicación entre los humanos y las computadoras, con el desarrollo de que las aplicaciones sean capaces de reconocer diferentes voces en la medida de que el habla se de manera natural. Aunque se obtiene muchas dificultades, ya que no se cuenta con un sistema que resuelva los problemas existentes relacionados con la variabilidad de las características de la señal acústica, no obstante, pueden ser dependientes o independientes del locutor en el caso de que sea:

- **Dependiente:** la probabilidad de un buen reconocimiento de la voz es alta, ya que las muestras que va a tener el reconocedor solo pertenecerá a una persona en particular facilitando que se reconozca el vocabulario
- **Independiente:** sea que es un sistema que puede reconocer la voz de cualquier persona, existe cierta dificultad debido a que las representaciones paramétricas de la voz dependen altamente del locutor.

4.7.1. Problemas Del Análisis Y Reconocimiento De Voz

Cuando el locutor habla lentamente, las palabras son más aisladas una de otras y esto da como resultado que la probabilidad de reconocimiento del sistema sea mayor, en cambio cuando el locutor habla de forma natural, realizando las pausas correspondientes que cada frase requiere, el grado de dificultad aumenta, ya que se hace difícil identificar donde inicia y donde termina una palabra, inclusive si están muy juntas cabe la posibilidad de que se confundan con una sola.

La extensión del vocabulario es importante, ya que en la medida de que vaya creciendo, así también va aumentado la dificultad de reconocimiento, o bien surgen nuevos problemas como lo es cuando se confunde una palabra por otra o cuando el sistema tarda más en reconocer las palabras, no obstante también se presenta la similitud entre palabras puesto que si tenemos un vocabulario extenso puede haber palabras que se parezcan aumentando la dificultad de reconocimiento, al mismo tiempo que en la búsqueda realizada para encontrar la palabra correcta tiene mayor probabilidad de que sea errónea.

En cuanto al ruido que es un factor que afecta en gran medida el nivel de desempeño del reconocedor, puede ser producido por el ambiente, por parte del locutor o inclusive música, a

la vez que también varía la eficiencia de acuerdo al estado de ánimo del locutor, la calidad del micrófono, entre otros factores tales como:

- La correcta elección y extracción de las características de la señal de voz.
- Tratar con corrección las variaciones inherentes a género.
- Velocidad de emisión, pronunciación y acentos.
- Tamaños de los vocabularios a reconocer.
- Ruido y distorsión de los entornos donde se utilizan.
- Inclusive hasta el estado de ánimo del locutor.

Pese a las dificultades se puede argumentar, que se ha logrado identificar por múltiples investigaciones y desarrollos de sistemas de uso real en la vida cotidiana de las personas, a una exactitud superior al 90%, siempre y cuando se considere analíticamente o lógicamente, la acotación de una o de otra manera.

4.8. Hardware

En términos generales se puede señalar que en la época actual el robot, está conformado por partes tangibles, es decir que agrupa todo un conjunto de elementos electrónicos, como fragmentos o partes fundamentales para el funcionamiento del robot, entre los cuales se destacan:

- Impresiones 3D
- Elementos de entrada y salida de datos (Tarjeta de Control)
- Componentes que suministrara o regulan la energía del sistema.
- Actuadores electrónicos que proporcionara movimiento a las articulaciones.

4.8.1. Impresión 3D

(López, 2016), argumenta que el inicio de la impresión 3D se remonta a 1976, en el momento cuando se inventó la impresora de inyección de tinta. A lo largo de los años, la tecnología ha evolucionado radicalmente sufriendo cambios en diferentes ramas de la industria, como se puede señalar en la siguiente Tabla 1:

*Tabla 1
Acontecimientos destacados (Impresora 3D)*

Año	Acontecimiento
1984	Charles Hull inventa la estereolitografía (SLA)
1989	Se concede la patente del SLS (Sinterización selectiva por láser). Scott Crump funda Stratasys. Hans Langer funda EOS GmbH
1992	Stratasys patenta su tecnología de FDM (Modelado por Deposición de Material Fundido)
1999	El Instituto de Medicina Regenerativa de la Universidad de Wave Forrest implanta en humanos los primeros órganos modificados por medio de implantes arteriales impresos en 3D y cubiertos con células del paciente.
2002	El Instituto de Medicina Regenerativa de la Universidad de Wave Forrest se imprime el primer órgano en 3D: un riñón completamente funcional.
2006	Se construye la primera máquina del tipo SLS. Básicamente, utiliza un láser para fundir materiales en el proceso de impresión 3D. Personalización masiva de fabricación de piezas industriales y más tarde, prótesis.

2007	Sale al mercado el primer sistema de 3D Systems por menos de 10.000 \$.
2008	Primera persona que camina sobre una pierna de prótesis impresa en 3D.
2009	Industrias MakerBot, compañía de Hardware de código abierto para las impresoras 3D comercializa en el mercado la primera impresora 3D en forma de kit.
2012	LayerWise realiza el primer implante de mandíbula inferior en 3D
2014	Engineering Co, difunde su sistema de impresión 3D “Atlas”, capaz de construir casas, Amazon abre su tienda de impresión 3D ofreciendo una amplia gama de productos.

La impresión 3D, o manufactura aditiva, es el grupo de tecnologías de fabricación con la posibilidad de imprimir, con una variedad técnicas, objetos solidos tridimensionales, esta tecnología implementa la fabricación e investigación de implantes médicos, piezas de arquitectura y construcción de prototipos en menor o mayor escala.

4.8.2. Módulo EasyVr 3.0

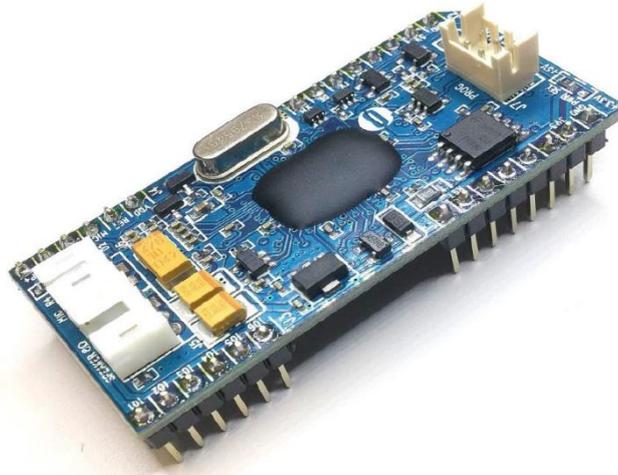


Figura 11 Modulo De Reconocimiento De Voz EasyVr. 3

Fuente: (VeeR, 2016)

Easy VR 3 es un módulo de reconocimiento de voz multipropósito diseñado para agregar fácilmente versátil, robusto y rentable. El modulo se puede usar con cualquier Host con un

UART interfaz alimentada a 3.3 V – 5 V, PIC y placas Arduino, algunos ejemplos de aplicaciones incluyen domótica, como interruptores de luz controlados por voz, cerraduras, cortinas o cocinas electrodomésticos, o agregar “audición” a los robots más populares en el mercador.

4.8.2.1. Características del Easy VR 3

- Hasta 64 (256 en Easy VR 3+) comandos dependientes del altavoz (SD – Speaker Dependet) o verificación del altavoz (SV – Speaker Verification) definidos por el usuario, que se pueden entrenar en cualquier idioma, dividido en un maximo de 16 grupos (hasta 32 comandos SD 0 5 SV cada uno).
- Tecnologia SonicNet para comunicaciones inalámbricas entre módulos o cualquier otra fuente de sonido (CD de Audio, DVD, Reproductor de MP3).
- Hasta aproximadamente (137 en Easy VR 3+) segundos de grabación y reproducción de mensajes en vivo.
- Capacidad de sincronización de labios en tiempo real.
- Salida de audio diferencial que admite directamente altavoces de 8 Ohms.
- Interfaz gráfica de usuario fácil de usar para programar comandos de voz y audio.

Idiomas que soporta el modulo:

- Ingles americano

- Inglés británico
- Francés
- Alemán
- Italiano
- Japonés
- Mandarín
- Español

4.8.2.2. *Easy VR 3 Como Placa de Desarrollo*

El módulo Easy VR 3 ha sido diseñado para permitir su uso como placa de desarrollo independiente cuando se combina con USB-Adaptador en serie

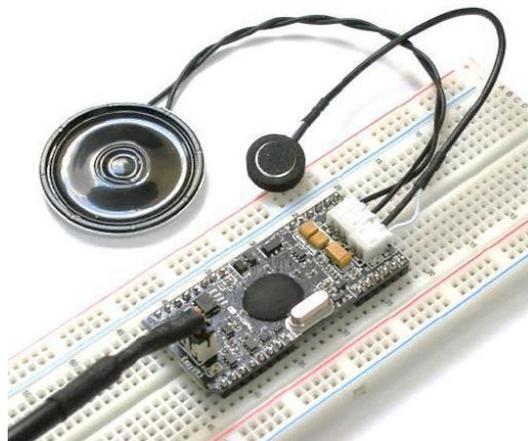


Figura 12 Easy VR 3 Como Placa de Desarrollo
Fuente: (VeeAR, 2016)

El cable QuickUSB se puede usar para programar comandos los voz y salidas de sonido directamente al módulo Easy VR3, probando rápidamente sus funciones desde la PC. Simplemente conectando el micrófono y un altavoz de 8 ohm al módulo y posteriormente adaptarle el cable QuickUSB.

4.8.2.3. *Procedimiento Independiente del Módulo Easy VR 3*

1. Conecte el micrófono a la toma de 2 pines MIC (J6)
2. Conectar el altavoz de 8 ohm a la toma de 3 pines SPEAKER (J5)
3. Conectar el cable QuickUSB a la toma de 3X2 pines (J7)
4. Conectar el extremo del USB al PC
5. Si la instalación es exitosa, se observará un puerto COM virtual en su administrador de dispositivos. (Figura 13)

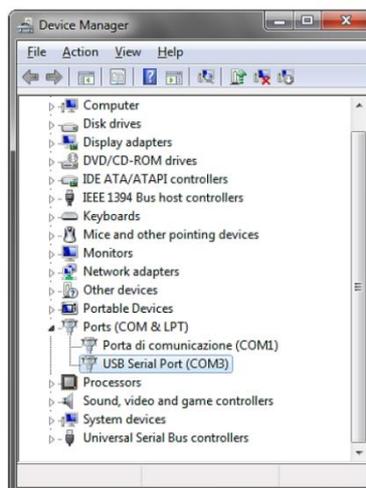


Figura 13 Administrador de Dispositivos
Fuente: Propia

6. Iniciar el Software Easy VR Commmander y elegir el puerto COM, clic en conectar.

4.8.2.4. Especificaciones Técnicas

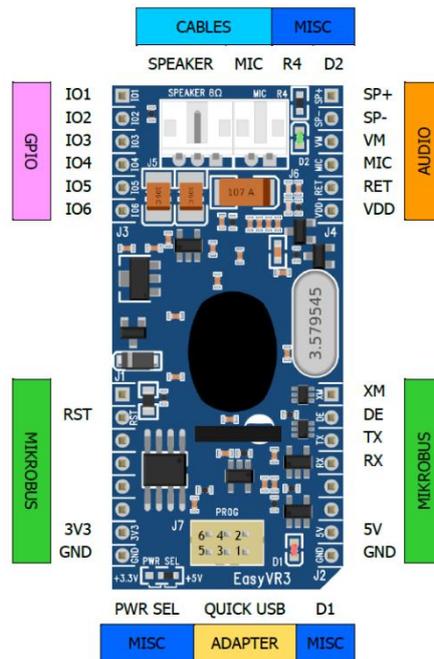


Figura 14 Easy VR 3 Especificaciones Técnicas

Fuente: (VeeR, 2016)

Los encabezados J1 y J2 son los conectores de interfaz mikroBus, que proporcionan seleccionable de 3.3 V / 5 V, al módulo y líneas de I/O digitales traducidas por voltaje, que incluyen: líneas de recepción / transmisión UART y pines de control.

Los encabezados J3 proporciona líneas de expansión de I/O configurables (entradas con pull-up interno débil por defecto), alimentadas con el voltaje lógico interno VDD.

El Encabezado J4 contiene las principales señales analógicas, como señales de micrófono y salidas DAC amplificadas, que también están disponibles en los conectores internos de ángulo recto J5 y J6.

El modulo se puede operar a través del conector de programación J7 solo Utilizando el cable adaptador QuickUSB.

Tabla 2
Características Técnicas Easy VR 3

Grupo	Nombre	Número	Pin	Tipo	Descripción	
• MIKROBUS	J1	1	-	-	(No Conectado)	
		2	RST	I	Reinicio activo bajo asincrónico (pull-up interno)	
		3-6	-	-	(No Conectado)	
		7	3V3	I	Entrada de alimentación de 3.3 V	
		8	GND	-	Ground	
	J2	1	XM	I	Selección de arranque (pull-down)	
		2	DE	O	(Reservado)	
		3	TX	O	Transmisión de datos en serie	
		4	RX	I	Recibir datos en serie	
		5-6	-	-	(No Conectado)	
		7	5V	I	Entrada de alimentación 5 V	
		8	GND	-	Ground	
	• GPIO	J3	1	IO1	I/O	I/O de propósito general (niveles lógicos VDD)
			2	IO2	I/O	I/O de propósito general (niveles lógicos VDD)
3			IO3	I/O	I/O de propósito general (niveles lógicos VDD)	
4			IO4	I/O	I/O de propósito general (niveles lógicos VDD)	
5			IO5	I/O	I/O de propósito general (niveles lógicos VDD)	
6			IO6	I/O	I/O de propósito general (niveles lógicos VDD)	
• AUDIO	J4	1	SP+	O	Salida de audio deferencial (puede manejar directamente un altavoz de 8 Ohm)	
		2	SP-	O		
		3	VM	O	Potencia del micrófono (para admitir micrófonos personalizados)	
		4	MIC	I	Entrada de audio de micrófono	
		5	RET	-	Retorno de micrófono (analog ground)	
		6	VDD	O	Tensión lógica interna (solo para referencia)	
• CABLES	J5	1	SP-	O	Salida de audio deferencial (puede manejar directamente un altavoz de 8 Ohm)	
		2	SP+	O		
		3	-	-	(No conectado)	
	J6	1	MIC	I	Entrada de audio de micrófono	
		2	RET	-	Retorno de micrófono (analog ground)	
• ADAPTADOR	J7	1	RX_P	O	Programming cable serial data receive	
		2	RTS_P	I	Programming cable request to send (reset/boot control)	
		3	GND	-	Programming cable ground	
		4	5V_P	I	Programming cable 5V DC power output	
		5	TX_P	I	Programming cable serial data transmit	
		6	CTS_P	O	Programming cable clear to send (tied to ground)	

4.8.2.5. *Interfaz de Comunicación*

Easy VR 3 se comunica a través de una interfaz serie asíncrona (comúnmente conocida como interfaz UART), con las siguientes características:

- **Velocidad de transmisión:** 9600 (predeterminado), 19200, 38700, 57600, 115200
- **Marco:** 8 Bits de datos, No parity, 1 bit de parada

La línea de datos de entrada del receptor es RX, mientras que la línea de datos de salida de transmisor es TX. No se utilizan líneas de apretón de manos.

4.8.2.6. *Protocolo de comunicación*

La comunicación con el módulo Easy VR utiliza una interfaz UART estándar compatible con niveles lógicos TTL / CMOS 3.3 – 5V, de acuerdo con el voltaje de alimentación VCC.

Una conexión típica a un HOST basado en MCU:

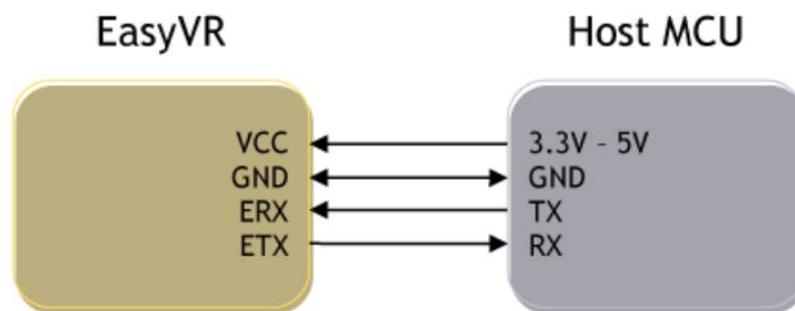


Figura 15 Protocolo de comunicación

Fuente: (VeeR, 2016)

La configuración inicial en el encendido es 9600 baudios, datos de 8 bits, sin parity, parada de 1 bit. La velocidad en baudios se puede cambiar más tarde para operar en el rango de 9600 a 115200 baudios.

El protocolo de comunicación utiliza caracteres ASCII imprimibles, que se pueden dividir en dos grupos principales:

Comando y caracteres de estado, respectivamente en las líneas TX y RX, elegidos entre letras minúsculas.

Argumentos de comando o detalles de estado, nuevamente en líneas TX y RX, que abarcan el rango de letras mayúsculas.

4.8.3. Arduino Uno



Figura 16 Arduino UNO

Fuente: Propia

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un

resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador. (Jadiaz, 2016)

4.8.3.1. Especificaciones Técnicas

Tabla 3
Características Técnicas Arduino UNO

Microcontrolador	Atmega328
Voltaje	5V
Voltaje Entrada	7 - 12V
Voltaje Salida	6 - 20V
Digital I/O Pins	14 (de los cuales 6 son salidas PWM)
PWM Digital I/O Pins	6
Entradas Analógicas	6
DC Current per I/O Pin	20 mA
DC Current For 3.3V Pin	50 mA
Flash Memory	32 KB (Atmega328) de los cuales 0.5 KB son utilizados para el arranque.
SRAM	2 KB (Atmega328)
EEPROM	1 KB (Atmega328)
Frecuencia del reloj	16 MHZ
Longitud	68.6 mm
Ancho	53.4 mm
Peso	25 g

4.8.4. Arduino Nano

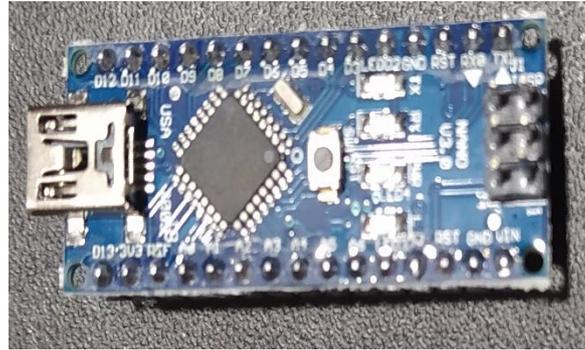


Figura 17 Arduino Nano

Fuente: Propia

Arduino Nano es una placa pequeña, completa y amigable que está basada en el Atmega328. tiene la misma funcionalidad que el Arduino Uno, pero mucho más compacto y con una presentación diferente. No posee conector para la alimentación externa, y funciona con un cable USB Mini-B (incluido) del cable estándar.

4.8.4.1. Especificaciones Técnicas

Tabla 4
Características Técnicas Arduino Nano

Microcontrolador	Atmega328
Arquitectura	AVR
Tensión de funcionamiento	5 V
Memoria Flash	32 KB de los cuales 2KB utilizados por el gestor de arranque
SRAM	2 KB
Velocidad de reloj	16 MHz
Pines IN analógicos	8
EEPROM	1 KB
Corriente CC por pines de E / S	40 mA (pines de E / S)
Voltaje de Entrada	7 – 12 V

Pines de E / S	22 (6 de los cuales son PWM)
Salida PWM	6
El consumo de energía	19 mA
Tamaño de PCB	18 x 45 mm
Peso	7 g

4.8.5. Servomotores

Los servomotores son dispositivos similares a un motor de corriente continua, que son capaces de ser controlados, tanto en velocidad como en su posición del eje principal dentro de un rango de operación y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control que pueden operarse mediante una señal codificada, tienen un peso reducido, además fiabilidad, estabilidad y versatilidad a baja tensión.

En proyectos o prácticas de la vida cotidiana, se usan servomotores para posicionar superficies de control como movimiento de palancas, radio control, títeres o robots.

4.8.5.1. Estructura interna

- **Motor de corriente continua**

Es el elemento que brinda movilidad al servo, de modo que cuando se aplica una corriente a sus dos terminales, este motor gira en un sentido, en cambio sí es aplicado inversamente, el sentido de giro se invierte.

- **Engranajes reductores**

Se encargan de convertir gran parte de la velocidad de giro del motor e corriente continua en torsión.

- **Circuito de control**

Es el encargado de controlar la posición del motor, recibe pulsos de entrada y ubica al motor en su nueva posición dependiendo de los pulsos recibidos.

4.8.5.2. Terminales

Los servomotores tienen 3 terminales:

- **Terminal positivo:** alimentación del motor (4 a 6 V)
- **Terminal negativo:** tierra del motor (0 V)
- **Entrada de señal:** señal de control del motor

Los colores de cada terminal varían con cada fabricante, el terminal positivo siempre es rojo el negativo varía entre marrón o negro y el terminal de entrada de señal suele ser de color blanco, naranja o amarillo

4.8.6. Servomotor MG996R

El servomotor, como el de la Figura 18, es un motor eléctrico que puede ser capaz de ser controlado, que se destaca por su buen torque (11 Kg), engranajes metálicos y gran robustez.

Utilizado principalmente en proyectos de robótica, como brazos robóticos y robots bípedos, puede rotar aproximadamente 180° (90° grados en cada dirección).



Figura 18 Servomotor MG996R

Fuente: (Lanfranco, 2016)

Tiene la facilidad de poder trabajar con diversas plataformas de desarrollo como:

- Arduino.
- PIC's.
- Raspberry Pi.
- Cualquier Microcontrolador.

Los terminales están distribuidos de la siguiente forma:

- **Café:** tierra (GND)
- **Rojo:** VCC (5 V)
- **Naranja:** Señal de control (PWM)

4.8.6.1. *Especificaciones Técnicas*

Tabla 5
Características Servomotor MG996R

Modelo	MG996R
Torque	9.4 Kg/cm (4.8 V), 11 Kg/cm (6 V)
Voltaje de operación	4.8 – 7.2 V
Peso	55 g
Plug	JR, FUTUBA general
Angulo de rotación	180° Maximo
Material de engranajes	Metal
Pulso ciclo	20 mS
Ancho del pulso	Entre 600 uS y 2400uS
Rango de temperatura	-30 a +60 °C

4.9. Software

4.9.1. Software Arduino IDE

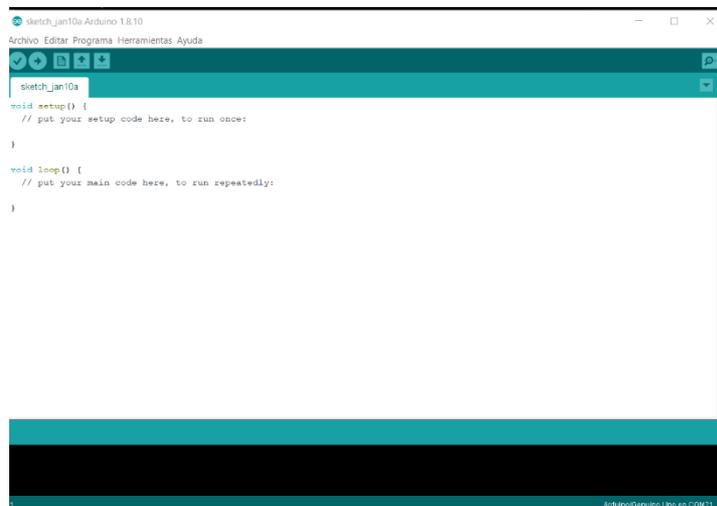


Figura 19 Arduino IDE Ventana Principal

Fuente: Propia

El software Arduino de código abierto (IDE) “Integrated Developmet Environment” (Entorno de Desarrollo Integrado). Este entorno de desarrollo permite realizar la programación de cada una de las placas de Arduino, es ejecutable en Windows, Mac OS X y Linux.

Programa informático compuesto por un conjunto de herramientas de programación, dedicado exclusivamente a un solo lenguaje de programación o bien puede utilizarse para varios, posteriormente este entorno de programación ha sido empaquetado como un programa de aplicación, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los programas de Arduino están compuestos por un solo fichero con extensión “ino”, el fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

4.9.2. Software EasyVR Commander

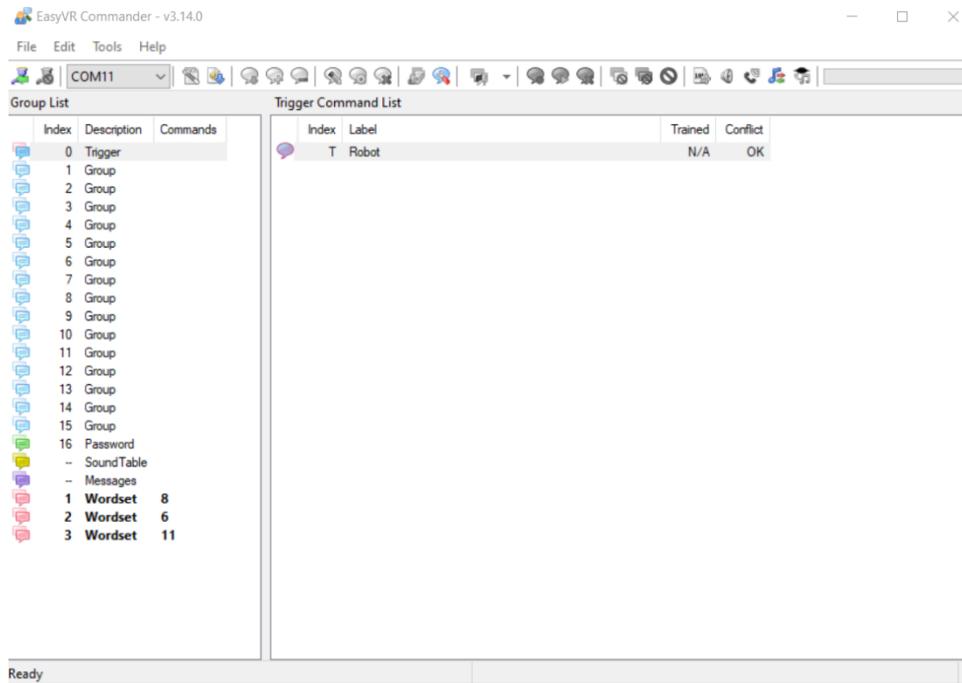


Figura 20 EasyVR Commander Ventana Principal

Fuente: Propia

El software EasyVR Commander se puede utilizar para configurar fácilmente su módulo EasyVr Conectado a su PC a través de un cable QuickUSB, o posteriormente por una placa adaptadora y mediante la placa host del microcontrolador con el programa “puente” proporcionado (disponible para la placa controladora ROBONOVA, Arduino, Parallax Sello Básico)

Software de desarrollo que puede definir grupos de comandos o contraseñas y generar una plantilla de código básica para manejar funciones o subrutinas del reconocimiento de voz.

4.9.3. Software Quick Synthesis 5

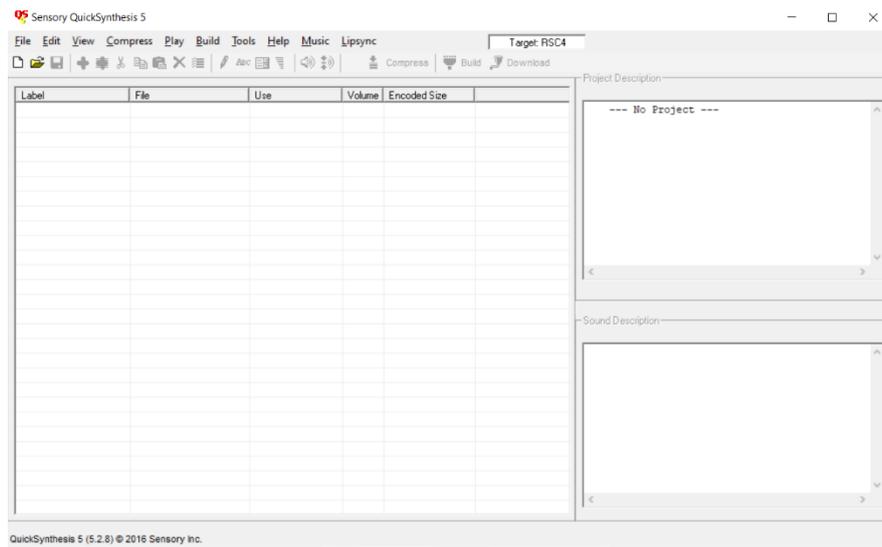


Figura 21 Sensory Quick Synthesis 5 Ventana Principal

Fuente: Propia

Quick Synthesis es un software diseñado como herramienta que ayuda a crear y gestionar síntesis de voz para Voice Extreme o distintas aplicaciones sensoriales de microcontroladores RSC.

4.9.4. Software Adobe Audition

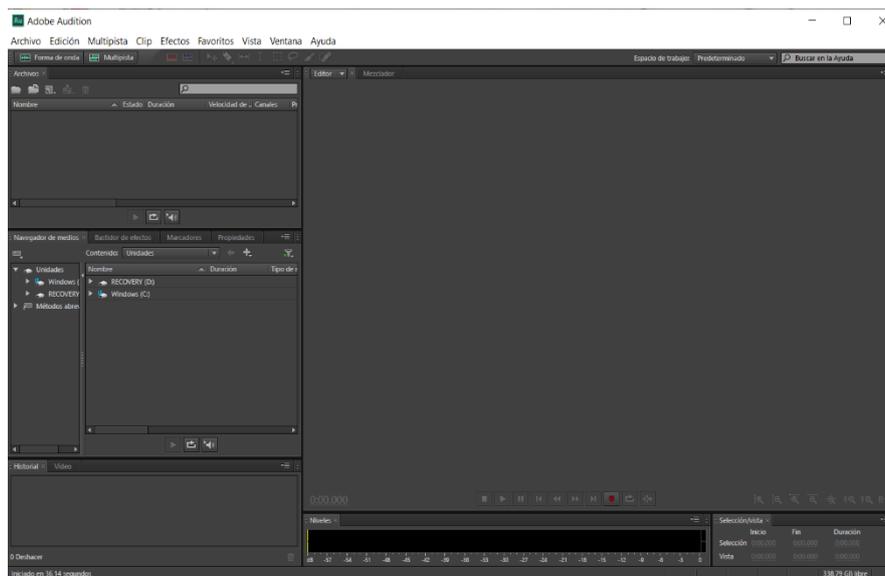


Figura 22 Adobe Audition Ventana Principal

Fuente: Propia

Es un software que tiene una forma de estudio de sonido destinado a la modificación o edición de audio digital con el cual se obtiene gran variedad de opciones y se puede realizar un trabajo con calidad ofreciendo un diseño de audio y video profesional con mezcla, edición y efectos de sonido, que permite un entorno de edición de ondas multipista con distinto formato, caracterizándose por disponer un mezclado tanto destructivo como no destructivo.

5. PROCEDIMIENTO – METODOLOGÍA

5.1. Diseño

5.1.1. Fuente variable 12V

En la actualidad muchos de los componentes electrónicos que se usa necesitan de corriente continua para su debido funcionamiento. Es por ello que es necesario la fuente variable, que permitirá ajustar los voltajes en un determinado rango según lo que se vaya alimentar. En este caso permite ajustar valores de voltaje entre 0 y 12 V con un valor de corriente de hasta 1.5 A.

5.1.1.1. Elaboración De Circuito Impreso

Para la elaboración del circuito impreso se siguió una secuencia de pasos, para poder realizar la implementación de los componentes que conformarían la fuente variable de 12 V, por lo cual se recurre al Software Proteus 8 Profesional (Figura 23), donde cual se diseñara el circuito impreso y posteriormente será estampado en una placa de baquelita.

1. Abrir el simulador Proteus.
2. Para la selección de componentes seleccione la opción Component Mode y a continuación seleccione la letra P.

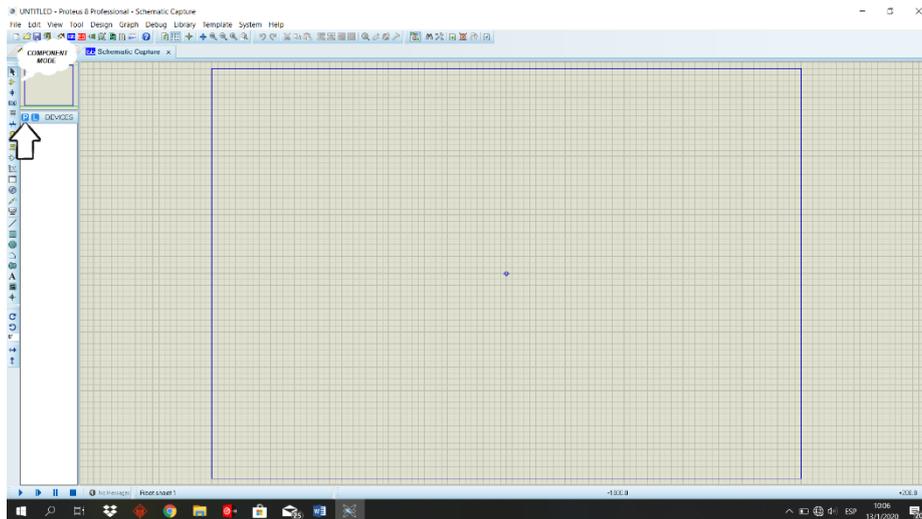


Figura 23 Ventana Principal Proteus

Fuente: Propia

3. Posteriormente se mostrará una ventana en la cual se introduce el código o el nombre del componente que se necesita. Una vez encontrado el componente se presionó doble clic para poder visualizarlo en la ventana principal.

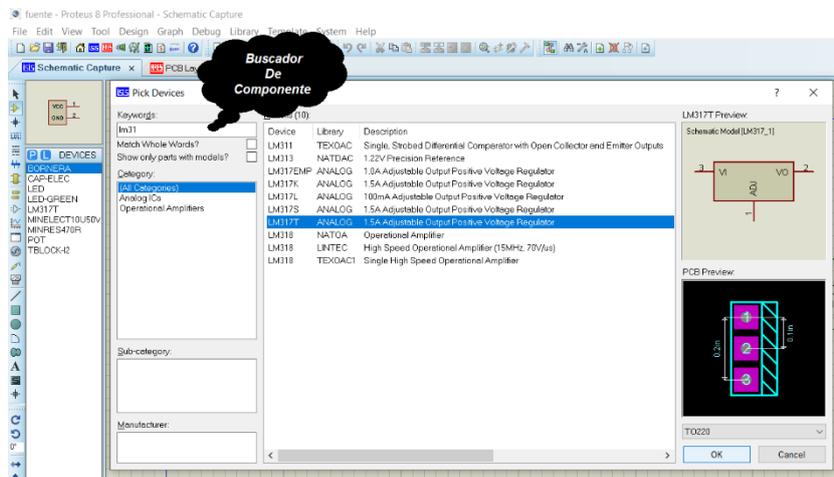


Figura 24 Ventana Paquete de Dispositivos

Fuente: Propia

4. Se comenzó a montar el circuito, en la ventana de diseño, arrastrando o colocando cada elemento en su debida posición. (Anexo 1)

5. Una vez finalizado el armado del circuito, seleccionar el icono de ARES para exportar el Circuito
6. En la siguiente ventana de ARES acomodar los componentes para diseñar el circuito impreso.

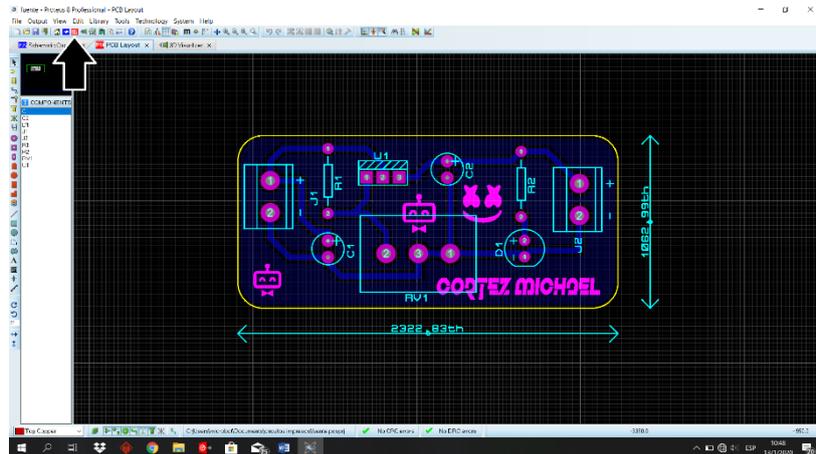


Figura 25 Ventana Principal De Ares

Fuente: Propia

7. Una vez acomodados todos los elementos del circuito, proceder a unir cada elemento con su debido pin. (Anexo 2)
8. Al finalizar totalmente el diseño, es posible imprimir el circuito eléctrico y poder plasmarlo en la placa de baquelita, y así posteriormente implementar y soldar los componentes electrónicos. (Anexo 3)

5.1.2. Amplificador de audio 20W

El dispositivo de salida o el módulo de audio son evidenciados de manera estratégica, para obtener un audio puntual en conformidad con el amplificador y la reproducción de audio.

5.1.2.1. *Elaboración del Circuito Impreso*

La elaboración del circuito impreso está planteada anteriormente, por una secuencia de pasos, para poder realizar la implementación de los componentes que conformarían el amplificador de Audio, por lo cual se recurre al Software Proteus 8 Profesional (Anexo 4 - 5), donde cual se diseñara el circuito impreso y posteriormente será estampado en una placa de baquelita.

Al finalizar totalmente el diseño, es posible imprimir el circuito eléctrico y poder plasmarlo en la placa de baquelita, y así posteriormente implementar y soldar los componentes electrónicos ver Anexo 6.

5.1.3. **Conexión Easy VR Shield 3 y Arduino Uno**

El Easy VR 3 proporciona con un paso estándar separado de 2.54mm, con pines machos que se pueden usar para conectar el modulo a una placa de pruebas, placa de prototipos, tablas personalizadas o tablas de soporte como el Easy VR Shield 3.

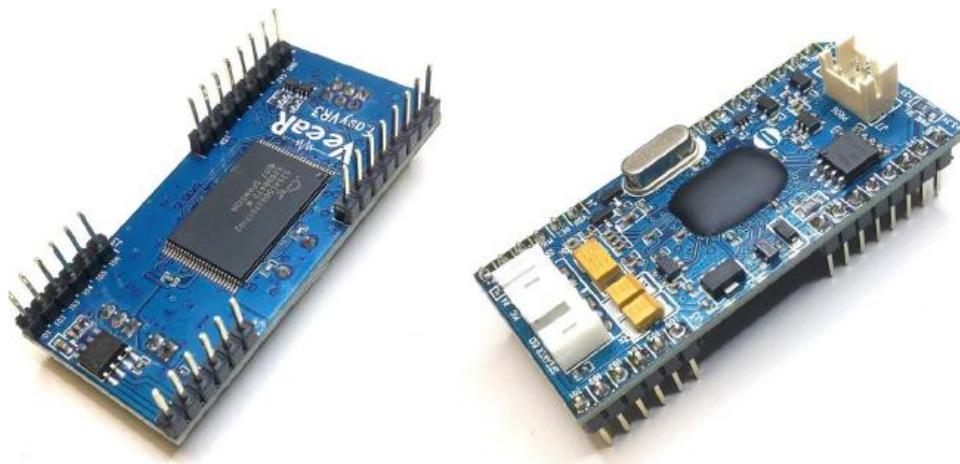


Figura 26 Easy VR 3

Fuente: (VeeR, 2016)

Al ensamblar el modulo, especialmente en preparación para Easy VR shield 3, asegúrese que los pines desde el lado inferior y el lado superior estén perfectamente soldados. (Figura 27)

Easy Vr Shield 3 es una placa adaptadora para el módulo Easy VR 3, diseñado para simplificar su uso entre los Arduinos. La Shield es compatible con cualquier placa Arduino que use UNO-R3, protección para los pines y que ejecute niveles de 3.3 V o 5V.



Figura 27 Easy VR Shield 3

Fuente: (VeeAR, 2016)

5.1.3.1. Características Easy VR Shield 3

- Admite placas principales de 5 V y 3,3 V a través del pin IOREF (el valor predeterminado es 5 V si este pin está ausente)
- Admite la conexión directa a la PC en placas principales con un chip USB / serie separado y un "modo puente" controlado por software especial en placas con solo interfaz USB nativa, para facilitar el acceso y la configuración con EasyVR Commander

- Permite diferentes modos de conexión en serie y también actualizaciones flash al módulo EasyVR incorporado (a través del puente de modo)
- Admite la reasignación de pines en serie utilizados por Shield (en modo SW)
- Proporciona un conector de salida de audio de 3.5 mm adecuado para auriculares o como salida de línea

Compatible con placas Arduino que tienen la interfaz 1.0 Shield (UNO R3) que incluye, entre otras:

- Arduino Zero
- Arduino Uno
- Arduino Mega
- Arduino Leonardo
- Due Arduino

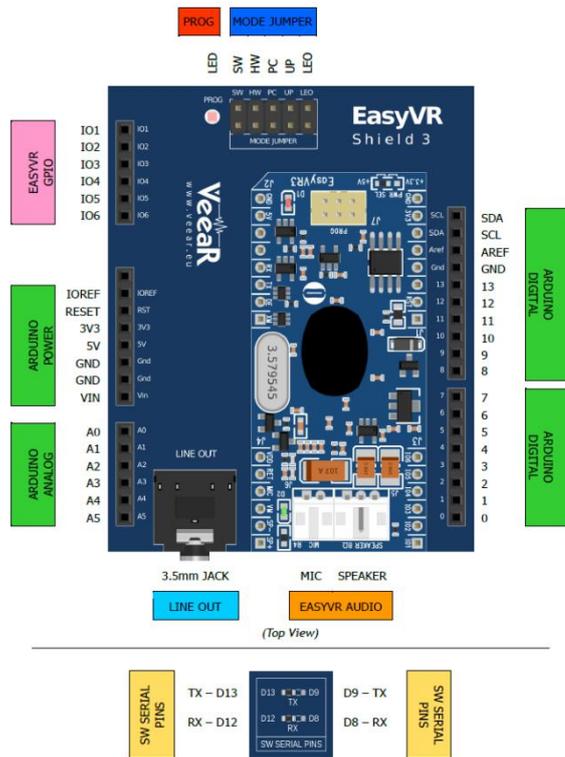


Figura 28 Easy Vr Shield 3 Especificaciones Técnicas

Fuente: (VeeAR, 2016)

Tabla 6
Características Técnicas Easy VR Shield 3

Grupo	Pin	Descripción
• ARDUINO HEADERS	-	Interfaz Arduino UNO-R3 Shield, conectores de paso (los pines 0-1 están en uso cuando J12 está configurado en UP, PC, HW o LEO) (los pines 12-13 u 8-9 están en uso cuando J12 está configurado en SW)
• EASYVR AUDIO	-	Conectores de cables de audio del módulo Easy VR 3 (micrófono y altavoz)
• LINE OUT	-	Jack estéreo / mono de 3.5 mm (Auriculares de 16 ohm – 32 ohm)
• MODE JUMPER	SW	Serie de Software Arduino (conectado a los pines 12-13 u 8-9)
	HW	Arduino Hardware Serial (conectado a los pines 0-1)
	PC	Modo PC (Arduino deshabilitado, Easy VR en moco comando)

	UP	Modo de actualización (Arduino deshabilitado, Easy VR modo de arranque)
	LEO	Actualización de Leonardo (Arduino habilitado, Easy VR en modo arranque)
• PROG	-	Indicador de luz roja para modos de programación (UP y LEO)
	RX	Use la resistencia para seleccionar el pin Software Serial RX: 12 u 8
• SW SERIAL PINS	TX	Use la resistencia para seleccionar el pin Software Serial RX: 13 o 9
	IO1	
	IO2	
• EASYVR GPIO	IO3	I / O de propósito general tal como se encuentra en el módulo Easy VR 3 incorporado (referenciado en el nivel lógico VDD interno)
	IO4	
	IO5	
	IO6	

5.1.3.2. Configuración de Puente

El funcionamiento del Easy VR Shield, se puede configurar diferentes opciones, donde están cuatro terminales con un jumper:

- **SW – Modo Serie de Software**

Se usó para controlar el módulo EasyVR desde su boceto Arduino a través de un puerto serie de software (usando los pines 12-13). También puede conectar el EasyVR Commander en este modo, siempre que ejecutar bocetos implementa el modo puente.

- **HW - Modo serie de Hardware**

Controlará el módulo EasyVR desde su boceto Arduino a través de la serie de hardware puerto (usando los pines 0-1).

- **PC - Modo de Conexión a PC**

Comprueba que la conexión directa con el EasyVR Commander. En este modo, el controlador Arduino es mantenido en el reinicio y solo se utiliza el adaptador USB / serie incorporado.

- **UP - Modo de Actualización de Flash**

Se utilizó para actualizaciones de firmware o para descargar datos de tablas de sonido y gramáticas personalizadas memoria flash de la placa del EasyVR Commander. En este modo, el controlador Arduino se mantiene en restablecer y solo se utiliza el adaptador USB / serie incorporado. El módulo EasyVR está configurado en modo de arranque.

5.1.4. Conexión de Arduino Uno con Servomotores

Para determinar el funcionamiento y la correcta conexión de los elementos electrónicos, como en este caso son los servomotores, se recurre al análisis y comprobación práctica, independientemente de cada uno para determinar su ubicación y precisión de movimiento en las articulaciones de la cabeza robótica.

Las conexiones dependerán del tipo de servomotor, por lo cual tiene 3 cables, uno ira a tierra, otro a la alimentación y el tercero aun pin PWM tal como se muestra en la Figura 29.

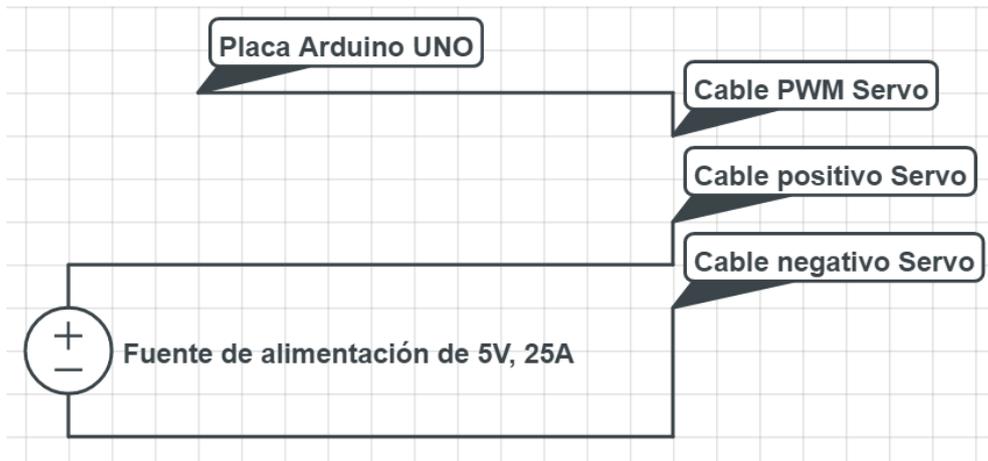


Figura 29 Conexión Servomotor Arduino

La programación del servomotor se debe utilizar una librería externa para poder controlar el servo desde Arduino, posteriormente se incorporará la librería, ubicarse en Programa / Incluye Library / Servo. (Figura 30)

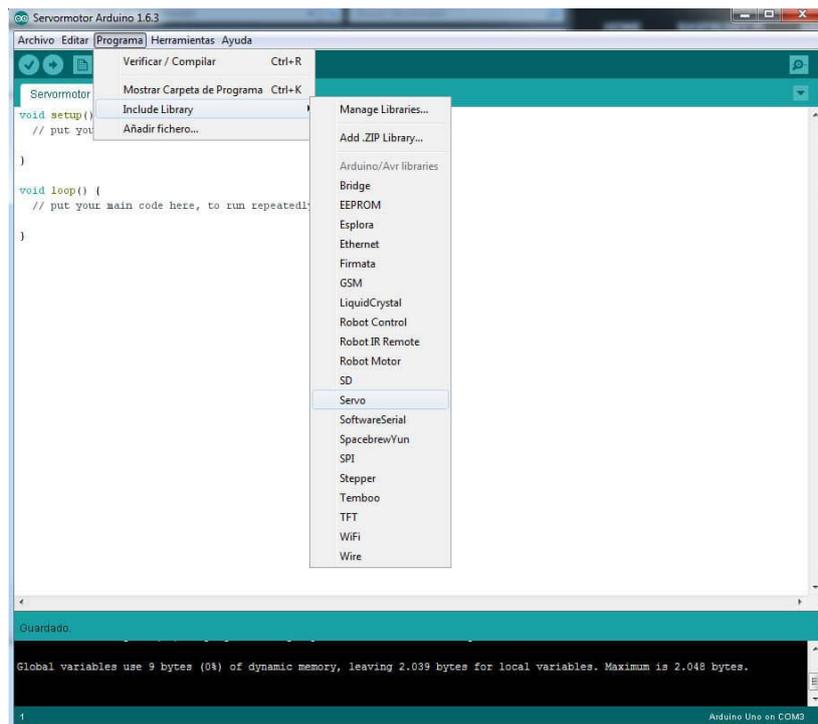


Figura 30 Librería Servo

Fuente: Propia

5.2. Construcción

5.2.1. Ensamblaje de Servomotores en la Cabeza Robótica

El sistema de ensamblaje de los servomotores en la cabeza robótica cumple la funcionalidad de mover las articulaciones, tanto como el del cráneo y la mandíbula, tal como se muestra en el Anexo 7.

Previamente determinar el Angulo de los servomotores 0° a 180° para obtener un funcionamiento óptimo y sin que tenga errores o complicaciones al momento de su aplicación. Cada uno de los ajustes se programó en el Software Arduino IDE, siendo un requisito determinante en la comprobación de los servomotores.

5.2.2. Ensamblaje de Módulos

Una vez terminado el ensamble de los servomotores, se procede a implementar los módulos como Easy VR Shield 3, Arduino UNO y NANO, planteando un diseño estético que permita el montaje y el desmontaje de los mismos, para lo cual se elaboró una caja contenedora, donde posteriormente se montara cada uno de los módulos en ubicaciones adecuadas. (Anexo 8-9)

5.2.3. Instalación de Parlantes 10W

El ensamble de los parlantes, se implementará en una caja de madera por razones que no cuenta con el cuerpo, para una mejor distribución de los equipos, para que cada uno mejore la salida de audio, con el amplificador de 20W. (Anexo 10)

5.3. Implementación

5.3.1. Sintetización De Sonidos De Voz

La sintetización de la tabla de sonidos se utilizará previamente el Software Adobe Audition, donde se grabará y editará la voz humanoide de la cabeza robótica, tomando en cuenta que los sonidos que puede soportar la tarjeta son de formato .wav 16 bit a 22050hz Canal: Mono.

1. Abrir el Software Adobe Audition
2. Una vez abierto el Software, ingresaremos a Archivo / Nuevo / Sesión Multipistas o podemos ingresar directamente con el atajo (Ctrl + N), tal como la Figura 31.

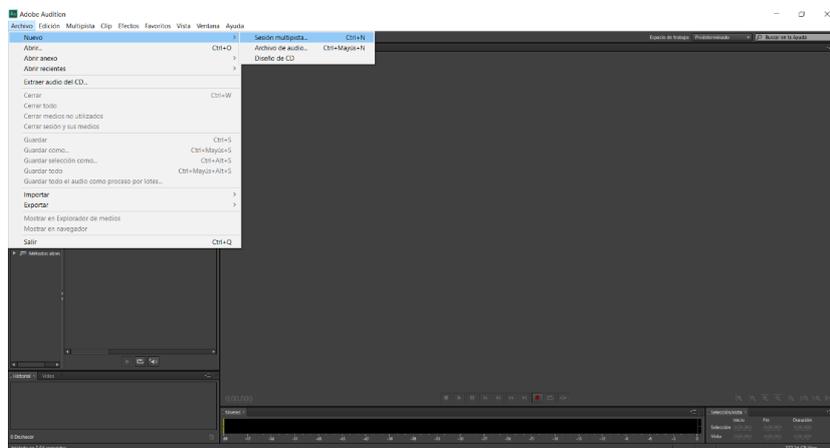


Figura 31 Nuevo Documento Adobe Audition

Fuente: Propia

3. A continuación, aparecerá un recuadro donde configuraremos, tal cual la Figura 32.

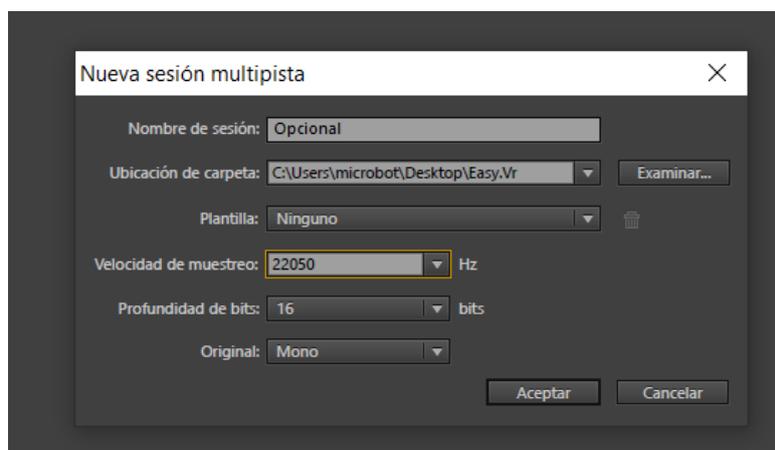


Figura 32 Formato de Audio

Fuente: Propia

- Posteriormente, en nuestra pista 1 o 2, iniciaremos la grabación del audio, tan solo se activará la letra R (se activara de un color rojo) y la letra S (se tornará en un color mostaza). Luego, activar el botón circular rojo en la parte inferior para proceder a grabar el audio.(Figura 33)

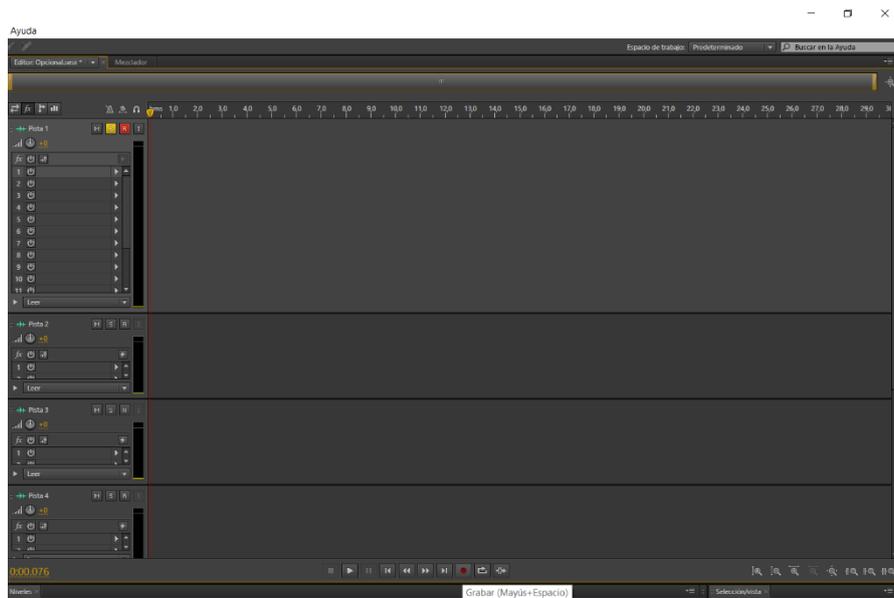


Figura 33 Ventana de Grabación

Fuente: Propia

- Al momento de realizar la grabación, es importante hablar a una distancia prudente, una vez finalizada la grabación, se incluirá efectos opcionalmente para pulir el audio.

5.3.2. Carga De Tabla De Sonidos

El procedimiento para cargar los sonidos de respuesta al módulo Easy VR shield 3, no se efectuará el Software Arduino, por lo cual se usará dos programas como Easy Commander y QuickSythesis.

Posteriormente antes de cargar la tabla de sonidos, se debe tomar en cuenta el siguiente formato:

- **Archivo:** .wav
- **Canal:** Mono
- 16 bits a 22050 hz

1. Abrir el Software QuickSynthesis

2. Clic en la hoja en blanco para abrir un nuevo documento. (Figura 34)

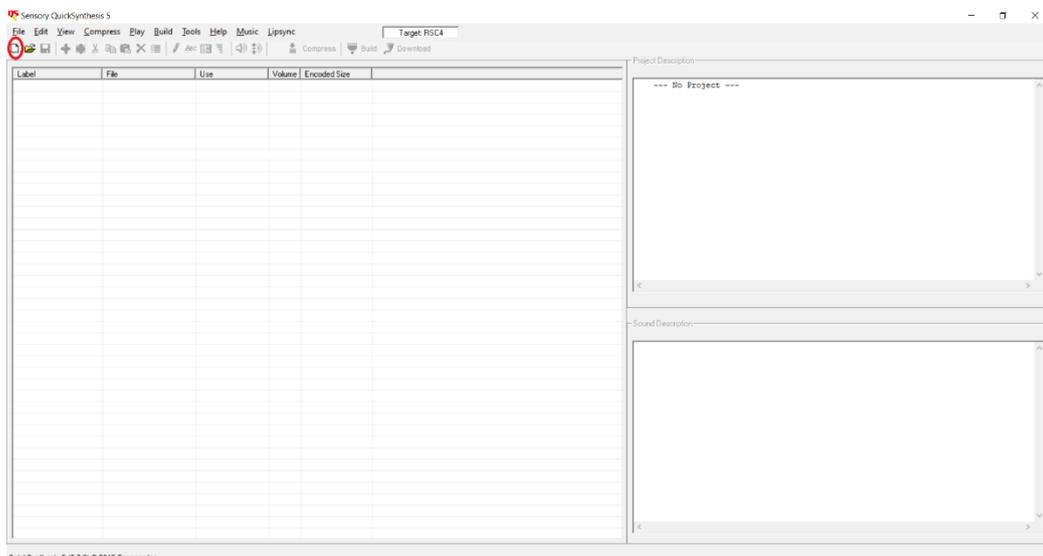


Figura 34 Nuevo Documento QuickSynthesis

Fuente: Propia

3. Damos Ok en RSC4

4. Asignamos un nombre al proyecto

5. Obtendremos una ventana nueva como la Figura 35, en la cual se dará clic en la cruz verde para, posteriormente seleccionar los audios sintetizados anteriormente.

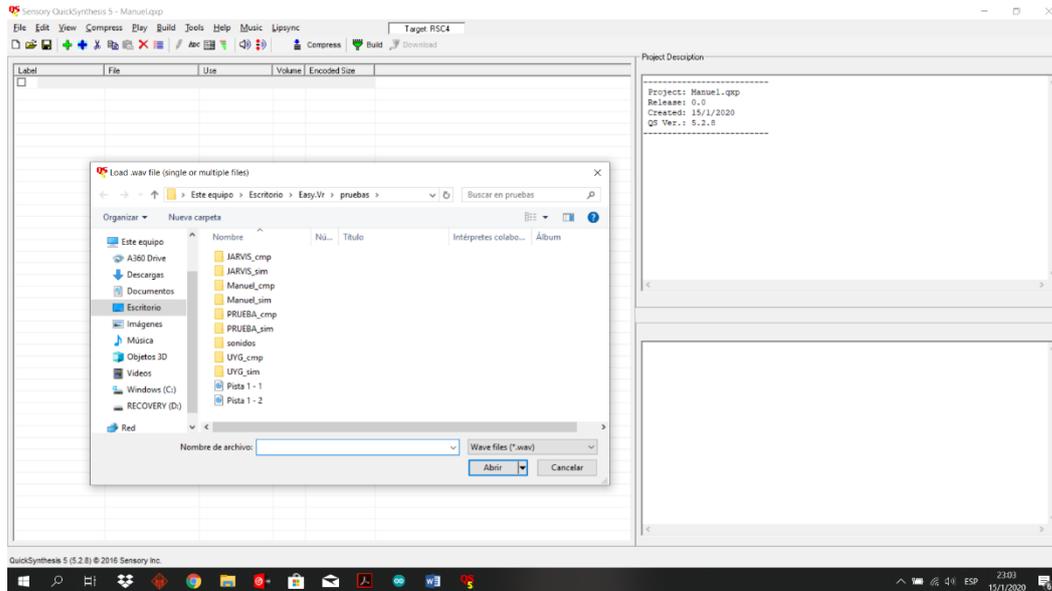


Figura 35 Carga de Tabla de Sonidos

Fuente: Propia

6. Una vez seleccionados los audios, se abrirá una ventana, en la cual se mantendrá los valores por default y clic en Ok for All.
7. Clic en auto Label All
8. Ahora clic en el botón verde que dice built y Ok.
9. Aparecerá una ventana de Build Error, clic en Sí.
10. Se ejecutará una compilación dentro del Software, una vez terminado quedará el texto en color verde, clic en Ok.
11. Finalmente, se procederá a guardar, presionado un clic en la casilla del disquete y luego al icono verde build para asegurar que el archivo esté en orden

5.3.3. Importación De Tabla De Sonido Al Software EasyVr. Commander

Luego de haber guardado el archivo de QuickSyntesis, automáticamente obtenemos un archivo con todos los sonidos, por lo cual se importarán al EasyVR Commander para cargarlos al módulo.

1. Conectar el módulo Easy VR Shield 3.
2. Asegurarse de poner el jumper del módulo en modo UP.
3. Abrir el Software EasyVR Commander y verificar el puerto COM.
4. Clic en el icono de una hoja en blanco con una bocina y flecha hacia abajo, (Update Custom Data) se encuentra en la parte superior. (Figura 36)

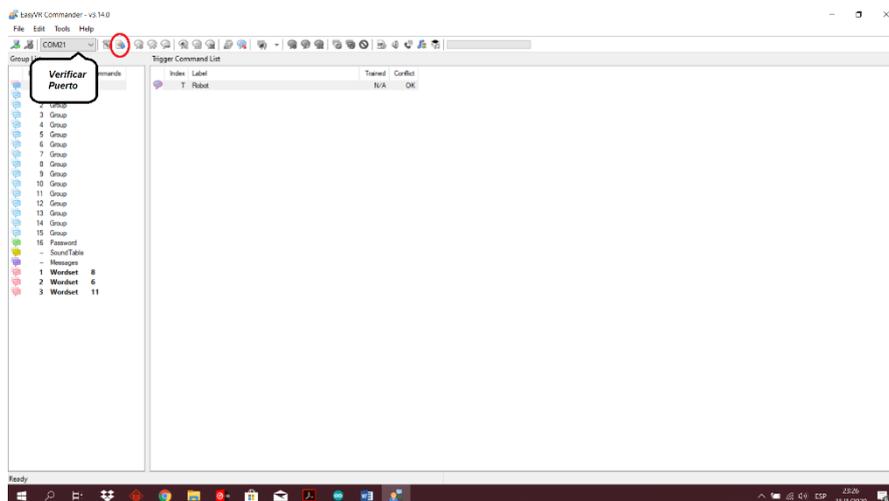


Figura 36 Easy Commander Carga Tabla De sonidos

Fuente: Propia

5. Se abrirá una venta en la cual se dará click en Import.
6. Seleccionar el archivo que creamos en QuickSyntesis. (Figura 37)

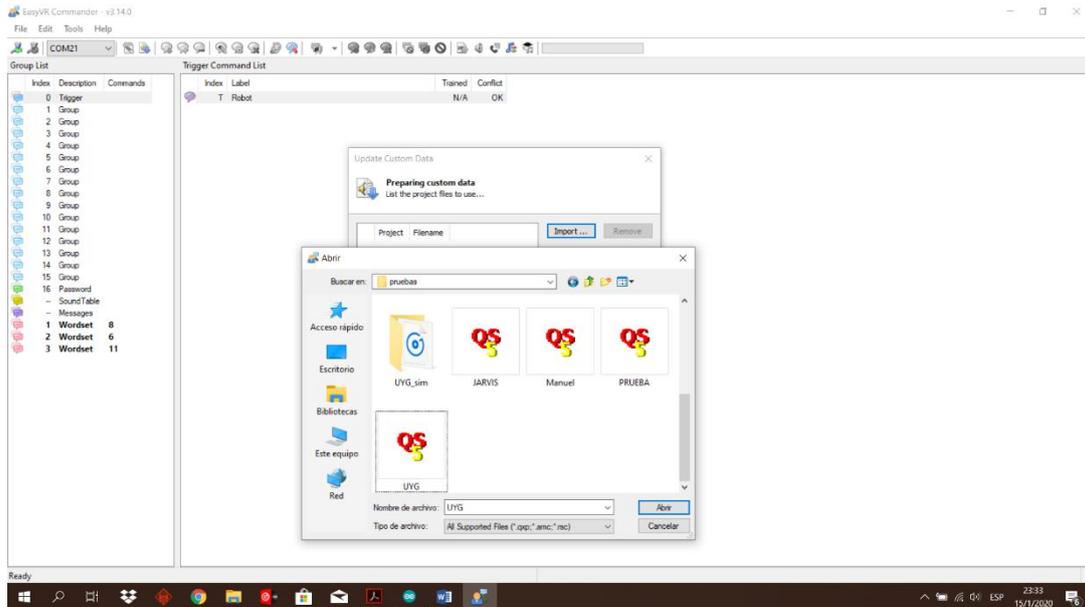


Figura 37 Selección de archivo de QuickSyntesis

Fuente: Propia

7. Previamente se activo la casilla SLOW transfer, para archivos pesados y si no es un archivo pesado, dar clic en download.
8. Concluyendo con el paso anterior, se desconecta el módulo y se cambia el jumper en modo PC, posteriormente asegurarse que en la casilla COM sea correspondiente con el módulo Easy VR shield 3.
9. Clic en el icono connect, que se encuentra en la parte superior. Dentro del apartado Sounds aparecerán los sonidos que hemos instalado. (Figura 38)

- Se procedió a configura el grupo Tigger, para poder ingresar al demás grupo tanto sea como el 1 o 2.
- Clic en el globo textual con un símbolo verde que se encuentra en la parte superior (add command), tal como se muestra en la Figura 40.

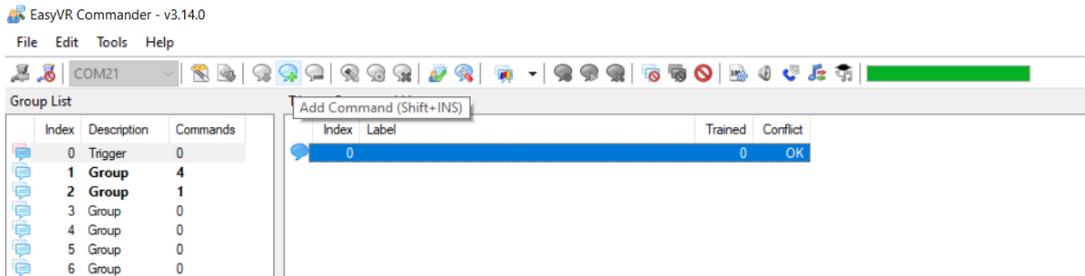


Figura 40 Add Command

Fuente: Propia

- Posteriormente se colocó un nombre opcional, en este caso, Mark.
- Luego se grabó la voz, dando clic en el icono de un globo textual con un engranaje (Train command), tal como se muestra en la Figura 41.

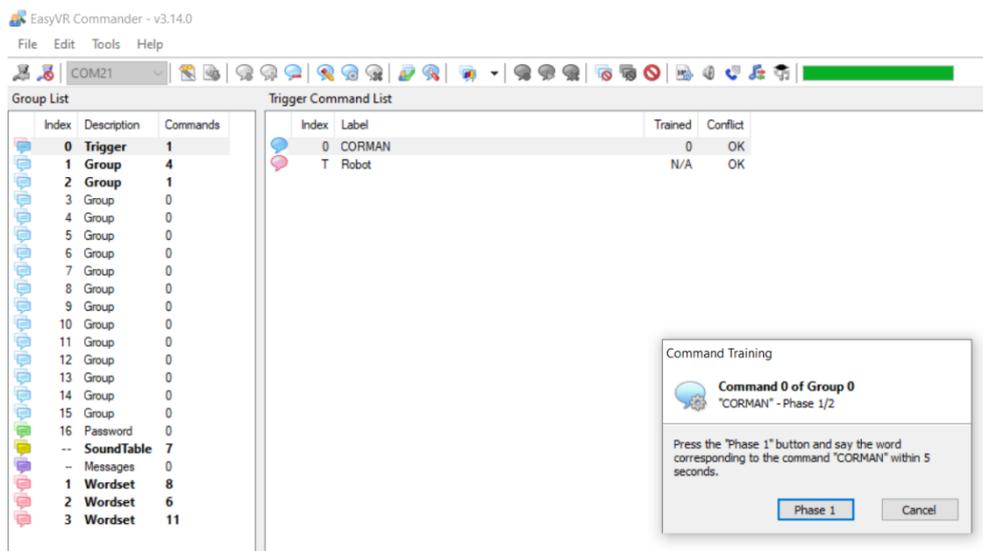


Figura 41 Train Command

Fuente: Propia

6. Aparecerá una ventana, en la cual se pronunciará la palabra planteada cerca del micrófono del módulo Easy VR.
7. Clic en Phase 1 y posteriormente en Phase 2 repitiendo el paso anterior (tratando de sonar lo más parecido posible).
8. Finalmente, si todo sale bien, la venta se quitará, si no se repitió los pasos para grabar la voz, por cuestión que encontró una incompatibilidad entre los dos sonidos.
9. Una vez finalizado el grupo Tigger, se procederá a configurar el Group 1 tal como se muestra en la Figura 42.

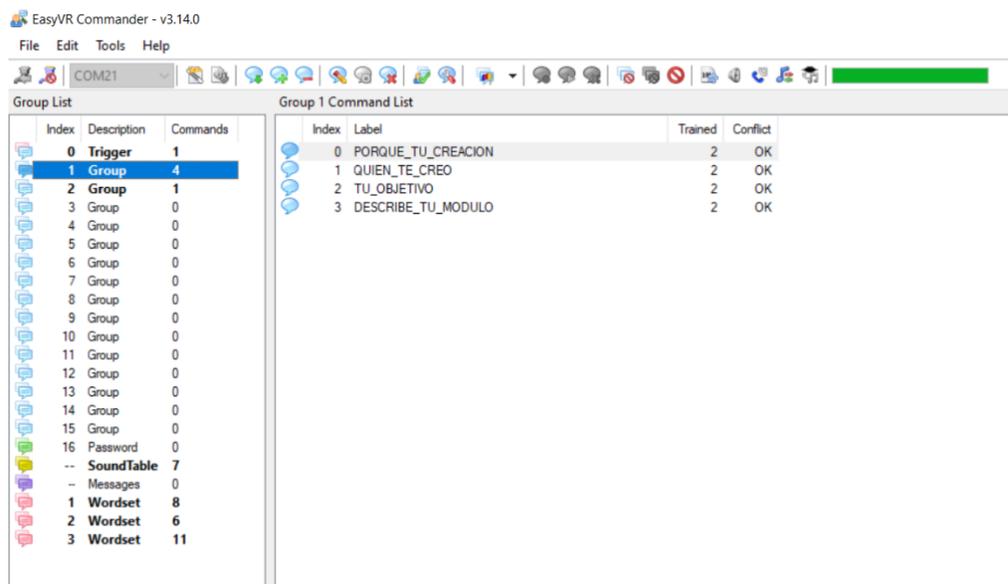


Figura 42 Group 1 Configuración de Comandos de Voz

Fuente: Propia

10. Una vez finalizado los comandos de voz, se procederá a exportar o generar el código IDE para poder editarlo en Arduino IDE, clic en la hoja con una barita (Generate Code).

5.3.5. Generación De Código IDE en el EasyVr Commander

La generación del código IDE, se aplicará una vez cuando todos los comandos de voz estén pre-establecidos, para poder exportar y editarlo en Arduino IDE. (Figura 43)

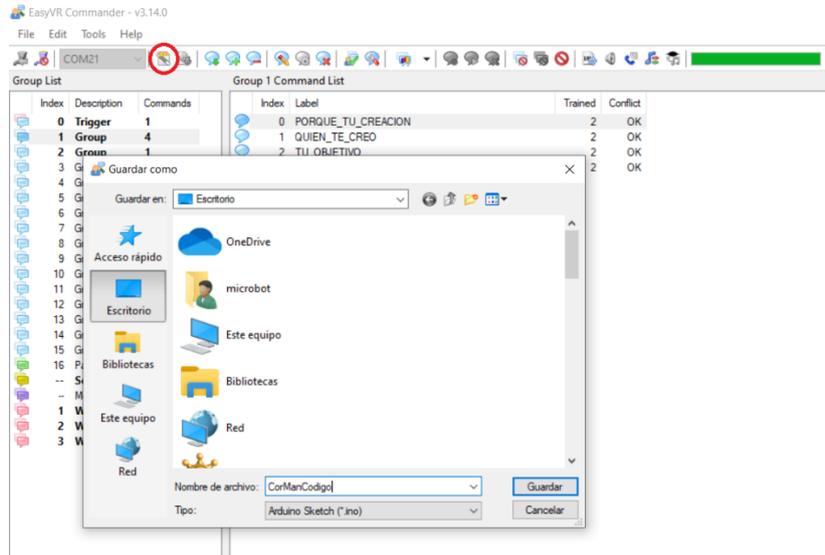


Figura 43 Generación de Código IDE

Fuente: Propia

Una vez obtenido el código, se procede a abrir el Software Arduino IDE y abrimos el código generado por el Software Easy Commander. (Figura 44)

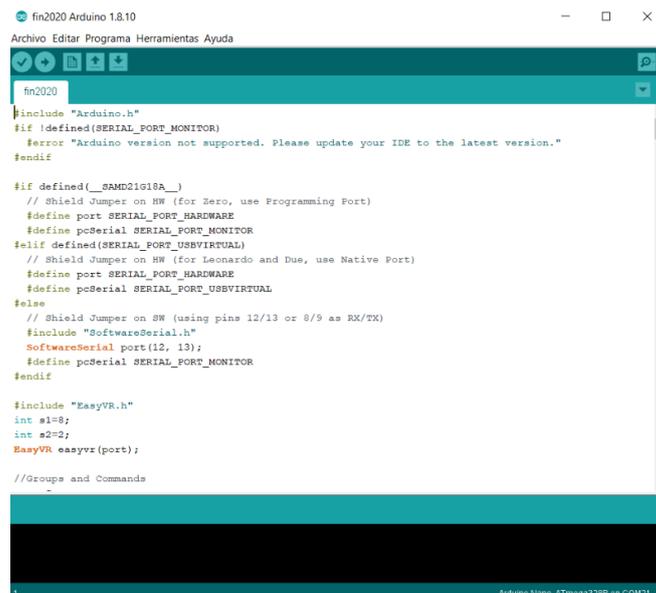


Figura 44 Código de Easy Commander en Arduino IDE

Fuente: Propia

5.3.6. Programación en Arduino IDE

Finalmente, una vez obtenido el código IDE se procederá a editarlo para controlarlo con la cabeza robótica y los audios pre-definidos anteriormente.

5.3.6.1. Programación del Reconocimiento De Voz

A continuación, se incluye el código modificado (las partes subrayadas son partes agregadas para que funcione correctamente la cabeza robótica), para el uso de los servomotores y la reproducción de los audios.

```
#include "Arduino.h"
#if ! definido ( SERIAL_PORT_MONITOR )
  #error "No se admite la versión de Arduino.
Actualice su IDE a la última versión". #terminara si

#if defined ( __SAM21G18A__ )
  // Shield Jumper en HW (para Zero, use Programming Port)
  #define port SERIAL_PORT_HARDWARE
  #define pcSerial SERIAL_PORT_MONITOR
#elif define ( SERIAL_PORT_USBVIRTUAL )
  // Shield Jumper en HW (para Leonardo y Due, use Native Port)
  #define puerto SERIAL_PORT_HARDWARE
  #define pcSerial SERIAL_PORT_USBVIRTUAL
#else
  // Escudo Jumper en SW (usando pasadores 12/13 o 8/9 como RX / TX)
  # include "SoftwareSerial.h"
  SoftwareSerial puerto ( 12 , 13 ) ;
  #define pcSerial SERIAL_PORT_MONITOR
#endif

#include "EasyVR.h"
int s1 = 8 ;
int s2 = 2 ;
EasyVR easyvr ( puerto ) ;

// Grupos y comandos
enum Grupos
{
  GROUP_0 = 0 ,
  GROUP_1 = 1 ,
  GROUP_2 = 2 ,
} ;

enumeración Group0
{
  G0_CORMAN = 0 ,
} ;

enum Grupo1
{
  G1_PORQUE_TU_CREACION = 0 ,
  G1_QUIEN_TE_CREO = 1 ,
  G1_TU_OBJETIVO = 2 ,
  G1_DESCRIBE_TU_MODULO = 3 ,
} ;
```

```

enumeración Group2
{
    G2_MANUEL = 0 ,
} ;

//
Conjuntos de palabras de enumeración de gramáticas y palabras { SET_1 = - 1 ,
SET_2 = - 2 , SET_3 = - 3 , } ;

enum Wordset1
{
    S1_ACTION = 0 ,
    S1_MOVE = 1 ,
    S1_TURN = 2 ,
    S1_RUN = 3 ,
    S1_LOOK = 4 ,
    S1_ATTACK = 5 ,
    S1_STOP = 6 ,
    S1_HELLO = 7 ,
} ;

enum Wordset2
{
    S2_LEFT = 0,
    S2_RIGHT = 1,
    S2_UP = 2,
    S2_DOWN = 3,
    S2_FORWARD = 4,
    S2_BACKWARD = 5,
};

enum Wordset3
{
    S3_ZERO = 0,
    S3_ONE = 1,
    S3_TWO = 2,
    S3_THREE = 3,
    S3_FOUR = 4,
    S3_FIVE = 5,
    S3_SIX = 6,
    S3_SEVEN = 7,
    S3_EIGHT = 8,
    S3_NINE = 9,
    S3_TEN = 10,
};

// use negative group for wordsets
int8_t group, idx;

void setup()
{
    pinMode(s1, OUTPUT);
    pinMode(s2, OUTPUT);
    // setup PC serial port
    pcSerial.begin(9600);
    bridge:
    // bridge mode?
    int mode = easyvr.bridgeRequested(pcSerial);
    switch (mode)
    {
    case EasyVR::BRIDGE_NONE:
        // setup EasyVR serial port
        port.begin(9600);
        // run normally
        pcSerial.println(F("Bridge not requested, run normally"));
    }
}

```

```

    pcSerial.println(F("---"));
    break;

case EasyVR::BRIDGE_NORMAL:
    // setup EasyVR serial port (low speed)
    port.begin(9600);
    // soft-connect the two serial ports (PC and EasyVR)
    easyvr.bridgeLoop(pcSerial);
    // resume normally if aborted
    pcSerial.println(F("Bridge connection aborted"));
    pcSerial.println(F("---"));
    break;

case EasyVR::BRIDGE_BOOT:
    // setup EasyVR serial port (high speed)
    port.begin(115200);
    pcSerial.end();
    pcSerial.begin(115200);
    // soft-connect the two serial ports (PC and EasyVR)
    easyvr.bridgeLoop(pcSerial);
    // resume normally if aborted
    pcSerial.println(F("Bridge connection aborted"));
    pcSerial.println(F("---"));
    break;
}

// initialize EasyVR
while (!easyvr.detect())
{
    pcSerial.println(F("EasyVR not detected!"));
    for (int i = 0; i < 10; ++i)
    {
        if (pcSerial.read() == '?')
            goto bridge;
        delay(100);
    }
}

pcSerial.print(F("EasyVR detected, version "));
pcSerial.print(easyvr.getID());

if (easyvr.getID() < EasyVR::EASYVR3)
    easyvr.setPinOutput(EasyVR::IO1, LOW); // Shield 2.0 LED off

if (easyvr.getID() < EasyVR::EASYVR)
    pcSerial.print(F(" = VRbot module"));
else if (easyvr.getID() < EasyVR::EASYVR2)
    pcSerial.print(F(" = EasyVR module"));
else if (easyvr.getID() < EasyVR::EASYVR3)
    pcSerial.print(F(" = EasyVR 2 module"));
else
    pcSerial.print(F(" = EasyVR 3 module"));
pcSerial.print(F(", FW Rev. "));
pcSerial.println(easyvr.getID() & 7);

easyvr.setDelay(0); // speed-up replies

easyvr.setTimeout(5);
easyvr.setLanguage(0); //<-- same language set on EasyVR Commander when code was
generated

group = EasyVR::TRIGGER; //<-- start group (customize)
}

void loop()
{
    if (easyvr.getID() < EasyVR::EASYVR3)
        easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
}

```

```

if (group < 0) // SI wordset/grammar
{
    pcSerial.print("Say a word in Wordset ");
    pcSerial.println(-group);
    easyvr.recognizeWord(-group);
}
else // SD group
{
    pcSerial.print("Say a command in Group ");
    pcSerial.println(group);
    easyvr.recognizeCommand(group);
}

do
{
    // allows Commander to request bridge on Zero (may interfere with user
    protocol)
    if (pcSerial.read() == '?')
    {
        setup();
        return;
    }
    // <<-- can do some processing here, while the module is busy
}
while (!easyvr.hasFinished());

if (easyvr.getID() < EasyVR::EASYVR3)
    easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off

idx = easyvr.getWord();
if (idx == 0 && group == EasyVR::TRIGGER)
{
    // beep
    easyvr.playSound(0, EasyVR::VOL_FULL);
    // print debug message
    pcSerial.println("Word: ROBOT");
    // write your action code here
    // group = GROUP_X\SET_X; <-- jump to another group or wordset
    return;
}
else if (idx >= 0)
{
    // beep
    easyvr.playSound(0, EasyVR::VOL_FULL);
    // print debug message
    uint8_t flags = 0, num = 0;
    char name[32];
    pcSerial.print("Word: ");
    pcSerial.print(idx);
    if (easyvr.dumpGrammar(-group, flags, num))
    {
        for (uint8_t pos = 0; pos < num; ++pos)
        {
            if (!easyvr.getNextWordLabel(name))
                break;
            if (pos != idx)
                continue;
            pcSerial.print(F(" = "));
            pcSerial.println(name);
            break;
        }
    }
    // perform some action
    action();
    return;
}
idx = easyvr.getCommand();

```

```

if (idx >= 0)
{
    // beep
    easyvr.playSound(0, EasyVR::VOL_FULL);
    // print debug message
    uint8_t train = 0;
    char name[32];
    pcSerial.print("Command: ");
    pcSerial.print(idx);
    if (easyvr.dumpCommand(group, idx, name, train))
    {
        pcSerial.print(" = ");
        pcSerial.println(name);
    }
    else
        pcSerial.println();
    // perform some action
    action();
}
else // errors or timeout
{
    if (easyvr.isTimeout())
        pcSerial.println("Timed out, try again...");
    int16_t err = easyvr.getError();
    if (err >= 0)
    {
        pcSerial.print("Error ");
        pcSerial.println(err, HEX);
    }
}
}

void action()
{
    switch (group)
    {
    case GROUP_0:
        switch (idx)
        {
        case G0_CORMAN:
            digitalWrite(s1,HIGH);
            delay(100);
            digitalWrite(s1,LOW);
            easyvr.playSound(1, EasyVR::VOL_FULL);
            // write your action code here
            group = GROUP_1; //\SET_X; <-- or jump to another group or wordset for
composite commands

            break;
        }
        break;
    case GROUP_1:
        switch (idx)
        {
        case G1_PORQUE TU CREACION:
            easyvr.playSound(4, EasyVR::VOL_FULL);
            // write your action code here
            group = GROUP_0; //\SET_X; <-- or jump to another group or wordset for
composite commands
            break;
        case G1 QUIEN TE CREO:
            digitalWrite(s2,HIGH);
            delay(1000);
            digitalWrite(s2,LOW);
            easyvr.playSound(2, EasyVR::VOL_FULL);
            // write your action code here
            group = GROUP_0; //\SET_X; <-- or jump to another group or wordset for
composite commands

```

```

        break;
    case G1 TU OBJETIVO:
        easyvr.playSound(3, EasyVR::VOL_FULL);
        // write your action code here
        group = GROUP_2; //\SET_X; <-- or jump to another group or wordset for
composite commands
        break;
    case G1 DESCRIBE TU MODULO:
        easyvr.playSound(5, EasyVR::VOL_FULL);
        // write your action code here
        group = GROUP_0; //\SET_X; <-- or jump to another group or wordset for
composite commands
        break;
    }
    break;
case GROUP_2:
    switch (idx)
    {
    case G2 CORMAN:
        easyvr.playSound(6, EasyVR::VOL_FULL);
        // write your action code here
        group = GROUP_1; //\SET_X; <-- or jump to another group or wordset for
composite commands
        break;
    }
}
}

```

5.3.6.2. Programación del movimiento de los servomotores (Código IDE)

A continuación, se incluye el código básico para posicionar el servo en los ángulos 0° a 180°, tanto como en la mandíbula y cráneo.

```

// Incluimos la librería para poder controlar el servo
#include <Servo.h>

// Declaramos la variable para controlar el servo
Servo s1;
Servo s2;

// Declaramos las variables
int pos1=0;
int pos2=0;
int p1 = 8;
int p2 = 2;
int vp1= 0;
int vp2= 0;
int t1= 5;
int t2= 8;
int t3= 5;
int t4= 15;

void setup()
{
    // Iniciamos el servo para que empiece a trabajar con el pin 9 - 10
    s1.attach(9);
    s2.attach(10);
    pinMode(p1, INPUT);
    pinMode(p2, INPUT);
}

```

```

//Dlecaramos Funciones Para Cada Servomotor (Mandibula y Craneo)
//Mandibula
void f1 ()
{
  for(pos1 = 85; pos1 < 140; pos1 += 1)
  {
    s1.write(pos1);
    delay(t2);
  }
  for(pos1 = 130; pos1 >= 85 ; pos1 -= 1)
  {
    s1.write(pos1);
    delay(t1);
  }
  for(pos1 = 85; pos1 < 140; pos1 += 1)
  {
    s1.write(pos1);
    delay(t2);
  }
  for(pos1 = 130; pos1 >= 85 ; pos1 -= 1)
  {
    s1.write(pos1);
    delay(t1);
  }
}
//Craneo
void f2 ()
{
  for(pos1 = 70; pos1 < 110; pos1 += 1)
  {
    s1.write(pos1);
    delay(t1);
  }
  for(pos1 = 110; pos1 >= 70 ; pos1 -= 1)
  {
    s1.write(pos1);
    delay(t2);
  }
  for(pos1 = 90; pos1 < 110; pos1 += 1)
  {
    s1.write(pos1);
    delay(t1);
  }
  for(pos1 = 110; pos1 >= 70 ; pos1 -= 1)
  {
    s1.write(pos1);
    delay(t1);
  }
}

void loop()
{
  vp1= digitalRead(p1);
  vp2= digitalRead(p2);

  if (vp1 == HIGH)
  {
    f1();
    f1();
  }

  if (vp2 == HIGH)
  {
    f2();
    f2();
  }
}

```

6. CONCLUSIONES

- El proceso de investigación de la cabeza robótica, se comprobó satisfactoriamente, mediante técnicas de investigación y procesos de comandos de voz pre-programados en el módulo Easy VR 3, visto que el modulo al recibir un comando de voz, la cabeza robótica reacciona y realiza los movimientos preestablecidos en la programación.
- Se consiguió elaborar una plataforma de comandos de voz pre-programados en el módulo Easy VR 3, aplicados mediante el Software EasyVr Commander, que obtiene una interfaz gráfica y características importantes, donde se puede realizar las ordenes de manera fácil y eficiente, debido a que se pudo grabar y reconocer voces definidas por el propio usuario.
- El diseño estético, eléctrico y electrónico constituye, de dos servomotores MG996R controlados por un Arduino NANO, una alimentación eléctrica, articulaciones de la cabeza robótica y el módulo Easy Vr 3 que esta ensamblada en un Arduino UNO, en el cual se efectuó de forma abierta, donde cada elemento funcional puede ser reemplazado, puesto que posee una interfaz atractiva, bajo un entorno fácil de manipular por cualquier usuario.
- En relación a las pruebas de funcionamiento se pudo descartar varios errores, que, en un principio, se desarrolló en la programación del módulo Easy VR 3, posteriormente a la variedad de Softwares que se utilizó previamente para configurar los comandos de voz y el audio de respuesta preestablecido para cada comando.

7. REEFERENCIAS

7.1. BIBLIOGRAFÍA

Anonimo. (2018). *Invox.Medical*. Recuperado el 03 de Diciembre de 2019, de La Inteligencia Artificial en el Reconocimiento de Voz: <https://invoxmedical.com/noticias/la-inteligencia-artificial-en-el-reconocimiento-de-voz/>

Astudillo, C., López, H., & Sotomayor, N. (2014). *Ensamblaje y Control De Un Rostro Robotico Que Simue Los Movimientos Del Rostro Humano*. Quito - Ecuador: Escuela Politecnica Nacional. Recuperado el 19 de Diciembre de 2019

Barrios, K., Lopez, J., Mendieta, S., Benavides, R., & Yessica, S. (2018). Sistema De Recnocimineto De Voz. *Un Enlace En La Comunicacion Hombre-Maquina*, 4, 92. Recuperado el 20 de Diciembre de 2019, de <file:///C:/Users/microbot/Downloads/1827-Texto%20del%20art%C3%ADculo-8766-1-10-20180623.pdf>

Billy, R. (2011). *The New Atlantis*. Recuperado el 18 de Diciembre de 2019, de Science, Virtue, And The Futere Of Humanity: <https://www.thenewatlantis.com/publications/machine-morality-and-human-responsibility>

Blum, J. (2014). *Arduino A Fondo*. (M. Pascual , Trad.) Madrid: Anaya Multimedia. Recuperado el 20 de Diciembre de 2019

CETI. (2015). *Academia.edu*. Recuperado el 05 de Noviembre de 2019, de Historia de la Robotica: https://www.academia.edu/31702044/Historia_de_la_robotica

Dimate, J. (2014). *Libro De Robotica*.

- Dr. Ponce, P. (2010). *Inteligencia Artificial Con Aplicaciones A La Ingenieria*. Mexico: Alfaomega.
- Emaze. (2011). *Evolucion de la Robotica*. Recuperado el 18 de Diciembre de 2019, de <https://app.emaze.com/@ALWICZZI/cronologa-de-los-eventos-ocurridos-en-la-evolucion-de-la-robotica#2>
- Garcia, A. (2012). *Inteligencia Artificial*. Madrid: Rc Libros. Recuperado el 20 de Diciembre de 2019, de <http://www.rclibros.es/pdf/InteligenciaArtificial.pdf>
- Gonzales, E. (2014). *Disciplinas*. Recuperado el 18 de Diciembre de 2019, de La Robotica: <https://sites.google.com/site/robotsovedientes/home/disciplinas-1>
- Jadiaz. (2016). *MiArduino*. Recuperado el 20 de Diciembre de 2019, de Arduino Uno: <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>
- Lanfranco, L. (2016). *Blog*. Recuperado el 10 de Enero de 2020, de MG996R: <http://lisandrolanfranco.com/2016/02/datasheet-mg996r-tower-pro-servo/>
- López, J. (2016). *Impresoras 3D*. CAP (Centro De Análisis Y Prospectiva Gabinete Técnico De La Guardia Civil). Recuperado el 23 de Diciembre de 2019, de https://intranet.bibliotecasgc.bage.es/intranet-tmpl/prog/local_repository/documents/17854.pdf
- Mejia, M & Nuñez, J. (2016). *Diseño y Construcion de un Robot Social Para Interacción Hombre-Maquina*. Sangolquí: Universidad De Las Fuerzas Armadas.
- Molina, J. (04 de Junio de 2015). *Profesor Molina*. Obtenido de Historia De La Robotica: <http://www.profesormolina.com.ar/index.htm>

- Moriello, S. (2008). *Megatendencias*. Recuperado el 19 de Diciembre de 2019, de Robot Sociales, La Nueva Generacion: https://www.tendencias21.net/Robots-sociales-la-nueva-generacion_a2833.html
- Ocaña, G. (2015). *Aula De Robotica*. Madrid: Dextra S.L. Recuperado el 19 de Diciembre de 2019
- Ramon, L. (2012). *Blogs Sociedad*. Recuperado el 20 de Diciembre de 2019, de Ramon Llull, El Ars Manga Y La INformatica: <https://blogs.elpais.com/turing/2012/10/ramon-llull-el-ars-magna-y-la-informatica.html>
- Rivera, J. (2016). *Research Gate*. Recuperado el 19 de Diciembre de 2019, de Que Es Un Robot.
- Ruales, C. (2015). *Diseños y Construcción De Un Rostro Robotico De Apariencia Humana, Con Capacidad De Ejecutar Gestos y Emociones, Para La Espe Extension Y Emociones, Para La Espe Extension Latacunga*. Latacunga: Universidad De Las Fuerzas Armadas. Recuperado el 19 de Diciembre de 2019
- Sanchez, M. (2007). *Timelines*. Recuperado el 18 de Diciembre de 2019, de Historia de la Robotica.
- Sandoval, E. (2012). *ResearchGate*. Recuperado el 19 de Noviembre de 2019, de Interaccion Humano Robot y sus Aplicaciones Educativas.: https://www.researchgate.net/publication/268746492_Interaccion_Humano_Robot_y_sus_Aplicaciones_Educativas
- Sanz, P. (2006). *RobInt-Apunes.pdf*. Recuperado el 19 de Diciembre de 2019, de Introducion A La Robotica Intiligente: <http://www3.uji.es/~sanzp/robot/RobInt-Apunes.pdf>

Takayuki, K. y Hiroshi, I. (2012). *Introduction to Network Robot Approach for Human-Robot Interaction*. Unit States: Taylor & Francis Group.

Toyota. (2015). *Innovation*. Recuperado el 19 de Diciembre de 2019, de Partner Robot: https://www.toyota-global.com/innovation/partner_robot/concept/history/index.html

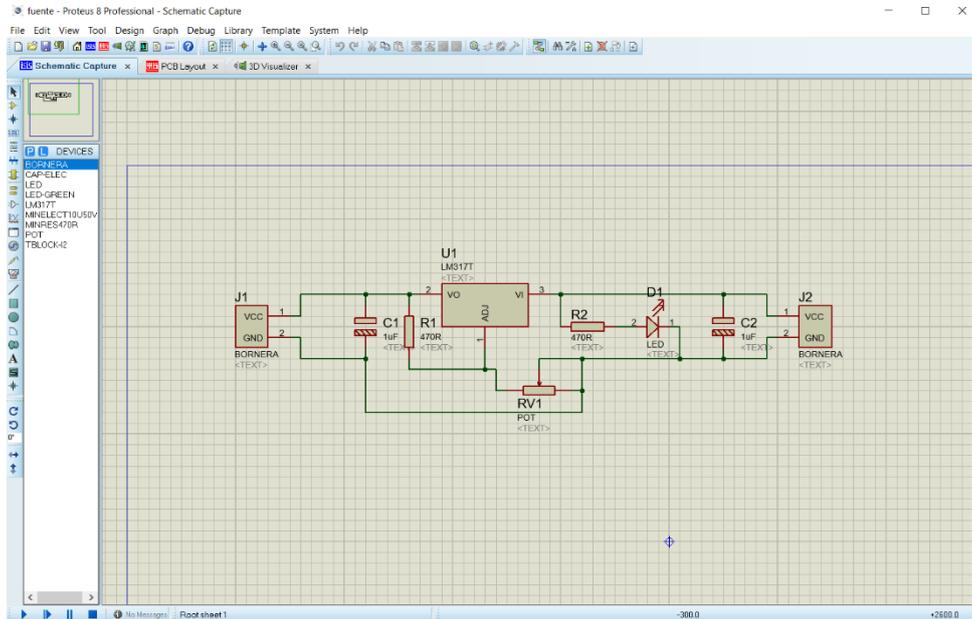
Universidad Calors III de Madrid. (2019). *RoboticsLab*. Recuperado el 19 de Diciembre de 2019, de Maggie.

Veear. (2016). *EasyVr 3 User Manual 1.0.16*. ROBOTTEHC. Recuperado el 24 de Diciembre de 2019

WordPress. (2017). *WordPress*. Recuperado el 20 de Diciembre de 2019, de Inteligencia Artificial: <https://inteligenciaartificial20172grupo5.wordpress.com/>

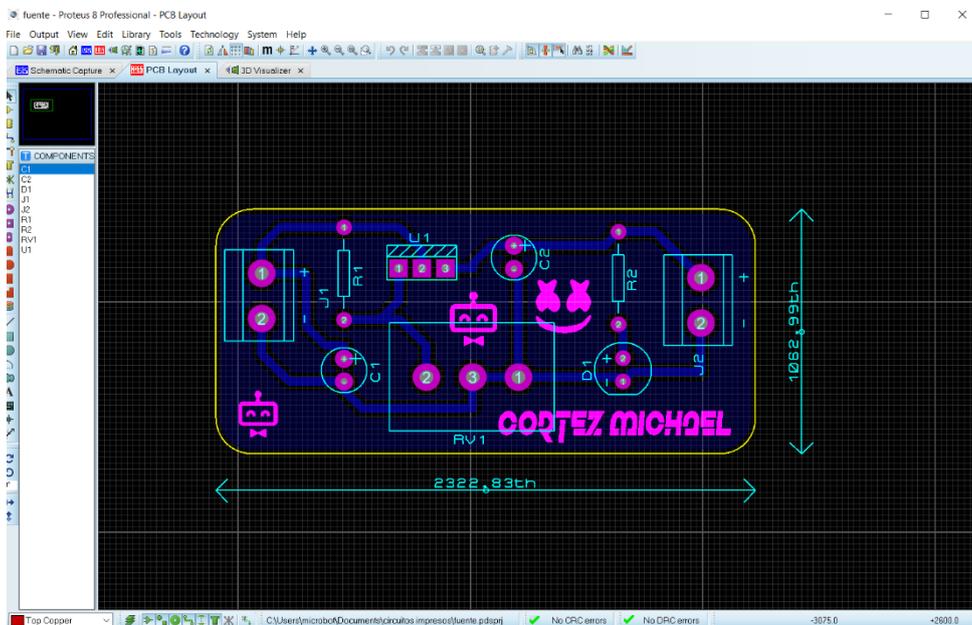
8. ANEXOS

8.1. Circuitos



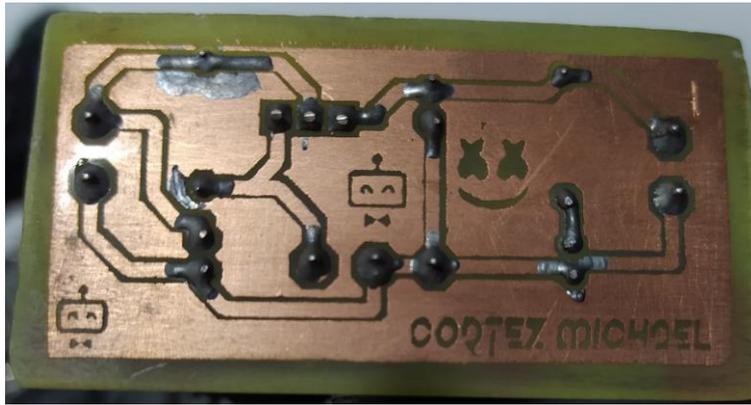
Anexo 1 Proteus 8 Ventana Principal Circuito Eléctrico Fuente Variable 12V

Fuente: Propia



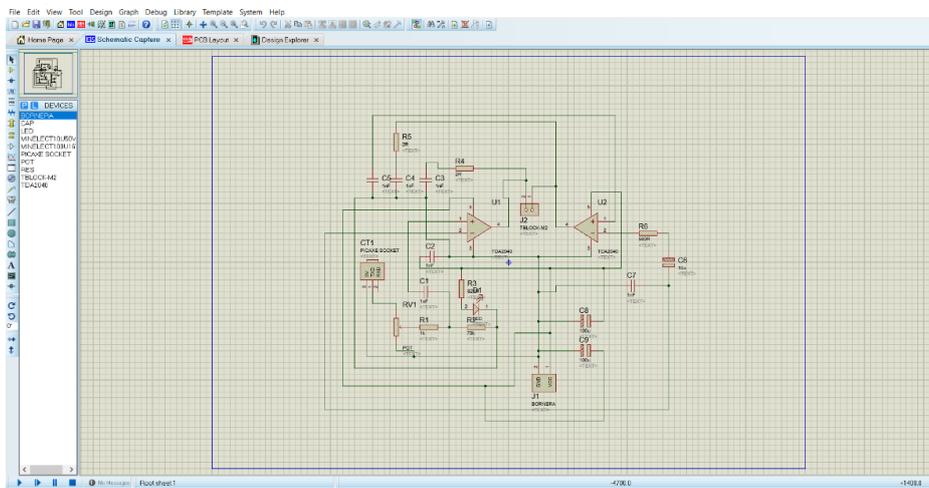
Anexo 2 Proteus 8 ISIS Ventana De Diseño Circuito Impreso Fuente Variable 12V

Fuente: Propia



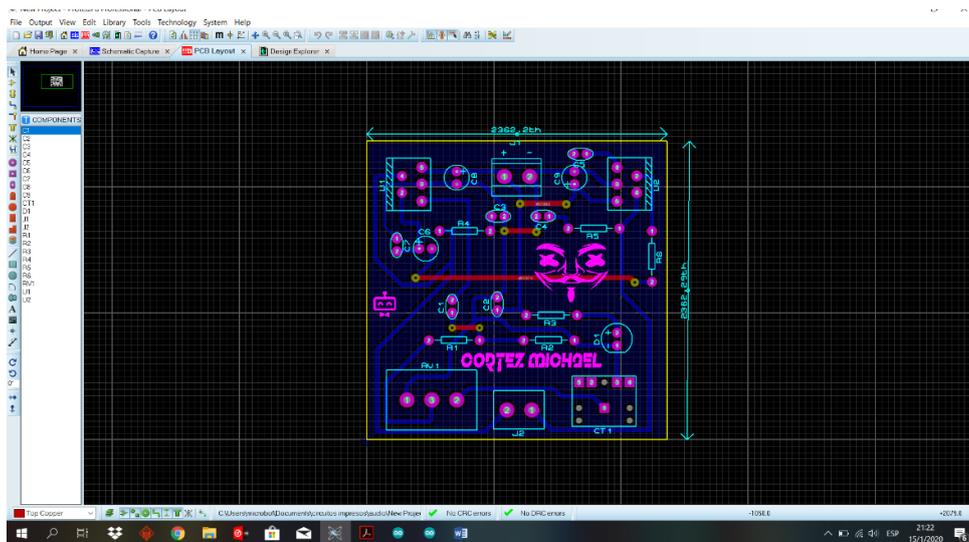
Anexo 3 Circuito Impreso

Fuente: Propia



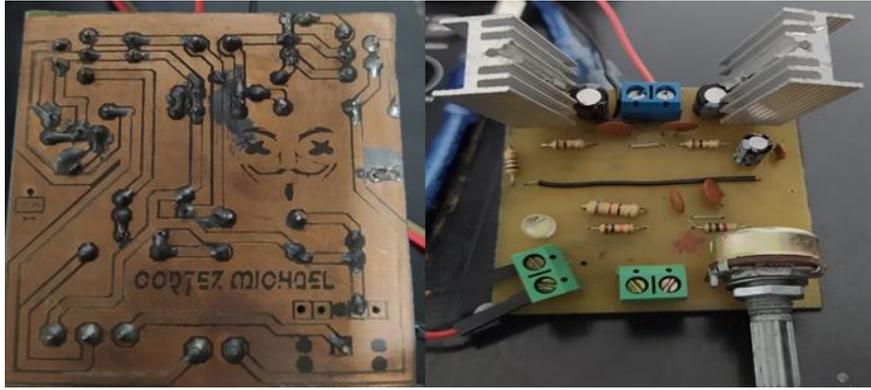
Anexo 4 Proteus 8 Ventana Principal Circuito Eléctrico Amplificador de 20W

Fuente: Propia



Anexo 5 Proteus 8 ISIS Ventana Principal Circuito Eléctrico Amplificador de 20W

Fuente: Propia



Anexo 6 Circuito Impreso Amplificador de 20W

Fuente: Propia

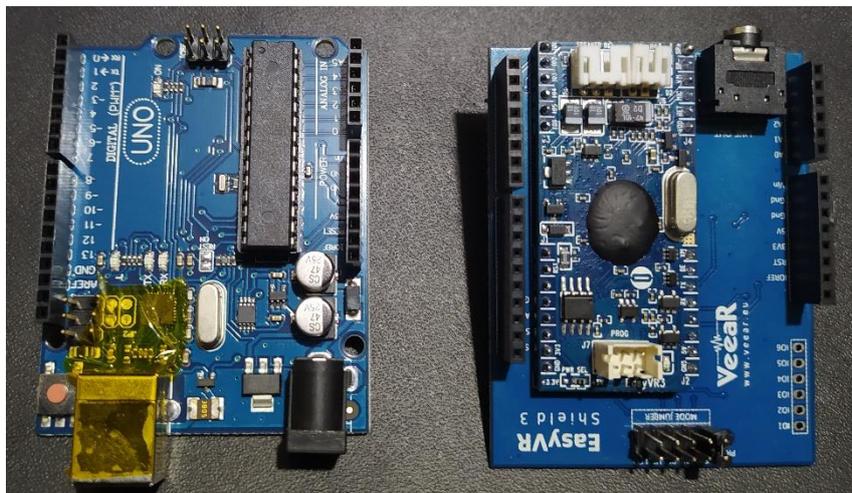
8.2. Servomotores



Anexo 7 Cabeza Robótica Servomotores

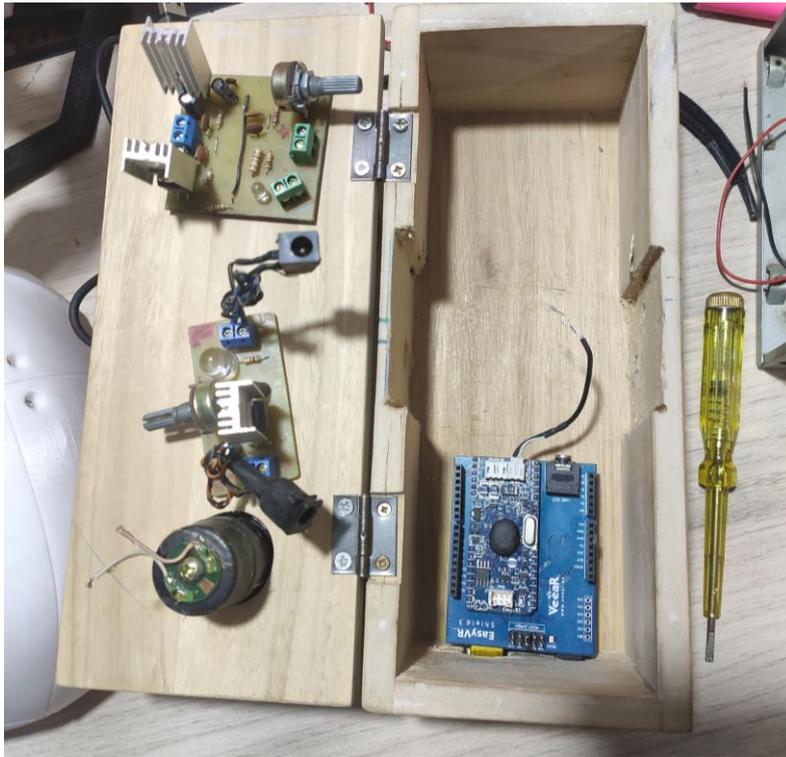
Fuente: Propia

8.3. Módulos



Anexo 8 Arduino y Easy VR Shield 3

Fuente: Propia



Anexo 9 Ensamblaje de Módulos

Fuente: Propia



Anexo 10 Cabeza Robótica

Fuente: Propia