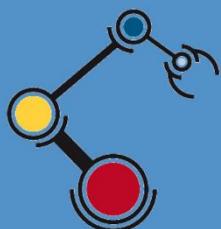




Escuela  
Politécnica  
Superior

# Diseño y programación de un robot humanoide de bajo coste



Máster Universitario en Automática  
y Robótica

## Trabajo Fin de Máster

Autor:  
Óscar Guillermo Hernández Ramírez

Tutores:  
Francisco Andrés Candelas Herías  
Carlos Alberto Jara Bravo

 Universitat d'Alacant  
Universidad de Alicante



## Resumen

Las técnicas de impresión 3D están facilitando el desarrollo de robótica de bajo coste, permitiendo el diseño y construcción de robots, aunque no son adecuados en entornos industriales o de producción; sin embargo, pueden ser usados en educación, entretenimiento, prototipos, sistemas de robótica sociales, entre otras aplicaciones. En este proyecto se detalla el proceso de construcción, programación y componentes involucrados para el control y puesta en marcha del robot humanoide Inmoov. Según (Langevin, 2016) el proyecto Inmoov es el primer robot humanoide de código abierto de tamaño real cuyas piezas pueden ser replicables con cualquier impresora 3D con un área de 12x12x12cm y es controlado mediante Arduino.

Entre los aspectos que diferencian este proyecto del proyecto Inmoov original están: control de objetivos, trayectorias, configuración de comandos de voz y audio del humanoide mediante una aplicación interactiva, persistente y escalable desde el ordenador, control de comando de voz desde el móvil mediante comunicación *bluetooth*.

## Palabras claves

Inmoov, robot humanoide, impresión 3D, Arduino, comando de voz.

## Abstract

3D printing techniques are facilitating the development of low-cost robotics, enabling the design and construction of robots, although, they are not suitable for industrial or production environments; however, they can be used in education, entertainment, as prototypes, in social robotics systems, among other applications.

The construction process, programming and the components involved in the control and the execution of the Inmoov humanoid robot it's detailed in these project. According to (Langevin, 2016) the Inmoov project is the first real size, open source humanoid robot whose pieces can be replicated with any 3D printer with an area of 12x12x12cm and is controlled by Arduino.

Among the aspects that differentiate this project from the original Inmoov project are: the control objectives, the paths, the settings of the humanoid voice and audio commands through an interactive, persistent and scalable application from the computer, also, the control of the voice command from the mobile through communication bluetooth.

## Keywords

Inmoov, humanoid robot, 3D printing, Arduino, voice command.

## Justificación

La facilidad de acceso a las diferentes técnicas de impresión 3d ha permitido el avance y desarrollo en la industria de la robótica. Tener conocimiento de modelado 3d junto con una impresora 3d permite replicar infinidad de objetos en diversos materiales a un coste cada vez más bajo. El proyecto Inmoov es el primer robot humanoide de tamaño real creado con impresora 3d y cuyo diseño de piezas es libre y se puede acceder mediante el sitio oficial: [www.inmoov.fr](http://www.inmoov.fr).

El recibir la preparación académica en un máster de automática y robótica y poder practicar temas de montaje, modificación de servomotores, control de sensores y programación de trayectorias con el diseño de plataforma y controlador estable y con facilidad de poder escalar a futuras adecuaciones; todo esto motiva a querer iniciar un proyecto de esta magnitud y poder investigar sobre problemas o dificultades que puedan surgir durante el desarrollo. El completar un proyecto de este tipo se logra una apertura a muchas aplicaciones de robótica aplicada a temas de medicina, hogar, diversión, educación, prototipos, entre otros temas.

## Agradecimientos

- A la Universidad Nacional Autónoma de Honduras por brindar el apoyo económico mediante gestiones realizadas por las autoridades **Mónico Oyuela** y **Rutilia Calderón** ellos me han dado la oportunidad de estudiar en España y han estado pendiente de mis solicitudes y avances académicos.
  
- A mis supervisores **Francisco Andrés Candelas Herías** y **Carlos Alberto Jara Bravo** por estar siempre disponibles y aclarar las dudas durante el desarrollo del proyecto.
  
- A mis compañeros por compartir sus experiencias y ser parte de equipos de trabajos, cuyo esfuerzo grupal y dedicado ha ayudado a cumplir con éxitos esos pequeños objetivos, fortaleciendo valores de compañerismo, responsabilidad, dedicación y aplicación en el desarrollo de actividades.

## **Dedicatoria**

- A Dios por darme fuerza, fortaleza y sabiduría para iniciar y poder llegar a esta etapa importante en el proceso de formación profesional.
- A mi padre y madre por confiar en mí y estar conmigo siempre, brindando su apoyo, orientación, cercanía y motivación a seguir estudiando.
- A mis hermanos, mi novia y demás familiares por estar conmigo, por compartir momentos inolvidables que fortalecen como persona y me motiva a lograr un esfuerzo constante y aplicación para el cumplimiento de metas.

## Citas

*“Sabemos lo que somos, pero aún no sabemos lo que podemos llegar a ser.”*

*William Shakespeare*

# Índice

Resumen .....	3
Palabras claves .....	3
Abstract.....	3
Keywords .....	3
Justificación .....	4
Agradecimientos.....	5
Dedicatoria.....	6
Citas .....	7
Índice de figuras .....	10
Índice de tablas .....	11
I. Introducción .....	12
II. Estado del arte .....	13
2.1. ¿Qué es un robot humanoide?.....	13
2.2. Mecánica y programación del robot humanoide.....	13
2.3. Servomotores .....	14
2.3.1. Principios y funcionamiento.....	15
2.4. Arduino .....	16
2.4.1. ¿Qué es Arduino? .....	17
2.4.2. ¿Por qué Arduino? .....	17
2.4.3. Conexión Arduino-Servomotor .....	18
2.5. Módulos Arduino .....	20
1.6. Impresión 3D .....	21
1.7. Impresión 3D y robot humanoide .....	23
1.8. Proyecto Inmoov .....	24
2.8.1. Partes y montaje .....	25
1.9. Software .....	28

2.9.1. Cura .....	28
1.9.2. C# .....	28
1.9.3. SQL Server .....	28
III.    Objetivos.....	30
Objetivo General.....	30
Objetivos Específicos .....	30
IV.    Metodología.....	31
4.1.    Plan de trabajo y cronograma .....	31
V.    Desarrollo del Robot Humanoide.....	33
5.1.    Montaje del humanoide .....	33
5.1.1.    Punto de partida .....	33
5.1.2.    Impresión y acabado de piezas .....	34
5.1.3.    Componentes Electrónicos .....	35
5.1.4.    Modificación de servomotores .....	36
5.1.5.    Ensamblaje .....	37
5.1.6.    Pruebas y estabilización de montaje .....	41
5.2.    Programación y control de humanoide .....	42
5.2.1.    Plataforma de control desde el ordenador .....	42
5.2.2.    Controlador del Arduino.....	45
5.2.3.    Comunicación entre Arduino y Ordenador .....	48
5.2.4.    Comunicación entre Arduino y Móvil.....	51
5.3.    Pruebas y resultados.....	52
5.3.1.    Ejemplo de configuración del humanoide .....	53
5.4.    Trabajos futuros .....	59
VI.    Conclusiones.....	60
VII.   Bibliografía y referencias .....	61
VIII.  Anexos .....	62

8.1.	Manual de Administrador .....	62
8.1.1.	Software necesario.....	62
8.1.2.	Iniciar aplicación en el ordenador .....	62
8.2.	Manual de usuario.....	68
8.2.1.	Pasos iniciales.....	68
8.2.2.	Controlar movimientos del humanoide .....	68
8.2.3.	Creación de Objetivos y Trayectorias .....	69
8.2.4.	Comando de voz y audio .....	71
8.2.5.	Detector de proximidad .....	74
8.3.	Especificaciones Técnicas de los Servomotores.....	77
8.4.	Especificaciones Técnicas del Módulo MP3 Player Mini .....	78
8.5.	Especificaciones del Módulo Bluetooth HC-05 .....	81
8.6.	Especificaciones Técnicas del Sensor de Distancia HC-SR04 .....	84
8.7.	Audios pre-grabados .....	85

## Índice de figuras

Figura 1:	Robot ADAM-V1, tomada de (Constantin et al., 2015) .....	14
Figura 2:	Componentes de un servo: a) carcasa; b) motor DC; c) potenciómetro; d) circuito de control, Fuente: (Candelas & Corrales, 2007).....	15
Figura 3:	Colores de los cables de los servomotores. fuente (Candelas & Corrales, 2007) 15	
Figura 4:	Pulsos PWM para controlar servos, Fuente: (Candelas & Corrales, 2007) .....	16
Figura 5:	Arduino MEGA ADK, Fuente: (Arduino, 2016).....	16
Figura 6:	Conexión Arduino -Servomotor, Fuente: (Arduino, 2016).....	19
Figura 7:	Objet260 Connex, fuente: (Mariano, 2013) .....	22
Figura 8:	Proceso de diseño de partes, fuente (Constantin et al., 2015).....	24
Figura 9:	Inmoov, fuente (Langevin, 2016) .....	24
Figura 10:	Mano derecha, fuente: (Langevin, 2016) .....	25
Figura 11:	Antebrazo, fuente: (Langevin, 2016) .....	25
Figura 12:	Hombro, fuente: (Langevin, 2016) .....	26
Figura 13:	Bicep, fuente: (Langevin, 2016) .....	26
Figura 14:	Torso, fuente: (Langevin, 2016) .....	27
Figura 15:	Cabeza, fuente: (Langevin, 2016) .....	27
Figura 16:	Definición de actividades del proyecto.....	31

Figura 17: Diagrama de Gantt .....	32
Figura 18: Antebrazo y mano izquierda .....	33
Figura 19: Interfaz OctoPrint.....	34
Figura 20: Piezas del torso, bicep y hombro .....	34
Figura 21: Ubicación de componentes .....	36
Figura 22: Extracción de potenciómetro del servomotor .....	36
Figura 23: Potenciómetro externo .....	37
Figura 24: Ensamblaje torso y bicep .....	37
Figura 25: Base del humanoide .....	38
Figura 26: Mecánica de la cabeza.....	38
Figura 27: Instalación de altavoces .....	39
Figura 28: Montaje final humanoide .....	40
Figura 29: Interfaz de usuario.....	43
Figura 30: Modelo relacional base de datos Inmoov, fuente: elaboración propia.....	44
Figura 31: Controlador Arduino, fuente: elaboración propia .....	47
Figura 32: Comunicación componentes, fuente: elaboración propia .....	48
Figura 33: Aplicación AMR_Voice Android .....	51
Figura 34: Articulaciones humanoide InMoov.....	68
Figura 35: Diagrama comando de voz y audio, fuente: elaboración propia.....	71
Figura 36: Diagrama detector de proximidad, fuente: elaboración propia .....	74
Figura 37: Módulo MP3 Player Mini, fuente: (DFRobot, 2016) .....	78
Figura 38: Diagrama de conexión: módulo, Arduino y altavoces, fuente: (DFRobot, 2016) .....	79
Figura 39: Ejemplo de nombre de audio, fuente: (DFRobot, 2016).....	80
Figura 40: Conexión Arduino Mega-Bluetooth HC-05, fuente: (Adrian, 2015).....	83
Figura 41: Sensor de Distancia de Ultrasonido HC-SR04 .....	84

## Índice de tablas

Tabla 2: Componentes electrónicos.....	35
Tabla 3: Descripción de tablas de base de datos .....	45
Tabla 4: Relación articulación-pin Arduino .....	46
Tabla 5: Definición de tramas .....	50
Tabla 6: Pruebas y resultados .....	53
Tabla 7: Definición de órdenes mediante comando de voz.....	54
Tabla 8: Definición de órdenes mediante detector de proximidad.....	54
Tabla 9: Identificación de objetivos .....	55
Tabla 1: Pinout, mp3 player mini, fuente: (DFRobot, 2016) .....	79
Tabla 10: Audios pre-grabados.....	85

## I. Introducción

La disponibilidad de las distintas técnicas de impresión 3D abre camino en el desarrollo de la industria de la robótica. Las impresoras 3D son utilizadas para replicar infinidad de objetos en diversos materiales a un coste cada vez más bajo y, de esta manera, quedan accesibles para las personas interesadas en temas de investigación de diseño, construcción, control y programación de robots. Aunque este tipo de robots no son apropiados para ambientes industriales y no son capaces de usarse en líneas de producción, son una buena elección para emplearlos en educación, entretenimiento, prototipos, sistemas de robótica sociales, entre otras aplicaciones.

Se muestra a continuación de forma detallada el desarrollo de proyecto consistente en el diseño y construcción de un humanoide basado en el proyecto Inmoov ([www.inmoov.fr](http://www.inmoov.fr)), para una mejor organización se ha estructurado el documento en: *estado del arte* donde se muestran las referencias bibliográficas que ayudaran a reforzar la parte teórica-práctica del proyecto, *objetivos* mostrando el objetivo general y objetivos específicos que se plantean como alcance del proyecto, *metodología* detallando la forma de trabajo, *desarrollo del robot humanoide* es la parte principal del proyecto donde se muestra todo el desarrollo del mismo, así como trabajos futuros, *conclusiones* donde se detallan las principales conclusiones del proyecto y por último las referencias bibliográficas y anexos, en esta última parte se muestra información adicional lo cual es importante recalcar del proyecto, entre ellos: el manual de usuario y administrador de la plataforma para el control del humanoide desde el ordenador y móvil.

## **II. Estado del arte**

### **2.1. ¿Qué es un robot humanoide?**

Según la Real Academia Española, define a término humanoide como: “Que tiene forma o características del ser humano”.

Desde el punto de vista de investigadores podemos mencionar a:

Según (Constantin, Marcoianu, Voiniciuc, Vinturis, & Ieee, 2015):

Un humanoide es una especie de robot que tiene apariencia y funciones como las de un ser humano. Independientemente de la finalidad para la que el humanoide fue creado ocurre problemas comunes tratados en robótica como ser: la percepción, el procesamiento de la información y la capacidad de tomar decisiones con la información adquirida. Se espera que en el futuro los robots humanoides sirvan como acompañantes, asistentes personales en la vida cotidiana y como reemplazo de personal en tareas que sería peligroso para el ser humano.

### **2.2. Mecánica y programación del robot humanoide**

En (Constantin et al., 2015) se menciona que la estructura general de un robot humanoide está compuesto por: torso, cabeza, dos brazos y dos piernas, aunque algunos tipos de humanoides pueden tener sólo una parte del cuerpo, por lo general de cintura para arriba. El mecanismo del cuerpo humano está compuesto de cartílagos y huesos en las articulaciones como rígido elementos y los músculos y tendones para movimientos. El rendimiento de los componentes mecánicos de hoy en día no puede igualar el rendimiento del cuerpo humano y en este caso es imposible reemplazar todos los componentes del sistema músculo-esquelético con componentes mecánicos actuales. En la figura 1 se muestra el humanoide ADAM-V1, las partes de este proyecto han sido construidas con impresoras 3D lo que ayuda en reducción de costes y peso.



Figura 1: Robot ADAM-V1, tomada de (Constantin et al., 2015)

### 2.3. Servomotores

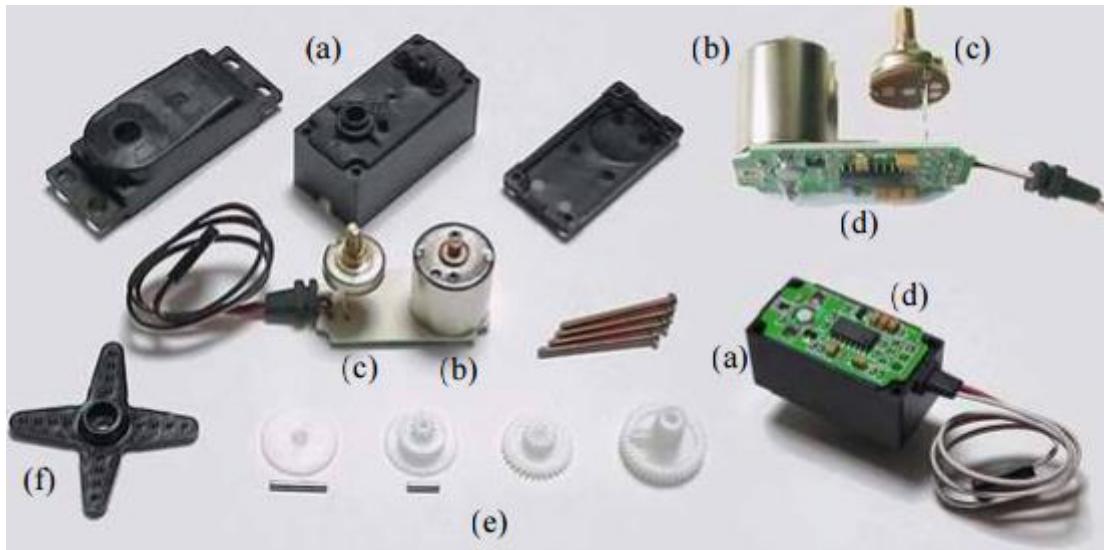
En el artículo (Candelas & Corrales, 2007), se describe un servomotor (o servo) como un motor de corriente continua o alterna que tiene la capacidad de ser controlado en posición. Es capaz de ubicarse en cualquier posición dentro de un rango de operación (generalmente de 180°) y mantenerse estable en dicha posición. Los servos se suelen utilizar en robótica, automática y modelismo (vehículos por radio-control) debido a su gran precisión en el posicionamiento.

Según (Neoteo, 2016), los servomotores son dispositivos considerados una de las materias primas en el diseño y la construcción de los robots. Si combinamos y administrámos los movimientos en un montaje mecánico adecuado, un grupo variable de servomotores puede dar motricidad y locomoción a sistemas controlados de manera local o remota. Desde pequeñas aplicaciones didácticas hasta el más complejo diseño robótico.

En general, los servos suelen estar compuestos por 4 elementos fundamentales:

- *Motor de corriente continua (DC)*: Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima. Si el voltaje aplicado a sus dos terminales es inverso, el sentido de giro también se invierte.
- *Engranajes reductores*: Tren de engranajes que se encarga de reducir la alta velocidad de giro del motor para acrecentar su capacidad de torque (o par-motor).
- *Sensor de desplazamiento*: Puede ser un potenciómetro colocado en el eje de salida del servo que se utiliza para conocer la posición angular del motor.

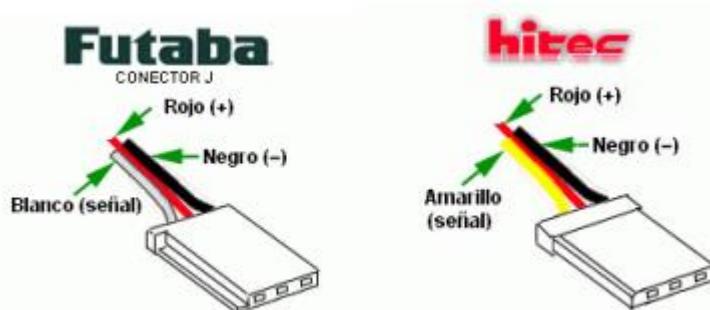
- *Circuito de control:* Es una placa electrónica que implementa una estrategia de control de la posición por realimentación. Para ello, este circuito compara la señal de entrada de referencia (posición deseada) con la posición actual medida por el potenciómetro. La diferencia entre la posición actual y la deseada es amplificada y utilizada para mover el motor en la dirección necesaria para reducir el error.



**Figura 2:** Componentes de un servo: a) carcasa; b) motor DC; c) potenciómetro; d) circuito de control, Fuente: (Candelas & Corrales, 2007)

### 2.3.1. Principios y funcionamiento

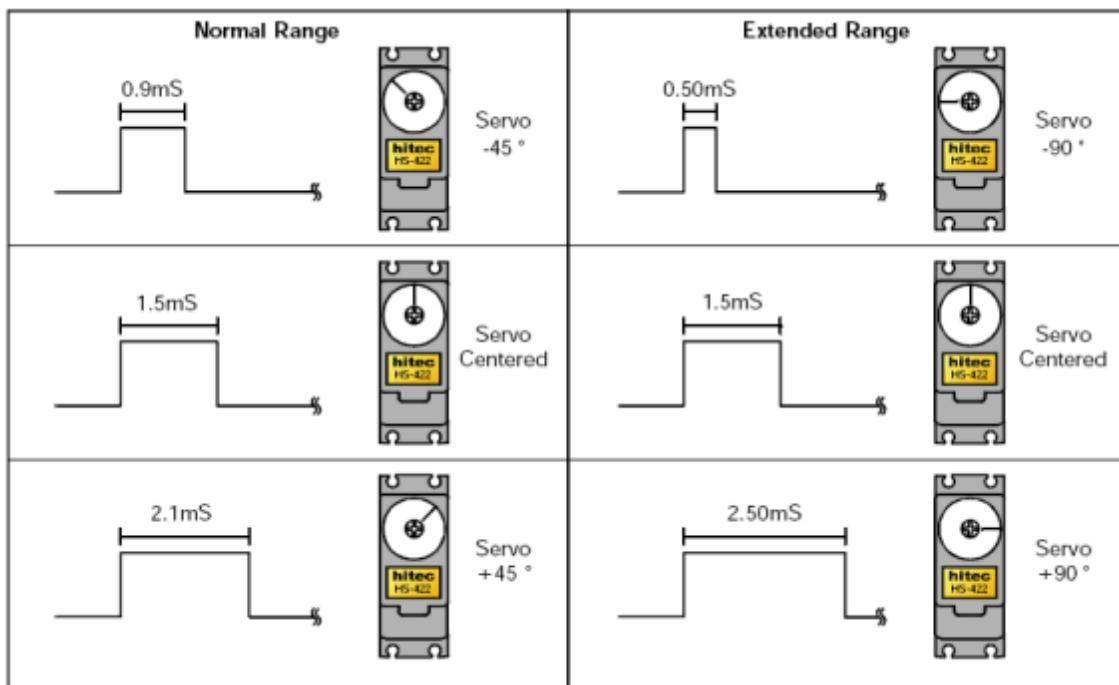
Los servos disponen de tres cables (Figura 3): dos cables de alimentación (positivo y negativo/masa) que suministran un voltaje 4.8-6V y un cable de control que indica la posición deseada al circuito de control mediante señales PWM (“*Pulse Width Modulation*”).



**Figura 3:** Colores de los cables de los servomotores. fuente (Candelas & Corrales, 2007)

Las señales PWM utilizadas para controlar los servos están formadas por pulsos positivos cuya duración es proporcional a la posición deseada del servo y que se repiten cada 20ms

(50Hz). Todos los servos pueden funcionar correctamente en un rango de movimiento de 90°, que se corresponde con pulsos PWM comprendidos entre 0.9 y 2.1ms. Sin embargo, también existen servos que se pueden mover en un rango extendido de 180° y sus pulsos de control varían entre 0.5 y 2.5ms (Figura 4). Antes de utilizar un servo habrá que comprobar experimentalmente su rango de movimiento para no dañarlo. Para mantener fijo un servo en una posición habrá que enviar periódicamente el pulso correspondiente; ya que si no recibe señales, el eje del servo quedará libre y se podrá mover ejerciendo una leve presión.



**Figura 4:** Pulsos PWM para controlar servos, Fuente: (Candelas & Corrales, 2007)

## 2.4. Arduino

Una de las plataformas de hardware abierto más utilizado es Arduino, a continuación se detallan algunos puntos importantes de esta plataforma, información obtenida de (Arduino, 2016):



**Figura 5:** Arduino MEGA ADK, Fuente: (Arduino, 2016)

#### **2.4.1. ¿Qué es Arduino?**

Arduino es una plataforma de código abierto (*open-source*) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros dispositivos. El microcontrolador de la placa se programa usando el “*Arduino Programming Language*” (basado en *Wiring*) y el “*Arduino Development Environment*” (basado en *Processing*). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.).

Las placas se pueden ensamblar a mano o encargarlas preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia *open-source*, por lo que eres libre de adaptarlas a tus necesidades. Arduino recibió una mención honorífica en la sección *Digital Communities del Ars Electronica Prix en 2006*.

#### **2.4.2. ¿Por qué Arduino?**

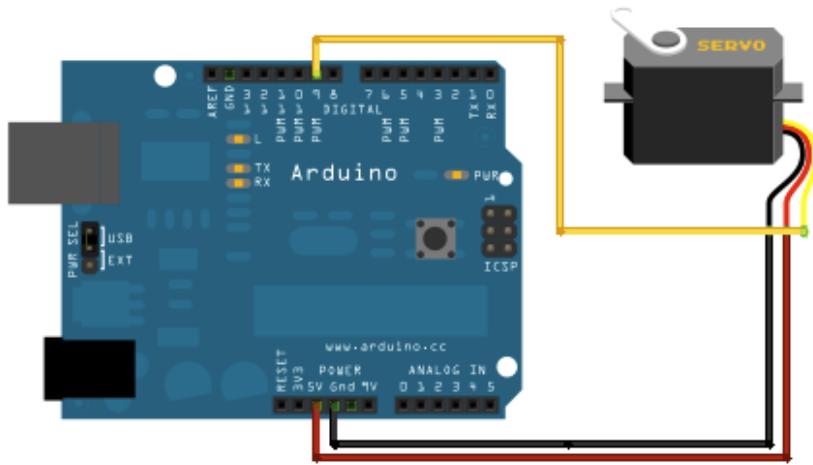
Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación física. *Parallax Basic Stamp*, *Netmedia's BX-24*, *Phidgets*, *MIT's Handyboard*, y muchas otras ofertas de funcionalidad similar. Todas estas herramientas toman los desordenados detalles de la programación de microcontrolador y la encierran en un paquete fácil de usar. Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas:

- **Económico:** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.

- **Entorno de programación simple y fácil de utilizar:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación *Processing*, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia *Creative Commons*, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.

#### **2.4.3. Conexión Arduino-Servomotor**

Según (Arduino, 2016), el cable de alimentación es típicamente rojo, y debe ser conectado al pin 5V de la placa Arduino o Genuino. El cable de tierra es por lo general de color negro o marrón y debe estar conectado a un conector de tierra en el tablero. El pasador de señal es normalmente de color amarillo o naranja y debe ser conectado al pin PWM.



**Figura 6:** Conexión Arduino -Servomotor, Fuente: (Arduino, 2016)

### Ejemplo control de servo mediante Arduino

```
#include <Servo.h>
Servo myservo; // Crear el objeto servo
int pos = 0; // Variable para controlar la posición del servo
void setup() {
    myservo.attach(9); //Attach el servo en pin 9
}

void loop() {
    for (pos = 0; pos <= 180; pos += 1) {
        myservo.write(pos); // Escribir posición en el servo
        delay(15); // Espera
    }
    for (pos = 180; pos >= 0; pos -= 1) {
        myservo.write(pos);
        delay(15);
    }
}
```

## 2.5. Módulos Arduino

### 2.5.1. Módulo MP3 Player Mini

Según (DFRobot, 2016), El *DFPlayer Mini* es un pequeño módulo MP3 de bajo costo que cuenta con una salida directa para un altavoz. Puede ser utilizado como un módulo independiente con baterías y pulsadores o también en combinación con un Arduino UNO u otros microcontroladores que soporten comunicación por RX/TX. Soporta formatos estándares de audio como MP3, WAV y WMA. Asimismo soporta tarjetas microSD con sistemas de archivos FAT16, FAT32. A través de una simple comunicación serial puedes comandar la reproducción de una pista en particular sin necesidad de muchas líneas de código.

#### Aplicaciones

- Reproducción de audio en sistemas de navegación.
- Reproducción de sonidos pregrabados en sistemas automáticos estación de buses.
- Sistemas de guía vehicular, retroceso, etc.
- Mensajes de bienvenida, atención al cliente.
- Sonidos para fallos de sistemas.
- Alarmas, entre otras aplicaciones.

Para visualizar mayor información del módulo, consultar la sección [8.4 Especificaciones Técnicas del Módulo MP3 Player Mini](#).

### 1.5.2. Módulo Bluetooth HC-05

El módulo HC-05 es utilizado para la trasferencia de información mediante comunicación *bluetooth*, entre las aplicaciones se encuentran:

- Comunicación inalámbrica entre microcontroladores.
- Comunicación inalámbrica entre computadoras y microcontroladores.
- Comunicación inalámbrica entre teléfonos móviles o tabletas y microcontroladores.

Para consultar mayor información del módulo *bluetooth* HC-05 consultar la sección [8.5 Especificaciones del Módulo Bluetooth HC-05](#).

### **1.5.3. Sensor de Distancia de Ultrasonido HC-SR04**

Según información obtenida de (Electronilab, 2016), el módulo HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots recientes. Es de fácil uso y programación con las placas de Arduino y microcontroladores.

Para consultar mayor información del sensor de distancia consultar la sección [8.6 Especificaciones Técnicas del Sensor de Distancia HC-SR04](#).

## **1.6. Impresión 3D**

Según (Mariano, 2013), la impresión 3D es un proceso que permite hacer objetos tridimensionales sólidos a partir de un modelo digital. La impresión 3D se consigue utilizando procesos aditivos (con aporte de material), donde se crea un objeto mediante las capas sucesivas de material. La impresión 3D difiere de los tradicionales mecanizados (técnicas de procesos sustractivos), que se basan principalmente en la eliminación de material por métodos tales como fresado, torneado, corte y perforación. Los procesos aditivos se realizan, generalmente, por una impresora de materiales utilizando la tecnología digital. Desde el inicio del siglo XXI ha habido un gran crecimiento en las ventas de estas máquinas, y su precio ha bajado considerablemente.

La tecnología se utiliza en joyería, calzado, diseño industrial, arquitectura, construcción, industria automotriz, ingeniería aeroespacial, industrias médicas y dentales, educación, sistemas de información geográfica, ingeniería civil, y muchos otros campos.

### **Etapas del proceso**

#### **1. Modelado**

Para la impresión 3D existente herramientas para el diseño asistido por ordenador (CAD) además de entornos virtuales que son guías para la impresión. Dependiendo de la máquina que se utiliza, el material o vinculante se deposita sobre el lecho de compilación o de la

plataforma hasta que la capa de material/aglutinante se completa y el modelo 3D final ha sido "impreso".

Los datos estándar de interfaz entre el software CAD y las máquinas de impresión es el formato de archivo STL. Un archivo STL se aproxima a la forma de una pieza o un ensamblaje con facetas triangulares. Pequeñas facetas producen una superficie de mayor calidad. El PLY es un formato de archivo de entrada generado por escáner, y archivos VRML (o WRL) se utilizan a menudo como entrada para las tecnologías de impresión 3D que son capaces de imprimir a color.

## 2. Impresión

Para realizar una impresión, la máquina lee el diseño y establece las capas sucesivas de líquido, polvo o material de lámina para construir el modelo a partir de una serie de secciones transversales. Estas capas, que corresponden a las secciones transversales virtuales a partir del modelo CAD, se unen o fusionan automáticamente para crear la forma final. La principal ventaja de esta técnica es su capacidad de crear casi cualquier forma o característica geométrica.

La resolución de la impresora describe espesor de la capa y la resolución X-Y en dpi (puntos por pulgada), o micrómetros. El espesor típico de la capa es de alrededor de 100 micras (0,1 mm), aunque algunas máquinas, tales como la *Objet Connex* y la *3D Systems ProJet* pueden imprimir capas tan delgadas como 16 micrómetros. La resolución XY es comparable a la de las impresoras láser. Las partículas (puntos 3D) son de alrededor de 50 a 100 micrómetros (0,05-0,1 mm) de diámetro.



Figura 7: Objet260 Connex, fuente: (Mariano, 2013)

La construcción de un modelo con métodos contemporáneos puede tardar de varias horas a varios días, dependiendo del método utilizado, el tamaño y la complejidad del modelo, tipo de máquina utilizada y del tamaño y número de los modelos de fabricación simultánea.

### 3. Acabado

Aunque la resolución del impreso producido es suficiente para muchas aplicaciones, en muchas ocasiones es necesario retirar pequeños fragmentos y pulir las piezas. Algunas técnicas de fabricación son capaces de utilizar múltiples materiales en el curso de la construcción de las piezas. Algunos también utilizan soportes durante la construcción. Los soportes son extraíble o soluble tras la finalización de la impresión, y se utilizan para apoyar voladizos durante la construcción.

#### 1.7. Impresión 3D y robot humanoide

De acuerdo con (Bestmann, Reichardt, & Wasserfall, 2015) el cual presenta un robot, llamado *hambot*, para competición de fútbol en este artículo se menciona que para lograr una reducción de costos al momento de construir un humanoide, es tomar en cuenta que las piezas pueden ser elaboradas con hojas de aluminio y plástico mediante impresión 3D, sin embargo, las hojas de aluminio fue utilizado en la primera versión, en esta versión el robot fue capaz de soporta y caminar, pero era incapaz de levantarse, en la segunda y actual versión el prototipo fue fabricado utilizando impresión 3D, con este diseño se logró la producción de piezas complejas que encajen exactamente entre sí, todas las piezas del robot fueron diseñadas con un máximo de 20x20x10 cm para que fueran imprimible con una impresora de consumo de bajo costo de Modelado por Deposición Fundida (FDM) utilizando materiales como el Acrilonitrilo Butadieno Estireno (ABS) y Ácido Poliláctico

Otro proyecto robot humanoide con impresora 3D detallado en el artículo (Constantin et al., 2015), el cual es llamado *ADAM-VI*, El proceso de todos los módulos del robot es: el diseño en 3D (*SolidWorks*, *3D Max Studio*, *OpenSCAD*, *SketchUp*), exportar en archivo con extensión stl, transformarse en código G (el lenguaje de control numérico más utilizado) con el programa CURA, la impresión con la impresora 3D y finalmente el montaje de todas las partes. La fabricación de piezas con tecnología de impresión 3D fabricación por adición, se requiere un período de sucesivas pruebas para determinar el tratamiento óptimo sistemas de temperatura y la capacidad de enfriamiento.

De acuerdo con (Constantin et al., 2015) el proceso para el diseño de las partes mecánicas del robot humanoide es:

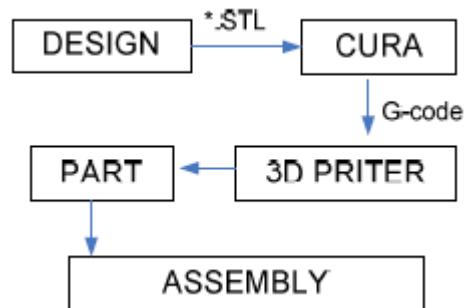


Figura 8: Proceso de diseño de partes, fuente (Constantin et al., 2015).

### 1.8. Proyecto Inmoov

De acuerdo con (Langevin, 2016) "Inmoov", es el primer robot humanoide de tamaño real de código abierto que se puede imprimir en 3D.

Gael Langevin es un escultor y diseñador francés, trabaja para las marcas más importantes desde hace más de 25 años. Inmoov es su proyecto personal, que se inició en enero de 2012 como el primer *Open Source* para prótesis de mano, ha dado lugar a proyectos como Biónico, E-Nable, y muchos otros. Este proyecto es replicable en cualquier impresora 3D en casa con un área de 12x12x12cm, se concibe como una plataforma de desarrollo para universidades, laboratorios.



Figura 9: Inmoov, fuente (Langevin, 2016)

### 2.8.1. Partes y montaje

A continuación se muestran el conjunto de piezas y partes principales del proyecto Inmoov, obtenido de (Langevin, 2016):

Los movimientos de los dedos de las manos son controlados mediante dos hilos plásticos que actúan como tendones con extremo común a la yema del dedo y el otro extremo es unido a la polea controlada por el servomotor cuya rotación tensa uno u otro hilo. En la figura 10 se muestra las partes del dedo pulgar.

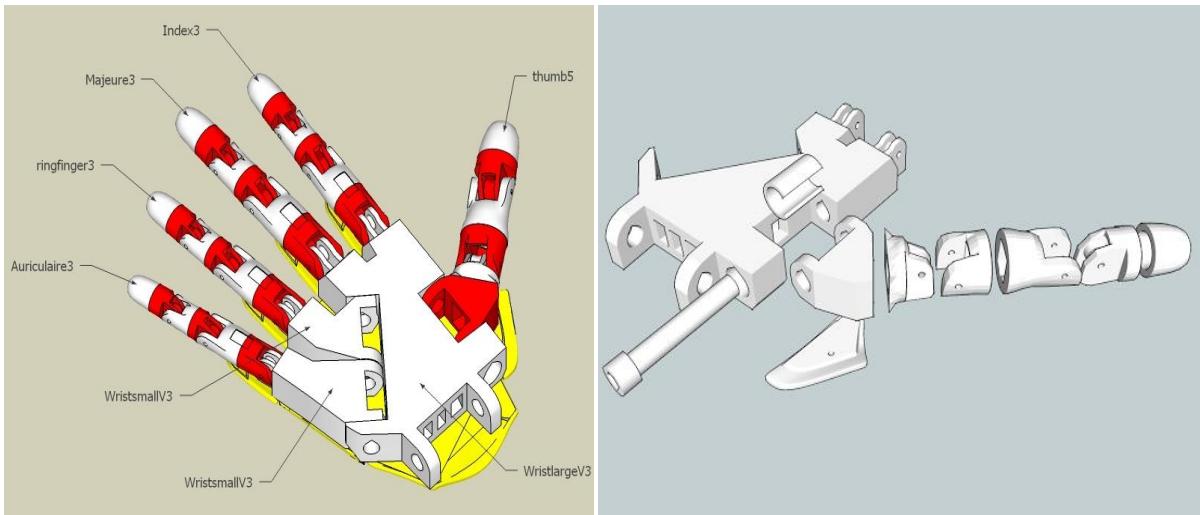


Figura 10: Mano derecha, fuente: (Langevin, 2016)

En la sección izquierda de la figura 11 se muestra las partes y engranajes que controlan los movimientos de la muñeca, en la sección derecha se muestra la ubicación de los servomotores, hilos plásticos y poleas que controlan los movimientos de los dedos y muñeca.

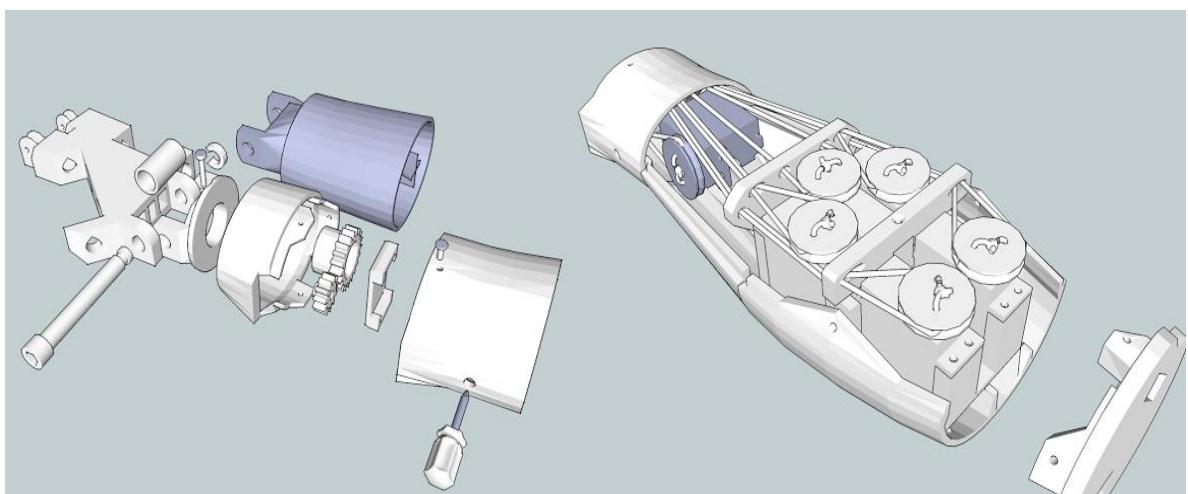
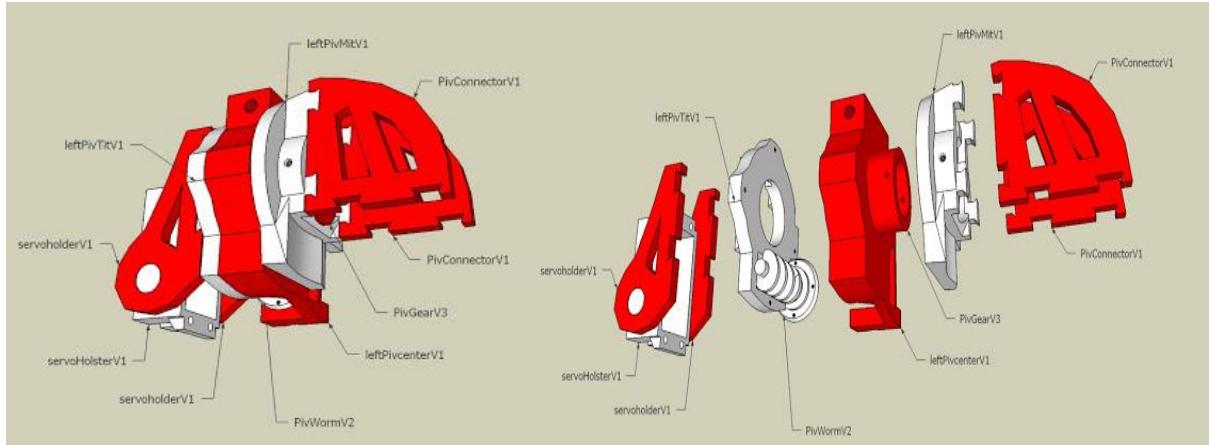


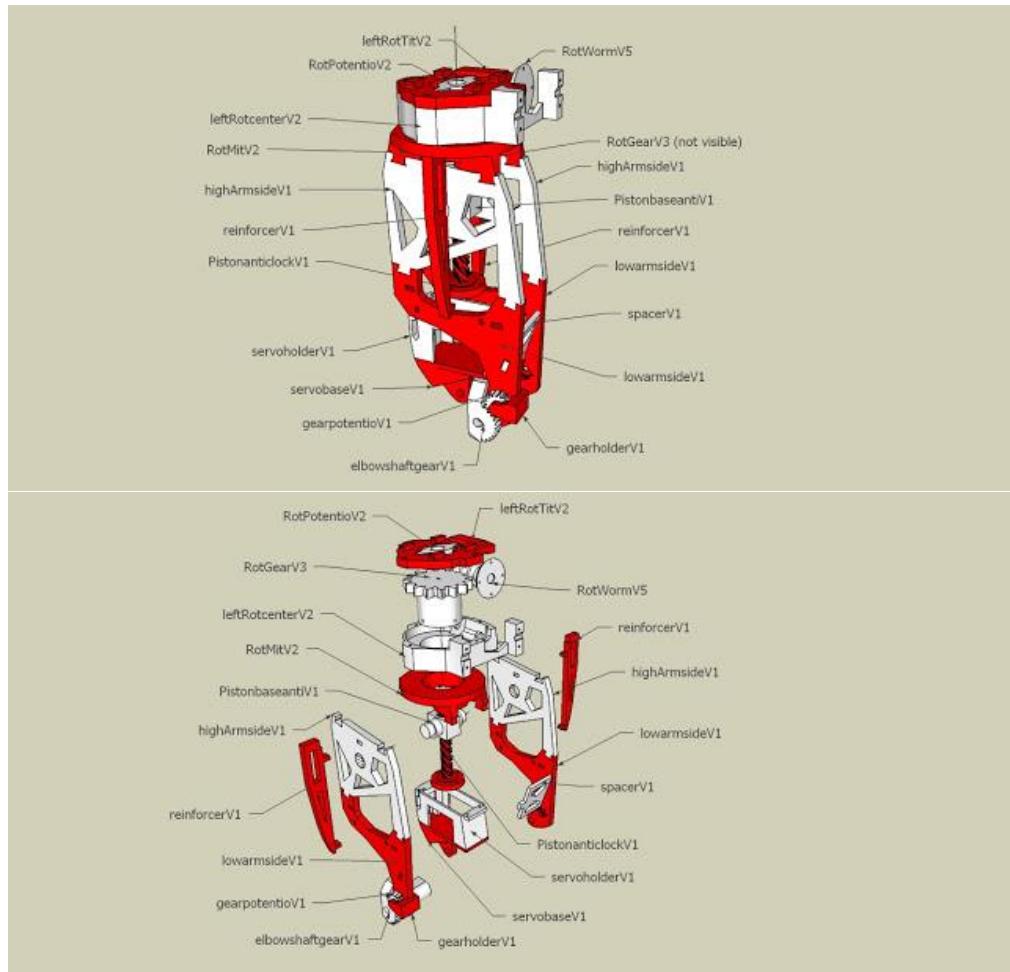
Figura 11: Antebrazo, fuente: (Langevin, 2016)

El hombro puede realizar dos movimientos: extensión lateral, frontal. Para controlar estos movimientos se utilizan engranajes para multiplicación del par. La figura 12 muestra las partes impresas necesarias para controlar los movimientos del hombro:



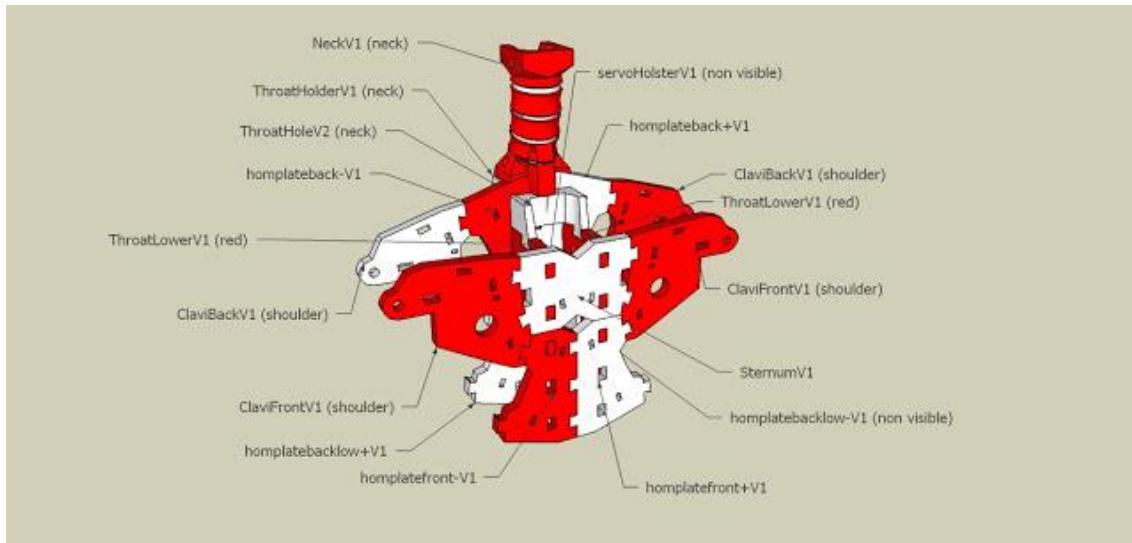
**Figura 12: Hombro, fuente: (Langevin, 2016)**

La figura 13 muestra las piezas y engranajes necesarios para poder realizar los movimientos: rotación del bicep y articulaciones del codo:



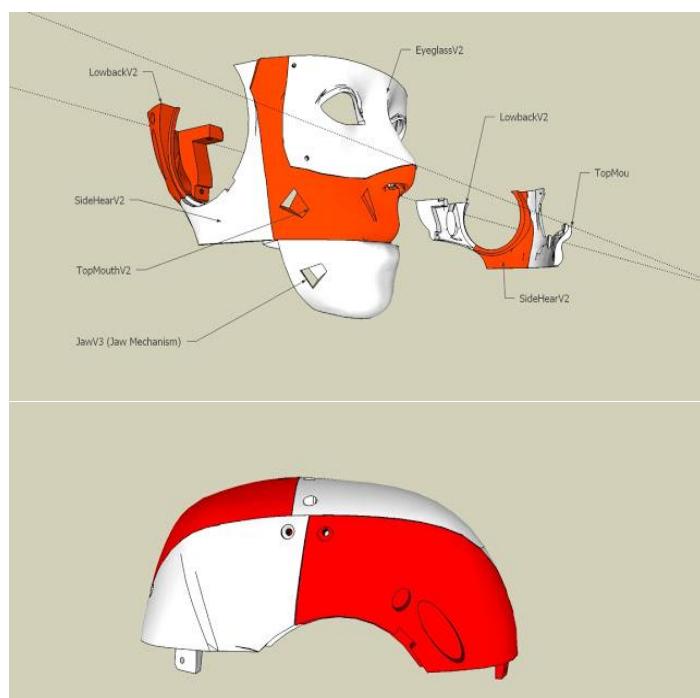
**Figura 13: Bicep, fuente: (Langevin, 2016)**

El torso está formado por un conjunto de piezas entrelazadas cuyo diseño permite la ubicación de engranajes y servomotores que controlan los movimientos: extensión lateral de los hombros, movimiento arriba-abajo de la cabeza, además de la ubicación de piezas que dan forma a pecho, en la figura 14 se puede observar el conjunto de piezas que forma el torso y cuello.



**Figura 14: Torso, fuente: (Langevin, 2016)**

La cabeza tiene la mecánica necesaria para poder realizar los movimientos de izquierda-derecha, arriba-abajo, así como movimiento de la mandíbula. El espacio de las orejas se puede colocar altavoces o algún modelo de orejas. En la figura 15 se muestran las piezas necesarias para la cabeza.



**Figura 15: Cabeza, fuente: (Langevin, 2016)**

## 1.9. Software

### 2.9.1. Cura

Según (Pablo, 2012), *Cura* es un software que permite convertir los archivos STL que contienen nuestro diseño 3D en piezas físicas en un solo entorno de trabajo.

Originalmente, Cura se llamaba *SkeinPyPy*. Este nombre proviene de la combinación de *Skeinforge* con PyPy. Skeinforge es un software que realiza el corte de las piezas 3D en capas (*slicer*) generando el código para que nuestras impresoras lo comprendan. Sin embargo, *Skeinforge* es bastante lento y le puede llevar más de una hora “cortar” modelos complejos.

Una vez configurado el CURA con los parámetros adecuados de la impresora 3d a utilizar, podemos visualizar la pieza a imprimir y generar el GCode que será el archivo que cargaremos para imprimir nuestra pieza.

### 1.9.2. C#

De acuerdo con (Microsoft, 2016b), C# es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

Visual C# es una implementación del lenguaje C# de Microsoft. Visual Studio ofrece compatibilidad con Visual C# con un completo editor de código, un compilador, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y de fácil uso y otras herramientas. La biblioteca de clases de .NET Framework ofrece acceso a numerosos servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

### 1.9.3. SQL Server

Según (Microsoft, 2016a) El motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. El motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.

Use motor de base de datos para gestionar bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos. Se puede usar SQL Server Management Studio para administrar los objetos de bases de datos y SQL Server Profiler para capturar eventos de servidor.

### **III. Objetivos**

#### **Objetivo General**

- Montaje y control de un robot humanoide basado en hardware y software abierto.

#### **Objetivos Específicos**

- Control y configuración de movimientos de robot humanoide mediante el ordenador.
- Control del humanoide mediante comandos de voz, detección de objetos y simulación de voz mediante sonidos pregrabados.
- Creación de plataforma para ordenador que sea configurable y escalable que permita almacenar y ejecutar configuraciones del robot humanoide.

## IV. Metodología

Se comenzará estudiando el proyecto Inmoov para conocer qué se puede realizar con el robot. Posteriormente, programar los controladores del Arduino y el ordenador para poder realizar movimientos simples con las partes construidas. Finalmente se programarán algunas tareas globales, involucrando trayectorias, comando de voz, audio y utilizando diversas partes físicas y se evaluará el rendimiento del robot.

### 4.1. Plan de trabajo y cronograma

El plan de trabajo inicia con el estudio del proyecto Inmoov y análisis del trabajo previo, luego se muestran dos etapas principales que son el montaje del humanoide y la programación y control del humanoide, el montaje consiste en todo la parte mecánica que va desde la impresión de piezas, instalación y modificación de servomotores y la etapa de programación y control del humanoide está dividida en la plataforma que se utilizará desde el ordenador y el controlador del Arduino; por último se realizarán las pruebas finales y documentación del proyecto detallando todos los aspectos y referencias necesarias para trabajos futuros. En la figura 16 se muestran el detalle de las actividades, duración de cada una así como la secuencia de ejecución.

		Nombre	Duración	Inicio	Terminado	Predecesores
1		Análisis y estudio del proyecto inmoov	2 days	1/03/16 8:00	2/03/16 17:00	
2		Estudio de documentos proporcionados	2 days	3/03/16 8:00	4/03/16 17:00	1
3		▣ Montaje del Humanoide	34,5 days	7/03/16 8:00	22/04/16 13:00	1;2
4		Instalación de herramientas necesarias	1 day	7/03/16 8:00	7/03/16 17:00	
5		Configuración de perfil de impresión	0,5 days	8/03/16 8:00	8/03/16 13:00	4
6		Impresión de piezas	15 days	8/03/16 13:00	29/03/16 13:00	5
7		Acabado de piezas impresas	8 days	29/03/16 13:00	8/04/16 13:00	6
8		Modificación de servomotores	5 days	7/03/16 8:00	11/03/16 17:00	
9		Ensamblaje	10 days	8/04/16 13:00	22/04/16 13:00	7;8
10		Pruebas y estabilización de montaje	12 days	7/03/16 8:00	22/03/16 17:00	
11		▣ Programación y control del humanoide	31 days	22/04/16 13:00	6/06/16 13:00	3
12		▣ Creación de plataforma para ordenador	10 days	22/04/16 13:00	6/05/16 13:00	
13		Módulo de control de objetivos y trayectorias	3 days	22/04/16 13:00	27/04/16 13:00	
14		Módulo de comando de voz	2 days	27/04/16 13:00	29/04/16 13:00	13
15		Módulo de control de audio	2 days	29/04/16 13:00	3/05/16 13:00	14
16		Módulo para guardar y recuperar información en	3 days	3/05/16 13:00	6/05/16 13:00	15
17		▣ Creación de controlador de arduino	14 days	6/05/16 13:00	26/05/16 13:00	12
18		Control de servomotores, interpolación de movimiento	3 days	6/05/16 13:00	11/05/16 13:00	
19		Controlador para comando de voz, comunicación	4 days	11/05/16 13:00	17/05/16 13:00	18
20		Controlador para audio	2 days	17/05/16 13:00	19/05/16 13:00	19
21		Definición y procesamiento de tramas	5 days	19/05/16 13:00	26/05/16 13:00	20
22		Comunicación Arduino-Ordenador	3 days	26/05/16 13:00	31/05/16 13:00	12;17
23		Organización de cableado, PCB, Arduino	4 days	31/05/16 13:00	6/06/16 13:00	22
24		Pruebas funcionamiento	4 days	6/06/16 13:00	10/06/16 13:00	11
25		Documentación	3 days	10/06/16 13:00	15/06/16 13:00	24

Figura 16: Definición de actividades del proyecto

La figura 17 muestra el diagrama de Gantt el cual se puede observar la secuencia, tiempo de inicio, fin de las actividades así como la ruta crítica, el número a la izquierda indica el número de actividad detallado en la figura 16.

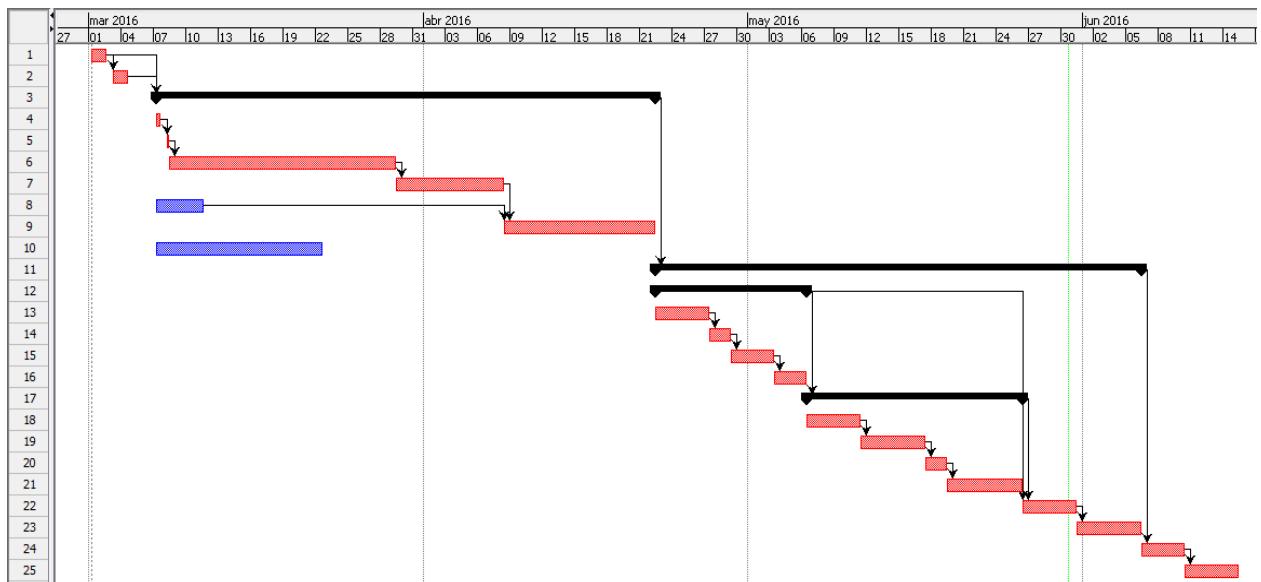


Figura 17: Diagrama de Gantt

## V. Desarrollo del Robot Humanoide

### 5.1. Montaje del humanoide

#### 5.1.1. Punto de partida

Entre los avances previos realizados en este proyecto están el montaje de las manos y antebrazos, este fue el punto de partida y es importante mencionar que se realizaron trabajos adicionales en las manos y antebrazos:

- Organización del cableado que controlan los servomotores.
- Mejoras en el acabado de antebrazos y manos, así como cambio de tornillos.
- Impresión y montaje de yemas de los dedos y parte frontal de las manos.
- Mejoras en las articulaciones de las muñecas y dedos.
- Calibración de servomotores.

La figura 18 muestra el antebrazo y mano izquierda, en las mismas condiciones se encontraba la mano y antebrazo derecho, junto con un conjunto de piezas impresas del torso, hombro y bicep (figura 20), fueron el punto de partida para el actual proyecto.



Figura 18: Antebrazo y mano izquierda

### 5.1.2. Impresión y acabado de piezas

Para la impresión de piezas se ha utilizado la herramienta *OctoPrint*, la cual permite el control y seguimiento de todos los aspectos de su impresora 3D y sus trabajos de impresión directamente desde un navegador web. La figura 19 muestra la interfaz *OctoPrint* donde se puede controlar los motores, ver estado de impresión, visualizar la pieza desde una cámara, entre otras funcionalidades.

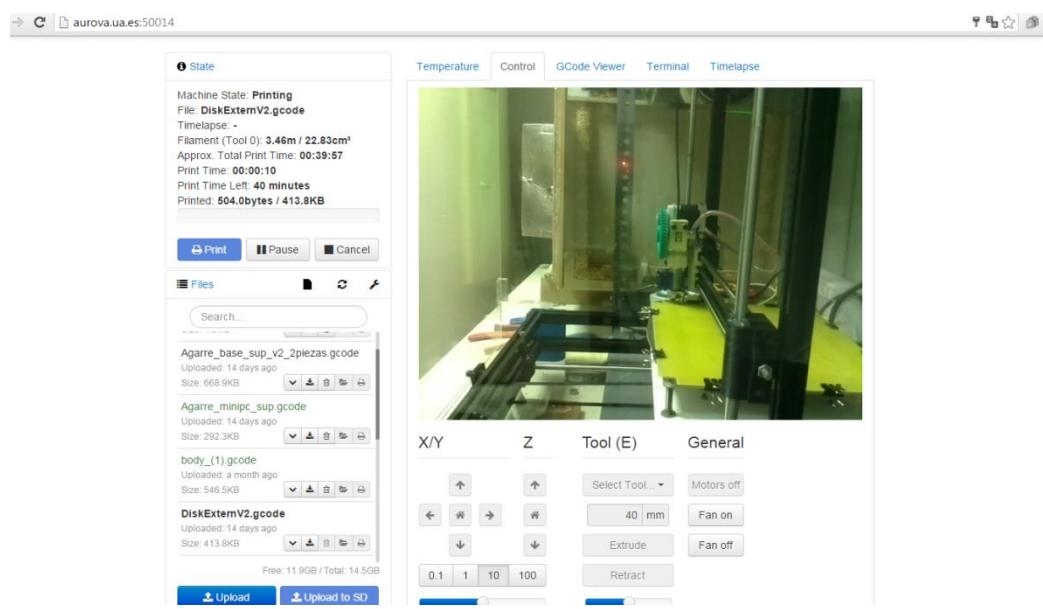


Figura 19: Interfaz OctoPrint

A continuación se muestra el conjunto de piezas del torso, bicep, hombro:



Figura 20: Piezas del torso, bicep y hombro

### 5.1.3. Componentes Electrónicos

A continuación se muestra los componentes electrónicos utilizados para el montaje del humanoide:

Descripción	Cantidad
Servomotor Power HD 1235 MG	10
Servomotor Power HD 1218 TH	13
Arduino MEGA	1
Altavoces 8 Ohm	2
Módulo Bluetooth HC-05	1
Módulo MP3 Player Mini	1
Placa PCB Arduino MEGA	1

Tabla 1: Componentes electrónicos

Para ver información se los Servomotores se puede consultar la sección [8.3 Especificaciones Técnicas de los Servomotores](#)

Las articulaciones con el servomotor *Power HD 1235 MG* son:

- Brazo izquierdo y derecho (8 servomotores en total):
  - Extensión frontal.
  - Extensión lateral.
  - Rotación del bicep.
  - Extensión del codo.
- Cabeza arriba-abaja (1 servomotor)
- Cabeza izquierda-derecha (1 servomotor)

Las articulaciones con el servomotor *Power HD 1218 TH*:

- Los diez dedos de las manos (10 servomotores).
- Muñeca izquierda y derecha (2 servomotores).
- Movimiento de la mandíbula (1 servomotor).

La figura 21 muestra la ubicación de componentes:

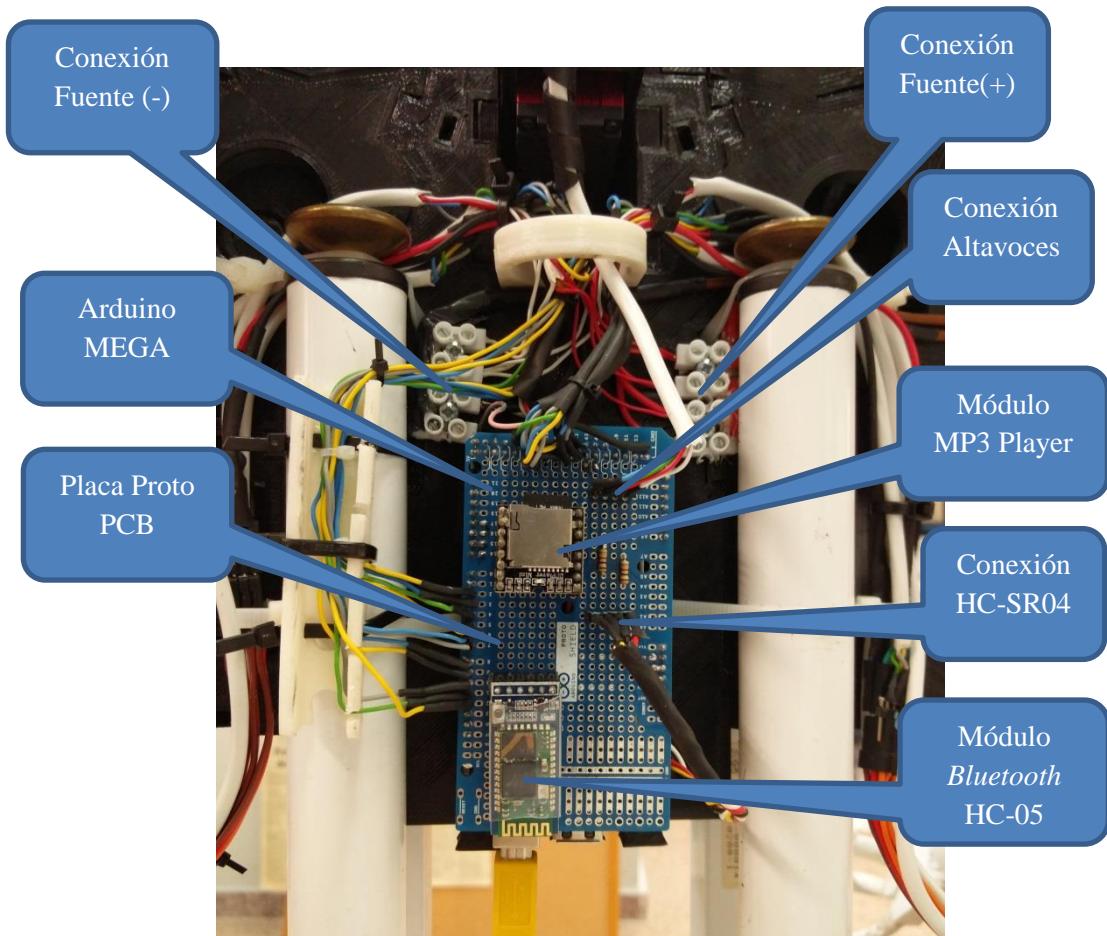


Figura 21: Ubicación de componentes

#### 5.1.4. Modificación de servomotores

Los servomotores utilizados pueden girar de 0 a 180 grados, lo cual si evaluamos cada una de las articulaciones del humanoide nos daríamos cuenta que con este giro podría ser suficiente, esto sería posible si las articulaciones fueran controladas directamente con el servomotor; sin embargo para el aumento del par existen un conjunto de engranajes que reducen el giro original, lo anterior obliga a realizar modificaciones de los servomotores consistentes en extraer el potenciómetro interno (ver figura 22).



Figura 22: Extracción de potenciómetro del servomotor

Una vez extraído el potenciómetro, se debe tener en cuenta lo siguiente:

- El potenciómetro debe ser ubicado en el eje de salida tras la reducción para poder medir la posición real.
- El cableado debe tener la longitud adecuada y debe ser ubicado cuidadosamente evitando interrupciones en el movimiento de las articulaciones.
- Una vez calibrado el ángulo máximo y mínimo permitido en el servomotor, se debe marcar la pieza y asegurar el potenciómetro, cualquier movimiento del potenciómetro puede ocasionar problemas graves ya que su calibración se ve afectada, ver figura 23.

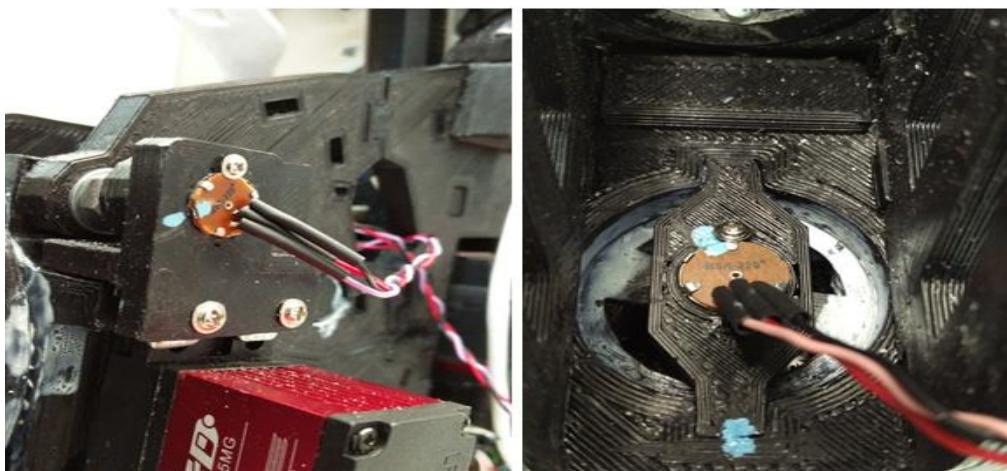


Figura 23: Potenciómetro externo

#### 5.1.5. Ensamblaje

Una vez impresas las piezas y realizado el acabado, procedemos a ensamblar las partes teniendo en cuenta que el acople sea lo más exacto posible para evitar vibraciones, en la figura 29 se muestra ensamblado el torso, bicep izquierdo y derecho, aun sin los servomotores:



Figura 24: Ensamblaje torso y bicep

Para que el humanoide pueda realizar los movimientos de las articulaciones, se construyó una base la cual esta sujetada de las piezas inferiores del torso:



Figura 25: Base del humanoide

La mecánica de la cabeza tiene dos servomotores que controlan el movimiento de la mandíbula y el movimiento horizontal de la cabeza, contiene dos altavoces con su respectivo cableado, los altavoces fueron colocados de tal forma que su reemplazo se realice de forma fácil en caso de un fallo.



Figura 26: Mecánica de la cabeza

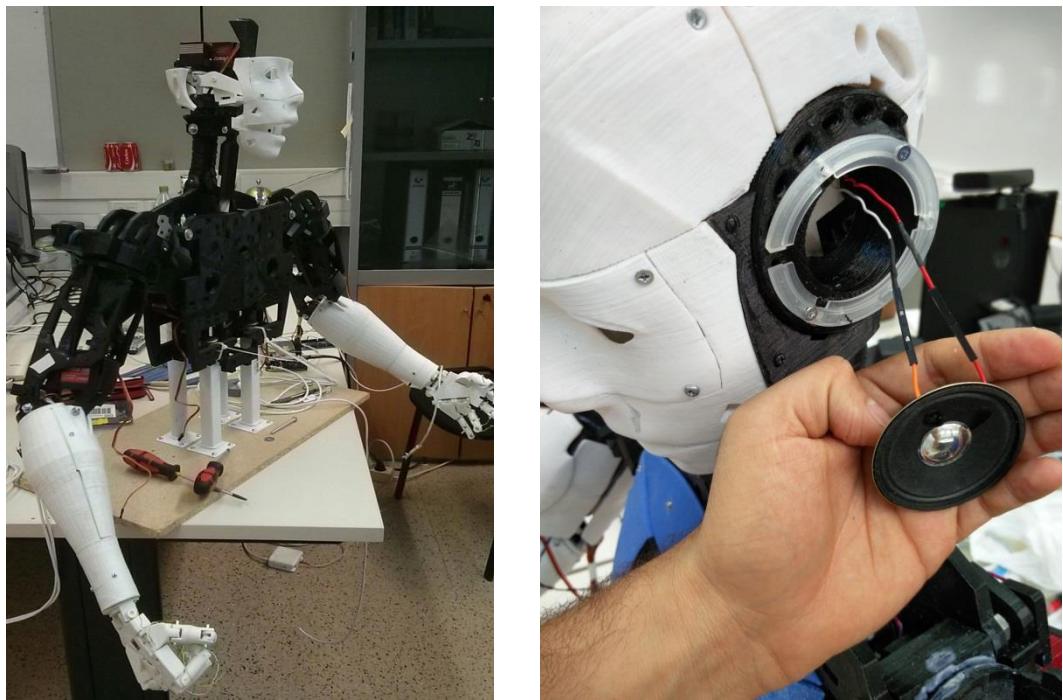
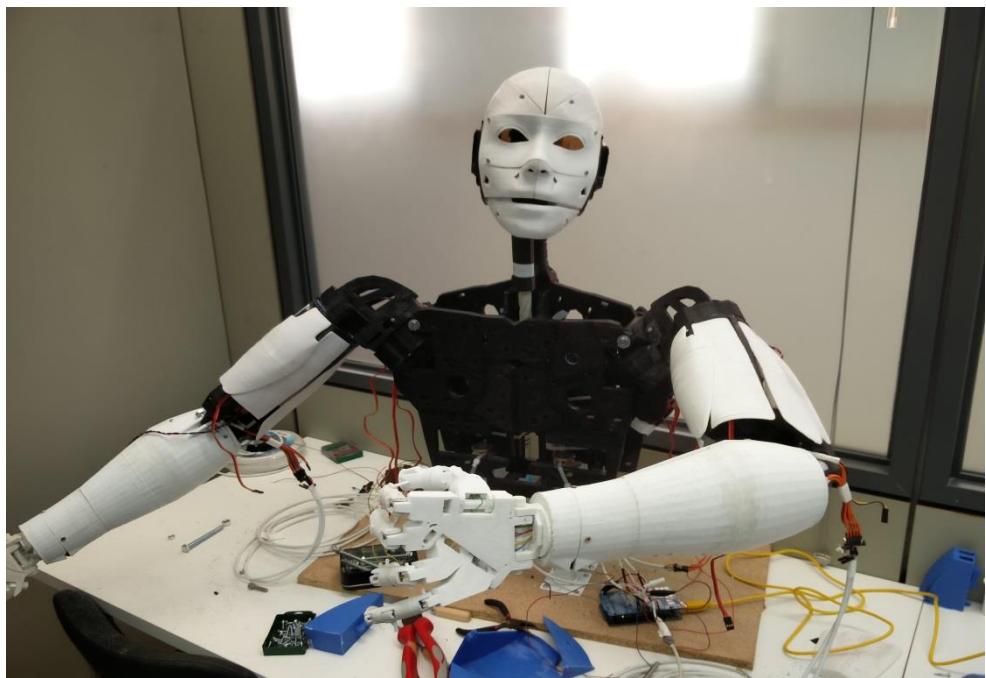
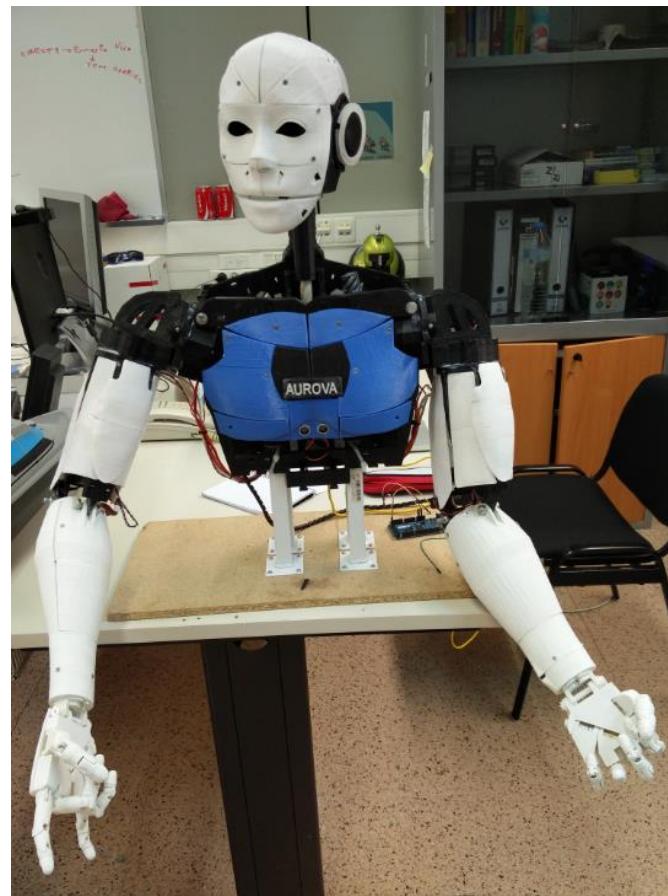
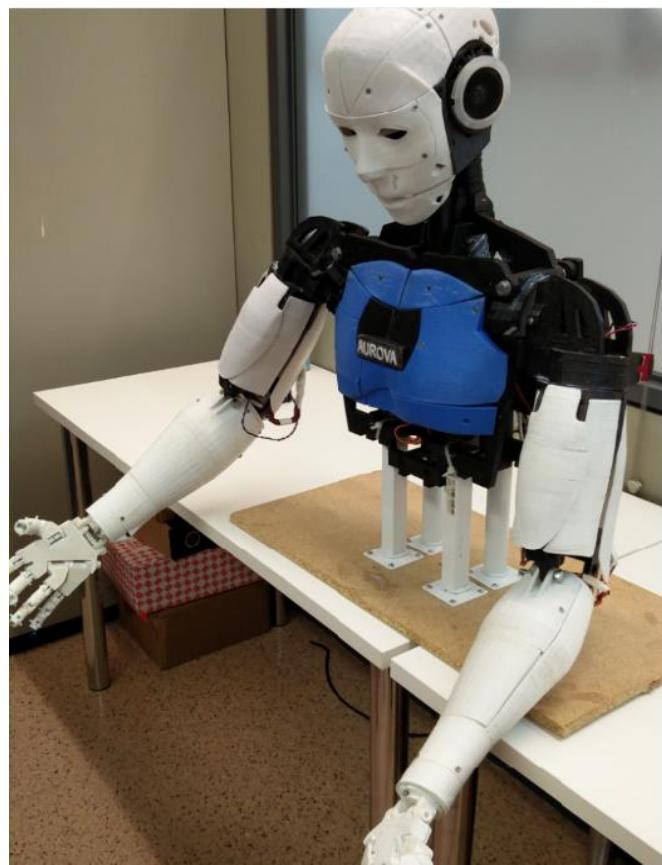


Figura 27: Instalación de altavoces

A continuación se muestran fotografías del montaje final del humanoide, en la sección posterior se mencionaran los aspectos que se tomaron en cuenta al momento de realizar las pruebas.





**Figura 28: Montaje final humanoide**

### **5.1.6. Pruebas y estabilización de montaje**

Los aspectos a tomar en cuenta para poder realizar las pruebas son:

1. Fricción adecuada de los engranajes

- Un ejemplo es el movimiento articular del hombro en el cual se tiene un conjunto de engranajes para poder mover todo el brazo.
- Se debe colocar lubricante para mejorar la rotación de los engranajes.

2. Soporte de los servomotores

- Cada servomotor debe ser bien asegurado para evitar vibraciones.

3. Ángulo máximo y mínimo de los servomotores

- Si se asigna un ángulo incorrecto es muy probable que se dañen piezas.

4. Interpolación de ángulo entre servomotores

- Para conseguir un movimiento con velocidad controlada.
- Para simular el movimiento de varios servomotores en paralelo.

5. Ubicación de cableado

- Se debe tener cuidado con la posición de los cables de alimentación, control del servomotor y potenciómetro ya que los movimientos pueden dañar cables que se cruzan entre movimiento de piezas.

6. Posición y firmeza de potenciómetros

- El potenciómetro mide la posición actual del motor y de esta forma poder controlar la posición máxima y mínima del servomotor, al estar externo al servomotor cualquier movimiento no controlado del mismo afectará los límites de giro.

7. Alimentación de los servomotores

- Se debe controlar la alimentación que recibe cada servomotor, al tener conectado muchos servomotores es posible que nuestra fuente de alimentación no sea la adecuada, otro factor a considerar es el tipo de cable a utilizar ya que al estar en funcionamiento muchos servomotores consumen

bastante corriente por lo que la sección de los cables debe ser la adecuada para evitar problemas de poca tensión en los servomotores.

## 5.2. Programación y control de humanoide

### 5.2.1. Plataforma de control desde el ordenador

#### 5.2.1.1. Requerimientos funcionales de la plataforma

1. Conexión y desconexión al puerto de comunicación asignado al Arduino.
2. Movimiento de las articulaciones del humanoide en tiempo real o por conjunto de articulaciones.
3. Definición de objetivos del humanoide
4. Un objetivo tiene un conjunto de trayectorias.
5. Toda información: objetivos, trayectorias, grado destino por articulación, entre otra información importante debe ser guardada en una base de datos.
6. Funcionalidad para poder recuperar objetivos guardados en base de datos.
7. Control de comando de voz: relación entre comando de voz y objetivo previamente definido.
8. Control de audio: relación objetivo, comando de voz y audio previamente guardado en memoria micro SD.

### 5.2.1.2. Interfaz de usuario

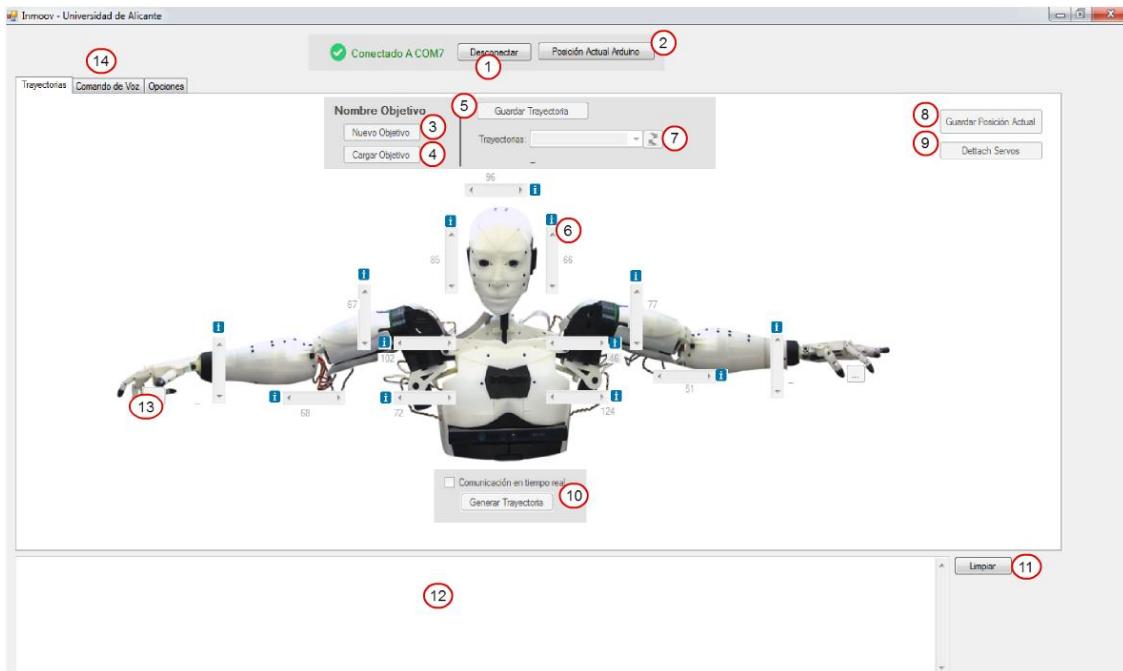


Figura 29: Interfaz de usuario

La figura 29 muestra la interfaz de usuario para control del humanoide desde el ordenador indicando cada una de las funcionalidades principales las cuales se detallan a continuación:

1. Permite desconectar al puerto de comunicación, si no se ha conectado al puerto la opción que aparece es la de conectar.
2. El Arduino por sí solo no almacena información una vez que se desconecta de la fuente de alimentación, para el control del humanoide es necesario guardar la posición actual de cada articulación esta posición se guarda en base de datos desde el ordenador, cuando se conecte al Arduino el primer paso es actualizar la última posición de cada uno de los servomotores.
3. Guarda a nivel de base de datos la descripción del objetivo, un objetivo tiene un conjunto de trayectorias.
4. Carga objetivos guardados en base de datos.
5. Guarda la posición actual de cada una de las articulaciones, para ello se debe indicar el nombre de la trayectoria, esta trayectoria es asignada al objetivo que se está definiendo.
6. El control *scrollBar* permite poder cambiar el grado actual de cada una de las articulaciones, cabe señalar que cada uno de los *scrollBar* tiene configurado el máximo y mínimo permitido por cada articulación.

7. El control *combobox* contiene los nombres de las trayectorias guardadas, cuando seleccionamos alguna automáticamente se muestran los grados por articulación.
8. Funcionalidad adicional que permite guardar en base de datos la posición actual de cada articulación, por defecto se guarda cuando se genera una trayectoria.
9. Funcionalidad adicional, permite liberar todos los servomotores.
10. Posiciona en humanoide según el movimiento configurado de cada articulación.
11. Limpia la consola
12. Visualización de consola, la consola son el conjunto de mensajes enviados por el Arduino y permite ver estado y depurar errores.
13. Para configurar las articulaciones de la mano se ha diseñado una pantalla adicional para tener un mejor control.
14. Funcionalidad que permite configurar el comando de voz, que será interpretado por el humanoide.

#### 5.2.1.3. Modelo relacional de base de datos

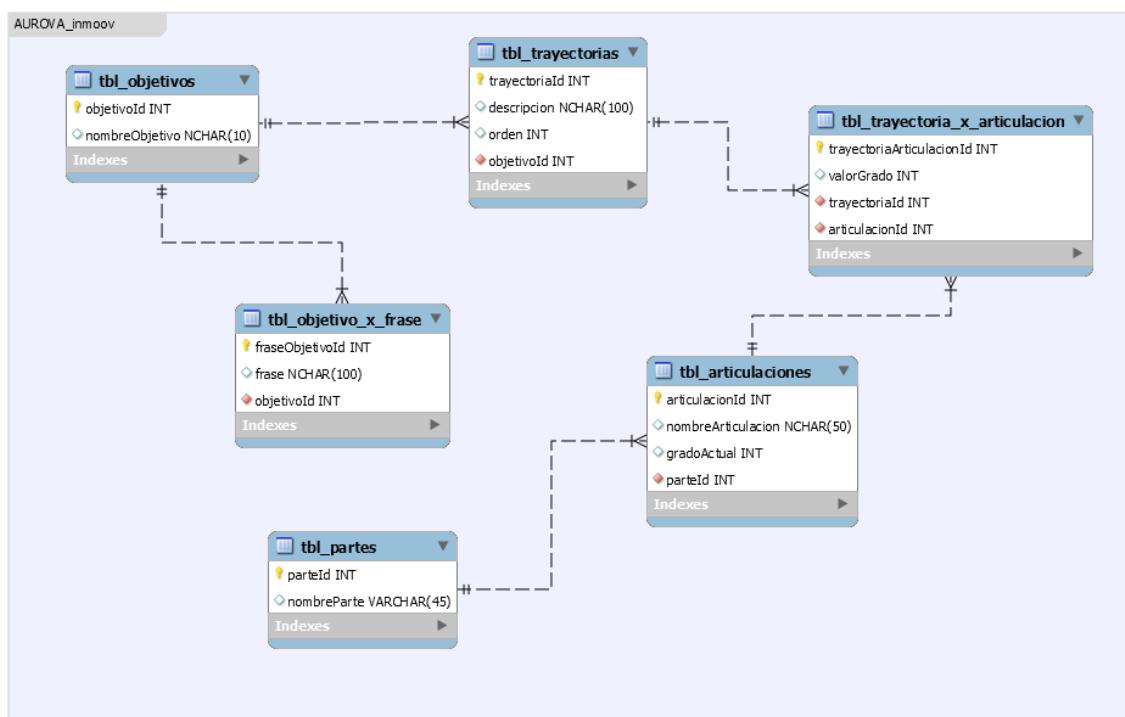


Figura 30: Modelo relacional base de datos Inmoov, fuente: elaboración propia

Descripción de tablas:

Nombre de tabla	Descripción
tbl_partes	Contiene el conjunto de partes del humanoide: cabeza, hombro derecho, entre otros.
tbl_objetivos	Almacena los objetivos, los cuales son definidos como un conjunto de trayectorias.
tbl_trayectorias	Almacena la descripción de trayectorias, una trayectoria tiene un conjunto de grados por articulación
tbl_trayectoria_x_articulación	Almacena la relación trayectoria por articulación con su respectivo grado en servomotor
tbl_articulaciones	Contiene el nombre de las articulaciones y almacena la posición actual de cada servomotor.
tbl_objetivo_x_frase	Almacena la relación entre la frase y el objetivo, una vez que el Arduino reciba determinada frase se procesará los movimientos configurados del objetivo.

Tabla 2: Descripción de tablas de base de datos

### 5.2.2. Controlador del Arduino

#### 5.2.2.1. Relación Servomotor(articulación) con pin del Arduino

Para poder realizar el controlador del Arduino es necesario tener un control de la relación de cada servomotor con su pin en Arduino, a continuación se muestra la tabla:

ID Articulación	Nombre de Parte	Nombre de Articulación	No. PIN en Arduino	Grado Mínimo	Grado Máximo
1	Brazo Derecho	Extensión Frontal Hombro	6	95	145
2	Brazo Izquierdo	Extensión Frontal Hombro	9	20	110
3	Brazo Derecho	Extensión Lateral Hombro	5	50	99

4	Brazo Izquierdo	Extensión Lateral Hombro	11	95	150
5	Brazo Derecho	Rotación del Bicep	7	50	90
6	Brazo Izquierdo	Rotación del Bicep	10	45	105
7	Brazo Derecho	Flexión del Codo	4	45	96
8	Brazo Izquierdo	Flexión del Codo	8	17	90
9	Brazo Derecho	Muñeca	30	50	120
10	Brazo Izquierdo	Muñeca	35	50	120
11	Mano Derecha	Anular	38	50	120
12	Mano Derecha	Meñique	40	50	120
13	Mano Derecha	Medio	32	50	120
14	Mano Derecha	Índice	36	50	120
15	Mano Derecha	Pulgar	36	50	120
16	Mano Izquierda	Anular	33	40	135
17	Mano Izquierda	Meñique	37	35	120
18	Mano Izquierda	Medio	31	5	138
19	Mano Izquierda	Índice	39	65	140
20	Mano Izquierda	Pulgar	41	43	115
21	Cabeza	Arriba-Abajo	12	20	85
22	Cabeza	Izquierda-Derecha	3	40	140
23	Cabeza	Mandíbula	2	75	125

Tabla 3: Relación articulación-pin Arduino

El funcionamiento del controlador Arduino se puede visualizar en el siguiente diagrama

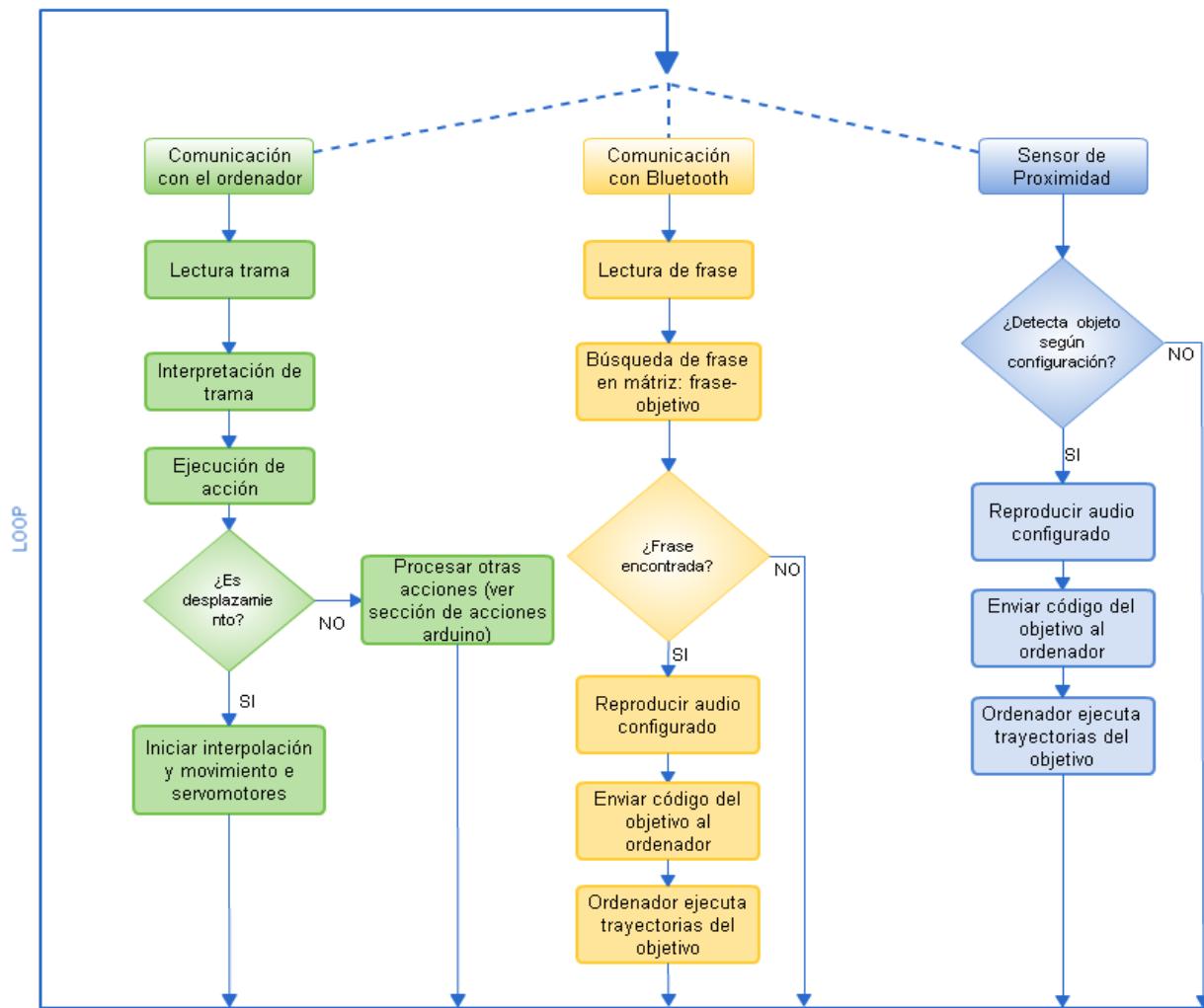


Figura 31: Controlador Arduino, fuente: elaboración propia

La figura 31 muestra el diagrama general del controlador interno del Arduino el cual está dividido en tres secciones principales:

1. Envío de secuencia desde el ordenador.
  - a. Se han definido un conjunto de tramas con formatos pre-establecidos, que ejecutan un conjunto de acciones como:
    - i. Actualización de posición inicial de los servomotores.
    - ii. Movimiento de servomotores.
    - iii. Mostrar información: posición actual, posición destino.
    - iv. Configuración de comando de voz.

2. Envío de frase desde módulo *bluetooth*.
  - a. Una vez enviada la frase se busca en la matriz de configuración de audio, si es encontrada envía el identificador del objetivo al ordenador para que este ejecute el conjunto de trayectorias.
3. Sensor de proximidad
  - a. Una vez que es detectado un objeto y está a una distancia configurada, el controlador localiza el identificador del objetivo y lo envía al ordenador para que este ejecute el conjunto de trayectorias almacenadas en la base de datos, además del identificador del objetivo el controlador reproduce el audio configurado.

### 5.2.3. Comunicación entre Arduino y Ordenador

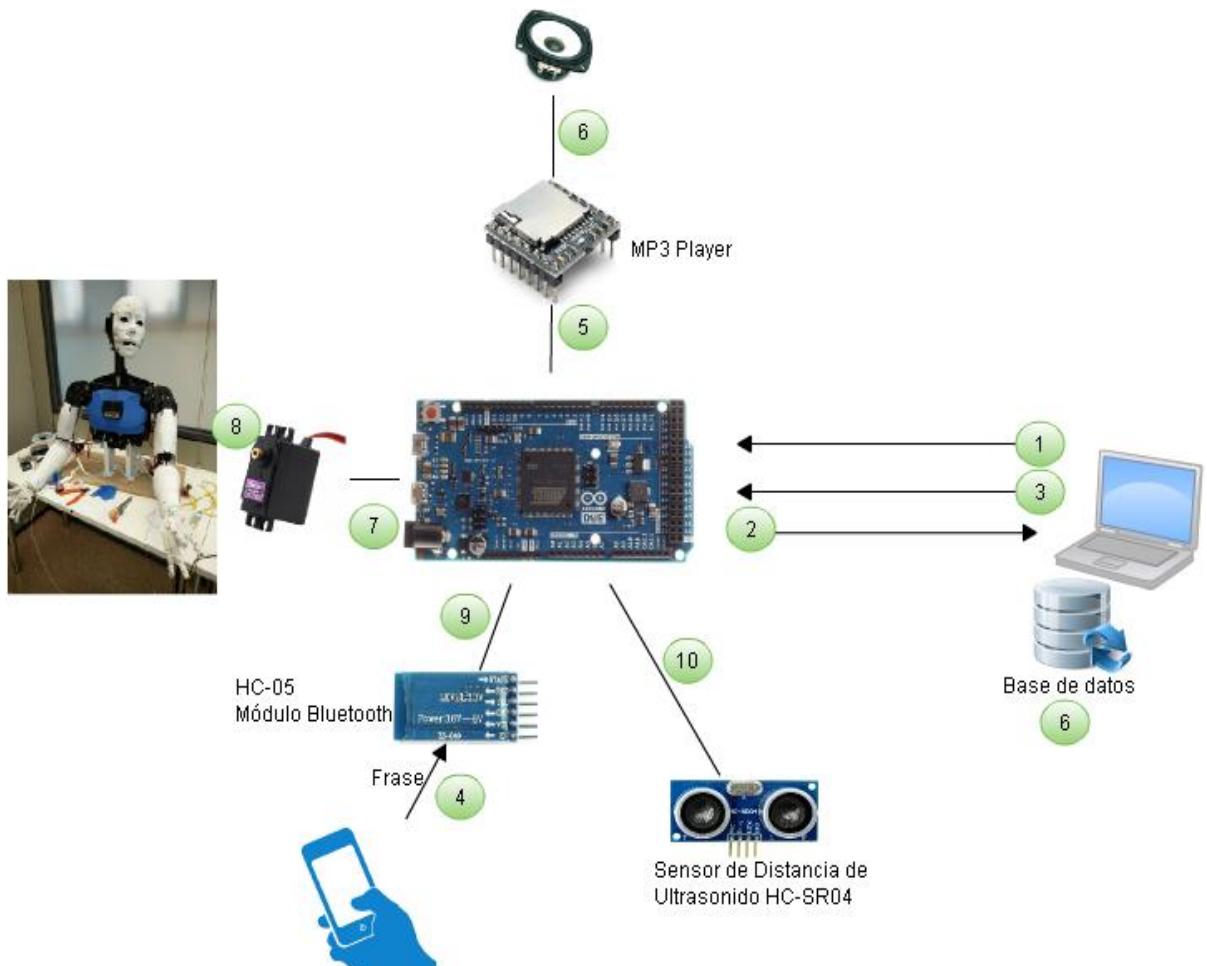


Figura 32: Comunicación componentes, fuente: elaboración propia

La figura 32 muestra la conexión entre los distintos componentes involucrados en el control del humanoide, a continuación se muestra una descripción de cada conexión:

1. Configuración inicial del comando de voz: Envío de trama que contiene la relación ID Objetivo a ejecutar, la frase (comando de voz) y el número de audio (para simulación de voz).
2. Si la frase enviada por el móvil es encontrada en la matriz de configuración, el Arduino enviará un mensaje al ordenador indicando el ID del objetivo y/o el número de audio a reproducir.
3. Una vez recibida la trama que indica el ID del objetivo, el ordenador ejecutará el conjunto de trayectorias almacenadas en base de datos según el objetivo identificado.
4. Comunicación desde el móvil al módulo *bluetooth* (ver sección comunicación móvil-Arduino).
5. El módulo *MP3 player mini*, es conectado mediante el Serial2 al Arduino Mega.
6. Toda información: objetivos, trayectorias, articulaciones, configuración de comando de voz, posición actual de los servomotores, es almacenada en base de datos SQL Server.
7. El Arduino controla la posición destino de cada servomotor, este control lo realiza mediante interpolación conociendo la posición actual y posición destino de los servomotores a controlar.
8. El humanoide Inmoov tiene veintitrés articulaciones que son controladas por veintitrés servomotores.
9. Para comunicar el Arduino con el móvil se utiliza el módulo *bluetooth* HC-05.
10. El sensor de distancia de ultrasonido HC-SR04 conectado al Arduino, nos permite identificar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm.

### 5.2.3.1. Definición de tramas

Trama	Descripción
\$01#	Utilizada para actualizar la posición inicial de los servomotores. Una vez enviada esta trama el controlador del Arduino esperará la posición de todos los servomotores separados por coma(,).
\$02#	Imprime la matriz de posición actual de todos los servomotores.
\$03#	Imprime la matriz de posición destino de todos los servomotores
\$04#	Posiciona cada uno de los servomotores a la posición actual.
\$05#	Realiza el <i>detach</i> de todos los servomotores.
\$06#	Activa controlador del Arduino en modo de configuración de comando de voz.
%OBJ-FRASE-AUDIO#	Una vez activada el modo de configuración de comando de voz, se envía la trama con la relación: identificador del objetivo – frase – número de audio.
\$07#	Fin de configuración de comando de voz.
\$08#	Imprimir frases configuradas: relación del identificador del objetivo con la fase.
\$09#	Imprimir información del controlador instalado en el Arduino.
\$10#	Activar el sensor HC-SR04.
\$11#	Desactivar el sensor HC-SR04.
\$12#	Activar modo de configuración del sensor HC-SR04.
\$13#	Desactivar modo de configuración del sensor HC-SR04.
\$14#	Imprimir configuración actual del sensor HC-SR04.

Tabla 4: Definición de tramas

#### 5.2.4. Comunicación entre Arduino y Móvil

- Para el control del humanoide mediante comando de voz se ha utilizado el complemento de *google* el cual procesa de forma rápida y eficiente la voz de cualquier persona, convirtiéndola a texto.
- Para poder tener acceso a este complemento necesitamos un móvil android con la aplicación *AMR\_Voice*, la cual puede ser descarga del siguiente enlace:  
[https://play.google.com/store/apps/details?id=robotspace.simplelabs.amr\\_voice&hl=es](https://play.google.com/store/apps/details?id=robotspace.simplelabs.amr_voice&hl=es)
- Una vez instalada se debe establecer la conexión mediante *bluetooth*, mediante la opción *connect robot*, esto conectará al módulo HC-05 conectado al Arduino.

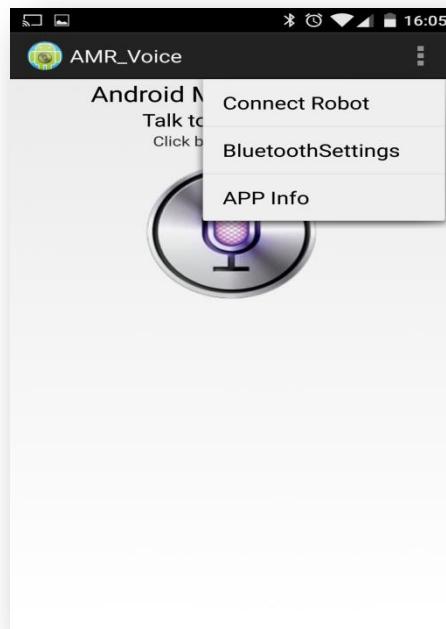


Figura 33: Aplicación AMR\_Voice Android

- Una vez configurada la aplicación y realizada la conexión, cada frase que se diga mediante el móvil será convertida a texto y enviada mediante *bluetooth* al Arduino el cual la capturará y procesará.

### 5.3. Pruebas y resultados

No.	Pruebas realizada	Resultados
1.	Control de las articulaciones de los dedos de la mano.	Prueba satisfactoria.
2.	Control de articulaciones cuyo servomotor ha sido modificado y extraído el potenciómetro	<p><b>Problema identificado:</b> el servomotor nunca se detiene, no localiza la posición destino.</p> <p>Para este problema se identificaron dos causas:</p> <p><b>Causa 1:</b> Mal cableado del controlador del servomotor al potenciómetro.</p> <p><b>Solución:</b> verificar cableado y corregirlo.</p> <p><b>Causa 2:</b> El servomotor encuentra la posición destino al contrario del movimiento de la articulación que controla.</p> <p><b>Solución:</b> invertir la polaridad del servomotor.</p>
3.	Control de un servomotor desde la consola del IDE de Arduino.	Prueba satisfactoria.
4.	Control de un servomotor desde la aplicación.	<p><b>Problema identificado:</b> daño a piezas al realizar movimientos de articulaciones.</p> <p><b>Causa:</b> errores de la parametrización de los grados mínimos y máximos de alguno de los servomotores</p> <p><b>Solución:</b> volver a imprimir piezas dañadas y realizar la correcta parametrización de los grados máximos y mínimos.</p>
5.	Ejecución de trayectoria, que controla varios servomotores mediante interpolación.	<p>Prueba satisfactoria con pocos servomotores en movimiento, después de aproximadamente cinco servomotores en ejecución, todos los servomotores que tenían el <i>attach</i> comenzaban a vibrar</p> <p><b>Problema identificado:</b> reducción de la tensión necesaria para el correcto funcionamiento de los servomotores, a raíz de cable principal muy delgado.</p> <p><b>Solución:</b> cambio del cableado principal por uno más grueso</p>

6.	Definición de objetivo, trayectorias y ejecución de las mismas.	Prueba satisfactoria, recuperando información de la base de datos, construyendo la trama y enviarla al Arduino.
7.	Control mediante comando de voz y relación con objetivo.	Prueba satisfactoria, se debe tener en cuenta que el tiempo que requiere la aplicación móvil para el procesamiento e identificación de la voz.
8.	Pruebas de altavoces y conexión a la placa principal.	<p><b>Problema identificado:</b> el módulo <i>MP3 player mini</i> no funciona al integrarlo a la placa principal.</p> <p><b>Causa:</b> al unir el módulo <i>MP3 player mini</i> a la placa principal la cual tenía un cableado interno que provocaba corto circuito.</p> <p><b>Solución:</b> quitar el cableado interno de la placa principal.</p>

Tabla 5: Pruebas y resultados

### 5.3.1. Ejemplo de configuración del humanoide

El diseño de la plataforma y controlador del Arduino ofrece flexibilidad de configuración según la funcionalidad que se requiera dar al humanoide, en este apartado explicaremos cuales son los pasos para configurar el humanoide según un planteamiento establecido.

#### Ejemplo:

Se requiere que el humanoide realice lo siguiente:

1. Cuando se envíe cierto comando de voz debe ejecutar una trayectoria y/o responder mediante sonido (simulación de voz mediante audio)

Comando de Voz	Trayectoria a Ejecutar	Respuesta del humanoide (audio a reproducir)
Hola	<i>Sin trayectoria</i>	Hola
Qué tal estás	Movimiento corto arriba-debajo de la cabeza.	Bien
Cuál es tu nombre	Extensión lateral corta de ambos brazos.	Mi nombre es Auron.
Dónde te construyeron	Movimiento corto arriba-	Universidad de Alicante

	debajo de la cabeza.	
Eres hombre	Movimiento corto arriba-abajo de la cabeza (repetir dos veces).	Si
A que grupo de investigación perteneces.	Colocar la cabeza en la posición central y brazos pegados al cuerpo.	Grupo de investigación Aurova.
Saluda	Extensión frontal del brazo derecho, extensión del codo, se debe estirar todos los dedos de la mano derecha.	<i>Sin respuesta</i>

Tabla 6: Definición de órdenes mediante comando de voz

2. Si el humanoide detecta la presencia de algo o alguien enfrente debe realizar lo siguiente:

Distancia mínima – Distancia máxima	Trayectoria a Ejecutar	Respuesta del humanoide (audio a reproducir)
30cm-50cm	Extensión lateral corta de ambos brazos.	Mi nombre es Auron
50cm-100cm	<i>Sin trayectoria</i>	Hola
100cm-150cm	Movimiento corto arriba-abajo de la cabeza.	Ven acércate un poco

Tabla 7: Definición de órdenes mediante detector de proximidad

#### 5.3.1.1. Configuración del humanoide

En la sección anterior se detallaron los aspectos necesarios para ser configurados en el humanoide, a continuación se detallan los pasos a seguir:

- a) Configuración de Objetivos y Trayectorias

El primer paso es identificar cuantos objetivos necesitamos, para nuestro ejemplo en estudio se requieren los siguientes objetivos:

No.	Descripción del objetivo	Nombre corto del objetivo (será utilizado para guardar el objetivo en la aplicación)
1	Movimiento corto arriba-abajo	Cabeza arriba-abajo
2	Extensión lateral corta de ambos brazos.	Extensión hombros
3	Movimiento corto arriba-abajo de la cabeza (repetir dos veces).	Cabeza arriba-abajo (dos)  *Para este objetivo se pueden utilizar las trayectorias del objetivo No. 1
4	Colocar la cabeza en la posición central y brazos pegados al cuerpo.	Cabeza inicio
5	Extensión frontal del brazo derecho, extensión del codo, se debe estirar todos los dedos de la mano derecha.	Saludar

Tabla 8: Identificación de objetivos

Se identificaron cinco objetivos, el siguiente paso es configurar estos objetivos en la aplicación, para ello podemos consultar la sección: [8.2.3 Creación de Objetivos y Trayectorias](#), donde se explica detalladamente los pasos a seguir para creación de objetivos y trayectorias.

*Nota:* una vez que estos objetivos sean configurados en la aplicación quedarán almacenados en base de datos y podrán ser utilizados posteriormente para la configuración en proceso o para otras configuraciones.

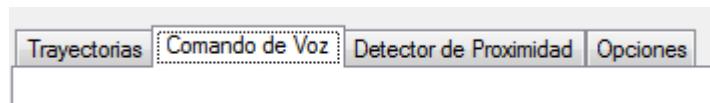
#### b) Configuración de comandos de voz

En la [Tabla No. 7](#) se muestran los diferentes comandos de voz con su respectiva trayectoria y respuesta del humanoide para configurar a nivel de aplicación cada uno de los comandos se debe consultar la sección [8.2.4. Comando de voz y audio](#).

Para efectos de ejemplo, mostraremos como configurar el segundo comando de voz:

Qué tal estás	Movimiento corto arriba-abajo de la cabeza.	Bien
---------------	---	------

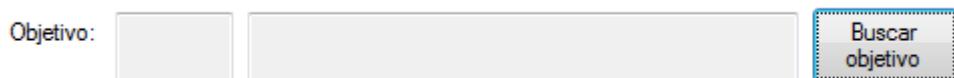
- Clic en comando de voz



- Clic en activar



- Buscamos el objetivo configurado previamente, llamado *Cabeza arriba-abajo*.



- Ingresamos el comando de voz

Frase:

*Nota:* El comando de voz debe ser ingresado exactamente como es procesado desde el móvil, para ello se puede realizar la conexión desde el móvil y enviar el comando de voz previo a la configuración en la aplicación con ello lograremos visualizar el texto procesado y enviado del móvil al Arduino, consultar la sección [5.3.4. Comunicación móvil-Arduino](#) para mayor detalle.

- Seleccionamos el audio a reproducir cuando se envíe el comando de voz

Audio:  ▾

- Clic en enviar, para enviar los parámetros ingresados desde la aplicación al controlador del Arduino.



En la sección [8.7 Audios pre-grabados](#) se puede ver el listado de voces que el humanoide puede simular actualmente.

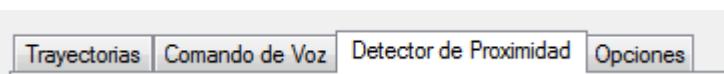
c) Configuración de detector de proximidad

En la [Tabla No. 8](#) se muestran las diferentes distancias a ser configuradas en el sensor de proximidad con su respectiva trayectoria y respuesta del humanoide, para configurar a nivel de aplicación cada uno de los comandos se debe consultar la sección [8.2.5. Detector de Proximidad](#).

A continuación mostraremos como configurar los parámetros desde la aplicación para uno de los requerimientos del ejemplo en estudio del sensor de proximidad:

30cm-50cm	Extensión lateral corta de ambos brazos.	Mi nombre es Auron
-----------	--	--------------------

- Clic en Detector de proximidad



- Clic en Activar configuración



- Ingresar los parámetros de distancia y periodicidad de detección

Distancia mínima(cm):	<input type="text" value="30"/>
Distancia máxima(cm):	<input type="text" value="50"/>
Repetir cada Seg.	<input type="text" value="20"/>

*Nota:* la periodicidad de detección (Repetir cada Seg.) es utilizada cuando el objeto esta estático y cuya detección se realizará cada cantidad de segundos que se configure en este parámetro.

- Seleccionamos el objetivo configurado previamente, llamado Extensión hombros



- Seleccionamos el audio a reproducir cuando se ejecute la detección

Audio:

Mi nombre es Auron



- Clic en enviar, para enviar los parámetros ingresados desde la aplicación al controlador del Arduino.



- Una vez completada la configuración se debe dar clic en Finalizar para indicarle al controlador del Arduino que la configuración del sensor de proximidad ha terminado.



*Nota:* a nivel de aplicación se puede Activar o Desactivar el sensor de proximidad para indicarle al controlador si debe ejecutar la verificación de los parámetros configurados o no.



## 5.4. Trabajos futuros

- Construcción del resto del cuerpo
  - a. *Estómago*: en la página del proyecto Inmoov se encuentra los modelos de las piezas y otros componentes necesarios para la construcción del estómago.
  - b. *Piernas*: no hay modelos de piezas creados actualmente, el proyecto Inmoov tiene incorporado ruedas que orientan al humanoide.
  - c. *Ojos*: en la página del proyecto se encuentra los modelos necesarios para construir los ojos mecánicos capaces de poder realizar movimientos similares a los de un humano.
  - d. *Sensores de presión en los dedos*: con el objetivo de poder detectar la fuerza que se debe aplicar para coger un objeto, se puede incorporar sensores de presión en las yemas de los dedos; con estos sensores y el control adecuado para asignar la posición deseada en cada uno de los servomotores que controlan los dedos.
- Mejoras en articulaciones que requieren una mayor fuerza
  - a. Hay articulaciones que realizan una mayor fuerza, por ejemplo: el movimiento lateral y frontal del hombro que tiene que poder soportar todo el peso de todo el brazo. Para dichas articulaciones sería recomendable imprimir los engranajes con una mejor calidad o elaborar estos engranajes con otro tipo de material más fuerte que el plástico.
- Comando de voz incorporado
  - a. El humanoide tiene incorporado el comando de voz mediante un móvil y utilizando el *plugin de google*, se puede realizar un módulo de reconocedor de voz incorporado en el humanoide que no dependa de un dispositivo adicional y que tenga igual o mejor calidad al momento de obtener, identificar y procesar la voz.
- Inteligencia artificial
  - a. Actualmente las frases que el humanoide entiende ya están pre-configuradas, con el reconocimiento de voz se puede diseñar se puede diseñar un módulo que pueda aprender de conversaciones con humanos, este módulo debe incluir
    - i. Definición y almacenamiento de trayectorias
    - ii. Almacenamiento de frases y relación con trayectorias

## **VI. Conclusiones**

1. La construcción de un humanoide con técnicas de impresión 3D y controlado mediante hardware abierto como lo es Arduino reduce los costos y pueden ser usados en educación, entretenimiento, prototipos, sistemas de robótica sociales, entre otras aplicaciones. Cabe señalar que diseñar un humanoide no es un proceso fácil ya que es necesario evaluar fuerza, pesos, par articulares, dimensiones, organización, entre otros factores; sin embargo, el proyecto Inmoov proporciona todos los diseños y listado de recursos necesarios para la construcción de un humanoide de bajo costo.
2. Desarrollando un controlador adecuado para Arduino junto con un protocolo de comunicación completo con tramas que permitan su verificación y procesamiento rápido se pueden establecer una comunicación Arduino-ordenador eficiente y permitir el control de un proyecto industrial, robótico o de otra índole.
3. El tratamiento y análisis del contenido lingüístico de una señal de voz no es proceso fácil, pero con ayuda del analizador de google el cual permite procesar de forma eficiente los distintos timbres de voz de las personas y al ser interpretadas y enviadas mediante un protocolo y módulo de comunicación al Arduino se puede realizar un control e interacción con el humanoide.

## VII. Bibliografía y referencias

- Arduino. (2016). Arduino. Retrieved from <https://www.arduino.cc/>
- Bestmann, M., Reichardt, B., & Wasserfall, F. (2015). Hambot: An Open Source Robot for RoboCup Soccer. In L. Almeida, J. Ji, G. Steinbauer, & S. Luke (Eds.), *Robocup 2015: Robot World Cup Xix* (Vol. 9513, pp. 339-346). Berlin: Springer-Verlag Berlin.
- Candelas, F., & Corrales, J. (2007). Servomotores.
- Constantin, D., Marcoianu, I., Voiniciuc, C., Vinturis, V., & Ieee. (2015). ASPECTS REGARDING THE DESIGN AND REALISATION OF A HUMANOID ROBOT *Proceedings of the 2015 7th International Conference on Electronics, Computers and Artificial Intelligence* (pp. P105-P108). New York: Ieee.
- Langevin, G. (2016). Inmoov. Retrieved from inmoov.fr
- Mariano. (2013). Retrieved from <http://tecnologiadelosplasticos.blogspot.com.es/2013/02/impresion-3d.html>
- Microsoft. (2016a). SQL Server. Retrieved from <https://www.microsoft.com/es-es/server-cloud/products/sql-server/>
- Microsoft. (2016b). Visual Studio 2015. Retrieved from <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>
- Neoteo. (2016). Servomotores: El primer paso hacia tu Robot. Retrieved from <http://www.neoteo.com/servomotores-el-primer-paso-hacia-tu-robot>
- Pablo. (2012). CURA – Software para Impresión 3D – Sprinter Vs Marlin. Retrieved from <https://www.tr3sland.com/2012/12/software-cura-para-impresion-3d-sprinter-vs-marlin/>
- Polulu. (2016). Robotics & Electronics. Retrieved from <https://www.pololu.com>
- Electrónicos Caldas. (2016). Retrieved from Electrónicos Caldas: <http://www.electronicoscaldas.com/modulos-rf/452-modulo-bluetooth-hc-05.html>
- Electronilab. (2016). Retrieved from <http://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>

## VIII. Anexos

### 8.1. Manual de Administrador

#### 8.1.1. Software necesario

Para el funcionamiento y adecuaciones al proyecto, se deben instalar las siguientes herramientas:

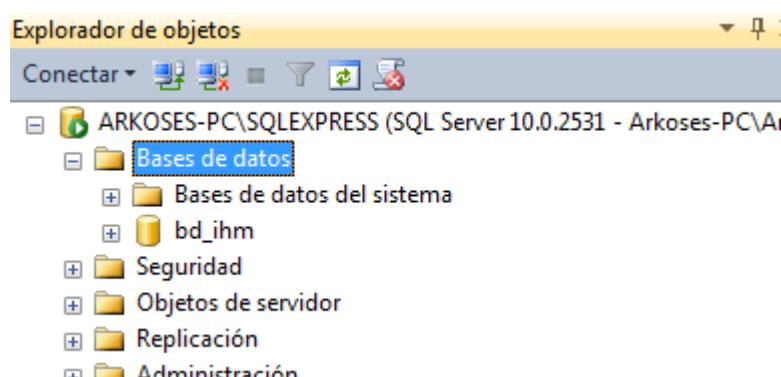
1. Visual Studio 2010 o superior
2. SQL Server 2012 (motor y administrador)
3. IDE Arduino

#### 8.1.2. Iniciar aplicación en el ordenador

##### 8.1.2.1. Restaurar base de datos

Para la restauración de base de datos se debe seguir los siguientes pasos:

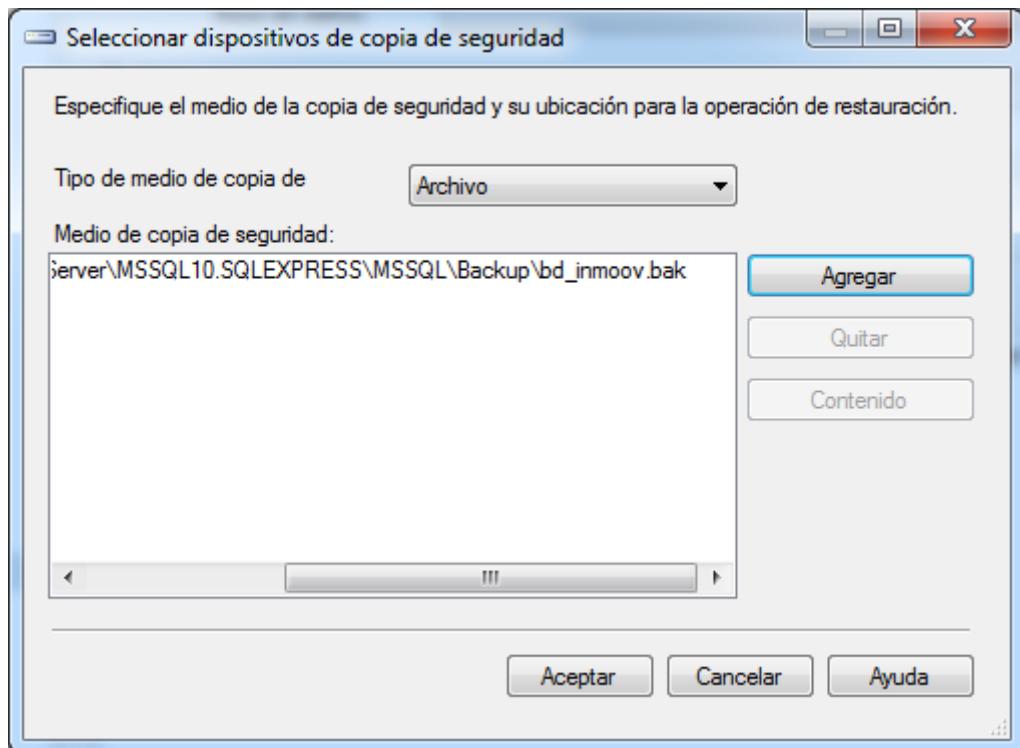
1. Abrir Microsoft SQL Server Management Studio.
2. Conectarse al motor SQL Server mediante la autenticación configurada.
3. Clic derecho en Base de datos → Restaurar base de datos...



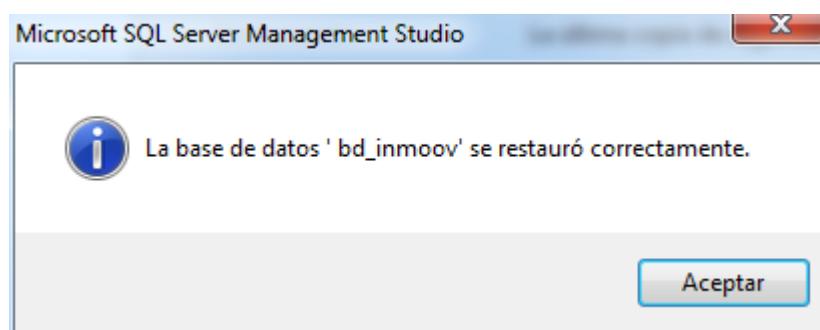
4. En origen seleccionar dispositivo:



5. Luego agregar y buscar la ubicación donde se encuentra el archivo “bd\_inmoov.bak” proporcionado



6. Clic en Aceptar
7. Luego clic en Aceptar
8. Si todo se procesó correctamente se mostrará el mensaje:

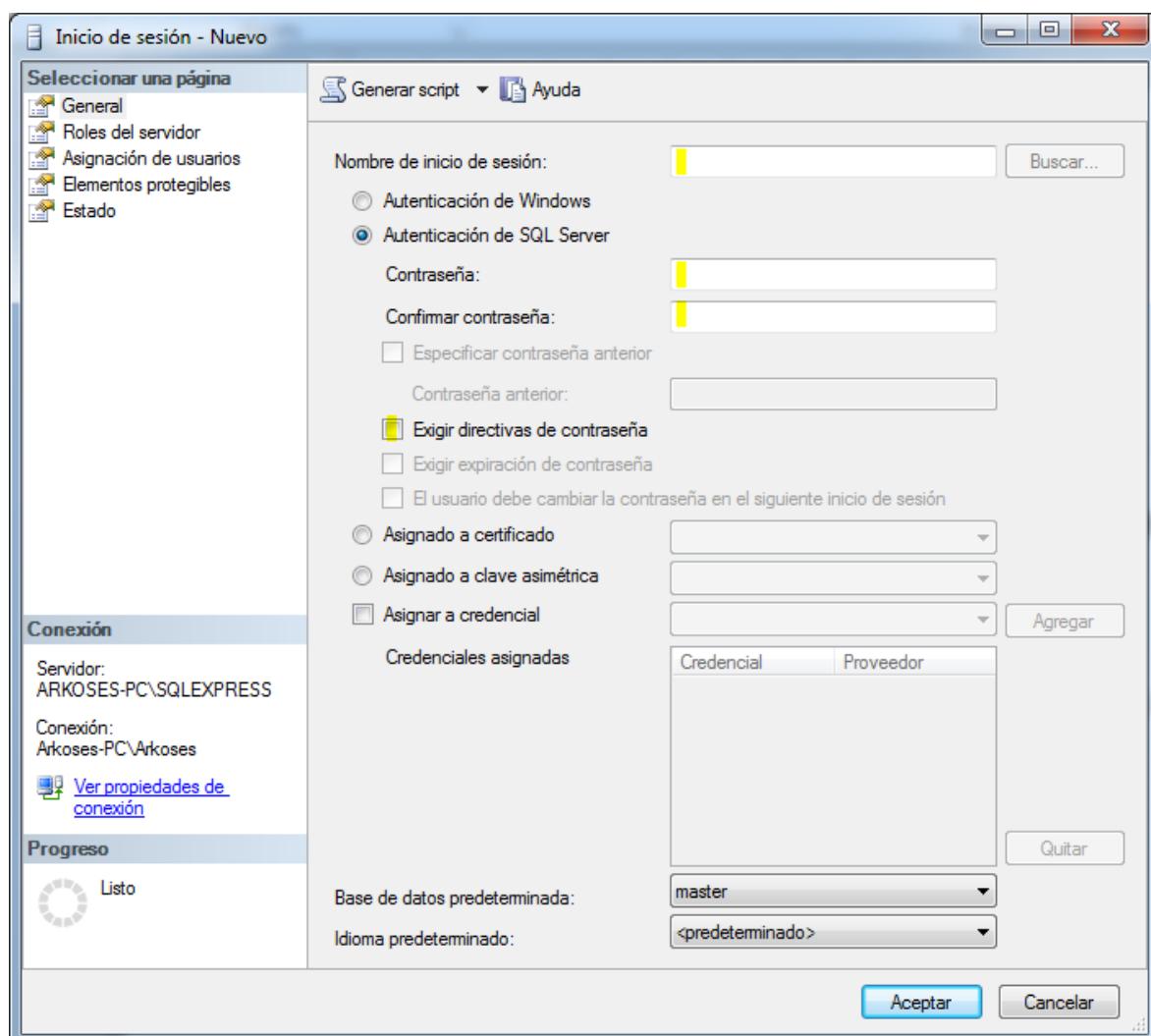


Una vez restaurada la base de datos, es necesario crear un nuevo Inicio de sesión, los pasos se describen a continuación:

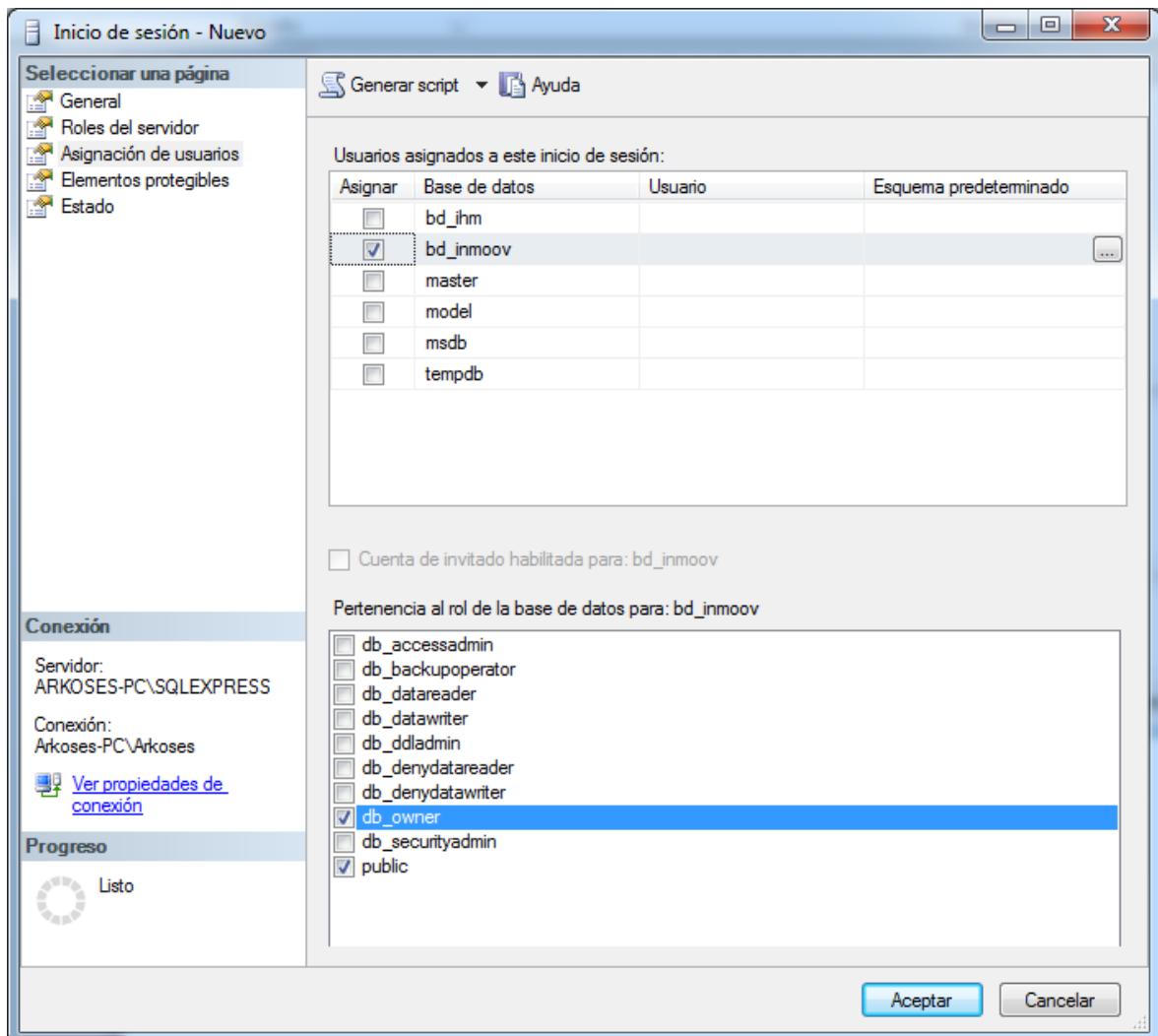
Nota: la conexión a la base de datos configurada en el proyecto C# es:

*Data Source=ARKOSES-PC\SQLEXPRESS;Initial Catalog=bd\_inmoov;UserID=aurova;Password=123*

1. Clic en Seguridad,
2. Clic derecho inicio de sesión, nuevo inicio de sesión
3. Ingresar el nombre del usuario, contraseña y quitar la marca en exigir directivas de contraseña :



4. En asignación de usuarios, seleccionar la base de datos bd\_inmoov y el rol db\_owner



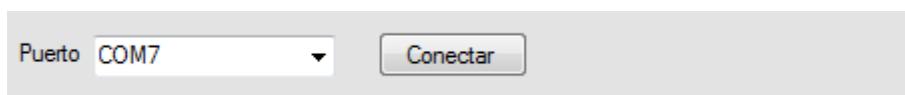
### 8.1.2.2. Iniciar aplicación

Antes de iniciar la aplicación se deben realizar los siguientes pasos:

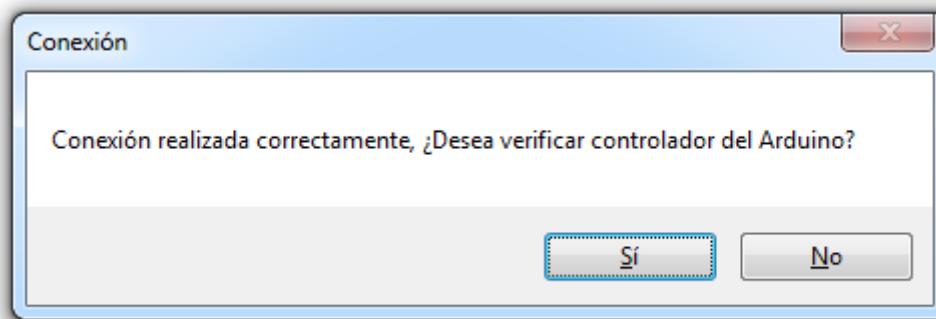
1. Verificar configuración de conexión a base de datos, archivo App.config parámetro *cadenaConexion*



2. Ejecutar la aplicación
3. Realizar la conexión al puerto que corresponde al Arduino



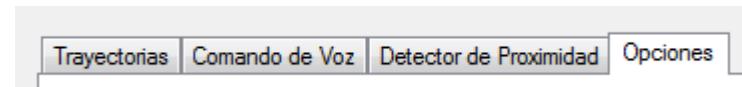
4. Verificar el controlador del Arduino, dando clic en SI al mensaje que se muestra después de realizar la conexión:



El resultado en la parte inferior de la ventana debe ser parecido a:

```
****Controlador Arduino InMoov****"  
**Versión: "+String(VERSION_ACTUAL  
**Autor: OSCAR HERNANDEZ  
**Fecha: Marzo-Julio 2016  
**Funcionalidad: Control desde aplicación en ordenador, comando de voz, audio;  
Puede continuar con la configuración del controlador Arduino;
```

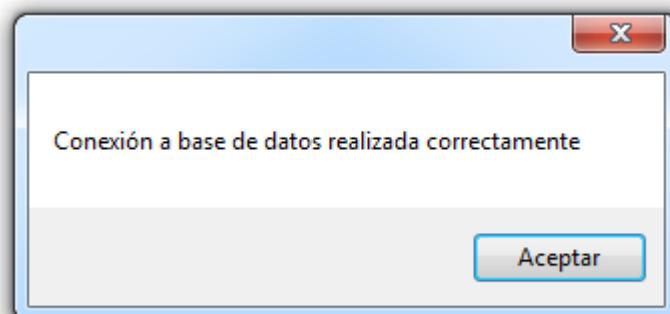
5. Verificar la conexión a base de datos, clic en Opciones



6. Clic en comprobar



Debería aparecer el siguiente mensaje:



## 8.2. Manual de usuario

### 8.2.1. Pasos iniciales

Para poder controlar el humanoide desde el ordenador se debe ver la sección **8.1.2 Iniciar aplicación en el ordenador** del manual de administrador.

### 8.2.2. Controlar movimientos del humanoide

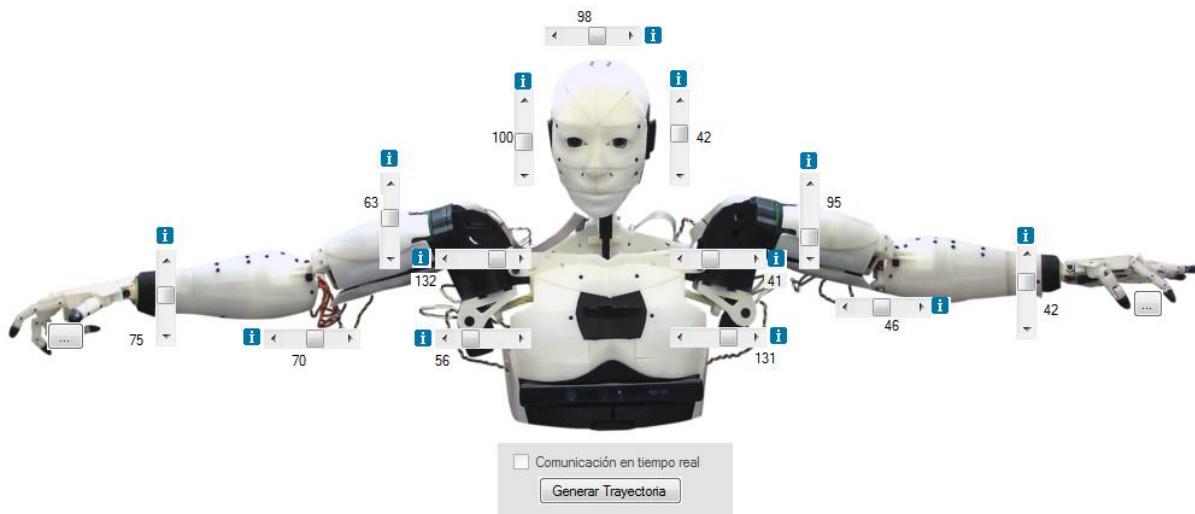


Figura 34: Articulaciones humanoide InMoov

Actualmente el humanoide tiene 23 articulaciones que corresponde a 23 servomotores, los movimientos se controlan asignando posiciones entre 0 y 180 a cada uno de los servomotores, los pasos a seguir para controlar los movimientos son:

1. Seleccionar el valor deseado en la(s) articulación(es) del humanoide.
2. La opción **i** que está en cada una de las articulaciones nos permite ver información adicional.
3. Clic en **Generar Trayectoria**: esta opción envía al Arduino el conjunto de posiciones deseadas de cada servomotor y las procesa mediante interpolación. Nota: es recomendable mover pocos servomotores a la vez, ir aumentando y viendo el comportamiento del humanoide.

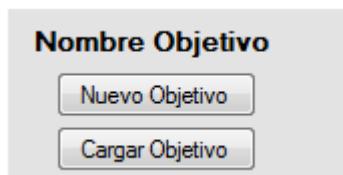
### 8.2.3. Creación de Objetivos y Trayectorias

Definiciones iniciales:

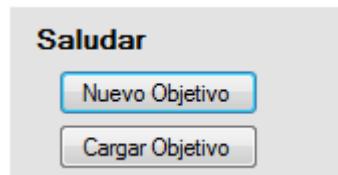
- *Articulación*: cada movimiento o grado de libertad del humanoide, es equivalente a cada servomotor (23 en total).
- *Trayectoria*: una trayectoria es un conjunto de movimientos articulares, es equivalente a asignar una posición deseada a cada servomotor.
- *Objetivo*: son un conjunto de trayectorias, el controlador ejecuta este conjunto de trayectorias en el orden establecido para alcanzar el objetivo. Ejemplos de objetivo: mover cabeza izquierda derecha, saludar, entre otros.

Los pasos a seguir para establecer un objetivo al humanoide son:

1. Clic en Nuevo Objетivo

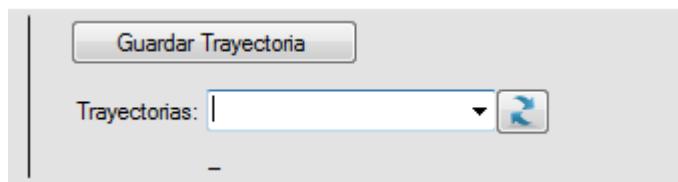


2. Ingresar el nombre del objetivo (por ejemplo *Saludar*) y dar clic en Aceptar
3. Con esto se habrá creado el objetivo almacenándolo en la base de datos y ya se habilitarán las opciones para poder agregarle trayectorias, además e visualizará el nombre del objetivo que se está definiendo:

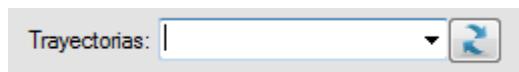


4. Una vez guardado el objetivo el siguiente paso es asignar los valores correspondientes a cada articulación (ver la sección [8.2.2 Controlar movimientos del humanoide](#) para más información).

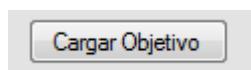
5. Una vez definido el valor de cada articulación se debe dar clic en Guardar Trayectoria



6. Se debe ingresar el nombre de la trayectoria y dar clic en Aceptar, con eso se asignará la trayectoria al objetivo que se está definiendo. Nota: en el orden que se definen las trayectorias, en ese orden se ejecutarán.
7. Se pueden visualizar las trayectorias guardadas seleccionando de la lista desplegable, cuando se seleccione una trayectoria automáticamente se cargan los valores configurados de cada articulación y se puede realizar la ejecución de la trayectoria si se desea.



8. La opción Cargar Objetivo, permite recuperar un objetivo guardado en base de datos.



#### 8.2.4. Comando de voz y audio

La configuración del comando de voz y audio se basa en la relación de la entrada: comando de voz y las salidas: movimientos del humanoide y/o audio, ver figura 40:

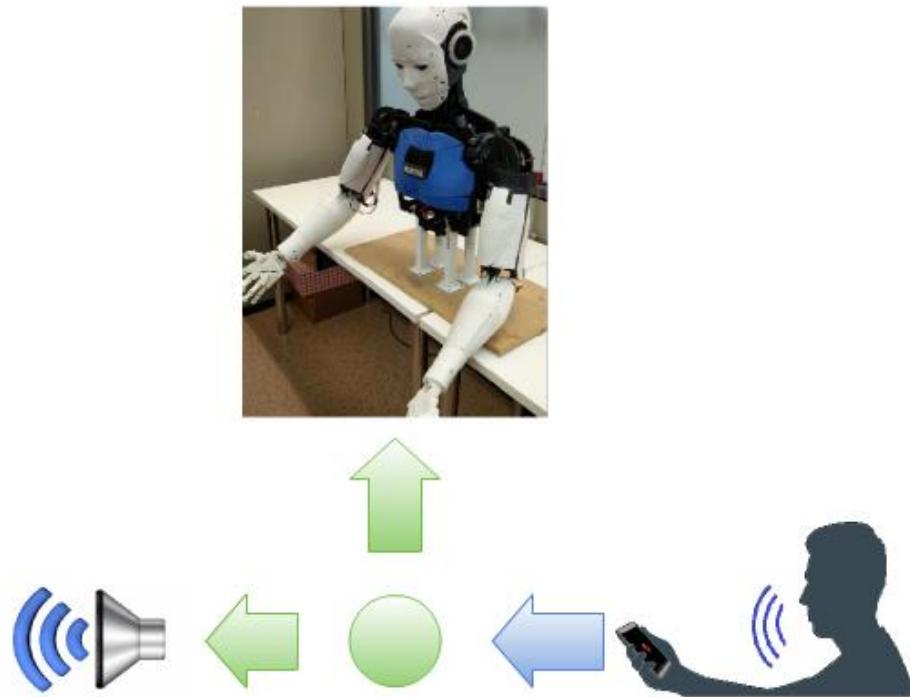
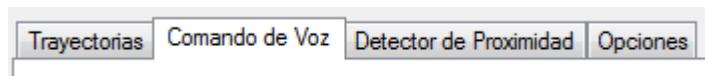


Figura 35: Diagrama comando de voz y audio, fuente: elaboración propia

Para configurar el comando de voz y audio se deben seguir los siguientes pasos:

1. Clic en comando de voz



2. Clic en Activar, activar habilita el controlador de Arduino en modo de configuración de comando de voz y audio

Activar      Finalizar      Ver Frases

Objetivo:

Frase:

Audio:

Al dar clic en Activar se debe visualizar un texto similar a:

*Iniciando configuración comando de voz...*

*Cadena de envío: %ID-FRASE-AUDIO# %3-N-N#*

3. Como ya se mencionó anteriormente la configuración de audio se basa en la relación de tres elementos: comando de voz (frase que el usuario dice), audio y movimientos del humanoide

Enviar

Objetivo:

Frase:

Audio:

Las posibles combinaciones pueden ser:

- Frase con audio.
  - Se debe seleccionar en objetivo “-1 NINGUNO” dando clic en Buscar Objetivo.
- Frase con objetivo.
  - Se debe seleccionar en Audio Sin Sonido de la lista desplegable.
- Frase con audio y objetivo.
  - Se debe ingresar la frase, seleccionar el objetivo y el sonido deseado.

*Nota:* Para ver como procesa la frase mediante el comando de voz, se puede conectar y enviar mediante el móvil, al realizar este proceso la frase se visualizará en la consola sin necesidad de configurarla previamente.

4. Una vez ingresada la frase, seleccionada el objetivo y audio se debe clic en enviar



5. Cuando se configuren todas las frases necesarias se debe dar clic en Finalizar, para desactivar el modo de configuración de comando de voz del controlador del Arduino, se debe visualizar en consola el siguiente texto:

*Configuración comando de voz finalizado...*



6. Para ver las frases que el controlador del Arduino tiene configuradas se debe dar clic en Ver Frases, con esto se desplegará en consola el Id del Objeto, la frase y el número de pista del audio configurado.



### 8.2.5. Detector de proximidad

La configuración del detector de proximidad se basa en la relación de la entrada: lectura sensor HC-SR04 y las salidas: movimientos del humanoide y/o audio, ver figura 41:

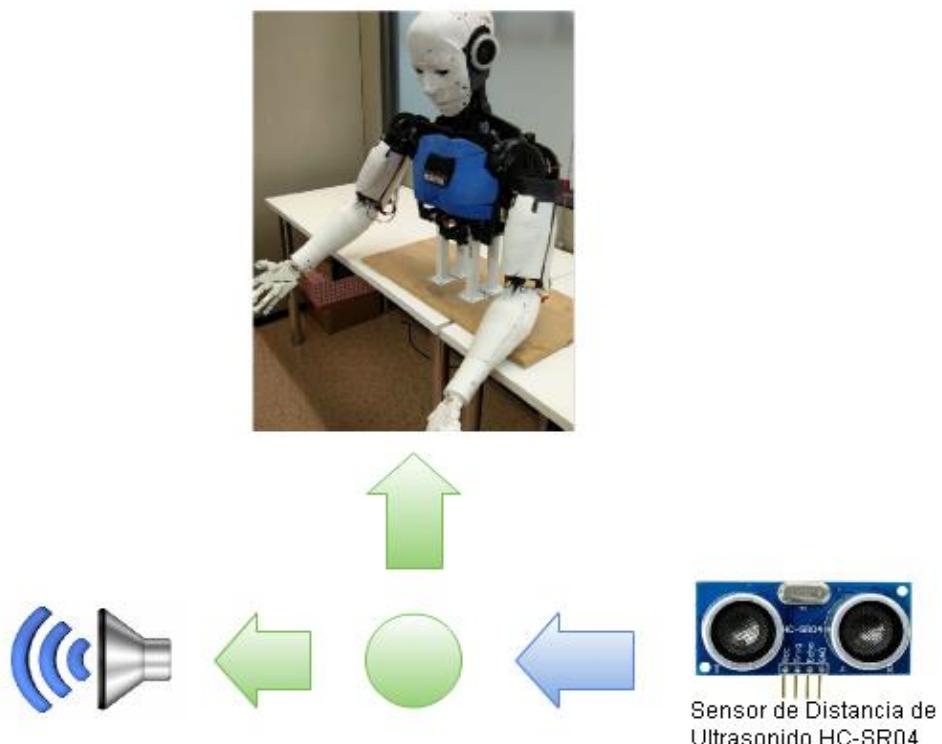


Figura 36: Diagrama detector de proximidad, fuente: elaboración propia

A continuación se muestran todas las opciones disponibles para configurar el detector de proximidad:

<b>Sensor</b> Activar     Desactivar	<b>Configuración</b> Activar     Finalizar     Ver Config.
Distancia mínima(cm): <input type="text" value="1"/>	
Distancia máxima(cm): <input type="text" value="1"/>	
Repetir cada Seg.: <input type="text" value="1"/>	
Objetivo:	<input type="text"/> <input type="text"/> Buscar objetivo
Audio:	<input type="text"/>
Enviar	

Para configurar el detector de proximidad se deben seguir los siguientes pasos:

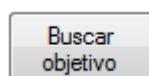
1. Clic en Activar esta opción habilita el controlador del Arduino en modo de configuración de detector de proximidad.



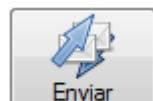
2. Cuando se está en modo de configuración de detector de proximidad, se deben ingresar los siguientes parámetros:

- *Distancia mínima*: valor o distancia mínima en centímetro para detección de objeto ubicado enfrente del sensor.
- *Distancia máxima*: valor o distancia máxima en centímetro para detección de objeto ubicado enfrente del sensor.
- *Repetir cada seg.* : Si el objeto se mantiene dentro del rango de detección el objetivo del humanoide y/o audio se repetirá durante el intervalo ingresado en este campo.
- *Objetivo*: es el conjunto de trayectorias a ejecutar en caso de identificar un objeto.
- *Audio*: es el sonido a reproducir en caso de identificar un objeto.

3. La opción Buscar Objetivo permite poder seleccionar un objetivo almacenado previamente



4. Una vez ingresado los parámetros de configuración se debe dar clic en enviar para construir la trama y enviarla al controlador del Arduino.



5. Cuando se ha finalizado el proceso de configuración de todos los rangos de detección de objetos se debe dar clic en Finalizar



6. El sensor por defecto está desactivado, se debe dar clic en Activar para que procese todas las configuraciones guardadas.



7. Si se desea desactivar detección de objetos, se puede dar clic en Desactivar



8. La opción Ver Config. Permite visualizar toda la configuración del sensor de proximidad almacenada en el controlador del Arduino



### **8.3. Especificaciones Técnicas de los Servomotores**

Especificaciones técnicas del Servomotor *Power HD 1235 MG*, obtenido de (Polulu, 2016):

#### Dimensiones

- Tamaño:       $59.5 \times 29.5 \times 54.3$  mm
- Peso:          170 g

#### Especificaciones:

- ¿Digital?:      Si
- Velocidad [6V]:    0.20 sec/60°
- Torque [6V]:      35 kg·cm
- Velocidad [7.4V]:   0.18 sec/60°
- Torque [7.4V]:      40 kg·cm
- Longitud del cable: 12 in

Especificaciones técnicas del Servomotor *Power HD 1218 MG*, obtenido de (Polulu, 2016):

#### Dimensiones

- Tamaño:       $40.3 \times 20.2 \times 37.2$  mm
- Peso:          64 g

#### Especificaciones:

- ¿Digital?:      Si
- Velocidad [6V]:    0.18 sec/60°
- Torque [6V]:      18.0 kg·cm
- Velocidad [7.4V]:   0.16 sec/60°
- Torque [7.4V]:      21.0 kg·cm
- Longitud del cable: 12 in

## 8.4. Especificaciones Técnicas del Módulo MP3 Player Mini

- Soporta frecuencias de muestreo (kHz): 8/11.025/12/16/22.05/24/32/44.1/48.
- 24 -bit DAC output, support for dynamic range 90dB, SNR support 85dB.
- Sistemas de archivos FAT16, FAT32 y tarjetas de hasta 32G.
- Se puede controlar por comandos seriales, señales digitales.
- Organización de archivos por carpeta. Soporta hasta 100 directorios y 255 pistas por directorio.
- 30 niveles de volumen y 6 -niveles de ecualización EQ.

En la figura 37 se muestra las entradas y salidas del módulo.

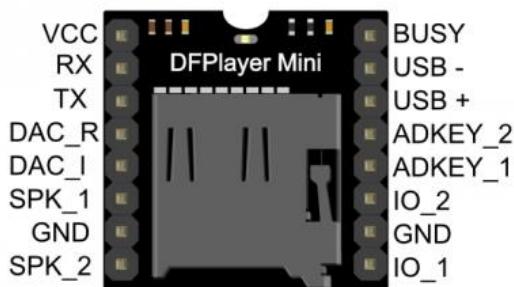


Figura 37: Módulo MP3 Player Mini, fuente: (DFRobot, 2016)

Número	Nombre	Descripción	Nota
1	VCC	Input Voltage	DC 3.2-5.0V; Typical: DC4.2
2	RX	UART serial input	
3	TX	UART serial output	
4	DAC_R	Audio output right channel	Drive earphone and amplifier
5	DAC_L	Audio output left channel	Drive earphone and amplifier
6	SPK2	Speaker	Drive speaker less than 3W
7	GND	Ground	Power Ground
8	SPK1	Speaker	Drive speaker less than 3W
9	IO1	Trigger port 1	Short press to play previous(long press to decrease volume)
10	GND	Ground	Power Ground
11	IO2	Trigger port 2	Short press to play next(long press to increase volume)

12	ADKEY1	AD port 1	Trigger play first segment
13	ADKEY2	AD port 2	Trigger play fifth segment
14	USB+	USB+ DP	USB Port
15	USB-	USB- DM	USB Port
16	Busy	Playing Status	Low means playing\High means no

Tabla 9: Pinout, mp3 player mini, fuente: (DFRobot, 2016)

Diagrama de conexión del módulo, Arduino y altavoces:

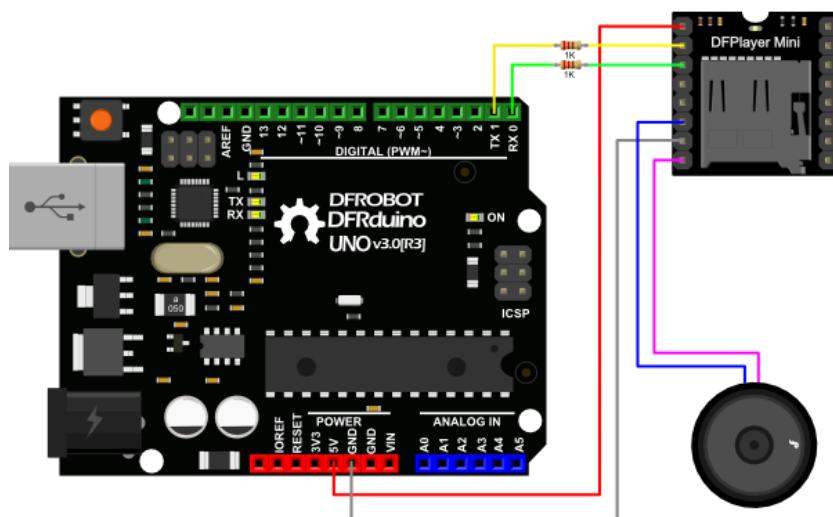


Figura 38: Diagrama de conexión: módulo, Arduino y altavoces, fuente: (DFRobot, 2016)

A continuación se muestra un ejemplo del uso del módulo *MP3 player mini*, obtenido de (DFRobot, 2016):

```
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>
SoftwareSerial mySerial(10, 11); // RX, TX

void setup () {
    Serial.begin (9600);
    mySerial.begin (9600);
    mp3_set_serial (mySerial);
    delay(1); // delay 1ms to set volume
    mp3_set_volume (15); // value 0~30
```

```

}

void loop () {
    boolean play_state = digitalRead(3); // connect Pin3 to BUSY pin
    of player

    if(play_state == HIGH) {
        mp3_next ();
    }
}

```

### Organización de audio en memoria:

Se debe tomar en cuenta lo siguiente:

2. En la microSD se debe crear el directorio llamado mp3.
3. El tipo de audio debe ser mp3, y su nombre debe iniciar con cuatro dígitos numéricos: por ejemplo:

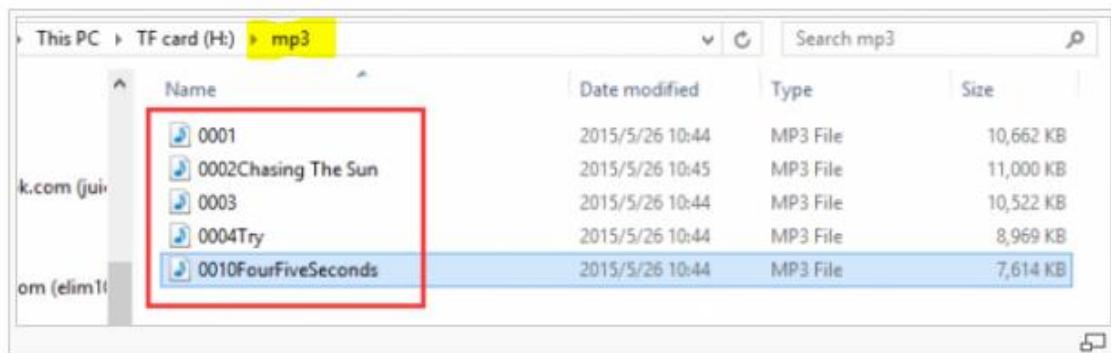


Figura 39: Ejemplo de nombre de audio, fuente: (DFRobot, 2016)

## 8.5. Especificaciones del Módulo Bluetooth HC-05

A continuación se detallan las características, aplicaciones, advertencias y configuración del módulo HC-05, información obtenida de (Electrónicos Caldas, 2016):

### **Características:**

- Especificación *bluetooth v2.0 + EDR (Enhanced Data Rate)*.
- Puede configurarse como maestro, esclavo, y esclavo con autoconexión (*Loopback*) mediante comandos AT.
- Chip de radio: CSR BC417143.
- Frecuencia: 2.4 GHz, banda ISM.
- Modulación: GFSK (*Gaussian Frequency Shift Keying*).
- Antena de PCB incorporada.
- Potencia de emisión:  $\leq$  4 dBm, Clase 2.
- Alcance 5m a 10 m.
- Sensibilidad:  $\leq$  -84 dBm a 0.1% BER.
- Velocidad: Asincrónica: 2.1 Mbps (max.)/160 kbps, sincrónica: 1 Mbps/1 Mbps
- Seguridad: Autenticación y encriptación (*Password* por defecto: 1234)
- Perfiles: Puerto serial Bluetooth
- Módulo montado en tarjeta con regulador de voltaje y 6 pines suministrando acceso a VCC, GND, TXD, RXD, KEY y status LED (STATE)
- Consumo de corriente: 50 mA
- El pin RX del módulo requiere resistencia de *pull-up* a 3.3 V (4.7 k a 10 k). Si el microcontrolador no tiene resistencia de pull-up interna en el pin Tx se debe poner externamente.
- Niveles lógicos: 3.3 V. Conectarlos a señales con voltajes mayores, como por ej. 5 V, puede dañar el módulo
- Voltaje de alimentación: 3.6 V a 6 V
- Dimensiones totales: 1.7 cm x 4 cm aprox.
- Temperatura de operación: -20 °C a +75 °C

### **Advertencias:**

- La comunicación Bluetooth entre dos módulos debe realizarse entre un módulo configurado como maestro y otro como esclavo. Para la comunicación *bluetooth* con computador, teléfono, PDA, tableta, etc., el módulo debe configurarse como esclavo.
- El nivel lógico de voltaje en las señales de comunicación es de 3.3 V. Conectarlos a señales con voltajes mayores, como por ej. 5 V, puede dañar el módulo.
- El pin RX del módulo requiere resistencia de *pull-up* a 3.3 V (4.7 k a 10 k). Si el microcontrolador no tiene resistencia de *pull-up* interna en el pin Tx se debe poner externamente.
- No conecte este dispositivo directamente a un puerto serial de computador, para hacer esto requerirá un conversor de serial TTL a RS232. También puede conectarse por USB utilizando un conversor USB a serial TTL.
- Para conexión vía inalámbrica con una computadora, la interface o adaptador *bluetooth* de esta debe soportar el perfil de puerto serie sobre *bluetooth*. También se puede conectar uno de estos módulos vía cable a la computadora y que se comunique con un segundo módulo.

### **Configuración:**

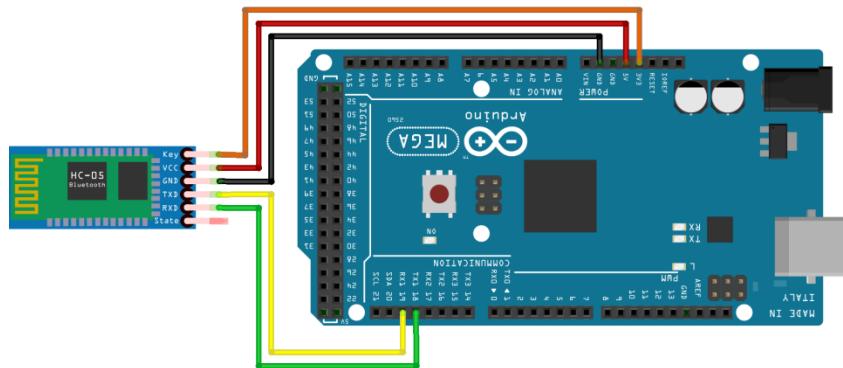
El módulo suele venir configurado como esclavo, con velocidad de transmisión serial de 38400 bps ó 9600 bps (Dependiendo del nivel de KEY al alimentarlo), 1 bit de parada, y sin bit de paridad, nombre: HC-05, contraseña: 1234

Para su configuración se puede conectar a el viejo puerto serial RS232 de la computadora a través de un convertidor TTL a RS232, o mejor empleando un conversor USB a serial TTL y utilizando el *Hyperterminal* de Windows u otro programa similar para enviar los comandos AT (Por ej. el SSCOM32, PuTTY, etc.). (A partir de Win Vista el *hyperterminal* ya no está incluido en el SO)

Por supuesto también se pueden enviar los comandos AT desde cualquier microcontrolador sin ayuda de computadoras.

Con Arduino de 3.3 V también se puede hacer fácilmente y sin ningún convertidor con un pequeño sketch que utiliza el monitor serial del IDE de Arduino para escribir los comandos AT y observar la respuesta del módulo. Como este monitor emplea la comunicación serial

que el Arduino utiliza para comunicarse con la computadora en los pines 0 y 1 digitales, se crea un puerto serial por software para pasar los datos al módulo Bluetooth empleando los pines digitales 10 y 11.



**Figura 40:** Conexión Arduino Mega-Bluetooth HC-05, fuente: (Adrian, 2015)

A continuación se muestra un ejemplo de uso del módulo *bluetooth* con Arduino, obtenido de (Adrian, 2015):

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10,11); // RX | TX
long bps=38400;
void setup() {
    Serial.begin(bps);
    BT.begin(bps);
    Serial.println("AT");
    BT.println("AT");
}
void loop() {
    if (BT.available()) {
        char c = BT.read();
        Serial.print(c);
    }
    if (Serial.available()) {
        char c = Serial.read();
        BT.print(c);
    }
}
```

## 8.6. Especificaciones Técnicas del Sensor de Distancia HC-SR04



Figura 41: Sensor de Distancia de Ultrasonido HC-SR04

### Características

- Dimensiones del circuito: 43 x 20 x 17 mm.
- Tensión de alimentación: 5 Vcc.
- Frecuencia de trabajo: 40 KHz.
- Rango máximo: 4.5 m.
- Rango mínimo: 1.7 cm.
- Duración mínima del pulso de disparo (nivel TTL): 10  $\mu$ s.
- Duración del pulso eco de salida (nivel TTL): 100-25000  $\mu$ s.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 ms.

### Pines de conexión:

- VCC.
- Trig (Disparo del ultrasonido).
- Echo (Recepción del ultrasonido).
- GND.

Para el cálculo de la distancia en metros se aplica la siguiente fórmula:

$$\text{Distancia} = [(\text{Tiempo entre Trig y el Echo}) * (\text{Velocidad del Sonido } 340 \text{ m/s})]/2$$

## 8.7. Audios pre-grabados

Actualmente están pregrabados los siguientes audios:

Código	No. Pista	Audio
2	-1	Sin Sonido
3	5	Bien
5	4	Gracias
6	1	Si
7	3	Hola
8	2	No
11	6	Nada
12	7	No quiero
13	8	Mi nombre es Auron
14	9	Quien eres
15	10	Soy Hombre
16	11	Y tu
17	12	El hondureño Óscar Hernández
18	13	Grupo de investigación AUROVA
19	14	No entiendo lo que dices
20	15	Universidad de Alicante
21	16	Ven, acércate un poco

Tabla 10: Audios pre-grabados