



POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es



Juan Miguel Sequeira Da Silva

05 TRABAJO FIN DE MASTER

INDUSTRIALES

TRABAJO FIN DE MASTER

SOFTWARE DE RECONOCIMIENTO FACIAL, TRACKING DE MOVIMIENTOS, PUPILAS Y APERTURA DE BOCA Y DISEÑO Y CONSTRUCCIÓN DE UN ROBOT HUMANOIDE PARA INTERPRETAR LOS MOVIMIENTOS

SEPTIEMBRE 2018

Juan Miguel Sequeira Da Silva

DIRECTOR DEL TRABAJO FIN DE MASTER:
Roque Jacinto Saltarén

TRABAJO FIN DE MASTER
PARA LA OBTENCIÓN DEL
TÍTULO DE MASTER EN
INGENIERÍA INDUSTRIAL



POLITÉCNICA

**UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES**



“SOFTWARE DE RECONOCIMIENTO FACIAL, TRACKING DE MOVIMIENTOS, PUPILAS Y APERTURA DE BOCA Y DISEÑO Y CONSTRUCCIÓN DE UN ROBOT HUMANOIDE PARA INTERPRETAR LOS MOVIMIENTOS”

Autor: Juan Miguel Sequeira da Silva

Tutor: Roque Jacinto Salterén

Septiembre 2018

Máster universitario de robótica

RESUMEN

El presente trabajo trata del desarrollo del software y el hardware de un robot social que puede ser utilizado para futuros desarrollos. El alcance del proyecto incluye el diseño mecánico, estructural y eléctrico de la cabeza robotizada a partir del cuello, incluyendo 3 ejes de rotación en la base del cuello, 2 ejes de rotación para el movimiento vertical y horizontal de los ojos y un eje para la apertura de la boca,

Adicionalmente se creó una GUI de matlab para el control del robot, es una interfaz de usuario donde se pueden ver los parámetros del robot y usarlo en modo manual o en automático.

El uso de las articulaciones del robot puede ser controlado con cualquier lenguaje de programación y una librería para la conexión con arduino y control de señales. De este modo, se pondrá el robot en disposición de otros usuarios que deseen complementar el funcionamiento del mismo, con otros algoritmos para lograr otros comportamientos o estudios referentes a la visión artificial.

El robot es capaz de reconocer la cara de la persona que tenga frente la cámara, hacer tracking de la posición de la cara y su inclinación con respecto al eje vertical, seguir e imitar el movimiento de las pupilas y la apertura de la boca.

Palabras clave: Robot, visión artificial, detección facial, detección de gestos.

ÍNDICE

Contenido

RESUMEN	1
ÍNDICE.....	2
ÍNDICE DE FIGURAS	3
1. ABREVIATURAS, UNIDADES Y ACRÓNIMOS	4
2. INTRODUCCIÓN	5
3. ESTADO DEL ARTE	6
-Honda P1:.....	6
-Honda P2:.....	7
-Honda P3:.....	7
-Nexi:	8
-Robot Nao:	9
-Robot Sophia:.....	10
-Robot Pepper:	11
4. OBJETIVOS	13
General:.....	13
Hardware:.....	13
Software:	14
5. METODOLOGÍA.....	15
Interfaz de usuario	15
Algoritmo de reconocimiento	17
Tracking.....	19
Ejecución de movimientos	20
Flujograma del algoritmo	22
Hardware - Electrónica.....	23
Hardware - Robot.....	27
6. RESULTADOS Y DISCUSIÓN.....	33
7. LINEAS FUTURAS	34
8. BIBLIOGRAFIA.....	35
9. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO	36
10. GLOSARIO	39
11. PLANOS.....	40

ÍNDICE DE FIGURAS

Figura 3. 1. Honda P1.	7
Figura 3. 2. Honda P2.	7
Figura 3. 3. Honda P3.	8
Figura 3. 4. Robot Nexi.	9
Figura 3. 5. Robot Nao.....	10
Figura 3. 6. Robot Sophia.	11
Figura 3. 7. Robot Pepper.....	12
Figura 5. 1. GUI de Matlab	15
Figura 5. 2 Representación de la GUI en curso.	17
Figura 5. 3 Tipos de características usadas por Viola y Jones. F1 y F2: características compuestas por dos rectángulos. F3 y F4: características compuestas por tres rectángulos. F5: características de cuatro rectángulos.	18
Figura 5. 4 Ejemplo de las características utilizadas en la detección de caras.....	18
Figura 5. 5 Detección de características.....	19
Figura 5. 6 Comparación de la nube de puntos utilizada para el tracking.	20
Figura 5. 7 Comparación en la determinación del ángulo de inclinación de la cabeza.	21
Figura 5. 8 Detección de las pupilas. Figura superior izquierda: imagen real. Figura superior derecha: imagen preprocesada. Figura inferior: cálculo de la posición de las pupilas.....	22
Figura 5. 9 Ejemplo de detección de apertura de la boca. Izquierda abierta, derecha cerrada.	22
Figura 5. 10 Flujograma de tareas del algoritmo.	23
Figura 5. 11 Esquemático de control de señales para servomotores.	24
Figura 5. 12 Detalle de conexión al arduino y a la fuente de voltaje.	25
Figura 5. 13 Arduino Nano.	26
Figura 5. 14 Esquema de conexión eléctrico.....	27
Figura 5. 15 Fotografía del robot construido.	28
Figura 5. 16 Vista frontal del montaje de los servos de movimiento horizontal, vertical y angular.....	29
Figura 5. 17 Vista lateral de la montura de los servos para el movimiento vertical, horizontal y angular.....	29
Figura 5. 18 Base articulada del robot.	30
Figura 5. 19 Mecanismo de apertura de la boca y movimiento de los ojos en el plano.....	31
Figura 5. 20 Vista trasera.....	32
Figura 9. 1 Tabla de componentes.	36
Figura 9. 2 Listado de tareas.	37
Figura 9. 3 Diagrama de Grantt del proyecto.....	38

1. ABREVIATURAS, UNIDADES Y ACRÓNIMOS

MSAC- M Estimator Sample Consensus

RANSAC – Random Sample Consensus.

PWM- Pulse Width Modulation

PLA - Ácido Poliláctico

2. INTRODUCCIÓN

En el máster de automática y robótica comprendimos la importancia y la utilidad de las aplicaciones que utilicen visión artificial. En la industria, muchos de los procesos de automatización se realizaban mediante la implementación de sensores de distintos tipos y lograr captar el mundo real y traducirlo en un lenguaje que una máquina pudiera seguir para llevar a cabo determinado trabajo. Con uso de cámaras, lentes y programación adecuada, se pueden reducir los costes de los procesos automatizados, consiguiendo de igual forma una fiabilidad y rendimiento óptimo. Incluso fuera de la industria, se han desarrollado aplicaciones de control ambiental con costes bajos y una utilidad muy amplia.

Es por ello que he dedicado este TFM a la visión artificial y el diseño de robots de bajo coste, además, creo firmemente que los robots sociales podrán ser incluidos en la sociedad con total normalidad y ayudarán en tareas diarias. El reconocimiento facial y la identificación de personal jugarán un rol importante en la seguridad y se logrará llevar un control más específico de asistentes en lugares públicos, haciendo frente también a amenazas terroristas o situaciones semejantes.

La finalidad de la construcción de este robot es consignar una base confiable y fácil de manipular mediante el software proporcionado o cualquier otro lenguaje de programación, en el cual puedan desarrollarse y probarse algoritmos de visión artificial de cualquier índole, profesional o no.

En este trabajo se explicarán las técnicas utilizadas para el reconocimiento de la cara, los ojos y la boca, así como el procesamiento previo de la imagen y los posteriores para obtener un tiempo de respuesta eficaz. Además, se explicará el ensamblado del robot y sus distintos componentes. También se propondrán distintas aplicaciones para el uso de dicho robot y posibles mejoras que pueden hacerse en el futuro.

3. ESTADO DEL ARTE

Antes de iniciar este proyecto, se hizo una extensa búsqueda de soluciones parecidas para el mismo problema. Se analizaron distintas aplicaciones, métodos y hardware. Con ello, se decidió cual era el alcance de este proyecto y sus posibles aplicaciones.

Primeramente haremos un análisis de la realidad del reconocimiento facial. Consiste básicamente en un algoritmo que detecta la forma de una cara en una imagen digital. Se pueden utilizar métodos holísticos, métodos de comparación de textura de la piel y métodos locales o geométricos, estos últimos presentan el desempeño menos óptimo. Su uso comienza comercialmente con la detección automática en cámaras fotográficas para mejorar la captura en las zonas de la cara y luego se fue extendiendo para otras aplicaciones como la biometría, controles de acceso y seguridad de información, tarjetas inteligentes y vigilancia policial y en cámaras de circuito cerrado. Es entonces donde separamos la detección de una cara con la identificación de dicha cara comparando características en una base de datos.

Los sistemas biométricos han sido poco confiables pero con el tiempo se han mejorado y han llegado a formar parte de etapas iniciales en redes de seguridad. Por ejemplo, los móviles más modernos utilizan técnicas 3D de identificación y comparación con muy pocos elementos de una base de datos, esto lo hace muy confiable. Por otro lado, en el control de mucho personal, ya comienza a bajar las estadísticas de reconocimientos positivos.

Sin embargo, la intención de este trabajo no es inclinarse hacia la biometría e identificación, aunque es considerado como una de las extensiones posibles que podrían realizarse en el futuro. A continuación se explicará en detalle algunos de los desarrollos tecnológicos pasados que inspiraron este trabajo.

-Honda P1:

Es el primer robot humanoide modelo de Honda con una altura total 1.9mt y un peso de 175 kg. Cuenta con una fuente de alimentación externa, y un ordenador a bordo para realizar acciones como encender o apagar un interruptor exterior, sostener el asa de una puerta o transportarse agarrando cosas. Se realizó un estudio sobre el movimiento coordinado del brazo y la pierna pero el robot no se considera independiente. En la figura 1 vemos una foto tomada en el año de su creación, 1993.



Figura 3. 1. Honda P1.

-Honda P2:

El primer robot humanoide bípedo y autónomo del mundo, terminado en diciembre de 1996. Inalámbrico y construido con todos los equipos necesarios como computadora, unidad de motor, batería, equipo de radio, y otros, todo ensamblado en la parte del cuerpo, tiene una altura total de 1.8mt, peso de 210kg. Es capaz de realizar acciones como caminar libremente, subir y bajar escaleras, empujar una plataforma móvil todo de forma inalámbrica y por operación automática. Figura 2.



Figura 3. 2. Honda P2.

-Honda P3:

Honda fue pionera en robótica humanoide amigable gracias a la investigación realizada en diversos prototipos hasta la aparición de su ASIMO. Posteriormente, y basándose en las lecciones aprendidas con el ASIMO durante más de una década decidieron diseñar diferentes tipos de humanoides,

adaptados a las características de su entorno. El robot P3 de Honda es un humanoide de 1,60 metros que pesa 130 kilos. El robot está construido en magnesio y no cuenta con una columna vertebral flexible, se le han situado brazos muy largos para poder alcanzar sitios lejanos sin necesidad de doblar el torso. Es el primer robot humanoide completamente independiente. Figura 3.



Figura 3. 3. Honda P3.

Se puede observar el parecido increíble entre estos tres humanoides de Honda, son la clara representación de una década de trabajo en cadena que dio pie a otros desarrollos en el área.

-Nexi:

Consiste solo de una cabeza robótica, el Nexi MDS es un robot manipulador móvil con habilidades expresivas bastante altas. Está diseñado para viajar sobre ruedas en un Segway, pero actualmente utiliza un conjunto adicional de ruedas de apoyo para operar como una plataforma estáticamente estable en su etapa inicial de desarrollo. Tiene manos para manipular objetos, ojos (videocámaras), oídos (una variedad de micrófonos), una cámara de infrarrojos 3-D y un telémetro láser para permitir el seguimiento en tiempo real de objetos, personas y voces, así como de la navegación en interiores.

El desarrollo de Nexi fue liderado por Personal Robots Group del MIT Media Lab, en colaboración con el profesor Rod Grupen de la Universidad de Massachusetts-Amherst. El proyecto fue financiado por un premio del Programa de Instrumentación de Investigación de la Universidad de Defensa Naval de Investigación de Defensa (DURIP) para desarrollar una nueva clase de robots que pueden involucrarse en formas sofisticadas de trabajo en equipo

de igual a igual con humanos en entornos inciertos. Tiene como objetivo desarrollar tecnologías y demostraciones para equipos compuestos por humanos y robots aéreos autónomos. Se puede ver en la figura 4 como aumenta el parecido con los humanos sin dejar de lado un aspecto caricaturesco y agradable a la vista.

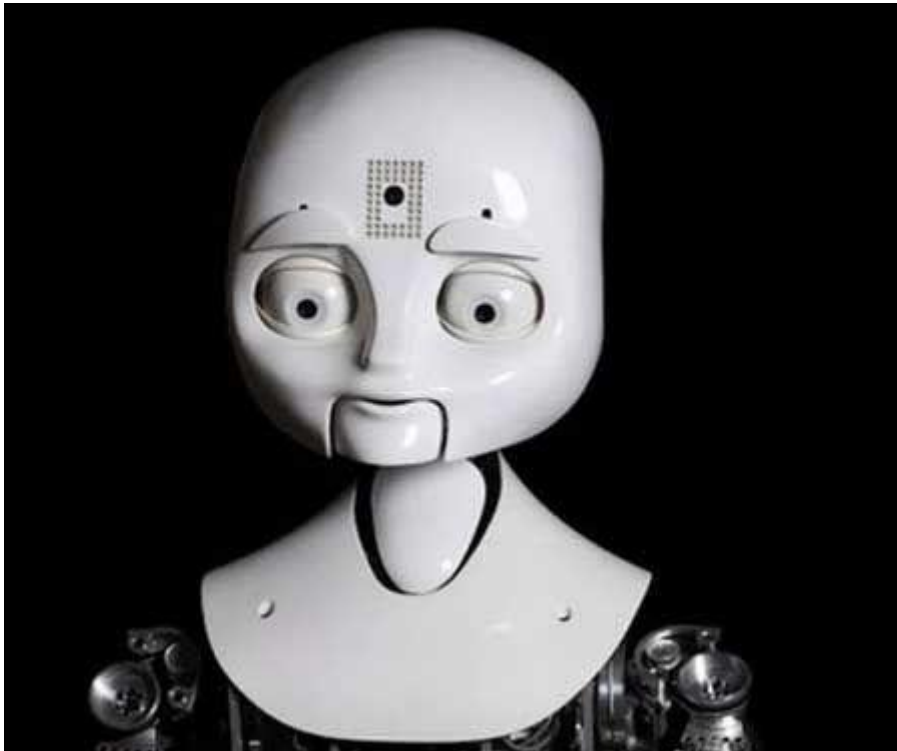


Figura 3. 4. Robot Nexi.

-Robot Nao:

Nao es un robot programable y autónomo, desarrollado por Aldebaran Robotics. El desarrollo del robot comenzó en 2004 y la edición Académica de Nao fue desarrollada para las universidades y laboratorios con fines de investigación y educación. Fue lanzado a las instituciones en 2008, y se puso a disposición del público antes de 2011. El robot ya ha entrado en uso en numerosas instituciones académicas de todo el mundo.

En diciembre de 2011, se lanzó Nao Next Gen, con mejoras de hardware y software, tales como cámaras de alta definición, una mayor robustez, sistemas anticolidión y una velocidad al caminar más rápida. Desde 2011, más de 200 instituciones académicas de todo el mundo han hecho uso del robot.

Las diferentes versiones de la plataforma del robot Nao tienen como característica de 14, 21 o 25 grados de libertad (DOF). Para la competición Robocup se creó un modelo especializado con 21 DOF y sin manos

accionadas. Todas las versiones de Nao Académico ofrecen una unidad de medición inercial con acelerómetro, girómetro y cuatro sensores de ultrasonidos que le proporcionan a Nao estabilidad y posicionamiento en el espacio. Las versiones de las piernas incluyeron ocho resistencias de detección de fuerza y dos topes.

El robot Nao también cuenta con un sistema multimedia a bordo basado en Linux, incluyendo cuatro micrófonos (para el reconocimiento de voz y la localización del sonido), dos altavoces (para la síntesis de texto a voz) y dos cámaras de alta definición (para visión artificial, como el reconocimiento facial y de formas). El tamaño es de 58cm de alto y su aspecto es parecido al Nexi, como se puede ver en la figura 5.



Figura 3. 5. Robot Nao.

-Robot Sophia:

Sophia es un robot humanoide desarrollado por la compañía Hanson Robótica. Ha sido diseñada para aprender y adaptarse al comportamiento humano y trabajar con humanos, y ha sido entrevistada por todo el mundo. En octubre 2017, se convirtió en una ciudadana saudí, siendo así el primer robot con ciudadanía de un país. Según ella, Sophia fue activada el 19 de abril de 2015. Está inspirada en la actriz Audrey Hepburn, y es conocida por su aspecto y comportamiento humano en comparación con variantes robóticas anteriores. Según el fabricante, David Hanson, Sophia tiene inteligencia artificial, procesamiento de datos visuales y reconocimiento facial. Sophia también imita gestos humanos y expresiones faciales y es capaz de contestar ciertas preguntas y tener conversaciones sencillas sobre temas predefinidos. El robot usa tecnología de reconocimiento de voz y está diseñada con capacidad de aprendizaje. Su software de IA analiza conversaciones y extrae datos que le permite mejorar sus respuestas con el tiempo. Su aspecto humanoide (figura 6)

es muy real y puede ser encontrado como incomodo para algunos espectadores.



Figura 3. 6. Robot Sophia.

-Robot Pepper:

Pepper es un robot humanoide de 120 centímetros, figura 7, programable y diseñado para interactuar con personas. Su tecnología le permite detectar tanto el lenguaje verbal como el no verbal, la posición de la cabeza y el tono de voz, para reconocer el estado emocional e individualizar cada interacción. Esto provoca un sentimiento de empatía y una conexión entre robot-usuario que favorecen una comunicación eficaz.

El robot Pepper supone un gran paso evolutivo hacia la inserción de los robots en entornos comerciales. A través de sus sensores, cámara en 3D y sus cuatro micrófonos, el humanoide se comunica de forma fluida tanto de forma individual como hacia un público numeroso.

Presenta movimientos muy flexibles, imitando los gestos humanos, y se desplaza en cualquier dirección hasta 3 km/h. Acompaña todas sus intervenciones con un elaborado lenguaje corporal totalmente programable para intensificar sus discursos e interacciones

Pepper es el primer humanoide diseñado específicamente para convivir y relacionarse con seres humanos. Puede interpretar el estado de ánimo de las personas de su entorno y modificar su comportamiento en base a su programación. Además, su forma de interactuar es completamente intuitiva y amigable puesto que reconoce gestos, sonidos, expresiones y tacto.

La interactividad es el concepto principal en el diseño de Pepper. Sus múltiples sensores táctiles y de sonido, junto a la cámara 3D, le permiten registrar e interpretar de forma detallada el entorno. Además, su pantalla táctil transmite y

recibe información. Este potente hardware ofrece múltiples opciones a la hora de programar y configurar el robot para que interactúe con su entorno.

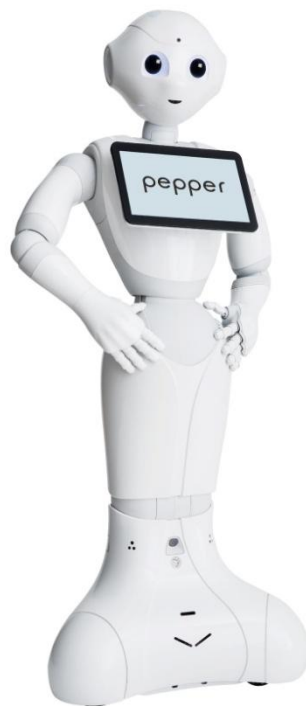


Figura 3. 7. Robot Pepper.

4. OBJETIVOS

Los objetivos de este trabajo serán desglosados según su índole:

General:

- Diseñar, fabricar y programar un robot social con aspecto humanoide de tamaño real para probar algoritmos de visión artificial.

Hardware:

- Diseñar un robot que pueda ser ensamblado sin dificultad, cuyos elementos principales puedan ser adquiridos a bajo coste y de muy fácil acceso.
- Utilizar una impresora 3D para la fabricación de las piezas a partir de PLA ecológico y de alta rigidez.
- Generar una lista de materiales de ensamblado necesarias para la construcción y el armado del robot
- Diseñar los mecanismos de acción y movimiento de cada articulación del robot para que cada movimiento sea lo más humanoide posible. También, es deseable que la construcción sea duradera y resistente, ya que serán tomados en cuenta los esfuerzos y la fatiga que recibirán los componentes así como las fuerzas transversales que sufrirán los ejes de los servo motores y las torsiones debido a las fuerzas flectoras.
- Diseñar los mecanismos de movimiento de los ojos en dirección vertical y horizontal y el mecanismo de apertura de la boca en dirección vertical.
- Ser capaz de reproducir los movimientos sin tener exceso de vibraciones en el sistema debido al paso de los engranajes de los servomotores o debido a la precisión baja de los *encoder* de los mismos.
- Utilizar la cara de un robot humanoide *open source* disponible y adaptarlo para nuestros fines.
- Diseñar un circuito sencillo de control y alimentación de los servos utilizando una fuente de 5VDC, premeditar el uso de condensadores y

resistencias para limitar la aceleración de los servos y lograr un movimiento mas suavizado.

Software:

- Escribir un algoritmo por etapas para el reconocimiento facial, de pupilas y de boca mediante distintas técnicas y con el uso de métodos de procesamiento de imágenes.
- Generar un método confiable de captura de imágenes para distintos dispositivos, de velocidad de procesamiento adecuada y con versatilidad de adaptación para dispositivos mediante matlab.
- Integrar un algoritmo de tracking a los métodos de detección implementados y reducir así el tiempo de procesamiento y aumentar la fiabilidad del algoritmo en general.
- Lograr transmitir la información captada mediante visión artificial en comandos para los servomotores con la finalidad de imitar los movimientos capturados de una manera suavizada y sutil.
- Diseñar una interfaz de usuario versátil e intuitiva para el uso del robot, así como generar un algoritmo de fácil comprensión para los futuros programadores del robot.
- Desarrollar una interfaz visual donde se revelen fácilmente las propiedades de la imagen captadas por el algoritmo, así como hacerlas accesibles o no en el algoritmo con la intención de reducir el tiempo de procesamiento.
- Utilizar distintos métodos de captura de imágenes, incluyendo el uso de móviles mediante aplicaciones existentes en la web.

5. METODOLOGÍA

En esta sección se explicará con detalle la metodología utilizada para resolver las funcionalidades y retos que surgieron durante la elaboración de este proyecto. Primeramente se explicarán aquellos métodos referentes a los algoritmos utilizados y luego aquellos relacionados a la mecánica o diseño del robot.

Interfaz de usuario

Se utilizó la herramienta Matlab para la programación de los algoritmos debido a su forma intuitiva de programar y la cantidad de librerías existentes para facilitar el trabajo. Se desarrolló la GUI de la figura siguiente:

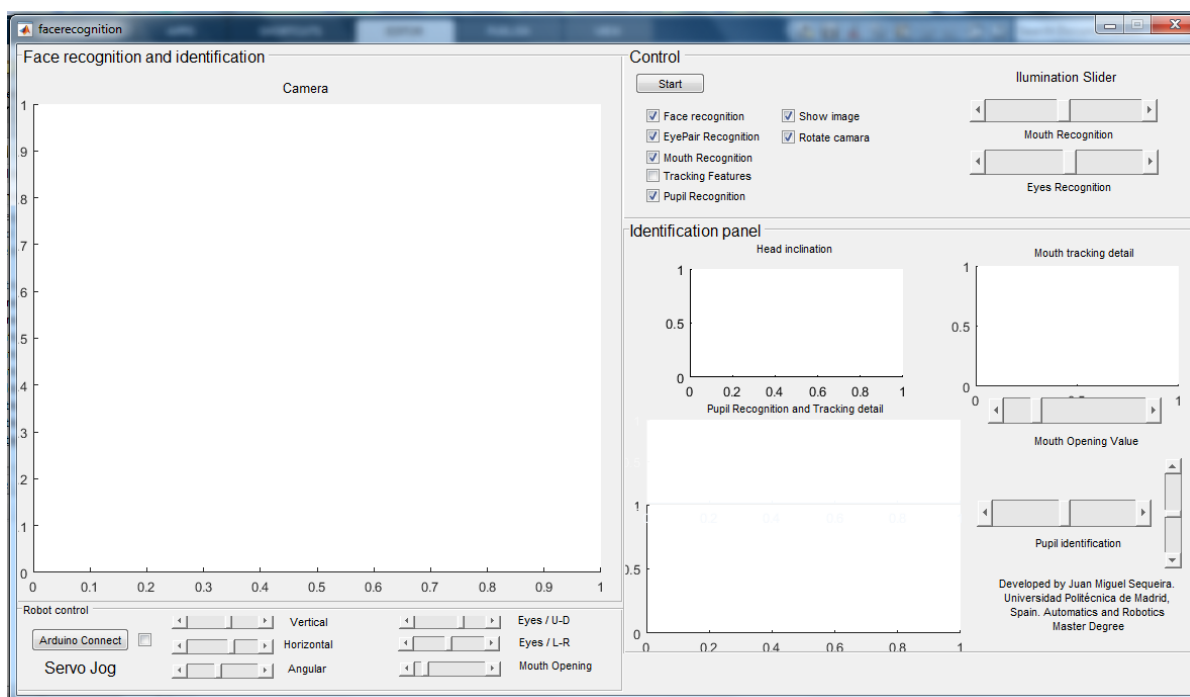


Figura 5. 1. GUI de Matlab

En ella podemos identificar 4 paneles distintos. El primero a la izquierda y superior es la ventana gráfica donde se puede visualizar la imagen obtenida de la cámara y las modificaciones realizadas a la imagen, así como los recuadros de la información reconocida o las propiedades de las figuras. En el panel

inferior derecho se puede mover al robot independientemente de la captura de la cámara, primeramente se pulsa el botón de conexión de arduino para establecer comunicación con el arduino, directamente conectado al USB del pc y del puerto previamente establecido. Una vez conectado, automáticamente el robot toma su posición inicial, en el cual se establecen como parámetros actuales el valor medio del rango posible de movimiento para servo. Todos los servos son de 200° pero el programa solo permite colocar cada servo dentro del rango establecido, así como modifica los valores máximos y mínimos de los slider, en caso de que se llegaran a cambiar esos valores, ya sea por el desarmado del robot o por el cambio de alguno de los servos. Con lo cual, mover el slider en la GUI moverá dicha articulación del robot dentro de los límites preestablecidos.

En el panel superior derecho se seleccionan los elementos de control y procesamiento que se deseen. Entre ellos está la detección de la cara, la detección del área del par de ojos, la detección de la apertura de la boca, la representación de las características encontradas en el algoritmo de tracking, el reconocimiento de las pupilas así como la representación en los paneles posteriores, la visualización de la imagen final y la rotación de la imagen en caso de que sea necesario.

Cada checkbox inhibe o permite el estudio o procesamiento de imagen de la característica en cuestión y además elimina la representación de los resultados encontrados. El checkbox para mostrar la imagen, permite estudiar los parámetros de la imagen pero no implementa la muestra de los resultados y además congela la imagen. Si este checkbox está sin tildar, la velocidad de procesamiento general del algoritmo aumenta considerablemente y los movimientos del robot pueden llegar a ser más suaves.

En este mismo panel, se encuentran dos slider para establecer los umbrales de inversión de la imagen y con ellos lograr resultados más acertados. Sin embargo, mientras corre el algoritmo, estos valores se ajustan automáticamente para adaptarse a las condiciones de luz del entorno y a las propiedades de la cámara utilizada.

En el panel inferior derecho se muestra la información de los resultados obtenidos representados numéricamente. El diagrama superior muestra el ángulo de inclinación de la cabeza con respecto al eje perpendicular de la cámara. El diagrama del medio muestra la imagen inversa ya procesada de la detección de los ojos, en ella podremos estimar la facilidad de encontrar el centro de las pupilas y para mejorarlo, podemos mover el slider del panel superior a la izquierda o derecha (disminuir umbral o aumentar, respectivamente) según sea necesario. El diagrama inferior muestra un plano cartesiano donde se representa con dos círculos azules el reconocimiento actual de las pupilas. El último diagrama del panel representado a la derecha, muestra la imagen inversa preprocesada del área de la boca ya detectada. A la

derecha inferior de este panel se encuentran 3 slider que representan la cantidad de apertura de la boca y la posición de las pupilas según el eje vertical u horizontal encontrado, mediante la relación de distancia que aporta la detección de la zona de los ojos.

La siguiente figura muestra la GUI en ejecución, con lo cual se puede hacer una idea del funcionamiento de la misma.

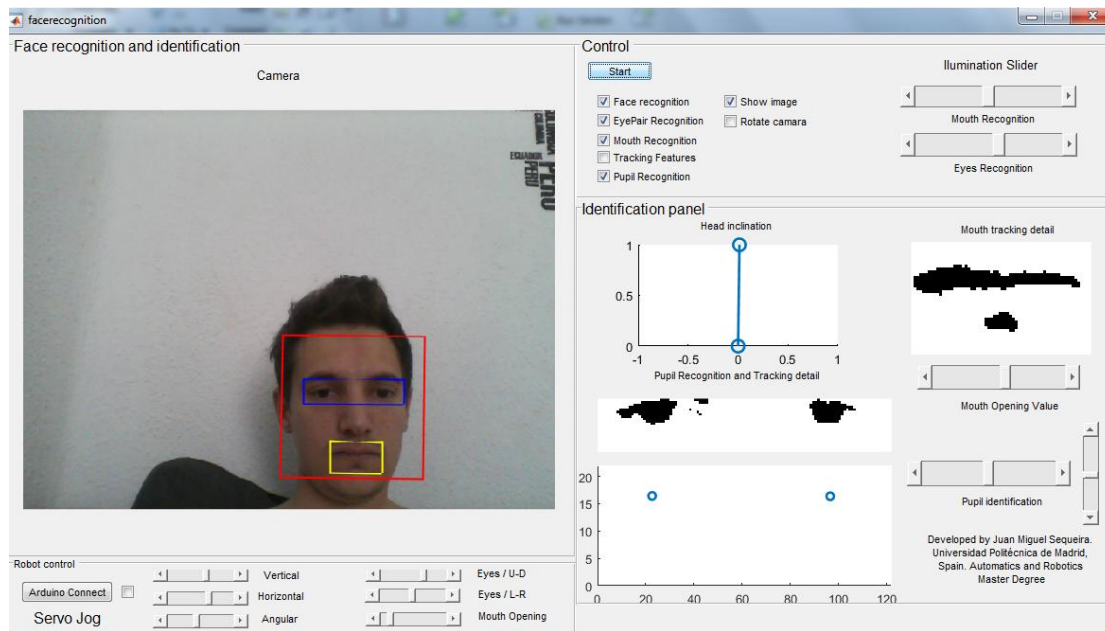


Figura 5. 2 Representación de la GUI en curso.

Algoritmo de reconocimiento

Una vez se pulsa el botón *Start* se establecen los parámetros iniciales predeterminados para poder comenzar el algoritmo, se coloca al robot en posición inicial y se abre el canal de video. Se toma una imagen y el primer paso es intentar encontrar una cara en la totalidad de la imagen. Esto se realiza mediante un algoritmo de Viola Jones.

Este algoritmo tiene un coste computacional muy bajo, sin embargo, arroja resultados positivos con un 99% de fiabilidad para los verdaderos positivos y solo el 3.3% de falsos positivos. Utiliza un sistema de detección en cascada mediante la transformación de una imagen real en escala de grises en una imagen integral. Dependiendo de los factores iniciales definidos como el área esperada de una cara dentro de la imagen en píxeles, el algoritmo divide en subsectores la imagen y analiza la posibilidad de encontrar una cara en cada sector, si no hay posibilidad elimina a dicho sector para el análisis en cascada al siguiente nivel, donde estudia nuevamente cada sector por separado.

En el trabajo de Viola y Jones [1], se utilizan clasificadores para distinguir objetos que son cara de objetos que no lo son, es decir, $K = 2$, siendo K la cantidad de clases de objetos en una imagen. Luego se construyen los clasificadores utilizando una variante de características conocidas como características de Haar-like features [2]. Reciben este nombre ya que su resultado es similar al de aplicar otros de Haar sobre la imagen, solo que éstos pueden calcularse más eficientemente. Están definidos por dos, tres o cuatro rectángulos y su cálculo consiste en hallar la diferencia entre la integral de la imagen en las zonas blancas y la integral de la imagen en las zonas negras.

Denominamos $\mathbb{F}$ al conjunto de tipos de características posibles:

$$\mathbb{F} = \left\{ F_1 : \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, F_2 : \begin{array}{|c|} \hline \square \\ \hline \end{array}, F_3 : \begin{array}{|c|c|} \hline \square & \blacksquare \\ \hline \end{array}, F_4 : \begin{array}{|c|c|} \hline \blacksquare & \square \\ \hline \end{array}, F_5 : \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array} \right\}$$

Figura 5. 3 Tipos de características usadas por Viola y Jones. F_1 y F_2 : características compuestas por dos rectángulos. F_3 y F_4 : características compuestas por tres rectángulos. F_5 : características de cuatro rectángulos.

Las características de Haar capturan muchas de las características presentes en una cara, por ejemplo, la zona de los ojos suele ser más oscura que la zona superior de las mejillas y el puente nasal suele ser más claro que los ojos. A este tipo de características, se les llama características simples por ser muy sencillos de calcular. Aportan poca información en forma individual, pero poseen la característica de poder calcularse en forma extremadamente veloz, en orden constante, y se pueden combinar para formar clasificadores que brinden mayor información. La etapa de entrenamiento descrita por Viola y Jones consiste justamente en esto, en formar clasificadores que sean capaces de discriminar entre caras y no caras, combinando otros clasificadores muy simples previamente construidos.

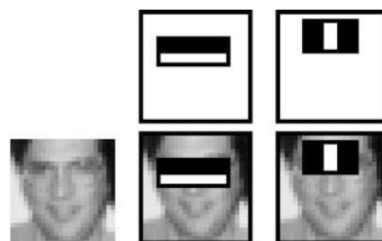


Figura 5. 4 Ejemplo de las características utilizadas en la detección de caras.

En la imagen se puede ver que la zona de los ojos es más oscura que la zona superior de las mejillas y el puente nasal es más claro que los ojos. Mediante el uso de estas características y un algoritmo AdaBoost se construye un clasificador [3].

Una vez encontradas las zonas de resultado positivo, se hace una relación entre la imagen integral y la imagen real (en píxeles) y se obtiene una matriz de

dimensiones $M \times 4$, siendo M la cantidad de caras encontradas y los cuatro valores corresponden a la posición del rectángulo que contiene la cara y sus dimensiones.

Si se obtiene información positiva de varias caras, se trabaja con la de mayor dimensión (la cara más cercana). Seguidamente y con el mismo método de detección pero con características distintas en el constructor, se establecen los parámetros del clasificado de la boca y del par de ojos. En paralelo, se ejecutan los dos clasificadores y se obtienen los rectángulos en los cuales se encuentran dichos objetos. De esta manera, podemos lograr la detección inicial de la figura 5.5.

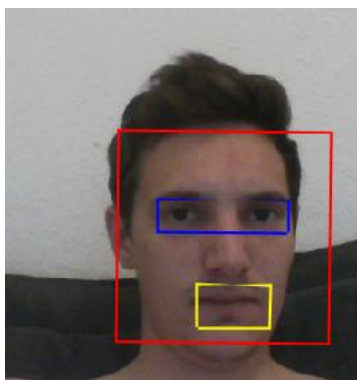


Figura 5. 5 Detección de características.

Tracking

Una vez detectada la cara principal, el algoritmo entra en el modo de tracking, para ello busca las características de Eigen, que distinto a un clasificador, busca por grupo de píxeles elementos contrastantes con los cual se puedan crear referencias fiables para buscar en la imagen siguiente. De esta manera, crea un mapa de puntos ubicado en un plano, luego estima la transformación geométrica que sufre la nube de puntos y arroja una matriz de transformación lineal.

Para ello, se utiliza un algoritmo de tracking con MSAC (Estimador de muestras consensuadas) que se basa en el método de RANSAC (Muestro consensuado aleatorio), que calcula los parámetros de un modelo matemático de un conjunto de datos observados y elimina, con cierto rango de permisibilidad, los valores atípicos del muestreo. Evidentemente, es un método no determinista utilizando probabilidades y arrojando resultados certeros. El método pasa por los siguientes pasos: selección de un conjunto aleatorio de los datos originales, realizar un nuevo conjunto de datos, comparar todos los datos originales con el modelo creado recientemente y consensuar definitivamente aquellos que cumplan con el modelo (se considera un buen subconjunto aquellos que logren obtener mayor cantidad de datos consensuados), y por último se realiza

nuevamente el procedimiento pero con aquellos puntos consensuados en el modelo aleatorio. Este proceso se realiza un numero finito de veces [4].

La GUI contiene un elemento de control para visualizar las características usadas en el tracking, un ejemplo se muestra en la figura 5.6.

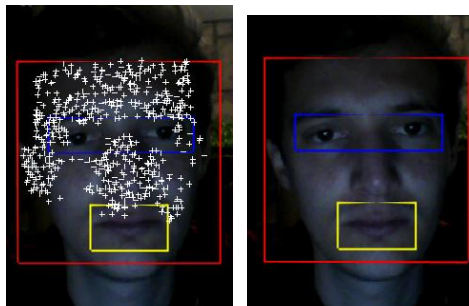


Figura 5. 6 Comparación de la nube de puntos utilizada para el tracking.

Una vez encontrada definida la nube de puntos, se procede a realizar la estimación geométrica del cambio entre una nube de puntos y la anterior mediante una función de matlab, para ello, se rellena una matriz de transformación geométrica donde los elementos de la diagonal corresponden a las traslaciones de los 3 ejes y los otros valores a las rotaciones entre ejes. Como la imagen está dispuesta en un plano y la nube de puntos también, solo se calculan las transformaciones en dos dimensiones. Acto seguido se aplican la transformación a los puntos que conforman los cuadros amarillo, azul y rojo, tomando como eje de rotación el centro del cuadro rojo que denota la detección de la cara, en los 3 casos. [5]

La funcionalidad del tracking reduce considerablemente la velocidad de procesamiento y el algoritmo está diseñado para reiniciar el proceso si no se encuentran todos los parámetros deseados, si se procesan 30 cuadros de imagen o si se pierde el 80% de los puntos en una nube de puntos.

Ejecución de movimientos

En paralelo, y si el tracking está funcionando correctamente según los parámetros, se determinan los valores que van a ejecutar los movimientos en el robot. Primeramente se calcula el ángulo de inclinación de la cabeza, como lo muestra la comparación de la figura 7. Para ello se utilizan los puntos que conforman el recuadro rojo de la cara y se calcula la inclinación respecto al eje vertical. Esta información se envía directamente a los parámetros del robot, haciendo pasar en todo momento una verificación de los máximos y mínimos físicos posibles para los servos, y ejecutando una sincronización entre ambos rangos de aplicación.

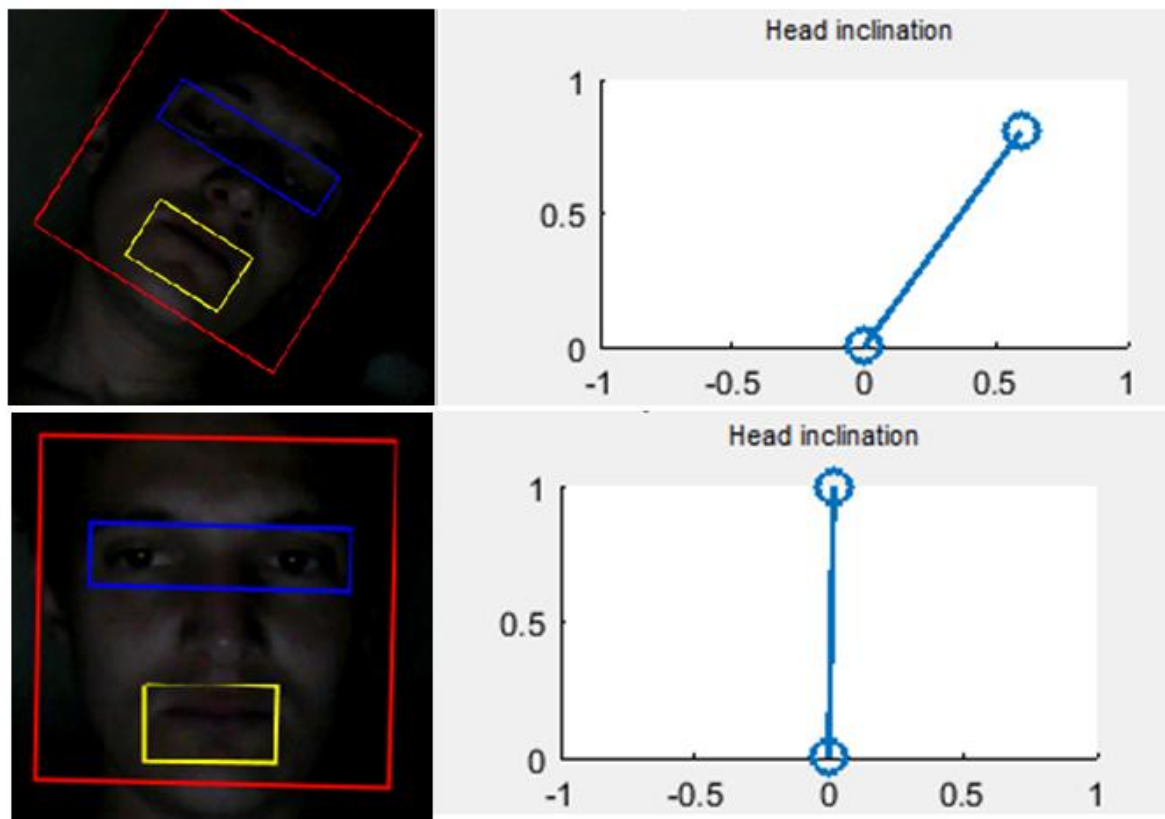


Figura 5. 7 Comparación en la determinación del ángulo de inclinación de la cabeza.

De la misma manera, se calcula la posición de la cara respecto a la totalidad de la imagen. En este proceso, si se encuentra que la cara no está dentro de los parámetros preestablecidos, se mueven los servos de posición vertical y horizontal, según convenga, para contrarrestar el desfase. Análogamente, se realiza el mismo proceso de sincronización de los valores obtenidos en la imagen y los máximos y mínimos posibles, así como un ajuste de rango para cada eje, en función de obtener los movimientos del robot lo más suaves posible.

La información de la ubicación de las pupilas dentro del recuadro de detección del par de ojos se actualiza con cada cuadro de imagen, se recorta solo la sección de los ojos, se realiza un procesamiento por umbrales en blanco y negro autoajutable y se utiliza una función de detección de zonas para encontrar círculos. Figura 8. El área de dichos círculos determina la precisión con la que se encuentra solo la pupila y se encuentra relacionada con el tamaño del recuadro de los ojos, con la intención de que si la persona se aleja físicamente del robot, entonces el área de las pupilas bien detectadas debe ser menor. Además del área, también se calcula el centroide de cada pupila. Con cada centroide, la información inicial del centro de las dos pupilas en ambos ejes y la relación entre el tamaño de la imagen cortada con el recuadro azul inicial y los centroides en cada *frame*, podemos indicar donde se encuentra cada pupila con respecto a la cara, y esta información se traduce a

movimientos en el robot utilizando los rangos máximos y mínimos y un ajuste de movimiento y percepción.

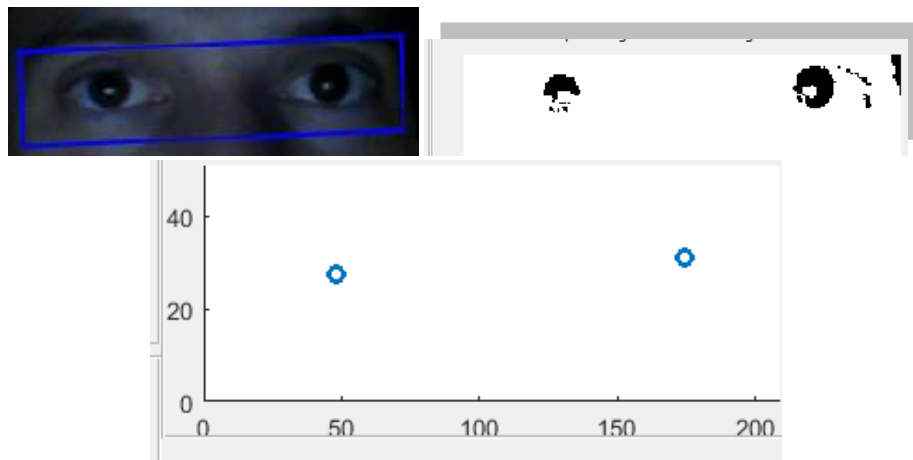


Figura 5. 8 Detección de las pupilas. Figura superior izquierda: imagen real. Figura superior derecha: imagen preprocesada. Figura inferior: cálculo de la posición de las pupilas.

Análogamente para la detección de la boca, se utiliza el detector y el tracking para seleccionar la zona de la boca dentro de cada *frame*, se extiende la detección de la imagen para asegurar que la boca abierta encaje dentro de la detección de la imagen y se procede a tratar el recorte de la imagen mediante umbrales en blanco y negro y se ejecuta una función de detección de zonas de matlab para calcular el área de la boca. Con esta área y la relación entre la detección de la boca encontrada y la detección de la cara para ajustar el cálculo cuando la persona esté cerca o lejos del robot, se puede estimar el porcentaje de apertura de la boca. Este cálculo se transmite al robot tomando en cuenta los máximos y mínimos del servo y su posición inicial cerrada.

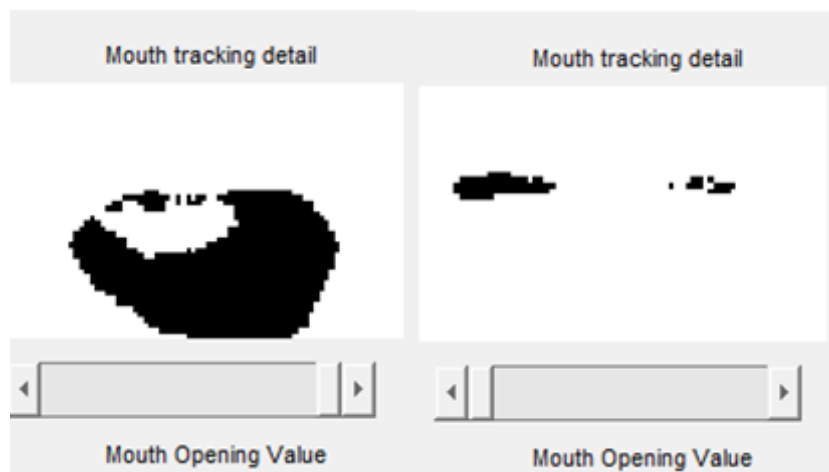


Figura 5. 9 Ejemplo de detección de apertura de la boca. Izquierda abierta, derecha cerrada.

Flujograma del algoritmo

En la figura 10 se muestra un flujograma de tareas que el algoritmo lleva a cabo una vez se inicializa el proceso mediante el botón de start. Vemos que el algoritmo se ejecuta por etapas y la detección correcta de los *features* o un bajo desempeño del sistema de tracking, son los que condicionan a las siguientes etapas y a la ejecución de movimientos.

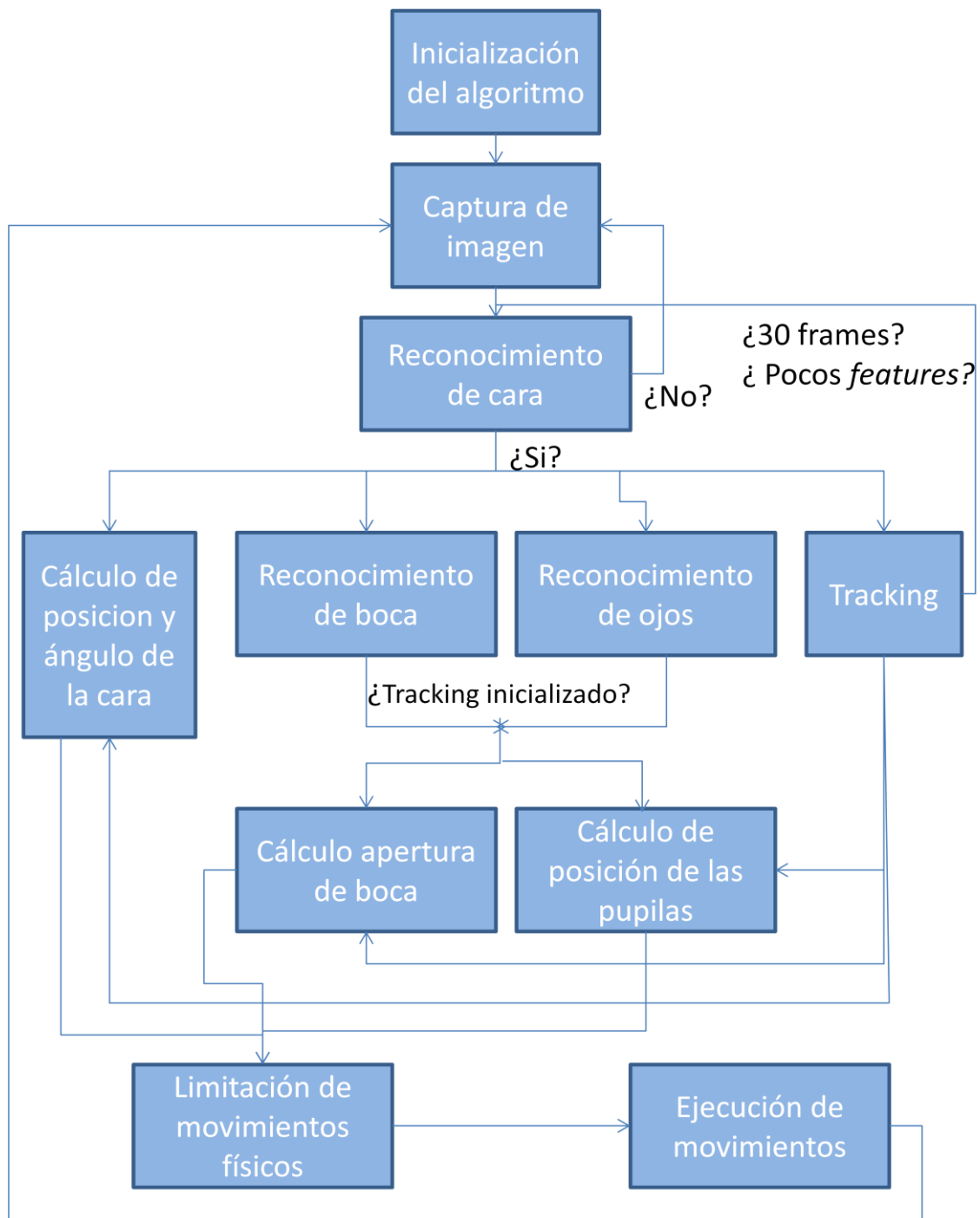


Figura 5. 10 Flujograma de tareas del algoritmo.

Hardware - Electrónica

A lo largo del algoritmo, en cuanto se obtiene el cálculo de alguna de las posiciones de los servos, se envía la señal correspondiente a ese arduino para ese servo.

Un servomotor es un dispositivo electromecánico que consiste en un motor eléctrico, un juego de engranes y una tarjeta de control, todo dentro de una carcasa de plástico. Un servo tiene la capacidad de ser controlado en posición. Es capaz de ubicarse en cualquier posición dentro de un rango de operación de 180° . Los servomotores funcionan por medio de modulación de ancho de pulso (PWM). Todos los servos disponen de tres cables, dos para alimentación Vcc y Gnd (4.8 a 6 [V]) y un tercero para aplicar el tren de pulsos de control, que hace que el circuito de control diferencial interno ponga el servo en la posición indicada. La frecuencia usada para mandar la secuencia de pulsos al servomotor es de 50 Hz, lo que significa que cada ciclo dura 20 ms. Las duraciones de cada pulso se interpretan como comandos de posicionamiento del motor, mientras que los espacios entre cada pulso son despreciados. El 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 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 11.

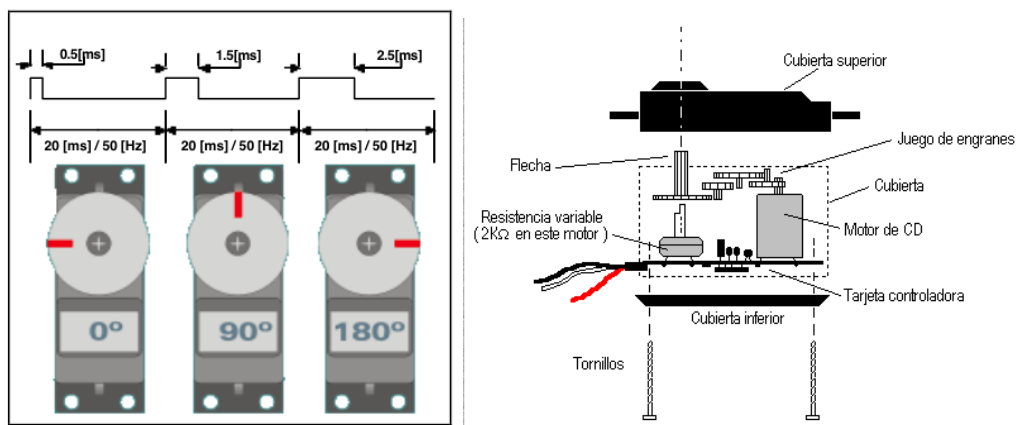


Figura 5. 11 Esquemático de control de señales para servomotores.

Los servos utilizados pueden llegar a ser rápidos, es por ello que para las articulaciones más críticas se regulan el máximo paso entre escrituras para tener un movimiento menos brusco (las dos primeras articulaciones, movimiento vertical y horizontal de la cara).

Se utiliza una fuente de voltaje de 5VDC y 4Amp para la alimentación de todos los servos en serie, por otra parte, el arduino se conecta mediante USB al PC, con lo cual su alimentación está garantizada. Figura 12.

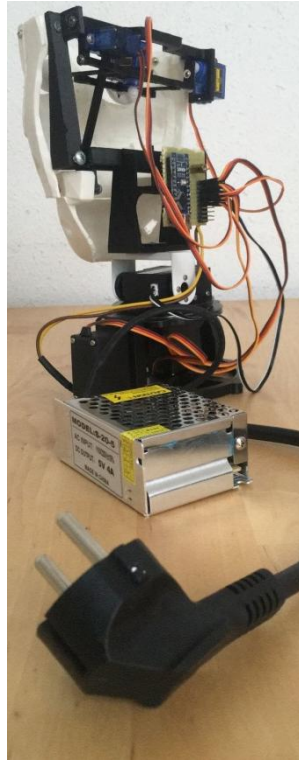


Figura 5. 12 Detalle de conexión al arduino y a la fuente de voltaje.

El hardware de control utilizado es una placa arduino Nano, basada en el ATmega328. No posee conector para alimentación externa, y funciona con un cable USB Mini-B. Se conforma de las siguientes características. Figura 13.

- Microcontrolador: Atmel ATmega328 (ATmega168 versiones anteriores)
- Tensión de Operación (nivel lógico): 5 V
- Tensión de Entrada (recomendado): 7-12 V
- Tensión de Entrada (límites): 6-20 V
- Pines E/S Digitales: 14 (de los cuales 6 proveen de salida PWM)
- Entradas Analógicas: 8 Corriente máx por cada PIN de E/S: 40 mA
- Memoria Flash: 32 KB (ATmega328) de los cuales 2KB son usados por el bootloader (16 KB – ATmega168)
- SRAM: 2 KB (ATmega328) (1 KB ATmega168)
- EEPROM: 1 KB (ATmega328) (512 bytes – ATmega168)
- Frecuencia de reloj: 16 MHz
- Dimensiones: 18,5mm x 43,2mm

El Arduino Nano posee selección automática de la fuente de alimentación y puede ser alimentado a través de:

- Una conexión Mini-B USB.
- Una fuente de alimentación no regulada de 6-20V (pin 30).
- Una fuente de alimentación regulada de 5V (pin 27)



Figura 5. 13 Arduino Nano.

El esquema del conexionado de los elementos eléctricos es sencillo y se muestra en el siguiente diagrama.

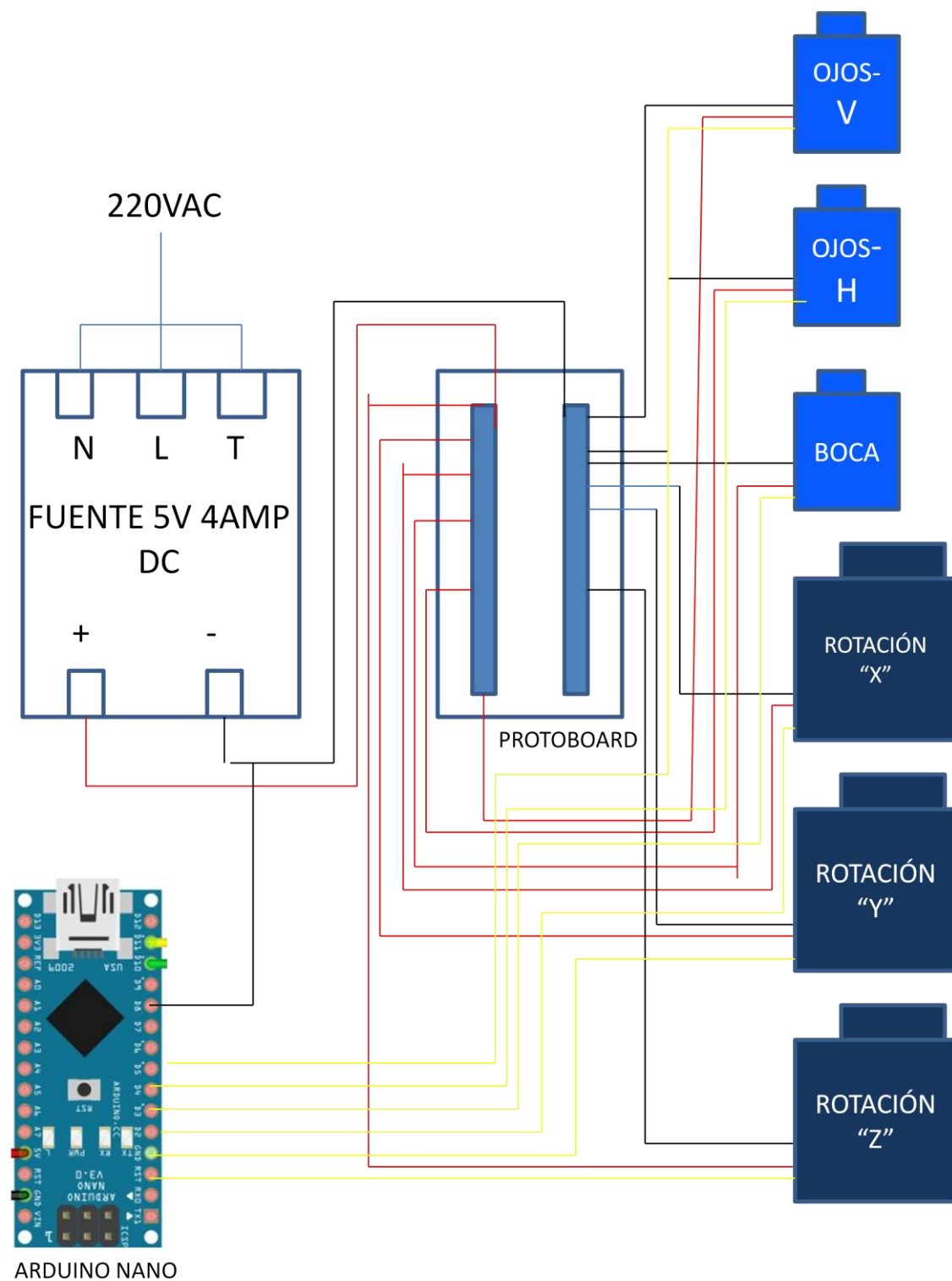


Figura 5. 14 Esquema de conexión eléctrico.

Hardware - Robot

Para lograr los movimientos reales de la cara, se diseñó un sistema mecánico de articulaciones intentando acercar los ejes de movimiento lo máximo posible,

con esto, logramos un movimiento muy natural. Esta es la ventaja del uso de materiales plásticos para la construcción de una cara humanoide, generalmente están contruidos con mucha robustez que puede llegar a ser innecesaria y causar que los movimientos sean muy lentos y abruptos. En la figura 15 se muestra una imagen general del robot ya construido.

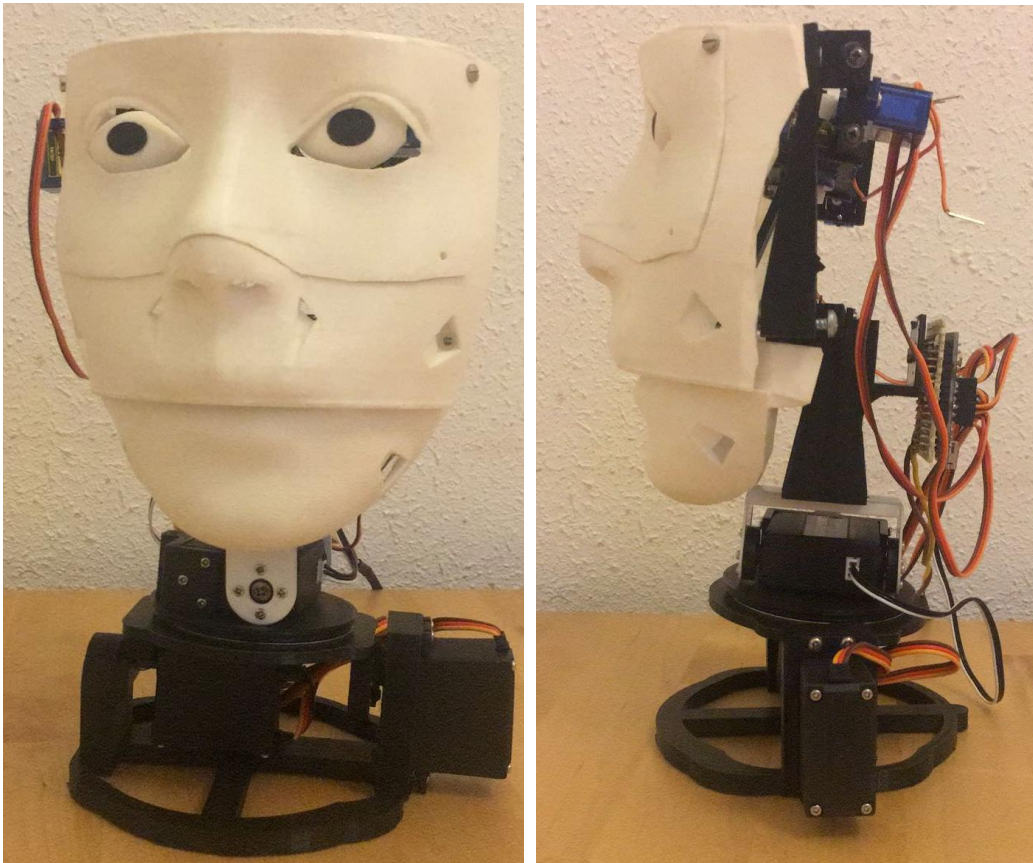


Figura 5. 15 Fotografía del robot construido.

Como se puede apreciar, el robot se posa sobre una base y el primer servo a la derecha, sostiene con dos puntos de anclaje toda la estructura y ejecuta el movimiento vertical de la misma. Sostenido a ella mediante ajuste a presión, se encuentra el segundo servo, que ejecuta el movimiento horizontal de la cara con el uso de un disco con sujeción para el siguiente servo y que está apoyado sobre un rodamiento plano de agujas para garantizar que el roce no sea un problema torsional para el servo. En la figura 16 y 17 se muestra un detalle.

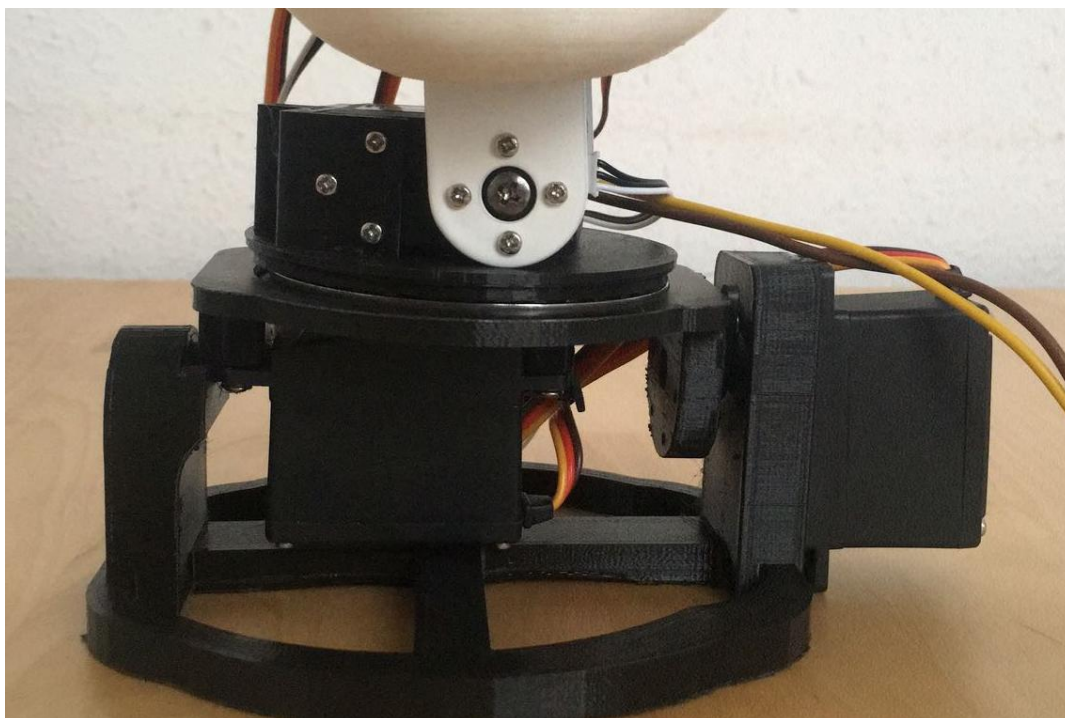


Figura 5. 16 Vista frontal del montaje de los servos de movimiento horizontal, vertical y angular

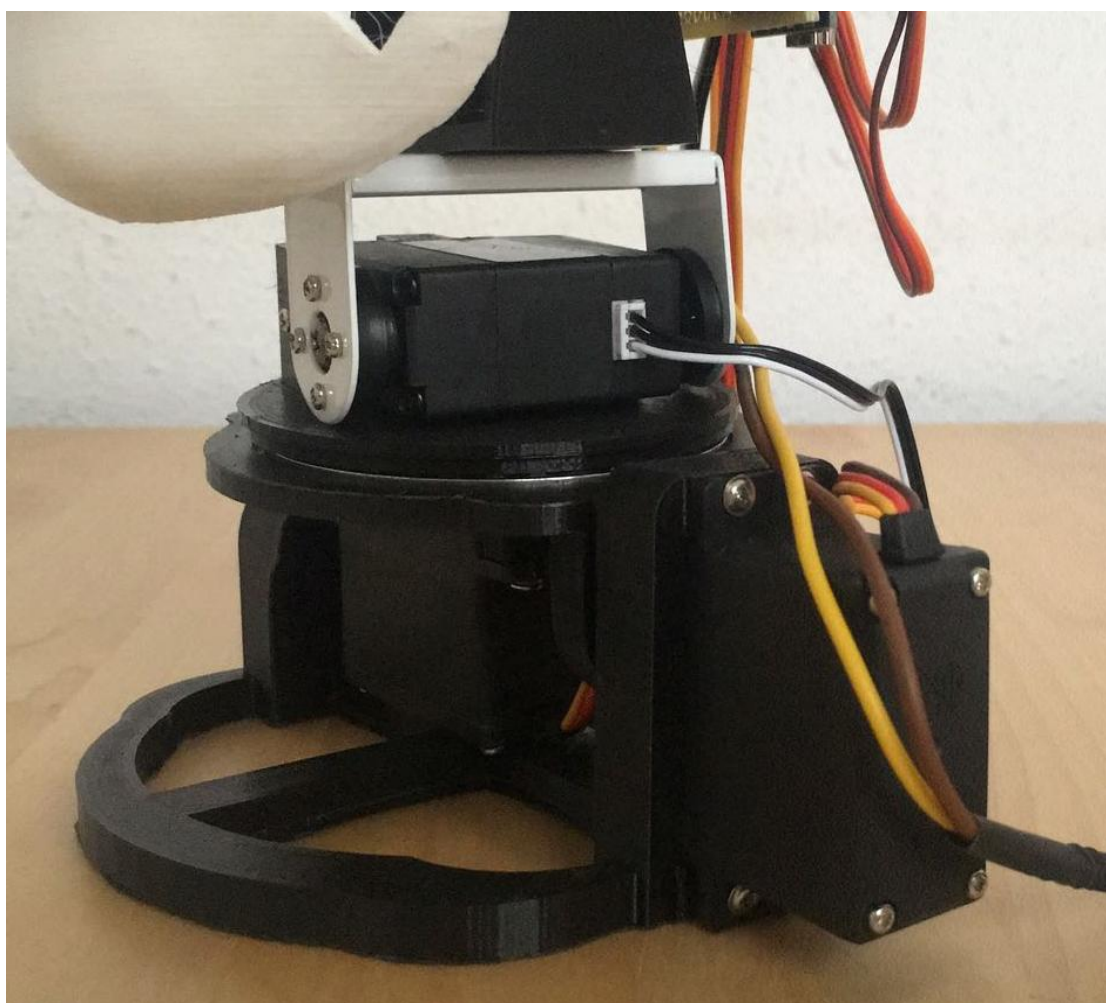


Figura 5. 17 Vista lateral de la montura de los servos para el movimiento vertical, horizontal y angular.

Justo arriba del disco se encuentra sujeto el servo donde se apoya la estructura que sujeta a toda la cara y otorga al robot el movimiento angular de la cabeza, incluyendo la cámara. Todos estos servos son del tipo MG995. En la figura 18 se muestra una representación del diseño de la base articulada, y en la figura 19 se muestra una imagen del mecanismo de apertura de la boca y el movimiento de los ojos.

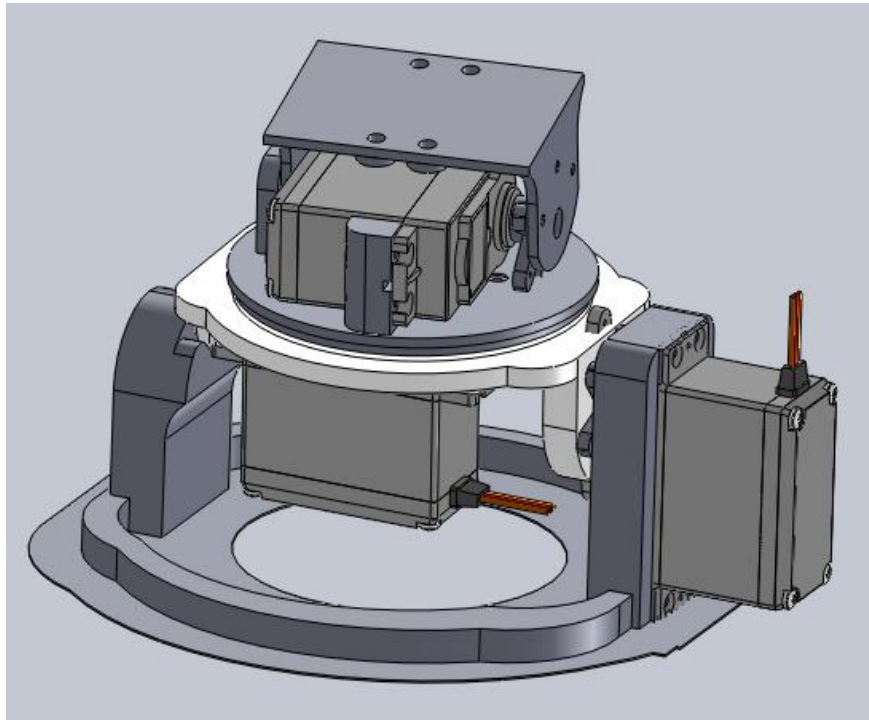


Figura 5. 18 Base articulada del robot.

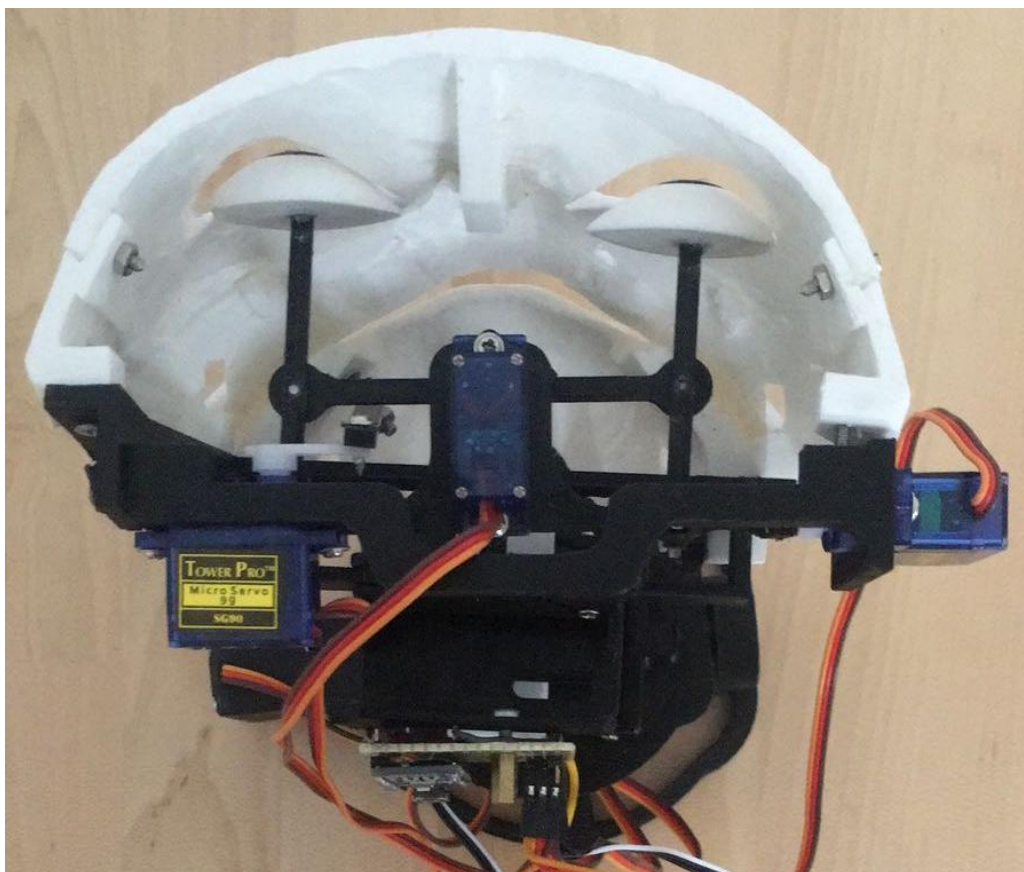


Figura 5. 19 Mecanismo de apertura de la boca y movimiento de los ojos en el plano.

La cara es un diseño de una librería libre para el desarrollo estudiantil creado por un artista francés. Su robot se llama inMoov y es un robot humanoide completo. En nuestro caso, solo utilizamos el diseño de la cara en representación de la piel del robot. De esta manera conseguimos simular un robot humanoide del tipo caricatura, alejándonos de una representación más real, sin cabello ni piel sintética, ya que podemos llegar a generar un rechazo visual por parte de un espectador escéptico y además se aligeran costos en materiales.

Se diseñó una estructura general que sostiene las piezas de la cara y la estructura mecánica de los ojos y de la boca.

Los ojos se mueven en dirección horizontal y vertical mediante dos servos 9G muy económicos, fiables y fáciles de encontrar en cualquier mercado. El servo de movimiento vertical mueve la estructura donde está sujeto el servo de movimiento horizontal, quien ejecuta el movimiento mediante un sistema de varillas sujetas a los ojos.

Por otra parte, la boca solo cuenta con una varilla sujeta a la articulación de la boca del inMoov y sujeto a otro servo 9G, además de tener una pequeña articulación redonda para obtener un movimiento sin rozaduras ni torsiones. En la figura 20, se muestra la imagen trasera de las conexiones con el arduino y el mecanismo de apertura de la boca.

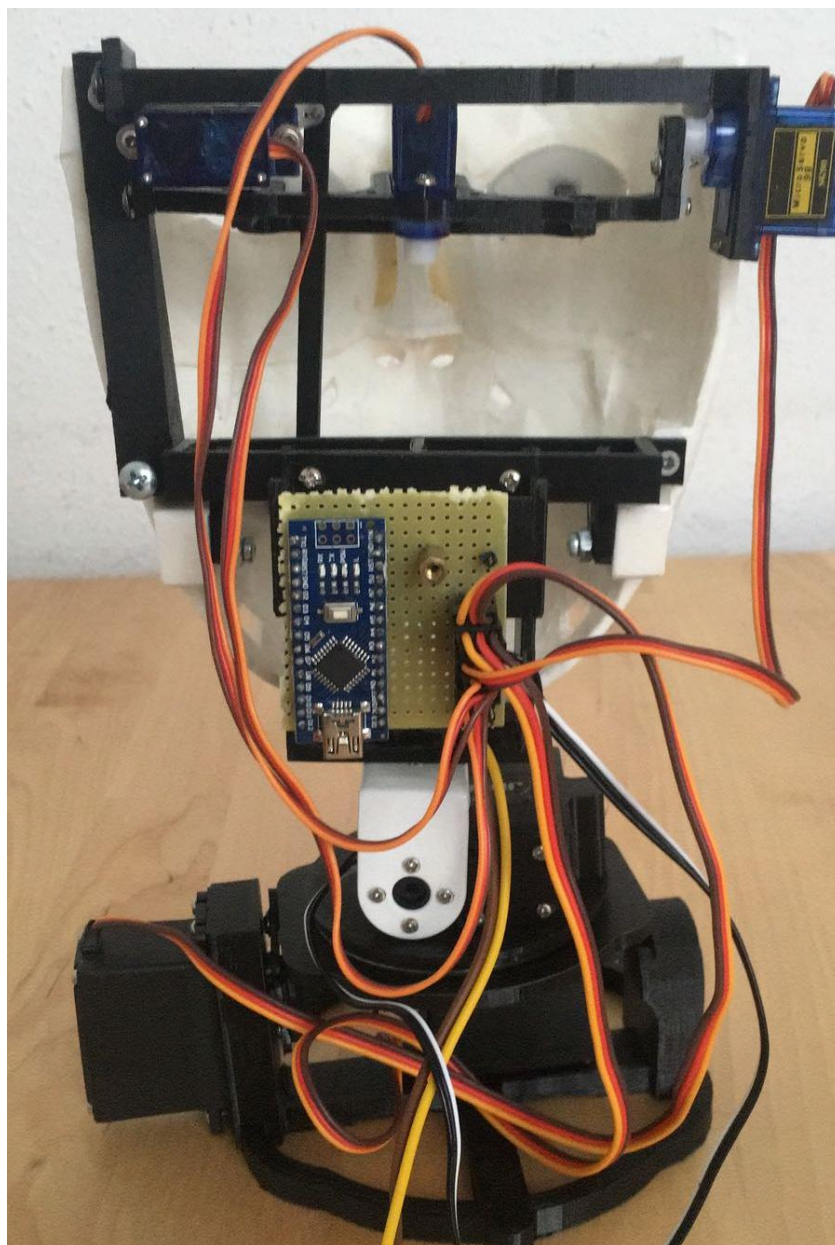


Figura 5. 20 Vista trasera.

6. RESULTADOS Y DISCUSIÓN

En el trabajo se demostró la posibilidad de diseñar y construir un robot de bajo costo para probar algoritmos de visión artificial, se comprobaron los siguientes puntos:

1. Los servomotores de bajo costo son aptos para este tipo de aplicaciones, tienen la robustez necesaria y son suficientemente precisos como para realizar movimientos a velocidad media sin afectar la naturalidad del movimiento.
2. Los métodos de detección y los métodos de tracking utilizados en el algoritmo son eficaces para los fines deseados, son precisos y se adaptan a las capacidades de las cámaras utilizadas.
3. Los mecanismos utilizados para lograr los movimientos son efectivos, fiables, no causan pérdidas por fuerzas de roce y están dimensionados correctamente para soportar las torsiones generadas por el peso propio del robot y la cámara.
4. El diseño del algoritmo cumple con la velocidad de procesamiento esperada, sin embargo, pudieran ser utilizadas otras librerías de otros lenguajes de programación para lograr mayor rapidez.
5. Los métodos de ahorro de tiempo de procesamiento utilizados, en términos de procesamiento y muestreo de imagen fueron efectivos.
6. En específico, el método de tracking utilizado basado en método RANSAC así como los clasificadores utilizados, generan un comportamiento correcto en el proceso de detección para las necesidades del sistema.

La intención de este proyecto es crear una base fiable y económica para las pruebas de algoritmos de distintos tipos que involucren a robots humanoides o robot colaborativos con o sin visión artificial. El alcance de este proyecto incluye la detección de caras, pupilas, boca, seguimiento e imitación. En un futuro, pueden desarrollarse algoritmos del tipo:

1. Detección de gestos, emociones, etc.
2. Pruebas de interacción mediante voz, desarrollo de un sistema de percepción visual y respuesta con algoritmos basados en inteligencia artificial.
3. Algoritmos basados en redes neuronales para aprendizaje de movimientos o premeditación y estimación de los mismos.
4. Algoritmo de reconocimiento facial y emparejamiento de características con una base de datos.
5. Reconocimiento de caras en la industria y algoritmos de seguridad y monitoreo de personal.
6. se podría desarrollar algoritmos parecidos utilizando otras librerías con otros lenguajes de programación para ganar mayor velocidad de procesamiento y versatilidad, además se pueden utilizar librerías de arduino para configurar las salidas análogas de cada servomotor, y adecuar el movimiento de los mismos dependiendo de las fuerzas generadas en su eje. De esta manera se puede conseguir un movimiento aún más natural.
7. Se pueden utilizar filtros de señales como el filtro de kalman para mejorar la precisión del algoritmo, aunque el tracking utilizado es suficientemente bueno, se pueden generar errores en la detección, sin embargo, si se filtran los resultados arrojados por los detectores, se puede mejorar el funcionamiento del algoritmo de manera preventiva.
8. Análogamente al punto anterior, se pueden utilizar filtros para la salida de los servomotores, haciendo control de la posición y la velocidad simultáneamente, de esta manera podremos configurar una velocidad máxima y una aceleración que permita eliminar los cambios bruscos en la posición, que se traducen en vibraciones y movimientos no naturales.

8. BIBLIOGRAFIA

- [1] Paul Viola and Michael Jones. Robust real-time face detection. International Journal of Computer Vision, 57:137-154, 2004.
- [2] Constantine Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In International Conference on Computer Vision (ICCV), páginas 555-562, 1998.
- [3] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Accepted conference on computer vision and pattern recognition 2001.
- [4] O. Chum and J. Matas, Randomized RANSAC with Td,d test, 13th British Machine Vision Conference, Septiembre 2002.
- [5] Hartley, R., and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2003.

9. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

La siguiente tabla contiene los elementos que constituyen el robot. Los elementos fabricados con impresora 3D son calculados de acuerdo al peso del plástico utilizado y el precio del consumible regular, con configuraciones de impresión media-alta.

Cantidad	Elemento	Precio
1	Base robot (STL)	2€
1	Mecanismo movimiento vertical (STL)	1.5€
1	Base de rotación y movimiento horizontal	1€
1	Rodamiento de bolas axial	13€
1	Estructura soporte de cara	2€
1	Mecanismos de ojos y boca	0.5€
1	Estructura cara (tres piezas)	4€
3	Servomotor 9G	2.99€
2	Servomotor S003	13€
1	Servomotor S003 anclajes angulares	18€
1	Arduino uno	2.5€
1	Fuente de poder	4.5€
1	Cable miniUSB	3€
1	Tornillería	4€
	TOTAL	91.97€

Figura 9. 1 Tabla de componentes.

Se realiza un diagrama de Grantt realizando las estimaciones iniciales del proyecto con los ajustes necesarios de acuerdo a su resolución. Se toman en cuenta 3 horas de trabajo diarias para dicha estimación, con lo cual, en horas-hombre se tiene un resultado de 288hr. Se adjunta la tabla de tareas y el diagrama.

Nombre de la tarea	Fecha de Inicio	Fecha final	Duración	Predecesores
Desarrollo Algoritmo	01/10/17	20/12/17	59d	
Pruebas detección	01/10/17	20/10/17	16d	
Pruebas de tracking	23/10/17	25/10/17	3d	2
Pruebas de captura	26/10/17	31/10/17	4d	3
Pruebas de funcionamiento	01/11/17	13/11/17	9d	4
Construcción de la GUI	14/11/17	14/11/17	1d	5

Programación de la GUI	15/11/17	28/11/17	10d	6
Programación de arduino	29/11/17	05/12/17	5d	7
Programacion de parametros de servos	06/12/17	06/12/17	1d	8
Programación y ajustes de movimiento	07/12/17	20/12/17	10d	9
Desarrollo Robot	21/12/17	01/02/18	31d	
Desarrollo de mecanismos	21/12/17	17/01/18	20d	10
Pruebas de ajuste	18/01/18	19/01/18	2d	12
Selección de materiales	22/01/18	22/01/18	1d	13
Acopio de piezas	23/01/18	24/01/18	2d	14
Armado del robot	25/01/18	26/01/18	2d	15
Conexiones y placa de arduino	29/01/18	29/01/18	1d	16
Pruebas de funcionamiento	30/01/18	01/02/18	3d	17
Elaboración del trabajo	16/01/18	02/02/18	14d	
Reunión con tutor	02/02/18	02/02/18	1d	18
Elaboración y acopio de documentación generada en el trabajo	16/01/18	22/01/18	5d	

Figura 9. 2 Listado de tareas.

En la siguiente página se anexa la figura 9.3, diagrama de Grantt del proyecto.

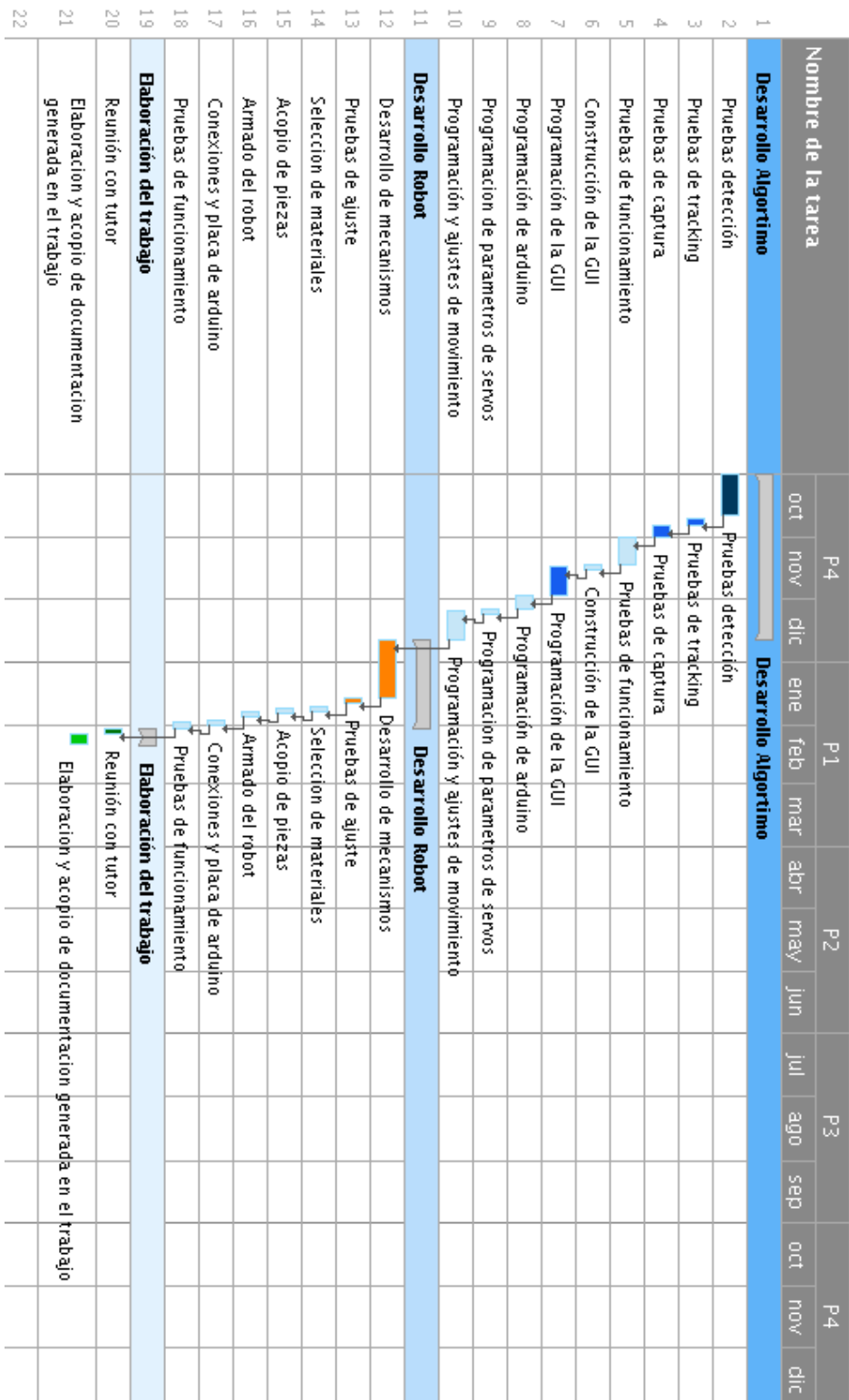


Figura 9. 3 Diagrama de Grantt del proyecto.

10. GLOSARIO

Frame; una imagen que compone un video.

Encoder: Sistema para el posicionamiento angular de ejes rotatorios.

inMoov: Robot open source.

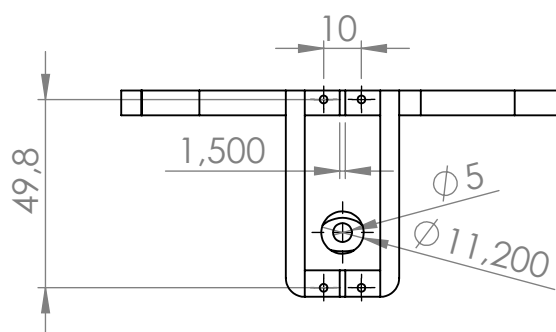
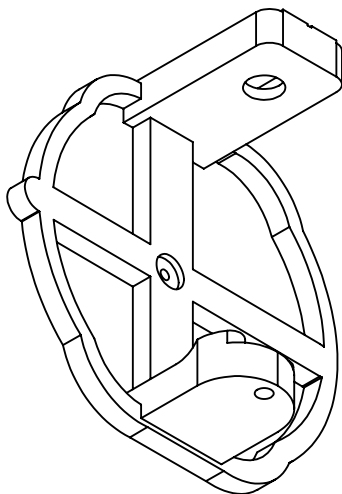
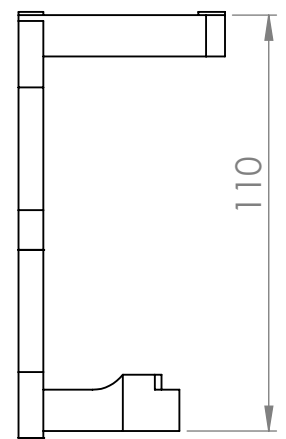
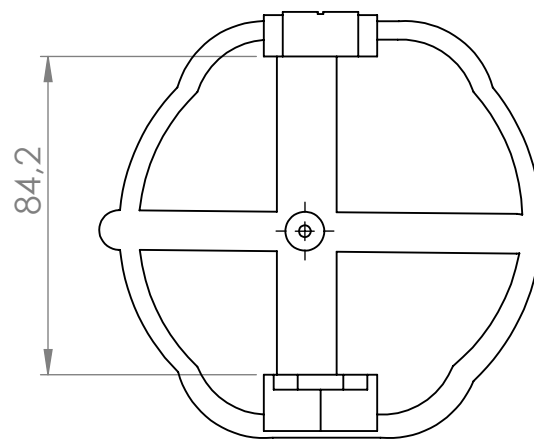
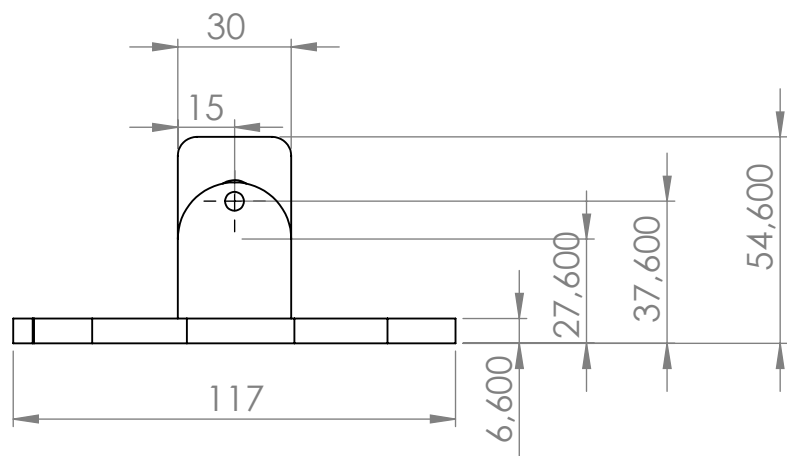
Arduino: Plataforma de prototipos electrónica de código abierto basada en hardware y software flexibles.

11. PLANOS

En las siguientes páginas se anexan los planos de las piezas que conforman el robot según la siguiente tabla.

Número de plano.	Nombre de la pieza.
18090801	Base principal
18090802	Placa soporte 1.
18090803	Soporte cuello.
18090804	Base estructura cabeza.
18090805	Base cabeza superior.
18090806	Base estructura ojos.
18090807	Eje rotación ojos.
18090808	Ojo.
18090809	Manivela boca.

Figura 11. 1 Tabla de planos.



ROBOT
BasePrincipal
IMPRESIÓN 3D

JmSequeira	FIRMA 26/03/18	CANTIDAD	COLOR
TOLERANCIA	ESCALA	1	NEGRO
		MATERIAL	ACABADO
+/-0.1mm	SCALE:1:2	PLA	Ninguno

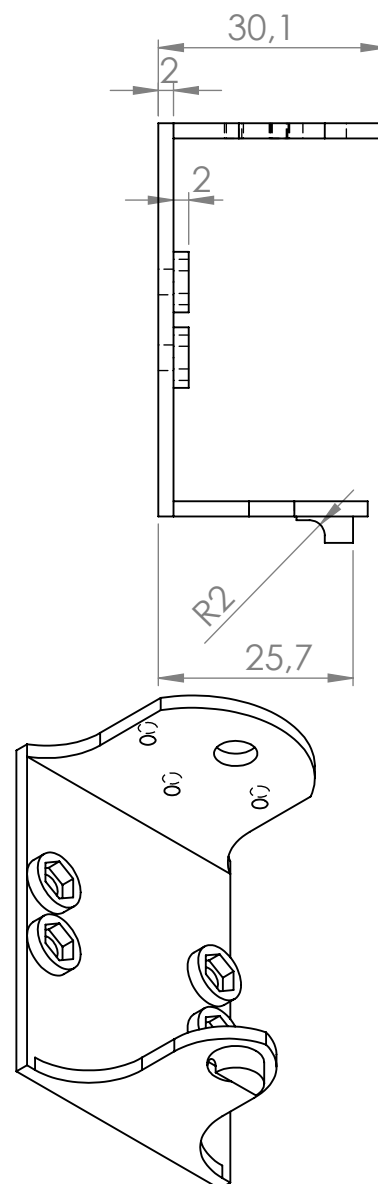
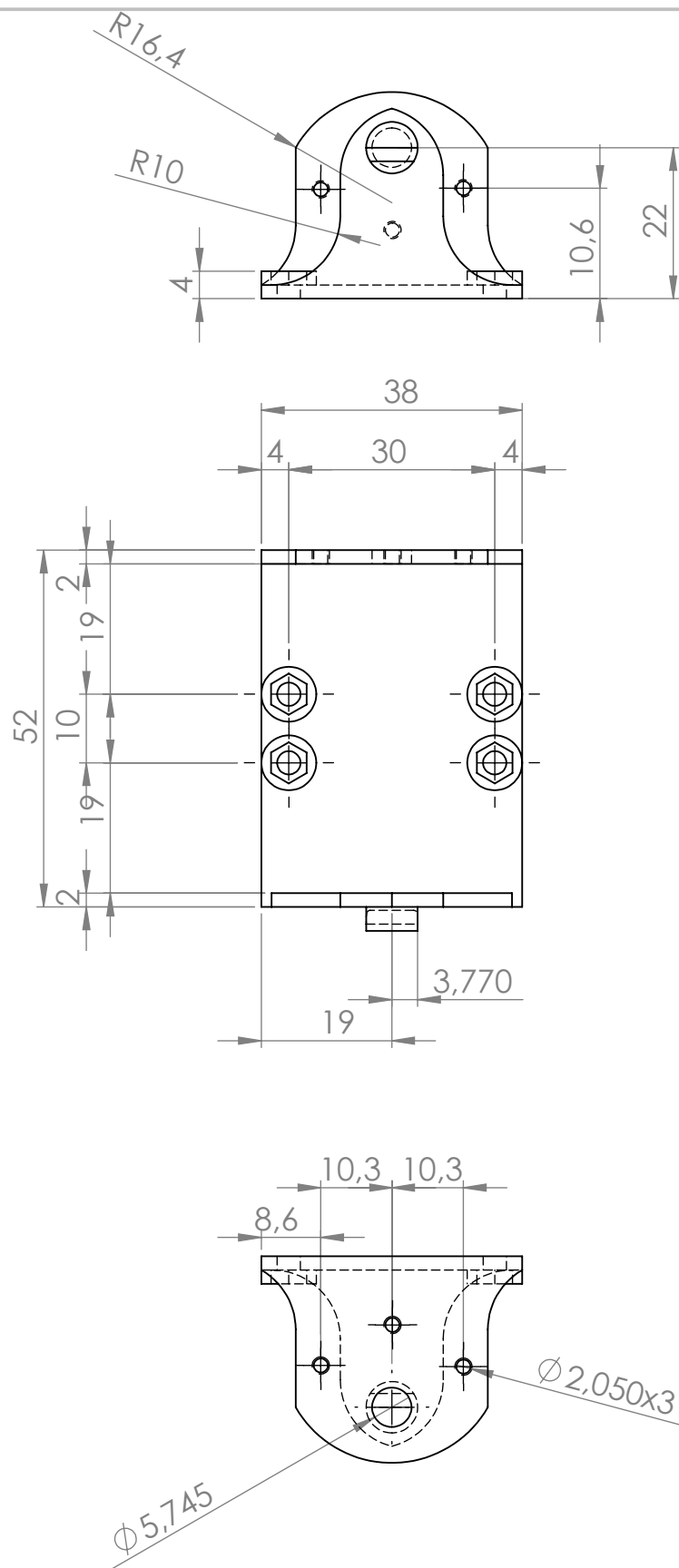


POLITÉCNICA

N

18090801





ROBOT
BaseEstructuraCabeza
IMPRESIÓN 3D

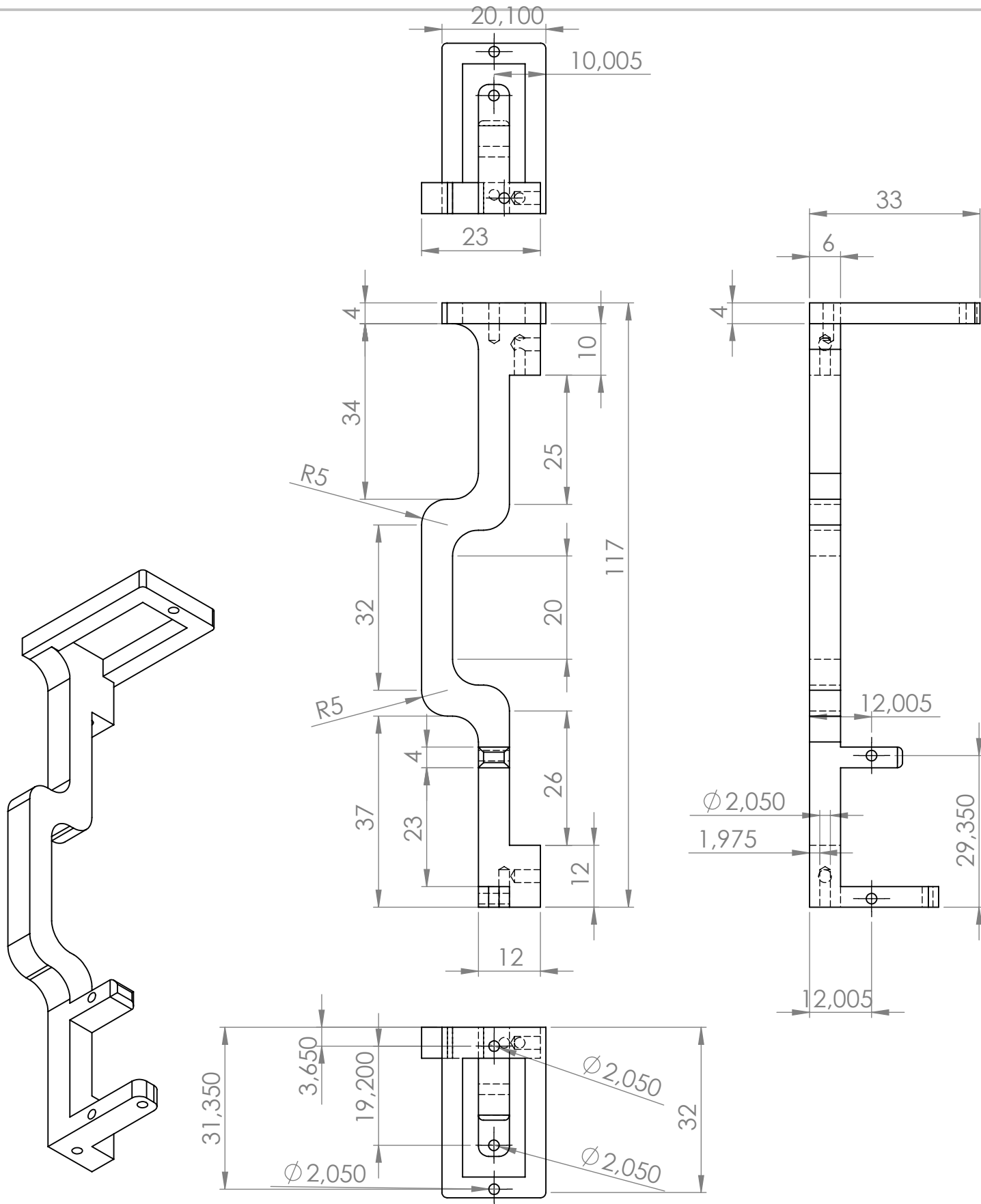
JmSequeira	FIRMA 26/03/18	CANTIDAD	COLOR
TOLERANCIA	ESCALA	1	NEGRO
		MATERIAL	ACABADO
+/-0.1mm	SCALE:1:1	PLA	Ninguno



POLITÉCNICA

N

18090804



ROBOT
BASE ESTRUCTURA OJOS
IMPRESIÓN 3D

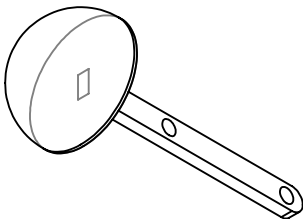
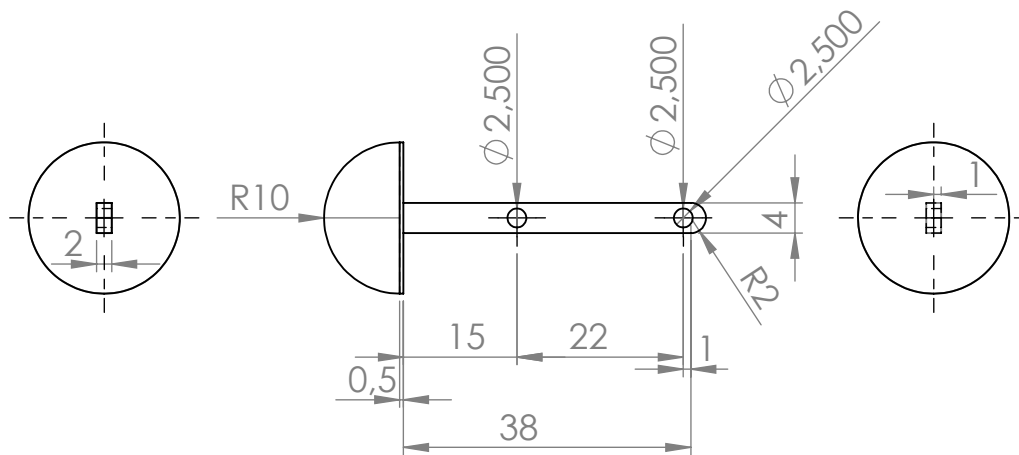
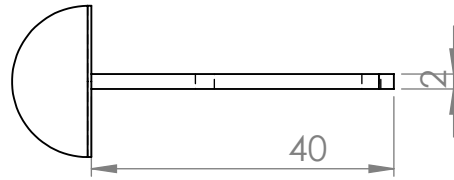
JmSequeira	FIRMA 26/03/18	CANTIDAD	COLOR
TOLERANCIA	ESCALA	1	NEGRO
		MATERIAL	ACABADO
+/-0.1mm	SCALE:1:1	PLA	Ninguno



POLITÉCNICA

N

18090806



ROBOT
OJO
IMPRESIÓN 3D

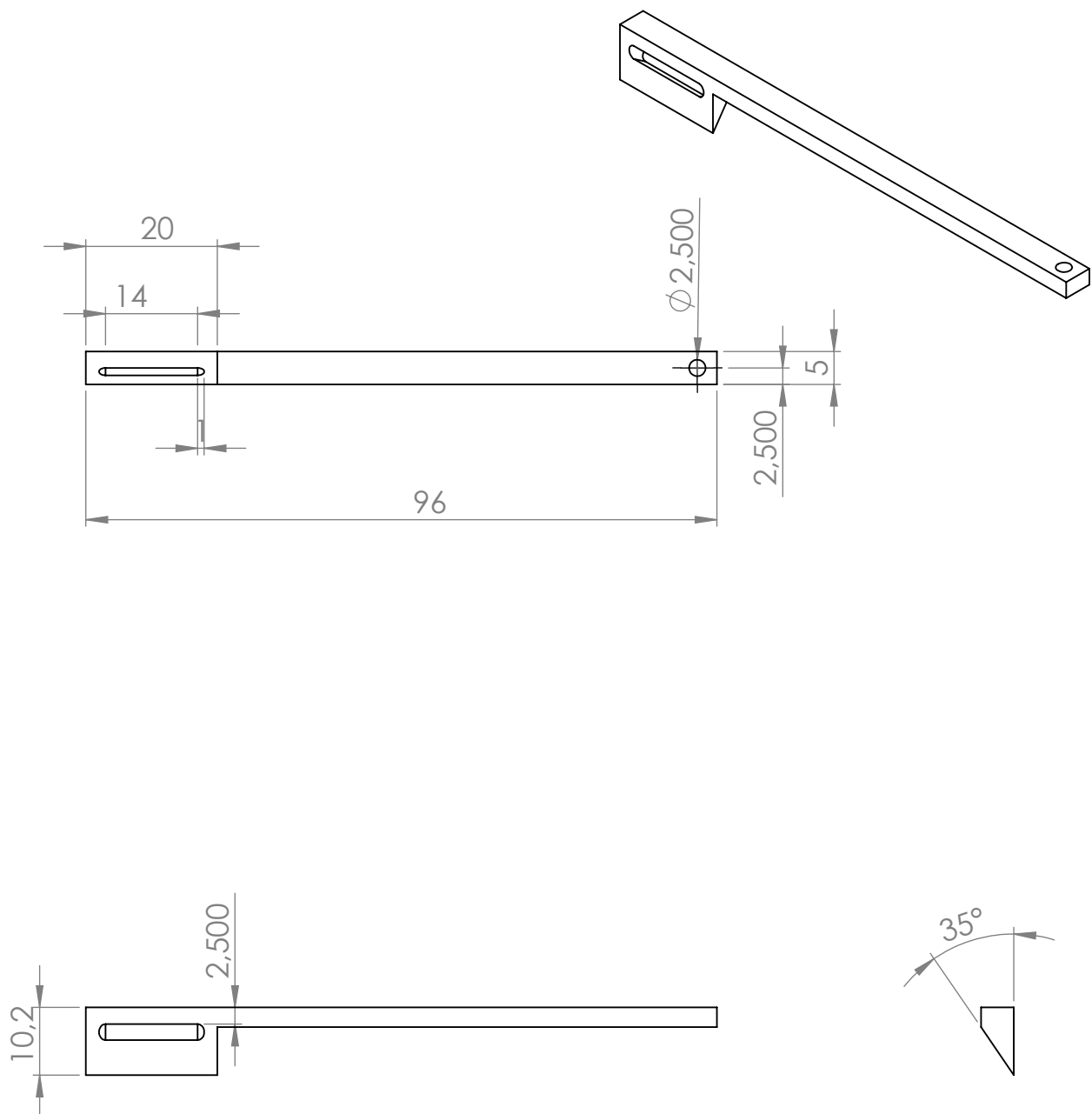
JmSequeira	FIRMA 26/03/18	CANTIDAD	COLOR
TOLERANCIA	ESCALA	1	NEGRO
		MATERIAL	ACABADO
+/-0.1mm	SCALE:1:1	PLA	Ninguno



POLITÉCNICA

N

18090808



ROBOT
MANIVELA BOCA
IMPRESIÓN 3D

JmSequeira	FIRMA 26/03/18	CANTIDAD	COLOR
TOLERANCIA	ESCALA	1	NEGRO
+/-0.1mm	SCALE:1:1	MATERIAL	ACABADO
		PLA	Ninguno



POLITÉCNICA

N

18090809