



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño del control automatizado para el seguimiento de  
personas por reconocimiento facial en tiempo real.

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Ciudad Real Villaescusa, Alejandro

Tutor/a: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
Escuela Técnica Superior de Ingeniería del Diseño

**DISEÑO DEL CONTROL AUTOMATIZADO PARA EL  
SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO  
FACIAL EN TIEMPO REAL**

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022



## RESUMEN

Con la realización del Trabajo de Fin de Grado (TFG) se pretende obtener un sistema de reproducción de video dinámico mediante un control automatizado basado en la programación en Python con Opencv; el control de actuadores basándonos en la programación y transmisión de datos por el canal serie en Arduino; y el diseño de una interfaz gráfica para posibilitar el control manual de la cámara. Por otro lado, la realización del proyecto incluye el diseño 3D de la estructura mecánica que habilitará el movimiento tanto de translación horizontal como vertical, así como una rotación sobre el eje vertical. En cuanto a los actuadores, se realizará el control sobre dos motores paso a paso y sobre un servomotor de 180°. Una de las aplicaciones tras un dimensionamiento a gran escala de este proyecto, puede ser la realización de videoconferencias en las que exista un movimiento por un escenario de la persona que figure en él.

## ABSTRACT

With the production of the Final Degree Project (FDP) it is intended to obtain a dynamic video playback system through an automated control based on Python programming with Opencv; the control of actuators based on the programming and transmission of data through the serial channel in Arduino; and the design of a graphical interface to enable manual control of the camera. On the other hand, the realization of the project includes the 3D design of the mechanical structure that will enable the movement of both horizontal and vertical translation as well as a rotation on the vertical axis. As for the actuators, the control will be carried out on two stepper motors and on a 180° servomotor. One of the applications after a large-scale dimensioning of this project may be the realization of videoconferences in which there is a movement through a scenario of the person who appears in it.

## RESUM

Amb la realització del Treball de Fi de Grau (TFG) es pretén obtenir un sistema de reproducció de vídeo dinàmic mitjançant un control automatitzat basat en la programació a Python amb Opencv; el control d'actuadors basant-nos en la programació i transmissió de dades pel canal sèrie a Arduino; i el disseny d'una interfície gràfica per possibilitar el control manual de la càmera. D'altra banda, la realització del projecte inclou el disseny 3D de l'estructura mecànica que habilitarà el moviment tant de translació horitzontal com vertical així com una rotació sobre l'eix vertical. Quant als actuadors, es realitzarà el control sobre dos motors pas a pas i sobre un servomotor de 180°. Una de les aplicacions després d'un dimensionament a gran escala d'aquest projecte, pot ser la realització de videoconferències en què hi hagi un moviment per un escenari de la persona que hi figuri.



## ÍNDICE DE DOCUMENTOS

Documento N°1: MEMORIA .....	4
Documento N°2: PLANOS .....	86
Documento N°3: PLIEGO DE CONDICIONES .....	112
Documento N°4: PRESUPUESTO .....	119



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °1: MEMORIA**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## ÍNDICE

1.0. Introducción al proyecto .....	7
1.1. Objeto.....	8
1.2. Antecedentes .....	9
1.3. Solución Adoptada .....	10
1.3.1. Diseño 3D de la estructura .....	11
1.3.2. Materiales.....	12
1.3.3. Software: Subsistema mecánico .....	14
1.3.3.1. Diseño 3D .....	15
1.3.3.2. Impresión de piezas.....	16
1.3.3.3. Corte láser .....	16
1.4. Componentes .....	17
1.4.1. Servo.....	17
1.4.2. Stepper .....	18
1.4.3. Drivers .....	19
1.4.4. Arduino .....	21
1.4.5. Módulo BT .....	23
1.4.6. Fuente de alimentación.....	23
1.4.7. Finales de carrera .....	24
1.5. Programación .....	25
1.5.1. Python .....	25
1.5.1.1. PyCharm .....	25
1.5.1.2. OpenCV.....	26
1.5.1.3. Librería Serial.....	26
1.5.1.4. Librería MediaPipe .....	27
1.5.1.4.1. Reconocimiento facial .....	27
1.5.1.4.2. Malla facial .....	28
1.5.2. Arduino .....	30
1.5.3. App Inventor.....	31
1.6. Estudio económico .....	33
1.7. Conclusiones y líneas abiertas.....	33
1.7.1. Líneas abiertas.....	33
1.7.1.1. Materiales.....	33
1.7.1.2. Diseño.....	34
1.7.1.3. Script y funcionalidad .....	34
1.7.1.4. Coste.....	34
1.7.2. Conclusiones.....	35
1.8. Bibliografía.....	36
Anejo N. º1: CÓDIGO .....	37
Anejo N. º2: DATASHEETS.....	73
Anejo N. º3: PLAN DE MANTENIMIENTO.....	82
Anejo N. º4: NORMATIVA.....	84

## ÍNDICE DE IMÁGENES

Imagen 1. Estructura final montada...	8
Imagen 2. Ejemplo de situación óptima a la cual adaptar el proyecto.....	9
Imagen 3. Solución propuesta para la situación de la imagen 2 .....	10
Imagen 4. Distintas fases del diseño de la estructura .....	11
Imagen 5. Conjunto de piezas resultantes de la adaptación del componente vertical de la estructura .....	13
Imagen 6. Pieza base previa a la adaptación.....	14
Imagen 7. Entorno Autodesk Inventor .....	15
Imagen 8. Control PWM de un servo.....	17
Imagen 9. Servo SG-90.....	18
Imagen 10. Bobinado de un motor paso a paso.....	18
Imagen 11. Circuito formado por Nema17 y su driver A.4988.....	19
Imagen 12. Tabla de configuración de los pines MSx.....	20
Imagen 13. Tabla para el cálculo de la tensión de referencia en funcion de la configuración de pasos.....	20
Imagen 14. . Amplitud de pulso de control .....	21
Imagen 15. Planta del Arduino Mega 2560 .....	22
Imagen 16. Conexionado del módulo bluetooth HC-06 .....	23
Imagen 17. Fuente de alimentación externa.....	24
Imagen 18. Esquema electrónico final de carrera .....	24
Imagen 19. Entorno de trabajo en PyCharm .....	25
Imagen 20. Interpretación de datos de coordenadas extraído por el algoritmo de reconocimiento facial ...	27
Imagen 21. Representación de los datos que extrae el algoritmo .....	28
Imagen 22. Modelo canónico del algoritmo de malla facial.....	29
Imagen 23. Representación gráfica de los datos que extrae el algoritmo .....	30
Imagen 24. Entorno de trabajo Arduino .....	30
Imagen 25. Entorno de trabajo MitAppInventor .....	31
Imagen 26. Entorno de programación MitAppInventor .....	32
Imagen 27. Interfaz gráfica de la aplicación móvil .....	70
Imagen 28. Bloque de código de conexión.....	71
Imagen 29. Bloque de código de control.....	71
Imagen 30. Creación de la aplicación .....	72
Imagen 31. Aplicación creada en el dispositivo móvil.....	72

## 1.0. Introducción al proyecto

La idea fundamental de este Trabajo de Fin de Grado surgió tras razonar aquello por lo que tenía y debía caracterizarse un ingeniero. La solución a esta pregunta fue esta: “Persona especializada en un cierto campo tecnológico con capacidad para adaptarse, analizar y aprender de forma autodidacta a cualquier otro ámbito”.

Por tanto, este proyecto lo que pretende mostrar son 2 cosas. La principal, una parte de los conocimientos adquiridos durante la realización del grado en Ingeniería Electrónica Industrial y Automática y, por otro lado, la capacidad de adaptación a campos o plataformas no enseñados durante esta. Es por ello que surge la idea de un proyecto que engloba disciplinas impartidas en las asignaturas del grado como la electrónica, la automática, la programación (Arduino) y así mismo incluye los campos del diseño 3D, programación en un lenguaje distinto (Python) y visión artificial en tiempo real.

Por otro lado, cabe destacar que la solución del proyecto no es única puesto que se pretende que se pueda ajustar para cualquier situación o donde se requiera su implementación, es decir, la solución adoptada es un prototipo cuyo funcionamiento se ha adecuado a las características de mi lugar de trabajo, con componentes de bajo presupuesto así como una reducida gama de materiales pero, con posibilidad a distintos calibres en cuanto a presupuesto, material, tamaño o forma tras la realización de ajustes en el diseño de la estructura y a ciertos parámetros de la programación.

Además, hay que mencionar que todo el proyecto ha sido realizado por mi persona sin disponer del soporte de una empresa en cuanto a conocimientos y apartado económico, desde el planteamiento inicial hasta la solución en cada uno de los subsistemas (diseño 3D de la estructura, material, componentes y script programado).

Se ha de mencionar también la ayuda recibida en colaboración con la ETSID puesto que se me ha suministrado una parte del hardware requerido y a la hora de la creación física de la estructura por parte de los técnicos de laboratorio de impresión gracias a la maquinaria que disponen.



## 1.1. Objeto

El objetivo de este Trabajo de Fin de Grado es realizar el diseño, programación y montaje de un dispositivo capaz de monitorizar, mediante visión artificial en tiempo real, la posición y orientación del rostro de un individuo y en función de ellas realizar el control automático sobre los actuadores para ubicar el rostro de la persona en el centro de la imagen.

Se dispondrá de 2 tipos de sensores, una webcam para realizar todo el análisis de la información gráfica y de finales de carrera para el correcto funcionamiento del movimiento de la estructura. En cuanto a actuadores contaremos con motores paso a paso y con un servomotor.

El proyecto contará con la capacidad de seleccionar una opción de control manual por si el usuario desea localizar el dispositivo en una posición determinada. Para ello se ha desarrollado una aplicación móvil que, mediante transferencia de datos por Bluetooth permite el control de la estructura.

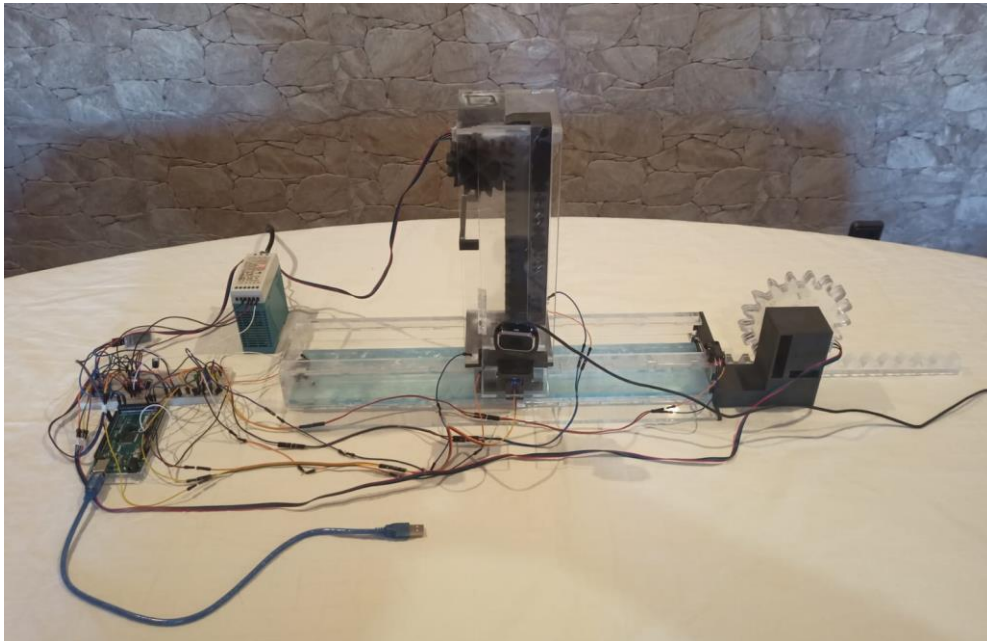


Imagen 1. Estructura final montada.

## 1.2. Antecedentes

Hoy en día, unos años tras el COVID-19 podemos decir que las formas de comunicación en la sociedad han evolucionado para adaptarse a cualquier situación, distancia o espacio. Es por ello que, en nuestro día a día se realizan una gran cantidad de reuniones de todo tipo de forma online. Para que estas sean totalmente prácticas, es necesario el dinamismo y la cercanía con la que se contaba en esas reuniones presenciales y como consecuencia una gran parte de la sociedad ha realizado inversiones en mejorar ese “set-up” del cual se dispone en casa o en el trabajo.

Debido a esto, el tener la capacidad de controlar la posición del dispositivo con el que dar ese soporte gráfico a las reuniones presenta ciertos beneficios. El primero de ellos, realizar un seguimiento facial para en todo momento estar de frente con el observador, esto es fundamental para conseguir ese efecto de cercanía y atención, vitales a la hora de mostrarnos frente a los demás, sobre todo en un entorno laboral. El segundo es la capacidad de controlar manualmente el dispositivo por si se quiere mostrar cualquier objeto, localizado en otra posición de la estancia, en el plano de la cámara. Estas opciones permiten al usuario mejorar la experiencia y calidad de las reuniones.



Imagen 2. Ejemplo de situación óptima a la cual adaptar el proyecto.

Este dispositivo se puede extrapolar a otras situaciones o campos como por ejemplo el relacionado con contenido audiovisual. Hoy en día con los avances en transmisiones en directo en distintas plataformas es de vital importancia contar con el control automatizado de todas las funcionalidades posibles en cuanto a la realización de estas, es por ello que, desde un centro de control, que puede ser remoto, se puede gestionar cada una de las tomas o cámaras de las que se disponga.

Un ejemplo práctico podría ser un plató de televisión donde un presentador vaya a realizar un monólogo y fuera mandatorio ese seguimiento de su persona o rostro mientras este se desplaza a lo largo del recinto o, donde en vez de requerir técnicos de imagen tras las cámaras exista un solo control de todo el dispositivo de forma remota.

En la imagen 2 se puede observar todo el equipo destinado a realizar estas labores de grabación en un programa en directo de televisión. Con una variante del proyecto adaptada a este espacio y requerimientos el programa podría ofrecer una mejor experiencia a los usuarios que diariamente conforma el público. Esta solución podría ser el hecho de establecer una estructura con sistema de guiado, basada en la misma funcionalidad que este proyecto, localizada con anclajes en el techo, habilitando la visión de todo el plató y toda la superficie de este mismo.



Imagen 3. Solución propuesta para la situación de la imagen 2.

### 1.3. Solución Adoptada

Como se ha mencionado previamente, el proyecto cuenta con una gran versatilidad a la hora de su implementación física. Es decir, en función del lugar y la necesidad para la cual se implemente podrá ser adaptado para cumplir con los requerimientos. Todo esto también vendrá determinado por el aspecto económico dando lugar así a opciones realizadas en distintos materiales o con mejor calidad y prestaciones.

En cuanto a la solución de este caso particular tiene que ser mencionado que se pretende mostrar la funcionalidad del proyecto, es decir, comprobar que la lectura de información de los sensores es correcta, que la interpretación del código procesa bien esta información y genera las instrucciones necesarias para enviar a los actuadores generando el movimiento óptimo.

A continuación, se muestra y justifica la elección de cada uno de los subsistemas que conforman la estructura, así pues, también se realiza una explicación detallada de la funcionalidad de estos.

### 1.3.1. Diseño 3D de la estructura

El diseño físico de la estructura requería de la capacidad de moverse en ambas direcciones de 2 ejes de translación para los movimientos horizontal y vertical y, por otro lado, la rotación sobre el eje vertical para el control de la perspectiva.

Para ello, el proceso de diseño se basó en un proceso en el que se creó un boceto a mano partiendo de cero, en ese proceso se detalla la forma en la que se va a realizar el movimiento y con que se van a transmitir las fuerzas. Por otro lado, hay que mencionar que, este es una idea inicial y genérica donde cada una de las piezas está sujeta a una amplia posibilidad de cambios tras analizar la viabilidad y funcionalidad.

Tras la verificación de la viabilidad se inicia el proceso de plasmación de ese boceto en el software de diseño 3D para formalizar todas las dimensiones y piezas de la estructura, ese periodo engloba todas las adaptaciones de tamaños, grosores e incluso modificaciones para reforzar la estructura, optimizar peso y material o mejorar su funcionalidad.

Por último, se procede a realizar el montaje de la estructura para comprobar que todo es correcto y que, tras implementar todos los componentes a esta, se cumple la función para la cual está diseñada.

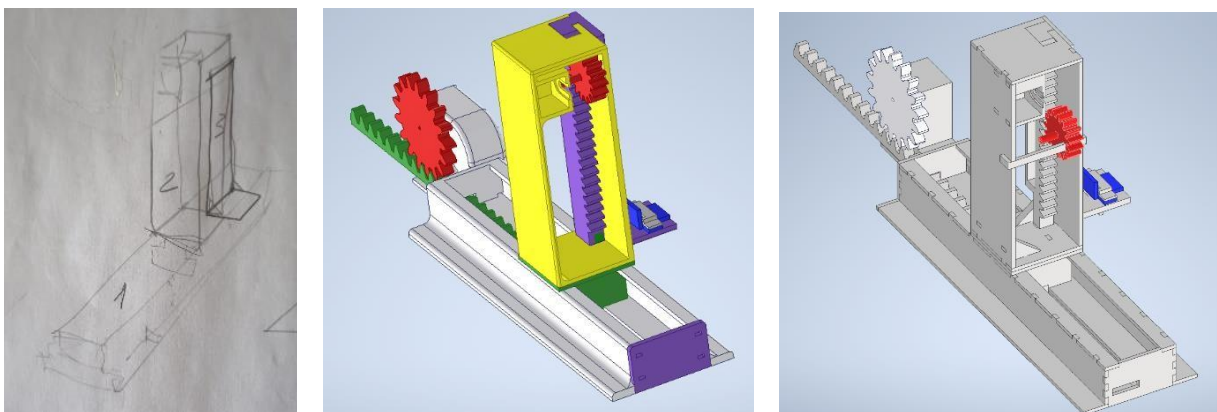


Imagen 4. Distintas fases del diseño de la estructura.

En el conjunto de imágenes se muestran, de una forma muy resumida las 3 fases, mencionadas previamente, por las cuales ha pasado el diseño del proyecto. En la primera se observa un boceto a mano inicial en el cual se intuye la naturaleza de los movimientos, así como lo que será la base de la estructura.

A continuación, vemos la estructura final diseñada en el software de edición 3D.

Por último, podemos observar la estructura una vez adaptada tras la elección de la solución para los materiales que la van a conformar y los métodos de obtención de esas piezas.

## 1.3.2. Materiales

Como bien se ha comentado previamente, el abanico de soluciones es amplio y de igual forma ocurre con la elección de materiales, en las figuras anteriores se aprecian 2 versiones de la estructura en función del material a usar, en la figura 2 se observa la estructura para construirse totalmente en material de impresión 3D (piezas macizas y con curvatura). Por contraparte, en la figura 3 se dispone de un diseño similar, pero con el detalle de disponer de una gran parte de piezas planas, estas serán óptimas para ser creadas en materiales como metal o metacrilato por corte láser.

La solución adoptada en este caso es una mezcla de ambos ejemplos mencionados. Debido al gran tamaño de la estructura, se convertía en una tarea inviable seleccionar el material de impresión 3D como opción para todo puesto que se hubiera tenido que realizar la estructura por secciones parciales de cada pieza y luego posteriormente el pegado de estas. Como solución a esto, se contempló el metacrilato como opción, siendo este un material con grandes capacidades mecánicas, con facilidad para su mecanizado por corte láser y su reducido coste.

Por tanto, la estructura contará con piezas planas de mayor dimensión realizadas en metacrilato cortado a láser y, piezas de menor tamaño o con mayores detalles realizadas en material de impresión 3D.

Destacar que para el diseño de las piezas de metacrilato se ha tenido que rehacer de 0 todo el diseño de la parte central de la imagen 4 pieza a pieza hasta conseguir su adaptación, mostrada en la parte derecha de esa misma imagen.

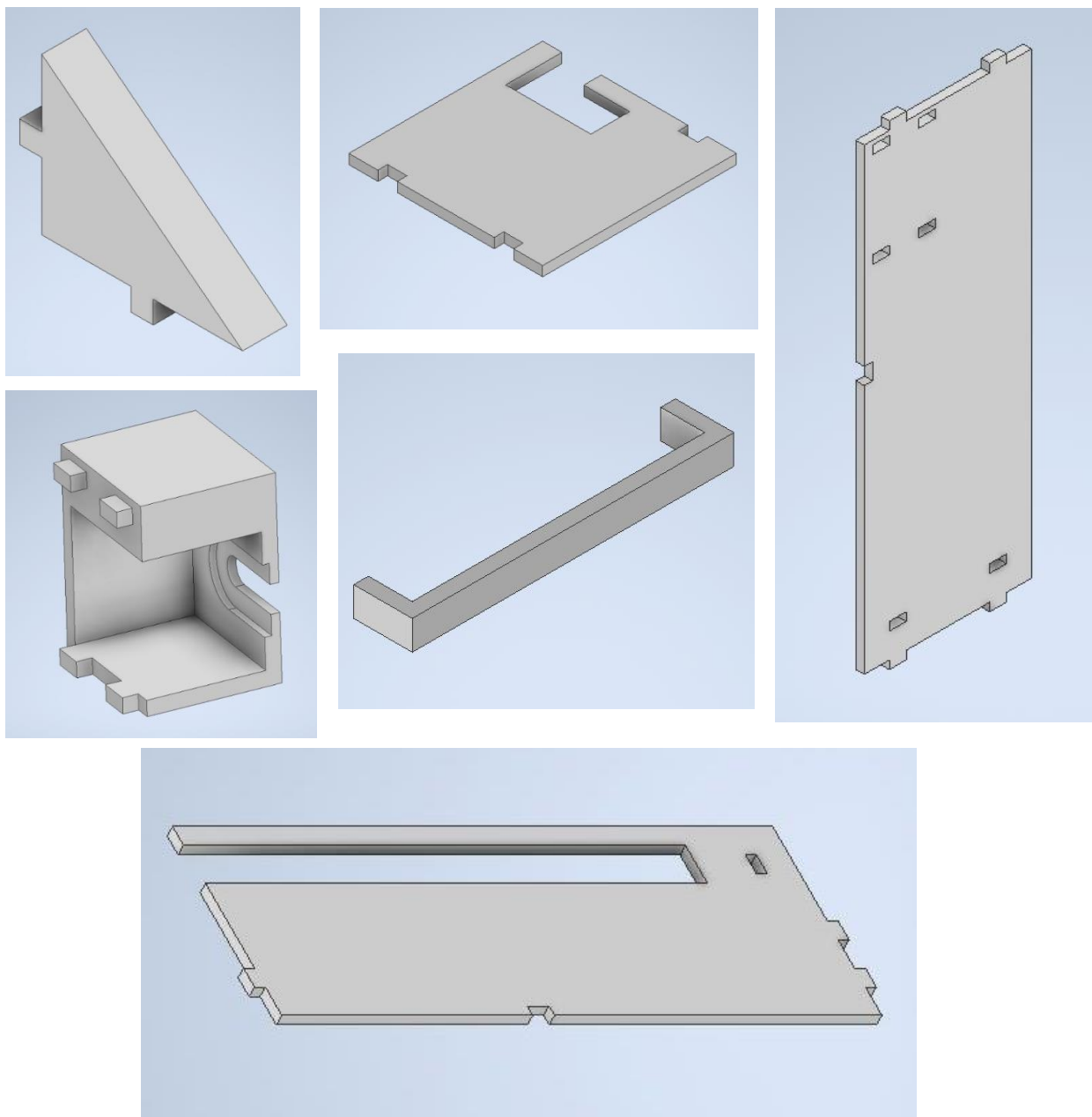


Imagen 5. Conjunto de piezas resultantes de la adaptación del componente vertical de la estructura.

Como se puede apreciar en la imagen 5, tras la selección de los materiales y métodos de mecanizado con los que se ha realizado la estructura, el diseño ha tenido que sufrir un remodelaje en el cual se ha pasado de una estructura sólida formada por 1 pieza a un conjunto equivalente con las adaptaciones requeridas compuesto por 6 tipos de piezas distintas.

De igual manera se ha procedido en cada uno de los subconjuntos de piezas para lograr el diseño definitivo de la estructura.

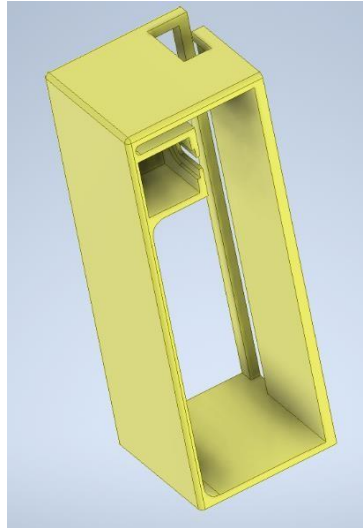


Imagen 6. Pieza base previa a la adaptación.

### 1.3.3. Software: Subsistema mecánico

En este apartado se menciona todos los softwares requeridos para los distintos campos abarcados por el subsistema mecánico del proyecto. En cada uno se explicará el motivo de uso, así como las principales ventajas e información respecto a ellos, además será expuesta la funcionalidad que ha proporcionado en sus apartados respectivos.

### 1.3.3.1. Diseño 3D

En cuanto al software utilizado, partiendo del desconocimiento acerca del campo del diseño 3D y tras haber buscado información y tutoriales acerca de estos, he usado Autodesk Inventor por distintas razones. La primera debido a que es un programa con una interfaz bastante intuitiva, la segunda puesto que había mucho contenido en internet para aprender acerca de su uso y como última razón, se incluía en el paquete de software gratuito con el cual contamos los alumnos miembros de la UPV, lo cual hace que sea muy afín con las características de presupuesto del proyecto.

Con este programa se han realizado los modelos CAD de la estructura en sus diferentes etapas (mencionadas previamente) así como, los ficheros de cada una de las piezas del conjunto general y sus planos (mostrados en el Documento N°2).

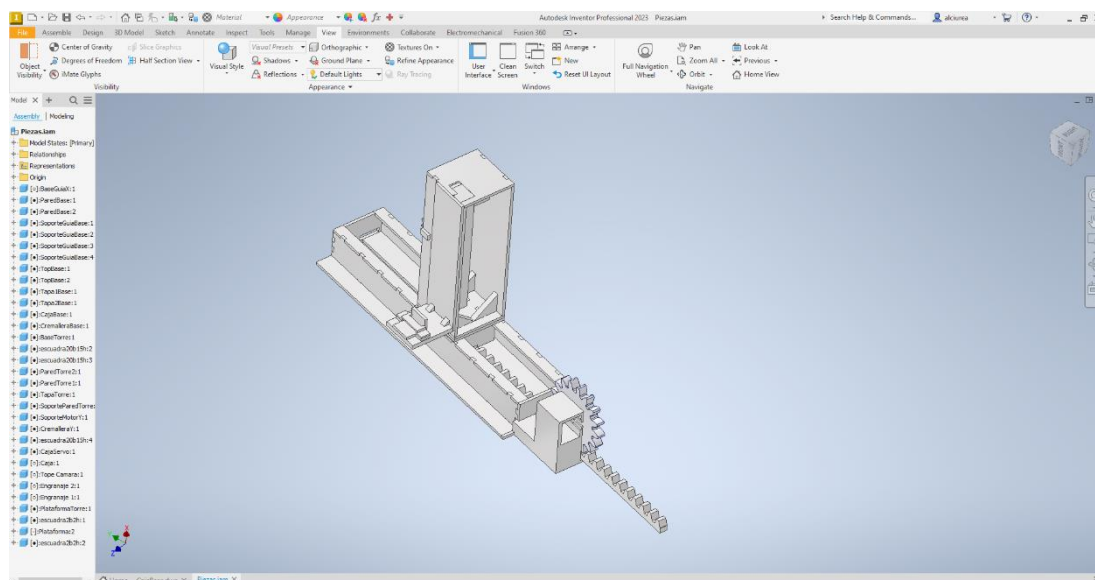


Imagen 7. Entorno Autodesk Inventor.



### 1.3.3.2. Impresión de piezas

Para la creación de ciertas piezas, y debido a que algunas de estas requerían de un gran detalle, se ha usado una máquina de sinterizado de nylon. La principal ventaja de ella es la optimización del trabajo respecto de las maquinas de filamento plástico convencionales.

El uso de esta tecnología permite la creación de una gran cantidad de piezas de tamaño reducido optimizándolo en el espacio útil de esta máquina.

Por otro lado, en términos de su funcionamiento es bastante práctico. Este se basa en lanzar una capa de polvo de nylon, a partir de ahí el láser de la máquina sinteriza las secciones requeridas para esa capa de polvo. Por consiguiente, una vez finalizado el proceso se obtendrá la totalidad del tanque de llenado con 2 productos distintos, el primero serán las piezas programadas, que serán macizas y de gran calidad, y el segundo, que representa al espacio total restante, que estará formado por polvo reciclable.

En cuanto a esto último hay que mencionar que el polvo que usa la maquina se compone por un 30% de polvo nuevo y un 70% de polvo reciclado. Estas cifras son lógicas puesto que el rendimiento de uso promedio por cada tanque de polvo suele rondar entre el 18 y 25%.

### 1.3.3.3. Corte láser

En el caso de las piezas restantes, cuyo denominador común es que son piezas planas y de grandes dimensiones, se ha usado metacrilato cortado por láser. Esto se debe a que el metacrilato es un material con buenas propiedades mecánicas y cuyo mecanizado es rápido y fácil en estos casos.

El funcionamiento de la maquinaria se basa en la modificación de la trayectoria del láser gracias a la localización y orientación correcta de 3 espejos con los que dispone, los 2 primeros situarán las coordenadas del plano X e Y. Por otro lado, el tercer espejo dirigirá el laser en la vertical sobre el punto de corte deseado. De esta forma y gracias a la facilidad de su uso se pueden realizar distintas piezas sobre una lámina que abarque la superficie útil de trabajo de este tipo de máquinas.

## 1.4. Componentes

Partiendo de la base que el proyecto es una solución con características y funcionalidades básicas adaptadas a un presupuesto reducido y al entorno de trabajo personal, el hardware usado no contará con una gran cantidad de componentes y estos tendrán un costo reducido, habiendo en el mercado una amplia variedad de opciones con mejor calidad o prestaciones.

A continuación, se muestran y explican de forma breve los componentes usados, en el anejo N°2 se encuentran todos los datasheets y especificaciones técnicas de cada modelo usado.

### 1.4.1. Servo

Los servomotores tienen la capacidad de establecer una posición dentro de su rango de movimiento y mantenerse en ella según el control aplicado.

Están formados por un motor de corriente continua y un circuito de control. Este último está basado en la modulación por ancho de pulso (PWM), cuyo funcionamiento radica en base a la duración del pulso en nivel alto respecto del pulso en nivel bajo, regulando estas duraciones se logrará que el servomotor se ponga en una posición u otra.

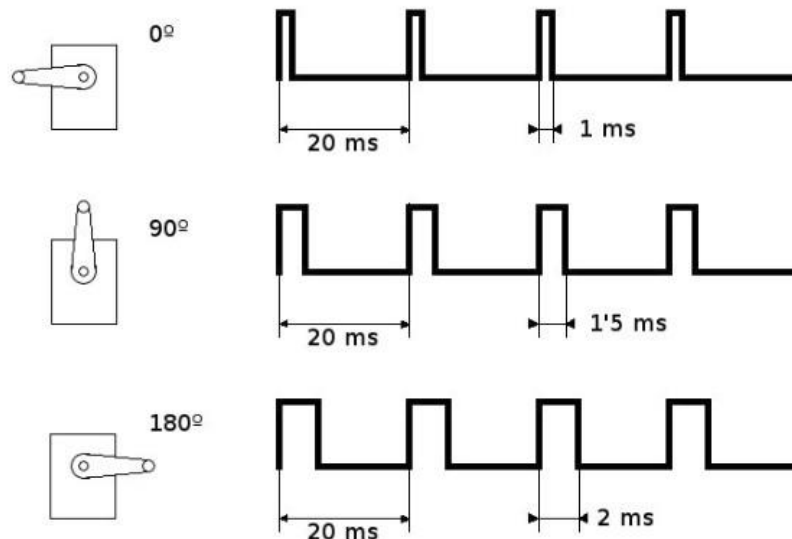


Imagen 8. Control PWM de un servo.

En el caso de la solución adoptada se usará un servomotor modelo SG90, este dispone de un rango de movimiento entre  $0^\circ$  y  $180^\circ$ , existen alternativas en el mercado con  $360^\circ$  de giro, pero no era necesario para el proyecto.

La finalidad de este será rotar el soporte de la webcam cuando sea necesario para ofrecer la posibilidad de localizar la cámara frente al rostro de una persona cuando esta realice un giro.



Imagen 9. Servo SG-90.

## 1.4.2. Stepper

Los stepper o motores paso a paso son motores de corriente continua sin escobillas en el que el total de la rotación se divide en un número de pasos.

En el proyecto se han usado 2 stepper modelo NEMA 17HS4401, comúnmente usado para la construcción de máquinas de impresión 3D o de CNC por sus características. Las más importantes son que cuenta con una capacidad de carga de 3.2kg/cm, es un motor bipolar y tiene la capacidad de regular la cantidad de pasos por vuelta, desde 200 hasta 3200 en función de la configuración de corriente y tensión de los drivers a los que esté asociado. En este caso, la configuración elegida será por tanto la de pasos completos (200 pasos/vuelta), con  $1,8^\circ$  por paso y un error de  $\pm 0,09^\circ$ .

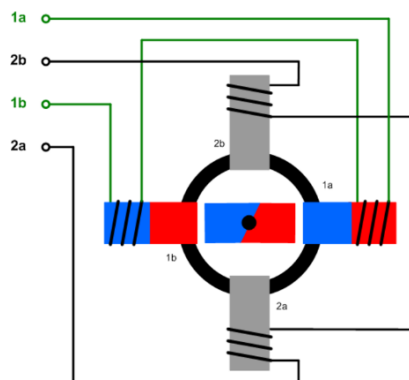


Imagen 10. Bobinado de un motor paso a paso.

Este es un ejemplo básico del esquema eléctrico interno que conforma los stepper mostrando las 2 bobinas que lo componen, estas se situarán en los pines que correspondan de cada uno de los drivers asociados para que estos gestionen su control.

### 1.4.3. Drivers

Para el control de los stepper, se requiere de unos drivers para facilitar la transmisión de información del script de control a los motores. En este caso se ha usado el modelo A-4988 el cual permite realizar un ajuste de los pasos por vuelta del stepper por tensión mediante la regulación de un potenciómetro en su superficie.

Por tanto, el esquema del circuito formado por todos los componentes quedará de la siguiente forma:

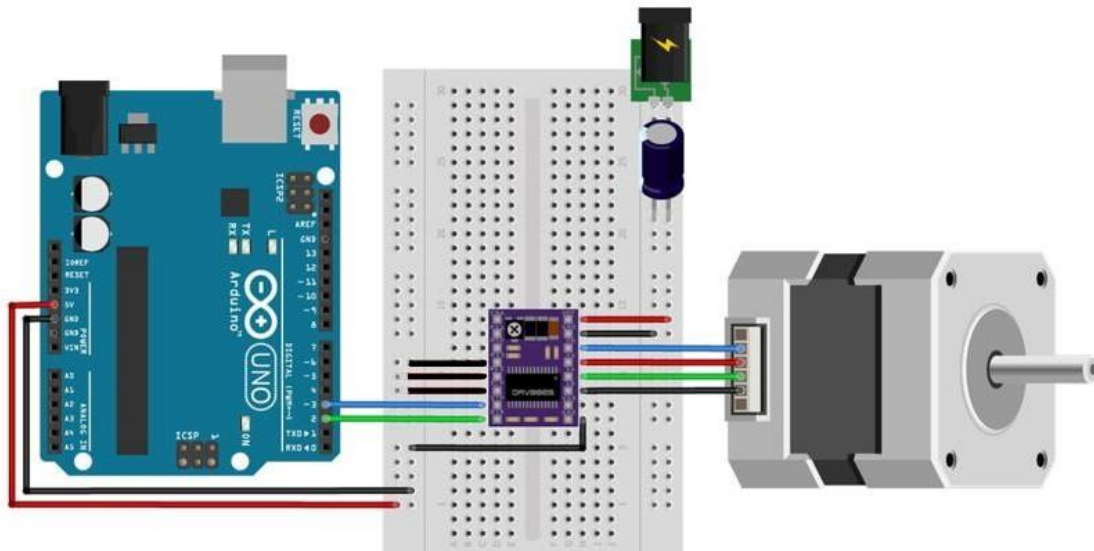


Imagen 11. Circuito formado por Nema17 y su driver A.4988.

Se puede apreciar que serán necesarios, además del stepper y del driver asociado, un condensador electrolítico de 100nF y una alimentación externa de 12V.

Por otro lado, para la configuración del driver será necesario acudir a su datasheet. A partir de este deberemos elegir el modo de funcionamiento en el que queremos que trabaje nuestro stepper. Entre las posibilidades se encuentran desde pasos enteros hasta dieciseisavos de paso, conllevando así una pérdida de par de giro de forma inversamente proporcional al número de pasos total.

En este caso se ha decidido usar pasos enteros luego deberemos ajustar los drivers en función de ello.

Cálculo teórico de ajuste de los drivers:

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Imagen 12. Tabla de configuración de los pines MSx.

Partiremos de la configuración de los pines MSX en función del tipo de paso que queramos.

A partir de ahí nos basaremos en una de las ecuaciones pertenecientes al datasheet que relacionan la tensión de referencia a ajustar con la corriente máxima.

-Datos necesarios: Corriente Max NEMA 17 = 1.7A  
Resistencia SMD “R100” = 0.1 Ω

-Cálculos:

$$I_{MAX} = \frac{V_{REF}}{8 \cdot R_{SMD}}$$

$$V_{REF} = \frac{1.7}{8 \cdot 0.1}$$

$$V_{REF} = 1.36V$$

Puesto que pretendemos trabajar en pasos completos, y siguiendo una tabla de referencia del datasheet deberemos de usar un 70.71% del valor calculado.

Full Step #	Half Step #	1/4 Step #	1/8 Step #	1/16 Step #	Phase 1 Current [% I <sub>tripMax</sub> ]	Phase 2 Current [% I <sub>tripMax</sub> ]	Step Angle (°)
	1	1	2	1	100.00	0.00	0.0
				2	99.52	9.80	5.6
			2	3	98.08	19.51	11.3
				4	95.69	29.03	16.9
		2	3	5	92.39	38.27	22.5
				6	88.19	47.14	28.1
			4	7	83.15	55.56	33.8
				8	77.30	63.44	39.4
1	2	3	5	9	70.71	70.71	45.0

Imagen 13. Tabla para el cálculo de la tensión de referencia en función de la configuración de pasos.

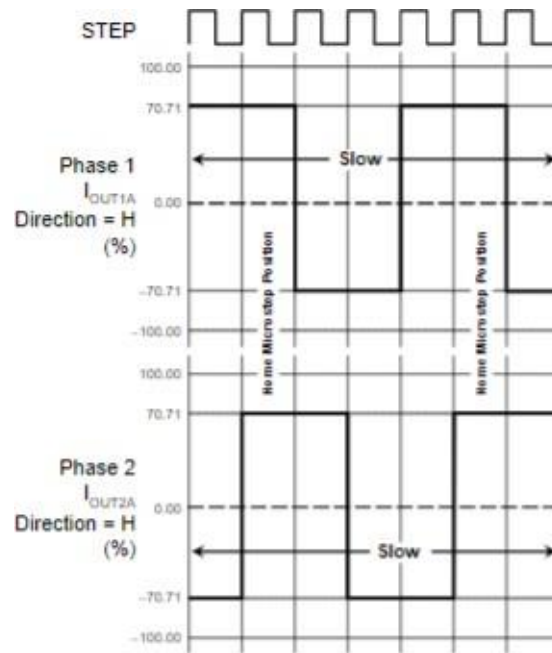


Imagen 14. Amplitud de pulso de control.

Por tanto, el valor de tensión de referencia al que tendremos que ajustar el driver será aproximadamente  $V_{REF} = 0.952V$ . Posteriormente se ajustará a este valor el potenciómetro de la superficie del driver con la ayuda de un multímetro.

#### 1.4.4. Arduino

Arduino es una plataforma “Open-Source”, es decir puede modificarse el diseño original según necesidades. Esto conlleva a la gran variedad de modelos de placas y módulos que se encuentran hoy en día en el mercado.

En este caso, según las necesidades de conexionado y prestaciones, se ha decidido usar un Arduino Mega 2560. Este modelo tiene la gran ventaja de tener una gran cantidad de pines digitales de E/S (un total de 54) así como 4 pares de pines de comunicación serie. Algunas de las características más importantes de este componente son:

256 kB de memoria FLASH (espacio disponible para almacenar el programa o sketch).

8 kB de memoria SRAM (es donde se crean las variables declaradas en el programa).

4 kB de memoria EEPROM (permite almacenar datos que se conserven, aunque se reinicie o falle la alimentación).

Frecuencia de CPU Máxima: 16MHz

Voltaje de Operación máximo: 6.0V (aunque se recomienda no sobrepasar los 5V).

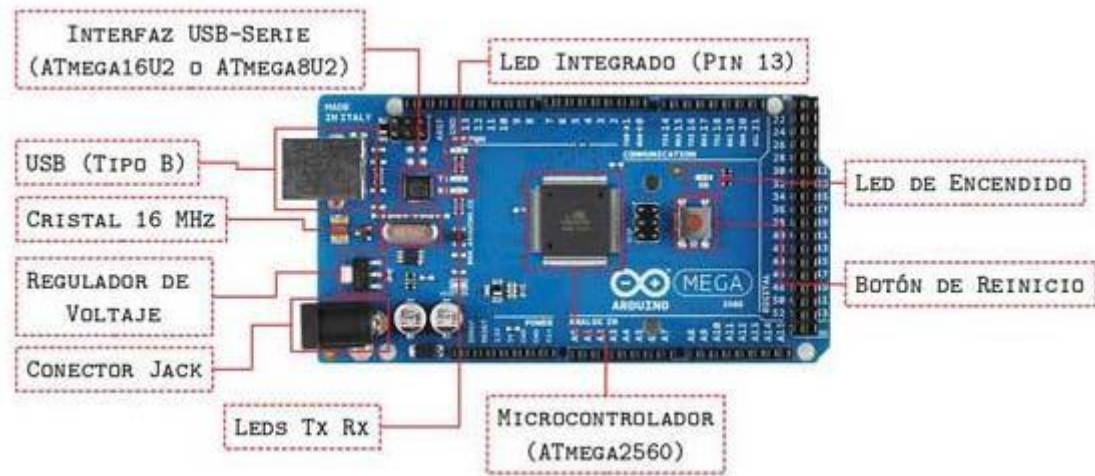


Imagen 15. Planta del Arduino Mega 2560.

En este proyecto el Arduino realizará las funciones de, lectura de información transmitida desde el script de Python con el procesado en tiempo real de la imagen, así como la información recibida desde la app mediante el módulo de bluetooth para el control manual de los actuadores.

A continuación, se hace mención la localización a nivel de pines de cada uno de los componentes electrónicos requeridos en el montaje:

- Finales de carrera: 47 (limitación de bajada), 49 (limitación de subida), 51 (limitación derecha) y 53 (limitación izquierda).
- Drivers: Dir y Step del motor con movimiento horizontal en los pines 22 y 24 respectivamente. En el caso del motor con movimiento vertical corresponden a los pines 26 y 27.
- Módulo Bluetooth: pines 14 y 15 (correspondientes a TX3 y RX3) para la comunicación serie.
- Servo: pin 4 para el control PWM.
- Alimentación: pin de 5V y GND.

### 1.4.5. Módulo BT

Para el control manual se ha creado desde cero una aplicación móvil basada en la transmisión de datos por bluetooth, puesto que el Arduino Mega no dispone de conexión bluetooth integrada se ha de conectar un módulo externo que nos posibilite ese tipo de conexión, es por ello que se ha incorporado el módulo bluetooth HC-06.

Este modelo en particular solo puede funcionar como esclavo a diferencia del modelo HC-05 que también lo puede hacer como maestro.

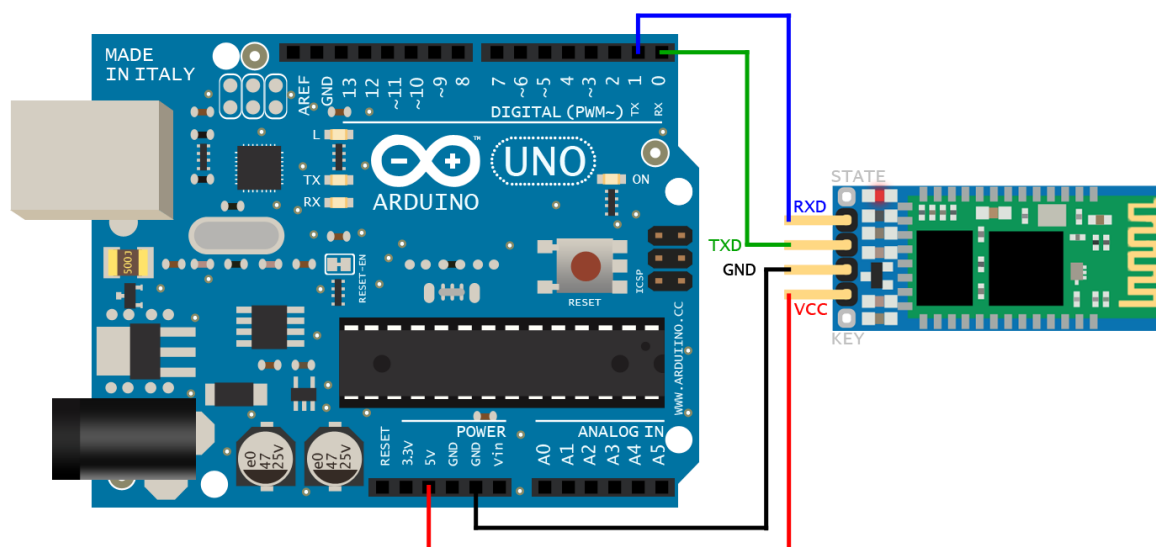


Imagen 16. Conexión del módulo bluetooth HC-06.

En la imagen se muestra el esquema de conexión del módulo respecto al Arduino, para ello lo que deberemos de tener en cuenta es que debe recibir la alimentación de 5V y, por otro lado, deberemos conectar los pines RXD y TXD a sus respectivos TX y RX de la placa. En el caso particular del Arduino Mega, se usarán los pines RX3 y TX3 puesto que por el canal de transmisión principal llegarán los datos provenientes del script de Python.

### 1.4.6. Fuente de alimentación

Además de la alimentación de 5V proporcionada por la placa Arduino deberemos alimentar una de las entradas de los drivers, para realizar el control del stepper, con una tensión de 12V. Es por ello que requeriremos de una fuente de alimentación externa que los proporcione.

Se ha usado pues, el modelo DIN-Rail de la marca MeanWell, este proporciona una potencia de 60W (12V y 5A) y dispone de 2 salidas con esas características, es decir, una para cada una de las alimentaciones de los 2 stepper.





Imagen 17. Fuente de alimentación externa.

## 1.4.7. Finales de carrera

Un final de carrera es un sensor de contacto con la finalidad de enviar señales para modificar el estado de un circuito electrónico. En este caso se han utilizado 4 unidades del modelo CESFONGER para limitar el recorrido de los movimientos de translación vertical y horizontal.

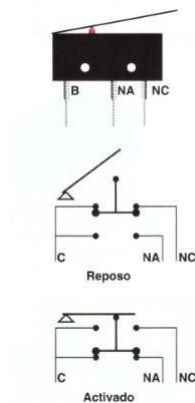


Imagen 18. Esquema electrónico final de carrera.

## 1.5. Programación

La programación ha sido una de las partes fundamentales del proyecto tanto para la obtención de información de los sensores y control de los actuadores, como para la creación de una interfaz gráfica para la app móvil. En el proyecto se han realizado scripts en distintos entornos de programación basados tanto en lenguaje C como en programación por bloques.

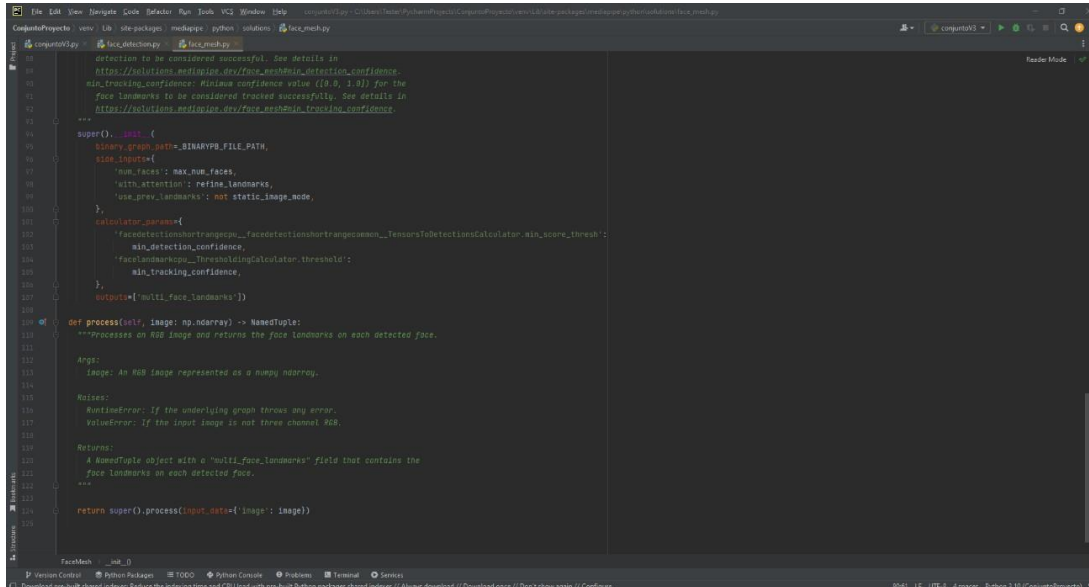
### 1.5.1. Python

Python es uno de los lenguajes de programación más versátiles que existen, su uso se da en distintos campos como pueden ser desarrollo web, visión artificial y procesamiento de imágenes o desarrollo de aplicaciones o videojuegos. Está basado en lenguaje C y es muy sencillo de leer y escribir debido a que se parece en gran medida al lenguaje humano. Otra de sus características es que es de código abierto luego permite desarrollar el software sin limitaciones.

El uso de Python en el proyecto permite obtener los datos requeridos de la captura de video. A partir de ahí se realizarán los cálculos necesarios para transformar esa información en una respuesta que genere los movimientos e instrucciones necesarias para articular los movimientos de la estructura. Posteriormente enviará esa información a través del canal serie al Arduino.

#### 1.5.1.1. PyCharm

PyCharm es un entorno de desarrollo usado para el lenguaje de programación en Python. La principal ventaja que ofrece es que no es solo un editor de código. Además, tiene un depurador e interprete para crear los programas que necesitemos, así como a detectar los posibles errores del código.



```

108 detection to be considered successful. See details in
109 https://politecnico.valencia.es/info/eng/men/entresolos_confidence.
110 min_tracking_confidence: #Minimum confidence value (0.0, 1.0) for the
111 face landmarks to be considered tracked successfully. See details in
112 https://politecnico.valencia.es/info/eng/men/entresolos_min_tracking_confidence.
113 """
114 super().__init__(
115     binary_graph_path=BINARY_GRAPH_PATH,
116     load_inputs={
117         'num_faces': MAX_NUM_FACES,
118         'min_tracking_confidence': min_tracking_confidence,
119         'use_prev_landmarks': not STATIC_IMAGE_MODE,
120     },
121     evaluator_params={
122         'face_detector': 'opencv',
123         'face_detector_params': {'face_detector_params': {'min_score_thresh':
124             MIN_DETECTION_CONFIDENCE,
125             'face_landmarks_5points_calculator_thresholds':
126                 MIN_TRACKING_CONFIDENCE,
127             'min_tracking_confidence':
128                 min_tracking_confidence,
129             'outputs': ['multi_face_landmarks']}
130     },
131 )
132
133 @staticmethod
134 def process(image: np.ndarray) -> NamedTuple:
135     """Processes an RGB image and returns the face landmarks on each detected face.
136     """
137     Args:
138         image: An RGB image represented as a numpy ndarray.
139     Returns:
140         A NamedTuple object with a 'multi_face_landmarks' field that contains the
141         face landmarks on each detected face.
142     """
143     return super().process(input_data={'image': image})
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Imagen 19. Entorno de trabajo en PyCharm.

### 1.5.1.2. OpenCV

OpenCV (“Open Computer Vision”) es una biblioteca de visión artificial desarrollada originalmente por Intel. Es una biblioteca de uso libre, multiplataforma (Linux, Mac OS X, Windows y Android) y para distintas arquitecturas de hardware (x64 o ARM). Además, entre sus áreas de aplicación se destacan: reconocimiento facial, robótica móvil, realidad aumentada, ...

Inicialmente está desarrollado en lenguaje C++ pero también es apto para lenguajes como Python, Java y Matlab entre otros.

En este caso, usaremos OpenCV para la obtención de la información generada por la cámara en tiempo real, más tarde esta será tratada en el mismo entorno para la obtención de la información que se requiere.

### 1.5.1.3. Librería Serial

La librería serial tiene la funcionalidad de permitir la comunicación serie para transmitir la información generada en el script en PyCharm a través de los puertos serie del Arduino para que este interprete la información.

En este caso será de gran utilidad puesto que requerimos de esa transmisión de datos para que, posteriormente, el script en Arduino lo interprete. Enviaremos pues, tras hacer los cálculos pertinentes, como se ha explicado en apartados previos, los caracteres (en código ASCII) correspondientes a las acciones a hacer.

## 1.5.1.4. Librería MediaPipe

La librería MediaPipe es aquella que nos va a permitir realizar el proceso de la detección de objetos 3D en tiempo real, su funcionamiento principal se basa en la detección de imágenes en 2D y la posterior estimación de las poses del objeto a través de un modelo implementado basado en machine learning.

En este caso se usarán los algoritmos de reconocimiento facial y de la malla facial con los que cuenta por defecto y, a partir de ellos, obtendremos todos los datos necesarios para su posterior análisis.

### 1.5.1.4.1. Reconocimiento facial

Este es uno de los algoritmos predefinidos dentro de la librería media pipe, su función básica el retornar las coordenadas de los píxeles a los que asociados a distintas partes del rostro (concretamente 6). En este caso el algoritmo generará un rectángulo alrededor del rostro. A partir de ahí y, tras la adaptación de unidades de píxeles a distancia real, trataremos de obtener el centro del rostro que es equivalente al centro del rectángulo generado.

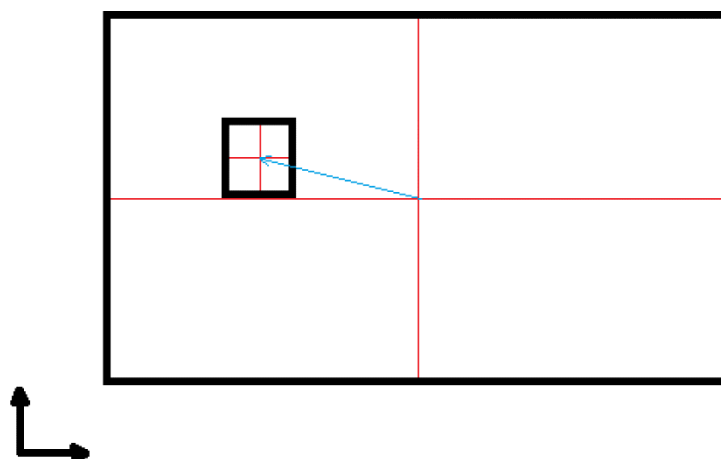


Imagen 20. Interpretación de datos de coordenadas extraído por el algoritmo de reconocimiento facial.

Más tarde, será sencillo generar las instrucciones de traslación en ambos ejes, lo único que habrá que hacer será extraer y comparar las coordenadas del centro de la imagen respecto del centro del rectángulo del rostro. A partir de esa comparación se creará pues, la información generada en cuanto al movimiento horizontal (izquierda o derecha) y al movimiento vertical (arriba y abajo).



Imagen 21. Representación de los datos que extrae el algoritmo.

### 1.5.1.4.2. Malla facial

El otro algoritmo que se ha usado es el de malla facial, este algoritmo nos generará una malla sobre el rostro. Esta malla está definida por una estructura basada en regiones triangulares entre los 468 puntos clave repartidos en ella. La ventaja de este algoritmo es que puede reconocer con una precisión adecuada la profundidad del rostro.

Por tanto, para el reconocimiento de la inclinación horizontal del rostro lo que se pretende hacer es, extraer unos puntos en específico y en base a su posición relativa, realizar unos cálculos geométricos para determinar la posición. En este caso, los puntos elegidos se dividirán en 3 secciones de la cara (nariz, patilla izquierda y patilla derecha).

Los puntos de la nariz son: 5, 6, 195 y 197.

Los puntos de la patilla izquierda son: 93, 127, 132 y 234.

Los puntos de la patilla derecha son: 323, 356, 361 y 454.

Una vez localizados los puntos hay que determinar cuándo un rostro está orientado a izquierda o derecha o si, por el contrario, está colocado de frente a la cámara.

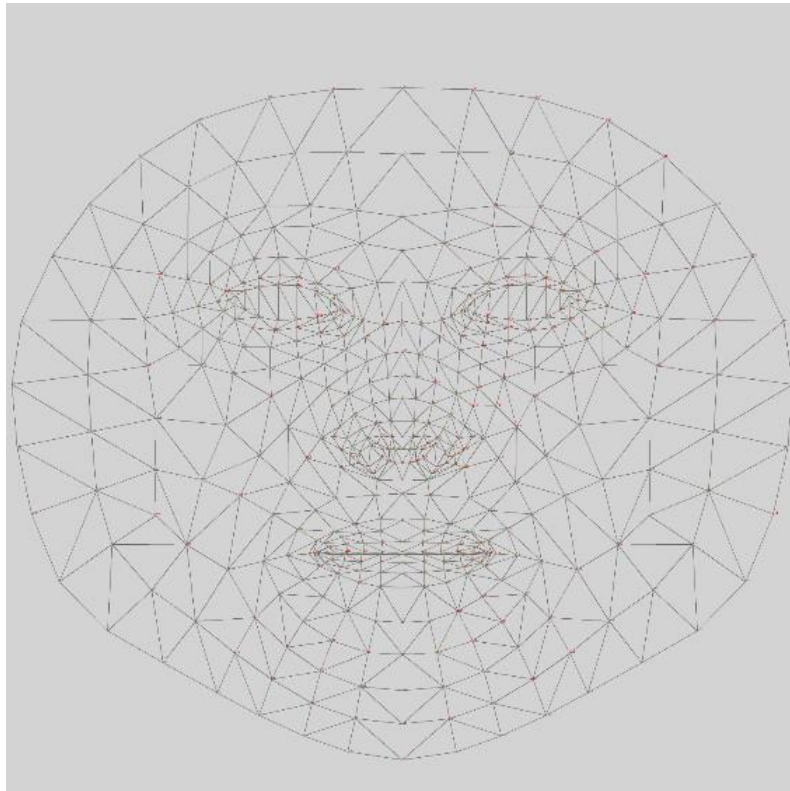


Imagen 22. Modelo canónico del algoritmo de malla facial.

Para ello y puesto que el algoritmo detecta profundidades, se observa en las pruebas que si el rostro se inclina los puntos de los extremos se siguen representando, pero por la parte trasera. Por tanto, se aprecia que, si está inclinado para un costado, los puntos de una patilla se acercarán en mayor medida que los de la otra a la hilera de puntos central. Se procederá pues a realizar una comparación de las distancias en el plano en 2D y en función de las distancias relativas de las hileras de puntos con las patillas se determinará la orientación del rostro.

Una vez extraídos todos los datos necesarios, se codificará la información y se enviará directamente al Arduino tras haberla convertido previamente a tipo carácter mediante el código ASCII. Es decir, para el envío de que la imagen esta inclinada hacia la izquierda se enviará un carácter 'I' en código ASCII por el canal serie.

A continuación, se adjunta un link para ver el modelo canónico con más precisión:  
[https://raw.githubusercontent.com/google/mediapipe/master/mediapipe/modules/face\\_geometry/data/canonical\\_face\\_model\\_uv\\_visualization.png](https://raw.githubusercontent.com/google/mediapipe/master/mediapipe/modules/face_geometry/data/canonical_face_model_uv_visualization.png)



Imagen 23. Representación gráfica de los datos que extrae el algoritmo.

## 1.5.2. Arduino

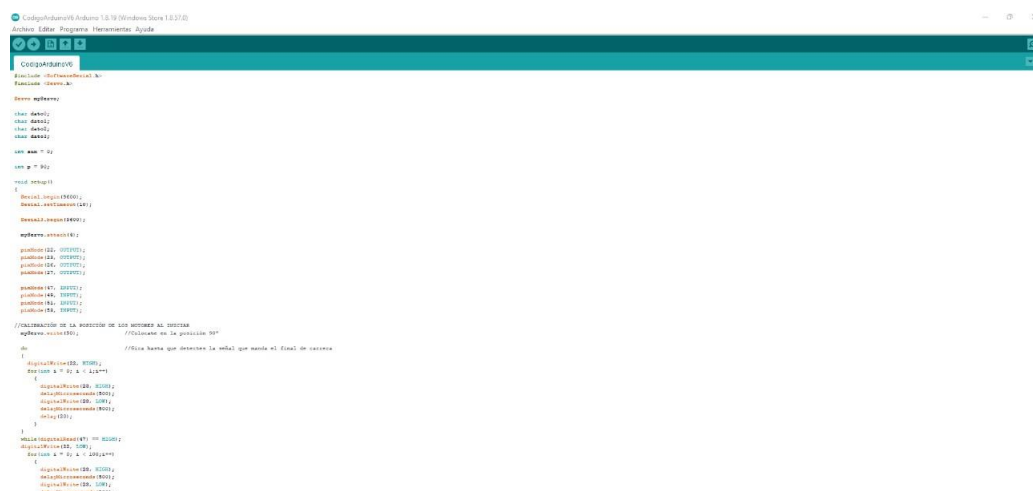


Imagen 24. Entorno de trabajo Arduino.

El script de Arduino permitirá controlar los actuadores en función de la información recibida previamente por los canales serie. Este buscará realizar un control (todo/nada) para compensar el error cometido entre las posiciones y orientaciones relativas del rostro respecto al centro de la imagen o la posición de orientación “de frente”. Hay que destacar que estos valores se han de conseguir por el método de “prueba/error” puesto que depende de distintos factores físicos debido a la solución adoptada. Estos pueden ser, desde el tamaño o recorrido de la estructura, distancia al usuario o el margen de error permitido.

Con relación a esto también se definirán unos márgenes para realizar una histéresis en el control de los movimientos puesto que, de no ser así, la estructura estaría en una oscilación constante para compensar ese error y eso no sería un comportamiento que proporcione una imagen estable. Estos márgenes se establecerán de igual forma que las relaciones para determinar las acciones requeridas y son válidos para la solución adoptada, en cualquier otro caso habrá que calibrar de nuevo por “prueba/error” en función de los parámetros anteriormente mencionados.

Por último, hay que comentar que también se usará la biblioteca Servo.h. Esta nos permitirá realizar de una manera más cómoda el control del SG90.

### 1.5.3. App Inventor

Como añadido al control automatizado explicado a lo largo de los puntos previos se ha incluido la posibilidad de activar un modo manual, es decir, mediante el uso de una app móvil podremos controlar el movimiento de la estructura sin importar los datos recibidos de la cámara.

Para la creación de la aplicación se ha recurrido a un software gratuito online llamado MitAppInventor. En esta plataforma podremos diseñar, con algunas limitaciones, tanto la interfaz gráfica como a nivel funcional de toda la aplicación. El lenguaje de programación es por bloques luego la realización de estos será bastante intuitiva pese a no haber realizado nada parecido previamente.

A continuación, se muestra el entorno mostrando las distintas interfaces, opciones y flujo de trabajo en ella:

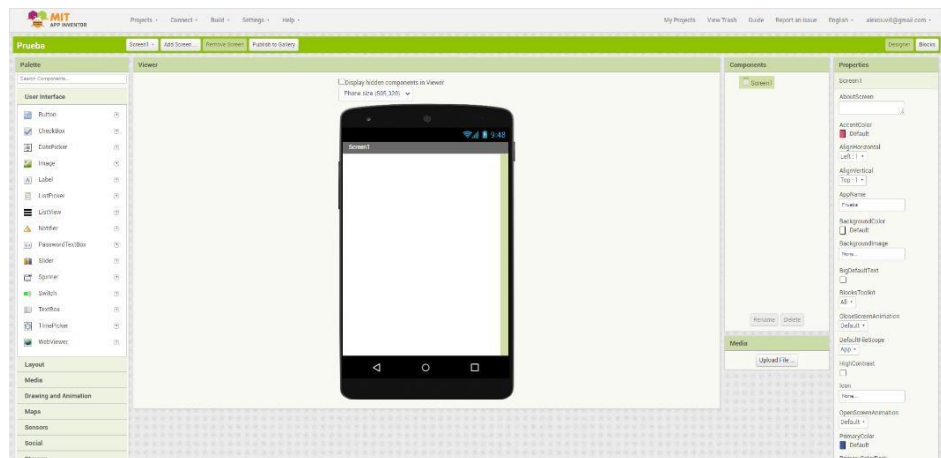


Imagen 25. Entorno de trabajo MitAppInventor.



Como se puede observar, una vez iniciado un proyecto dispondremos de una interfaz bastante limpia en la cual se visualizan, a la izquierda, un menú con todas las opciones y funciones que se pueden adaptar a la app. En el lado opuesto contaremos con un menú de las funciones que hayamos implementado, así como un submenú, en función del que se haya seleccionado, para ofrecer la posibilidad de realizar una pequeña modificación de parámetros como tamaño de botones, localización o transparencia entre ellos.

Una vez configurado todo lo relativo a la interfaz gráfica de la aplicación, pulsando en el botón de “Blocks” situado en la parte superior derecha, podremos avanzar al menú de programación.

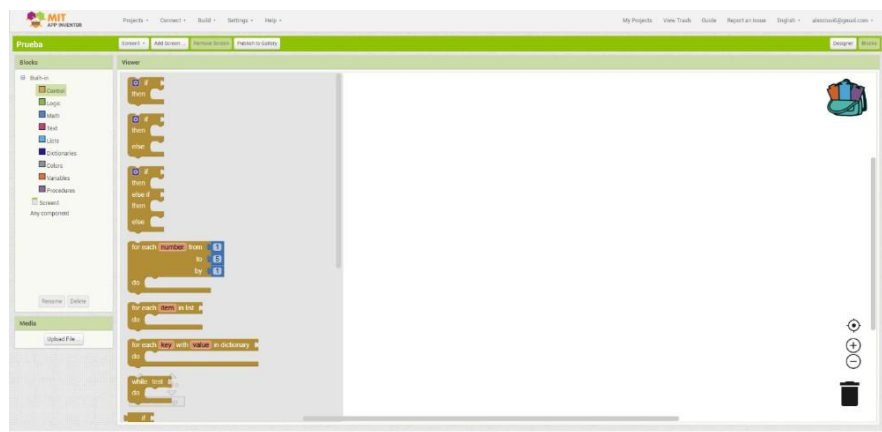


Imagen 26. Entorno de programación MitAppInventor.

En este menú contaremos con un apartado que contiene los distintos comandos y opciones de bloques para configurar el funcionamiento de la aplicación. Para diseñar el conjunto de bloques ubicaremos estos mismos en la zona blanca, con solo arrastrar los bloques conseguiremos conectarlos entre sí y realizar la función deseada.

Como comentaba previamente este menú de bloques está dividido por conjuntos de bloques de distintos colores según la funcionalidad que tengan.

Destacar que, para mejorar el factor personalización, este software nos permite subir archivos en formato de imagen para graficar distintos iconos u ofrecer la posibilidad de ver, desde la misma aplicación la evolución del estado de algún parámetro.

## 1.6. Estudio económico

En cuanto al presupuesto asociado al proyecto, toda la información detallada tanto a este como al estudio económico o inversión total realizada se encuentra localizada en el documento N.º 4: Presupuesto.

En dicho documento se justifica el importe total resultante de **DOS MIL CINCUENTA Y DOS EUROS CON DIECIOCHO CÉNTIMOS..**

## 1.7. Conclusiones y líneas abiertas

### 1.7.1. Líneas abiertas

Como bien se ha comentado a lo largo de todo el documento, la solución de este proyecto se adecua al caso específico para el cual está diseñado. La principal ventaja que presenta es que prácticamente su totalidad se puede/debe adaptar para el caso específico en el cual se quiera implementar. Hay que destacar que, la mejor solución será aquella que se adapte a las necesidades y requerimientos propuestos.

A continuación, se exponen distintas posibilidades en cuanto a la selección de componentes, diseño o código para adecuarlo a las posibilidades del sistema.

#### 1.7.1.1. Materiales

- Aluminio: material con buenas características mecánicas y baja densidad.
- Filamento 3D: capacidad de realizar con gran precisión cada una de las piezas logrando un diseño más personalizado.
- Madera: coste reducido, sencillez de corte de las piezas.

### 1.7.1.2. Diseño

-Guía horizontal curva: modificación de la trayectoria horizontal obteniendo un arco de circunferencia del radio R requerido según la situación.

-Orientación facial vertical: así como se puede realizar el control de la orientación vertical de la cámara se podría articular un movimiento de rotación sobre el eje horizontal permitiendo así la detección de si un rostro se inclina hacia arriba o hacia abajo.

-Orientación facial eje Z: de igual forma que la explicada previamente, pero consiguiendo articular un movimiento de rotación sobre el eje Z (atravesando el plano XY) para conseguir rotar la cámara y que la imagen no se observe volteada (respecto al eje horizontal).

### 1.7.1.3. Script y funcionalidad

-Pad de control físico: uso de un joystick implementado de forma física en el circuito para transmitir el control del sistema.

-Interfaz de control en ordenador: de igual forma que se puede realizar un control manual de forma remota en la app móvil, pero en el mismo dispositivo donde se ejecutan los códigos.

-Almacenaje en memoria de posiciones: contar con la posibilidad de almacenar en una matriz las posiciones a las que tienen que acudir los actuadores una vez se seleccione de estando en el modo de control manual.

### 1.7.1.4. Coste

Como en cualquier ámbito, el coste de un proyecto es un determinante a la hora de su realización, en función de esta variable se podrá realizar un diseño más robusto, liviano, de mayor tamaño o con componentes más precisos y fiables. A partir de ahí hay que mencionar que se puede realizar el proyecto con materiales y componentes más baratos consiguiendo un resultado limitado o, por el contrario, realizar una inversión mayor obteniendo así movimientos más fluidos, una estructura depurada o un alcance mayor del rango de la imagen.

En cualquier caso, hay que mencionar que el proyecto será factible para casi cualquier bolsillo.

## 1.7.2. Conclusiones

En este apartado se mencionan todas las conclusiones relacionadas tanto con la realización del proyecto, así como las mejoras en competencias transversales tras haberlo logrado.

La realización de este proyecto parte de un proceso de selección, tras el descarte deliberado de otras opciones a nivel personal, considero que es una pequeña muestra de todas las capacidades y recursos adquiridos a lo largo del grado.

Inicialmente, consideraba que el proyecto presentaría la mayor dificultad en la adaptación de los distintos campos (electrónica, programación, control, diseño 3D y visión artificial), una vez finalizado he podido observar que al fin y al cabo cada una de las partes requiere de la otra y, tras la correcta división y programación de las tareas, el objetivo se alcanza de una forma asequible.

Como añadido cabe mencionar que, sería de gran ayuda la inclusión de todos aquellos campos que he mencionado donde mi conocimiento era reducido o nulo al inicio del proyecto. Sería de gran utilidad poder estudiarlos a lo largo del grado aunque fuera de forma reducida.

Quiero comentar que, paralelamente a todo lo que conlleva el desarrollo de un proyecto desde el inicio, existe una parte fundamental en cuanto a trámites. En este caso radica en la impresión de la estructura 3D y una ayuda en la compra de algunos componentes. Los trámites relacionados con el primer caso radican en cuanto al ajuste del diseño a ciertos materiales, así como el razonamiento de la mejor solución para facilitarlos o el envío y creación de planos y apuntes extra sobre las piezas para ser interpretado.

En el caso del segundo los trámites radican en insistencia y el hecho de intentar aprovechar los recursos, es decir, desde el mismo descubrimiento de la posibilidad de la ayuda para materiales, hasta la selección de estos con los condicionantes impuestos por la universidad o la gestión y comunicación con los aquellos con los que se debe realizar. He de destacar aquí la ayuda de mi tutor que descubrió esta posibilidad, así como su involucración y consejos para la realización de los trámites.

He de recalcar que habrá que proteger la propiedad intelectual de todo el diseño en caso de su comercialización para autorizar su empleo por terceros.



## 1.8. Bibliografía

- Información de todo lo relacionado con la visión artificial en tiempo real en Python.  
<https://google.github.io/mediapipe/>
- Curso aprendizaje programación en Python.  
<https://youtube.com/playlist?list=PLU8oAlHdN5BlvPxziopYZRd55pdqFwkeS>
- Curso aprendizaje diseño 3D Inventor.  
<https://youtube.com/playlist?list=PLrnrP-LizQAiqbgODEc2gC1M6nFHtuLBo>
- Configuración driver A-4988.  
<https://www.youtube.com/watch?v=L0RwBb-lLts&t=286s>

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

## **Anejo N. °1: CÓDIGO**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## ÍNDICE

1. Código.....	39
1.1. Código Python.....	39
1.2. Código Arduino .....	47
1.3. MitAppInventor .....	70



## 1. Código

En este anejo se adjuntan principalmente todo el código necesario para el correcto funcionamiento del sistema. Destacar que en él se incluyen comentarios en puntos determinados para facilitar la comprensión del funcionamiento del código, además se hace un pequeño énfasis en la explicación del proceso de creación de la aplicación móvil.

### 1.1. Código Python

El código implementado en Python es el que se muestra a continuación, destacar que se incluyen los comentarios indicados en el mismo, así como la explicación detallada de algunas partes que lo requieren para la correcta comprensión.

#Importación de librerías

```
import cv2
import mediapipe as mp
import serial
import math
```

#Configuración del puerto serie del Arduino

```
com = serial.Serial("COM11", 9600, write_timeout=10)
```

#Declaración de los movimientos a enviar al arduino y conversión a bytes para poder enviarlo

```
derecha = bytes("d", 'ascii')
izquierda = bytes("i", 'ascii')
inclinadoi = bytes("h", 'ascii')
inclinadod = bytes("e", 'ascii')
recto = bytes("r", 'ascii')
arriba = bytes("w", 'ascii')
abajo = bytes("j", 'ascii')
parox = bytes("p", 'ascii')
paroy = bytes("l", 'ascii')
```

#Creación de las variables que almacenan los datos recogidos por las funciones  
#face\_mesh y face\_detection. Modificación del dibujo con drawing\_utils.

```
Malla_Facial = mp.solutions.face_mesh
Detector = mp.solutions.face_detection
```





Dibujo = mp.solutions.drawing\_utils

#Realización de la videocaptura y establecimiento de los píxeles de la imagen

```
videocap = cv2.VideoCapture(0, cv2.CAP_DSHOW) #Selección de la cámara a usar (0 = cámara por
#defecto en portatil, 1 = cámara externa)

videocap.set(3, 1280)
videocap.set(4, 720)
```

#Modificación de los parámetros de la malla facial

```
with Malla_Facial.FaceMesh(
    static_image_mode=False,
```

#Para mejorar la calidad, latencias y precisión en detección de rostros a tiempo real

```
max_num_faces=1, #Máximo 1 cara detectada
min_detection_confidence=0.75) as face_mesh: #Precisión de detección
```

#Modificación de los parámetros de la detección de rostros

```
with Detector.FaceDetection(
    min_detection_confidence = 0.75, model_selection = 0) as rostros:
```

#Model selection para elegir en que rango enfoca mejor las caras, en este caso el 0 = cercano y según se

#aproxime a 1 = lejano

#Creamos el bucle infinito

```
while True:
```

#Si no detecta imagen sale del bucle automáticamente

```
ret, frame = videocap.read()
if ret == False:
    break
```

#Ajuste de la imagen

```
frame = cv2.flip(frame, 1)
frameRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

#conversión de color BGR a RGB para ambos algoritmos



```
resultados = face_mesh.process(frameRGB)
resultado = rostros.process(frameRGB)

#Creación de listas de almacenaje de datos

px = []
py = []
lista = []

listacentro = []
listarostro = []

#Detección de la inclinación del rostro

if resultados.multi_face_landmarks is not None:

    #Ajuste de los parámetros de visualización de en pantalla de la malla

    for face_landmarks in resultados.multi_face_landmarks:
        Dibujo.draw_landmarks(frame, face_landmarks,
            Malla_Facial.FACEMESH_CONTOURS,
            Dibujo.DrawingSpec(color=(255, 0, 0), thickness=1, circle_radius=1),
            Dibujo.DrawingSpec(color=(0, 255, 0), thickness=1))

    #Bucle para almacenar todos los puntos en la lista de la malla

    for id, puntos in enumerate(face_landmarks.landmark):

        al, an, c = frame.shape
        x, y = int(puntos.x * an), int(puntos.y * al)

        px.append(x)
        py.append(y)
        lista.append([id, x, y])
```



#Declaración de los puntos de interés para reconocer la inclinación de la cara

```
if len(lista) == 468:
```

```
    #Puntos izquierda
```

```
    x1, y1 = lista[127][1:]
```

```
    x2, y2 = lista[234][1:]
```

```
    x3, y3 = lista[93][1:]
```

```
    x4, y4 = lista[132][1:]
```

```
    #Puntos derecha
```

```
    x5, y5 = lista[356][1:]
```

```
    x6, y6 = lista[454][1:]
```

```
    x7, y7 = lista[323][1:]
```

```
    x8, y8 = lista[361][1:]
```

```
    #Puntos centrales
```

```
    x9, y9 = lista[6][1:]
```

```
    x10, y10 = lista[197][1:]
```

```
    x11, y11 = lista[195][1:]
```

```
    x12, y12 = lista[5][1:]
```

```
    #Cálculo de las longitudes de los puntos de izquierda respecto al centro
```

```
    cx, cy = (x1 + x9) // 2, (y1 + y9) // 2  
    longitud = math.hypot(x9 - x1, y9 - y1)
```

```
    cx2, cy2 = (x2 + x10) // 2, (y2 + y10) // 2  
    longitud2 = math.hypot(x10 - x2, y10 - y2)
```

```
    cx3, cy3 = (x3 + x11) // 2, (y3 + y11) // 2  
    longitud3 = math.hypot(x11 - x3, y11 - y3)
```

```
    cx4, cy4 = (x4 + x12) // 2, (y4 + y12) // 2  
    longitud4 = math.hypot(x12 - x4, y12 - y4)
```



#Cálculo de las longitudes de los puntos de la derecha respecto al centro

```
cx5, cy5 = (x5 + x9) // 2, (y5 + y9) // 2  
longitud5 = math.hypot(x9 - x5, y9 - y5)
```

```
cx6, cy6 = (x6 + x10) // 2, (y6 + y10) // 2  
longitud6 = math.hypot(x10 - x6, y10 - y6)
```

```
cx7, cy7 = (x7 + x11) // 2, (y7 + y11) // 2  
longitud7 = math.hypot(x11 - x7, y11 - y7)
```

```
cx8, cy8 = (x8 + x12) // 2, (y8 + y12) // 2  
longitud8 = math.hypot(x12 - x8, y12 - y8)
```

#Condiciones para determinar la inclinación

```
if longitud > (longitud5 + 60) and longitud2 > (longitud6 + 60) and longitud3 > (longitud7 +  
60) and longitud4 > (longitud8 + 60):  
    print("inclinadod")  
    com.write(inclinadod)
```

```
if longitud < (longitud5 - 60) and longitud2 < (longitud6 - 60) and longitud3 < (longitud7 - 60)  
and longitud4 < (longitud8 - 60):  
    print("inclinadoi")  
    com.write(inclinadoi)
```

```
if ((longitud5 - 59) < longitud < (longitud5 + 59)) and ((longitud6 - 59) < longitud2 <  
(longitud6 + 59)) and ((longitud7 - 59) < longitud3 < (longitud7 + 59)) and ((longitud8 - 59) <  
longitud4 < (longitud8 + 59)):  
    print("recto")  
    com.write(recto)
```

#Detección de la posición del rostro

```
if resultado.detections is not None:  
    for rostro in resultado.detections:
```

```
        #Bucle para almacenar todos los datos en la lista del rostro
```



```
for id2, puntos2 in enumerate(resultado.detections):

#Extracción de los valores de ancho y alto respecto al tamaño de la imagen

al, an, c = frame.shape

#Valor del centro de la imagen

centroancho = int(an / 2)
centroalto = int(al/2)

#Extracción de las coordenadas X e Y iniciales del rectangulo generado en la cara

x = puntos2.location_data.relative_bounding_box.xmin
y = puntos2.location_data.relative_bounding_box.ymin

#Pasamos x e y a píxeles

x, y = int(x * an), int(y * al)

#Extracción el ancho y el alto en PÍXELES

ancho = puntos2.location_data.relative_bounding_box.width
alto = puntos2.location_data.relative_bounding_box.height
px1, py1 = int(ancho * an), int(alto * al)

#Extracción el punto central del rectangulo cara

centrox = x + (px1//2)
centroy = y + (py1//2)

#Almacenaje de los datos en las listas

listacentro.append([id, centrox, centroy])
listacentro.append([x, y, px1, px1])

#Mostrar un punto en el centro de la cara

cv2.circle(frame, (centrox, centroy), 3, (0, 0, 255), cv2.FILLED)
cv2.line(frame, (centrox, 0), (centrox, 720), (0, 0, 255), 2)
cv2.line(frame, (0, centroy), (1280, centroy), (0, 0, 255), 2)
```



#Condiciones para detectar la posición en el eje X de la cara

```
if centrox < centroancho - 50:
```

```
    print("Izquierda")
```

```
    com.write(izquierda)
```

```
elif centrox > centroancho + 50:
```

```
    print("Derecha")
```

```
    com.write(derecha)
```

```
elif centroancho - 49 < centrox < centroancho + 49:
```

```
    print("ParadaX")
```

```
    com.write(parox)
```

#Condiciones para detectar la posición en el eje Y de la cara

```
if centroy < centroalto - 30:
```

```
    print("Abajo")
```

```
    com.write(abajo)
```

```
if centroy > centroalto + 30:
```

```
    print("Arriba")
```

```
    com.write(arriba)
```

```
if centroalto - 29 < centroy < centroalto + 29:
```

```
    print("ParadaY")
```

```
    com.write(paroy)
```

#Mostrar la imagen

```
cv2.imshow("Frame", frame)
```

#Salir del bucle presionando Esc

```
k = cv2.waitKey(1) & 0xFF
```

```
if k == 27:
```

```
    break
```

#Cierre del video y de las ventanas

```
com.close()
```

```
videocap.release()
```

```
cv2.destroyAllWindows()
```

En cualquier caso, puesto que la interpretación del código fuera de su entorno de programación es bastante más compleja. A continuación, se adjunta un enlace a un editor de código basado en Python online, en él se encuentra todo el código usado con la estructura requerida (puesto que en Python no se requieren de llaves delimitadoras y es imprescindible la estructura de tabulaciones).

<https://www.online-python.com/b96FGxdYPA>

## 1.2. Código Arduino

A continuación, se presenta el código creado en el entorno de programación de Arduino para que el Arduino Mega 2650 procese toda la información y controle de manera adecuada todos los actuadores. En él se incluyen comentarios para mostrar de una forma más clara la funcionalidad o el objetivo de ciertas partes del código.

```
//Librerías a usar

#include <SoftwareSerial.h>
#include <Servo.h>

Servo myServo;      //Definición de la variable del servo

char dato0;        //Definición de variables de lectura
char dato1;
char dato2;
char dato3;

int aux = 0;

int p = 90;        //posición por defecto del servo

void setup()
{
  Serial.begin(9600);    //Inicialización de variables y configuración de puertos
  Serial.setTimeout(10);

  Serial3.begin(9600);
```





```
myServo.attach(4);
```

```
pinMode(22, OUTPUT);
```

```
pinMode(23, OUTPUT);
```

```
pinMode(26, OUTPUT);
```

```
pinMode(27, OUTPUT);
```

```
pinMode(47, INPUT);
```

```
pinMode(49, INPUT);
```

```
pinMode(51, INPUT);
```

```
pinMode(53, INPUT);
```

```
//CALIBRACIÓN DE LA POSICIÓN DE LOS MOTORES AL INICIAR
```

```
myServo.write(90);          //posición 90° del servo
```

```
do                          //Gira hasta que detecte la señal que manda el final de carrera
```

```
{
```

```
  digitalWrite(26, HIGH);
```

```
  for(int i = 0; i < 1;i++)
```

```
  {
```

```
    digitalWrite(27, HIGH);
```

```
    delayMicroseconds(500);
```

```
    digitalWrite(27, LOW);
```

```
    delayMicroseconds(500);
```

```
    delay(20);
```

```
  }
```

```
}
```

```
while(digitalRead(47) == HIGH);
```

```
digitalWrite(26, LOW);
```



```
for(int i = 0; i < 100;i++)
{
    digitalWrite(27, HIGH);
    delayMicroseconds(500);
    digitalWrite(27, LOW);
    delayMicroseconds(500);
    delay(10);
}

do //Gira hasta que detectes la señal que manda el final de carrera
{
    digitalWrite(22, HIGH);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
        delay(20);
    }
}
while(digitalRead(51) == HIGH);
digitalWrite(22, LOW);
for(int i = 0; i < 100;i++)
{
    digitalWrite(23, HIGH);
    delayMicroseconds(500);
    digitalWrite(23, LOW);
    delayMicroseconds(500);
```



```
        delay(10);
    }
}

void loop()
{
    if(aux == 0)
    {
        while(Serial.available() > 0 && aux == 0)    //Siempre que se envíen datos por el
        canal serie y estemos en modo automatico
        {
            if(Serial3.read() == 'A')                //Condición de modo manual
            {
                aux = 1;
            }

            if(Serial.available() % 3 == 0)          //Lectura de datos si hay 3 variables o
            múltiplo
            {
                //ACCIONES ANTE LOS DATOS

                dato1 = Serial.read();                //Lectura de los datos y switch para
                elegir el movimiento adecuado

                if(dato3 == ('h' || 'e'))
                {
                    dato1 = 'p';
                }

                switch(dato1)
                {
                    case 'h':                          //INCLINADO I
```



```
myServo.write(p);
if(p>176)
{
  p=180;
}
else
{
  p=p+4;

  digitalWrite(22, LOW);
  if(digitalRead(53) == LOW)
  {
    delay(100);
  }
  else
  {
    for(int i = 0; i < 1;i++)
    {
      digitalWrite(23, HIGH);
      delayMicroseconds(500);
      digitalWrite(23, LOW);
      delayMicroseconds(500);
    }
  }
}
break;
case 'e':          //INCLINADO D
myServo.write(p);
if(p<4)
```



```
{
  p=0;
}
else
{
  p=p-4;
  if(digitalRead(51) == LOW)
  {
    delay(100);
  }
  else
  {
    digitalWrite(22, HIGH);
    for(int i = 0; i < 1;i++)
    {
      digitalWrite(23, HIGH);
      delayMicroseconds(500);
      digitalWrite(23, LOW);
      delayMicroseconds(500);
    }
  }
}
break;
case'r':          //RECTO
  myServo.write(p);
  break;

case 'i':          //IZQUIERDA
  if(digitalRead(53) == LOW)
```



```
{
    delay(100);
}
else
{
    digitalWrite(22, LOW);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'd':          //DERECHA
    if(digitalRead(51) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(22, HIGH);
        for(int i = 0; i < 1;i++)
        {
            digitalWrite(23, HIGH);
            delayMicroseconds(500);
            digitalWrite(23, LOW);
            delayMicroseconds(500);
        }
    }
}
```



```
    }  
  }  
  break;  
case 'p':          //PARO X  
  
  break;  
  
case 'w':          //ARRIBA  
  if(digitalRead(49) == LOW)  
  {  
    delay(100);  
  }  
  else  
  {  
    digitalWrite(26, LOW);  
    for(int i = 0; i < 1;i++)  
    {  
      digitalWrite(27, HIGH);  
      delayMicroseconds(500);  
      digitalWrite(27, LOW);  
      delayMicroseconds(500);  
    }  
  }  
  break;  
case 'j':          //ABAJO  
  if(digitalRead(47) == LOW)  
  {  
    delay(100);  
  }  
}
```



```
else
{
digitalWrite(26, HIGH);
for(int i = 0; i < 1;i++)
{
digitalWrite(27, HIGH);
delayMicroseconds(500);
digitalWrite(27, LOW);
delayMicroseconds(500);
}
}
break;
case 'l':          //PARO Y
break;
}

dato2 = Serial.read();
if(dato1 == ('h' || 'e'))
{
dato2 = 'p';
}
switch(dato2)
{
case 'h':          //INCLINADO I
myservo.write(p);
if(p>176)
{
p=180;
}
}
```





```
else
{
  p=p+4;
  digitalWrite(22, LOW);
  if(digitalRead(53) == LOW)
  {
    delay(100);
  }
  else
  {
    for(int i = 0; i < 1;i++)
    {
      digitalWrite(23, HIGH);
      delayMicroseconds(500);
      digitalWrite(23, LOW);
      delayMicroseconds(500);
    }
  }
}
break;
case 'e': //INCLINADO D
  myServo.write(p);
  if(p<4)
  {
    p=0;
  }
  else
  {
    p=p-4;
```



```
if(digitalRead(51) == LOW)
{
    delay(100);
}
else
{
    digitalWrite(22, HIGH);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
    }
}
break;
case'r':          //RECTO
    myServo.write(p);
    break;

case 'i':        //IZQUIERDA
    if(digitalRead(53) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(22, LOW);
```



```
for(int i = 0; i < 1;i++)
{
    digitalWrite(23, HIGH);
    delayMicroseconds(500);
    digitalWrite(23, LOW);
    delayMicroseconds(500);
}
}
break;
case 'd':                //DERECHA
if(digitalRead(51) == LOW)
{
    delay(100);
}
else
{
    digitalWrite(22, HIGH);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'p':                //PARO X
break;
```



```
case 'w':                //ARRIBA
    if(digitalRead(49) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(26, LOW);
        for(int i = 0; i < 1;i++)
        {
            digitalWrite(27, HIGH);
            delayMicroseconds(500);
            digitalWrite(27, LOW);
            delayMicroseconds(500);
        }
    }
    break;
case 'j':                //ABAJO
    if(digitalRead(47) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(26, HIGH);
        for(int i = 0; i < 1;i++)
        {
            digitalWrite(27, HIGH);
            delayMicroseconds(500);
```



```
        digitalWrite(27, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'l':           //PARO Y
    break;
}

dato3 = Serial.read();
if(dato2 == ('h' || 'e'))
{
    dato3 = 'p';
}
switch(dato3)
{
    case 'h':       //INCLINADO I
        myServo.write(p);
        if(p>176)
        {
            p=180;
        }
        else
        {
            p=p+4;
            if(digitalRead(53) == LOW)
            {
                delay(100);
            }
        }
    }
}
```



```
else
{
digitalWrite(22, LOW);
for(int i = 0; i < 1;i++)
{
digitalWrite(23, HIGH);
delayMicroseconds(500);
digitalWrite(23, LOW);
delayMicroseconds(500);
}
}
}
break;
case 'e': //INCLINADO D
myServo.write(p);
if(p<4)
{
p=0;
}
else
{
p=p-4;
if(digitalRead(51) == LOW)
{
delay(100);
}
}
else
{
digitalWrite(22, HIGH);
```



```
for(int i = 0; i < 1;i++)
{
    digitalWrite(23, HIGH);
    delayMicroseconds(500);
    digitalWrite(23, LOW);
    delayMicroseconds(500);
}
}
}
break;

case'r':           //RECTO
myServo.write(p);
break;

case 'i':         //IZQUIERDA
if(digitalRead(53) == LOW)
{
    delay(100);
}
else
{
    digitalWrite(22, LOW);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
    }
}
```



```
}  
break;  
case 'd':           //DERECHA  
if(digitalRead(51) == LOW)  
{  
  delay(100);  
}  
else  
{  
  digitalWrite(22, HIGH);  
  for(int i = 0; i < 1;i++)  
  {  
    digitalWrite(23, HIGH);  
    delayMicroseconds(500);  
    digitalWrite(23, LOW);  
    delayMicroseconds(500);  
  }  
}  
break;  
case 'p':           //PARO X  
break;  
  
case 'w':           //ARRIBA  
if(digitalRead(49) == LOW)  
{  
  delay(100);  
}  
else  
{
```





```
digitalWrite(26, LOW);
for(int i = 0; i < 1;i++)
{
    digitalWrite(27, HIGH);
    delayMicroseconds(500);
    digitalWrite(27, LOW);
    delayMicroseconds(500);
}
}
break;
case 'j':          //ABAJO
if(digitalRead(47) == LOW)
{
    delay(100);
}
else
{
    digitalWrite(26, HIGH);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(27, HIGH);
        delayMicroseconds(500);
        digitalWrite(27, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'l':          //PARO Y
break;
```



```
    }  
    for(int i = 0;i<Serial.available();i++)  
    {  
        dato0 = Serial.read();  
    }  
}  
}  
}  
if(aux == 1)          //Entramos en el modo auxiliar y se procede con la lectura  
igual que en el automático  
{  
    dato0 = Serial3.read();  
    switch(dato0)  
    {  
        case 'H':          //INCLINADO I  
            myServo.write(p);  
            if(p<4)  
            {  
                p=0;  
            }  
            else  
            {  
                p=p-4;  
            }  
            break;  
        case 'E':          //INCLINADO D  
            myServo.write(p);  
            if(p>176)  
            {  
                p=180;  
            }  
        }  
    }  
}
```



```
}  
else  
{  
    p=p+4;  
}  
break;  
case'R':                //RECTO  
    myServo.write(p);  
    break;  
  
case 'I':                //IZQUIERDA  
    if(digitalRead(53) == LOW)  
    {  
        delay(100);  
    }  
    else  
    {  
        digitalWrite(22, LOW);  
        for(int i = 0; i < 1;i++)  
        {  
            digitalWrite(23, HIGH);  
            delayMicroseconds(500);  
            digitalWrite(23, LOW);  
            delayMicroseconds(500);  
        }  
    }  
    break;  
case 'D':                //DERECHA  
    if(digitalRead(51) == LOW)
```



```
{
    delay(100);
}
else
{
    digitalWrite(22, HIGH);
    for(int i = 0; i < 1;i++)
    {
        digitalWrite(23, HIGH);
        delayMicroseconds(500);
        digitalWrite(23, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'P':          //PARO X
    break;

case 'W':          //ARRIBA
    if(digitalRead(49) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(26, LOW);
        for(int i = 0; i < 1;i++)
        {
            digitalWrite(27, HIGH);
```



```
        delayMicroseconds(500);
        digitalWrite(27, LOW);
        delayMicroseconds(500);
    }
}
break;
case 'J':                //ABAJO
    if(digitalRead(47) == LOW)
    {
        delay(100);
    }
    else
    {
        digitalWrite(26, HIGH);
        for(int i = 0; i < 1;i++)
        {
            digitalWrite(27, HIGH);
            delayMicroseconds(500);
            digitalWrite(27, LOW);
            delayMicroseconds(500);
        }
    }
    break;
case 'L':                //PARO Y
    break;
```



```
case 'B':           //MODO AUTOMÁTICO
    aux = 0;
    break;
}
}
}
```

De igual forma que en apartado anterior, para la correcta lectura y comprensión del código es necesario tenerlo correctamente estructurado, es por ello que a continuación se adjunta un enlace a un editor online de Arduino en el que se encuentra todo el código.

<https://create.arduino.cc/editor/alexciuvi6/7a6e2d5b-941c-4f7d-ba6b-f1a27fe8cce7/preview>

## 1.3. MitAppInventor

A continuación, se exponen los bloques de código usados a lo largo de la creación de la aplicación móvil que se conecta por bluetooth al sistema. Además, se muestra en más profundidad el entorno de trabajo.

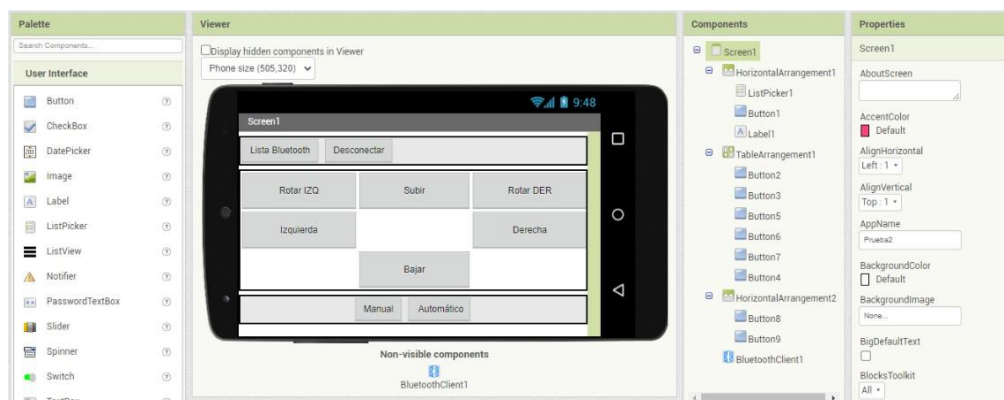


Imagen 27. Interfaz gráfica de la aplicación móvil.

En la imagen anterior se puede observar el diseño final de la interfaz gráfica de la aplicación móvil. En ella se disponen distintos botones orientados al flujo de transmisión para realizar los movimientos, cambiar de modo automático al manual y conectar con los dispositivos vía bluetooth.

Por otro lado, si nos dirigimos al menú de los bloques observaremos aquellas funciones implementadas para realizar la transmisión de datos y modificación de la interfaz gráfica. A continuación, se explica cada uno de los bloques y su utilidad.

La estructura de los bloques se divide en dos partes claramente definidas, en la primera se habilitará la conexión bluetooth así como se notificará en pantalla del estado de esta. Por otro lado, la segunda parte nos permitirá realizar el control de los movimientos de la estructura, en esta parte se enviará todos los datos al módulo bluetooth para que posteriormente el Arduino sea capaz de leerlos.

```
when Screen1.Initialize
do set Label1.Text to "Sin Conexion"

when ListPicker1.BeforePicking
do set ListPicker1.Elements to BluetoothClient1.AddressesAndNames

when ListPicker1.AfterPicking
do evaluate but ignore result call BluetoothClient1.Connect
address ListPicker1.Selection
if BluetoothClient1.IsConnected
then set Label1.Text to "Conectado"
else set Label1.Text to "Error al conectar"

when Button1.Click
do call BluetoothClient1.Disconnect
set Label1.Text to "Sin Conexion"
```

Imagen 28. Bloque de código de conexión.

Como se puede observar, en este tramo de código se inicializarán todos los labels y para informar del estado de la conexión, también usaremos un selector para conectarnos al dispositivo bluetooth de nuestra elección, para la desconexión usaremos un botón implementado en la interfaz. Destacar que, en caso de producirse una conexión fallida, la aplicación nos lo reportará con un mensaje en pantalla.

```
when Button2.Click
do call BluetoothClient1.SendText
text "H"

when Button3.Click
do call BluetoothClient1.SendText
text "W"

when Button4.Click
do call BluetoothClient1.SendText
text "E"

when Button5.Click
do call BluetoothClient1.SendText
text "I"

when Button6.Click
do call BluetoothClient1.SendText
text "D"

when Button7.Click
do call BluetoothClient1.SendText
text "J"

when Button8.Click
do call BluetoothClient1.SendText
text "A"

when Button9.Click
do call BluetoothClient1.SendText
text "B"
```

Imagen 29. Bloque de código de control.

En esta segunda parte dispondremos de los bloques que nos permitirán enviar la información, así como elegir el tipo de control que queremos. Su funcionamiento es muy sencillo y se basa en la pulsación de los botones para realizar el control, cada uno de los botones enviará un carácter asociado a un movimiento. Como se puede apreciar, en este caso se han dispuesto los bloques con la configuración de la interfaz gráfica para facilitar la comprensión o modificación posterior de estos. Mencionar que también contaremos con la transmisión, de igual manera, de los datos para activar el modo manual o el automático.



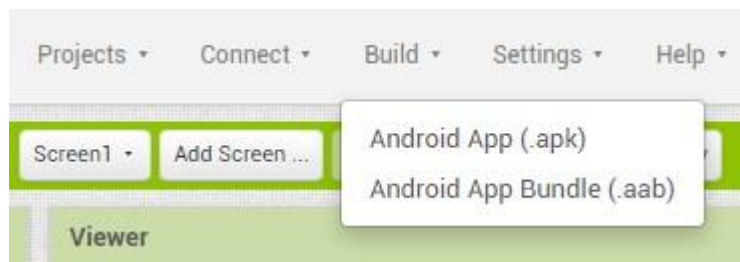


Imagen 30. Creación de la aplicación.

Por último, se va a mencionar el proceso para la creación y descarga de esta aplicación. Desde el mismo software online, nos ubicaremos en la parte superior del entorno y elegiremos la opción de “build” como una aplicación (extensión “.apk”), a partir de ahí, se generará un código QR que una vez escaneado con la app del software (ubicada en las tiendas de aplicaciones móviles) nos descargará directamente la aplicación en nuestro dispositivo.



Imagen 31. Aplicación creada en el dispositivo móvil.

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

## **Anejo N. °2: DATASHEETS**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## ÍNDICE

2.	Datasheets.....	75
2.1.	Fuente de alimentación.....	75
2.2.	Stepper NEMA 17 .....	77
2.3.	Driver A-4988 .....	78
2.4.	Servo SG90 .....	81
2.5.	Arduino Mega 2560.....	81

## 2. Datasheets

En este anejo se adjuntan los datasheets de los componentes específicos que conforman todo el sistema.

### 2.1. Fuente de alimentación

<https://octopart.com/datasheet/mdr-60-12-mean+well-3991779>



60W Single Output Industrial DIN Rail Power Supply

**MDR-60** series



■ Features :

- Universal AC input/Full range
- Protections: Short circuit / Overload / Over voltage
- Cooling by free air convection
- Can be installed on DIN rail TS-35/7.5 or 15
- NEC class 2 / LPS compliant (24V,48V only)
- LED indicator for power on
- DC OK relay contact
- No load power consumption<0.75W
- 100% full load burn-in test
- 3 years warranty



**SPECIFICATION**

MODEL	MDR-60-5	MDR-60-12	MDR-60-24	MDR-60-48	
OUTPUT	DC VOLTAGE	5V	12V	24V	48V
	RATED CURRENT	10A	5A	2.5A	1.25A
	CURRENT RANGE	0 ~ 10A	0 ~ 5A	0 ~ 2.5A	0 ~ 1.25A
	RATED POWER	50W	60W	60W	60W
	RIPPLE & NOISE (max.) Note.2	80mVp-p	120mVp-p	150mVp-p	200mVp-p
	VOLTAGE ADJ. RANGE	5 ~ 6V	12 ~ 15V	24 ~ 30V	48 ~ 56V
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±1.0%	±1.0%	±1.0%	±1.0%
	LOAD REGULATION	±1.5%	±1.0%	±1.0%	±1.0%
	SETUP, RISE TIME Note.5	500ms, 30ms/230VAC	500ms, 30ms/115VAC at full load		
HOLD UP TIME (Typ.)	50ms/230VAC	20ms/115VAC at full load			
INPUT	VOLTAGE RANGE	85 ~ 264VAC	120 ~ 370VDC		
	FREQUENCY RANGE	47 ~ 63Hz			
	EFFICIENCY (Typ.)	78%	86%	88%	87%
	AC CURRENT (Typ.)	1.8A/115VAC	1A/230VAC		
	INRUSH CURRENT (Typ.)	COLD START 30A/115VAC	60A/230VAC		
LEAKAGE CURRENT	<1mA/240VAC				
PROTECTION	OVERLOAD	105 ~ 150% rated output power Protection type : Constant current limiting, recovers automatically after fault condition is removed			
	OVER VOLTAGE	6.25 ~ 7.25V	15.6 ~ 18V	31.2 ~ 36V	57.6 ~ 64.8V
		Protection type : Shut down o/p voltage, re-power on to recover			
FUNCTION	DC OK SIGNAL	Relay contact rating(max.): 30V/1A resistive			
ENVIRONMENT	WORKING TEMP.	-20 ~ +70°C (Refer to "Derating Curve")			
	WORKING HUMIDITY	20 ~ 90% RH non-condensing			
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH			
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)			
	VIBRATION	Component : 10 ~ 500Hz, 2G 10min./1cycle, period for 60min. each along X, Y, Z axes ; Mounting : Compliance to IEC60068-2-6			
SAFETY & EMC (Note 4)	SAFETY STANDARDS	UL508, UL60950-1, TUV EN60950-1 approved, NEC class 2 / LPS compliant (24V,48V only)			
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC I/P-FG:1.5KVAC O/P-FG:0.5KVAC			
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:>100M Ohms / 500VDC / 25°C / 70% RH			
	EMC EMISSION	Compliance to EN55011, EN55022 (CISPR22), EN61204-3 Class B, EN61000-3-2, -3			
	EMC IMMUNITY	Compliance to EN61000-4-2, 3, 4, 5, 6, 8, 11, EN55024, EN61000-6-2, EN61204-3, heavy industry level, criteria A			
OTHERS	MTBF	299.2K hrs min. MIL-HDBK-217F (25°C)			
	DIMENSION	40*90*100mm (W*H*D)			
	PACKING	0.33Kg; 42pcs/14.8Kg/0.82CUFT			
NOTE	<p>1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature.</p> <p>2. Ripple &amp; noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uF &amp; 47uF parallel capacitor.</p> <p>3. Tolerance : includes set up tolerance, line regulation and load regulation.</p> <p>4. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives.</p> <p>5. Length of set up time is measured at first cold start. Turning ON/OFF the power supply may lead to increase of the set up time.</p>				



60W Single Output Industrial DIN Rail Power Supply

**MDR-60 series**

**Mechanical Specification** Case No. 962A Unit:mm

Install DIN rail TS35/7.5 or TS35/15

**Block Diagram**

**DC OK Relay Contact**

Contact Close	PSU turns on / DC OK.
Contact Open	PSU turns off / DC Fail.
Contact Ratings (max.)	30V/1A resistive load.

**Derating Curve**

**Output Derating VS Input Voltage**



## 2.3. Driver A-4988

<https://pdf1.alldatasheet.com/datasheet-pdf/view/338780/ALLEGRO/A4988.html>

### A4988

### DMOS Microstepping Driver with Translator and Overcurrent Protection

**ELECTRICAL CHARACTERISTICS<sup>1</sup>** at  $T_A = 25^\circ\text{C}$ ,  $V_{BB} = 35\text{ V}$  (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. <sup>2</sup>	Max.	Units
<b>Output Drivers</b>						
Load Supply Voltage Range	$V_{BB}$	Operating	8	–	35	V
Logic Supply Voltage Range	$V_{DD}$	Operating	3.0	–	5.5	V
Output On Resistance	$R_{DSON}$	Source Driver, $I_{OUT} = -1.5\text{ A}$	–	320	430	m $\Omega$
		Sink Driver, $I_{OUT} = 1.5\text{ A}$	–	320	430	m $\Omega$
Body Diode Forward Voltage	$V_F$	Source Diode, $I_F = -1.5\text{ A}$	–	–	1.2	V
		Sink Diode, $I_F = 1.5\text{ A}$	–	–	1.2	V
Motor Supply Current	$I_{BB}$	$f_{PWM} < 50\text{ kHz}$	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	$I_{DD}$	$f_{PWM} < 50\text{ kHz}$	–	–	8	mA
		Outputs off	–	–	5	mA
<b>Control Logic</b>						
Logic Input Voltage	$V_{IN(1)}$		$V_{DD} \times 0.7$	–	–	V
	$V_{IN(0)}$		–	–	$V_{DD} \times 0.3$	V
Logic Input Current	$I_{IN(1)}$	$V_{IN} = V_{DD} \times 0.7$	–20	<1.0	20	$\mu\text{A}$
	$I_{IN(0)}$	$V_{IN} = V_{DD} \times 0.3$	–20	<1.0	20	$\mu\text{A}$
Microstep Select	$R_{MS1}$	MS1 pin	–	100	–	k $\Omega$
	$R_{MS2}$	MS2 pin	–	50	–	k $\Omega$
	$R_{MS3}$	MS3 pin	–	100	–	k $\Omega$
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of $V_{DD}$	5	11	19	%
Blank Time	$t_{BLANK}$		0.7	1	1.3	$\mu\text{s}$
Fixed Off-Time	$t_{OFF}$	OSC = VDD or GND	20	30	40	$\mu\text{s}$
		$R_{OSC} = 25\text{ k}\Omega$	23	30	37	$\mu\text{s}$
Reference Input Voltage Range	$V_{REF}$		0	–	4	V
Reference Input Current	$I_{REF}$		–3	0	3	$\mu\text{A}$
Current Trip-Level Error <sup>3</sup>	$err_1$	$V_{REF} = 2\text{ V}$ , $\%I_{TRIPMAX} = 38.27\%$	–	–	$\pm 15$	%
		$V_{REF} = 2\text{ V}$ , $\%I_{TRIPMAX} = 70.71\%$	–	–	$\pm 5$	%
		$V_{REF} = 2\text{ V}$ , $\%I_{TRIPMAX} = 100.00\%$	–	–	$\pm 5$	%
Crossover Dead Time	$t_{DT}$		100	475	800	ns
<b>Protection</b>						
Overcurrent Protection Threshold	$I_{OCPST}$		2.1	–	–	A
Thermal Shutdown Temperature	$T_{TSD}$		–	165	–	$^\circ\text{C}$
Thermal Shutdown Hysteresis	$T_{TSDHYS}$		–	15	–	$^\circ\text{C}$
VDD Undervoltage Lockout	$V_{DDUVLO}$	$V_{DD}$ rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

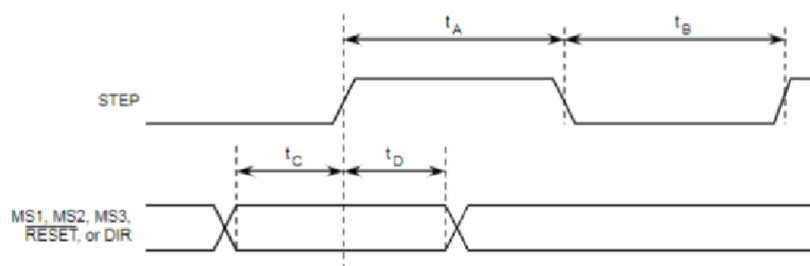
<sup>1</sup>For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

<sup>2</sup>Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

<sup>3</sup> $V_{ERR} = [(V_{REF}/8) - V_{SENSE}] / (V_{REF}/8)$ .

## A4988

### DMOS Microstepping Driver with Translator and Overcurrent Protection



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	$t_A$	1	$\mu\text{s}$
STEP minimum, LOW pulse width	$t_B$	1	$\mu\text{s}$
Setup time, input change to STEP	$t_C$	200	ns
Hold time, input change to STEP	$t_D$	200	ns

Figure 1. Logic Interface Timing Diagram

Table 1. Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase







## 2.4.Servo SG90

<https://datasheetspdf.com/pdf-file/791970/TowerPro/SG90/1>

## 2.5.Arduino Mega 2560

<https://store.arduino.cc/products/arduino-mega-2560-rev3>

\*Destacar que en el caso del Arduino el enlace redirige a la web del fabricante, una vez ahí se encuentran de manera muy accesible documentos más allá del datasheet, entre estos se encuentra el diseño de la PCB u otros de gran relevancia.

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

## **Anejo N. °3: PLAN DE MANTENIMIENTO**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

### 3. Plan de mantenimiento

El mantenimiento del sistema consistirá en distintas acciones a modo de mantenimiento preventivo repartidas a lo largo de los distintos componentes electrónicos, elementos estructurales y a nivel de programación. A continuación, se explica cada acción de forma detallada.

En cuanto a todas las piezas de metacrilato, hay que limpiarlas con el uso de agua, alcohol y algodón. Para el correcto funcionamiento de las translaciones habrá que eliminar el polvo o aplicar un lubricante sobre estas, en cualquier caso y como último recurso existirá la posibilidad de desmontar la torre de la base aplicando un disolvente en la pared suelta y, a partir de ahí facilitar su limpieza.

Mencionar que contamos con una estructura abierta, especialmente la torre, esto facilita el acceso, limpieza y si hiciera falta desmontar la estructura o alguna pieza.

Otra acción es la calibración del código si se usa en entornos distintos o para otras distancias, sobre todo en relación con el enfoque de la cara, el método es exactamente el mismo que el de prueba-error mencionado en apartados anteriores.

La comprobación visual y mediante equipos de verificación del cableado permitirá detectar fallos en el conexionado por desgaste o mala colocación de los cables.

Para los finales de carrera se usarán unos leds que determinan con luz cuando están detectando pulsación. En función de eso podremos comprobar que su lectura de información es correcta.

Revisión de la integridad física de la estructura para comprobar si tiene daños como fisuras, grietas, curvatura de las piezas o algo que pueda comprometer el funcionamiento.

Revisar los drivers puesto que son componentes muy frágiles y pese al disipador puede pasarles de todo, desde a fundirse como a romperse por esfuerzos mecánicos, en cuyo caso sustituir por uno ya configurado.

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro
------------------------------------



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

## **Anejo N. °4: NORMATIVA**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

El proyecto contará con distintas directivas de aplicación obligada:

**-Compatibilidad Electromagnética (EMC):**

DIRECTIVA 2014/30/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 26 de febrero de 2014 sobre la armonización de las legislaciones de los Estados miembros en materia de compatibilidad electromagnética (DOUE L 96/79), 29/3/2014.

**-Equipos radioeléctricos y equipos terminales de telecomunicación y reconocimiento mutuo de su conformidad (RTTE):**

DIRECTIVA 2014/53/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 16 de abril de 2014 relativa a la armonización de las legislaciones de los Estados miembros sobre la comercialización de equipos radioeléctricos, y por la que se deroga la Directiva 1999/5/CE (DOUE L.153/62), 22/5/2014.

**-Máquinas:**

DIRECTIVA 2006/42/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 17 de mayo de 2006 relativa a las máquinas y por la que se modifica la Directiva 95/16/CE (DOUE L.157/24), 9/6/2006.

**-Restricción de uso de sustancias peligrosas en aparatos eléctricos y electrodomésticos (ROHS):**

DIRECTIVA 2011/65/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 8 de junio de 2011 sobre restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos (DOUE L.174/88), 1/7/2011.

**-Indicación del consumo de energía y otros recursos por parte de los productos relacionados con la energía, mediante el etiquetado y una información normalizada (ErP):**

DIRECTIVA 2010/30/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 19 de mayo de 2010 relativa a la indicación del consumo de energía y otros recursos por parte de los productos relacionados con la energía, mediante el etiquetado y una información normalizada (DOUE L.163/1), 18/6/2010.

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °2: PLANOS**

AUTOR: Ciudad-Real Villaescusa, Alejandro

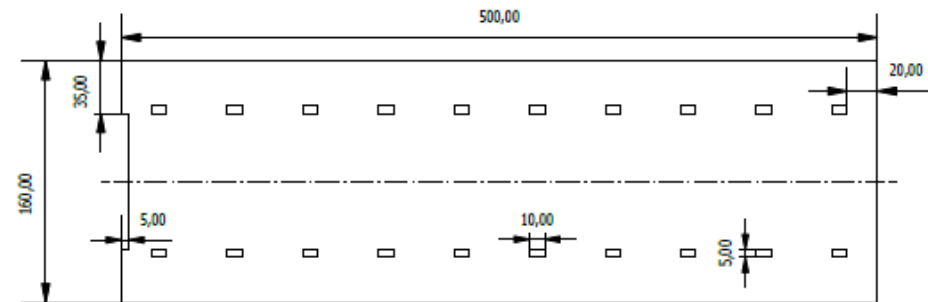
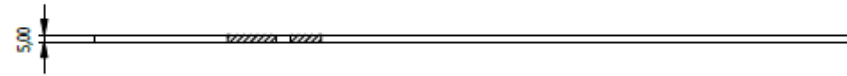
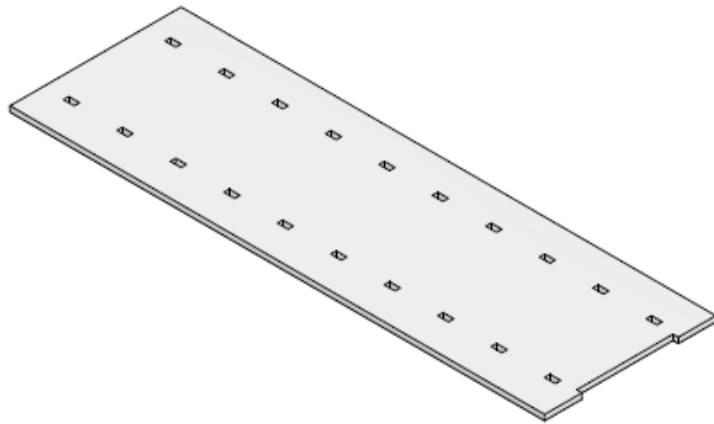
TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## ÍNDICE DE PLANOS

Plano 1. Base guía X.....	88
Plano 2. Pared base .....	89
Plano 3. Soporte guía base .....	90
Plano 4. Top base.....	91
Plano 5. Tapa 1 base.....	92
Plano 6. Tapa 2 base.....	93
Plano 7. Caja base.....	94
Plano 8. Cremallera base.....	95
Plano 9. Base torre... ..	96
Plano 10. Escuadra 20B15H.....	97
Plano 11. Pared torre 2 .....	98
Plano 12. Pared torre 1 .....	99
Plano 13. Tapa torre.....	100
Plano 14. Soporte pared torre.....	101
Plano 15. Soporte motor Y .....	102
Plano 16. Cremallera Y .....	103
Plano 17. Caja servo .....	104
Plano 18. Soporte cámara .....	105
Plano 19. Encaje cámara .....	106
Plano 20. Engranaje 2.....	107
Plano 21. Engranaje 1.....	108
Plano 22. Plataforma torre.....	109
Plano 23. Escuadra 2B2H.....	110
Plano 24. Plataforma .....	111





Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



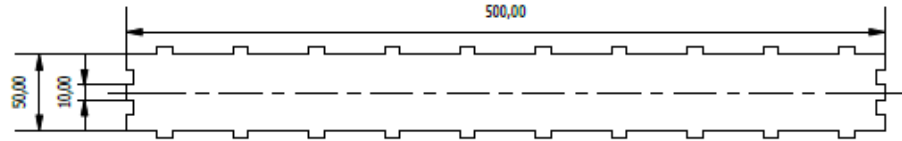
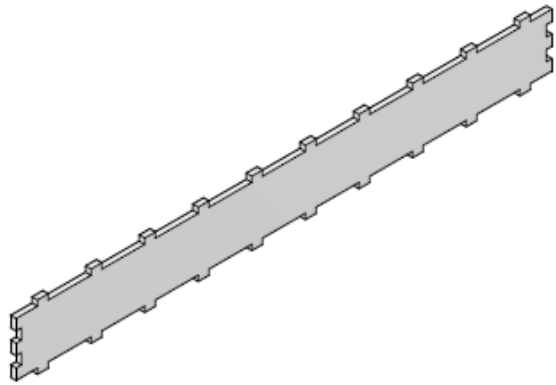
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:5

Nombre Plano: Base Guia X

Nº Plano:  
1



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



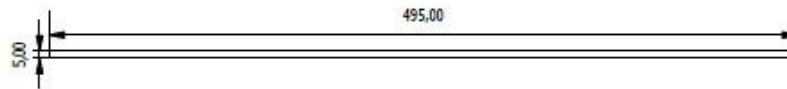
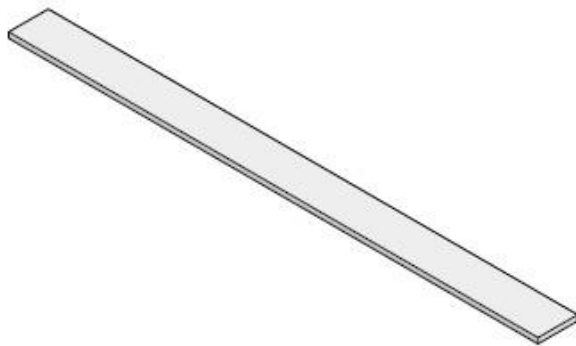
Escuela Técnica Superior de Ingeniería del Diseño



Autor: Ciudad-Real Villaescusa, Alejandro

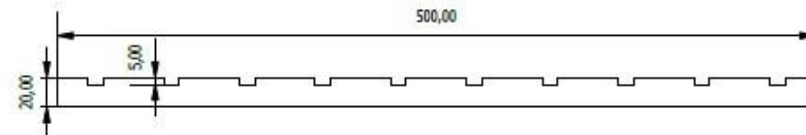
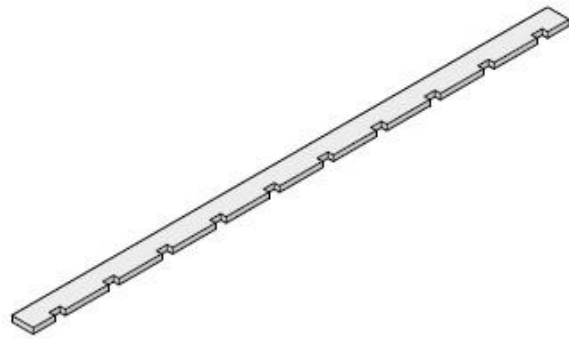
Escala:  
**1:5**

Nombre Plano: Pared Base

Nº Plano:  
**2**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		Fecha: 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	 Escuela Técnica Superior de Ingeniería del Diseño	Escala: <b>1:5</b>
		Nombre Plano: Soporte Guia Base Nº Plano: <b>3</b>



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

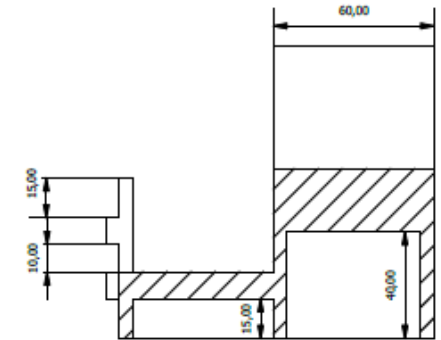
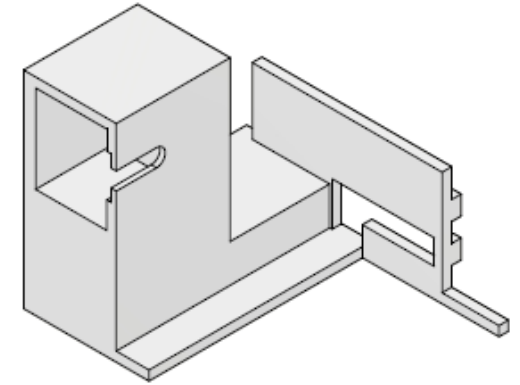
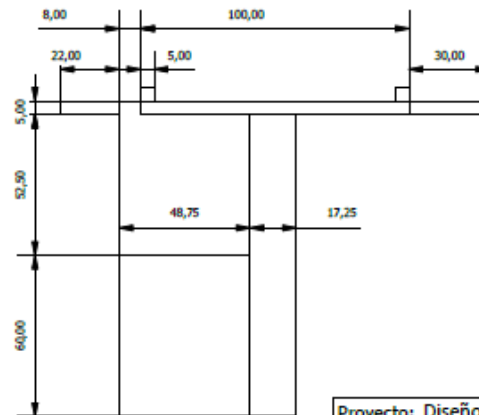
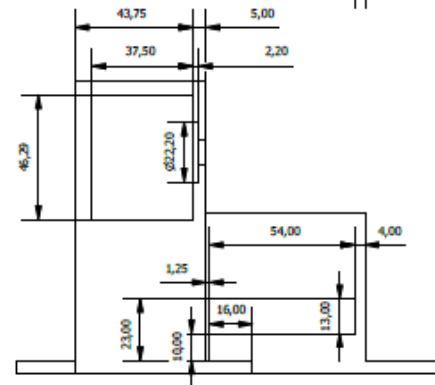
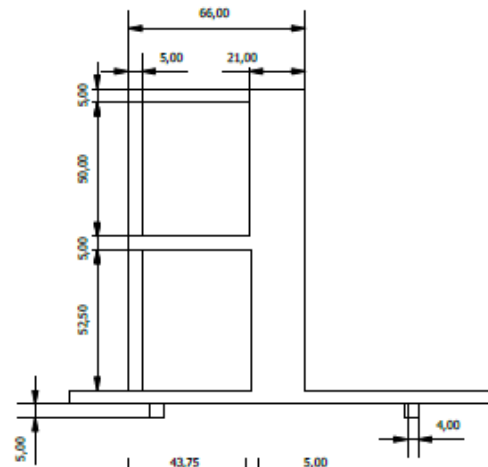
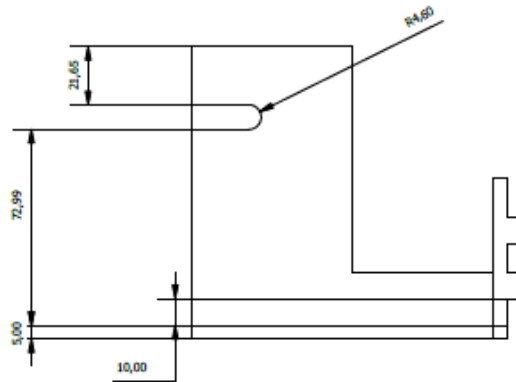
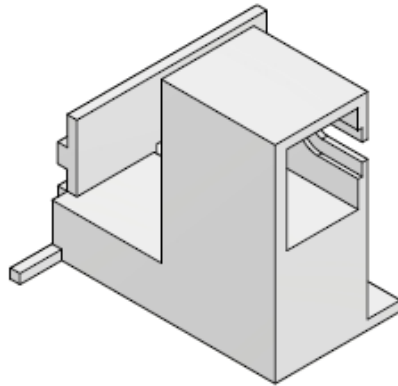
Autor: Ciudad-Real Villaescusa, Alejandro



Escala:  
**1:5**

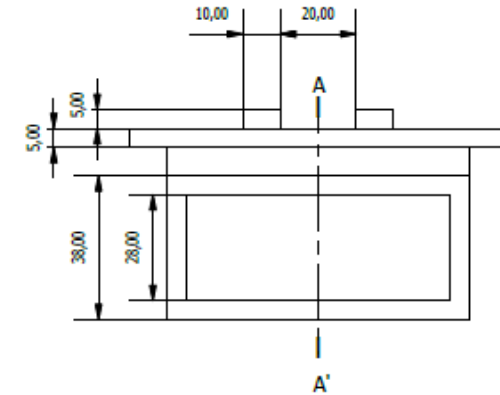
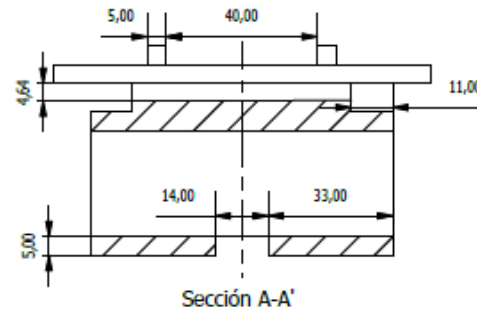
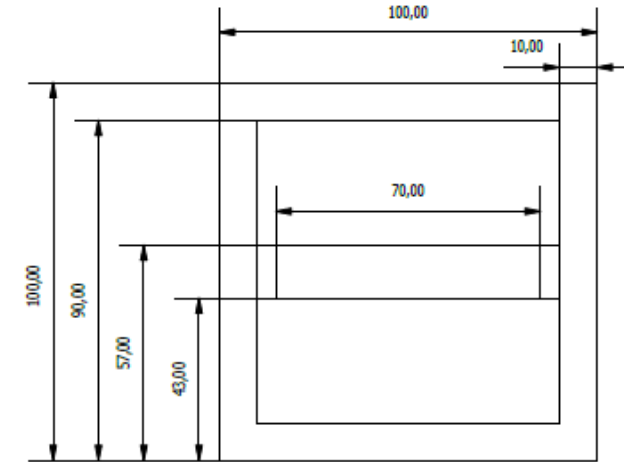
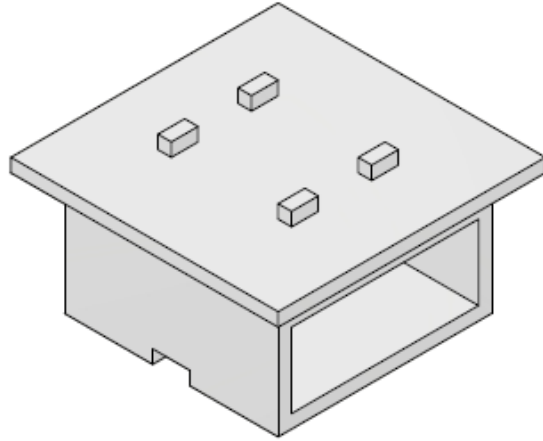
Nombre Plano: Top Base

Nº Plano:  
**4**





Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		Fecha: 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  Escuela Técnica Superior de Ingeniería del Diseño	Autor: Ciudad-Real Villaescusa, Alejandro	Escala: 1:2
	Nombre Plano: Tapa 2 Base	Nº Plano: 6



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



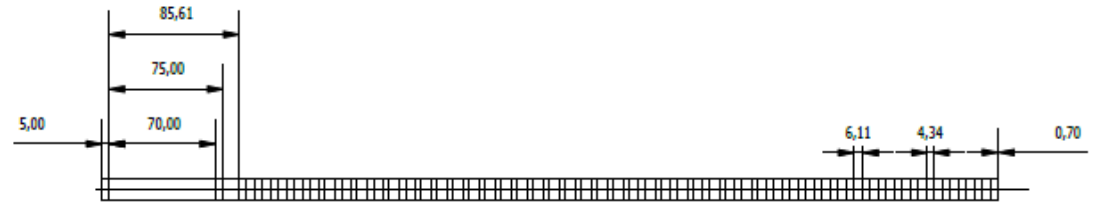
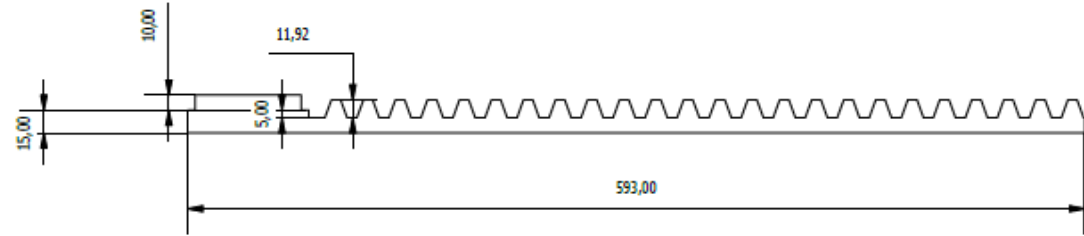
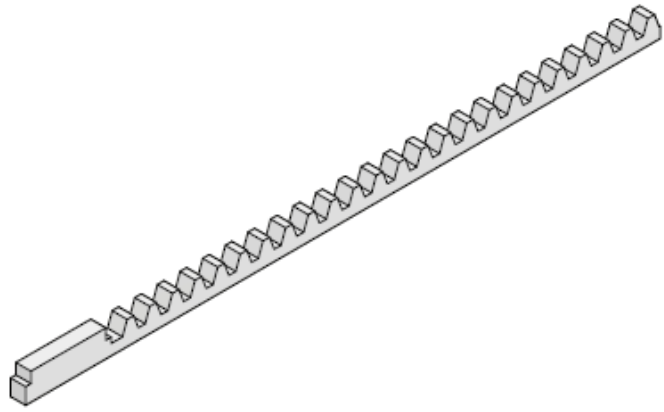
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:2

Nombre Plano: Caja Base

Nº Plano:  
7



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

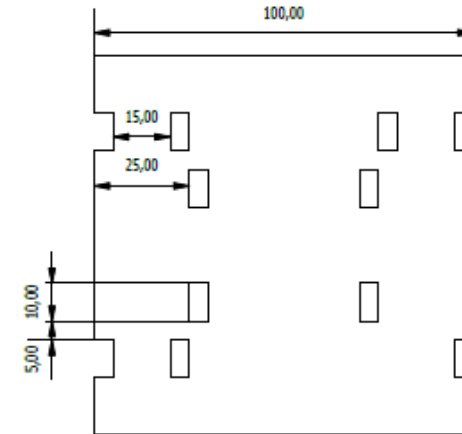
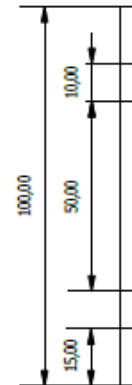
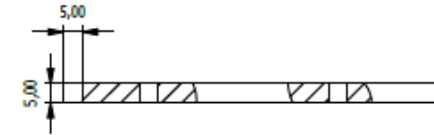
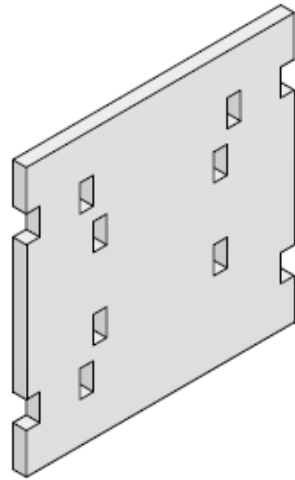
Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1/5

Nombre Plano: Cremallera Base

Nº Plano:  
8





Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



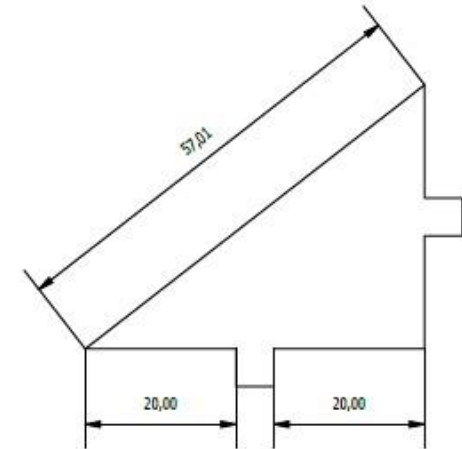
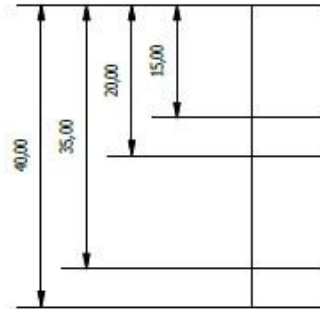
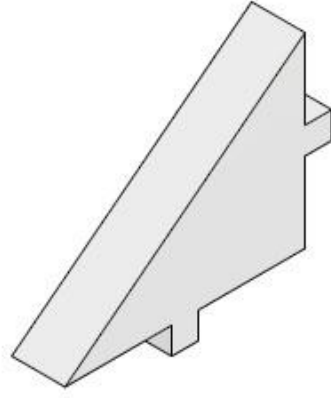
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:2

Nombre Plano: Base Torre

Nº Plano:  
9



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÉCNICA  
DE VALÈNCIA



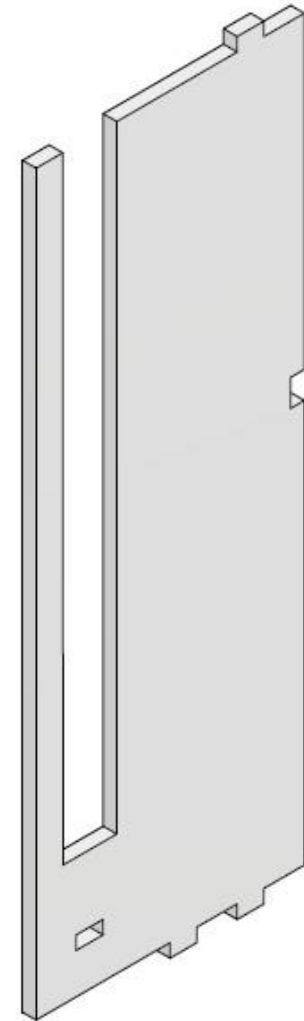
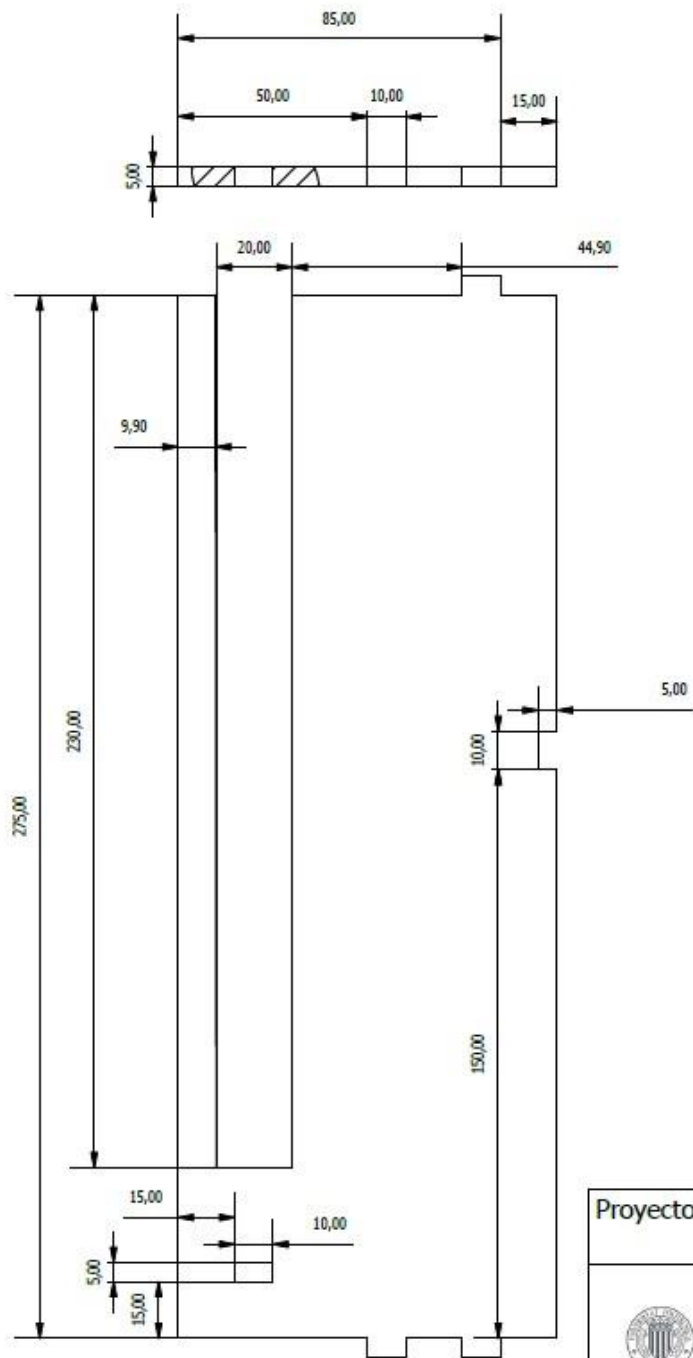
Escuela Técnica Superior de Ingeniería del Diseño



Autor: Ciudad-Real Villaescusa, Alejandro

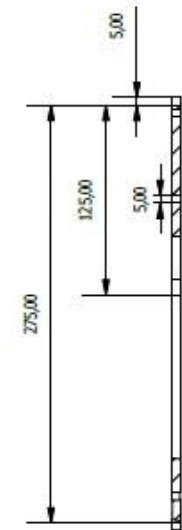
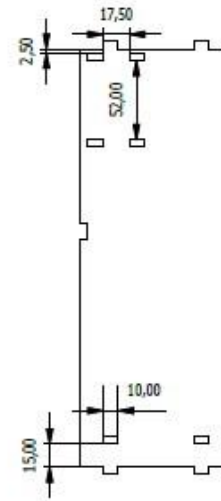
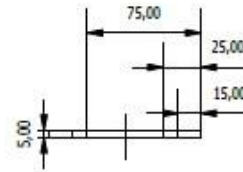
Escala:  
**1:1**

Nombre Plano: Escuadra 20B15H

Nº Plano:  
**10**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		Fecha: 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	 Escuela Técnica Superior de Ingeniería del Diseño	Autor: Ciudad-Real Villaescusa, Alejandro
		Nombre Plano: Pared Torre 2
		Escala: 1:2
		Nº Plano: 11



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITECNICA  
DE VALÈNCIA



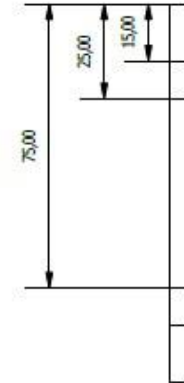
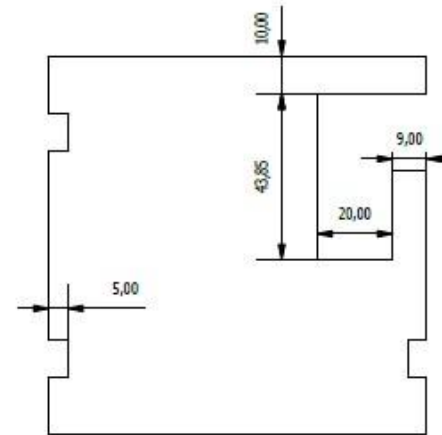
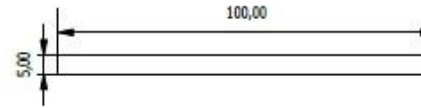
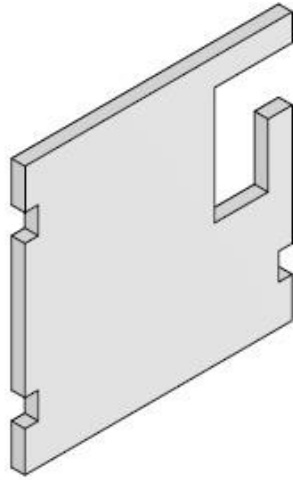
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
**1:5**

Nombre Plano: Pared Torre 1

Nº Plano:  
**12**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



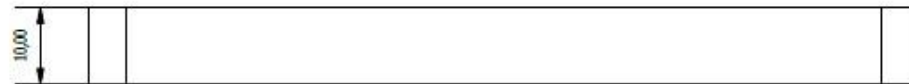
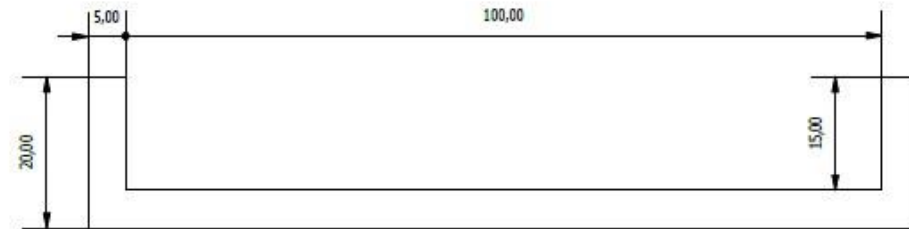
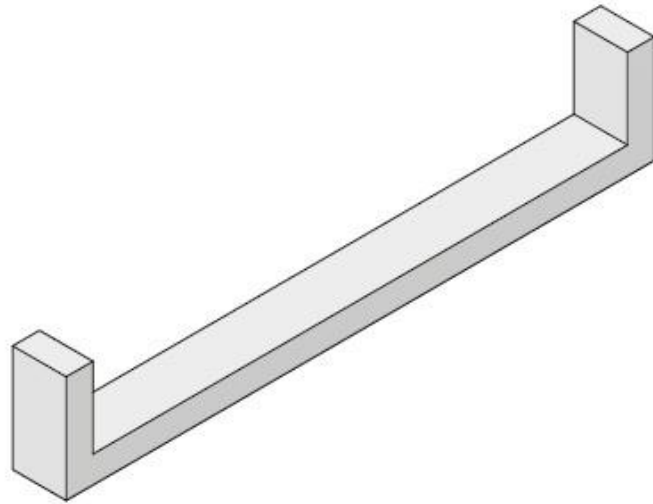
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:2

Nombre Plano: Tapa Torre

Nº Plano:  
13



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



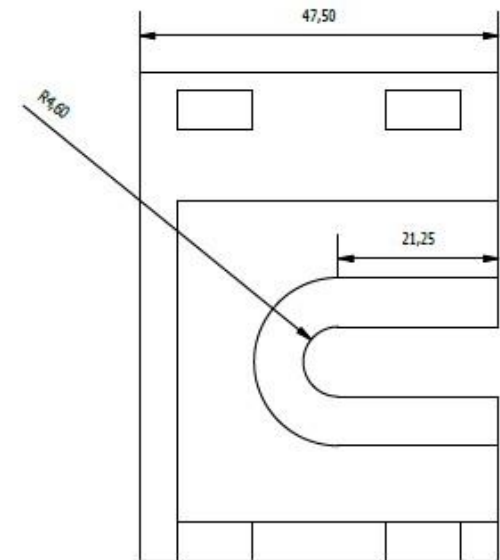
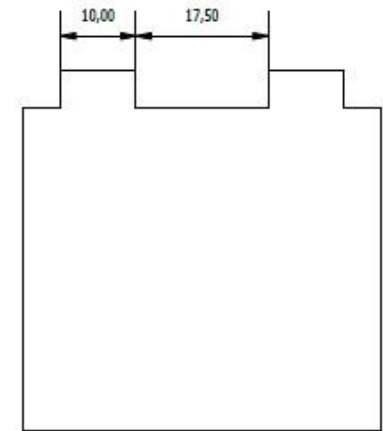
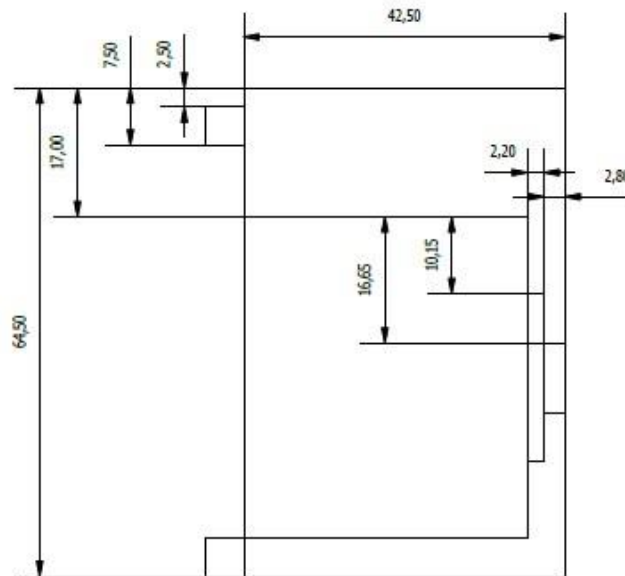
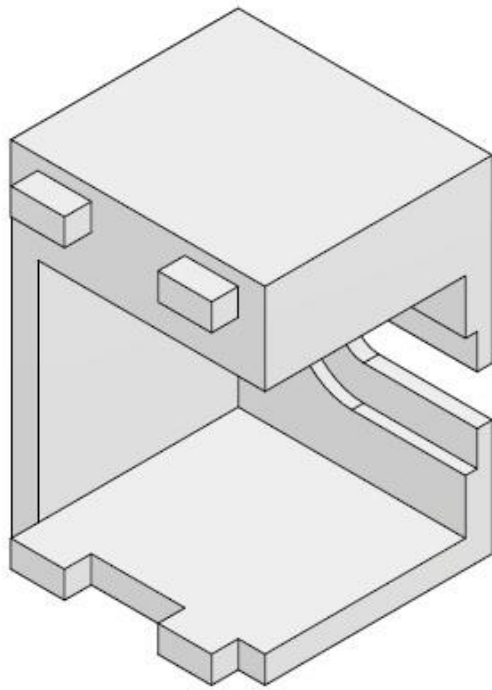
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
**1:1**

Nombre Plano: Soporte Pared Torre

Nº Plano:  
**14**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



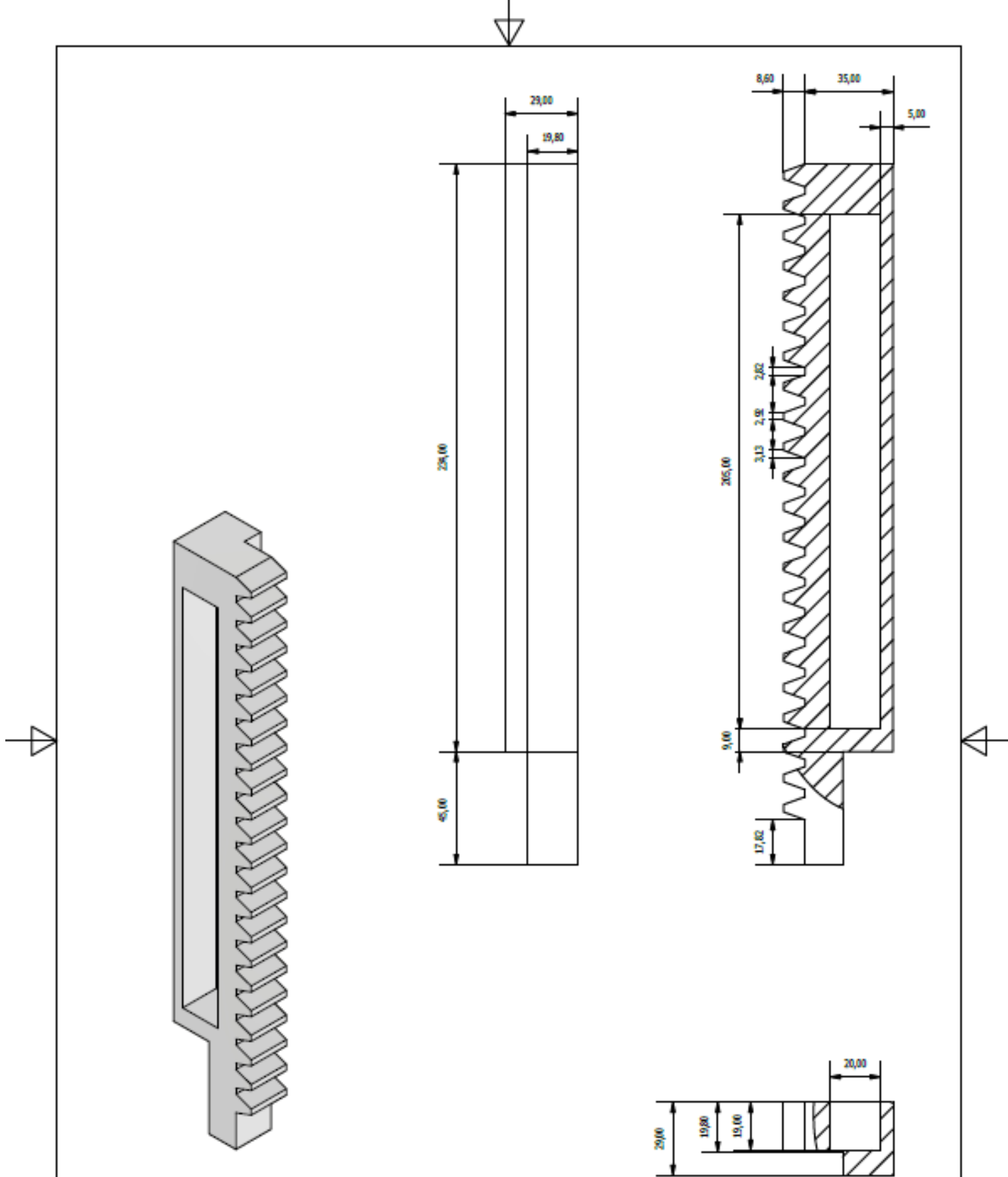
Escuela Técnica Superior de Ingeniería del Diseño



Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:1

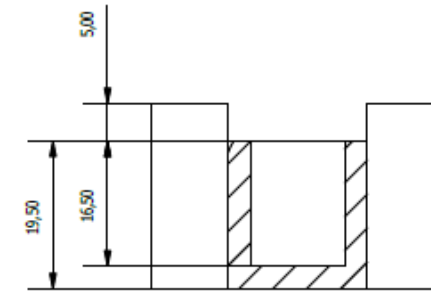
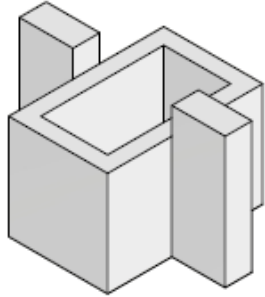
Nombre Plano: Soporte Motor Y

Nº Plano:  
15

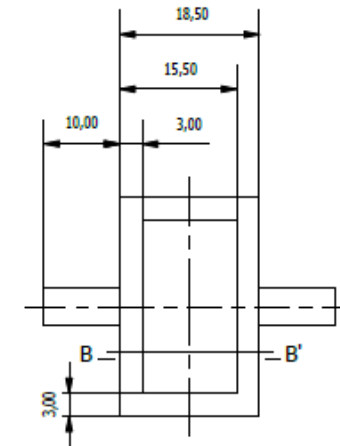
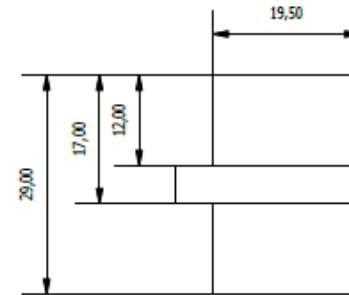




<b>Proyecto:</b> Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		<b>Fecha:</b> 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	 Escuela Técnica Superior de Ingeniería del Diseño	<b>Autor:</b> Ciudad-Real Villaescusa, Alejandro
		<b>Nombre Plano:</b> Cremallera Y
		<b>Escala:</b> 1:2
		<b>Nº Plano:</b> 16

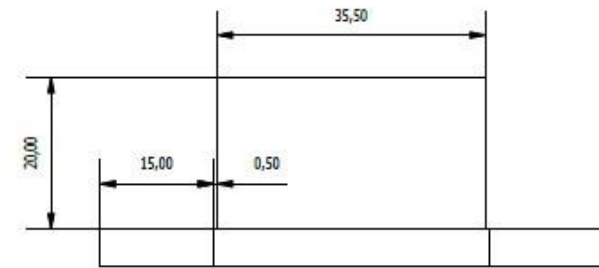
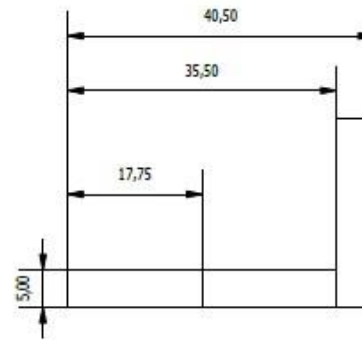
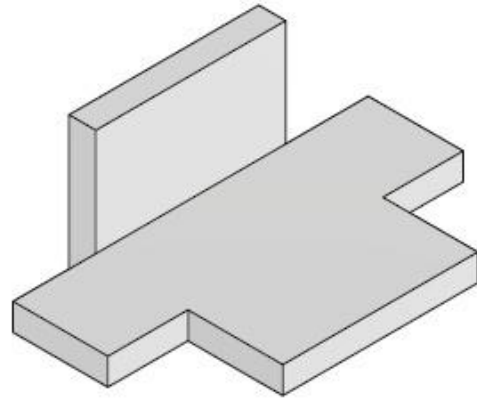




Sección B-B'



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		Fecha: 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	 Escuela Técnica Superior de Ingeniería del Diseño	Autor: Ciudad-Real Villaescusa, Alejandro
		Nombre Plano: Caja Servo
		Escala: 1:1
		Nº Plano: 17



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



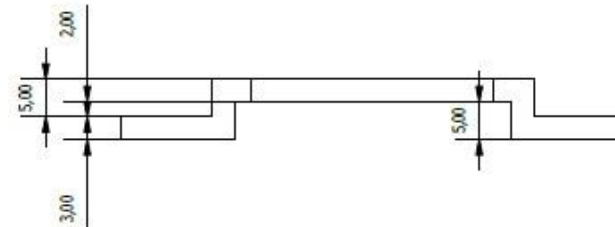
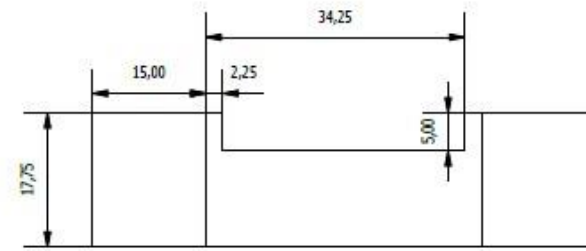
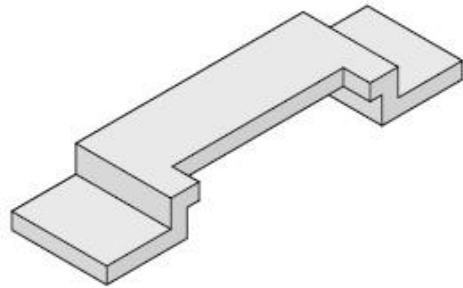
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
**1:1**

Nombre Plano: Soporte Camara

Nº Plano:  
**18**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



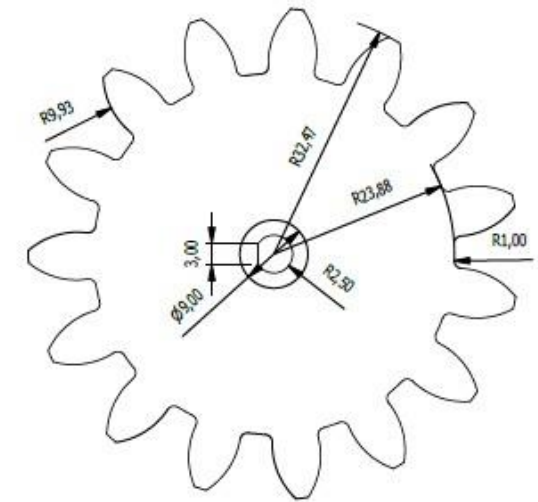
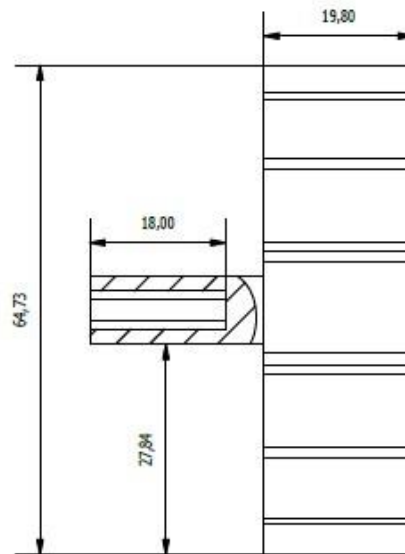
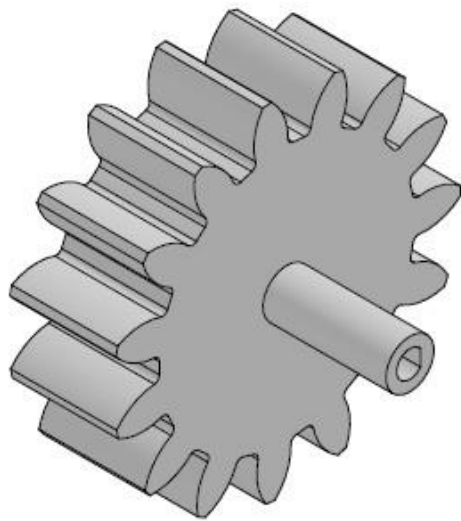
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
**1:1**

Nombre Plano: Encaje Cámara

Nº Plano:  
**19**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



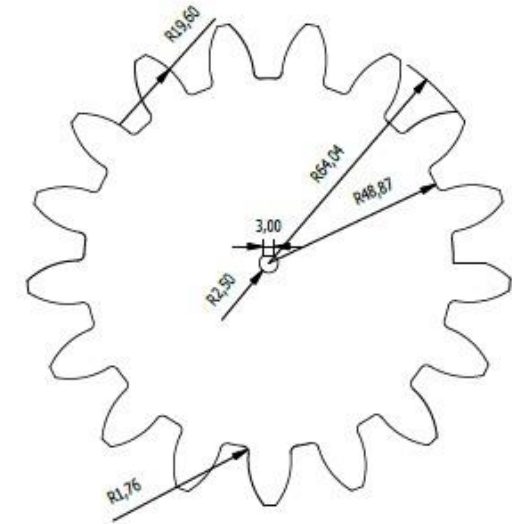
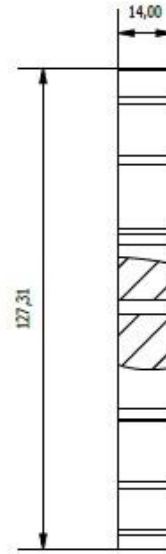
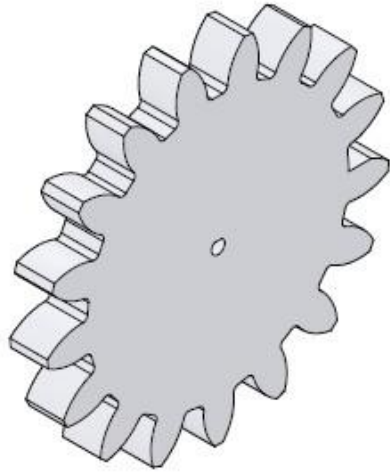
Escuela Técnica Superior de Ingeniería del Diseño



Autor: Ciudad-Real Villaescusa, Alejandro

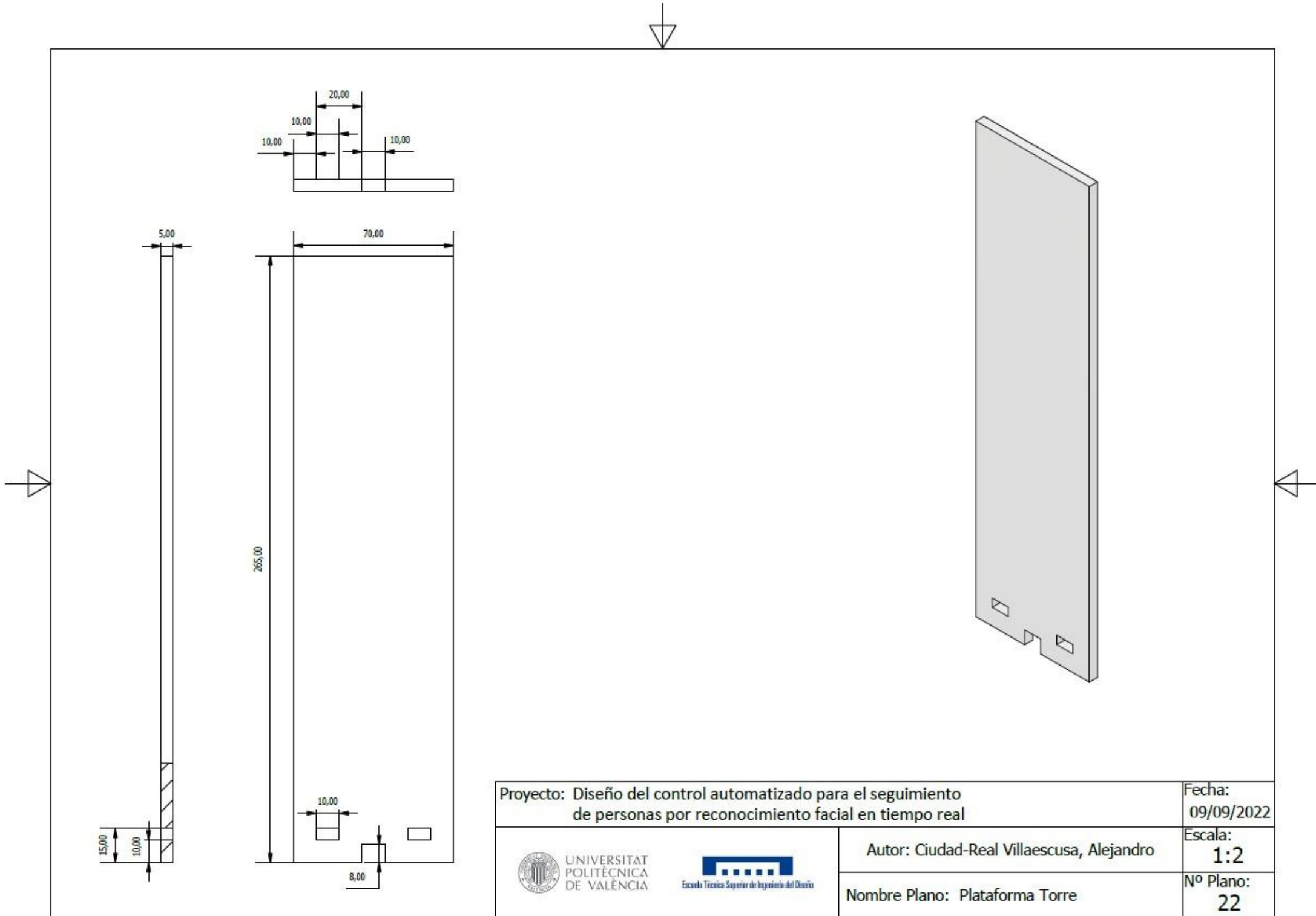
Escala:  
**1:1**

Nombre Plano: Engranaje 2

Nº Plano:  
**20**



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real		Fecha: 09/09/2022
 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	 Escuela Técnica Superior de Ingeniería del Diseño	Autor: Ciudad-Real Villaescusa, Alejandro
		Nombre Plano: Engranaje 1
		Escala: 1:2
		Nº Plano: 21



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha: 09/09/2022

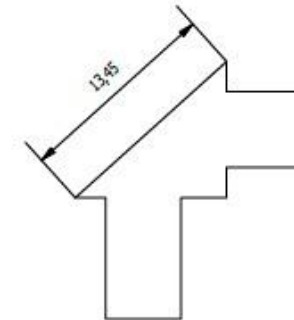
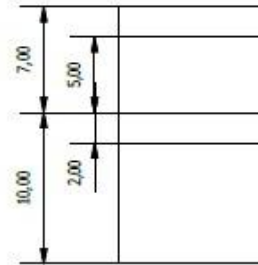
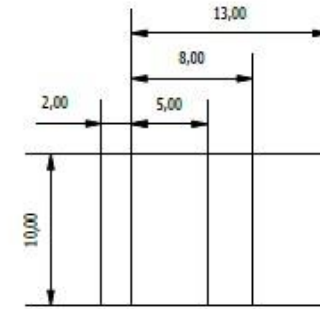
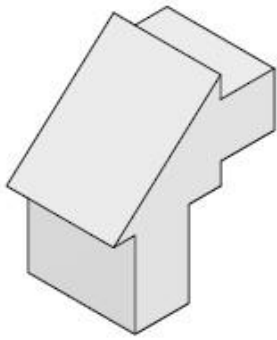


Autor: Ciudad-Real Villaescusa, Alejandro

Escala: 1:2

Nombre Plano: Plataforma Torre

Nº Plano: 22



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



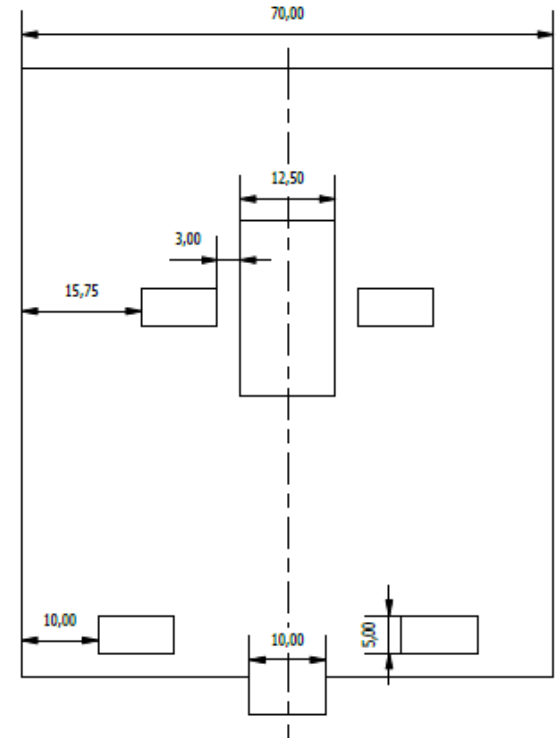
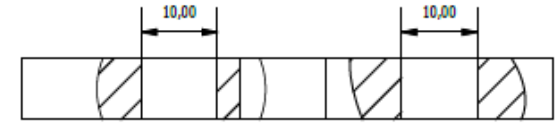
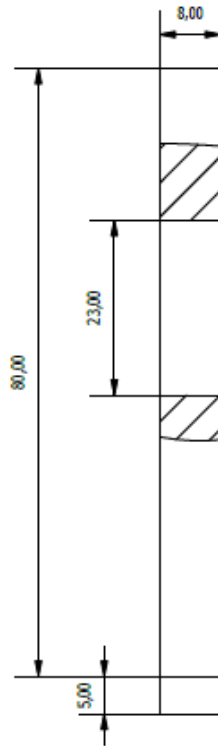
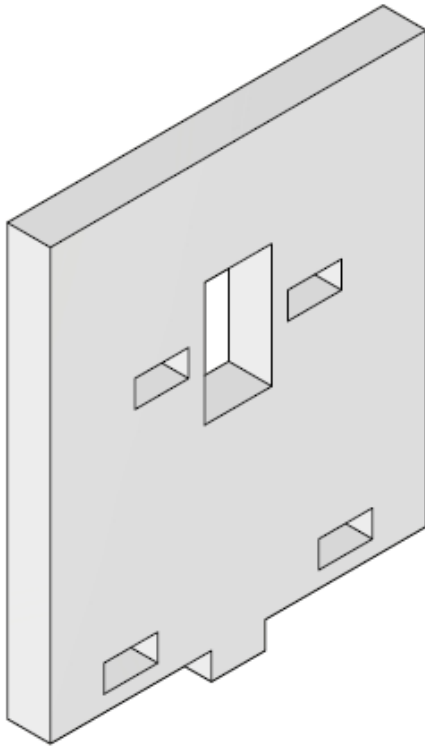
Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
2:1

Nombre Plano: Escuadra2B2H

Nº Plano:  
23



Proyecto: Diseño del control automatizado para el seguimiento de personas por reconocimiento facial en tiempo real

Fecha:  
09/09/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Autor: Ciudad-Real Villaescusa, Alejandro

Escala:  
1:1

Nombre Plano: Plataforma

Nº Plano:  
24





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °3: PLIEGO DE CONDICIONES**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022



## ÍNDICE

3. Pliego de condiciones .....	114
3.1. Objeto.....	114
3.1.1. Condiciones generales.....	114
3.1.1.1. Vigencia.....	114
3.1.1.2. Modificaciones .....	115
3.2. Materiales.....	115
3.2.1. Estructura.....	115
3.2.1.1. Metacrilato.....	115
3.2.1.2. Sinterizado de nylon .....	115
3.2.2. Hardware .....	115
3.2.2.1. Stepper.....	115
3.2.2.2. Drivers.....	115
3.2.2.3. Servo.....	116
3.2.2.4. Fuente de alimentación.....	116
3.2.2.5. Finales de carrera.....	116
3.2.2.6. Condensadores.....	116
3.2.2.7. Arduino.....	116
3.2.2.8. Módulo Bluetooth.....	116
3.3. Condiciones de la ejecución.....	117
3.4. Prueba de servicio.....	117
3.5. Certificaciones.....	117
3.5.1. Marcado CE.....	117

## 3. Pliego de condiciones

### 3.1. Objeto

El objetivo de este Trabajo de Fin de Grado es realizar el diseño, programación y montaje de un dispositivo capaz de monitorizar, mediante visión artificial en tiempo real, la posición y orientación del rostro de un individuo y en función de ellas realizar el control automático sobre los actuadores para ubicar el rostro de la persona en el centro de la imagen.

Se dispondrá de 2 tipos de sensores, una webcam para realizar todo el análisis de la información gráfica y de finales de carrera para el correcto funcionamiento del movimiento de la estructura. En cuanto a actuadores contaremos con motores paso a paso y con un servomotor.

El proyecto contará con la capacidad de seleccionar una opción de control manual por si el usuario desea localizar el dispositivo en una posición determinada. Para ello se ha desarrollado una aplicación móvil que, mediante transferencia de datos por Bluetooth permite el control manual de la estructura.

#### 3.1.1. Condiciones generales

Este proyecto tiene carácter de obligado cumplimiento una vez sellado y legalizado, debiendo ser objeto de aprobación previa todas aquellas modificaciones al mismo durante su ejecución.

##### 3.1.1.1. Vigencia

Este Pliego de Condiciones, con todos sus articulados, estará en vigor durante la fabricación y hasta su completa finalización. Frente a posibles discrepancias, el orden de prioridad de los documentos básicos del Proyecto será el siguiente:

- 1).- Planos.
- 2).- Pliego de Condiciones.
- 3).- Presupuesto.
- 4).- Memoria.

### 3.1.1.2. Modificaciones

Durante la fabricación del dispositivo, se podrán realizar todas las modificaciones que se estimen oportunas para mejorar o adaptar el sistema, siempre que las estas no modifiquen el funcionamiento del dispositivo.

## 3.2. Materiales

### 3.2.1. Estructura

#### 3.2.1.1. Metacrilato

Se usará metacrilato para diferentes piezas caracterizadas por tener como mínimo 2 superficies planas y paralelas entre sí. Estas presentarán distintos grosores dentro de un rango entre 5cm y 8cm mostrados en el documento de planos previos. En caso de necesitar grosores mayores se superpondrán capas de este material para conseguir el grosor requerido.

#### 3.2.1.2. Sinterizado de nylon

Se usará nylon sinterizado para crear aquellas piezas de tamaño reducido, forma irregular o con gran componente de detalle.

### 3.2.2. Hardware

#### 3.2.2.1. Stepper

Se instalarán 2 motores paso a paso bipolares modelo NEMA 17-44HS01 con capacidad de carga de 3.2 kg/cm. Estos se asociarán a sus respectivos drivers para realizar un control óptimo.

#### 3.2.2.2. Drivers

Serán requeridos 2 drivers A-4988 con capacidad de control de motores paso a paso bipolares de hasta 2A.

### 3.2.2.3. Servo

Usaremos 1 servo modelo SG-90 con 180° de giro y capacidad de soportar cargas de hasta 1,6 kg/cm.

### 3.2.2.4. Fuente de alimentación

Dispondremos de una fuente de alimentación externa modelo MeanWell LPV-60-12, esta se encargará de suministrar 12V y 5A para ejercer acción sobre los motores paso a paso. Esta fuente cuenta con una conexión directa a la red eléctrica.

### 3.2.2.5. Finales de carrera

Se usarán 4 unidades del modelo CESFONJER, este tipo de final de carrera dispone de un micro interruptor de límite asociado a una bisagra que dispone de un roller.

### 3.2.2.6. Condensadores

Se requerirán 2 condensadores electrolíticos de 100nF asociados al circuito de alimentación externa de los drivers para evitar la rotura de la placa.

### 3.2.2.7. Arduino

Contaremos con 1 Arduino Mega 2560 con capacidad máxima de 54 pines digitales y 4 puertos de comunicación serie.

### 3.2.2.8. Módulo Bluetooth

Se usará 1 módulo bluetooth modelo HC-06 con capacidad de trabajo únicamente como esclavo y transmisión de datos por el puerto serie del Arduino Mega 2560.

### 3.3. Condiciones de ejecución

El sistema estará alimentado por dos fuentes distintas, la primera de ellas será de 5V, es decir, la misma que proporciona el Arduino mega, para el uso de esa fuente bastará con conectar la placa a una fuente USB, en este caso el mismo ordenador. Por otro lado, será requerida una fuente auxiliar para alimentar los motores paso a paso, esta requerirá de una conexión a la red eléctrica para poder obtener la energía adaptada para nuestros requerimientos.

### 3.4. Prueba de servicio

La prueba de servicio constará de distintas fases. La fase inicial se producirá de forma automática puesto que está programada para cada puesta en marcha del proyecto, básicamente es una calibración de la posición inicial, en el caso del servo colocará la cámara en su posición de 90°, es decir, de frente a un hipotético usuario, de igual forma ubicará los márgenes de la estructura para ambas translaciones gracias a los finales de carrera y tras esto se ubicará en el punto medio de estas trayectorias.

La segunda fase consistiría en la calibración de, tanto los parámetros de decisión de movimiento en Python como la de la acción de los actuadores de Arduino, esta fase es en parte opcional en función de si el sistema ha sido movido de otro set-up o por el contrario el usuario va a situarse en otro rango de profundidad de la imagen.

Como tercera fase será necesario una comprobación del correcto funcionamiento de la aplicación, desde el correcto emparejamiento de dispositivos hasta la activación de las funciones de movimiento o las limitaciones de recorrido producidas por los finales de carrera.

### 3.5. Certificaciones

#### 3.5.1. Mercado CE

Cada material y componente usado contarán con el marcado CE previo a su compra, será determinado por el suministrador y otorgado por las entidades a las que corresponda.

El proyecto contará con distintas directivas de aplicación obligada:

-Compatibilidad Electromagnética (EMC): <https://www.marcado-ce.com/directivas-europeas-de-nuevo-enfoque/compatibilidad-electromagnetica-emc.html>

DIRECTIVA 2014/30/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 26 de febrero de 2014 sobre la armonización de las legislaciones de los Estados miembros en materia de compatibilidad electromagnética (DOUE L 96/79), 29/3/2014.

-Equipos radioeléctricos y equipos terminales de telecomunicación y reconocimiento mutuo de su conformidad (RTTE): <https://www.marcado-ce.com/directivas-europeas-de-nuevo-enfoque/equipos-radioelectricos-terminales-de-telecomunicacion-reconocimiento-mutuo-conformidad.html>

DIRECTIVA 2014/53/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 16 de abril de 2014 relativa a la armonización de las legislaciones de los Estados miembros sobre la comercialización de equipos radioeléctricos, y por la que se deroga la Directiva 1999/5/CE (DOUE L.153/62), 22/5/2014.

-Máquinas: <https://www.marcado-ce.com/directivas-europeas-de-nuevo-enfoque/maquinas.html>

DIRECTIVA 2006/42/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 17 de mayo de 2006 relativa a las máquinas y por la que se modifica la Directiva 95/16/CE (DOUE L.157/24), 9/6/2006.

-Restricción de uso de sustancias peligrosas en aparatos eléctricos y electrodomésticos (ROHS): <https://www.marcado-ce.com/directivas-europeas-de-nuevo-enfoque/restriccion-uso-de-sustancias-peligrosas-en-aparatos-electricos-electrodomesticos.html>

DIRECTIVA 2011/65/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 8 de junio de 2011 sobre restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos (DOUE L.174/88), 1/7/2011.

-Indicación del consumo de energía y otros recursos por parte de los productos relacionados con la energía, mediante el etiquetado y una información normalizada (ErP): <https://www.marcado-ce.com/directivas-europeas-de-nuevo-enfoque/indicacion-consumo-energia-etiquetado-normalizada.html>

DIRECTIVA 2010/30/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 19 de mayo de 2010 relativa a la indicación del consumo de energía y otros recursos por parte de los productos relacionados con la energía, mediante el etiquetado y una información normalizada (DOUE L.163/1), 18/6/2010.

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro
------------------------------------



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4: PRESUPUESTO**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022





## ÍNDICE

### 4. Presupuesto

4.1. Precios unitarios .....	121
4.2. Precios descompuestos .....	121
4.3. Mediciones .....	122
4.4. Presupuesto .....	123
4.5. Resumen .....	123



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4.1: PRECIOS UNITARIOS**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## 4.1. Precios unitarios

4.1. Cuadro de Precios Unitarios			
Ref.	Unidad	Descripción	Precio (€)
<b>Materiales</b>			
m1	ud.	Microcontrolador (Arduino Mega 2560)	47,68
m2	ud.	Stepper (NEMA 17)	9,99
m3	ud.	Driver A-4988	2,00
m4	ud.	Finales de carrera	1,12
m5	ud.	Servomotor SG90	2,40
m6	ud.	Modulo Bluetooth HC-06	9,99
m7	ud.	Fuente de alimentación MeanWell 12V 5A	25,34
m8	m3.	Metacrilato	60,00
m9	kg.	Nylon	40,00
<b>M.O.D (Mano de obra directa)</b>			
h1	h.	Técnico de impresión	15,00
h2	h.	Programador	15,00
h3	h.	Diseñador 3D	15,63
h4	h.	Ayudante	10,62
<b>Maquinaria</b>			
mq1	h.	Máquina de impresión 3D	5,00
mq2	h.	Máquina de corte láser	1,50
mq3	h.	AutoDesk Inventor	4,00
mq4	h.	Hardware programación (ordenador)	1,00



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4.2: PRECIOS DESCOMPUESTOS**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## 4.2. Precios descompuestos

4.2. Cuadro de precios descompuestos					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
<b>Sub 1</b>	Ud.	Subsistema de programación basado en microcontrolador (Arduino Mega 2560, software Python y software MiAppInventor así como las programaciones respectivas) y el diseño 3D de la estructura móvil totalmente ejecutados y probados.			
<b>Materiales</b>					
m1	ud.	Arduino Mega 2560	47,68	1,00	47,68
<b>M.O.D.</b>					
h2	h.	Programador para Arduino	15,00	24,00	360,00
h2	h.	Programador para Python	15,00	12,00	180,00
h2	h.	Programador y diseño app para MiAppInventor	15,00	1,00	15,00
h3	h.	Diseño de estructura 3D	15,63	15,00	234,45
h4	h.	Comprobación de funcionamiento (prueba de servicio)	15,00	6,00	90,00
<b>Maquinaria</b>					
mq3	h.	AutoDesk Inventor	4,00	15,00	60,00
mq4	h.	Hardware programación	1,00	58,00	58,00
<b>Medios Auxiliares</b>					
	%	M.A. sobre costes directos	5%	985,13	49,26
				<b>Precio de ejecución material</b>	<b>1034,39</b>

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
<b>Sub 2</b>	Ud.	Subsistema de materiales formado por todos aquellos componentes físicos desde el material de metacrilato usado para el corte laser, el material usado para la realización de piezas en 3D y todo el hardware que compone el sistema de sensores y actuadores (stepper, drivers, servos, finales de carrera, módulo bluetooth y fuente de alimentación externa) totalmente ejecutados y probados.			
<b>Materiales</b>					
m2	ud.	Stepper (NEMA 17)	9,99	2,00	19,98
m3	ud.	Driver A-4988	2,00	2,00	4,00
m4	ud.	Finales de carrera	1,12	4,00	4,48
m5	ud.	Servomotor SG90	2,40	1,00	2,40
m6	ud.	Modulo Bluetooth HC-06	9,99	1,00	9,99
m7	ud.	Fuente de alimentación MeanWell 12V 5A	25,34	1,00	25,34
m8	m3.	Metacrilato	60,00	1,00	60,00
m9	kg.	Nylon	40,00	1,00	40,00
<b>M.O.D.</b>					
h1	h.	Técnico de impresión	15,00	5,00	75,00
h4	h.	Ayudante Montaje	10,62	6,00	63,72
<b>Maquinaria</b>					
mq1	h.	Máquina de impresión 3D	5,00	20,00	100,00
mq2	h.	Máquina de corte láser	1,50	4,00	6,00
<b>Medios Auxiliares</b>					
	%	M.A. sobre costes directos	5%	410,91	20,55
				<b>Precio de ejecución material</b>	<b>431,46</b>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4.3: MEDICIONES**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

### 4.3. Mediciones

4.3. Mediciones			
Capítulo	Unidad	Valoración	Cantidad
Sub 1	Ud.	Subsistema de programación basado en microcontrolador (Arduino Mega 2560, software Python y software MiAppInventor así como las programaciones respectivas) y el diseño 3D de la estructura móvil totalmente ejecutados y probados.	1,00
Sub 2	Ud.	Subsistema de materiales formado por todos aquellos componentes físicos desde el material de metacrilato usado para el corte laser, el material usado para la realización de piezas en 3D y todo el hardware que compone el sistema de sensores y actuadores (stepper, drivers, servos, finales de carrera, módulo bluetooth y fuente de alimentación externa) totalmente ejecutados y probados.	1,00



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4.4: PRESUPUESTO**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022



## 4.4. Presupuesto

4.4. Presupuesto					
Capítulo	Unidad	Valoración	Precio (€)	Cantidad	Importe (€)
Sub 1	Ud.	Subsistema de programación basado en microcontrolador (Arduino Mega 2560, software Python y software MiAppInventor así como las programaciones respectivas) y el diseño 3D de la estructura móvil.	1034,39	1	1034,39
Sub 2	Ud.	Subsistema de materiales formado por todos aquellos componentes físicos desde el material de metacrilato usado para el corte laser, el material usado para la realización de piezas en 3D y todo el hardware que compone el sistema de sensores y actuadores (stepper, drivers, servos, finales de carrera, módulo bluetooth y fuente de alimentación externa)	431,46	1	431,46

El presupuesto de ejecución material asciende a la cantidad de **MIL CUATROCIENTOS SESENTA Y CINCO EUROS CON OCHENTA Y UN CÉNTIMOS**.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# **DISEÑO DEL CONTROL AUTOMATIZADO PARA EL SEGUIMIENTO DE PERSONAS POR RECONOCIMIENTO FACIAL EN TIEMPO REAL**

**Documento N. °4.5: RESUMEN**

AUTOR: Ciudad-Real Villaescusa, Alejandro

TUTOR: Adán Roca, Vicente

CURSO ACADÉMICO: 2021/2022

## 4.5. Resumen

4.5. Resumen			
Capítulo	Unidad	Valoración	Importe (€)
Sub 1	Ud.	Subsistema de programación	1034,39
Sub 2	Ud.	Subsistema de materiales	431,46

<b>Presupuesto de ejecución material</b>	<b>1465,84</b>
CG(13%)	190,56
BI(6%)	87,95
<b>Suma</b>	<b>1744,35</b>
IVA (21%)	307,83
<b>Total presupuesto general</b>	<b>2052,18</b>

El presupuesto general asciende a la cantidad de **DOS MIL CINCUENTA Y DOS EUROS CON DIECIOCHO CÉNTIMOS.**

Valencia, septiembre de 2022

El autor:

Ciudad-Real Villaescusa, Alejandro