



# PROYECTO MECATRONICO

UNIVERSIDAD DE VALLADOLID

## PROYECTO 2015 2016-GRUPO1

Proyecto Mecatrónico compuesto principalmente por un brazo robótico sobre una estación industrial que cuenta con varios mecanismos como una puerta de ingreso y una banda transportadora de la cual se hará una recolección de piezas mediante el movimiento del brazo robótico que podrá controlarse por un guante acoplado con sensores flex y además se tiene otro control mediante una aplicación para dispositivos Android mediante la cual se controlará los grados de libertad del robot, ambos tipos de control no funcionaran simultáneamente se tendrá la opción de elegir entre ellos

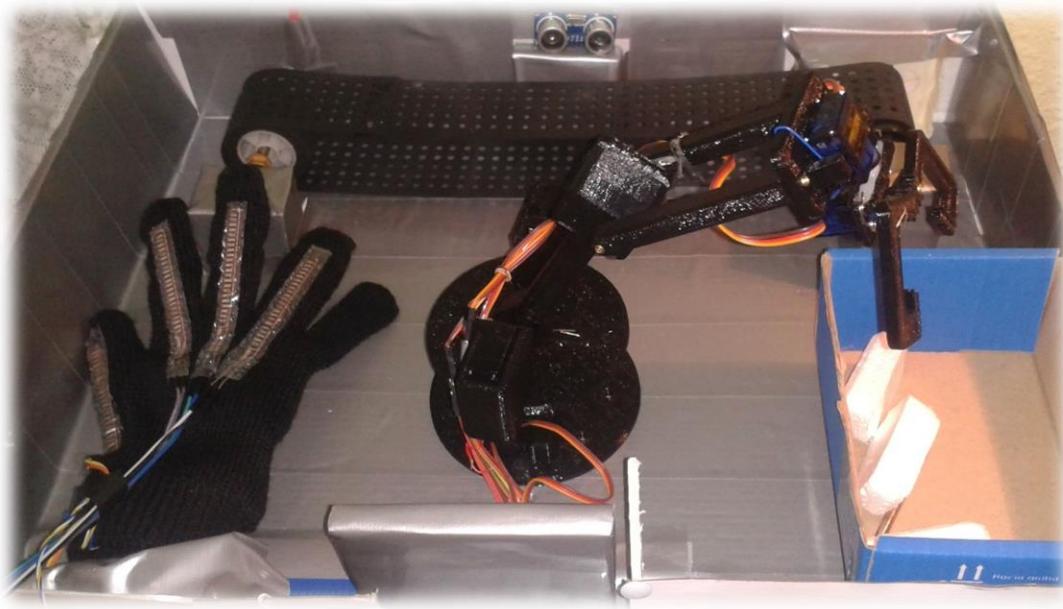
Ken Vera Chan  
[kenveracha@yahoo.es](mailto:kenveracha@yahoo.es)

Pilar Estefanía Samaniego  
[pilar.samaniego@hotmail.com](mailto:pilar.samaniego@hotmail.com)

# ÍNDICE

- 1. INTRODUCCIÓN**
- 2. ELEMENTOS UTILIZADOS EN EL PROYECTO**
- 3. CONSTRUCCIÓN Y DISEÑO EN INVENTOR**
  - 2.1. PLANO BASE DEL ROBOT**
  - 2.2. PLANO BASE DE LA PINZA DEL ROBOT**
  - 2.3. PLANO BASE DE LA PINZA SUJECCIÓN**
  - 2.4. PLANO BRAZO PRIMERO SERVO EXTREMO SUPERIOR**
  - 2.5. PLANO BRAZO SUPERIOR UNIÓN CON EL EXTREMO DE LA PINZA**
  - 2.6. PLANO BRAZO SUPERIOR UNIÓN CON EL EXTREMO DE LA PINZA PARTE II**
  - 2.7. PLANO EXTREMO BRAZO SUSTENTADOR DE LA ARTICULACIÓN**
  - 2.8. PLANO ESLABÓN SOBRE BASE DEL ROBOT**
  - 2.9. PLANO ESLABÓN SOBRE BASE DEL ROBOT**
  - 2.10. PLANO EXTREMO ROTATORIO DE LA BASE:**
  - 2.11. PLANO DEL NEXO ENTRE LA ZONA BRAZO INFERIOR Y SERVOS PARALELOS DE BASE**
  - 2.12. PLANO PERNO DE ENGANCHE DE PIEZAS CON EL SERVO IMPLICADO**
  - 2.13. PLANO PINZA PRIMERA EXTREMO GUIADO POR EL SERVO**
  - 2.14. PLANO PINZA SEGUNDA EXTREMO GUIADO POR UN PERNO**
  - 2.15. PLANO PIEZA SOPORTE Y ROTACIÓN DEL EXTREMO DEL ROBOT**
  - 2.16. PLANO SOPORTE DE SUJECCIÓN DEL SERVO DE LAS PINZAS**
- 4. IMÁGENES DE LAS PIEZAS EN INVENTOR**
  - 4.1. TABLA DE PIEZAS DISEÑADAS**
  - 4.2. IMAGEN DEL ENSAMBLAJE FINAL DEL ROBOT COMPLETO**
  - 4.3. OTROS ENSAMBLAJES DE CONSTRUCCIÓN INICIAL**
- 5. ACONDICIONAMIENTO DE LOS SENORES**
  - 8.1. PRECIO DE LOS SERVOS UTILIZADOS**
    - 8.1.1. SERVO DE LA BASE ROTATORIA DEL ROBOT**
    - 8.1.2. SERVOS DE LA ROTACIÓN DEL BRAZO ROBOT**
    - 8.1.3. SERVOS DE ROTACIÓN DE LAS ARTICULACIONES RESTANTES**
    - 8.1.4. MOTOR DE ROTACIÓN PARA PUERTA ACCESO Y CINTA TRANSPORTADORA**
  - 8.2. PRECIO TORNILLOS Y PERNOS**
  - 8.3. PRECIO DE LAS PIEZAS IMPRESAS**
- 6. CONEXIÓN DEL DRIVER PARA LOS MOTORES DC**
- 7. FOTOGRAFÍAS DE LA IMPRESIÓN 3D**
- 8. CÁLCULOS DE LA CINEMÁTICA Y DINÁMICA DEL ROBOT**
  - 8.1. ESTUDIO CINEMÁTICO DEL ROBOT**
  - 8.2. ESTUDIO DINÁMICO DEL ROBOT**
  - 8.3. ESTUDIO DINÁMICO DE LA PUERTA ACCESO Y EL DE LA CINTA TRASPORTADORA**
- 9. DIAGRAMA DE FLUJO DEL CODIGO EN ARDUINO**
- 10. CÓDIGOS DE PROGRAMACIÓN ARDUINO**
- 11. PROGRAMA DE LA APLICACIÓN DESARROLLADA PARA DISPOSITIVOS ANDROID**
- 12. PRECIO DEL PROYECTO**
- 13. PROBLEMAS ENCONTRADOS Y SOLUCIONES ADOPTADAS**
- 14. CONCLUSION DEL PROYECTO**
- 15. BIBLIOGRAFÍA**

## PROYECTO 2015 2016



### **1. INTRODUCCIÓN**

El proyecto consiste en la implementación de un brazo robótico de 5 grados de libertad con una pinza en su extremo basado en un sistema de engranaje unitario y mini servo, el robot que está formado por un sin número de piezas diseñadas en Inventor las cuales fueron impresas posteriormente en una impresora 3D, este brazo robótico es controlado mediante dos sistemas, el primero consiste en un Guante el cual podrá ser operado por cualquier persona, y otro que es un control mediante un dispositivo Android al cual se la instalado una aplicación desarrollada exclusivamente para este proyecto. Mediante el control con el guante sensible a cualquier movimiento de la mano el robot produce un movimiento similar y teniendo en el dedo pulgar el control para la pinza y además consta de un potenciómetro en el guante para el control de un eje adicional. Por el contrario mediante la aplicación para Android se tendrá un control para cada uno de los grados de libertad del robot que en total son 5.

## 2. ELEMENTOS UTILIZADOS EN EL PROYECTO

De manera General el proyecto está compuesto por los siguientes mecanismos y elementos:

- ✓ **Brazo Robótico de 5 grados de libertad**



1. Articulación para el giro de toda la estructura robótica,
2. Articulación para los primeros brazos del robot
3. Articulación para los eslabones siguientes
4. Articulación para la orientación de la pinza
5. Articulación para la apertura y cierre de la pinza.

El brazo robótico está constituido por:

- Las piezas mecánicas impresas en una impresora 3D.
- Mini servos que permitan el movimiento articular de los componentes.
- Tornillos específicos.
- Pernos para permitir el giro articular con la ayuda de un servo.

- ✓ **Mecanismo de Cinta transportadora** constituido por un tornillo acoplado a un motor DC de 12 V y piñones.



- ✓ **Mecanismo para la apertura y cierre de una puerta de ingreso**, constituido por un tornillo acoplado al motor junto con otro mecanismo Piñón-Cremallera.



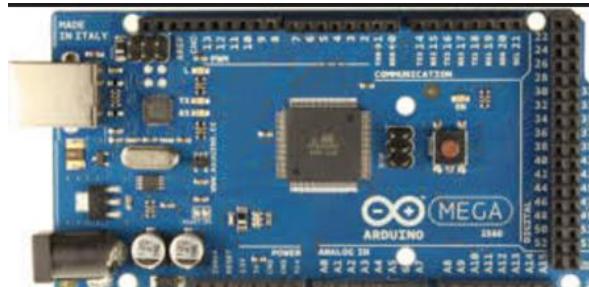
- ✓ **Guante sensible a los movimientos de la mano**, construido a base de 4 sensores Flex y un potenciómetro para el control adicional del primer grado de libertad del robot.



Mediante el movimiento de la mano el Brazo Robótico realizará los mismos movimientos, de tal manera que el operario pueda controlar con el movimiento de su mano todas las acciones del robot, si es para agarre de un objeto se hará el uso de la pinza mediante el movimiento del dedo pulgar, el potenciómetro

permitirá girar la estructura en general del robot para posicionarlo sobre un sitio u otro, de mejor manera esto será visualizado en la demostración práctica.

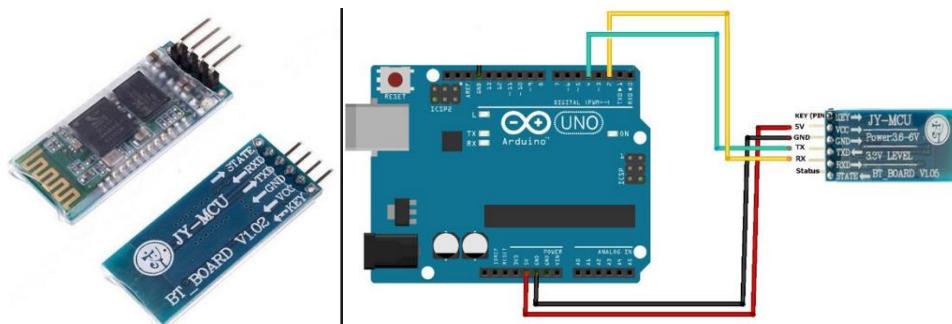
✓ ARDUINO MEGA 2560



Mediante nuestro arduino se hará el control total de nuestro proyecto, el mismo que presenta las siguientes características:

El Arduino Mega está basado en el microcontrolador ATMega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 Mhz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente se lo conecta al PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

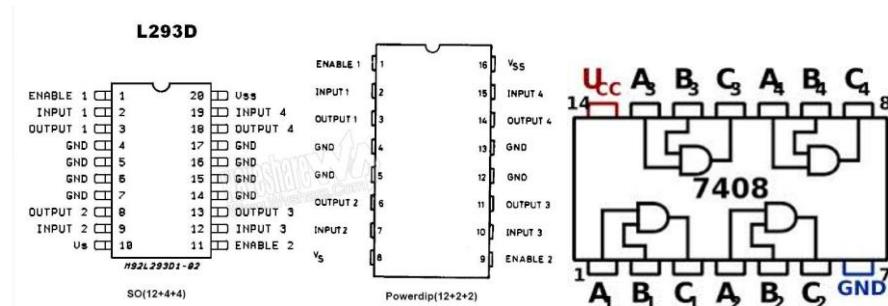
✓ MODULO BLUETOOTH HC06



Los módulos de bluetooth HC-06 son módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras.

Mediante este dispositivo estableceremos la comunicación entre nuestro Arduino Mega y nuestro dispositivo Android y se establecerá la comunicación bluetooth respectiva.

✓ PUENTE H L293D Y COMPUERTA AND 74LS08



Se utilizará el puente H junto con una compuerta AND para poder hacer el control de los motores DC de la puerta de ingreso y de la cinta transportadora, con la puerta And se podrá establecer el sentido de giro de los motores.

Adicional a todos estos elementos se tendrán elementos adicionales como leds, buzzer, pulsadores, dip switches para realizar funciones como subir o bajar la velocidad de giro de los motores, cambiar el sentido de giro, etc.

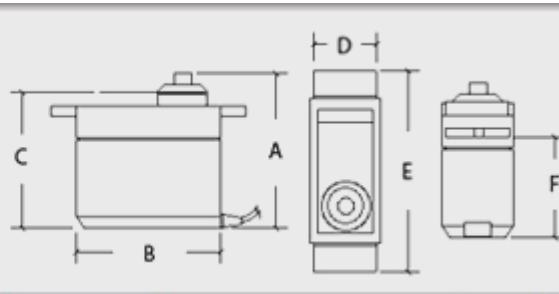
✓ SERVOMOTOR Power HD Standard Servo 3001HB



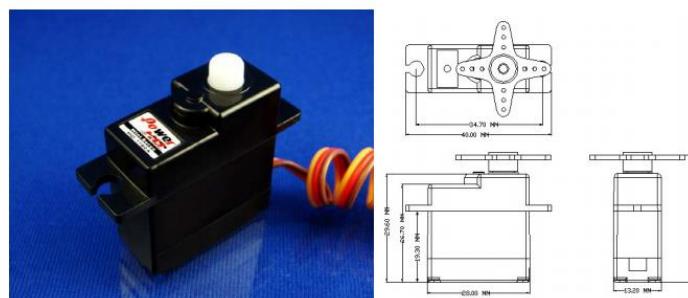
**Model:** HD-3001HB  
**Voltage:** 4.8~6.0v  
**Speed:** 0.15~0.12sec  
**Torque:** 3.5~4.4kg/cm  
**Weight:** 43g  
**Size:** 40.7x20.5x39.5mm

Este servomotor se utilizará para la base de nuestro brazo Robótico

Peso (g)	43
Torque (kg)	3.5
Velocidad(Sec/60deg)	0.15
A(mm)	46
B(mm)	41
C(mm)	42
D(mm)	21
E(mm)	56
F(mm)	29



✓ SERVOMOTOR Power HD Standard Servo 1160 A



Este servomotor se utilizará para la sujeción y movimiento de los dos primeros brazos del Robot.

- Torque:2.7kg a 6v
- Peso: 16g
- Voltaje:4.8v-6v
- Velocidad de funcionamiento: 0.12sec/60 ° (sin carga esté funcionando)
- Temperatura de almacenamiento:-15C ~ 55C
- Dimensión: 29x11.7x30.2mm

**Electrical Specification (Function of the Performance) :**

No.	Item	4.8V	6.0V
4-1	Operating speed (at no load)	0.12 sec/60°	0.11 sec/60°
4-2	Running current (at no load)	160 mA	180 mA
4-3	Stall torque (at locked)	2.0 kg-cm	2.7 kg-cm
4-4	Stall current (at locked)	680 mA	800 mA
4-5	Idle current (at stopped)	4 mA	5 mA

Note: Item 4-2 definition is average value when the servo running with no load

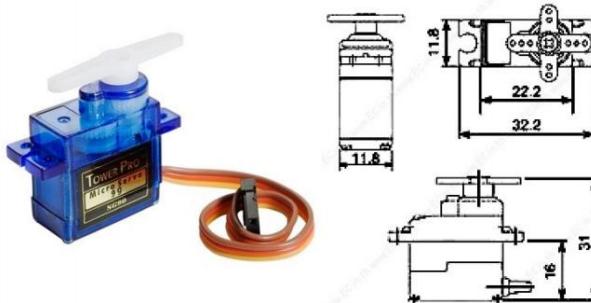
**Mechanical Specification :**

No.	Item	Specification
5-1	Overall Dimensions	See the drawing
5-2	Limit angle	180± 10°
5-3	Weight	17 ± 1 grams (without servo horn)
5-4	Connector wire gauge	# 26 PVC
5-5	Connector wire length	150 ± 5 mm
5-6	Horn gear spline	25T/Φ5.77
5-7	Horn type	Cross , Disk , Star , Bar
5-8	Reduction ratio	1/313

**Control Specification :**

No.	Item	Specification
6-1	Control system	Pulse width modification
6-2	Amplifier type	Analog controller
6-3	Operating travel	90° (when 1000→2000 μsec)
6-4	Neutral position	1500 μsec
6-5	Dead band width	7 μsec
6-6	Rotating direction	Clockwise (when 1500→2000 μsec)
6-7	Pulse width range	800→2200 μsec
6-8	Maximum travel	Approx 165°(when 800→2200 μsec)

✓ MINI SERVO tower pro sg90 9g



Especificaciones:

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf-cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μs
- Temperature range: 0 °C – 55 °C

Este servomotor se utilizará para las demás articulaciones del brazo Robótico siendo un total de 3 servos.

✓ SENSOR ULTRASONICO HC SR 04



Este sensor permitirá la detección de objetos sobre la cinta transportadora, se determinará la distancia y según esto se sabrá si hay o no presencia de un objeto.

Los módulos incluyen transmisores ultrasónicos, el receptor y el circuito de control.

Número de pines:

VCC: Alimentación +5V (4.5V min – 5.5V max)  
TRIG: Trigger entrada (input) del sensor (TTL)  
ECHO: Echo salida (output) del Sensor (TTL)  
GND

Corriente de reposo: < 2mA

Corriente de trabajo: 15mA

Ángulo de medición: 30º

Ángulo de medición efectivo: < 15º

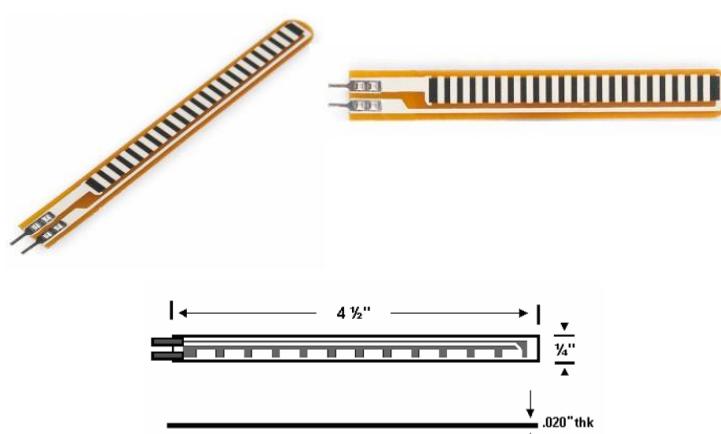
Detección de 2cm a 400cm o 1" a 13 pies (Sirve a más de 4m, pero el fabricante no garantiza una buena medición).

"Resolución" La precisión puede variar entre los 3mm o 0.3cm.

Dimensiones: 45mm x 20mm x 15mm

Frecuencia de trabajo: 40KHz

#### ✓ FLEX SENSOR



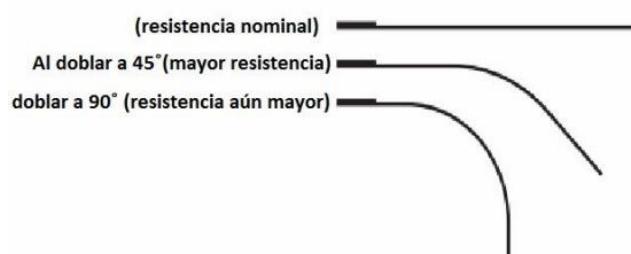
El Sensor Flex produce una resistencia variable en función del grado al que esté doblada.

Convierte la curvatura en distintos valores de resistencia eléctrica.

Son por lo general en la forma de una delgada tira de 5 cm de largo que varía en resistencia de aproximadamente 10 a 100 kOhms.

#### Características:

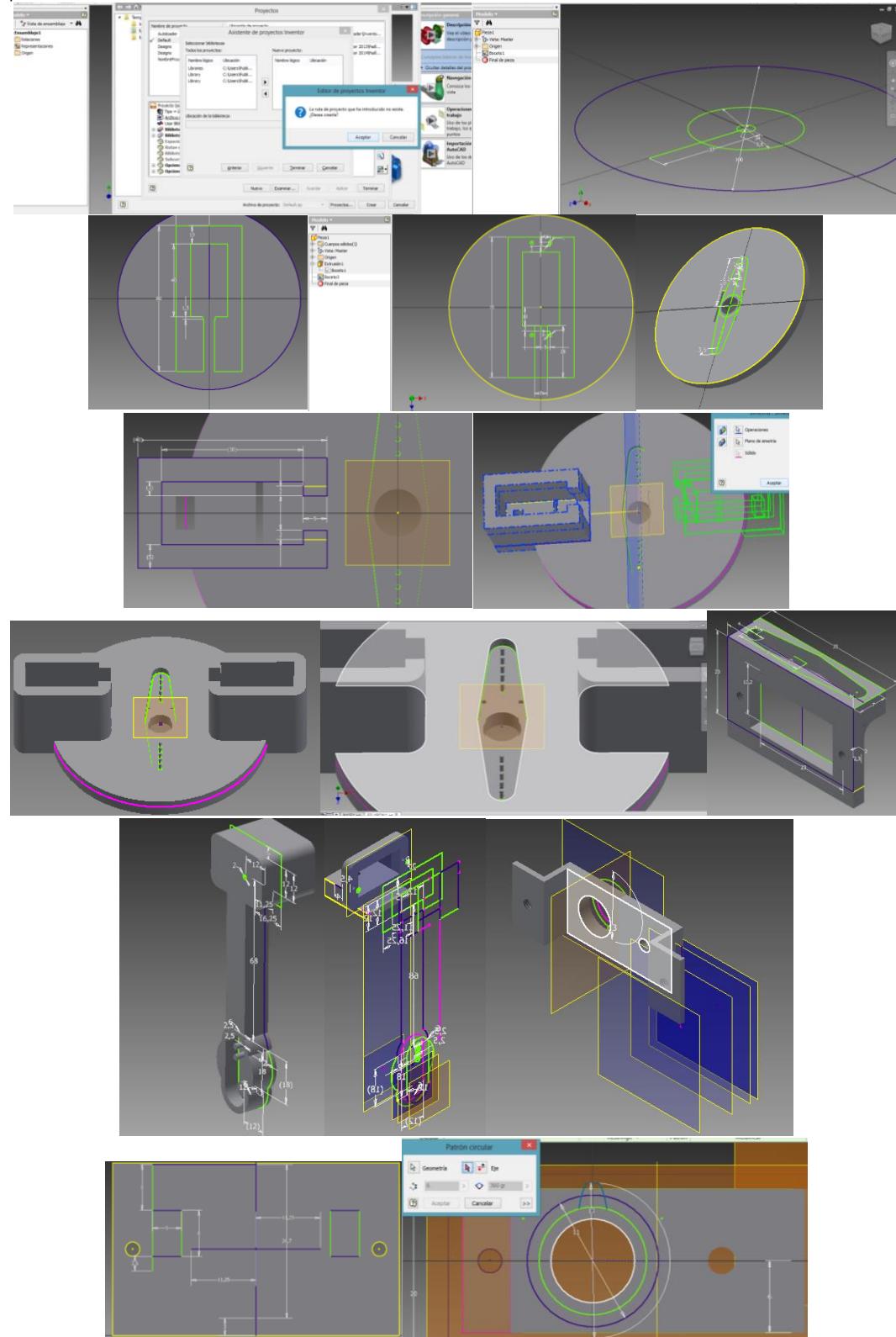
- ✓ Tolerancia de la Resistencia: ± 30%.
- ✓ Potencia nominal: 0,50 Volts continuos.
- ✓ La resistencia al no estar doblada es: 25K Ohms.
- ✓ Rango de la curva de la resistencia: 45K a 125K Ohms (dependiendo del radio de curvatura).
- ✓ Altura: 0.43 mm (0.017").
- ✓ Rango de temperatura: -35 ° C a +80 ° C.
- ✓ Voltaje: 5 a 12 Volts.

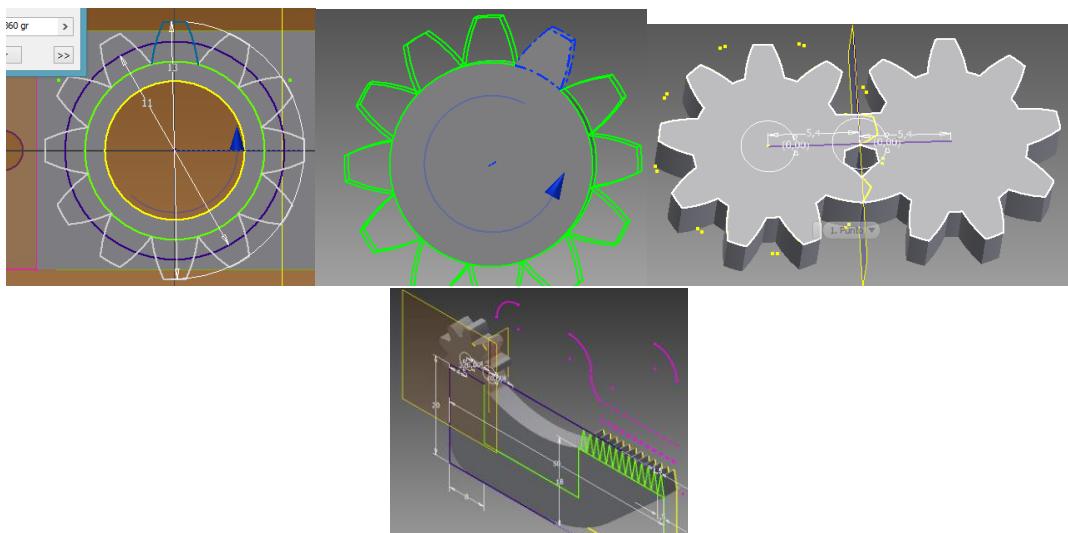


### 3. CONSTRUCCIÓN y DISEÑO EN INVENTOR

#### 3.1. BOCETOS INICIALES

A continuación se muestra los bocetos iniciales realizados en el inventor de cada una de las piezas para el brazo robótico, se empezó por el diseño de la base y en función de este diseño en base a planos paralelos se prosiguió con el diseño de las demás piezas para hacer un ajuste perfecto de las medidas en el espacio.



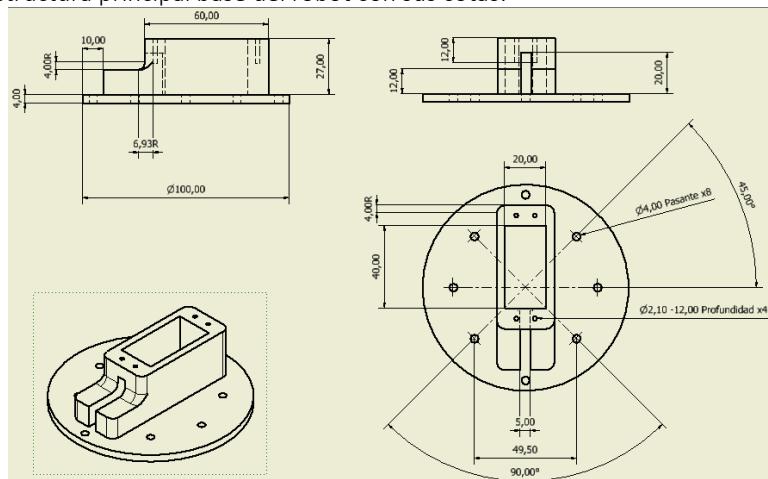


Cada boceto fue creado en base a las medidas de cada uno de los elementos que se iban a utilizar, están diseñados para el modelo y tamaño exacto de los servomotores empleados así como de los demás elementos de sujeción como son las piezas de acoplamiento para cada uno de los ejes acoplados a los miniservos.

Los piñones fueron creados de igual manera con medidas para ajustarse al diseño del brazo robótico.

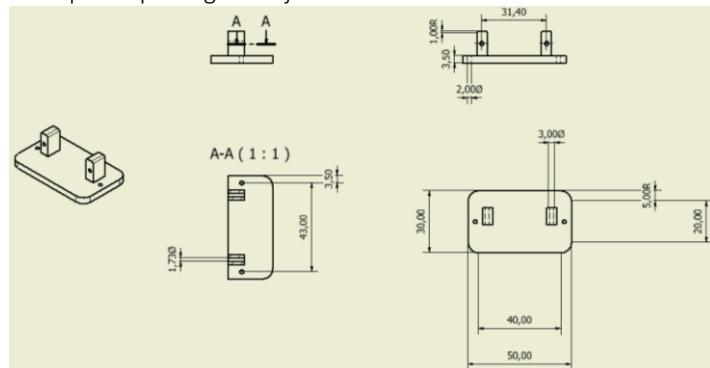
### 3.2. PLANO BASE DEL ROBOT:

Plano de la estructura principal base del robot con sus cotas:



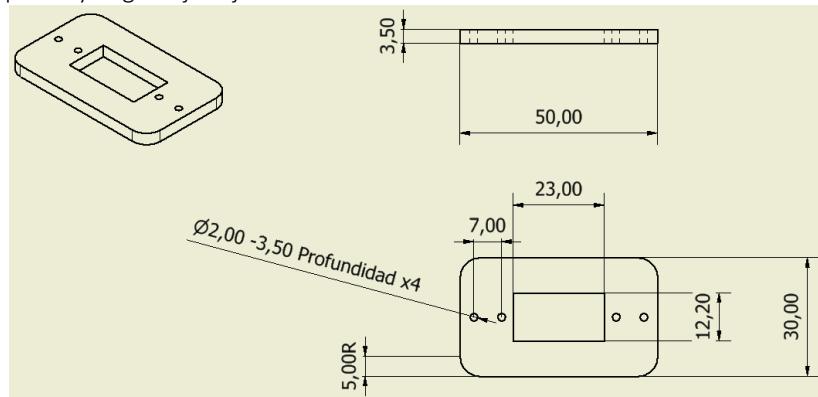
### 3.3. PLANO BASE DE LA PINZA DEL ROBOT:

En el extremo del robot se encuentra esta estructura de sujeción en la que se colocarán el servo y las piezas que sostendrán el conjunto de la pinza que cogerá objetos.



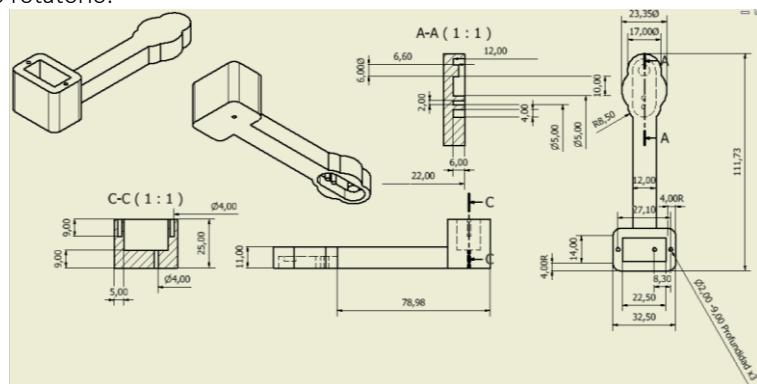
### 3.4. PLANO BASE DE LA PINZA SUJECCIÓN:

En el mismo extremo, explicado en el apartado anterior, se colocará esta pieza para poder mejorar la fijación del sistema de servo, pinzas y engranajes fijos a los anteriores.



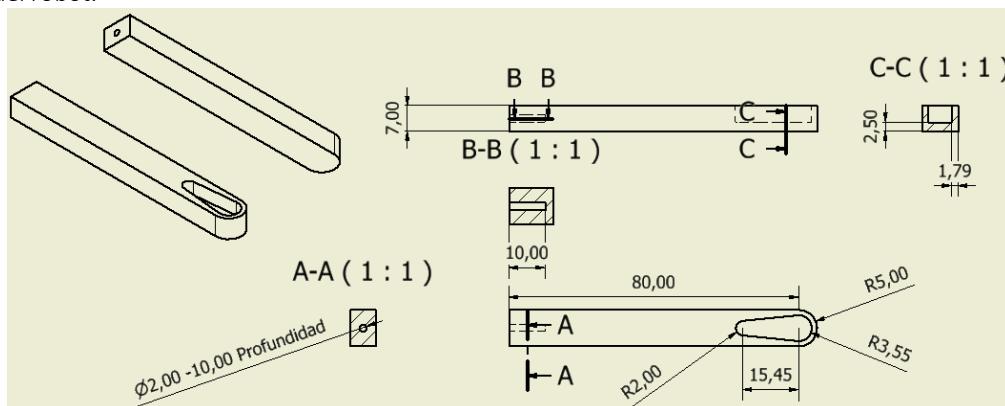
### 3.5. PLANO BRAZO PRIMERO SERVO EXTREMO SUPERIOR:

En esta pieza se define el hueco rectangular para poder colocar el servo que permitirá el movimiento de la parte siguiente del brazo en cuyo extremo se hallaría la pinza. Este brazo estará siendo movido por un servo colocado en la parte de la base del mismo robot. Obsérvese que la estructura posee diferentes vistas para poder definir la abertura redondeada en donde irá una pieza auxiliar que junto con el servo permitirá el movimiento rotatorio.



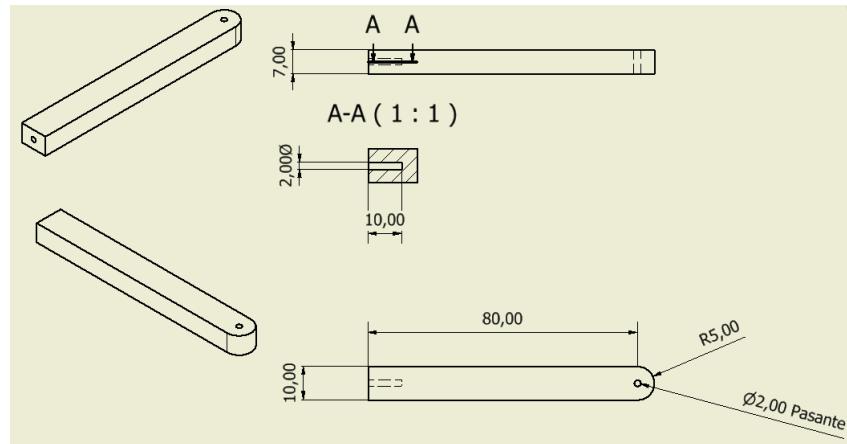
### 3.6. BRAZO SUPERIOR, UNIÓN CON EL EXTREMO DE LA BASE DE LA PINZA:

Esta estructura se encuentra en la parte superior del brazo que es nexo entre la parte inferior del mismo brazo y la pinza del robot.



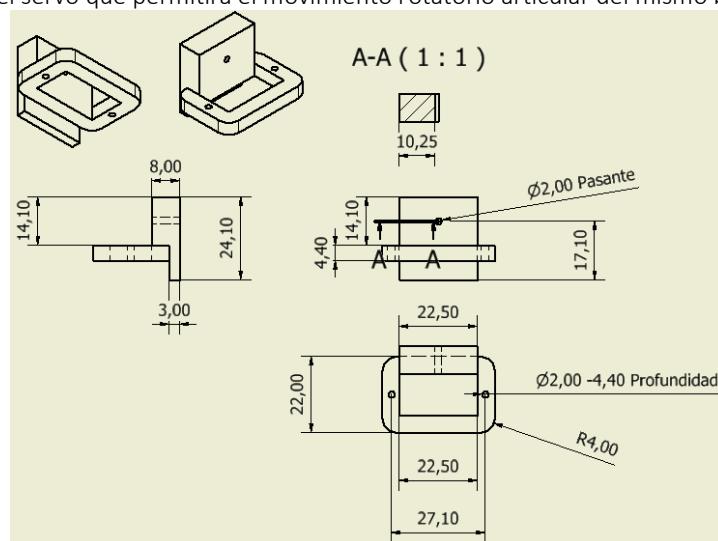
### 3.7. BRAZO SUPERIOR ,UNIÓN CON EL EXTREMO DE LA BASE DE LA PINZA PARTE II:

Esta estructura está colocada paralelamente a la anterior formando entonces, el conjunto del brazo de la zona superior.



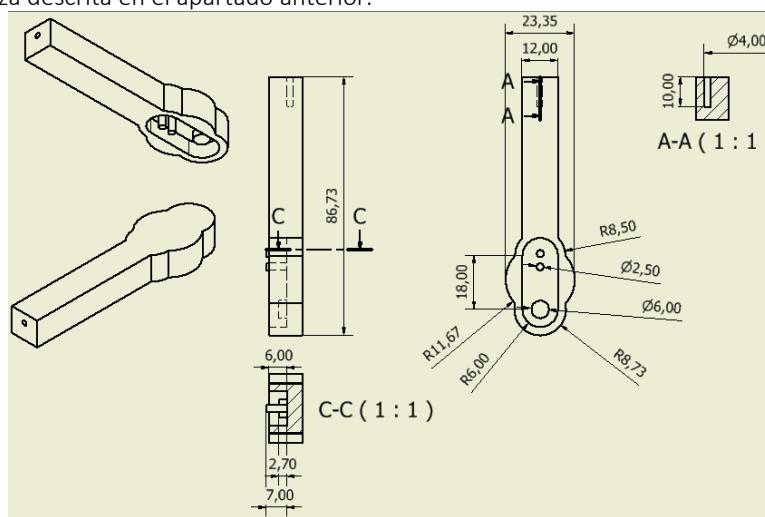
### 3.8. EXTREMO BRAZO SUSTENTADOR DE LA ARTICULACIÓN:

Esta estructura va colocada sobre uno de los extremos de la zona inferior del brazo en donde irá colocada en el rectángulo ahuecado el servo que permitirá el movimiento rotatorio articular del mismo brazo.



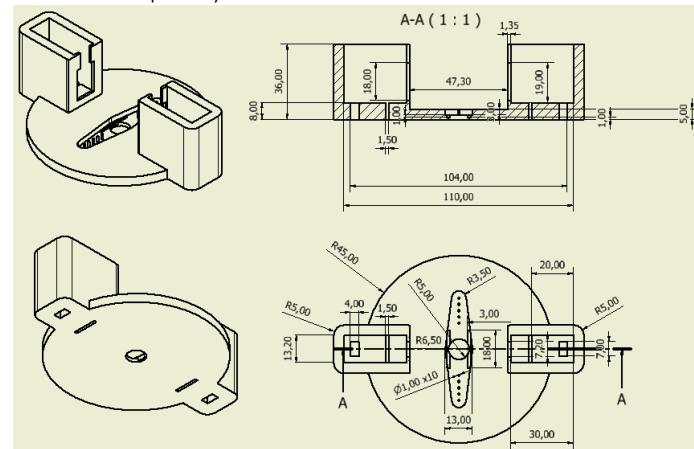
### 3.9. BRAZO PRIMERO SERVO EXTREMO SUPERIOR PARTE II:

Esta es la otra parte de la zona inferior del brazo en la que se observa el encaje idéntico a la estructura paralela que se mencionó en apartados anteriores, sin embargo en este caso se observa un taladro en un extremo para poder encajar la pieza descrita en el apartado anterior.



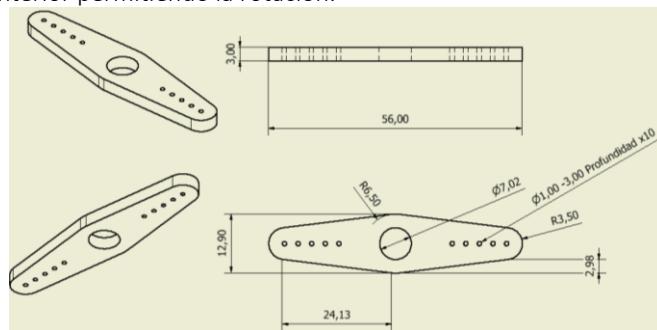
### 3.10. PLANO ESLABÓN SOBRE BASE DEL ROBOT:

Para que el sistema del robot pueda funcionar se precisa de una estructura como la descrita a continuación en la que se puede observar dos cubículos huecos en donde irían colocados los sendos servos que permitían el movimiento rotatorio del sistema brazo del robot. En su base se encuentra una abertura redondeada elíptica en la que iría insertada una pieza con un orificio circular cilíndrica para que quede insertado un servo que permitiría el movimiento rotatorio de la misma pieza y con ello el resto del brazo del robot.



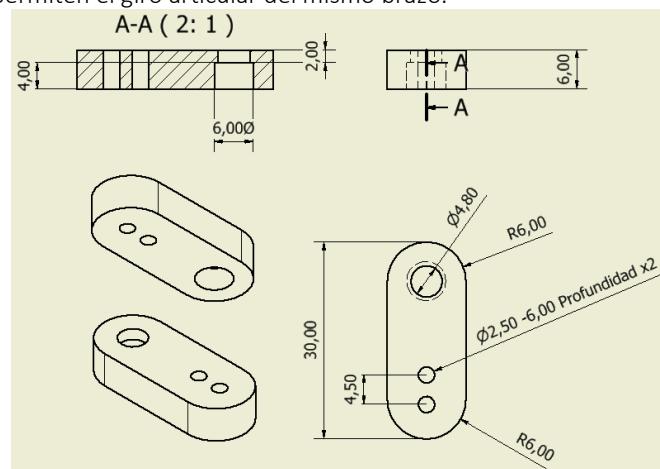
### 3.11. EXTREMO ROTATORIO DE LA BASE:

Esta estructura encaja con la anterior haciendo que exista juego entre el servo que se conectaría con ésta y la descrita en el apartado anterior permitiendo la rotación.



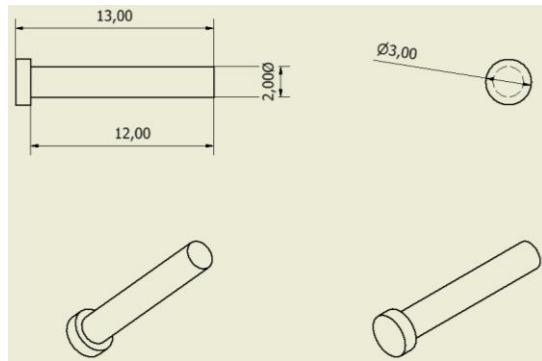
### 3.12. NEXO ENTRE LA ZONA BRAZO INFERIOR Y SERVOS PARALELOS DE BASE:

Esta pieza permite la conexión entre la zona inferior del brazo y la estructura que contiene a sendos servos que colocados enfrentados, permiten el giro articular del mismo brazo.



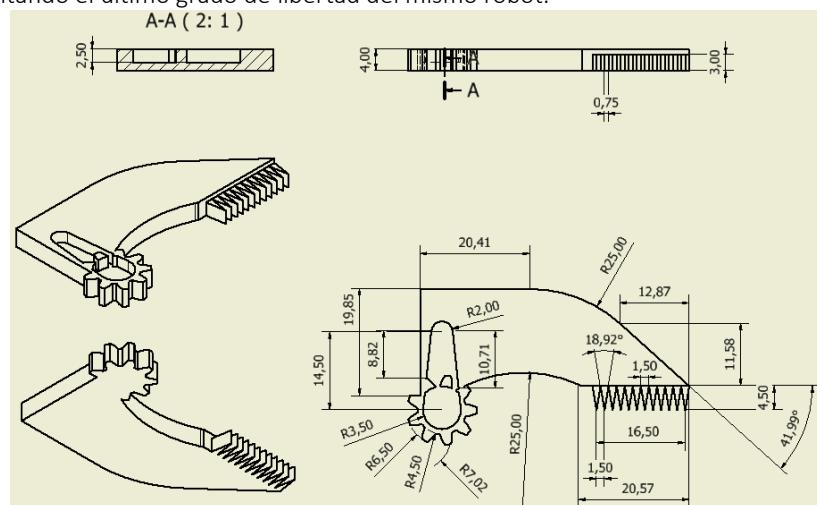
### 3.13. ENGANCHE DE PIEZAS CON EL SERVO IMPLICADO:

La siguiente pieza es de unión y permite la cohesión entre una pieza móvil y un servo, al no ser de rosca permite una rotación y no una fijación.



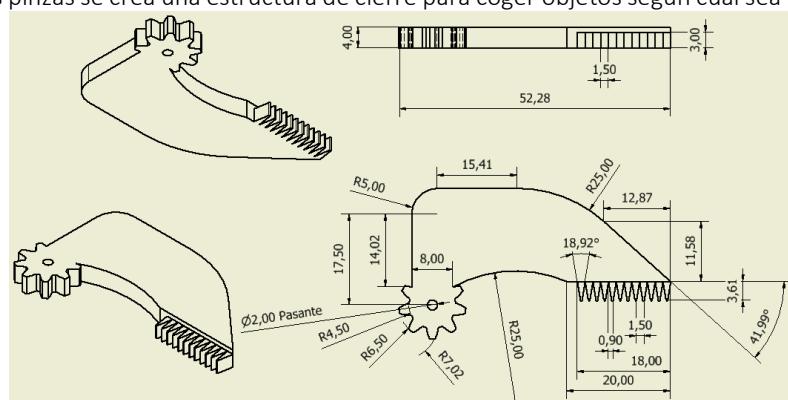
### 3.14. PLANO PINZA PRIMERA EXTREMO GUIADO POR EL SERVO:

Esta pieza parte de la pinza permite el giro con el servo, el cual se colocará en el hueco del engranaje permitiendo la rotación aumentando el último grado de libertad del mismo robot.



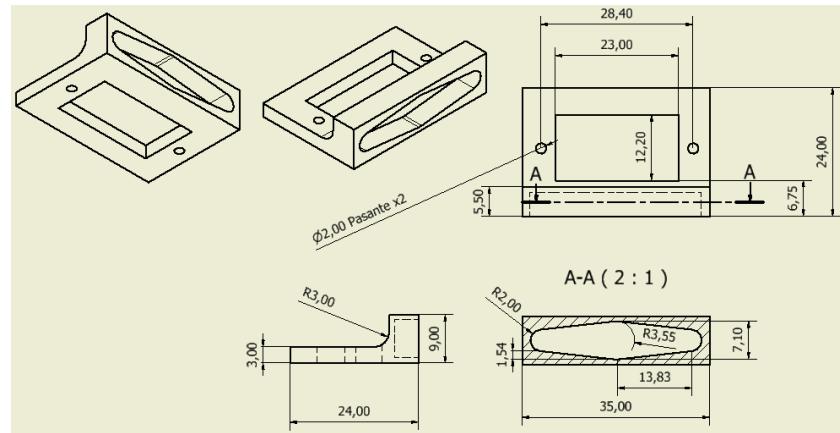
### 3.15. PLANO PINZA SEGUNDA EXTREMO GUIADO POR UN PERNO:

Esta pieza se moverá a causa del otro extremo de la pinza que a su vez su movimiento se verá afectado por un servo. Entre ambas pinzas se crea una estructura de cierre para coger objetos según cual sea la posición del robot.



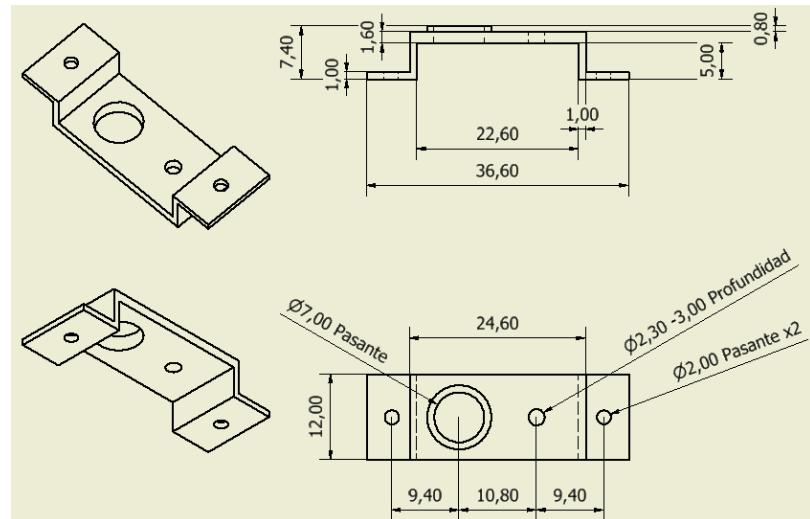
### 3.16. PIEZA DE SOPORTE Y ROTACIÓN DEL EXTREMO DEL ROBOT (ORIENTACION DE LA PINZA):

Esta pieza tiene la función de rotar conteniendo al sistema de pinzas y soportes observándose el hueco rectangular en donde irá encajado el servo que permita el cierre o apertura de las respectivas pinzas con sendos agujeros para sujetar el soporte que cargue con el servo insertado.



### 3.17. SOPORTE DE SUJECCIÓN DEL SERVO DE LAS PINZAS:

La siguiente pieza tiene la función de sujeción del servo cuyo rotor saliente encajaría a juego por el hueco cilíndrico de diámetro más amplio en donde se encontraría con las pinzas ajustando un cierre o una apertura según el giro.

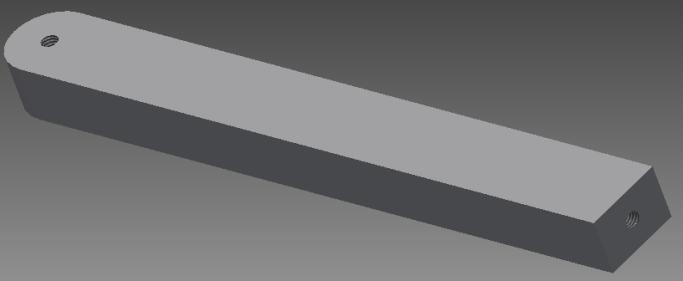
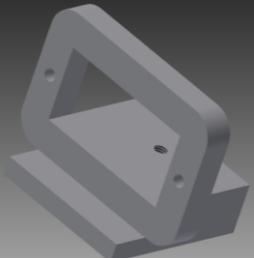
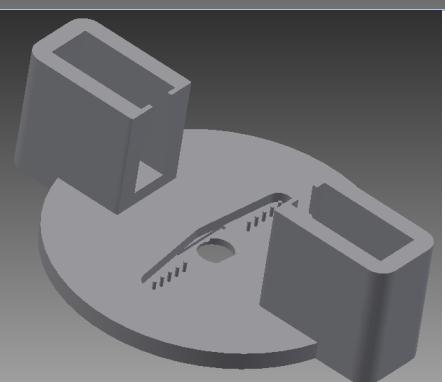
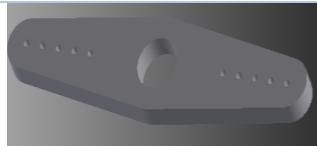
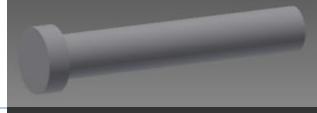
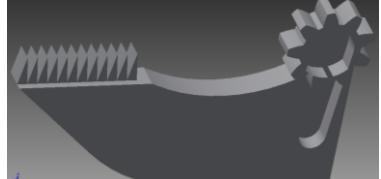


## 4. IMÁGENES DE LAS PIEZAS EN INVENTOR:

### 3.1. TABLA DE PIEZAS DISEÑADAS

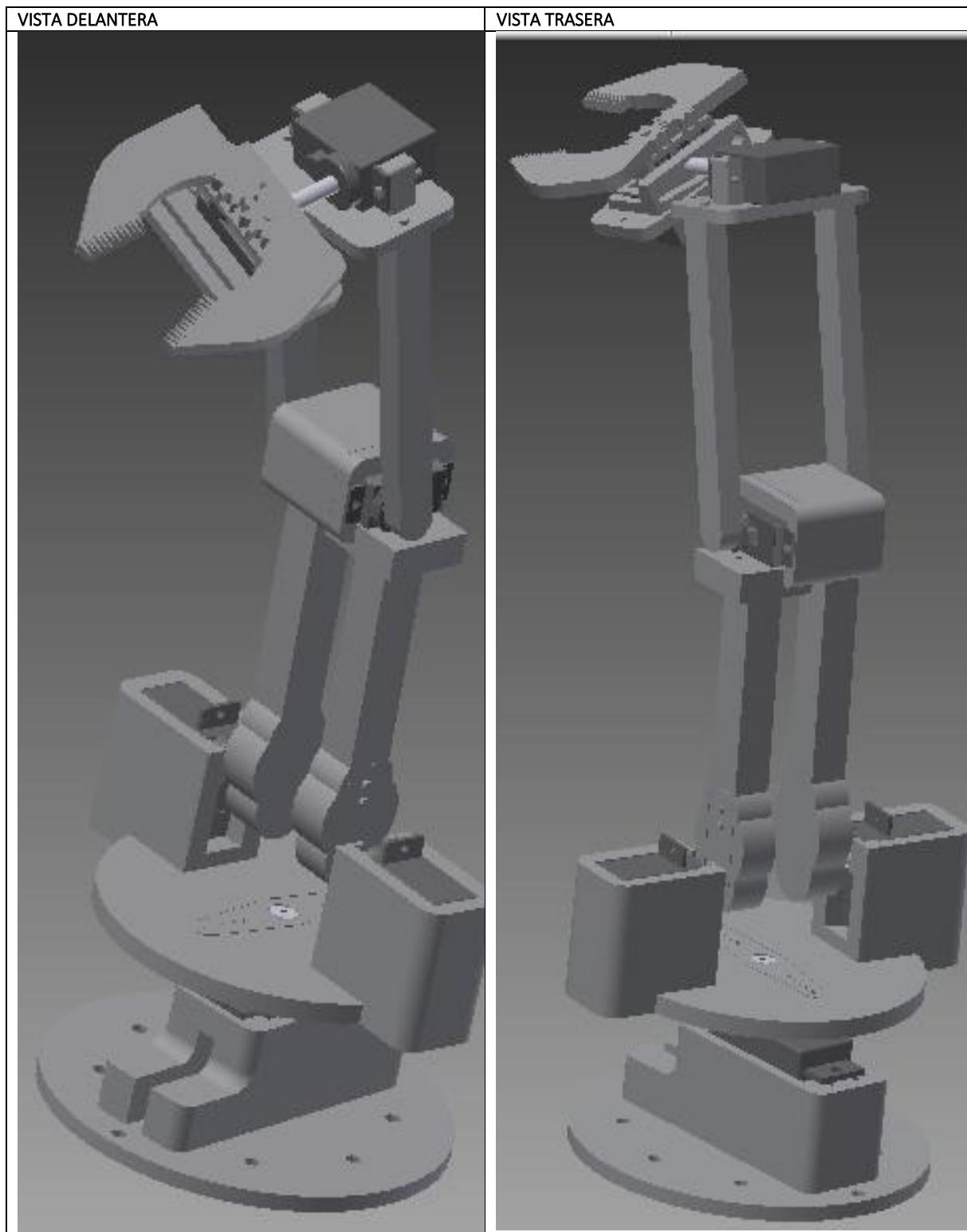
A continuación, se van a representar en una tabla la serie de piezas del robot en cuestión construido:

BASE DEL ROBOT		
BASE DE LA PINZA DEL ROBOT		
BASE DE LA PINZA SUJECCIÓN		
BRAZO PRIMERO SERVO EXTREMO SUPERIOR		
BRAZO SUPERIOR UNIÓN CON EL EXTREMO DE LA PINZA		

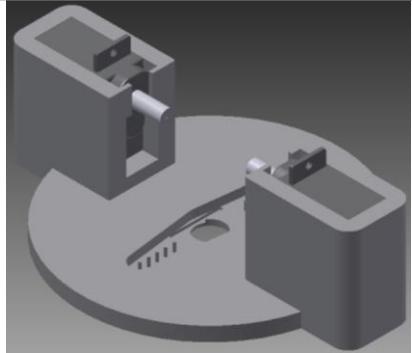
BRAZO SUPERIOR UNIÓN CON EL EXTREMO DE LA PINZA PARTE II		
EXTREMO BRAZO SUSTENTADOR DE LA ARTICULACIÓN		
BRAZO PRIMERO SERVO EXTREMO SUPERIOR PARTE II		
ESLABÓN SOBRE BASE DEL ROBOT		
EXTREMO ROTATORIO DE LA BASE		
NEXO ENTRE LA ZONA BRAZO INFERIOR Y SERVOS PARALELOS DE BASE		
PERNO DE ENGANCHE DE PIEZAS CON EL SERVO IMPLICADO		
PINZA PRIMERA EXTREMO GUIADO POR EL SERVO		

PINZA SEGUNDA EXTREMO GUIADO POR UN PERNO		
PIEZA SOPORTE Y ROTACIÓN DEL EXTREMO DEL ROBOT		
SOPORTE DE SUJECCIÓN DEL SERVO DE LAS PINZAS		
MINI SERVO UTILIZADO EN LAS ARTICULACIONES, MOVIMIENTOS ROTATORIOS Y DE APERTURA Y CIERRE DE LA PIEZA DEL EXTREMO DEL ROBOT		
TORNILLO PARA FIJACIONES DE ELEMENTOS DEL BRAZO DEL ROBOT		

**3.2. IMAGEN DEL ENSAMBLAJE FINAL DEL ROBOT COMPLETO:**



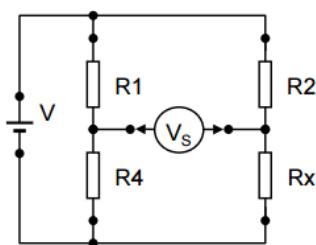
### 3.3. OTROS ENSAMBLAJES DE CONSTRUCCIÓN INICIAL:

<p><u>Colocación de servos en la base robot</u></p> <p>En esta imagen se observan dos mini servos colocados perfectamente en las ranuras específicas que se observan en la imagen en la pieza de sujeción del sistema robot, obsérvese la necesaria precisión de una búsqueda del más aproximado servo en referencia a las exigencias del mecanismo global y a las dimensiones que están limitadas por la pieza que los porta</p>		
<p><u>Colocación del componente rotatorio en el servo</u></p> <p>En la imagen se observa un mini servo conectado al componente rotatorio que juntos irán ensamblados a la pieza que se observa encima (con los dos mini servos expuestos)</p>		

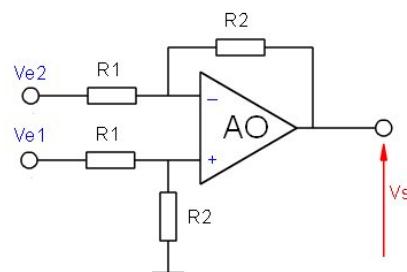
## 5. ACONDICIONAMIENTO DE LOS SENSORES

Los sensores flex fueron acondicionados con el circuito siguiente:

En primer lugar un puente de Wheatstone para tener una señal diferencial en función de la variación de la resistencia del sensor:



Seguido de un amplificador diferencial para amplificar la tensión diferencial que sale del puente anterior.



Siendo Rx la entrada del sensor, se utilizó este circuito para obtener un valor de 0 V cuando el sensor no está flexionado y otra tensión cuando se encuentra doblado, aproximadamente con los cálculos realizados se obtuvo:

$$R3=10K$$

$$R4=10 K$$

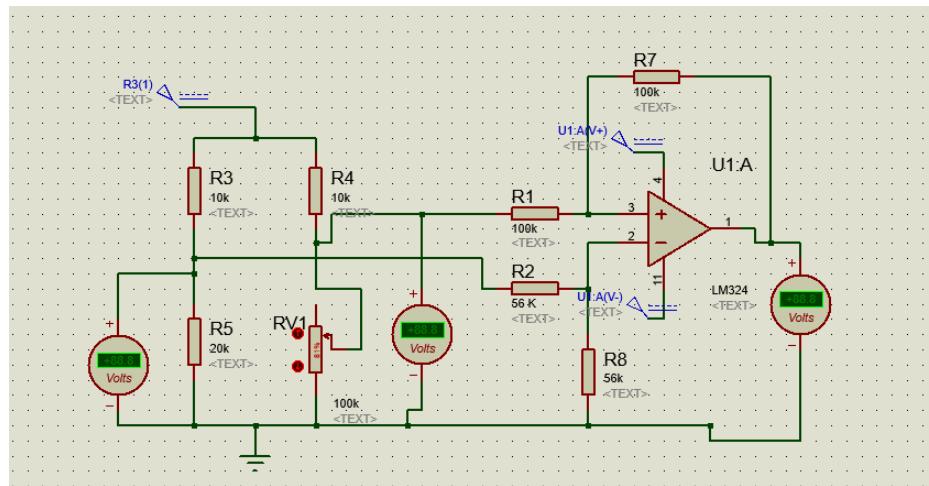
$$R5=20 K \text{ (La misma resistencia del sensor sin flexión)}$$

Con estos valores se tienen tensiones diferenciales entre 0 y 2,3 voltios siendo necesario un amplificador diferencial para poder tener a la salida una tensión entre 0 y 5 V. La salida diferencial de este puente de Wheatstone pasa entonces a un amplificador diferencial como se puede ver en el circuito siguiente.

Este amplificador tendrá una ganancia de aproximadamente 1,77 para lo cual las resistencias utilizadas son de

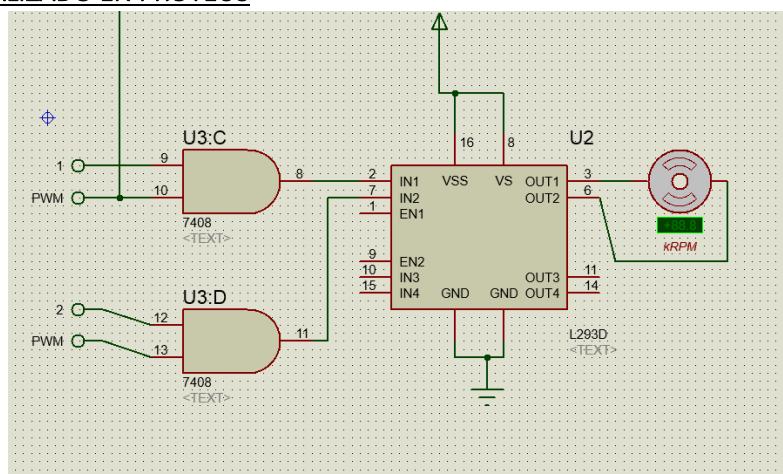
$$R1=R2=100 k$$

$$R7=R8= 56 K.$$



## 6. CONEXIÓN DEL DRIVER PARA LOS MOTORES DC

DIAGRAMA REALIZADO EN PROTEUS



Claramente se puede ver cómo será la lógica de control para el motor la PWM generada es la misma ya sea en el uno u otro sentido, lo que nos permitirá elegir o uno u otro sentido será nuestra compuerta And que según las señales que entran a 1 o 2 envían su salida a las entradas del puente H y este a su vez envía las señales directas a los motores.

Los pines de control para los motores son:

**MOTOR1 :**

**PIN 22:** ENTRADA MEDIANTE DIPSWITCH PARA EL SENTIDO DE GIRO

**PINES 24 Y 25 :** INTERRUPCIONES DE AUMENTO O DISMINUCIÓN DE LA PWM, ED DE VELOCIDAD DE GIRO

**PIN 11:** POR EL CUAL SALE LA PWM1 DE CONTROL DE VELOCIDAD

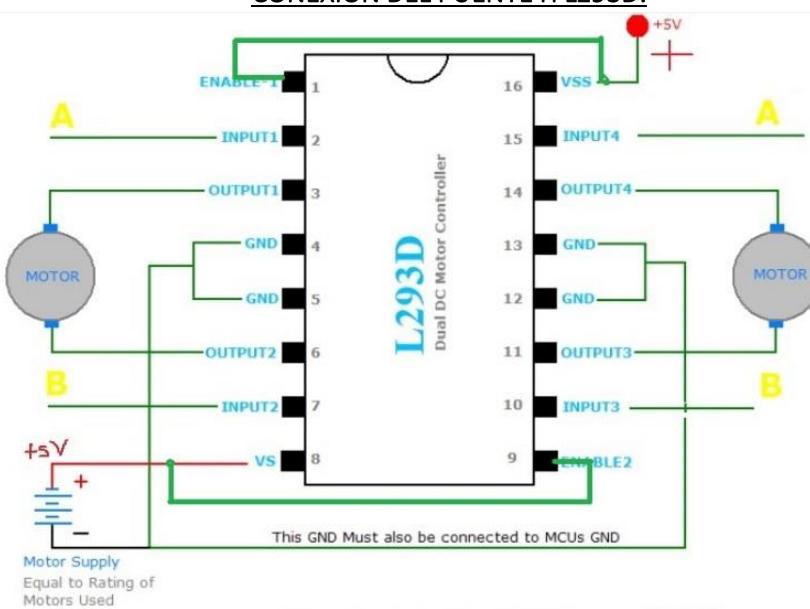
**MOTOR2 :**

**PIN 23:** ENTRADA MEDIANTE DIPSWITCH PARA EL SENTIDO DE GIRO

**PINES 26 Y 27 :** INTERRUPCIONES DE AUMENTO O DISMINUCIÓN DE LA PWM, ED DE VELOCIDAD DE GIRO

**PIN 12:** POR EL CUAL SALE LA PWM1 DE CONTROL DE VELOCIDAD

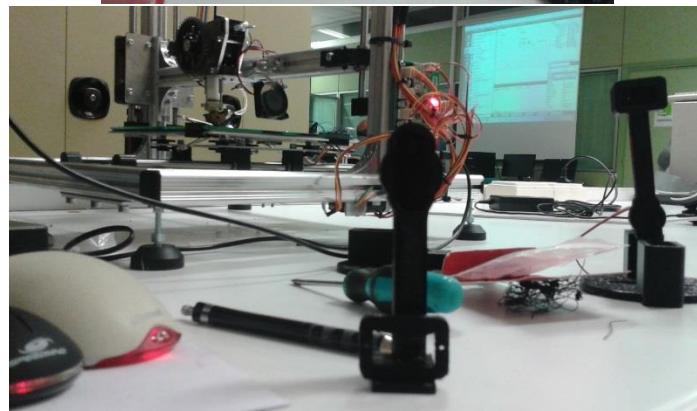
CONEXIÓN DEL PUENTE H L293D:

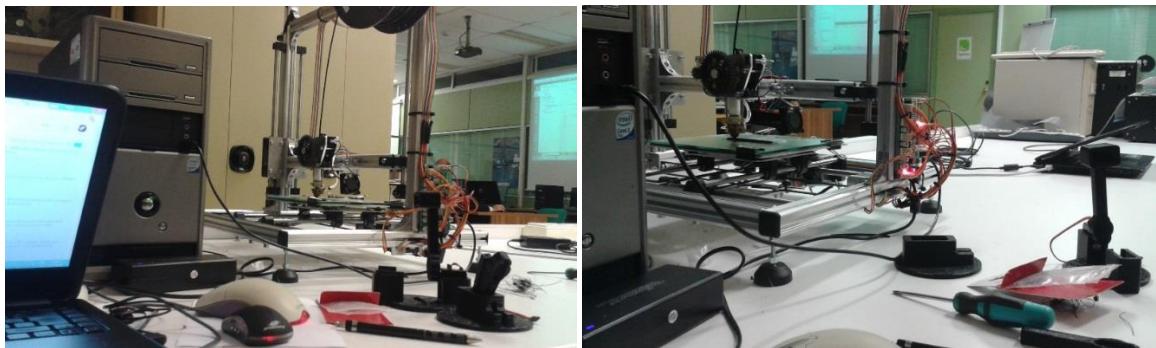


**Motor Controller Using L293D**

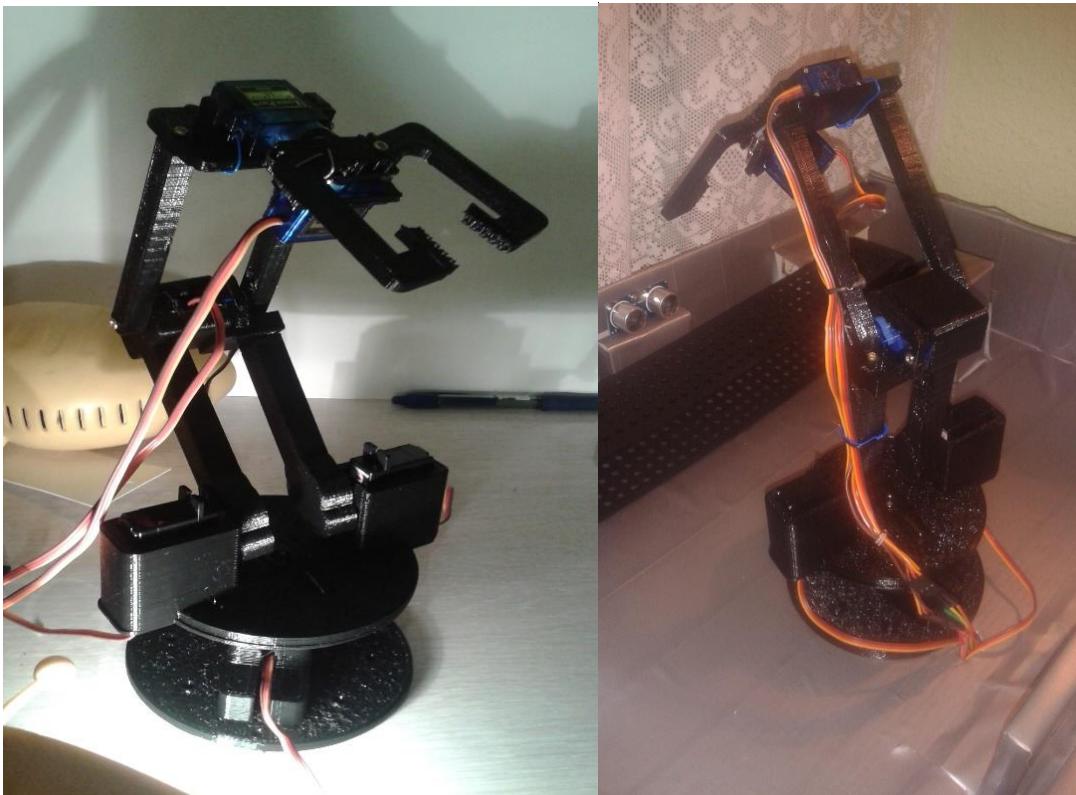
## 7. FOTOGRAFÍAS DE LA IMPRESIÓN 3D:

Se muestran a continuación las fotografías de la impresión de las piezas en la impresora 3D.



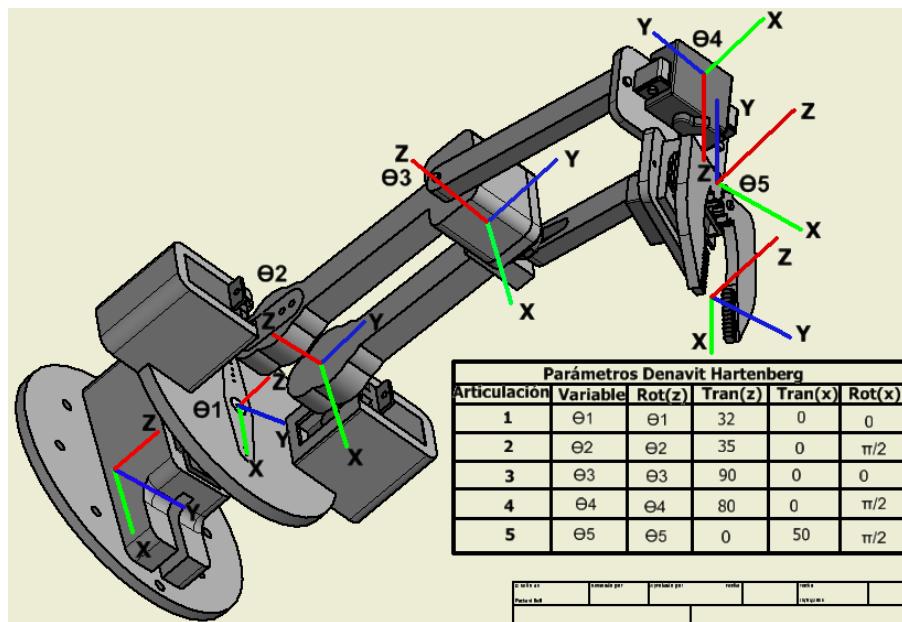


En esta fotografía se observa el montaje finalizado del proyecto del robot con las piezas impresas en 3 dimensiones, en una de las cuales se ve el robot solo, y en la otra se lo ve sobre la maqueta final del proyecto.



## 8. CÁLCULOS DE LA CINEMÁTICA Y DINÁMICA DEL ROBOT:

### 8.1. ESTUDIO CINEMÁTICO DEL ROBOT:



En éste apartado se pretende analizar la cinemática del robot fabricado en donde se ha empleado el método de Denavit Hartenberg de cara al estudio del movimiento y posición del TCP0 del robot (que se haya en la distancia media entre las dos pinzas del extremo del brazo robot)

Se han colocado diferentes sistemas de coordenadas para iniciar el análisis obteniendo posteriormente los parámetros de Denavit Hartenberg y construir las siguientes matrices homogéneas que son paso para poder llegar a las ecuaciones de la posición del extremo del robot:

NOTA: la fuente de obtención de las matrices mencionadas pertenece al software Matlab R2015.

A1 =

$$\begin{bmatrix} \cos(a), & -\sin(a), & 0, & 0 \\ \sin(a), & \cos(a), & 0, & 0 \\ 0, & 0, & 1, & 32 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Matriz A1 para el ángulo 'a'

Matriz A3 para el ángulo 'c'

A4 =

$$\begin{bmatrix} \cos(d), & 0, & \sin(d), & 0 \\ \sin(d), & 0, & -\cos(d), & 0 \\ 0, & 1, & 0, & 80 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Matriz A4 para el ángulo 'd'

A2 =

$$\begin{bmatrix} \cos(b), & 0, & \sin(b), & 0 \\ \sin(b), & 0, & -\cos(b), & 0 \\ 0, & 1, & 0, & 35 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Matriz A2 para el ángulo 'b'

A5 =

$$\begin{bmatrix} \cos(e), & 0, & \sin(e), & 50*\cos(e) \\ \sin(e), & 0, & -\cos(e), & 50*\sin(e) \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Matriz A5 para el ángulo 'e'

A3 =

$$\begin{bmatrix} \cos(c), & -\sin(c), & 0, & 0 \\ \sin(c), & \cos(c), & 0, & 0 \\ 0, & 0, & 1, & 90 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

Tras haber obtenido las matrices homogéneas se pasará a obtener la matriz T que sería el producto de las cinco matrices anteriores conociéndose por lo tanto la posición del TCPO, extremo de la pinza, del robot que son los tres primeros componentes de la cuarta columna:

$$X = 50\sin(e)(\cos(a)\sin(b) + \cos(b)\sin(a)) - 50\cos(e)(\cos(c)\cos(d)(\sin(a)\sin(b) - \cos(a)\cos(b)) - \sin(c)\sin(d)(\sin(a)\sin(b) - \cos(a)\cos(b))) + 170\cos(a)\sin(b) + 170\cos(b)\sin(a)$$

$$Y = 170\sin(a)\sin(b) + 50\cos(e)(\cos(c)\cos(d)(\cos(a)\sin(b) + \cos(b)\sin(a)) - \sin(c)\sin(d)(\cos(a)\sin(b) + \cos(b)\sin(a))) + 50\sin(e)(\sin(a)\sin(b) - \cos(a)\cos(b)) - 170\cos(a)\cos(b)$$

$$Z = 50\cos(e)(\cos(c)\sin(d) + \cos(d)\sin(c)) + 67$$

Se pretendió conseguir la matriz Jacobiana 'J' para analizar las singularidades del robot, puntos inalcanzables en su zona de trabajo, pero al obtenerla, partiendo de que la misma es muy extensa, no poseía la propiedad de ser cuadrada de manera que no se pudo analizar las posibles singularidades que tuviera intentando resolver su determinante.

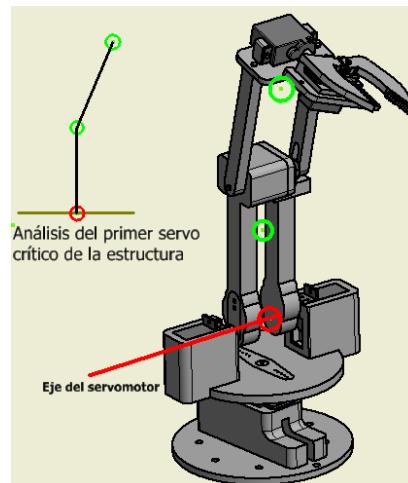
Esto se debe a que en un primer análisis se conoció que el robot poseía más grados de libertad que dimensiones en donde trabajaba; 5 grados de libertad frente > 3 dimensiones de zona de trabajo, de manera que quedaría redundante el mismo sistema completo debido a la añadidura de 2 grados más de libertad.

## **8.2. ESTUDIO DINÁMICO DEL ROBOT:**

Los servomotores que tienen una funcionalidad más crítica son los que se analizarán en este apartado, ya que soportan el peso inercial basado en giros, sin embargo, los demás meramente generan giros sin mera preocupación acerca de los posibles problemas de fuerzas rotativas adversas. Uno de los primeros servos que se analizará es el mostrado en la figura que en realidad es una pareja de servos cerca de la base para poder mover el brazo entero que soportan.

Para poder comprender el modelo dinámico se ha simplificado a un modelo bidimensional con los puntos verdes referentes a los centros de gravedad de cada zona concreta ya que éstos representan las masas totales de cada conjunto de piezas involucradas en el movimiento;

- El círculo verde inferior es centro de gravedad de únicamente de sendos brazos en cuyos extremos sustentan el servomotor del centro de la estructura.
- El círculo verde superior hace referencia al centro de gravedad que existe en la estructura completa por encima del servomotor del centro del brazo.



### Estudio del par de servos próximos a la base

Así pues, partiendo del modelo bidimensional se extraerá el modelo cinemático directo de ambos círculos verdes a partir de los cuales se verá qué momento ejercen sobre el círculo rojo en donde se encontraría el eje del servomotor (Nota: sistema de referencia; Z vertical en el modelo bidimensional y X horizontal en el mismo):

Círculo verde inferior (ángulo 'a' entre la línea vertical y la base horizontal):

$$X = 0,07123 * \cos(a); \\ Z = 0,07123 * \sin(a);$$

Círculo verde superior (ángulo 'b' entre una línea imaginaria prolongada de la línea que unía ambos círculos verdes y la misma línea que une ambos círculos verdes):

$$X = 0,07123 * \cos(a) + L * \cos(a) * \cos(b); \\ Z = 0,07123 * \sin(a) + L * \sin(a) * \sin(b);$$

L longitud entre amos puntos verdes (colocar valor distancia)

Ahora se procederá a calcular el momento que debe contrarrestar el servomotor en función del ángulo que se tenga de 'a' y 'b':

Aplicando la regla de la mano derecha entorno al eje del servo, se tiene que:

$$\vec{M} = \vec{r} \wedge \vec{F} = \vec{r} \wedge (\mathbf{m} * \vec{a}) = \vec{r} \wedge \left( \mathbf{m} * \frac{\partial \vec{V}}{\partial t} \right) = \vec{r} \wedge \left( \mathbf{m} * \frac{\partial^2 \vec{P}}{\partial^2 t} \right)$$

- Para el primer círculo verde, se tiene:

$$\begin{aligned} \vec{P} &= (0,07123 * \cos(a))\vec{i} + (0,07123 * \sin(a))\vec{k} \\ \vec{V} &= (-0,07123 * \sin(a))\vec{i} + (0,07123 * \cos(a))\vec{k} \\ \vec{a} &= (-0,07123 * \cos(a))\vec{i} + (-0,07123 * \sin(a))\vec{k} + (-9,8)\vec{k} \\ \vec{M} &= m * \begin{vmatrix} i & j & k \\ \cos(a) & 0 & 0 \\ -0,07123 * \cos(a) & 0 & -0,07123 * \sin(a) - 9,8 \end{vmatrix} = \\ &= -m * ((-0,07123 * \sin(a) - 9,8) * \cos(a))\vec{j} \\ |\vec{M}| &= 0,07123 * m * (\sin(a) + 9,8) \cos(a) \end{aligned}$$

- Para el segundo círculo verde, se tiene:

$$\begin{aligned} \vec{P} &= (0,07123 * \cos(a) + L * \cos(a) * \cos(b))\vec{i} + (71,23 * \sin(a) + L * \sin(a) * \sin(b))\vec{k} \\ \vec{V} &= (-0,07123 * \sin(a) - L * \sin(a) * \cos(b) - L * \cos(a) * \sin(b))\vec{i} + (0,07123 * \cos(a) + L * \cos(a) * \sin(b) + L * \sin(a) * \cos(b))\vec{k} \\ \vec{a} &= (-0,07123 * \cos(a) - L * \cos(a) * \cos(b) + L * \sin(a) * \sin(b) + L * \sin(a) * \sin(b) - L * \cos(a) * \cos(b))\vec{i} \\ &\quad + (-0,07123 * \sin(a) - L * \sin(a) * \sin(b) + L * \cos(a) * \cos(b) + L * \cos(a) * \cos(b) - L * \sin(a) * \sin(b))\vec{k} + (-9,8)\vec{k} = (a1)\vec{i} + (a2 - 9,8)\vec{k} \\ \vec{M} &= m * \begin{vmatrix} i & j & k \\ 0,07123 * \cos(a) + L * \cos(a) * \cos(b) & 0 & 0,07123 * \sin(a) + L * \sin(a) * \sin(b) \\ a1 & 0 & a2 - 9,8 \end{vmatrix} \end{aligned}$$

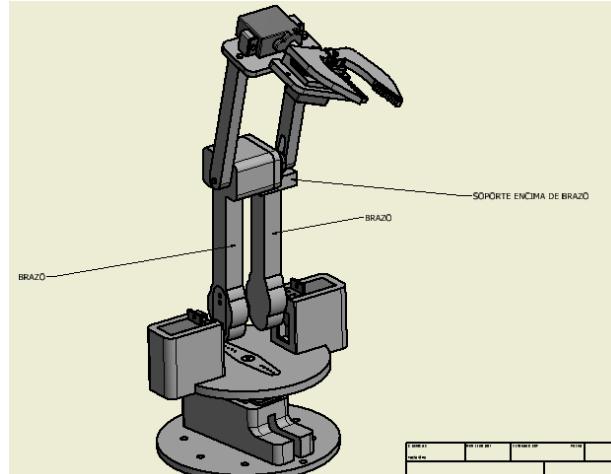
$$|\vec{M}| = -m * (((0,07123 * \cos(a) + L * \cos(a) * \cos(b)) * (0,07123 * \sin(a) - 2 * L * \cos(a) * \cos(b) + 2 * L * \sin(a) * \sin(b) + 9,8) - (0,07123 * \sin(a) + L * \sin(a) * \sin(b)) * (0,07123 * \cos(a) + 2 * L * \cos(a) * \cos(b) - 2 * L * \sin(a) * \sin(b))))$$

Para el momento máximo que debería soportar el servomotor analizado, se tendrían que los ángulos 'a' valiera 0 grados y 'b' valiera 0 grados posición tumbado el robot hacia adelante totalmente:

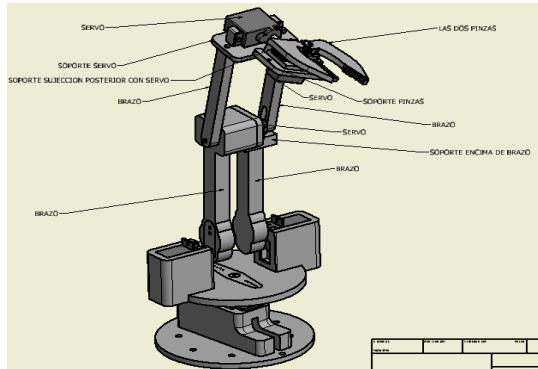
- Caso M1:

$$|\vec{M}| = 0,07123 * 9,8 * m = 0,698054 * m \text{ N} \cdot \text{m}$$

[BORRAR DESPUES DE CALCULAR, FALTA DEFINIR L, COMO LA LONGITUD APROXIMADA CON REGLA ENTRE LOS DOS CIRCULOS VERDES DEL DIBUJO Y m COMO LA MASA DEL CONJUNTO DE PIEZAS PARA ESTE MOMENTO DE FUERZA M1 DE LAS PIEZAS SEÑALADAS A CONTINUACION EN EL DIBUJO QUE TAMBIEN SE HA DE BORRAR DESPUES DEL CALCULO]



EN EL CASO DEL MOMENTO DE FUERZA M2 LA CONSTANTE m SERÁ LA MASA DE LA SUMA TOTAL DE LOS SIGUIENTES ELEMENTOS DEL DIBUJO SIGUIENTE QUE TAMBIEN DEBE SER BORRADO TRAS EL CÁLCULO:



- Caso M2:

$$|\vec{M_2}| = m * (2 * L - 9,8) * (0,07123 + 2 * L) N \cdot m$$

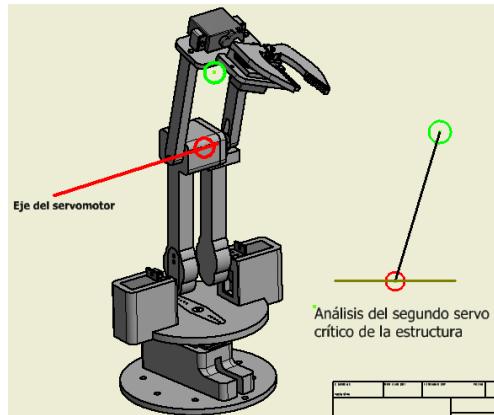
Y para el otro caso desfavorable se tendrían; ángulo 'a' un valor de 180 grados y 'b' de 0 grados en la que el robot estaría tumbado hacia atrás totalmente:

- Caso M1:

$$|\vec{M_1}| = -0,07123 * 9,8 * m = -0,698054 * m N \cdot m$$

- Caso M2:

$$|\vec{M_2}| = m * (2 * L + 9,8) * (0,07123 + 2 * L) N \cdot m$$



### Estudio del servo zona central del brazo del robot

Se procederá de la manera anterior partiendo del modelo cinemático directo para el sistema bidimensional representado a la derecha del robot como modelo simplificado de la estructura general.

Para no perder la notación se seguirá con el sistema de coordenadas expuesto anteriormente en donde la línea de referencia horizontal seguirá siendo el eje X mientras que la vertical imaginaria y perpendicular a la anterior será el eje Z, de modo que el estudio continúa como sigue. Téngase en cuenta que la distancia entre ambos círculos rojo y verde (centro de gravedad del sistema de estos dos extremos superiores del brazo robot junto con todo lo que carga encima hasta el extremo del mismo) será aproximadamente de 77,5 mm:

$$X = 0,0775 * \cos(c); \\ Z = 0,0775 * \sin(c);$$

Siendo 'c' el ángulo que hay entre el extremo superior del brazo que girará a causa del servo en estudio y la línea horizontal que pasa por su eje de rotación.

$$\vec{M} = \vec{r} \wedge \vec{F} = \vec{r} \wedge (\mathbf{m} * \vec{a}) = \vec{r} \wedge \left( \mathbf{m} * \frac{\partial \vec{V}}{\partial t} \right) = \vec{r} \wedge \left( \mathbf{m} * \frac{\partial^2 \vec{P}}{\partial^2 t} \right)$$

- Para el único círculo verde en estudio, se tiene:

$$\begin{aligned} \vec{P} &= (0,0775 * \cos(a))\vec{i} + (0,0775 * \sin(a))\vec{k} \\ \vec{V} &= (-0,0775 * \sin(a))\vec{i} + (0,0775 * \cos(a))\vec{k} \\ \vec{a} &= (-0,0775 * \cos(a))\vec{i} + (-0,0775 * \sin(a))\vec{k} + (-9,8)\vec{k} \end{aligned}$$

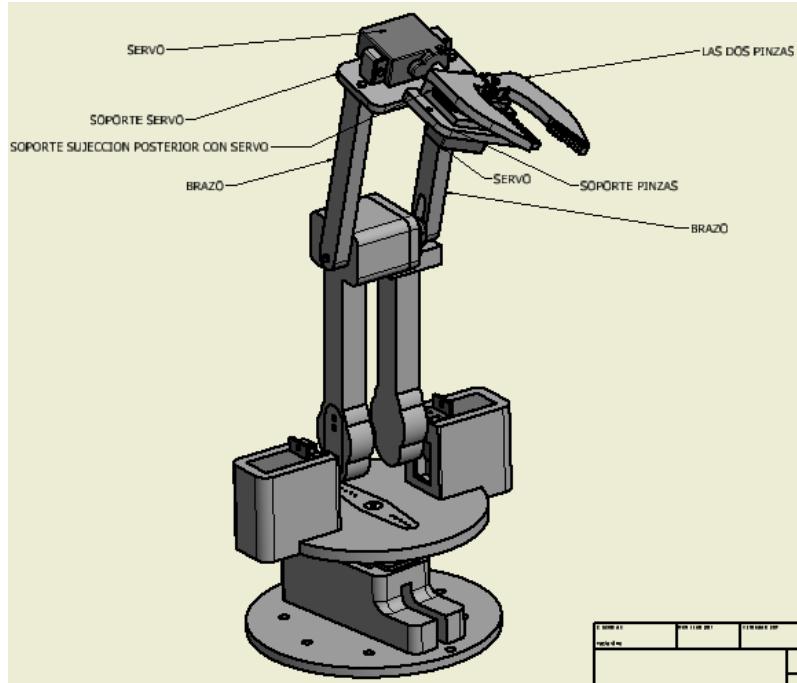
$$\begin{aligned} \vec{M} &= m * \begin{vmatrix} i & j & k \\ \cos(a) & 0 & 0 \\ -0,0775 * \cos(a) & 0 & -0,0775 * \sin(a) - 9,8 \end{vmatrix} = \\ &= -m * ((-0,0775 * \sin(a) - 9,8) * \cos(a))\vec{j} \end{aligned}$$

$$|\vec{M}| = 0,0775 * m * (\sin(a) + 9,8) \cos(a)$$

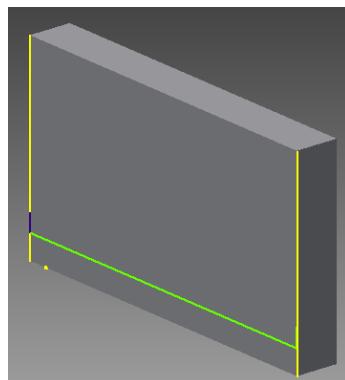
Para el momento máximo que debería soportar el servomotor analizado, se tendría el ángulo 'c' valiera 0 grados que sería la posición tumbada del extremo superior del brazo robot hacia adelante totalmente:

$$|\vec{M}| = 0,7595 * m \cdot N \cdot m$$

EN ESTE CASO EL MOMENTO DE FUERZA M3 LA CONSTANTE  $m$  SERÁ LA MASA DE LA SUMA TOTAL DE LOS SIGUIENTES ELEMENTOS DEL DIBUJO SIGUIENTE QUE TAMBIÉN DEBE SER BORRADO TRAS EL CÁLCULO:



### **8.3. ESTUDIO DINÁMICO DE LA PUERTA DE ACCESO Y EL DE LA CINTA TRASPORTADORA:**



#### **Puerta de acceso:**

En el caso de la puerta de acceso, se tiene que el par que genera siempre será constante indiferentemente de la posición de la puerta, cuyo valor se extrajo de la distancia entre los dientes de la cremallera de cierre y la media anchura de la puerta debido a la colocación del centro de gravedad de ésta y su valor es de 0,011 mm que con el momento calculado sale lo siguiente:

BORRAR: L es la anchura de la puerta, m su masa:

$$|\vec{M}| = |\vec{r}|x|\vec{F}| \cdot \sin(r^\circ F) = |\vec{r}|x|\vec{F}| \cdot \sin(90) = |\vec{r}|x|\vec{F}| = 0,011 + \left(\frac{L}{2}\right) \cdot 9,8 \cdot m =$$

$$|\vec{M}_6| = 0,011 + \left(\frac{3}{2}\right) \cdot 9,8 \cdot 0,015 \text{ N} \cdot m = 0,2315$$

$$\text{Potencia: } P = \left(\frac{\pi}{30}\right) \cdot 41,888 \frac{\text{rad}}{\text{sg}} \cdot [0,011 + \left(\frac{3}{2}\right) \cdot 9,8 \cdot 0,015] = 1,015 \text{ [W]}$$

$$\text{Rendimiento: } \eta = 64,7\%$$

#### **Cinta transportadora:**

En este caso sólo se tendría que tener en cuenta la masa de la cinta y la de la carga que transporte para poder calcular el torque máximo en el estudio, lo que supone que junto con el radio del tambor de giro que es la mitad de los 25 mm de diámetro que serían 12,5 mm setienen los siguientes cálculos:

Longitud cinta: 2 m

Ancho cinta: 0,10 m

Densidad poleas: 1,05g/cm<sup>3</sup>

Masa rodillos: 0,010 Kg

Velocidad cinta: 0,5 m/s

Diámetro rodillos: 0,10 m

Coeficiente rozamiento: 0,3

Carga máxima de: 0,10 Kg

P cinta por unidad de longitud: 0,085 Kg/m

Tensión: 5V

Intensidad: 1 Amperio

$$n(in) = \frac{60}{\pi} \cdot \frac{V1}{d1} = \frac{60}{\pi} \cdot \frac{0,5m}{0,1m} = 95,5 rpm$$

$$m(cinta) = 0,086 \frac{Kg}{m} \cdot 2 m = 0,17 Kg$$

$$m(rodillos) = 1,05 \frac{g}{cm^3} \cdot \left[ \frac{1Kg}{1000g} \right] \cdot \left[ 100 \frac{cm}{1m} \right]^3 \cdot \pi \cdot (0,05m)^2 \cdot 0,1m = 0,8246 Kg$$

$$m(carga, maxima) = 0,1 Kg$$

$$F(roz) = \mu \cdot m(total) \cdot g = 0,3 \cdot (0,1 + 0,8246 + 0,17) \cdot \frac{9,81m}{sg^2} = 3,2214 N$$

$$Mi = \frac{d1}{2} \cdot \frac{f1}{n} = \frac{d1}{2} \cdot \frac{\mu \cdot m \cdot g}{n} = \frac{0,1}{2} \cdot \frac{3,2214N}{95,5} = 0,00168 N \cdot m$$

Reductor:

$$r(maxima) = \frac{n(maxima)}{n(minima)} = \frac{400}{200} = 2$$

Se elige 1,5

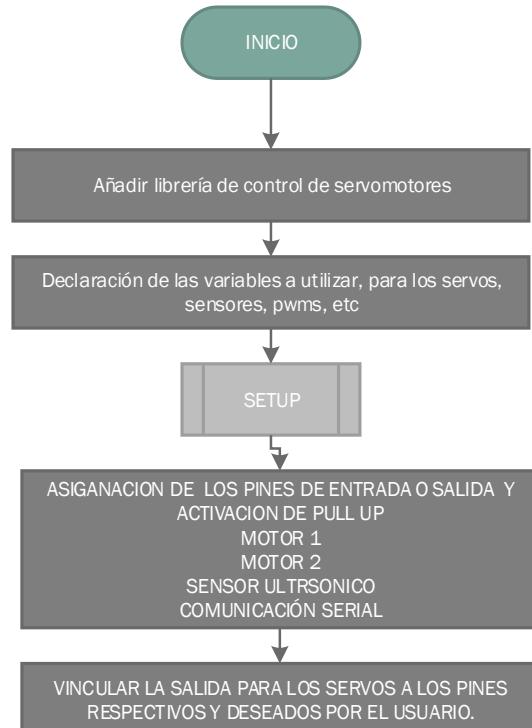
Reductora: máximo par: 0,0017

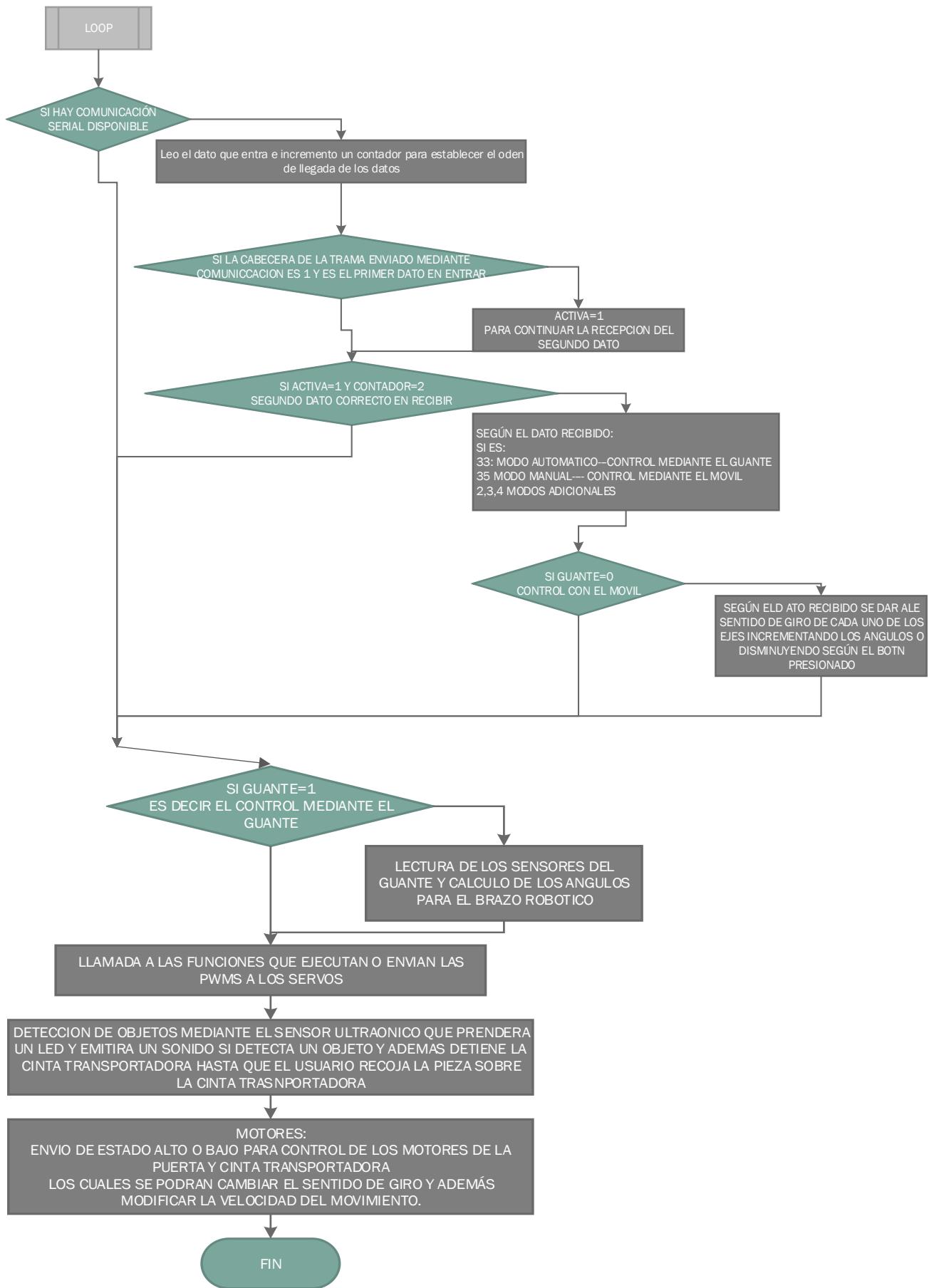
Máximo par intermitente: 0,002 rendimiento: 75%

n(motor):  $1,5 \cdot 200 = 300$  rpm

$M$  (motor):  $0,00168 / (1,5 \cdot 0,75) = 0,0015$  MNm

## 9. DIAGRAMA DE FLUJO DEL CODIGO EN ARDUINO





## 10. CÓDIGO DE PROGRAMACIÓN ARDUINO

El código a continuación está explicado de mejor manera en un **DIAGRAMA DE FLUJO** en el apartado 6, sin embargo el código tiene sus respectivos comentarios para su fácil comprensión.

```
#include <Servo.h> //LIBRERIA SERVOMOTORES
//*****+MOTORES
long T0=0;
int PWM=10;
int PWM2=10;//MOTOR2
//*****ULTRASONICO
int speakerPin = 10;
int distancia=20;
int pulso;
int trig=8;
int echo=9;
int led=52;
//*****SERIAL
int cabecera='5';
int contador=0;
int activacion=0;
//*****SERVOS
//movimiento servomotor controlando el angulo con un potenciometro
//=====SERVOMOTORES--MINIMANIPULADOR=====
Servo base;//SERVOMOTOR DEL PRIMER EJE DE GIRO
Servo baseder;//SERVOMOTOR DEL SEGUNDO EJE DE GIRO LADO IZQUIERDO
Servo baseizq;//SERVOMOTOR DEL SEGUNDO EJE DE GIRO LADO DERECHO
Servo brazo;//SERVOMOTOR DEL TERCER EJE DE GIRO
Servo basepinza;//SERVOMOTOR DE LA BASE DE LA PINZA
Servo pinza;//SERVOMOTOR DE LA PINZA
//=====PULSOS MINIMO Y MAXIMO DE LOS SERVOMOTORES=====
//const int pulsoMin=650;//0°
//const int pulsoMax=2550;//180°
//=====VARIABLES UTILIZADAS EN EL PROGRAMA=====
unsigned long angulo0=80;
unsigned long angulo1=70;
unsigned long angulo2=110;
unsigned long angulo3=83;
unsigned long angulo4=97;
unsigned long angulo5=73;
int direccion=0;
int guante=1;
float sensor1;
//=====PROGRAMA PRINCIPAL=====
void setup()
{
    //*****+MOTORES
    //%%%%%%%%%%%%%MOTOR 1%%%%%%%%%%%%%
    pinMode(22, INPUT);//ENTRADA ADELANTE/ATRAS MOTOR1
    digitalWrite(22,HIGH);//ACTIVO PULL UP
    pinMode(18, INPUT);//ENTRADA AUMENTAN
    digitalWrite(18,HIGH);//ACTIVO PULL UP
    pinMode(19, INPUT); //ENTRADA DISMINUYE
    digitalWrite(19,HIGH);//ACTIVO PULL UP
    pinMode(24, OUTPUT);//ADELANTE
    pinMode(25,OUTPUT);//ATRAS
    pinMode(11,OUTPUT);//SEÑAL
    attachInterrupt(5,AUMENTA, CHANGE);//PIN18
    attachInterrupt(4,DISMINUYE, CHANGE);//PIN19
```

```

//*****MOTOR 2*****
pinMode(23, INPUT); //ENTRADA ADELANTE/ATRAS MOTOR1
digitalWrite(23,HIGH); //ACTIVO PULL UP
pinMode(20, INPUT); //ENTRADA AUMENTAN
digitalWrite(20,HIGH); //ACTIVO PULL UP
pinMode(21, INPUT); //ENTRADA DISMINUYE
digitalWrite(21,HIGH); //ACTIVO PULL UP
pinMode(26, OUTPUT); //ADELANTE
pinMode(27,OUTPUT); //ATRAS
pinMode(12,OUTPUT); //SEÑAL
attachInterrupt(3,AUMENTA2, CHANGE); //PIN20
attachInterrupt(2,DISMINUYE2, CHANGE); //PIN21
//*****ULTRASONICO
pinMode(speakerPin,OUTPUT);
pinMode(trig, OUTPUT); // ponemos el pin 4 como trig
pinMode(echo, INPUT); // ponemos el pin 2 como entrada eco
//*****SERIAL
pinMode(13,OUTPUT); //SALIDA DIR
digitalWrite(10,LOW);
Serial.begin(9600);
Serial2.begin(9600);
//*****SERVOS
// put your setup code here, to run once:
base.attach(2, 650, 2550); //pin servo1//EJE1=0-175°//ardu uno 3
baseder.attach(3, 650, 2550); //pin servo2//EJE2.1 //ardu uno 5
baseizq.attach(4, 650, 2500); //pin servo1//EJE2.2=0-180° vlor medio =55//ardu uno 6
brazo.attach(5, 650, 2550); //pin servo2//EJE3=0-180° mitad 83°///ardu uno 9
basepinza.attach(6, 650, 2550); //pin servo2//EJE4//centro=97°//ardu uno 10
pinza.attach(7, 650, 2550); //pin servo2//EJE5//73(abierto)-125(cerrado)//ardu uno 11
//=====PULSOS MINIMO Y MAXIMO DE LOS SERVOMOTORES=====
//const int pulsoMin=650;//0°
//const int pulsoMax=2550;//180°
//=====VARIABLES UTILIZADAS EN EL PROGRAMA=====
}
void loop()
{
    if (Serial2.available() > 0)
    {
        contador=contador+1;
        // digitalWrite(13,HIGH);
        cabecera = Serial2.read(); //lee en ascii
        // Serial1.write(cabecera); //llega en ascii

        if(cabecera==1 && contador==1)
        {
            activacion=1;
            Serial.println("llegue aqui");
        }
        if(contador==2 && activacion==1 && cabecera==33) //=====MODO AUTOMATICO=====
        //manejado por la mano el guante
        {
            contador=0;
            activacion=0;
            guante=1;
        }
    }
}

```

```
    Serial.println("AUTOMATICO--GUANTE") ;
    digitalWrite(13,HIGH);
}
if(contador==2 && activacion==1 && cabecera==35)//=====MODO MANUAL=====
//=controlado por el telefono
{
    contador=0;
    activacion=0;
    guante=0;
    Serial.println("MANUAL---TELEFONO") ;
    digitalWrite(13,LOW);
}
if(contador==2 && activacion==1 && cabecera==2)
{
    contador=0;
    activacion=0;
    Serial.println("ADELANTE") ;
    digitalWrite(13,HIGH);
}
if(contador==2 && activacion==1 && cabecera==3)
{
    contador=0;
    activacion=0;
    Serial.println("DERECHA") ;
    digitalWrite(13,LOW);
}
if(contador==2 && activacion==1 && cabecera==4)
{
    contador=0;
    activacion=0;
    Serial.println("IZQUIERDA") ;
    digitalWrite(13,LOW);
}
if(guante==0)
{
    if(contador==2 && activacion==1 && cabecera==10)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO HORARIO 1") ;
        angulo0=angulo0+5;
        digitalWrite(13,HIGH);
    }
    if(contador==2 && activacion==1 && cabecera==11)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO ANTIHORARIO 1") ;
        angulo0=angulo0-5;
        digitalWrite(13,LOW);
    }
    if(contador==2 && activacion==1 && cabecera==20)
    {
        contador=0;
        activacion=0;
```

```
        Serial.println("GIRO HORARIO 2") ;
        angulo1=angulo1+5;
        angulo2=180-angulo1;
        digitalWrite(13,HIGH);
    }
    if(contador==2 && activacion==1 && cabecera==21)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO ANTIHORARIO 2") ;
        angulo1=angulo1-5;
        angulo2=180-angulo1;
        digitalWrite(13,LOW);
    }
    if(contador==2 && activacion==1 && cabecera==30)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO HORARIO 3") ;
        angulo3=angulo3+5;
        digitalWrite(13,HIGH);
    }
    if(contador==2 && activacion==1 && cabecera==31)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO ANTIHORARIO 3") ;
        angulo3=angulo3-5;


---


        digitalWrite(13,LOW);
    }
    if(contador==2 && activacion==1 && cabecera==40)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO HORARIO 4") ;
        angulo4=angulo4+5;
        digitalWrite(13,HIGH);
    }
    if(contador==2 && activacion==1 && cabecera==41)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO ANTIHORARIO 4") ;
        angulo4=angulo4-5;
        digitalWrite(13,LOW);
    }
    if(contador==2 && activacion==1 && cabecera==50)
    {
        contador=0;
        activacion=0;
        Serial.println("GIRO HORARIO 5") ;
        angulo5=angulo5+5;
        digitalWrite(13,HIGH);
    }
    if(contador==2 && activacion==1 && cabecera==51)
    {
```

```

        {
            contador=0;
            activacion=0;
            Serial.println("GIRO ANTIHORARIO 5") ;
            angulo5=angulo5-5;
            digitalWrite(13,LOW);
        }
    }
    if(contador==2 && activacion==1)
    {
        contador=0;
        activacion=0;
    }

}
if(guante==1)
{
    //=====BASE=====
    angulo0=analogRead(0);
    angulo0=((angulo0*angulo0)/(1023*4));
    angulo0=map(angulo0,0,255,0,180);
    //Serial.println(angulo0);
    //=====MOVIMIENTO BRAZO TOTAL=====
    angulo1=analogRead(1);//LEO EL VALOR DEL SENSORFLEX1;

    angulo2=analogRead(2);//LEO EL VALOR DEL SENSORFLEX2;
    angulo1=(angulo1+angulo2)/2;
    angulo3=angulo1;
    angulo1=((angulo1*angulo1)/(1023*4));
    angulo1=map(angulo1,0,255,55,200);//del valor medio a 180°
    angulo2=180-angulo1;

    angulo3=((angulo3*angulo3)/(1023*4)); //se establece valor entre 0 y 180°
    angulo3=map(angulo3,0,200,70,180);

    angulo4=97;
    //=====PINZA=====
    angulo5=analogRead(3);//LEO EL VALOR DEL SENSORFLEX4;
    angulo5=((angulo5*angulo5)/(1023*4));
    angulo5=map(angulo5,0,200,73,170);
}

fbase(angulo0);
fbaseder(angulo1);//entre 0 y 180
fbaseizq(angulo2); //negado del anterior
fbrazo(angulo3);//de 0 a 180
fbasepinza(angulo4);//angulo medio=97 centro
fpinza(angulo5);//angulo medio=97 centro
Serial.print("ANGULO 0 ");Serial.println(angulo0);
// Serial.print("ANGULO 1 ");Serial.println(angulo2);
// Serial.print("ANGULO 2 ");Serial.println(angulo3);
//Serial.print("ANGULO 3 ");Serial.println(angulo4);
//Serial.print("ANGULO 4 ");Serial.println(angulo5);

*****+ULTRASONICO*****
digitalWrite(trig, HIGH); // lo activamos

```

```
delayMicroseconds(10); // esperamos 10 microsegundos
digitalWrite(trig, LOW); // lo desactivamos
pulso = pulseIn(echo, HIGH); //tiempo de llegada del eco
// 
distancia = ((float(pulso/1000.0))*34.32)/2; // pasar tiempo a cm.
if(distancia<6)
{
    PWM=0;
    analogWrite(11,PWM);
    digitalWrite(led,HIGH);
    for (long i = 0; i < 250 * 1L; i+= 956 * 2) //l=constante LONG
    {
        analogWrite(speakerPin,130);
    }
}
else
{
    PWM=250;
    analogWrite(11,PWM);
    digitalWrite(led,LOW);
    analogWrite(speakerPin,0);
}

//*****MOTORES DC*****
if(digitalRead(22))
{
    digitalWrite(24,LOW);
    digitalWrite(25,HIGH);

    //Serial.println("MOTOR 1 ADELANTE");
    // Serial.println(PWM);
}
else
{
    digitalWrite(25,LOW);
    digitalWrite(24,HIGH);
    // Serial.println("MOTOR 1 ATRAS");
    // Serial.println(PWM);
}
if(digitalRead(23))
{
    digitalWrite(26,LOW);
    digitalWrite(27,HIGH);

    // Serial.println("MOTOR 1 ADELANTE");
    // Serial.println(PWM2);
}
else
{
    digitalWrite(27,LOW);
    digitalWrite(26,HIGH);

    // Serial.println("MOTOR 1 ATRAS");
    // Serial.println(PWM2);
}

}

//*****SERVOS*****
```

```
//*****SERVOS*****
void fbase(int angulo0)//ANGULO ENTRE 0 Y 165°
{
    if(angulo0<10)
    {
        angulo0=10;
    }
    if(angulo0>165)
    {
        angulo0=165;
    }
    base.write(angulo0);
}
void fbaseder(int angulo1)
{
    if(angulo1<80)
    {
        angulo1=80;
    }
    if(angulo1>170)
    {
        angulo1=170;
    }
    baseder.write(angulo1);
}
void fbaseizq(int angulo2)
{
    if(angulo2<90)
    {
        angulo2=90;
    }
    if(angulo2>170)
    {
        angulo2=170;
    }
    baseizq.write(angulo2);
}

void fbrazo(int angulo3)
{
    if(angulo3<0)
    {
        angulo3=0;
    }
    if(angulo3>180)
    {
        angulo3=180;
    }
    brazo.write(angulo3);

}
void fbasepinza(int angulo4)
{
    if(angulo4<0)
```

```

    if (angulo4<0)
    {
        angulo4=0;
    }
    if (angulo4>180)
    {
        angulo4=180;
    }
    basepinza.write(angulo4);

}

void fpinza(int angulo5)
{
    if (angulo5<73)
    {
        angulo5=73;
    }
    if (angulo5>125)
    {
        angulo5=125;
    }
    pinza.write(angulo5);
}

//*****MOTORES*****
void AUMENTA()
{
    ..
    {

analogWrite(11,PWM);
if ( millis() > T0 + 250)
    {   PWM=PWM+15;
        T0 = millis();
    }

if(PWM>255)
{PWM=255;}
}
void DISMINUYE()
{
    analogWrite(11,PWM);
if ( millis() > T0 + 250)
    {   PWM=PWM-15;
        T0 = millis();
    }

if(PWM<0)
{PWM=0;}
}
void AUMENTA2()
{
    analogWrite(12,PWM2);
}

```

```

if ( millis() > T0 + 250)
{
    PWM2=PWM2+15;
    T0 = millis();
}

if(PWM2>255)
{PWM2=255;}
}
void DISMINUYE2()
{

analogWrite(12,PWM2);
if ( millis() > T0 + 250)
{
    PWM2=PWM2-15;
    T0 = millis();
}

if(PWM2<0)
{PWM2=0;}
}

```

---

## 11. PROGRAMA DE LA APLICACIÓN DESARROLLADA PARA DISPOSITIVOS ANDROID



# App Inventor

El software utilizado para el desarrollo de la aplicación para los dispositivos Android es el APP INVENTOR el cual es un software libre que únicamente requiere una cuenta de correo de google para permitir la edición de cualquier programa.

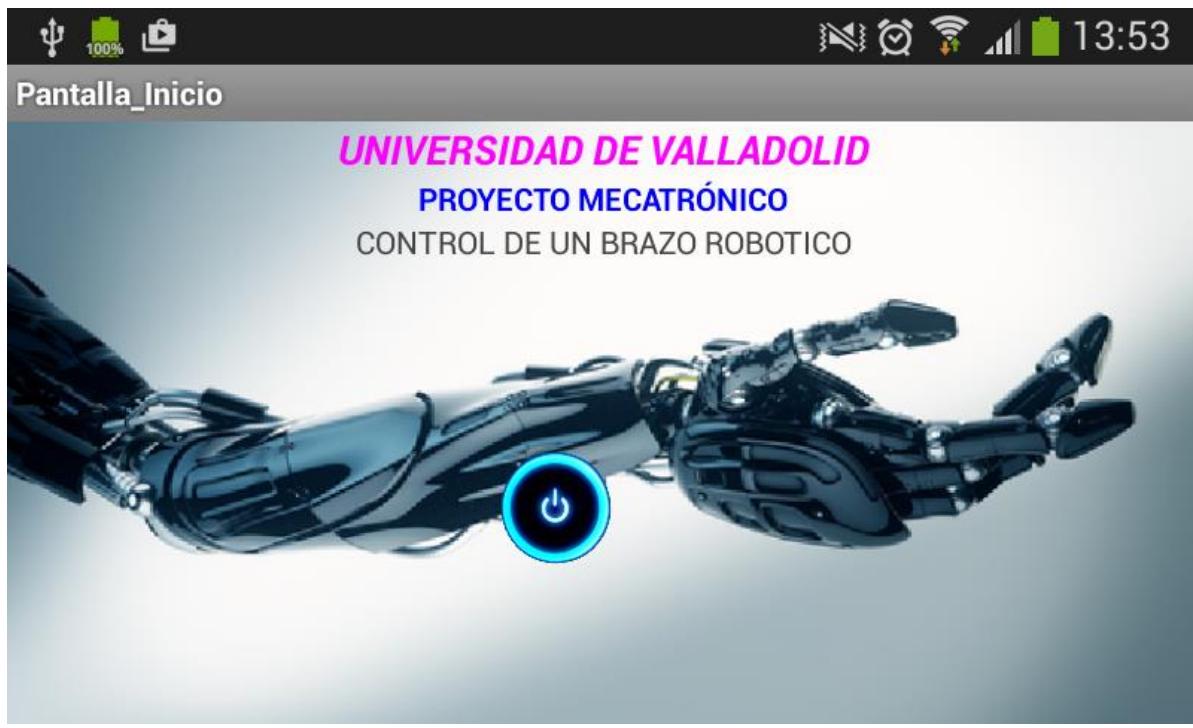
App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android. Para desarrollar aplicaciones con App Inventor sólo se necesita un navegador web y un teléfono o Tablet Android. App Inventor se basa en un servicio web que permite almacenar el trabajo y ayuda a realizar un seguimiento de sus proyectos.

El desarrollo del programa en esta plataforma consistirá básicamente en el envío de una serie de comandos según las necesidades que tenga el usuario, en este caso se ha considerado que desde la aplicación móvil se hará el control de cada uno de los grados de libertad del Robot, se tendrán botones para controlar el sentido de giro de cada uno de los ejes o articulaciones del brazo Robótico además se tendrá un control para poder pasar de un modo a otro, es decir se tendrá el primer modo en el cual el brazo Robótico es controlado por el guante directamente por el usuario y el segundo modo en el cual el control está totalmente realizado por el dispositivo Android.

Para poder llevar a cabo esta comunicación desde el dispositivo Android y nuestro Arduino Mega el cual está basado en el microcontrolador ATMega2560, se lo ha realizado mediante comunicación bluetooth, utilizando un módulo bluetooth como es el HC 06 el cuyas características fueron mencionadas en un inicio. Este recibe mediante comunicación bluetooth los datos que envía el dispositivo Android y los transmite por comunicación serial hacia nuestro Arduino Mega, el cual según la serie de comandos enviados ejecutará una u otra tarea.

## 11.1. PANTALLA DE INICIO

La pantalla inicial de nuestra aplicación es la mostrada en la imagen está será la primera que aparecerá al iniciar la aplicación



## 11.2. PANTALLA DE CONTROL

Al presionar en el botón de encendido nos dirige a la pantalla siguiente en donde habrá que establecer la comunicación bluetooth y se tendrá acceso al control para el Brazo Robótico.

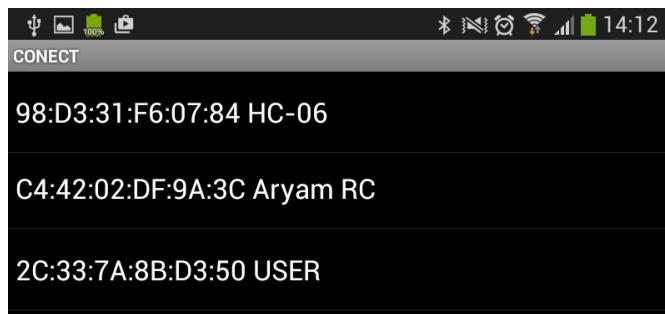


EL manejo de la comunicación bluetooth y el uso de esta pantalla se lo realizará de la siguiente manera:

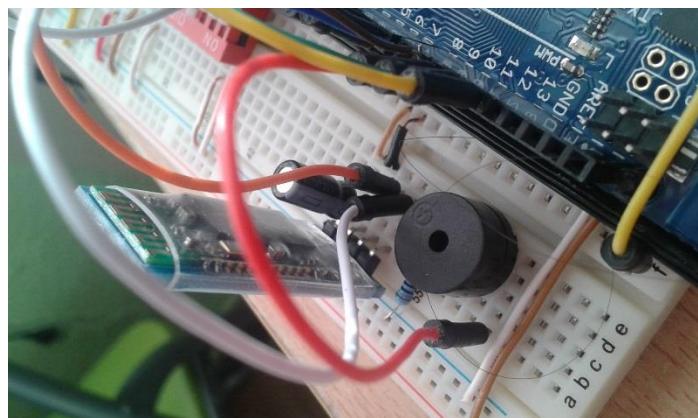
- ✓ Se activa el bluetooth para la conexión con cualquier dispositivo ya sea desde una tablet o desde un móvil.



- ✓ En el botón CONNECT presionamos y aparecerá la lista de dispositivos a conectar y entre ellos si ya está asociado el HC 06 lo seleccionamos si no en caso contrario habrá que emparejarlo(el código usualmente es 1234)



- ✓ Una vez conectado el módulo bluetooth detiene el parpadeo del led indicador, informando que la comunicación ha sido establecida y ya se puede proceder a realizar cualquier maniobra con el móvil.



- ✓ Para la desconexión únicamente se presiona el botón de DISCONNECT e inmediatamente el Robot dejará este modo de control y se quedará establecido por el modo de control mediante el guante.



### **11.3. PANTALLA DE CONTROL- MANEJO DE LOS BOTONES:**

Cada botón tiene una funcionalidad específica hay botones en la parte central los cuales son adicionales para una expansión futura del proyecto, los mismos que pueden utilizar para añadir el control cinemático inverso del Brazo Robótico y permitir el movimiento de los ejes X Y Z, dichos cálculos del modelo cinemático inverso se encuentran realizados en la parte de cálculos de este informe. Son adicionales y en nuestro Arduino únicamente sería de insertar funciones adicionales.



El botón AutoPilot lo que permite es el control mediante el móvil, y desactivará por completo el control realizado mediante el Guante Sensible a movimientos.

Si activamos este botón el resto de funcionalidades queda desactivada relacionadas con el guante.



Mediante estos botones se nos permitirá el control de nuestras articulaciones del Robot ya sea en un sentido o en otro sentido teniendo desde la 1 hasta la 5

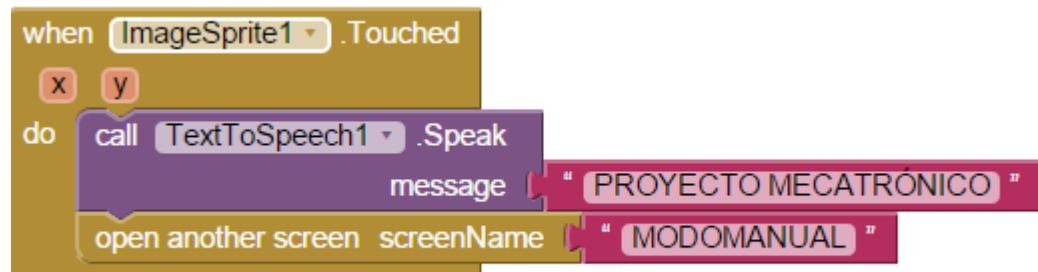
1. Corresponde al movimiento de la base del robot
2. Corresponde al movimiento de la segunda articulación del robot de los primeros brazos del robot
3. Movimiento de los brazos extremos del robot los cuales sujetarán la base para la pinza y la piza de agarre.
4. Movimiento de la base de la pinza para definir la orientación de la misma.
5. Movimiento de la pinza para agarrar o dejar piezas sobre un determinado sitio.



Con este botón lo que se nos permitirá hacer es el cambio de control a un control realizado por el guante que manejará el operario. Al activar esta funcionalidad todas las demás quedan totalmente desactivadas, es decir el único control posible sobre el robot es mediante nuestro guante.

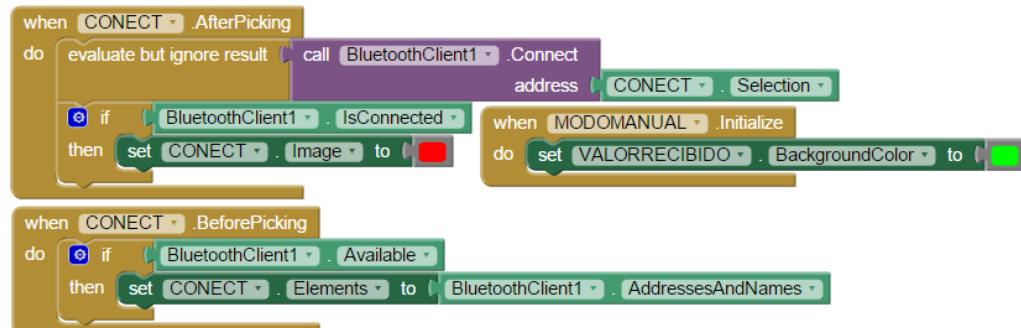
#### 11.4. PROGRAMACIÓN DE LA APLICACIÓN

##### PANTALLA 1:

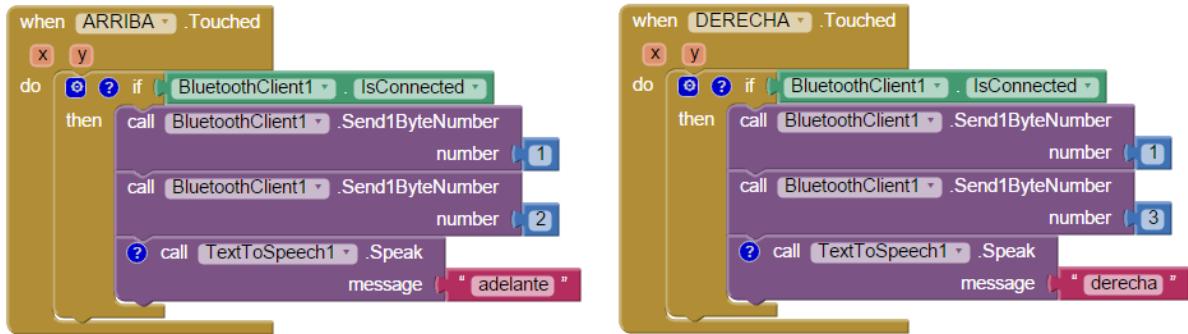


##### PANTALLA 2:

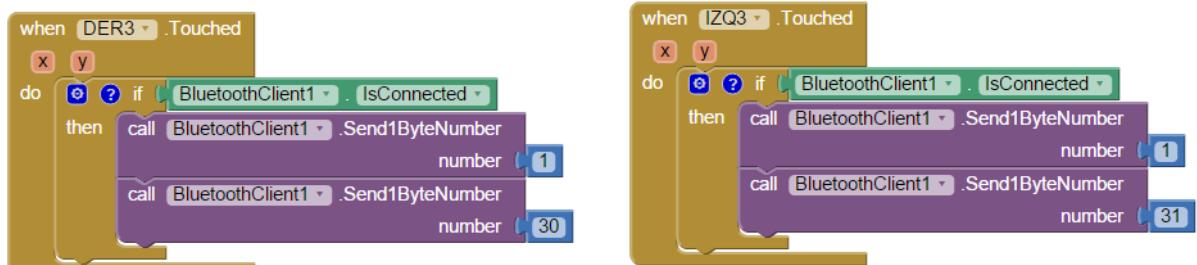
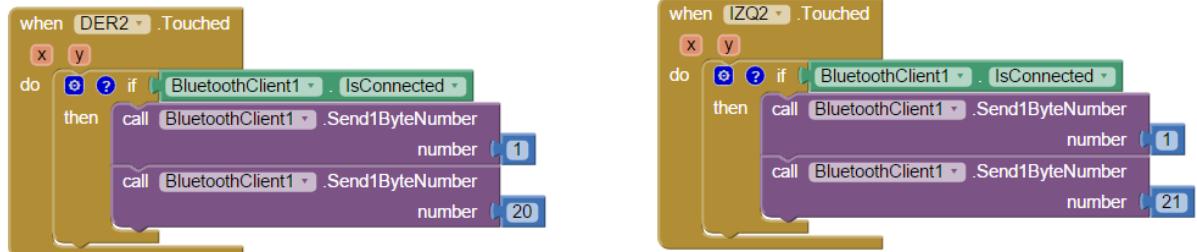
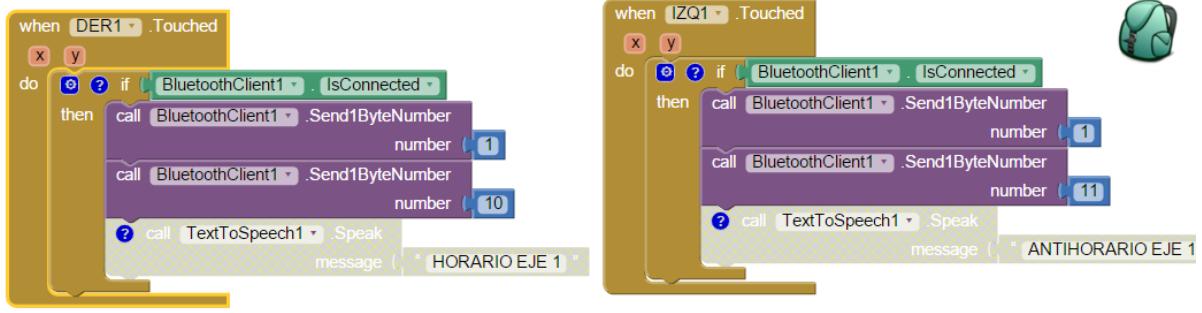
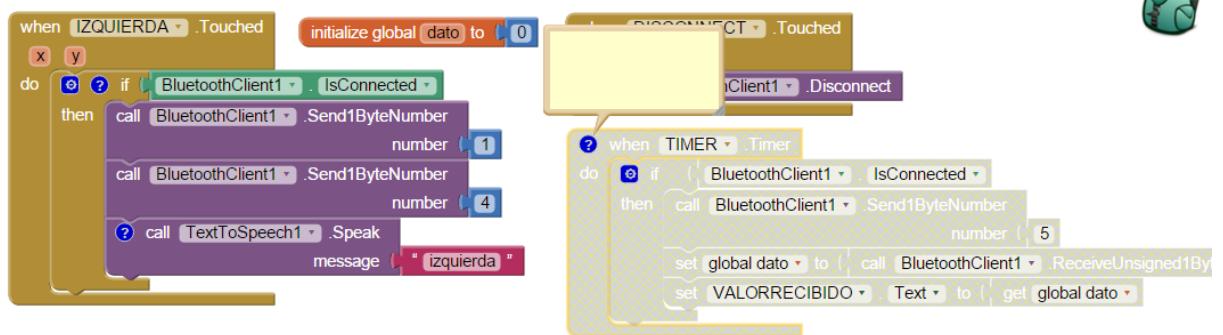
- Establecimiento de la comunicación bluetooth

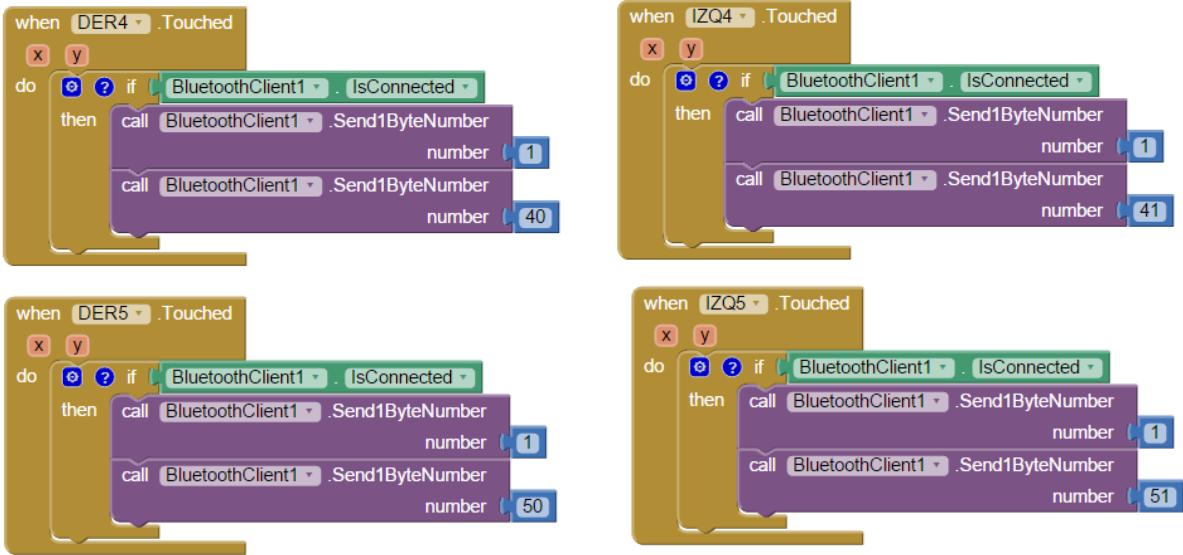


- Envió de datos según el botón presionado

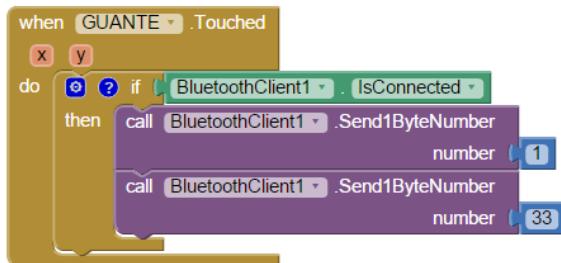


En este siguiente bloque puede verse un bloque deshabilitado el cual permitirá en aplicaciones futuras o dentro de la misma aplicación la recepción de datos en tiempo real por ejemplo en el caso de que se requiera hacer un control de temperatura se podría visualizar la misma en nuestra pantalla del móvil.

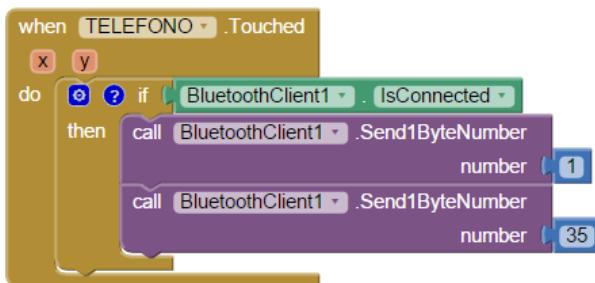




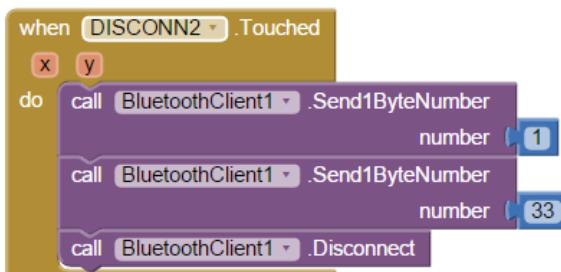
Envio del comando para el control mediante el guante



Envío del comando para el control mediante el móvil



Desconexión del módulo bluetooth



## 12. PRESUPUESTO DEL PROYECTO

PRECIO DE LOS SERVOS UTILIZADOS:



www.pololu.com  
Power HD standard servo 3001HB.

ESPECIFICACIONES DEL SERVOMOTOR BASE ROTATORIA DEL ROBOT	
DIMENSIONES	40,7x20,5x39,5 mm
PESO	0,043 Kg
VELOCIDAD A 6V	8,7266 rad/sg
VELOCIDAD A 4,8V	6,9813 rad/sg
TORQUE	0,044 Kg·m
PRECIO	9,95 €/unidad
RENDIMIENTO MAX CON CARGA A 6V	82,23%
ENLACE DEL PRODUCTO	<a href="https://www.pololu.com/product/1058/specs">https://www.pololu.com/product/1058/specs</a>



ESPECIFICACIONES DEL SERVOMOTOR MOVIMIENTO BRAZO ROBOT	
DIMENSIONES	40X13,20X34,70 mm
PESO	0,018 Kg
VELOCIDAD A 6V	9,52 rad/sg
SIN CARGA	0,18 Amperios
VELOCIDAD A 4,8V	8,73 rad/sg
SIN CARGA	0,16 Amperios
TORQUE A 6V	0,027 Kg·m
TORQUE A 4,8V	0,002 Kg·m
PRECIO	7,95 €/unidad
RENDIMIENTO MAX CON CARGA A 6V	
ENLACE DEL PRODUCTO	<a href="https://www.pololu.com/file/0J318/HD-1160A.pdf">https://www.pololu.com/file/0J318/HD-1160A.pdf</a>



ESPECIFICACIONES DEL SERVOMOTOR RESTANTES ARTICULACIONES	
DIMENSIONES	22,2x11,8x31 mm
PESO	0,009 Kg
VELOCIDAD A 4,8V	10,472 rad/sg
TORQUE	0,018 Kg·m
PRECIO	4,60€/unidad
ENLACE DEL PRODUCTO	<a href="http://www.micropik.com/PDF/SG90Servo.pdf">http://www.micropik.com/PDF/SG90Servo.pdf</a>

### 12.1.1. MOTOR DE ROTACIÓN PARA PUERTA ACCESO Y CINTA TRANSPORTADORA:



ESPECIFICACIONES DEL SERVOMOTOR PUERTA ACCESO Y CINTA TRANSPORTADORA	
DIMENSIONES	68x25Ø mm
PESO	0,070 Kg
VELOCIDAD A 4,8V	41,888 rad/sg
TORQUE	0,015 Kg·m
PRECIO	3,58 €/unidad
ENLACE DEL PRODUCTO	<a href="http://www.dx.com/p/zndiy-bry-dc-12v-400rpm-dc-6v-200rpm-high-torque-gear-motor-silver-300222#.VpvaMChDIU">http://www.dx.com/p/zndiy-bry-dc-12v-400rpm-dc-6v-200rpm-high-torque-gear-motor-silver-300222#.VpvaMChDIU</a>

#### **PRECIO TORNILLOS Y PERNOS:**



#### TORNILLOS

Lote de 100 unidades de tornillo con las especificaciones exigidas a un precio de **2,07€** como solo se va a pedir un único lote el precio tan sólo es de: **1x2,07€ = 2,07€** totales.

Tornillos de 2 mm de diámetro y 12 mm de longitud buscados en el siguiente enlace:  
PERNOS (TORNILLOS ESPECIFICAMENTE POR DEFECTO)



Lote de 50 unidades de pernos (tornillos) con las especificaciones exigidas a un precio de **1,88€** como solo se va a pedir un único lote el precio tan sólo es de: **1x1,88€ = 1,88€** totales.

Al no haber encontrado los pernos en cuestión se han utilizado tornillos de 20 mm de longitud con tuercas para poder conseguir el agarre aproximado de un perno concreto.

Tornillos de 2 mm de diámetro y 20 mm de longitud.

Para poder completar el perno se han tenido que comprar tuercas para el agarre del mismo:

#### TUERCAS (PARA EL PERNO)

Lote de 50 unidades de tuercas con las especificaciones exigidas a un precio de **1,17€** como solo se va a pedir un único lote el precio tan sólo es de: **1x1,17€ = 1,17€** totales.

Tuercas de 2 mm de diámetro y 12 mm de longitud.

#### COMPUERTA 74LS08



**DM74Ls08N 74LS08PC y puerta 4-element 2-Bipolar 14 pin Pdip 74LS08N**  
de National or Fairchild

**EUR 1,63**

Sólo hay 4 en stock. Cómpralo cuanto antes.

Ilumin

#### PUENTE H L293D



**L293D chip IC del cond**  
de Genérico

**EUR 3,18**

Más opciones de compra

**EUR 3,18 nuevo (2 ofertas)**

#### MODULO BLUETOOTH HC 06



**XINTE Inalámbrica Bluetooth Serial esclavo módulo HC-06**  
de XINTE

**EUR 3,85**

#### SENSORES FLEX



**BricoGeek - Sensor flex 7cm**  
de BricoGeek

**EUR 9,10 + EUR 5,95 de envío**

Sólo hay 2 en stock. Cómpralo cuanto antes.

## ARDUINO MEGA



SainSmart - Tarjeta de programación Mega 2560 para Arduino de SainSmart

**EUR 18,99 EUR 35,00** Premium  
Recíbelo el **lunes, 18 enero**

Más opciones de compra  
**EUR 18,99** nuevo (2 ofertas)

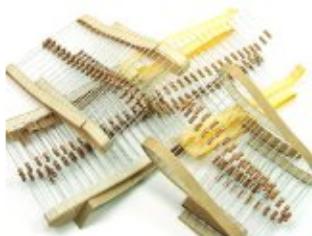
## AMPLIFICADORES LM324 PARA ACONDICIONAMIENTO



Lm324N Quad Op Amp Gp  $\pm$  16V/32 V 14 pin Pdip X2Pcs  
de ST or Texas

**EUR 1,63**

## RESISTENCIAS VARIAS:



Piezas / pcs. 200 x Resistencias / Resistor, 0.25W, 4 x 50 de Gen

**EUR 3,49** + EUR 4,49 de envío  
Más opciones de compra  
**EUR 2,68** nuevo (2 ofertas)

## CABLE PARA PROTOBOARD



Ecloud Shop 65 x Cable Jumper Wire para Protoboard Breadboard  
de Ecloud Shop

**EUR 2,99**

## SENSOR ULTRASONICO



Neuftech® Ultrasónico módulo HC-SR04 sensor transductor de medición de Neuftech

**EUR 4,59** Producto Plus  
Añadir a un pedido que cumpla las condiciones para recibirlo antes de **lunes, 18 enero**

## MOTOR DC 12 V



DC 12 V C

de Sourcingm

**EUR 3,58**

#### LEDS Y BUZZER



SODIAL(R) 75 x 3mm Diodos Emisores de Luz LED  
de SODIAL(R)

**EUR 1,36**

Más opciones de compra  
**EUR 1,36** nuevo (3 ofertas)



Piezas 2 x 5V / 12mm Zumbador Buzzer Beeper ,  
de Just-Honest

**EUR 3,15** + EUR 4,49 de envío

#### **PRESUPUESTO PROYECTO MECATRONICO**

CANTIDAD	ITEM	PRECIO UNITARIO	TOTAL ITEM
1	SERVOMOTOR HD 3001 HB	9,95	9,95
2	SERVOMOTOR HD 1160 A	7,95	15,9
3	micro-servo-9g-towerpro	4,6	13,8
15	TORNILLOS, PERNOS Y TUERCAS	1,5	22,5
1	PUENTE H L293D	3,18	3,18
1	MODULO BLUETOOTH HC 06	3,85	3,85
4	SENSORES FLEX	9,1	36,4
1	ARDUINO MEGA 2560	18,99	18,99
2	AMPLIFICADORES LM354 PARA ACONDICIONAMIENTO	1,63	3,26
1	RESISTENCIAS VARIAS:	2,68	2,68
1	CABLE PARA PROTOBOARD	3	3
1	SENSOR ULTRASONICO	4,59	4,59
1	CINTA PLOMA QUE CUBRE LA MAQUETA	2	2
2	MOTORES DC 12 V	3,58	7,16
4	LEDS	1,36	5,44
1	BUZZER - MICROFONO	1,55	1,55
<b>TOTAL DEL PROYECTO</b>			<b>154,25</b>

## **13. PROBLEMAS ENCONTRADOS Y SOLUCIONES ADOPTADAS**

- ⊕ Una de las principales dificultades al inicio del proyecto fue el diseño de las piezas en Inventor ya que al ser construidas para imprimir en la impresora 3D su diseño tenía que ser preciso y exacto, por lo tanto tomó un tiempo, y una que otra pieza sustituida por una de diferente medida.
- ⊕ El desarrollo de la aplicación para Android fue complicado inicialmente ya que no se estaba familiarizado con este entorno, y entender como es el envío de datos para poder establecer el protocolo de comunicación con el módulo bluetooth
- ⊕ La programación en Arduino se complicó en tanto a la comunicación con el módulo bluetooth puesto que no se sabía que codificación enviaba el móvil hacia el módulo bluetooth y este a su vez al arduino mega. Sin embargo finalmente todo pudo solucionarse utilizando los demás puertos seriales para verificar los datos de llegada.
- ⊕ La implementación misma del hardware de la cinta trasportadora ya que la disponibilidad de materiales era escasa, es decir se requería de un cinta que roce o tenga un alto coeficiente de rozamiento con los rodillos sobre los cuales girará, finalmente se la construyó con partes de una impresora antigua y una banda que fue acondicionada para esta aplicación
- ⊕ La construcción del guante también tuvo sus inconvenientes inicialmente al tratar de acoplar los sensores a un guante, no se establecían bien los movimientos pero finalmente todo funcionó de la manera correcta al cocer cada sensor a los dedos de los guantes.
- ⊕ En el caso de ensamblajes, se tiene que varias piezas se tuvieron que precisar al milímetro en el encaje para poder simular una presentación evitando que se cancelaran ciertos grados de libertad en el proyecto final y dado que el Autodesk Inventor es muy exigente en ese aspecto de encajar conjuntos y componentes de manera exacta, se han tenido que hacer arreglos de aproximación para poder permitir crear las simulaciones de movimiento con las respectivas secuencias grabadas en video.

## **14. CONCLUSION**

- ⊕ El proyecto Mecatrónico ha permitido nuestra familiarización con el mundo Mecatrónico en general y nos ha enriquecido en la adquisición de nuevos conocimientos y manejo de nuevas herramientas como es el caso del desarrollo de aplicaciones para Android, además ha sido de gran ayuda por el hecho de hacer uso de un sinnúmero de elementos nuevos como un módulo bluetooth y entre otros, los cuales después deben ser montados en conjuntos para un adecuado funcionamiento. Mediante este proyecto se han profundizado y aclarado conceptos de la teoría los cuales se los ha aplicado en todo el desarrollo del proyecto de manera práctica, además de entender como es el funcionamiento de los mecanismos y como controlar estos mediante Arduino ya como un conjunto en general.

## **15. BIBLIOGRAFÍA:**

- ✓ Arduino: Oscar Torrente Arteneo.
- ✓ Apuntes de clase (campus virtual)
- ✓ <http://ai2.appinventor.mit.edu/> - Aplicación para Android
- ✓ Bibliografía adicional para selección del material se incluye en el respectivo ítem.