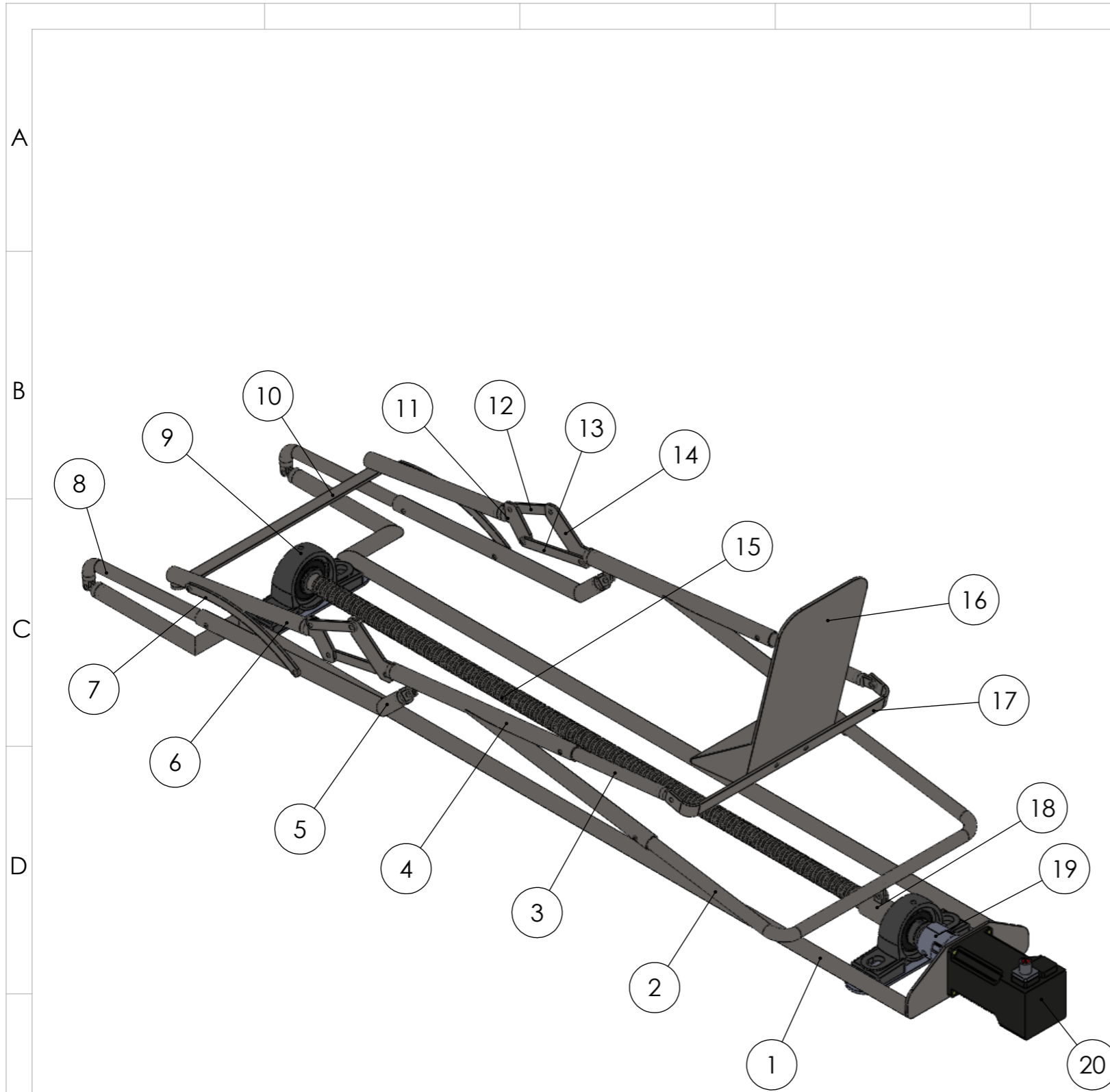


## **ANEXOS**

## **ANEXO 1**

### **Planos del diseño mecánico**

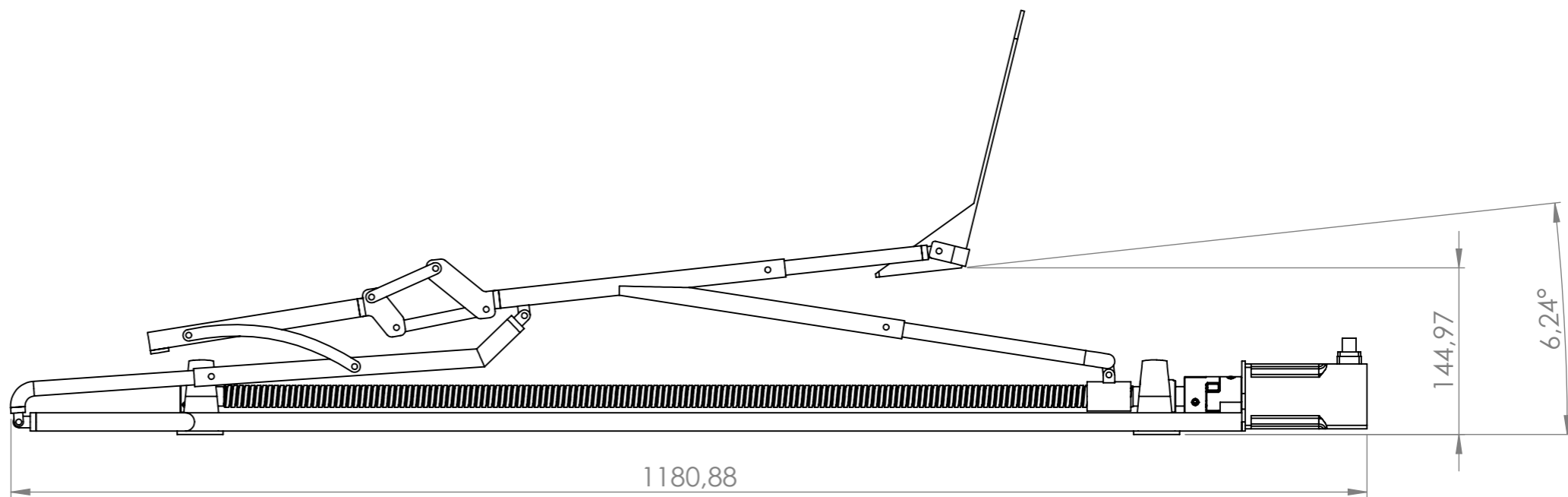
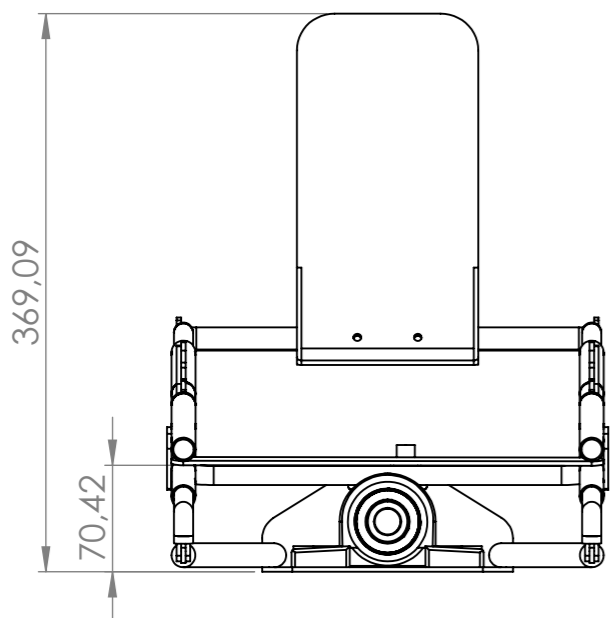
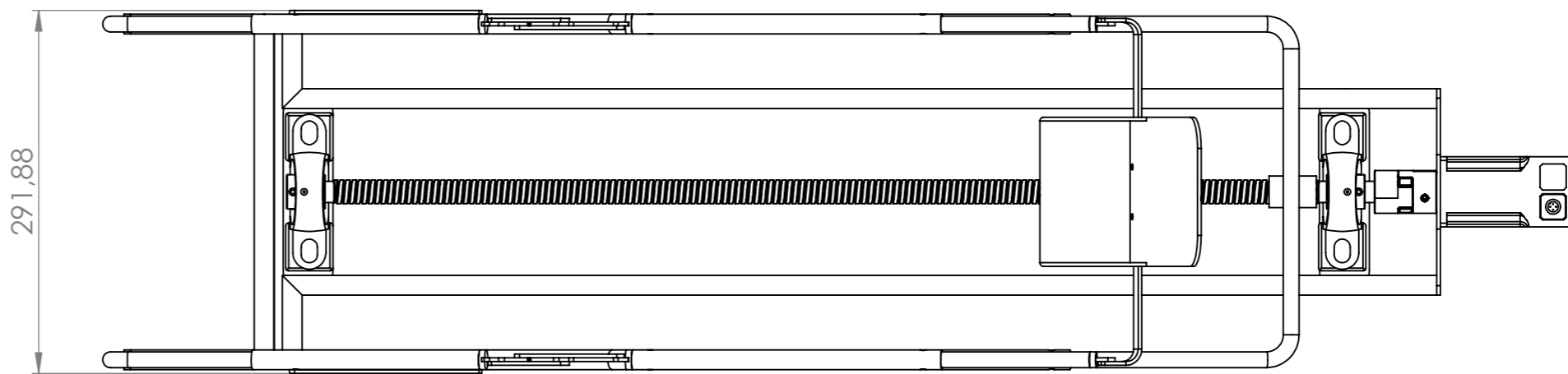


20	MOTOR		1	NEMA 23
19	ACOPLE DE MANDÍBULA		1	LOVEJOY L070
18	TUERCA	CIME20-001-017	1	
17	BRIDA SOPORTE PIE	CIME20-001-016	1	
16	SOPORTE PIE	CIME20-001-015	1	
15	TORNILLO DE POTENCIA	CIME20-001-014	1	
14	BARRA D	CIME20-001-013	2	
13	BARRA A	CIME20-001-012	2	
12	BARRA C	CIME20-001-011	2	
11	BARRA B	CIME20-001-010	2	
10	BARRA SOPORTE FÉMUR	CIME20-001-009	1	
9	CHUMACERA		2	UCP 203
8	TUBO BRAZO EXTENSIÓN	CIME20-001-008	2	
7	BARRA CIRCULAR BRAZO-FÉMUR	CIME20-001-007	2	
6	TUBO FÉMUR	CIME20-001-006	2	
5	TUBO BRAZO	CIME20-001-005	2	
4	TUBO TIBIA	CIME20-001-004	2	
3	TUBO TIBIA EXTENSIÓN	CIME20-001-003	2	
2	TUBO SOPORTE TIBIA EXTENSIÓN	CIME20-001-002	1	
1	BASE	CIME20-001-001	1	
N.º DE ORDEN	DENOMINACIÓN	N.º DE DIBUJO	N.º	OBSERVACIONES

PESO:		12.7 kg			
Fecha:		Nombre:		TÍTULO:	
Dibujó:		Grijalva C.		REHABILITADOR POLICÉNTRICO PASIVO DE RODILLA	
Revisó:		Ing. Gamez B.		ESC: 1:5	
Aprobó:		Ing. Gamez B.		Nº DE DIBUJO: CIME20-001	
<b>UTN</b>					
Edición	Modificación	Fecha	Nombre		

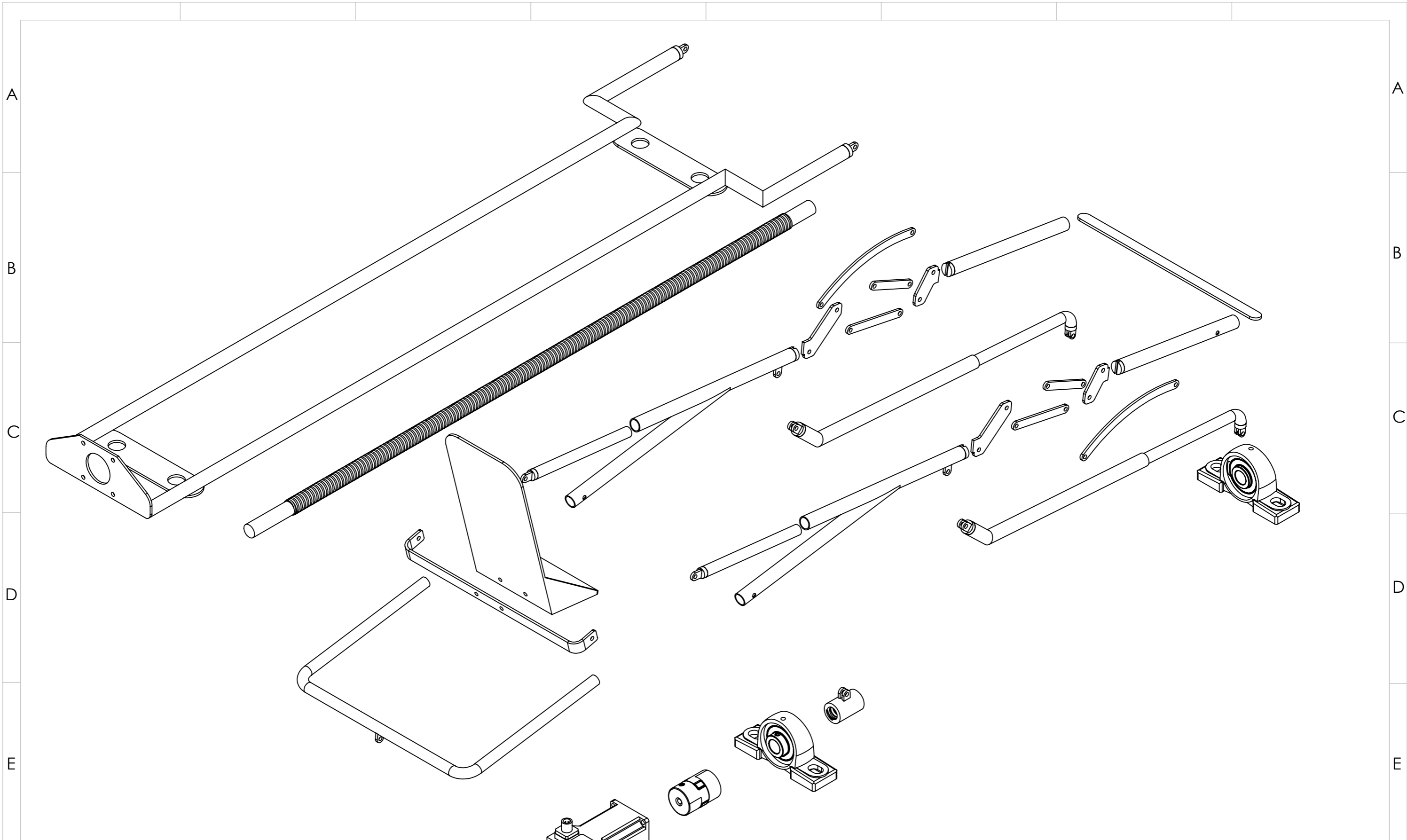
A  
B  
C  
D  
E  
F

A  
B  
C  
D  
E  
F

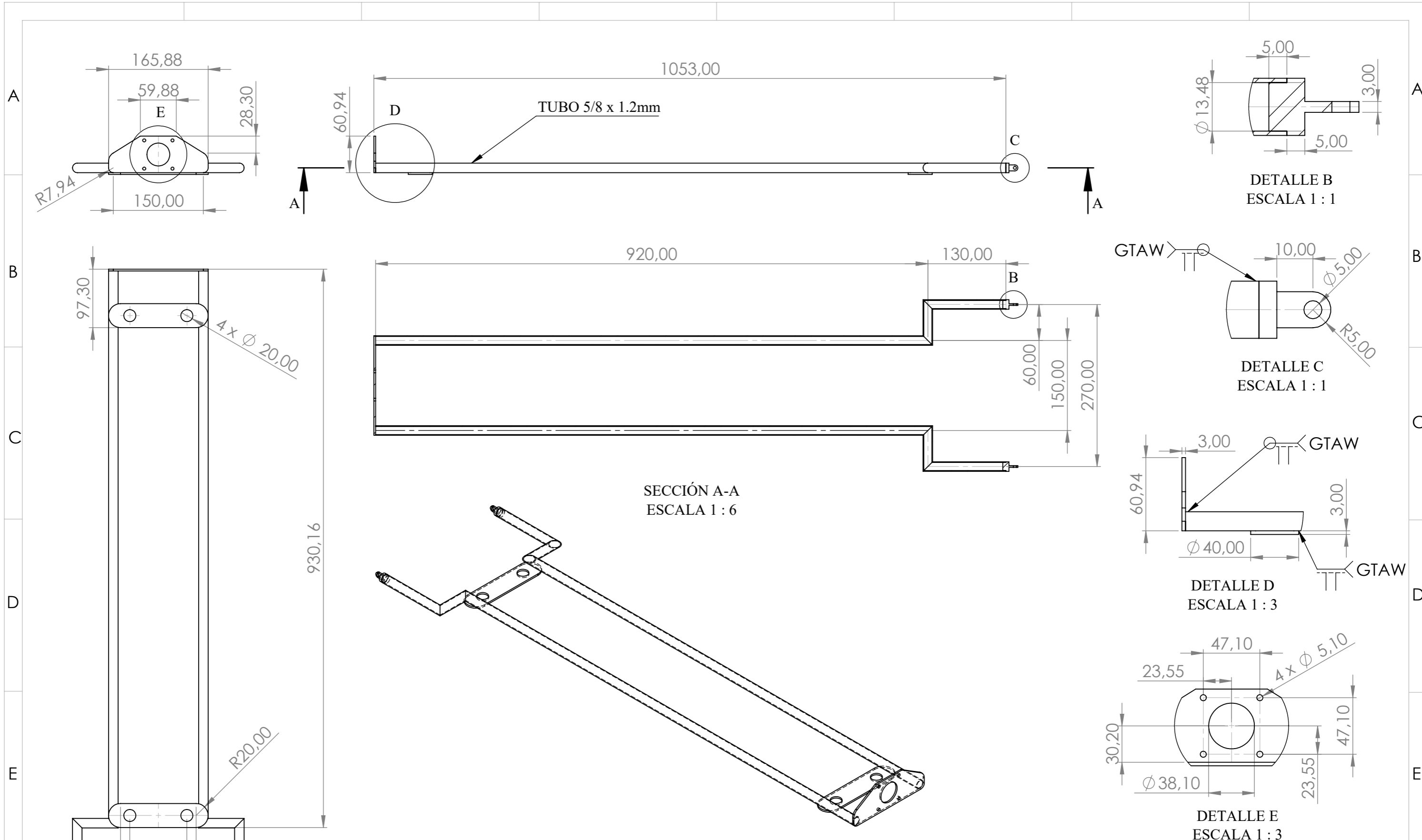


				PESO:			
				12.7 kg			
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	DIBUJO DE CONJUNTO	1:5
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO:	
						CIME20-002	
Edición	Modificación	Fecha	Nombre				

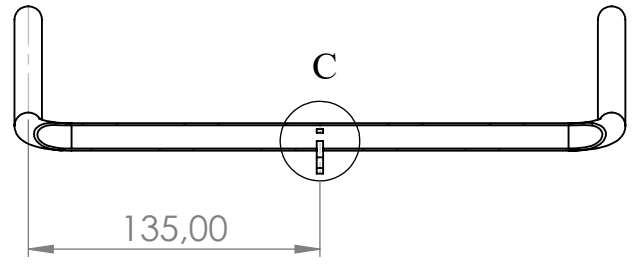
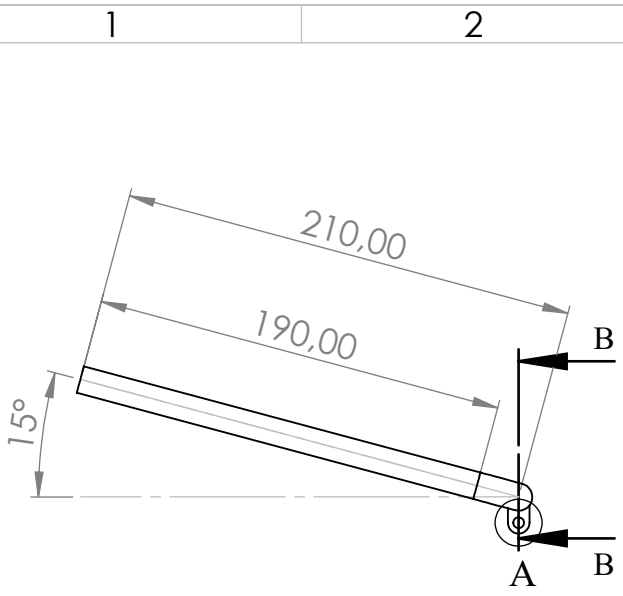




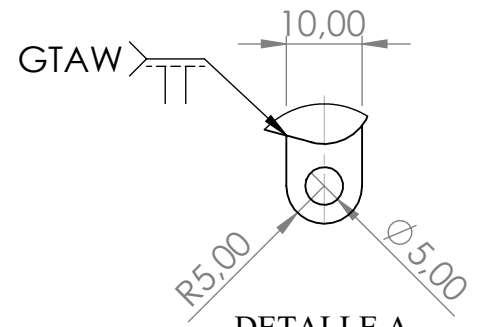
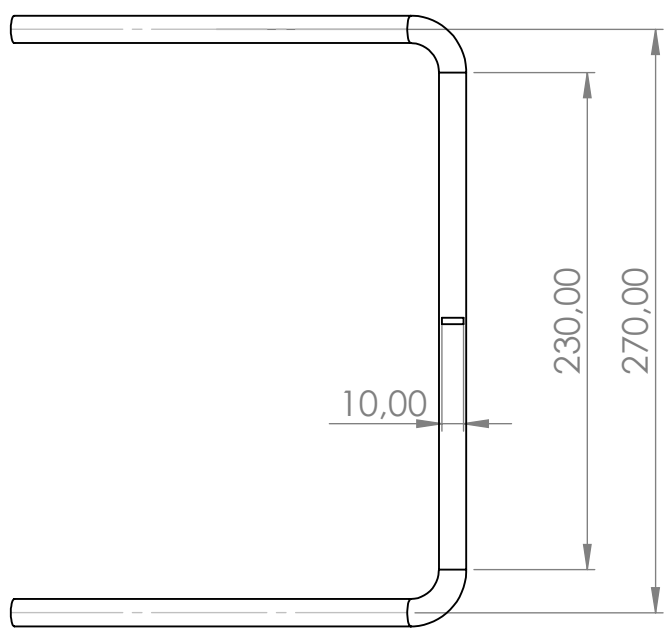
				Fecha:	Nombre:	TÍTULO: <b>DIBUJO ISOMÉTRICO EN EXPLOSIÓN</b>	ESC: <b>1:4</b>
				Dibujó:	Grijalva C.		
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.	N° DE DIBUJO: CIME20-003	
Edición	Modificación	Fecha	Nombre	<b>UTN</b>			



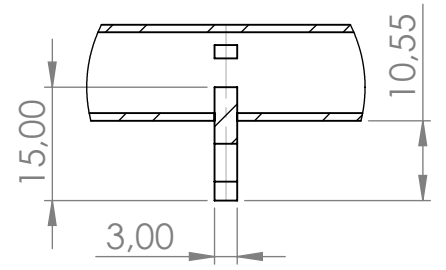
				PESO:	TOLERANCIA:	MATERIAL:	
				1.22 kg	+/- 1mm	ACERO INOXIDABLE 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	BASE	1:6
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO:	
						CIME20-001-001	
Edición	Modificación	Fecha	Nombre				



SECCIÓN B-B  
ESCALA 2 : 7



DETALLE A  
ESCALA 1 : 1

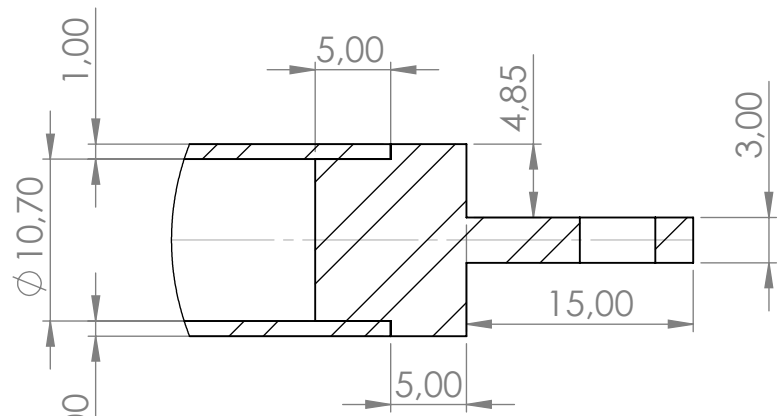
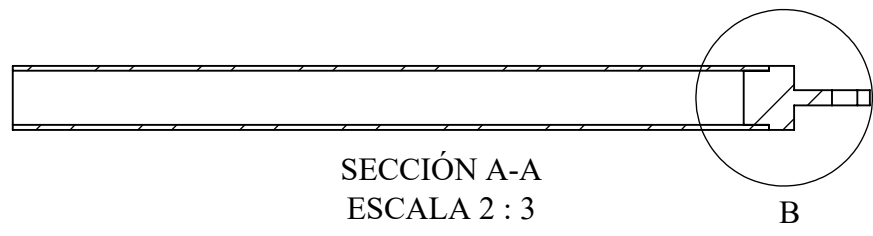
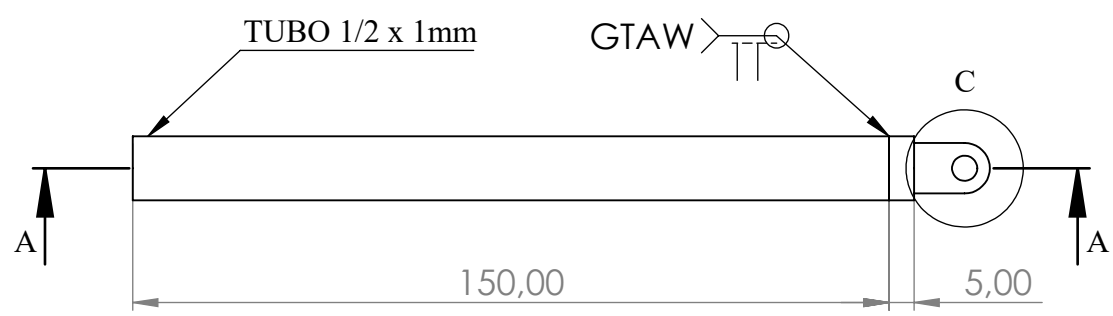
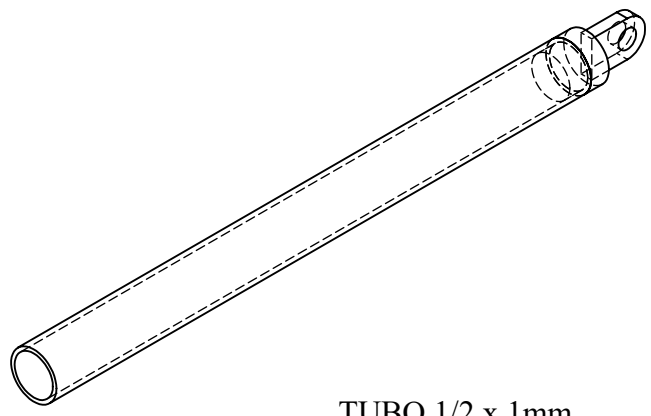


DETALLE C  
ESCALA 1 : 1

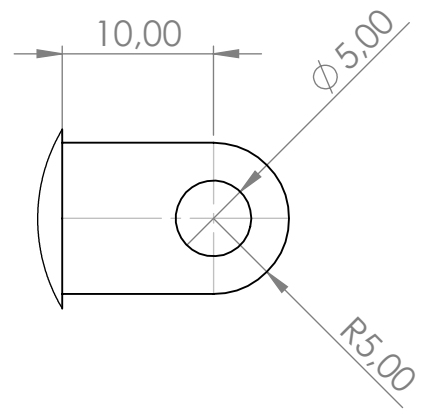
				PESO: 200.39 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO: TUBO SOPORTE TIBIA EXTENSIÓN	ESC: 2:7
				Dibujó:	Grijalva C.		
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.	Nº DE DIBUJO: CIME20-001-002	
Edición	Modificación	Fecha	Nombre	<b>UTN</b>			

1 2 3 4

A  
B  
C  
D  
E



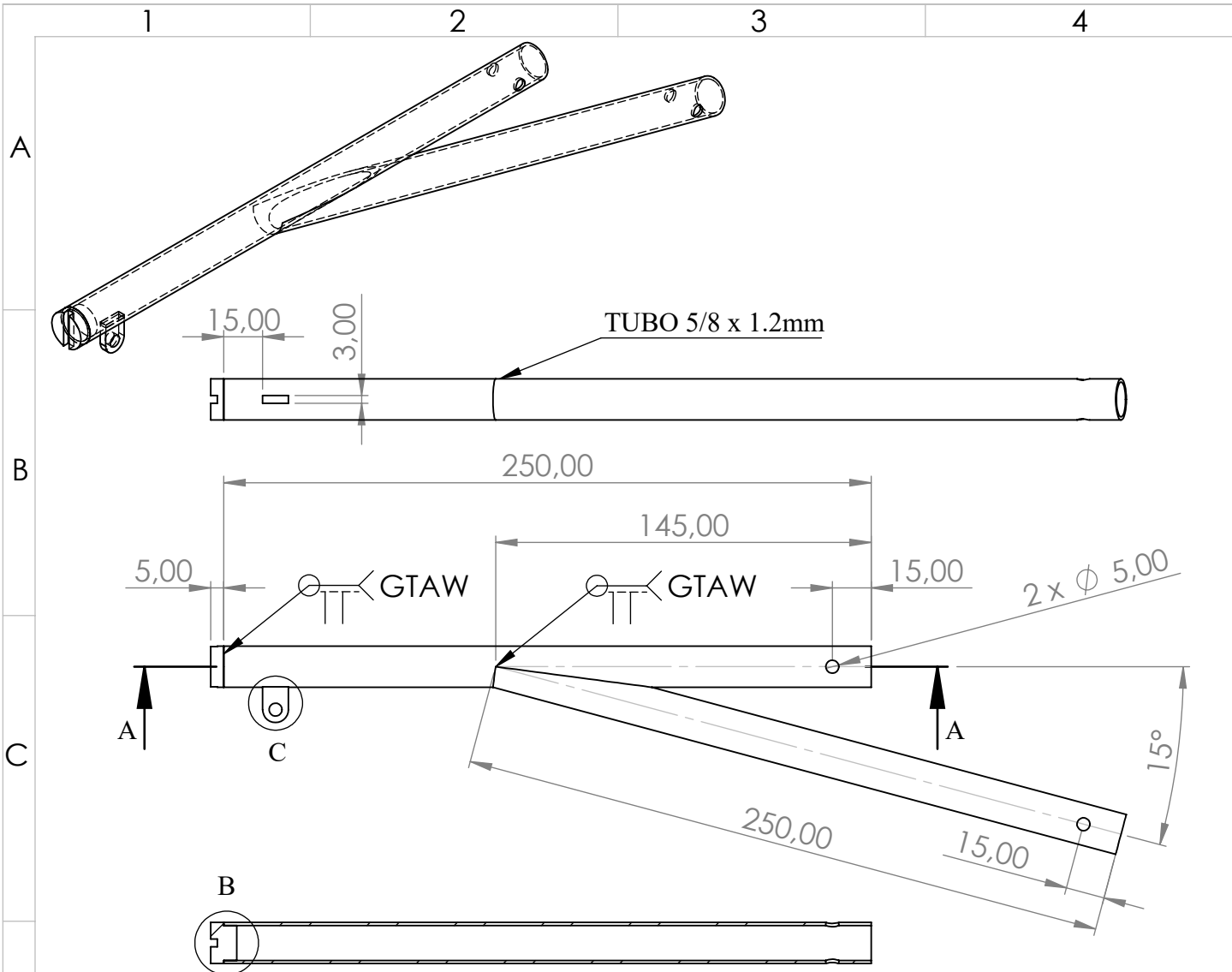
DETALLE B  
ESCALA 2 : 1



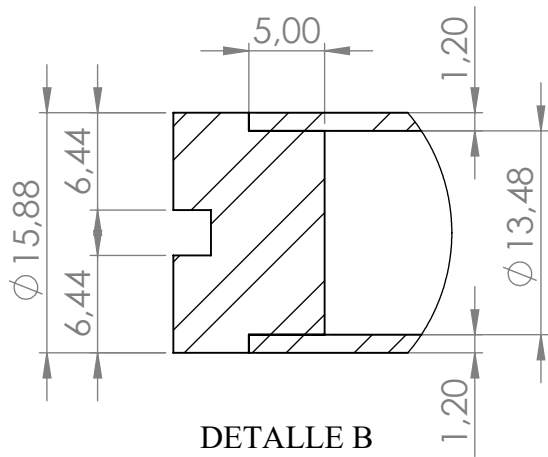
DETALLE C  
ESCALA 2 : 1

				PESO: 55.64 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO: TUBO TIBIA EXTENSIÓN	
				Dibujó:	Grijalva C.	ESC: 2:3	
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO: CIME-001-003	
Edición	Modificación	Fecha	Nombre				

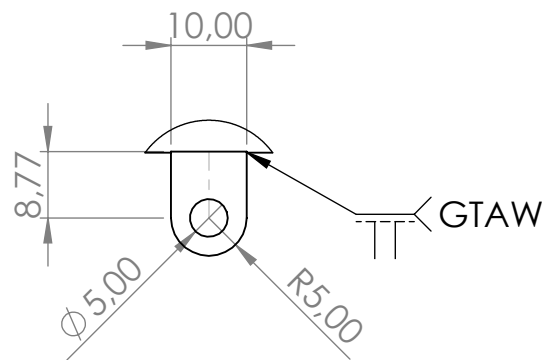




SECCIÓN A-A  
ESCALA 2 : 5

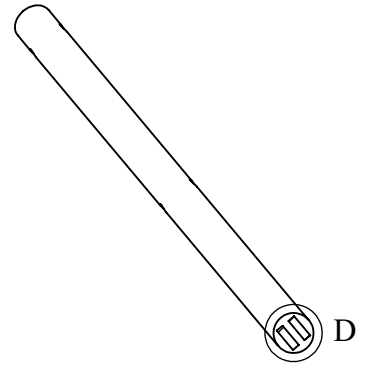
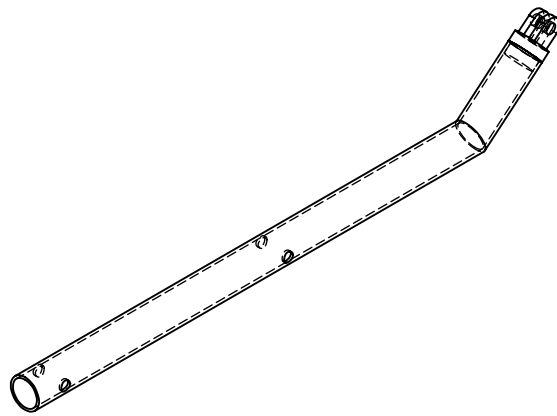


DETALLE B  
ESCALA 2 : 1

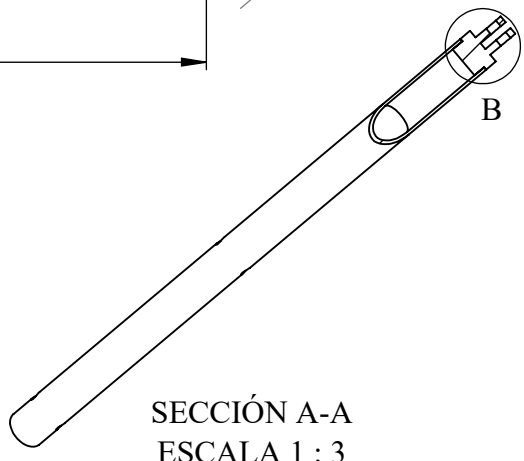
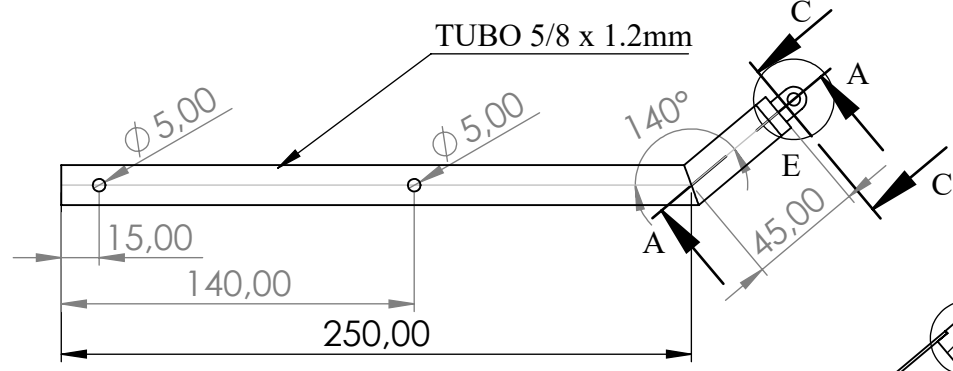


DETALLE C  
ESCALA 1 : 1

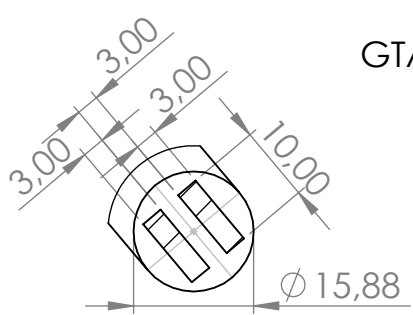
				PESO: 226.02 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO: TUBO TIBIA	
				Dibujó:	Grijalva C.	ESC: 2:5	
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO: CIME20-001-004	
Edición	Modificación	Fecha	Nombre				



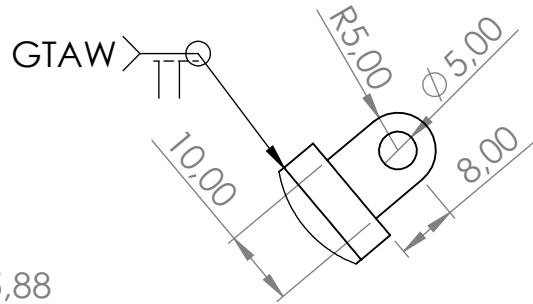
SECCIÓN C-C  
ESCALA 1 : 3



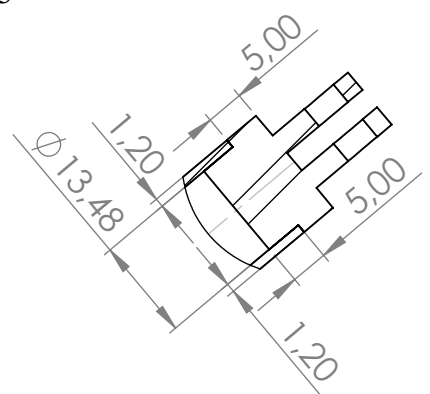
SECCIÓN A-A  
ESCALA 1 : 3



DETALLE D  
ESCALA 1 : 1

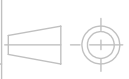


DETALLE E  
ESCALA 1 : 1



DETALLE B  
ESCALA 1 : 1

				PESO:	TOLERANCIA:	MATERIAL:	
				146 g	+/- 1mm	ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	TUBO BRAZO	1:2
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO:	CIME20-001-005
Edición	Modificación	Fecha	Nombre				



1 2 3 4

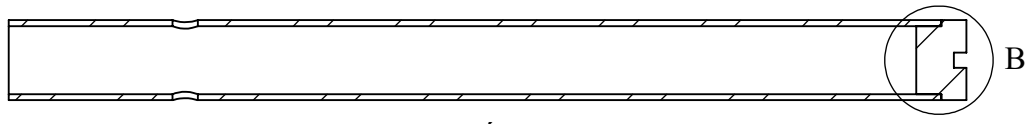
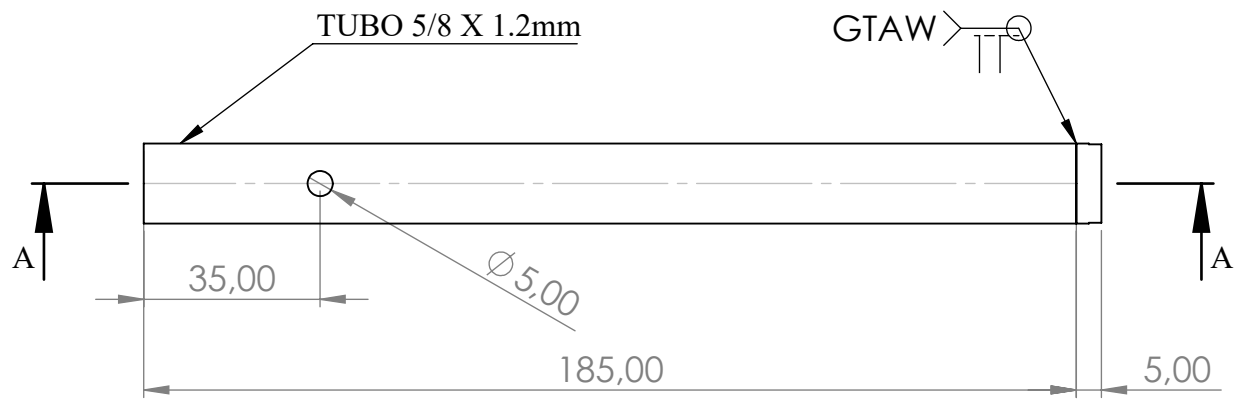
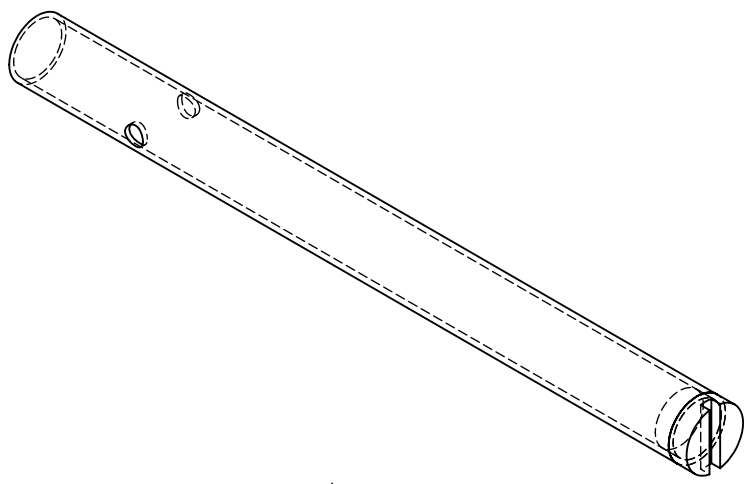
A

B

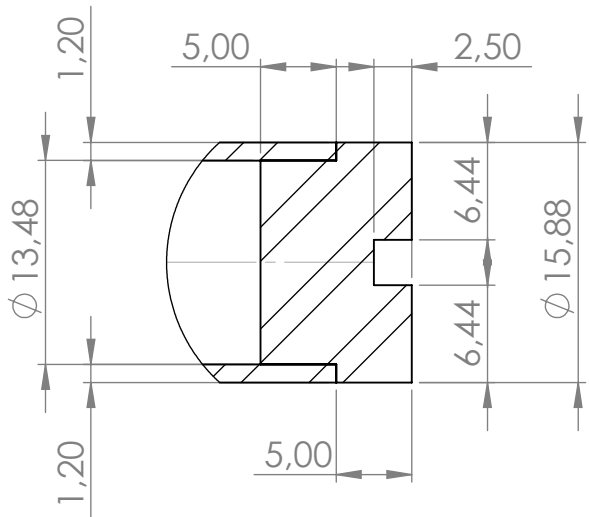
C

D

E



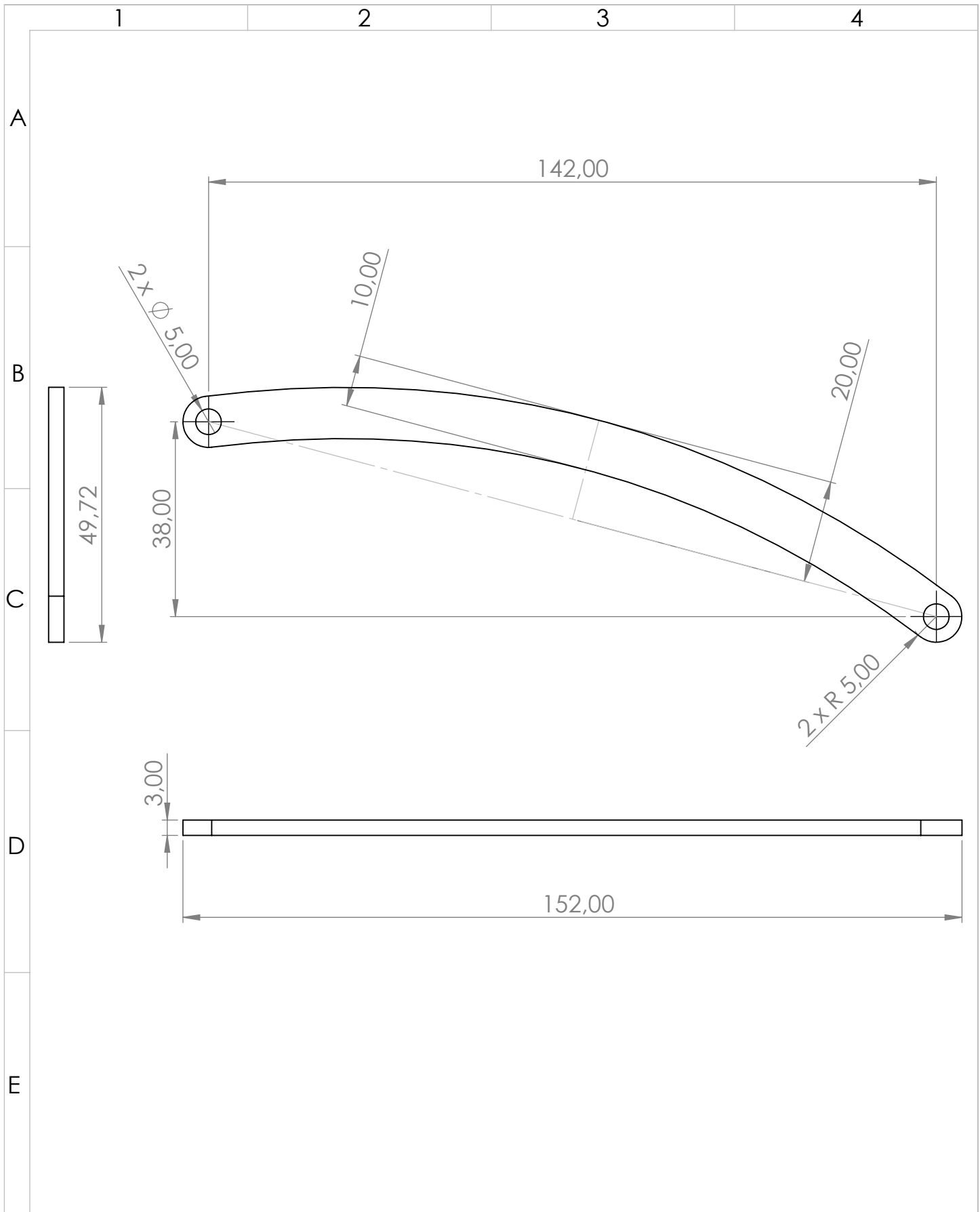
SECCIÓN A-A  
ESCALA 2 : 3



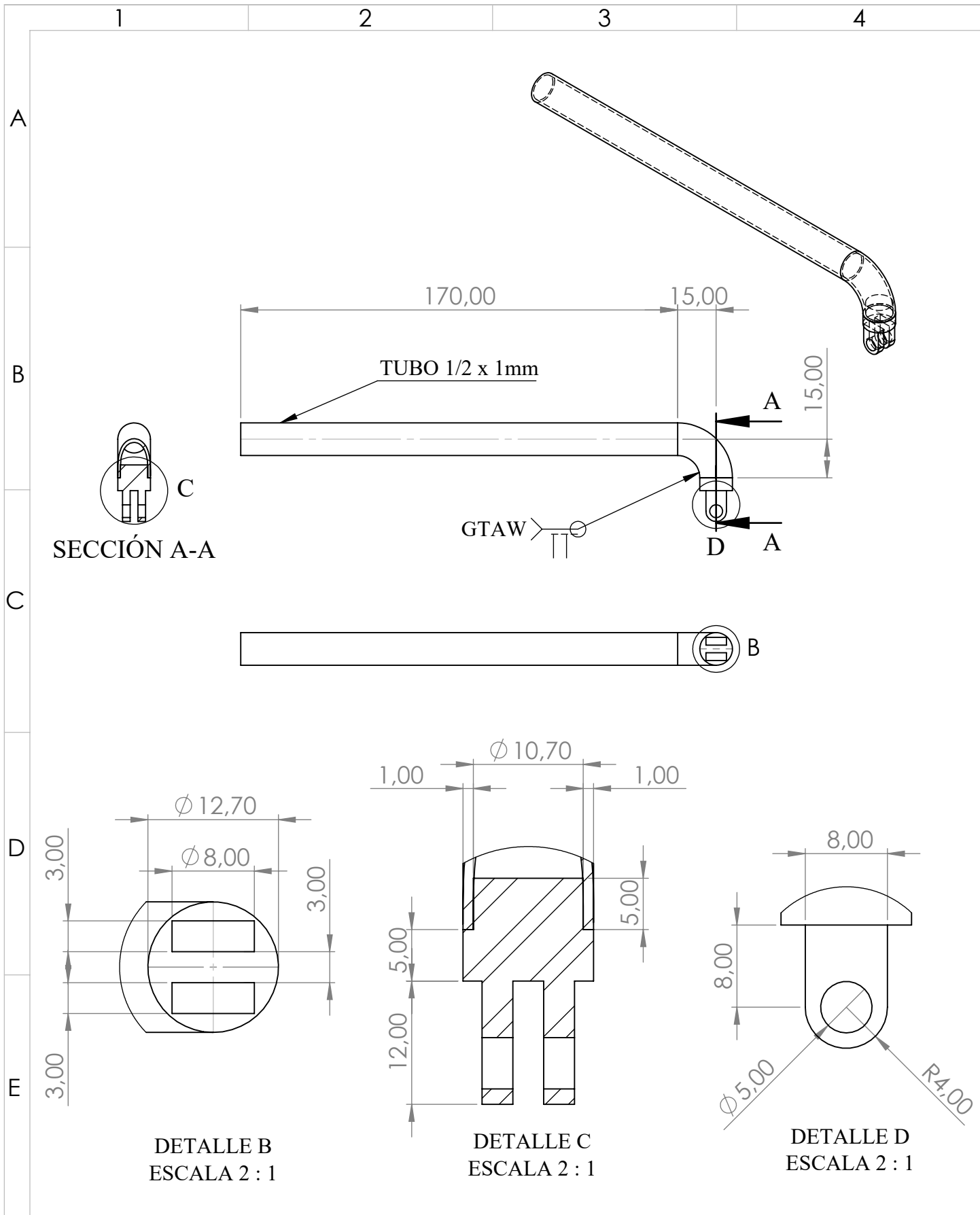
DETALLE B  
ESCALA 2 : 1

				PESO:	TOLERANCIA:	MATERIAL:	
				94.17 g	+/- 1mm	ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	TUBO FÉMUR	2:3
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO:	CIME20-001-006
Edición	Modificación	Fecha	Nombre				

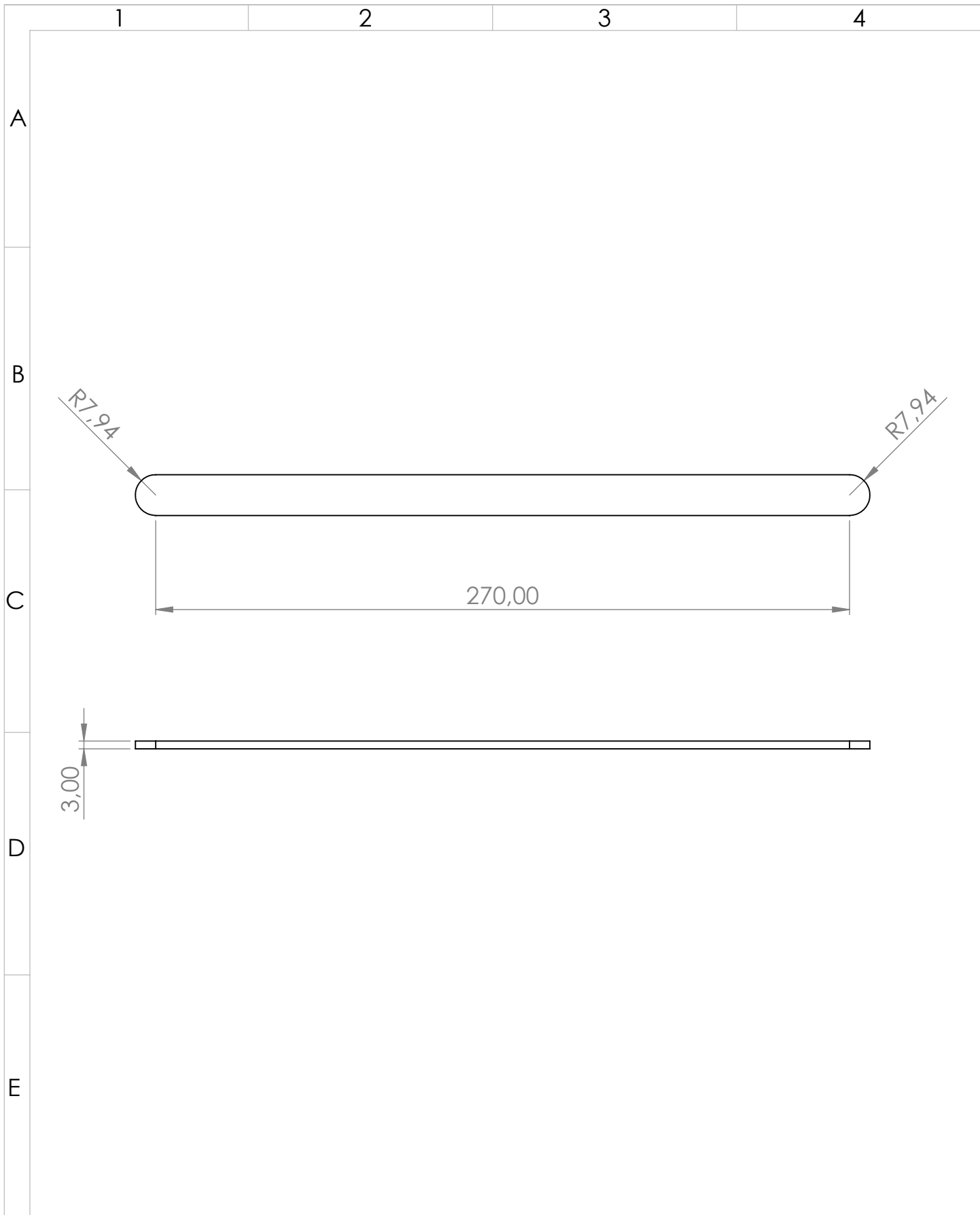




				PESO:	TOLERANCIA:	MATERIAL:		
				37.18 g	+/- 1mm	ACERO INOXIDABLE 304		
				Fecha:	Nombre:	TÍTULO:	ESC:	
				Dibujó:	Grijalva C.	BARRA CIRCULAR BRAZO-FÉMUR	1:2	
				Revisó:	Ing. Gamez B.			
				Aprobó:	Ing. Gamez B.	Nº DE DIBUJO:		
				<b>UTN</b>		CIME20-001-007		
Edición	Modificación	Fecha	Nombre					



				PESO:	TOLERANCIA:	MATERIAL:			
				68.92 g	+/- 1mm	ACERO AISI 304			
				Fecha:	Nombre:	TÍTULO:	ESC:		
				Dibujó:	Grijalva C.			TUBO BRAZO EXTENSIÓN	1:2
				Revisó:	Ing. Gamez B.				
				Aprobó:	Ing. Gamez B.				
				UTN		Nº DE DIBUJO:	CIME20-001-008		
Edición	Modificación	Fecha	Nombre						



				PESO:	TOLERANCIA:	MATERIAL:			
				107.62 g	+/- 1mm	ACERO INOXIDABLE 304			
				Fecha:	Nombre:	TÍTULO:	ESC:		
				Dibujó:	Grijalva C.			BARRA SOPORTE FÉMUR	1:2
				Revisó:	Ing. Gamez B.				
				Aprobó:	Ing. Gamez B.	Nº DE DIBUJO:			
				UTN		CIME20-001-009			
Edición	Modificación	Fecha	Nombre						



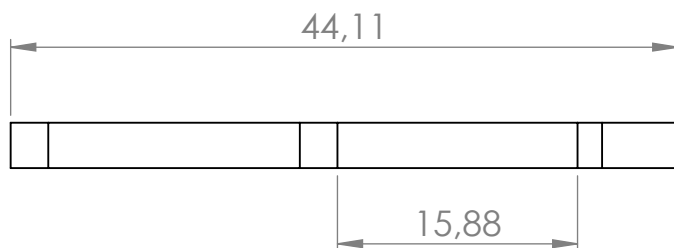
1

2

3

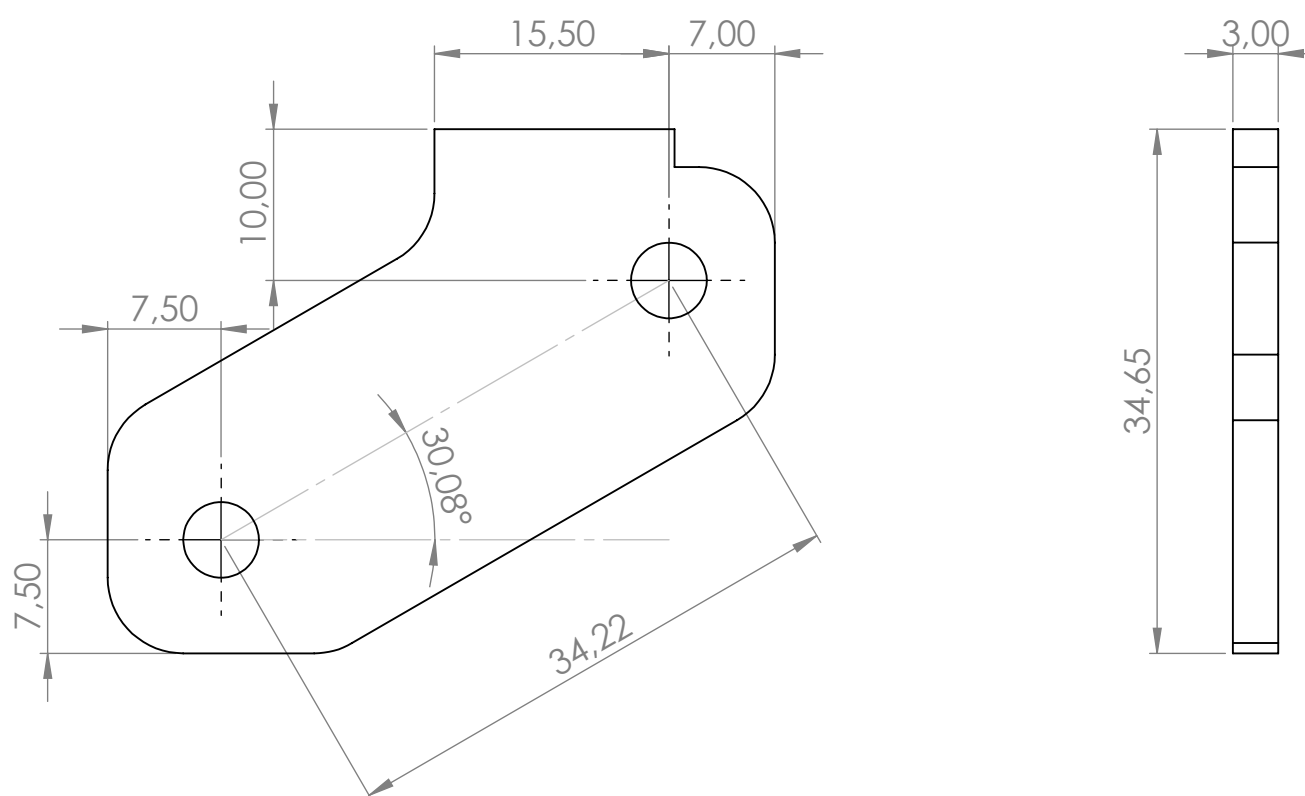
4

A



B

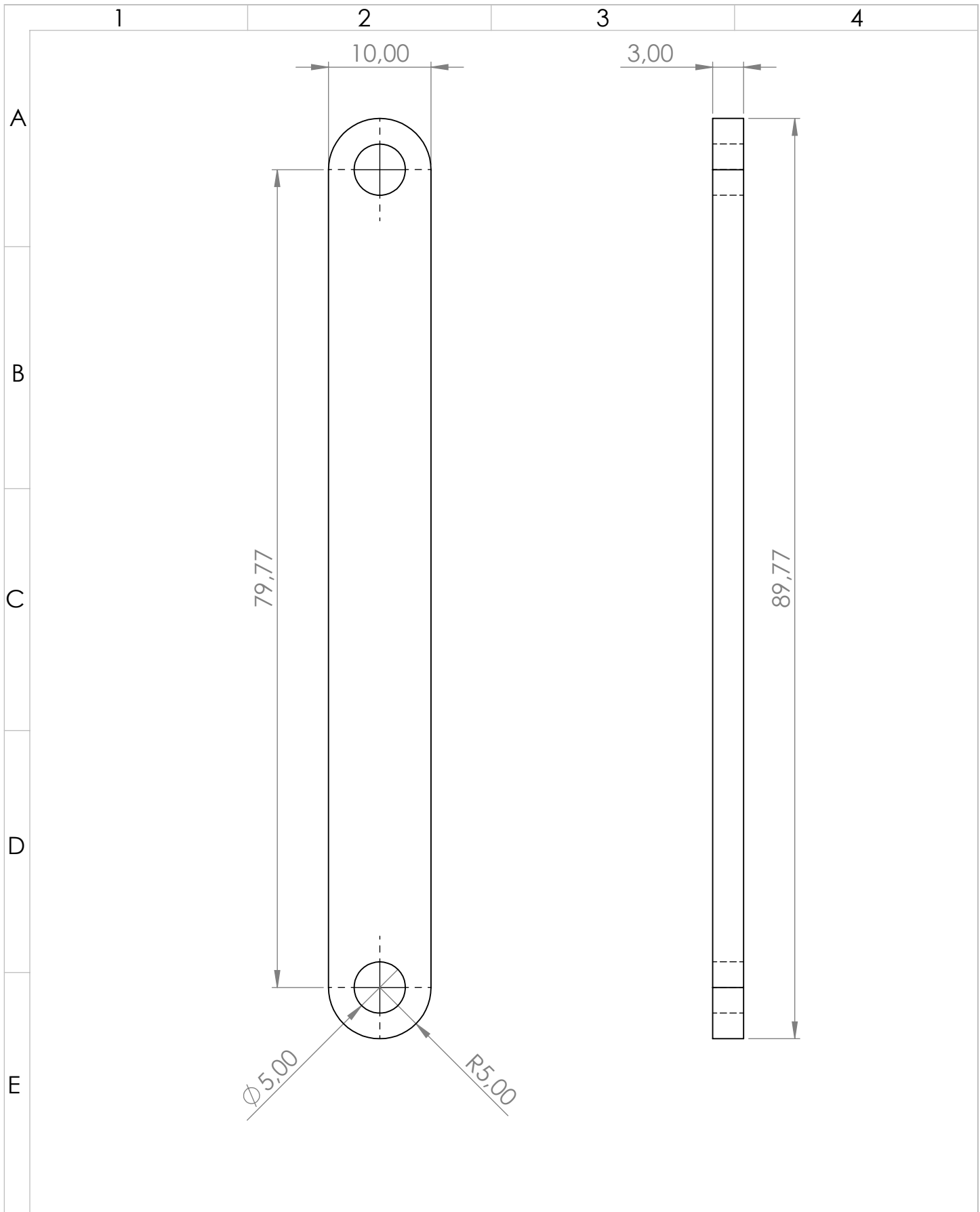
C




D

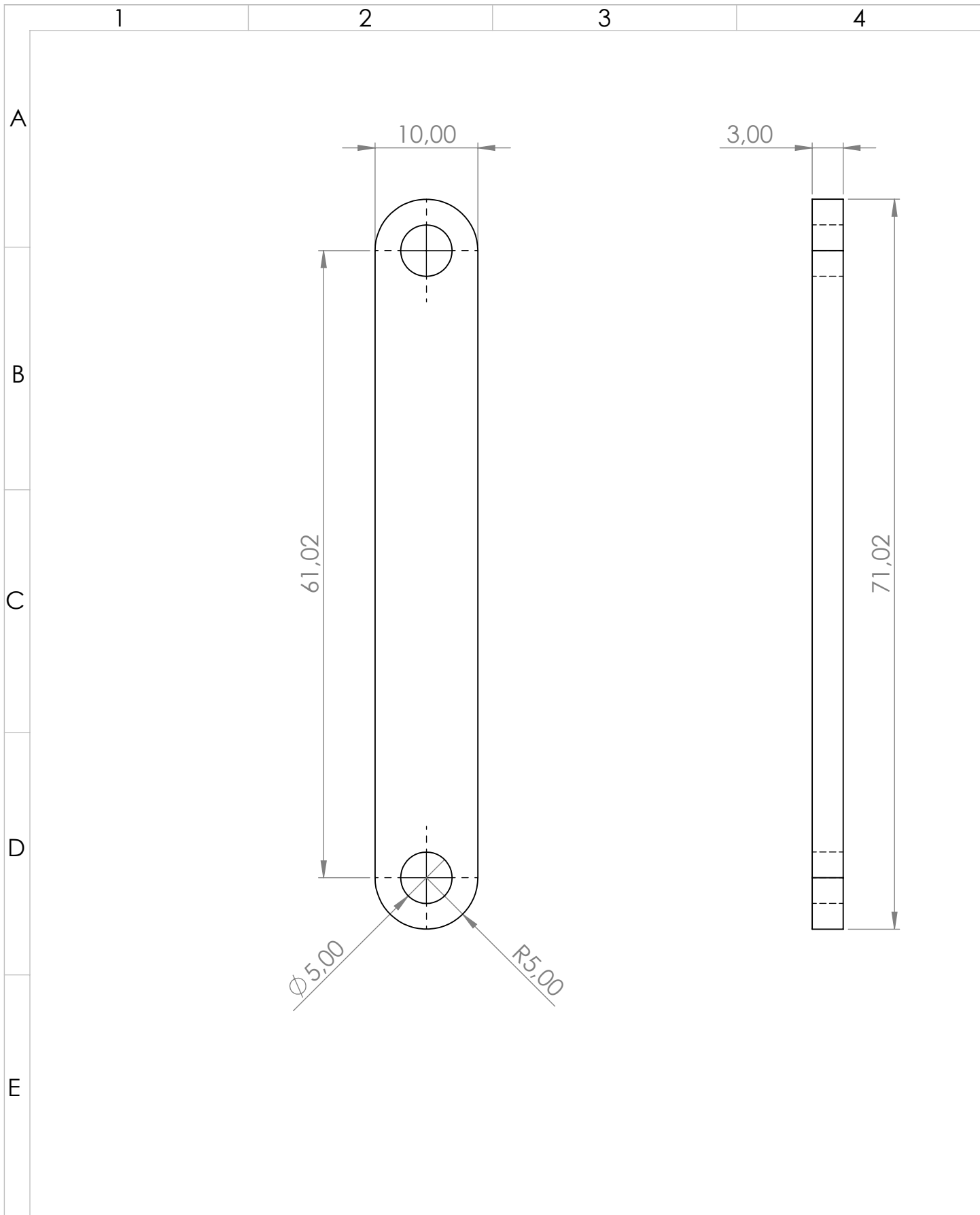
E


				PESO: 22.21 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO INOXIDABLE 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	BARRA B	2:1
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO:	
Edición	Modificación	Fecha	Nombre			CIME20-001-010	

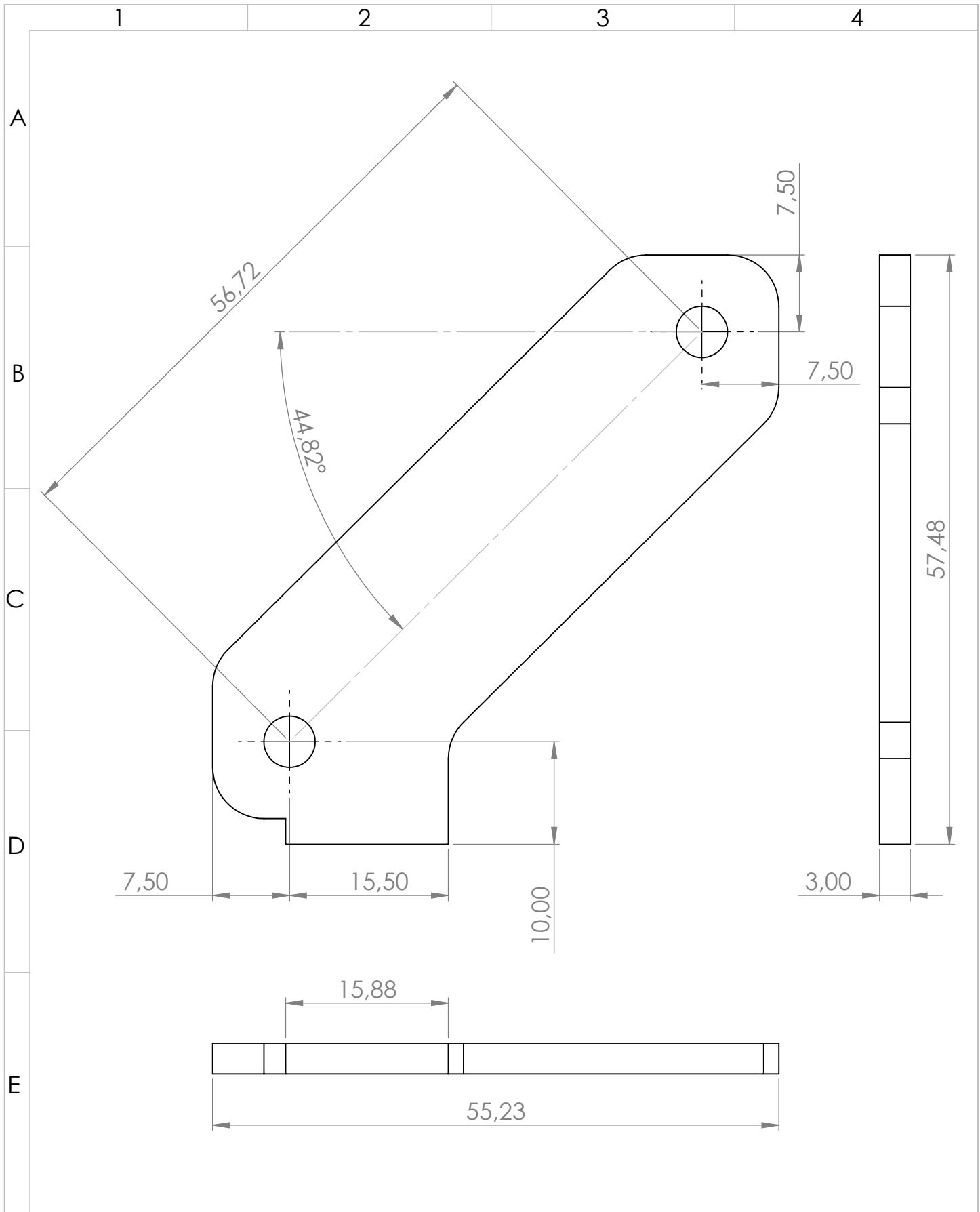



				PESO:	TOLERANCIA:	MATERIAL:	
				15.59 g	+/- 1mm	ACERO INOXIDABLE 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	BARRA C	2:1
				Revisó:	Ing. Erazo V.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO:	
Edición	Modificación	Fecha	Nombre			CIME20-001-011	

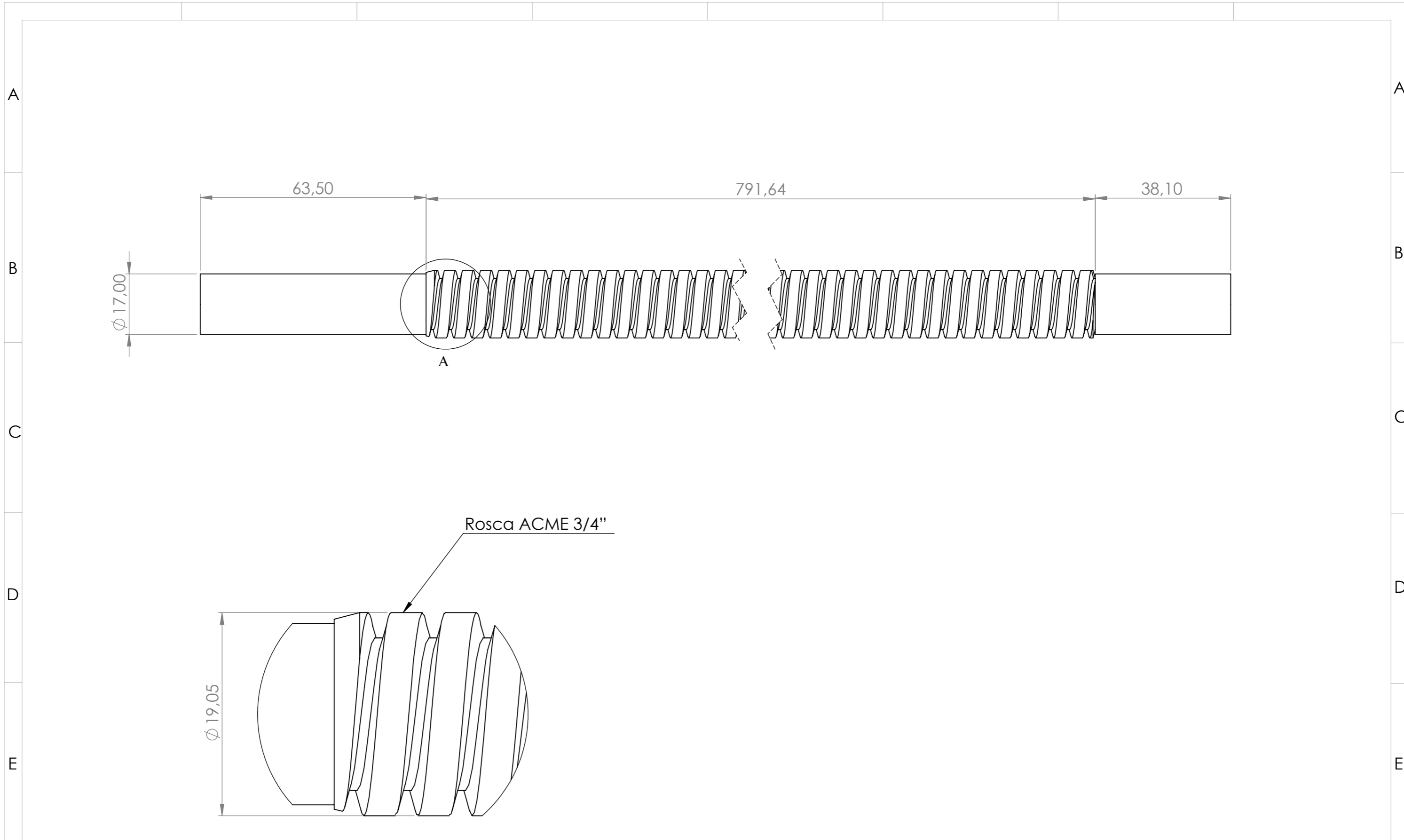




				PESO: 15.59 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO INOXIDABLE 304	
				Fecha:	Nombre:	TÍTULO: BARRA A	ESC: 2:1
				Dibujó:	Grijalva C.		
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO: CIME20-001-012	
Edición	Modificación	Fecha	Nombre				

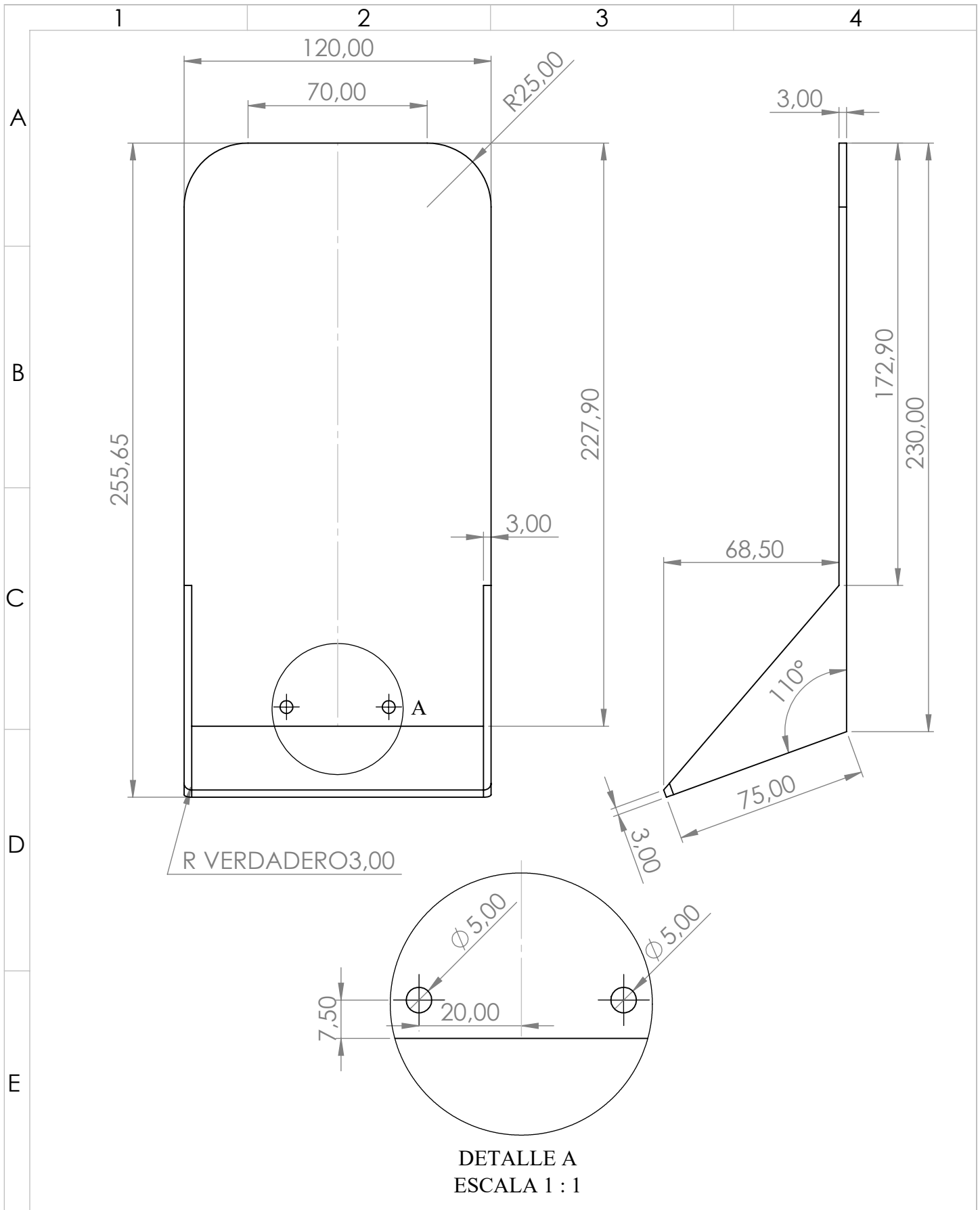



				PESO: 34.77 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO INOXIDABLE 304	
				Fecha:	Nombre:	TÍTULO: BARRA D	ESC: 2:1
				Dibujó:	Grijalva C.		
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO: CIME20-001-013	
Edición	Modificación	Fecha	Nombre				

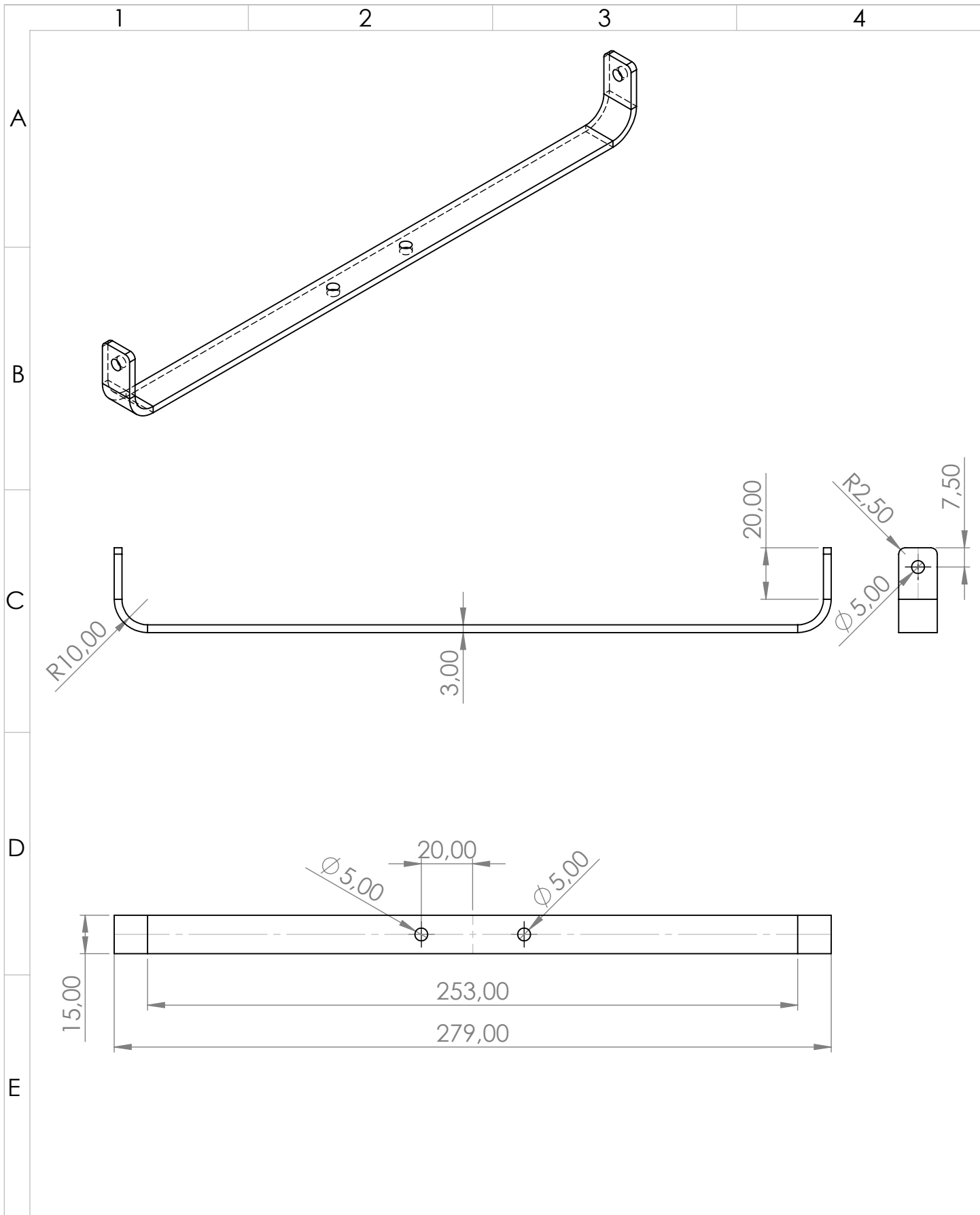



DETALLE A  
ESCALA 3 : 1

				PESO: 1750.3 g	TOLERANCIA: +/- 1mm	MATERIAL: ACERO AISI 1018	
					Fecha:	Nombre:	TÍTULO: TORNILLO DE POTENCIA
					Dibujó:	Grijalva C.	ESC: 1:1
					Revisó:	Ing. Gamez B.	
					Aprobó:	Ing. Gamez B.	
				UTN		N° DE DIBUJO: CIME20-001-014	
Edición	Modificación	Fecha	Nombre				



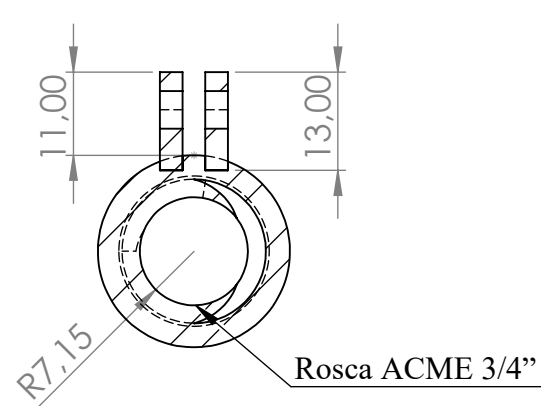
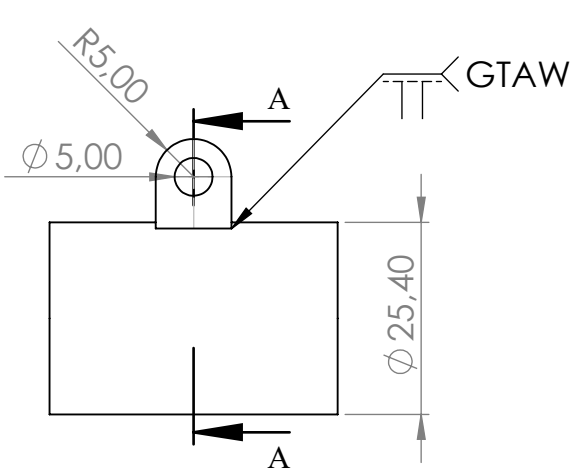
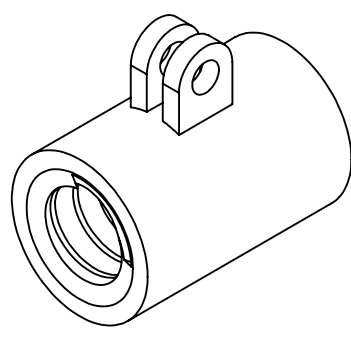
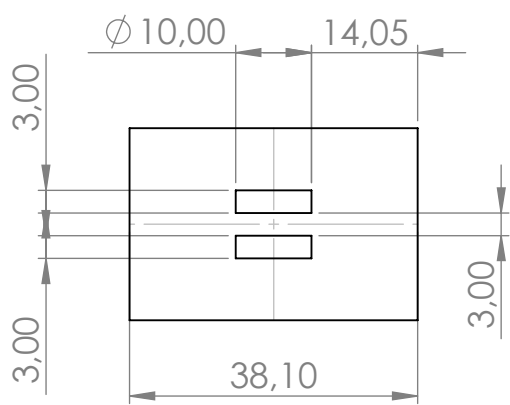
				PESO:	TOLERANCIA:	MATERIAL:	
				955.29 g	+/- 1mm	ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	SOPORTE PIE	1:2
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		N° DE DIBUJO:	CIME20-001-015
Edición	Modificación	Fecha	Nombre				
							



				PESO:	TOLERANCIA:	MATERIAL:	
				116.47 g	+/- 1mm	ACERO AISI 304	
				Fecha:	Nombre:	TÍTULO:	
				Dibujó:	Grijalva C.	BRIDA SOPORTE PIE	
				Revisó:	Ing. Gamez B.	ESC:	
				Aprobó:	Ing. Gamez B.	1:2	
				UTN		Nº DE DIBUJO:	
Edición	Modificación	Fecha	Nombre			CIME20-001-016	

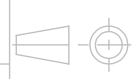
1 2 3 4

A  
B  
C  
D  
E



SECCIÓN A-A

				PESO:	TOLERANCIA:	MATERIAL:	
				95.31 g	+/- 1mm	ACERO AISI 1018	
				Fecha:	Nombre:	TÍTULO:	ESC:
				Dibujó:	Grijalva C.	TUERCA	1:1
				Revisó:	Ing. Gamez B.		
				Aprobó:	Ing. Gamez B.		
				UTN		Nº DE DIBUJO:	CIME20-001-017
Edición	Modificación	Fecha	Nombre				



## ANEXO 2

### Especificaciones del acople de mandíbula “Lovejoy” L035 a L225

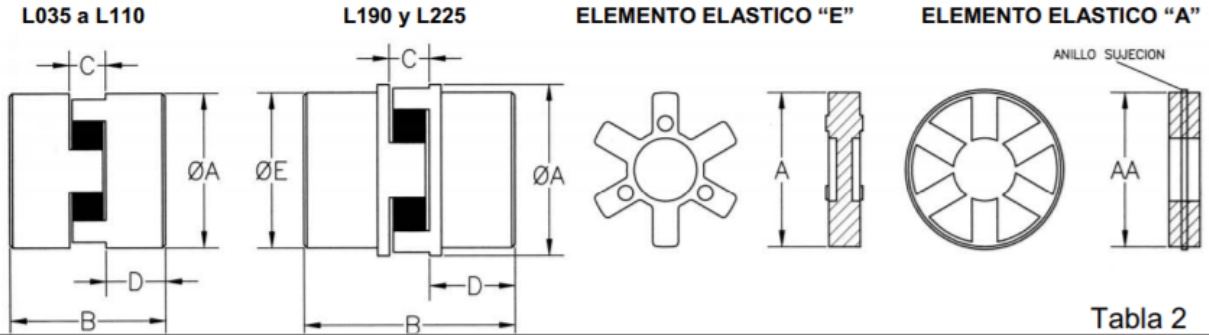
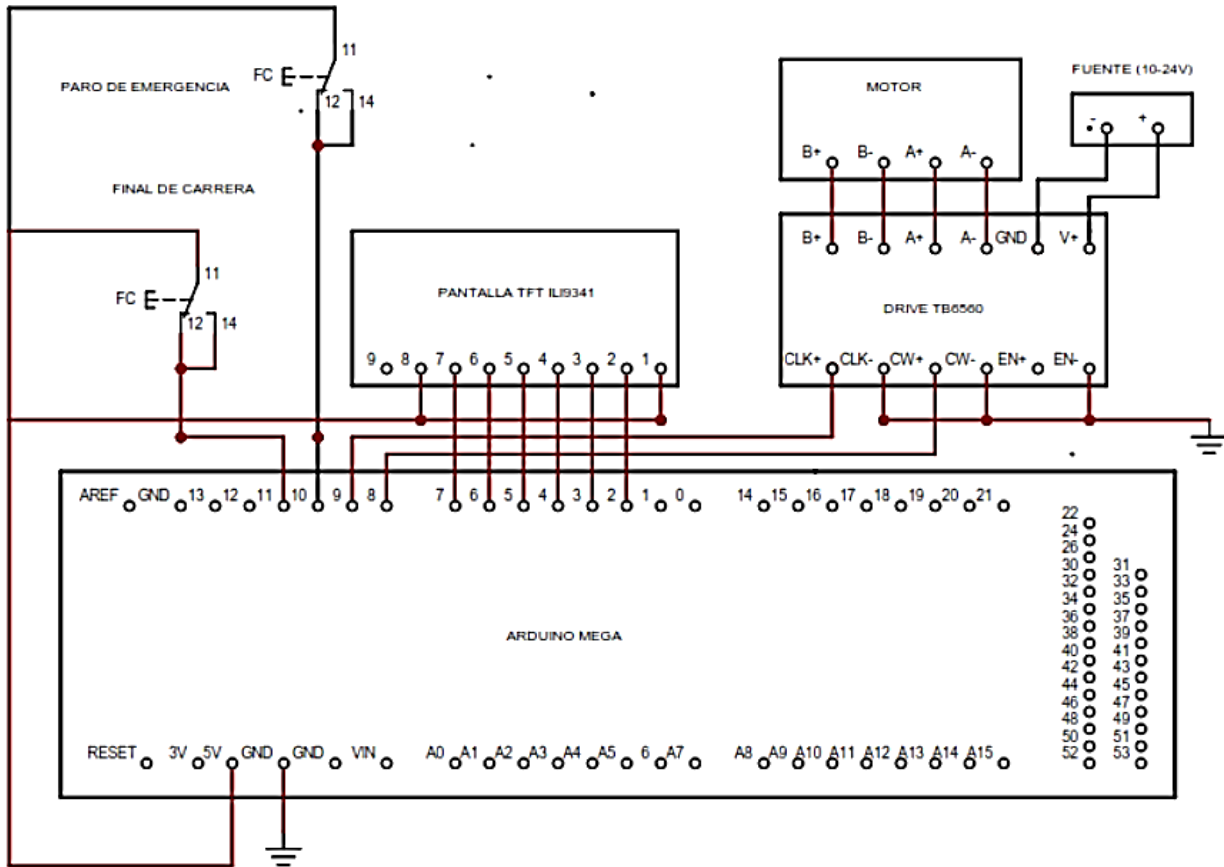


Tabla 2

DIMENSIONES ( cotas en mm )									MATERIAL MANGON				
									ACERO SINT.	FUNDICION	ALUMINIO	ACERO INOX.	BRONCE
(x) Material standard													
TIPOS	d máx.	A	B	C	D	E	AA	Peso Kg.	ACERO SINT.	FUNDICION	ALUMINIO	ACERO INOX.	BRONCE
L.035	10	15.9	20.6	7.1	6.8	15.9		0.05	x			x	
L.050	15	27.4	43.6	12.2	15.7	27.4		0.14	x		x	x	
L.070	19	34.5	50.8	12.2	19.0	34.5		0.27	x		x	x	
L.075	24	44.5	54.0	12.7	20.8	44.5		0.45	x		x	x	x
L.090	28	53.6	54.0	13.2	20.8	53.6	65	0.68	x		x	x	x
L.095	28	53.6	63.5	13.2	25.4	53.6	65	0.81	x		x	x	x
L.100	35	64.3	89.0	18.0	35.0	64.3	78	1.58	x		x	x	x
L.110	42	84.1	108.0	22.3	42.7	84.1	98	2.97	x		x	x	x
L.150	48	95.3	114.3	25.4	44.4	95.3	116	4.10	x		x	x	x
L.190	55	114.3	123.8	25.4	49.3	101.6	132	7.65		x		x	x
L.225	65	127.0	136.5	25.4	55.4	108.0	138	10.35		x		x	

# ANEXO 3

## Plano del diseño electrónico





# ANEXO 4

## Especificaciones del motor paso a paso “NEMA 23”

SPECIFICATION	CONNECTION	BIPOLAR
VOTAGE(VOC)		4.20
AMPS/PHASE		3.50
RESISTANCE/PHASE(Ohms)@25°C		1.20±10%
INDUCTANCE/PHASE(mH)@1KHz		5.00±20%
HOLDING TORQUE(Nm) [lb-in]		3.00[26.55]
STEP ANGLE(°)		1.80
STEP ACCURACY(NON-ACCUM)		±5.00%
ROTOR INERTIA(g-cm²)		800.00
WEIGHT(Kg) [lb]		1.60[3.53]
TEMPERATURE RISE:MAX,80°C ( MOTOR STANDSTILL;FOR 2PHASE ENERGIZED )		
AMBIENT TEMPERATURE -10°C--50°C[14°F--122°F]		
INSULATION RESISTANCE 100 Mohm ( UNDER NORMAL TEMPERATURE AND HUMIDITY )		
INSULATION CLASS B 130°C[266°F]		
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE )		
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A —	BLK	A
2	A  —	GRN	A
3	B —	RED	B
4	B  —	BLU	B

FULL STEP 2 PHASE-Ex. . WHEN FACING MOUNTING END (X)

STEP	A	B	A	B	
1	+	+	-	-	↓ CCW ↑ CW
2	-	+	+	-	
3	-	-	+	+	
4	+	-	-	+	

APVD	
CHKD	
DRN	
SCALE	SIGNATURE      DATE

### STEPPER MOTOR

23HS45-3504S

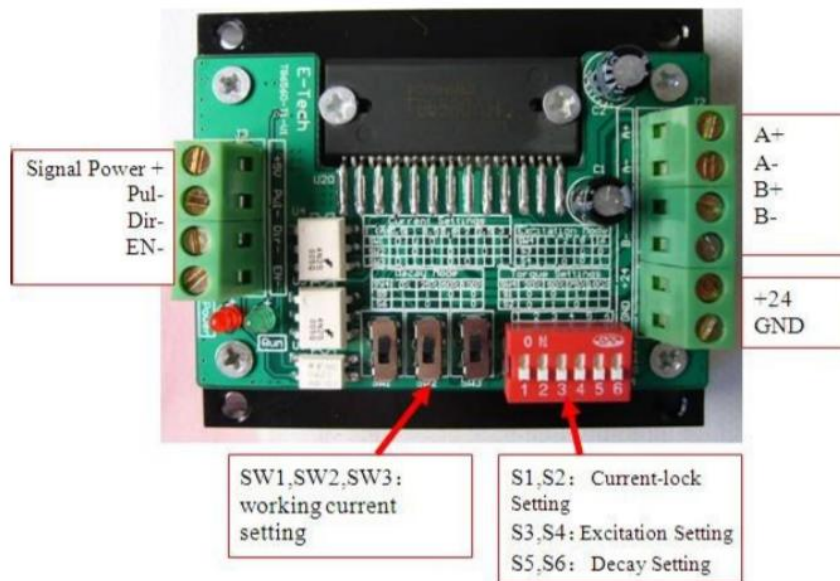
## ANEXO 5

### Especificaciones del controlador TB6560

Electrical Specifications( $T_j=25^\circ\text{C}$ )

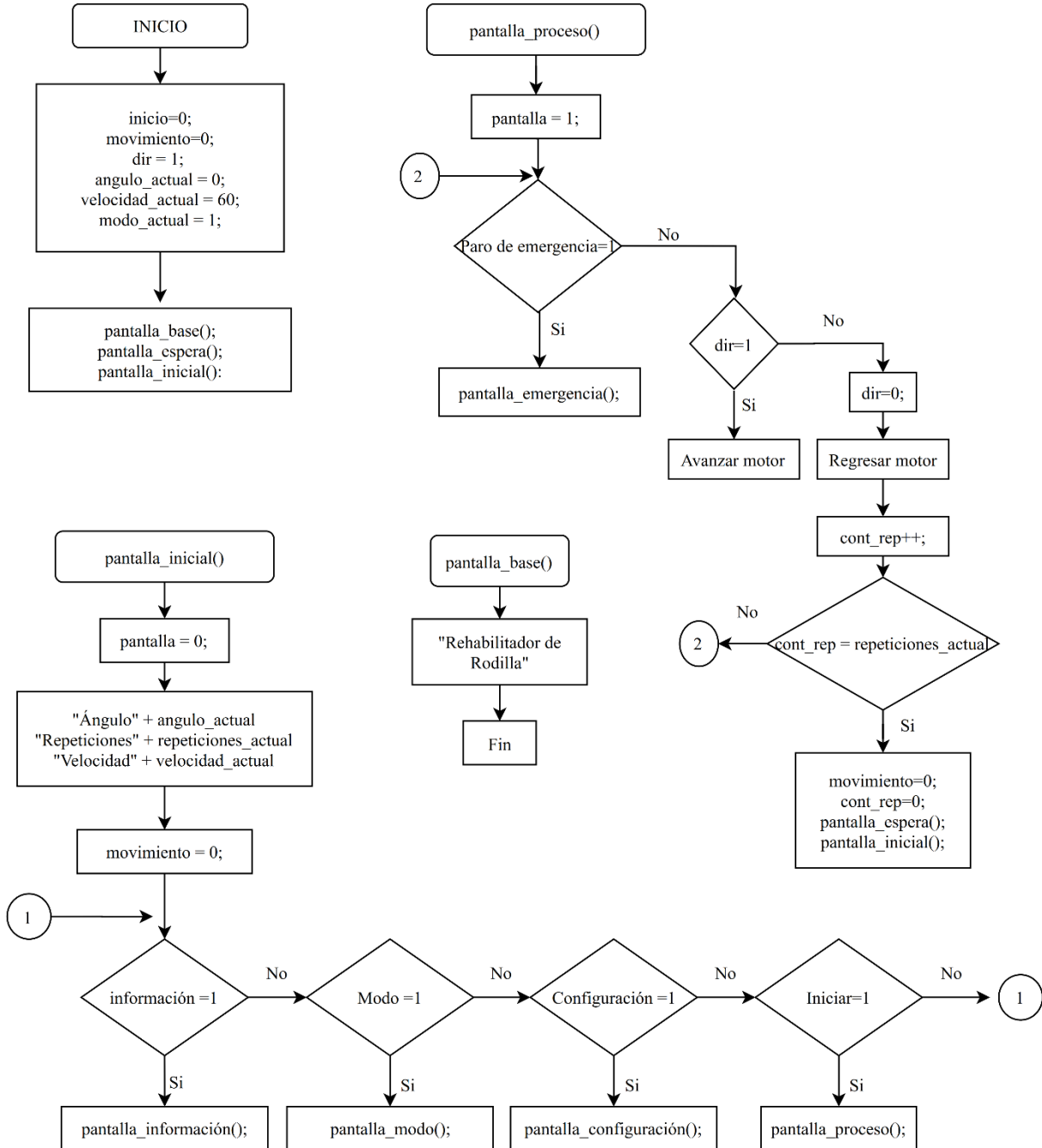
Parameters	TB6560-3AXIS			
	Min	Typical	Max	Unit
Output current	0.6	-	3	A
Input voltage	7	24	32	VDC
Inner Frequency	640		20000	Hz
Outer input frequency	0	-	16000	Hz
Connector Voltage	H	4.5	5.5	VDC
	L	0	0	0.5

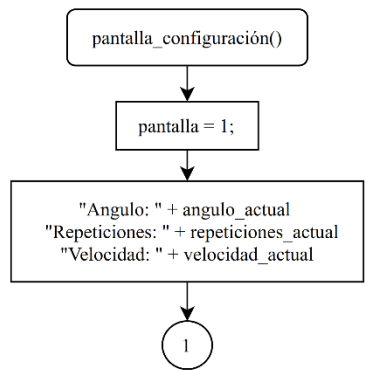
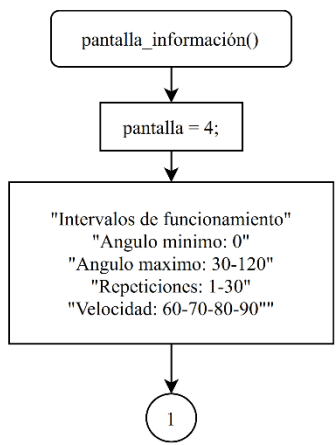
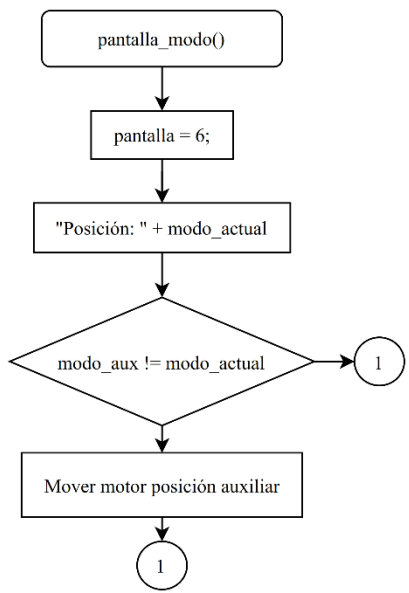
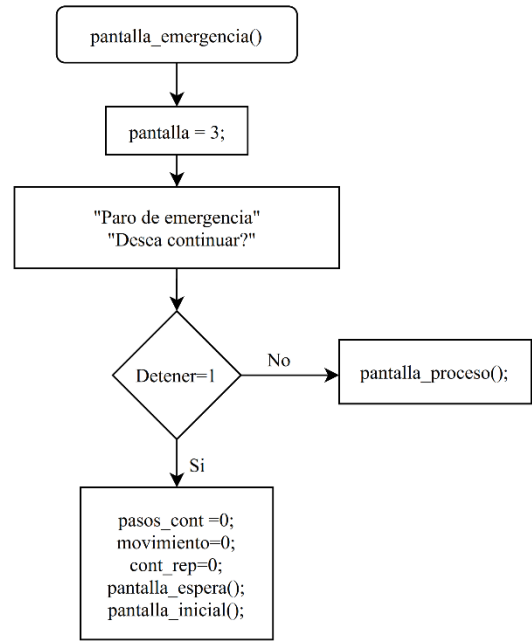
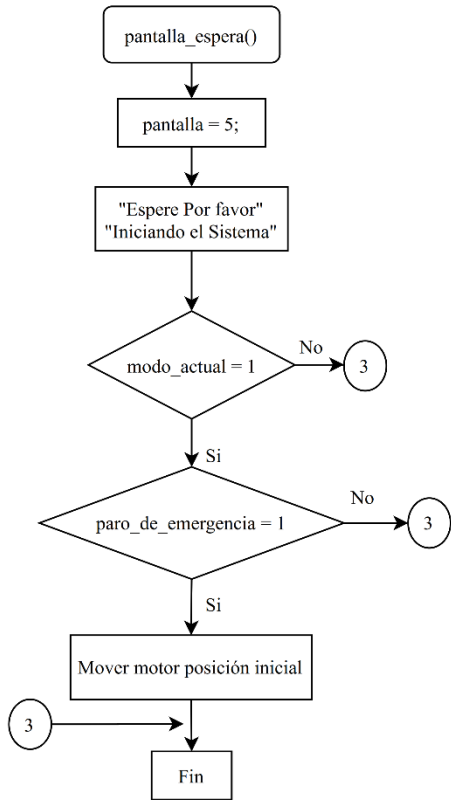
TB6560-3Axis Interface Definition



## ANEXO 6

### Flujograma del dispositivo





## ANEXO 7

### Código de programación del dispositivo

```
/* Librerías */
#include <TimerOne.h>
#include <EEPROM.h>
#include <UTFT.h>
#include <URTough.h>
/* Configuración pantalla y touch */
UTFT myGLCD(ITDB32WC, 38, 39, 40, 41);
URTough myTouch(6, 5, 4, 3, 2);
extern uint8_t SmallFont[];
/* Variables */
int x, y;
byte inicio = 0;
byte pantalla = 5;
byte repeticiones_actual = 0;
byte angulo_actual = 0;
byte velocidad_actual = 60;
byte modo_actual = 1;
byte repeticiones_aux = 0;
float angulo_aux = 0;
byte velocidad_aux = 60;
byte modo_aux = 1;
byte cont_rep = 0;
unsigned long angulo_v[2];
// Coeficientes curvas de movimiento
float coef[3][3] =
{
  143.26046, -131.48606, 36337.61842,
  134.87860, -132.66428, 29666.03135,
  150.49342, -149.81315, 32867.06796
};
boolean e = 0;
boolean a = 0;
boolean b = 1;
boolean dir = 1;
boolean pasos_ini = 0;
unsigned long pasos_cont = 0;
unsigned long pasos_lim = 5430;
unsigned long angulos[3][19] =
{
  5430, 7070, 8790, 10596, 12500, 14500, 16650, 18850, 21350,
  23850, 26650, 29650, 32900, 36500, 40450, 44950, 50050, 55950,
  60000,
  6972, 8421, 9945, 11551, 13249, 15050, 16968, 19018, 21220,
  23599, 26186, 29020, 32154, 35658, 39632, 44223, 49655, 56311,
  64906,
  7158, 8551, 10006, 11528, 13124, 14801, 16569, 18437, 20417,
  22525, 24777, 27195, 29806, 32641, 35744, 39172, 42998, 47330,
  52320
};
unsigned long velocidades[4] = {1333, 1500, 1714, 2000};
boolean movimiento = 0;
/* Variables lectura de interruptores*/
int buttonState[3];
int lastButtonState[3];
unsigned long lastDebounceTime[3];
unsigned long debounceDelay = 50;
byte reading[3];
byte pin_rd[3] = {10, 11, 12};

void setup()
{
  Serial.begin(115200);
  // Paro de emergencia
  pinMode(10, INPUT_PULLUP);

  // Finales de carrera
  pinMode(11, INPUT);
  pinMode(12, INPUT_PULLUP);
  // Control motor PAP
  pinMode(8, OUTPUT); //pasos
  pinMode(9, OUTPUT); //dirección
  digitalWrite(9, 1);
  dir = 1;
  // Variables de control almacenados
  angulo_actual = EEPROM.read(0);
  repeticiones_actual = EEPROM.read(1);
  velocidad_actual = EEPROM.read(2);
  modo_actual = EEPROM.read(3);
  modo_aux = modo_actual;
  delay(500);
  // Configuración Pantalla
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setFont(SmallFont);
  // Configuración Touch
  myTouch.InitTouch();
  myTouch.setPrecision(PREC_MEDIUM);
  delay(500);
  // Inicio del sistema
  pantalla_base();
  pantalla_espera();
  pantalla_inicial();
  movimiento = 0;
  Timer1.initialize(1333); // Velocidad de movimiento
  Timer1.attachInterrupt(pasos);
}

void loop()
{
  // Paro de emergencia
  if (e)
  {
    pantalla_emergencia();
    e = 0;
  }
  // Control motor PAP
  if (movimiento)
  {
    lectura(); // Lectura paro de emergencia
    if (pasos_ini)
    {
      pasos_ini = 0;
      pasos_cont++;
      a = 1 - a;
      digitalWrite(8, a); // Envío de pasos
      if (dir)
        angulo_aux = coef[modo_actual - 1][1] * exp(-
float(pasos_cont) / coef[modo_actual - 1][2]) + coef[modo_actual -
1][0];
      else
        angulo_aux = coef[modo_actual - 1][1] * exp(-float(2 *
pasos_lim - pasos_cont) / coef[modo_actual - 1][2]) +
coef[modo_actual - 1][0];
      // Detección de cambio en el ángulo
      angulo_v[0] = angulo_v[1];
      angulo_v[1] = round(angulo_aux);
      if (angulo_v[0] != angulo_v[1])
      {
        Serial.print(pasos_cont);
        Serial.print(" ");
      }
    }
  }
}
```

```

Serial.print(angulo_aux);
Serial.print(" ");
Serial.println(dir);
myGLCD.setColor(255, 255, 0);
myGLCD.print("Angulo", 50, 111);
// Parámetros
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(0, 255, 0);
if (angulo_v[1] > 100)
{
myGLCD.printNumI(angulo_v[1], 150, 111);
}
else if (angulo_v[1] >= 10)
{
myGLCD.print(" ", 150, 111);
myGLCD.printNumI(angulo_v[1], 158, 111);
}
else
{
myGLCD.print(" ", 150, 111);
myGLCD.printNumI(angulo_v[1], 164, 111);
} }
// Detección para el cambio de giro
if (pasos_cont == pasos_lim)
{
digitalWrite(9, 0);
dir = 0;
}
// Final del movimiento (posición inicial)
if (pasos_cont >= (2 * pasos_lim) || ((digitalRead(11) == 0)
&& (pasos_cont >= (1.8 * pasos_lim))))
{
digitalWrite(9, 1);
dir = 1;
pasos_cont = 0;
cont_rep++;
if (cont_rep == repeticiones_actual)
{
pasos_cont = 0;
movimiento = 0;
cont_rep = 0;
pantalla_espera();
delay(1000);
pantalla_inicial();
} } } }
else //Motor detenido
{
if (myTouch.dataAvailable())
{
myTouch.read();
x = myTouch.getX();
y = myTouch.getY();
deteccionBotones();
Serial.println(pantalla);
} } }
/***/ Rutinas para graficar las diferentes pantallas
void pantalla_base()
{
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Rehabilitador pasivo de rodilla", CENTER, 10);
myGLCD.setColor(100, 100, 100);
myGLCD.drawRoundRect(1, 1, 319, 239);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect(20, 30, 20 + 280, 30 + 170);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect(20, 30, 20 + 280, 30 + 170);
}
void pantalla_espera()
{
pantalla = 5;
myGLCD.setColor(0, 0, 0);
// Caja de visualización
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect(21, 31, 19 + 280, 29 + 170);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Espere por favor", CENTER, 105);
myGLCD.print("Iniciando el sistema...", CENTER, 125);
// Espera el reinicio del sistema con la lectura del final de carrera
if (modo_actual == 1)
{
if (digitalRead(11) == 1)
{
digitalWrite(9, 0);
dir = 0;
while (digitalRead(11) == 1)
{
a = 1 - a;
digitalWrite(8, a);
delay(4);
}
pasos_cont = 0;
digitalWrite(9, 1);
dir = 1;
} }
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect(21, 31, 19 + 280, 29 + 170);
// Mensaje de espera
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Sistema listo", CENTER, 115);
delay(1000);
}
void pantalla_inicial()
{
pantalla = 0;
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect(21, 31, 19 + 280, 29 + 170);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Bienvenidos", 25, 111);
// Parámetros seleccionados
myGLCD.print("Repeticiones", 150, 111);
myGLCD.setColor(255, 255, 0);
myGLCD.print("Angulo", 150, 59);
myGLCD.print("Velocidad", 150, 163);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(0, 255, 0);
if (angulo_actual < 100)
{
myGLCD.print(" ", 250, 111);
myGLCD.printNumI(angulo_actual, 258, 59);
}
else myGLCD.printNumI(angulo_actual, 250, 59);
if (repeticiones_actual < 10)
{
myGLCD.print(" ", 258, 111);
myGLCD.printNumI(repeticiones_actual, 264, 111);
}
else
myGLCD.printNumI(repeticiones_actual, 258, 111);
if (velocidad_actual < 100)
{
myGLCD.print(" ", 250, 163);
myGLCD.printNumI(velocidad_actual, 258, 163);
}
else
myGLCD.printNumI(velocidad_actual, 250, 163);
dBinicio();
}

```

```

    pantalla = 0;
}
void pantalla_informacion()
{
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
dBinformacion();
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Intervalos de funcionamiento", CENTER, 35);
// Contenido Información
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 0);
myGLCD.print("Angulo minimo:", 25, 65 );
myGLCD.print("Angulo maximo:", 25, 90 );
myGLCD.print("Repeticiones:", 25, 115);
myGLCD.print("Velocidad:" , 25, 140);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(0, 255, 0);
myGLCD.print("0" , 145, 65 );
myGLCD.print("30 - 120", 145, 90 );
myGLCD.print("1 - 30" , 145, 115);
myGLCD.print("60 - 70 - 80 - 90" , 145, 140);
/*
    myGLCD.print("70" , 145, 155);
    myGLCD.print("80" , 145, 170);
    myGLCD.print("90" , 145, 185);*/
pantalla = 4;
}
void pantalla_configuracion()
{
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Configuracion", CENTER, 35);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 0);
myGLCD.print(" Angulo" , 25, 59);
myGLCD.print("Repeticiones" , 25, 111);
myGLCD.print("Velocidad" , 25, 163);
// Parámetros
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(0, 255, 0);
if (angulo_actual < 100)
{
myGLCD.print(" ", 200, 111);
myGLCD.printNumI(angulo_actual, 208, 59);
}
else myGLCD.printNumI(angulo_actual, 200, 59);
if (repeticiones_actual < 10)
{
myGLCD.print(" ", 200, 111);
myGLCD.printNumI(repeticiones_actual, 208, 111);
}
else myGLCD.printNumI(repeticiones_actual, 200, 111);
if (velocidad_actual < 100)
{
myGLCD.print(" ", 200, 163);
myGLCD.printNumI(velocidad_actual, 208, 163);
}
else myGLCD.printNumI(velocidad_actual, 200, 163);
dBconfiguracion();
pantalla = 1;
}
void pantalla_proceso()
{
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (2, 201, 300, 237);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
pantalla = 2;
// Inicio rutina de movimiento
movimiento = 1;
pasos_lim = angulos[modo_actual - 1][angulo_actual / 5 - 6];
Timer1.initialize(velocidades[9 - velocidad_actual / 10]);
}
void pantalla_emergencia()
{
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (2, 201, 300, 237);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Paro de emergencia", CENTER, 100);
myGLCD.print("Desea continuar?", CENTER, 120);
dBemergencia();
pantalla = 3;
}
void pantalla_mod0()
{
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (2, 201, 300, 237);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 0);
myGLCD.print("Posicion", 25, 59);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(0, 255, 0);
myGLCD.print(" ", 200, 111);
myGLCD.printNumI(modo_actual, 208, 59);
dBmodo();
pantalla = 6;
}
void pantalla_espera_mod0(byte cambioP)
{
pantalla = 7;
switch (cambioP)
{
case 1:
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (2, 201, 300, 237);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
// Mensaje de espera
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Espere por favor", CENTER, 105);
myGLCD.print("Reubicando sistema...", CENTER, 125);
break;
case 2:
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (21, 31, 19 + 280, 29 + 170);
// Mensaje de espera
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Sistema listo", CENTER, 115);
delay(200);
break;
}}
/**/ Rutinas para dibujar botones en las diferentes pantallas ***/
void dBmodo()
{
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (2, 201, 100, 237);
byte d1 = 67;
byte d2 = 30;
byte d3 = 4;

```

```

byte d4 = d1 + d3;
for (byte Cx = 2; Cx < 3; Cx++)
{
    myGLCD.setColor(0, 0, 255);
    myGLCD.fillRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx + d1,
205 + d2);
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx +
d1, 205 + d2);
    myGLCD.setBackColor(0, 0, 255);
    myGLCD.setColor(255, 255, 255);
    switch (Cx)
    {
        case 2:
            myGLCD.print("Atras", 167, 213);
            break;
    } }
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (140, 43, 175, 83);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (140, 43, 175, 83);
myGLCD.print("-", 150, 59);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (240, 43, 275, 83);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (240, 43, 275, 83);
myGLCD.print("+", 250, 59);
}
void dBemergencia()
{
    myGLCD.setColor(0, 0, 0);
    myGLCD.fillRoundRect (2, 201, 100, 237);
    byte d1 = 67;
    byte d2 = 30;
    byte d3 = 4;
    byte d4 = d1 + d3;
    for (byte Cx = 2; Cx < 4; Cx++)
    {
        myGLCD.setColor(0, 0, 255);
        myGLCD.fillRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx + d1,
205 + d2);
        myGLCD.setColor(255, 255, 255);
        myGLCD.drawRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx +
d1, 205 + d2);
        myGLCD.setBackColor(0, 0, 255);
        myGLCD.setColor(255, 255, 255);
        switch (Cx)
        {
            case 2:
                myGLCD.print("Detener", 167, 213);
                break;
            case 3:
                myGLCD.print("Contin.", 241, 213);
                break;
        } }
}
void dBinicio()
{
    myGLCD.setColor(0, 0, 0);
    myGLCD.fillRoundRect (2, 201, 200, 237);
    byte d1 = 67;
    byte d2 = 30;
    byte d3 = 4;
    byte d4 = d1 + d3;
    for (byte Cx = 2; Cx < 4; Cx++)
    {
        myGLCD.setColor(0, 0, 255);
        myGLCD.fillRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx + d1,
205 + d2);
        myGLCD.setColor(255, 255, 255);

```

```

        myGLCD.drawRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx +
d1, 205 + d2);
        myGLCD.setBackColor(0, 0, 255);
        myGLCD.setColor(255, 255, 255);
        switch (Cx)
        {
            case 2:
                myGLCD.print("Config.", 167, 213);
                break;
            case 3:
                myGLCD.print("Iniciar", 241, 213);
                break;
        } }
}
void dBinformacion()
{
    myGLCD.setColor(0, 0, 0);
    myGLCD.fillRoundRect (2, 201, 100, 237);
    myGLCD.fillRoundRect (101, 201, 200, 237);
    myGLCD.fillRoundRect (233, 201, 300, 237);
    byte d1 = 67;
    byte d2 = 30;
    byte d3 = 4;
    byte d4 = d1 + d3;
    for (byte Cx = 2; Cx < 3; Cx++)
    {
        myGLCD.setColor(0, 0, 255);
        myGLCD.fillRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx + d1,
205 + d2);
        myGLCD.setColor(255, 255, 255);
        myGLCD.drawRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx +
d1, 205 + d2);
        myGLCD.setBackColor(0, 0, 255);
        myGLCD.setColor(255, 255, 255);
        switch (Cx)
        {
            case 2:
                myGLCD.print("Atras", 167, 213);
                break;
        } }
}
void dBconfiguracion()
{
    byte d1 = 67;
    byte d2 = 30;
    byte d3 = 4;
    byte d4 = d1 + d3;
    for (byte Cx = 0; Cx < 4; Cx++)
    {
        myGLCD.setColor(0, 0, 255);
        myGLCD.fillRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx + d1,
205 + d2);
        myGLCD.setColor(255, 255, 255);
        myGLCD.drawRoundRect (20 + d4 * Cx, 205, 20 + d4 * Cx +
d1, 205 + d2);
        myGLCD.setBackColor(0, 0, 255);
        myGLCD.setColor(255, 255, 255);
        switch (Cx)
        {
            case 0:
                myGLCD.print("info.", 30, 213);
                break;
            case 1:
                myGLCD.print("Posicion", 93, 213);
                break;
            case 2:
                myGLCD.print("Atras", 167, 213);
                break;
            case 3:
                myGLCD.print("Iniciar", 241, 213);
                break;
        } }
}

```



```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (140, 43, 175, 83);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (140, 43, 175, 83);
myGLCD.print("-5", 150, 59);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (120, 95, 155, 135);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (120, 95, 155, 135);
myGLCD.print("-1", 130, 111);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (160, 95, 195, 135);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (160, 95, 195, 135);
myGLCD.print("-5", 170, 111);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (140, 147, 175, 187);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (140, 147, 175, 187);
myGLCD.print("<", 150, 163);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (240, 43, 275, 83);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (240, 43, 275, 83);
myGLCD.print("+5", 250, 59);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (220, 95, 255, 135);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (220, 95, 255, 135);
myGLCD.print("+1", 230, 111);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (260, 95, 295, 135);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (260, 95, 295, 135);
myGLCD.print("+5", 270, 111);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (240, 147, 275, 187);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (240, 147, 275, 187);
myGLCD.print(">", 255, 163);
}
void deteccionBotones()
{
// Botones inferiores de control
if ((y >= 205) && (y <= 235))
{
if ((x >= 20) && (x <= 87))
{
switch (pantalla)
{
case 1:
waitButton(20, 205, 87, 235);
pantalla_informacion();
break;
}
}
if ((x >= 91) && (x <= 158))
{
switch (pantalla)
{
case 1:
waitButton(91, 205, 158, 235);
pantalla_modos();
break;
}
}
if ((x >= 162) && (x <= 229))
{
switch (pantalla)
{
case 0:
waitButton(162, 205, 229, 235);

```

```

pantalla_configuracion();
break;
case 1:
waitButton(162, 205, 229, 235);
EEPROM.write(0, angulo_actual);
EEPROM.write(1, repeticiones_actual);
EEPROM.write(2, velocidad_actual);
pantalla_inicial();
break;
case 3: // Paro de emergencia (Detener)
waitButton(162, 205, 229, 235);
pasos_ini = 0;
pasos_cont = 0;
movimiento = 0;
cont_rep = 0;
pantalla_espera();
delay(1000);
pantalla_inicial();
break;
case 4:
waitButton(162, 205, 229, 235);
pantalla_configuracion();
break;
case 6:
waitButton(162, 205, 229, 235);
if (modo_aux != modo_actual)
{
EEPROM.write(3, modo_actual);
pantalla_espera_modos(1);
switch (modo_aux)
{
case 1:
modo_aux = modo_actual;
switch (modo_actual)
{
case 2:
digitalWrite(9, 1);
dir = 1;
for (int i = 0; i < 8000; i++)
{
a = 1 - a;
digitalWrite(8, a);
}
delayMicroseconds(1333);
break;
case 3:
digitalWrite(9, 1);
dir = 1;
for (int i = 0; i < 14800; i++)
{
a = 1 - a;
digitalWrite(8, a);
delayMicroseconds(1333);
}
break;
case 2:
modo_aux = modo_actual;
switch (modo_actual)
{
case 1:
digitalWrite(9, 0);
dir = 0;
while (digitalRead(11) == 1)
{
a = 1 - a;
digitalWrite(8, a);
}
delayMicroseconds(1333);
}
}
}
}
}

```

```

        digitalWrite(9, 1);
        dir = 1;
        break;
    case 3:
        digitalWrite(9, 1);
        dir = 1;
        for (int i = 0; i < 6800; i++)
        {
            a = 1 - a;
            digitalWrite(8, a);
delayMicroseconds(1333);
        }
        digitalWrite(9, 1);
        dir = 1;
        break;
    }
    break;
    case 3:
        modo_aux = modo_actual;
        switch (modo_actual)
        {
            case 1:
                digitalWrite(9, 0);
                dir = 0;
                while (digitalRead(11) == 1)
                {
                    a = 1 - a;
                    digitalWrite(8, a);
delayMicroseconds(1333);
                }
                digitalWrite(9, 1);
                dir = 1;
                break;
            case 2:
                digitalWrite(9, 0);
                dir = 0;
                for (int i = 0; i < 6800; i++)
                {
                    a = 1 - a;
                    digitalWrite(8, a);
                    delayMicroseconds(1333);
                }
                digitalWrite(9, 1);
                dir = 1;
                break;
            }
        }
        pantalla_espera_modos(2);
    }
    pantalla_configuracion();
    break;
} }
if ((x >= 233) && (x <= 300))
{
    switch (pantalla)
    {
        case 0:
            waitButton(233, 205, 300, 235);
            pasos_cont = 0;
            pantalla_proceso();
            break;
        case 1:
            waitButton(233, 205, 300, 235);
            EEPROM.write(0, angulo_actual);
            EEPROM.write(1, repeticiones_actual);
            EEPROM.write(2, velocidad_actual);
            pasos_cont = 0;
            pantalla_proceso();
            break;
            case 3: // Paro de emergencia (Contiuar)
                waitButton(233, 205, 300, 235);
                pantalla_proceso();
                break;
            } } }
// Botones internos de configuración
if (pantalla == 6)
{
    myGLCD.setBackColor(0, 0, 0);
    if ((y >= 43) && (y <= 83))
    {
        if ((x >= 140) && (x <= 175))
        {
            waitButton(140, 43, 175, 83);
            modo_actual -= 1;
            if (modo_actual <= 1)
                modo_actual = 1;
            myGLCD.setColor(0, 255, 0);
            myGLCD.print(" ", 200, 59);
myGLCD.printNumI(modo_actual, 208, 59);
        }
        if ((x >= 240) && (x <= 275))
        {
            waitButton(240, 43, 275, 83);
            modo_actual += 1;
            if (modo_actual >= 3)
                modo_actual = 3;
            myGLCD.setColor(0, 255, 0);
            myGLCD.print(" ", 200, 59);
            myGLCD.printNumI(modo_actual, 208, 59);
        } } }
    if (pantalla == 1)
    {
        myGLCD.setBackColor(0, 0, 0);
        if ((x >= 140) && (x <= 175))
        {
            if ((y >= 43) && (y <= 83))
            {
                waitButton(140, 43, 175, 83);
                angulo_actual -= 5;
                if (angulo_actual <= 30)
                    angulo_actual = 30;
                myGLCD.setColor(0, 255, 0);
                if (angulo_actual < 100)
                {
                    myGLCD.print(" ", 200, 59);
                    yGLCD.printNumI(angulo_actual, 208, 59);
                }
            }
            else
                myGLCD.printNumI(angulo_actual, 200, 59);
        }
        if ((y >= 147) && (y <= 187))
        {
            waitButton(140, 147, 175, 187);
            velocidad_actual -= 10;
            if (velocidad_actual <= 60)
                velocidad_actual = 60;
            myGLCD.setColor(0, 255, 0);
            if (velocidad_actual < 100)
            {
                myGLCD.print(" ", 200, 163);
myGLCD.printNumI(velocidad_actual, 208, 163);
            }
            else
                myGLCD.printNumI(velocidad_actual, 200, 163);
        } }
        if ((x >= 240) && (x <= 275))
        {
            if ((y >= 43) && (y <= 83))
            {
                waitButton(240, 43, 275, 83);
                angulo_actual += 5;

```

```









    if (angulo_actual >= 120)
        angulo_actual = 120;
    myGLCD.setColor(0, 255, 0);
    if (angulo_actual < 100)
    {
        myGLCD.print(" ", 200, 59);
    }
myGLCD.printNumI(angulo_actual, 208, 59);
    }
    else
        myGLCD.printNumI(angulo_actual, 200, 59);
    }
    if ((y >= 147) && (y <= 187))
    {
        waitButton(240, 147, 275, 187);
        velocidad_actual += 10;
        if (velocidad_actual >= 90)
            velocidad_actual = 90;
        myGLCD.setColor(0, 255, 0);
        if (velocidad_actual < 100)
        {
            myGLCD.print(" ", 200, 163);
        }
myGLCD.printNumI(velocidad_actual, 208, 163);
    }
    else
        myGLCD.printNumI(velocidad_actual, 200, 163);
    }
    }
    if ((y >= 95) && (y <= 135))
    {
        if ((x >= 120) && (x <= 155))
        {
            waitButton(120, 95, 155, 135);
            repeticiones_actual--;
            if (repeticiones_actual <= 1 || repeticiones_actual > 200)
                repeticiones_actual = 1;
            myGLCD.setColor(0, 255, 0);
            if (repeticiones_actual < 10)
            {
                myGLCD.print(" ", 200, 111);
            }
myGLCD.printNumI(repeticiones_actual, 208, 111);
        }
        else
            myGLCD.printNumI(repeticiones_actual, 200,
111);
    }
    if ((x >= 160) && (x <= 195))
    {
        waitButton(160, 95, 195, 135);
        repeticiones_actual -= 5;
        if (repeticiones_actual <= 1 || repeticiones_actual > 200)
            repeticiones_actual = 1;
        myGLCD.setColor(0, 255, 0);
        if (repeticiones_actual < 10)
        {
            myGLCD.print(" ", 200, 111);
        }
myGLCD.printNumI(repeticiones_actual, 208, 111);
    }
    else
        myGLCD.printNumI(repeticiones_actual, 200,
111);
    }
    if ((x >= 220) && (x <= 255))
    {
        waitButton(220, 95, 255, 135);
        repeticiones_actual++;
        if (repeticiones_actual >= 30)
            repeticiones_actual = 30;
        myGLCD.setColor(0, 255, 0);
        if (repeticiones_actual < 10)
        {
            myGLCD.print(" ", 200, 111);
        }
myGLCD.printNumI(repeticiones_actual, 208, 111);
    }
    }
    else
        myGLCD.printNumI(repeticiones_actual, 200,
111);
    }
    if ((x >= 260) && (x <= 295))
    {
        waitButton(260, 95, 295, 135);
        repeticiones_actual += 5;
        if (repeticiones_actual >= 30)
            repeticiones_actual = 30;
        myGLCD.setColor(0, 255, 0);
        if (repeticiones_actual < 10)
        {
            myGLCD.print(" ", 200, 111);
        }
myGLCD.printNumI(repeticiones_actual, 208, 111);
    }
    else
        myGLCD.printNumI(repeticiones_actual, 200, 111);
    }
} } }
void waitButton(int x1, int y1, int x2, int y2) //Rutina de espera en
los botones
{
    myGLCD.setColor(255, 0, 0);
    myGLCD.drawRoundRect (x1, y1, x2, y2);
    while (myTouch.dataAvailable())
        myTouch.read();
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (x1, y1, x2, y2);
}
void Emergencia() //Rutina de paro de emergencia
{
    if (pantalla == 2)
        if (!e)
        {
            e = 1;
            pantalla = 3;
            movimiento = 0;
        }
}
void pasos() //Inicio de movimiento del motor
{
    if (movimiento)
    {
        pasos_ini = 1;
    }
}
void lectura(byte pin) //Lectura de pulsadores
{
    reading[pin] = digitalRead(pin_rd[pin]);
    if (reading[pin] != lastButtonState[pin])
        lastDebounceTime[pin] = millis();
    if ((millis() - lastDebounceTime[pin]) > debounceDelay) //Anti-
rebote
    {
        if (reading[pin] != buttonState[pin]) //Detección de cambio de
estado
        {
            buttonState[pin] = reading[pin];
            if (buttonState[pin] == 0)
                switch (pin)
                {
                    case 0:
                        Emergencia();
                        break;
                    case 1:
                        break;
                    case 2:
                        break;
                }
        }
    }
    lastButtonState[pin] = reading[pin];
}

```

## ANEXO 8

### Pruebas de funcionamiento con pacientes sanos

*Tabla de pruebas de funcionamiento con pacientes sanos.*

Con paciente	Sin paciente	Ángulo
		0°
		60°
		90°
		120°

## ANEXO 9

### Análisis de costos del dispositivo

*Tabla de costos de los materiales de fabricación directa*

Cantidad	Descripción	V.	V.
		Unitario	Total
1	Motor Nema 23	90	90
1	Controlador TB6560	35	35
1	Pantalla	20	20
1	Fuente de voltaje 24V / 5A	25	25
1	Microcontrolador Arduino Mega	18	18
6	Tubo acero inoxidable AISI 304 5/8" X 1.2 (metro)	2	12
6	Tubo acero inoxidable AISI 304 1/2" X 1.0 (metro)	1.5	9
2	Barra de acero de transmisión 3/4" (metro)	3.6	7.2
1	Barra de acero de transmisión 1" (metro)	4	4
2	Chumacera UCP 203	16	32
1	Platina de acero inoxidable	35	35
1	Acople Lovejoy L070	32	32
1	Insumos directos (tapizado, colchoneta)	30	30
	TOTAL		349.2

*Tabla de costos de los materiales de fabricación indirecta*

Cantidad	Descripción	V. Unitario	V. Total
1	Materiales mecánicos varios (tornillos, pasadores, etc.)	20	20
1	Materiales electrónicos varios (cables, botones, etc.)	20	20
1	Insumos varios (estampado, pintura, etc.)	30	30
1	Impresiones y empastado	40	40
3	Transporte	20	60
1	Carcasa (Impresión 3D)	30	30
	TOTAL		200

*Tabla de costos de los servicios básicos*

Cantidad	Descripción	V. Unitario	V. Total
3	Energía eléctrica (tarifa básica)	6.5	19.5
3	Agua potable (tarifa básica)	3	9
3	Teléfono celular (tarifa básica)	3	9
	TOTAL		37.5

*Tabla de costos directos*

Descripción	V. Total
Insumos directos de fabricación	349,2
Mano de obra directa	400
TOTAL	749,2

*Tabla de costos indirectos*

Descripción	V. Total
Servicios básicos	37.5
Mano de obra indirecta	100
Gastos administrativos	100
Insumos indirectos de fabricación	200
<b>TOTAL</b>	<b>437.5</b>

*Tabla de costos total*

Descripción	V. Total
Costos directos	749.2
Costos indirectos	437.5
<b>TOTAL</b>	<b>1186.7</b>

*Tabla de precio de venta*

Descripción	V. Total
Costo total	1186.7
Utilidad (15%)	178
<b>TOTAL</b>	<b>1364.7</b>