



Extrusora de filamentos poliméricos

*Este trabajo final de carrera fue presentado al
Departamento de Ingeniería Electrónica y Computación
de la Facultad de Ingeniería de la Universidad Nacional
de Mar del Plata, como requisito parcial para la
obtención del título de Ingeniero en Electrónica*

Alumno:

Leandro Cesar Cristobal Lombardi

Director:

Dr. Emiliano Penovi

Laboratorio de Instrumentación y Control - FI UNMdP

Co-Director:

Dr. Amadeo Daniel Sosa

Laboratorio de Metrología - FI UNMdP

24 de Octubre de 2019



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Dedicatoria

A mi familia, quienes siempre han estado ahí para apoyarme y me han ayudado a levantarme cada vez que tropezaba y a mis amigos, los que ayudaron a que nunca baje los brazos.

“Todos tus sueños pueden hacerse realidad si tienes el coraje de perseguirlos”.

Walt Disney.

Agradecimientos

Quiero agradecer a mi director, Dr. Ing. Emiliano Penovi, por la ayuda constante y guía a lo largo de todo el proyecto. A mi co-director, Dr. Ing. Amadeo Daniel Sosa, el cual siempre estuvo disponible para ayudarme y fue la pieza fundamental de la parte mecánica del proyecto. Al Sr. Luciano Ditonto que junto al Dr. Sosa mecanizaron las piezas del equipo desarrollado. Al Dr. Ing. Jonatan Fischer, por su gran ayuda al momento de implementar el control discreto. Al Dr. Ing. Patricio Donato, al Ing. Gustavo Uicich, al Ing. Walter Kloster y al Dr. Ing. Nicolás Wassinger, los cuales estuvieron siempre a disposición a lo largo del proyecto. Al PhD. José Carella, quien aportó su experiencia en extrusión de polímeros. Al Dr. Gustavo Abraham y al Dr. Pablo Caracciolo quienes son los destinatarios del equipo y han estado siempre atentos a las necesidades y predispuestos a brindar una colaboración para que el proyecto llegue a buen puerto. A la Subsecretaría de Gestión y Coordinación de Políticas Universitarias dependiente Ministerio de Educación y Deportes. Al Programa Universidad, Diseño y Desarrollo Productivo 2015, por la financiación del Proyecto *Extrusora para impresión 3D* que permitió disponer de fondos para la adquisición de algunos componentes de la extrusora.

A todos ustedes, mi más sincero agradecimiento.

Índice general

1. Introducción	1
1.1. Resumen del proyecto ejecutado	2
2. Análisis del proceso térmico	3
2.1. Introducción	3
2.2. Modelo del sistema de control	5
2.2.1. Filtro digital	6
2.2.2. Modelado del proceso térmico	7
2.2.3. Modelo de estados	8
2.2.4. Caracterización de la planta	10
2.2.5. Controlador del sistema	11
2.2.5.1. Controlador ON-OFF	11
2.2.5.2. Controlador ON-OFF con zona proporcional	13
2.2.5.3. Controlador PID	15
3. Implementación	20
3.1. Análisis circuital	20
3.2. Alimentación	21
3.3. Medición de temperatura	22
3.3.1. Generador de corriente constante	25
3.3.2. Compensación de la resistencia del cable (R_w)	26
3.3.3. Filtrado	27
3.3.4. Ajuste de offset	30
3.3.5. Multiplexado y adquisición	31
3.4. Señales de entrada y salida del sistema	33
3.5. Procesamiento	36
3.5.1. Comunicación	36
3.5.2. Interfaz de usuario	36
3.5.3. Máquina de estados	37
3.5.4. Modulación de potencia	43
4. Diseño mecánico	46
4.1. Introducción	46
4.2. Ajuste por interferencia	47
4.3. Transmisión de potencia	49

5. Resultados experimentales	52
5.1. Proceso de fusión del termoplástico	52
5.2. Controlador PID	53
6. Aspectos a mejorar	59
6.1. Respecto al diseño del PCB	59
6.2. Respecto a la caracterización del sistema	59
6.3. Respecto al calentamiento	59
6.4. Respecto a la adquisición	60
6.5. Respecto al motor y a la transmisión de potencia	60
6.6. Respecto al control del calibre del filamento	60
7. Conclusiones	61
A. Planos mecánicos	62
B. Planos circuitales	79
C. Código	88
C.1. Código general	88
C.2. Código librería MCP23008	100
C.3. Código interfaz (MatLAB)	104

Índice de figuras

1.1. Ensamblaje	2
2.1. Diagrama en bloques físico (conceptual)	4
2.2. Diagrama en bloques del sistema	5
2.3. Diagrama de Bode del filtro digital	6
2.4. Modelo eléctrico análogo del problema térmico	7
2.5. Respuesta al escalón medida	10
2.6. Transferencia del controlador ON-OFF	11
2.7. Ensayo de un controlador ON-OFF	12
2.8. Zoom del ripple en un controlador ON-OFF	13
2.9. Comportamiento de un controlador ON-OFF con zona proporcional a lo largo del rango de trabajo	14
2.10. Transferencia controlador PID	15
2.11. Diagrama de Bode G_{total} (asintótico)	16
2.12. Diagrama de Bode del controlador PID	17
2.13. Diagrama de Bode G_{total}	18
3.1. Diagrama en bloques general	20
3.2. Diagrama en bloques fuente de alimentación	21
3.3. Convertidor Buck 5VCC	22
3.4. Convertidor Buck 12VCC	22
3.5. Diagrama en bloques de la etapa de medición de temperatura	23
3.6. Esquema circuital instrumentación	24
3.7. Generador de corriente constante	25
3.8. Pt100 instrumentado con 3 hilos	26
3.9. Circuito compensador de R_w	27
3.10. Filtro pasa bajos	28
3.11. Diagrama de Bode del filtro pasa bajos	29
3.12. Esquema del circuito de ajuste de offset	30
3.13. Esquema de multiplexación	31
3.14. Distribución de pines NodeMCU-ESP8266	33
3.15. Información adquirida en MatLAB	37
3.16. Diagrama de la máquina de estados principal	38
3.17. Diagrama de la máquina de estados secundaria	38
3.18. Esquema circuital modulación por ciclos enteros	44
3.19. Modulación por ciclos enteros	45

4.1.	Gráfico interferencia entre camisas	49
4.2.	Curvas de velocidad-torque del motor	50
5.1.	Termograma DSC de una muestra de LDPE PG 7008	53
5.2.	Ensayo de calentamiento	54
5.3.	Zoom sobrepico de temperatura para $T_{ref} = 100^{\circ}C$	55
5.4.	Zoom sobrepico de temperatura para $T_{ref} = 125^{\circ}C$	56
5.5.	Zoom sobrepico de temperatura para $T_{ref} = 160^{\circ}C$	56
5.6.	Error de temperatura con el controlador PID implementado	57
5.7.	Valor del integrador en el controlador PID implementado	57
5.8.	Filamentos obtenidos con y sin control de estiramiento	58
A.1.	Ensamblaje con lista de componentes	62
A.2.	Vista en explosión del sistema modelado	63
A.3.	Plano del pistón	64
A.4.	Plano de la camisa interior	65
A.5.	Plano de la resistencia	66
A.6.	Plano de la base	67
A.7.	Plano de la boquilla	68
A.8.	Plano de la camisa exterior (delantera)	69
A.9.	Plano de la camisa exterior (trasera)	70
A.10.	Plano de la tapa trasera	71
A.11.	Plano de la cremallera	72
A.12.	Plano del piñón	73
A.13.	Plano de la base (estructura)	74
A.14.	Plano del motor	75
A.15.	Plano del soporte delantero (pivot)	76
A.16.	Plano del soporte trasero (cremallera)	77
A.17.	Plano del perno pivot	78
B.1.	Esquema del control por ciclos enteros	80
B.2.	Esquema del control del motor paso a paso	81
B.3.	Esquema de los convertidores DC/DC	82
B.4.	Esquemas de capacitores de filtro y referencia de tensión	83
B.5.	GPIO's (Entradas y salidas)	84
B.6.	Esquema de la instrumentación (canal I)	85
B.7.	Esquema de la instrumentación (canal II)	86
B.8.	Esquema del procesamiento	87

Índice de cuadros

2.1. Comportamiento de un controlador ON-OFF con zona proporcional a lo largo del rango de trabajo	14
3.1. Tabla de entradas/salidas NodeMCU-ESP8266	34
3.2. Tabla de entradas/salidas MCP2003	35
3.3. Tabla parámetros etapa de comunicación	36
3.4. Estados futuros posibles a partir del estado inicio	39
3.5. Estados futuros posibles a partir del estado posicionamiento	39
3.6. Estados futuros posibles a partir del estado preparacion	39
3.7. Estados futuros posibles a partir del estado calentamiento	40
3.8. Estados futuros posibles a partir del estado extrusión	41
3.9. Estados futuros posibles de la maquina de estados secundaria	42
3.10. Estados futuros posibles a partir del estado extrusión	42
3.11. Estados futuros posibles a partir del estado Fin	43
4.1. Propiedades de los materiales	46
4.2. Características eléctricas del motor 86HS45	50
4.3. Características mecánicas del motor 86HS45	51

Capítulo 1

Introducción

En la actualidad, la tecnología de impresión 3D ha evolucionado exponencialmente como método para la fabricación de diversos tipos de piezas, tanto prototipos y piezas simples, como productos finales altamente técnicos tales como componentes biomédicos personalizados, piezas para la industria automotriz y aeronáutica, entre otros. La impresión 3D, también conocida como manufactura por adición, consiste en la fabricación de objetos físicos a partir de la deposición por capas de un filamento fundido de PLA, ABS u otro termoplástico, siguiendo un modelo digital previamente generado en un ordenador.

Estos filamentos son producidos a través de un proceso de extrusión. Una extrusora es una máquina que permite la fabricación de filamentos al forzar el paso de un polímero termoplástico y/o material compuesto por una boquilla a temperatura elevada. Un termoplástico es un material que a cierta temperatura, se vuelve deformable o flexible, derritiéndose al calentarse y endureciéndose cuando se enfría lo suficiente. Los termopásticos se diferencian de los termoestables o termorrígidos (*baquelita, goma vulcanizada*, por ejemplo) en que estos últimos no funden al elevarlos a altas temperaturas, sino que se degradan, siendo imposible volver a moldearlos.

Los mecanismos más comunes de extrusión comprenden los basados en el uso de un tornillo sinfín, y los basados en el uso de un pistón. En el caso de la extrusión a tornillo, el mismo se encuentra en el interior de una camisa calefaccionada por zonas, produciendo el gradiente necesario para la correcta fundición del polímero. Luego, la rotación del tornillo permite comprimir el polímero fundido, permitiendo lograr la presión necesaria para que este pase a través de la boquilla. Este proceso es continuo y permite ir agregando material por medio de una tolva, lo cual es ideal cuando se requiere producir grandes cantidades de filamento. Por otro lado, las extrusoras a pistón, están constituidas por una camisa calefaccionada por la cual se desplaza un émbolo, el cual generará la presión necesaria para extruir el material por la boquilla cuando este alcance la temperatura de plastificación. Este mecanismo resulta ser el más adecuado cuando la cantidad de material a extruir es reducida, condición que se da en producciones de polímeros a baja escala desarrollados en laboratorio.

Este proyecto aborda la construcción de un prototipo de extrusora a pistón, por requerimiento de la División Polímeros Biomédicos de INTEMA (Intituto de Investigaciones en Ciencia y Tecnología de Materiales (UNMdP-CONICET)). Dicha máquina permitirá la producción de filamentos a partir de polímeros para uso médico, sintetizados en dicho laboratorio.

1.1. Resumen del proyecto ejecutado

El objetivo del proyecto consistió en el desarrollo de un prototipo funcional de extrusora a pistón para producir un filamento continuo y de sección constante para impresión 3D a partir de un polímero termoplástico incorporado en pellets o en polvo.

La componente mecánica del prototipo contó con una camisa de acero calefaccionada a resistencias, en el interior de la cual se dispuso un pistón cuyo desplazamiento se operó mediante sistema de piñón y cremallera, impulsado por un motor paso a paso.

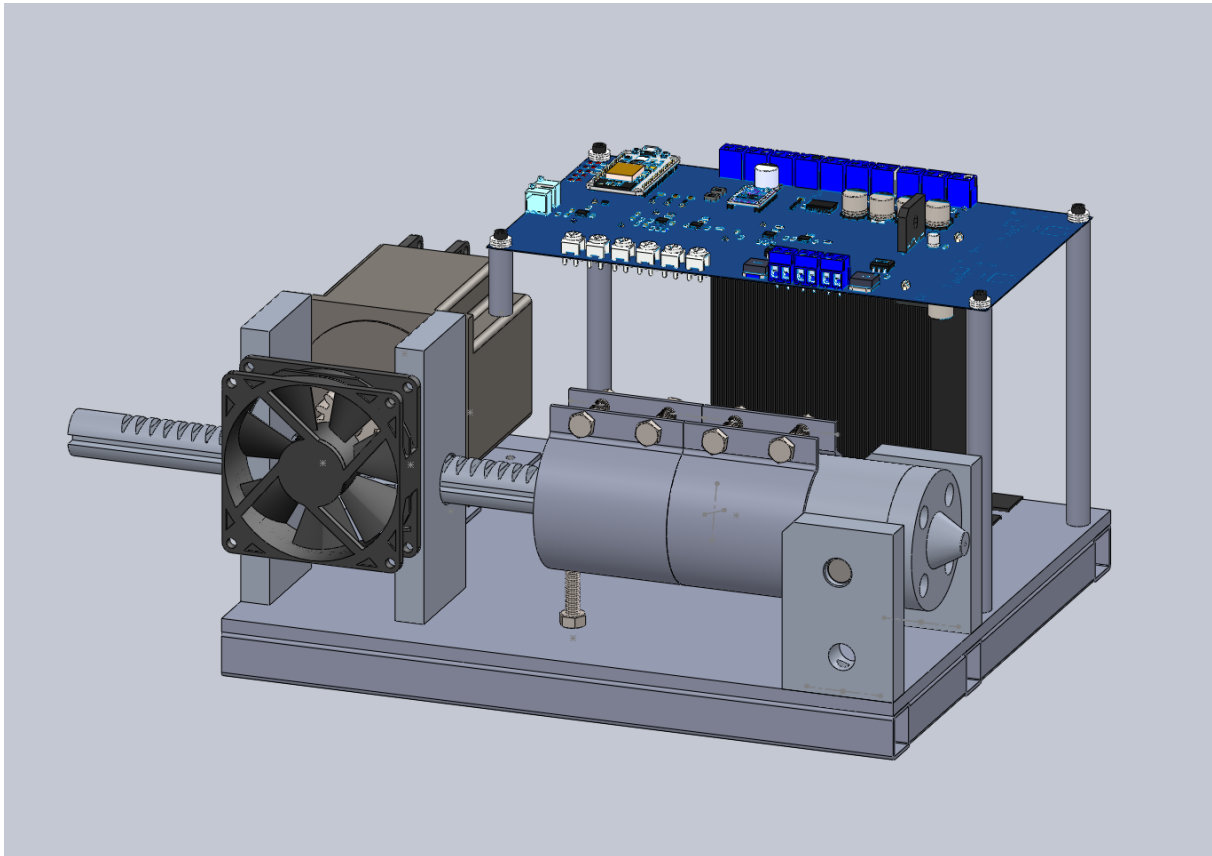


Figura 1.1: Ensamblaje

El subsistema electrónico construido se orientó al control del movimiento del pistón, la regulación de la temperatura y la interface usuario.

El sistema de control que permite la ejecución de perfiles de temperatura programables por un usuario se diseñó empleando un controlador PID y un sensor tipo Pt100 para garantizar la precisión requerida. El firmware de interface, de control de temperatura y de movimiento del pistón se implantó en una placa de desarrollo de bajo costo basada en ESP8266 de Expressif.

El diseño del conjunto mecánico se realizó en entorno CAD y las piezas componentes resultantes se fabricaron en su totalidad en el *taller de mecánica de la Facultad de Ingeniería de la UNMdP*. Por último, se ensayó el funcionamiento del equipo, se ajustaron los parámetros necesarios y se propusieron posibles mejoras para una potencial iteración del presente proyecto en otro a futuro.

Capítulo 2

Análisis del proceso térmico

2.1. Introducción

En este capítulo, se presenta el modelo del sistema de control con sus bloques fundamentales, junto a un detalle de cada bloque. Aquí se incluye diferentes tipos de controladores ensayados y el modelado de la planta. En la figura 2.1 puede observarse un esquema conceptual de la extrusora. El principio de funcionamiento es el de hacer presión sobre un polímero termoplástico calentado a una determinada temperatura. El material se encuentra dentro de una camisa metálica, rodeada por resistencias calefactoras. En uno de los extremos de la camisa, se encuentra la boquilla de extrusión la cual tiene un orificio de salida de $\phi = 1,75$ mm. La presión es ejercida por un motor, el cual por medio de un sistema de piñón y cremallera, desplaza el pistón hacia la boquilla. El cooler se encarga de refrigerar el motor. La temperatura se mide por medio de un sensor colocado en el pistón y el sistema de control se encarga de comandar la lógica para el correcto funcionamiento del equipo.

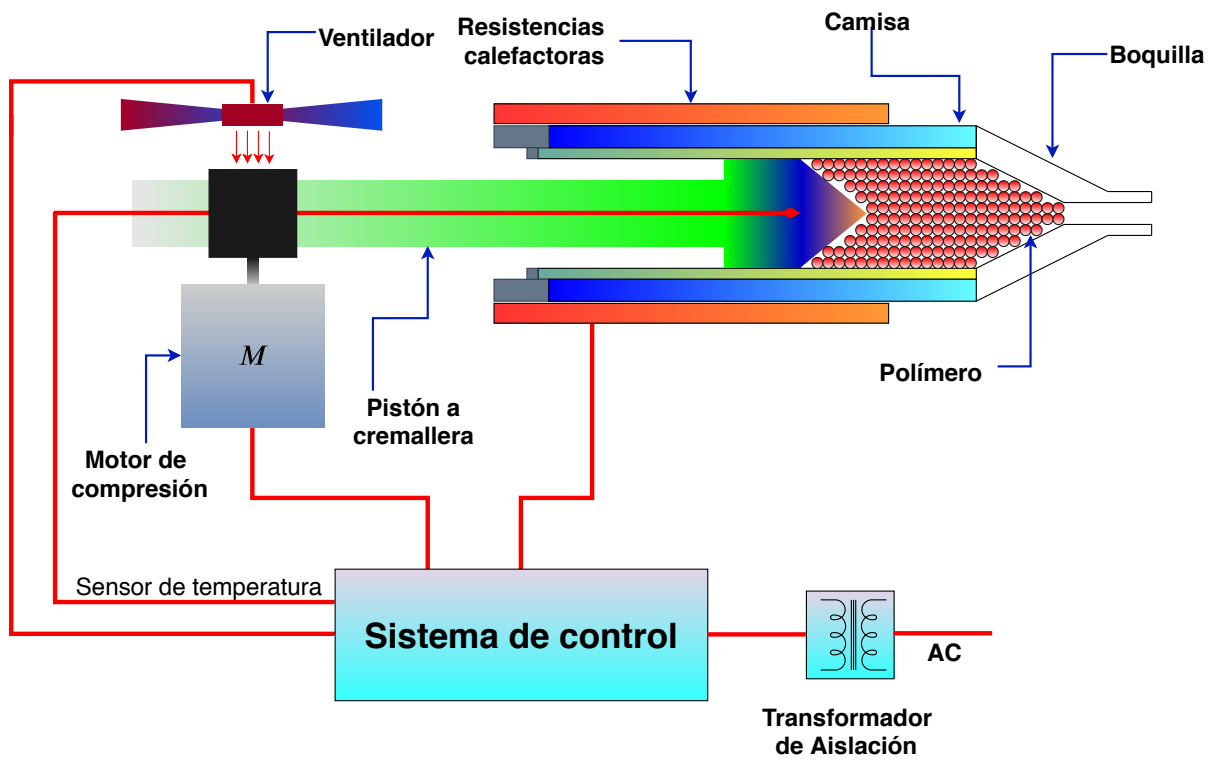


Figura 2.1: Diagrama en bloques físico (conceptual)

2.2. Modelo del sistema de control

El sistema térmico a controlar puede describirse con el diagrama en bloques de la figura 2.2

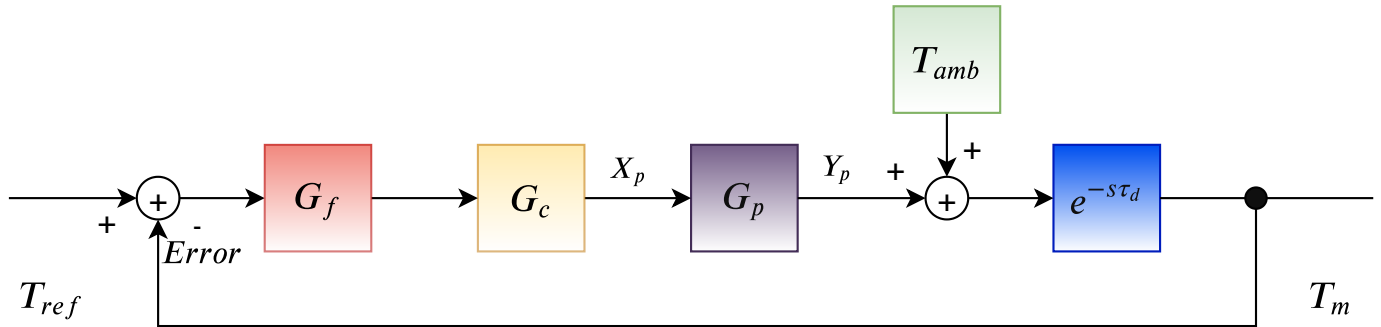


Figura 2.2: Diagrama en bloques del sistema

En donde:

- G_f es la transferencia del filtro de media móvil digital.
- G_c es la función transferencia del controlador.
- T_{amb} es la temperatura ambiente.
- G_p es la función transferencia de la planta, siendo X_p e Y_p las señales de entrada y de salida respectivamente.
- τ_d es el tiempo de retardo del sistema. Se observó que los retardos de calentamiento y de enfriamiento eran iguales y su valor es de 75 segundos.
- T_{ref} es la temperatura de referencia. Considerando lo explicado en la sección 5.1, fue necesario generar una referencia de temperatura escalonada haciendo que estos valores coincidan con aquellos en donde el material poseía un pico endotérmico. Esto permitió que el material cambie de estado de manera gradual.
- $Error$ es la diferencia entre la referencia de temperatura, T_{ref} y la temperatura medida, T_m .

2.2.1. Filtro digital

Se utilizó un filtro digital de media móvil para atenuar el ruido que no pudo ser adecuadamente limitado por el filtro pasabajos de entrada. En la figura 2.3 se muestra un diagrama de Bode de amplitud y fase del filtro. Dicho filtro se diseñó para que tuviera una respuesta plana en magnitud y que no aportara fase, en la zona de trabajo del sistema. Debido a esto es que no es considerado en los cálculos posteriores.

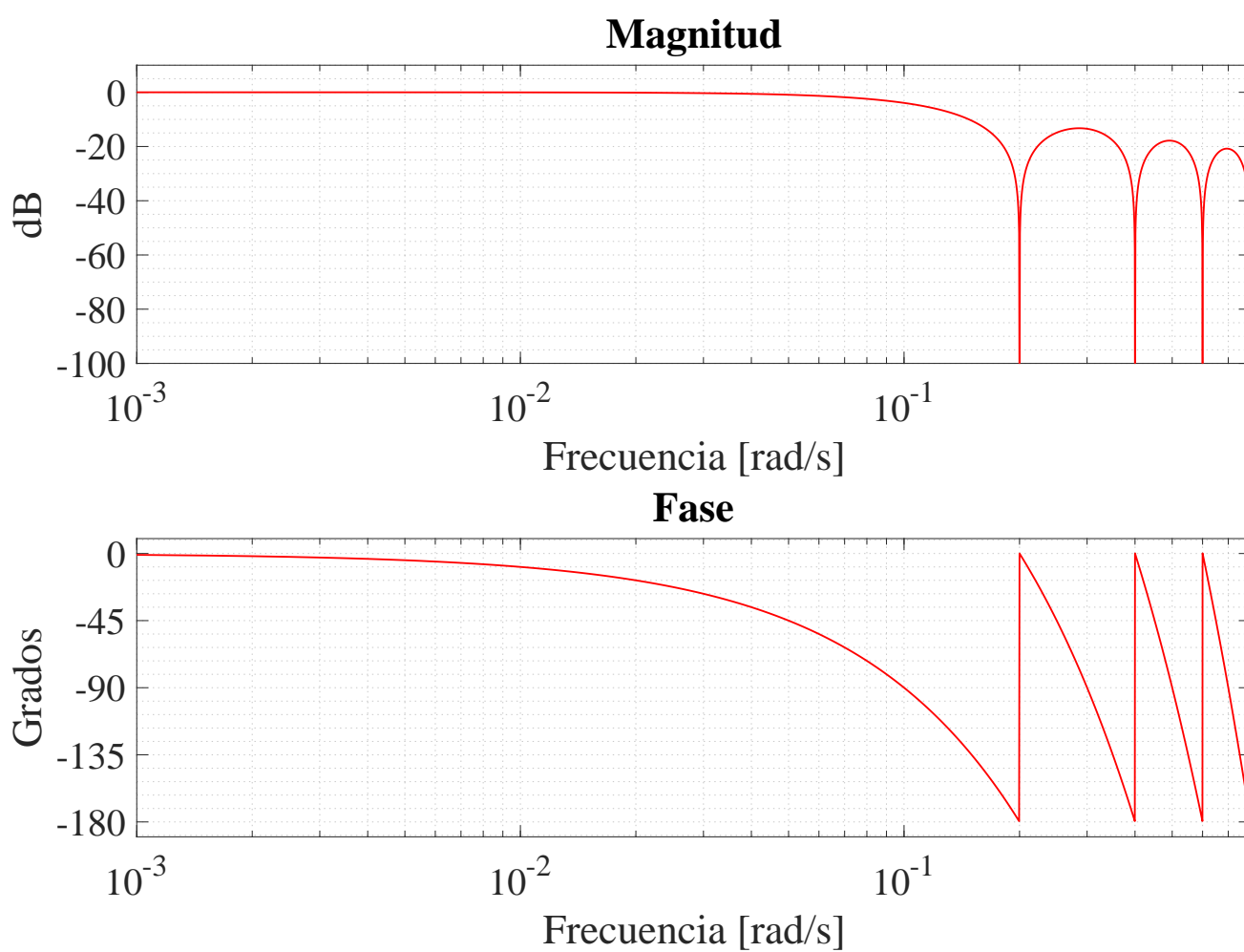


Figura 2.3: Diagrama de Bode del filtro digital

2.2.2. Modelado del proceso térmico

En la figura 2.4 se puede observar el modelo análogo eléctrico equivalente del proceso térmico a controlar.

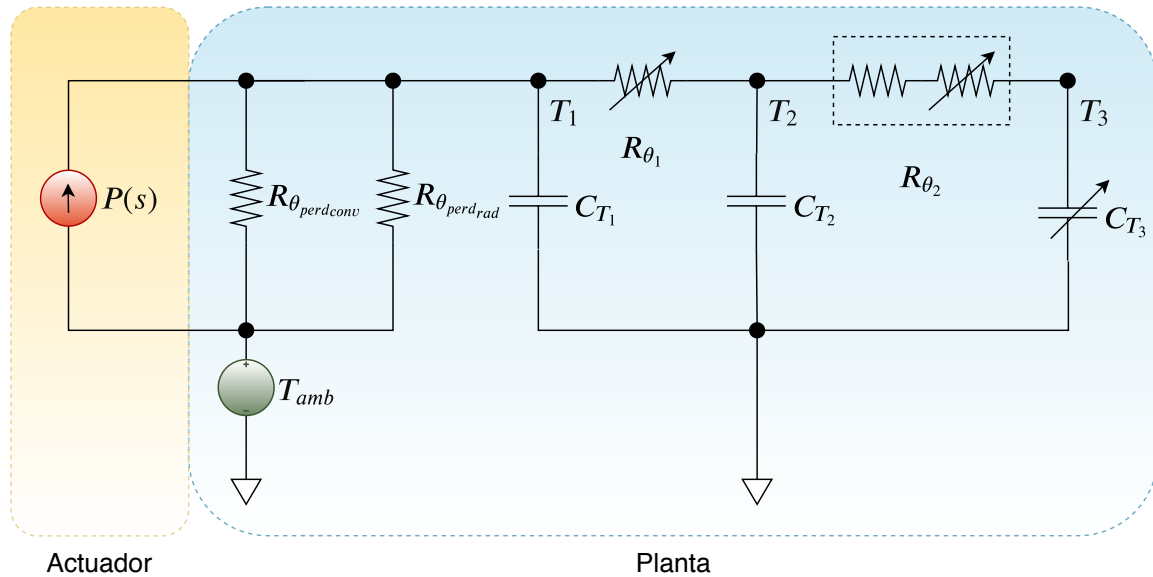


Figura 2.4: Modelo eléctrico análogo del problema térmico

En donde:

- $P(s)$ Es la potencia aportada por las resistencias calefactoras.
- $R_{\theta_{perdconv}}$ Es la resistencia de pérdidas debida a la convección.
- $R_{\theta_{perdrad}}$ Es la resistencia de pérdidas asociada a la radiación.
- T_1 Es la temperatura externa de la camisa de aluminio.
- C_{T_1} Es la capacidad calorífica de la masa de aluminio (camisa).
- T_2 Es la temperatura de la camisa de acero inoxidable.
- C_{T_2} Es la capacidad calorífica de la masa acero inoxidable (camisa).
- R_{θ_1} Es la resistencia térmica entre la camisa de aluminio y la camisa de acero inoxidable. Este valor puede considerarse como una resistencia variable con la temperatura debida al huelgo de $30\mu m$ existente entre la camisa de acero y la de aluminio a $250^{\circ}C$ causado por una limitación en el proceso de ensamblaje.
- R_{θ_2} puede considerarse compuesto por una resistencia térmica fija (entre la camisa de acero inoxidable y el pistón) y una variable entre la camisa de acero inoxidable y el polímero. Este valor puede considerarse variable con la temperatura debido

a que inicialmente, cuando el polímero se encuentra pelletizado, existen pequeñas superficies de contacto a través de las cuales el calor es transmitido y a medida que la temperatura aumenta los espacios vacíos van completándose con material formando una única superficie de contacto de gran tamaño. Dado esto, puede considerarse que R_{θ_2} es pequeño cuando el polímero se encuentra fundido, por lo que $T_3 \approx T_2$

- C_{T_3} Es la capacidad calorífica de la masa de polímero a extruir. Teniendo en cuenta lo anteriormente explicado y considerando la entalpía absorbida durante el cambio de fase a temperatura constante, puede representarse a nivel eléctrico como un condensador variable no lineal.

En el modelo de la Fig. 2.4 T_3 representa la temperatura medida en el proceso. A efectos de no incrementar innecesariamente la complejidad del modelo y debido a que la constante de tiempo asociada es despreciable respecto del resto de los componentes, la dinámica del sensor fue ignorada.

2.2.3. Modelo de estados

En la ecuación 2.1 se observa el modelo de estados que describe a la planta mostrada en la figura 2.4. Es importante aclarar que este modelo es una representación linealizada de un modelo no lineal.

$$\begin{cases} \dot{T}(s) = AT(s) + BU(s) \\ Y(s) = CT(s) + DU(s) \end{cases} \quad (2.1)$$

La ecuación 2.2 representa a la matriz de estados:

$$A = \begin{bmatrix} \frac{1}{\left(\frac{R_{perdeq}R_{\theta_1}}{R_{perdeq}+R_{\theta_1}}\right)\frac{1}{C_{T_1}}} & \frac{1}{R_{\theta_1}C_{T_1}} \\ -\left(\frac{1}{R_{\theta_1}C_{T_2}}\right) & -\left(\frac{1}{R_{\theta_1}C_{T_2}}\right) \end{bmatrix} \quad (2.2)$$

En donde:

$$R_{perdeq} = \frac{R_{\theta_{perdconv}} R_{\theta_{perdrad}}}{R_{\theta_{perdconv}} + R_{\theta_{perdrad}}} \quad (2.3)$$

La ecuación 2.4 representa a la matriz de entrada.

$$B = \begin{bmatrix} \frac{1}{C_{T_1}} \\ 0 \end{bmatrix} \quad (2.4)$$

La ecuación 2.5 representa a la matriz de salida:

$$C = [0 \ 1] \quad (2.5)$$

La ecuación 2.6 representa a la matriz de transmisión directa.

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.6)$$

La ecuación 2.7 representa al vector de entradas (también llamado vector de control):

$$U(s) = [P(s) \ 0] \quad (2.7)$$

Por último se define a 2.8 como el vector de estados:

$$T(s) = \begin{bmatrix} T_1(s) \\ T_2(s) \end{bmatrix} \quad (2.8)$$

Operando a partir de la ecuación 2.1, se obtiene la función transferencia de la ecuación 2.9

$$G_p = \frac{Y(s)}{U(s)} = (C(sI - A)^{-1}B + D) \quad (2.9)$$

$$\frac{\left(\frac{1}{C_{T_1} C_{T_2} R_{\theta_1}} \right)}{s^2 + \left[\frac{1}{R_{\theta_1} C_{T_2}} + \frac{1}{\left(\frac{R_{\theta_1} R_{perdeq}}{R_{\theta_1} + R_{perdeq}} \right) C_{T_1}} \right] s + \left(\frac{1}{C_{T_1} C_{T_2} R_{\theta_1} R_{perdeq}} \right)} \quad (2.10)$$

En la ecuación 2.10 se puede observar que la dinámica del sistema depende de 2 polos, uno relacionado con las resistencias de pérdidas y la resistencia térmica entre el acero y aluminio y el otro solamente con la resistencia térmica entre el acero y aluminio. Debido a que los materiales son buenos conductores y la aislación térmica del equipo tiene como objetivo minimizar las pérdidas, se puede anticipar que el valor de las resistencias térmicas será mucho menor que el de las resistencias de pérdidas. Al mismo tiempo, C_{T_1} es mucho mas pequeño que C_{T_2} dado que el aluminio es mejor conductor que el acero inoxidable. Esto determinó que el polo dominante será debido a las resistencias térmicas del material (aluminio y acero).

En la ecuación 2.11 se presenta una función transferencia de 2º orden. Las expresiones 2.10 y 2.11 permiten comprender la relación entre los elementos físicos y parámetros de la función.

$$G_p(S) = \frac{R_{\theta}}{[s^2 + (\omega_{p_1} + \omega_{p_2}) s + \omega_{p_1} \omega_{p_2}]} \quad (2.11)$$

2.2.4. Caracterización de la planta

A efectos de caracterizar la dinámica del sistema térmico y para poder determinar los parámetros asociados del modelo, se realizó un ensayo de respuesta temporal.

En el ensayo representado en la figura 2.5, se sometió a la planta a un escalón de potencia de 150 W midiéndose la temperatura en el extremo del extrusor con un sensor Pt100 instrumentado apropiadamente y adquirido con un ADC de 10 bits. Una vez alcanzado el régimen permanente, la acción de control fue revertida a 0W lo que permitió evaluar el comportamiento durante el enfriamiento.

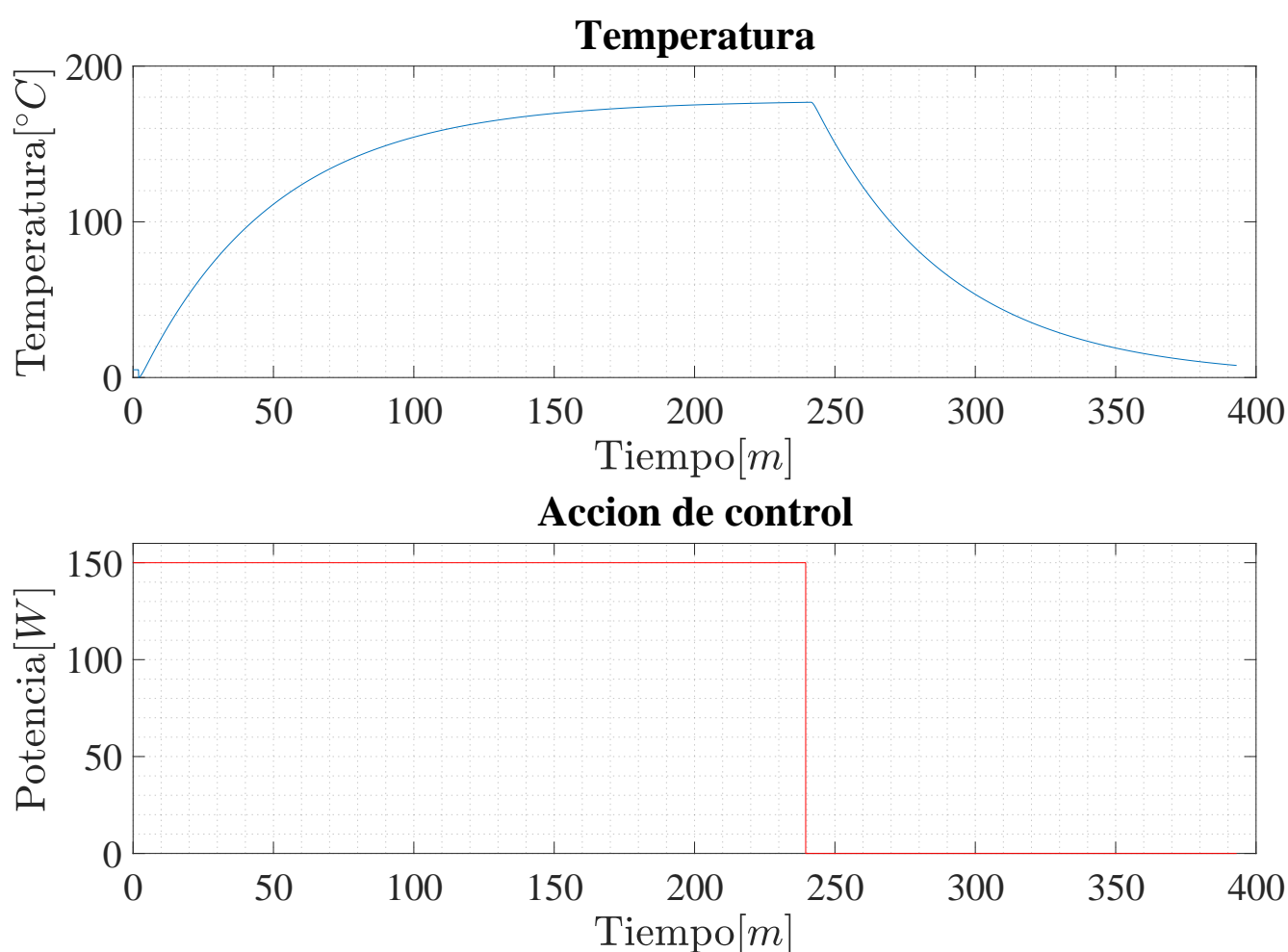


Figura 2.5: Respuesta al escalón medida

Notándose del ensayo que las constantes y los retardos de calentamiento y enfriamiento resultaron correspondientemente iguales entre si.

Definiendo:

$$G_p(S) = \frac{Y_p(S)}{X_p(S)} \quad (2.12)$$

En donde $Y_p(S)$ es la salida de la planta, $X_p(S)$ es la entrada y $G_p(S)$ es la función de transferencia. Aproximando la función de transferencia a la de un sistema de segundo orden con retardo, se obtuvo:

$$G_p(S) = \frac{1,18 \frac{^{\circ}\text{C}}{\text{W}}}{\left(\frac{s}{3,4540e-04 \frac{\text{rad}}{\text{s}}} + 1\right) \left(\frac{s}{0,0224 \frac{\text{rad}}{\text{s}}} + 1\right)} \quad (2.13)$$

2.2.5. Controlador del sistema

Para el controlador del sistema se proponen las siguientes prestaciones:

- Error nulo al escalón
- Sobrepico máximo del 1%
- Máximo ancho de banda

2.2.5.1. Controlador ON-OFF

El controlador ON-OFF es el control alineal más simple. Se basa en aplicar toda la potencia disponible en caso que el error sea positivo y no aplicar potencia si el error es menor o igual a cero, tal como puede representarse en la figura 2.6

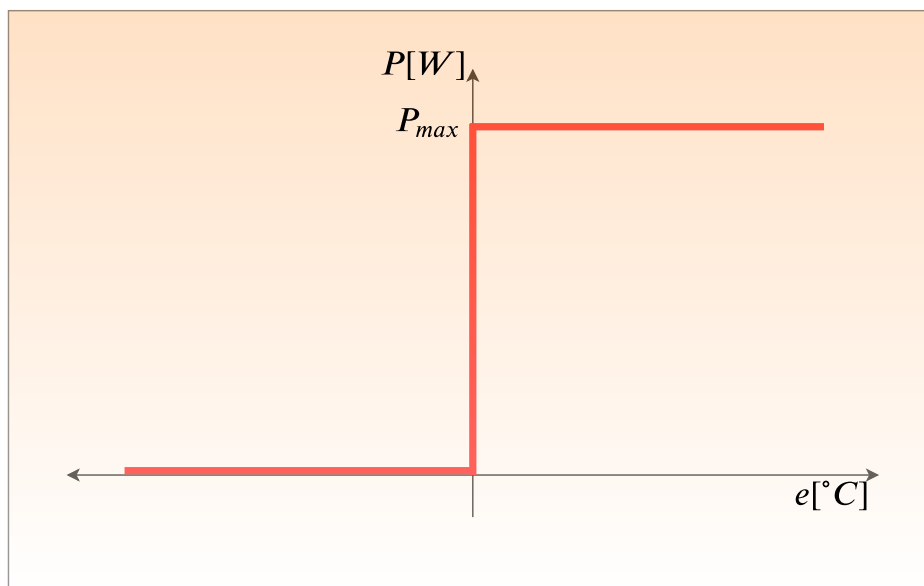


Figura 2.6: Transferencia del controlador ON-OFF

El inconveniente de este tipo de controles aplicados a sistemas con un elevado retardo de medición, radica en que la acción de control es tomada en función al valor de error correspondiente a t_d segundos antes. En consecuencia la variable real del sistema presenta un ripple, el cual es función de la dinámica del sistema y del propio valor de retardo de medición.

Implementar un control de este tipo, significa mejorar la dinámica del proceso, pero sin posibilidades de seguir de manera precisa una referencia de calentamiento debido a las limitaciones inherentes de este tipo de control alineal.

En la figura 2.7, se puede observar el resultado de utilizar un control ON-OFF sobre la planta como también la acción de control. Resulta importante destacar que además del valor del ripple, este no es simétrico respecto a la temperatura de referencia. La razón de esto es el error de calibración que presentan este tipo de controles.

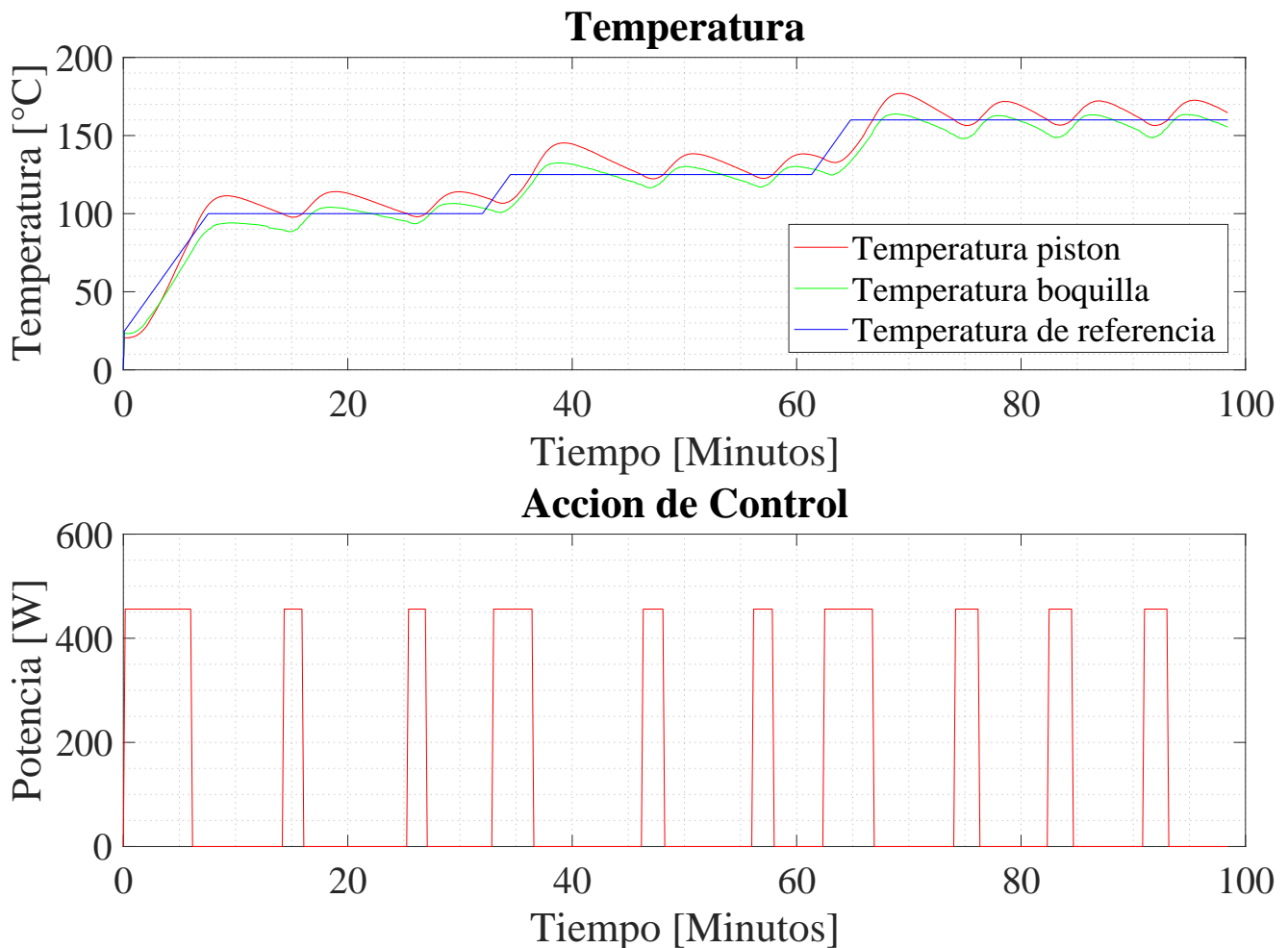


Figura 2.7: Ensayo de un controlador ON-OFF

Observando la forma del ripple de la figura 2.8, se puede hallar el período del ripple

es de 8,42 minutos y su amplitud pico a pico $15,2^{\circ}\text{C}$.

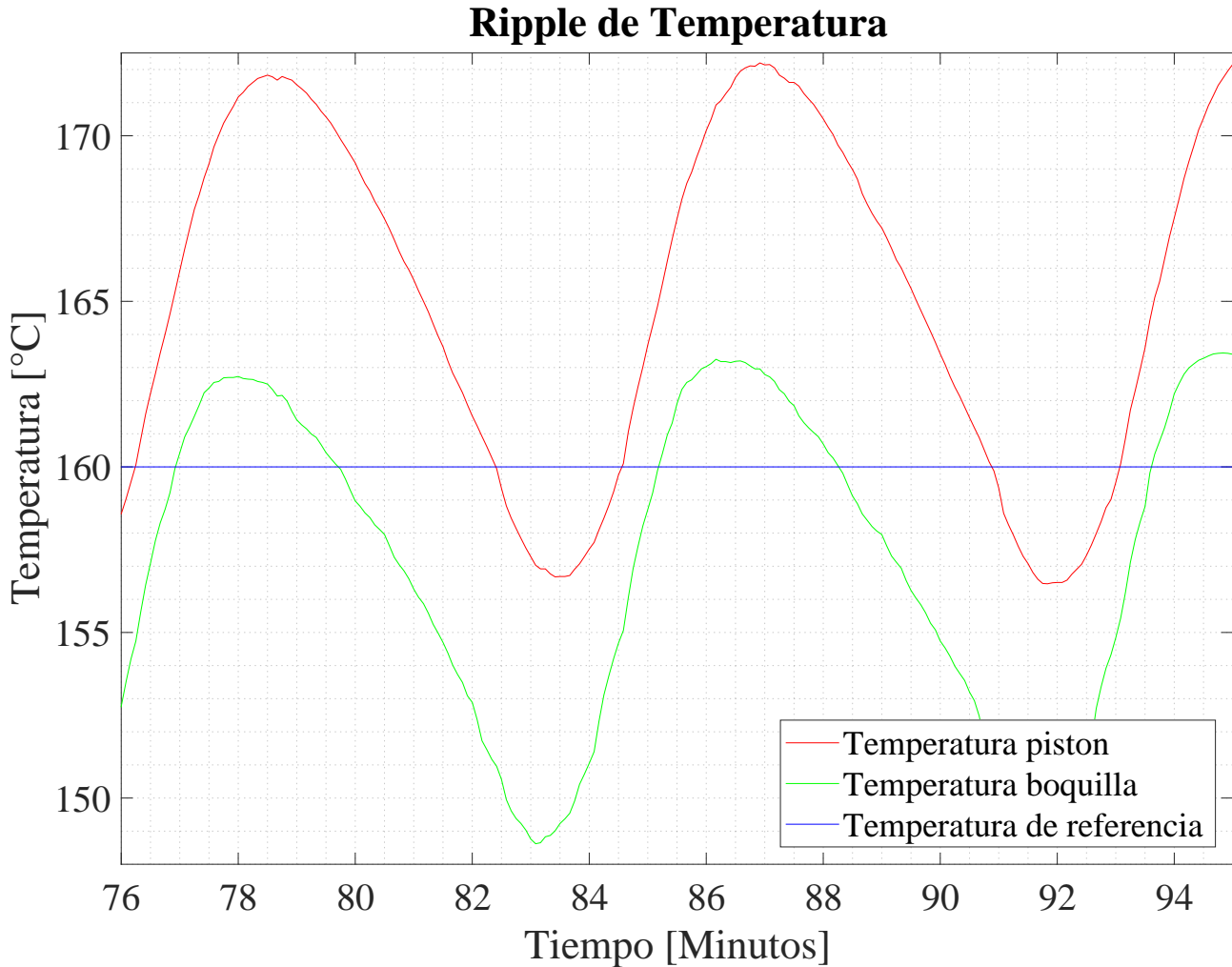


Figura 2.8: Zoom del ripple de las temperaturas de referencia, del pistón y de la boquilla en un controlador ON-OFF

2.2.5.2. Controlador ON-OFF con zona proporcional

Habiendo diseñado el control ON-OFF, surgió la idea de implementar una zona proporcional al mismo. Se pudo establecer de manera teórica que para evitar que una vez dentro de la zona lineal el control salga fuera de la misma debido al ripple, el ancho de la zona proporcional debía ser $2\Delta T_{Hpp}$, es decir, el doble del valor pico a pico del ripple.

$$T_{ref} = \begin{cases} T_{ref1} = 100^{\circ}\text{C} \\ T_{ref2} = 125^{\circ}\text{C} \\ T_{ref3} = 160^{\circ}\text{C} \end{cases} \quad (2.14)$$

Luego de realizado el ensayo con las referencias de temperatura mostradas en la ecuación 2.14, se pudo ver que implementar este tipo de control no solo acarrea el inconveniente de ser incapaz de seguir una referencia, tal como se presentó como corolario de la sección 2.2.5.1, sino que también, el sistema solo estaría funcionando en la zona alineal (*dinámica más rápida*), en el rango en donde la diferencia de temperatura entre el valor de referencia y el medido sea mayor a $2\Delta T_{Hpp}$ ($30^\circ C$). Partiendo de una temperatura ambiente promedio de $20^\circ C$, el sistema operará del siguiente modo:

Expresión	Temperatura	Zona
$T_{amb} < T \leq (T_{ref_1} - 2\Delta T_{Hpp})$	$20^\circ C < T \leq 69,6^\circ C$	no lineal
$(T_{ref_1} - 2\Delta T_{Hpp}) < T \leq T_{ref_1}$	$69,6^\circ C < T \leq 100^\circ C$	lineal
$T_{ref_1} < T \leq T_{ref_2}$	$100^\circ C < T \leq 125^\circ C$	lineal
$T_{ref_2} < T \leq (T_{ref_3} - 2\Delta T_{Hpp})$	$125^\circ C < T \leq 129,6^\circ C$	no lineal
$(T_{ref_3} - 2\Delta T_{Hpp}) < T \leq T_{ref_3}$	$129,6^\circ C < T \leq 160^\circ C$	lineal

Cuadro 2.1: Comportamiento de un controlador ON-OFF con zona proporcional a lo largo del rango de trabajo

En la figura 2.9, puede apreciarse de manera grafica lo expresado en la tabla 2.1

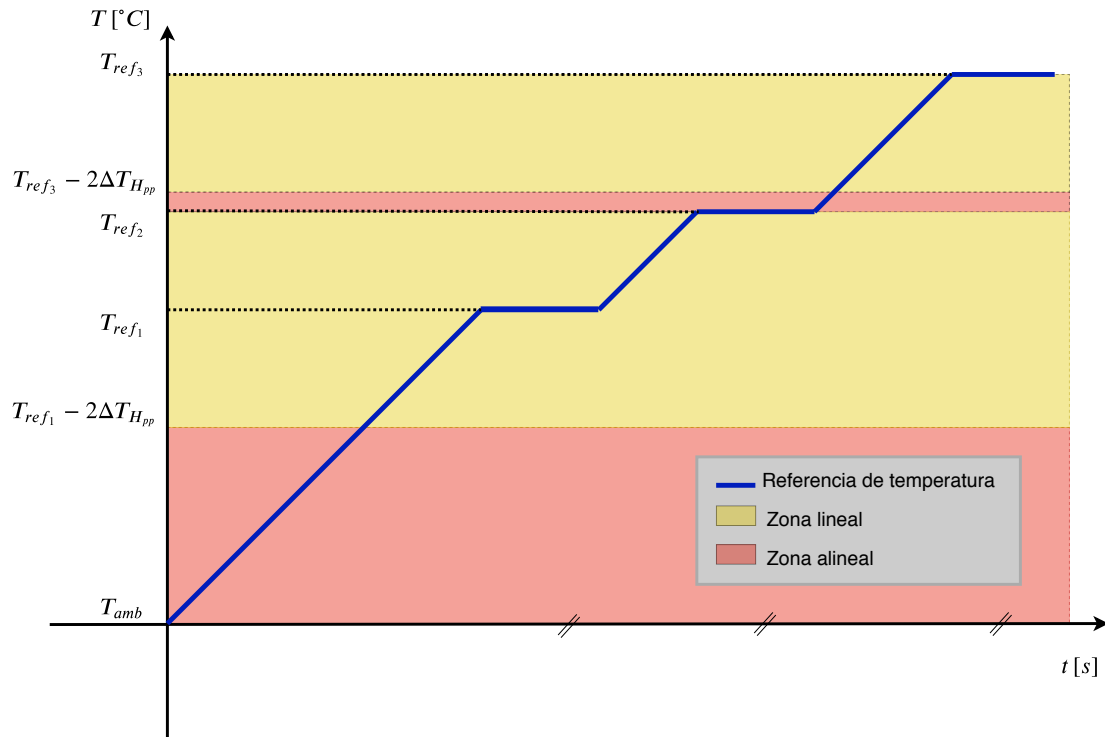


Figura 2.9: Comportamiento de un controlador ON-OFF con zona proporcional a lo largo del rango de trabajo

Debido a esto, este controlador no fue implementado.

2.2.5.3. Controlador PID

Finalmente se propuso el uso de un controlador lineal, sin zona ON-OFF del tipo proporcional - integrador - derivativo (PID).

La función transferencia de este tipo de controladores se muestra en la ecuación 2.15:

$$G_c(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{\frac{s}{\omega_p} + 1} \quad (2.15)$$

Las constantes k_p , k_i y k_d determinaran las posiciones de las singularidades de la función transferencia.

En la figura 2.10 puede observarse una representación asintótica de la transferencia del controlador.

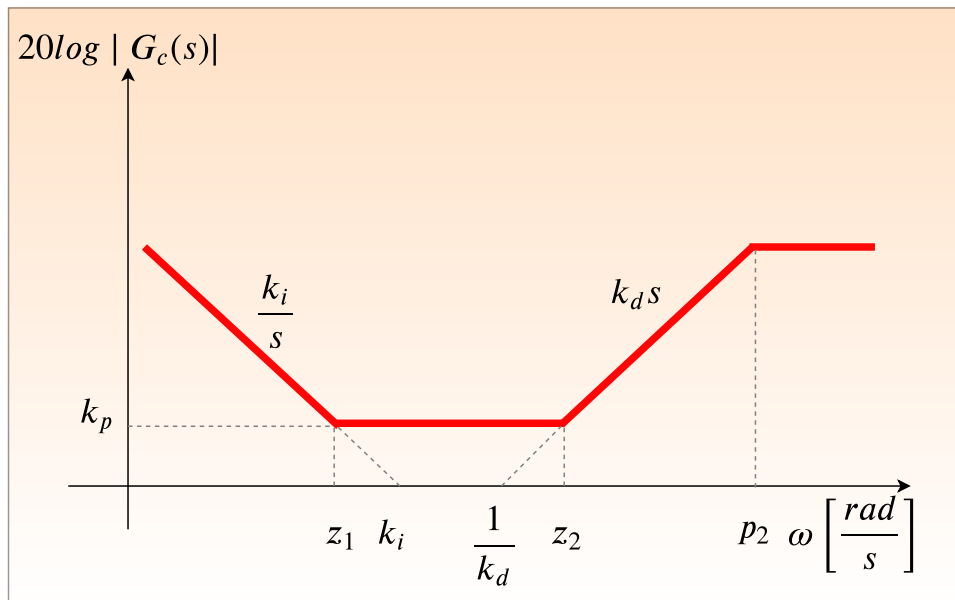


Figura 2.10: Transferencia controlador PID

Para obtener la función transferencia del controlador se utilizó la técnica de *loopshaping*. Esta se basa en diseñar una función transferencia gráficamente a partir del diagrama de Bode basándose en los requerimientos del sistema. Como criterio de diseño, se puede considerar que si se tiene un control tipo PID, este poseerá dos ceros y dos polos en donde los ceros del controlador cancelaran a los polos de la planta. En cuanto a los polos del controlador, uno se encuentra en el origen (lo que determina una característica de sistema tipo I) y el segundo polo se agrega para que el control sea físicamente realizable.

El procedimiento realizado fue hallar la transferencia de lazo, considerando un controlador de ganancia unitaria. Luego se halló la ganancia de la planta a la frecuencia en donde el margen de fase del sistema era superior a 60° , de forma tal de no tener sobrepico en la respuesta a lazo cerrado del sistema y se ajustó la ganancia del controlador. Los valores obtenidos determinaron un controlador tal que la acción de control no saturaba.

En caso de haberlo hecho, debería haberse implementado alguna técnica para evitar que el término integral diverja (antiwindup). Los valores hallados fueron:

$$\begin{cases} k_p = 4,6894 \\ k_i = 0,0032 \\ k_d = 271,1011 \end{cases} \quad (2.16)$$

En la figura 2.11 puede observarse el diagrama de Bode asintótico del control y en la imagen 2.12 puede observarse el diagrama de Bode detallado.

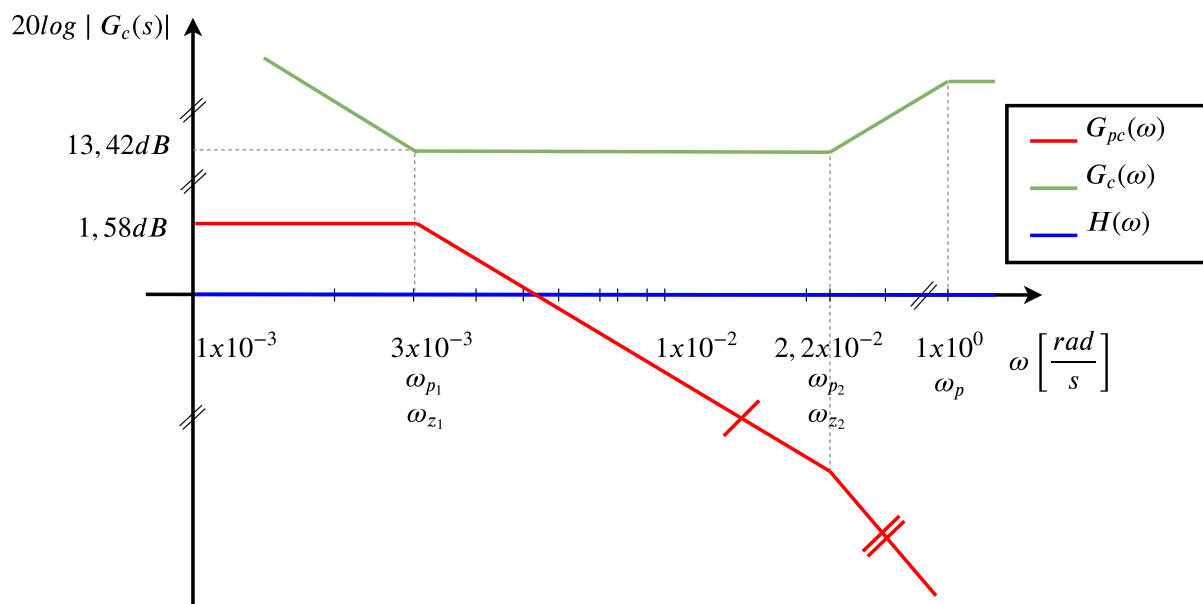


Figura 2.11: Diagrama de Bode G_{total} (asintótico)

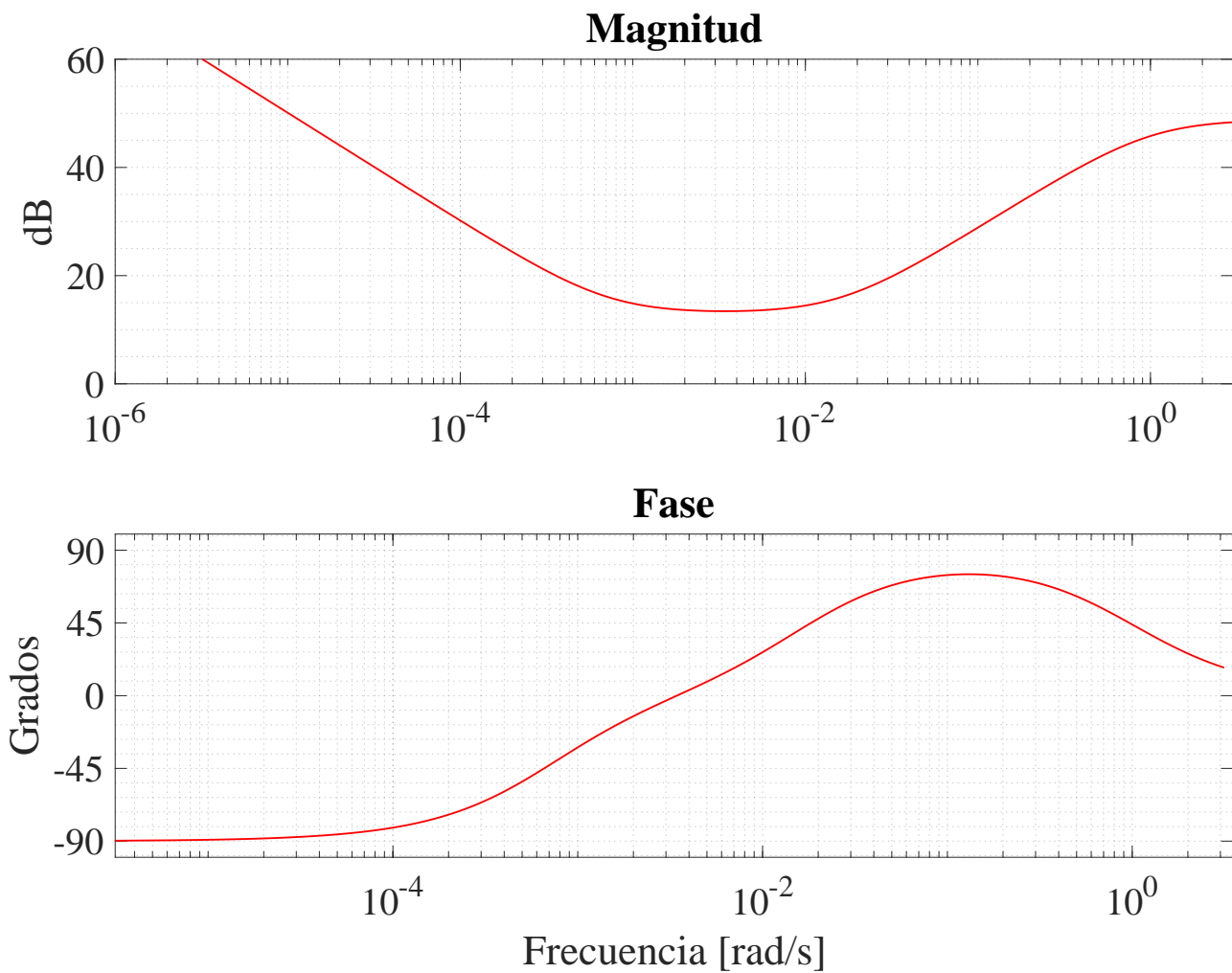


Figura 2.12: Diagrama de Bode del controlador PID

En la figura 2.13 podremos observar el diagrama de bode de $G_{total} = G_c(s)G_p(s)H(s)e^{s\tau_d}$ en donde se verifica que el $M_\Phi = 69^\circ \big|_{\omega_{0dB}=1,9x10^{-3} \frac{rad}{s}}$.

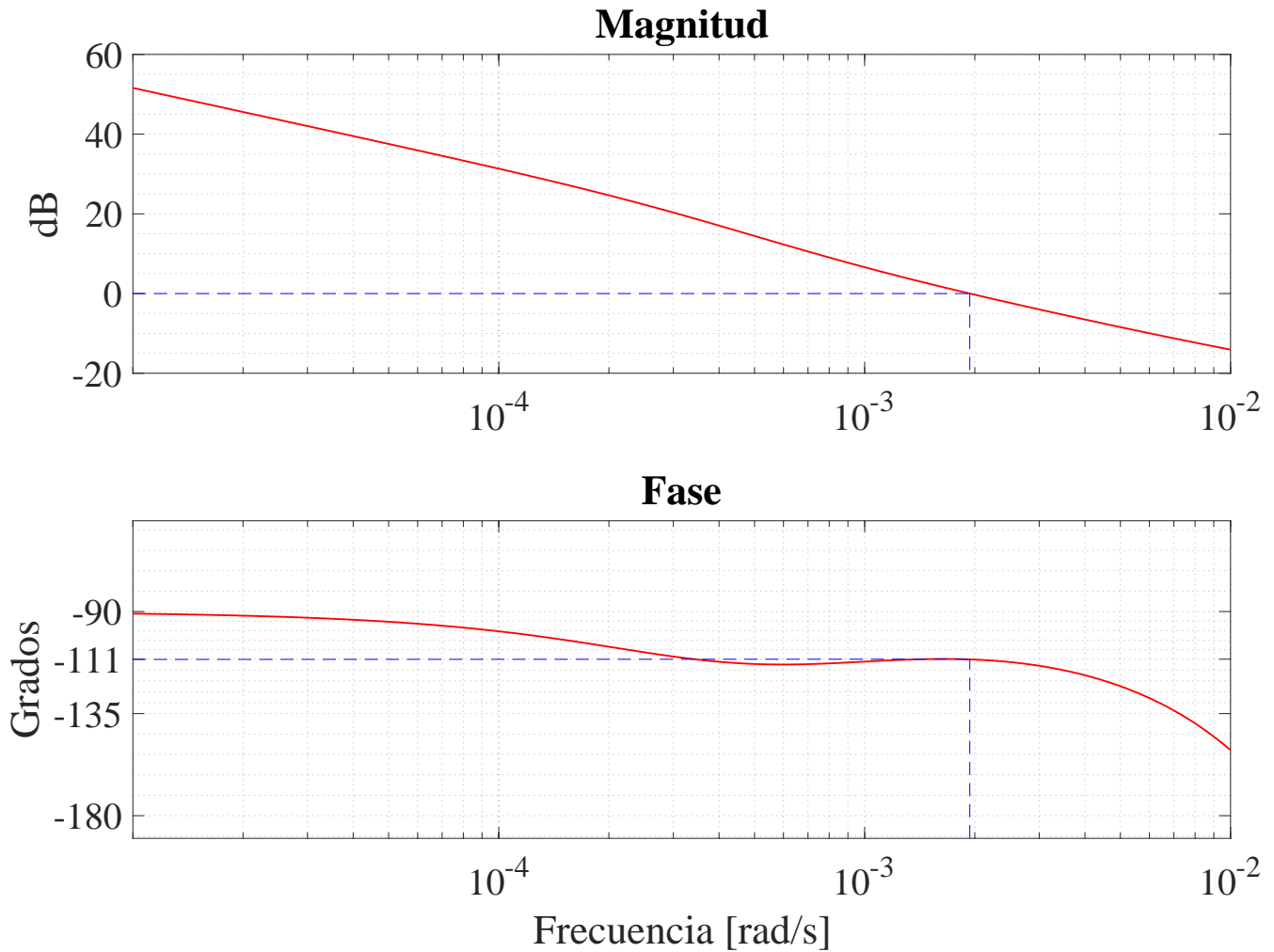


Figura 2.13: Diagrama de Bode G_{total}

Control PID discreto

A partir de lo presentado en la sección 2.2.5.3 se implementó un controlador PID discreto. Para pasar del campo continuo al campo discreto, se transformó la variable continua s en la variable discreta z utilizando la aproximación Backward-Euler, representada en la ecuación 2.17.

$$z = e^{sT} \approx \frac{1}{1 + sT_s} \quad (2.17)$$

$$s \approx \frac{1 - z^{-1}}{T_s} \quad (2.18)$$

Por simplicidad se presenta la ecuación en diferencia asociada a cada componente del controlador PID, es decir, una ecuación par el bloque proporcional, otra para el bloque

integral y otra para el bloque derivativo. Esta forma de presentar los equivalentes discretos permite una visualización mas clara y al mismo tiempo representa la manera en la que el controlador fue implementado en el código.

A continuación, en las ecuaciones 2.19, 2.20 y 2.21 se presentan las funciones de transferencia continuas asociadas a la componente proporcional, integral y derivativa respectivamente.

$$Y_{c_p}(s) = k_p X(s) \quad (2.19)$$

$$Y_{c_i}(s) = \frac{k_i}{s} X(s) \quad (2.20)$$

$$Y_{c_d}(s) = \frac{k_d s}{\frac{s}{\omega_p} + 1} X(s) \quad (2.21)$$

Discretizando las ecuaciones por medio de la aproximación Backward Euler representada en la ecuación 2.18, se obtienen las ecuaciones 2.22, 2.23 y 2.24.

$$y_{c_{p_d}}[n] = k_p x[n] \quad (2.22)$$

$$y_{c_{i_d}}[n] = k_i T_s x[n] + y_{c_{i_d}}[n-1] \quad (2.23)$$

$$y_{c_d}[n] = \frac{k_d \omega_p}{1 + T_s \omega_p} x[n] - \frac{k_d \omega_p}{1 + T_s \omega_p} x[n-1] + \frac{1}{1 + T_s \omega_p} y_{c_d}[n-1] \quad (2.24)$$

Por lo tanto, la salida total del controlador es la suma de las componentes discretas representadas en las ecuaciones 2.22, 2.23 y 2.24.

$$\left\{ \begin{array}{l} k_p = 4,6894 \\ k_i = 0,0032 \\ k_d = 271,1011 \\ T_s = 5s \\ \omega_p = 1 \frac{rad}{s} \end{array} \right. \quad (2.25)$$

Capítulo 3

Implementación

3.1. Análisis circuital

En la figura, se puede observar un diagrama en bloques completo del sistema.

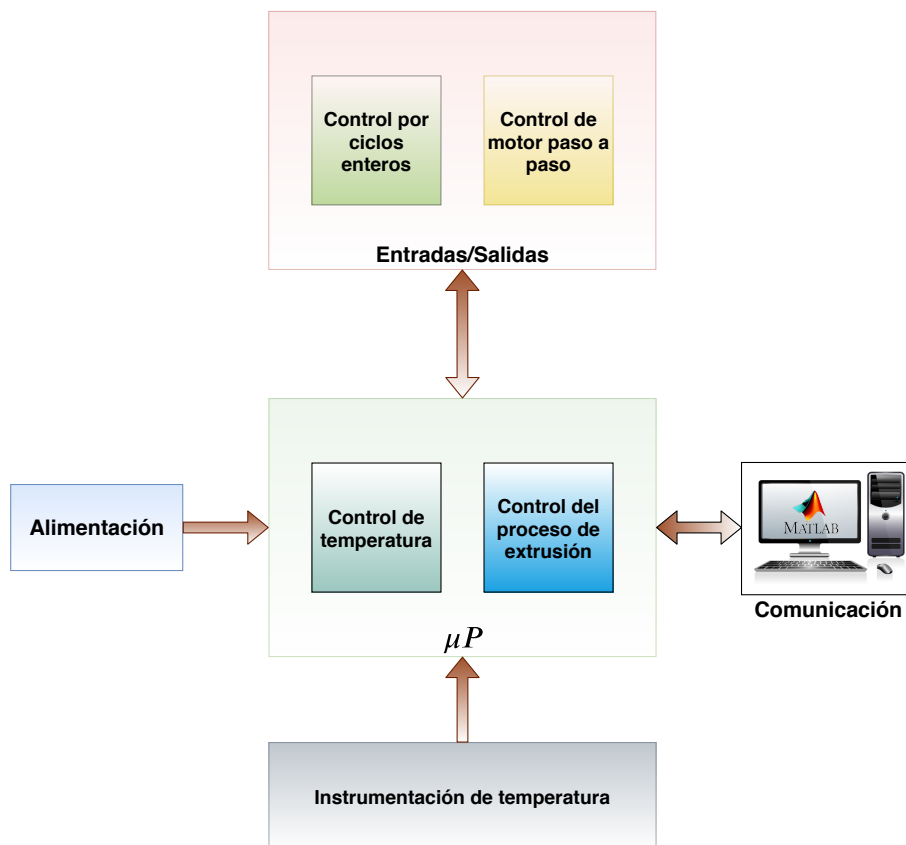


Figura 3.1: Diagrama en bloques general

A continuación se detallarán estos bloques fundamentales.

3.2. Alimentación

El equipo posee 2 fuentes de alimentación. Una generada a partir de un transformador de 220VAC a 48VAC encargada de alimentar el controlador del motor paso a paso que mueve el pistón de la extrusora y la otra generada a partir de un transformador de 220VAC a 12VAC para alimentar la circuitería del sistema. Esta última se rectifica y filtra, obteniendo así una tensión continua de aproximadamente 12V, denominada V_{in} . Posteriormente, dicha tensión es regulada mediante dos convertidores buck de 5V y 12V los cuales pueden verse en las figuras 3.3 y 3.4 respectivamente. El diseño de de estos convertidores fue tomado de la placa de evaluación LM2596S-ADJEVM [1], eligiendo los valores de las resistencias para obtener la tensión de salida deseada.

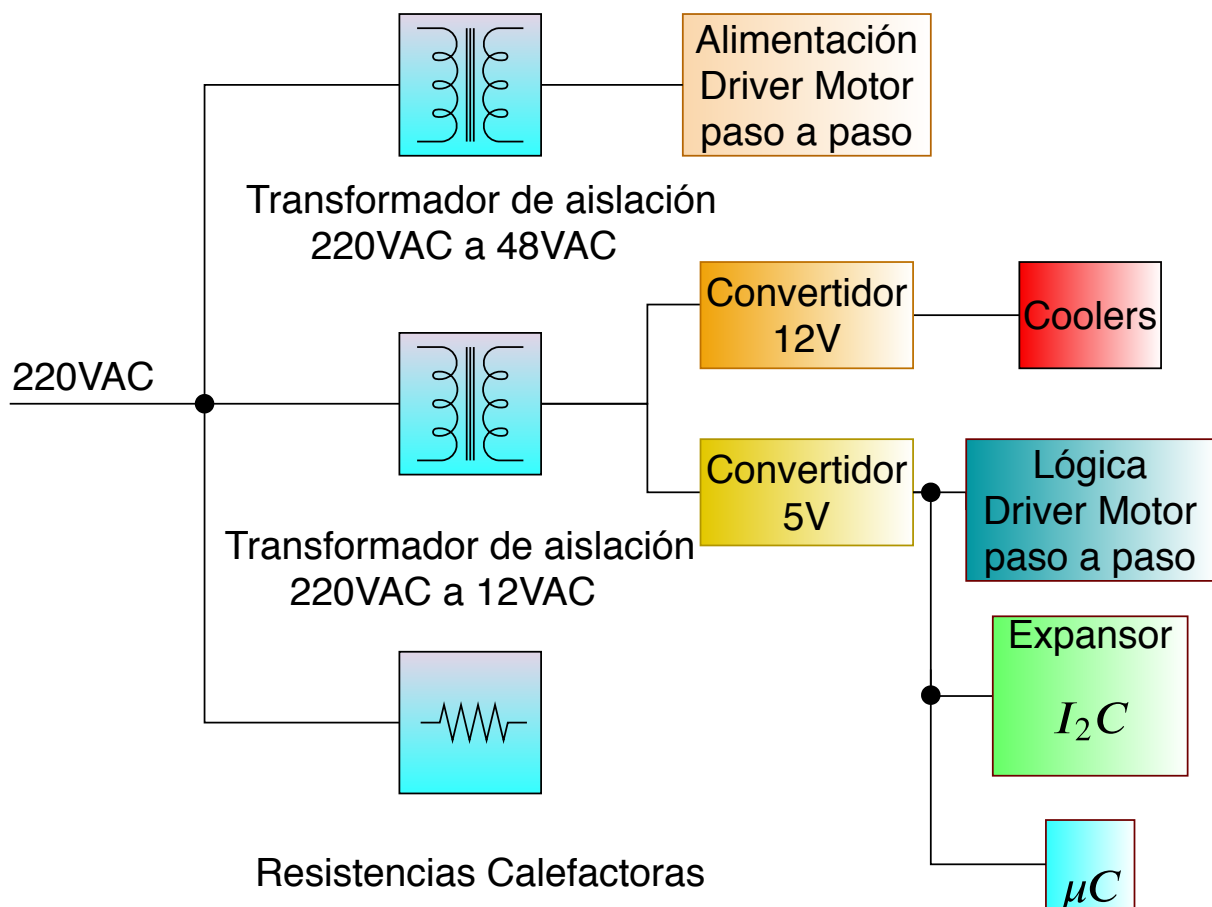


Figura 3.2: Diagrama en bloques fuente de alimentación

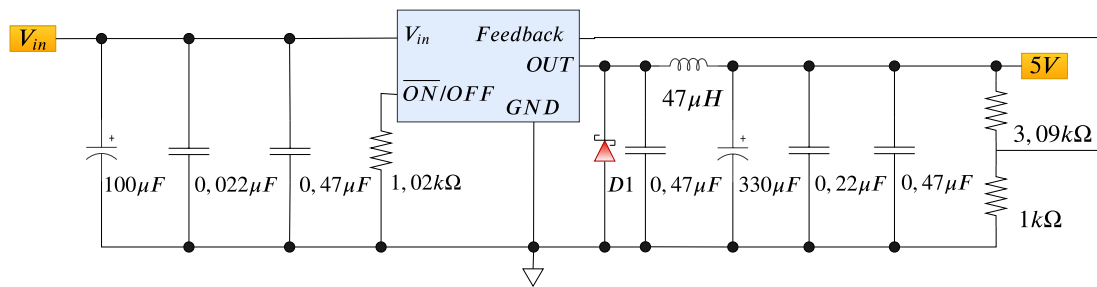


Figura 3.3: Convertidor Buck 5VCC

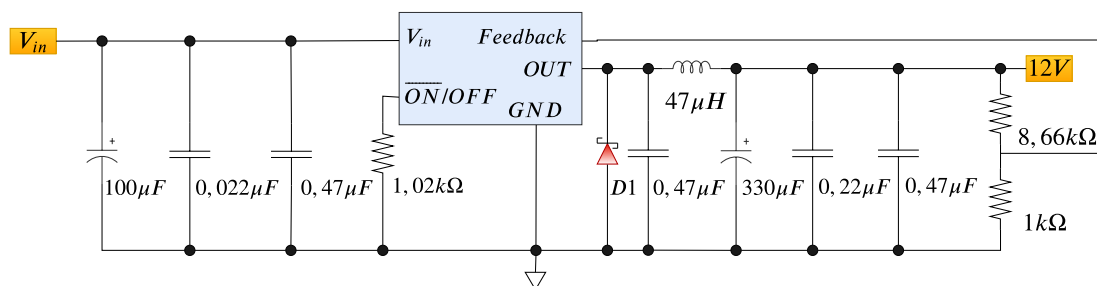


Figura 3.4: Convertidor Buck 12VCC

Existe una alimentación diferente para alimentar el controlador del motor paso a paso de la extrusora. Es un transformador de ± 24 VAC. El conexionado se realizó sin utilizar el punto medio, por lo tanto, la tensión de salida es de 48 VAC¹

3.3. Medición de temperatura

En la figura 3.5 se presenta un diagrama en bloques de la etapa de sensado. Este está compuesto por un sensor de temperatura del tipo Pt100, instrumentado en configuración 3 hilos, un filtro pasa-bajos activo y una etapa de corrección de offset. La función de esta etapa es ajustar el rango de las variables sensadas al rango de entrada del ADC para maximizar el rango de sensado. Es por ello que esta etapa en conjunto con la ganancia del filtro determinaran un mapeo de valores para lograr que el fondo de escala sea 3,3V (250°C).

¹Debido a que el controlador del motor admite ser alimentado por una tensión no regulada, la señal no fue rectificad.

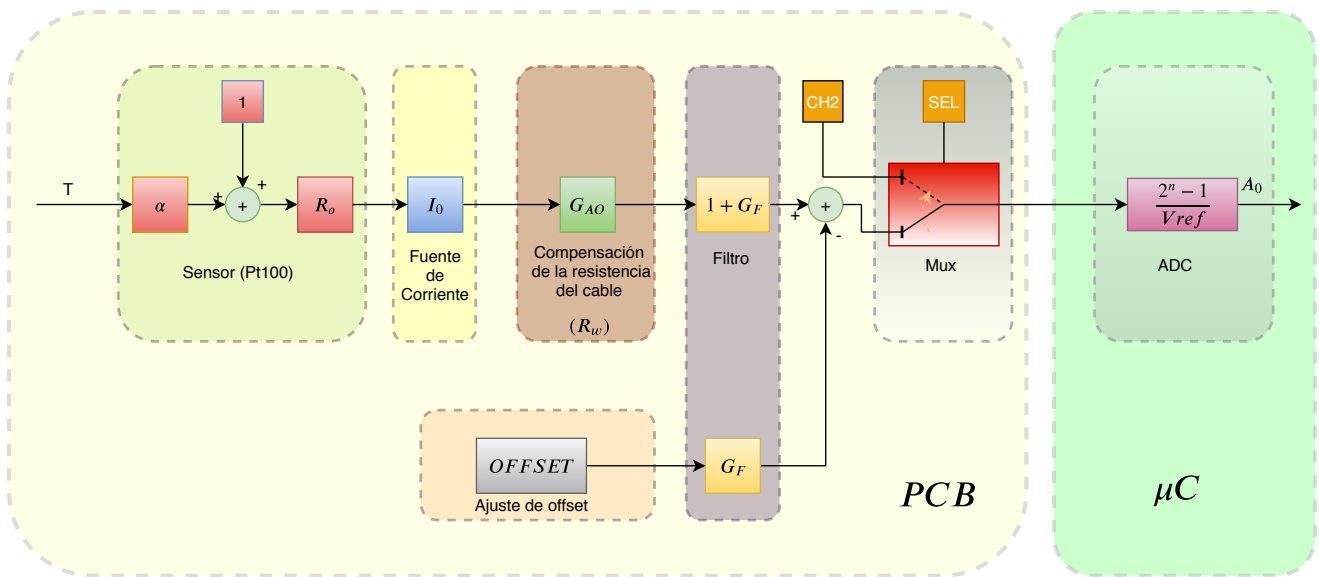


Figura 3.5: Diagrama en bloques de la etapa de medición de temperatura

En la figura 3.6 se puede observar un esquema circuital de la etapa de instrumentación. En el mismo se pueden identificar 6 etapas:

- Medición de temperatura
- Generador de corriente constante
- Compensación de la resistencia del cable (R_w)
- Filtrado
- Ajuste de offset
- Multiplexación de los canales de entrada

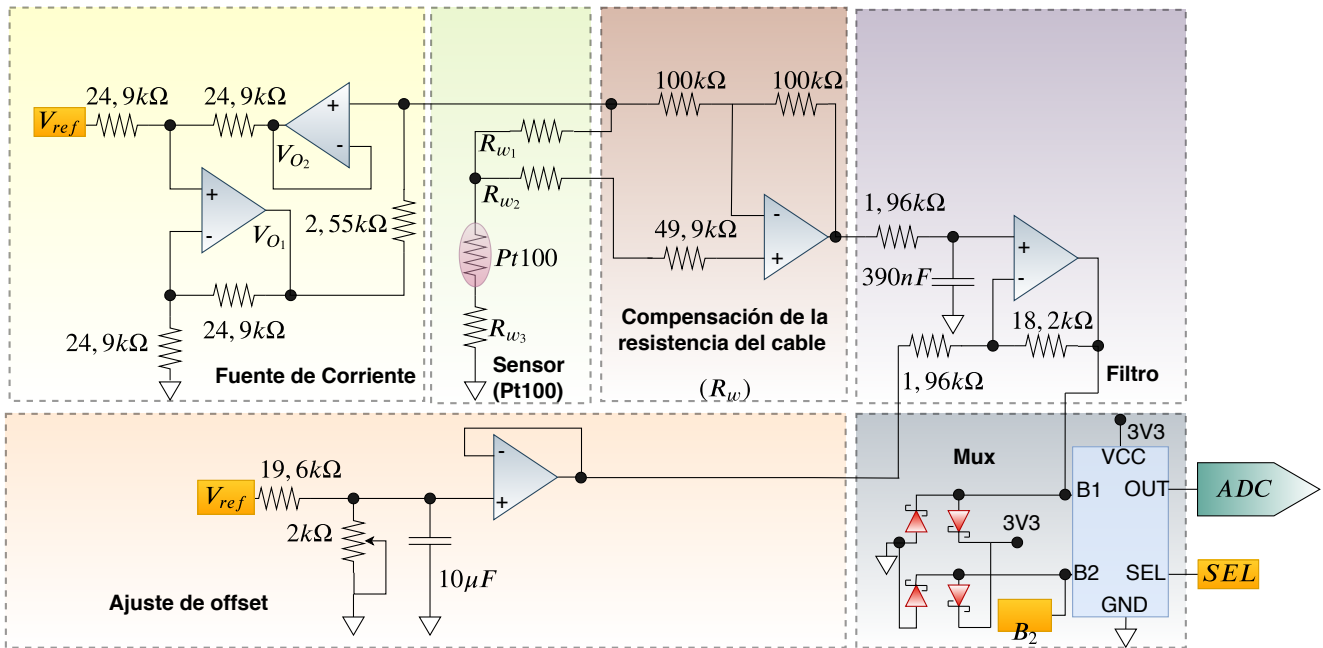


Figura 3.6: Esquema circuital instrumentación

A continuación, en las secciones 3.3.1, 3.3.2, 3.3.3 y 3.3.4 se detallarán las etapas anteriormente nombradas:

3.3.1. Generador de corriente constante

Este circuito permite generar una corriente constante, controlando la tensión sobre una resistencia de referencia, R_{REF} . Este esquema implementa una realimentación de tensión cuyo objetivo es tomar como referencia la tensión medida sobre el $Pt100$ para garantizar un valor constante de la corriente. La precisión del generador de corriente constante de la figura 3.7 está determinada por la resistencia de referencia $2,55K\Omega$. La estabilidad a largo plazo de la referencia de tensión $REF2925AIDBZT$ de Texas Instruments es de 24ppm para $0 \text{ horas} \leq t \leq 1000 \text{ horas}$ y de 15 ppm para $1000 \text{ horas} \leq t \leq 2000 \text{ horas}$. La exactitud del generador de corriente depende de la referencia y del valor real de R_{REF} .

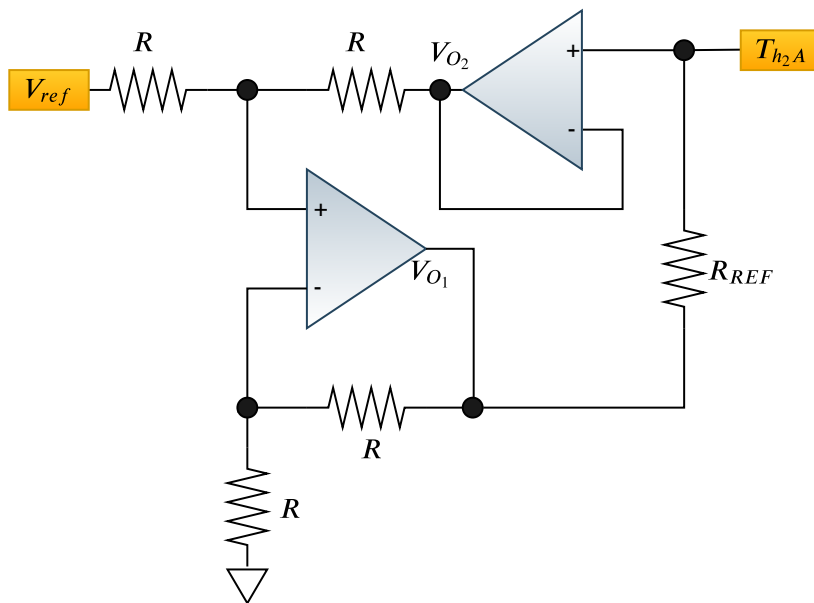


Figura 3.7: Generador de corriente constante

A continuación se detallan las ecuaciones que definen la corriente:

$$I_0 = \frac{V_{O1} - V_{O2}}{R_{REF}} \quad (3.1)$$

$$V_{O1} = \left(\frac{V_{REF} - V_{O2}}{2} + V_{O2} \right) * 2 = V_{REF} + V_{O2} \quad (3.2)$$

$$\frac{V_{REF} + V_{O2} - V_{O2}}{R_{REF}} = \frac{2,5V}{2,55k\Omega} = 980,3\mu A \quad (3.3)$$

3.3.2. Compensación de la resistencia del cable (R_w)

Debido a que un RTD es una resistencia variante con la temperatura, para realizar la medición, se hará circular una corriente conocida a través de esta resistencia y al medir la tensión, se puede obtener indirectamente el valor de la temperatura. El inconveniente radica en que los cables de conexionado entre el circuito y la etapa de medición, son cables que poseen una resistencia por unidad de longitud que también varía con la temperatura. Es por ello que este esquema implementa una instrumentación por 3 hilos, la cual emplea un 3^{er} conductor, por el cual no circula corriente, de manera de poder medir la caída de tensión en el cable (R_{w_1}). Th_{1A} representa uno de los bornes de la termocupla A . Th_{2A} representa otro de los bornes de la misma termocupla y por último, V_{TP_1} representa el *test point* 1.

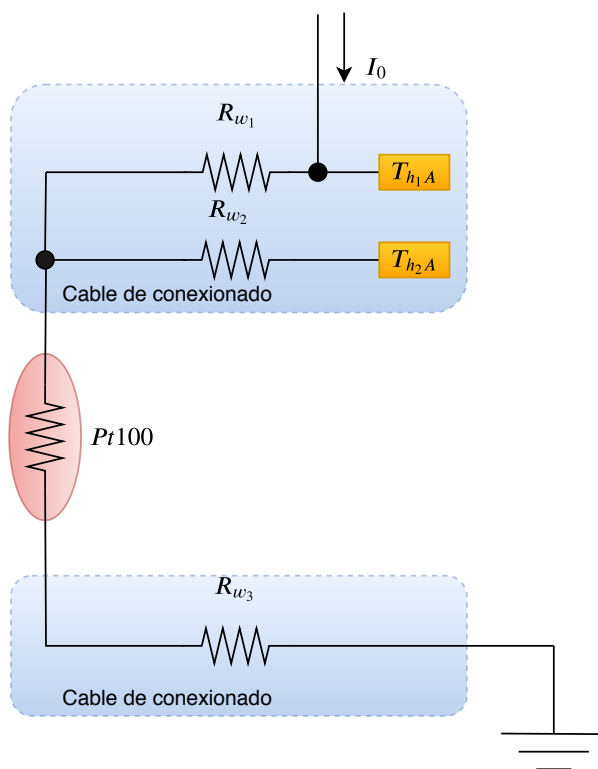


Figura 3.8: Pt100 instrumentado con 3 hilos

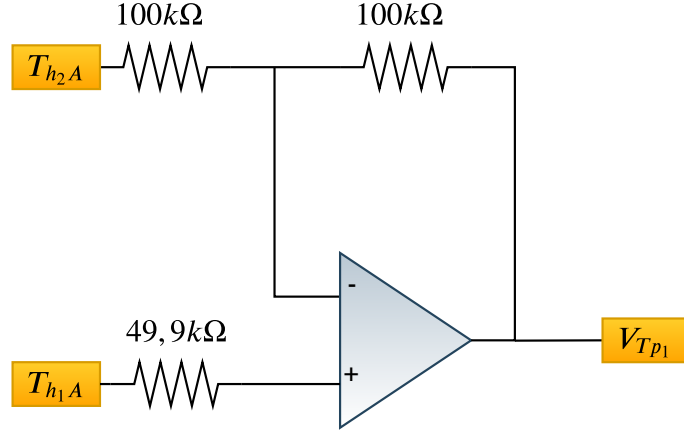


Figura 3.9: Circuito compensador de R_w

$$V_{TP1} = V_{Th1A} \left(-\frac{100K\Omega}{100K\Omega} \right) + V_{Th2A} \left(1 + \frac{100K\Omega}{100K\Omega} \right) = 2V_{Th2A} - V_{Th1A} \quad (3.4)$$

$$V_{Th2A} = I_0(RTD + R_{w3}) \quad (3.5)$$

$$V_{Th1A} = I_0(R_{w1} + RTD + R_{w3}) \quad (3.6)$$

Reemplazando (3.5) y (3.6) en (3.4)

$$V_{TP1} = 2I_0(RTD + R_{w3}) - I_0(R_{w1} + RTD + R_{w3}) \quad (3.7)$$

Como se asume que los cables de conexionado entre el sensor y la placa de adquisici3n son iguales, R_{w1} , R_{w2} y R_{w3} poseen el mismo valor. Por lo tanto, reescribiendo 3.7, obtenemos 3.8:

$$V_{TP1} = I_0RTD \quad (3.8)$$

Por lo tanto, en la ecuaci3n 3.8 se puede observar que la medici3n realizada con el RTD, utilizando la instrumentaci3n por 3 hilos, permiti3 independizarse de la impedancia del cable de conexionado. Es importante aclarar que la precisi3n de esta instrumentaci3n esta relacionada con la precisi3n de la fuente de corriente.

3.3.3. Filtrado

En la figura 3.10, se puede observar el esquema circuital del filtro utilizado².

²Un procedimiento para el dise1o de un filtro anti aliasing puede ser partiendo de la f_s y definiendo un valor de rechazo deseado se halla a continuaci3n la frecuencia de corte del filtro.

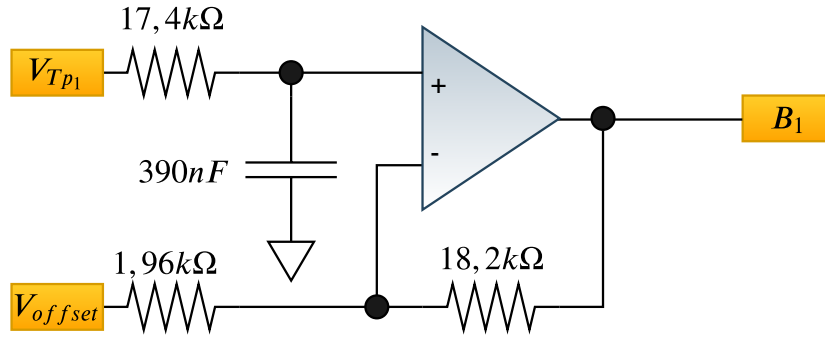


Figura 3.10: Filtro pasa bajos

$$V_{TP2} = V_{Ref}G_{AO}(1 + G_f) - V_{offset}G_f \quad (3.9)$$

$$G_f = \frac{V_{TP2} - V_{Ref}G_{AO}}{V_{Ref}G_{AO} - V_{offset}} \quad (3.10)$$

$$G_f = 10,28 \quad (3.11)$$

$$G_{AO} = 1,003 \quad (3.12)$$

Este, posee una ganancia de $G_f = 1 + \frac{18,2k\Omega}{1,96k\Omega} = 10,28^3$ y $f_c = 22,45 \frac{rad}{s}$. El objetivo de este filtro es eliminar ruido de alta frecuencia y evitar el aliasing [2]. En la ecuación 3.13 puede observarse la función transferencia del filtro.

$$G_f(j\omega) = \frac{\frac{18,2k\Omega}{1,96k\Omega}}{1 + j\omega RC} \quad (3.13)$$

³En la sección 3.3.4 se justificará el valor de G_f .

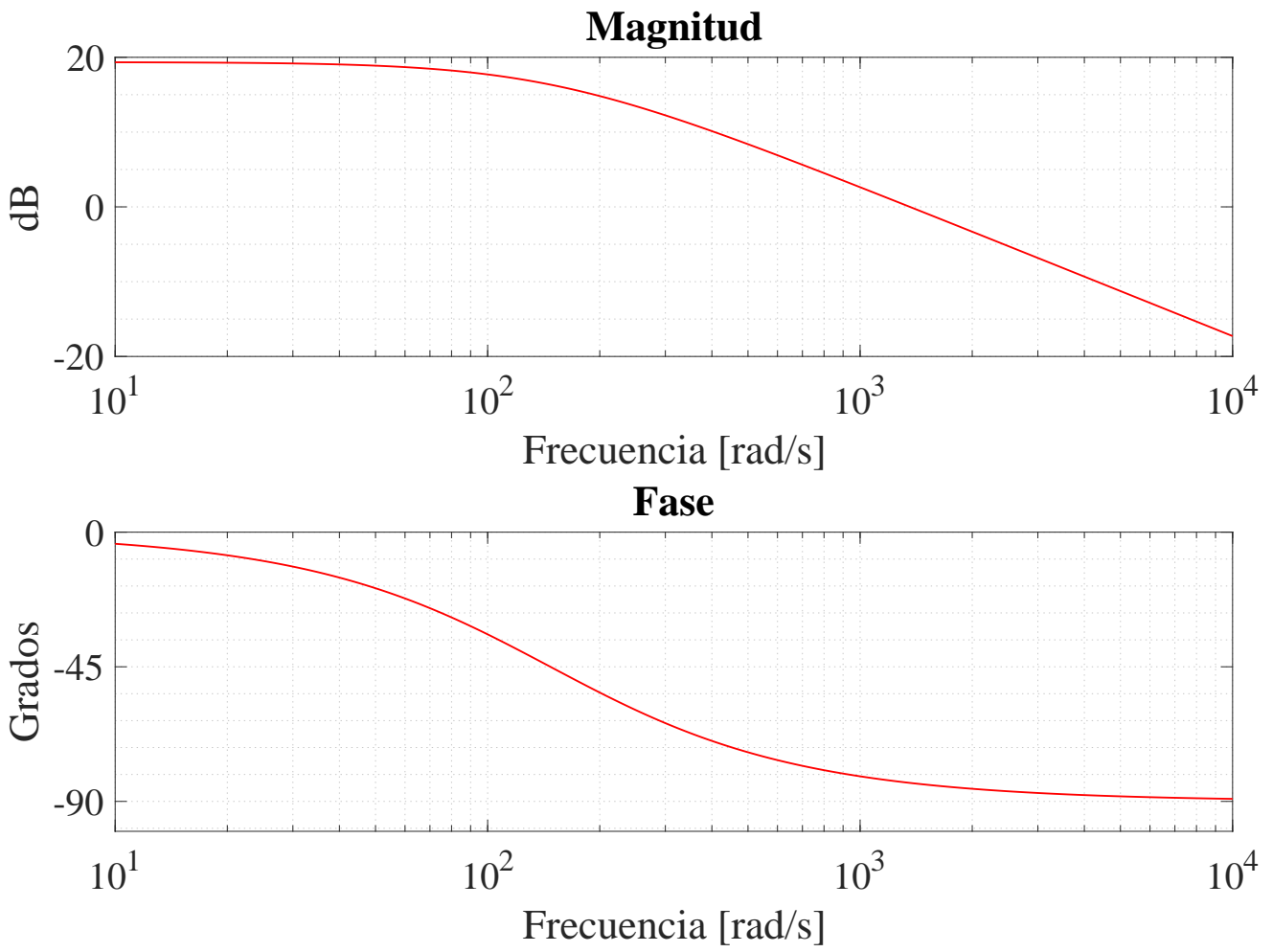


Figura 3.11: Diagrama de Bode del filtro pasa bajos

3.3.4. Ajuste de offset

En la fig. 3.12 se puede observar el esquema circuitual de la etapa de corrección de offset. Esto se logró ajustando la ganancia a $G_f = 10,28$ y $V_{offset} = 101,64mV$. La resolución del ADC es:

$$Resolucion_{tension} = \frac{3,3V}{1024bits} = 3,2 \frac{mV}{bit} \quad (3.14)$$

$$Resolucion_{temperatura} = \frac{250^{\circ}C}{1024bits} = 0,24 \frac{^{\circ}C}{bit} \quad (3.15)$$

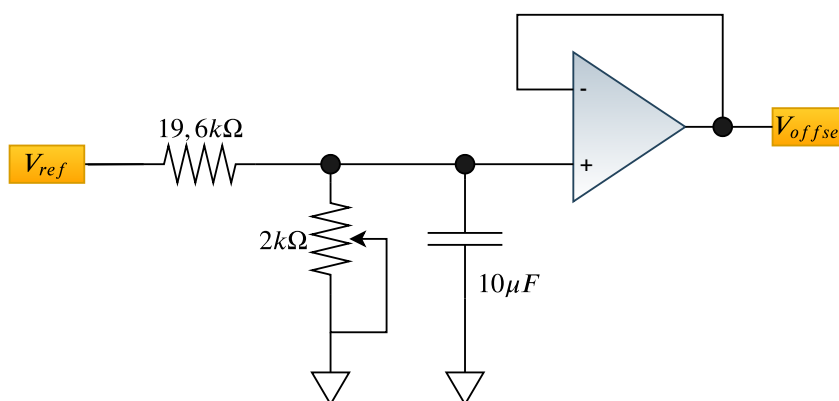


Figura 3.12: Esquema del circuito de ajuste de offset

3.3.5. Multiplexado y adquisición

Debido a que la placa de control posee un solo ADC, se implementó un esquema de multiplexado el cual permitió implementar dos canales de medición a pesar de tener uno solo de adquisición. Las entradas del multiplexor poseen enclavadores de tensión debido a que la tensión máxima de las entradas es $V_{CC} + 0,5V$. El ADC posee 10 bits y una tensión de fondo de escala de 1V, pero la placa de desarrollo posee un divisor resistivo para poder utilizar una tensión máxima de 3,3V. El sistema posee 2 sensores de temperatura. En B_1 y B_2 se pueden leer los valores de tensión asociados a la temperatura medida por cada sensor (luego de las etapas previamente explicadas).

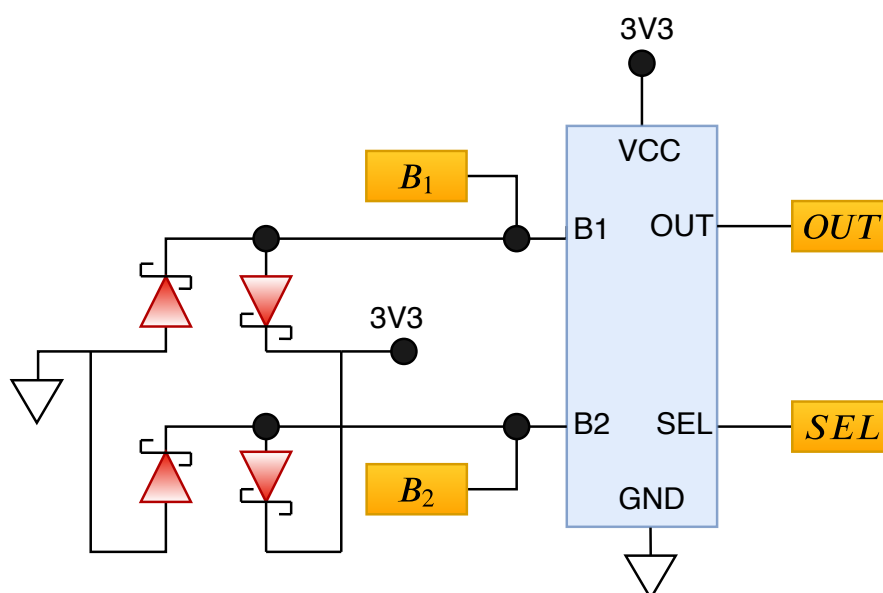


Figura 3.13: Esquema de multiplexación

La resistencia del *Pt100* en función de la temperatura se aproximó por la ecuación 3.16:

$$RTD(T) = R_0 (1 + \alpha T) \quad (3.16)$$

En donde $R_0 = 100\Omega$, $\alpha = 3,908310^{-3} \frac{1}{^\circ C}$

En la ecuación 3.17 se muestra el valor de tensión leído por el ADC.

$$\{[(\alpha T + 1) R_0 I_0 G_{AO} (1 + G_f)] - (V_{offset} G_f)\} = A_0 \quad (3.17)$$

$$A_0 \left(\frac{2^n - 1}{V_{ref}} \right) = D_0 \quad (3.18)$$

Siendo A_0 el valor de tensión leído por el conversor analógico digital y D_0 el valor digital convertido.

Por lo tanto, en la ecuación 3.17, para obtener el valor de temperatura utilizado en el código. El proceso de normalización implementado para convertir el valor digital D_0 en un valor de temperatura T , se define en la ecuación 3.19.

$$\frac{1}{\alpha} \left[\frac{D_0 + (V_{offset} G_f)}{R_0 I_0 G_{AO} (1 + G_f)} - 1 \right] = T \quad (3.19)$$

3.4. Señales de entrada y salida del sistema

En esta sección se presentarán de manera detallada las entradas y salidas del sistema [3] como así los métodos implementados para llevar a cabo las acciones necesarias para el funcionamiento del equipo. En la tabla 3.1 se presentarán las entradas y salidas de la placa de desarrollo NodeMCU-ESP8266 en conjunto con sus pines y la descripción de la función y en la tabla 3.2 se presenta la misma información pero relacionada al expansor de puertos *I2C* MCP2003. La frecuencia de adquisición se definió en $1kHz$.

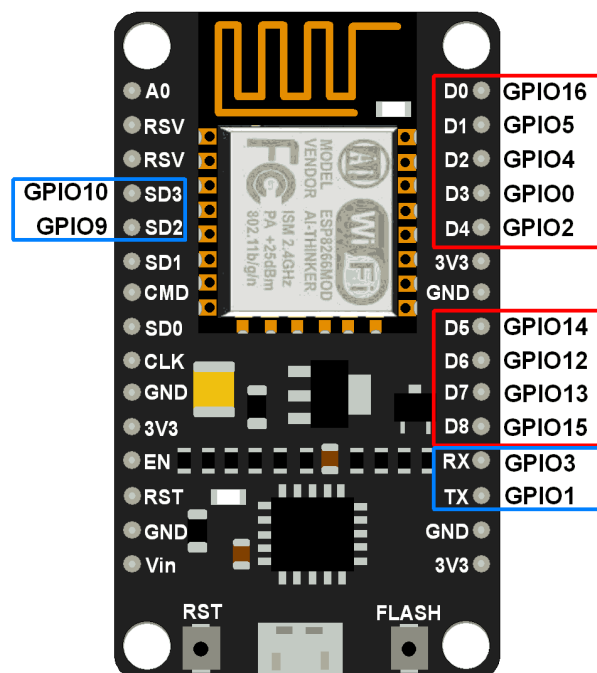


Figura 3.14: Distribución de pines NodeMCU-ESP8266

Variable	Pin	E/S	Función
LedPlaca	15	Salida	Led testigo de muestreo. Esta salida se puso en alto al inicio de la rutina de muestreo y luego en bajo al finalizar la misma. Esto permitió medir la duración de la interrupción y determinar si era menor a T_s . A futuro puede ser utilizada como entrada para un sensor que pueda medir el diámetro del filamento.
LedSel	12	Salida	Led asociado al bit de selección del canal del multiplexor
Interrupciones	14	Entrada	Interrupciones en el MCP2003 (No implementado).
DirBobinadora	2	Salida	Define el sentido de giro del motor de bobinado.
StepBobinadora	0	Salida	Define la velocidad de giro del motor de bobinado.
ResistenciaCuerpo	13	Salida	Control de la resistencia trasera.
ResistenciaBoquilla	16	Salida	Control de la resistencia delantera.
SCL	5	Salida	Serial Clock (<i>I2C</i>). Genera la base de tiempo para controlar al expansor de puertos MCP2003
SDA	4	Salida	Serial Data (<i>I2C</i>). Canal de datos entre el expansor y la placa de control

Cuadro 3.1: Tabla de entradas/salidas NodeMCU-ESP8266

Para configurar los pines SDA y SCL debe editarse la línea de código que se encarga de definirlos, en la librería *Adafruit_MCP23008.cpp*⁴ tal como se muestra a continuación:

```
1 Wire.begin(4,5);
```

⁴Ver código en el apéndice C.2

Variable	Pin	E/S	Función
DirExtrusion	1	Salida	Define el sentido de giro del motor de extrusión.
VelExtrusion	2	Salida	Define la velocidad de giro del motor de extrusión.
Coolers	4	Salida	Controla un circuito con MOSFET de canal N, para encender o apagar los coolers.
FinDeCarreraT	5	Entrada	Una ranura sobre la parte trasera de la cremallera acciona un switch que permite detectar el fin de carrera.
FinDeCarreraD	6	Entrada	Una ranura sobre la parte delantera de la cremallera hace cambiar el valor de un switch y genera un cambio de estado.
ParadaDeEmerg	7	Entrada	Switch de seguridad.

Cuadro 3.2: Tabla de entradas/salidas MCP2003

3.5. Procesamiento

Introducción

A continuación se detallarán aspectos relativos al procesamiento de la información adquirida en las entradas detalladas en la sección 3.4 y su representación.

3.5.1. Comunicación

Se implementó una comunicación entre una PC y la placa por USB. El puerto COM fue configurado con los parámetros mostrados en el cuadro a continuación:

Función	Parámetro	Valor	Detalle
Baud Rate	serialport.BaudRate	115200	Número de símbolos por segundo en el canal de comunicación digital [4].
Control de flujo de datos	serialport.FlowControl	ninguno	Especifica el tipo con control de flujo de datos a utilizar (handshaking) [5].
Bit de paridad	serialport.Parity	ninguno	Especifica el tipo de control de paridad [6].
Bit de stop	serialport.StopBits	1	Indica la cantidad de bits utilizados para determinar el fin del byte [7].
Bits de información	serialport.DataBits	8	Especifica la cantidad de bits de datos a transmitir [8].
Tamaño del buffer de entrada	serialport.InputBufferSize	5000000	Especifica la cantidad de bits de información que pueden ser almacenados en el buffer de entrada del software durante una operación de lectura [9].
Timeout	serialport.Timeout	10 s	Especifica el tiempo máximo en segundos para completar un proceso de lectura o escritura [10].

Cuadro 3.3: Tabla de parámetros de la etapa de comunicación

3.5.2. Interfaz de usuario

En la imagen 3.15 pueden observarse los gráficos adquiridos y representados por MatLAB. Aquí puede verse la temperatura del pistón, la *temperatura del cuerpo* y la

temperatura de referencia. La potencia inyectada, el error de temperatura (entre la temperatura del pistón y la temperatura de referencia) y el valor del integrador. También se representa un valor llamado *inicio timer*. Este valor indica cuando se inicializa un timer cuya función es medir el tiempo desde que la temperatura medida alcanza el 99% de la temperatura de referencia hasta que la rampa de referencia evoluciona a su próximo valor. Como trabajo posterior se prevé el desarrollo de una interfaz para que el usuario final pueda interactuar con el equipo. En este caso, la única información de la mostrada en la figura 3.15 que se representará será la referencia de temperatura y la temperatura medida, ya que el resto es información irrelevante para un usuario.

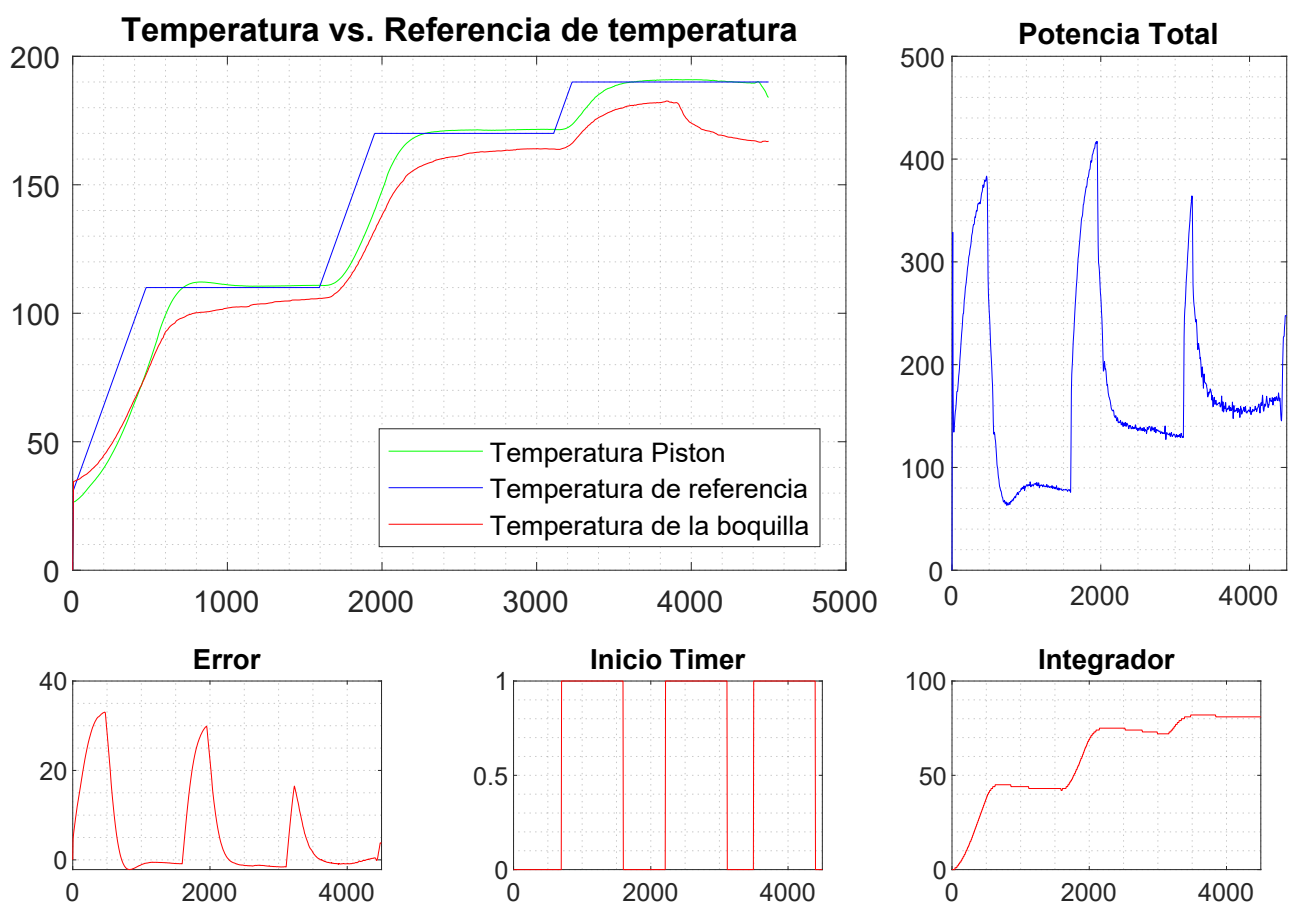


Figura 3.15: Información adquirida en MatLAB

3.5.3. Máquina de estados

El proceso de extrusión se implementó mediante una máquina de estados. Se optó por esta técnica con el objetivo de establecer de manera clara y sencilla las relaciones entre

cada etapa del proceso, que variables determinan la evolución de un estado a otro y que salidas se ven modificadas en cada una de las transiciones. El prototipo posee 2 fines de carrera, uno en cada extremo de la cremallera y un switch de parada de emergencia. Se define como alto (HIGH) al fin de carrera presionado. Una consideración de seguridad determina que si el switch Parada de Emergencia se encuentra en bajo, la maquina de estados detenga su evolución, deteniendo el funcionamiento del motor y colocando las salidas de las resistencias en bajo. A continuación se detallarán los estados que la componen y sus funciones principales. Dentro de la maquina de estados principal, existe una segunda máquina de estados que permite hacer evolucionar un contador, con el objetivo de establecer la posición correcta del embolo sobre el fin del recorrido. Esto es debido a que la posición se determina cuando el fin de carrera cambia de estado al pasar la segunda ranura de la cremallera por encima del mismo.

Diagrama de estados

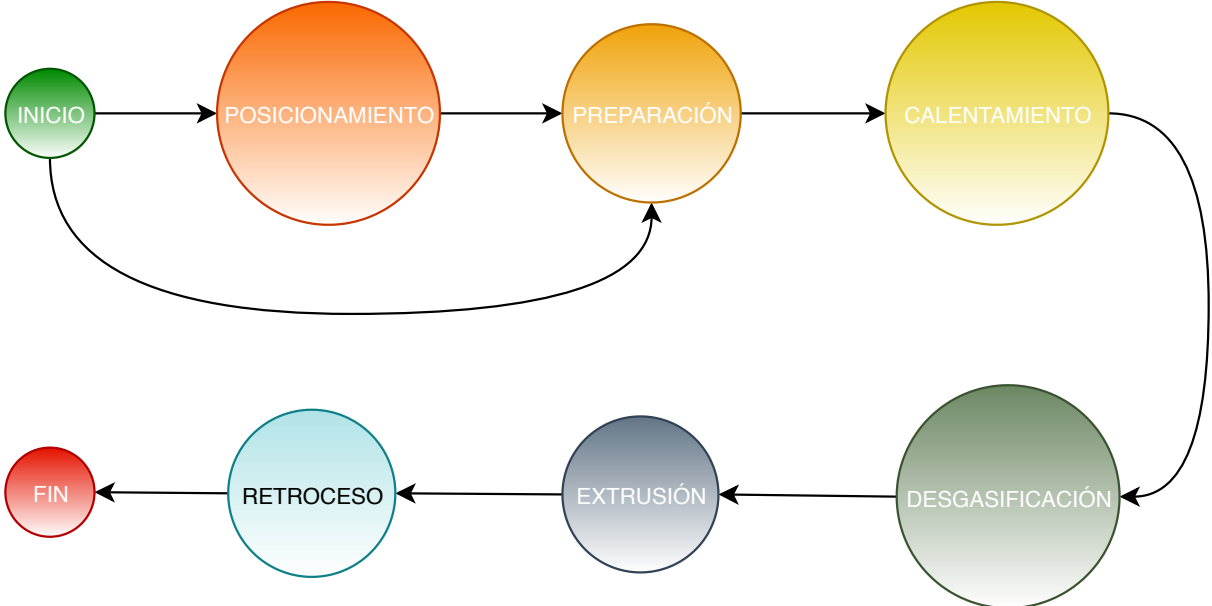


Figura 3.16: Diagrama de la máquina de estados principal

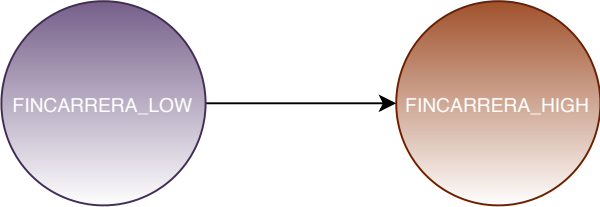


Figura 3.17: Diagrama de la máquina de estados secundaria

Inicio

Este estado puede definirse como un estado de transición. Esto significa que por defecto, la maquina de estados comienza en este estado y en función de la condición del fin de carrera delantero, evolucionará a los estados posicionamiento o preparación. En este estado, se accionan los coolers.

A continuación, en la tabla 3.4 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Inicio	FinDeCarreraDelantero == HIGH	Posicionamiento	-
	FinDeCarreraDelantero == LOW	Preparación	

Cuadro 3.4: Estados futuros posibles a partir del estado inicio

Posicionamiento

Este estado se encarga de posicionar la cremallera en una posición de referencia. Esto se implementó haciendo desplazar la misma hasta que FinDeCarreraDelantero == LOW. A continuación, en la tabla 3.5 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Posicionamiento	FinDeCarreraDelantero == HIGH	Posicionamiento	Ver 3.5.3
	FinDeCarreraDelantero == LOW	Preparación	

Cuadro 3.5: Estados futuros posibles a partir del estado posicionamiento

Salidas asociadas al estado posicionamiento

- ***PeriodoMotor=Ton*2***: define el ancho del pulso del PWM que determina la velocidad del motor que desplaza la cremallera.
- ***MotorRetroceso()***: esta función implementa un control de velocidad del motor de extrusión por PWM.

Preparación

Este estado se encarga de posicionar la cremallera en un determinado punto de la camisa. Es importante destacar que esta posición depende de varios factores. Uno de ellos es la cantidad de material aportado y la forma del mismo, dado que el volumen de un material pelletizado es mucho mayor que la de un material en forma de polvillo. Si no se tiene en cuenta este factor, el sistema intentará comprimir un polímero en estado sólido y el motor comenzará a perder pasos una vez que haya superado el torque dinámico máximo.

A continuación, en la tabla 3.6 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Preparacion	ContadorPosicion < PosicionPistonArranque	Preparación	Ver 3.5.3
	ContadorPosicion ≥ PosicionPistonArranque	Calentamiento	

Cuadro 3.6: Estados futuros posibles a partir del estado preparacion

Salidas asociadas al estado preparación

- ***PeriodoMotor=Ton*2***: define el ancho del pulso del PWM que determina la velocidad del motor que desplaza la cremallera.
- ***MotorAvance()***: esta función implementa un control de velocidad del motor de extrusión por PWM.
- ***ContadorPosicion++***: incrementa el valor del contador que permitirá identificar cuando la cremallera llegue a la posición preestablecida.
- ***delay(1500)***: Genera un delay de 1,5 segundos.
- ***ContadorPosicion=0***: Resetea el valor del contador. Este contador se incrementa a cada paso del motor hasta llegar a la posición deseada.

Calentamiento

Este estado se encarga principalmente de iniciar la rutina de calentamiento y de ejecutar el control de temperatura. Una vez inicializada la rutina de calentamiento, este estado se encarga de comparar si la temperatura medida alcanza el 99 % de la temperatura de referencia. Si esto es verdadero, inicializa un timer para garantizar que el sistema regule su temperatura al valor de referencia durante un determinado tiempo, tiempo necesario para que el polímero absorba energía y cambie de fase. Una vez que el timer llega a su valor de referencia en minutos, el valor de la referencia de temperatura cambia y comienza a crecer con la pendiente de calentamiento original hasta el nuevo valor de referencia de temperatura, repitiendo este proceso. En caso que se haya cumplido la ultima etapa de calentamiento (se consideró un perfil de calentamiento en 3 etapas), se da inicio al proceso de extrusión. Debido a que el timer debe medir fielmente el tiempo entre etapas de calentamiento, este se implementó dentro de la función ***muestreo()***, garantizando una base de tiempo confiable, dado que esta función se ejecuta cada $PeriodoMuestreo = 1ms$.

A continuación, en la tabla 3.7 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Calentamiento	IndiceTemperatura > 2 (Ver 3.5.3)	Desgasificación	Ver 3.5.3

Cuadro 3.7: Estados futuros posibles a partir del estado calentamiento

Condiciones de cambio asociadas al estado calentamiento

Como se comentó anteriormente, la evolución del estado ***calentamiento*** al estado ***desgasificación*** se hace de manera indirecta. Esto quiere decir que no se hace directamente dentro de la propia maquina de estados sino dentro de una función auxiliar llamada ***TimerTemperatura()***, la cual determina una frecuencia de ejecución precisa

y sirve como base de tiempo de referencia para establecer el período de tiempo previamente configurado entre etapas de calentamiento. Una vez alcanzada la última etapa y habiéndose cumplido el tiempo preestablecido, el sistema evoluciona al estado desgasificación si $\text{IndiceTemperatura} > 2$.

Salidas asociadas al estado calentamiento

- ***MuestreoTemperatura = 1***: Esta bandera determina que estando en la función *muestreo*, se ejecute la función que permite comenzar a adquirir la temperatura.
- ***InicioCalentamiento = 1***: Esta bandera determina que estando en la función *muestreo*, se ejecute la función que permite iniciar el calentamiento.
- ***InicioTimerTemp = 1***: Esta bandera determina que estando en la función *muestreo*, el *TimerTemp* comience a incrementarse hasta llegar al valor de referencia

Desgasificación

Este estado actualmente no se encuentra implementado.

Extrusión

Este estado se encarga de hacer que el pistón se desplace, presionando al polímero sobre la boquilla. La velocidad de desplazamiento debe ser baja para evitar que el motor pierda pasos. Si se intenta hacer que el polímero se extruya por la boquilla a una velocidad relativamente alta, la viscosidad del mismo generará una resistencia haciendo que se supere el torque dinámico máximo del motor.

A continuación, en la tabla 3.8 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Extrusión	$\text{ContadorAvance} < 2$	Extrusión	Ver 3.5.3
	$\text{ContadorAvance} \geq 2$	Retroceso	

Cuadro 3.8: Estados futuros posibles a partir del estado extrusión

Salidas asociadas al estado extrusión

- ***PeriodoMotor=Ton*1200***: define el ancho del pulso del PWM que determina la velocidad del motor que desplaza la cremallera.
- ***MotorAvance()***: esta función implementa un control de velocidad del motor de extrusión por PWM.

Máquina de estados secundaria

Debido a que la cremallera posee 2 ranuras en el extremo trasero debido a un error de mecanizado, se implementó una segunda maquina de estados, cuya función es incrementar el valor de un contador, el cual determinará la evolución desde el estado extrusión al de retroceso. A continuación, en la tabla 3.9 se detallará la información relativa a estos estados.

Estado Actual	Condición	Estado siguiente	Salida
FINCARRERA_HIGH	FinDeCarreraTrasero == LOW	FINCARRERA_LOW	Ver 3.5.3
FINCARRERA_LOW	FinDeCarreraTrasero == LOW	FINCARRERA_LOW	-
	FinDeCarreraTrasero == HIGH	FINCARRERA_HIGH	

Cuadro 3.9: Estados futuros posibles de la maquina de estados secundaria

Salidas asociadas al estado FINDECARRERA _ HIGH

- **ContadorAvance++**: incrementa el valor del contador que permitirá identificar cuando el pistón llegue al final de su recorrido (en avance).

Retroceso

Este estado se encarga de hacer retroceder la cremallera, una vez finalizado el proceso de extrusión. Una vez ocurrido esto, el sistema se encuentra listo para limpiarse, debido a que el pistón usualmente queda con restos de polímero fundido. Por otro lado se experimentó que si no se hacía retroceder el pistón luego de finalizado el proceso de extrusión, el mismo se pegaba a la camisa, dificultando el comienzo de un nuevo ciclo⁵.

A continuación, en la tabla 3.10 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Retroceso	FinDeCarreraDelantero == HIGH	Retroceso	Ver 3.5.3
	FinDeCarreraDelantero == LOW	Fin	

Cuadro 3.10: Estados futuros posibles a partir del estado extrusión

Salidas asociadas al estado retroceso

- **MuestreoTemperatura = 1**: Esta bandera determina que estando en la función *muestreo*, se ejecute la función que permite comenzar a adquirir la temperatura.
- **InicioCalentamiento = 0**: Esta bandera determina que estando en la función *muestreo*, no se ejecute la función que permite iniciar el calentamiento.
- **PeriodoMotor=Ton*2**: define el ancho del pulso del PWM que determina la velocidad del motor que desplaza la cremallera.
- **MotorRetroceso()**: esta función implementa un control de velocidad del motor de extrusión por PWM.

⁵Esto se solucionó inicializando la maquina de estados en el estado calentamiento.

Fin

Este estado se encarga de resetear la maquina de estados, cambiando el valor de la variable inicio. También detiene los coolers (los cuales se accionaron previo al estado inicio) y detiene la adquisición. Implementa también un encendido y apagado de una serie de leds sobre el PCB indicando que el proceso ha finalizado.

A continuación, en la tabla 3.11 se detallará la información relativa a este estado.

Estado Actual	Condición	Estado siguiente	Salida
Fin	-	Fin	-

Cuadro 3.11: Estados futuros posibles a partir del estado Fin

Salidas asociadas al estado Fin

- ***CoolersOff()***: esta función detiene el giro de los coolers que se encargan de refrigerar el piñon del motor y el circuito electrónico.
- ***MuestreoTemperatura = 0***: esta bandera determina que estando en la función *muestreo*, no se ejecute la función que permite comenzar a adquirir la temperatura.
- ***InicioCalentamiento = 0***: esta bandera determina que estando en la función *muestreo*, no se ejecute la función que permite iniciar el calentamiento.
- ***LedPlacaToggle()***: define el ancho del pulso del PWM que determina la velocidad del motor que desplaza la cremallera.
- ***Start = 1***: esta bandera determina que el sistema quede listo para un nuevo proceso de extrusión, **debido a que esta variable se condiciona al estado inicio**.

3.5.4. Modulación de potencia

Para regular la potencia inyectada al sistema, se implementó una modulación por ancho de pulso. Como se puede observar en la figura 3.18, el circuito correspondiente a la etapa de potencia está implementada mediante un triac NTGS3446. Dicho circuito permite, por un lado brindar la aislación y protección eléctrica necesaria, entre la etapa de baja y alta tensión y por el otro generar una acción de control por ciclos enteros. De esta forma la potencia aplicada consiste en medios ciclos de la tensión de red y las conmutaciones del triac se realizan por los cruces por cero de la tensión de red, evitando la generación de armónicos.

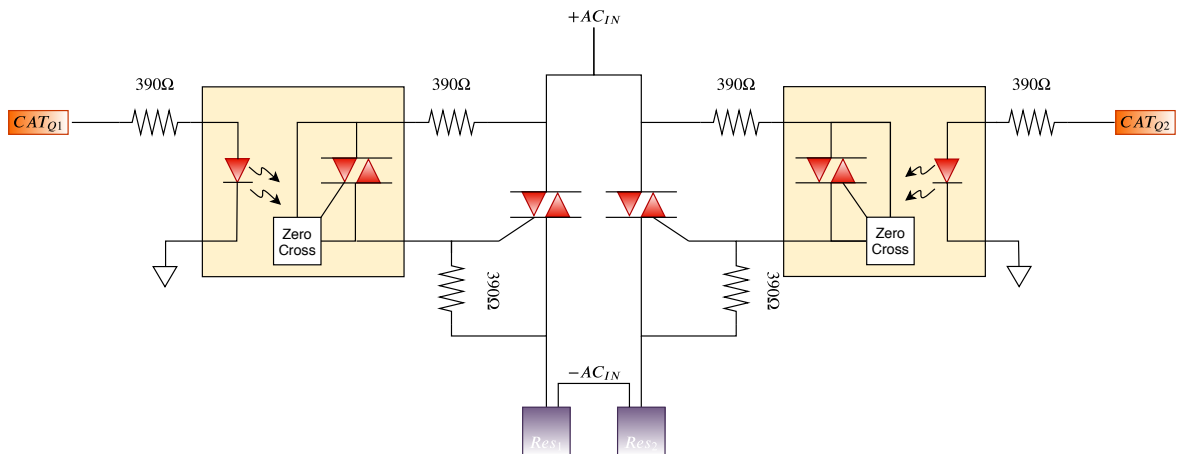


Figura 3.18: Esquema circuital modulación por ciclos enteros

El valor mas pequeño que se puede discernir en la forma de onda de red es el de un semiciclo, cuyo período es de $10ms$.

La placa de control NodeMCU-ESP8266 posee salidas PWM configurables. El inconveniente es que la frecuencia mínima es de 100 Hz. Esto significa que el periodo máximo es de $10ms$, exactamente la duración de un semiciclo de señal. Por lo tanto la únicas posibilidades podrían haber sido aplicar un semiciclo de señal o no aplicar nada, es decir, un control ON-OFF. Debido a esta limitación, se optó por implementar un PWM vía software. El mismo se basa en una rampa de periodo conocido y una referencia (que es el ciclo de trabajo). Cuando la rampa se encuentra por debajo de la referencia, la salida esta en alto y en caso contrario, la salida se encuentra en bajo.

La elección del periodo del PWM, debería ser tal que el polo de la planta proporcione un filtrado adecuado. El impacto finalmente es un ripple de temperatura filtrado. Si se considera un periodo de PWM de 5 segundos y el periodo del semiciclo es de $10ms$, en un periodo de PWM se pueden discernir 500 semiciclos de la señal de red. La potencia total es $464W$ y equivale a una resolución de temperatura de $1,09^{\circ}C$.

La potencia máxima de cada resistencia calefactora era levemente diferente, debido a diferencias constructivas, lo que redundaba en un gradiente de temperatura entre ambas. La potencia máxima de la resistencia del cuerpo (resistencia trasera) era de $228 W$, mientras que la potencia de la resistencia delantera era de $236 W$. Debido a esto se implementó una corrección a nivel de código para evitar el gradiente de temperatura entre los dos actuadores.

A continuación, en la figura 3.19 pueden observarse las formas de onda de tensión, corriente y potencia sobre una de las resistencias calefactoras y la señal de control.

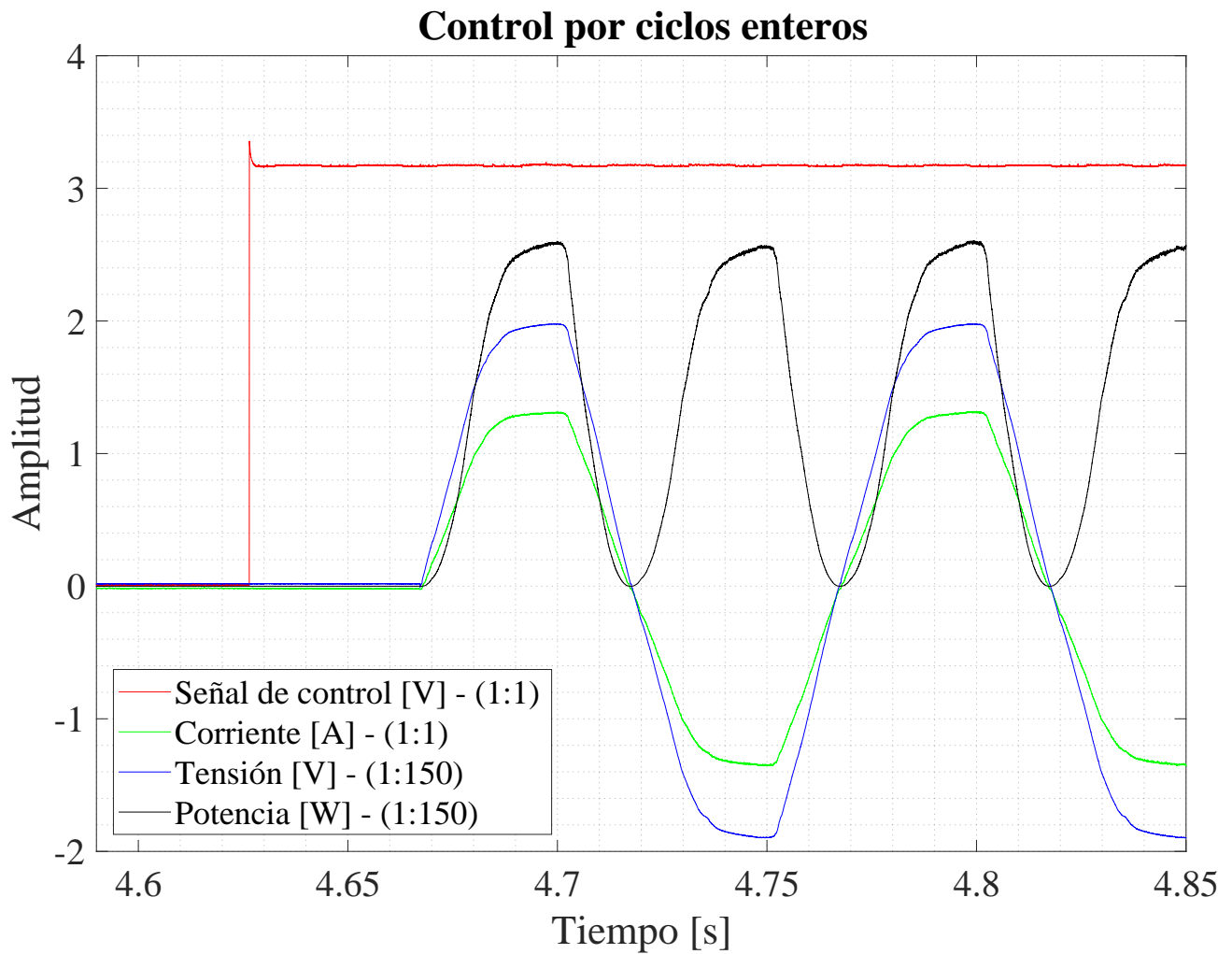


Figura 3.19: Modulación por ciclos enteros

En la ecuación 3.20 se presenta la ecuación que permite calcular la potencia sobre cada resistencia calefactora, en donde P es la potencia, V_{RMS} es el valor cuadrático medio de la tensión e I_{RMS} es el valor cuadrático medio de la corriente, ambos sobre la resistencia calefactora. D es el ciclo de trabajo.

$$P = V_{RMS}I_{RMS}D \quad (3.20)$$

Capítulo 4

Diseño mecánico

4.1. Introducción

En este capítulo se presentan de manera resumida las operaciones mecánicas realizadas para fabricar las piezas del equipo prototipado. Se presentan también los materiales elegidos, sus características, ventajas y desventajas frente a otros disponibles en el mercado. En el apéndice A podrán encontrarse los planos de cada pieza del equipo.

El equipo consta básicamente de una camisa de acero inoxidable, inserta en una camisa de aluminio. Por su interior se desplaza un embolo que es el encargado de desplazar el material y extruirlo sobre la boquilla (ésta se encuentra en el extremo de la camisa).

¿Por qué se implementó un sistema de dos camisas insertas y no una sola? Existen varias razones, las cuales se detallan a continuación. En la tabla 4.1 se presentan diversas características del aluminio y acero inoxidable¹.

Material	Conductividad térmica	Punto de fusión	Dureza	Densidad
Acero Inox.	$16,3 \frac{W}{\text{°Cm}}$	$1370^{\circ}\text{C} - 1400^{\circ}\text{C}$	$110HB$	$7850 \frac{kg}{cm^3}$
Aluminio	$237 \frac{W}{\text{°Cm}}$	660°C	$230HB$	$2700 \frac{kg}{cm^3}$

Cuadro 4.1: Propiedades del aluminio y del acero inoxidable

Debido a que el oxido de aluminio se deposita sobre la superficie del metal y al tener este un coeficiente de expansión térmica menor que el propio metal, en caso de que la camisa hubiese sido completamente de aluminio este compuesto con características cerámicas se desprendería y se transformaría en una impureza adicionada al polímero.

Otra opción fue la de realizar la camisa completa de acero inoxidable, pero debido a la dureza del material, el costo de las herramientas para mecanizarlo es muy superior al de las herramientas para mecanizar aluminio. Otro punto importante fue la diferencia en la conductividad térmica de cada material siendo que el aluminio conduce unas 14,5 veces mejor el calor que el acero inoxidable.

¹Los valores de dureza se encuentran expresados en escala Brinell

Debido a estas razones, se optó por utilizar una camisa de acero inoxidable inserta en un cuerpo macizo de aluminio, cuyo objetivo principal fue el de homogeneizar la temperatura para que el polímero en el interior de la camisa de acero inoxidable se caliente uniformemente.

Una ventaja importante es que en caso de existir en el futuro algún desgaste sobre la camisa de acero inoxidable, esta puede mecanizarse nuevamente sin tener que cambiar el cuerpo de la extrusora.

Por otro lado, tanto el pistón como la camisa al ser del mismo material, poseen el mismo coeficiente de expansión térmica lo que evitará que el huelgo entre estas piezas dependa de la temperatura.

4.2. Ajuste por interferencia

Debido a que se requería mantener el contacto entre la camisa de acero inoxidable y la camisa de aluminio en todo el rango de temperatura es que se llevó a cabo un ajuste por interferencia entre ambas piezas. Este tipo de ajuste se produce cuando se quiere acoplar dos piezas, una en el interior de la otra, siendo $\phi_{pieza_{interior}} > \phi_{pieza_{exterior}}$. En este caso, la pieza interior es la camisa de acero inoxidable y la pieza exterior es la camisa de aluminio. Debido a que estos materiales poseen diferentes coeficientes de dilatación térmica, fue necesario calcular la dilatación de cada pieza para un determinado ΔT . La diferencia de esos valores es la interferencia entre las piezas a temperatura ambiente. La ecuación que permitió realizar este cálculo es la 4.1.

$$\Delta\phi = \phi\alpha\Delta T; \tag{4.1}$$

Siendo $\Delta\phi$ el incremento del diámetro de un material de diámetro ϕ , coeficiente de dilatación térmica α , sometido a una variación de temperatura ΔT .

En la ecuación 4.2 se observa la dilatación sufrida por la camisa de aluminio y en la ecuación 4.3 la dilatación sufrida por la camisa de acero, cuando el conjunto pasa de $T_{amb} = 20^{\circ}C$ a $T_{trabajo_{max}} = 225^{\circ}C$. Se define entonces $\Delta T = T_{trabajo_{max}} - T_{amb} = 205^{\circ}C$.

$$\Delta\phi_{aluminio} = 26,3mm23x10^{-6} \frac{1}{^{\circ}C} 205^{\circ}C = 124\mu m \quad (4.2)$$

$$\Delta\phi_{acero} = 26,3mm12x10^{-6} \frac{1}{^{\circ}C} 205^{\circ}C = 64,7\mu m \quad (4.3)$$

La diferencia de los valores obtenidos en las ecuaciones 4.2 y 4.3 puede observarse en la ecuación 4.4.

$$\Delta\phi_{aluminio} - \Delta\phi_{acero} = 59,3\mu m \quad (4.4)$$

Debido a una limitación en el proceso de ensamblaje, se implementó un valor de interferencia menor, obtenido a partir de la ecuación 4.5.

$$\frac{\Delta\phi_{aluminio} - \Delta\phi_{acero}}{2} = 29,65\mu m \quad (4.5)$$

Con el valor de interferencia hallado en la ecuación 4.5 se puede asegurar contacto pleno de las piezas concéntricas hasta una temperatura igual a $117,18^{\circ}C$. Por lo tanto, se puede concluir que a $250^{\circ}C$ existirá un huelgo de $40,6\mu m$. En la figura 4.1, el área en amarillo representa el intervalo de temperaturas para las cuales existirá interferencia, mientras que el área sombreada en verde, representa el intervalo de temperaturas para las cuales existirá un huelgo entre ambas camisas.

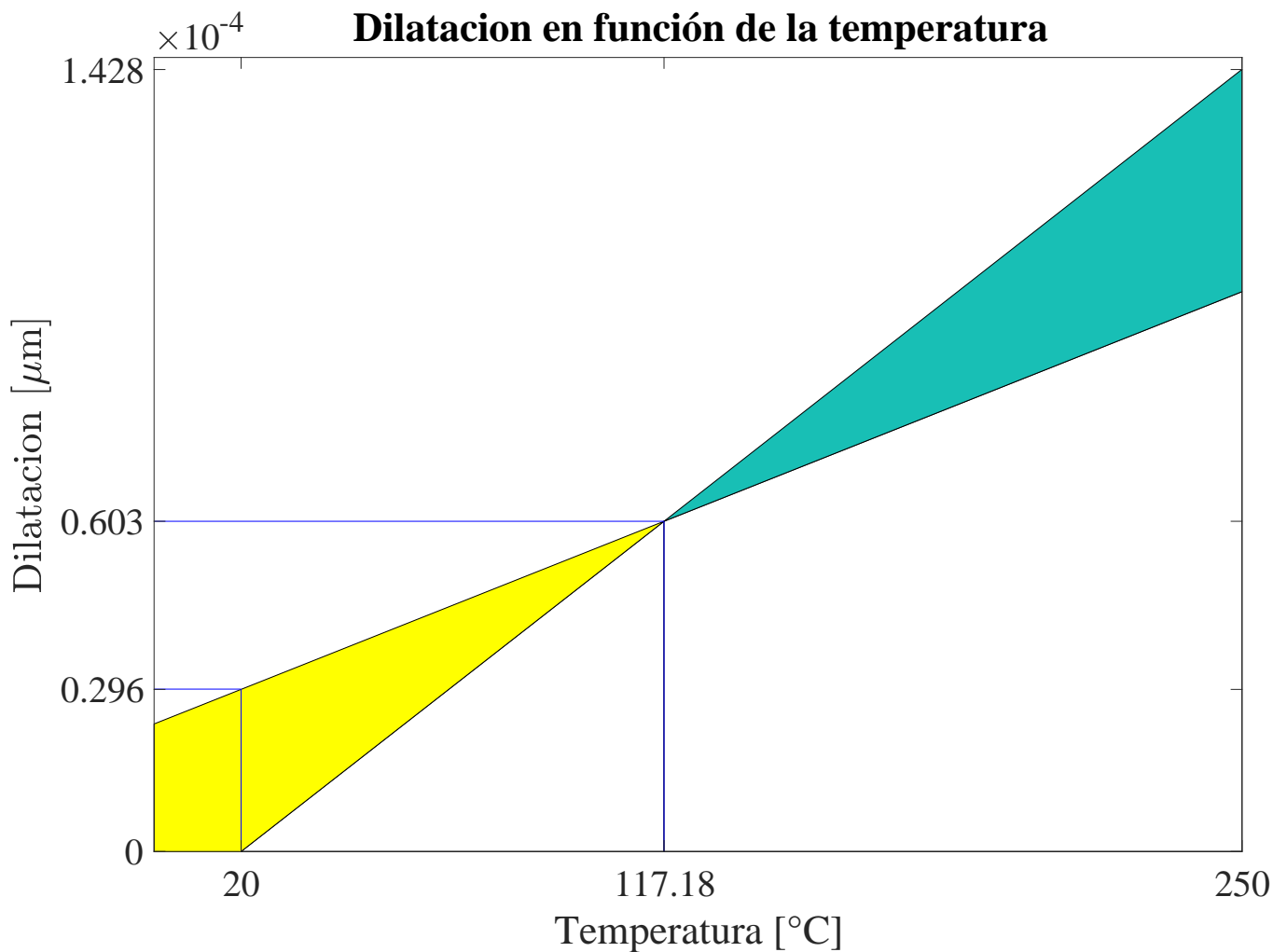


Figura 4.1: Gráfico interferencia entre la camisa de aluminio y la camisa de acero inoxidable

4.3. Transmisión de potencia

La extrusión del polímero se llevó a cabo mediante un motor paso tipo híbrido. Este tipo de motores, combina las mejores prestaciones de un motor de reluctancia variables con uno de imanes permanentes. El mismo posee 4 bobinas independientes y puede ser utilizado en las siguientes configuraciones:

- Unipolar
- Bipolar
 - Serie

- Paralelo

Teniendo en cuenta que el controlador solo permite el conexionado de motores bipolares (4 conductores), se debió adoptar la configuración de conexionado de los bobinados del motor. Una de las opciones era colocar las bobinas en paralelo de a pares dando como resultado bobinas con inductancia equivalente menor a la que posee cada bobina individualmente. Debido a que el torque depende de la magnitud de la inductancia, se estaría configurando al motor cuyo torque es bajo a bajas velocidades. La otra opción posible era conectar las bobinas en serie, esto permitirá obtener una inductancia mayor (y en consecuencia un mayor torque) a costa de una disminución de velocidad. La razón es que al momento de producir el giro, como el inductor no puede cambiar su valor de corriente instantáneamente, este deberá descargarse antes de volverse a cargar, por consiguiente, si la inductancia es mayor, este tiempo será mayor así que por lo tanto, la velocidad de rotación será menor. En la figura 4.2 se pueden observar las diferentes curvas velocidad-torque posibles del motor 86HS45.

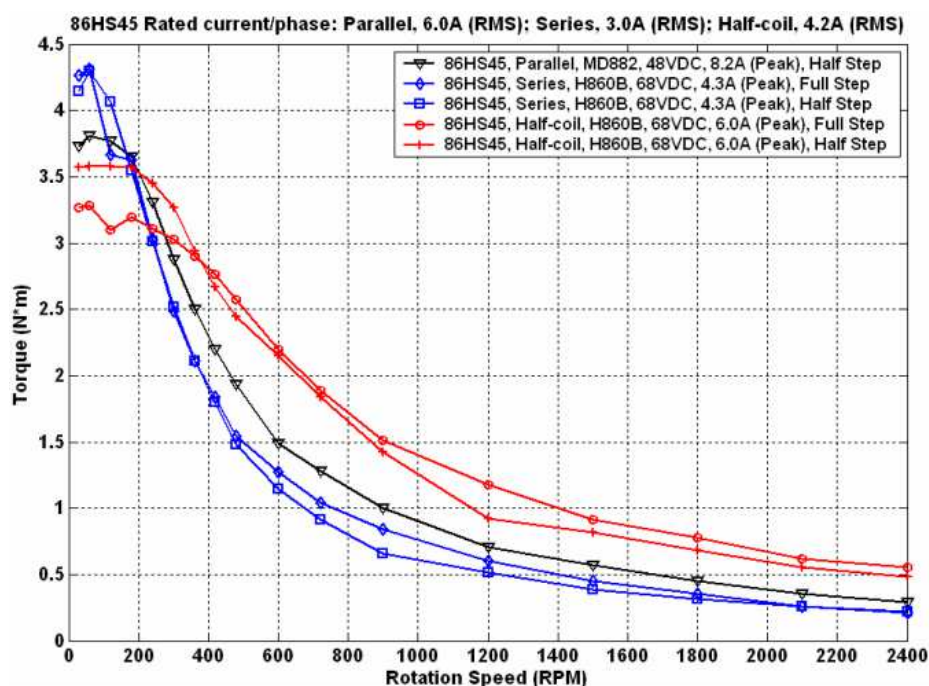


Figura 4.2: Curvas de velocidad-torque del motor

En la tabla 4.2 se muestran las características eléctricas del motor utilizado.

Modelo	Tipo de Conexionado	Corriente (A)	Resistencia (Ω)	Inductancia (mH)	
86HS45	Bipolar	Paralelo	6,0	$0,375 \pm 10\%$	$3,4 \pm 20\%$
		Series	3,0	$1,5 \pm 10\%$	$14 \pm 20\%$
	Unipolar	4,2	$0,75 \pm 10\%$	$3,4 \pm 20\%$	

Cuadro 4.2: Características eléctricas del motor 86HS45

En la tabla 4.3 se muestran las características mecánicas del motor utilizado.

Modelo	Torque de mantenimiento $Oz - in (N.m)$	Longitud $in (mm)$	Número de terminales	Inercia del rotor $Oz - in - s^2$ $(g.cm^2)$	Torque de retención $Oz - in (g.cm)$	Peso $Oz (kg)$
86HS45	637,2 (4,5)	3,13 (80)	8	0,01988 (1400)	18,2 (1300)	81,1 (2,3)
	637,5 (4,5)					
	453,12 (3,2)					

Cuadro 4.3: Características mecánicas del motor 86HS45

Dado que se requería el mayor torque posible, se optó por la configuración **Bipolar - Serie**

El motor utilizado, posee un torque máximo de $4,5Nm$. Debido a que se utilizó un engranaje sobre una cremallera, el diámetro primitivo del engranaje juega un papel fundamental a la hora de lograr una máxima transmisión de potencia.

$$P = M\omega \quad (4.6)$$

En donde P es la potencia [W], M es el par motor [Nm] y ω es la velocidad angular $[\frac{rad}{s}]$.

Expresando la potencia como el producto de la *fuerza tangencial* y la *velocidad tangencial*

$$F_t v_t = M\omega \quad (4.7)$$

Sabiendo que $v_t = \omega r$, siendo ω la velocidad angular y r el radio primitivo del piñón.

$$F_t v_t = M \frac{v_t}{r} \quad (4.8)$$

Obteniendo la expresión de la fuerza tangencial, la cual es inversamente proporcional al radio primitivo del piñón, implica que cuanto mas pequeño sea este, mayor fuerza tangencial se imprimirá sobre la cremallera. Esto puede observarse en la ecuación 4.9.

$$F_t = \frac{M}{r} \quad (4.9)$$

Reemplazando r por $30mm$ y $M = 4,5Nm$, se obtiene la expresión 4.10.

$$F_t = \frac{4,5Nm}{30 \times 10^{-3}m} = 150N. \quad (4.10)$$

Dado que el Φ_{int} de la camisa de acero inoxidable es $21,65mm$, en la expresión 4.11 se puede calcular la presión ejercida sobre la boquilla

$$P = \frac{F}{A} = \frac{150N}{\pi r^2} = 407460,5 \frac{N}{m^2} = 40746 \frac{\vec{kg}}{m^2}. \quad (4.11)$$

Aplicando el principio de Bernoulli y suponiendo que el polímero fundido se comporta como un fluido ideal y sin fricción, se podría hallar un valor aproximado de la velocidad de salida del material.

Capítulo 5

Resultados experimentales

Introducción

En este capítulo se presentarán los resultados experimentales de los ensayos realizados con el controlador PID implementado, previa presentación de la dinámica térmica del polímero, la cual es necesaria para comprender la forma de la referencia de temperatura implementada.

5.1. Proceso de fusión del termoplástico

En la fig. 5.1 se presenta un termograma DSC¹ de una muestra de polietileno de baja densidad *LDPE*² *PG 7008*. La información provista por este ensayo permitió comprender el comportamiento del polímero en función de la temperatura.

Un polímero semicristalino posee una fracción de sus macromoléculas ordenada (fase cristalina) y una fracción desordenada, sin orden o estructura repetitiva (fracción amorfa). Durante el proceso de fusión, la fase cristalina absorbe calor lo cual se manifiesta como un pico endotérmico en el termograma de DSC de la figura 5.1. El área bajo la curva se relaciona con la entalpía de fusión ΔH_m o el calor absorbido por el material a presión constante. Se consideró como temperatura de fusión, T_m al valor correspondiente al mínimo en el termograma.

¹Differential Scanning Calorimetry

²Low Density Polyethylene

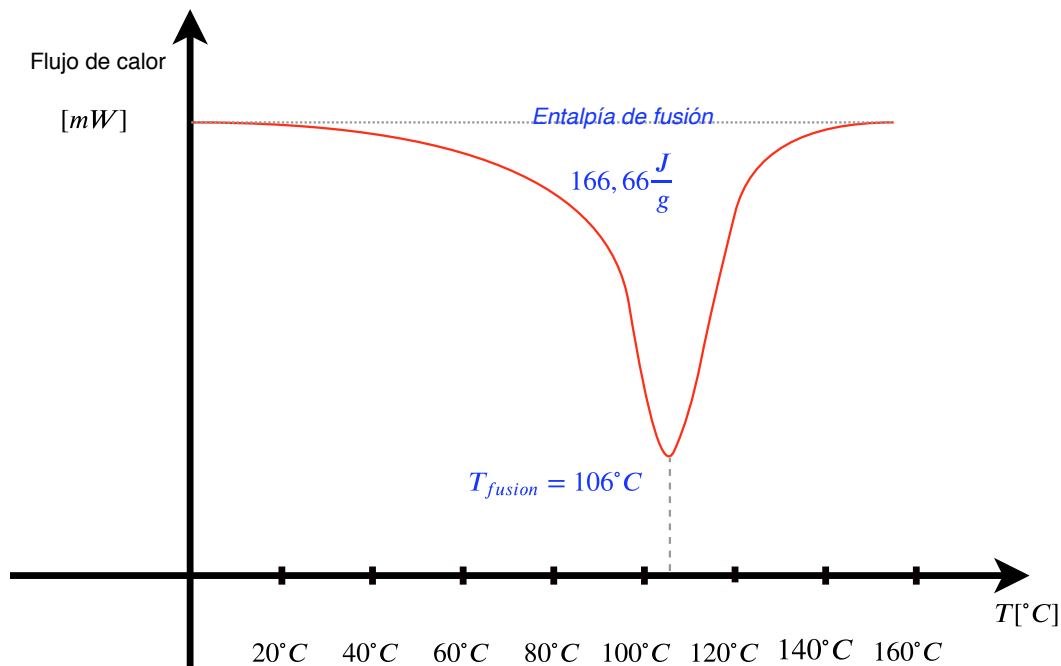


Figura 5.1: Termograma DSC de una muestra de LDPE PG 7008

El termograma se registró empleando una velocidad de calentamiento de $10 \frac{^{\circ}C}{min}$ en el rango de temperatura mostrado. La temperatura de fusión del LDPE resultó de $106^{\circ}C$ con una entalpía de fusión de $166,66 \frac{J}{g}$.

En la calorimetría diferencial de barrido se emplea una muestra y una referencia que deben mantenerse a la misma temperatura durante la etapa de calentamiento y el calor adicional que debe aportarse a la muestra (o a la referencia, si es que el cambio es exotérmico) es el valor medido. De esta manera los cambios de entalpía pueden ser medidos directamente.

El área de los picos es una medida de la energía requerida para mantener la referencia y la muestra a igual temperatura.

Si una muestra y un material de referencia se calientan a una velocidad conocida en un ambiente controlado, el incremento de la temperatura de la muestra y de la referencia será aproximadamente el mismo, dependiendo de las diferencias de calor específico. En consecuencia, si ocurre un cambio de fase, la temperatura de la muestra puede aumentar (proceso exotérmico) o bien disminuir (endotérmico) con respecto a la temperatura de la referencia. Esta diferencia debida al cambio de calor está relacionada directamente con el flujo de calor diferencial necesario para mantener las temperaturas iguales.

5.2. Controlador PID

En el ensayo de la figura 5.2 pueden observarse dos gráficos. Uno de ellos temperatura en función del tiempo y el otro, potencia en función del tiempo. En el caso de la

temperatura, se representa la referencia y la temperatura de pistón y de boquilla (puede observarse como la temperatura de boquilla, tiende a igualar la temperatura del pistón luego de un cierto tiempo). En el caso del gráfico de potencia, puede observarse la manera como el sistema entrega potencia y luego mantiene la misma constante cuando se alcanzó la temperatura de referencia, compensando así mayormente las pérdidas del sistema al ambiente.

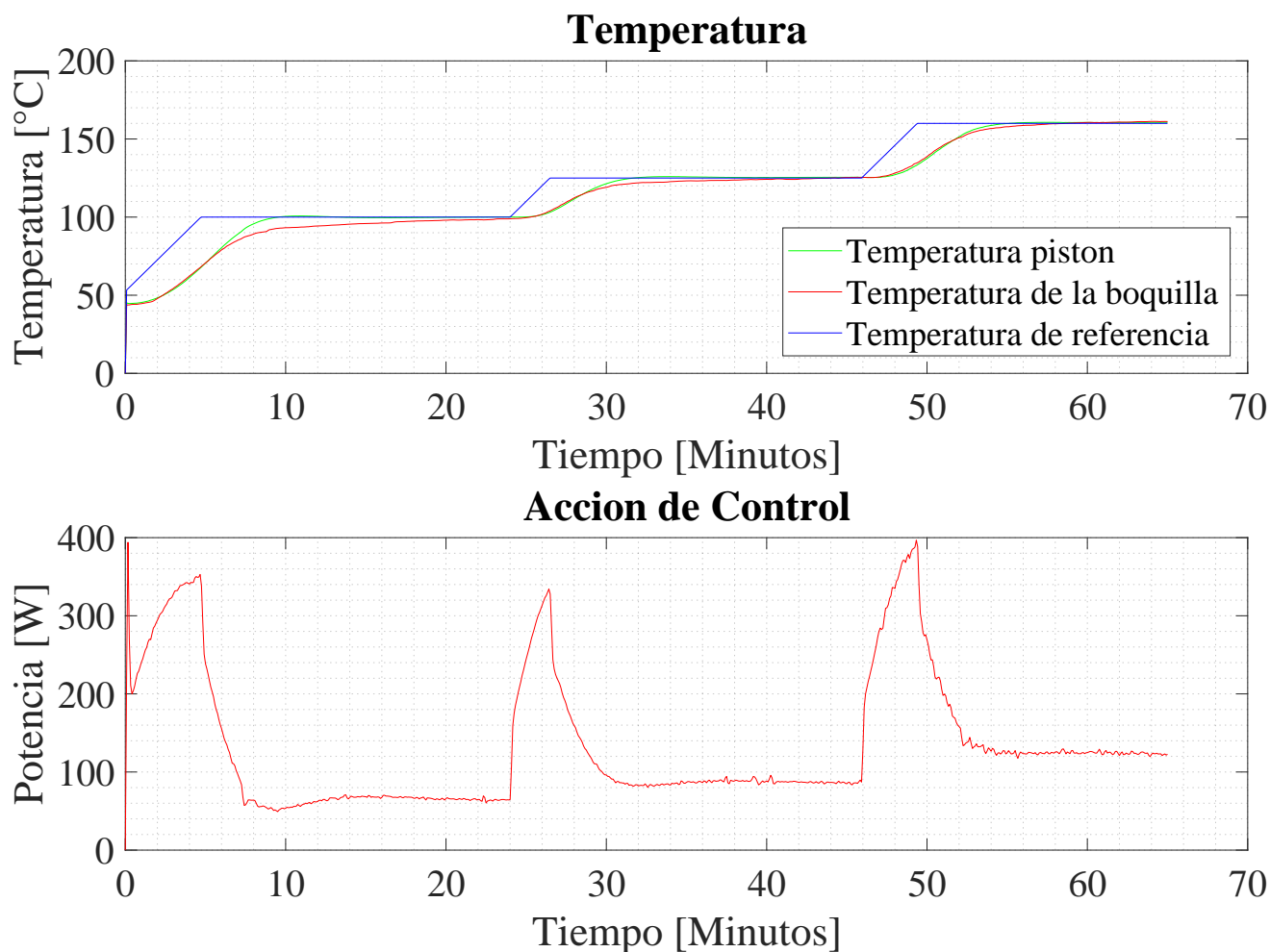


Figura 5.2: Temperaturas de referencia, del pistón y de la boquilla en un ensayo de calentamiento

A continuación, en las figuras 5.3, 5.4 y 5.5 puede observarse un zoom de la temperatura medida y la temperatura de referencia en función del tiempo para $T_{ref} = 100^{\circ}C$, $T_{ref} = 125^{\circ}C$ y $T_{ref} = 160^{\circ}C$ respectivamente. Se evidenció la no linealidad del sistema, al hallar valores porcentuales de sobrepico diferentes para cada T_{ref} .

$$Sobrepico|_{T_{ref}=100^{\circ}C} = \frac{100,7^{\circ}C - 100^{\circ}C}{100^{\circ}C} 100 \% = 0,7 \% \quad (5.1)$$

$$Sobrepico|_{T_{ref}=125^{\circ}C} = \frac{125,9^{\circ}C - 125^{\circ}C}{125^{\circ}C} 100 \% = 0,72 \% \quad (5.2)$$

$$Sobrepico|_{T_{ref}=160^{\circ}C} = \frac{160,6^{\circ}C - 160^{\circ}C}{160^{\circ}C} 100 \% = 0,37 \% \quad (5.3)$$

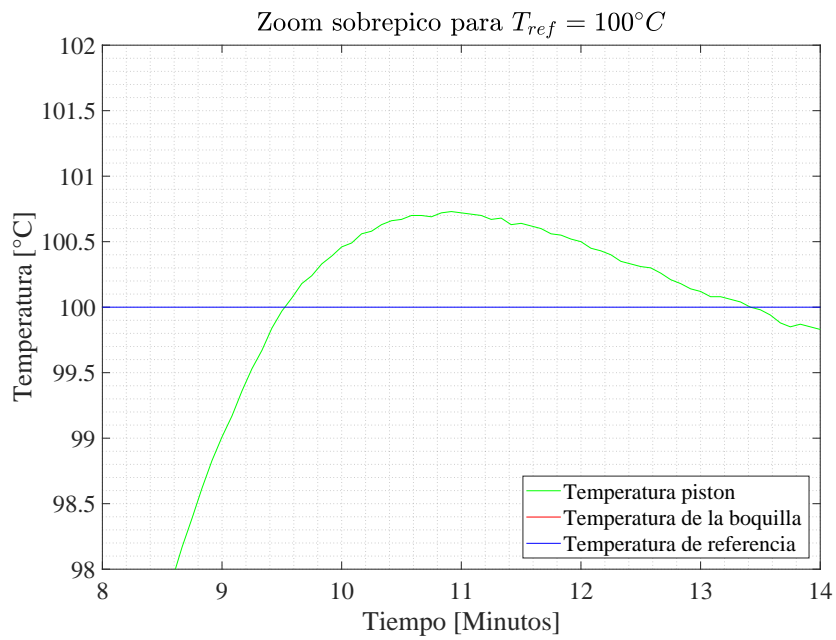


Figura 5.3: Zoom sobrepico de temperatura para $T_{ref} = 100^{\circ}C$

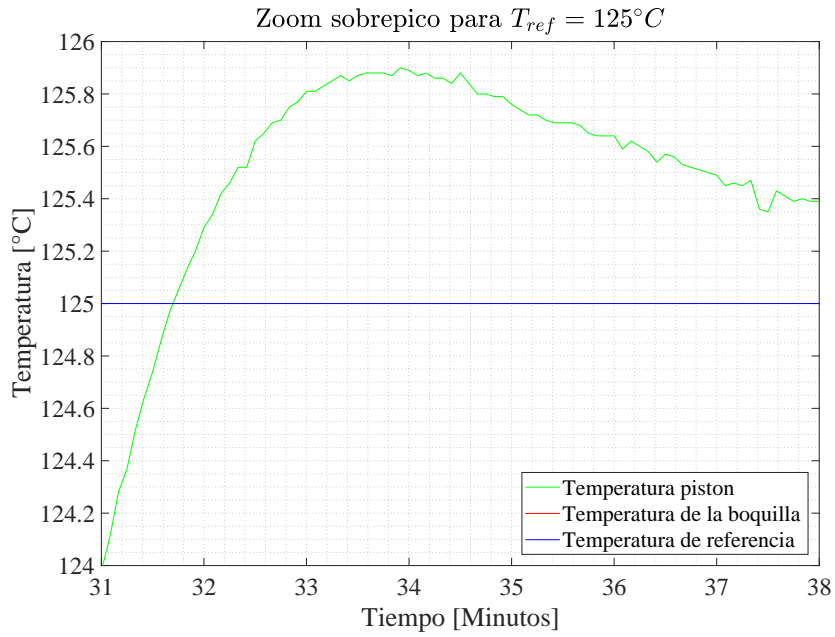


Figura 5.4: Zoom sobrepico de temperatura para $T_{ref} = 125^{\circ}C$

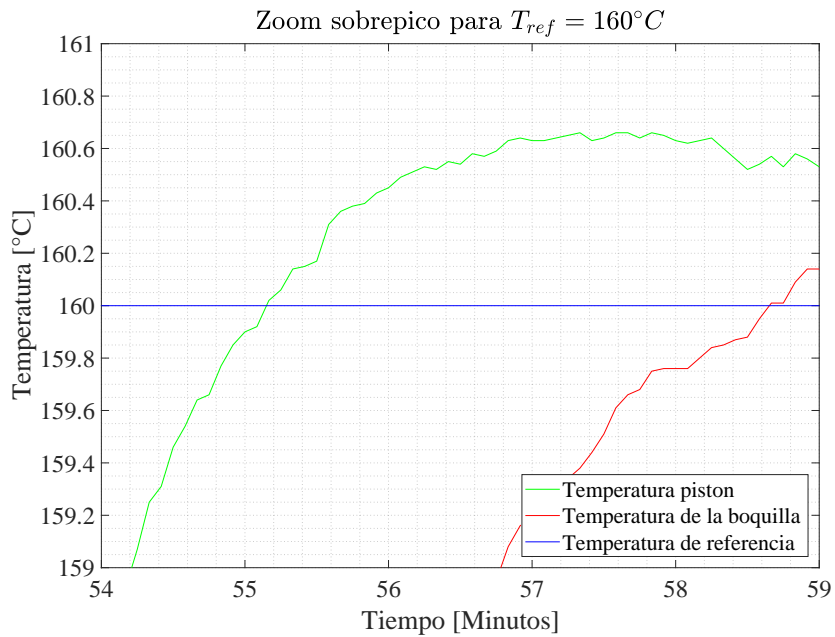


Figura 5.5: Zoom sobrepico de temperatura para $T_{ref} = 160^{\circ}C$

Respecto a las especificaciones del control, los valores de sobrepico hallados en las ecuaciones 5.1, 5.2 y 5.3 fueron menores al 1%, por lo tanto puede considerarse como una especificación cumplida.

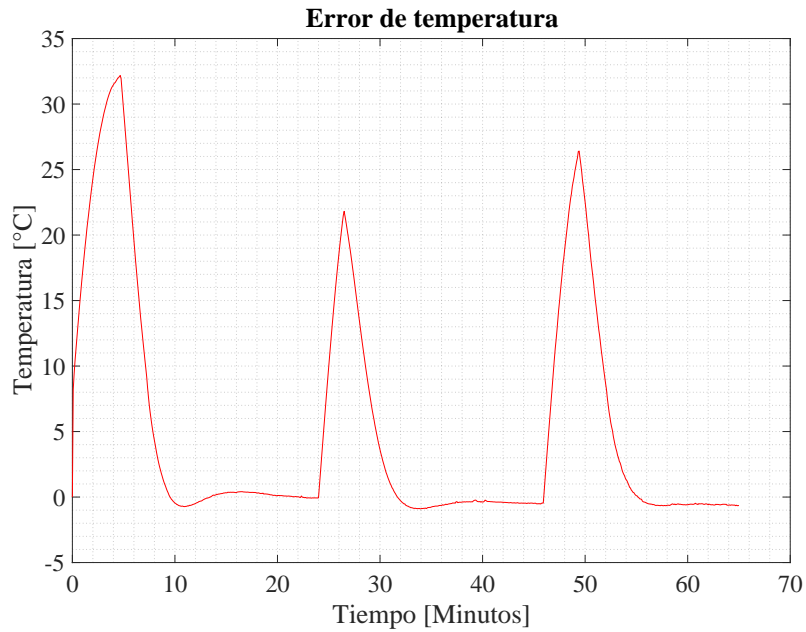


Figura 5.6: Error de temperatura con el controlador PID implementado

El error obtenido tiende a cero para las tres temperaturas de referencia.

En la figura 5.7, puede observarse la salida del bloque integrador y cómo este no diverge, tal como se explicó en la sección 2.2.5.3.

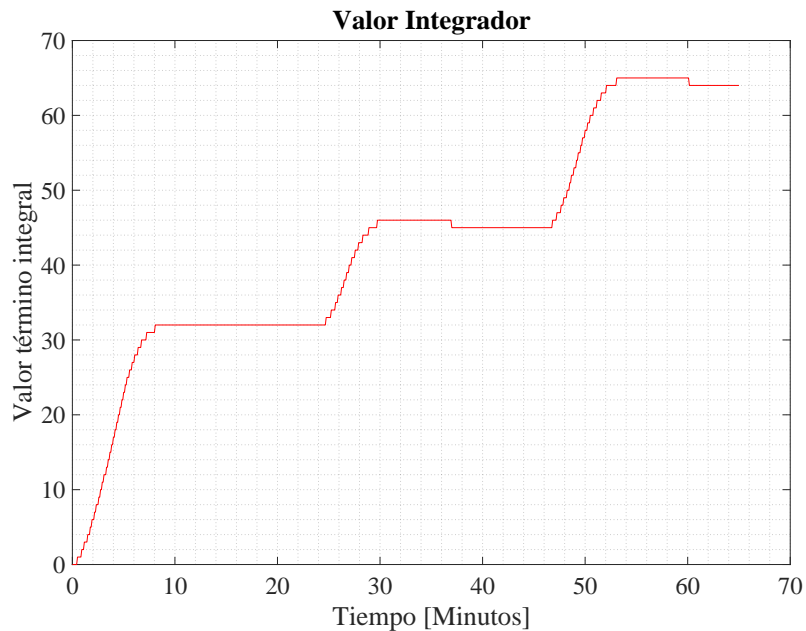


Figura 5.7: Valor del integrador en el controlador PID implementado

En la figura 5.8 se pueden observar 2 muestras de filamento obtenido compuesto de

LDPE 7008 PG. El filamento superior fue obtenido sin estiramiento. En cambio el inferior, fue obtenido estirando el material a medida que salía de la boquilla y bobinándose sobre un carrete (procedimiento manual).

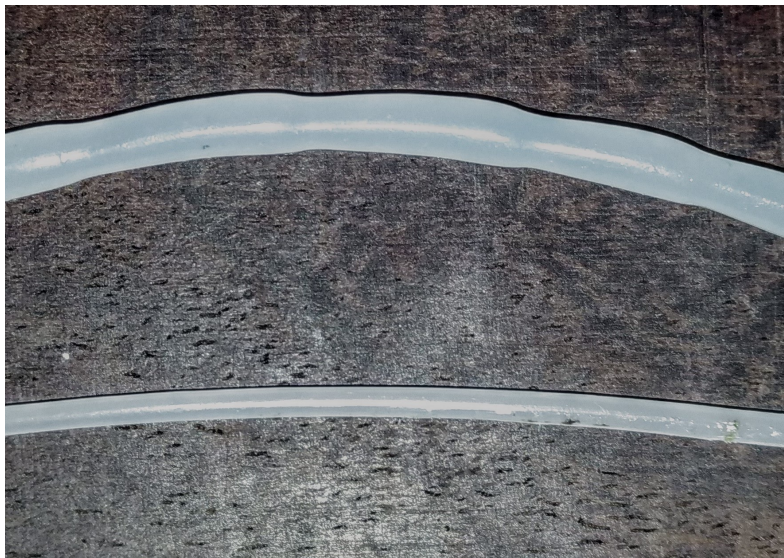


Figura 5.8: Filamentos obtenidos con y sin control de estiramiento

Capítulo 6

Aspectos a mejorar

6.1. Respecto al diseño del PCB

Dado que existen en el PCB 2 canales para la medición de temperatura y uno de ellos se encuentra más cerca del ADC que el otro, el más alejado, se comportaba como una antena al tener un camino más extenso, siendo afectado por el ruido. Debido a esto es que se implementó un filtro digital con $N = 5000$. Una posible mejora para un diseño a futuro puede ser mejorar el layout y que los planos de masa tenga diversas vias para permitir su conexión con las otras capas del circuito, disminuyendo el camino de retorno de la señal.

6.2. Respecto a la caracterización del sistema

Inicialmente, el sistema fue caracterizado mediante un escalón de potencia, tal como se explicó en la sección 2.2.4. A partir del modelo obtenido, se realizaron diversas simulaciones con MatLAB. Luego, al realizar los ensayos experimentales, las respuestas obtenidas no coincidían completamente con las simulaciones obtenidas previas. Teniendo en cuenta esto, se caracterizó el sistema con la señal de potencia obtenida experimentalmente y la respuesta del modelo simulado, se aproximó a la respuesta del modelo real en ciertos intervalos. La razón de esto fue debida a que el sistema es alineal y su comportamiento es diferente para distintos rangos de temperatura como así también a que el modelo obtenido fue de segundo orden cuando el sistema real es mucho más complejo. Una posible mejora puede introducirse si se caracteriza al sistema con una señal variable y no con un escalón. Puede atacarse también al sistema con una onda senoidal, sabiendo que el sistema es pasa bajos, este tenderá a la temperatura determinada por el valor medio de la potencia senoidal de entrada, pero esa variación rápida excitará ciertos polos que con un escalón no se ven afectados.

6.3. Respecto al calentamiento

Debido a que las resistencias fueron donadas al proyecto, no se realizó un análisis respecto a que otros métodos de calentamiento podrían haberse utilizado. Dado que las

resistencias tienen forma de omega, inicialmente la medición de temperatura se realizó en el centro de las mismas, sobre el aluminio. Esto se debió a que en el punto de medición, la temperatura no era uniforme y era diferente a la del interior de la camisa, lugar en donde el polímero iba a fundirse. Una mejora puede ser la utilización de resistencias tipo cartucho, insertas en la camisa de aluminio permitiendo generar un flujo de calor uniforme y evitando un gradiente de temperatura excesivo entre el cuerpo y la boquilla de la extrusora.

6.4. Respecto a la adquisición

Inicialmente, las entradas del multiplexor no tenían enclavadores para fijar los valores máximos y mínimos, por lo que en ciertas pruebas preliminares durante el diseño del control, el sobrepico sobrepasó la tensión máxima admisible $V_{max} = V_{CC} + 0,5$ quemando la entrada del mismo, haciendo este inutilizable.

6.5. Respecto al motor y a la transmisión de potencia

Debido a que el proyecto fue realizado en diferentes etapas, algunas de ellas no se realizaron en un orden preestablecido. Un ejemplo de ello fue que se adquirió el motor previo al diseño del equipo. En este caso, se realizó un diseño para maximizar el torque útil del motor, debido a que este era un factor requerido para la extrusión del polímero. Un motor con un torque mayor, podría haber servido para acelerar el proceso de extrusión. Otra opción podría haber sido la utilización de un engranaje más pequeño, pero la limitación en este caso la imponía la fresa módulo $N^{\circ}2$ cuyo rango era de 14 a 20 dientes.

A partir de lo presentado en la sección 4.3, el procedimiento para el cálculo del radio primitivo del piñón debería haber sido de la siguiente manera. A partir de la ecuación de Bernoulli e imponiendo una velocidad de salida del material, se calcula la presión necesaria. Desafectando ese factor por el área transversal interna de la camisa de acero inoxidable, se obtiene la fuerza. Por último, con esa fuerza y el torque máximo disponible del motor a la velocidad de rotación deseada, se calcula el radio primitivo del engranaje.

6.6. Respecto al control del calibre del filamento

Inicialmente se pensaba que al salir el filamento por una boquilla cuyo $\phi = 1,75mm$, este tendría el mismo diámetro. Esta suposición fue incorrecta debido a que el material se expande al contacto con el aire. Resta implementar en una etapa futura un sensado del diámetro y un control sobre la bobinadora para lograr el calibre requerido.

Capítulo 7

Conclusiones

En el presente trabajo se presentó el desarrollo de un prototipo funcional de extrusora a pistón para polímeros biomédicos. El desarrollo de un equipo de estas características como requisito para optar por el grado de ingeniero electrónico, fue todo un desafío particularmente desde el hecho que no fue un desarrollo puramente electrónico. El diseño mecánico del equipo jugó un rol preponderante debido a que comercialmente no existen extrusoras a pistón de este tamaño. El estudio de diferentes fenómenos, observados en los ensayos, permitieron adquirir conocimientos en diversas áreas concernientes al proyecto tales como el comportamiento de los polímeros termoplásticos en función de la temperatura, la transferencia de calor por diferentes fenómenos, dilatación térmica de los materiales, ruido sobre señales adquiridas, caracterización de sistemas térmicos no lineales, implementación de un control PID, entre otras. Con el desarrollo del informe se pretendió volcar toda la información necesaria para que a futuro pueda mejorarse el diseño del equipo sirviendo este como prototipo inicial.

Apéndice A

Planos mecánicos

En este anexo se presentarán los planos principales de cada parte del equipo.

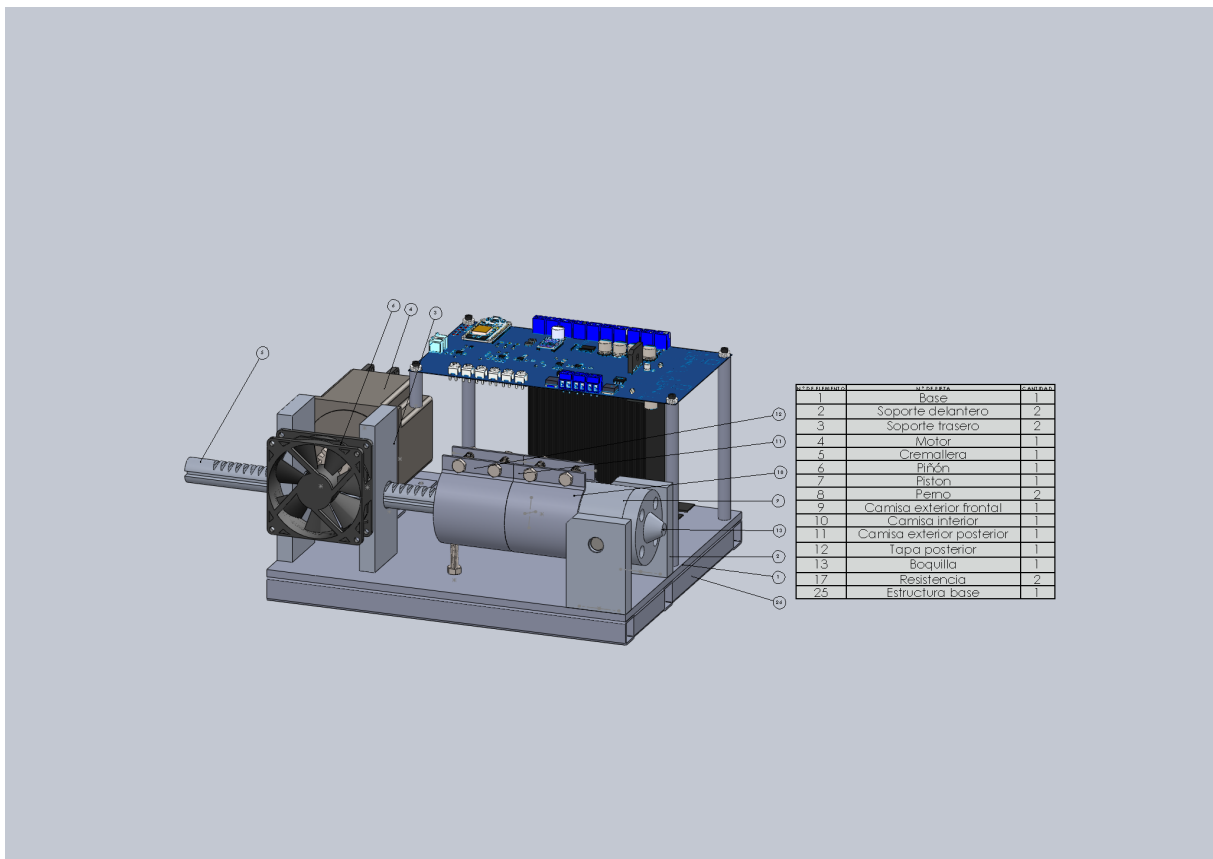


Figura A.1: Ensamblaje con lista de componentes

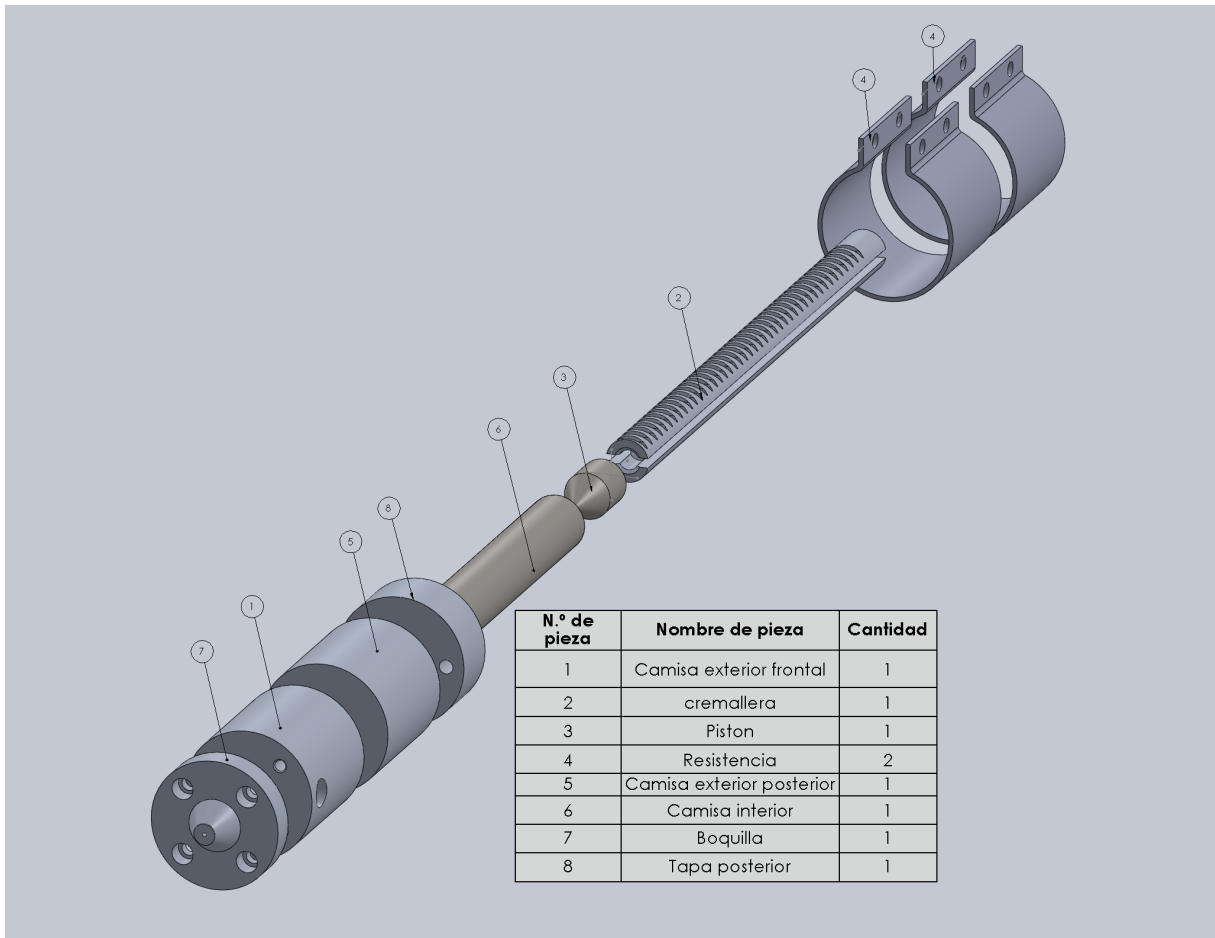


Figura A.2: Vista en explosión del sistema modelado

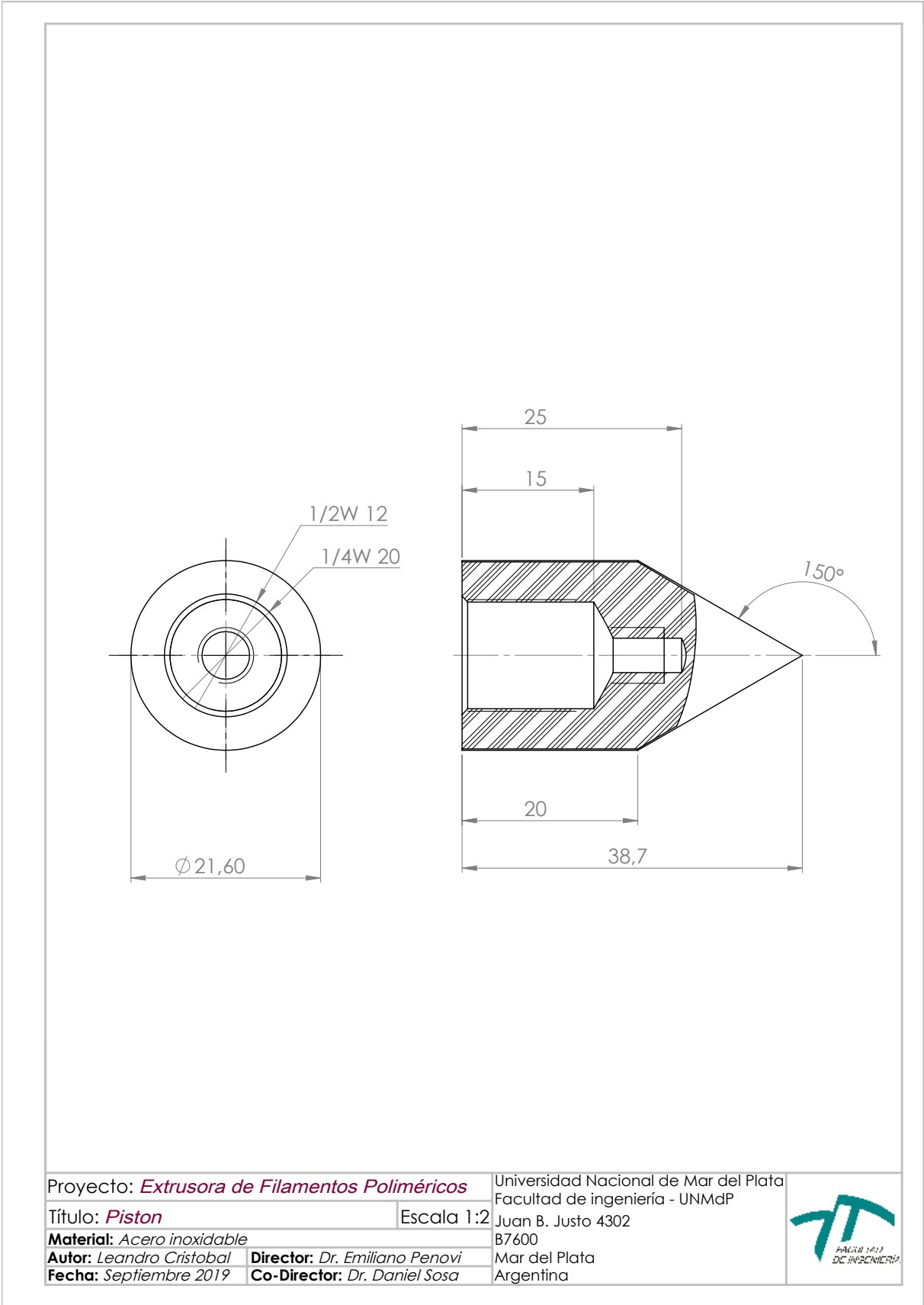


Figura A.3: Plano del pistón

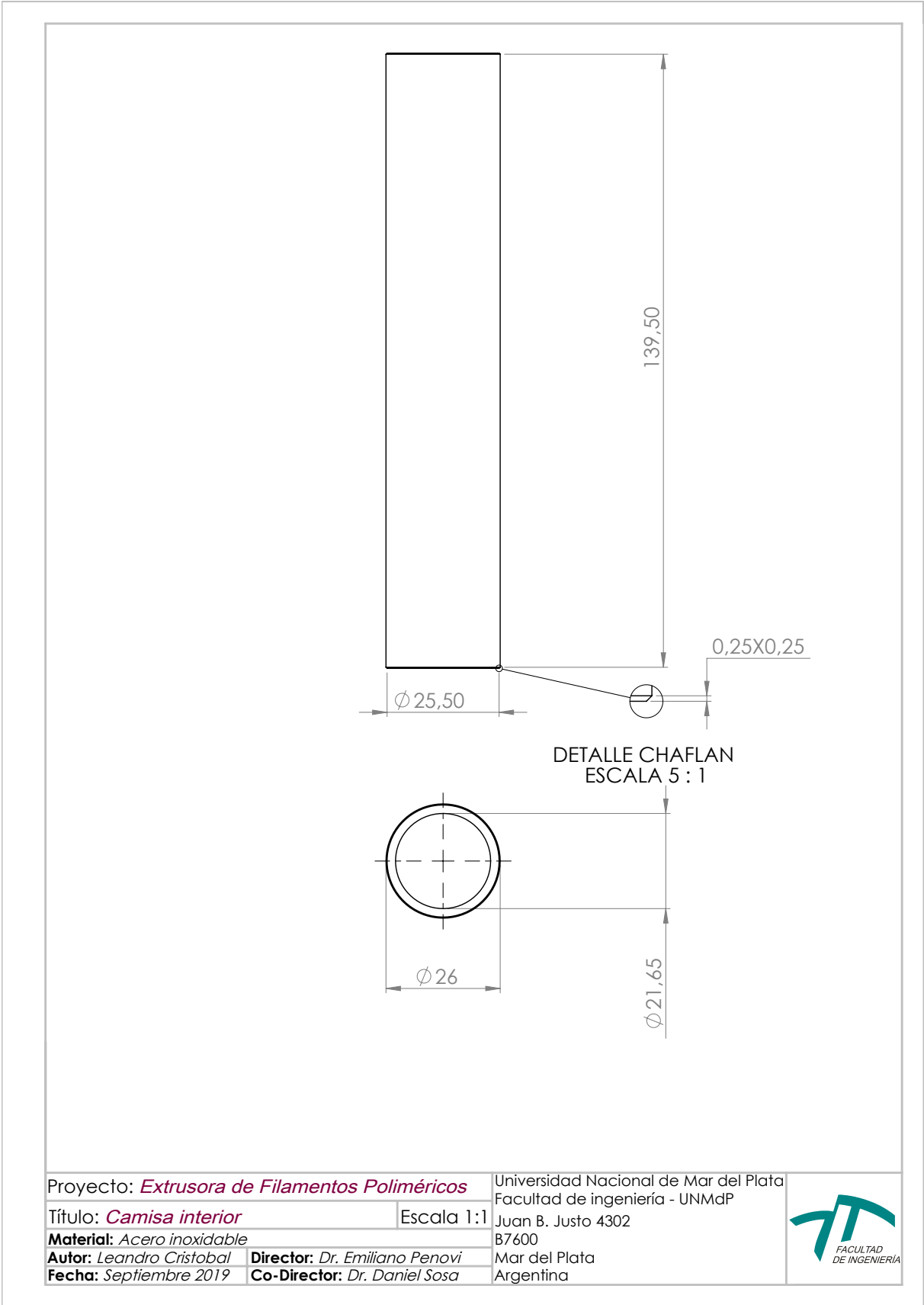


Figura A.4: Plano de la camisa interior

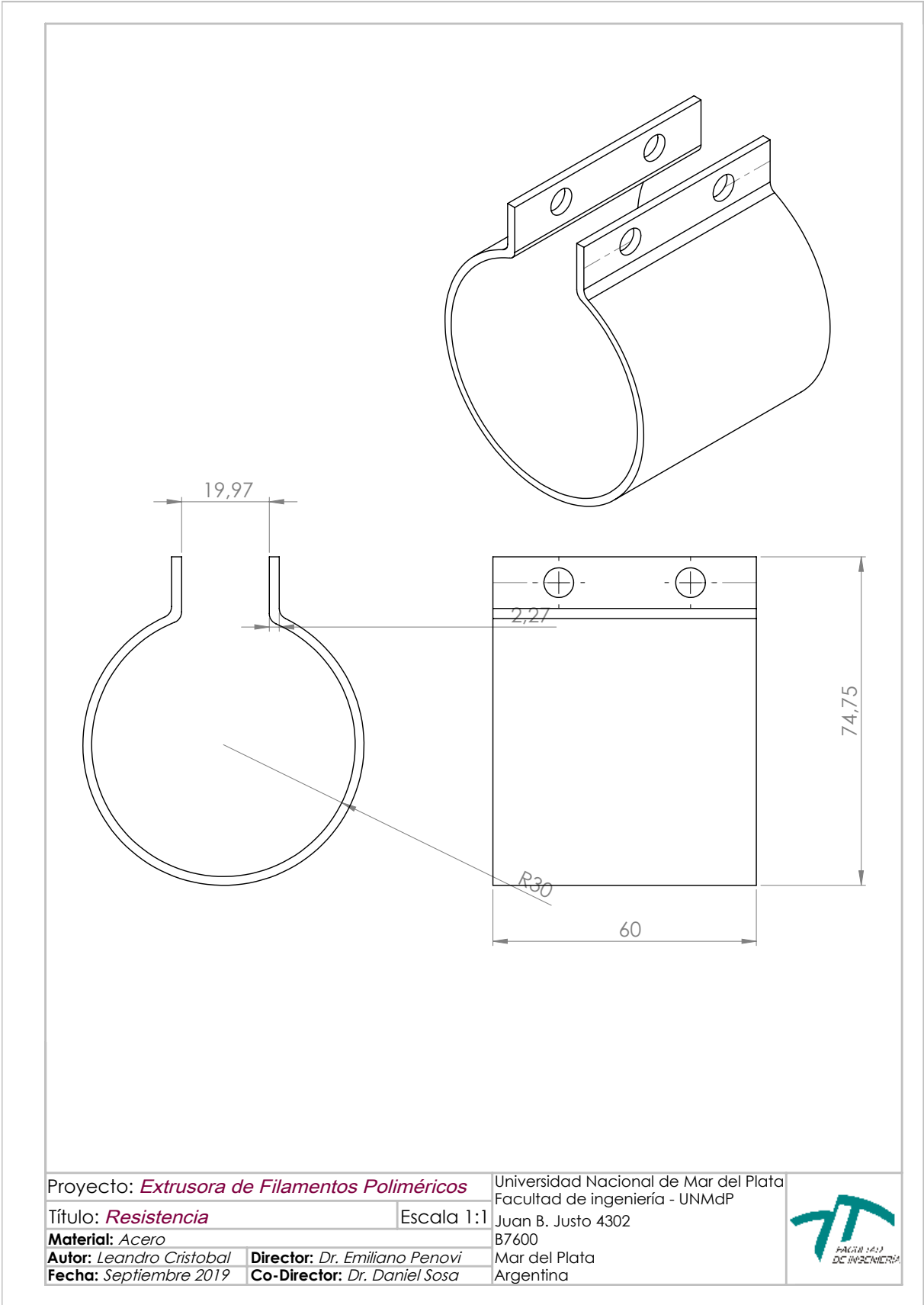


Figura A.5: Plano de la resistencia

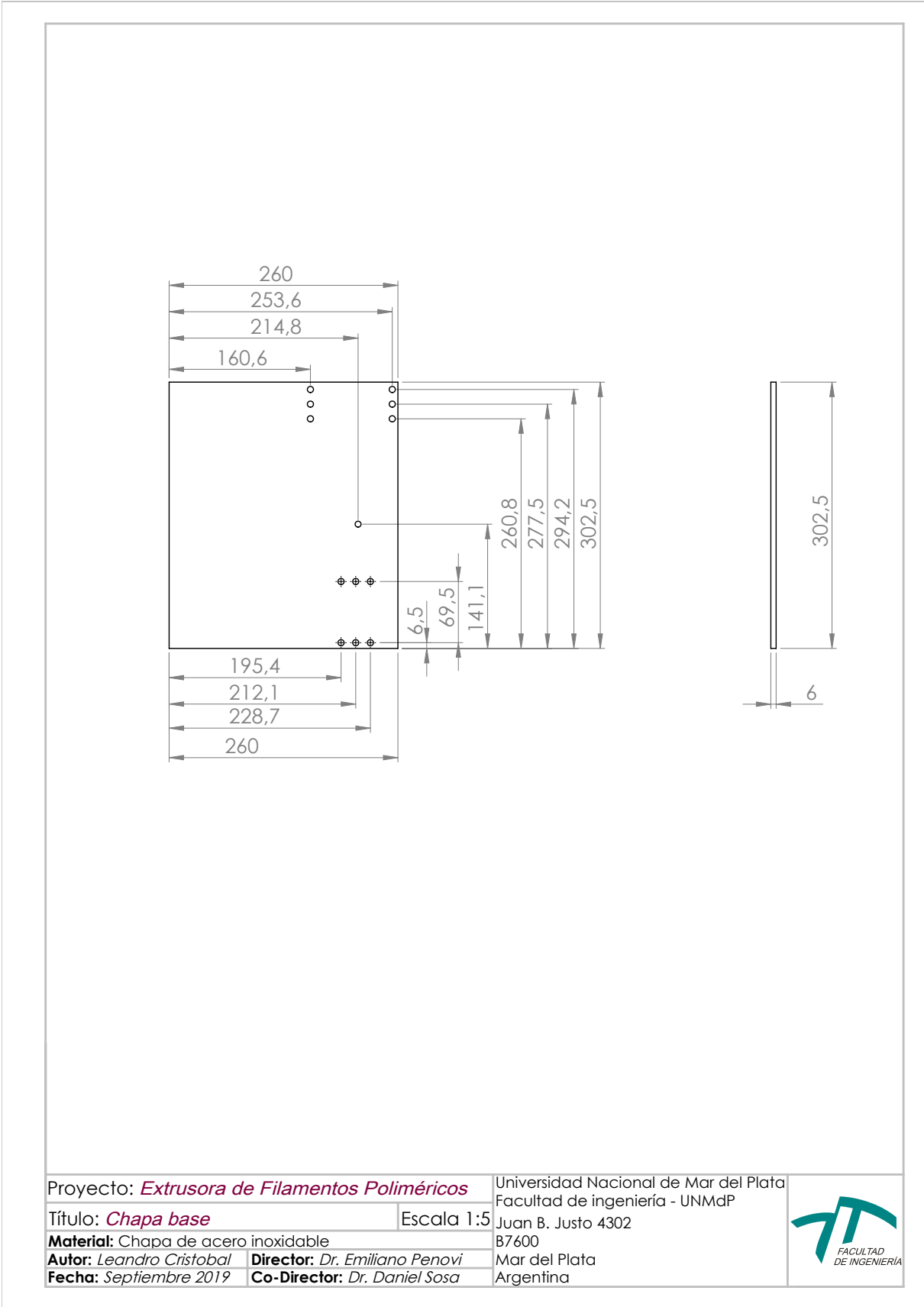


Figura A.6: Plano de la base

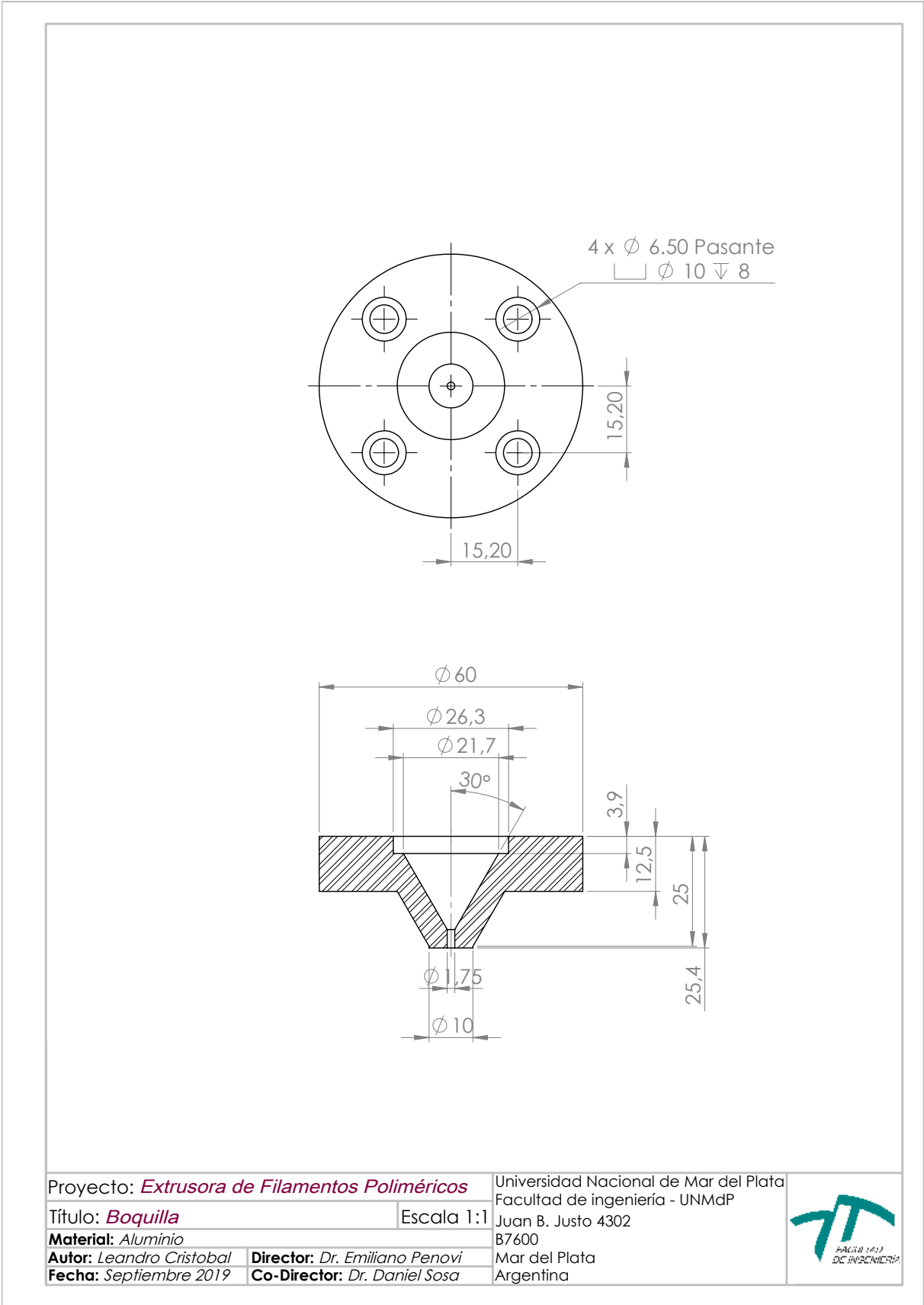


Figura A.7: Plano de la boquilla

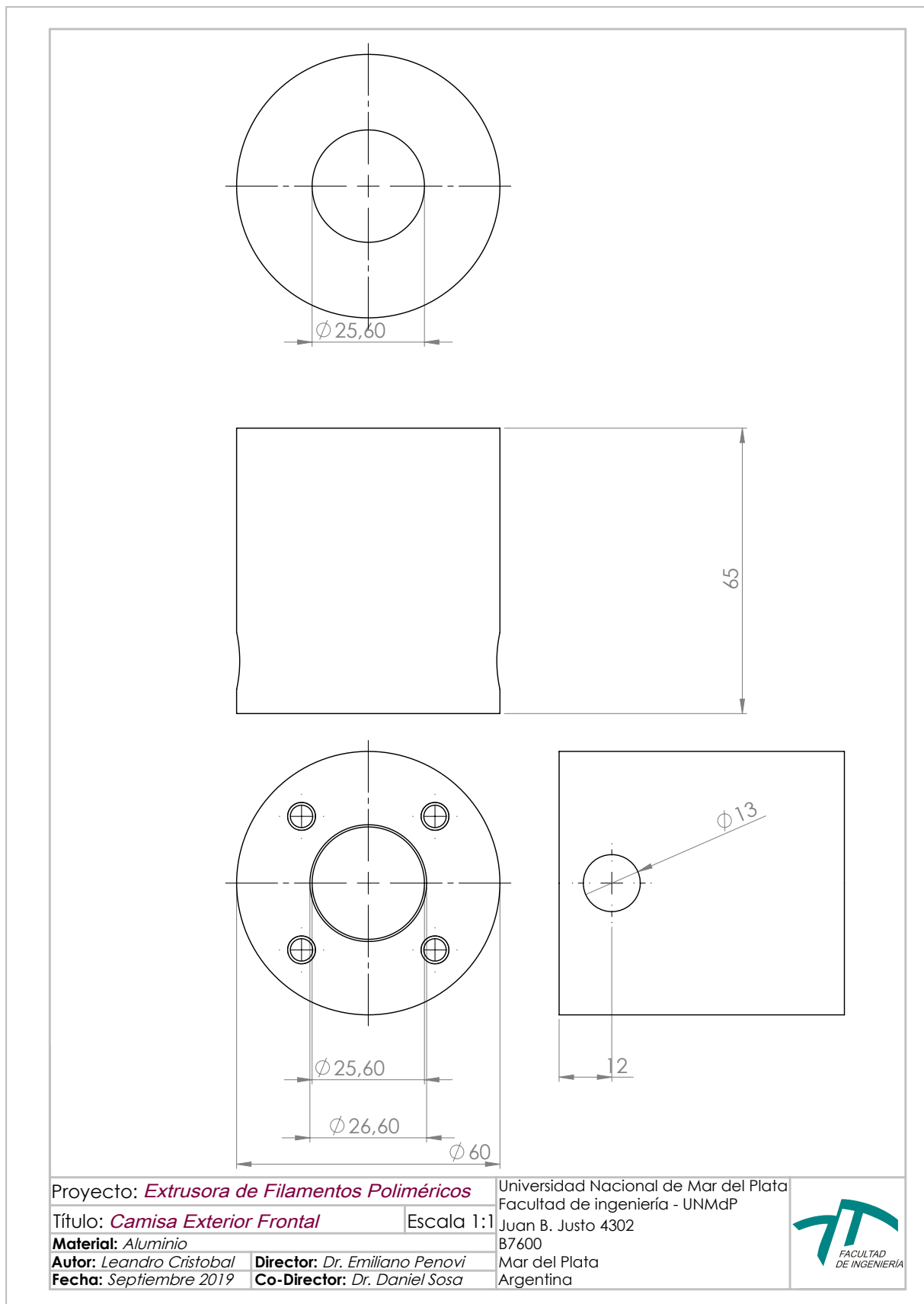


Figura A.8: Plano de la camisa exterior (delantera)

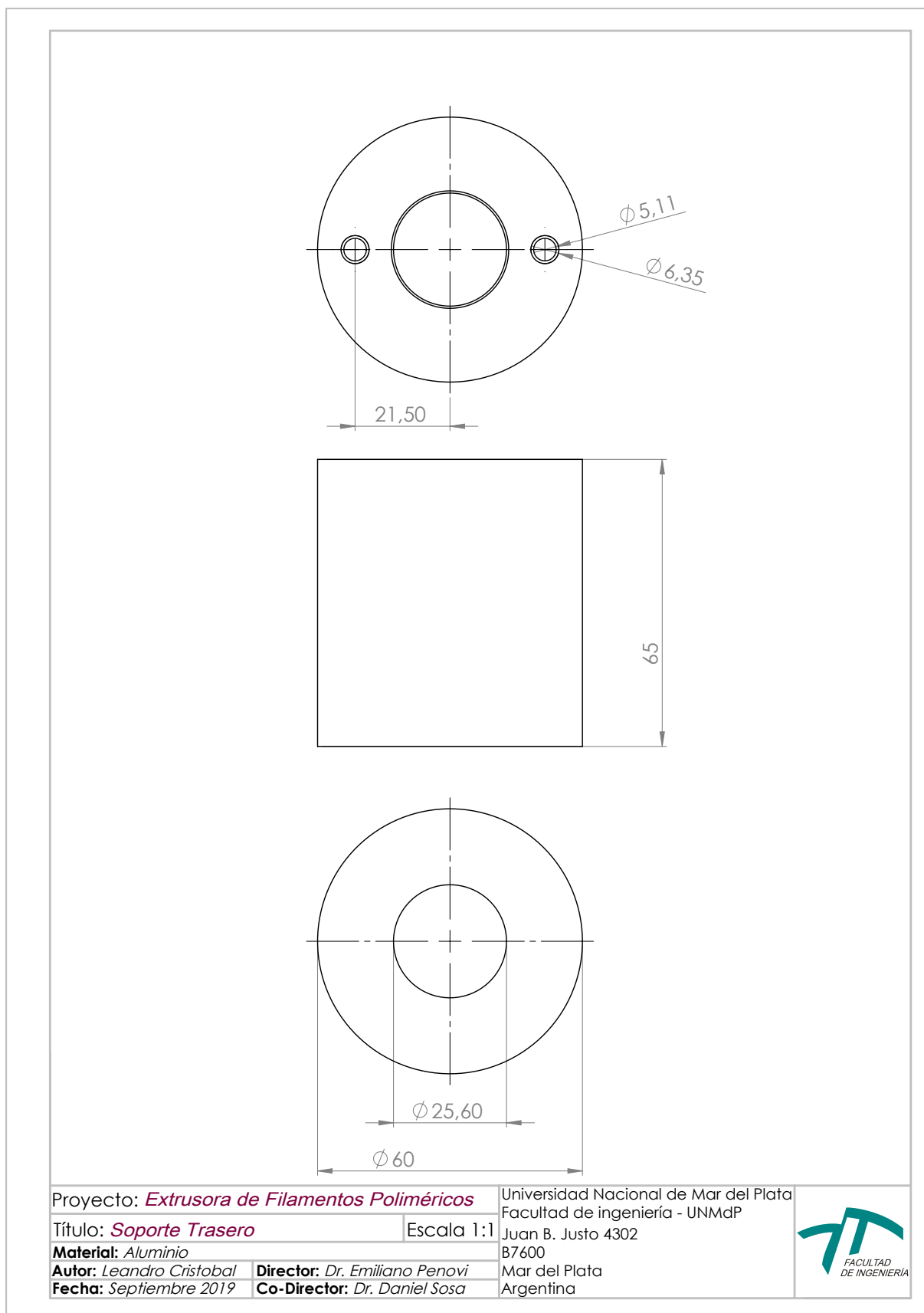


Figura A.9: Plano de la camisa exterior (trasera)

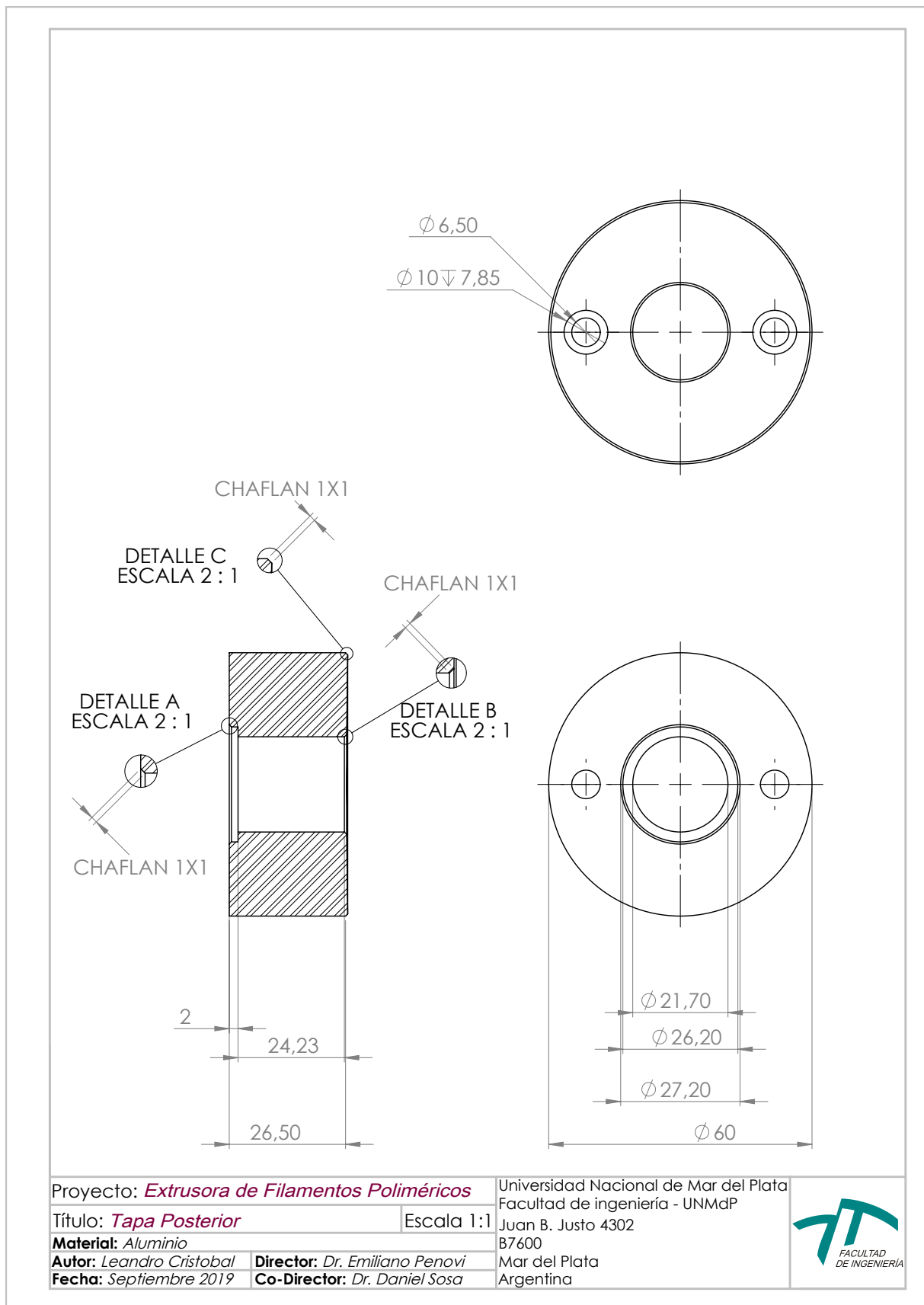


Figura A.10: Plano de la tapa trasera

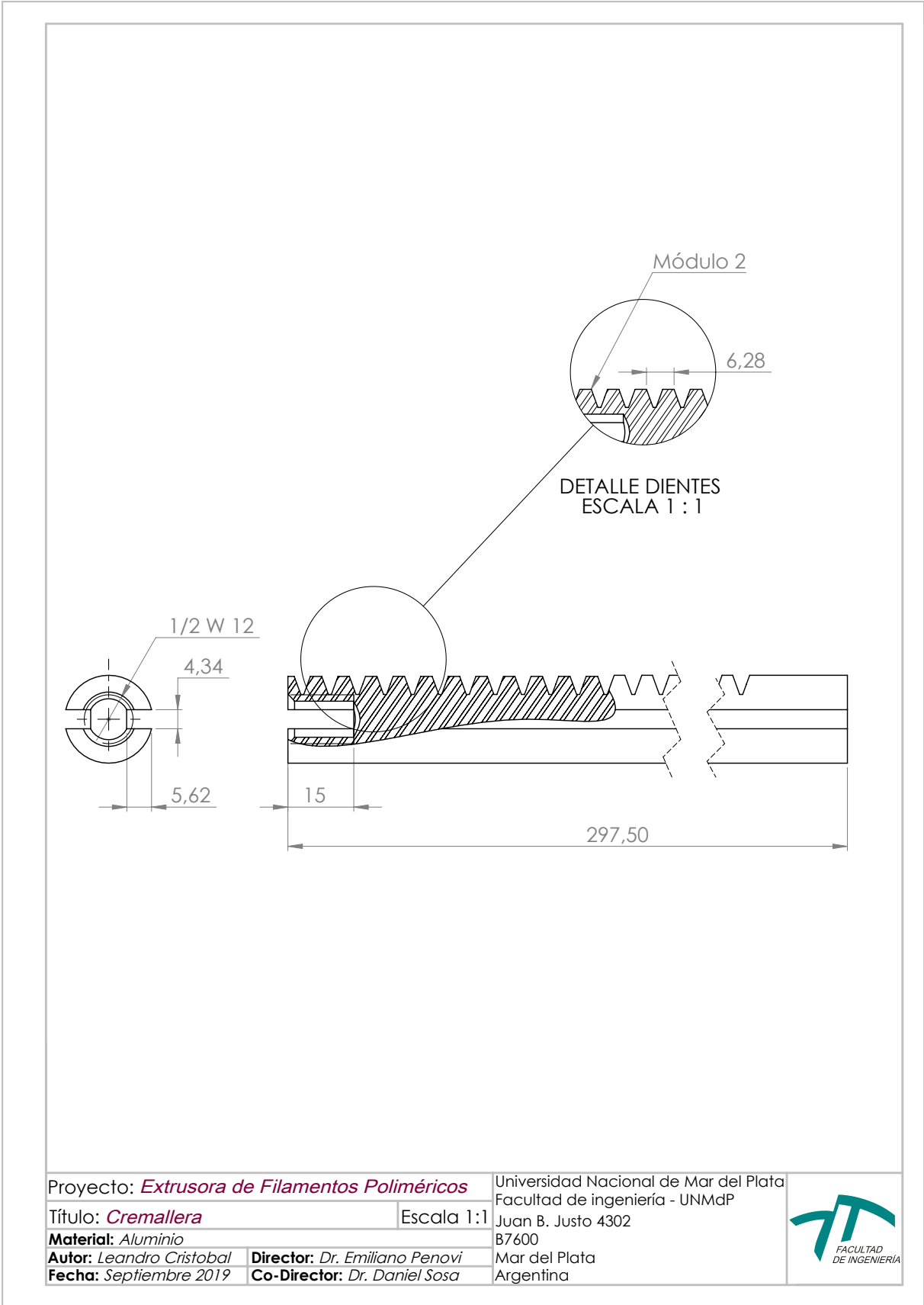


Figura A.11: Plano de la cremallera

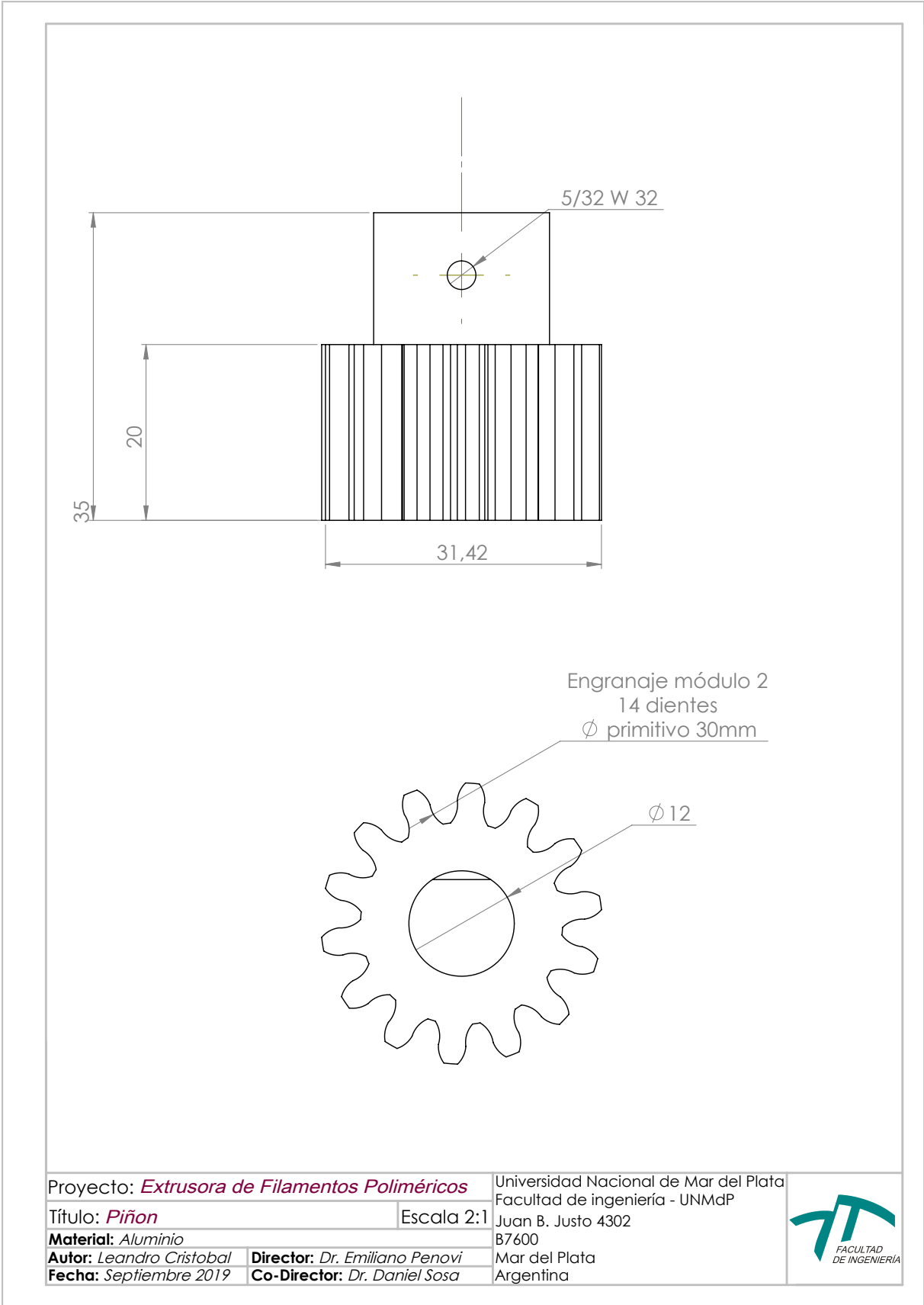


Figura A.12: Plano del piñón

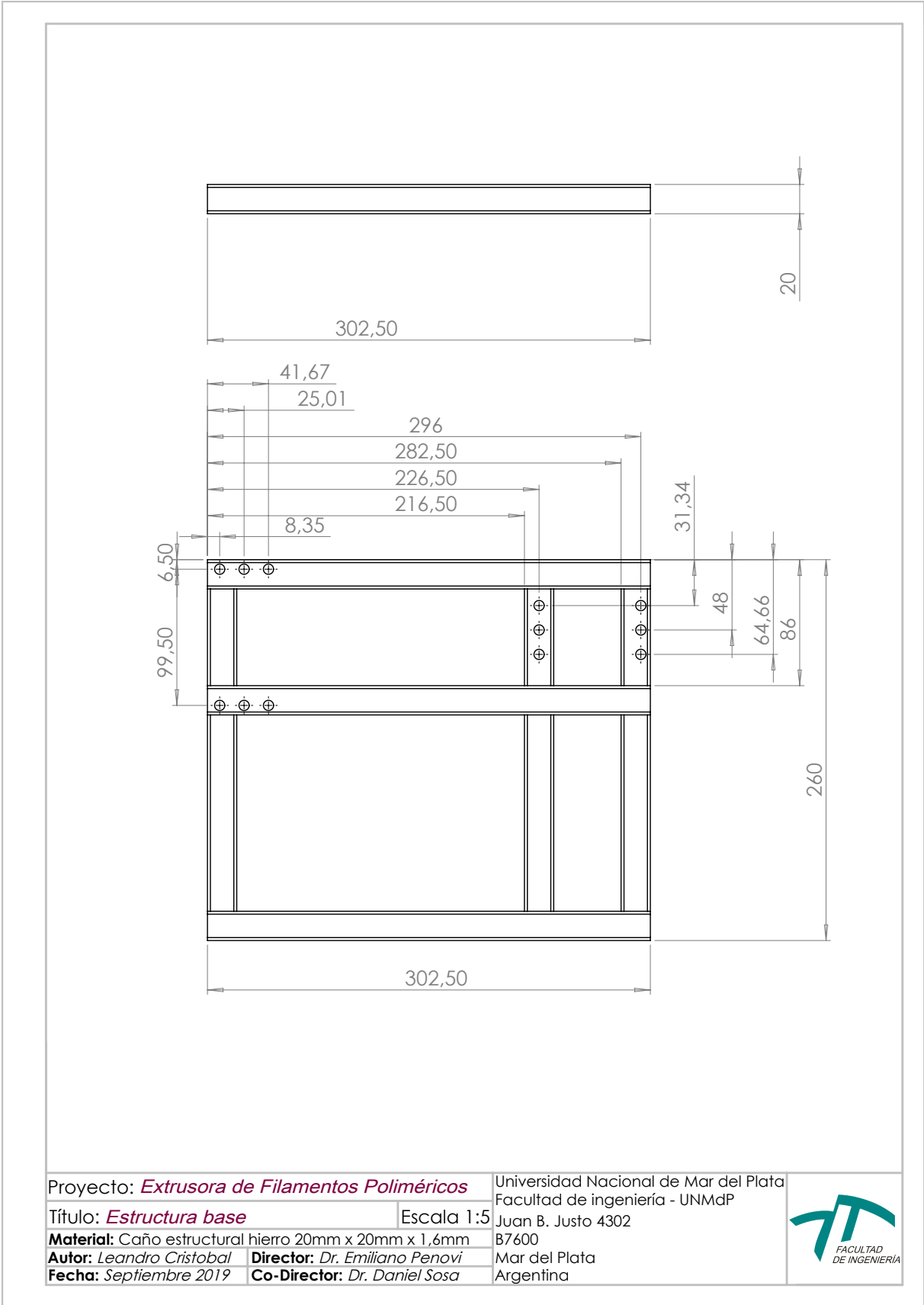


Figura A.13: Plano de la base (estructura)

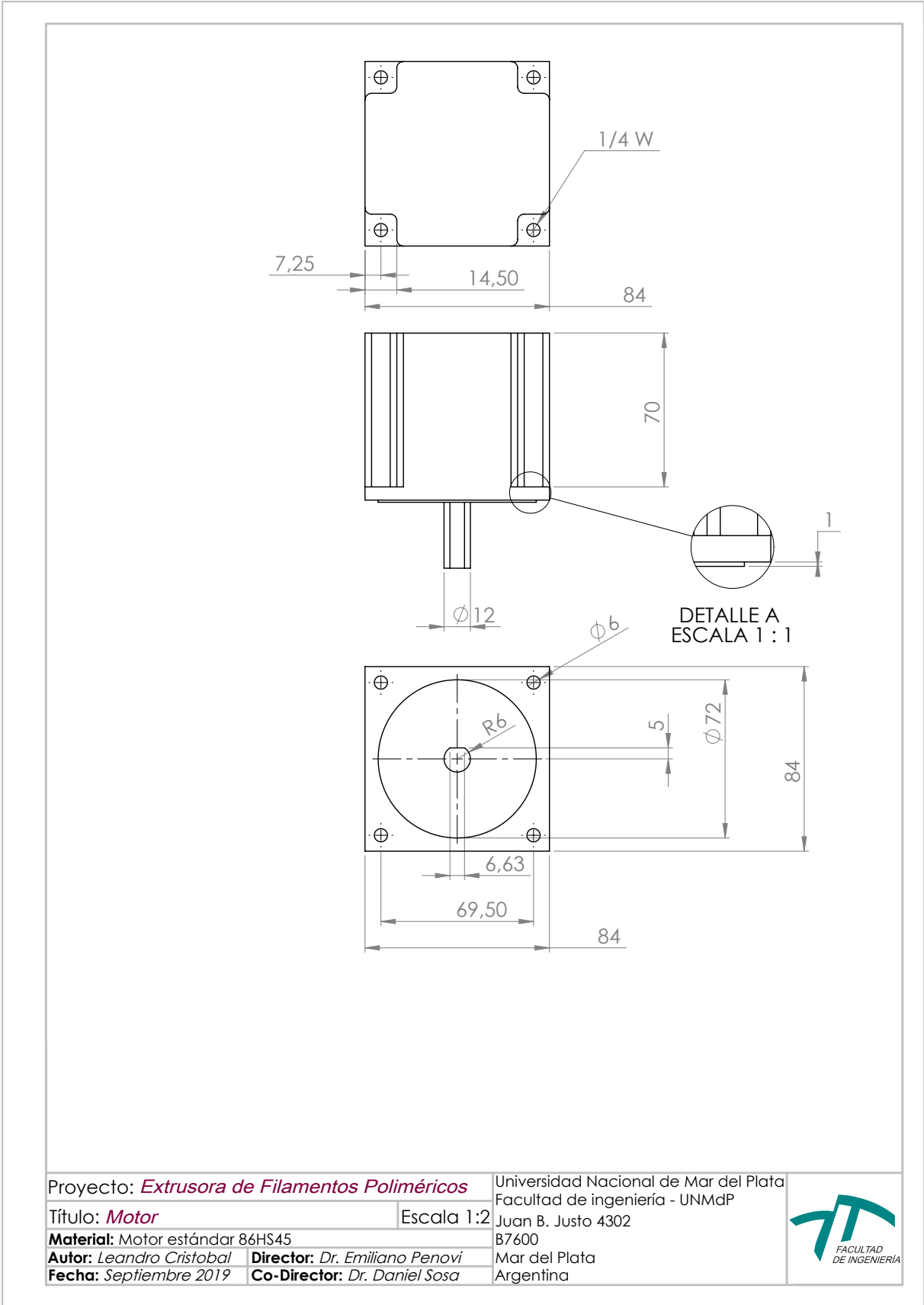
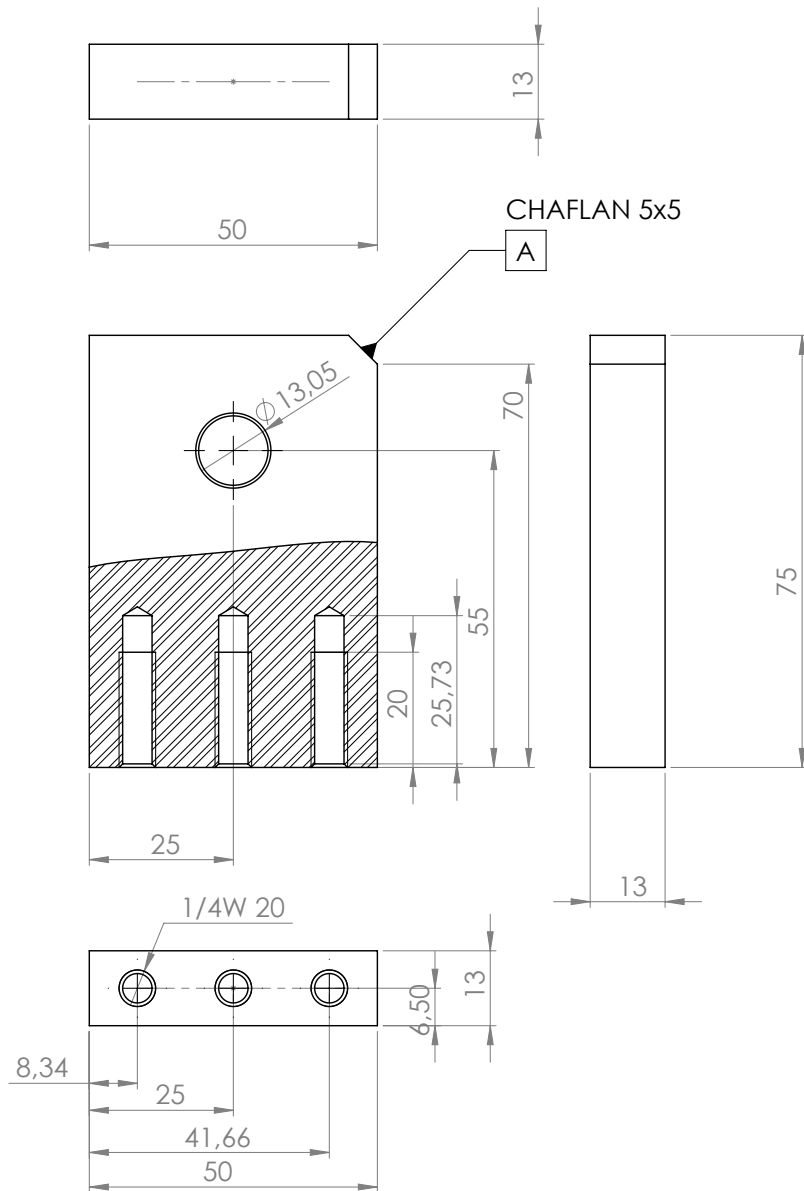


Figura A.14: Plano del motor



Proyecto: *Extrusora de Filamentos Poliméricos*

Título: *Soporte Delantero*

Material: *Aluminio*

Autor: *Leandro Cristobal*

Fecha: *Septiembre 2019*

Director: *Dr. Emiliano Penovi*

Co-Director: *Dr. Daniel Sosa*

Escala 1:1

Universidad Nacional de Mar del Plata

Facultad de ingeniería - UNMdP

Juan B. Justo 4302

B7600

Mar del Plata

Argentina



Figura A.15: Plano del soporte delantero (pivot)

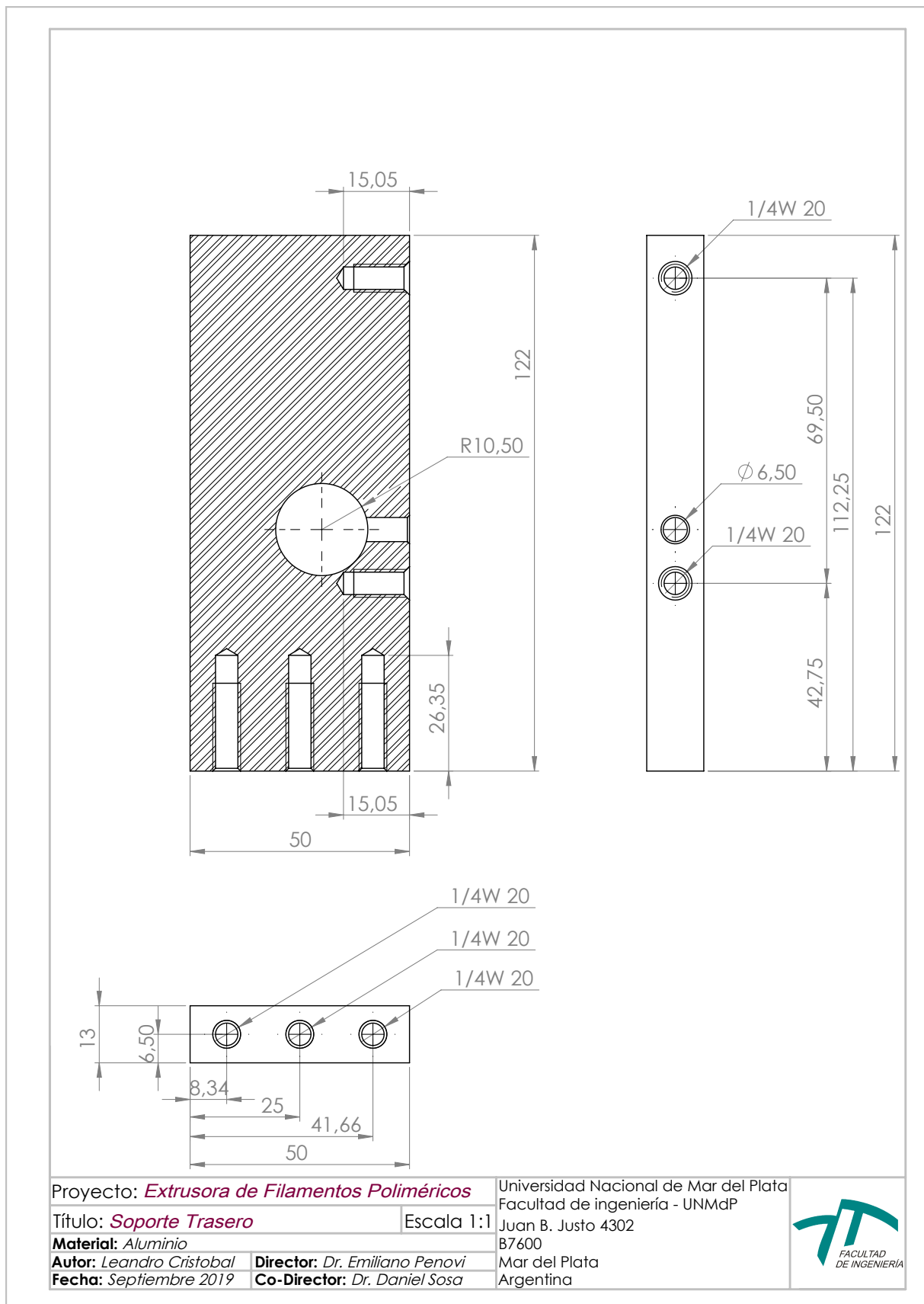


Figura A.16: Plano del soporte trasero (cremallera)

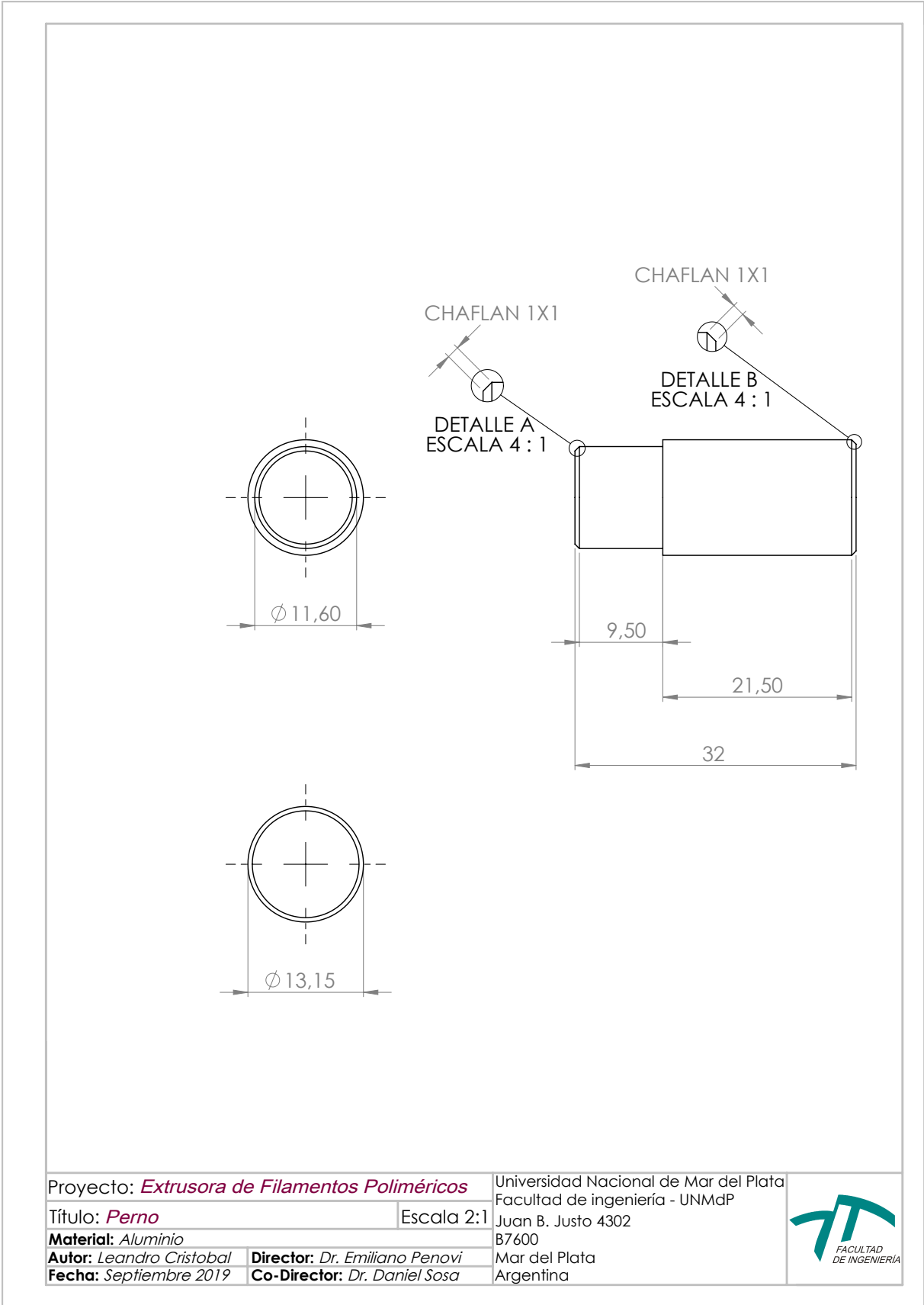
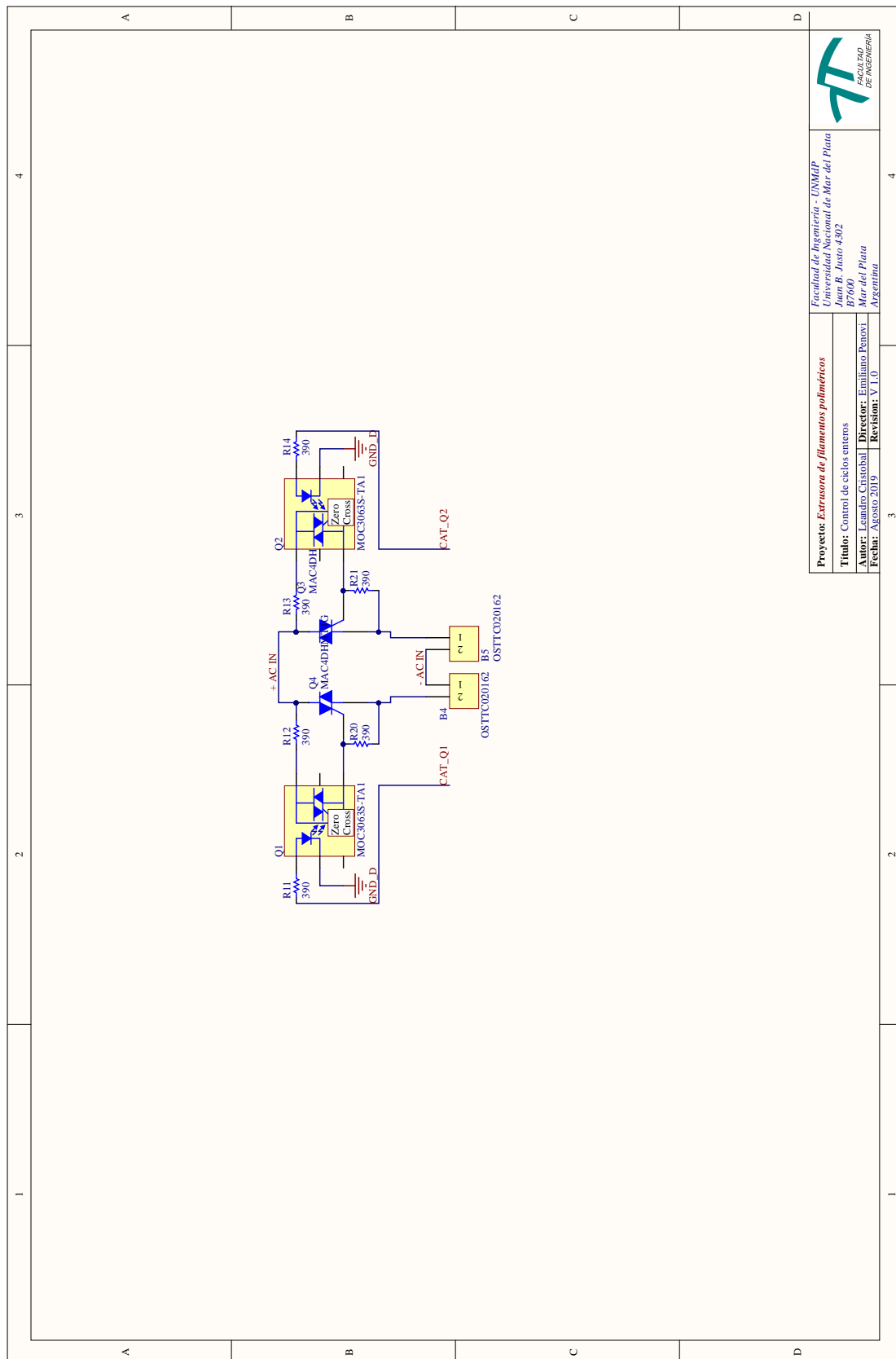


Figura A.17: Plano del perno pivot

Apéndice B

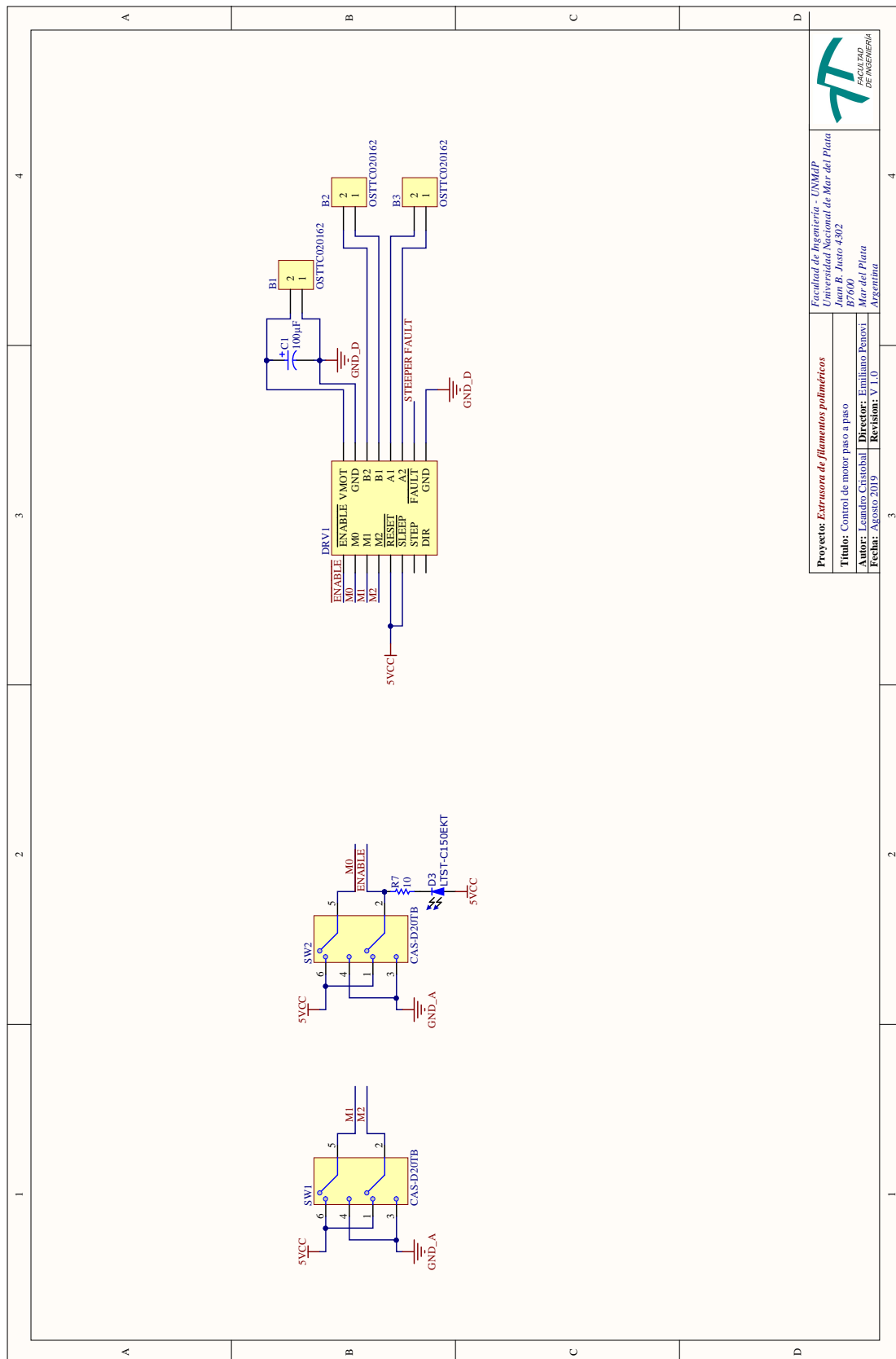
Planos circuitales



Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 B7600
 Mar del Plata
 Argentina

Proyecto: *Extrusora de filamentos poliméricos*
 Título: Control de ciclos enteros
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019
 Director: Emiliano Penovi
 Revisión: V 1.0

Figura B.1: Esquema del control por ciclos enteros

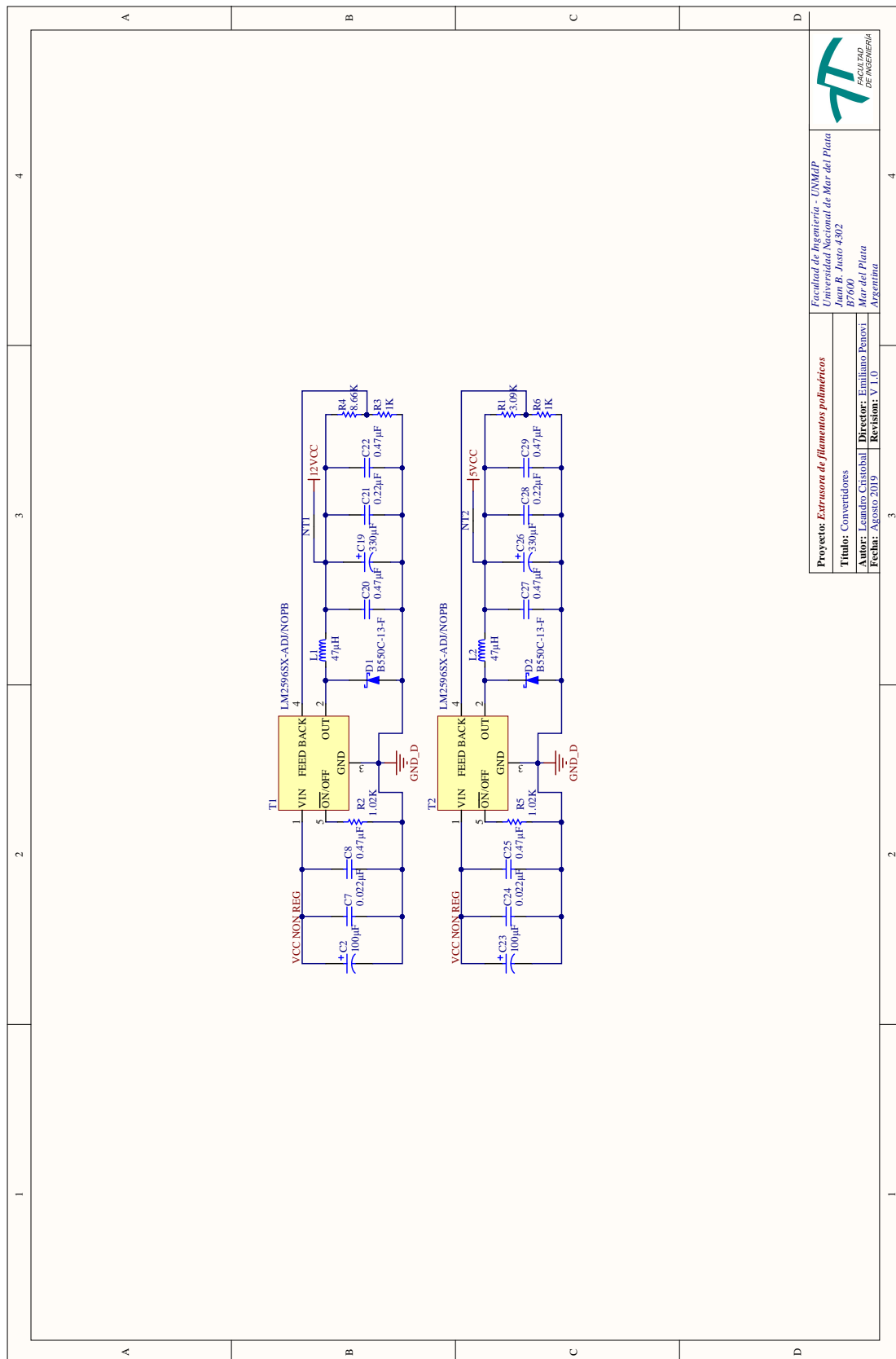


Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 B7600
 Mar del Plata
 Argentina

Proyecto: *Extrusora de filamentos poliméricos*
 Título: Control de motor paso a paso
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019

Director: Emiliano Penovi
 Revisión: V 1.0

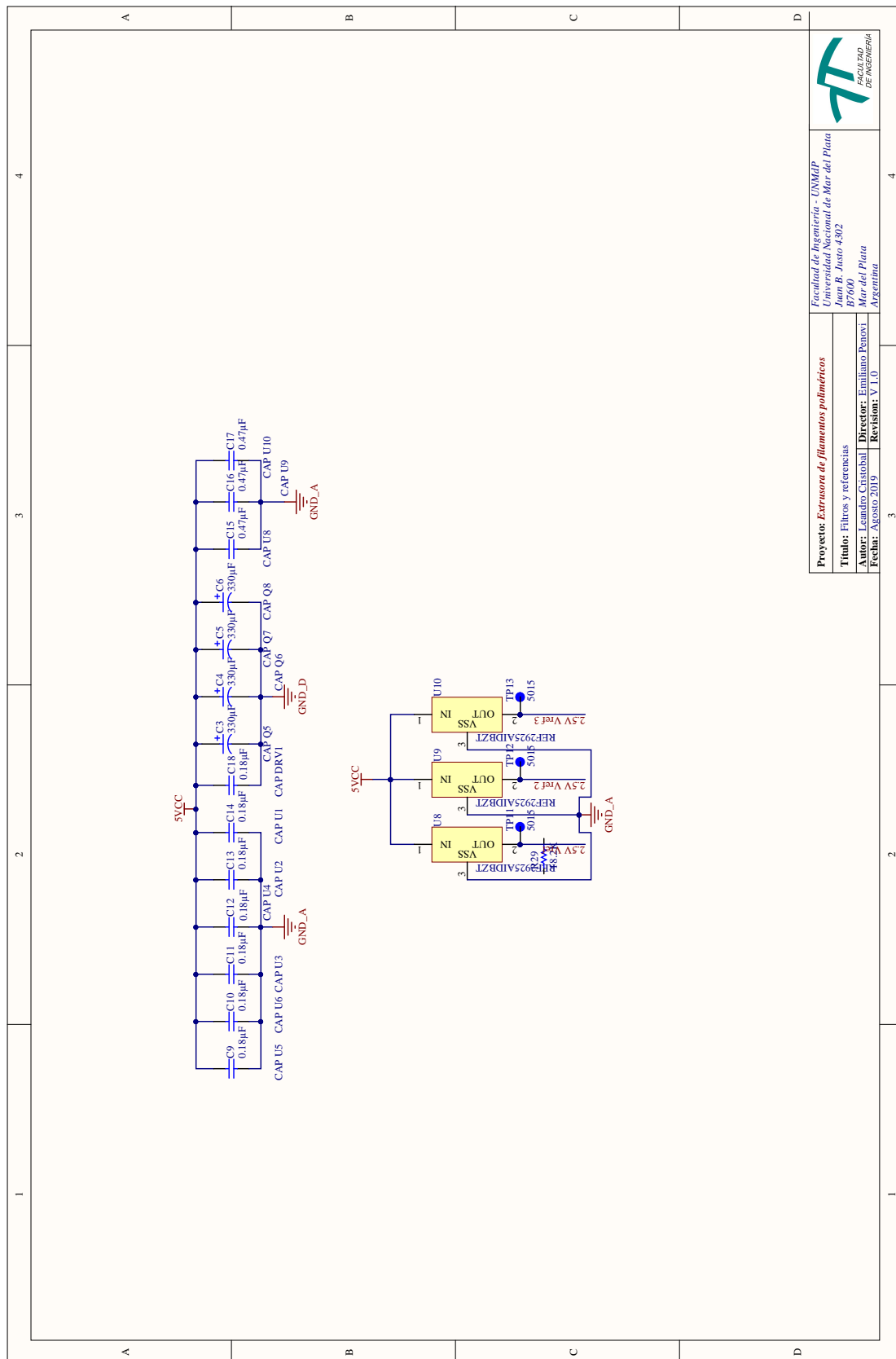
Figura B.2: Esquema del control del motor paso a paso



Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 7600
 Mar del Plata
 Argentina

Proyecto: Extrasora de filamentos poliméricos
 Título: Convertidores
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019
 Director: Emiliano Penovi
 Revisión: V 1.0

Figura B.3: Esquema de los convertidores DC/DC

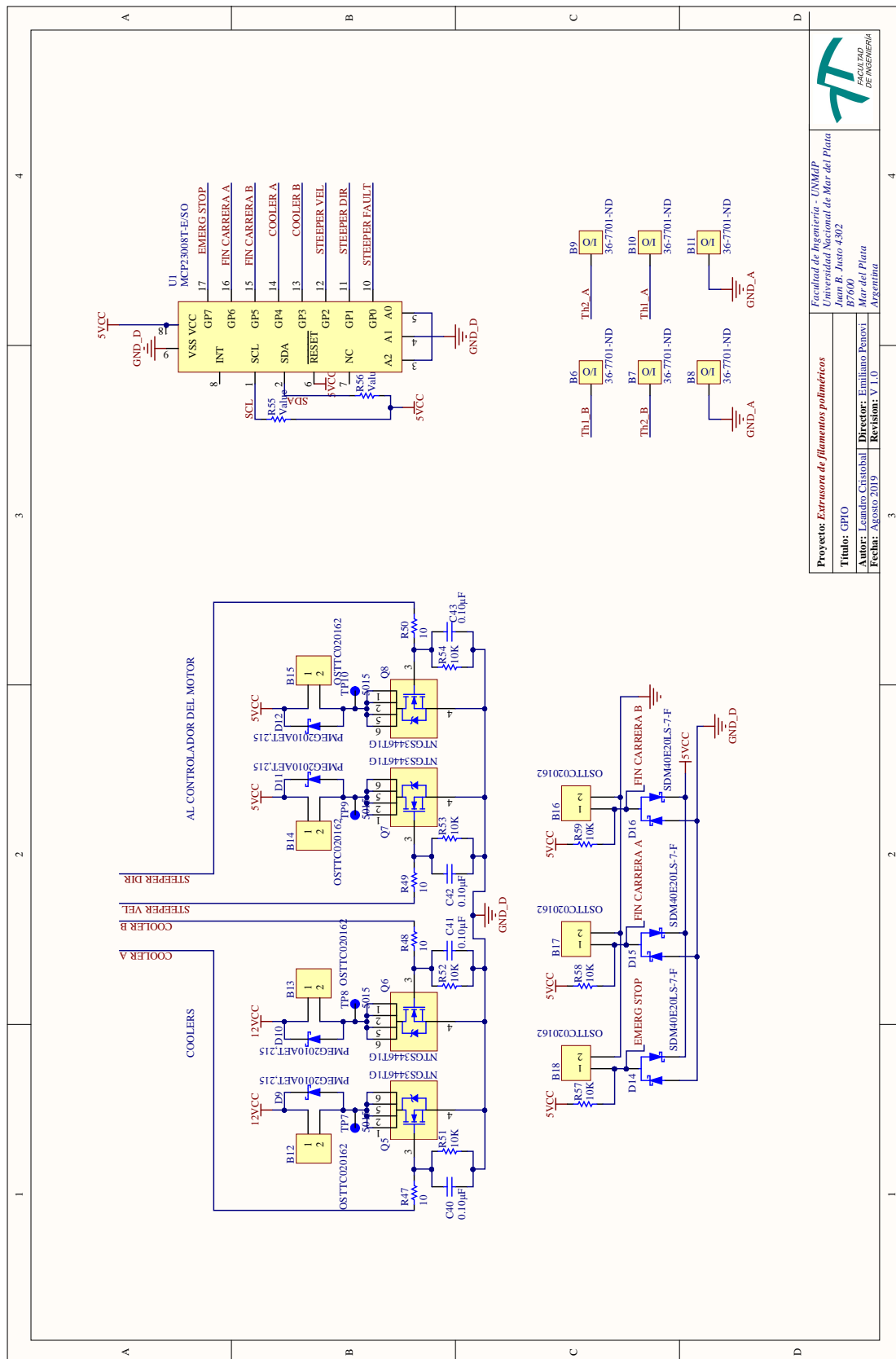


Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 7600
 Mar del Plata
 Argentina

Proyecto: *Extrusora de filamentos poliméricos*
 Título: Filtros y referencias
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019

Director: Emiliano Penovi
 Revisión: V 1.0

Figura B.4: Esquemas de capacitores de filtro y referencia de tensión



Proyecto: Extrusora de filamentos poliméricos
 Título: GPIO
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019
 Director: Emiliano Penovi
 Revisión: V 1.0

Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 B7600
 Mar del Plata
 Argentina

Figura B.5: GPIO's (Entradas y salidas)

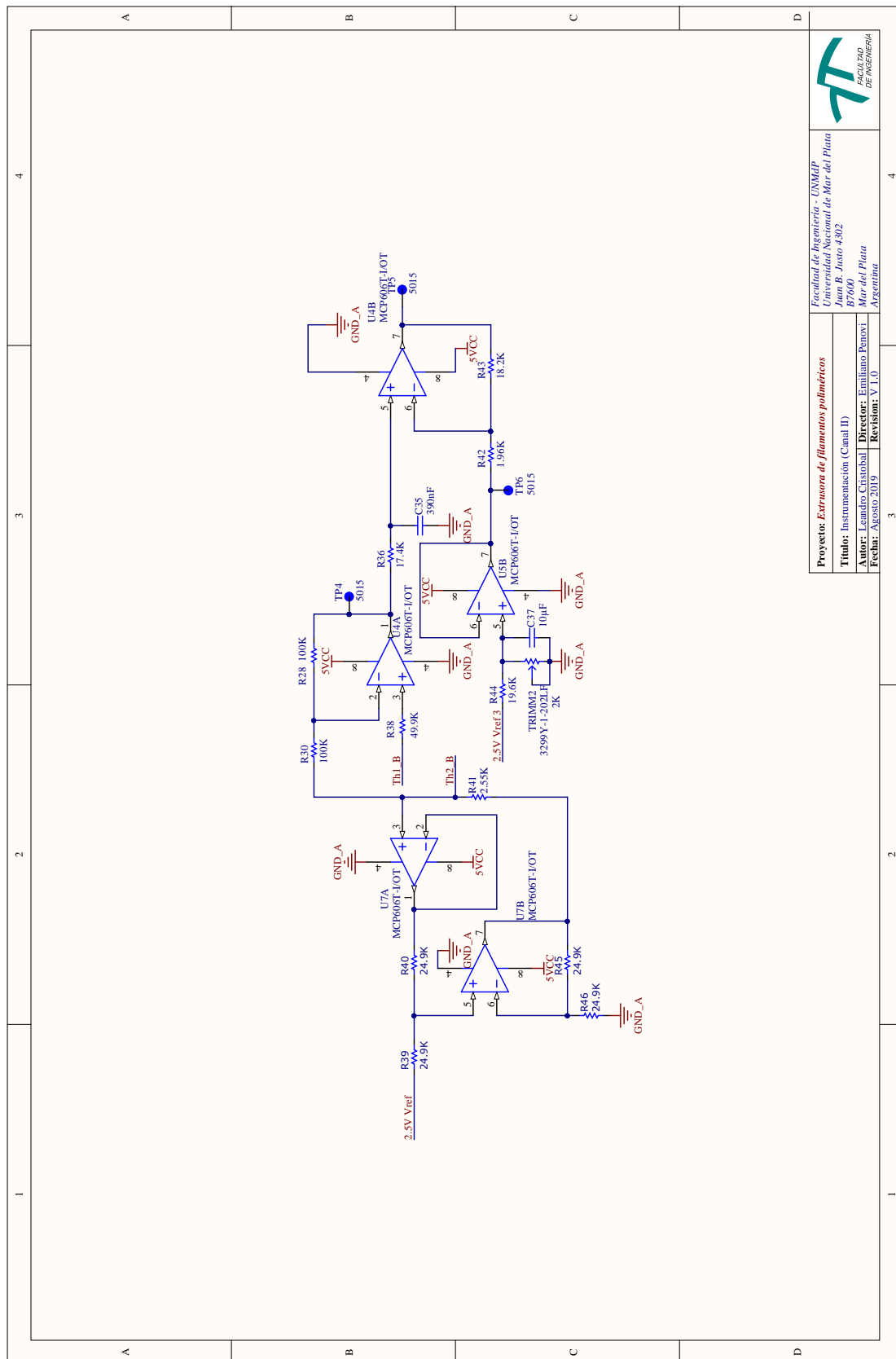


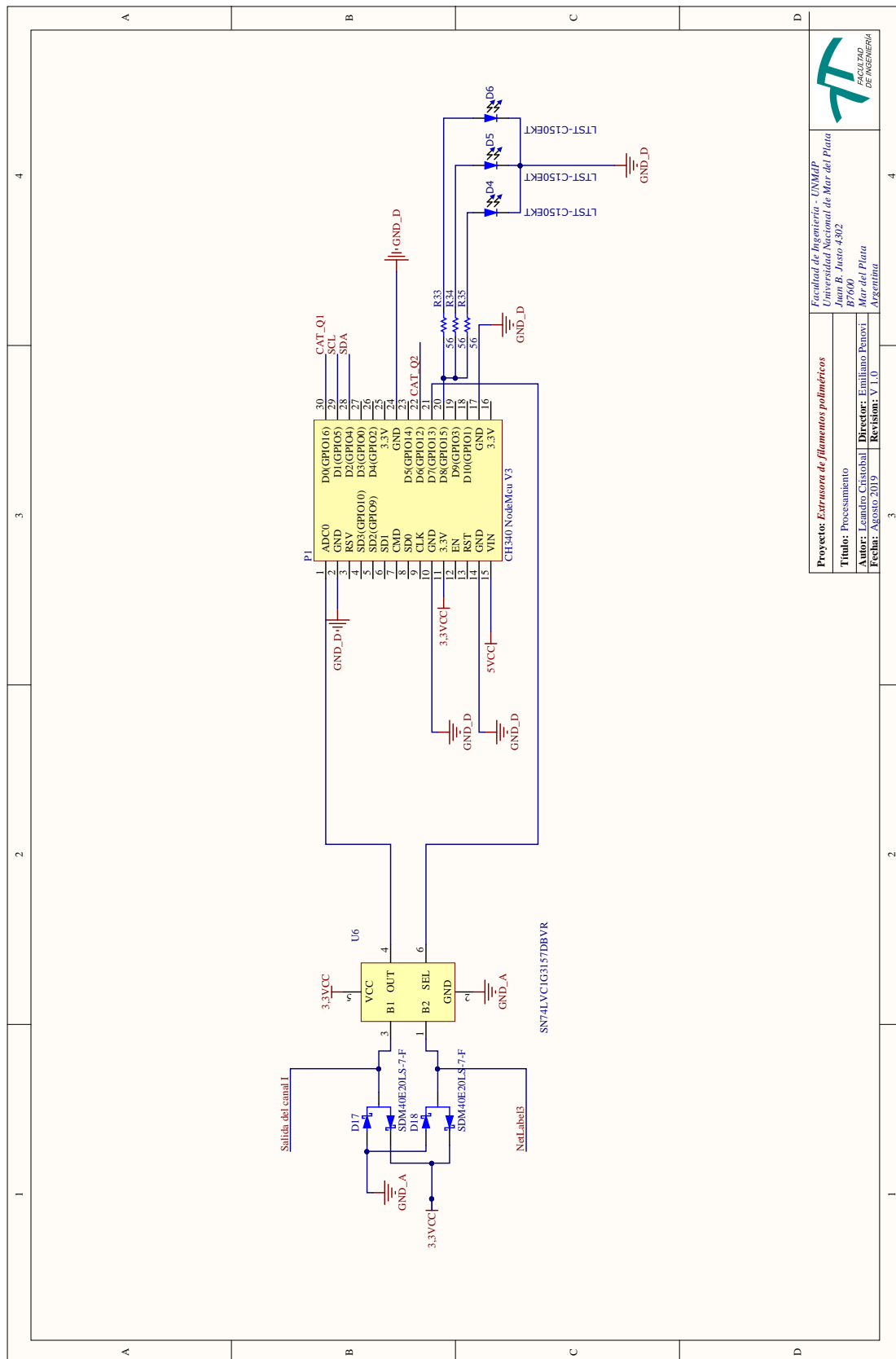
Figura B.7: Esquema de la instrumentación (canal II)



Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 B7600
 Mar del Plata
 Argentina

Proyecto: *Extrusora de filamentos poliméricos*
 Título: Instrumentación (Canal II)
 Autor: Leandro Cristóbal | Director: Emiliano Penovi
 Fecha: Agosto 2019 | Revisión: V 1.0

1	2	3	4
---	---	---	---



Facultad de Ingeniería - UNMAP
 Universidad Nacional de Mar del Plata
 Juan B. Justo 4302
 B7600
 Mar del Plata
 Argentina

Proyecto: *Extrusora de filamentos poliméricos*
 Título: Procesamiento
 Autor: Leandro Cristóbal
 Fecha: Agosto 2019
 Director: Emiliano Penovi
 Revisión: V 1.0

Figura B.8: Esquema del procesamiento

Apéndice C

Código

C.1. Código general

```
1 enum Estados {INICIO, PREPARACION, POSICIONAMIENTO, CALENTAMIENTO,
   DESGASIFICACION, EXTRUSION, RETROCESO, FIN, ERROR_, PAUSA_DESGASIFICACION,
   PAUSA_EXTRUSION, DETENIDO, FINCARRERA_LOW, FINCARRERA_HIGH, FINCARRERA
   };//defino los estados
2
3 #include <Wire.h>
4 #include "Adafruit_MCP23008.h"
5 #include <Ticker.h>
6 #include <ESP8266WiFi.h>
7
8 #define LedPlaca 15
9 #define LedSel 12
10 #define Interrupciones 14
11 #define DirBobinadora 2
12 #define StepBobinadora 0
13 #define ResistenciaCuerpo 13
14 #define ResistenciaBoquilla 16
15
16 uint8_t state = INICIO;
17 uint8_t StateSensor = FINCARRERA_HIGH;
18 int ContadorPosicion = 0;
19 int ContadorLedFin = 0;
20 int Contador3 = 0;
21 int Contador4 = 0;
22 int ContadorAvance = 0;
23 int Start = 1;
24 int Desgasificacion = 1;
25 int ControlError = 0;
26 int PosicionPistonArranque = 2500;//antes 2500 antes 3000
27 int CicloDeTrabajoBoquilla = 0;
28 int CicloDeTrabajoCuerpo = 0;
29 int CicloDeTrabajoBobinadora = 500;
```

```

30 int BandaProporcional = 3;
31 int Potencia = 1023;
32 int MuestreoTemperatura = 0;
33 int InicioCalentamiento = 0;
34 int Muestra = 1;
35 int PendienteTemperaturaCuerpo = 10; //En °C/minuto
36 int PendienteTemperaturaBoquilla = 10; //En °C/minuto
37 int PmaxBoquilla = 236; //Activo las dos resistencias al mismo tiempo
38 int PmaxCuerpo = 228;
39 int FrecuenciaPWM = 900;
40 int i = 0;
41 int j = 0;
42 int Bandera;
43 int TimerTemp[3] = {0, 0, 0};
44 int IndiceTemperatura = 0;
45 int TimerTempSat[3] = {1200*1000, 1200*1000, 900*1000}; // 20 minutos, 20
    minutos, 15 minutos (timer cada etapa)
46 int ValorIntegrador;
47
48 int InicioTimerTemp = 0;
49 int PeriodoMotor = 0;
50 int Ton = 10;
51 int ContadorPeriodoMotor = 0;
52
53 float FrecMuestreo = 1000;//
54 float PeriodoMuestreo = (1/FrecMuestreo); //Frecuencia de muestreo
55 float FrecuenciaControl = 0.2; //Cada 5 segundos
56 float PeriodoMuestreoControl = (1/FrecuenciaControl);
57 int Ventana = FrecMuestreo*PeriodoMuestreoControl; //este valor lo utilizo
    para filtrar y acceder al control.
58 int FondoEscalaPWM = Ventana;
59 float TemperaturaCuerpo[2];
60 float TemperaturaCuerpoAux;// = 0;
61 float TemperaturaCuerpoRef = 210;
62 float TemperaturaCuerpoRampa;
63 float DeltaT=TemperaturaCuerpoRef-TemperaturaCuerpoRampa;
64 float Tiempo=DeltaT/10;
65 float IncrementoRampaCuerpo = 0;
66 float PotenciaCuerpo[2];//={0, 0};//El indice 0 es el error en k, El indice 1
    es el error en k-1
67 float PotenciaCuerpoSat;
68 float PotenciaBoquilla;
69 float PotenciaBoquillaSat;
70 float TemperaturaBoquilla;
71 float TemperaturaBoquillaAux;
72 float VectorTemperaturaCuerpoRef[3] = {100, 125, 160};//Temperaturas de
    referencia de cada etapa
73 float DeltaP = 0;

```

```

74 float Error[3];//={0, 0, 0};//El indice 0 es el error en k, El indice 1 es el
    error en k-1, El indice 2 es el error en k-2
75 float Uint[2]={0, 0};
76 float Uder[2]={0, 0};
77 float Ts = PeriodoMuestreoControl;
78
79 float Kp = 5.1647;
80 float Ki = 0.0021;
81 float Kd = 261.2708;
82
83 float wp = 1;
84 float Kaw =Ki*0; Constante AntiWindUP (No implementada)
85
86 Ticker blinker;
87 Adafruit_MCP23008 mcp;
88
89 void MotorAvance();
90 void MotorRetroceso();
91 void ControlBoquilla();
92 void ControlCuerpo();
93 void ResistenciasOff();
94 void CoolersOn();
95 void CoolersOff();
96 void LedPlacaToggle();
97 void AdquisicionTemperaturaCuerpo();
98 void AdquisicionTemperaturaBoquilla();
99 void muestreo();
100 void Bobinadora();
101 void ActualizoDatos();
102 void IncrementoReferencia();
103 void IncrementoReferenciaCuerpo();
104 void FiltroCuerpo();
105 void PWM();
106 void PWMBobinadora();
107 void TimerTemperatura();
108
109 void setup(){
110     Serial.begin(115200);
111     while (!Serial);
112     delay(2000);
113     mcp.begin();
114     mcp.pinMode(1, OUTPUT);
115     mcp.pinMode(2, OUTPUT);
116     mcp.pinMode(4, OUTPUT);
117     mcp.pinMode(5, INPUT);
118     mcp.pinMode(6, INPUT);
119     mcp.pinMode(7, INPUT);
120     pinMode(Interrupciones, INPUT);

```

```

121 pinMode(LedPlaca, OUTPUT);
122 pinMode(LedSel, OUTPUT);
123 pinMode(StepBobinadora,OUTPUT);
124 pinMode(DirBobinadora,OUTPUT);
125 pinMode(ResistenciaCuerpo, OUTPUT);
126 pinMode(ResistenciaBoquilla, OUTPUT);
127 digitalWrite(ResistenciaBoquilla,LOW);
128 digitalWrite(ResistenciaCuerpo,LOW);
129 digitalWrite(DirBobinadora,HIGH);
130 digitalWrite(LedSel,HIGH);
131 TemperaturaCuerpo[0] = (((analogRead(A0)*3.3/1023.0 +
    0.10164*37.34)/(38.34*1.003*0.000987*100.0))-1)/0.0039);
132 TemperaturaCuerpoRampa = TemperaturaCuerpo[0]*1.1; // inicializo la rampa
    10% por encima del valor de temperatura del cuerpo (esto es para evitar
    que al arranque se cumpla la condición del 99% de Tref)
133 analogWriteFreq(FrecuenciaPWM);
134 IncrementoReferencia();
135 state = INICIO;
136 blinker.attach(PeriodoMuestreo, muestreo); // Cada PeriodoMuestreo
    (segundos), ejecuto la función muestreo
137 }
138 void loop(){
139
140 int FinDeCarreraTrasero = mcp.digitalRead(5);
141 int FinDeCarreraDelantero = mcp.digitalRead(6);
142 int ParadaDeEmergencia = mcp.digitalRead(7);
143
144 if (ParadaDeEmergencia == HIGH){
145     CoolersOn();
146     switch(state)
147     {
148         case INICIO:
149             if(FinDeCarreraDelantero == HIGH){
150                 state = POSICIONAMIENTO;
151             }
152             else{
153                 state = PREPARACION;
154             }
155             break;
156         case POSICIONAMIENTO:
157             if (FinDeCarreraDelantero == HIGH) {
158                 PeriodoMotor=Ton*2;
159                 MotorRetroceso();
160                 state = POSICIONAMIENTO;
161             }
162             else{
163                 state = PREPARACION;
164                 delay(1500);

```

```

165     }
166     break;
167 case PREPARACION:
168     if(ContadorPosicion < PosicionPistonArranque){
169         PeriodoMotor=Ton*2;
170         MotorAvance();
171         ContadorPosicion++;
172         state = PREPARACION;
173     }
174     else{
175         delay(1500);
176         ContadorPosicion=0;
177         state = CALENTAMIENTO;
178     }
179     break;
180 case CALENTAMIENTO:
181     MuestreoTemperatura = 1;
182     InicioCalentamiento = 1;
183     if((TemperaturaCuerpo[0] >
184         0.99*VectorTemperaturaCuerpoRef[IndiceTemperatura] ) && (Start=1)){
185         InicioTimerTemp = 1;
186     }
187     break;
188 case DESGASIFICACION:
189     state = EXTRUSION;
190     break;
191
192 case EXTRUSION:
193     if(ContadorAvance<2){
194         PeriodoMotor=Ton*80*10*1.5;
195         MuestreoTemperatura = 1;
196         InicioCalentamiento = 1;
197         MotorAvance();
198         state = EXTRUSION;
199         switch(StateSensor){
200             case FINCARRERA_HIGH:
201                 if(FinDeCarreraTrasero == LOW){
202                     ContadorAvance++;
203                     StateSensor = FINCARRERA_LOW;
204                 }
205                 break;
206             case FINCARRERA_LOW:
207                 if(FinDeCarreraTrasero == LOW){
208                     StateSensor = FINCARRERA_LOW;
209                 }
210             else {
211                 StateSensor = FINCARRERA_HIGH;

```

```

212     }
213     break;
214 }
215 }
216 else{
217
218     state = RETROCESO;
219 }
220 break;
221 case RETROCESO:
222 MuestreoTemperatura = 1;
223 InicioCalentamiento = 0;
224 if(FinDeCarreraDelantero == HIGH){
225
226     PeriodoMotor=Ton*2;
227     MotorRetroceso();
228     state = RETROCESO;
229 }
230 else{
231     state = FIN;
232 }
233 break;
234 case FIN:
235     CoolersOff();
236     state = FIN;
237     MuestreoTemperatura = 0;
238     InicioCalentamiento = 0;
239     Start = 1;
240 break;
241 case ERROR_:
242 if(ControlError == 1){
243     MuestreoTemperatura = 1;
244     InicioCalentamiento = 0;
245     state = ERROR_;
246 }
247 else{
248     state = FIN;
249 }
250 break;
251 case PAUSA_DESGASIFICACION:
252 if(Start == 1){
253     state = DESGASIFICACION;
254 }
255 else{
256     MuestreoTemperatura = 1;
257     InicioCalentamiento = 1;
258     state = PAUSA_DESGASIFICACION;
259 }

```

```

260     break;
261     case PAUSA_EXTRUSION:
262     if(Start == 1){
263         state = EXTRUSION;
264     }
265     else {
266         MuestreoTemperatura = 1;
267         InicioCalentamiento = 1;
268         state = PAUSA_EXTRUSION;
269     }
270     break;
271     case DETENIDO:
272     if(ParadaDeEmergencia==1){
273         MuestreoTemperatura = 0;
274         InicioCalentamiento = 0;
275         state = DETENIDO;
276     }
277     else{
278         state = INICIO;
279     }
280     break;
281 }
282 }
283 else{
284     CoolersOff();
285     analogWrite(StepBobinadora,512);
286     ResistenciasOff();
287 }
288 }
289 void muestreo(){
290     digitalWrite(LedPlaca,HIGH); //Toggle LED Pin
291     if (MuestreoTemperatura == 1) {
292         AdquisicionTemperaturaBoquilla(); //Adquiero cada 1 milisegundos
293         AdquisicionTemperaturaCuerpo();
294     }
295     if (InicioCalentamiento == 1) {
296         IncrementoReferenciaCuerpo();
297         if (Contador3==Ventana) {
298             ControlBoquilla();
299             ControlCuerpo();
300             Contador3=0;
301         }
302         PWM();
303     }
304     if (MuestreoTemperatura == 1) {
305         ActualizoDatos();
306     }
307     if (InicioTimerTemp==1){

```

```

308     TimerTemperatura();
309 }
310
311     ContadorPeriodoMotor++;
312     digitalWrite(LedPlaca,LOW);
313 }
314 void MotorAvance(){
315     mcp.digitalWrite(1, HIGH);//SETEO DIRECCIÓN
316     if (ContadorPeriodoMotor<PeriodoMotor){
317         if(ContadorPeriodoMotor<Ton){
318             mcp.digitalWrite(2,HIGH);
319         }
320         else{
321             mcp.digitalWrite(2, LOW);
322         }
323     }
324     else{
325         ContadorPeriodoMotor = 0;
326     }
327 }
328 void MotorRetroceso(){
329     mcp.digitalWrite(1, LOW);//SETEO DIRECCIÓN
330     if (ContadorPeriodoMotor<PeriodoMotor){
331         if(ContadorPeriodoMotor<Ton){
332             mcp.digitalWrite(2,HIGH);
333         }
334         else{
335             mcp.digitalWrite(2, LOW);
336         }
337     }
338     else{
339         ContadorPeriodoMotor = 0;
340     }
341 }
342
343 void CoolersOn (){
344     mcp.digitalWrite(4,HIGH);
345 }
346 void CoolersOff (){
347     mcp.digitalWrite(4,LOW);
348 }
349 void LedPlacaToggle (){
350     while(ContadorLedFin<10){
351         digitalWrite(LedPlaca,HIGH);
352         delay(500);
353         digitalWrite(LedPlaca,LOW);
354         delay(500);
355         ContadorLedFin++;

```



```

356     }
357     ContadorLedFin=0;
358 }
359 void ResistenciasOff(){
360     InicioCalentamiento = 0;
361     PotenciaBoquillaSat=0;
362     PotenciaCuerpoSat=0;
363 }
364 void AdquisicionTemperaturaCuerpo(){
365     digitalWrite(LedSel,HIGH);
366     TemperaturaCuerpoAux += (((analogRead(A0)*3.3/1023.0 +
367         0.10164*37.34)/(38.34*1.003*0.000987*100.0))-1)/0.0039);
367     Contador3++;
368
369     if (Muestra%Ventana==0){
370     TemperaturaCuerpo[0] = TemperaturaCuerpoAux/Ventana;
371     TemperaturaCuerpoAux = 0;
372     }
373 }
374
375 void AdquisicionTemperaturaBoquilla(){
376     digitalWrite(LedSel,LOW);
377     TemperaturaBoquillaAux += (((analogRead(A0)*3.3/1023.0 +
378         0.10164*37.34)/(38.34*1.003*0.000987*100.0))-1)/0.0039);
378
379     if (Muestra%Ventana==0){
380     TemperaturaBoquilla = TemperaturaBoquillaAux/Ventana;
381     TemperaturaBoquillaAux = 0;
382     }
383 }
384
385 void ControlCuerpo(){
386     Error[0]=TemperaturaCuerpoRampa-TemperaturaCuerpo[0];
387
388     DeltaP =
389         (PotenciaCuerpoSat+PotenciaBoquillaSat)-(PotenciaCuerpo[0]+PotenciaBoquilla);
389
390     //Parte Proporcional
391     PotenciaCuerpo[0]= Kp*Error[0];
392
393     //Parte Integral
394     Uint[0] = Uint[1] + Error[0];
395     Uint[1] = Uint[0];
396     PotenciaCuerpo[0] += (Ki*Ts)*Uint[1];
397     ValorIntegrador=(Ki*Ts)*Uint[1];
398
399     //Parte Derivativa
400

```

```

401 Uder[0] = (Kd*wp/(1+wp*Ts))*(Error[0]-Error[1]) + (1/(1+wp*Ts))*Uder[1];
402 Uder[1] = Uder[0];
403 PotenciaCuerpo[0] += Uder[0];
404
405 //Inicio Reasignacion
406 Error[2]=Error[1];
407 Error[1]=Error[0];
408 PotenciaCuerpo[1]=PotenciaCuerpo[0];
409 //Fin Reasignacion
410
411 if (PotenciaCuerpo[0]>PmaxCuerpo){
412     PotenciaCuerpoSat=PmaxCuerpo;
413 }
414 else if (PotenciaCuerpo[0]<0){
415     PotenciaCuerpoSat=0;
416 }
417 else {
418     PotenciaCuerpoSat=PotenciaCuerpo[0];
419 }
420 CicloDeTrabajoCuerpo=PotenciaCuerpoSat*FondoEscalaPWM/PmaxCuerpo;
421
422 ContadorLedFin++;
423 }
424 void ControlBoquilla(){
425     PotenciaBoquilla=PotenciaCuerpo[0];
426     if (PotenciaBoquilla>PmaxBoquilla){
427         PotenciaBoquillaSat=PmaxBoquilla;
428     }
429     else if (PotenciaBoquilla<0){
430         PotenciaBoquillaSat=0;
431     }
432     else {
433         PotenciaBoquillaSat=PotenciaBoquilla;
434     }
435     CicloDeTrabajoBoquilla=PotenciaBoquillaSat*FondoEscalaPWM/PmaxBoquilla;
436
437 }
438 void ActualizoDatos(){
439     if (Muestra%Ventana==0){
440
441         Serial.print("A{"); //EX TCU
442         Serial.print(TemperaturaBoquilla);
443         Serial.print("}");
444
445         Serial.print("C{"); //EX ACB
446         Serial.print(PotenciaCuerpoSat+PotenciaBoquillaSat);
447         Serial.print("}");
448

```

```

449 Serial.print("D{"); //EX TBR
450 Serial.print(TemperaturaCuerpoRampa);
451 Serial.print("}");
452
453 Serial.print("E{"); //EX TCU
454 Serial.print(TemperaturaCuerpo[0]);
455 Serial.print("}");
456
457 Serial.print("F{"); //EX ETC
458 Serial.print(Error[0]);
459 Serial.print("}");
460
461 Serial.print("M{");
462 Serial.print(Muestra);
463 Serial.print("}");
464
465 Serial.print("X{");
466 Serial.print(ValorIntegrador);
467 Serial.print("}");
468
469 Serial.print("U{");
470 Serial.print(InicioTimerTemp);
471 Serial.print("}");
472
473 Serial.print("Z{");
474 Serial.print(CicloDeTrabajoCuerpo);
475 Serial.print("}");
476
477 Serial.println("");
478 }
479 Muestra++;
480 }
481 void IncrementoReferencia(){
482     IncrementoRampaCuerpo=PendienteTemperaturaCuerpo/(60*FrecMuestreo);
483 }
484 void IncrementoReferenciaCuerpo(){
485     if (TemperaturaCuerpoRampa>=VectorTemperaturaCuerpoRef [IndiceTemperatura]){
486         TemperaturaCuerpoRampa=VectorTemperaturaCuerpoRef [IndiceTemperatura];
487     }
488     else{
489         TemperaturaCuerpoRampa=TemperaturaCuerpoRampa+IncrementoRampaCuerpo;
490     }
491 }
492 void PWM(){
493     if (i<FondoEscalaPWM){
494         i++;
495     }
496     else {

```

```

497     i=0;
498 }
499 if (i<CicloDeTrabajoCuerpo){
500     digitalWrite(ResistenciaCuerpo,HIGH);
501 }
502 else {
503     digitalWrite(ResistenciaCuerpo,LOW);
504 }
505 if (i<CicloDeTrabajoBoquilla){
506     digitalWrite(ResistenciaBoquilla,HIGH);
507 }
508 else {
509     digitalWrite(ResistenciaBoquilla,LOW);
510 }
511 }
512 void PWMBobinadora(){
513     if (j<FondoEscalaPWM){
514         j++;
515     }
516     else {
517         j=0;
518     }
519     if (j<CicloDeTrabajoBobinadora){
520         digitalWrite(StepBobinadora,HIGH);
521     }
522     else {
523         digitalWrite(StepBobinadora,LOW);
524     }
525 }
526 void TimerTemperatura(){
527     TimerTemp[IndiceTemperatura]++;
528     if(TimerTemp[IndiceTemperatura] > TimerTempSat[IndiceTemperatura]){
529         IndiceTemperatura++;
530         InicioTimerTemp = 0;
531         if (IndiceTemperatura>2){
532             IndiceTemperatura = 2;
533             state = DESGASIFICACION;
534             InicioTimerTemp = 0;
535         }
536     }
537 }

```

C.2. Código librería MCP23008

```
1  /*****
2   This is a library for the MCP23008 i2c port expander
3
4   These displays use I2C to communicate, 2 pins are required to
5   interface
6   Adafruit invests time and resources providing this open source code,
7   please support Adafruit and open-source hardware by purchasing
8   products from Adafruit!
9
10  Written by Limor Fried/Ladyada for Adafruit Industries.
11  BSD license, all text above must be included in any redistribution
12  *****/
13
14  #if ARDUINO >= 100
15    #include "Arduino.h"
16  #else
17    #include "WProgram.h"
18  #endif
19  #include <Wire.h>
20  #ifdef __AVR
21    #include <avr/pgmspace.h>
22  #elif defined(ESP8266)
23    #include <pgmspace.h>
24  #endif
25  #include "Adafruit_MCP23008.h"
26
27
28  //////////////////////////////////////
29  // RTC_DS1307 implementation
30
31  void Adafruit_MCP23008::begin(uint8_t addr) {
32    if (addr > 7) {
33      addr = 7;
34    }
35    i2caddr = addr;
36
37    Wire.begin(4,5); //Definición de pines de comunicación I2C: 4 SDA, 5 SCL
38
39    // set defaults!
40    Wire.beginTransmission(MCP23008_ADDRESS | i2caddr);
41  #if ARDUINO >= 100
42    Wire.write((byte)MCP23008_IODIR);
43    Wire.write((byte)0xFF); // all inputs
44    Wire.write((byte)0x00);
45    Wire.write((byte)0x00);
46    Wire.write((byte)0x00);
```

```

47 Wire.write((byte)0x00);
48 Wire.write((byte)0x00);
49 Wire.write((byte)0x00);
50 Wire.write((byte)0x00);
51 Wire.write((byte)0x00);
52 Wire.write((byte)0x00);
53 #else
54 Wire.send(MCP23008_IODIR);
55 Wire.send(0xFF); // all inputs
56 Wire.send(0x00);
57 Wire.send(0x00);
58 Wire.send(0x00);
59 Wire.send(0x00);
60 Wire.send(0x00);
61 Wire.send(0x00);
62 Wire.send(0x00);
63 Wire.send(0x00);
64 Wire.send(0x00);
65 #endif
66 Wire.endTransmission();
67
68 }
69
70 void Adafruit_MCP23008::begin(void) {
71   begin(0);
72 }
73
74 void Adafruit_MCP23008::pinMode(uint8_t p, uint8_t d) {
75   uint8_t iodir;
76
77
78   // only 8 bits!
79   if (p > 7)
80     return;
81
82   iodir = read8(MCP23008_IODIR);
83
84   // set the pin and direction
85   if (d == INPUT) {
86     iodir |= 1 << p;
87   } else {
88     iodir &= ~(1 << p);
89   }
90
91   // write the new IODIR
92   write8(MCP23008_IODIR, iodir);
93 }
94

```

```

95 uint8_t Adafruit_MCP23008::readGPIO(void) {
96     // read the current GPIO input
97     return read8(MCP23008_GPIO);
98 }
99
100 void Adafruit_MCP23008::writeGPIO(uint8_t gpio) {
101     write8(MCP23008_GPIO, gpio);
102 }
103
104
105 void Adafruit_MCP23008::digitalWrite(uint8_t p, uint8_t d) {
106     uint8_t gpio;
107
108     // only 8 bits!
109     if (p > 7)
110         return;
111
112     // read the current GPIO output latches
113     gpio = readGPIO();
114
115     // set the pin and direction
116     if (d == HIGH) {
117         gpio |= 1 << p;
118     } else {
119         gpio &= ~(1 << p);
120     }
121
122     // write the new GPIO
123     writeGPIO(gpio);
124 }
125
126 void Adafruit_MCP23008::pullUp(uint8_t p, uint8_t d) {
127     uint8_t gppu;
128
129     // only 8 bits!
130     if (p > 7)
131         return;
132
133     gppu = read8(MCP23008_GPPU);
134     // set the pin and direction
135     if (d == HIGH) {
136         gppu |= 1 << p;
137     } else {
138         gppu &= ~(1 << p);
139     }
140     // write the new GPIO
141     write8(MCP23008_GPPU, gppu);
142 }

```

```

143
144 uint8_t Adafruit_MCP23008::digitalRead(uint8_t p) {
145     // only 8 bits!
146     if (p > 7)
147         return 0;
148
149     // read the current GPIO
150     return (readGPIO() >> p) & 0x1;
151 }
152
153 uint8_t Adafruit_MCP23008::read8(uint8_t addr) {
154     Wire.beginTransaction(MCP23008_ADDRESS | i2caddr);
155     #if ARDUINO >= 100
156     Wire.write((byte)addr);
157     #else
158     Wire.send(addr);
159     #endif
160     Wire.endTransmission();
161     Wire.requestFrom(MCP23008_ADDRESS | i2caddr, 1);
162
163     #if ARDUINO >= 100
164     return Wire.read();
165     #else
166     return Wire.receive();
167     #endif
168 }
169
170
171 void Adafruit_MCP23008::write8(uint8_t addr, uint8_t data) {
172     Wire.beginTransaction(MCP23008_ADDRESS | i2caddr);
173     #if ARDUINO >= 100
174     Wire.write((byte)addr);
175     Wire.write((byte)data);
176     #else
177     Wire.send(addr);
178     Wire.send(data);
179     #endif
180     Wire.endTransmission();
181 }

```

C.3. Código interfaz (MatLAB)

```
1 % Defino las variables auxiliares
2 %-----
3 clc
4 clear
5 waittime= 1.e-03;%Variable auxiliar para esperar un caracter
6 first_boot = 1;
7 %disp('MSG: Available serial ports. '+ serialliist')
8 serialport_COM = 'COM4';
9 serialport_BR = 115200;
10 serialport = instrfind('Port', serialport_COM);
11 % -----
12 % Comienzo del programa principal
13 % -----
14 % close por if already created
15 if ~isempty( serialport )
16     fclose( serialport );
17     delete( serialport );
18     clear serialport
19 end
20
21 %-----
22 % Set UART parameters
23 serialport = serial(serialport_COM);
24
25 serialport.BaudRate = serialport_BR;
26 serialport.FlowControl = 'none';
27 serialport.Parity = 'none';
28 serialport.StopBits = 1;
29 serialport.DataBits = 8;
30 serialport.InputBufferSize = 5000000;
31 serialport.Timeout = 10;
32
33
34 %-----
35 % Open port and wait for data
36 %-----
37 fopen(serialport);
38 pause(.2)
39 disp('MSG: COM READY.')
```

```
40
41 flushinput(serialport); % Clean serial port input buffer
42
43 RX_LENGTH = 5000000; % Size of buffer
44 TX_LENGTH = 30; % Size of buffer
45 contador=0;
46
```

```

47 TemperaturaBoquilla = 0;
48 TemperaturaCuerpo = 0;
49 TemperaturaAmbiente = 0;
50 TemperaturaRampaCuerpo = 0;
51 ErrorTemperaturaBoquilla = 0;
52 AccionControl = 0;
53 AccionControlCuerpo = 0;
54 Muestras = 0;
55 n_sample = 1;
56 PotenciaCuerpo = 0;
57 ErrorTemperaturaCuerpo = 0;
58 Indice = 0;
59 Tiempo = 0;
60 ValorIntegrador = 0;
61 TimerCalentamiento = 0;
62 while max(Muestras) < 100000000
63     RX_BUFFER = char(fscanf(serialport));
64     TemperaturaBoquilla_temp = str2num(char(regex(RX_BUFFER,
65         '(?<=A{(.*)}(?=)')', 'match')));
66     TemperaturaBoquilla = [TemperaturaBoquilla TemperaturaBoquilla_temp];
67     TemperaturaCuerpo_temp = str2num(char(regex(RX_BUFFER,
68         '(?<=E{(.*)}(?=)')', 'match' )));
69     TemperaturaCuerpo = [TemperaturaCuerpo TemperaturaCuerpo_temp];
70     AccionControl_temp = str2num(char(regex(RX_BUFFER, '(?<=C{(.*)}(?=)')',
71         'match' )));
72     AccionControl = [AccionControl AccionControl_temp];
73     TemperaturaAmbiente_temp = str2num(char(regex(RX_BUFFER,
74         '(?<=U{(.*)}(?=)')', 'match' )));
75     TemperaturaAmbiente = [TemperaturaAmbiente TemperaturaAmbiente_temp];
76     TemperaturaRampaCuerpo_temp = str2num(char(regex(RX_BUFFER,
77         '(?<=D{(.*)}(?=)')', 'match' )));
78     TemperaturaRampaCuerpo = [TemperaturaRampaCuerpo
79         TemperaturaRampaCuerpo_temp];
80     ErrorTemperaturaCuerpo_temp=str2num(char(regex(RX_BUFFER,
81         '(?<=F{(.*)}(?=)')', 'match' )));
82     ErrorTemperaturaCuerpo = [ErrorTemperaturaCuerpo
83         ErrorTemperaturaCuerpo_temp];
84     Muestras_temp=str2num(char(regex(RX_BUFFER, '(?<=M{(.*)}(?=)')', 'match'
85         )));
86     Muestras = [Muestras Muestras_temp];
87     ValorIntegrador_temp=str2num(char(regex(RX_BUFFER, '(?<=X{(.*)}(?=)')',
88         'match' )));
89     ValorIntegrador = [ValorIntegrador ValorIntegrador_temp];
90     TimerCalentamiento_temp=str2num(char(regex(RX_BUFFER,
91         '(?<=U{(.*)}(?=)')', 'match' )));
92     TimerCalentamiento = [TimerCalentamiento TimerCalentamiento_temp];
93     Tiempo = Muestras/1000;
94     TiempoMinutos = Tiempo/60;

```

```

84     if mod(Muestras(end), 20000) == 0
85         figure(3),subplot(3,3,[1 2 4 5]),plot(Tiempo,TemperaturaCuerpo,'g'),
            hold on;
86         figure(3),subplot(3,3,[1 2 4 5
            ]),plot(Tiempo,TemperaturaRampaCuerpo,'b');
87         figure(3),subplot(3,3,[1 2 4 5]),plot(Tiempo,TemperaturaBoquilla,'r'),
            grid minor, hold off, title('Temperatura vs. Referencia de
            temperatura'); legend('Temperatura Piston','Temperatura de
            referencia','Temperatura de la boquilla','Location','southeast');
88         figure(3),subplot(3,3,7),plot(Tiempo,ErrorTemperaturaCuerpo,'r'), grid
            minor, title('Error');
89         figure(3),subplot(3,3,[3 6]),plot(Tiempo,AccionControl,'b'), grid
            minor, title('Potencia Total');
90         figure(3),subplot(3,3,[8]),plot(Tiempo,TimerCalentamiento,'r'), grid
            minor, title('Inicio Timer');
91         figure(3),subplot(3,3,[9]),plot(Tiempo,ValorIntegrador,'r'), grid
            minor, title('Integrador');
92         serialport.BytesAvailable
93         save('EnsayoPID','AccionControl','ErrorTemperaturaCuerpo',
94             'Muestras','TemperaturaBoquilla','TemperaturaCuerpo',
95             'TemperaturaRampaCuerpo','Tiempo','TiempoMinutos',
96             'ValorIntegrador');
97     end
98     pause(waittime)
99 end

```

Bibliografía

- [1] “Lm2596s-adjevm users guide,” Texas Instruments, Tech. Rep., Junio 2016. [Online]. Available: <http://www.ti.com/lit/ug/snvu536/snvu536.pdf>
- [2] B. C. Baker, “Anti-aliasing analog filters for data acquisition systems,” Microchip Technology Inc., Tech. Rep., 1999. [Online]. Available: <http://ww1.microchip.com/downloads/en/appnotes/00699b.pdf>
- [3] “Nodemcu documentation,” Tech. Rep., 2019. [Online]. Available: <https://nodemcu.readthedocs.io/en/master/>
- [4] “Baudrate,” MathWorks, Tech. Rep., 2019. [Online]. Available: [mathworks.com/help/matlab/matlab_external/ baudrate.html](https://www.mathworks.com/help/matlab/matlab_external/ baudrate.html)
- [5] “Flowcontrol,” MathWorks, Tech. Rep., 2019. [Online]. Available: <https://www.mathworks.com/help/instrument/flowcontrol.html>
- [6] “Parity,” MathWorks, Tech. Rep., 2019. [Online]. Available: https://www.mathworks.com/help/matlab/matlab_external/parity.html
- [7] “Stopbits,” MathWorks, Tech. Rep., 2019. [Online]. Available: <https://www.mathworks.com/help/instrument/stopbits.html>
- [8] “Databits,” MathWorks, Tech. Rep., 2019. [Online]. Available: https://www.mathworks.com/help/matlab/matlab_external/databits.html
- [9] “Inputbufferize,” MathWorks, Tech. Rep., 2019. [Online]. Available: <https://www.mathworks.com/help/instrument/inputbufferize.html>
- [10] “Timeout,” MathWorks, Tech. Rep., 2019. [Online]. Available: <https://www.mathworks.com/help/instrument/timeout.html>