

UNIVERSIDAD MIGUEL HERNÁNDEZ DE
ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y
AUTOMÁTICA INDUSTRIAL



DESARROLLO E IMPLEMENTACIÓN DE UN
CALENTADOR PARA SISTEMAS DE
MICROPERFUSIÓN

TRABAJO DE FIN DE GRADO

Julio – 2020

AUTOR: Pablo Valero Martínez

DIRECTOR/ES: David Marroquí Sempere

Laura Butron García



RESUMEN

Entre los estudios que se realizan sobre cultivos neuronales, uno de los más extendidos es el estudio de los potenciales eléctricos generados por neuronas al ser estimuladas por estímulos físicos o químicos. En el Instituto de Investigación, Desarrollo e Innovación en Biotecnología Sanitaria de Elche se estudia el funcionamiento y modulación de un tipo de proteínas de membrana llamadas canales iónicos. Concretamente se investiga el diseño y desarrollo de fármacos dirigidos a la modulación de canales iónicos que son altamente sensibles a variaciones de temperatura, lo que les otorga la función de termorreceptores. Por ello, en su estudio se requiere un control preciso sobre los cambios de temperatura. Aunque en la actualidad existen equipos profesionales en el mercado para este fin, estos tienen un coste muy elevado.

Es por ello por lo que el objetivo de este trabajo es el desarrollo de un equipo de bajo coste que permita a los investigadores trabajar en fases incipientes del estudio y comprobar que estos equipos son adecuados para ellos y se adaptan a sus necesidades.

Para ello, primero se estudia la oferta actual del mercado para asentar las bases del trabajo y evaluar cuales son las características comunes a estos equipos. Después se desarrolla una plataforma de hardware y software que permita realizar las funciones de calentamiento de forma precisa. Además, se prueban distintos tipos de controladores de temperatura para comprobar cuál es el más adecuado para la aplicación.

Por último, se valida el sistema final y se elabora un presupuesto que indica cuales serían los costes de producción, tanto en pequeña como a gran escala.



ÍNDICE GENERAL

RESUMEN	3
ÍNDICE GENERAL	5
ÍNDICE DE TABLAS	8
ÍNDICE DE FIGURAS	9
1. INTRODUCCIÓN	12
1.1. OBJETIVOS DEL PROYECTO.....	13
1.2. PROPUESTA DE SOLUCIÓN	13
1.3. ESTUDIO DE MERCADO	14
1.3.1. <i>Definición del perfil de usuario</i>	16
1.3.2. <i>Estudio de la oferta actual</i>	16
1.3.3. <i>Análisis de productos sustitutivos</i>	20
1.3.4. <i>Análisis de precios</i>	21
1.3.5. <i>Conclusiones del estudio de mercado</i>	22
1.3.6. <i>Análisis Dafo</i>	23
1.4. REQUISITOS DEL PROYECTO	23
1.5. DISTRIBUCIÓN DE LA MEMORIA.....	24
2. DISEÑO DE HARDWARE	26
2.1. PLACA DE DESARROLLO ARDUINO UNO.....	26
2.2. SENSOR DE TEMPERATURA	27
2.2.1. <i>Sensor RTD</i>	28
2.2.2. <i>Termistor</i>	28
2.2.3. <i>Sensores digitales de temperatura</i>	29
2.3. ELEMENTOS CALEFACTORES.....	31
2.3.1. <i>Calentamiento mediante infrarojos</i>	31
2.3.2. <i>Celula de Peltier</i>	32
2.3.3. <i>Resistencias calefactoras</i>	33
2.4. ALIMENTACIÓN.....	36
2.5. DRIVERS	38
2.6. PANTALLA.....	39
2.7. SOPORTE DE LA AGUJA	41
2.7.1. <i>Validación térmica de los prototipos</i>	41
2.7.2. <i>Prototipo 1</i>	43
2.7.3. <i>Prototipo 2</i>	48
2.7.4. <i>Prototipo 3</i>	52

Índice general

2.7.5. Prototipo Final.....	57
3. DISEÑO DE SOFTWARE	61
3.1. ENTORNO DE TRABAJO	61
3.1.1. Arduino IDE	61
3.1.2. Librerías.....	62
3.2. ALGORITMO DE CONTROL	65
3.2.1. Control Todo/nada	65
3.2.2. Control PID	68
4. INTERFAZ DE USUARIO	80
4.1. PANTALLA PRINCIPAL	80
4.2. PANTALLA MENÚ.....	81
4.3. PANTALLA INTRODUCIR TEMPERATURA.....	82
4.4. PANTALLA CALIBRAR SISTEMA	82
4.5. PANTALLA INFO	83
5. FABRICACIÓN SISTEMA FINAL	84
5.1. PLACAS DE CIRCUITO IMPRESO	84
5.1.1. Diseño PCB	84
5.2. CARCASA DEL SISTEMA	86
5.2.1. Diseño 3D	86
5.2.2. Impresión 3D	90
5.3. PRESUPUESTO DE FABRICACIÓN.....	91
6. PRUEBAS EN ENTORNO REAL	95
6.1. MEDICIÓN DE TEMPERATURA CON CÁMARA TÉRMICA	95
6.2. PRUEBA DEL DISPOSITIVO CON CIRCULACIÓN DE AGUA	96
6.3. PRUEBA DEL DISPOSITIVO CON CIRCULACIÓN DE SOLUCIÓN DE LABORATORIO	97
6.4. PRUEBA DE MEDIDA DE LA TEMPERATURA REAL.....	97
6.5. PRUEBA DEL DISPOSITIVO CON DISTINTAS VELOCIDADES DE FLUIDO.....	98
7. CONCLUSIONES	100
8. MEJORAS Y FUTURAS LÍNEAS DE DESARROLLO	102
9. REFERENCIAS.....	104
10. ANEXOS.....	109
10.1. CÓDIGO	109
10.1.1. Main.cpp	109
10.1.2. Librería LCD	110

Índice general

10.1.3. Librería PID.....	116
10.1.4. Librería memoria.....	120
10.2. PLANOS.....	122
10.2.1. Prototipo 1	122
10.2.2. Prototipo 2	122
10.2.3. Prototipo final	123
10.2.4. Carcasa LCD.....	123
10.2.5. Carcasa aguja.....	124
10.3. ESQUEMA ELÉCTRICO	125
10.4 TABLAS DE PROPIEDADES DE LOS MATERIALES UTILIZADOS PARA LA SIMULACIÓN TÉRMICA	126
10.5 MANUAL DE USUARIO	128



Índice de Tablas

ÍNDICE DE TABLAS

TABLA 1. CARACTERÍSTICAS DEL CALENTADOR DE PERFUSIÓN DE MULTICHANNEL SYSTEMS [11].....	17
TABLA 2. CARACTERÍSTICAS DEL CALENTADOR DE PERFUSIÓN DE DIGITIMER [12]	18
TABLA 3. CARACTERÍSTICAS DEL CALENTADOR DE PERFUSIÓN DE WARNER INSTRUMENTS [13]	19
TABLA 4. CARACTERÍSTICAS DEL CALENTADOR DE JERINGUILLA DE WARNER INSTRUMENTS [14].....	21
TABLA 5. RESUMEN DE PRECIOS DE CALENTADORES DE PERFUSIÓN COMERCIALES.....	21
TABLA 6. CARACTERÍSTICAS DE LA PLACA ARDUINO UNO [16]	27
TABLA 7. RESUMEN DE LAS PRUEBAS DE CONFIGURACIÓN DE RESISTENCIAS	36
TABLA 8. PINES DE CONEXIÓN DEL KEYPAD LCD SHIELD.....	40
TABLA 9. TABLA DE PARÁMETROS DE LA SIMULACIÓN TÉRMICA DEL PRIMER PROTOTIPO	45
TABLA 10. TEMPERATURAS OBTENIDAS DE LA PRIMERA SIMULACIÓN DEL PROTOTIPO 1.....	46
TABLA 11. TEMPERATURAS OBTENIDAS DE LA SEGUNDA SIMULACIÓN TÉRMICA DEL PROTOTIPO 1	47
TABLA 12. TABLA DE PARÁMETROS DE LA SIMULACIÓN TÉRMICA DEL SEGUNDO PROTOTIPO	50
TABLA 13. TEMPERATURAS OBTENIDAS DE LA SIMULACIÓN DEL PROTOTIPO 2	52
TABLA 14. TABLA DE PARÁMETROS DE LA SIMULACIÓN TÉRMICA DEL TERCER PROTOTIPO.....	56
TABLA 15. TEMPERATURAS OBTENIDAS DE LA SIMULACIÓN DEL PROTOTIPO 3	57
TABLA 16. TABLA DE PARÁMETROS DE LA SIMULACIÓN DEL PROTOTIPO CUATRO	59
TABLA 17. RESULTADOS NUMÉRICOS DE LA SIMULACIÓN TÉRMICA DEL PROTOTIPO FINAL	60
TABLA 18. TABLA RESUMEN DE LOS CÁLCULOS NECESARIOS PARA HALLAR LOS PARÁMETROS DE CONTROL DEL ALGORITMO PID	74
TABLA 19. RESULTADOS DEL ALGORITMO DE AUTO-TUNE	79
TABLA 20. NORMAS DE DISEÑO UTILIZADAS EN LA FABRICACIÓN DE LAS PLACAS DE CIRCUITO IMPRESO DEL PROYECTO	85
TABLA 21. PARÁMETROS DE IMPRESIÓN 3D UTILIZADOS PARA LA FABRICACIÓN DE LAS CARCASAS DEL SISTEMA.....	90
TABLA 22. PRESUPUESTO DE COMPONENTES	91
TABLA 23. PRESUPUESTO DE FABRICACIÓN DEL DISPOSITIVO	94
TABLA 24. COMPOSICIÓN DE LA SOLUCIÓN DE LABORATORIO EMPLEADA POR EL IDIBE	97
TABLA 25. PROPIEDADES DEL ALUMINIO UTILIZADAS PARA LA SIMULACIÓN TÉRMICA	126
TABLA 26. PROPIEDADES DEL COBRE UTILIZADAS PARA LA SIMULACIÓN TÉRMICA	126
TABLA 27. PROPIEDADES DE LA RESINA EPOXI UTILIZADAS PARA LA SIMULACIÓN TÉRMICA.....	126
TABLA 28. PROPIEDADES DEL FR4 UTILIZADAS PARA LA SIMULACIÓN TÉRMICA	127

ÍNDICE DE FIGURAS

FIGURA 1. PLACA MEA DE MULTICHANNEL SYSTEMS [3]	12
FIGURA 2. RESPUESTA DE UN SISTEMA CON OVERSHOOT [7]	15
FIGURA 3. CALENTADOR DE PERFUSIÓN DE MULTICHANNEL SYSTEMS [11]	17
FIGURA 4. CALENTADOR DE PERFUSIÓN DE DIGITIMER [12]	18
FIGURA 5. CALENTADOR DE PERFUSIÓN DE WARNER INSTRUMENTS [13]	19
FIGURA 6. CALENTADOR DE JERINGUILLA DE WARNER INSTRUMENTS [14]	20
FIGURA 7. ANÁLISIS DAFO	23
FIGURA 8. ESQUEMA DE ENTRADAS Y SALIDAS DE LA PLACA ARDUINO UNO [16]	26
FIGURA 9. CURVA DE RESPUESTA DE UN TERMISTOR [17]	29
FIGURA 10. SENSOR DE TEMPERATURA TC74A	30
FIGURA 11. SISTEMA INDUSTRIAL DE CALENTAMIENTO MEDIANTE TECNOLOGÍA INFRARROJA [19]	31
FIGURA 12. CALENTADOR INFRARROJO DE ANILLA [20]	32
FIGURA 13. DIAGRAMA DE FUNCIONAMIENTO PLACA DE PELTIER [21]	33
FIGURA 14. RESULTADOS DE LA PRUEBA DE CALENTAMIENTO DE DOS RESISTENCIAS DE 5Ω EN PARALELO ALIMENTADAS MEDIANTE 12 V. TEMPERATURA AMBIENTE: 25°C	34
FIGURA 15. RESULTADOS DE LA PRUEBA DE CALENTAMIENTO DE DOS RESISTENCIAS DE 5Ω EN SERIE ALIMENTADAS MEDIANTE 12 V. TEMPERATURA AMBIENTE: 29°C	35
FIGURA 16. RESULTADOS PRUEBA DE CALENTAMIENTO DE UNA RESISTENCIA DE 5 Ω ALIMENTADA MEDIANTE 12 V. TEMPERATURA AMBIENTE: 29°C	35
FIGURA 17. CIRCUITO DE ALIMENTACIÓN DE LA PLACA CONTROLADORA	37
FIGURA 18. DRIVER Y MOSFET PARA EL CONTROL DE POTENCIA EN LA RESISTENCIA CALEFACTORA	39
FIGURA 19. EJEMPLO DE LCD KEYPAD SHIELD [27]	40
FIGURA 20. MODELO 3D DE LA RESISTENCIA PARA SIMULACIÓN TÉRMICA	42
FIGURA 21.- PRIMER PROTOTIPO DE CALENTADOR	44
FIGURA 22.- PROCESO DE FABRICACIÓN DEL PRIMER PROTOTIPO	44
FIGURA 23. RESULTADO DE LA PRIMERA SIMULACIÓN TÉRMICA DEL PROTOTIPO 1	46
FIGURA 24. RESULTADO DE LA SEGUNDA SIMULACIÓN TÉRMICA DEL PROTOTIPO 2	47
FIGURA 25. PROTOTIPO NÚMERO 2	48
FIGURA 26.- PROTOTIPO NÚMERO DOS	49
FIGURA 27. RESULTADO DE LA SIMULACIÓN TÉRMICA DEL PROTOTIPO 2	51
FIGURA 28. ESQUEMÁTICO PROTOTIPO 3	53
FIGURA 29. DISEÑO DE SOPORTE PARA LA AGUJA BASADO EN UNA PCB	53
FIGURA 30. PROTOTIPO 2 FABRICADO	54
FIGURA 31. DISEÑO TRIDIMENSIONAL DEL PROTOTIPO 3 PARA LA SIMULACIÓN TÉRMICA	55
FIGURA 32. RESULTADO DE LA SIMULACIÓN TÉRMICA DEL PROTOTIPO 3	56

Índice de Figuras

FIGURA 33. PROTOTIPO NÚMERO 3.....	58
FIGURA 34. PROTOTIPO FINAL FABRICADO	59
FIGURA 35. RESULTADO DE LA SIMULACIÓN TÉRMICA DEL PROTOTIPO FINAL	60
FIGURA 36 . DIAGRAMA DE FLUJO DEL ALGORITMO TODO/NADA	66
FIGURA 37. CÓDIGO DEL ALGORITMO TODO/NADA	67
FIGURA 38. GRÁFICAS DEL RESULTADO DE LA PRUEBA DEL ALGORITMO TODO/NADA CON UN SETPOINT DE 60°C.....	68
FIGURA 39. DIAGRAMA DE BLOQUES DE UN CONTROLADOR PID	69
FIGURA 40. RESULTADOS DE LAS PRUEBAS DEL ALGORITMO PID CON UN SETPOINT DE 80°C.....	71
FIGURA 41. GRÁFICA AMPLIADA UNA VEZ ESTABILIZADA LA TEMPERATURA	72
FIGURA 42. EJEMPLO DE IMPLEMENTACIÓN DEL RELÉ	74
FIGURA 43. SEÑAL DE LA LECTURA DE TEMPERATURA EN LA PRACTICA.....	75
FIGURA 44. SISTEMA SIMPLE DE IDENTIFICACIÓN DE PICOS	76
FIGURA 45. SISTEMA AVANZADO DE DETECCIÓN DE MÁXIMOS Y MÍNIMOS	76
FIGURA 46. EJEMPLO DE PROBLEMA DERIVADO DEL RUIDO EN EL ALGORITMO DE AUTO-TUNE	77
FIGURA 47. SOLUCIÓN AL RUIDO EN EL ALGORITMO DE AUTO-TUNE MEDIANTE LA IMPLEMENTACIÓN DE BANDA DE DISPARO77	
FIGURA 48.- DIAGRAMA DE FLUJO DEL ALGORITMO DE AUTO-TUNE	78
FIGURA 49.- GRAFICA DE EJECUCIÓN DEL ALGORITMO DE AUTO-TUNE SOBRE EL PROTOTIPO FINAL CON 3 CICLOS Y SETPOINT DE 70°C. TEMPERATURA AMBIENTE 25°C	79
FIGURA 50. DIAGRAMA DE MENÚS DEL SISTEMA.....	80
FIGURA 51. PANTALLA PRINCIPAL DEL MENÚ	81
FIGURA 52. PANTALLA 1 DEL MENÚ	81
FIGURA 53. PANTALLA 2 DEL MENÚ	82
FIGURA 54. PANTALLA INTRODUCIR TEMPERATURA DEL MENÚ	82
FIGURA 55. PANTALLA CALIBRACIÓN DEL MENÚ	83
FIGURA 56. PANTALLA INFO DEL MENÚ	83
FIGURA 57. DISEÑO DE LA PLACA DE CONTROL	86
FIGURA 58. MODELO TRIDIMENSIONAL DE LA ELECTRÓNICA	87
FIGURA 59. DISEÑO FINAL DEL CUADRO DE CONTROL	88
FIGURA 60. DISEÑO DE LA CARCASA DE LA UNIDAD CALEFACTORA	89
FIGURA 61. CALCULADORA DE COSTES DE IMPRESIÓN 3D.....	93
FIGURA 62. COSTES DE IMPRESIÓN 3D	93

Índice de Figuras



1. INTRODUCCIÓN

El objeto de este trabajo concierne al diseño, fabricación y validación de un calentador de bajo coste para un equipo de microperfusión. La perfusión es una técnica utilizada en diversos campos científicos que consiste en la introducción de distintas sustancias en cultivos celulares para estudiar su reacción a dichos estímulos [1].

En este caso concreto el equipo de perfusión es utilizado por el Instituto de Investigación, Desarrollo e Innovación en Biotecnología Sanitaria de Elche, en adelante IDiBE, para el estudio del comportamiento de cultivos de células neuronales a distintos estímulos. En el IDiBE estudian estos cultivos utilizando una matriz de electrodos (MEA, Multichannel Systems®, Figura 1) que permite obtener la magnitud de los potenciales eléctricos generados por los canales iónicos de las neuronas al ser estimuladas [2].

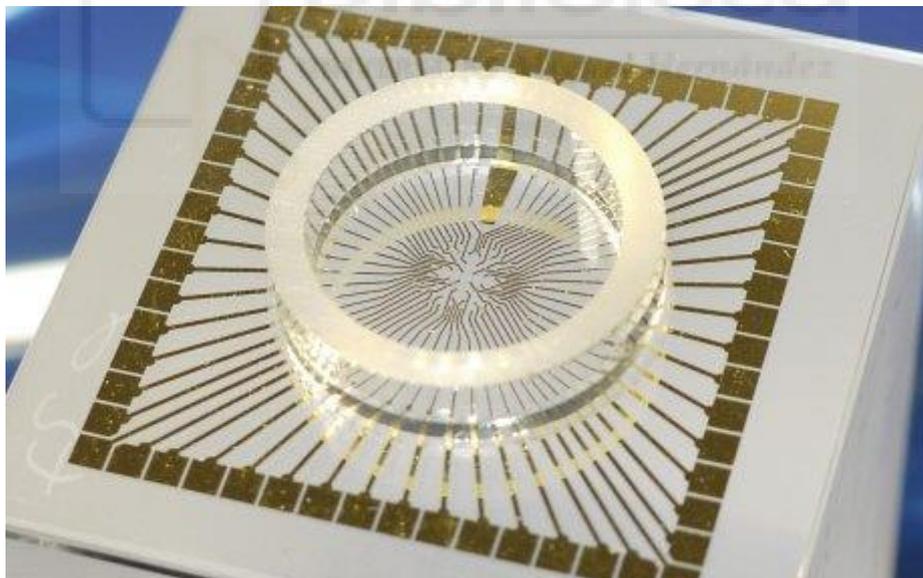


Figura 1. Placa MEA de Multichannel Systems [3]

En concreto, existe una familia de canales iónicos denominados “Receptores de Potencial Transitorio o TRP”, también conocidos como “TermoTRP”, ya que están relacionados con la percepción de variaciones en la temperatura. Cada uno de los canales que integra esta familia es sensible a un rango concreto de temperaturas. El más estudiado es TRPV1 que se activa a temperaturas entre

Introducción

40°C a 50°C [4]. Algunos de estos canales iónicos sufren un proceso de desensibilización al ser sometidos a temperaturas elevadas durante un periodo de tiempo prolongado [5]. Es por esto por lo que todo el proceso de calefacción y medida de los valores estudiados debe producirse de forma rápida para que los datos obtenidos sean válidos.

Actualmente existen en el mercado equipos capaces de adaptar la temperatura del fluido a la requerida para estos estudios. El elevado coste de estos equipos limita su uso de forma extensiva. En este trabajo se pretende desarrollar una alternativa de bajo coste y comprobar si su desempeño es válido para ser empleado en este tipo de aplicaciones.

1.1. OBJETIVOS DEL PROYECTO

El objetivo del trabajo es el planteamiento, desarrollo e implementación de un sistema calentador basado en resistencias para su uso en sistemas de microperfusión. El proyecto contempla el análisis de necesidades del dispositivo, implementación hardware, implementación del sistema de control, calibración de los sensores de temperatura, diseño y fabricación de las envolventes del equipo y validación del prototipo.

1.2. PROPUESTA DE SOLUCIÓN

A través de este proyecto se busca el diseño de un sistema completo capaz de adecuar el fluido a las condiciones de temperatura requeridas y que cuente con una interfaz sencilla para poder ser manejado por personal no experto en electrónica.

El dispositivo contará con dos elementos bien diferenciados. Por un lado, con la unidad de control, interfaz y alimentación. Por otro, con la unidad calefactora.

Introducción

El control de temperatura se realizará mediante un controlador PID¹. Este algoritmo de control está bien indicado para trabajar con sistemas térmicos ya que es muy flexible a la hora de adaptarse a la dinámica de cualquier sistema. Además, se implementará un algoritmo de *auto tune* que servirá para calibrar el control si fuese necesario por un cambio en las condiciones del entorno de trabajo.

Para la interacción con el usuario, a través de la unidad de control, se empleará una pantalla LCD y una botonera. Se buscará que su diseño sea lo más simple e intuitivo posible. En la interfaz, el usuario podrá ver en cada momento todas las variables que le conciernen sin necesidad de preocuparse de lo que esté ocurriendo a nivel interno.

El fluido circula por una aguja hipodérmica, la cual es la encargada de inyectarlo en el MEA. Por ello, es necesario que la unidad calefactora esté preparada para incorporar este elemento.

1.3. ESTUDIO DE MERCADO

La microperfusión es una técnica utilizada en el campo de la biotecnología entre muchos otros que consiste en la introducción de un fluido dentro de un cultivo celular o tejido para estudiar su respuesta ante dicho fluido. Este fluido puede contener diversas sustancias o medicamentos en función de cuál sea el objetivo del estudio. Existen ciertos casos en que, además, es necesario que este fluido se encuentre a una determinada temperatura para provocar las reacciones deseadas. Para ello se utilizan los calentadores de microperfusión, cuya función es la de adecuar la temperatura del fluido a las condiciones requeridas por los investigadores.

Para el correcto funcionamiento de estos dispositivos es necesario que además de alcanzar la temperatura deseada, este proceso se realice de forma precisa y

¹ Un controlador PID es un instrumento utilizado en aplicaciones de control industrial para regular la temperatura, el flujo, la presión, la velocidad y otras variables del proceso. Los controladores PID utilizan un mecanismo de retroalimentación de bucle de control para controlar las variables del proceso. [6]

Introducción

segura. En la mayoría de aplicaciones realizar un calentamiento con *overshoot*²(Figura 2) puede provocar una degradación o destrucción tanto del elemento a estudiar como de las componentes del fluido perfusionado. Además, si la temperatura del sistema presenta oscilaciones sobre el valor deseado, no será posible realizar las observaciones sobre el sistema de manera correcta. Es por esto por lo que en el presente estudio de mercado observaremos como uno de los principales valores ofrecidos por los fabricantes de los dispositivos analizados, es la precisión del calentador. La precisión es la medida en grados centígrados que nos indica cual es la máxima desviación con la que trabaja el sistema.

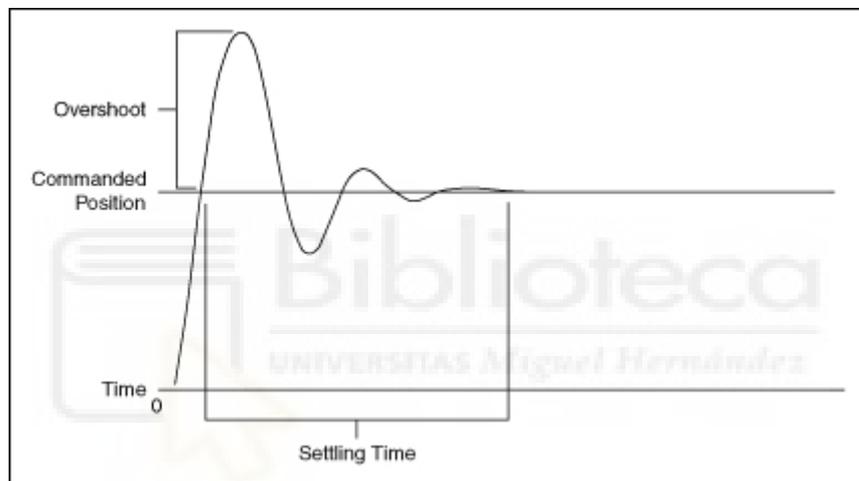


Figura 2. Respuesta de un sistema con *overshoot* [7]

Otro de los puntos clave, y sobre el que menos datos se han encontrado en la realización del estudio, es el tiempo de estabilización, tiempo requerido por el sistema para alcanzar la temperatura deseada de forma estable. Este es un parámetro crucial en ciertas aplicaciones, como la que se ha planteado en la introducción, ya que si sometemos la muestra demasiado tiempo a una temperatura elevada podemos llegar a degradarla [5].

Teniendo todo esto en cuenta, el objetivo del presente estudio de mercado es el de obtener un conocimiento sobre las alternativas disponibles actualmente y

² En un sistema de control se define el *overshoot* como la máxima desviación existente entre valor medido y el valor deseado.

Introducción

qué ofrecen respecto a los aspectos comentados en este capítulo, para poder plantear unos objetivos del proyecto viables y consecuentes con las necesidades de los potenciales usuarios del dispositivo.

1.3.1. DEFINICIÓN DEL PERFIL DE USUARIO

El usuario final de este tipo de sistemas de control de temperatura tiene un perfil científico. Son investigadores, biólogos y técnicos de laboratorio. Los equipos estudiados suelen utilizarse en campos muy diversos y aunque sus usuarios tengan un perfil científico, no tienen por qué disponer de los conocimientos sobre el uso de sistemas electrónicos complejos.

1.3.2. ESTUDIO DE LA OFERTA ACTUAL

Actualmente no muchas empresas disponen de dispositivos dedicados exclusivamente a la microperfusión. Las tres principales empresas que realizan esta tarea son *Multichannel Systems* [8], *Digitimer* [9] y *Warner Instruments* [10].

1.3.2.1. MULTICHANNEL SYSTEMS

La solución que esta empresa ofrece a los calentadores de perfusión es el PH01 (Figura 3), un sistema compacto con una gran precisión.

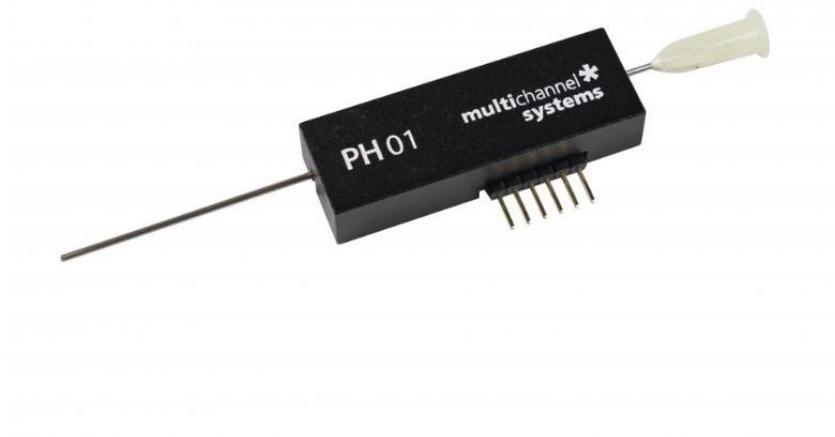


Figura 3. Calentador de perfusión de Multichannel Systems [11]

Las características principales de este dispositivo se describen en la Tabla 1.

Tabla 1. Características del calentador de perfusión de Multichannel Systems [11]

Dimensiones	111 mm x 10.5 mm x 24 mm (con cánula) 48.5 mm x 10.5 mm x 24 mm (sin cánula)
Rango de trabajo	10°C a 40°C
Precisión	+/- 0.1 °C
Resistencia	22 Ω
Tiempo de estabilización	0.5 min a 5 min

Este es el dispositivo del que más información se ha encontrado durante el estudio de mercado. Es el único fabricante que ofrece información respecto al tiempo de estabilización del sistema. Además, es el que más precisión ofrece, aunque cuenta con el rango de trabajo más pequeño de todos.

Introducción

1.3.2.2. DIGITIMER:

El sistema *ThermoClamp Heated Perfusion* de Digitimer (Figura 4) es el dispositivo que ofrece más opciones de los que se han encontrado durante la realización de este estudio. Trabajando en combinación con una unidad de control, es capaz de ajustar sus parámetros de control automáticamente para adecuar su funcionamiento a cada aplicación.



Figura 4. Calentador de perfusión de Digitimer [12]

Las características principales de este dispositivo se describen en la Tabla 2.

Tabla 2. Características del calentador de perfusión de Digitimer [12]

Dimensiones	No especificado
Rango de trabajo	10°C a 50°C
Precisión	+/- 1 °C
Resistencia	No especificado
Tiempo de estabilización	No especificado

Uno de sus puntos más fuertes es la ausencia de componentes metálicos en las zonas por las que discurre el fluido, esto consigue un mejor aislamiento del sistema al ruido. Sin embargo, el fabricante no ofrece ninguna información respecto al tiempo de estabilización.

Introducción

1.3.2.3. WARNER INSTRUMENTS:

Warner Instruments dispone de dos dispositivos distintos, el SH-27B y el SF-28 (Figura 5). Ambos dispositivos cuentan con las mismas propiedades, siendo la única diferencia el caudal de trabajo para el que han sido diseñados. Este caudal es de 10mL/min en el caso del SH-27B y de 2mL/min en el caso del SF-28.



Figura 5. Calentador de perfusión de Warner Instruments [13]

Las características principales de este dispositivo se describen en la Tabla 3.

Tabla 3. Características del calentador de perfusión de Warner Instruments [13]

Dimensiones	12.5 mm x 12.5 cm (D x L)
Rango de trabajo	10°C a 50°C
Precisión	± 0.2°C (según condiciones)
Resistencia	10Ω
Tiempo de estabilización	No especificado
Caudal	10mL/min(SH-27B) 2mL/min(SF-28)

Introducción

Destacan de este equipo su sencillez y facilidad de uso, así como la inclusión de una sonda de temperatura externa para monitorizar la temperatura del MEA sobre el que estemos trabajando.

El fabricante ofrece información adicional a la suministrada por su competencia, especificando el caudal de trabajo de cada dispositivo, lo que puede resultar interesante a la hora de adaptarlo al trabajo que estemos realizando. Al igual que el anterior no ofrece ninguna información acerca del tiempo de estabilización.

1.3.3. ANÁLISIS DE PRODUCTOS SUSTITUTIVOS

Existen en el mercado un tipo de calentadores que no trabajan sobre el conducto de entrada al dispositivo, sino que lo hacen calefactando la jeringuilla que almacena el líquido a utilizar.

Un ejemplo de este tipo de calentadores es la serie SWS (Figura 6) de la empresa *Warner Instruments*. Cuentan con distintas versiones dependiendo de la jeringuilla que se vaya a utilizar en el sistema. Sus características principales se encuentran resumidas en la Tabla 4.



Figura 6. Calentador de jeringuilla de Warner Instruments [14]

Introducción

Tabla 4. Características del calentador de jeringuilla de Warner Instruments [14]

Resistencia	18 Ω
Temperatura máxima	65°C
Precisión	+/-1°C
Voltaje de alimentación	12V

Su principal ventaja frente a un calentador como los expuestos anteriormente, es que el fluido en este caso se encuentra aislado del dispositivo, por lo que puede ser utilizado para varios fluidos sin riesgo de contaminación. Los calentadores en línea requieren ser limpiados para el cambio del fluido, dependiendo de la aplicación. Esta limpieza puede ser complicada en algunos dispositivos en los que el acceso a los canales interiores no es sencillo.

Por el contrario, es necesario disponer de un calentador independiente por cada fluido que vaya a ser utilizado.

1.3.4. ANÁLISIS DE PRECIOS

En este apartado se analizan los precios de los dispositivos presentados en el apartado 1.2.3. Los precios aproximados de los distintos dispositivos se resumen en la Tabla 5.

Tabla 5. Resumen de precios de calentadores de perfusión comerciales

Dispositivo	Precio
PH01 (Multichannel Systems)	1471 € [11]
Thermo Clamp (Digitimer)	1731,28 € [15]
SH-27B (Warner Instruments)	2241,85 [13]

Introducción

Como podemos observar el precio medio de estos dispositivos ronda los 1800€. Este precio en todos los casos incluye la correspondiente unidad de control, necesaria para el funcionamiento del sistema.

Este dato nos sirve como punto de partida para estimar cuánto debería costar nuestro dispositivo. Teniendo en cuenta que el objetivo del trabajo es la fabricación de un producto de bajo coste que pueda cumplir con los requisitos de los investigadores en fases incipientes del estudio, no debería exceder en ningún caso una cuarta parte del coste de cualquiera de los dispositivos analizados en este capítulo.

1.3.5. CONCLUSIONES DEL ESTUDIO DE MERCADO

El presente estudio ha servido para conocer cuáles son las características del mercado en el que estamos trabajando. Nos encontramos ante un producto de nicho, lo que significa precios altos y baja disponibilidad.

En todos los productos analizados encontramos sistemas completos que incluyen el dispositivo y la unidad de control. Esto hace que aumente su facilidad de uso de cara al usuario siempre y cuando trabajes dentro del ecosistema de cada fabricante. En ninguno de los casos se ha encontrado que las empresas ofrezcan piezas de repuesto, lo que hace suponer que en caso de rotura será necesario sustituir el producto completo.

En nuestro caso el planteamiento del producto es y debe ser distinto al de las empresas que comercializan los productos analizados. El producto que se desarrolle en este trabajo debe poder cumplir con las especificaciones mínimas, pero suficientes, para fases iniciales de trabajos de investigación. El objetivo no es fabricar un producto que pueda competir en este mercado, ya que esto es inviable debido a factores técnicos y económicos, sino servir para que el usuario pueda decidir si necesita realizar la inversión en un equipo profesional o no.

1.3.6. ANÁLISIS DAFO

Podemos resumir la situación y el encaje del proyecto en el mercado mediante el análisis DAFO mostrado en la Figura 7.

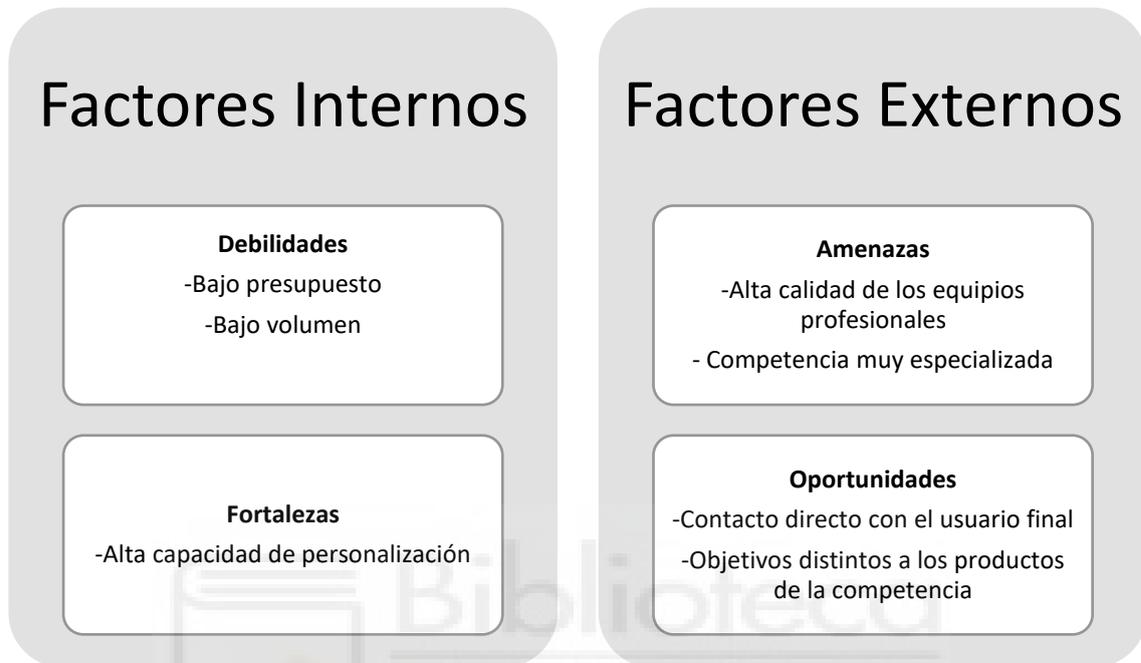


Figura 7. Análisis DAFO

1.4. REQUISITOS DEL PROYECTO

Con todo lo analizado en el presente capítulo podemos enumerar todos los requisitos que ha de cumplir el dispositivo para considerarse apto para su uso en laboratorio a nivel profesional:

- El sistema debe ser de fácil reproducción
- El sistema debe incorporar el elemento conductor del fluido cedido por el IDiBE
- El sistema debe contar con una interfaz sencilla
- El calentamiento debe producirse en un tiempo aproximado de un minuto.

Introducción

- El sistema debe proporcionar lecturas precisas de la temperatura de $\pm 1^{\circ}\text{C}$.
- El sistema debe realizar un control preciso de la temperatura de $\pm 1^{\circ}\text{C}$.
- Sistema que no introduzca ruido eléctrico en el fluido a calentar.
- Rango de calentamiento: $25^{\circ}\text{C} - 70^{\circ}\text{C}$
- Sistema fácil de utilizar incluso para personas no expertas en electrónica.
- Planificación de desarrollo basada en posibilidad de reparaciones, etc.
- Coste de fabricación inferior a 400€

1.5. DISTRIBUCIÓN DE LA MEMORIA

La memoria de este trabajo está distribuida de la siguiente manera:

- Capítulo 1: Introducción

En este capítulo se plantean los objetivos del trabajo y se realiza un estudio de mercado para conocer las alternativas y poder ofrecer una solución mejor.

- Capítulo 2: Diseño de Hardware

El segundo capítulo está dedicado al diseño de los componentes físicos del calentador de microperfusión. En este capítulo se explican cuáles han sido los criterios seguidos a la hora de elegir cada componente del sistema. Además, se explica en detalle cuáles han sido los procedimientos empleados en la valoración de dichos componentes.

- Capítulo 3: Diseño de Software

Este capítulo se centra en el código que hace que todo el sistema funcione correctamente. Por un lado, se analizan cuáles son las librerías y como se estructura el código que vamos a utilizar. Por otro, se explican qué algoritmos de control hemos implementado, cómo y cuál ha sido su rendimiento.

Introducción

- Capítulo 4: Interfaz de usuario

El cuarto capítulo trata sobre cómo está distribuido el sistema de menús, cuáles son las opciones disponibles para el usuario y cómo se ha diseñado pensando en un uso sencillo para personal no experto.

- Capítulo 5: Fabricación del sistema final

En este capítulo se muestra el procedimiento seguido para la fabricación del dispositivo final. En este proceso se han seguido diversas técnicas como la mecanización, la fabricación de PCB o la impresión 3D. Además, contiene un presupuesto detallado de los materiales utilizados y la mano de obra, así como un escalado en función de la cantidad.

- Capítulo 6: Pruebas en entorno real

En este capítulo se realiza la validación del sistema mediante la realización de distintas pruebas que simulan el comportamiento del equipo en un entorno real.

- Capítulo 7: Conclusiones

Con el sistema ya validado, este capítulo se centra en resumir todo el trabajo y analizar si se han cumplido con los objetivos propuestos.

- Capítulo 8: Mejoras y futuras líneas de desarrollo.

Por último, se analizan las posibles mejoras que podrían realizarse en el sistema más allá de este trabajo.

2. Diseño de Hardware

2. DISEÑO DE HARDWARE

En el siguiente apartado se describen los elementos hardware de los que consta el sistema. El esquema final de la electrónica puede encontrarse en el anexo 10.3.

2.1. PLACA DE DESARROLLO ARDUINO UNO

Las placas de desarrollo Arduino son sistemas embebidos que cuentan con todos los elementos necesarios para el prototipado y desarrollo de sistemas electrónicos que interactúen con el mundo real (Figura 8).

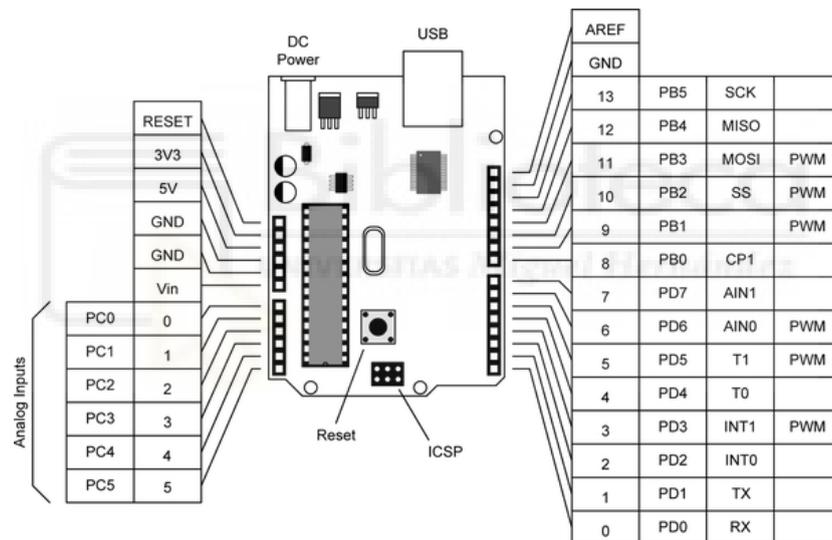


Figura 8. Esquema de entradas y salidas de la placa Arduino Uno [16]

En el caso de la placa Arduino Uno está basado en el microcontrolador ATmega328. Sus características más destacadas son las siguientes (Tabla 6):

2. Diseño de Hardware

Tabla 6. Características de la placa Arduino Uno [16]

Microcontrolador	ATmega328
Tensión de funcionamiento	5V
Voltaje de alimentación	7-12V
Pines Digitales	14(6pwm)
Pines Analógicos	6
Memoria Flash	32KB
SRAM	2KB
EEPROM	1KB
Velocidad de reloj	16MHz

La elección de la placa Arduino Uno viene motivada por varios factores. Primero por la cantidad de recursos disponibles para su programación. Este hecho ayuda en gran medida a reducir el tiempo de desarrollo del sistema.

Además, gracias a utilizar Arduino toda la filosofía del proyecto se basa en la utilización de *Shields*, módulos aislados y apilables uno encima de otro. Esto ayuda al cumplimiento de uno de los objetivos del proyecto, la facilidad de reparación. Al ser un producto basado en módulos independientes es muy sencillo sustituir uno de ellos en caso de rotura sin que afecte al sistema.

2.2. SENSOR DE TEMPERATURA

La medición de temperatura es uno de los aspectos críticos para el correcto funcionamiento del sistema. Si no trabajamos con una media precisa el algoritmo de PID puede no funcionar como se espera.

2. Diseño de Hardware

Se han valorado tres tipos de sensores de temperatura diferentes para la realización de este trabajo. A continuación, se expondrán las tres tipologías, explicando sus ventajas e inconvenientes y la justificación del sensor elegido.

2.2.1. SENSOR RTD

Los sensores RTD funcionan mediante la variación de la resistencia interna debido a cambios de temperatura. Estos sensores se utilizan generalmente para aplicaciones de baja temperatura como esta.

El principal inconveniente de estos sensores para su uso con Arduino es que este no es capaz de detectar el cambio de voltaje en el sensor cada vez que varía la temperatura. Esto es debido a que estos cambios son de un orden de magnitud inferior al que Arduino puede detectar. Si bien es cierto que esto es un problema salvable utilizando circuitos que adapten la señal, como un puente de Wheatstone, esto supondría que de ser necesaria su sustitución por otro sensor RTD sería necesario cambiar las resistencias del puente de Wheatstone para adaptarlas al nuevo sensor. Podría utilizarse un puente de Wheatstone con una resistencia variable, pero la calibración no está al alcance de un usuario no experimentado.

2.2.2. TERMISTOR

Los termistores son sensores de temperatura basados en semiconductores. La temperatura también se mide en función a la variación de la resistencia, pero este proceso ocurre de forma distinta. En los termistores es la variación en la concentración de los portadores la que provoca el cambio en la resistencia del componente.

Los termistores trabajan en un rango menor de temperaturas que los sensores RTD, pero aun suficiente para cumplir con los requisitos del proyecto. Los precios de los termistores también son de media más bajos que los de los sensores RTD, lo que encaja con la filosofía de bajo coste del dispositivo.

2. Diseño de Hardware

El principal problema de los termistores es que no son lineales tal y como puede verse en su curva de respuesta (Figura 9). Esto no presenta un inconveniente a la hora de leer la temperatura, ya que los fabricantes disponen de modelos matemáticos que permiten leer con precisión la temperatura con estos sensores. El motivo por el que este tipo de sensores ha sido descartado para este trabajo es que, en caso de rotura, si no se sustituye por el mismo sensor exacto, sería necesario realizar cambios en el código para adaptarlo a la curva de respuesta de este nuevo sensor. En cambio, si utilizamos un sensor digital que funcione mediante el protocolo I2C como los que se describen en el siguiente capítulo, solo sería necesario actualizar la dirección de lectura en el código.

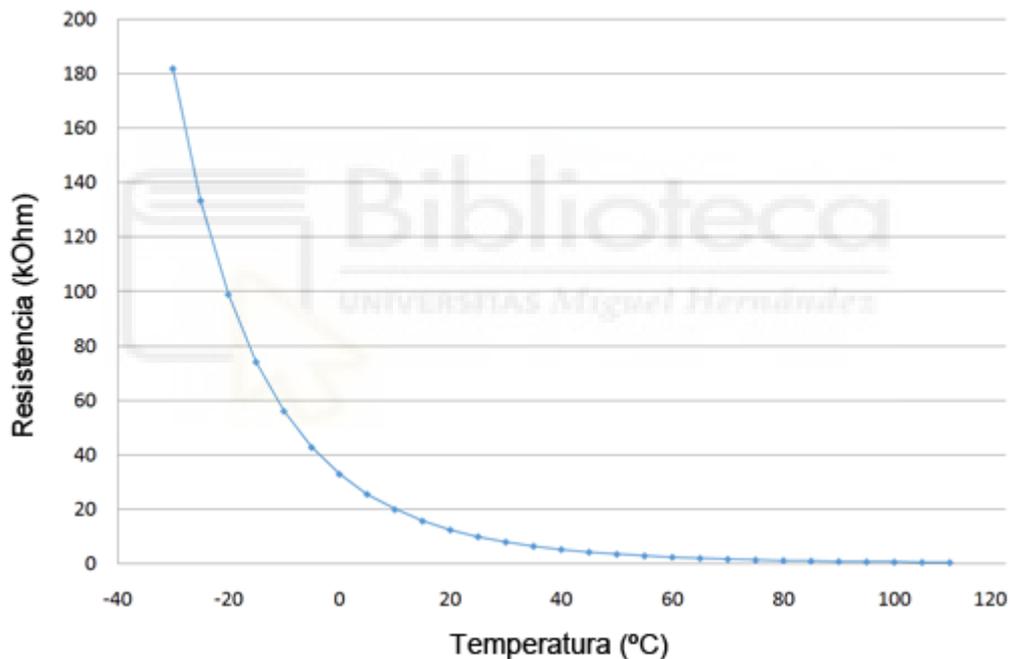


Figura 9. Curva de respuesta de un termistor [17]

2.2.3. SENSORES DIGITALES DE TEMPERATURA

Los sensores digitales de temperatura son dispositivos que incorporan, además del sensor de temperatura, el circuito de control que adapta la señal facilitando su lectura. De esta forma evitamos el uso de componentes extra y disminuimos la carga de trabajo del microcontrolador.

2. Diseño de Hardware

Es por ello por lo que a la hora de elegir un sensor de temperatura se ha optado por un sensor digital y con un envío de datos por segundo adecuado para la aplicación. El sensor elegido es el TC74A con una resolución de 1°C, que realiza la conversión de la temperatura en el integrado y la envía mediante I2C convertida en una palabra de 8 bits. Esta precisión es suficiente para cumplir con los requisitos del proyecto.

Además, su encapsulado (TO-220) permite un contacto directo con el disipador de calor, lo que minimiza errores de lectura (Figura 10). El contacto directo del encapsulado con el dispositivo hace que este quede conectado a la tierra del circuito digital. Esto puede producir ruido en las lecturas del sensor de temperatura, por lo que es necesario aislarlo eléctricamente, por ejemplo, con mica.



Figura 10. Sensor de temperatura TC74A

Para trabajar con I2C es necesario utilizar dos resistencias de *pull-up*. Estas resistencias mantienen el nivel alto de las líneas del bus y su uso asegura que el cambio de estado se produzca dentro del tiempo establecido en el protocolo [18].

Para su uso con Arduino pueden utilizarse las resistencias de *pull-up* internas de las que dispone el dispositivo. En este dispositivo se ha preferido utilizar resistencias externas ya que las que proporciona Arduino son de un valor

2. Diseño de Hardware

pequeño, lo que supone que la velocidad de comunicación es menor y que la distancia de trabajo disminuye.

2.3. ELEMENTOS CALEFACTORES

Los elementos calefactores son los encargados de suministrar energía al sistema para, en conjunto con el sensor de temperatura, controlar la temperatura del sistema.

Se han valorado tres tipos de elementos calefactores diferentes para la realización de este trabajo. A continuación, se expondrán las tres tipologías, explicando sus ventajas e inconvenientes y la justificación del elemento elegido.

2.3.1. CALENTAMIENTO MEDIANTE INFRAROJOS

La tecnología de calentamiento por infrarrojos funciona mediante la transmisión de ondas electromagnéticas. Esta tecnología se utiliza en la industria para trabajos de secado de barnices y pinturas o el calentamiento controlado de elementos plásticos o vidrio.

Sus principales ventajas son la aplicación de calor sin contacto, un tiempo de reacción corto, una gran precisión y la capacidad de adaptarse a multitud de sistemas.

En la industria suelen encontrarse en dos formatos, lámparas (Figura 11) y calentadores de anilla (Figura 12).



Figura 11. Sistema industrial de calentamiento mediante tecnología infrarroja [19]

2. Diseño de Hardware

Para este dispositivo el más indicado sería el de anilla. Lamentablemente no es común encontrarlo en un formato lo suficientemente pequeño como para poder utilizarlo en esta aplicación. Es por ello por lo que esta opción ha sido descartada.



Figura 12. Calentador infrarrojo de anilla [20]

2.3.2. CELULA DE PELTIER

Las células de Peltier (Figura 13) son dispositivos de calentamiento basados en semiconductores que funcionan mediante el efecto termoeléctrico. En concreto, funciona mediante el efecto Peltier que describe como en un circuito cerrado formado por metales diferentes aparece una diferencia de temperatura al hacer circular corriente por él. Esto se debe a que al circular electrones entre regiones con distinta densidad de carga se comportan como un gas ideal de manera que se expanden enfriando la región de baja densidad. Esto implica que siempre que se de el efecto Peltier se crearán dos zonas distintas, una fría que será la región de menor densidad y una caliente, que será la de mayor densidad. Este efecto podría ser aprovechado en el trabajo de forma que usáramos la zona caliente para calentar el dispositivo y la fría cuando necesitáramos disminuir la temperatura.

2. Diseño de Hardware

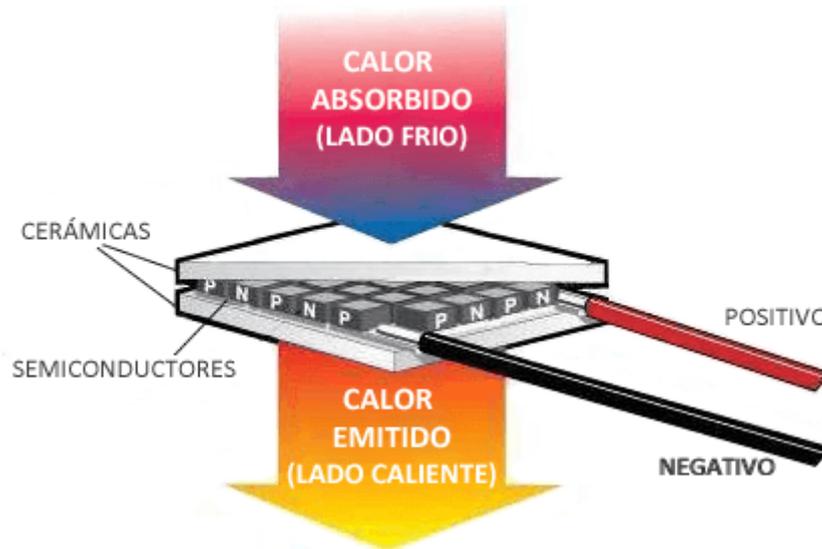


Figura 13. Diagrama de funcionamiento placa de Peltier [21]

Su principal desventaja y el motivo por el que han sido descartadas para este trabajo es su baja eficiencia, en torno al 20-25% [21]. Esto unido a las pérdidas en la transmisión de calor hacia el soporte de la aguja lo hace inviable para un sistema de baja potencia como este.

2.3.3. RESISTENCIAS CALEFACTORAS

Las resistencias calefactoras funcionan haciendo uso del efecto Joule. El efecto Joule es un fenómeno por el cual, si por un conductor circula corriente eléctrica, parte de la energía cinética de los electrones se transforma en calor debido a los choques que estos sufren entre sí mismos y con los átomos del elemento conductor.

Existe una gran variedad de formatos en los que podemos encontrar las resistencias calefactoras. Para este proyecto, al igual que en el capítulo 2.2.2.3, se ha elegido una resistencia calefactora encapsulada en el formato TO-220 por considerar que es más sencillo para su instalación en superficies planas como las que vamos a utilizar y nos garantiza el máximo contacto aumentando la eficiencia del sistema. En concreto se ha elegido el modelo PF2205 de la empresa Riedon [22].

2. Diseño de Hardware

2.3.3.1 SELECCIÓN Y CONFIGURACIÓN DE LAS RESISTENCIAS CALEFACTORAS

Todas las pruebas realizadas con las resistencias calefactoras se han realizado midiendo la temperatura alcanzada sobre una placa de cobre de 75mm x 20mm x 4mm. Esta placa de cobre ha sido la que finalmente se ha utilizado en el capítulo 2.2.7.4 para la fabricación del prototipo final.

En una primera aproximación se consideró el uso de dos resistencias de 5Ω conectadas en paralelo y alimentadas mediante 12V. Las pruebas con este circuito han resultado en una velocidad de calentamiento de $2,42^{\circ}\text{C/s}$ con un consumo de 4,8A y una potencia de 57,6W (Figura 14).

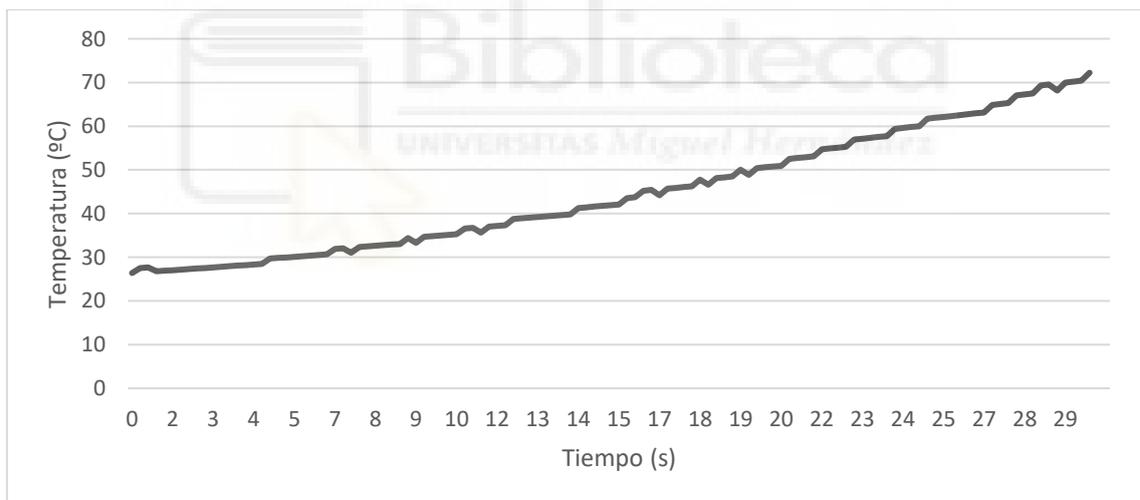


Figura 14. Resultados de la prueba de calentamiento de dos resistencias de 5Ω en paralelo alimentadas mediante 12 V. Temperatura ambiente: 25°C

Para la segunda prueba la configuración elegida es de dos resistencias de 5Ω conectadas en serie y alimentadas mediante 12V. Las pruebas con este circuito han resultado en una velocidad de calentamiento de $0,5^{\circ}\text{C/s}$ con un consumo de 1,09A y una potencia de 11,881 W (Figura 15).

2. Diseño de Hardware

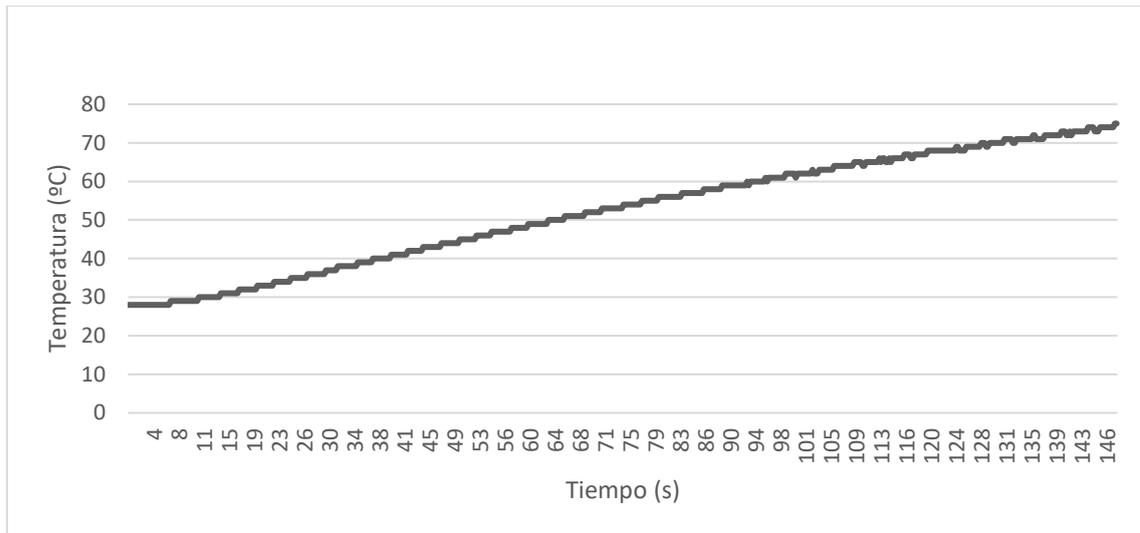


Figura 15. Resultados de la prueba de calentamiento de dos resistencias de 5Ω en serie alimentadas mediante 12 V. Temperatura ambiente: 29°C

La tercera y última configuración probada es de una sola resistencia de 5Ω alimentada mediante 12V. Las pruebas con este circuito han resultado en una velocidad de calentamiento de 1°C/s con un consumo de 2,06A y una potencia de 21,21W (Figura 16).

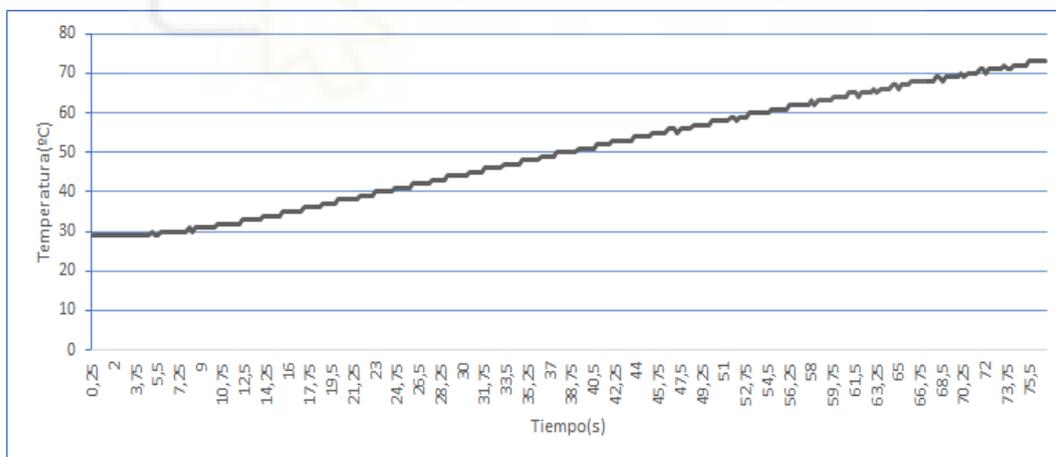


Figura 16. Resultados prueba de calentamiento de una resistencia de 5Ω alimentada mediante 12 V. Temperatura ambiente: 29°C

Como criterio para la selección de configuración final nos basamos en los requisitos del proyecto, establecidos en base a las necesidades de los experimentos del IDiBE. Es decir, que buscamos un sistema capaz de pasar de

2. Diseño de Hardware

25°C a 70°C en menos de un minuto. Las conclusiones de todas estas pruebas se resumen en la Tabla 7.

Tabla 7. Resumen de las pruebas de configuración de resistencias

Configuración	Intensidad (A)	Potencia (W)	Velocidad (°C/s)	Tiempo de establecimiento (s)
2 resistencias Paralelo	4,8	57,6	2,42	18,59
2 resistencias Serie	1,09	11,88	0,5	90
1 Resistencia	2,06	21,21	1	45

La conclusión que obtenemos de las pruebas realizadas es que tanto la configuración de dos resistencias en paralelo como la de una sola resistencia cumplen con el criterio temporal. La principal diferencia es el consumo. Dado que 4,8A es un valor superior a la potencia disponible dada por la fuente de alimentación, la elección final es utilizar una sola resistencia de 5Ω.

Para el modelo final sin embargo no es posible utilizar una sola resistencia debido a que durante los tiempos de calentamiento esta no puede soportar toda la potencia. Por ello se han utilizado 4 resistencias de 5Ω en dos grupos en paralelo de dos resistencias, el circuito eléctrico completo se encuentra en el Anexo 10.3. La resistencia final sigue siendo la misma, pero de esta forma repartimos la potencia en las cuatro resistencias para evitar que se dañen.

2.4. ALIMENTACIÓN

La alimentación eléctrica del dispositivo vendrá dada por la red eléctrica a través de una fuente de alimentación externa que proporciona 12V y 3A, una potencia total de 36W. La elección de una fuente de alimentación externa viene motivada por la reducción del tamaño del dispositivo y la reducción del calor

2. Diseño de Hardware

generado dentro del dispositivo. La conexión física con el circuito se realiza mediante una clavija *Jack* de 5.5mm de diámetro. El uso de este tamaño de clavija está muy extendido por lo que en caso de ser necesaria su reposición no supondría ningún problema.

En el diseño de nuestro circuito hemos manejado dos voltajes distintos, 12V y 5V. Para que el dispositivo pueda disponer de ambos voltajes es necesario una etapa de adaptación de voltaje que suministre 5V a los elementos que lo requieran. Para ello primero se barajó la utilización de un convertidor *DC/DC* lineal, pero se descartó por su baja eficiencia.

Una posible solución al problema de calentamiento es utilizar un convertidor *DC/DC* conmutado. Estos dispositivos tienen una eficiencia de conversión de energía mayor que los lineales. Además, debido a la alta frecuencia de conmutación, los componentes pasivos son de tamaño reducido lo que evita pérdidas en forma de calor [23]. Concretamente se utiliza el R-78E5.0-1.0 [24] que proporciona una salida de 5V y es capaz de suministrar hasta 5W.

Al circuito de alimentación se le han añadido dos condensares para lidiar con los picos de corriente y las distorsiones según las recomendaciones del fabricante [24]. El circuito final se muestra en la Figura 17.

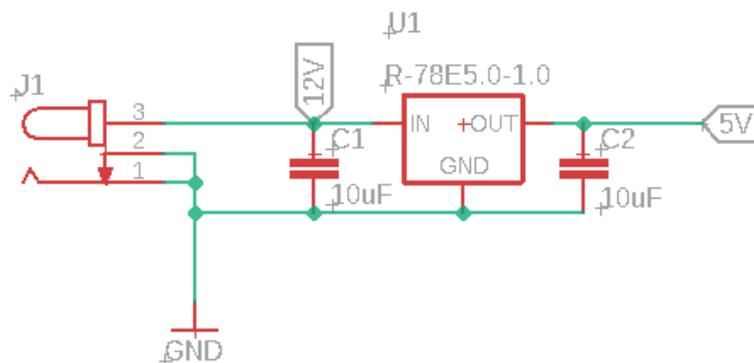


Figura 17. Circuito de alimentación de la placa controladora

2.5. DRIVERS

La tensión e intensidad necesaria para trabajar con la resistencia calefactora son superiores a las que una placa Arduino puede manejar. Para poder controlar la resistencia calefactora mediante señales del microcontrolador utilizaremos un MOSFET. Concretamente se ha utilizado el transistor MOSFET de canal N IRF510.

Aun así, las salidas digitales de la placa Arduino no son capaces de trabajar el voltaje necesario para controlar el MOSFET. Es necesario añadir un *driver*³ que maneje el transistor. Se ha optado por implementar un circuito *totem-pole*.

Un circuito totem-pole se utiliza para aumentar la carga y la velocidad de conmutación con la que puede trabajar una salida digital. Trabaja utilizando un par de transistores, uno encargado de conectar la puerta del MOSFET con la alimentación y otro para descargar la puerta del MOSFET rápidamente cuando la señal de control sea cero [26]. El hecho de utilizar este circuito y no un solo transistor es debido al comportamiento capacitivo del MOSFET. Cuando la entrada del MOSFET es puesta a nivel alto se almacena carga eléctrica en ella. Cuando queremos dar un nivel bajo debemos descargarla lo más rápido posible. Si no contamos con una forma correcta de descargar la puerta del MOSFET, aunque el circuito de control haya dado un nivel bajo, el elemento a controlar seguirá a un nivel alto hasta que la puerta del MOSFET no se descargue completamente. Utilizando un segundo transistor permitimos que el MOSFET pueda descargarse de manera rápida conectando la puerta a tierra. Para implementar este circuito se ha utilizado un transistor PNP 2N2907 y un transistor PNP2N2222. La configuración final del hardware de control de la resistencia puede verse en la Figura 18.

³ En electrónica, un **driver** o **controlador** es un circuito eléctrico u otro componente electrónico utilizado para controlar otro circuito o componente. [25]

2. Diseño de Hardware

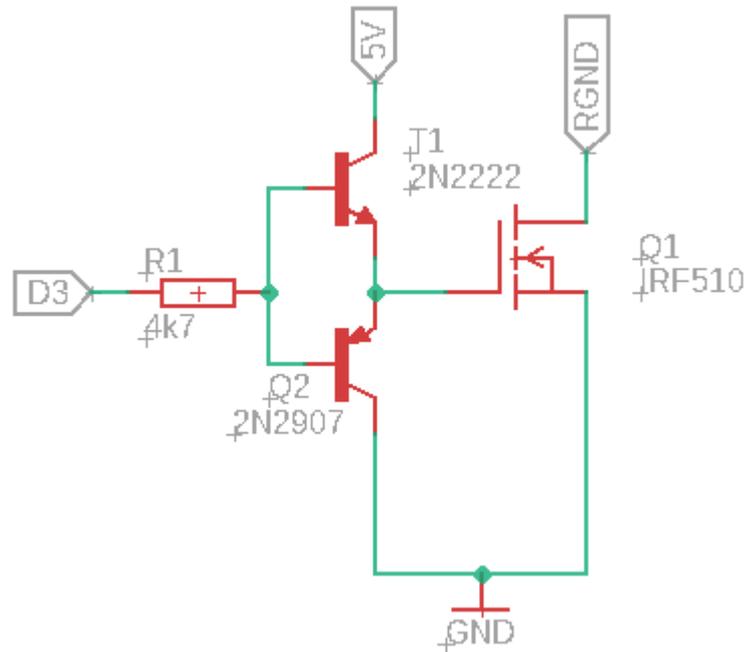


Figura 18. Driver y MOSFET para el control de potencia en la resistencia calefactora

2.6. PANTALLA

Toda la interacción con el dispositivo será realizada mediante una pantalla y una botonera. Para implementar la pantalla y la entrada de datos por parte del usuario se ha utilizado un *LCD keypad Shield* (Figura 19). Estos *Shields* se basan en un diseño *Open-Source* que fabrican y comercializan muchas empresas. La ventaja en el uso de esta *Shield* como interfaz es la facilidad para encontrar un recambio si fuese necesario. Además, cuentan con una gran variedad de recursos libres para su programación.

2. Diseño de Hardware



Figura 19. Ejemplo de LCD Keypad Shield [27]

La Shield utilizada incluye una pantalla LCD de 16x2, 5 botones y toda la electrónica necesaria para controlarlos. Los pines de Arduino necesarios para hacer funcionar esta *Shield* se encuentran en la Tabla 8.

Tabla 8. Pines de conexión del keypad lcd Shield

Pin	Función
Analógico 0	Botones
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS
Digital 9	Enable
Digital 10	Control de luz trasera

Con esta configuración nos quedan libres el resto de pines analógicos y los pines digitales del 0 al 3. Estos pines son más que suficientes para la implementación del sistema.

2. Diseño de Hardware

Para el manejo de la botonera los 5 botones están conectadas mediante resistencias a una única entrada analógica de la placa Arduino. De esta forma leyendo el valor en este pin podemos conocer qué botón se ha pulsado.

2.7. SOPORTE DE LA AGUJA

La aguja es el elemento principal del sistema. Su cometido es introducir el fluido caliente dentro del MEA. En nuestro caso, la aguja ha sido suministrada por el IDiBE. El soporte de la aguja es el elemento encargado de ser el soporte físico del sistema calefactor y del elemento de circulación del fluido a calentar. Además, tiene el objetivo de dotar de inercia térmica al sistema para que sea capaz de mantener la temperatura de forma estable.

Para su diseño se han partido de las siguientes premisas:

- El volumen del componente debe contar con el tamaño suficiente como para ser capaz de almacenar la energía necesaria para mantener la temperatura de forma estable, pero no ser demasiado voluminoso como para requerir de más energía de la que le puede ser suministrada para ser calentado.
- Fácil de fabricar y de reproducir.
- Estabilidad térmica.

Hasta la fabricación del soporte final se han probado cuatro prototipos distintos que se detallarán a continuación. Los planos detallados de cada prototipo pueden encontrarse en el anexo 10.2

2.7.1. VALIDACIÓN TÉRMICA DE LOS PROTOTIPOS

Para poder validar los prototipos en base a los criterios establecidos en el capítulo anterior, se han realizado diversos análisis térmicos bajo distintas condiciones de trabajo utilizando la herramienta de simulación térmica del

2. Diseño de Hardware

programa Autodesk Fusion 360 [28]. Esta herramienta nos permite basarnos en los modelos 3D de cada uno de los prototipos y simular su comportamiento ante una fuente de calor, en nuestro caso las resistencias calefactoras. El objetivo principal de estas simulaciones es comprobar si los prototipos cuentan con la suficiente inercia térmica como para alcanzar y mantener la temperatura deseada establecida en los requisitos del proyecto en 70°C. Además, servirán para conocer si existe un peligro potencial de dañar las resistencias térmicas en el caso de que la temperatura final del sistema, bajo un calentamiento sin control, supere la temperatura de trabajo. En el caso de la resistencia PF2205 este valor es de 175°C [22].

Para efectuar los análisis se ha creado un modelo en 3D de las resistencias a las que se le han asignado las propiedades térmicas de los materiales de los que están fabricado, cobre para las partes metálicas y resina epoxi para los encapsulados (Figura 20). La ventaja de utilizar esta herramienta reside en que todos los parámetros térmicos, tanto de las resistencias como de cada uno de los prototipos, son calculados automáticamente en base al material asignado y a la geometría de la pieza.

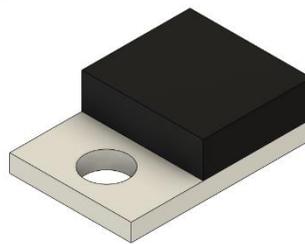


Figura 20. Modelo 3D de la resistencia para simulación térmica

Este modelo también se ha utilizado para simular el sensor de temperatura, ya que este comparte encapsulado. Utilizaremos los estudios térmicos para conocer la diferencia de temperatura existente entre el sensor y el punto medio de la aguja. Conocer este valor resulta de gran utilidad ya que permite conocer posibles desviaciones entre la temperatura real y la medida y tenerlas en cuenta a la hora de implementar el control de temperatura.

2. Diseño de Hardware

Cada uno de los siguientes apartados, dedicados a los distintos prototipos, contará con un subapartado donde se mostrarán y explicarán las simulaciones térmicas realizadas sobre dicho prototipo. En todas las simulaciones se han considerado las resistencias térmicas como una fuente de calor, cuya potencia dependerá de su configuración electrónica, y el soporte de la aguja como un sumidero térmico que transmite calor hacia el ambiente mediante convección y radiación. Las propiedades físicas de los materiales utilizados para la simulación térmica pueden encontrarse en el anexo 10.4.

2.7.2. PROTOTIPO 1

A la hora de diseñar un soporte para la aguja, la primera idea ha sido la de adaptar un disipador de temperatura. La elección de un disipador viene motivada por varios factores. El primero, es que es un elemento de bajo coste y de fácil adquisición, por lo que podría ser fácil de encontrar para posibles investigadores que quisieran recrear el dispositivo en su centro de trabajo. Además, los disipadores son elementos diseñados para absorber el calor generado por un componente electrónico, que en nuestro caso son las resistencias calefactoras. Otro motivo para elegir un disipador es que en una primera fase del proyecto se barajó la posibilidad de incorporar un ventilador que ayudara con el control de temperatura y los disipadores están pensados para transferir el calor al aire mediante convección utilizando ventilación forzada.

Para poder utilizar el disipador, se ha pensado en diseñar una envolvente que contuviera todos los elementos del sistema. Esta envolvente (Figura 21) consta de dos piezas de aluminio unidas por tornillería. En la parte superior se sitúa un ventilador y en la parte media un disipador de temperatura. El uso del ventilador ha sido descartado en posteriores versiones del sistema debido a que se ha considerado que con los nuevos prototipos no es necesario para un control de temperatura óptimo. Los elementos de transmisión de calor se sitúan en la parte de abajo, así como el sensor de temperatura. La aguja se sitúa en un agujero transversal en la pieza de abajo.

2. Diseño de Hardware

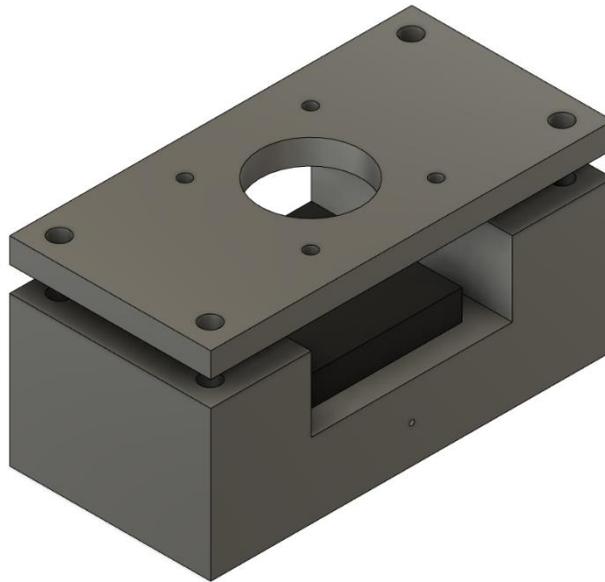


Figura 21.- Primer prototipo de calentador

Este prototipo ha sido descartado en las primeras pruebas debido a su volumen, lo que hace muy difícil de calentar y enfriar con rapidez el sistema tal y como se demuestra en el siguiente apartado. Además, el agujero en el que se sitúa la aguja complica su fabricación (Figura 22) con los medios disponibles, ya que es pequeño y de mucha longitud. Esto hace que las brocas empleadas en su fabricación se rompan.



Figura 22.- Proceso de fabricación del primer prototipo

2. Diseño de Hardware

2.7.2.1. SIMULACIÓN TÉRMICA DEL PROTOTIPO 1

Para la simulación térmica de este prototipo se han utilizado dos configuraciones de resistencias calefactoras distintas utilizando el mismo material para el prototipo en ambas, el aluminio. Los parámetros de las simulaciones pueden observarse en la Tabla 9.

Tabla 9. Tabla de parámetros de la simulación térmica del primer prototipo

Propiedad	Valor
Temperatura ambiente	25°C
Coeficiente de transmisión de calor por convección del soporte	10 W/m ² K [29]
Emisividad del soporte	0.05 [30]

En la primera se han configurado tal y como se diseñaron en el capítulo 2.2.3.3.1, dos grupos en paralelo de dos resistencias de 5Ω en serie. Con esta configuración cada resistencia desarrolla una potencia de 7W. Los resultados de la simulación pueden verse en la Figura 23

2. Diseño de Hardware

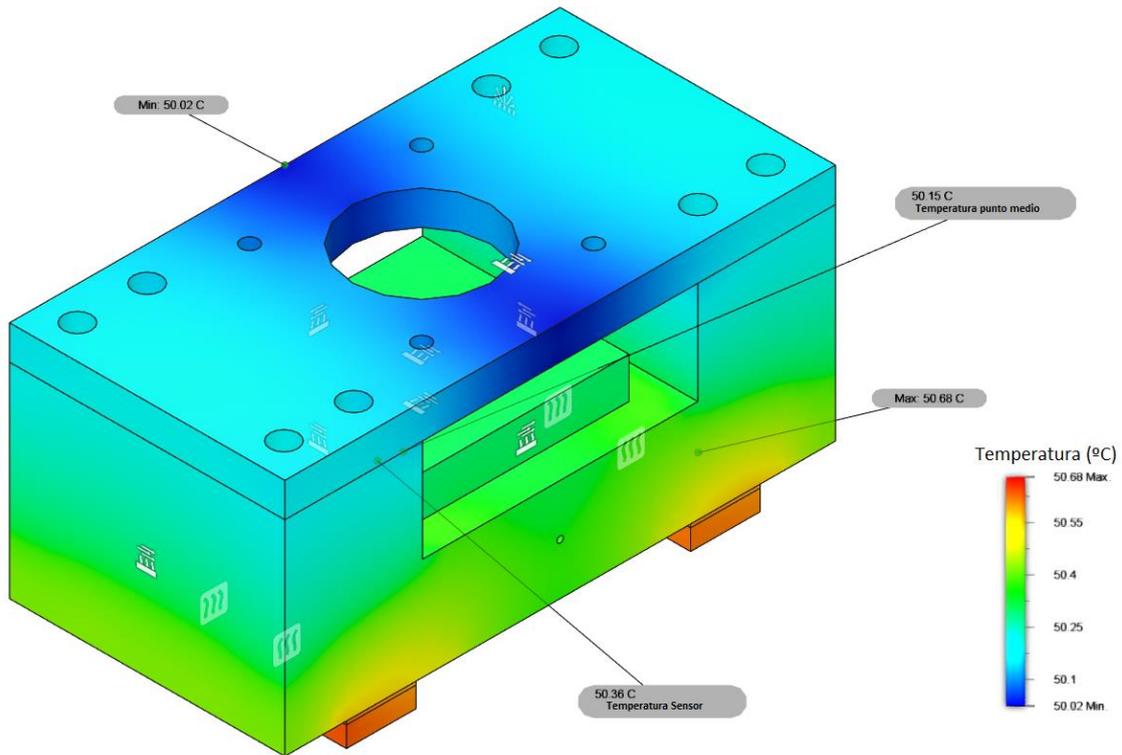


Figura 23. Resultado de la primera simulación térmica del prototipo 1

Tal y como puede observarse en la simulación la energía proporcionada por las resistencias es insuficiente para calentar el sistema hasta la temperatura de trabajo. Esto es debido principalmente al gran volumen del prototipo. Un resumen de las temperaturas obtenidas de la simulación puede encontrarse en la Tabla 10

Tabla 10. Temperaturas obtenidas de la primera simulación del prototipo 1

Propiedad	Valor
Temperatura Máxima	50,68°C
Temperatura Mínima	50,02°C
Temperatura en el sensor	50,36°C
Temperatura en el punto medio de la aguja	50,15°C

Antes de descartar este prototipo se ha realizado una segunda simulación térmica. En esta se han configurado las cuatro resistencias en paralelo de forma

2. Diseño de Hardware

que cada una desarrolla una potencia de 28 W. El resultado de esta simulación puede verse en la Figura 24.

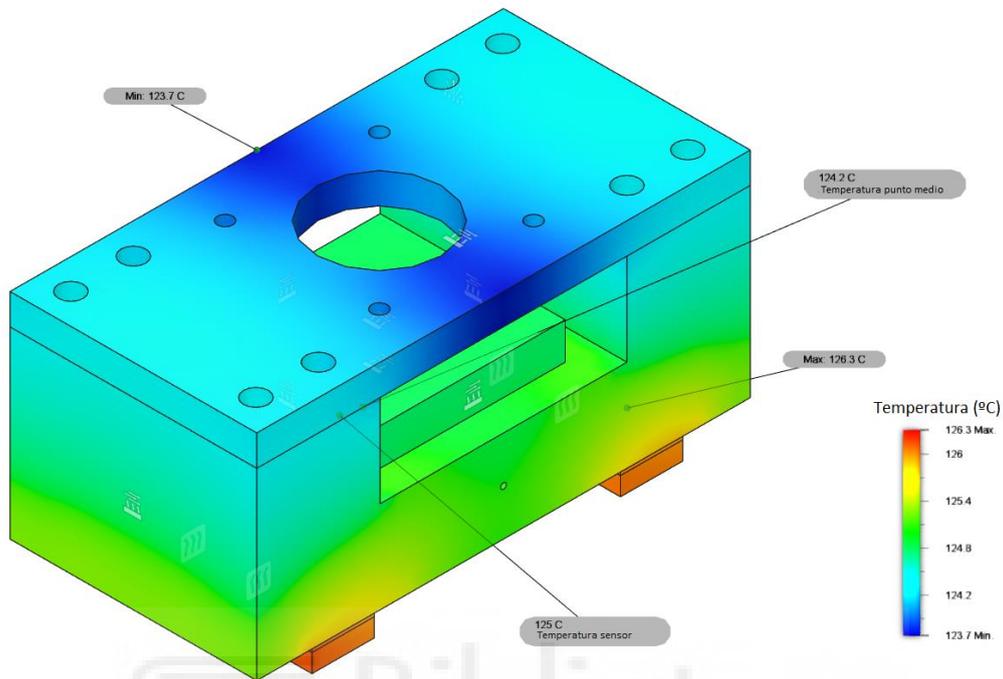


Figura 24. Resultado de la segunda simulación térmica del prototipo 2

Como puede observarse en esta simulación con las cuatro resistencias configuradas en paralelo si es posible alcanzar la temperatura objetivo. Aun así, este prototipo ha sido descartado ya que para lograr que cada resistencia alcance los 28W sería necesaria una intensidad de 9A, muy superior a la que la fuente de alimentación puede suministrar. Las temperaturas obtenidas de la simulación pueden encontrarse en la Tabla 11.

Tabla 11. Temperaturas obtenidas de la segunda simulación térmica del prototipo 1

Propiedad	Valor
Temperatura Máxima	126,3°C
Temperatura Mínima	123,7°C
Temperatura en el sensor	125°C
Temperatura en el punto medio de la aguja	124,2°C

Lo que podemos deducir de ambas simulaciones si observamos las temperaturas obtenidas, es que la diferencia de temperatura existente entre el

2. Diseño de Hardware

sensor y el punto medio de la aguja es inferior a 1°C . Sabiendo esto, podemos estar seguros de que las lecturas del sensor de temperatura estarán muy cerca de las temperaturas reales.

2.7.3. PROTOTIPO 2

Descartado el uso de una envoltura para el disipador, se decidió utilizar un disipador más grande que pudiera albergar el resto de elementos del sistema por sí mismo (Figura 25). Las ventajas de este prototipo son que dado su pequeño volumen se requiere de un menor tiempo de calentamiento.

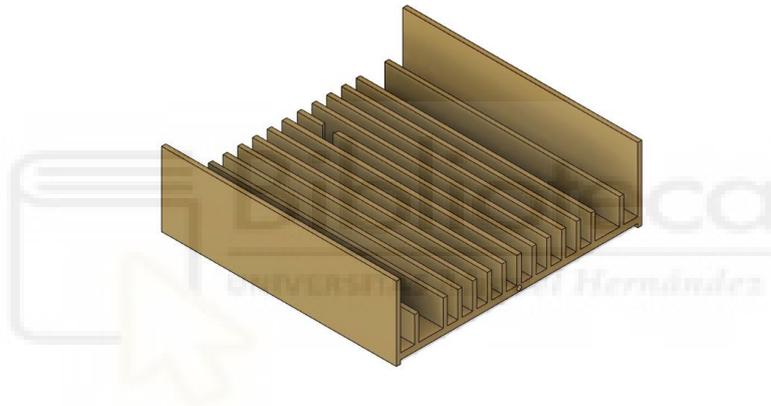


Figura 25. Prototipo número 2

Para anclar los elementos de control de temperatura se realizaron taladros en el disipador donde estos elementos iban situados. La colocación de estos elementos puede verse en la figura 26.

2. Diseño de Hardware



Figura 26.- Prototipo número dos

El principal problema de esta versión viene a la hora de incorporar la aguja. Dado el poco espacio disponible para taladrar el disipador, es inviable realizar un agujero transversal ya que esto rompería la pieza.

La solución a este problema pasa por modificar el disipador para crear un canal en el que colocar un tubo de nylon relleno de pasta térmica y cerrado por los extremos. En el interior de este tubo, estaría situada la aguja. Todo esto unido al disipador mediante una pieza realizada por impresión 3D.

Debido a la fragilidad y el pequeño tamaño del disipador, no es posible la fabricación.

2. Diseño de Hardware

2.7.3.1. SIMULACIÓN TÉRMICA DEL PROTOTIPO 2

La configuración de las resistencias de este prototipo es dos resistencias de 5Ω en serie desarrollando una potencia de 14W cada una. El material asignado al prototipo es el aluminio. En la Tabla 12 pueden observarse los parámetros utilizados para la simulación térmica del segundo prototipo.

Tabla 12. Tabla de parámetros de la simulación térmica del segundo prototipo

Propiedad	Valor
Temperatura ambiente	25°C
Coeficiente de transmisión de calor por convección del soporte	10 W/m ² K [29]
Emisividad del soporte	0.05 [30]

Los resultados de la simulación pueden verse en la Figura 27.



2. Diseño de Hardware

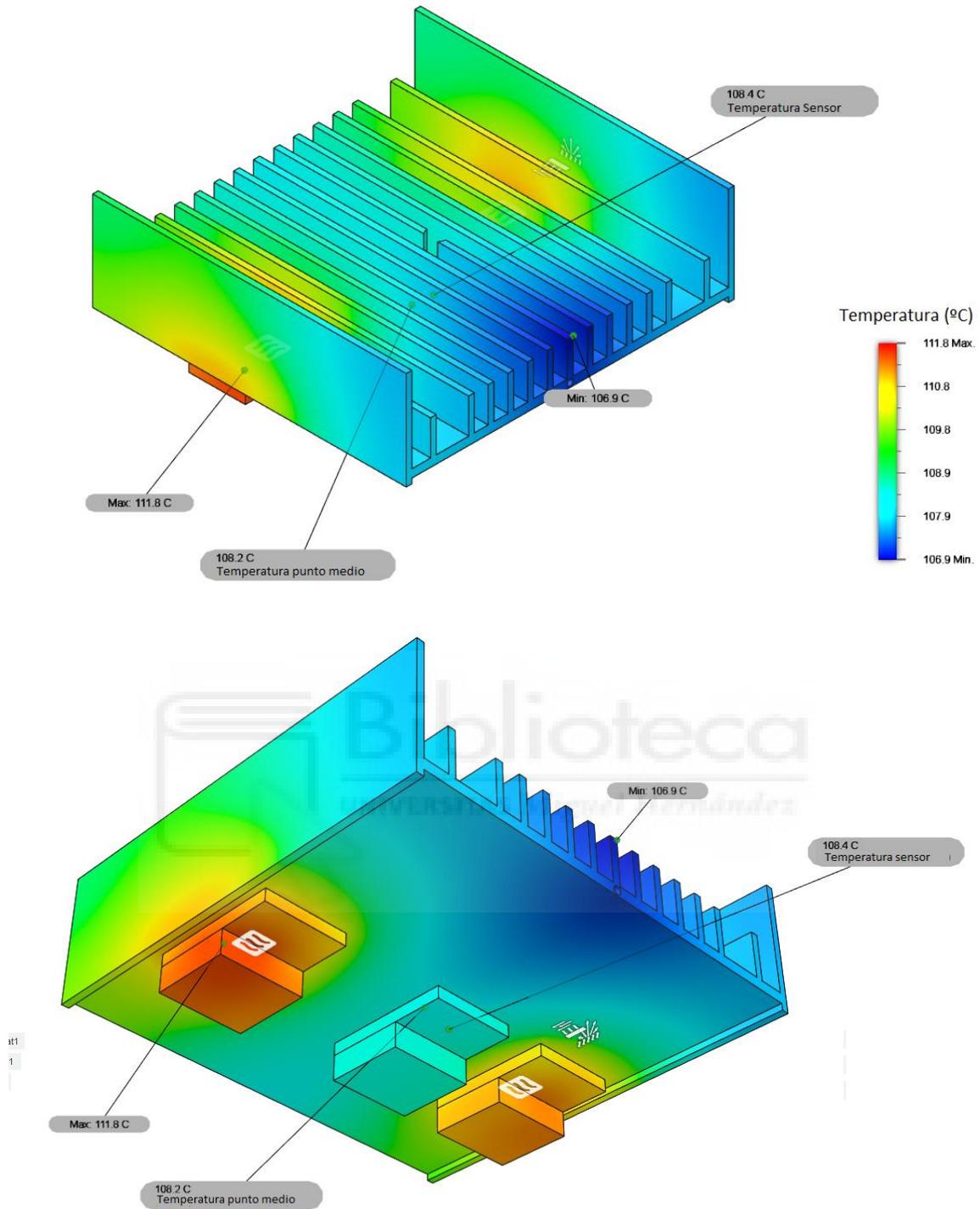


Figura 27. Resultado de la simulación térmica del prototipo 2

Como puede apreciarse en la simulación, este prototipo con la configuración de resistencias anteriormente mencionada es capaz de alcanzar la temperatura de

2. Diseño de Hardware

trabajo. Los resultados numéricos de esta simulación pueden encontrarse en la Tabla 13.

Tabla 13. Temperaturas obtenidas de la simulación del prototipo 2

Propiedad	Valor
Temperatura Máxima	111,8°C
Temperatura Mínima	106,9°C
Temperatura en el sensor	108,4°C
Temperatura en el punto medio de la aguja	108,2°C

De esta tabla podemos deducir que, aunque exista disparidad entre la temperatura máxima y mínima, lo que implica que la temperatura no se distribuye uniformemente por el dispositivo, la temperatura en el sensor y en el punto medio difieren unos pocos decimales. Esto nos indica que las lecturas del sensor serán representativas de la temperatura real de la aguja.

2.7.4. PROTOTIPO 3

Para el tercer prototipo se ha realizado una aproximación totalmente diferente. En lugar de fabricar una pieza mecanizada, muy costosa y complicada con los medios disponibles, se ha optado por fabricar una placa de circuito impreso que además de servir de elemento de interconexión de la resistencia calefactora y el sensor de temperatura, lleva soldada la aguja.

La placa de circuito impreso del sistema calefactor, que puede verse en la Figura 28 y su esquemático en la Figura 29, requiere utilizar las dos caras disponibles en su diseño. En una cara, la *TOP*, se ha creado un gran plano de cobre que se utilizará como elemento transmisor de la energía térmica. Por la otra, la *BOTTOM*, están situados tanto el sensor de temperatura como la resistencia calefactora.

2. Diseño de Hardware

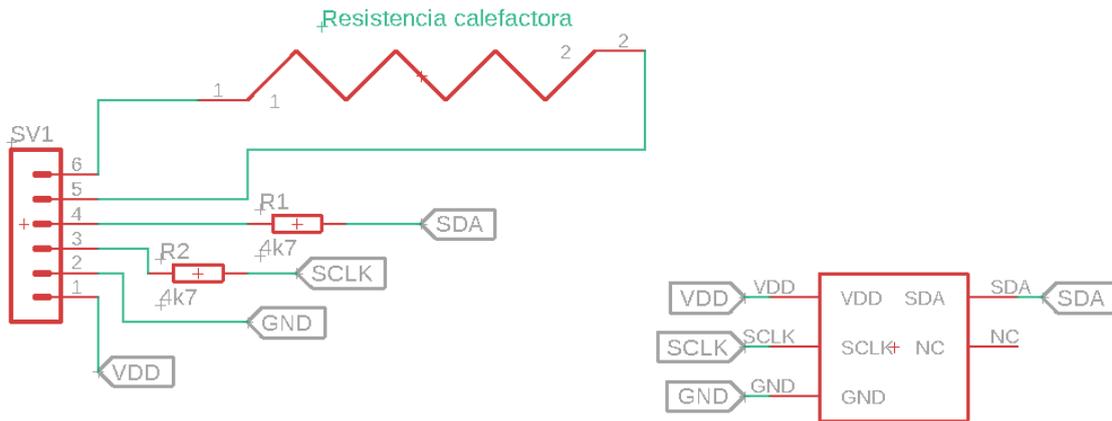


Figura 28. Esquemático prototipo 3

Además, se han incluido en esta placa las resistencias requeridas para la transmisión de datos del sensor. De esta forma liberamos espacio en la placa principal. La conexión entre ambas se realizará directamente con un cable soldado en las salidas de la placa.

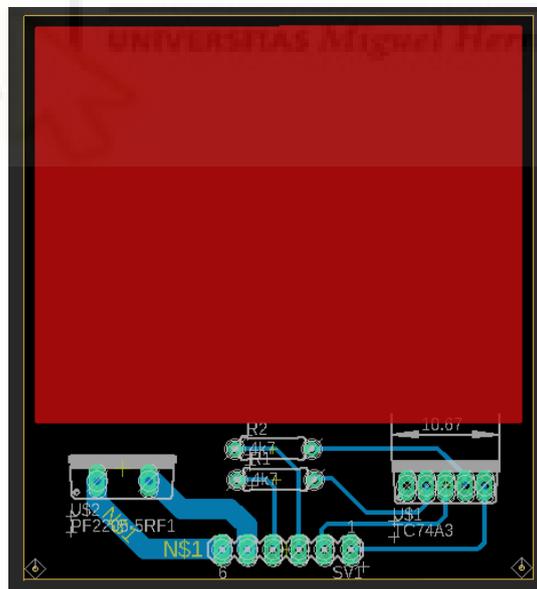


Figura 29. Diseño de soporte para la aguja basado en una PCB

Como la superficie de cobre es muy pequeña como para poder mantener una estabilidad térmica se ha soldado una lámina de cobre de 5mm de espesor que ayuda a mejorar la dinámica del sistema. Esta placa de cobre hace más lento el

2. Diseño de Hardware

proceso de calentamiento, pero ayuda a estabilizar las temperaturas una vez se ha alcanzado el *set point* debido a que aumenta la inercia térmica del sistema.

En este sistema la aguja, de acero inoxidable, va soldada al cobre. Esto es un problema ya que la soldadura entre el acero inoxidable y el cobre presenta dificultades utilizando los medios disponibles, soldadura con estaño. La adhesión entre el acero inoxidable y el estaño es muy baja, si bien esto no es un impedimento total para realizar la soldadura, requiere de un trabajo previo de preparación de la zona a soldar. Para poder realizar la soldadura ha sido necesario lijar la superficie de la aguja para aumentar su porosidad. Aun así, este proceso no es sencillo y existe un gran riesgo de deteriorar la aguja en la operación de lijado. Es por ello por lo que este prototipo ha sido descartado ya que, en caso de necesitar ser sustituida, es un trabajo que se ha considerado complicado para el usuario final. En la Figura 30 puede encontrarse una imagen del prototipo fabricado donde puede apreciarse la soldadura.

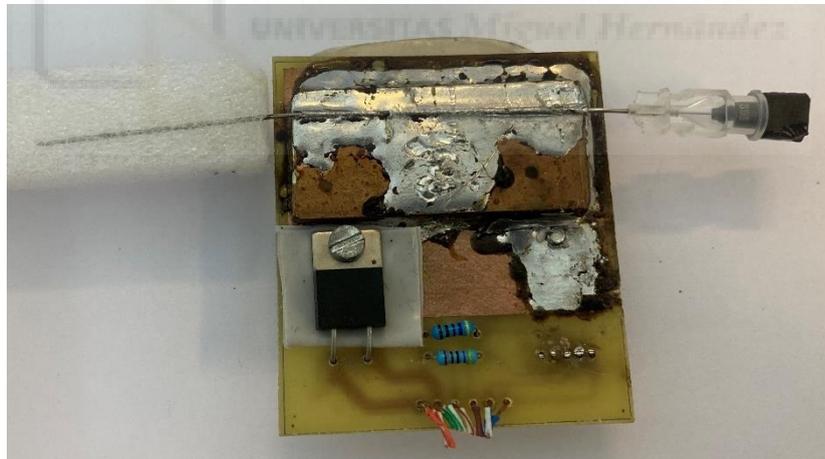


Figura 30. Prototipo 2 fabricado

2. Diseño de Hardware

2.7.4.1 SIMULACIÓN TÉRMICA DEL PROTOTIPO 3

Dado que este prototipo se ha diseñado como una placa de circuito impreso ha sido necesario realizar el diseño de un modelo tridimensional (Figura 31) para someterlo a la simulación térmica.

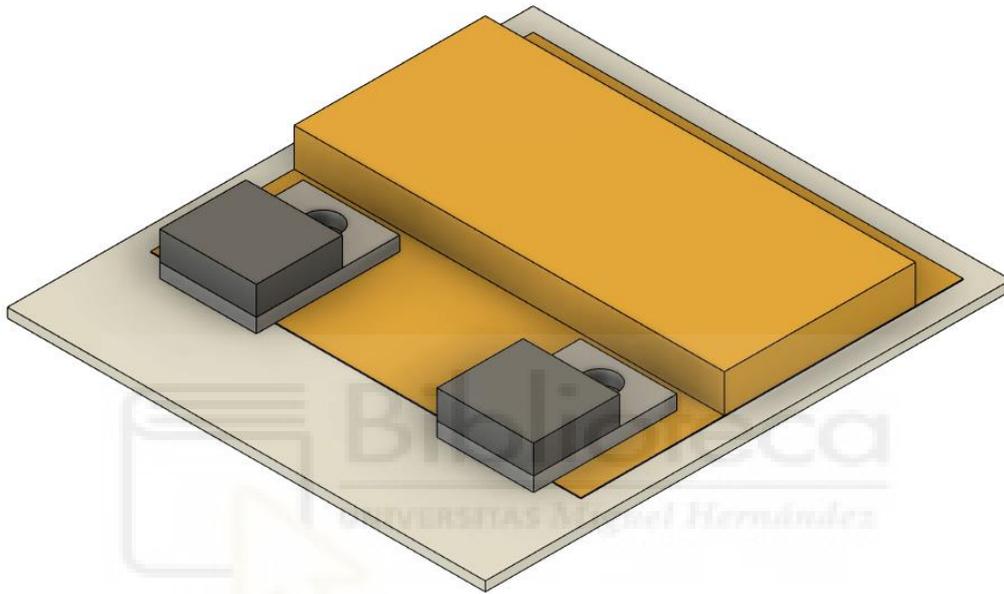


Figura 31. Diseño tridimensional del prototipo 3 para la simulación térmica

En este diseño se ha asignado el material FR4 para la base de la placa y cobre para el resto de elementos. En este caso el prototipo cuenta con una sola resistencia calefactora que desarrolla una potencia de 28 W. Los parámetros utilizados para la simulación pueden encontrarse en la Tabla 14.

2. Diseño de Hardware

Tabla 14. Tabla de parámetros de la simulación térmica del tercer prototipo

Propiedad	Valor
Temperatura ambiente	25°C
Coefficiente de transmisión de calor por convección del soporte	10 W/m ² K [29]
Emisividad del soporte	0.025 [30]

Los resultados de la simulación pueden verse en la Figura 32.

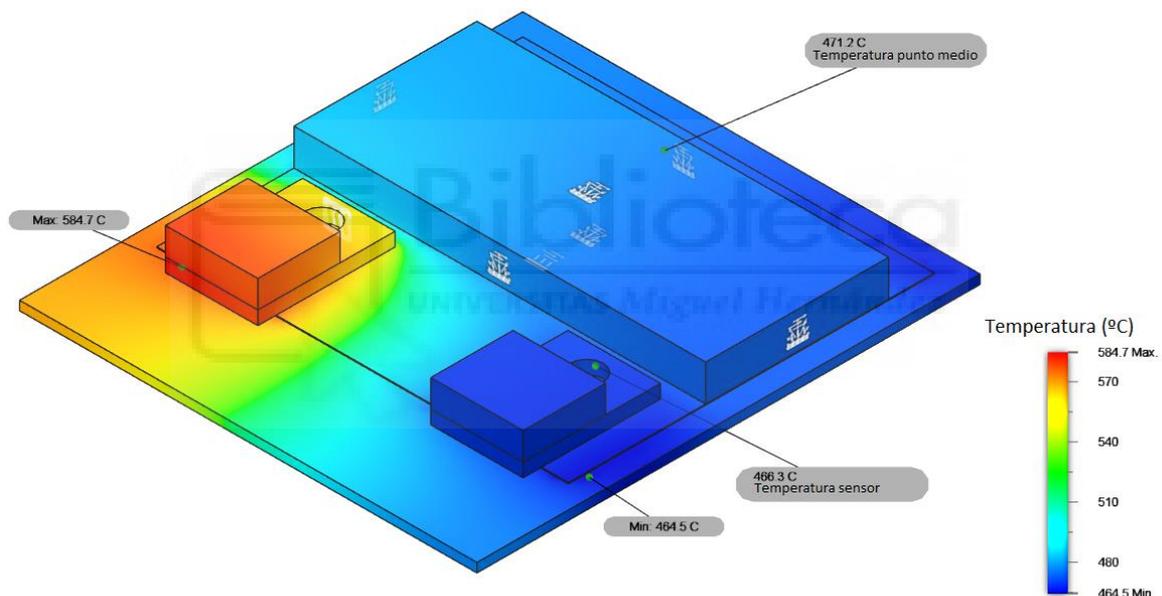


Figura 32. Resultado de la simulación térmica del prototipo 3.

En la simulación podemos apreciar que el prototipo es capaz de alcanzar sobradamente la temperatura de trabajo. De hecho, se puede ver como sometido a un calentamiento sin control la resistencia supera la temperatura de trabajo recomendada de 175°C lo que llevaría a su rotura. Esto es debido a la poca superficie con la que cuenta el prototipo para disipar temperatura, lo que hace que almacene toda la energía suministrada por la resistencia. Este hecho ha sido comprobado de forma empírica en el laboratorio. Si quisiéramos utilizar este prototipo para el trabajo final, debería implementarse una limitación en la

2. Diseño de Hardware

corriente que la resistencia puede consumir para impedir su rotura. Esto supondría un aumento en el tiempo de establecimiento del sistema y haría que no cumpliera con los requisitos del proyecto. Las temperaturas obtenidas de la simulación pueden encontrarse en la Tabla 15.

Tabla 15. Temperaturas obtenidas de la simulación del prototipo 3

Propiedad	Valor
Temperatura Máxima	584,7°C
Temperatura Mínima	464,5°C
Temperatura en el sensor	466,3°C
Temperatura en el punto medio de la aguja	471,2°C

Como puede observarse en la Tabla 15 la distribución de temperaturas en el dispositivo es muy heterogénea. Esto supone un gran inconveniente a la hora de utilizar este prototipo, ya que la temperatura medida por el sensor no será representativa de la temperatura del sistema ni de la temperatura de la aguja. Este, entre los motivos citados anteriormente, ha llevado a descartar el prototipo.

2.7.5. PROTOTIPO FINAL

Con todo lo aprendido del diseño y fabricación de los anteriores sistemas se fabricó un sistema formado por dos piezas de cobre, pero que no requerían de labores de mecanizado (Figura 33).

Estas dos piezas son una plancha de cobre de 75 mm x 20 mm x 4 mm y un tubo de cobre de 75 mm de longitud, 4 mm de diámetro externo y 3 mm de diámetro interno. La idea detrás de este montaje es que, además de su fácil fabricación, su reparación resulte sencilla. Esto se consigue debido a que la aguja va introducida dentro del tubo de cobre, relleno de pasta térmica y sujeta utilizando un adhesivo especial resistente a la temperatura.

2. Diseño de Hardware



Figura 33. Prototipo número 3

De esta forma cuando se requiera cambia la aguja, solo es necesario retirar el adhesivo, sustituirla y volver a sellarlo todo. Este diseño aporta una gran ventaja con respecto a los que hemos visto en el estudio de mercado, ya que a diferencia de estos permite la sustitución de la aguja.

El proceso de fabricación es el siguiente. Primero se realizaron los taladros para sujetar las resistencias y el sensor de temperatura. Después, se procedió a soldar el tubo a la placa de cobre. Para ello se ha utilizado la misma técnica empleada en trabajos de fontanería, la soldadura mediante soplete de gas. Para esta soldadura es necesario utilizar un agente decapante que ayuda en la soldadura y proporciona una unión fuerte. El prototipo fabricado puede verse en la Figura 34.

2. Diseño de Hardware

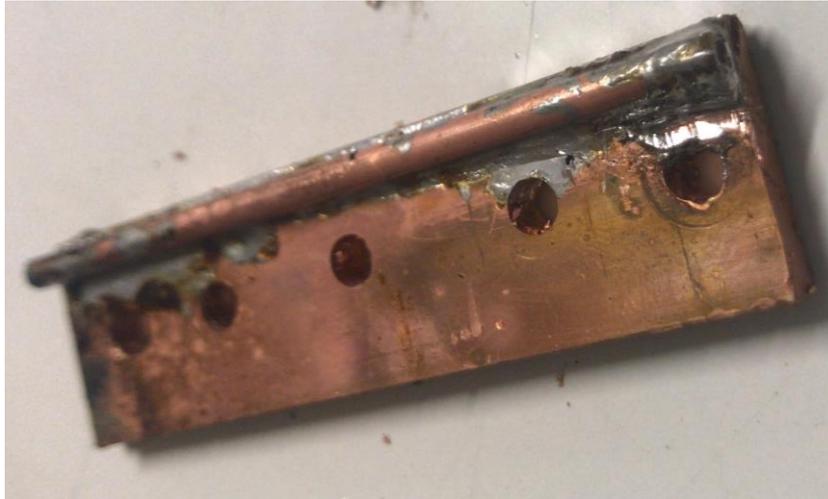


Figura 34. Prototipo final fabricado

Una vez con el sistema taladrado y soldado se atornillan todas las resistencias y el sensor de temperatura. Para ello se utiliza un producto diseñado para mejorar la transmisión de calor en el montaje de disipadores, lo que mejora la eficiencia de las resistencias, así como la precisión del sensor de temperatura.

2.7.5.1. SIMULACIÓN TÉRMICA DEL PROTOTIPO FINAL

Para la simulación térmica de este prototipo se ha utilizado la configuración de resistencias del capítulo 2.2.3.3.1, dos grupos en paralelo de dos resistencias de 5Ω en serie. Con esta configuración cada resistencia desarrolla una potencia de 7W. En este caso el material del prototipo es cobre. Los parámetros de la simulación pueden encontrarse en la Tabla 16.

Tabla 16. Tabla de parámetros de la simulación del prototipo cuatro

Propiedad	Valor
Temperatura ambiente	25°C
Coeficiente de transmisión de calor por convección del soporte	10 W/m ² K [29]
Emisividad del soporte	0.025 [30]

2. Diseño de Hardware

Los resultados de la simulación pueden verse en la Figura 35.

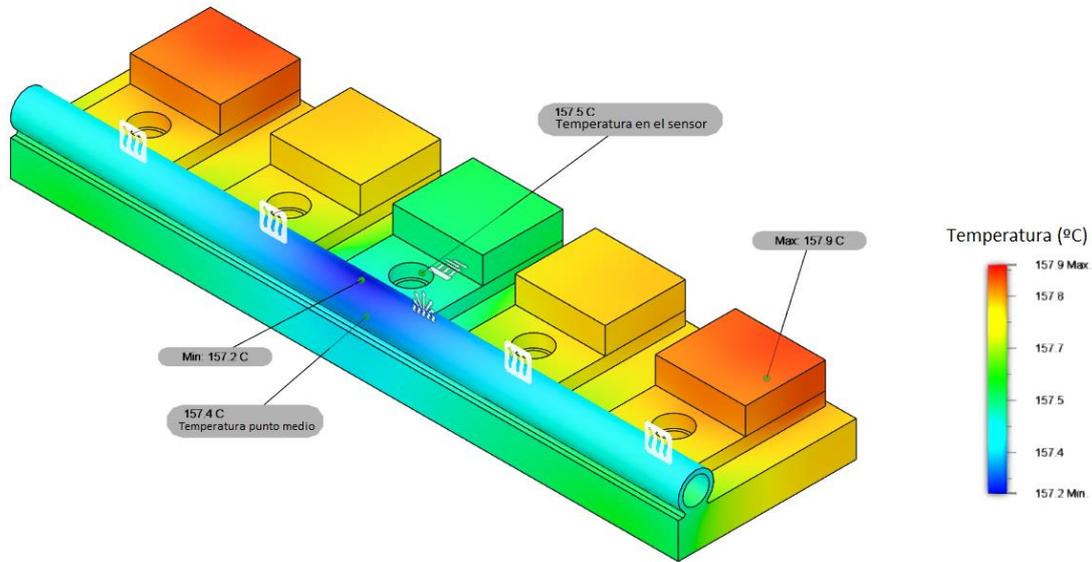


Figura 35. Resultado de la simulación térmica del prototipo final

Como puede verse en la simulación el prototipo es capaz de alcanzar y mantener la temperatura de trabajo. Los resultados numéricos de la simulación se encuentran en la Tabla 17.

Tabla 17. Resultados numéricos de la simulación térmica del prototipo final

Propiedad	Valor
Temperatura Máxima	157,9°C
Temperatura Mínima	157,2°C
Temperatura en el sensor	157,5°C
Temperatura en el punto medio de la aguja	157,4°C

Los resultados de la simulación son los más homogéneos de todos. Esto nos indica que este prototipo final nos va a permitir realizar un control de temperatura muy preciso, ya que distribuye el calor por todo él de forma que la diferencia entre los puntos más dispares solo sea de unas décimas de grado.

3. DISEÑO DE SOFTWARE

3.1. ENTORNO DE TRABAJO

En los siguientes apartados se describe tanto el entorno de desarrollo de Arduino IDE, así como otros módulos y complementos de *software* necesarios para el proyecto o utilizados en él.

3.1.1. ARDUINO IDE

Arduino IDE es un entorno de desarrollo que facilita la tarea de desarrollar y compilar código para las tarjetas de prototipado Arduino [31].

Entre sus principales características destacan su consola *serial*, fundamental para las primeras etapas del desarrollo del proyecto y su lenguaje de programación, basado en C++ lo que permite un rápido aprendizaje si ya se cuenta con una base de conocimiento en este lenguaje.

La consola *serial* de Arduino permite la comunicación *serial* con el dispositivo. Esto permite entre otras cosas poder conocer los valores de lectura del sensor de temperatura sin necesidad de ningún otro dispositivo, así como el valor de cualquier variable que intervenga en el sistema.

El lenguaje de programación de Arduino está basado en C++. Es un lenguaje de programación orientado a objetos. La ventaja de trabajar con el lenguaje de Arduino es que además de utilizar instrucciones propias, permite utilizar comandos estándar de C++ en su programación.

3.1.2. LIBRERÍAS

Una librería es un fragmento de código escrito en un lenguaje de programación que ofrecen al desarrollador la capacidad de reutilizar código para optimizar el tiempo de desarrollo.

A diferencia del código principal las librerías no se ejecutan de forma autónoma, si no que su cometido es ser llamadas por otros programas para ejecutar su función cuando estos los requieran.

3.1.2.1. LIBRERÍA SERIAL

La placa de desarrollo Arduino posee soporte hardware para la comunicación *serial* utilizando los pines 0 y 1, que además se comunican con el PC vía USB. Esta comunicación se realiza gracias al chip UART (*Universal asynchronous receiver-transmitter*) que Arduino lleva incorporado y que permite la comunicación con el microcontrolador.

La librería *Serial* permite gestionar esta comunicación de forma cómoda. Con ella podemos establecer, entre otros parámetros, la velocidad de la comunicación y gestionar el envío y recepción de datos.

La librería *Serial*, hasta las últimas versiones, trabaja mediante interrupciones. Por esto no se utiliza en el código final para evitar que las interrupciones puedan afectar a los tiempos del control PID.

3.1.2.2. LIBRERÍA WIRE

La librería *Wire* permite la comunicación de la placa Arduino con dispositivos que utilicen el protocolo I2C.

Este sistema de comunicación utiliza dos líneas de transmisión *SDA* Y *SCL*. La línea *SDA* es la que transmite los datos, mientras que la línea *SCL* transmite la señal de reloj.

Diseño de software

Esta librería utiliza 7 bits para identificar la dirección del dispositivo y un octavo bit para indicar si el microcontrolador lee o escribe en él.

3.1.2.3. LIBRERÍA LIQUIDCRYSTAL

La librería *LiquidCrystal* contiene una serie de instrucciones de bajo nivel que facilitan la tarea de comunicar la placa Arduino con un *display* LCD que esté basado en el chip HD44780 o compatibles.

Funciona creando un objeto que representa al *display LCD*. Una ventaja de utilizar esta librería es que permite la creación de caracteres personalizados. Por defecto los caracteres que puede representar una pantalla LCD están limitados. Gracias a esta librería se pueden crear hasta 8 caracteres personalizados, lo que se ha utilizado para crear los caracteres que representan los grados Celsius y el selector del menú.

3.1.2.4. LIBRERÍA EEPROM

La placa Arduino cuenta con una memoria EEPROM que permite almacenar información incluso cuando la placa está apagada. La librería contiene las instrucciones necesarias para escribir y leer en esta memoria.

En el proyecto se ha utilizado para almacenar los parámetros de control PID, permitiendo que no sea necesario calcularlos cada vez que se utilice el dispositivo.

3.1.2.5. LIBRERÍAS PROPIAS

Para mejorar la legibilidad del código y su posterior reutilización en otros proyectos, todo el código del proyecto se ha dividido en 3 librerías propias que agrupan todas las funciones necesarias para el funcionamiento del sistema. A continuación, se detalla su contenido.

3.1.2.5.1. LIBRERÍA PID

La librería PID contiene todas las funciones necesarias para el control de temperatura. Entre sus funciones principales destacan:

- Medir temperatura: Esta función es la encargada de comunicarse con el sensor I2C para realizar la lectura de la temperatura del sistema.
- Autotune: Esta función se encarga de calibrar el sistema y encontrar los parámetros óptimos para que el algoritmo PID funcione correctamente.
- PID: Es la función encargada de realizar el control de la temperatura y mantenerla en el punto que el usuario desee.

3.1.2.5.2. LIBRERÍA MENÚ

La librería menú es la encargada de hacer funcionar la interfaz del sistema y permitir al usuario controlarlo. Su programación se soporta sobre la librería *LiquidCrystal* mencionada anteriormente.

Funciona utilizando un bloque de código principal que es el encargado de detectar cuando se ha pulsado alguna tecla del sistema e implementa la lógica necesaria para actualizar la pantalla y las variables del sistema de acuerdo con dicha entrada.

El resto del menú trabaja mediante una función para cada submenú que deseemos mostrar al usuario, que serán llamadas mediante la función anterior.

3.1.2.5.3. LIBRERÍA MEMORIA

La librería memoria utiliza la *EEPROM* para guardar y leer los parámetros del PID. Cada vez que arrancamos el sistema carga los parámetros almacenados

Diseño de software

en la memoria. De la misma forma, cada vez que calibramos el sistema almacenamos dichos parámetros en memoria.

3.2. ALGORITMO DE CONTROL

Para implementar un control de temperatura se han evaluado los algoritmos descritos en el presente apartado.

Para su evaluación se han realizado todas las pruebas con las mismas condiciones que en el capítulo 2.2.3. Todas las pruebas se han realizado con las resistencias calefactoras unidas a una placa de cobre de 75 mm x 20 mm x 4 mm. Como *setpoint* se han establecido distintas temperaturas, siendo presentadas en este trabajo las que mejores resultados han mostrado. Todas las pruebas de calentamiento parten de la misma temperatura ambiente de 27°C.

3.2.1. CONTROL TODO/NADA

Es el sistema de control más simple. Consiste en revisar si la variable a controlar está por encima o por debajo del *setpoint*. Si se encuentra por debajo encendemos los elementos calefactores, mientras que si se encuentra por encima los apagamos. Su funcionamiento se muestra en la Figura 36.

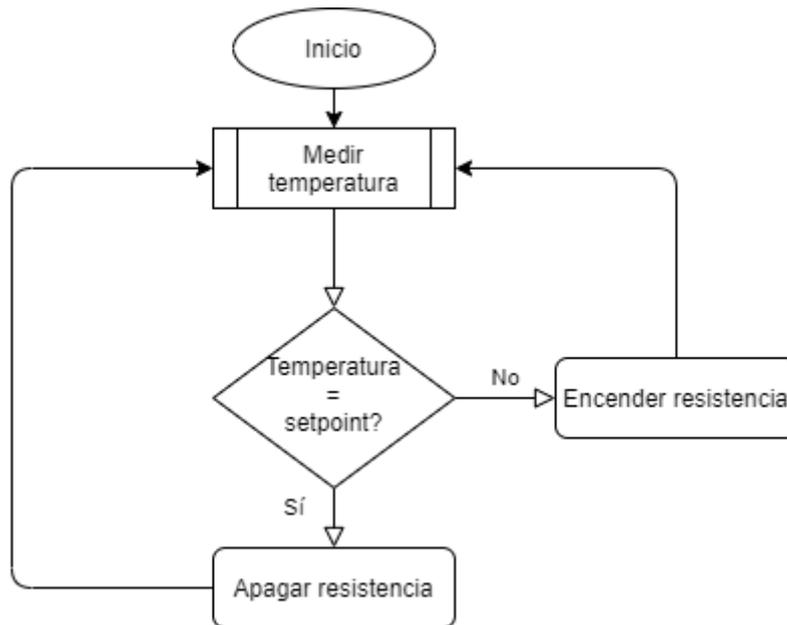


Figura 36 . Diagrama de flujo del algoritmo TODO/NADA

Este tipo de algoritmo de control solo es adecuado para aplicaciones de bajo rendimiento como podría ser un equipo de calefacción doméstico. Este algoritmo no ofrece la precisión necesaria para cumplir con los requisitos del proyecto.

Pero no es solo un problema de precisión, además que una conmutación constante puede producir fatiga en los elementos de control, más importante en sistemas mecánicos, provoca sobre oscilaciones en la variable de control. En el proceso que se va a controlar, una sobre oscilación puede deteriorar los elementos químicos utilizados, además de producir errores en los experimentos. Esto podría solucionarse implementando un control TODO/NADA con histéresis. En este tipo de controles se incorpora un retardo en el cambio de la señal para reducir el número de conmutaciones. El principal problema de introducir histéresis en el control TODO/NADA es la pérdida de precisión, por lo que este tipo de control no es el adecuado para sistemas donde se desee una temperatura constante y uniforme como es el caso.

3.2.1.1. IMPLEMENTACIÓN DEL ALGORITMO TODO/NADA

Dada la sencillez del algoritmo TODO/NADA su implementación no excede de unas pocas líneas. En el código que se muestra en la Figura 37 podemos apreciar el funcionamiento. En un primer lugar se mide la temperatura del sistema. Si esta, es menor que la temperatura objetivo, damos la orden de activar el circuito calefactor. Si por el contrario es menor o igual, lo desactivamos.

```
void todo_nada(int tempo){
  medirtemp();
  if (temp<tempo){
    analogWrite(resistencia,255);
  }

  else if (temp>=tempo){
    analogWrite(resistencia,0);
  }
}
```

Figura 37. Código del algoritmo TODO/NADA

3.2.1.2. PRUEBAS DEL ALGORITMO TODO/NADA

Los resultados de las pruebas del algoritmo TODO/NADA, mostrados en la Figura 38, han demostrado que este no es un sistema de control válido para la aplicación planteada. Por un lado, nos encontramos con una sobre oscilación considerable en el sistema. Al tratarse de un sistema que debe ser preciso para evitar daños en el objeto a estudiar, esta sobre oscilación podría dañar las muestras y alterar los datos del estudio.

Por otro lado, la precisión en el control a la temperatura es mayor que un grado. Esto choca frontalmente con uno de los requisitos planteados en el trabajo.

Es por ello por lo que se considera que el algoritmo TODO/NADA no es apto para el calentador de microperfusión.

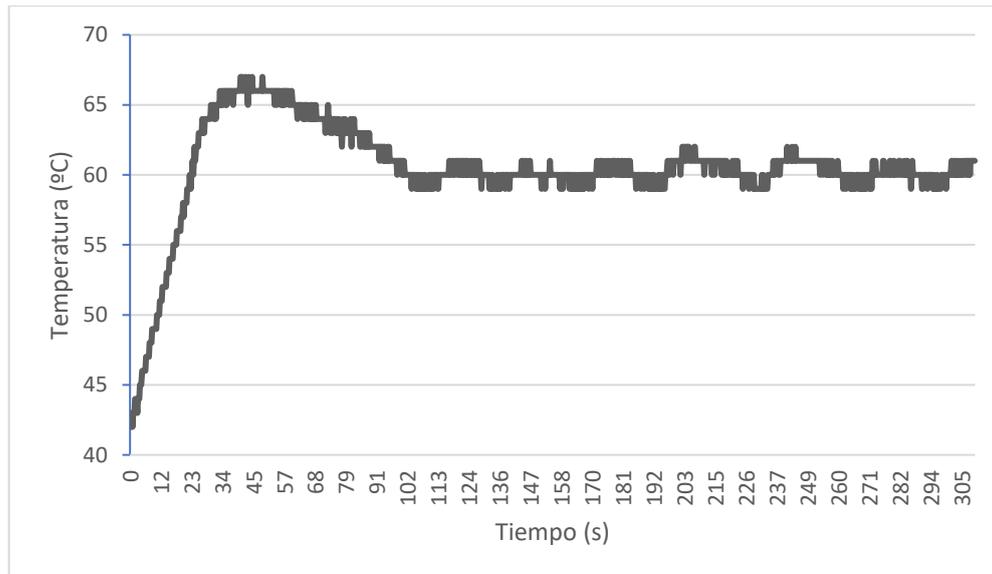


Figura 38. Gráficas del resultado de la prueba del algoritmo TODO/NADA con un setpoint de 60°C.

3.2.2. CONTROL PID

El algoritmo de control PID es uno de los más usados en la industria debido a su robustez, versatilidad y precisión. Trabaja mediante realimentación, calculando la acción de control en base al error entre el *setpoint* y la variable de control.

Este algoritmo de control calcula la acción de control en base a tres parámetros distintos: el proporcional, el derivativo y el integral.

El valor proporcional depende del error en el momento de la lectura. Este parámetro se calcula multiplicando una constante, K_p , por el error.

El valor integral depende del comportamiento del sistema en momentos anteriores. De esta forma se calcula como se ha comportado el error durante el tiempo y se multiplica por una constante, K_i .

El valor derivativo intenta predecir cómo será el error en el futuro para intentar corregirlo antes de que suceda. Trabaja con el valor inmediatamente anterior y

Diseño de software

mira si el error va en aumento o decrecimiento, así como la velocidad en la que está ocurriendo. Una vez calculado se multiplica por una constante K_d .

Su funcionamiento se muestra mediante un diagrama de bloques en la Figura 39.

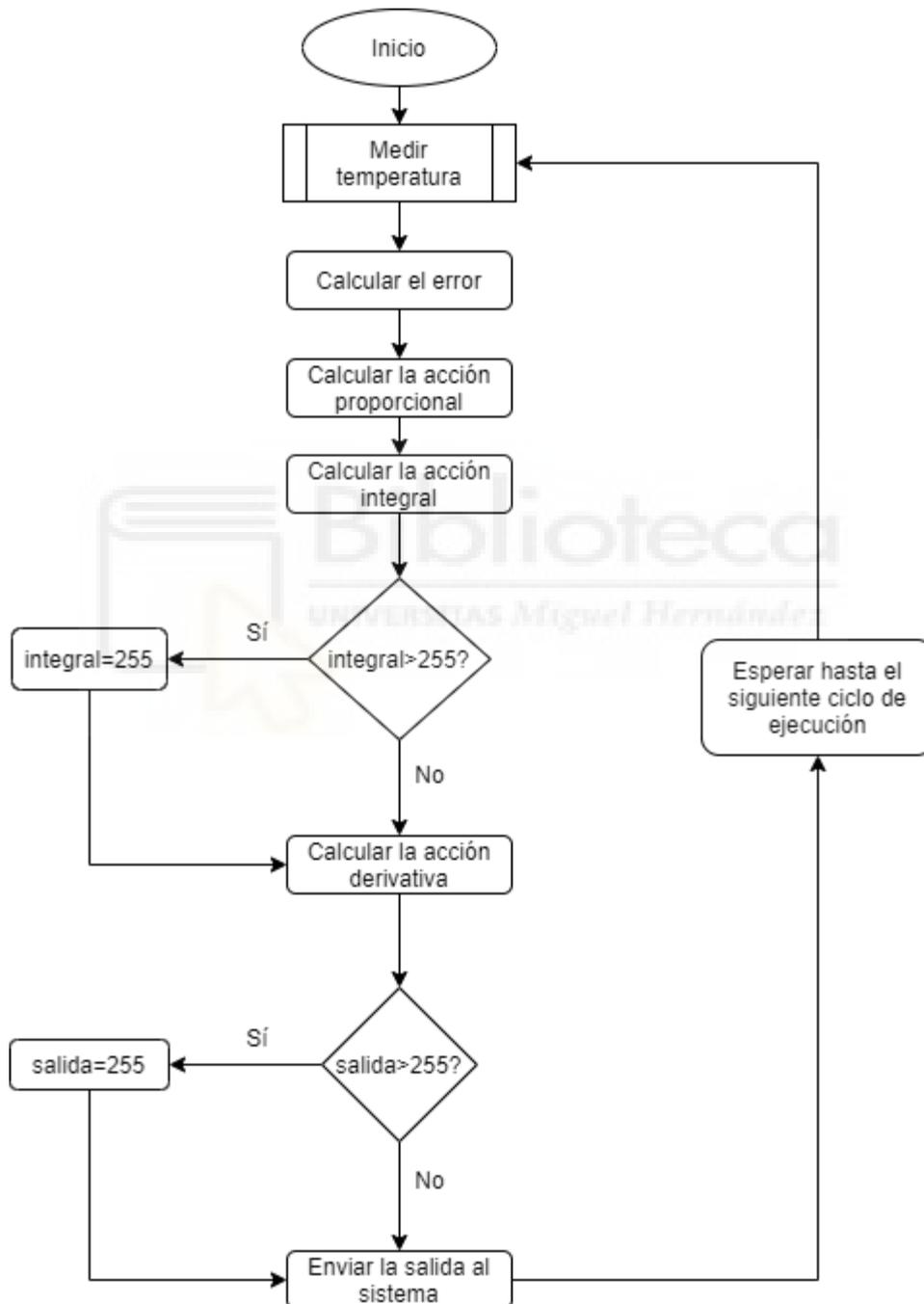


Figura 39. Diagrama de bloques de un controlador PID

3.2.2.1. IMPLEMENTACIÓN DEL ALGORITMO PID

La implementación del algoritmo de control PID no requiere de muchas líneas de código, ya que el grueso de la complejidad proviene del cálculo de los parámetros de control, explicados en el capítulo 3.2.2.3. El código puede consultarse en el Anexo 10.1.3

Como se ha explicado anteriormente, el control PID funciona actuando sobre el error del sistema. Por ello, lo primero será calcular dicho error. El valor de este parámetro se obtiene de la diferencia entre el valor objetivo, *setpoint*, y el valor actual.

En el caso de un control PID completo necesitamos conocer la derivada del error y la integral. En tiempo discreto podemos obtener este valor utilizando el periodo de la función PID. Este periodo se define en función de las necesidades del sistema y es una entrada de la función.

El término integral del error se calcula acumulando el error en todas las ejecuciones del algoritmo. Como se puede ver en la función este término del error cobra más importancia mientras más tiempo lleve el sistema funcionando.

Para conseguir un correcto funcionamiento del algoritmo PID es necesario implementar un sistema anti *Windup*. El *Windup* es un fenómeno que se produce cuando los sistemas a controlar tienen una dinámica lenta. En estos, mientras se alcanza el *setpoint* el error integral va acumulándose cada vez más, llegando al punto de que cuando el sistema alcanza el *setpoint* tenemos un error integral acumulado demasiado grande. Esto produce sobre oscilaciones muy grandes del sistema llegando incluso a abrirse el lazo de control y resultando en un sistema prácticamente incontrolable.

Para evitarlo existen diversos métodos, en este trabajo se ha empleado el más sencillo, que consiste en limitar la magnitud máxima del error integral a un valor equivalente a la máxima salida del sistema, en nuestro caso 255.

Diseño de software

El término derivativo, pensado para cambios rápidos del sistema, solo se manifiesta cuando existe un cambio en el valor absoluto del error, si no solo se aplican el efecto proporcional y el integral.

Una vez calculados, la salida del sistema se determina en función de estos valores y los parámetros de control del sistema. Por motivos de seguridad la salida está limitada a la máxima salida con la que trabaja nuestro dispositivo.

3.2.2.2. PRUEBAS DEL ALGORITMO PID

Los resultados de las pruebas del algoritmo PID, mostrados en la Figura 40, han demostrado que este sistema de control suple todas las carencias del TODO/NADA probado anteriormente.

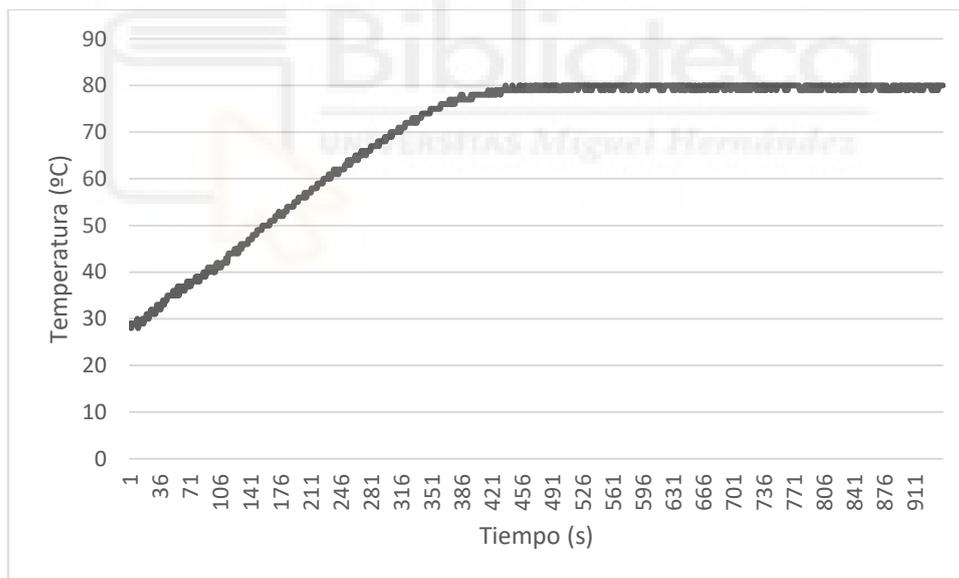


Figura 40. Resultados de las pruebas del algoritmo PID con un setpoint de 80°C

En la Figura 41, se muestra la gráfica ampliada en la zona estabilizada. Aquí se puede observar como las temperaturas están acotadas entre $\pm 1^{\circ}\text{C}$.

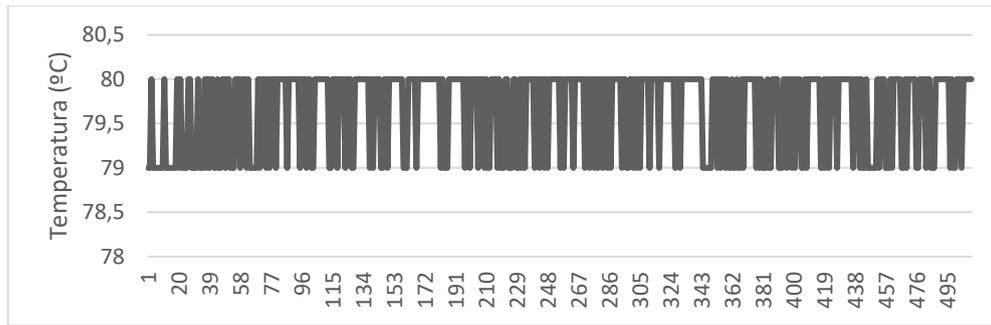


Figura 41. Gráfica ampliada una vez estabilizada la temperatura

3.2.2.3. AUTO-TUNE

El algoritmo de control PID requiere para su correcto funcionamiento de tres constantes que han de ser calculadas previamente. Como se ha explicado en el apartado anterior, estas constantes son K_p , K_i y K_d .

Es posible calcular estas constantes a mano y dejarlas introducidas en el código del programa. El problema es que, de esta forma, ante un cambio en las condiciones de trabajo el equipo podría funcionar erróneamente. Es por ello por lo que se ha decidido implementar un algoritmo que calcule dichas constantes, un algoritmo de *auto-tune*. Este algoritmo no se ejecuta continuamente, si no que será el propio usuario el que decida cuando debe ejecutarlo antes de cada experimento en función de sus condiciones de trabajo. Este algoritmo se ejecutará cuando el usuario ejecute la calibración desde el menú del sistema.

Los algoritmos de *auto-tune* intentan averiguar cuál es el comportamiento del sistema a controlar y entonces calcular las constantes de control para el PID. Existen varias formas de hacer esto, pero la gran mayoría funcionan de la misma manera, cambiando la salida del sistema y observando su respuesta. En este proyecto se ha utilizado el método de relé propuesto por Astrom y Hagglund en 1994 [32].

La idea detrás del método de relé es encontrar la ganancia y el periodo crítico que utilizan Ziegler y Nichols [33] en su método, pero de una forma automática y controlada. En el método de Ziegler y Nichols, estos valores se obtienen

Diseño de software

incrementando escalonadamente la ganancia del controlador PID hasta que la salida del sistema se vuelve inestable. El inconveniente es que no estamos actuando directamente sobre el rango de valores que se utilizaran en la aplicación final y, además, aumentar la ganancia de forma descontrolada puede llevar a la destrucción del sistema a controlar. Sin embargo, si se introduce un relé en el bucle de control, la mayoría de procesos comenzaran a oscilar. Es de esta oscilación de donde podemos averiguar los valores de ganancia y periodo necesarios para calcular los parámetros del controlador PID. La ventaja principal de este método es que no necesitamos conocer el proceso en sí, siempre y cuando podamos actuar sobre él. Además, cuenta con el añadido de que podemos decidir la temperatura sobre la que el auto-tune oscilará. La elección de esta temperatura es clave ya que de ella depende la dinámica del controlador PID. Este algoritmo nos permite seleccionar una temperatura de auto-tune cercana a la temperatura de trabajo, lo que nos otorgara un controlador calibrado especialmente para trabajar en esa zona.

El funcionamiento de este método se describe a continuación (Figura 42). Comenzando con un sistema en estado estacionario, se aplica una salida escalón de un valor D. Cuando el parámetro de control cruce la línea de disparo, se le aplicará a la salida un escalón de valor inverso al anterior. El valor de disparo debe de estar cerca del valor de trabajo del sistema.

Analizando el comportamiento de la variable a controlar podemos calcular el valor de los parámetros de control. El análisis se basa en el valor de los picos generados y la distancia entre dichos picos.

Diseño de software

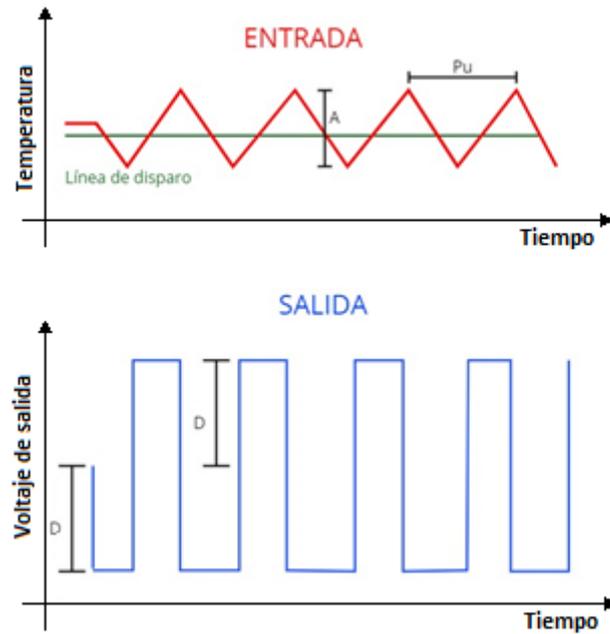


Figura 42. Ejemplo de implementación del relé

Una vez realizado, los parámetros se calculan siguiendo las fórmulas establecidas por Ziegler y Nichols [33] que se muestran en la Tabla 18.

Tabla 18. Tabla resumen de los cálculos necesarios para hallar los parámetros de control del algoritmo PID

Tipo de control	K_p	K_i	k_d
PI	$0.4 * K_u$	$0.48 * K_u / P_u$	0
PID	$0.6 * K_u$	$1.2 * K_u / P_u$	$0.075 * K_u * P_u$
Siendo $K_u = 4 * D / (A * \pi)$			

3.2.2.3.1. IMPLEMENTACIÓN DEL ALGORITMO AUTO-TUNE

El principal problema de implementar un algoritmo como este en la práctica es el ruido. En la práctica, la señal leída de la temperatura tendrá una forma parecida a la que puede apreciarse en la Figura 43.

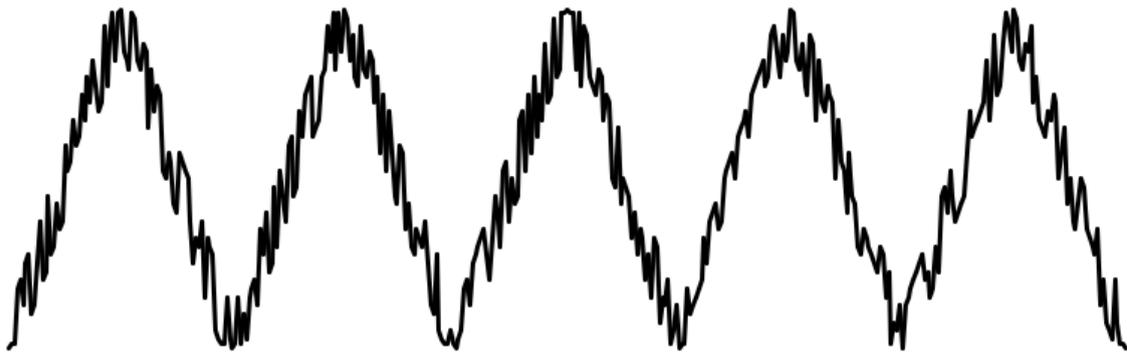


Figura 43. Señal de la lectura de temperatura en la practica

Un error en la lectura puede hacer que el sistema identifique como picos valores que no lo son. Esto haría pensar al algoritmo que hemos alcanzado un valor máximo o mínimo cuando realmente no es así, por lo que no tendríamos los parámetros reales de la dinámica del sistema y todo el cálculo posterior sería erróneo.

El mínimo cambio en la dirección de la señal puede hacer creer al sistema que nos encontramos ante un pico tal y como se muestra en la Figura 44. Esto lleva a un cálculo incorrecto de los parámetros de control.

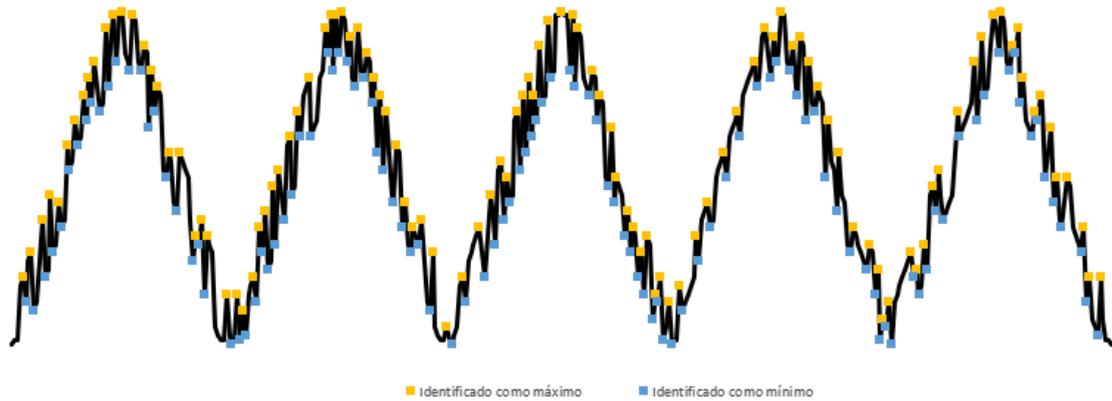


Figura 44. Sistema simple de identificación de picos

Para solucionar esto, el código implementado compara cada pico encontrado con el último pico detectado. Este proceso se realiza cada vez que se encuentra en la región superior o inferior de la línea de disparo con todos los posibles picos. De esta forma la identificación de picos se realiza de manera correcta, Figura 45.

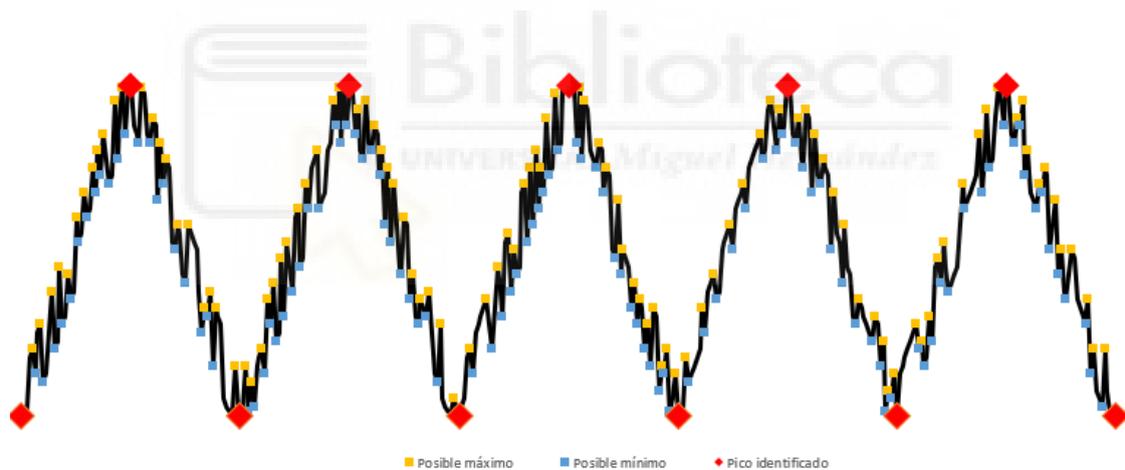


Figura 45. Sistema avanzado de detección de máximos y mínimos

El ruido también puede tener otro efecto negativo en la ejecución del algoritmo de *auto-tune*. Cuando la señal se encuentra cruzando la línea de disparo, pequeñas alteraciones debidas a este pueden hacer que el sistema crea que ha cruzado la línea de disparo sin que esto haya llegado a producirse realmente, Figura 46. Esto provocaría un mal funcionamiento del relé y haría que no se identificara correctamente la dinámica del sistema.

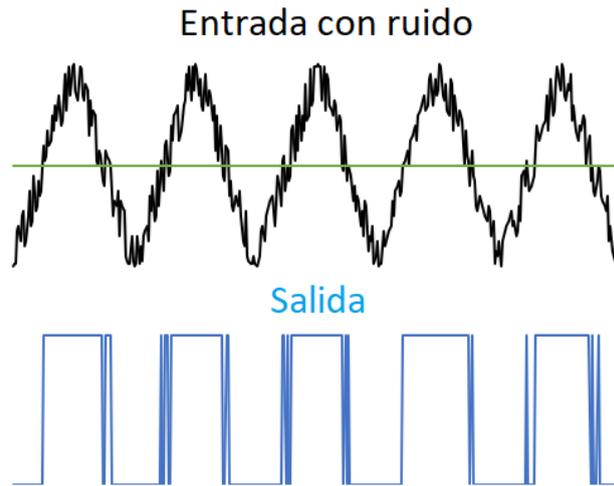


Figura 46. Ejemplo de problema derivado del ruido en el algoritmo de auto-tune

Para solucionarlo, en lugar de trabajar con una línea de disparo se ha trabajado con una banda de disparo, Figura 47. Esta banda indica la tolerancia que ha de tener el sistema al ruido.



Figura 47. Solución al ruido en el algoritmo de auto-tune mediante la implementación de banda de disparo

En la Figura 48 se encuentra el diagrama de flujo de el algoritmo de *auto-tune*. Dado que el código del algoritmo es muy extenso se encuentra íntegramente en el anexo 10.1.3.

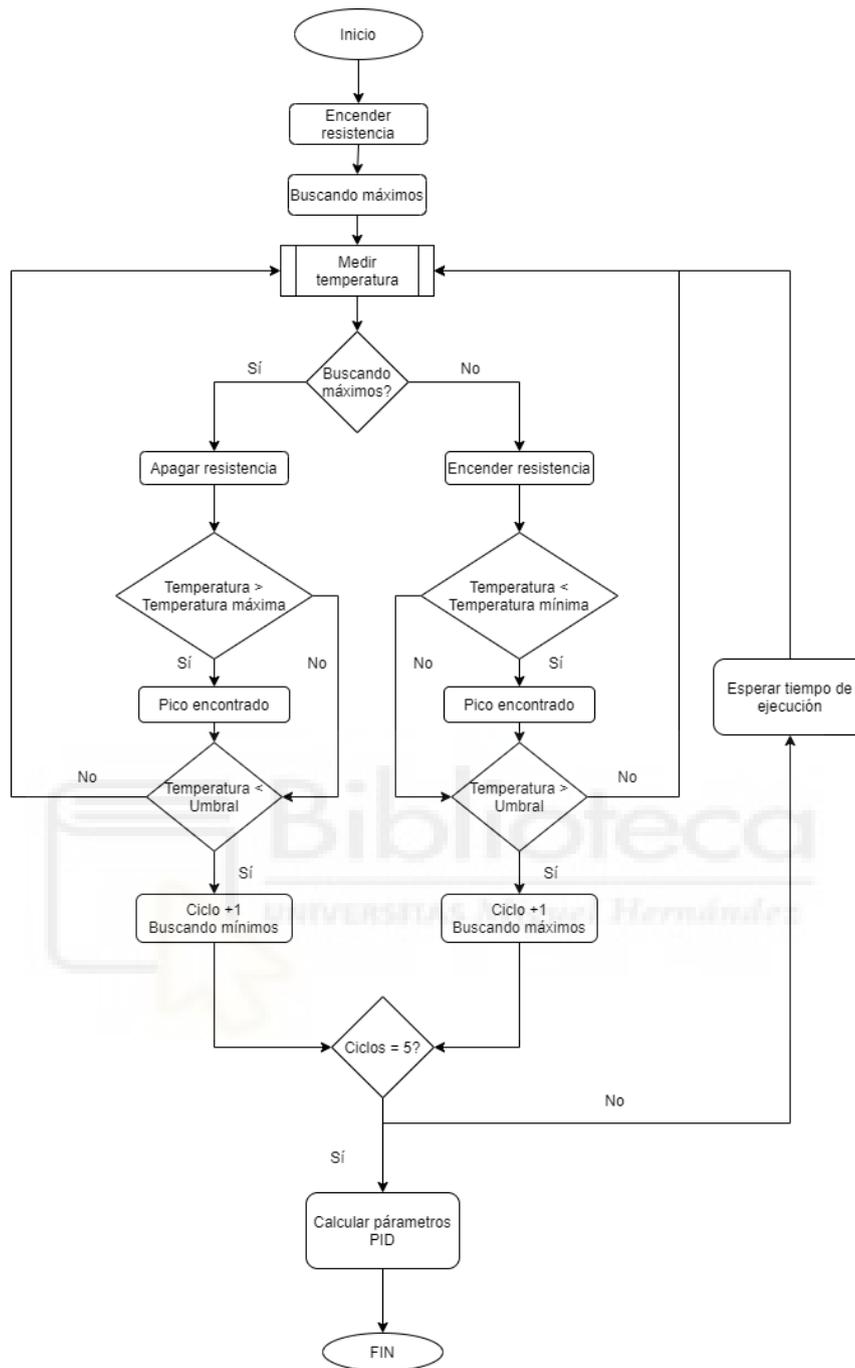


Figura 48.- Diagrama de flujo del algoritmo de auto-tune

Diseño de software

3.2.2.3.2. PRUEBAS ALGORITMO AUTO-TUNE

Para calibrar el sistema se ha utilizado el algoritmo de *auto-tune* con 3 ciclos de trabajo. Como se puede observar en la Figura 49, aun existiendo ruido en la medida este no ha afectado al desempeño del código.

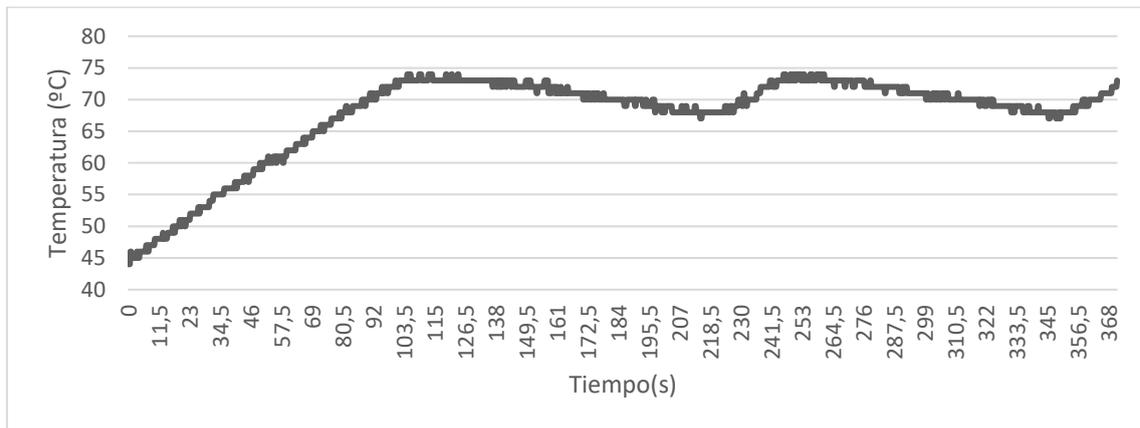


Figura 49.- Grafica de ejecución del algoritmo de auto-tune sobre el prototipo final con 3 ciclos y setpoint de 70°C. Temperatura ambiente 25°C

A continuación, en la Tabla 19, se muestran los parámetros de control calculados por el algoritmo de *auto-tune*.

Tabla 19. Resultados del algoritmo de auto-tune

Kp	24.50
Ki	0.39
Kd	398.01

4. Interfaz de usuario

4. INTERFAZ DE USUARIO

La interfaz de usuario se ha creado de forma que sea sencilla e intuitiva. Consta de una pantalla principal, donde se puede ver la información más importante del sistema: temperatura actual, *setpoint* y estado del sistema, y de un menú para acceder a las diversas funciones que el usuario tiene disponibles. Para una mejor comprensión del funcionamiento del sistema puede consultarse el manual de usuario que se encuentra en el Anexo 10.5.

El esquema de la interfaz puede apreciarse en la Figura 50.

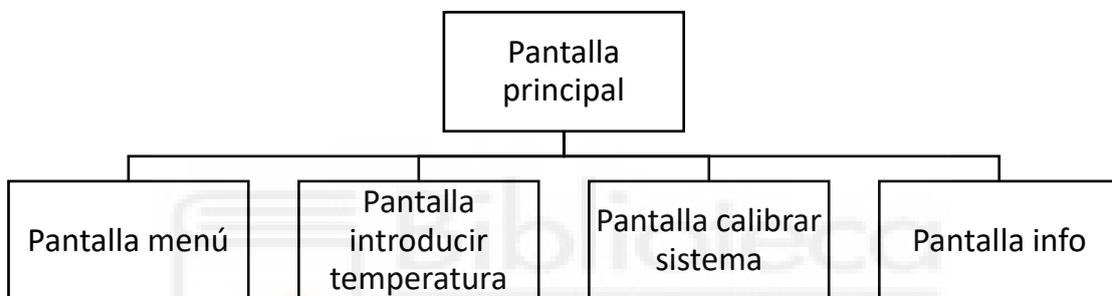


Figura 50. Diagrama de menús del sistema

4.1. PANTALLA PRINCIPAL

En la pantalla principal (Figura 51) se muestran la temperatura actual y el *setpoint* introducido por el usuario. Desde esta pantalla y pulsando la tecla izquierda se accede al menú.

4. Interfaz de usuario



Figura 51. Pantalla principal del menú

4.2. PANTALLA MENÚ

El menú es desde donde el usuario puede acceder a todas las funciones del sistema. Dado el tamaño del LCD se divide en dos pantallas.

En la primera (Figura 52) se le muestran al usuario la opción de ver la temperatura y la de cambiar la temperatura. La primera opción dirige al usuario hacia la pantalla principal. La segunda, hacia la introducción del *setpoint*.



Figura 52. Pantalla 1 del menú

En la segunda pantalla del menú (Figura 53) se muestran las opciones de calibrar e información. La opción de calibrar ejecuta el algoritmo de *auto-tune*. La de información, muestra al usuario los parámetros de control almacenados en la EEPROM.

4. Interfaz de usuario



Figura 53. Pantalla 2 del menú

Para acceder a cualquiera de estas opciones el usuario puede navegar con los botones arriba y abajo. Para entrar en la opción correspondiente debe pulsar derecha.

4.3. PANTALLA INTRODUCIR TEMPERATURA

En esta pantalla el usuario puede seleccionar la temperatura de trabajo mediante los botones arriba y abajo y fijarla mediante el botón derecho (Figura 54). Una vez fijada el sistema devolverá al usuario a la pantalla principal.



Figura 54. Pantalla Introducir temperatura del menú

4.4. PANTALLA CALIBRAR SISTEMA

Al acceder a esta pantalla (Figura 55) se le pide confirmación al usuario para iniciar el protocolo de la calibración del sistema. Este consiste en llamar a la función *auto-tune*. Durante la ejecución de esta funcionalidad se le impide al

4. Interfaz de usuario

usuario realizar cualquier acción para no interferir con el correcto funcionamiento del sistema. Solo le será posible cancelarlo. Una vez calibrado el sistema se le devolverá a la pantalla principal. La calibración del sistema debe realizarse a temperatura ambiente.



Figura 55. Pantalla Calibración del menú

4.5. PANTALLA INFO

En esta pantalla (Figura 56) el usuario puede consultar cuales son los parámetros de control almacenados en la memoria EEPROM.

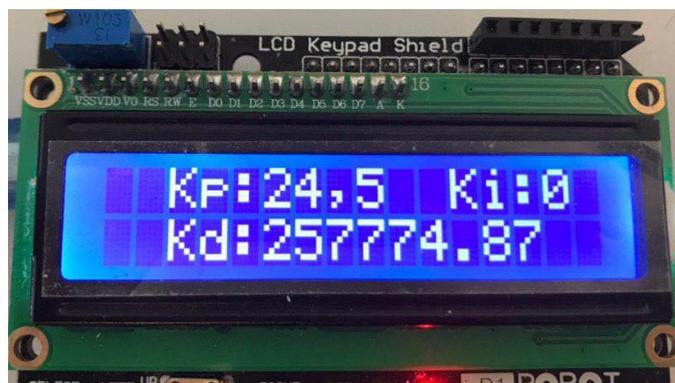


Figura 56. Pantalla info del menú

5. Fabricación sistema final

5. FABRICACIÓN SISTEMA FINAL

La fabricación del sistema final se divide en dos etapas. Por un lado, la parte electrónica, que consta de la fabricación de las placas de circuito impreso y el montaje de los componentes. Por otro la fabricación de las dos carcasas que protegerán el sistema, realizadas mediante impresión 3D.

5.1. PLACAS DE CIRCUITO IMPRESO

Con el fin de fabricar un sistema final de acabado profesional, es necesario trasladar todos los circuitos electrónicos diseñados en las fases previas a una placa de circuito impreso.

5.1.1. DISEÑO PCB

El diseño de las placas de circuito impreso es el paso previo a su fabricación. En esta etapa decidimos cual va a ser la colocación de los componentes electrónicos y cómo van a ser conectados entre sí. Para realizar este trabajo debemos basarnos en los esquemáticos del circuito y en las restricciones físicas del espacio donde va a ir montada la placa.

5.1.1.1. SOFTWARE DE DISEÑO

Para el diseño de las placas de circuito impreso se ha utilizado el *software* de diseño Autodesk EAGLE [34]. La elección de este *software* viene motivada por su facilidad de uso y el alto número de recursos disponibles al alcance del usuario. Al ser uno de los programas más utilizados, están disponibles numerosas librerías de componentes electrónicos ya creados para su uso directo en el diseño.

Además, este *software* cuenta con herramientas de exportación de los diseños, tanto en 2D como en 3D, que posteriormente a esta fase han resultado muy útiles a la hora de realizar el diseño de las carcasas del sistema.

5. Fabricación sistema final

5.1.1.2. NORMAS DE DISEÑO

Dado que las placas son fabricadas en los laboratorios de la Universidad Miguel Hernández se han utilizado las mismas normas de diseño que emplean los alumnos en sus prácticas de laboratorio. El objetivo de utilizar estas normas es que la fabricación posterior de las placas de circuito impreso sea viable con los medios técnicos de los que se dispone en la Universidad.

Las normas de diseño utilizadas se exponen en la Tabla 20.

Tabla 20. Normas de diseño utilizadas en la fabricación de las placas de circuito impreso del proyecto

Cara de enrutado	TOP
Clearence	8 mil (0.2032 mm)
Distance: Copper/Dimension	20 mil (0.508 mm)
Sizes: Minimum Width	24 mil (0.6096 mm)
Sizes: Minimum Drill	10 mil (0.254 mm)
Restrings: Pads	16 mil (0.4064 mm)

5.1.1.3. DISEÑO DE LA PLACA DE CONTROL

La placa de control (Figura 57) se trata de una *Shield*. Este tipo de placas se diseñan de forma que puedan ser encajadas en otras placas electrónicas de forma que queden perfectamente integradas y su funcionalidad no se vea afectada.

En nuestro caso la shield debería encajar en una placa de desarrollo Arduino. Afortunadamente existen librerías de EAGLE pensadas para el diseño de *Shields* [33]. Estas librerías ya aportan las restricciones dimensionales que ha de cumplir nuestra shield para que encaje correctamente. Principalmente es necesario que respetemos la posición de los pines del Arduino y, en nuestro caso, tengamos

5. Fabricación sistema final

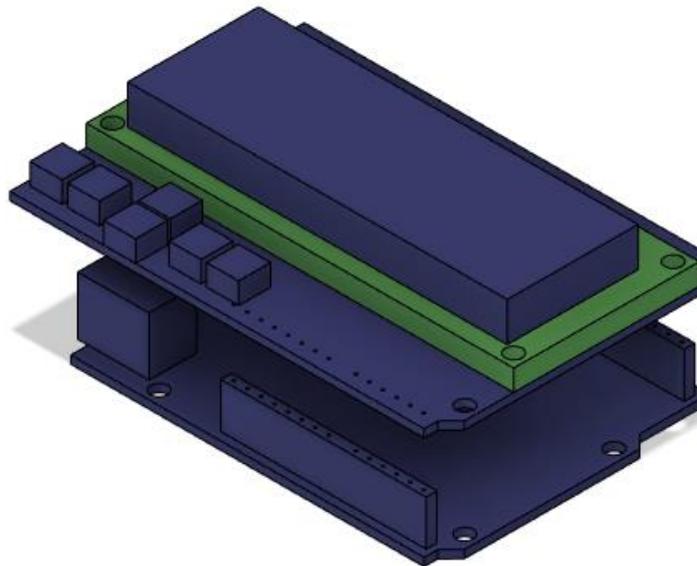


Figura 58. Modelo tridimensional de la electrónica

Se han incluido los logos del Grupo de Electrónica Industrial de la UMH y del IDiBE. Para conseguir un mejor acabado, los logos están diseñados en otro color que la carcasa. Conseguir imprimir esto con una impresora 3D mono cabezal es un proceso tedioso pero sencillo si se planifica desde la fase de diseño. Es por ello por lo que los logos están diseñados de forma que midan exactamente 0.4 mm de grosor. Esto es la medida de dos capas de impresión 3D estándar. Además, han sido exportados al archivo 3D final por separado. Veremos más detalles sobre esto en el apartado 5.2.2. Los planos detallados de ambas carcasas pueden encontrarse en el anexo 10.2

5.2.1.1. DISEÑO DE LA CARCASA DEL CUADRO DE CONTROL

La carcasa del cuadro de control, que puede verse en la Figura 59, ha sido diseñada en dos partes unidas mediante dos tornillos Allen

5. Fabricación sistema final

El diseño incluye un hueco para la pantalla LCD, espacio para la botonera, otro para el conector *Jack* y un último para la bornera. Además, se han incluido numerosas rendijas de ventilación para evitar que se sobrecaliente el MOSFET.

Para mejorar la experiencia del usuario, se han diseñado unos pulsadores para la botonera que ayudan a integrar el diseño en una sola pieza.



Figura 59. Diseño final del cuadro de control

5.2.1.2. DISEÑO DE LA CARCASA DEL SISTEMA CALEFACTOR

La carcasa del sistema calefactor sigue el mismo tipo de diseño en dos partes que la del anterior sistema. En este caso, al ser la única placa de electrónica de diseño propio se han combinado ambos diseños para poder disponer de cuatro agujeros de montaje. Los tornillos utilizados en estos agujeros son lo

5. Fabricación sistema final

suficientemente largos para hacer además la función de patas del sistema. Esta carcasa puede verse en la Figura 60.

En cuanto a la ventilación, también cuenta con rendijas laterales. Estas rendijas son utilizadas para expulsar el aire caliente proveniente de la refrigeración de la aguja

Por último, el diseño cuenta con una rendija frontal grande para la salida de cables y una pequeña rendija lateral donde es situada la aguja.



Figura 60. Diseño de la carcasa de la unidad calefactora

5. Fabricación sistema final

5.2.2. IMPRESIÓN 3D

Para la impresión 3D se ha utilizado una impresora FDM⁴ mono cabezal. Conseguir el efecto multicolor en este tipo de impresoras 3D es algo tedioso pero sencillo si se ha diseñado teniendo en cuenta el trabajo de impresión. Para ello se han preparado dos archivos distintos. Por un lado, tenemos el archivo que contiene las letras y por otro el archivo que contiene el diseño general. El proceso consiste en cuadrar las coordenadas de ambos archivos en el *software* de impresión y preparar ambos trabajos de impresión por separado. Después, una vez en la impresora realizamos primero la impresión de las letras. Cuando hayan acabado y sin retirarlas de la superficie de trabajo, ejecutamos el otro trabajo de impresión. De esta forma pueden completarse ambos trabajos de impresión de manera que los dos archivos queden integrados.

En cuanto al material de impresión, se ha utilizado el filamento de impresión 3D de PLA INGEO 850 de la empresa 3DFILAMENTS S.L. [36]. La elección de este material viene motivada por su resistencia a altas temperaturas, sobre todo en el rango que vamos a trabajar. Debido a su temperatura de reblandecimiento puede contener la parte calefactora sin ningún problema.

Por último, en la Tabla 21, se encuentra los parámetros de impresión utilizados en estas piezas:

Tabla 21. Parámetros de impresión 3D utilizados para la fabricación de las carcasas del sistema

Altura de capa	0.2 mm
Temperatura de impresión	210°C
Temperatura de la base de impresión	50°C
Velocidad de impresión	50 mm/s

⁴ Las siglas FDM (Fused deposition modeling) hacen referencia a una tecnología de fabricación aditiva que consiste en la deposición de material plástico fundido en sucesivas capas de forma controlada.

5. Fabricación sistema final

5.3. PRESUPUESTO DE FABRICACIÓN

A continuación, en la Tabla 22, se muestra un cuadro con los costes de los componentes empleados para la fabricación del dispositivo.

Tabla 22. Presupuesto de componentes

Componente (código Digkey)	Precio Unitario (€)	Cantidad Usada	Precio (€)	Precio Total (€)	Precio cada 10(€)	Precio cada 100 (€)
Convertor CC/CC Reemplazo del regulador lineal 1 Salida 5V 1A Entrada 8V 28V (945-2201-ND)	2,91	1	2,91	2,91	2,82	2,57
Condensadores electrolíticos de aluminio 10µF 400V Radial, encapsulado 10000 h a 105°C (493-13313-1-ND)	0,73	2	1,46	1,46	0,538	0,369
Power Barrel Connector Jack 2.10mm DI (0.083"), 5.50mm DE (0.217") Orificio pasante, ángulo recto (839-1516-ND)	0,87	1	0,87	0,87	0,758	0,654
Transistor bipolar (BJT) NPN 40V 800mA 300MHz 500mW Orificio pasante TO-18 (2N2222ACS-ND)	2,19	2	4,38	4,38	1,966	1,5802
Transistor bipolar (BJT) PNP 60V 600mA 200MHz 400mW Orificio pasante TO-18 (2N2907ACS-ND)	2,19	1	2,19	2,19	1,966	1,5802
Canal N Orificio pasante 100V 5.6 A (Tc) 43W (Tc) TO- 220AB (IRF510PBF-ND)	0,79	1	0,79	0,79	0,702	0,5472
Resistencias de orificio pasante Axial Revestimiento pirorretardante, seguridad Película de carbono 4.7 kΩs ±5% 0.25W, 1/4W (CF14JT4K70CT- ND)	0,09	3	0,27	0,27	0,036	0,0145
Resistencias de orificio pasante Axial Revestimiento pirorretardante, seguridad Película de carbono 330 Ωs ±5% 0.25W, 1/4W	0,09	1	0,09	0,09	0,036	0,0145

5. Fabricación sistema final

(CF14JT330RCT-ND)						
ATmega328 Arduino Uno SMD R3 AVR® AT mega AVR MCU 8 bits Embedded Evaluation Board (1050-1041-ND)	19,63	1	19,63	19,63	19,63	19,63
Botón Interfaz humana Arduino Platform Evaluation Expansion Board (DFR0009-ND)	9,1	1	9,1	9,1	9,1	9,1
Placas Arduino u otras placas MCU – Conector (1568-1413-ND)	1,34	1	1,34	1,34	1,34	1,34
Resistencias de orificio pasante TO- 220-2 Detección de corriente, a prueba de llamas, resistente a pulsos, seguridad Película delgada 5 Ωs ±1% 50W (696-1354-ND)	3,93	1	3,93	3,93	3,553	2,6457
sensor de temperatura Digital, local -40°C ~ 125°C 7 b TO-220-5 (TC74A3-5.0VAT- ND)	1,26	1	1,26	1,26	1,26	0,9556
Una sola unidad:	Cien Unidades:					
51,32 €	45,11 €					

Para el cálculo de los costes de impresión 3D se ha utilizado la herramienta de cálculo de presupuestos de la empresa 3DFILAMENTS.S.L.[37] (Figura 61). Con el fin de simplificar los cálculos se ha incluido el precio de la mano de obra dentro del presupuesto de impresión. Los resultados pueden observarse en la Figura 62

5. Fabricación sistema final

Coste material[€/kg]:	<input type="text" value="24"/>
Coste Luz[€/kWh]:	<input type="text" value="0.14"/>
Consumo medio[W]:	<input type="text" value="300"/>
Coste impresora[€]:	<input type="text" value="3000"/>
Tiempo amortización[años]:	<input type="text" value="5"/>
Horas de impresión diarias:	<input type="text" value="15"/>
Tasa de fallos[%]:	<input type="text" value="2"/>
Mano de obra[€/h]:	<input type="text" value="10"/>
Horas de pre y post procesado:	<input type="text" value="2"/>
Masa de la pieza[g]:	<input type="text" value="101"/>
Tiempo de impresión[h]:	<input type="text" value="11.8"/>
<input type="button" value="Calcular"/>	

Figura 61. Calculadora de costes de impresión 3D

Electricidad:	0.50€
Material:	2.42€
Mano de obra:	20.00€
Amortización	1.89€
Total:	25.30€

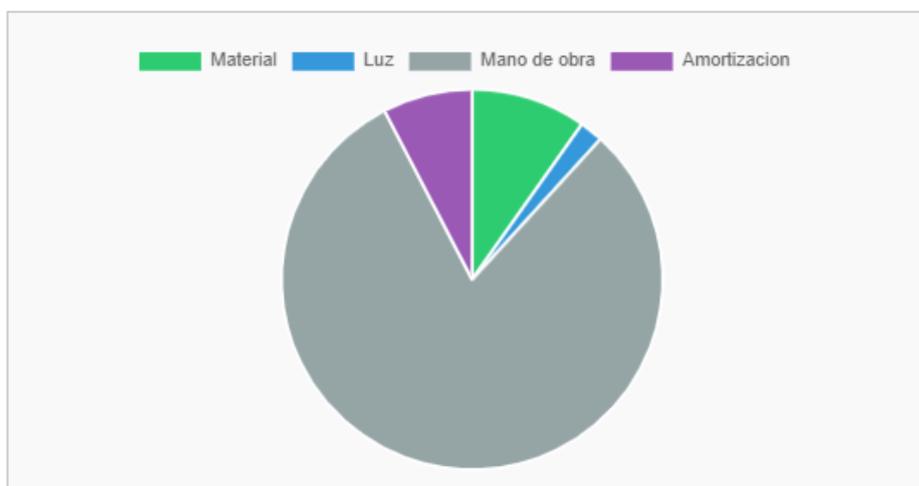


Figura 62. Costes de impresión 3D

5. Fabricación sistema final

Teniendo en cuenta ambos presupuestos, los costes finales de fabricación para 1, 10 y 100 se muestran en la Tabla 23. Hay que tener en cuenta que los costes de mano obra se reparten por unidad fabricada según la cantidad. De este modo, la misma mano de obra es requerida para la fabricación de 1 y de 10 unidades.

Tabla 23. Presupuesto de fabricación del dispositivo

	Coste de componentes	Coste de impresión 3D	Coste de mano de obra	Coste unitario	Total
1 unidad	51,32 €	4,81 €	20,00 €	76,13 €	76,13 €
10 unidades	49,12 €	4,81 €	2,00 €	55,93 €	559,30 €
100 unidades	45,11 €	4,81 €	2,00 €	51,92 €	5.192,00 €



6. PRUEBAS EN ENTORNO REAL

Para poder validar el dispositivo y considerarlo apto para su uso en trabajos de investigación deben llevarse a cabo una serie de pruebas inspiradas en la aplicación real del sistema que corroboren su correcto funcionamiento en un entorno de trabajo real.

Lamentablemente debido a la situación excepcional provocada por la crisis del covid-19 estas pruebas no han podido ser realizadas. Aun así, en este apartado van a definirse en profundidad estas pruebas para que puedan ser realizadas en futuros trabajos.

6.1. MEDICIÓN DE TEMPERATURA CON CÁMARA TÉRMICA

Con el fin de corroborar el correcto funcionamiento del calentador de microperfusión es necesario comprobar que las lecturas proporcionadas por el sensor de temperatura sean correctas. Para esto es necesario medir las temperaturas alcanzadas en el sistema mediante un dispositivo externo al sistema correctamente calibrado.

Para ello se ha considerado que la mejor manera de realizar esta prueba es mediante el uso de una cámara térmica. Las cámaras térmicas son dispositivos capaces de detectar la radiación infrarroja emitida por un cuerpo y traducir esta información en una imagen coloreada visible para el ojo humano.

El procedimiento es el siguiente. Primero debemos establecer el setpoint en una temperatura conocida y dejar que el sistema alcance la temperatura deseada. Después analizaremos con la cámara térmica si la temperatura alcanzada, indicada por el sensor del sistema, se corresponde con la temperatura real. Hemos de tener en cuenta que la precisión del sistema debería ser de $\pm 1^{\circ}\text{C}$. Cualquier desviación superior a este valor no es válida. Si este fuese

6. Pruebas en entorno real

el caso deberían de ser revisados tanto los elementos calefactores del sistema como el algoritmo de control de temperatura y el propio sensor de temperatura.

6.2. PRUEBA DEL DISPOSITIVO CON CIRCULACIÓN DE AGUA

El objetivo de esta prueba es observar si el comportamiento y la velocidad de calentamiento del sistema varían ante la circulación de fluido.

El fluido circulante por el sistema absorberá parte del calor generado por las resistencias. Este hecho afectará principalmente a la velocidad de calentamiento del sistema. Con esta prueba debemos comprobar que el retraso provocado en el calentamiento por la presencia de fluido no haga que el tiempo de establecimiento supere el tiempo objetivo.

Para realizar la prueba deberemos realizar un montaje similar al que será utilizado en la aplicación real. En este montaje contamos con el sistema conectado un embolo mediante un tubo plástico. Estos émbolos se sitúan en altura utilizando un soporte universal de laboratorio. Lo situamos en altura para facilitar la circulación del fluido. El efecto de la altura en la prueba se analizará en el capítulo 6.4.

Una vez realizado el montaje realizaremos una primera prueba midiendo el tiempo de establecimiento de la temperatura sin circulación de fluido. Con este dato registrado, realizaremos una segunda prueba con fluido en el sistema. El resultado de ambas pruebas debe servirnos para corroborar que la presencia de fluido no supone un aumento significativo en el tiempo de establecimiento.

6. Pruebas en entorno real

6.3. PRUEBA DEL DISPOSITIVO CON CIRCULACIÓN DE SOLUCIÓN DE LABORATORIO

Para comprobar el correcto funcionamiento del sistema en un entorno de trabajo real, es importante asegurar que la respuesta del sistema sigue siendo válida cuando se utiliza el mismo fluido utilizado en la aplicación final. La composición de este fluido se muestra en la Tabla 24.

Tabla 24. Composición de la solución de laboratorio empleada por el IDiBE

NaCl	0.14M
KCl	0,004M
CaCl ₂	0,002M
MgCl ₂	0.002M
HEPES	0.01M
Glucosa	0.005M
D-Manitol	0.02M

La importancia de esta prueba reside en que el calor específico de cada fluido afecta a la velocidad de establecimiento del sistema. En cuanto a la prueba en sí, su desarrollo es igual a la del capítulo 6.2.

6.4. PRUEBA DE MEDIDA DE LA TEMPERATURA REAL

Las pruebas planteadas anteriormente se han centrado en el correcto funcionamiento del sistema calefactor, pero para validar el dispositivo para su uso en un entorno de trabajo real es necesario comprobar que la temperatura alcanzada por el propio fluido es la correcta. Debemos asegurarnos de que el fluido que finalmente será introducido en el MEA se encuentre a la temperatura marcada como *setpoint*.

6. Pruebas en entorno real

El proceso de calentamiento por conducción no ocurre instantáneamente, sino que requiere de tiempo para que la temperatura de ambos cuerpos se iguale. En nuestro caso al encontrarse un fluido en circulación por el elemento calefactor, es necesario comprobar que el tiempo de residencia del fluido en el interior de este es suficiente como para alcanzar la temperatura deseada.

Para comprobarlo, debemos realizar el mismo montaje que en el capítulo 6.2, solo que en esta prueba nuestro objetivo es medir la temperatura del fluido en el instante que abandona el sistema. Para realizar la medición puede utilizarse una cámara térmica o un termómetro externo.

6.5. PRUEBA DEL DISPOSITIVO CON DISTINTAS VELOCIDADES DE FLUIDO

Con lo visto en el apartado anterior, podemos concluir que el tiempo de residencia es un parámetro que nos indica el tiempo mínimo que ha de estar el fluido dentro del sistema para alcanzar la temperatura correcta. Expresado de otra forma, existe un caudal máximo al cual puede circular el fluido por el interior del sistema. A partir de este, el dispositivo no será capaz de suministrar energía suficientemente rápido como para que el fluido alcance el *setpoint*.

Para establecer un rango de caudales de fluido funcionales para el sistema, será necesario desarrollar una prueba en la que, haciendo circular distintos caudales, midamos la temperatura de salida del fluido y la comparemos con el *setpoint*.

El montaje de la prueba utilizará los mismos elementos que las pruebas anteriores, pero esta vez su configuración será diferente. En este caso el soporte universal irá variando su altura, de esta manera podremos regular el caudal de entrada de fluido al sistema. El microcalentador de perfusión será colocado en vertical en línea con el tubo que transporta el fluido desde el émbolo. Esto es

6. Pruebas en entorno real

debido a que de esta forma evitaremos tener pérdidas de carga⁵ que afecten a el caudal del fluido.

Para regular el caudal del fluido variaremos la altura del embolo. El valor del caudal del fluido deberá estimarse de forma experimental, midiendo la variación de volumen en la jeringuilla y dividiendo entre el tiempo. Con el fin de evitar desviaciones es aconsejable realizar la prueba varias veces para cada caudal. Este procedimiento puede aplicarse de forma inversa para obtener la altura necesaria a la que colocar el embolo dado un caudal objetivo.



⁵ La pérdida de carga en una tubería o canal es la pérdida de presión que se produce en un fluido debido a la fricción de las partículas del fluido entre sí y contra las paredes de la tubería que las conduce.

7. CONCLUSIONES

El objetivo del presente trabajo era el de desarrollar un dispositivo calefactor de bajo coste para equipos de perfusión que sirviera como alternativa a los equipos profesionales en etapas tempranas del trabajo de laboratorio. Para ello, era necesario conocer con detalle las características principales que ofrecen los equipos comerciales. Con este fin se ha realizado un estudio de mercado en el que se han analizado los dispositivos de las tres principales empresas que fabrican equipos para este sector. Con la información obtenida y conociendo los requerimientos del IDiBE se han establecido los requisitos sobre los que se ha desarrollado todo el proyecto.

La primera parte del trabajo ha consistido en determinar qué elementos de *hardware* iban a conformar el sistema. Para ello se han realizado múltiples pruebas con los componentes principales del sistema. Uno de los aspectos más destacables de este punto es la concepción del dispositivo mediante el uso de *Shields*. Esto ha permitido crear un dispositivo con capacidad de ser mejorado y ampliado en futuros trabajos, así como con facilidad de ser reparado.

En cuanto al *hardware*, sin duda, otro de los puntos clave, es el diseño del chasis donde se sitúan los elementos calefactores. Para llegar al diseño final se han descartado varios prototipos. El principal problema con ellos ha sido la falta de medios técnicos para su fabricación. Aun así, el diseño final destaca y se diferencia de los productos comerciales en el tipo de instalación de la aguja que conduce el fluido a calentar. En este diseño, la aguja es fácilmente sustituible lo que reduce el coste de reparación y le otorga mayor versatilidad.

Una vez elegido y configurado el *hardware*, lo siguiente ha sido diseñar el *software*. Se ha elegido trabajar con Arduino por su bajo coste, fácil reparación y simplicidad, todos ellos objetivos del proyecto. A la hora de realizar un controlador de temperatura existen diversas opciones. En este trabajo se han analizado las más populares, enfrentando sus resultados hasta decantarnos por la implementación de un controlador PID.

7. Conclusiones

Para obtener los parámetros de control necesarios para el correcto funcionamiento del controlador, se ha trabajado con un algoritmo de *auto-tune* conocido como *relay method*. Gracias a la implementación de este algoritmo, se ha conseguido un sistema robusto y preciso que cumple con los requisitos del proyecto tal y como puede observarse en el capítulo 3.2.2. La precisión de $\pm 1^{\circ}\text{C}$ obtenida en el control de temperatura es equiparable a la proporcionada por algunos equipos comerciales vistos en el estudio de mercado.

El precio del dispositivo también ha cumplido con los requisitos del proyecto, establecido el presupuesto inicial en 400€. En este punto es importante destacar que, a diferencia de los productos analizados en el estudio de mercado, al dispositivo desarrollado no se le han imputado los costes típicos de la actividad empresarial. Este hecho no desvirtúa la comparación ya que el objetivo de este trabajo no es desarrollar un dispositivo comercial sino una alternativa de bajo coste que pueda ser fabricada en los centros de investigación.

En conclusión, podemos considerar que el trabajo presentado ha cumplido los objetivos propuestos y que ha resultado en el desarrollo de un dispositivo calefactor de bajo coste para equipos de perfusión. El desarrollo de este trabajo demuestra que es posible la fabricación de equipos de bajo coste para aplicaciones científicas que cumplan con los objetivos de etapas tempranas de investigación.

8. Mejoras y futuras líneas de desarrollo

8. MEJORAS Y FUTURAS LÍNEAS DE DESARROLLO

El dispositivo desarrollado cumple con los objetivos propuestos en este trabajo, aun así, queda lejos de un dispositivo profesional que pueda competir con las alternativas del mercado.

Aun habiendo obtenido una precisión en el control de $\pm 1^{\circ}\text{C}$, equiparable a algunos dispositivos del mercado, existen diversas aplicaciones que exigen un nivel de precisión mayor. Para aumentar la precisión del calentador sería necesario trabajar con un sensor de temperatura que también ofreciera una mayor exactitud a la hora de medir la temperatura del sistema. Además, podría profundizarse más en la parametrización del controlador PID e implementar otros algoritmos de *auto-tune*.

En cuanto al *hardware*, podrían implementarse diversos elementos que mejoren el rendimiento y prestaciones del dispositivo. La adición de un ventilador al conjunto haría que el sistema fuese capaz de regular mejor la temperatura y ofrecer una respuesta más rápida ante descensos bruscos del *setpoint*. Para controlarlo podría implementarse un controlador *On/off* o utilizarse la salida negada del actual controlador PID. Este punto fue considerado en etapas incipientes del proyecto, pero finalmente fue descartado al no encontrar una forma óptima de incorporar un ventilador al sistema. Puede observarse en los esquemáticos del sistema final que ya está preparado para su incorporación.

Otro elemento que dotaría al sistema de más versatilidad sería la integración de más entradas para sensores de temperatura de cualquier tipo. Para poder abarcar el mayor tipo de sensores posible, una posible opción sería la de integrar algún estándar de conexión como I2C, USB o algunos más modernos como el *IO-Link*. Queda pendiente para futuros trabajos el determinar cuál sería el más adecuado.

8. Mejoras y futuras líneas de desarrollo

En cuanto a los algoritmos de control, para mejorar su adaptabilidad a distintas condiciones de trabajo, sería necesario implementar una función que permitiera que el usuario pudiera introducir el setpoint al ejecutar el algoritmo de Autotune. De esta forma le estaríamos permitiendo que la dinámica del controlador se asemejara más a la dinámica del sistema a controlar.

Como último punto a profundizar en futuros trabajos, queda pendiente implementar una interfaz de usuario más avanzada. Debido al limitado espacio de la pantalla con la que se ha trabajado, no ha sido posible crear una representación profunda de los datos que ofrecer al usuario. En este tipo de dispositivos se valora el hecho de poder ofrecer gráficas e históricos, así como la posibilidad de poder exportar los datos para poder trabajar con ellos. Para subsanar esto se podría aprovechar la estructura modular del dispositivo para sustituir la actual pantalla por una pantalla *LCD* táctil o utilizar la conexión USB de Arduino para programar una interfaz en el ordenador.



9. Referencias

9. REFERENCIAS

[1] Merriam Webster. (s. f.). microperfusion. Recuperado de <https://www.merriam-webster.com/medical/microperfusion>

[2] L. Butron. (s. f.). Multi Electrode Array

[3] Multi Channel Systems MCS GmbH. (s. f.). MEA2100-Systems. [Fotografía] Recuperado de <https://www.multichannelsystems.com/products/mea2100-systems#meas>

[4] Everaerts, W., Gees, M., Alpizar, Y. A., Farre, R., Leten, C., Apetrei, A., ... & Nilius, B. (2011). The capsaicin receptor TRPV1 is a crucial mediator of the noxious effects of mustard oil. *Current Biology*, 21(4), 316-321.

[5] Islas, L. D., De-la-Rosa, V., Rodríguez-Cortés, B., Rangel-Yescas, G. E., & Elias-Viñas, D. (2015). A simple method for fast temperature changes and its application to thermal activation of TRPV1 ion channels. *Journal of neuroscience methods*, 243, 120-125.

[6] Omega Engineering Limited. (s. f.). PID Controller: Types, What It Is & How It Works. Recuperado de <https://www.omega.co.uk/prodinfo/pid-controllers.html>

[7] Stack Exchange Inc. . (s. f.). Definición de overshoot [Gráfico]. Recuperado de <https://electronics.stackexchange.com/questions/179878/exact-definition-of-overshoot>

[8] Multi Channel Systems MCS GmbH. (s. f.-b). www.multichannelsystems.com | Innovations in Electrophysiology. Recuperado de <https://www.multichannelsystems.com/>

[9] Digitimer North America, LLC. (s. f.). Digitimer | Research and Clinical Instruments Manufacturer. Recuperado de <https://digitimer.com/>

9. Referencias

- [10] Warner Instruments, LLC. (s. f.). Warner Instruments. Recuperado de <https://www.warneronline.com/>
- [11] Multi Channel Systems MCS GmbH. (s. f.-b).PH01. Recuperado de <https://www.multichannelsystems.com/products/ph01>
- [12] Digitimer North America, LLC. (s. f.-b). ThermoClamp Heated Perfusion. Recuperado de <https://digitimer.com/products/research-electrophysiology/perfusion-systems/new-thermoclamp-heated-perfusion/>
- [13] Warner Instruments, LLC. (s. f.-a). (SH-27B, SF-28) Single Inline Solution Heaters. Recuperado de <https://www.warneronline.com/single-inline-solution-heaters-sh-27b-sf-28>
- [14] Warner Instruments, LLC. (s. f.-b). (SWS Series) Syringe Warmer. Recuperado de <https://www.warneronline.com/syringe-warmer-sws-series>
- [15] AutoMate Scientific, Inc. (s. f.). ThermoClamp-1 Temperature Control System [1 Ch.]. Recuperado de <https://www.autom8.com/shop/temperature-control/thermoclamp-inline-heater/thermoclamp-1-temperature-control-system-1-ch/>
- [16] Cinjordiz, C. (2018, 13 marzo). Arduino Uno R3, tutorial especificaciones electrónicas y programación. Recuperado de <https://www.infootec.net/arduino/>
- [17] Llamas, L. (2015, 10 abril). Relación entre temperatura y resistencia en un termistor [Gráfico]. Recuperado de <https://www.luisllamas.es/wp-content/uploads/2015/04/arduino-termistor-grafica.png>
- [18] NXP Semiconductors N.V. (2014). I2C-bus specification and user manual. Recuperado de <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>

9. Referencias

[19] Quartz Bubble Heaters. (s. f.). Lámpara calefactora infrarroja [Fotografía]. Recuperado de <http://spanish.quartztubeheater.com/sale-11438316-sk15-single-tube-quartz-infrared-tube-heating-elements-industrial-infrared-heaters.html>

[20] Heater Factory. (s. f.). Calentador Infrarrojo de anillo [Fotografía]. Recuperado de <http://www.cnheatingelement.com/band-heaters/quartz-infrared-band-heater/china-customized-nano-far-infrared-quartz.html>

[21] Llamas, L. (2016, 29 julio). Diagrama de una placa Peltier [Diagrama]. Recuperado de <https://www.luisllamas.es/arduino-peltier/>

[22] Riedon Inc. (2019). PF2200 Series Hoja de datos. Recuperado de <https://riedon.com/media/pdf/PF2200.pdf>

[23] Gudino, M. (2017, 12 julio). Tipos de convertidores interruptores de CC a CC. Recuperado de <https://www.arrow.com/es-mx/research-and-events/articles/types-of-switching-dc-dc-converters>

[24] Recom Power. (s. f.). R-78E5.0-0.5. Recuperado de https://www.digikey.es/product-detail/en/recom-power/R-78E5.0-0.5/945-1648-5-ND/2834904?utm_adgroup=Power%20Supplies&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&productid=&gclid=Cj0KCQjwzZj2BRDVARIsABs3l9JMuH80sGxY3yipC-UN2dWOCdzXdWXk3SCP1wG27XAf_P8MD08bhoaAhSFEALw_wcB

[25] colaboradores de Wikipedia. (2020, 11 febrero). Driver. Recuperado de <https://es.wikipedia.org/wiki/Driver>

[26] Wikipedia contributors. (s. f.). Push–pull output. Recuperado de https://en.wikipedia.org/wiki/Push%E2%80%93pull_output.

9. Referencias

[27] Minibots. (2016, 3 febrero). LCD Keypad Shield [Diagr ma]. Recuperado de <https://minibots.wordpress.com/2016/02/03/componente-lcd-keypad-shield-para-fritzing/>

[28] Autodesk | Fusion 360 Integrated CAD, CAM, and CAE software. (2020). Softwar. Recuperado de <https://www.autodesk.com/products/fusion-360/overview>

[29] Convective Heat Transfer Coefficients Table Chart. (s. f.). Recuperado de https://www.engineersedge.com/heat_transfer/convective_heat_transfer_coefficient_13378.htm

[30] colaboradores de Wikipedia. (2019, 26 septiembre). Emisividad. Recuperado de <https://es.wikipedia.org/wiki/Emisividad>

[31] Arduino IDE (Versi n 1.8.8) [Softwar]. (2019). Recuperado de <https://www.arduino.cc>

[32] Astrom, K. J., & Hagglund, T. (1995, septiembre). New tuning methods for PID controllers, 2456-2462. Presentado en 3rd European Control Conference, Roma, Italia.

[33] Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. trans. ASME, 64(11).

[34] EAGLE | PCB Design and Electrical Schematic Software (Versi n 9.5.1) [Softwar]. (2019). Recuperado de <https://www.autodesk.com>

[35] SparkFun Electronics. (s. f.). SparkFun Eagle Libraries. Recuperado de <https://github.com/sparkfun/SparkFun-Eagle-Libraries>

[36] 3DFILAMENTS S.L. (s. f.). esFil PLA INGEO 850. Recuperado de <https://www.3dfils.com/es/esfil-pla-ingeo-850>

9. Referencias

[37] 3DFilaments S.L. (s. f.). Calculadora costes impresión 3D. Recuperado de <https://www.3dfils.com/es/content/calculadora-costes-impresion-3d.html>



10. Anexos

10. ANEXOS

10.1. CÓDIGO

En este anexo se encuentra el código final que utiliza el sistema. Para facilitar su comprensión el código se ha estructurado en distintas librerías. Todo el código está perfectamente comentado explicándolo línea a línea.

10.1.1. MAIN.CPP

Este es el archivo principal del proyecto, el encargado de cargar todas las librerías, inicializar el sistema y ejecutar el bucle principal.

```
1. //Incluimos las librerias necesarias
2. #include "LCD.h"
3. #include "PID.h"
4. #include "memoria.h"
5.
6. void setup () {
7.     PID_inicio (); //Función de inicio del PID
8.     LCD_inicio (); //Función de inicio del LCD
9.     leer_eeeprom (); //Leemos los valores del PID de la memoria
10.     Eeprom
11.     menu1 ();
12. }
13. void loop() {
14.     medirtemp (); //En cada ciclo de ejecución leemos la
15.     temperatura
16.     pid(setpoint, 50); //Función encargada de ejecutar el PID
17.     autoTune(70, 5, 50, 250); //Función encargada de ejecutar
18.     el Autotune
19.     LCD_loop (); //Función encargada de controlar el contenido
20.     mostrado en la pantalla LCD
21. }
```

10. Anexos

10.1.2. LIBRERÍA LCD

Esta librería es la encargada de mostrar el menú e interactuar con el usuario.

10.1.2.1. LCD.CPP

```
1. //Incluimos las librerías necesarias
2. #include <LiquidCrystal.h>
3. #include<arduino.h>
4. #include "LCD.h"
5.
6. //Declaración de las variables necesarias para el uso del LCD
7. int menu_activo = 1; //Variable que controla que menú esta
  activo
8. int menu_cambio = 0; //Variable que controla si se ha producido
  un cambio en el menu activo
9. int menu_accion; //Variable que controla que tecla del LCD ha
  sido pulsada
10. int selector = 0; //Variable que indica la posición del
  selector en el menu principal
11. String estado; //Variable que almacena el mensaje ha
  mostrar al usuario
12. String elementos_menu[] = {"Ver temp", "Cambiar
  temp", "Calibrar", "Info"}; //Array que almacena los nombres de
  los distintos submenus
13.
14. //Arrays que almacenan caracteres personalizados que son
  necesarias para el menu
15.
16. byte menuCursor[8] = { //Almacena el caracter personalizado
  del selector
17.     B01000, // *
18.     B00100, // *
19.     B00010, // *
20.     B00001, // *
21.     B00010, // *
22.     B00100, // *
23.     B01000, // *
24.     B00000 //
25. };
26. byte grados[8] = { //Almacena el caracter personalizado
  del icono de grados
27.     B11100, //***
28.     B10100, //* *
29.     B11100, //***
30.     B00000, //
31.     B00000, //
32.     B00000, //
33.     B00000, //
34.     B00000 //
35. };
36. //Teclado
37.
38. int teclas = 5;
39. int tecla_in;
40. int tecla = -1;
41. int tecla_old = -1;
42.
43. void set_estado(String x) {
```

10. Anexos

```
44.     estado = x;
45.     }
46.
47.     void LCD_inicio() {
48.         //Inicialización de la pantalla LCD
49.         LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
50.         lcd.begin(16, 2);
51.         lcd.clear();
52.
53.         //Creamos los caracteres personalizados
54.         lcd.createChar(0, menuCursor);
55.         lcd.createChar(3, grados);
56.
57.         set_estado("Calentador uP");//Cambiamos el mensaje de
estado
58.
59.     };
60.
61.
62.     int devolver_tecla(unsigned int x) { //Esta función
devuelve que tecla ha sido pulsada en dunción del valor
analógico leído
63.         int result = 0;
64.         if (x < 50) {
65.             result = 1; // Derecha
66.         } else if (x < 195) {
67.             result = 2; // Arriba
68.         } else if (x < 380) {
69.             result = 3; // Abajo
70.         } else if (x < 790) {
71.             result = 4; // Izquierda
72.         }
73.         return result;
74.     }
75.
76.
77.     int accion() {
78.
79.         tecla_in = analogRead(0);    // Leemos el valor de la
pulsacion
80.         tecla = devolver_tecla(tecla_in);    // Obtenemos el
boton pulsado
81.
82.         if (tecla == 0) { //Si ninguna tecla ha sido pulsada
salimos de la función
83.             menu_cambio = 0;
84.             return 10;
85.         }
86.
87.         else    // Si se ha detectado pulsación
88.
89.         {
90.             delay(100); // Espera para evitar los rebotes de las
pulsaciones
91.             tecla_in = analogRead(0);    // Leemos el valor de la
pulsacion
92.             tecla = devolver_tecla(tecla_in);    // Obtenemos el
boton pulsado
93.             menu_cambio = 1; //indicamos que ha habido un cambio en
el menu
```

10. Anexos

```
94.         return tecla; //Devolvemos la tecla pulsada
95.
96.
97.     }
98. }
99.
100. //El contenido que debe ser dibujado en función del menu
    actual se almacena en una función independiente
101.
102. void menu1() { //Pantalla principal
103.     lcd.clear();
104.     lcd.setCursor(5, 0);
105.     lcd.print(temp);
106.     lcd.setCursor(7, 0);
107.     lcd.print("/");
108.     if (setpoint == 0) {
109.         lcd.print("00");
110.     } else {
111.         lcd.print(setpoint);
112.     }
113.     lcd.write(byte(3));
114.     lcd.print("C");
115.
116. }
117. void menu2() { //Menu de selección de temperatura
118.     lcd.clear();
119.     lcd.setCursor(1, 0);
120.     lcd.print("Introduce temp");
121.     lcd.setCursor(6, 1);
122.     lcd.print(setpoint_selector);
123.     lcd.write(byte(3));
124.     lcd.print("C");
125. }
126. void menu3() { //Menu de Autotune
127.     lcd.clear();
128.     lcd.setCursor(2, 0);
129.     lcd.print("Ejecutando");
130.     lcd.setCursor(2, 1);
131.     lcd.print("Calibracion");
132.
133. }
134. void menu4() { //Menu de info
135.     lcd.clear();
136.     lcd.setCursor(0, 0);
137.     lcd.print("Kp:");
138.     lcd.print(kp);
139.     lcd.print(" ");
140.     lcd.print("Ki:");
141.     lcd.print(ki);
142.     lcd.setCursor(0, 1);
143.     lcd.print("Kd:");
144.     lcd.print(kd);
145. }
146. void menu_principal() { //Menu principal
147.     lcd.clear();
148.
149.
150.
151.     switch (selector) { //En función de la posición del
        selector se dibujaran unos elementos u otros
```

10. Anexos

```
152.         case 0:
153.             lcd.setCursor(0, 0);
154.             lcd.write(byte(0));
155.             lcd.setCursor(1, 0);
156.             lcd.print(elementos_menu[0]);
157.             lcd.setCursor(1, 1);
158.             lcd.print(elementos_menu[1]);
159.             break;
160.         case 1:
161.             lcd.setCursor(0, 1);
162.             lcd.write(byte(0));
163.             lcd.setCursor(1, 0);
164.             lcd.print(elementos_menu[0]);
165.             lcd.setCursor(1, 1);
166.             lcd.print(elementos_menu[1]);
167.             break;
168.         case 2:
169.             lcd.setCursor(0, 0);
170.             lcd.write(byte(0));
171.             lcd.setCursor(1, 0);
172.             lcd.print(elementos_menu[2]);
173.             lcd.setCursor(1, 1);
174.             lcd.print(elementos_menu[3]);
175.             break;
176.         case 3:
177.             lcd.setCursor(0, 1);
178.             lcd.write(byte(0));
179.             lcd.setCursor(1, 0);
180.             lcd.print(elementos_menu[2]);
181.             lcd.setCursor(1, 1);
182.             lcd.print(elementos_menu[3]);
183.             break;
184.     }
185. }
186. }
187.
188. void controlar_menu() { //Esta es la principal funcion de
    control del LCD, en función de la tecla pulsada y el menú activo
    se ejecutan unas ordenes u otras.
189.     menu_cambio = 0;
190.     switch (menu_accion) {
191.         case 1: //pulsamos derecha
192.             switch (menu_activo) {
193.                 case 0: //Estando en el menu principal
194.                     switch (selector) {
195.                         case 0:
196.                             menu_activo = 1;
197.                             menu1();//vamos al menu 1
198.
199.                             break;
200.                         case 1:
201.                             menu_activo = 2;
202.                             menu2();//vamos al menu 2
203.
204.                             break;
205.                         case 2:
206.                             menu_activo = 3;
207.                             ejecutar = 2; //Ejecutamos el Autotune
208.                             menu3();//Vamos al menu 3
209.
```

10. Anexos

```
210.         break;
211.     case 3:
212.         menu_activo = 4;
213.         menu4();//vamos al menu 4
214.
215.         break;
216.
217.
218.     }
219.
220.     break;
221.
222.     case 1://Estando en el menu 1
223.     break;
224.     case 2://Estando en el menu 2
225.
226.         if (ejecutar != 2) { //si esta haciendo autotune
no podemos activar el pid
227.             setpoint = setpoint_selector;
228.             ejecutar = 1; //Ejecutamos el PID
229.
230.         }
231.         else {
232.             menu_activo = 3;
233.             menu3();
234.             break;
235.
236.
237.
238.
239.         }
240.         menu_activo = 1;
241.         menu1();
242.         break;
243.         break;
244.     case 3://Estando en el menu 3
245.     break;
246.     case 4://Estando en el menu 4
247.     break;
248.     case 5://Estando en el menu 5
249.     break;
250. }
251. break;
252. case 2: //pulsamos arriba
253.     switch (menu_activo) {
254.         case 0://Estando en el menu principal
255.             selector--;
256.             if (selector < 0) {
257.                 selector = 0;
258.             }
259.             menu_principal();
260.             break;
261.         case 2:
262.             setpoint_selector++;
263.             menu2();
264.             break;
265.     }
266.     break;
267. case 3://pulsamos abajo
268.     switch (menu_activo) {
```

10. Anexos

```
269.         case 0://estando en el menu principal
270.             selector++;
271.             if (selector > 3) {
272.                 selector = 3;
273.             }
274.             menu_principal();
275.             break;
276.         case 2:
277.             setpoint_selector--;
278.             menu2();
279.             break;
280.     }
281.     break;
282.     case 4: //pulsamos izquierda
283.         menu_activo = 0;
284.         menu_principal();
285.         break;
286.     }
287. }
288. }
289. }
290.
291.
292. void LCD_loop() { //Función que controla el menu
293.
294.     //Si nos encontramos en la pantalla principal
295.     //actualizamos solo la temperatura sin tener que redibujar, de
296.     //esta forma no refrescamos
297.     //el LCD constantemente
298.     if (menu_activo == 1 ) {
299.         lcd.setCursor(5, 0);
300.         lcd.print(temp);
301.         lcd.setCursor(1, 1);
302.         lcd.print(estado);
303.     }
304.
305.     menu_accion = accion(); //Comprobamos si se ha pulsado
306.     //alguna tecla
307.     if (menu_cambio == 1) { //Si se ha pulsado alguna tecla,
308.         //llamamos a la función de control del LCD
309.         controlar_menu();
310.     }
311. }
```

10. Anexos

10.1.2.2. LCD.H

```
1. extern int temp, setpoint, ejecutar, setpoint_selector;
2. extern double kp, ki, kd;
3.
4. void LCD_inicio();
5. void LCD_loop();
6. void set_estado(String x);
7. void menu_principal();
8. void menu1();
9. void menu2();
10.    void menu3();
11.    void menu4();
12.    void controlar_menu();
13.    int devolver_tecla(unsigned int x);
14.    int accion();
```

10.1.3. LIBRERÍA PID

10.1.3.1. PID.CPP

```
1. #include <Wire.h>
2. #include<arduino.h>
3. #include "PID.h"
4.
5. #define byte uint8_t
6.
7. //Declaración de variables
8.
9. int temp; //Variable que almacena la temperatura actual
10.    int setpoint_selector; //Variable que almacena el Setpoint
    introducido por el usuario
11.    int setpoint = 0; //Variable que almacena el Setpoint
    introducido por el usuariodel PID
12.    int ejecutar = 0; //ejecutar:0->apagado,1->pid,2->autotune
13.
14.    //Valores por defecto del PID
15.    double kp = 24.5;
16.    double ki = 0.39;
17.    double kd = 389.01;
18.
19.
20.    int last_temp, error, picomax, picomin;
21.    unsigned long tiempoi ;
22.    unsigned long tiempo = 0 ;
23.    int ciclo;
24.
25.    int picos[5];
26.    unsigned long tpicos[5];
27.    int posicion = 0;
28.    int valorpico;
29.
30.    int errorp = 0;
31.    int derivative, integral, tultimo;
32.    unsigned long tvalorpico;
```

10. Anexos

```
33.     byte start = true;
34.     byte maximo;
35.
36.
37.     const int resistencia = 3;
38.     const int ventilador = 9;
39.
40.     int TC74A0 = 72;
41.
42.     void PID_inicio() { //Función de inicio del PID
43.         Wire.begin(); //Iniciamos la comunicación I2C
44.     }
45.
46.     int medirtemp() { //Función que mide la temperatura
47.
48.         Wire.beginTransmission(TC74A0); //Establecemos conexión
con el sensor de temperatura
49.         Wire.write(0); //Pedimos el dato de temperatura al sensor
50.         Wire.endTransmission(); //Finalizamos la comunicación con
el sensor
51.         Wire.requestFrom(TC74A0, 1); //Leemos la temperatura
52.
53.         if (Wire.available()) { //Si la conexión ha sido correcta
54.             last_temp = temp; //Guardamos la temperatura actual
como la ultima
55.             temp = Wire.read(); //Leemos la ultima temperatura
56.
57.         }
58.         else {
59.
60.             if (temp == 0) {
61.                 temp = last_temp;
62.             }
63.         }
64.         return temp;
65.
66.
67.     }
68.
69.     void pid(int tempo, int iteration_time) { //Función del
controlador PID
70.
71.         if (ejecutar == 1) { //Si el PID esta activo
72.             medirtemp(); //Medimos la temperatura actual
73.             error = tempo - temp; //Calculamos el error
74.             integral = integral + (error * iteration_time); //Calcu
lamos el error acumulado
75.
76.             if (integral > 255) { //Anti Wind-Up
77.                 integral = 255;
78.             };
79.
80.             derivative = (error - errorp) / iteration_time; //Calcu
lamos el cambio del error
81.
82.
83.             int salida = kp * error + ki * integral + kd * derivati
ve; //Calculamos la salida del PID
84.
85.             //Limites de Salida de arduino
86.             if (salida > 255) {
```

10. Anexos

```
87.         salida = 255;
88.     }
89.     if (salida < 0) {
90.         salida = 0;
91.     }
92.
93.     errorp = error; //Almacenamos el error actual como el
ultimo error
94.     analogWrite(resistencia, salida); //Aplicamos el
control a la resistencia
95.
96.     if (temp == Setpoint) {
97.         set_estado("Calentamiento completo");
98.     }
99.     else {
100.        set_estado("Calentando");
101.    }
102.
103.    delay(iteration_time);
104. }
105. }
106.
107. void autoTune(int templ, int ciclos, int escalon, int tmues
treo) {
108.
109.     if (ejecutar == 2) {
110.         set_estado("Calibrando");
111.         setpoint = 0;
112.         if (!start) {
113.
114.             analogWrite(resistencia, 255);
115.             medirtemp();
116.             if (temp > templ) {
117.
118.                 start = true;
119.                 maximo = true;
120.                 return;
121.             }
122.             delay(250);
123.             return;
124.         }
125.
126.         unsigned long tactual = millis();
127.         if ((tactual - tultimo) > tmuestreo) {
128.
129.             if (maximo == true) {
130.
131.                 analogWrite(resistencia, 0);
132.                 tultimo = tactual;
133.                 tactual = millis();
134.                 medirtemp();
135.                 if (temp > valorpico) {
136.                     valorpico = temp;
137.                     tvalorpico = tactual;
138.                 }
139.
140.                 if (temp < templ - 2) {
141.
142.                     picos[ciclo] = valorpico;
143.                     tpicos[ciclo] = tvalorpico;
144.                     ciclo += 1;
```

10. Anexos

```
145.
146.         valorpico = 1000;
147.         maximo = false;
148.         Serial.print("Ciclo:");
149.         Serial.println(ciclo);
150.     }
151. }
152. if (maximo == false) {
153.
154.     analogWrite(resistencia, 255);
155.     tultimo = tactual;
156.     tactual = millis();
157.     medirtemp();
158.     if (temp < valorpico) {
159.         valorpico = temp;
160.         tvalorpico = tactual;
161.     }
162.
163.     if (temp > templ + 2) {
164.
165.         picos[ciclo] = valorpico;
166.         tpicos[ciclo] = tvalorpico;
167.         ciclo += 1;
168.         valorpico = 0;
169.         maximo = true;
170.         Serial.print("Ciclo:");
171.         Serial.println(ciclo);
172.     }
173. }
174. }
175. if (ciclo == ciclos) {
176.     for (int c = 0; c < 10; c++) {
177.         if (picos[c] > picomax) {
178.             picomax = picos[c];
179.         }
180.     }
181.     for (int d = 0; d < 10; d++) {
182.         if (picos[d] < picomin) {
183.             picomin = picos[d];
184.         }
185.     }
186.     double ku = 4 * 255 / (picomax - picomin) * 3.1416;
187.     kp = 0.6 * ku;
188.     ki = 1.2 * ku / (tpicos[2] - tpicos[0]);
189.     kd = 0.075 * ku * (tpicos[2] - tpicos[0]);
190.     ejecutar = 0;
191.     grabar_parametros();
192.     set_estado("Calibracion completada");
193.     return;
194. }
195. }
196. }
```

10. Anexos

10.1.3.2. PID.H

```
1. extern void set_estado(String x);
2. extern void grabar_parametros();
3. void PID_inicio();
4. void pid(int tempo, int iteration_time);
5. void autoTune(int templ, int ciclos, int escalon, int tmuestreo)
;
6. int medirtemp();
```

10.1.4. LIBRERÍA MEMORIA

Esta librería contiene las funciones necesarias para leer y escribir de la eeprom

10.1.4.1. MEMORIA.CPP

```
1. #include <EEPROM.h>
2. #include "memoria.h"
3.
4. int eeprom_addr = 0;
5.
6. void grabar_parametros() {
7.     EEPROM.put( eeprom_addr, kp ); //Grabamos el valor
8.     eeprom_addr += sizeof(double); //Obtener la siguiente
posicion para escribir
9.     EEPROM.put( eeprom_addr, ki ); //Grabamos el valor
10.    eeprom_addr += sizeof(double); //Obtener la siguiente
posicion para escribir
11.    EEPROM.put( eeprom_addr, kd ); //Grabamos el valor
12.    eeprom_addr = 0; //Obtener la siguiente posicion para
escribir
13.
14. }
15. void leer_eeprom() {
16.     EEPROM.get( eeprom_addr, kp ); //Leemos el valor
17.     eeprom_addr += sizeof(double); //Obtener la siguiente
posicion para leer
18.     EEPROM.get( eeprom_addr, ki ); //Leemos el valor
19.     eeprom_addr += sizeof(double); //Obtener la siguiente
posicion para leer
20.     EEPROM.get( eeprom_addr, kd ); //Leemos el valor
21.     eeprom_addr = 0; //Obtener la siguiente posicion para
leer
22.
23. }
```

10. Anexos

10.1.4.2. MEMORIA.H

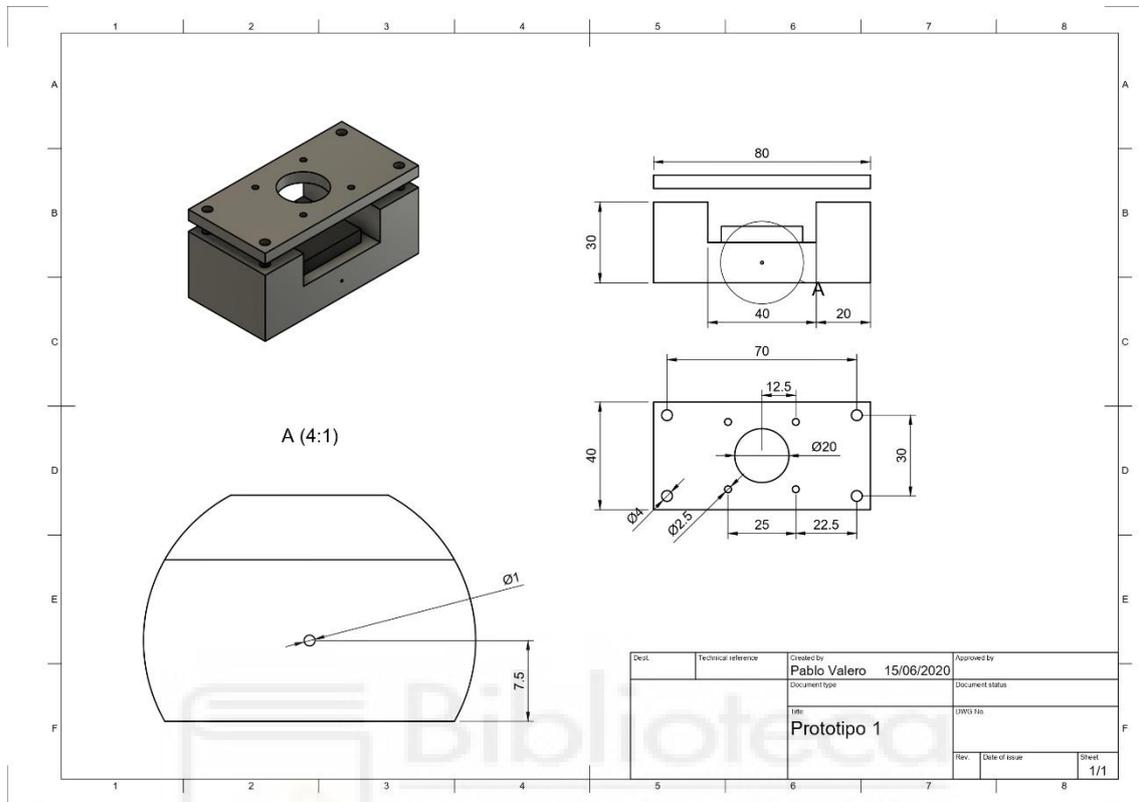
```
1. extern double kp, ki, kd;  
2.  
3. void grabar_parametros();  
4. void leer_eeprom();
```



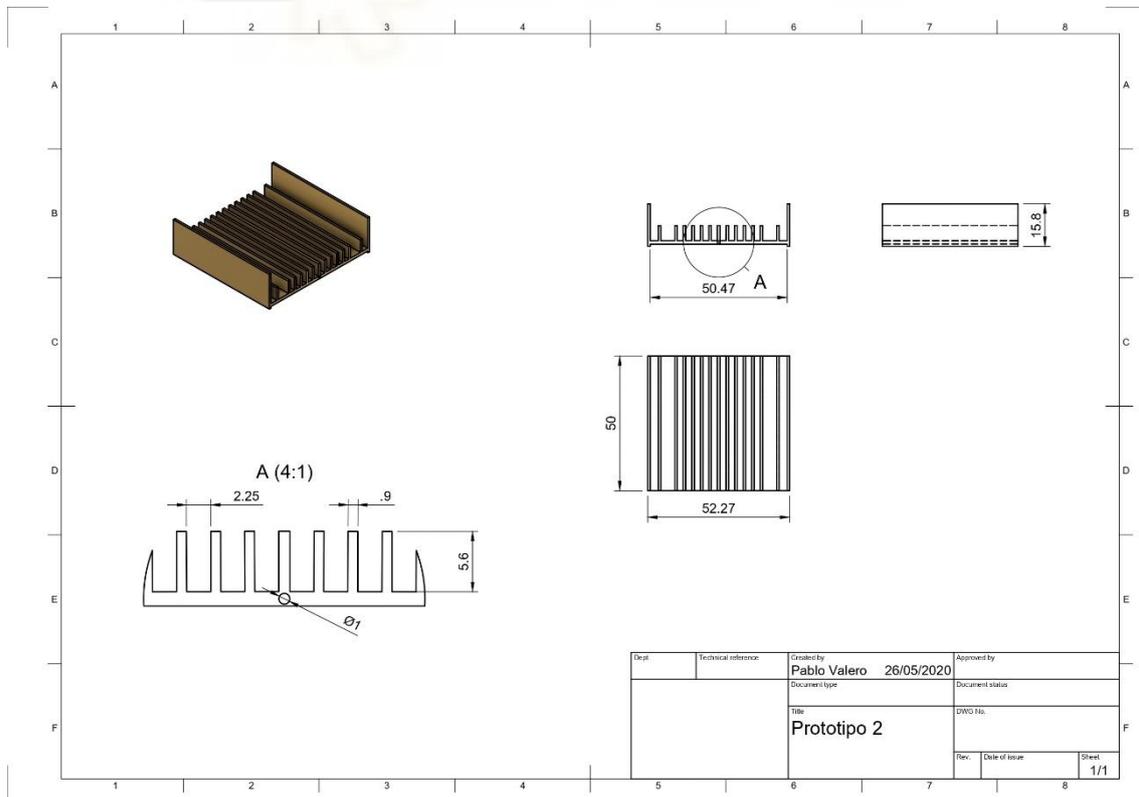
10. Anexos

10.2. PLANOS

10.2.1. PROTOTIPO 1

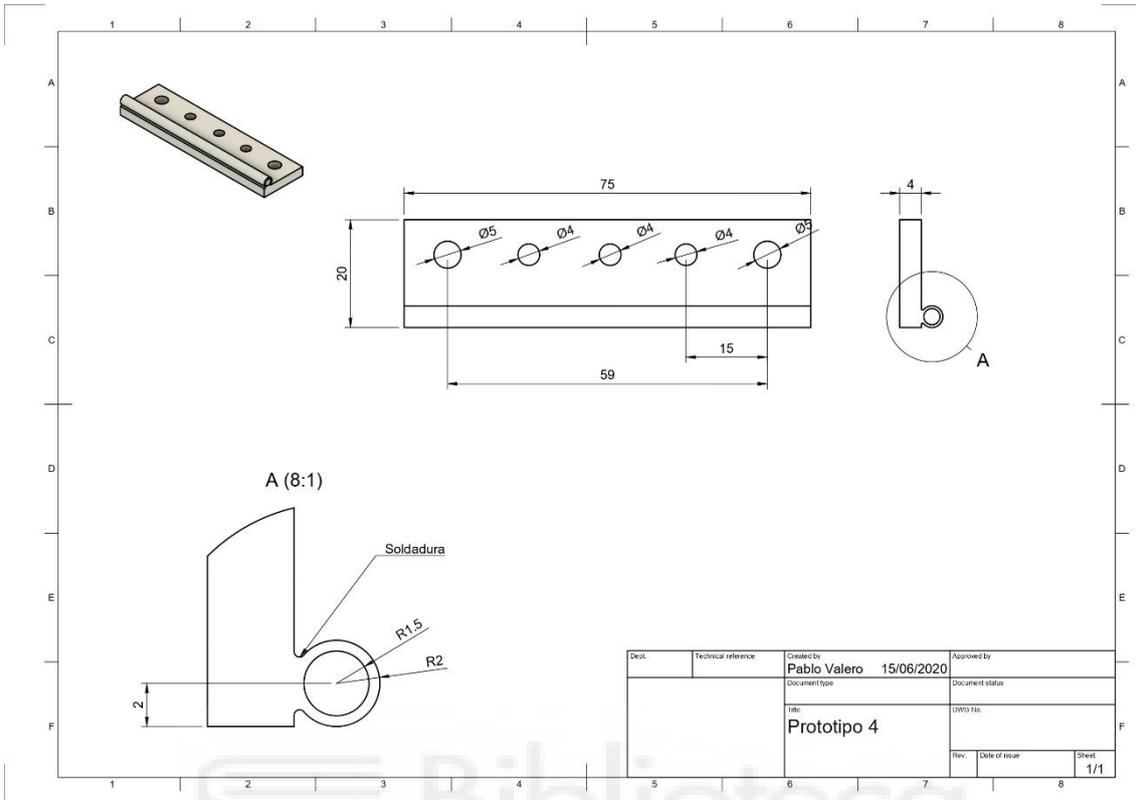


10.2.2. PROTOTIPO 2

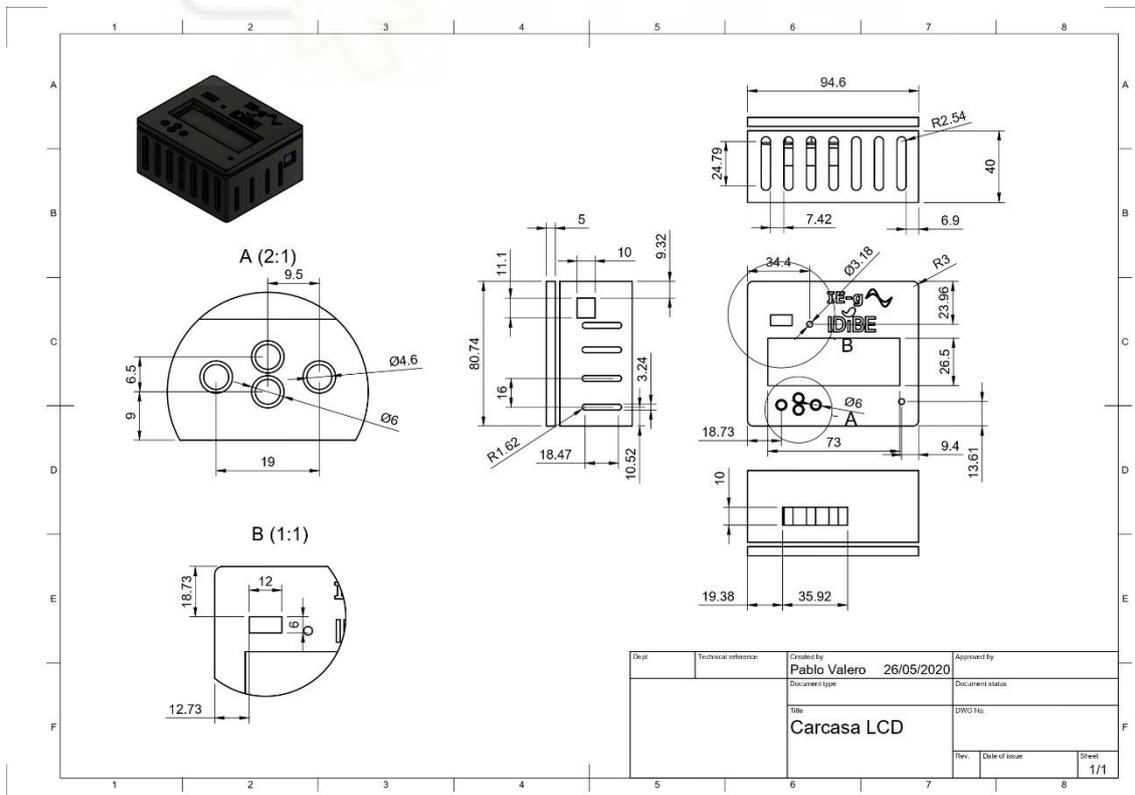


10. Anexos

10.2.3. PROTOTIPO FINAL

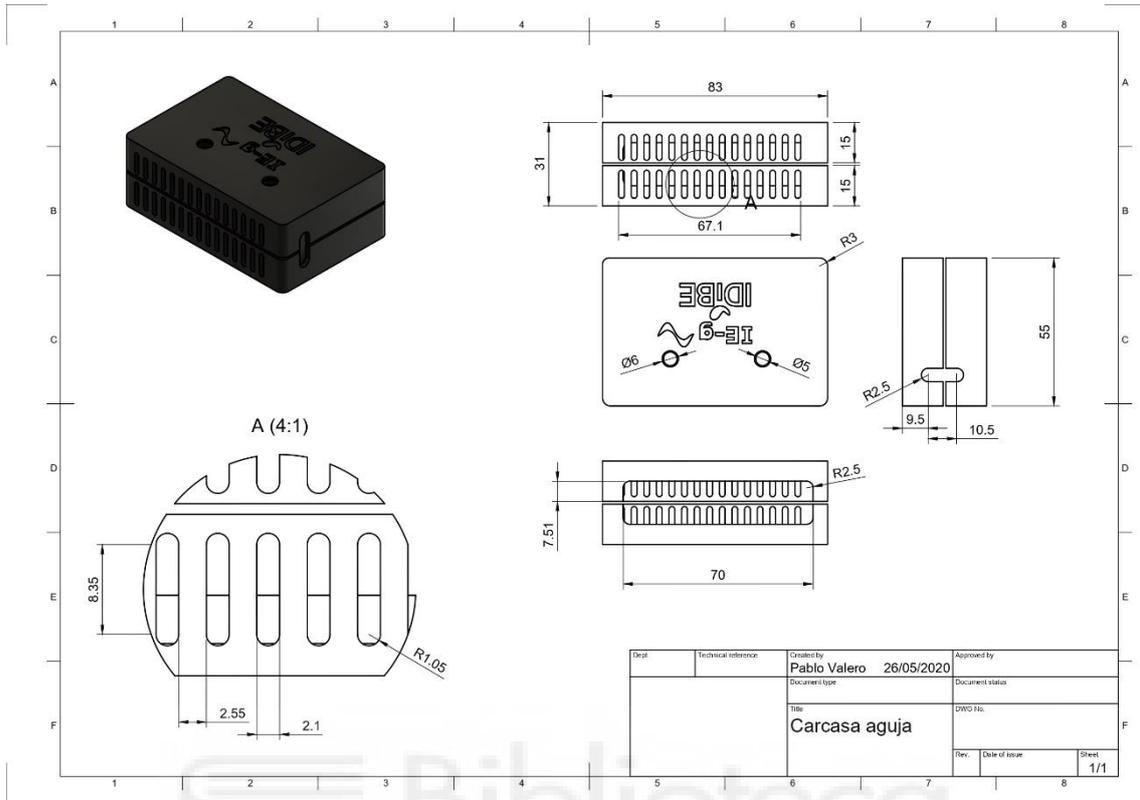


10.2.4. CARCASA LCD



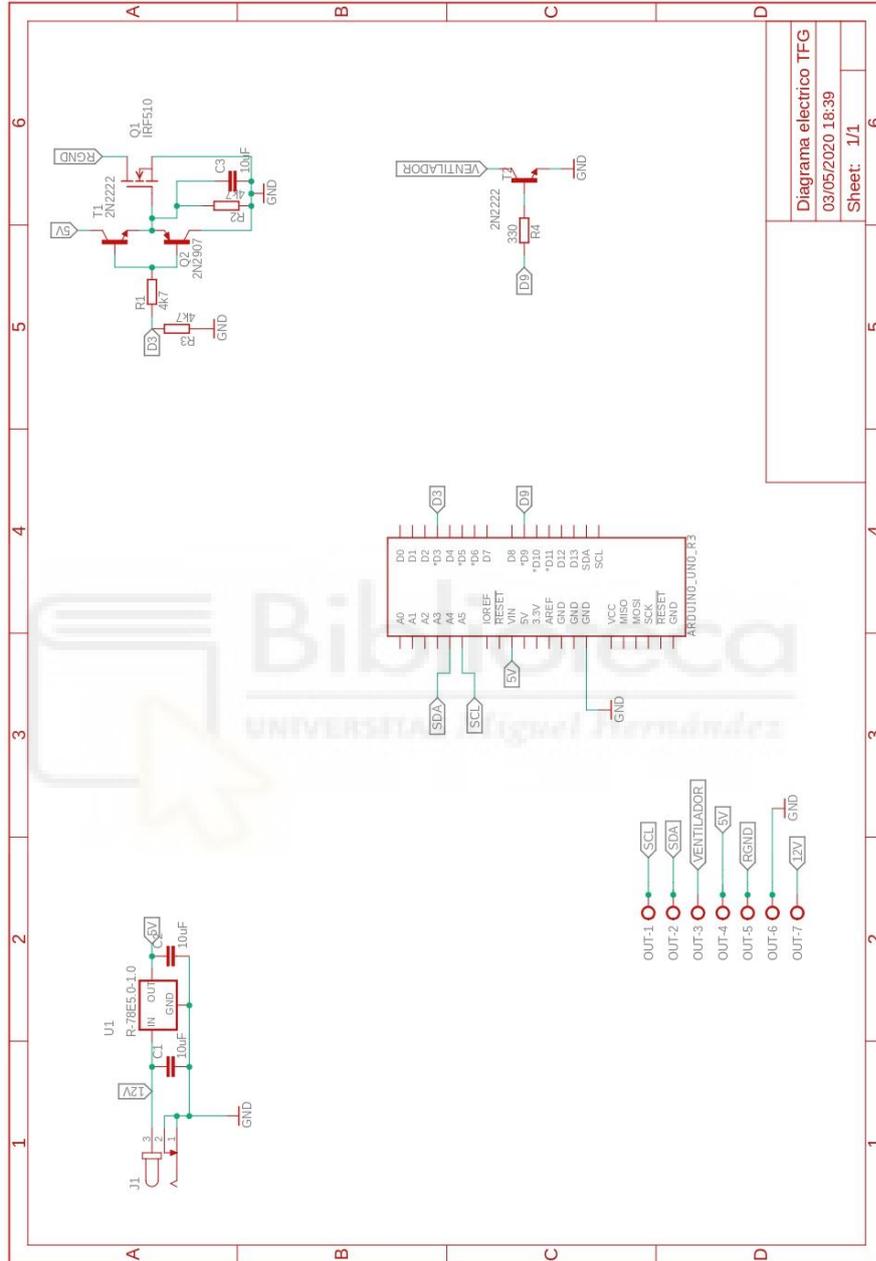
10. Anexos

10.2.5. CARCASA AGUJA



10. Anexos

10.3. ESQUEMA ELÉCTRICO



10. Anexos

10.4 TABLAS DE PROPIEDADES DE LOS MATERIALES UTILIZADOS PARA LA SIMULACIÓN TÉRMICA

10.4.1 ALUMINIO

Tabla 25. Propiedades del aluminio utilizadas para la simulación térmica

Densidad	2.7E-06 kg / mm ³
Conductividad térmica	0,23 W / (mm C ^o)
Coefficiente de expansión térmica	2.36E-05 / C
Calor específico	897 J / (Kg C ^o)

10.4.2 COBRE

Tabla 26. Propiedades del cobre utilizadas para la simulación térmica

Densidad	8.94E-06 kg / mm ³
Conductividad térmica	0,401 W / (mm C ^o)
Coefficiente de expansión térmica	1.67E-05 / C
Calor específico	450 J / (Kg C ^o)

10.4.3 RESINA EPOXI

Tabla 27. Propiedades de la resina epoxi utilizadas para la simulación térmica

Densidad	1.14E-06 kg / mm ³
Conductividad térmica	4.68E-04 W / (mm C ^o)
Coefficiente de expansión térmica	3.98E-05 / C
Calor específico	1000 J / (Kg C ^o)

10. Anexos

10.4.4 FR4

Tabla 28. Propiedades del FR4 utilizadas para la simulación térmica

Densidad	1.85E-06 kg / mm ³
Conductividad térmica	3.5E-04 W / (mm C ⁰)
Coefficiente de expansión térmica	1E-07 / C ⁰
Calor específico	950 J / (Kg C ⁰)



10.5 MANUAL DE USUARIO

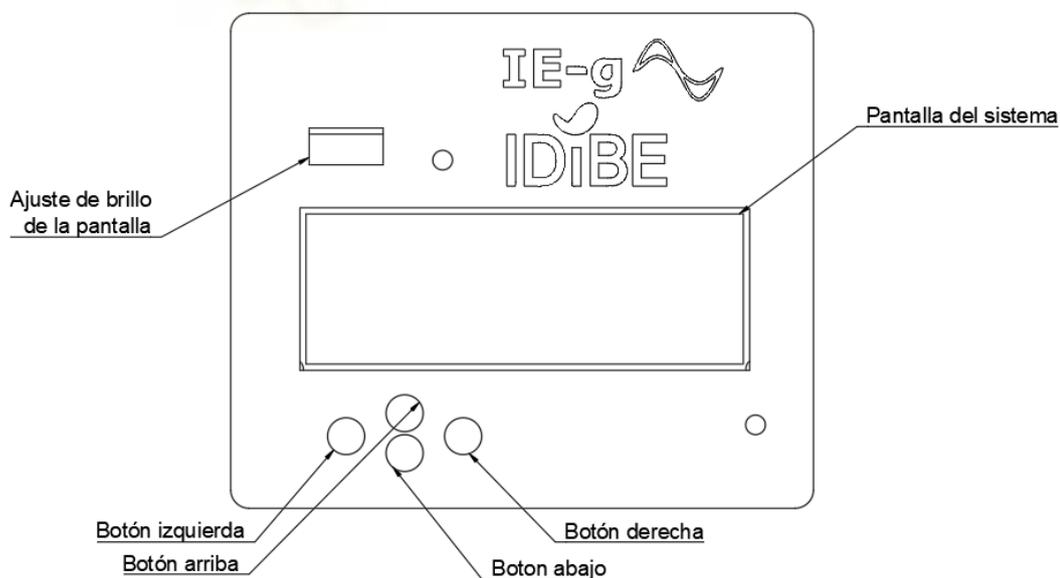
μPerfusion Heater

Manual de usuario

Fecha	30/06/2020
Versión	1.0
Autor	Pablo Valero Martínez

Características técnicas	
Dimensiones	83mm x 55mm x 31mm
Rango de trabajo	Ambiente-70°C
Precisión	±1°C
Resistencia	5Ω
Tiempo de estabilización	5 minutos

Interfaz del sistema



Uso básico

Introducir temperatura

1. Desde la pantalla principal presiona la tecla izquierda para acceder al menú.
2. Selecciona la opción introducir temp
3. Utiliza las teclas arriba y abajo para seleccionar la temperatura de trabajo
4. Confirma la selección con la tecla derecha

Calibrar el sistema

1. Desde la pantalla principal presiona la tecla izquierda para acceder al menú.
2. Selecciona la opción calibrar sistema
3. El sistema comenzara a calibrarse debes esperar a que el proceso termine
4. Una vez calibrado puedes consultar los parámetros del controlador en el menú Info.

Nota: Es recomendable calibrar el sistema siempre que cambien las condiciones de trabajo. La temperatura de calibración es 70°C por defecto. Si deseas cambiarla debes modificar el código adjunto con el dispositivo.

