

AGOSTO 2019

Proyecto Final Integrador

Diseño e implementación de un controlador de temperatura para fermentadores de cerveza



Carrera: Ingeniería Electrónica

Proyecto Final Integrador

Diseño e implementación de un controlador de temperatura para fermentadores de cerveza.

Autores: *Belloli, Enzo y Agostini Ferrero, Inti*

Director: *Pierini, Juan Pablo*

Co-Director: *García, Adrián*

Universidad Nacional de Rio Negro

Sede Andina

Ingeniería Electrónica

San Carlos de Bariloche

Resumen

El presente proyecto está orientado al desarrollo de un dispositivo que permita mejorar y modernizar una etapa del proceso de producción de cerveza artesanal. En este sentido se entrevista a productores locales de cerveza para comprender el modo de funcionamiento y las limitaciones que se les presentan a los cocineros durante diferentes etapas de la producción de cerveza artesanal. Se propuso abordar la problemática desde el punto de vista del control térmico del proceso de fermentación de cerveza.

Se realiza un estudio sobre la fermentación de cerveza y la recolección de datos sistemática en varios reactores de fermentación de modo de construir un modelo matemático que sea fiel a los datos de planta.

Se estudia la técnica de lógica difusa para la implementación de un control del tipo encendido/apagado y se aplica el control al modelo matemático antes mencionado.

Se diseña, fabrica y realiza la puesta en marcha de la electrónica que permita la implementación de lo estudiado, cumpliendo con los requerimientos desprendidos de las conversaciones con diferentes actores de la producción local tanto de cerveza como de equipos de refrigeración.

Por último, se implementa el software de la plataforma embebida y del servidor Web.

A estos puntos se los complementa con los esquemáticos del hardware y un pequeño análisis de mercado.

El proyecto atraviesa las etapas de investigación, diseño, fabricación, puesta en marcha y verificación pensando en la generación de un producto prototipo que permita ser el punto de partida para una producción a mayor escala.

I - Contenido

I -	Contenido.....	6
II -	Introducción.....	16
1.	Objetivos del proyecto.....	16
1.1.	Investigación.....	16
1.2.	Diseño y construcción de una plataforma.....	16
1.3.	Implementación del sistema de control y pruebas de campo 16	
1.4.	Desarrollo e Implementación del firmware final.....	16
2.	Estructura del documento.....	16
III -	Modelo Matemático de la planta a controlar.....	19
1.	Introducción al proceso de fabricación de cerveza.....	19
2.	Desarrollo del Modelo matemático.....	21
2.1.	Objetivo.....	21
2.2.	Modelado de la Población de levadura.....	22
2.3.	Balance de energía del fermentador.....	26
2.4.	Registro de datos en planta.....	29
2.5.	Ajuste del Modelo utilizando los datos recopilados.....	35
3.	Validación del modelo.....	45
3.1.	Análisis de los datos.....	45
3.2.	Ajuste de los parámetros del modelo.....	47
IV -	Estudio de lógica difusa.....	51
1.	Diferencias y similitudes entre la Lógica "Clásica" y la Lógica Difusa 52	
2.	Funciones de membrecía.....	53
3.	Operaciones entre conjuntos difusos.....	54
4.	Variable lingüística.....	55
5.	Control difuso.....	56
5.1.	Método de inferencia de Mandani.....	56
5.2.	Defusificación.....	58
6.	Implementación del controlador.....	59
6.1.	Estructura de la biblioteca.....	59
6.2.	Variables lingüísticas.....	59
6.3.	Reglas.....	63
6.4.	Funciones.....	65
V -	Mediciones y resultados.....	68
1.	Simulaciones.....	68

2.	Errores	75
3.	Procesamiento de los datos.....	77
4.	Ensayo en la fábrica.....	78
VI -	Diseño del sistema	88
1.	Diagrama en bloques	88
2.	Requerimientos de la plataforma y selección de componentes ...	89
2.1.	Subsistema: Adquisición de datos	89
2.2.	Subsistema: Actuadores	91
2.3.	Subsistema: Comunicaciones	92
2.4.	Subsistema: Interfaz con el usuario	93
2.5.	Subsistema: Potencia	94
2.6.	Subsistema: Memoria	96
2.7.	Subsistema: Control.....	96
3.	Diseño del PCB	98
3.1.	Placa A.....	98
3.2.	Placa B	100
4.	Lista de componentes	100
5.	Gabinete.....	103
VII -	Costeo	106
1.	Productos similares en el mercado local	106
2.	Compra de materiales.....	107
3.	Costo final del producto.....	110
VIII -	Software.....	114
1.	Arquitectura del software.....	114
2.	Interfaces de usuario.....	115
2.1.	Servidor Web	115
2.2.	Conexión Bluetooth	117
3.	Estructura de la memoria	118
IX -	Conclusiones	122
X -	Anexo I - Esquemáticos y PCB	126
1.	Placa A.....	126
1.1.	Esquemáticos.....	126
1.2.	PCB	134
2.	Placa B	136
2.1.	Esquemáticos.....	136
2.2.	PCB	137
XI -	Anexo II - Ensamblado y puesta en marcha	140
1.	Objeto	140

2. Alcance.....	140
3. Documentación aplicable.....	140
4. Definiciones	140
5. Procedimiento de ensamblado	141
5.1. Equipamiento necesario	141
5.2. Placa A.....	141
5.3. Placa B	149
6. Procedimiento para la verificación del software	149
6.1. Equipamiento necesario	149
6.2. Pasos a seguir.....	149
XII - Bibliografía	154

Índice de figuras

Figura III-1: Diagrama del proceso de fabricación de la cerveza.	19
Figura III-2: Evolución de la población a lo largo de las fases de la fermentación(16).....	20
Figura III-3: Diagrama de la planta a modelar.....	21
Figura III-4: Gráfica del modelo poblacional logístico.	24
Figura III-5: Gráfica del modelo poblacional logístico con muerte por alcohol	25
Figura III-6: Gráfica del modelo poblacional logístico con muerte por alcohol.	26
Figura III-7: Diagrama resumido del modelo.	27
Figura III-8: Gráfica de la temperatura generada por el modelo.	28
Figura III-9: Esquema del datalogger utilizado	30
Figura III-10: Resistencia vs Temperatura NTC 10Kohm.....	30
Figura III-11: Resistencia vs Temperatura NTC 10k Ω	33
Figura III-12: Datos sin procesar adquiridos en el primer muestreo.....	34
Figura III-13: Gráfica los datos procesados, evolución de la temperatura en un reactor de fermentación.	35
Figura III-14: Identificación de las características de la curva de temperatura registrada.....	37
Figura III-15: Primer ajuste del modelo comparado con los datos reales.....	37
Figura III-16: Curva del Índice de reproducción de la levadura Vs Temperatura.....	38
Figura III-17: Segundo Ajuste del Modelo, incluyendo curva de crecimiento y temperatura ambiente variable	39
Figura III-18: Tipos de enfriadores	40
Figura III-19: Diagrama de temperaturas en el sistema Chiller-Fermentador.....	40
Figura III-20: Diagrama en bloques del Modelo del Sistema de enfriado.	42
Figura III-21: interconexión Fermentador/Sistema de refrigeración	42
Figura III-22: Implementación del modelo del fermentador.	43
Figura III-24: Ajuste final del modelo completo de la planta.	44
Figura III-23: Detalle de la etapa de fermentación del ajuste final.	44
Figura III-25: Datos segundo muestreo	45
Figura III-26: Datos segundo muestreo detallado	46
Figura III-27: Comparación entre el modelo y los datos	47
Figura III-28: Detalle de la etapa de fermentación	48
Figura III-29: Comparación entre los dos ajustes del modelo	49
Figura IV-1- Partes de la función de membrecía	53
Figura IV-2: Diagrama de control realimentado.....	56
Figura IV-3: Procesos del controlador.....	56
Figura IV-4: Función Delta	57
Figura IV-5: Ejemplo de Fusificación	58
Figura IV-6: Estructura de la librería.....	59
Figura IV-7: Diagrama de flujo de las entradas.....	62
Figura IV-8: Detalle de las estructuras de datos.....	62
Figura IV-9: Función de membrecía	63
Figura IV-10- Estructura de las reglas de control.....	64
Figura IV-11: Estructuras para las reglas	65
Figura IV-12: Diagrama de flujo del controlador	66
Figura V-1: Estructura del sistema con el controlador en software (Modelo con controlador).....	68
Figura V-2: Estructura del sistema con el controlador en hardware (Modelo con CdT).....	68
Figura V-3: Comparación del modelo con controlador vs modelo con CdT	69
Figura V-4: Parámetros ideales.....	71
Figura V-5: Sim –Producción de calor incrementada,	71
Figura V-6: Sim – Producción de calor disminuida.....	72
Figura V-7: Sim – Temperatura ambiente límite inferior.....	72
Figura V-8: Sim – Temperatura ambiente límite superior.....	73

Figura V-9: Sim - Temperatura ambiente límite inferior con variación	73
Figura V-10: Sim – Temperatura ambiente límite superior con variación	74
Figura V-11: Sim – Temperatura del refrigerante baja.....	74
Figura V-12: Sim – Aislación termina del reactor disminuida	75
Figura V-13 Gráfica de los errores de simulación	75
Figura V-14: Calibración del sensor	76
Figura V-15 : Calibración del filtro IIR	77
Figura V-16: Esquema de interconexión en la fábrica	78
Figura V-18: Filtro IIR en los datos reales	79
Figura V-17: Datos de la prueba de campo	79
Figura V-19: Configuración utilizada en la prueba en planta.....	80
Figura V-20: Ajuste del modelo a la prueba de campo.....	82
Figura V-21: Simulación de control sobre los datos de la prueba de campo.....	83
Figura V-22: Datos adquiridos durante la segunda experiencia de campo.....	84
Figura V-23: Detalle del primer cambio de setpoint y consecuente activación del control.....	85
Figura V-24: Detalle del segundo cambio de setpoint y consecuente activación del control.....	85
Figura VI-1: Diagrama simplificado del controlador de temperatura	88
Figura VI-2: Diagrama de la plataforma.....	89
Figura VI-3: Conexiones del ADC según el tipo de sensor.....	91
Figura VI-4: Pantalla ilustrativa.....	94
Figura VI-5: Render del frente del equipo.....	94
Figura VI-6: Placa A-TOP/BOTTOM.....	98
Figura VI-7: Placa A-TOP, Planos de tierra.....	99
Figura VI-8: Placa B-TOP/BOTTOM.....	100
Figura VI-9: Gabinete.....	103
Figura VI-10: Representación vista explosionada del CdT.....	104
Figura VI-11: Representación del CdT	104
Figura VII-1: Costo final por unidad.....	111
Figura VII-2: Evolución de los costos frente a diferentes cantidades producidas.....	112
Figura VIII-1: Diagrama de flujo del software implementado en el prototipo.....	114
Figura VIII-2: Página web del prototipo.....	117
Figura VIII-3: Interfaz de la aplicación Bluetooth usada para controlar el prototipo.....	118
Figura VIII-4: Estructura implementada en la memoria.....	120

Índice de tablas

Tabla III-1: Caracterización del sensor NTC.....	29
Tabla III-2: Parámetros utilizados.....	29
Tabla III-3: Valores medidos de los componentes utilizados en el datalogger....	32
Tabla III-4: Parámetros de ajuste de la simulación.....	36
Tabla III-5: Variación de los parámetros entre el primer y segundo ajuste.....	48
Tabla IV-1: Funciones de membrecía para el error	60
Tabla IV-2: Funciones de membrecía para dT/dt	60
Tabla IV-3: Funciones de membrecía para el refrigerante.....	61
Tabla IV-4: Reglas SI-ENTONCES.....	64
Tabla IV-5: Funciones de la librería de Fuzzy Logic	65
Tabla V-1: Parámetros de simulación	70
Tabla VI-1: Sensores y características admisibles (9)	90
Tabla VI-2: Consumos típicos	92
Tabla VI-3 : Temperatura máxima alcanzada por el triac (10)	92
Tabla VI-4: Temperatura máxima que alcanzan los reguladores.....	95
Tabla VI-5: Consumos parciales y totales del dispositivo	95
Tabla VI-6: Consumo a la red eléctrica.....	96
Tabla VI-7: Tamaño de la memoria	96
Tabla VI-8: Lista de señales y periféricos	98
Tabla VI-9: Lista de componentes.....	102
Tabla VII-1: Controladores de temperatura comerciales (11)	106
Tabla VII-2: Lista de compra a Digikey	108
Tabla VII-3: Lista de compras en Aliexpress	109
Tabla VII-4: Lista de compra a PCBWay.....	109
Tabla VII-5: Lista de compra a comercios locales	109
Tabla VII-6: Costo de producción	110
Tabla VII-7: Proyección simplificada de producción	111
Tabla VIII-1: Descripción de los estados de la función APP_Tasks().	115

Glosario

Arduino UNO: es una plataforma electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas.

Datalogger: es un dispositivo que almacena información de un sensor (o varios) en el tiempo y permite luego recuperar esa información.

OTG: USB On-The-Go conocido también por el acrónimo USB OTG y como USB host es una extensión de la norma USB 2.0 que permite a los dispositivos USB tener mayor flexibilidad en la gestión de la interconexión.

Debugger: o depurador, sirve para encontrar errores de código o hardware.

Siglas y Acrónimos

CdT Controlador de Temperatura.

FL Fuzzy Logic o Lógica Difusa.

FM Función de Membrecía.

PID Proporcional, integral Derivativo.

IIR Respuesta de impulso infinito o Infinite impulse response por sus siglas en ingles.

MISO Multiple Inputs Simple Output, múltiples entradas una sola salida.

NTC Negative Temperature Coefficient, Coeficiente de temperatura negativo.

II - Introducción.

1. Objetivos del proyecto

Para establecer los objetivos generales del Proyecto Final Integrador se tomó como punto de partida las necesidades de las pequeñas y medianas empresas de cerveza artesanal de la región y las herramientas que aprendimos en el transcurso de la carrera. Se definió entonces diseñar un producto que satisfaga las necesidades de los productores locales de cerveza de un control más automático, configurable y fiable, brindando al maestro cervecero la posibilidad de tener un mayor control del proceso.

Se plantean cuatro metas;

1.1. Investigación

Realizar una investigación para entender, dentro del proceso de fabricación de cerveza artesanal, la etapa de fermentación y cuáles son los agentes que determinan su evolución. Además recopilar información acerca del comportamiento termodinámico dentro y fuera del fermentador, con el fin de utilizar esa información para generar un modelo matemático que permita mejorar el diseño del controlador.

Asimismo se busca implementar un sistema de control que sea destacable entre los que se encuentran en el mercado y proponemos implementar una librería en lenguaje de programación C, para un control difuso.

1.2. Diseño y construcción de una plataforma

Diseñar y construir una plataforma sobre la cual implementar y verificar el sistema de control. Dicha plataforma está delimitada por una serie de requerimientos (capítulo VI) entre los que se destaca la conectividad a otros dispositivos (computadoras, celulares).

1.3. Implementación del sistema de control y pruebas de campo

Ensayar el sistema de control tanto en laboratorio como en una planta utilizando la plataforma construida. Se busca verificar tanto el modelo matemático como el sistema de control.

1.4. Desarrollo e Implementación del firmware final

Diseñar e implementar un firmware para la plataforma que implemente todas las funcionalidades deseadas para el producto final.

Diseñar e implementar un software compatible con el sistema operativo Windows versión "seven" y posteriores que sirva de interfaz con la plataforma.

Diseñar e implementar un software compatible con el sistema operativo Android versión siete y posteriores que sirva de interfaz con la plataforma.

2. Estructura del documento

El documento está constituido por nueve capítulos y dos anexos. En ellos se desarrollan todos los aspectos que involucran desde el estudio y análisis del problema hasta la implementación y puesta en marcha. La estructura no sigue necesariamente el desarrollo temporal en el que se llevaron a cabo las actividades.

Se describe brevemente a continuación el contenido de cada capítulo.

Capítulo 3 – Identificación de la planta

Se estudia brevemente el proceso de fabricación de cerveza haciendo foco en la etapa de interés, la fermentación. Se estudia el desarrollo del proceso de fermentación y se diseña un modelo matemático termodinámico del mismo. Además se realizan mediciones en una planta real para recolectar datos que permitan validar el modelo. Para finalizar se contrasta el modelo matemático con los datos medidos y se ajustan los parámetros necesarios para que el modelo responda correctamente.

Capítulo 4 – Estudio de la lógica difusa

En este capítulo se estudia el método de control elegido. Se realiza una pequeña introducción a los fundamentos de este método de control, la lógica difusa, con el objetivo de entender y aplicar un control difuso. Por último se analiza el problema en cuestión y se implementa la librería en lenguaje de programación C.

Capítulo 5 – Mediciones y resultados

Se realizan simulaciones para comprobar el correcto funcionamiento del algoritmo de control, utilizando el modelo desarrollado en el capítulo 3, para luego llevar a cabo la experiencia en la planta real.

Capítulo 6 – Diseño del sistema

Se detalla el proceso de diseño del sistema, iniciando por diagramas en bloques, detalle de requerimientos y selección de componentes críticos. Además se muestran algunos criterios del diseño del PCB, la lista de componentes completa y el gabinete.

Capítulo 7 – Costeo

Se realiza un estudio de mercado en la búsqueda de características que hagan de nuestro producto algo novedoso. Se detalla la compra de materiales y se hace una proyección del costo final del producto. Por último se analiza el precio y las posibles ventajas.

Capítulo 8 – Software

Contiene una descripción general del código implementado sobre el microcontrolador para las pruebas de campo realizadas en el capítulo 5.

Capítulo 9 – Conclusiones

Para finalizar se analiza el desarrollo del proyecto final integrador en su conjunto, desde la proyección de tiempos que se realizó al inicio del trabajo, el alcance que se esperaba, las pruebas realizadas y una proyección de cómo se debería continuar.

Los dos capítulos siguientes son los anexos I y II, los cuales detallan el proceso de diseño y puesta en marcha utilizado para verificar el correcto funcionamiento del prototipo.

Capítulo 10- Anexo I – Esquemáticos y PCB

Capítulo 11- Anexo II – Ensamblado y puesta en marcha

Capítulo 12- Bibliografía

III - Modelo Matemático de la planta a controlar

1. Introducción al proceso de fabricación de cerveza

1.1.1 Etapas del proceso

El proceso de fabricación de la cerveza consiste en seis etapas, las cuales se pueden observar en la siguiente figura: (1)



Figura III-1: Diagrama del proceso de fabricación de la cerveza.

Malteado: Los granos de cereal se sumergen en agua para que comiencen a germinar, luego se secan con aire caliente para interrumpir la germinación. Dependiendo del grado de tostado se le dará a la futura cerveza distintos componentes de sabor, color y aroma.

Molienda y Maceración: el cereal previamente malteado se muele y mezcla con agua, con el propósito de extraer el azúcar del grano, el producto final de esta etapa es un líquido claro con alto contenido de azúcares conocido como "Mosto".

Cocción: El mosto se pone a hervir con el objetivo de eliminar las bacterias que hayan podido aparecer durante el proceso, y es justo en este momento cuando se añade el lúpulo, ingrediente que aportará el aroma y amargor deseado.

Fermentación: Luego de la cocción del Mosto, este se enfría a una temperatura adecuada para la levadura a utilizar y se traspa a al recipiente en el cual se realizará la fermentación, luego se agrega la levadura previamente activada comenzando propiamente la fermentación, la cual dependiendo del tipo de levadura durará entre 4 y 7 días.

Maduración: El líquido resultante se mantiene un tiempo en un tanque de maduración, donde reposa en frío para que el sabor y los aromas logrados durante el proceso se estabilicen, se clarifique la cerveza y se carbonate.

Embotellado: Terminada la maduración, se procede a filtrar la cerveza y embotellarla, para evitar que la misma entre en contacto con el aire y cualquier otro contaminante manteniendo así sus características.

Durante la etapa de fermentación va a ser el único momento del proceso en el que nuestro dispositivo va a actuar, controlando la temperatura del mosto para mantener la misma dentro del rango óptimo para la fermentación de la cerveza. En la próxima sección se explicará con más detalle esta etapa con el objetivo de desarrollar luego un modelo matemático de la misma.

1.1.2 Etapa de Fermentación

La etapa de fermentación inicia una vez que la levadura se agrega al "Mosto", y teniendo en cuenta el comportamiento de la levadura se puede dividir en tres fases:

Fase de Latencia o "Lag Phase": Esta es la primera etapa, empieza cuando la levadura entra en contacto con el Mosto y finaliza aproximadamente unas 15 horas después. Durante este periodo las células de levadura reaccionan químicamente con el oxígeno, los minerales y aminoácidos del mosto para producir proteínas y enzimas esenciales para su reproducción.

Fase de Crecimiento Exponencial: Luego de que la levadura se ha adaptado al nuevo medio, comienza a reproducirse de manera exponencial, consumiendo los azúcares presentes en el Mosto, produciendo CO₂, alcohol y liberando calor al medio. Durante esta fase se consumirán la mayor parte de los azúcares, comenzando primero por los azúcares más simples como la glucosa, fructosa y sacarosa, y luego siguiendo por los más complejos como la maltotriosa.

En esta fase es crucial el control de temperatura del fermentador, ya que es cuando se genera la mayor parte del sabor del producto final, el cual depende de la temperatura a la cual se realiza la fermentación.

Fase Estacionaria: Al finalizar la fase anterior, la mayor parte de los azúcares del Mosto ya fueron consumidos, lo que desacelera la velocidad de reproducción de la levadura y además el nivel de alcohol presente en el medio ya es significativo lo que genera un aumento en la tasa de mortalidad de las células de levadura. Estos dos factores confluyen para que la población se estanque y con el tiempo comience a decrecer. En esta etapa la temperatura del fermentador se comienza a reducir gradualmente para propiciar la decantación de la levadura y la clarificación de la cerveza, entrando a la próxima etapa de la producción de cerveza, la Maduración.

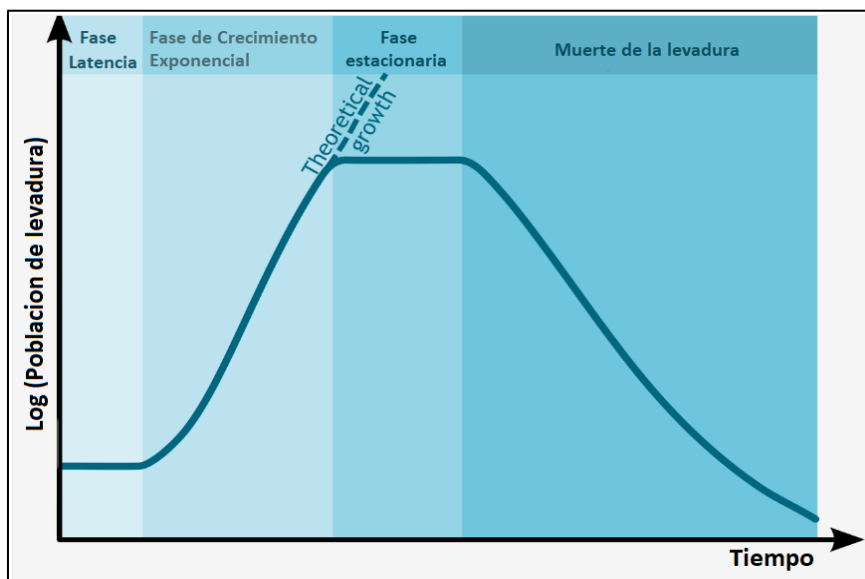


Figura III-2: Evolución de la población a lo largo de las fases de la fermentación(16)

2. Desarrollo del Modelo matemático

2.1. Objetivo

El objetivo fundamental de esta sección es entender las dinámicas del proceso a controlar y poder así estudiar qué estrategia de control usar, y por otro lado independizarnos de la necesidad de probar nuestro futuro algoritmo de control sobre un fermentador real, por dos razones, la primera es que el proceso de fermentación dura de días a semanas con lo que no sería práctico en una primera instancia de desarrollo probarlo de esta manera ya que supondría mucho tiempo y la segunda razón es que sería arriesgado para el productor de cerveza ya que si por alguna razón el controlador funcionaría mal se podría arruinar todo el lote de cerveza.

En la Figura III-3 se puede observar un diagrama de la instalación típica de una fábrica de cerveza, el mismo está compuesto por el fermentador y un sistema de refrigeración, activado por medio de una electroválvula controlada por la acción de un controlador de temperatura.

Durante el proceso de fermentación el mosto genera calor elevando la temperatura del interior del fermentador, si bien parte de este calor se pierde por conducción hacia el ambiente, cuando la temperatura del ambiente es elevada o el volumen del fermentador es suficientemente grande es necesario retirar calor de una manera forzada, para mantener la temperatura en un rango que asegure una fermentación óptima. Con este objetivo se utiliza el sistema de refrigeración el cual activado por el controlador de temperatura, disminuye la temperatura del fermentador cuando es necesario.

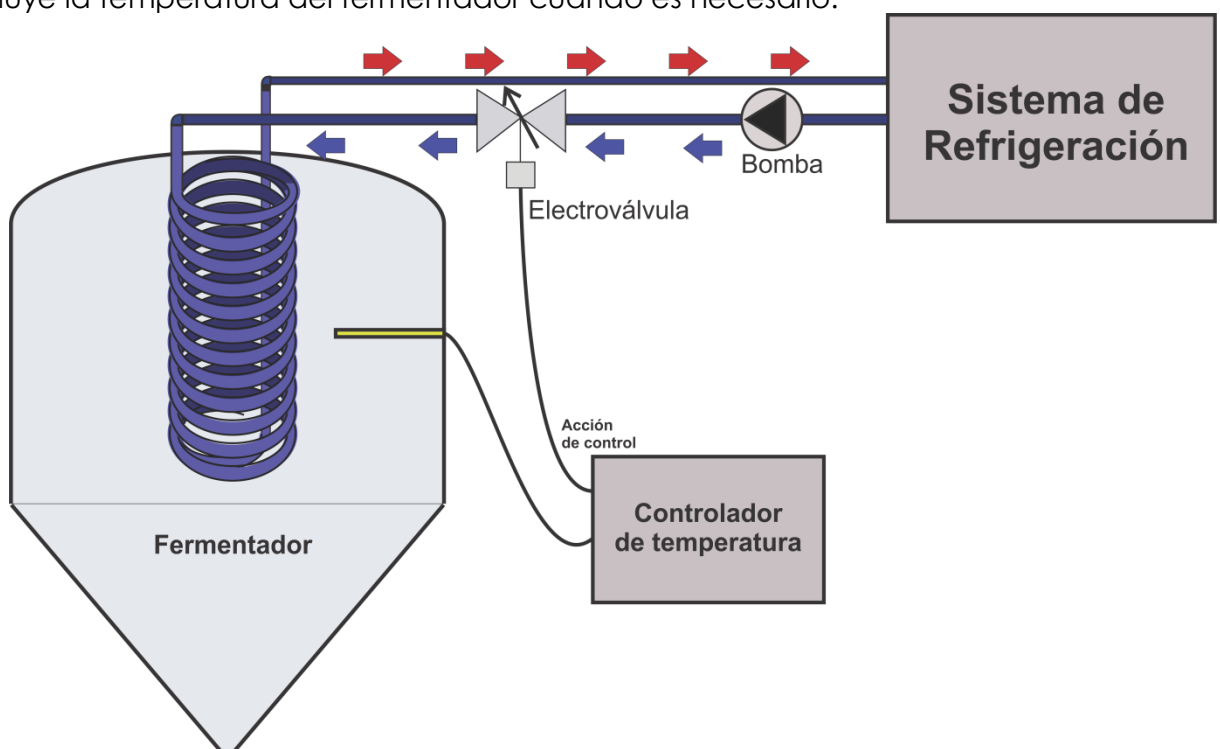


Figura III-3: Diagrama de la planta a modelar

El desarrollo del modelo consistirá en cuatro etapas, en la primera se diseñará un modelo que determine la evolución de la población de levadura, esto es necesario ya que el calor generado en el fermentador es proporcional a la población activa de levadura, una vez obtenido el modelo poblacional se utilizará en la segunda etapa para calcular la evolución de la temperatura del fermentador.

En la tercera etapa se presentarán datos reales de un fermentador de cerveza obtenidos en una cervecería local, se analizarán los mismos, y finalmente en la cuarta etapa se ajustará el modelo a los datos antes mencionados agregando además al modelo un sistema de enfriado, logrando así obtener un modelo completo de la planta a controlar.

2.2. Modelado de la Población de levadura

Si bien durante el proceso de fermentación ocurren muchas reacciones, y se producen y consumen distintos compuestos químicos, para los fines prácticos del modelo, considerando que solo interesa modelar la temperatura del proceso, se puede realizar la siguiente simplificación.



La reproducción de la levadura ocurre mediante el proceso químico que "consume" los azúcares liberando energía calórica y produciendo como subproductos, CO₂ y alcoholes. Estos alcoholes generados son un componente "tóxico" para las células de la levadura, por lo que a medida que aumenta la concentración de los mismos disminuye la tasa neta de reproducción de las levaduras.

2.2.1 Modelo exponencial o "Malthusian growth model"

El modelo exponencial, creado por Thomas Malthus (1766-1834), supone que la velocidad de crecimiento de una población es proporcional a la población en ese instante de tiempo, o lo que es lo mismo:

$$\frac{dN(t)}{dt} = k N(t)$$

Siendo:

$N(t)$: Población en el instante t

$\frac{dN(t)}{dt}$: Crecimiento instantáneo de la población

k : Constante que determina la velocidad con que crece

La solución de esta ecuación diferencial es la siguiente:

$$N(t) = N_0 e^{kt}$$

Siendo: N_0 la población inicial.

Si bien es bastante lógico lo que propone el modelo, como se puede ver en la solución, la población crece exponencialmente a medida que pasa el tiempo, esto se debe a que el modelo no contempla ningún limitante del ambiente como puede ser por ejemplo la cantidad de comida disponible.

2.2.2 Modelo Logístico

Este modelo fue desarrollado por Pierre-François Verhulst (1804–1849), y propone que el crecimiento de la población es exponencial en cuanto la población sea chica con respecto a la capacidad máxima del ambiente, pero a medida que la población comienza a aumentar el crecimiento comienza a disminuir, llegando asintóticamente a la capacidad máxima.

En nuestro caso el ambiente de la levadura sería el fermentador, y la capacidad máxima estaría determinada por la cantidad de azúcar inicial en el Mosto, las ecuaciones que modelan esto son las siguientes.

$$\frac{dN(t)}{dt} = \gamma N(t) S(t)$$

$$\frac{dS(t)}{dt} = -\eta \gamma N(t) S(t)$$

Siendo:

$S(t)$: Cantidad de azúcar en el instante t

$\frac{dS(t)}{dt}$: Variación instantánea del azúcar

γ : Tasa de crecimiento por tiempo y concentración de azúcar

η : Cantidad de azúcar consumida para generar un nuevo individuo

Observando las ecuaciones podemos ver que ahora tenemos un sistema de ecuaciones diferenciales no lineal ya que en ambas ecuaciones hay un producto entre la población de levaduras y la cantidad de azúcar $S(t)$. En este nuevo sistema la velocidad de crecimiento de la población es proporcional a la población en ese instante y a la cantidad de azúcar disponible, esto se debe a la suposición de que para generar un nuevo individuo la célula de levadura requiere azúcar, por lo que en un plano general cuanto mayor sea la cantidad de azúcar será mayor la probabilidad de que esa célula pueda reproducirse.

La segunda ecuación del sistema determina la dinámica de la cantidad de azúcar presente en el medio, y establece que esta decrece proporcionalmente al factor η , el cual es constante y lo podemos pensar como la cantidad de azúcar que se consume para generar un nuevo individuo, y el resto de los términos los cuales determinan la velocidad de reproducción de la levadura.

Fijando S_0 e N_0 (cantidad inicial de azúcar y población inicial) como las condiciones iniciales, la solución analítica del sistema es la siguiente:

$$N(t) = \frac{k}{1 + \frac{(k - N_0)}{N_0} e^{-rt}}$$

Dónde:

$r = \gamma (S_0 + \eta N_0)$ Velocidad de crecimiento

$k = \frac{S_0}{\eta} + N_0$ Capacidad del ambiente

El desarrollo de la solución puede observarse en "MATHEMATICAL MODELS IN POPULATION DYNAMICS- Salisbury, Alexander (2011) - p.52" (2).

A continuación se puede observar gráficamente la solución, como propone el modelo la población de levadura crece exponencialmente en los primeros instantes, cuando el azúcar disponible es prácticamente ilimitado, pero a medida que la población sigue creciendo y el azúcar es consumido la velocidad de reproducción decrece y la población llega a la capacidad límite del

ambiente, lo que en la sección de Fermentación denominamos la fase estacionaria.

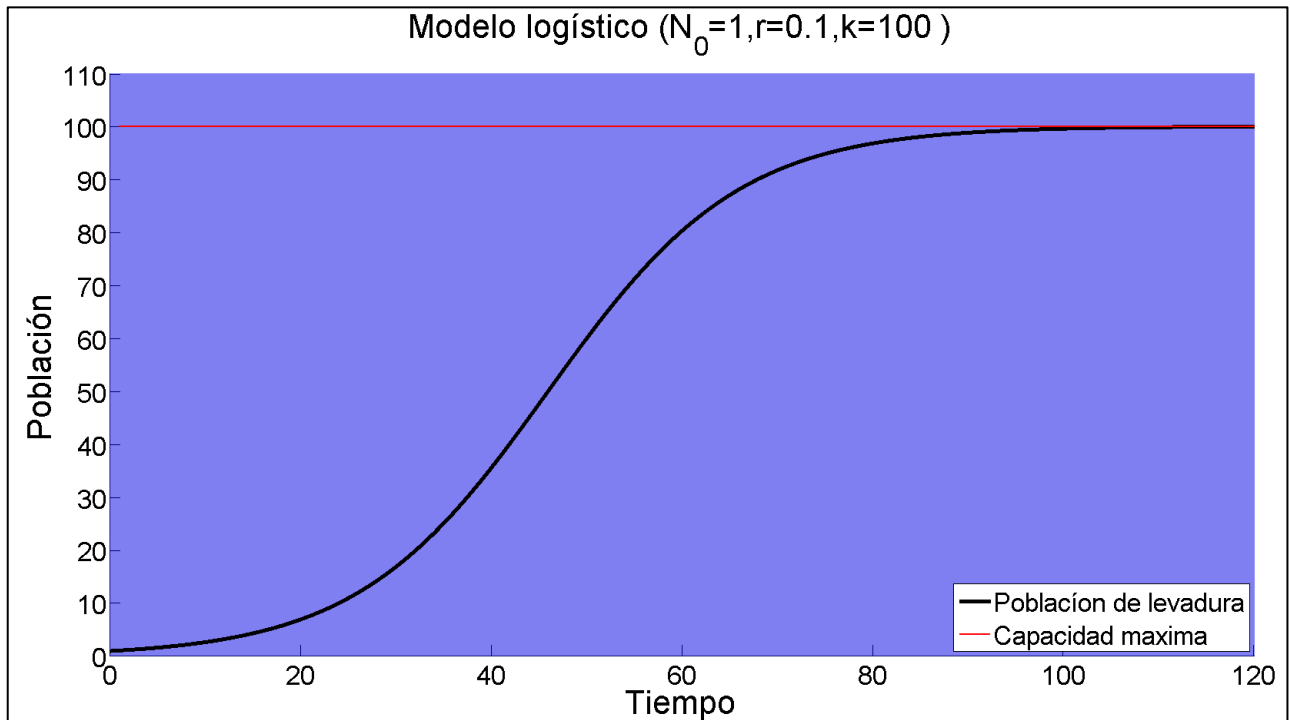


Figura III-4: Gráfica del modelo poblacional logístico.

2.2.3 Modelo Logístico con Muerte por alcohol

Si bien el modelo anterior es correcto en la primera etapa del proceso, una vez alcanzada la fase estacionaria la población con el paso del tiempo comienza a disminuir, debido entre otras cosas a la presencia de alcohol en el medio el cual es tóxico para las células de levadura. A continuación se muestra el sistema de ecuaciones diferenciales teniendo en cuenta la muerte de la levadura por la acción del alcohol, el mismo fue obtenido a partir de "Students Rise to the Challenge of Modeling Yeast Growth Despite Sour Hiccups from Imperfect Data- Caldwell, Alicia (2013)- p.19" (3).

$$\frac{dN(t)}{dt} = \gamma N(t) S(t) - \beta A(t) N(t) \quad (1)$$

$$\frac{dS(t)}{dt} = -\eta \gamma N(t) S(t) \quad (2)$$

$$\frac{dA(t)}{dt} = \alpha S(t) N(t) \quad (3)$$

En la ecuación (1) se agregó el término negativo $\beta A(t) N(t)$ el cual representa el decremento de población generado por muertes por alcohol, el mismo es proporcional a β que puede interpretarse como la tasa de mortalidad por alcohol, $A(t)$ que es la cantidad de alcohol en el medio en el instante t , y también es proporcional a la población actual de levadura $N(t)$ ya que cuanto mayor sea más probable es que un individuo muera.

La ecuación (2) se mantiene sin cambios, y se agrega la ecuación (3) para incluir la dinámica de crecimiento de la cantidad de alcohol en el medio. La cantidad de alcohol en el medio crece proporcionalmente con la población de levadura $N(t)$, ya que al existir más individuos la producción de alcohol es mayor, y es proporcional también a la cantidad de azúcar $S(t)$, ya

que el alcohol es generado a partir de esta, y por último a la constante α que puede interpretarse como la cantidad de alcohol generada a partir de una cantidad de azúcar.

El sistema de ecuaciones se simuló y dependiendo de las condiciones iniciales y los distintos parámetros $(\beta, \alpha, \eta, \gamma)$ se dan dos casos:

Primer caso:

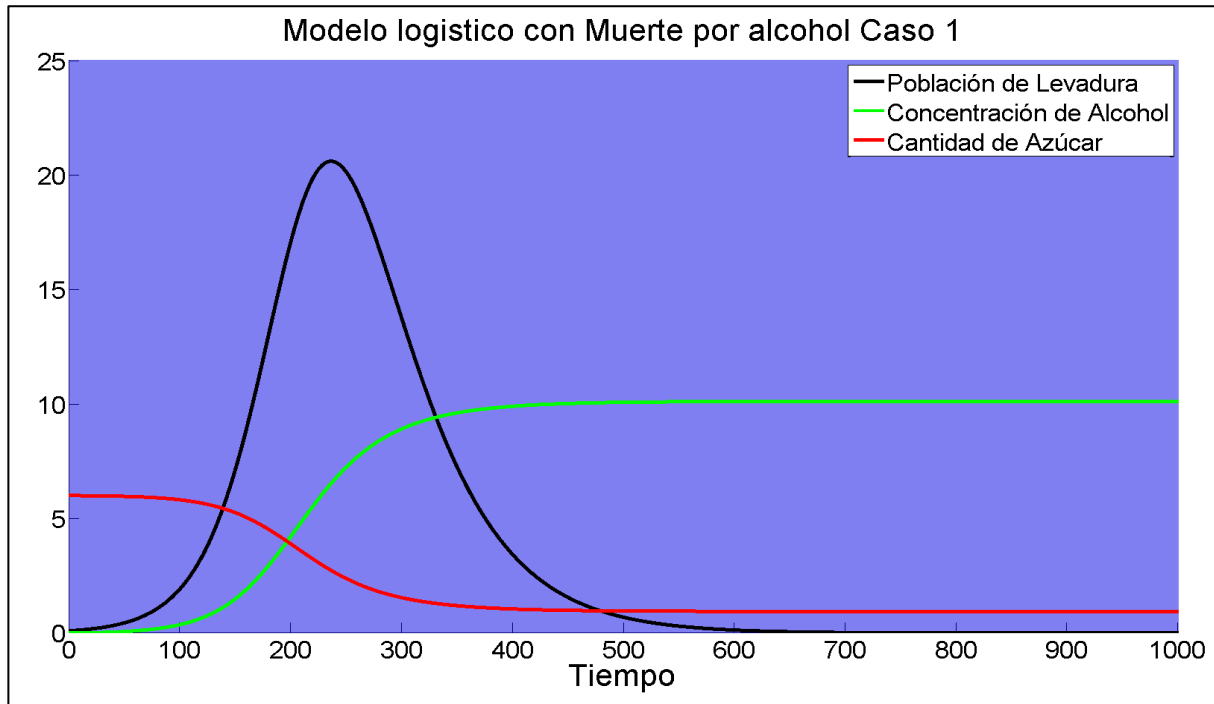


Figura III-5: Gráfica del modelo poblacional logístico con muerte por alcohol

Parámetros usados $\eta = 0.1, \gamma = 0.005, \beta = 0.021, \alpha = 0.001, N_0 = 0.1, S_0 = 6, A_0 = 0$

Como se puede observar, la población de levadura crece hasta el momento en que la cantidad de alcohol presente mata una cantidad de células de levadura suficiente como para inhibir el crecimiento de la población, momento a partir del cual la población comienza a disminuir hasta su desaparición.

En este caso el limitante es el alcohol, ya que como se ve todavía queda azúcar restante en el medio.

Segundo caso:

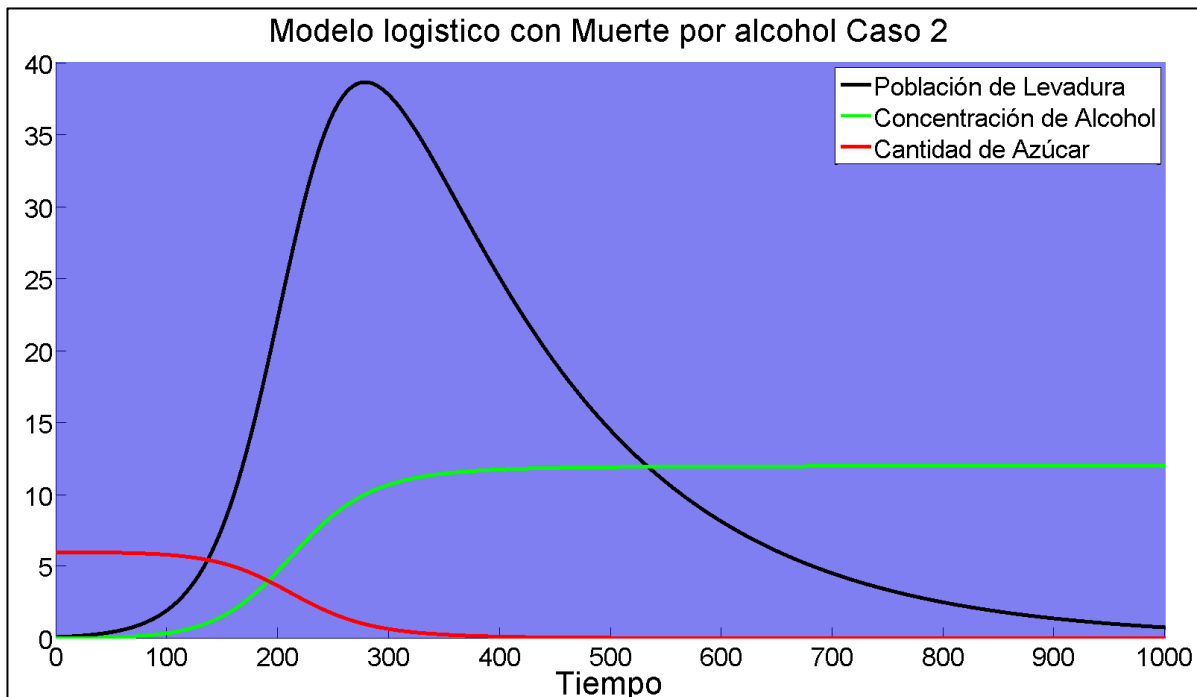


Figura III-6: Gráfica del modelo poblacional logístico con muerte por alcohol.
Parámetros usados $\eta = 0.1, \gamma = 0.005, \beta = 0.005, \alpha = 0.001, N_0 = 0.1, S_0 = 6, A_0 = 0$

En este caso la cantidad de células que mueren por alcohol es mucho menor y esto genera que la población de la levadura crezca logarítmicamente hasta que se agota el azúcar disponible, luego de lo cual cesa la reproducción de la levadura y el alcohol presente genera una lenta disminución de la población de la levadura.

2.3. Balance de energía del fermentador

El proceso de fermentación puede resumirse de la siguiente forma: (Figura III-7)

- La levadura $N(t)$ crece proporcionalmente a $N(t)S(t)$, e inversamente proporcional a $A(t)N(t)$.
- La concentración de azúcar $S(t)$ decrece proporcionalmente al crecimiento de la levadura.
- La concentración de alcohol $A(t)$ crece proporcionalmente al crecimiento de la levadura.
- El mosto genera calor proporcional a $N(t)$ y $S(t)$, elevando la temperatura del fermentador $T_{fermentador}(t)$.
- Hay una transferencia de calor con el ambiente, proporcional a $[T_{fermentador}(t) - T_{ambiente}(t)]$, que impacta en la $T_{fermentador}(t)$.
- Hay una transferencia de calor con el sistema de control $Q_R(t)$, el cual es utilizado para controlar $T_{fermentador}(t)$, esta transferencia será modelada en la sección 2.5.2, por lo que no será tenida en cuenta en esta instancia.

Hasta ahora el modelo desarrollado contempla las dinámicas propias de la levadura, el azúcar y el alcohol, por lo que para obtener un modelo

completo de la planta es necesario incluir las dos últimas interacciones antes resumidas.

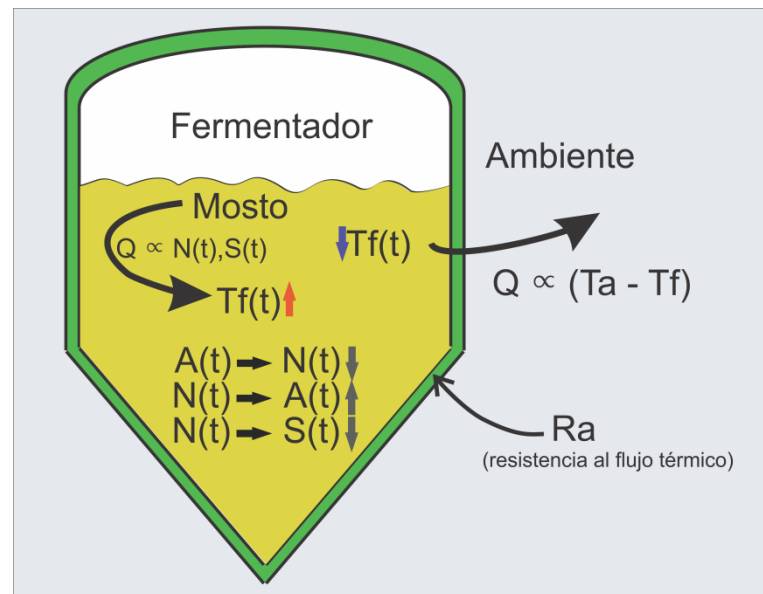


Figura III-7: Diagrama resumido del modelo.

La temperatura del fermentador puede ser escrita como la sumatoria del calor generado por la levadura y la pérdida hacia el ambiente o el refrigerante:

$$\frac{dT_{\text{fermentador}}(t)}{dt} = Q_L(t) - Q_A(t) - Q_R(t) \quad (4)$$

Dónde:

$$\frac{dT_{\text{fermentador}}(t)}{dt} = \text{Variación instantánea de la temperatura del fermentador}$$

$Q_L(t)$ = Calor generado por la levadura

$Q_A(t)$ = Calor ganado/ perdido hacia el ambiente

$Q_R(t)$ = Calor perdido por el refrigerante (en esta sección no será considerado)

En el proceso de la fermentación, como ya se mencionó, se consumen azúcares y se genera CO_2 , alcohol y calor, por lo tanto se puede asumir que el calor generado por la levadura $Q_L(t)$ es proporcional a la cantidad de azúcar consumida.

$$Q_L(t) = k_{t,\text{levadura}} \frac{dS(t)}{dt} \quad (5)$$

Entonces reemplazando la ecuación (2) en (5), se puede escribir al calor generado por la levadura como

$$Q_L(t) = k_{t,\text{levadura}} \eta \gamma N(t) S(t) \quad (6)$$

A su vez la transferencia de calor con el ambiente $Q_A(t)$, en la que solo se tendrá en cuenta la pérdida por conducción, es proporcional a la diferencia entre la temperatura del fermentador y la temperatura ambiente:

$$Q_A(t) = k_{t,\text{ambiente}} [T_{\text{ambiente}}(t) - T_{\text{fermentador}}(t)] \quad (7)$$

Reemplazando (6) y (7) en (4), podemos escribir a la variación de la temperatura en el fermentador como:

$$\frac{dT_{fermentador}(t)}{dt} = k_{t,levadura} \eta \gamma N(t) S(t) - k_{t,ambiente} [T_{fermentador}(t) - T_{ambiente}(t)] - Q_R(t) \quad (8)$$

Las constantes $k_{t,ambiente}$ y $k_{t,levadura}$, dependen del volumen de mosto, la aislación del fermentador, la cantidad de calor liberado por la levadura, el calor específico del mosto, y la conductividad con el ambiente. Y dado que $Q_R(t)$ como ya se mencionó nos será tenido en cuenta, en lo que respecta a la simulación ese término será considerado cero.

En la siguiente gráfica se puede ver la evolución temporal de la temperatura del modelo de fermentador sin refrigeración para los siguientes parámetros y condiciones iniciales.

Parámetros usados: $\gamma = 0.005, \eta = 0.5, \beta = 0.004, \alpha = 0.01, k_{t,levadura} = 3.5$

$k_{t,ambiente} = 0.008, T_{ambiente} = 10^\circ C.$

Condiciones iniciales: $N_0 = 0.1, S_0 = 10, A_0 = 0$

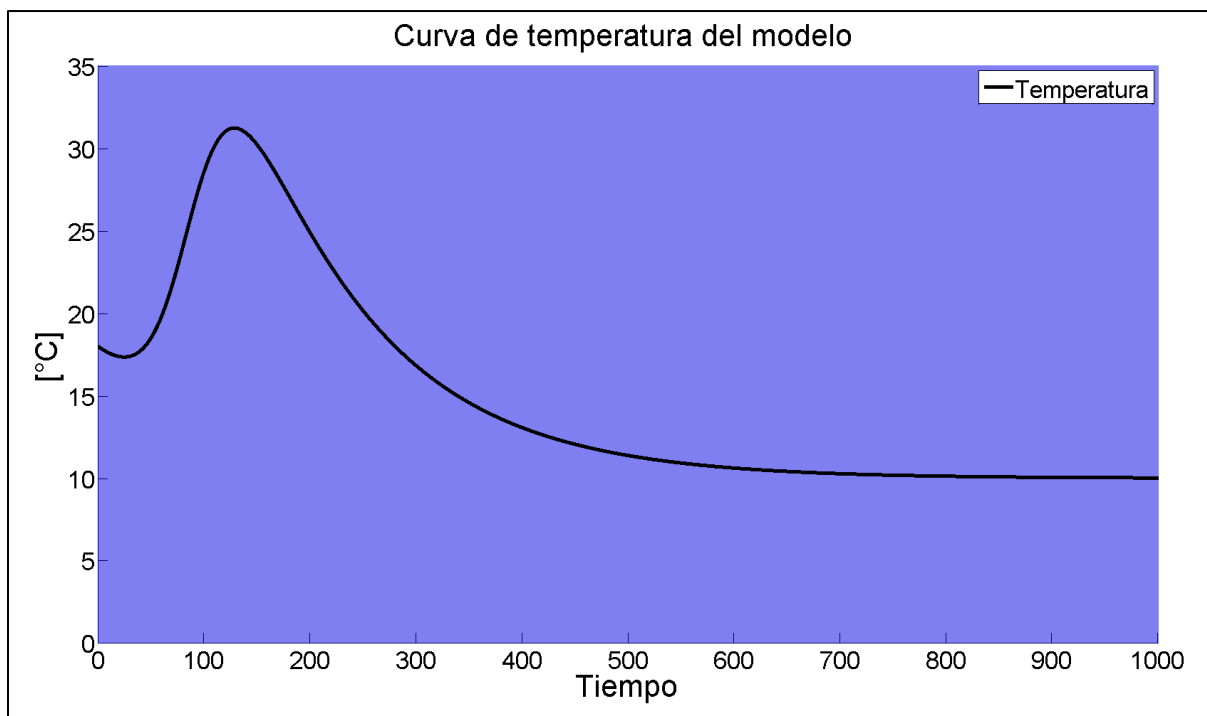


Figura III-8: Gráfica de la temperatura generada por el modelo.

Como se puede observar la temperatura del modelo comienza con una pendiente negativa debido a la pérdida de calor con el ambiente y que la levadura está en la fase LAG, luego hay un crecimiento exponencial debido a la explosión en la población de levadura alcanzando el pico máximo de temperatura, seguido por una pendiente negativa debido a la muerte de la levadura y la pérdida de calor hacia el ambiente, finalizando en un valor cercano a los $10^\circ C$ (temperatura ambiente usada).

2.4. Registro de datos en planta

2.4.1 Caracterización de los sensores

Con el objetivo de validar el modelo matemático se procedió a realizar un registro de la temperatura a lo largo de la etapa de fermentación, para esto se utilizaron sensores NTC los cuales son altamente alinéales, y debido a que sólo conocíamos la resistencia de los mismos a 25 °C, realizamos una calibración utilizando la ecuación de Steinhart–Hart la cual modela la resistencia de un semiconductor a diferentes temperaturas, esta ecuación fue obtenida de "Thermistor Calibration and the Steinhart-Hart Equation, APPLICATION NOTE" (4).

$$\frac{1}{T} = A + B \ln(R) + C [\ln(R)]^3$$

Partiendo de esta ecuación y utilizando tres temperaturas distintas, se puede llegar a un sistema de ecuaciones como el siguiente.

$$\begin{bmatrix} 1 & \ln(R_1) & \ln^3(R_1) \\ 1 & \ln(R_2) & \ln^3(R_2) \\ 1 & \ln(R_3) & \ln^3(R_3) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \frac{1}{T_1} \\ \frac{1}{T_2} \\ \frac{1}{T_3} \end{bmatrix}$$

Por lo que realizando tres mediciones de la resistencia de los sensores (R_1 , R_2 , R_3) en tres temperaturas conocidas (T_1 , T_2 , T_3) se puede obtener los parámetros A, B y C. La medición del sensor arrojó los siguientes valores de Resistencia vs Temperatura:

Temperatura [°C]	Resistencia [Ω]
5	25449
15	15736
25	10006

Tabla III-1: Caracterización del sensor NTC

Los valores de temperatura fueron seleccionados de tal forma que se cubriera el rango esperado de temperaturas a medir, y que de esta forma la curva generada siguiera con el menor error posible a los valores del sensor en este rango.

Utilizando los valores medidos y el sistema de ecuaciones antes descripto, se despejaron los siguientes valores para los tres parámetros necesarios, estos luego se utilizaron en el programa que procesa las muestras tomadas, para traducir los valores de resistencia a temperatura.

Parámetro	Valor
A	0,001138942
B	0,000232762
C	0,000000091

Tabla III-2: Parámetros utilizados

2.4.2 Hardware utilizado

La medición de temperatura se realizó utilizando un Arduino UNO como datalogger, se midió utilizando sensores NTC la temperatura del fermentador y del ambiente y por último se registró el estado de la electroválvula.

Para registrar el estado de la electroválvula se utilizó una fuente de 220VCA a 5VCC, la cual se conectó en paralelo a la salida del controlador que accionaba la electroválvula en la fábrica. A continuación se muestra un diagrama de las conexiones del datalogger.

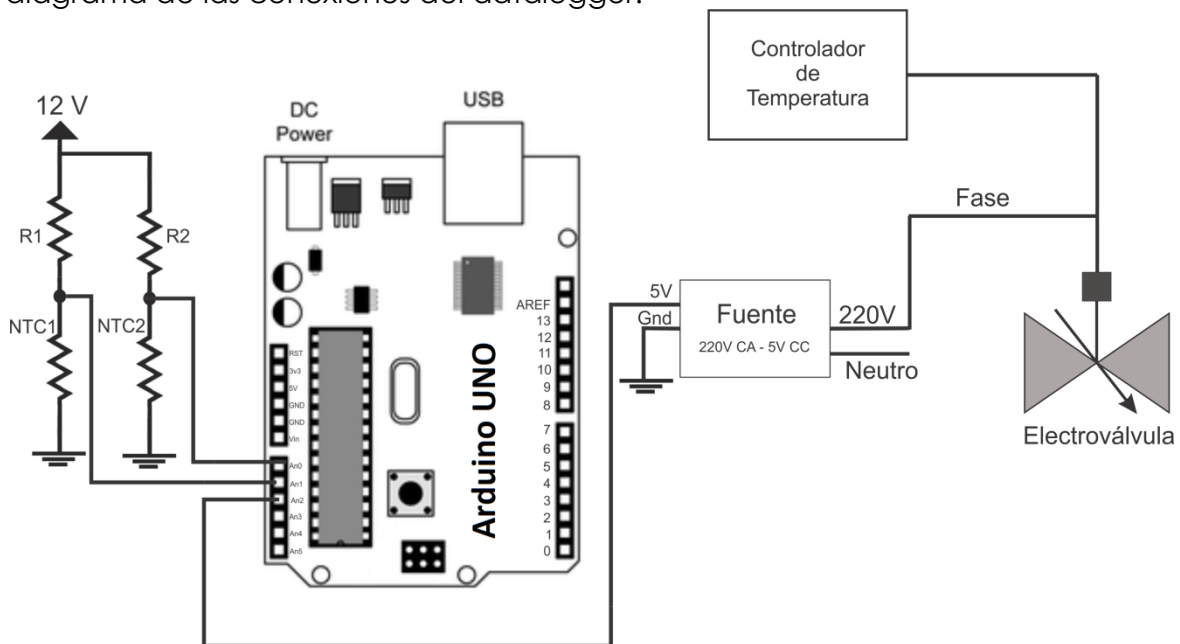


Figura III-9: Esquema del datalogger utilizado

2.4.3 Selección de componentes, resolución y error

Selección de componentes

Con los parámetros despejados en la caracterización del sensor (A, B, C), se generó la curva que puede observarse a continuación.

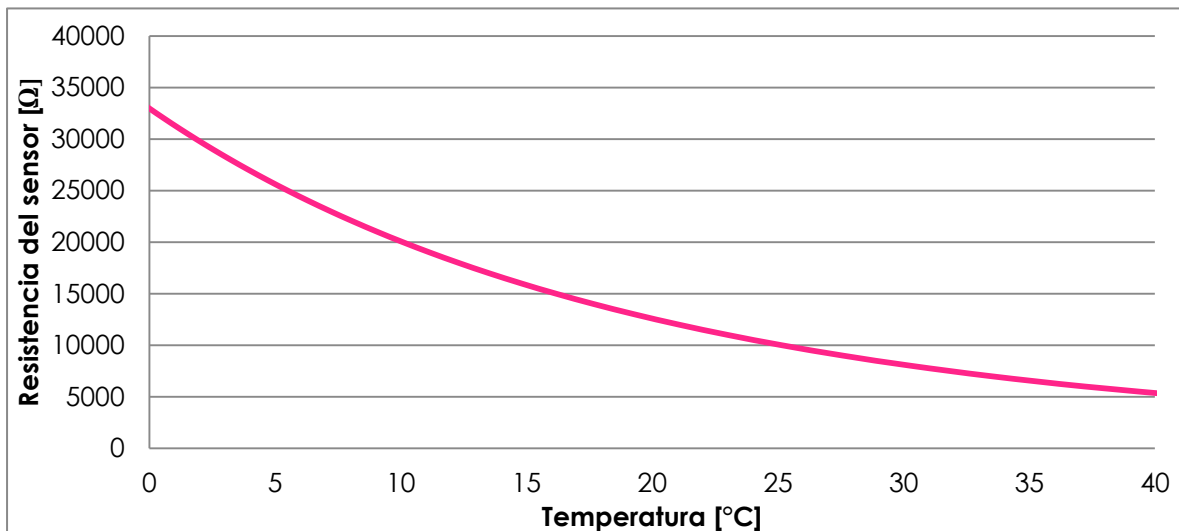


Figura III-10: Resistencia vs Temperatura NTC 10Kohm

Dado que se esperaban medir temperaturas dentro del rango de 4 a 30 °C, el valor máximo de resistencia a medir sería 27kΩ. Por esta razón sumado a que el voltaje máximo que puede tolerar un pin analógico del ARDUINO UNO sin generar daños en el microcontrolador es $V_{CC}+0.5V$ (siendo $V_{CC} = 5V$), se seleccionó un valor de R_1 y R_2 de 39kΩ, este valor genera un voltaje máximo en el pin analógico de 4,9 V.

Cálculo de resolución

El ARDUINO UNO posee un ADC con una resolución de 10 bits, lo que se traduce en una resolución en voltaje de:

$$1 \text{ LSB [V]} = \frac{5 \text{ V}}{2^{10}} = 0,00488 \text{ V}$$

Esto se traduce en una resolución en la medición de la resistencia NTC de:

$$1 \text{ LSB [\Omega]} = \frac{0,00488 \text{ V} \times 39000 \Omega}{12 \text{ V} - 0,00488 \text{ V}} = 15,86 \Omega$$

Ya que la variación de resistencia no es lineal con respecto a la temperatura para poder calcular la resolución en °C es necesario tomar ciertas consideraciones, el análisis se realizara en el entorno de los 21°C, esta temperatura es elegida debido a que será el valor que tomara el Setpoint durante la etapa activa de la fermentación.

Las variaciones de resistencia disminuyen a medida que la temperatura aumenta, aprovechando este fenómeno se acotara la resolución en las cercanías de 21°C, linealizando entre 21°C y 30°C, tomando el segundo valor muy por encima de la temperatura esperada de fermentación con el propósito de calcular la peor resolución esperada.

La resistencia en los puntos elegidos es:

$$R_{21^{\circ}\text{C}} = 12040 \Omega$$

$$R_{30^{\circ}\text{C}} = 8109 \Omega$$

Por lo tanto podemos acotar la variación en torno a los 21°C de la siguiente forma:

$$\frac{d\Omega}{d^{\circ}\text{C}_{21^{\circ}\text{C}}} \leq \frac{(R_{30^{\circ}\text{C}} - R_{21^{\circ}\text{C}})}{30^{\circ}\text{C} - 21^{\circ}\text{C}} = -436,77 \Omega/^{\circ}\text{C}$$

Utilizando la resolución en Ω calculada anteriormente podemos llegar a que la resolución en °C estará acotada de la siguiente forma:

$$1 \text{ LSB [\Omega]} \leq \frac{15,86 \Omega}{436,77 \frac{\Omega}{^{\circ}\text{C}}} = 0,036 \text{ }^{\circ}\text{C}$$

Calculo de error

Los factores de error los podemos dividir en dos grupos:

Factores dinámicos: variación de la fuente de alimentación de 12V, variación del voltaje de referencia interno del Arduino UNO, variación en la resistencias R11/R2 debida a la potencia disipada/temperatura ambiente.

Factores estáticos: variación en las resistencias R1/R2 debido a la tolerancia de fabricación/error de medición de las mismas, error en la medición de la fuente de 12V.

Los factores dinámicos serán despreciados ya que la corriente que circula por el divisor resistivo, la potencia que disipan las resistencias y la carga que se le presenta a la fuente de 12V son mínimas, y las variaciones en el voltaje de referencia de 12V será minimizado en el post procesamiento de las muestras.

En cuanto a los factores estáticos previa a la instalación del datalogger se midió con un multímetro UNI-T UT61D, el valor real de las resistencias y la fuente de 12V con el fin de eliminar el error debido a la tolerancia de fabricación, según el fabricante el multímetro posee una precisión de $\pm (0.5\%+1)$ en

mediciones de voltaje y $\pm (1.0\%+2)$ en mediciones de resistencia, a continuación se realiza la propagación del error en las mediciones.

Componente	Valor medido	Error
R1 39k Ω	38,38 k Ω	$\pm(383,8 + 20)$
R2 39k Ω	38,50 k Ω	$\pm(385 + 20)$
Fuente 12V	12,20 V	$\pm(0,061 + 0,01)$
NTC 10k 21°C	12,04 k Ω	$\pm(120,4 + 20)$

Tabla III-3: Valores medidos de los componentes utilizados en el datalogger

EL análisis de error se realizara teniendo en cuenta R2, ya que presenta el mayor error, y se hará en el entorno de los 21°C región en la que se espera estén los datos de mayor interés.

EL error de cuantización $\Delta_{cuantizacion}$ se puede calcular como:

$$\Delta_{cuantizacion} = \pm \frac{1 \text{ LSB [V]}}{2} = \pm 0,00244 \text{ V}$$

A 21°C el voltaje en el sensor será: 2,9895 V 2,8664 V

$$V_{ntc_{max}} = \frac{(12,20 + 0,071)V \times (12040 + 140,4) \Omega}{(12040 \Omega + 38500 \Omega - 405 \Omega)} = 2,9896 \text{ V}$$

$$V_{ntc_{min}} = \frac{(12,20 - 0,071)V \times (12040 - 140,4) \Omega}{(12040 \Omega + 38500 \Omega + 405 \Omega)} = 2,8254 \text{ V}$$

$$V_{ntc} = 2,9074 \pm 0,0821 \text{ V}$$

Lo que será registrado por el Arduino UNO como:

$$\text{Valor medido} = 595 \pm 17$$

Traducido a voltaje como

$$\text{Valor medido [V]} = \text{Valor medido} \times \frac{5V}{2^{10}} \pm \Delta_{cuantizacion} = 2,9052 \pm 0,0830$$

Traducido a resistencia como

$$\text{Valor medido } [\Omega] = \text{Valor medido [V]} \times \frac{(38,50 \text{ k}\Omega \pm 405\Omega)}{(12 \pm 0,071)V - \text{Valor medido [V]}}$$

$$\text{Valor medido } [\Omega]_{max}$$

$$= (2,9052 + 0,0439)V \times \frac{(38,50 \text{ k}\Omega + 405 \Omega)}{(12,20 - 0,061)V - (2,9052 + 0,0439)V}$$

$$= 12718,8818 \Omega$$

$$\text{Valor medido } [\Omega]_{min}$$

$$= (2,9052 - 0,0439)V \times \frac{(38,50 \text{ k}\Omega - 405 \Omega)}{(12,20 + 0,061)V - (2,9052 - 0,0439)V}$$

$$= 11378,6889 \Omega$$

$$\text{Valor medido } [\Omega] = 12048,7504 \pm 670,06614 \Omega$$

Dado que en la cercanía de los 21°C el sensor puede pensarse como un dispositivo lineal como se puede ver en la Figura III-11, el error calculado en Ω , se puede traducir a un error constante en $^{\circ}\text{C}$, usando la aproximación utilizada en el cálculo de la resolución de la siguiente forma:

$$\Delta^{\circ}C \leq \frac{670,06614 \Omega}{436,77 \frac{\Omega}{^{\circ}C}} = \pm 1,5314^{\circ}C$$

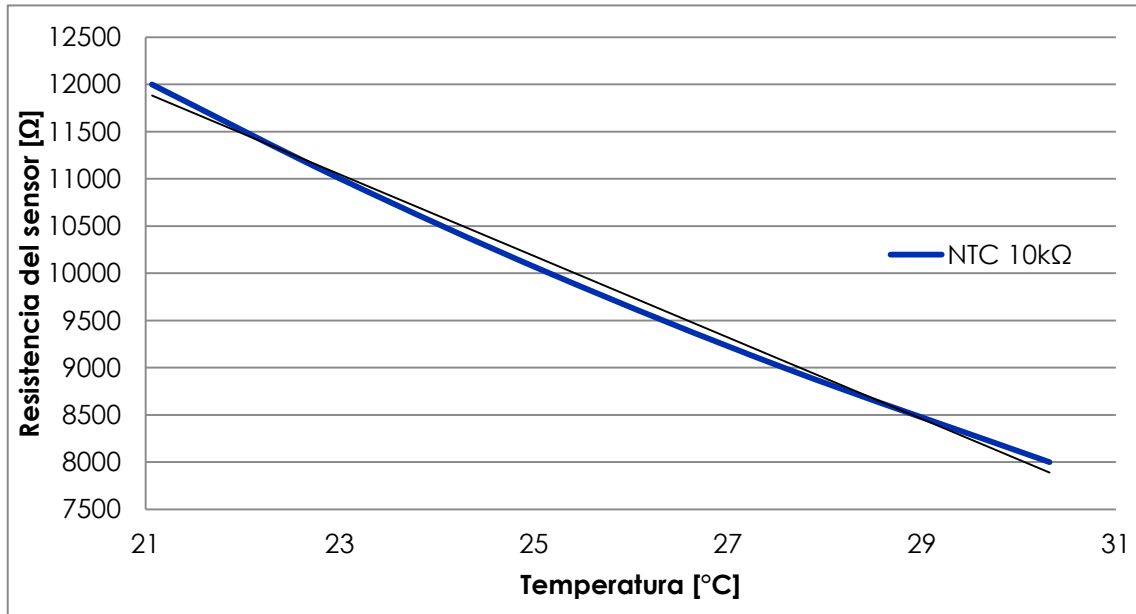


Figura III-11: Resistencia vs Temperatura NTC 10k Ω

2.4.4 Datos recopilados

El registro se realizó en el mes de en Abril de 2018 y duró alrededor de un mes, ya que abarcó tanto la etapa de fermentación como la de madurado, durante este periodo se tomó muestras cada un segundo, las que luego se procesaron realizando un promedio por minuto y luego un promedio móvil que abarcaba 20 minutos.

La variedad de levadura utilizada por la cervecería en este caso es una ALE US-05 cuya temperatura óptima de fermentación está en el rango de 15-24 °C, previo a la inoculación (agregado de la levadura al mosto) la misma fue mezclada con agua tibia (temperatura no medida) para hidratarla y acelerar su activación.

El fermentador en cuestión es un fermentador plástico de forma cónica, con un volumen de 500 litros, cubierto por una manta térmica para reducir la conducción de calor con el ambiente.

A continuación se muestran los datos sin procesar de los primeros 11 días del registro, los cuales contienen la totalidad del proceso de fermentación y el inicio del madurado.

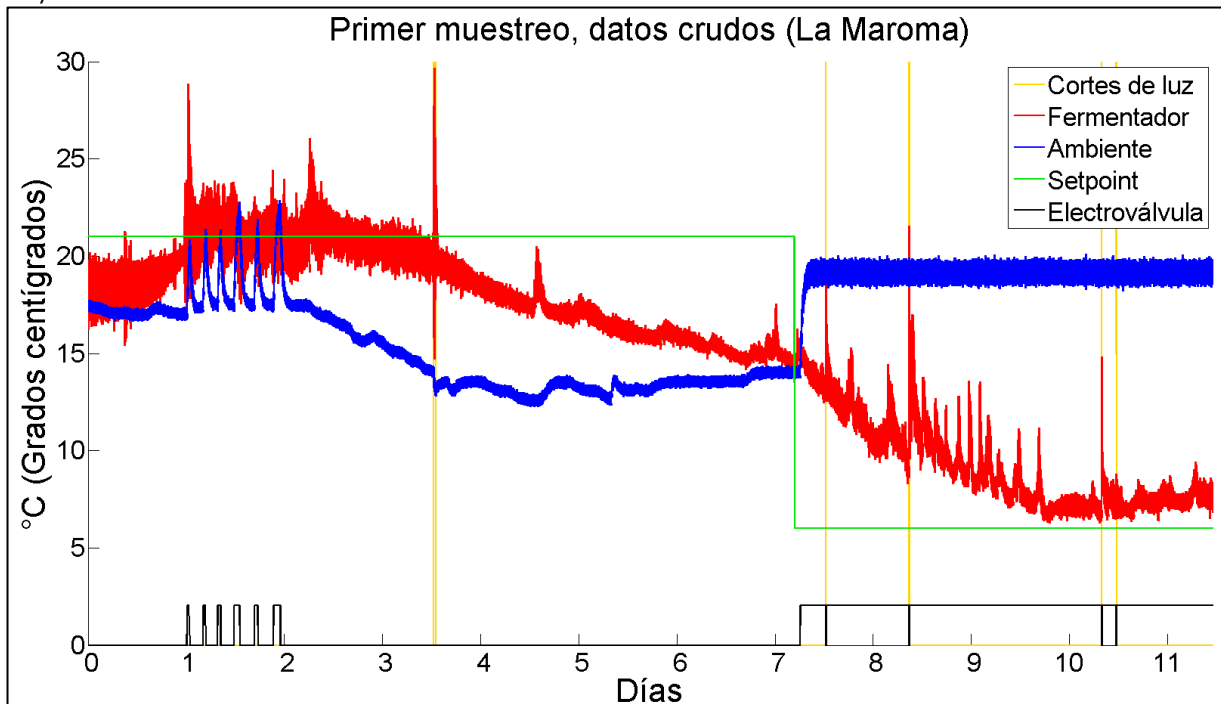


Figura III-12: Datos sin procesar adquiridos en el primer muestreo

En la Figura III-12 se observan tres problemas, el primero y más evidente es que la medición de la temperatura en el fermentador tiene mucho ruido. Esto se debió a que el sensor utilizado no era apto para estar en contacto con líquidos y si bien se esperaba que no hubiera líquidos en el alojamiento del sensor en el fermentador, al finalizar el muestreo se retiró el sensor y se pudo observar que el mismo estaba mojado debido a la condensación dentro del alojamiento.

El segundo problema fue que el sensor que utilizamos para medir la temperatura ambiente estaba pegado en el exterior del contenedor plástico que protegía el Arduino UNO y la fuente de 5V utilizada para registrar la electroválvula, el propósito de esto era evitar que la humedad y condensación del ambiente generara un mal funcionamiento de los dispositivos antes mencionados y la pérdida de los datos del registro. Si bien el contenedor cumplió su propósito, provocó debido a la hermeticidad del mismo que el calor generado por la fuente de 5V alterara las mediciones de la temperatura ambiente, por esta razón es que se puede observar en la Figura III-12, que en los intervalos en que la electroválvula se prende, hay picos en la temperatura ambiente.

El tercer y último problema fue la presencia de reiterados cortes de electricidad en la fábrica, generados por los mismos operarios debido a que el sistema de refrigeración presentaba problemas, los cortes del suministro eléctrico pueden observarse en la Figura III-12 como líneas verticales de color naranja. Durante estos cortes el datalogger no fue capaz de continuar con la medición debido a que carecía de una fuente alternativa de energía por esta razón es que se pueden observar cambios bruscos en la temperatura en estos puntos.

A pesar de los inconvenientes antes mencionados se pudo procesar los datos y aislar la información relevante presente en los mismos, la primera ac-

ción que se realizó fue eliminar las zonas en que la temperatura ambiente había sido afectada, reemplazando todo el periodo por el último valor conocido, para luego realizar un promedio sobre los datos obteniendo una muestra por minuto.

Como se puede observar en la figura siguiente, luego del procesamiento los datos son mucho más claros y los cambios de temperatura reflejan lo que se esperaría observar en un reactor de fermentación basándonos en la información sobre la etapa de fermentación desarrollada en la sección 1.2 de este mismo capítulo.

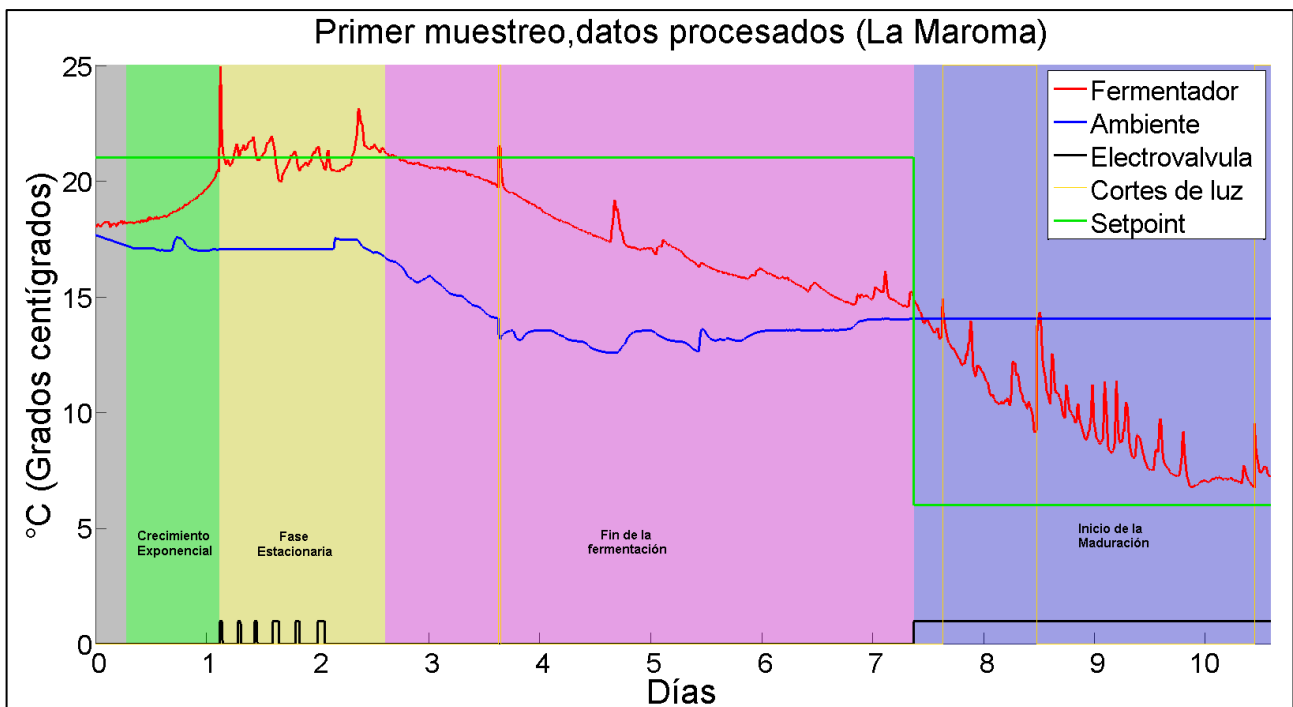


Figura III-13: Gráfica los datos procesados, evolución de la temperatura en un reactor de fermentación.

Las secciones de colores del gráfico delimitan, aproximadamente, las distintas etapas del proceso, en gris se puede observar la fase de adaptación de la levadura, seguida por la fase de crecimiento exponencial en color verde, luego en amarillo la fase estacionaria y en rosa la etapa de finalización de la fermentación en la que la población de la levadura comienza disminuir, y finalmente en azul se ve el comienzo de la etapa de maduración.

Luego de la etapa de crecimiento exponencial, se puede observar como el sistema de enfriado comienza a actuar, manteniendo la temperatura alrededor de los 21 °C. La actividad de la electroválvula está representada por la línea de color negro, tomando el valor 0 cuando la misma está cerrada, momento en el cual el sistema de enfriado no actúa sobre el fermentador y tomando el valor 1 en el caso contrario.

2.5. Ajuste del Modelo utilizando los datos recopilados

2.5.1 Ajuste del Modelo Sin Enfriador

Los parámetros del modelo fueron ajustados en función de los datos recopilados ya que si bien los datos adquiridos son del sistema a lazo cerrado, es posible realizar el ajuste debido a que la naturaleza del control es ON/OFF, por lo que el control solo actúa en algunas etapas del proceso y nuestros datos contienen mucha información del proceso a lazo abierto.

El procedimiento para ajustar los parámetros fue un proceso iterativo en el que se observaban discrepancias entre la curva del modelo y los datos obtenidos, se realizaba la modificación de un parámetro del modelo que estuviese relacionado con la característica a corregir y se volvía a iterar hasta que la curva del modelo reprodujera fielmente los datos. A continuación se muestra una tabla con los distintos parámetros del modelo, una descripción de los mismos y una explicación de cómo se ajustó su valor.

Parámetro	Descripción	Criterios de ajuste
γ (Gama)	Índice de reproducción de la levadura.	La modificación de este índice modifica la pendiente en la etapa de crecimiento exponencial, fue ajustado teniendo en cuenta la forma de la pendiente de los datos antes del accionar de la electroválvula.
β (Beta)	Índice de mortalidad por alcohol.	El alcohol es el parámetro que regula la pendiente de disminución de la población de levaduras luego de la etapa estacionaria, por lo que para ajustarlo se tuvo en cuenta la duración de la etapa de fermentación
α (Alfa)	Índice de producción de alcohol.	
η (Eta)	Índice de consumo de azúcar.	El azúcar es el limitante principal del crecimiento de la población de levadura y por lo tanto estos parámetros están directamente relacionados con el pico de temperatura, la duración de la fase de LAG, la velocidad de crecimiento de la población.
S_0	Cantidad inicial de azúcar	
$k_{t,ambiente}$	Conductividad calorífica hacia el ambiente.	Determina la cantidad de calor que fluye hacia el ambiente, fue ajustado teniendo en cuenta la pendiente de la temperatura del fermentador luego de que finalice la fermentación.
$k_{t,levadura}$	Constante de producción de calor, por la levadura.	Determina la cantidad de calor que genera la levadura y por lo tanto su aumento o disminución modifican la pendiente en la primera etapa de la fermentación y el pico de temperatura alcanzado por el fermentador si no fuese controlado. Fue ajustado teniendo en cuenta principalmente la pendiente inicial de la temperatura.
N_0	Población inicial de levadura.	Determina la duración de la fase de LAG de la levadura, y para ajustarla se tuvo en cuenta el inicio de la curva debida al crecimiento exponencial de la levadura.
A_0	Cantidad inicial de alcohol.	Se supone que la concentración inicial de alcohol es nula, por lo que no fue necesario ajustar este parámetro.
T_0	Temperatura inicial del fermentador.	Se ajustó según la temperatura inicial de los datos.
$T_{ambiente}$	Temperatura promedio del ambiente.	Junto con el coeficiente de conductividad calorífica hacia el ambiente determina la pendiente de la curva de temperatura luego de la fermentación y también el valor final de temperatura.
t	Constante temporal.	Esta constante fue agregada al modelo con el objetivo de establecer una relación temporal entre las muestras y el modelo, sirve para disminuir o aumentar la velocidad de todas las dinámicas del modelo, permitiendo "contraer o expandir" el eje de tiempo

Tabla III-4: Parámetros de ajuste de la simulación

A continuación se muestra un análisis del gráfico de la evolución de la temperatura del fermentador, resaltando las características utilizadas en el ajuste.

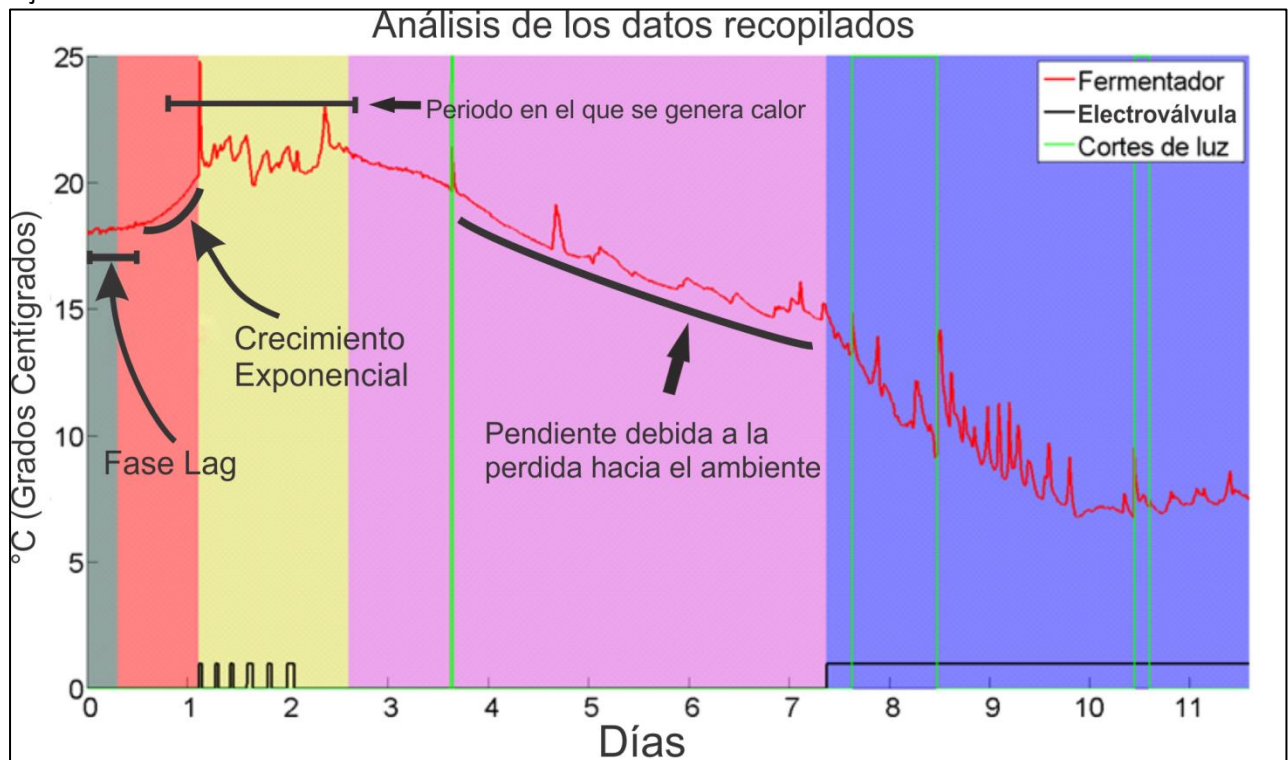


Figura III-14: Identificación de las características de la curva de temperatura registrada

Luego de ajustar los parámetros el modelo representaba satisfactoriamente los datos, como se puede observar en la siguiente figura, teniendo en cuenta que el modelo en esta etapa todavía no incluye el sistema de refrigeración, por lo que la temperatura del modelo sobrepasa los 21°C del setpoint.

Parámetros usados: $\gamma = 0.0023$, $\eta = 0.007$, $\beta = 0.0005$, $\alpha = 0.01$, $N_0 = 500$, $S_0 = 350$, $A_0 = 0$, $T_0 = 18$, $k_{t,levadura} = 0.00049$, $k_{t,ambiente} = 1.3$, $T_{ambiente} = 14$, $t = 0.000275$

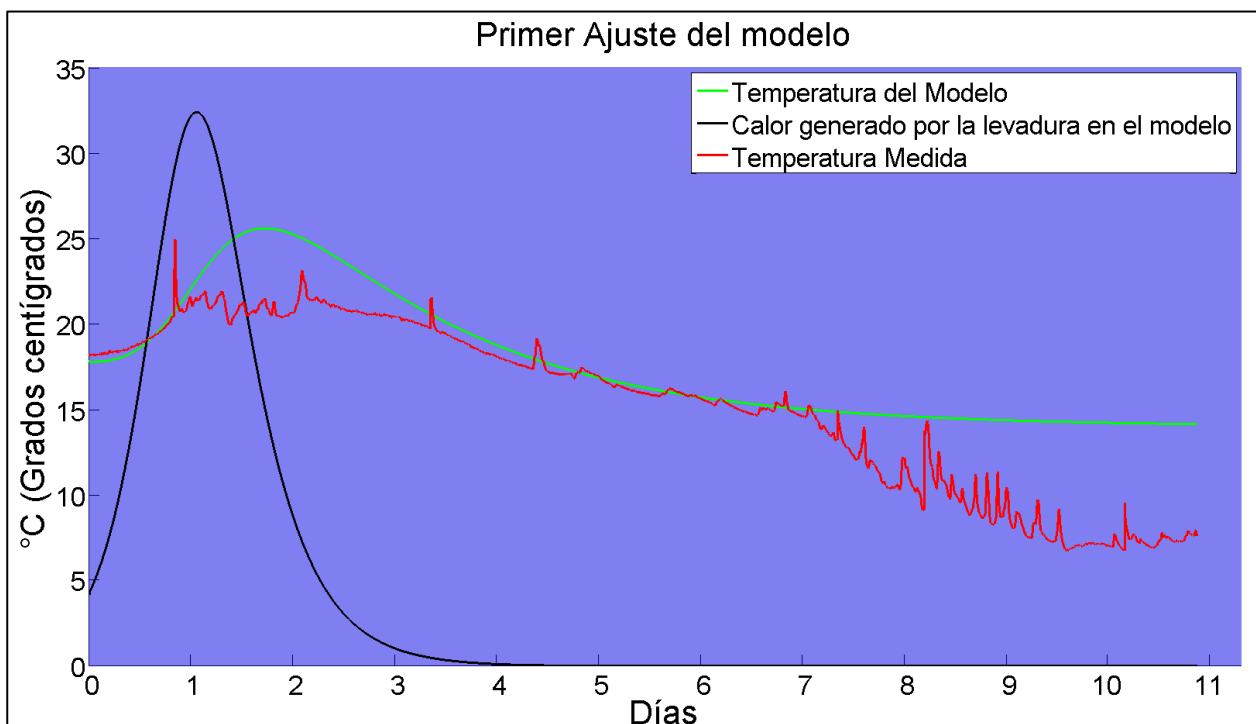


Figura III-15: Primer ajuste del modelo comparado con los datos reales.

Si bien el modelo ya es lo suficientemente completo como para simular las dinámicas de la levadura y los procesos térmicos dentro del fermentador, en vista de que se pretende agregar un sistema de enfriado al sistema es necesario incluir una relación entre el índice de reproducción de la levadura y la temperatura del fermentador, con el fin de que en la próxima sección al agregar el sistema de refrigeración, haya una interacción entre ambos sistemas y la curva de temperatura generada no sea simplemente la suma de la salida de ambos sistemas.

Para lograr esto se incluye en el modelo una curva que relaciona el índice de reproducción de la levadura con la temperatura del medio que habita, esto además de generar la interacción antes mencionada impedirá que la temperatura del fermentador aumente sin límite frente a cambios en los parámetros ya que si se dan las condiciones necesarias para que aumente la temperatura por encima del valor límite que la levadura puede resistir esta dejara de reproducirse y morirá por el efecto del alcohol.

La curva utilizada para relacionar índice de reproducción con la temperatura se puede observar en la Figura III-16, esta función se creó a partir de la información obtenida de "*Thermotolerant Yeast Strains Adapted by Laboratory Evolution Show Trade-Off at Ancestral Temperatures and Preadaptation to Other Stresses - Luis Caspeta, Jens Nielsen(2015)- p.2 Figura 1-A*" (5), ya que medir esta variación estaba fuera de los alcances del proyecto. La magnitud de la curva se ajustó en función del índice de crecimiento ya antes encontrado en el primer ajuste del modelo contra los datos.

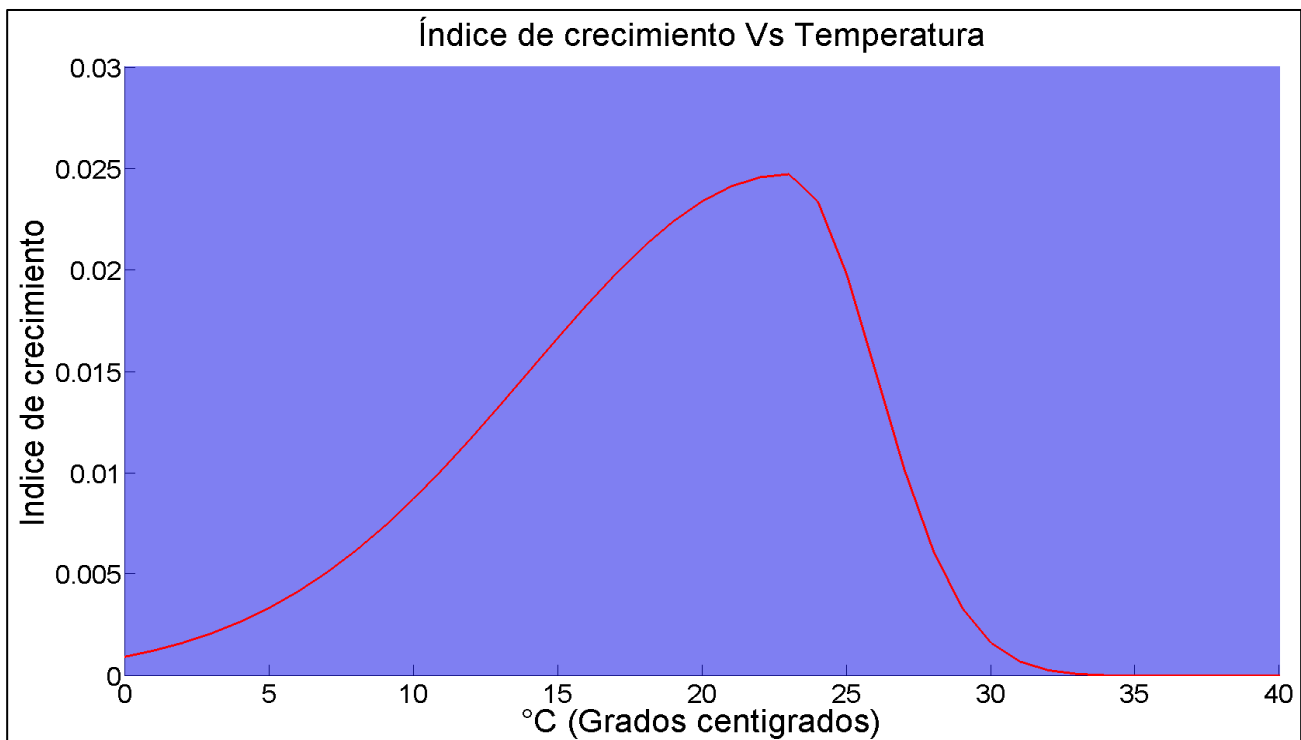


Figura III-16: Curva del Índice de reproducción de la levadura Vs Temperatura.

En la Figura III-15 se puede observar, que en las primeras horas del proceso la temperatura no disminuye, mientras que en nuestro modelo disminuye por la pérdida de calor hacia el ambiente, hasta que la levadura genera suficiente calor como para contrarrestar esta pérdida, por esta razón se suplanta la temperatura ambiente constante, utilizada en el primer ajuste, por

los datos recopilados de la temperatura ambiente, mostrados en la Figura III-13.

A continuación se muestran los resultados de la simulación luego de implementar los dos cambios antes explicados.

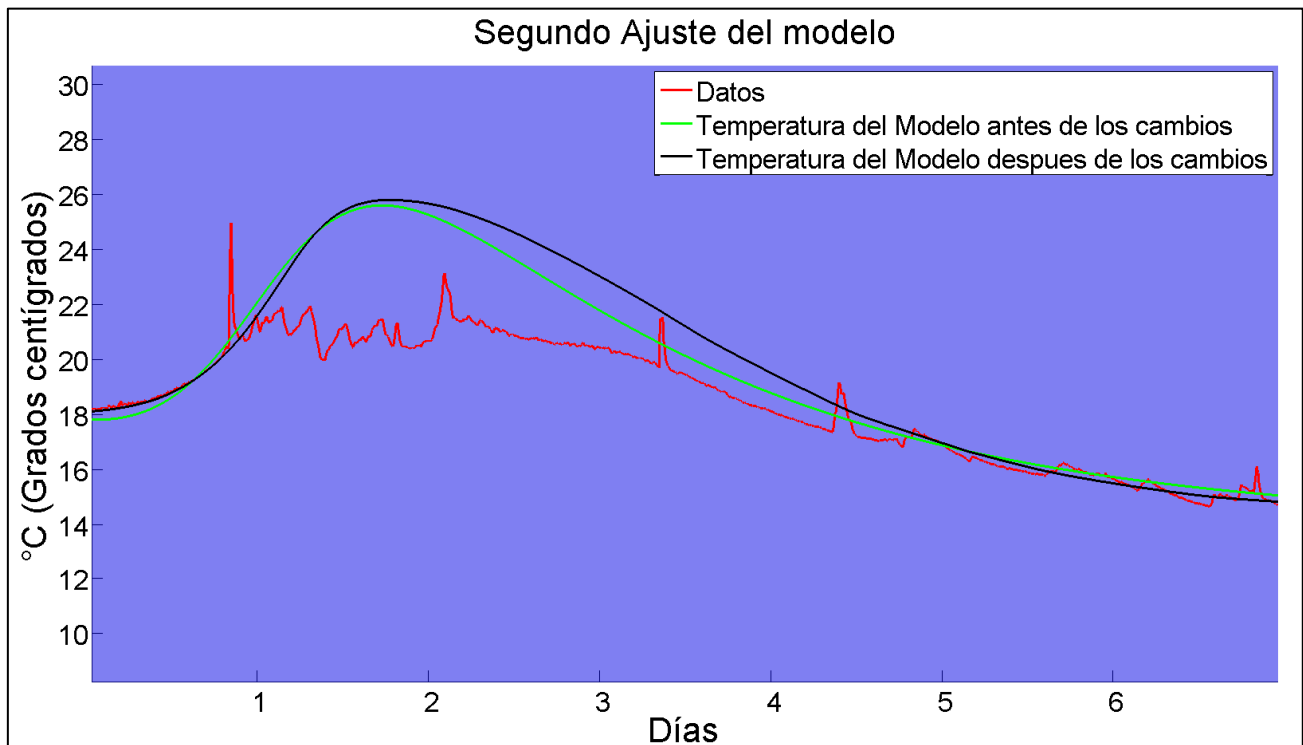


Figura III-17: Segundo Ajuste del Modelo, incluyendo curva de crecimiento y temperatura ambiente variable
 Parámetros usados $\gamma = 0.0023$, $\eta = 0.007$, $\beta = 0.0005$, $\alpha = 0.01$, $N_0 = 345$, $S_0 = 350$, $A_0 = 0$, $T_0 = 18$, $k_{t,levadura} = 0.00045$, $k_{t,ambiente} = 1.3$, $t = 0.000275$

Como se puede observar ahora el modelo coincide mucho mejor durante el primer día con los datos registrados, ya que la temperatura ambiente al iniciar la fermentación es mayor a el valor medio antes usado, y por lo tanto el calor perdido hacia el ambiente es menor. También se puede observar como la curva de temperatura es menos abrupta, esto se debe a que el índice de reproducción disminuye cuando la temperatura sube por encima de los 23°C lo que desacelera el proceso y por lo tanto la producción de calor.

2.5.2 Modelado matemático del sistema de refrigeración

En las secciones anteriores se desarrolló un modelo funcional que contempla las dinámicas propias del fermentador y la levadura, por lo tanto para tener un modelo completo del sistema que vamos a controlar con nuestro dispositivo lo único que resta modelar es el sistema de enfriado.

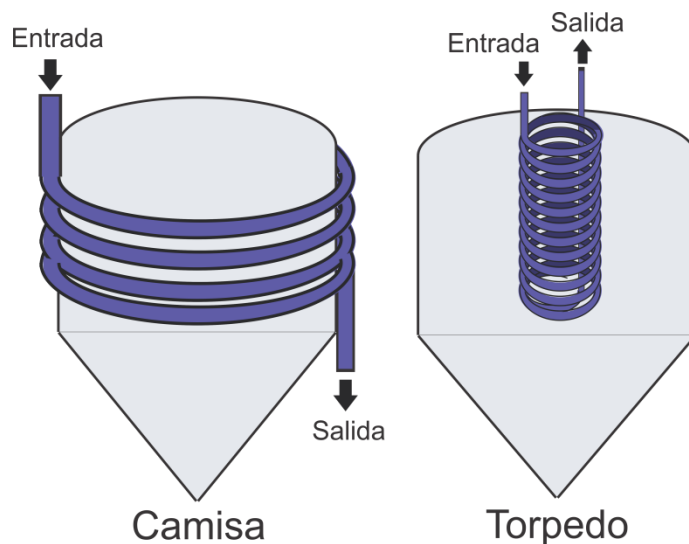


Figura III-18: Tipos de enfriadores

Los sistemas de enfriado para fermentadores están compuestos por un sistema de tubos, que pueden estar ubicados alrededor del tanque fermentador (camisa) o es su interior (torpedo) (Figura III-19), por estos se hace circular un líquido refrigerante cuyo paso es controlado mediante una electroválvula. El líquido refrigerante proviene de un enfriador o chiller, que mantiene la temperatura del líquido constante, y por medio de una bomba se fuerza su circulación por los tubos del fermentador. A continuación se muestra un diagrama del sistema completo.

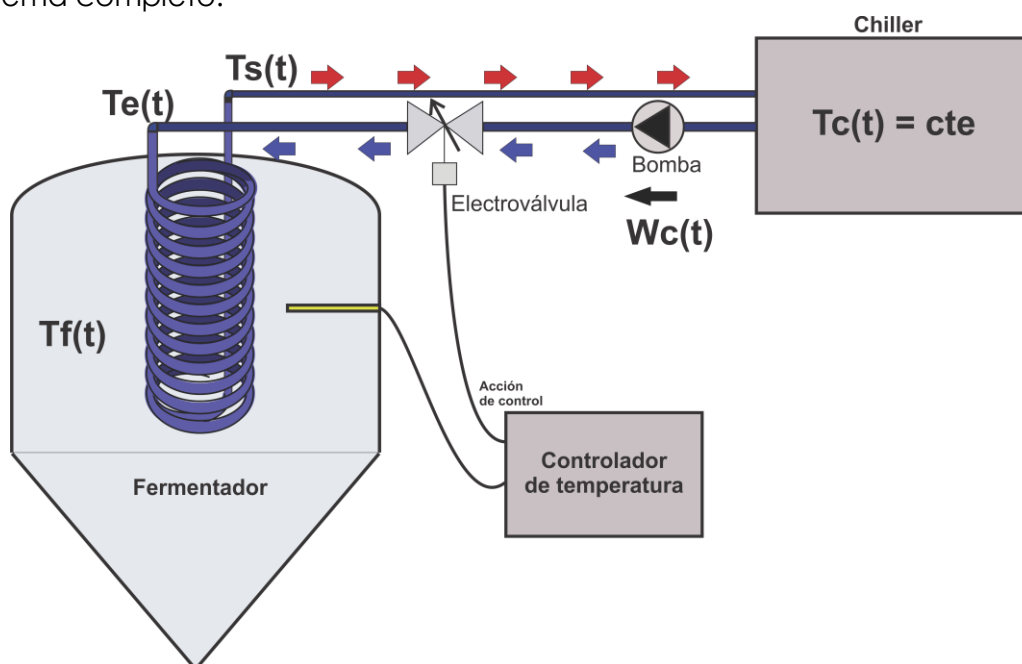


Figura III-19: Diagrama de temperaturas en el sistema Chiller-Fermentador

El modelo se desarrollará basándonos en las siguientes hipótesis:

- La temperatura del líquido refrigerante en el chiller $T_c(t)$ es constante, no se considera el efecto que podría tener la sobrecarga del chiller con el accionar simultáneo de fermentadores.
- La temperatura de entrada del refrigerante al fermentador $T_e(t)$ es igual a $T_c(t)$ debido a la aislación de los caños que transportan el refrigerante.
- El caudal que circula por el sistema de refrigeración $W_c(t)$ es constante, dado que se considera una bomba ideal despreciando las posibles perturbaciones que generaría tener otros fermentadores compartiendo el sistema de refrigeración.
- El coeficiente de transferencia por conducción si bien en la realidad varia con la temperatura, será considerado constante debido a que la temperatura del fermentador $T_f(t)$ se mantiene dentro de un rango acotado.
- Dado que todas las características del sistema que caracterizan la transferencia térmica modifican proporcionalmente al calor, se englobará bajo una sola constante $k_{Tcooler}$, la cual será proporcional a la resistencia térmica, el caudal del refrigerante, el calor específico y demás constantes propias de las características físicas del sistema.

El calor que se transfiere desde el fermentador al refrigerante $Q_R(t)$ se puede escribir entonces como:

$$Q_r(t) = k_{Tcooler} A_v(t - t_d) [T_f(t) - T_c(t)] \quad (9)$$

El coeficiente $A_v(t)$ (Apertura válvula) de la ecuación anterior, fue agregado con el propósito de tener en cuenta que el sistema de refrigeración no siempre está activo, ya que como se puede observar en la Figura III-19, el flujo de refrigerante es controlado por una electroválvula. Por este motivo en la simulación $A_v(t)$ tomará el valor "1" cuando el sistema de enfriado está activo y "0" en caso contrario. A su vez el coeficiente $A_v(t)$ está desplazado temporalmente t_d para tener en cuenta retardos de transporte en la planta debido al tiempo que le toma al líquido refrigerante circular por las tuberías.

Agregando $Q_r(t)$ a la ecuación (8) (dinámica de $T_{fermentador}(t)$), finalmente obtenemos el sistema de ecuaciones diferenciales que representa la planta en su totalidad, Fermentador/Ambiente/Enfriador.

$$\frac{dN(t)}{dt} = \gamma S(t)N(t) - \beta A(t)N(t)$$

$$\frac{dS(t)}{dt} = -\eta \gamma S(t)N(t)$$

$$\frac{dA(t)}{dt} = \alpha S(t)N(t)$$

$$\begin{aligned} \frac{dT_{fermentador}(t)}{dt} = & k_{t,levadura} \eta \gamma N(t) S(t) - k_{t,ambiente} [T_{fermentador}(t) - T_{ambiente}(t)] \\ & - k_{Tcooler} A_v(t - t_d) [T_f(t) - T_c(t)] \end{aligned}$$

2.5.3 Ajuste a datos de planta del modelo con circuito refrigerante

La ecuación del sistema de refrigeración fue implementada en software, y se le agregó un bloque que, accionado por la entrada "Electroválvula" conecta o desconecta el calor resultante a la salida del bloque permitiendo de esta forma "prender o apagar" el sistema de refrigeración $A_v(t)$, a su vez la entrada "Electroválvula pasa por un bloque de retardo el cual retrasa la señal en t_d .

A continuación se puede observar un diagrama en bloques de la implementación, en el cual las entradas $T_{fermentador}(t)$ y **Electroválvula** están resaltadas en celeste, las constantes $k_{Tcooler}$ y $T_c(t)$ en gris, y la salida del sistema en amarillo.

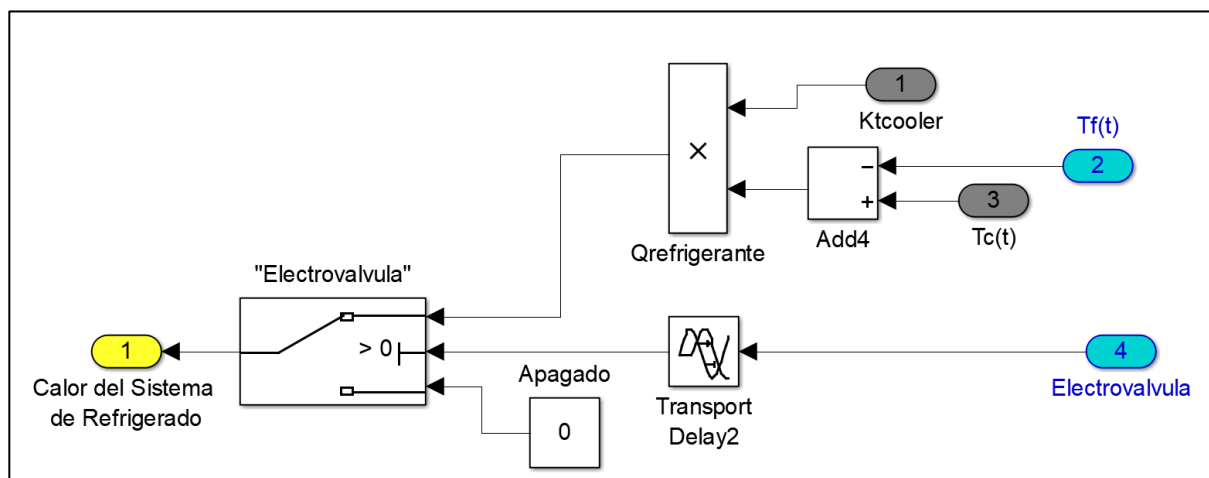


Figura III-20: Diagrama en bloques del Modelo del Sistema de enfriado.

Para ajustar los parámetros $k_{Tcooler}$ y $T_c(t)$, se interconectaron los bloques del modelo del enfriador y el fermentador (con los parámetros calculados en la sección 2.5.1), y se inyectó en la entrada "Electroválvula" los datos adquiridos del accionar de la electroválvula en la cervecería (Figura III 21). Luego de realizar la conexión de los bloques se varió los parámetros con el fin hacer coincidir la magnitud de pendiente de los datos con el modelo en las zonas que actúa la refrigeración.

A continuación se muestra el diagrama de la interconexión de los modelos del Fermentador y el sistema de refrigeración, seguido por un diagrama de la implementación del modelo del fermentador.

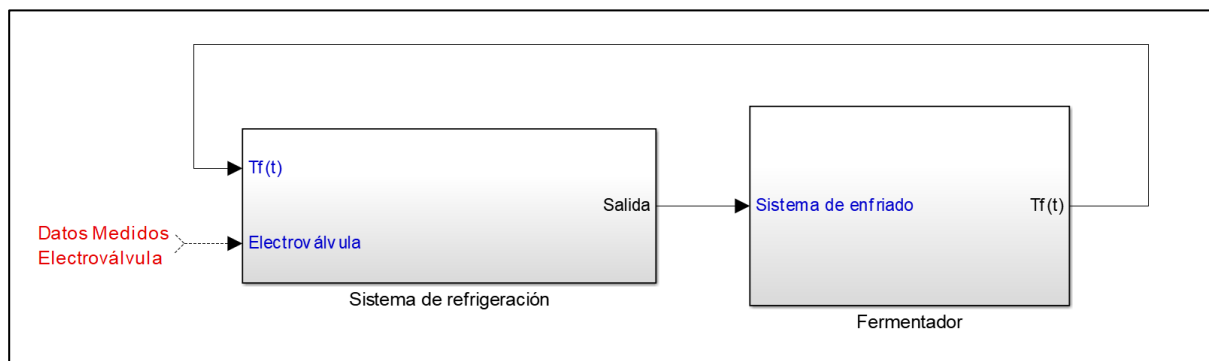


Figura III-21: interconexión Fermentador/Sistema de refrigeración

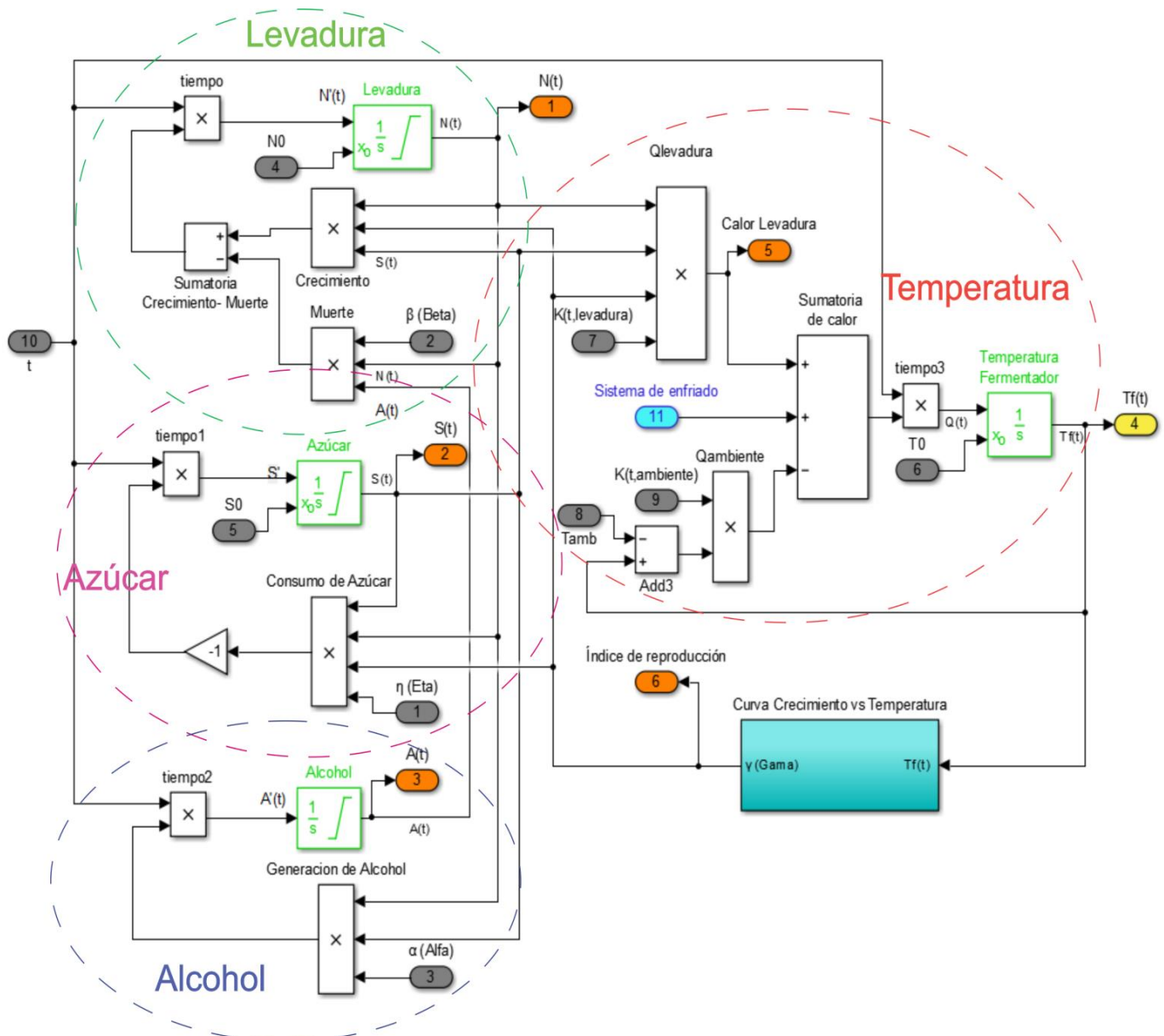


Figura III-22: Implementación del modelo del fermentador.

Los distintos círculos punteados del gráfico resaltan las áreas en que están implementadas las dinámicas de cada una de las variables, en gris se pueden observar los parámetros del modelo, en anaranjado las distintas variables internas del modelo, en celeste la entrada del sistema de refrigeración y en amarillo la salida de temperatura del fermentador.

En la figura III-23 y III-24 se puede observar el resultado de la simulación con los parámetros ajustados, la diferencia que se puede observar entre el modelo y los datos durante el último accionar de la electroválvula, se puede deber a un cambio en el coeficiente de conductividad $k_{T_{cooler}}$, el cual en el modelo fue supuesto constante. Sin embargo la etapa de mayor interés es la de fermentación, ya que es sobre la cual se realizarán las pruebas del algoritmo de control, y en esta zona el modelo se ajusta suficientemente bien a los datos considerando que estamos utilizando la señal adquirida de la electroválvula y no hay un lazo cerrado entre $T_{fermentador}(t)$ y el accionar del enfriador, por lo que determinamos que las simplificaciones y dinámicas tomadas en

cuenta en el modelo son suficientes para poder probar en primer instancia el controlador.

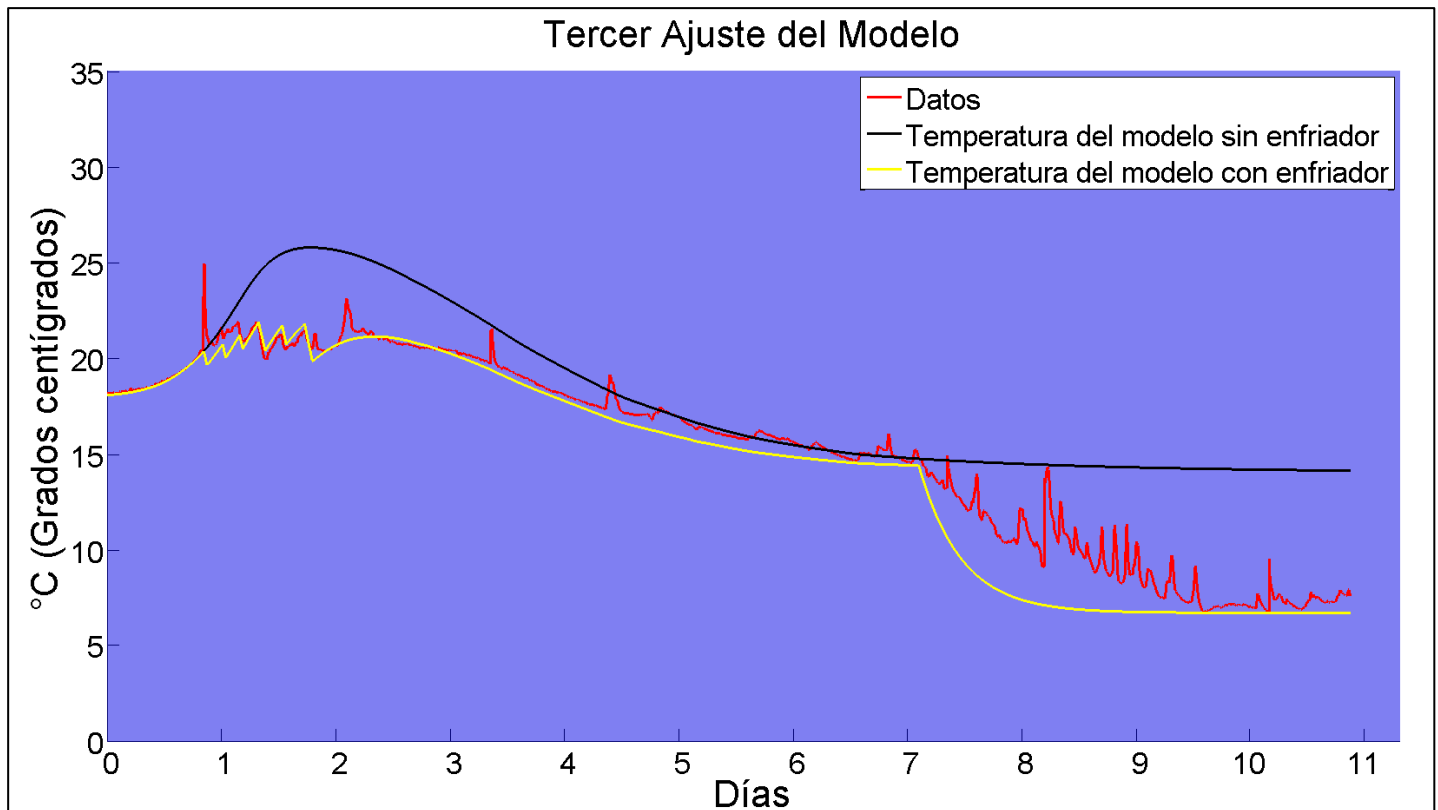


Figura III-24: Detalle de la etapa de fermentación del ajuste final.

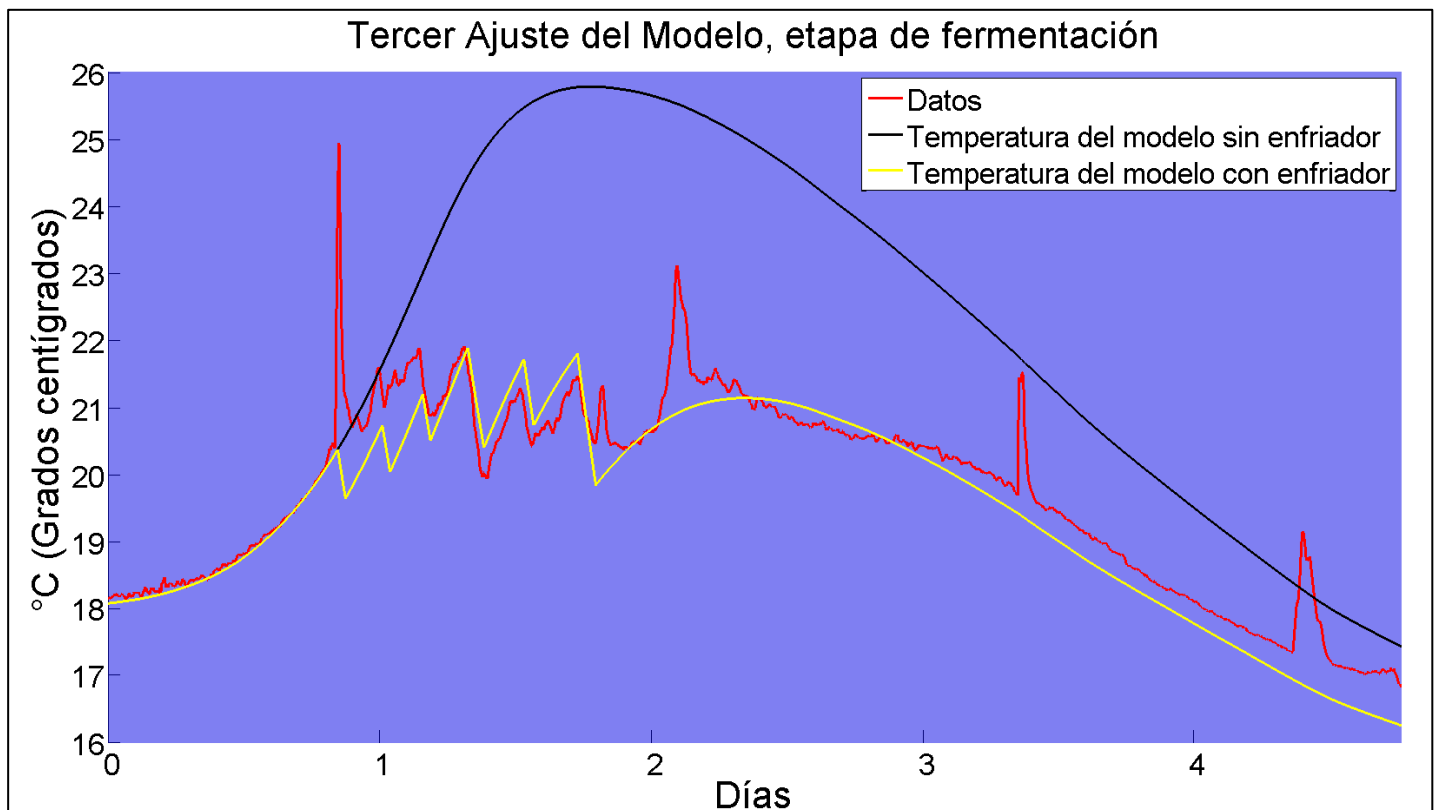


Figura III-23: Ajuste final del modelo completo de la planta.

Parámetros usados $\eta = 0.007$, $\beta = 0.0005$, $\alpha = 0.01$, $N_0 = 345$, $S_0 = 350$, $A_0 = 0$,
 $T_0 = 18$, $k_{t,levadura} = 0.00045$, $k_{t,ambiente} = 1.3$, $t = 0.000275$, $k_{Tcooler} = 5.5$, $T_c(t) = 5$

3. Validación del modelo

3.1. Análisis de los datos

Con el propósito de comprobar la validez del modelo se realizó un segundo muestreo en la fábrica de cerveza "La Maroma" ubicada en la localidad de El Bolsón, nuevamente utilizando el Arduino UNO como datalogger (Figura III.9), pero en esta oportunidad se usaron sensores recubiertos por un encapsulado metálico apto para estar en contacto con líquidos.

El registro duro 22 días y comenzó el 6 de Septiembre de 2018, la variedad de levadura utilizada por la cervecería en este caso es SafAle S-04 del fabricante Fermentis, previo a la inoculación la misma fue hidratada y luego incorporada al fermentador con el mosto ya cargado.

Se registró la temperatura del fermentador, el ambiente, el caño del refrigerante y el accionar de la electroválvula. Esta vez el setpoint fue fijado en 22°C, y dado que la temperatura del fermentador no sobrepasó este valor el controlador de temperatura no actuó durante la fermentación, lo que resultó sumamente beneficioso para la comprobación de la validez del modelo.

En el gráfico se puede observar en los primeros cuatro días el aumento de la temperatura del fermentador debido a la fermentación, seguido por la pendiente de pérdida de calor hacia el ambiente, ambas características ya observadas en los datos recopilados en la sección anterior.

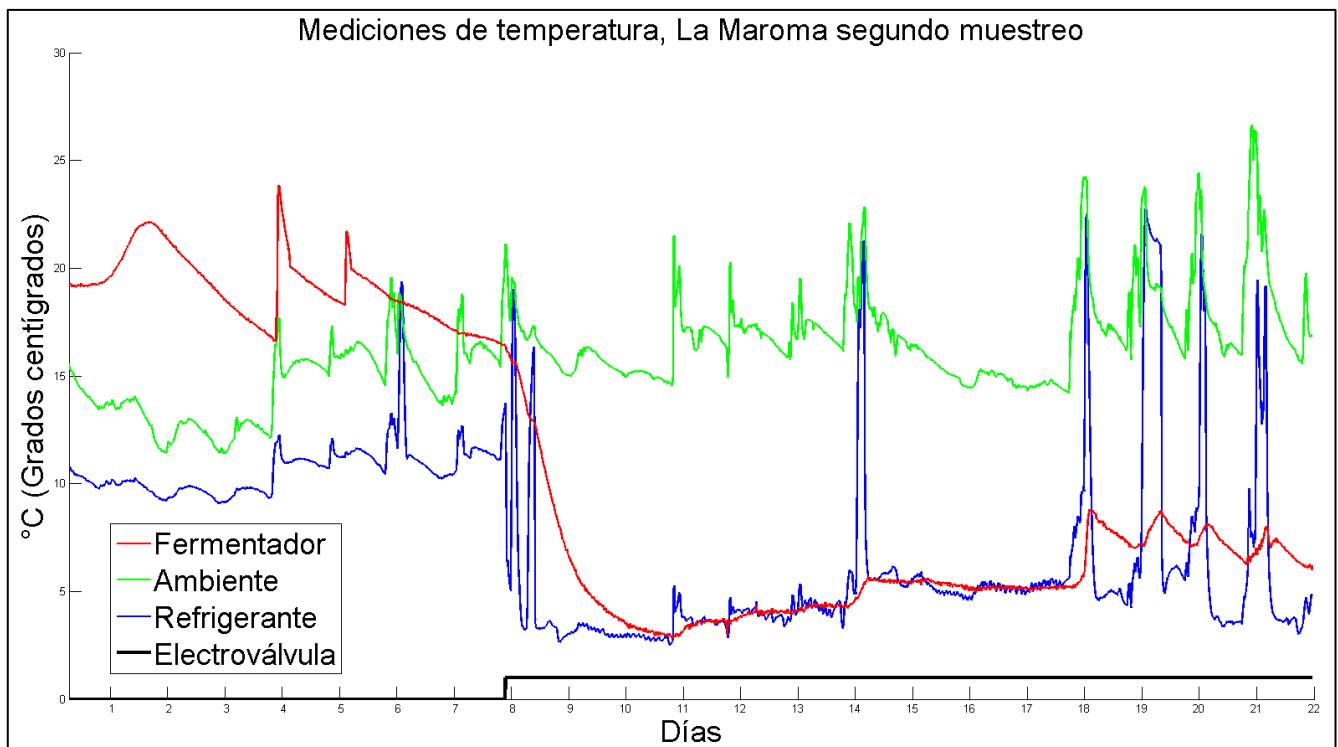


Figura III-25: Datos segundo muestreo

En los días 4 y 5 la temperatura del fermentador presenta dos máximos que alcanzan valores por sobre la temperatura ambiente, esto solo es posible con la existencia de una fuente de calor externa al sistema, dado que la fermentación ya había finalizado. Luego de consultar con los operarios, se concluye que esos valores fueron causados por la circulación de agua caliente por el sistema de refrigeración del fermentador, con el fin de intentar reactivar la fermentación debido a la baja temperatura ambiente.

En la medición de la temperatura del caño del Refrigerante se puede observar un cambio luego de la activación de la electroválvula debido a la circulación de refrigerante, esta medición será utilizada luego en la simulación para suplantar la constante que hasta ahora había sido usada como la temperatura del refrigerante. En los días 14, 18, 19, 20 y 21 se puede observar como la temperatura medida tiende a la temperatura ambiente, esto se debe a que los operarios debieron cerrar las válvulas manuales anteriores a la electroválvula impidiendo la circulación del refrigerante, como nos explicaron esta acción es tomada cuando se elabora una nueva cerveza, ya que el sistema de refrigeración es compartido con los intercambiadores de calor que enfrían el mosto antes de introducirlo a los fermentadores. Cuando se cocina una nueva cerveza el refrigerante se calienta, por lo que al cerrar las válvulas manuales evitan que el sistema de refrigeración caliente los fermentadores.

A continuación se muestra un gráfico del registro con anotaciones para ilustrar el análisis de los datos.

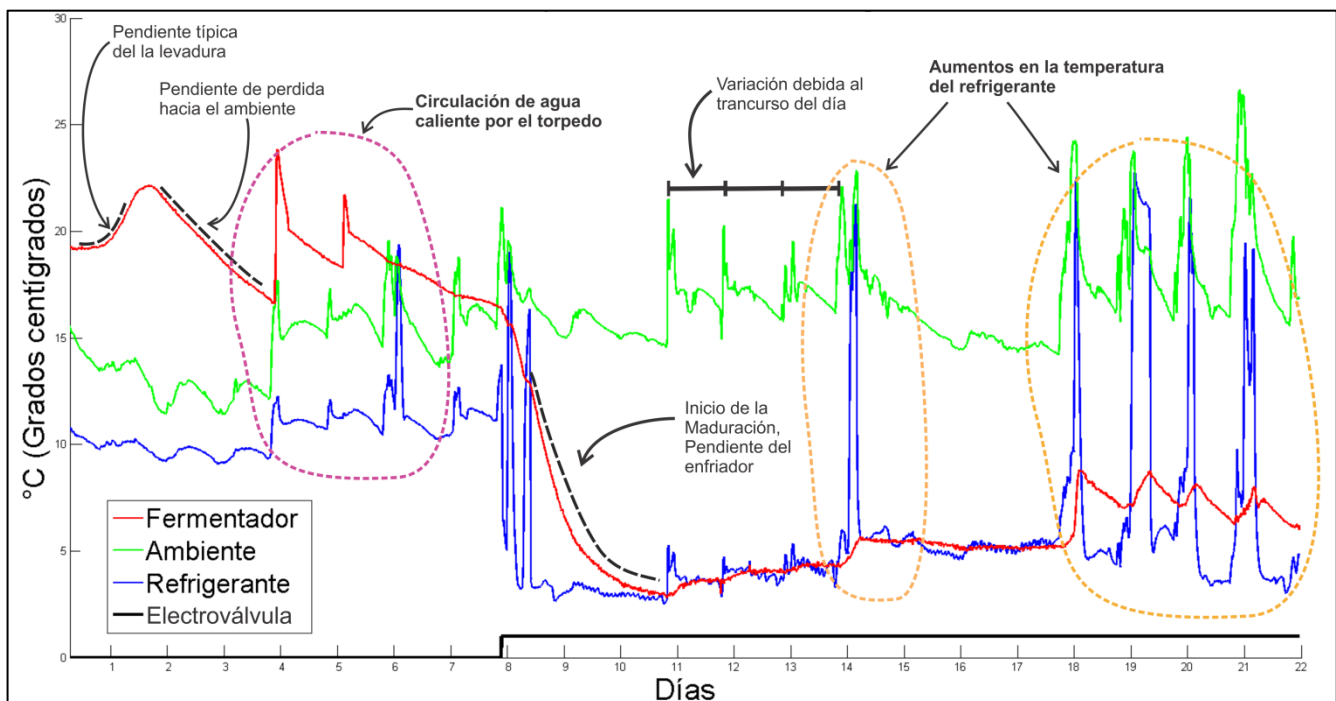


Figura III-26: Datos segundo muestreo detallado

3.2. Ajuste de los parámetros del modelo

Al igual que en la sección 2.5.3, se utilizaron los datos de la electroválvula como entrada del modelo de refrigeración, y los datos de la temperatura ambiente junto con los datos de la temperatura del refrigerante como parámetros del modelo del fermentador, para realizar la simulación y ajustar los parámetros del modelo con el fin de comprobar si las dinámicas del mismo representaban fielmente a la planta.

Para la simulación, la temperatura del refrigerante fue desplazada con el fin de eliminar el offset propio de la medición, ya que fue realizada sobre el caño del refrigerante y no sobre el refrigerante mismo. Dado que a temperatura promedio del chiller ronda los -1.1°C y la media del registro es de 6.5°C , se restaron 7°C a la temperatura del refrigerante para utilizarla en el modelo.

Para poder simular la circulación de agua caliente en los días 4 y 5, se modificó la señal de la electroválvula agregando dos pulsos con un ancho proporcional al pico registrado, y en los primeros 7 días, se fijó la temperatura del refrigerante en 65°C (este valor se obtuvo mediante prueba y error en la

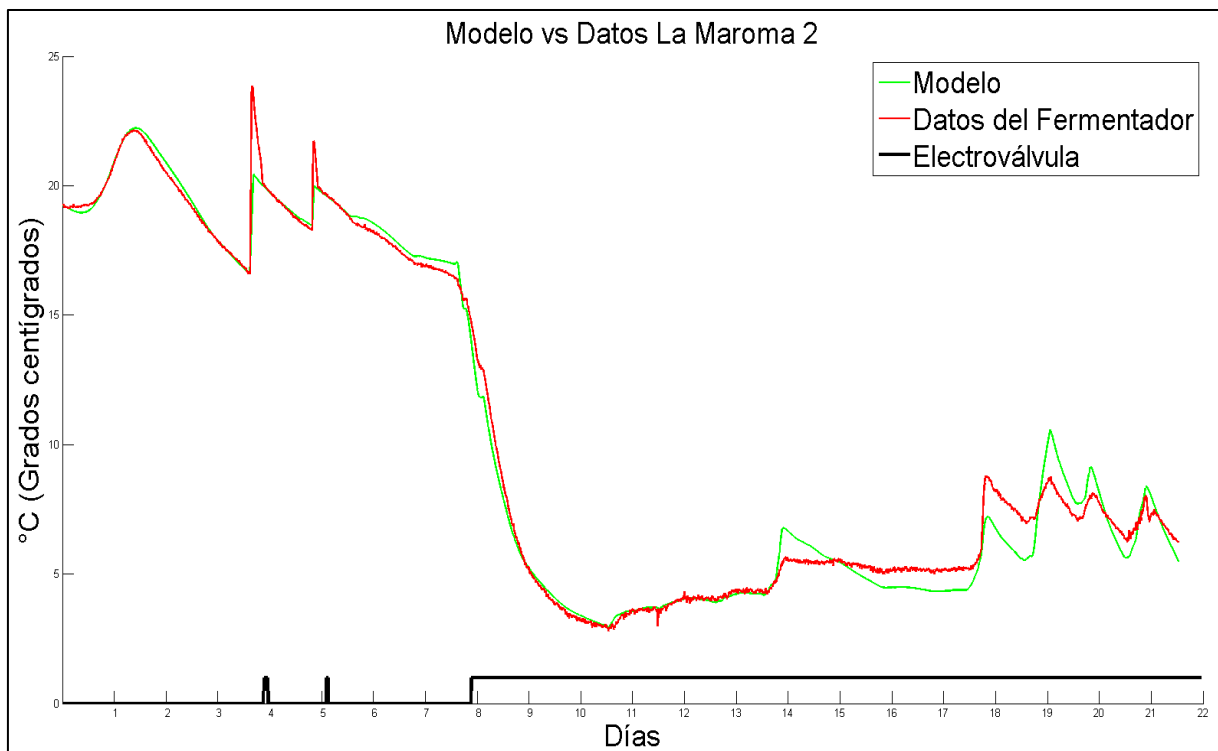


Figura III-27: Comparación entre el modelo y los datos

simulación).

Parámetros usados $\eta = 0.007, \beta = 0.0005, \alpha = 0.005, N_0 = 900, S_0 = 400, A_0 = 0,$
 $T_0 = 19,25, k_{t,levadura} = 0.000165, k_{t,ambiente} = 1.2, k_{tcooler} = 3, t = 0.000275$

En la Figura III.28, se puede ver como el modelo representa satisfactoriamente a la planta con una similitud especialmente alta durante los primeros 13 días, periodo durante el cual, el sistema de refrigeración no fue desactivado por los operarios. A continuación se puede observar en detalle los primeros 4 días del proceso, periodo durante el cual se desarrolla la fermentación y de mayor interés para el proyecto.

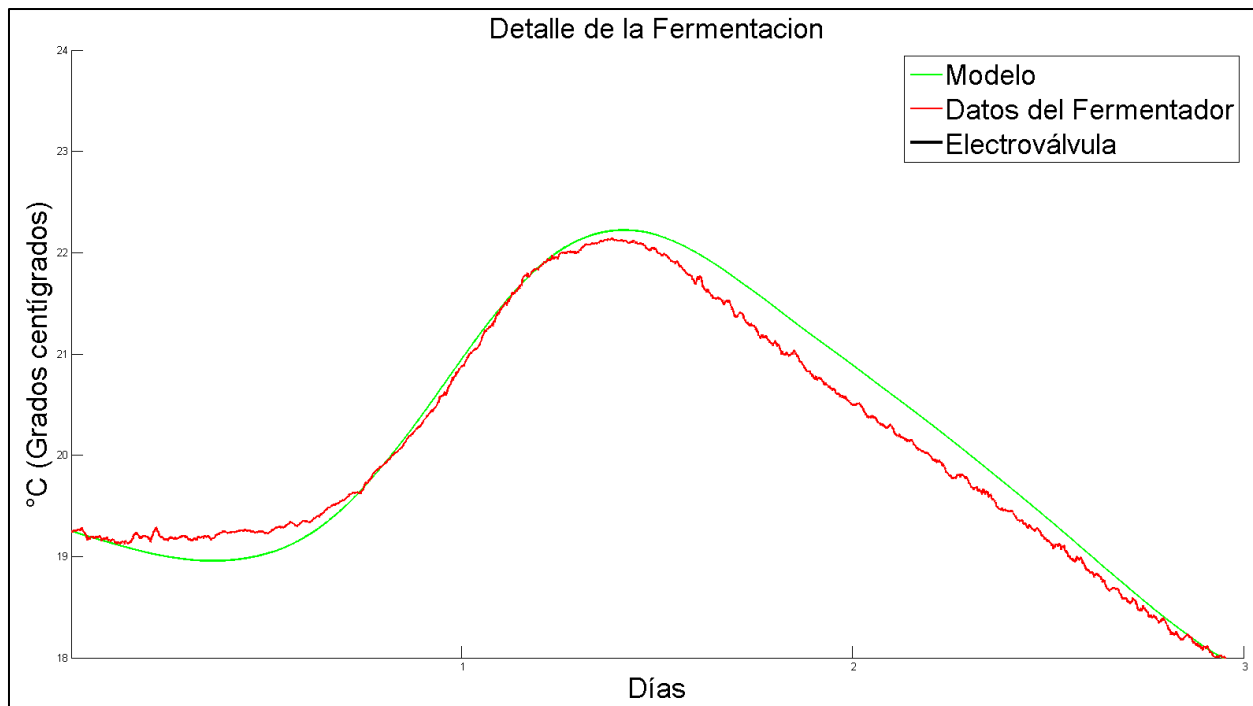


Figura III-28: Detalle de la etapa de fermentación

Durante este segundo ajuste varios parámetros fueron modificados, a continuación se muestra una tabla con una explicación de los cambios.

Parámetro	Valor	Causa
T_0 (temperatura inicial)	18°C → 19,25°C	La temperatura inicial del mosto no era la misma.
N_0 (población inicial de levadura)	345 → 900	Al cambiar el tipo de levadura es normal que estos parámetros cambien.
$k_{t,levadura}$ (constante de calor de la levadura)	0,00045 → 0,000165	
α (tasa de mortalidad por alcohol)	0,01 → 0,005	
$k_{tcooler}$ (constante del enfriador)	5,5 → 1,3	En este último ajuste la temperatura del refrigerante fue medida por lo que fue necesario modificar el $k_{tcooler}$ anteriormente usado.
$T_c(t)$ (temperatura del refrigerante)	5°C → - 1,1°C (media)	
$k_{t,ambiente}$	1,3 → 1,2	El cambio en este parámetro fue menor, ya que el fermentador era idéntico.

Tabla III-5: Variación de los parámetros entre el primer y segundo ajuste.

En la Figura III.29 se puede observar la comparación entre ambos ajustes, en la misma se puede observar como la cantidad de calor generada por el nuevo ajuste es menor debido al decremento de $k_{t,levadura}$, la fase LAG es menor debido al cambio en N_0 , y la pérdida de calor por el sistema de refrigeración es menor también debida a el decremento en $k_{tcooler}$.

Sin embargo se puede observar que a pesar de los cambios la dinámica del modelo representa satisfactoriamente a la planta a lazo abierto a pesar de los distintos cambios entre los dos muestreos por lo que damos por concluido el desarrollo del modelo.

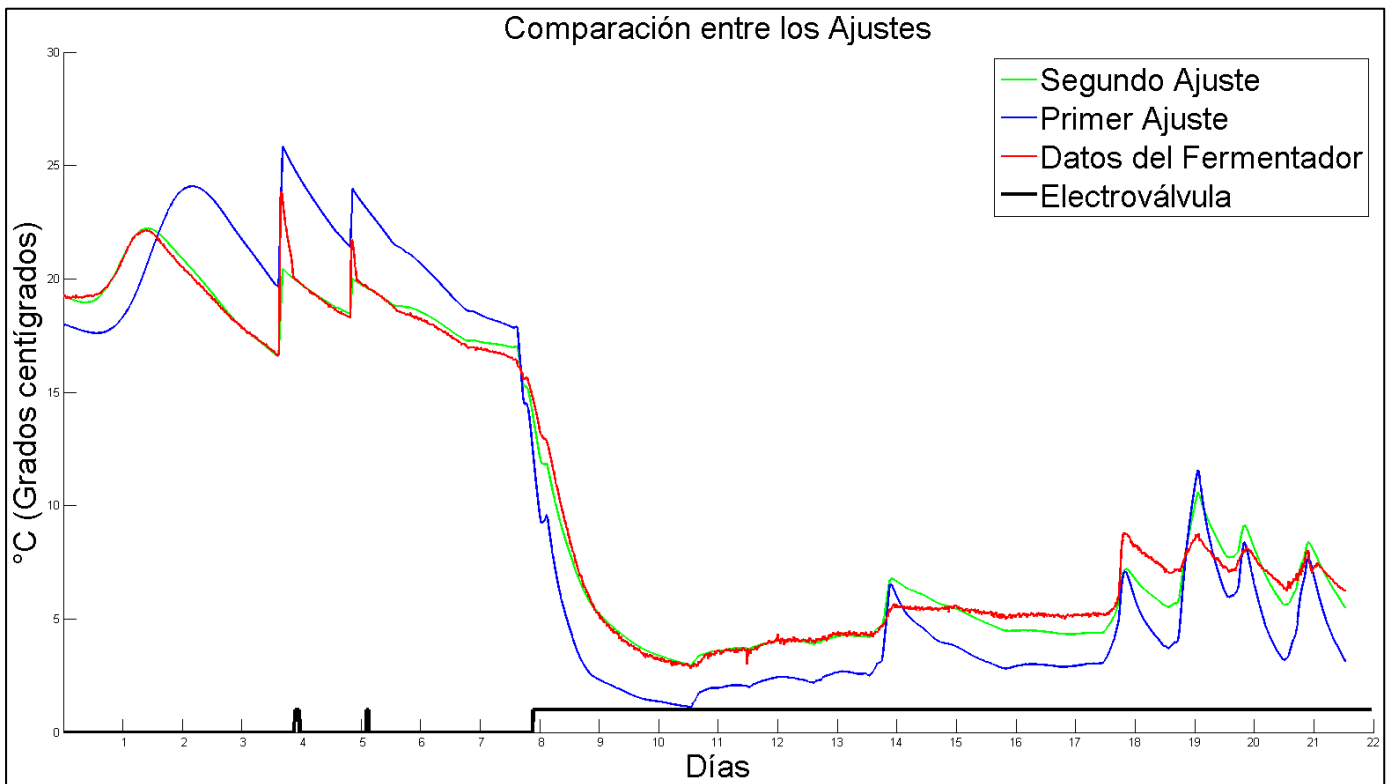


Figura III-29: Comparación entre los dos ajustes del modelo

IV - Estudio de lógica difusa

El presente capítulo ofrece una introducción a la lógica difusa donde se busca adquirir las herramientas elementales para luego comprender la implementación en lenguaje de programación "C". La implementación fue diseñada con una herramienta de cálculo por computadora y evaluada con los datos adquiridos durante la identificación de la planta del capítulo anterior. Una vez que se obtuvieron los resultados deseados se procedió a la implementación final en C para el micro controlador.

Se elige estudiar e implementar esta técnica de control principalmente porque tiene muy buen desempeño en áreas de control donde no resulta fácil encontrar un modelo matemático práctico. Si bien se podría utilizar un controlador ON/OFF (encendido/apagado) nos resultó interesante implementar una técnica de control no convencional para esta aplicación¹.

Un controlador ON/OFF funciona utilizando una banda de error. Es decir, si la temperatura del reactor supera cierto límite la salida es ON y si vuelve a estar dentro de "la banda" la salida es OFF. Si pensamos en un sistema que tiene una respuesta lenta, es decir que hay un tiempo considerable entre que se ejecuta la acción de control y podemos ver su efecto, este tipo de controlador reacciona de manera tardía produciendo que la temperatura oscile dentro y fuera de la banda.

Necesariamente debemos pensar en un controlador con una salida del tipo ON/OFF ya que las electroválvulas son de este estilo, no es posible graduarlas. Entonces como esa es una condición del sistema que queremos controlar el foco para mejorar el diseño esta puesto en las variables que describen el proceso.

En primera instancia la lógica difusa requiere, como se analiza más adelante, que las variables sean expresadas en diferentes conjuntos, es decir a diferencia de un control con una banda de error aquí requerimos que existan más de una banda (tantas como se crea conveniente) esto necesariamente permite que se procesen las variables con más detalle, pudiendo actuar de diferente manera para cada una de estas "nuevas" bandas.

Este tipo de controlador permite, una vez diseñado, agregar tantas entradas como se requieran sin mayores dificultades, por ejemplo además del error una entrada que ayudaría a predecir el comportamiento de la temperatura es la derivada de la temperatura. De esta manera mirando solamente el signo de la misma podemos definir si está creciendo o no. Otro factor que se puede agregar es la cantidad de días desde que inició el proceso, como se vio en el capítulo anterior la fermentación ocurre en diferentes etapas, muy diferenciadas, entonces la inercia térmica al inicio es muy diferente que al final del proceso, por lo que teniendo en cuenta esto se puede actuar acorde.

Esto último también se aplica para la salida, es posible agregar tantas salidas como se requieran sin necesidad de rediseñar. La flexibilidad es tal, que se puede utilizar una salida ON/OFF para la electroválvula y una salida PWM para una manta térmica. Todas estas características se pueden incorporar en diferentes etapas sin la necesidad de tener que modificar la estructura inicial convirtiéndolo en un controlador escalable.

¹ Ver Capítulo VI: Costos; Subtitulo: Productos similares.

“Lógica difusa (Fuzzy Logic) es una lógica. Lógica hace referencia al estudio de los métodos y principios del razonamiento humano. La lógica clásica, la que utilizamos día a día, trabaja sobre proposiciones (conclusiones o decisiones) que son verdaderas o falsas. Cada proposición tiene un opuesto. La lógica clásica entonces utiliza combinaciones de variables que representan proposiciones. Como cada variable se refiere a una proposición hipotéticamente verdadera o falsa, cualquier combinación de ellas eventualmente asume un valor de verdad (verdadero o falso) nunca un valor entre medio. No puede ocurrir que algo sea verdadero y falso al mismo tiempo.

El contenido principal de la lógica clásica es el estudio de reglas que permitan a funciones producir nuevas variables lógicas. Si suponemos n variables x_1, x_2, \dots, x_n tales que $\{x_1$ es verdadera, x_2 es falsa, \dots, x_n es falsa $\}$ y suponemos una nueva variable lógica “ y ” puede ser definida por una regla como función de $\{x_1, \dots, x_n\}$ que tiene un particular y único valor de verdad (verdadero o falso). Por ejemplo: “Si x_1 es verdadero Y x_2 es verdadero Y \dots Y x_n es falso ENTONCES y_1 es falso”. Es por esto que la lógica clásica es llamada también una lógica de dos valores.

El supuesto fundamental en que la lógica clásica está basada es que cada proposición es o verdadera o falsa. Este principio ha sido cuestionado por muchos filósofos a lo largo de los años y ahora es bien sabido y aceptado que muchas proposiciones son parcialmente falsas y verdaderas. Para poder describir esas parcialidades son necesarias nuevas reglas que permitan extender y generalizar “la lógica de dos valores a múltiples valores.” (6)

1. Diferencias y similitudes entre la Lógica “Clásica” y la Lógica Difusa

Tal como fue descrito en el anterior apartado, la lógica clásica (LC) o aristotélica posee un universo de valores con dos elementos, 1 o 0. Los elementos solo pueden pertenecer a un solo grupo. Por ejemplo podemos hablar de ropa “mojada” o “seca”. En la lógica difusa (LD) los elementos pueden pertenecer en diferente medida a todos los conjuntos, es decir la ropa puede estar tanto mojada como seca. Si en vez de definir solo “mojada” y “seca” definimos que los posibles conjuntos son “mojada”, “muy mojada”, “poco mojada”, etc un elemento puede pertenecer en diferente medida a cada uno de los conjuntos. Lo que define en que cantidad un elemento pertenece a un conjunto es la “función de membrecía”, $\mu(x) \in [0;1]$.

Estos valores lingüísticos (“muy mojada”, “poco mojada”, etc) son descripciones vagas de una realidad que depende mucho de la mirada de quien las enuncia y es justamente esta situación el fuerte de la lógica difusa. La lógica difusa y en particular los controladores basados en esta lógica poseen la capacidad de procesar información con alto grado de incertidumbre. Ya no es necesario identificar el modelo matemático exacto de un sistema para poder controlarlo, basta con tener el conocimiento de la experiencia humana y un conocimiento general del problema que se está intentando resolver.

Si bien esto facilita mucho el diseño también tiene una contrapartida, definir lo que se llaman “reglas de control”. Estas reglas son operaciones lógicas del tipo SI-ENTONCES que traducen entradas difusas en salidas concretas. Tanto para especificar las reglas como las funciones de membrecía es necesario muchas veces de la intuición humana y/o de la experiencia de un operario.

2. Funciones de membrecía.

Estas funciones definen la pertenencia de un elemento a un conjunto dado. Es decir que cada conjunto tiene su propia función de membrecía (FM) y son parte fundamental de los controles difusos.

Características:

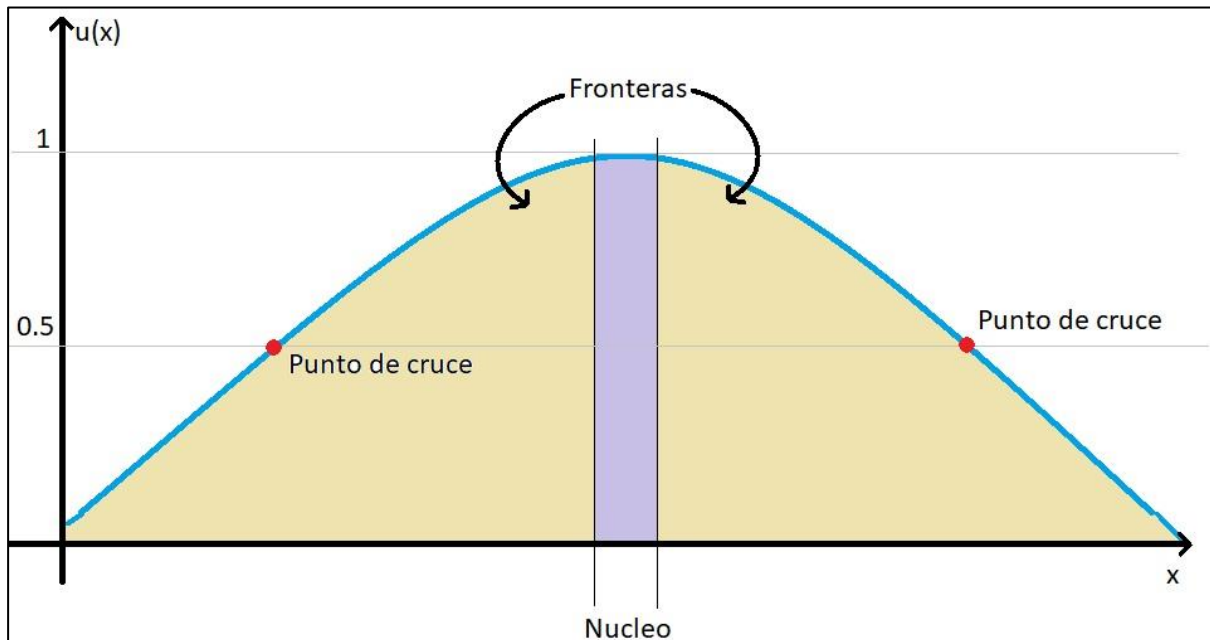


Figura IV-1- Partes de la función de membrecía

Tal como se ve en la figura IV-1, la imagen de la función de membrecía está acotada entre cero y uno (incluidos).

- $\mu(x) = 1$, se llama NUCLEO de la función de membrecía.
- $\mu(x) = 0.5$, se llaman PUNTOS DE CRUCE, la distancia entre ellos se define como el "ancho de banda".
- $0 > \mu(x) < 1$, se llaman FRONTERAS.
- El SOPORTE de la función es igual a la suma entre el núcleo y las fronteras.

Existen ciertas características que nos permiten clasificar las funciones de membrecía.

- Si la FM nunca alcanza el valor de 1 se dice que es una función del tipo ANORMAL en cambio si esta alcanza el valor de 1 se denomina NORMAL.
- Si la FM es simétrica con respecto al núcleo se la denomina SIMETRICA, caso contrario ASIMETRICA.
- Si la FM parte desde el valor cero y en el infinito mantiene un valor fijo diferente a cero se la denomina ABIERTA POR DERECHA, en el caso inverso, si comienza en un valor fijo diferente a cero y en el infinito su valor es cero se la denomina ABIERTA POR IZQUIERDA.
- Si la FM tiene ambos de sus extremos en cero se la denomina CERRADA.
- Si $\frac{d\mu(x_0)}{dx} = 0$ y $\frac{d\mu(x_0+\alpha)}{dx} > 0$ y $\frac{d\mu(x_0-\alpha)}{dx} < 0$ se dice que la FM es CONVEXA.

- Si $\frac{d\mu(x_0)}{dx} = 0$ y $\frac{d\mu(x_0+\alpha)}{dx} < 0$ y $\frac{d\mu(x_0-\alpha)}{dx} < 0$ se dice que la FM es CONCAVA.

3. Operaciones entre conjuntos difusos

De las propiedades matemáticas que poseen los conjuntos clásicos la única que no se cumple en la lógica difusa es la de complementariedad. Es decir que si defino un conjunto A y hago la unión de este conjunto con su complemento el resultado no es igual a uno. Es fácil de ver esto si recordamos que los elementos pueden pertenecer a más de un conjunto y que esa pertenencia depende de la función de membrecía de tal o cual conjunto.

Norma T

La norma T es una operación de intersección de dos conjuntos o más, que se representa con la letra T y posee ciertas características:

$$\mu_{A \cap B}(x, y) = T(\mu_A(x), \mu_B(y))$$

1. **Acotada:** $T(0,0) = 0$, $T(a,1) = T(1,a) = a$.
2. **Monótona:** $T(a,b) \leq T(c,d)$ si y solo si $a \leq c$ y $b \leq d$.
3. **Conmutativa:** $T(a,b) = T(b,a)$
4. **Asociativa:** $T(a,T(b,c)) = T(T(a,b),c)$

Existe principalmente 4 operaciones del tipo T:

- Intersección normal: $T_{min}(a(x), b(x)) = \min_x(a(x), b(x))$
- Producto algebraico: $T_{pa}(a(x), b(x)) = a(x) \cdot b(x)$
- Producto frontera: $T_{pf}(a(x), b(x)) = \max_x(0, a(x) + b(x) - 1)$
- Producto drástico: $T_{pd}(a(x), b(x)) = \begin{cases} a & \text{si } b(x) = 1 \\ b & \text{si } a(x) = 1 \\ 0 & a, b < 0 \end{cases}$

Norma S

La norma S es una operación de unión de dos conjuntos o más que se representa con la letra S y posee ciertas características:

$$\mu_{A \cup B}(x, y) = S(\mu_A(x), \mu_B(y))$$

1. **Acotada:** $S(1,1) = 1$, $T(a,0) = T(0,a) = a$.
2. **Monótona:** $S(a,b) \leq S(c,d)$ si y solo si $a \leq c$ y $b \leq d$.
3. **Conmutativa:** $S(a,b) = S(b,a)$
4. **Asociativa:** $S(a,S(b,c)) = S(T(a,b),c)$

Existe principalmente 4 operaciones del tipo S:

- Unión normal: $S_{max}(a(x), b(x)) = \max_x(a(x), b(x))$
- Suma algebraico: $S_{sa}(a(x), b(x)) = a(x) + b(x) - a(x) \cdot b(x)$
- Suma frontera: $S_{sf}(a(x), b(x)) = \min_x(1, a(x) + b(x))$
- Producto drástico: $T_{pd}(a(x), b(x)) = \begin{cases} a & \text{si } b(x) = 0 \\ b & \text{si } a(x) = 0 \\ 1 & a, b < 0 \end{cases}$

Norma N

La norma N es una operación de complemento de un conjunto.

$$\mu_{\bar{A}}(x) = N(\mu_A(x))$$

1. **Acotada:** $N(0) = 1, N(1)=0$
2. **Monótona:** $N(a) < N(b)$ si y solo si $a < b$
3. **Involutiva:** $N(N(a)) = a$
 - Complemento clásico: $N_c(a(x)) = 1 - a(x)$
 - Complemento de Sugeno $N_s(a(x)) = \frac{1-a(x)}{1+s \cdot a(x)}, s > -1$
 - Complemento de Yaguer $N_c(a(x)) = (1 - a(x)^w)^{\frac{1}{w}}, w > 0$

Inclusión

La inclusión de conjuntos está dada por las funciones de pertenencia de la siguiente manera:

Supongo que tengo dos conjuntos difusos A y B con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$ respectivamente, se puede decir que A está incluido en B si y solo si $\mu_A(x) < \mu_B(x)$ para todo x.

Producto cartesiano

Dado que no es posible crear funciones entre conjuntos difusos (ya que muchas veces un elemento de un conjunto se relaciona con más de un elemento de otro conjunto) en LD se utiliza el termino RELACION.

Una de las relaciones que se utilizan es el producto cartesiano que relaciona dos elementos para traducir matemáticamente una implicación.

Supongo dos conjuntos; A con elementos x_1, x_2, \dots, x_n y función de membrecía $\mu_A(x)$ y el conjunto B con elementos y_1, y_2, \dots, y_n y función de membrecía $\mu_B(y)$

$$R_{AxB} = AxB = \{x_i X y_j\}$$

$$\mu_{AxB}(x, y) = \min_x(\mu_a(x), \mu_b(y))$$

Composición

$$R_3 = R_1 \circ R_2 = \mu_{R_3}(x, z) = \max[\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))]$$

Ejemplo

$$R_1 = AxB = \begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{matrix}$$

$$R_2 = BxC = \begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{matrix}$$

$$R_3 = \begin{matrix} \max(\min(1,5), \min(2,7)) & \max(\min(1,6), \min(2,8)) \\ \max(\min(3,5), \min(4,7)) & \max(\min(3,6), \min(4,8)) \end{matrix}$$

Principio de extensión

Si definimos una función entre conjuntos difusos que tome elementos del conjunto de partida A y crea un nuevo conjunto B con la salida de esa función, dicho conjunto tendrá una función de membrecía definida como;

$$f: A \rightarrow B, y = f(x), x \in A, y \in B$$

$$\mu_B(y) = \max_y(\mu_A(x))$$

4. Variable lingüística

Cuando se define resolver un problema con LD lo primero que se debe hacer es definir las variables lingüísticas que se utilizaran. Estas variables definen los conjuntos difusos, los elementos y las funciones de membrecía;

- $V(x)$, Valores que acepta la variable.
- x, Nombre de la variable.

- X , Universo.
- $\mu(x)$, Función de membresía.

Ejemplo

$V(x) = \{\text{muy mojado, poco mojado, mojado}\}$

$x =$ cantidad de agua

$X = [0, 1]$ litro

$\mu(x) = \text{gauss}(x, \text{sigma} = 2, \text{centrada en } 0,5)$

5. Control difuso

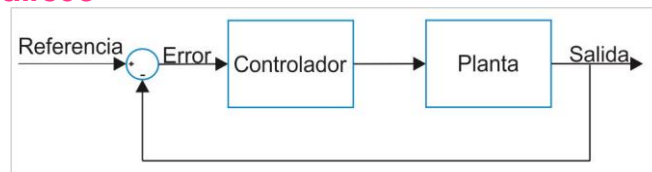


Figura IV-2: Diagrama de control realimentado



Figura IV-3: Procesos del controlador

Tanto el “pre procesamiento” como el “pos procesamiento” son etapas de acondicionamiento de la señal, tanto para la entrada como para la salida. Estas etapas pueden ser de amplificación, filtrado digitalización, etc.

Fusificación: En esta etapa, a partir del valor certero que entra al controlador, se obtiene la pertenencia a cada una de los valores lingüísticos. (7)

Reglas de control: relaciona cada una de las expresiones SI-ENTONCES, con los valores encontrados en la Fusificación.

Defusificación: Dado que el resultado del bloque “reglas de control” devuelve un valor difuso, es necesario volver a traducir eso en una acción de control certera.

5.1. Método de inferencia de Mandani

El método de inferencia nos permite traducir las expresiones lógicas en expresiones matemáticas para poder ser programadas.

Antes de continuar con el método propiamente dicho vamos a advertir el significado de una expresión que nos permitirá simplificar la explicación luego.

La expresión “ x es A ” significa que el valor “ x pertenece con cierto valor de pertenencia al conjunto A ”. Debido a que hablamos de pertenencia se entiende que el conjunto A es a priori un conjunto difuso.

Vamos a expresar propositivamente y matemáticamente el controlador difuso.

Propositivamente	Matemáticamente	
x_0 es A' e y_0 es B' , es la entrada	$R = A' \times B'$	
Reglas de control		
Si x es A_1 e y es B_1 , ENTONCES z es C_1	$R_1 = A_1 \times B_1 \times C_1$	$C_1' = R \circ R_1$
Si x es A_2 e y es B_2 , ENTONCES z es C_2	$R_2 = A_2 \times B_2 \times C_2$	$C_2' = R \circ R_2$
.	.	
.	.	
Si x es A_i e y es B_i , ENTONCES z es C_n	$R_n = A_i \times B_i \times C_n$	$C_n' = R \circ R_n$
Conclusión		
z es C'		$C' = C_1' \cup C_2' \cup \dots \cup C_n'$

Cabe destacar que tanto x_0 como y_0 son entradas certeras por lo que A' y B' puede definirse como conjuntos con un solo elemento y con una función de pertenencia del estilo DELTA o SINGLETON.

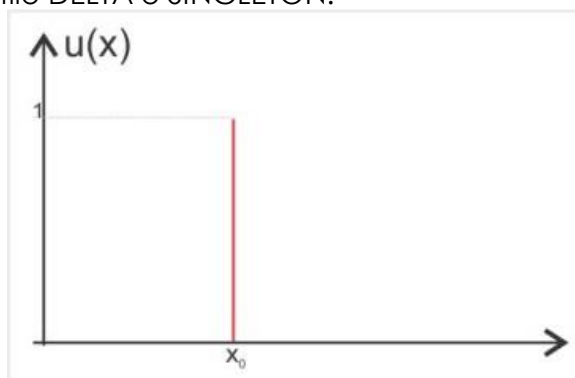


Figura IV-4: Función Delta

$$\mu_{C'_i}(z) = \max_{x,y} [\min(\mu_R(x,y), \mu_{R_i}(x,y,z))]]$$

$$\mu_{C'_i}(z) = \max_{x,y} [\min(\min(\mu_{A'}(x), \mu_{B'}(y)), \min(\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z)))]]$$

$$\mu_{C'_i}(z) = \max_{x,y} [\min(\mu_{A'}(x), \mu_{B'}(y), \mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z))]]$$

$$\mu_{C'_i}(z) = \min \left\{ \max_x (\min(\mu_{A'}(x), \mu_{A_i}(x))), \max_y (\min(\mu_{B'}(y), \mu_{B_i}(y))), \mu_{C_i}(z) \right\}$$

Como A' y B' son del tipo delta, el mínimo entre ambas funciones de pertenencia es siempre la de la regla evaluada en ese punto, por lo tanto

$$\mu_{C'_i}(z) = \min(\mu_{A_i}(x_0), \mu_{B_i}(y_0), \mu_{C_i}(z))$$

A esto se le llama VALOR DE FUSIFICACIÓN, aun que en verdad no es un único valor sino una función.

Gráficamente esto se puede analizar de la siguiente manera y simplifícadamente a solos dos reglas y una sola entrada x_0 .

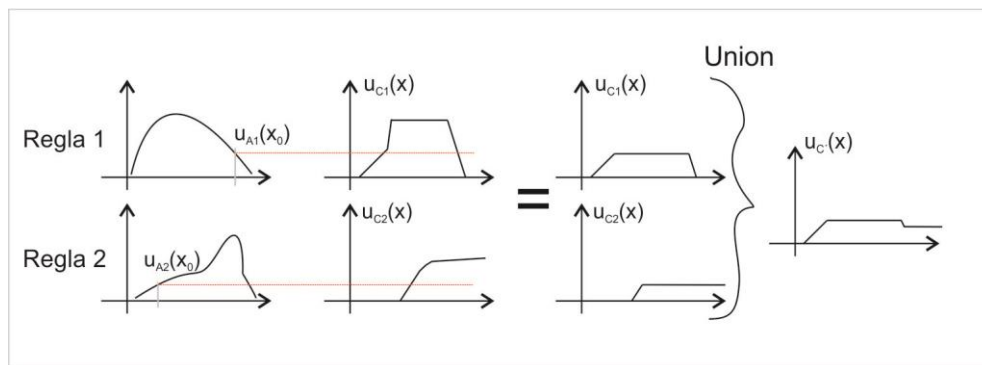


Figura IV-5: Ejemplo de Fusificación

El siguiente paso es el de la Defusificación y para esto existen varios métodos que a partir de la función $\mu_C(x)$ computan el valor de salida.

5.2. Defusificación

Existen varios métodos de Defusificación, la utilización de alguno en particular depende del problema.

Si suponemos que y_0 es la salida y $\mu_{C'}(z)$ como el valor de fuzzyficación;

Centroide

$$y_0 = \frac{\sum z \cdot \mu_{C'}(z)}{\sum \mu_{C'}(z)}$$

Bisectriz

Se basa en encontrar el punto que divida el área de la función de pertenencia en dos partes iguales. Si supongo una función que pertenece al intervalo $[A, B]$.

$$\int_A^{y_0} \mu_{C'}(z) dz = \int_{y_0}^B \mu_{C'}(z) dz$$

Máximo central

La salida es el promedio de los máximos de la función.

Máximos más pequeños y más grandes

La salida es el máximo más pequeño o el máximo más grande de la función de pertenencia.

6. Implementación del controlador

Para la implementación del controlador se generó una biblioteca² codificada en el lenguaje de programación C. El código se realiza en base a un artículo publicado Gregory Viot (8), pero tal como se aclaró al inicio del capítulo antes de la implementación propiamente dicha se utilizó un software de computadora específico de controladores de lógica difusa que permitió verificar las reglas de control.

En este apartado se ven las diferentes partes y estructuras del código de la librería utilizando como punto de partida las variables del problema nuestro.

6.1. Estructura de la biblioteca

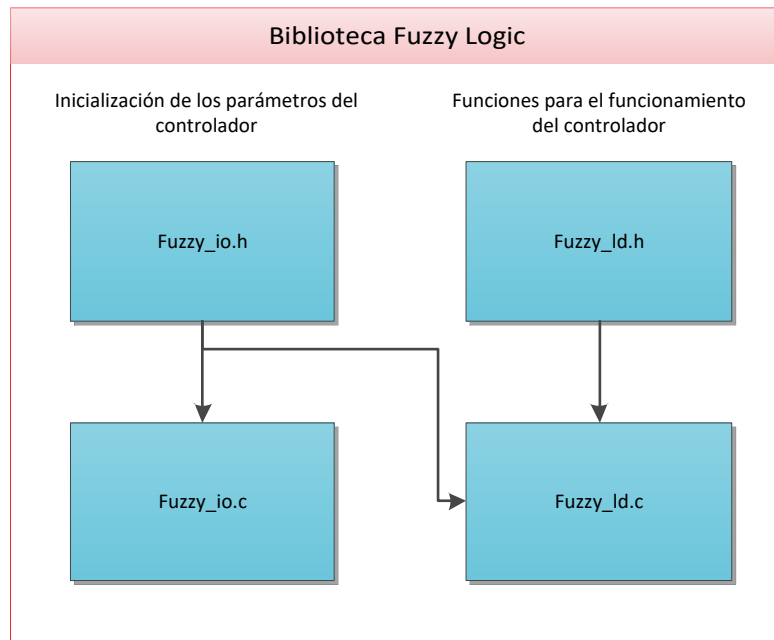


Figura IV-6: Estructura de la librería.

La librería consta de cuatro archivos tal como se ve en la figura IV-6. Las flechas indican donde está incluido el archivo de donde parte la flecha:

1. Fuzzy_io.h: Contiene las estructuras, definiciones y la declaración de la función de inicialización.
2. Fuzzy_io.c: Contiene la implementación de la función de inicialización.
3. Fuzzy_ld.h: Contiene las declaraciones de las funciones necesarias para el funcionamiento del controlador.
4. Fuzzy_ld.c: Contiene la implementación de las funciones declaradas en fuzzy_ld.h.

6.2. Variables lingüísticas

Se eligieron tres variables lingüísticas de entrada, el "error", la "tasa de cambio de la temperatura" y "Refrigerante".

² Def: En informática, una biblioteca o librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

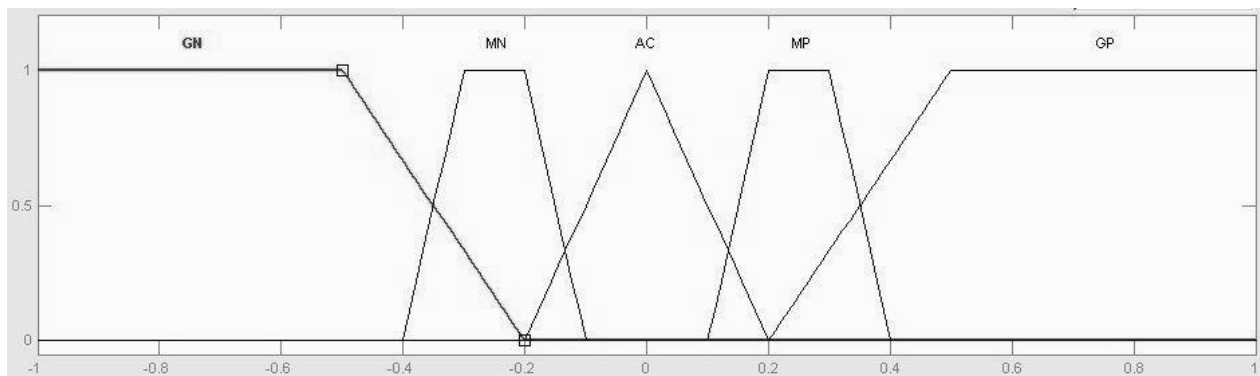
6.2.1 Entradas

Error

El error es la diferencia entre la temperatura del reactor de fermentación y el valor que se desea que tenga el tanque (set point).

Designación	Descripción	Rango	Función de pertenencia
GP	Grande positivo	$e > 0.2^{\circ}\text{C}$	Trapezoide(0.2,0.5,inf)
MP	Mediano positivo	$0.1^{\circ}\text{C} < e < 0.4^{\circ}\text{C}$	Trapezoide(0.1,0.2,0.3,0.4)
AC	Aceptable	$-0.2^{\circ}\text{C} < e < 0.2^{\circ}\text{C}$	Triangular(-0.2,0,0.2)
MN	Mediano negativo	$-0.1^{\circ}\text{C} > e > -0.4^{\circ}\text{C}$	Trapezoide(-0.4,-0.3,-0.2,-0.1)
GN	Grande negativo	$e < -0.2^{\circ}\text{C}$	Trapezoide(-inf,-0.5, -0.2)

Tabla IV-1: Funciones de membresía para el error

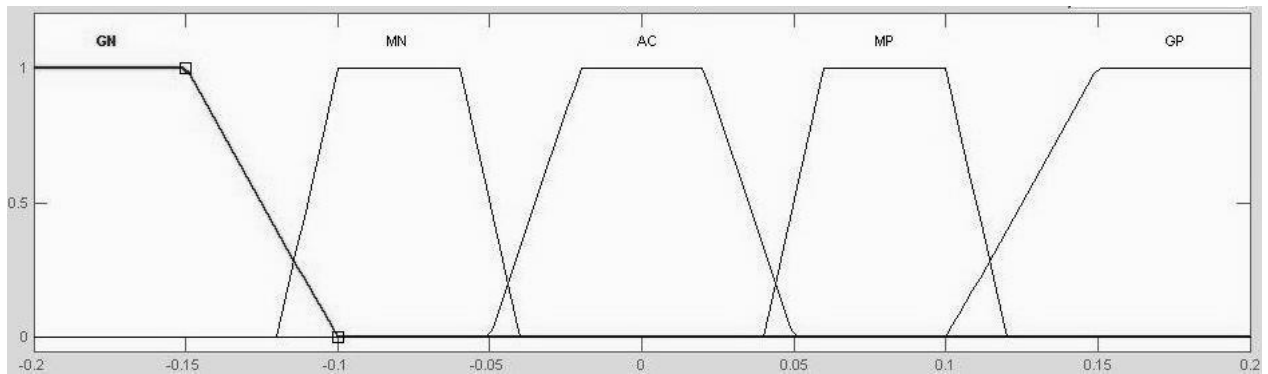


Tasa de cambio de la temperatura

Con esto se pretende, aprovechando que es un proceso lento, predecir el comportamiento de la temperatura.

Designación	Descripción	Rango	Función de pertenencia
GP	Grande positivo	$dT > 0.1^{\circ}\text{C}/\text{m}$	Trapezoide(0.1,0.15, inf)
MP	Mediano positivo	$0.04^{\circ}\text{C}/\text{m} < dT < 0.12^{\circ}\text{C}/\text{m}$	Trapezoide(0.04,0.06,0.1,0.12)
AC	Aceptable	$0.05^{\circ}\text{C}/\text{m} > dT > -0.05^{\circ}\text{C}/\text{m}$	Trapezoide(-0.05,0.02,0.02,0.05)
MN	Mediano negativo	$-0.04^{\circ}\text{C}/\text{m} > dT > -0.12^{\circ}\text{C}/\text{m}$	Trapezoide(-0.12,-0.1,-0.06,-0.04)
GN	Grande negativo	$dT < -0.1^{\circ}\text{C}/\text{m}$	Trapezoide(-inf,-0.15, -0.1)

Tabla IV-2: Funciones de membresía para dT/dt



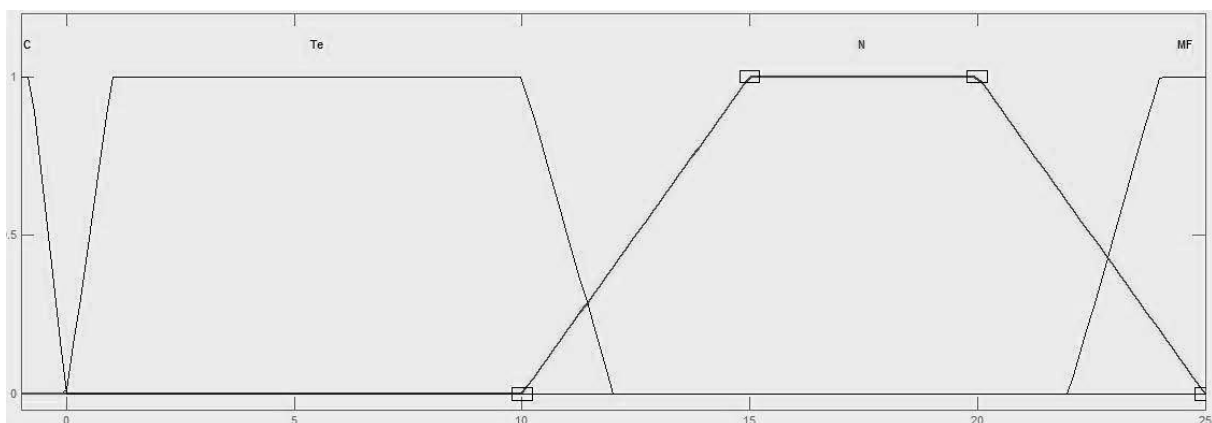
Refrigerante

El condicionante principal de este proceso es la temperatura del refrigerante en relación con el setpoint. Es decir una temperatura de refrigerante de 10°C puede resultar alta para un setpoint de 5 °C, pero totalmente funcional para un setpoint de 20°C o incluso 11°C.

Entonces $TR = \text{Setpoint} - \text{Temperatura refrigerante}$

Designación	Descripción	Rango	Función de pertenencia
C	Caliente	$TR < 0^{\circ}\text{C}$	Trapezoide(-inf, -1, 0)
N	Templado	$0^{\circ}\text{C} < TR < 12^{\circ}\text{C}$	Trapezoide(0,1,10,12)
F	Normal	$10^{\circ}\text{C} < TR < 25^{\circ}\text{C}$	Trapezoide(10, 15, 20, 25)
MF	Muy frio	$TR > 20^{\circ}\text{C}$	Trapezoide(22, 24, inf)

Tabla IV-3: Funciones de membrecía para el refrigerante



6.2.2 Salida

Teniendo en cuenta que la salida es del tipo Encendido-Apagado

Designación	Descripción	Rango	Función de pertenencia
A	Abierto	$E > 0.4$	Trapezoide(0.4,0.6,inf)
C	Cerrado	$E < 0.6$	Trapezoide(0.6,0.4,-inf)

Los rangos utilizados para cada una de las entradas y la salida fueron establecidos según diferentes métodos y criterios. Por ejemplo para definir el rango para el error se tuvo en cuenta la magnitud del error de medición de los termistores NTC (0.1°C). Para la derivada de la temperatura se analizaron los datos recolectados en la primer muestra y en base a las distintas pendientes de crecimiento de las diferentes etapas se determinó cuando se podría considerar grande o no.

6.2.3 Estructuras para las variables

Estas estructuras serán las encargadas de relacionar cada entrada con su función de membrecía y al mismo tiempo las entradas entre sí.

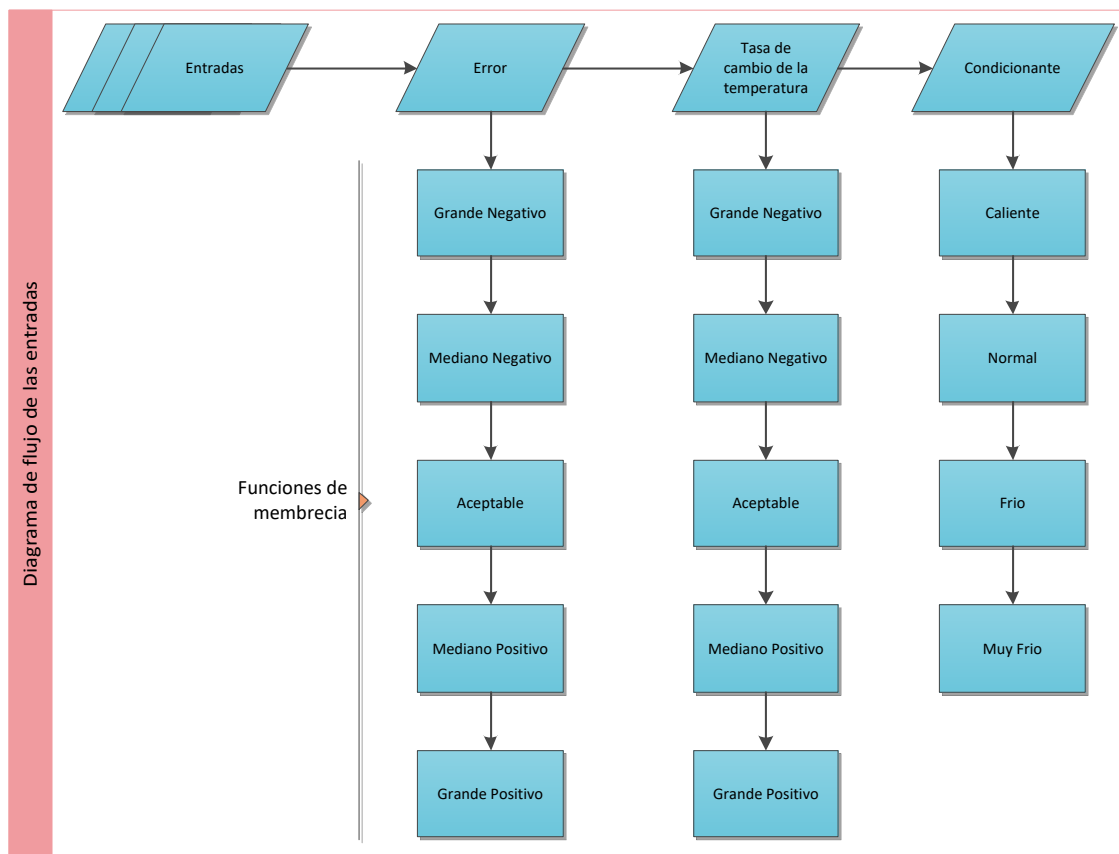


Figura IV-7: Diagrama de flujo de las entradas

En la figura IV-7 se muestra la estructura de interrelación de las entradas y sus funciones de membresía. La salida posee una estructura similar pero con un solo nodo y dos funciones de membresía asociadas.

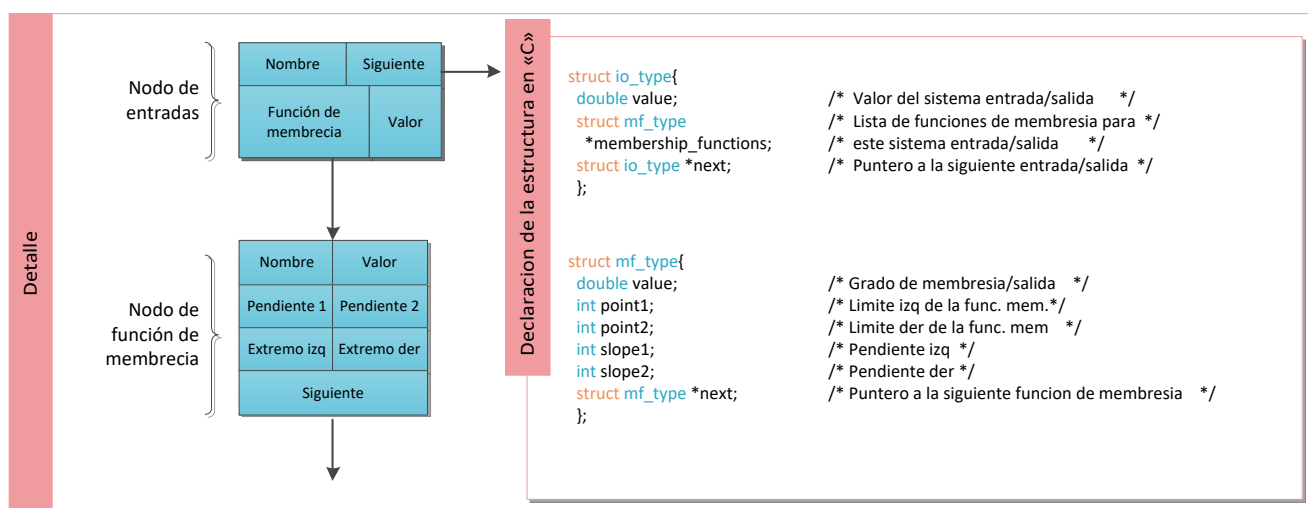


Figura IV-8: Detalle de las estructuras de datos

En el detalle podemos observar que el nodo "entradas" posee cuatro campos. De estos el único que se modifica es el de valor, el cual representa el valor cuantitativo de la variable de entrada. En tanto en el nodo de función de membresía el campo "valor" no es una cantidad medible, sino que calculable en función de la entrada. A esto se llama grado de pertenencia.

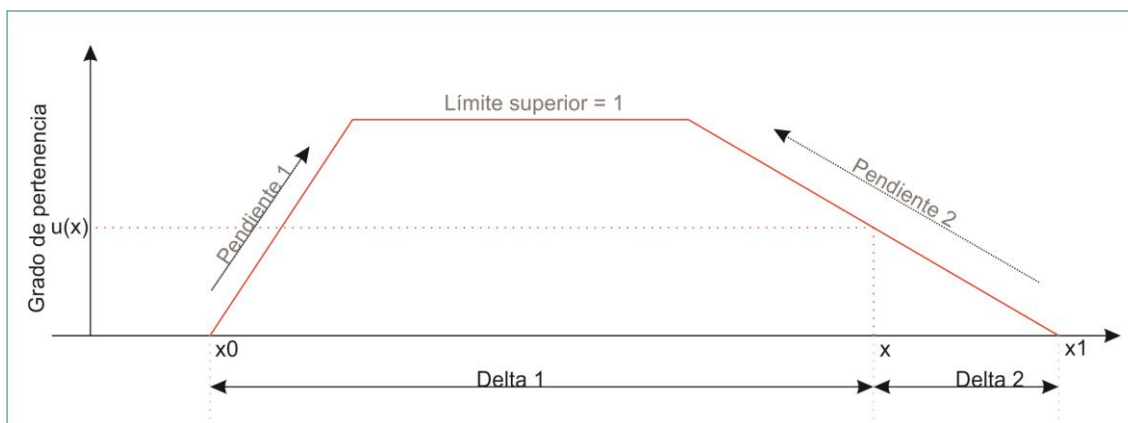


Figura IV-9: Función de membrecía

El procedimiento algebraico para resolver "el grado de pertenencia" considerando que $\Delta_1 = x - x_0$ y $\Delta_2 = x_1 - x$ es el siguiente:

1. Si tanto Δ_1 , como Δ_2 son menores o iguales a cero, significa que el valor no está comprendido dentro de la curva. Es decir, grado de pertenencia = 0.
2. Si no grado de pertenencia = $\min(\Delta_1 * \text{pendiente}_1, \Delta_2 * \text{pendiente}_2, \text{límite superior})$

6.3. Reglas

Como ya se explicó antes, las reglas son la base de la traducción del conocimiento humano a una expresión matemática. En base al previo conocimiento de que se trata de un proceso lento sin cambios abruptos y a la investigación y modelo matemático generado de la planta es que se establecen las siguientes reglas de control.

R	SI	Error	dT	Condicionante	ENTONCES	Salida
1	SI	-	-	Caliente	ENTONCES	Cerrado
2	SI	Grande Negativo	-	-	ENTONCES	Cerrado
3	SI	Mediano Negativo	Grande Negativo	-	ENTONCES	Cerrado
4	SI	Mediano Negativo	Mediano Negativo	-	ENTONCES	Cerrado
5	SI	Mediano Negativo	Aceptable	-	ENTONCES	Cerrado
6	SI	Mediano Negativo	Mediano Positivo	-	ENTONCES	Cerrado
7	SI	Mediano Negativo	Grande Positivo	Muy Frio	ENTONCES	Abierto
8	SI	Mediano Negativo	Grande Positivo	Templado	ENTONCES	Abierto
9	SI	Mediano Negativo	Grande Positivo	Normal	ENTONCES	Abierto
10	SI	Aceptable	Grande Negativo	-	ENTONCES	Cerrado
11	SI	Aceptable	Mediano Negativo	-	ENTONCES	Cerrado
12	SI	Aceptable	Aceptable	-	ENTONCES	Cerrado
13	SI	Aceptable	Mediano Positivo	Normal	ENTONCES	Abierto
14	SI	Aceptable	Mediano Positivo	Templado	ENTONCES	Abierto
15	SI	Aceptable	Mediano Positivo	Muy Frio	ENTONCES	Cerrado
16	SI	Aceptable	Grande Positivo	Normal	ENTONCES	Abierto
17	SI	Aceptable	Grande Positivo	Templado	ENTONCES	Abierto
18	SI	Aceptable	Grande Positivo	Muy Frio	ENTONCES	Abierto
19	SI	Mediano Positivo	Grande Negativo	-	ENTONCES	Cerrado
20	SI	Mediano Positivo	Mediano Negativo	-	ENTONCES	Cerrado
21	SI	Mediano Positivo	Aceptable	Normal	ENTONCES	Abierto
22	SI	Mediano Positivo	Aceptable	Templado	ENTONCES	Cerrado
23	SI	Mediano Positivo	Aceptable	Muy Frio	ENTONCES	Cerrado
24	SI	Mediano Positivo	Mediano Positivo	Normal	ENTONCES	Abierto
25	SI	Mediano Positivo	Mediano Positivo	Templado	ENTONCES	Abierto

26	SI	Mediano Positivo	Mediano Positivo	Muy Frio	ENTONCES	Cerrado
27	SI	Mediano Positivo	Grande Positivo	Normal	ENTONCES	Abierto
28	SI	Mediano Positivo	Grande Positivo	Templado	ENTONCES	Abierto
29	SI	Mediano Positivo	Grande Positivo	Muy Frio	ENTONCES	Abierto
30	SI	Grande Positivo	Grande Negativo	-	ENTONCES	Cerrado
31	SI	Grande Positivo	Mediano Negativo	-	ENTONCES	Cerrado
32	SI	Grande Positivo	Aceptable	Normal	ENTONCES	Abierto
33	SI	Grande Positivo	Aceptable	Templado	ENTONCES	Abierto
34	SI	Grande Positivo	Aceptable	Muy Frio	ENTONCES	Abierto
35	SI	Grande Positivo	Mediano Positivo	Normal	ENTONCES	Abierto
36	SI	Grande Positivo	Mediano Positivo	Templado	ENTONCES	Abierto
37	SI	Grande Positivo	Mediano Positivo	Muy Frio	ENTONCES	Abierto
38	SI	Grande Positivo	Grande Positivo	Normal	ENTONCES	Abierto
39	SI	Grande Positivo	Grande Positivo	Templado	ENTONCES	Abierto
40	SI	Grande Positivo	Grande Positivo	Muy Frio	ENTONCES	Abierto

Tabla IV-4: Reglas SI-ENTONCES

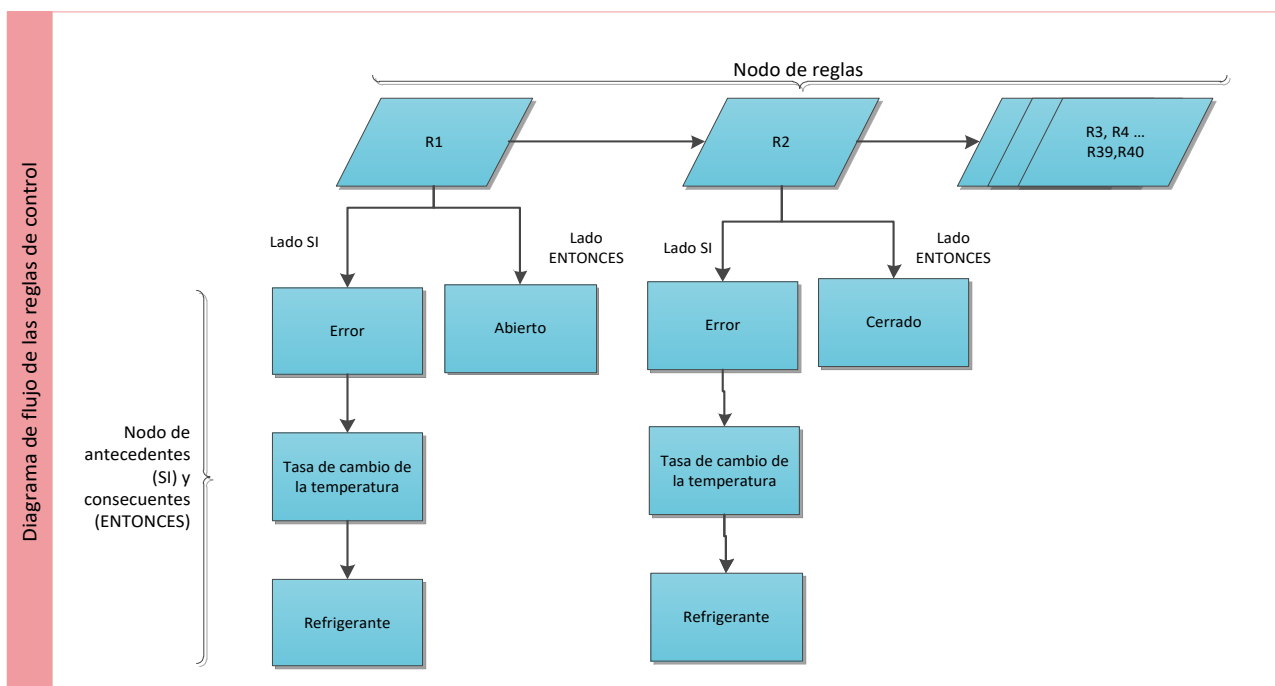


Figura IV-10- Estructura de las reglas de control

La programación del sistema de reglas también necesita de una estructura de datos que permita entrelazar los distintos nodos.

Estructuras de datos

```

struct rule_element_type{
    double *value;                /* Puntero a antecedente/salida */
    struct rule_element_type *next; /* Siguiete antecedente o salida */
};

struct rule_type{
    struct rule_element_type *if_side; /* Lista de antecedentes */
    struct rule_element_type *then_side; /* Lista de salida (consecuentes) */
    struct rule_type *next; /* Siguiete regla */
};

```

Figura IV-11: Estructuras para las reglas

6.4. Funciones

A continuación se listan todas las funciones de la librería.

Funciones		Descripción
Tipo	Descriptor	
void	fuzzy_init()	Inicializa las estructuras de datos necesarias para ejecutar la librería.
io_type*	new_io()	Inicializa una lista a la entrada/salida.
void	add_io(io_type **raiz, int valor, mf_type *lista_funciones_membrecia)	Agrega a la lista de entrada/salida una entrada/salida nueva.
mf_type*	new_mf()	Inicializa una lista de funciones de membrecía.
void	add_mf(mf_type **raiz, int valor, int punto1, int punto2, float pendiente1, float pendiente2)	Agrega a la lista de funciones de membrecía un nuevo elemento.
rule_type*	new_rule();	Inicializa una lista de reglas de control.
void	add_rule(rule_type **raiz, rule_element_type *lista_lado_si, rule_element_type *lista_lado_entonces)	Agrega a la lista de reglas de control, una nueva regla.
rule_element_type*	new_rule_element()	Inicializa una lista para lado si/entonces de las reglas.
void	add_rule_element(rule_element_type **raiz, int *valor)	Agrega un elemento a la lista de lado si/entonces
void	fuzzification(struct io_type *System_Inputs)	Calcula los valores de fuzzificación para cada función de membrecía.
void	rule_evaluation(struct rule_type *rules)	Evalúa las reglas de control.
void	defuzzification(struct io_type *System_Outputs)	Calcula los valores certeros de salida a partir de los valores difusos.
void	compute_degree_of_membership(struct mf_type *mf, int input)	Calcula el grado de pertenencia de una entrada a una función de membrecía.
int	compute_area_of_trapezoid(struct mf_type *mf)	Calcula el área del trapecoide para cada función de membrecía.
void	get_system_inputs(float dato[NUM_ENTRADAS], struct io_type *si)	Guarda en la lista de entradas lo valores para fuzzificar.
int	put_system_outputs(struct io_type *so)	Traduce los valores certeros de salida en un único valor para ser utilizado.
int	fuzzy_control(float dato[NUM_ENTRADAS], struct io_type *entrada, struct rule_type *reglas, struct io_type *salida)	Retorna la salida del sistema de control a determinadas entradas y reglas. Solo es válido para sistemas MISO.

Tabla IV-5: Funciones de la librería de Fuzzy Logic

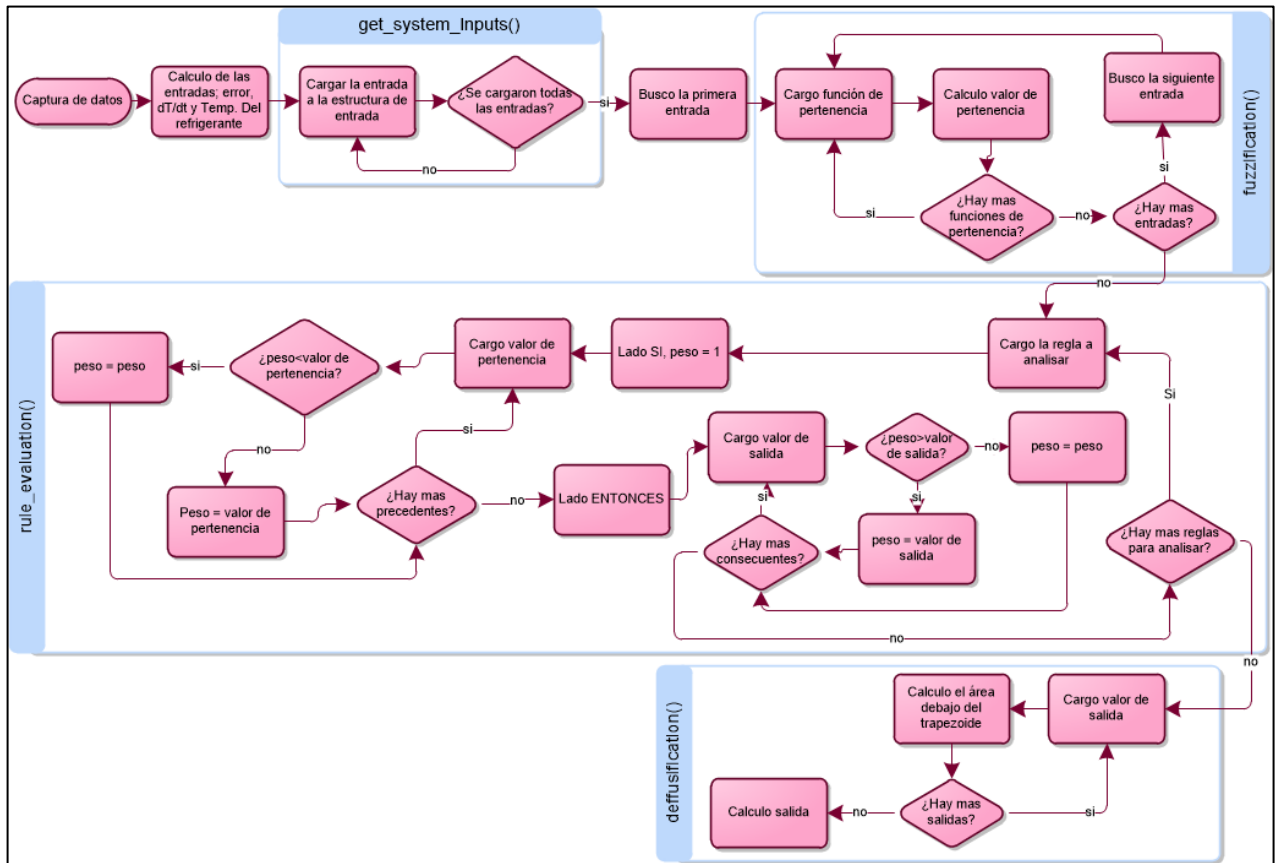


Figura IV-12: Diagrama de flujo del controlador

V - Mediciones y resultados

El siguiente capítulo tiene como objetivo general mostrar el procedimiento que se llevó a cabo para verificar el funcionamiento del CdT en la fábrica. Antes de poder instalarlo y que efectivamente funcione controlando el reactor de fermentación es necesario asegurar el correcto desempeño e identificar posibles escenarios en los que el controlador pueda fallar.

1. Simulaciones

En esta primera etapa se retomará el modelo matemático diseñado en el capítulo III al cual se le agregará el controlador diseñado en el capítulo IV.

En base al conjunto controlador-planta se crean dos sistemas, el primero cuyo controlador será implementado en el mismo software utilizado para realizar el modelo y el segundo cuyo controlador estará implementado en hardware, a partir de ahora llamaremos a estos sistemas "**Modelo con controlador**" y "**Modelo con CdT**" respectivamente. De esta forma al comparar la respuesta de ambos sistemas se comprobará que la implementación en hardware es fiel a lo diseñado en software, a continuación se pueden observar los diagramas de los sistemas a comparar.

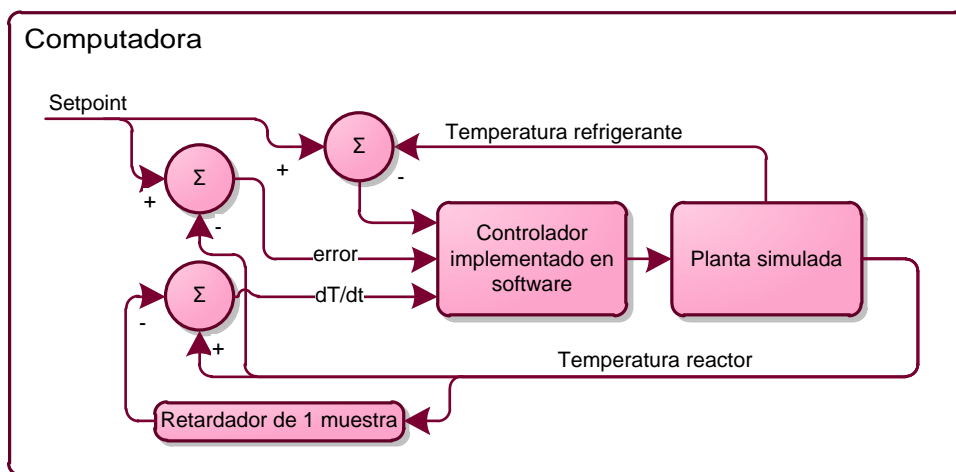


Figura V-1: Estructura del sistema con el controlador en software (Modelo con controlador).

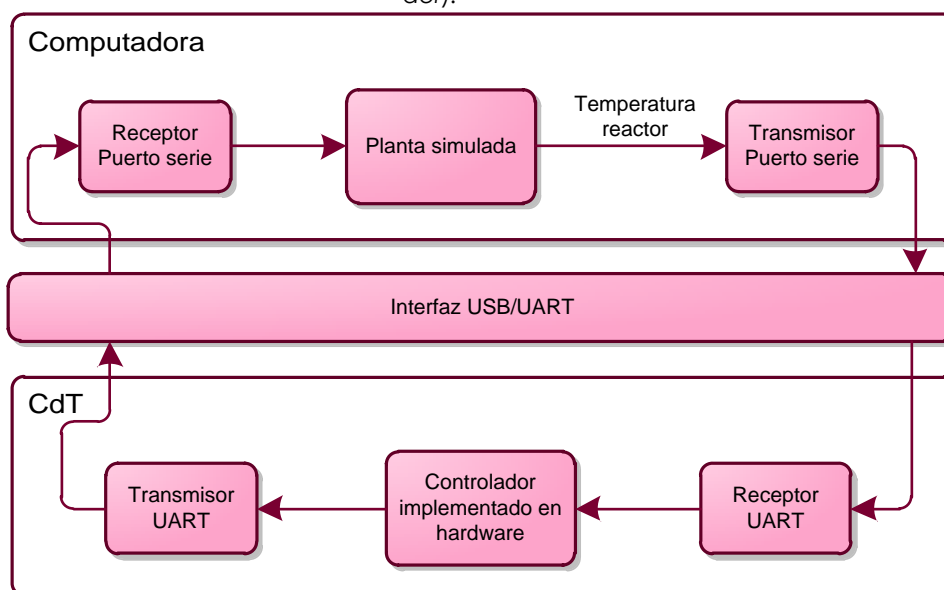


Figura V-2: Estructura del sistema con el controlador en hardware (Modelo con CdT).

Luego de realizar la simulación en ambos sistemas con los parámetros obtenidos en la sección III.2.5.2, se llegó a los siguientes resultados.

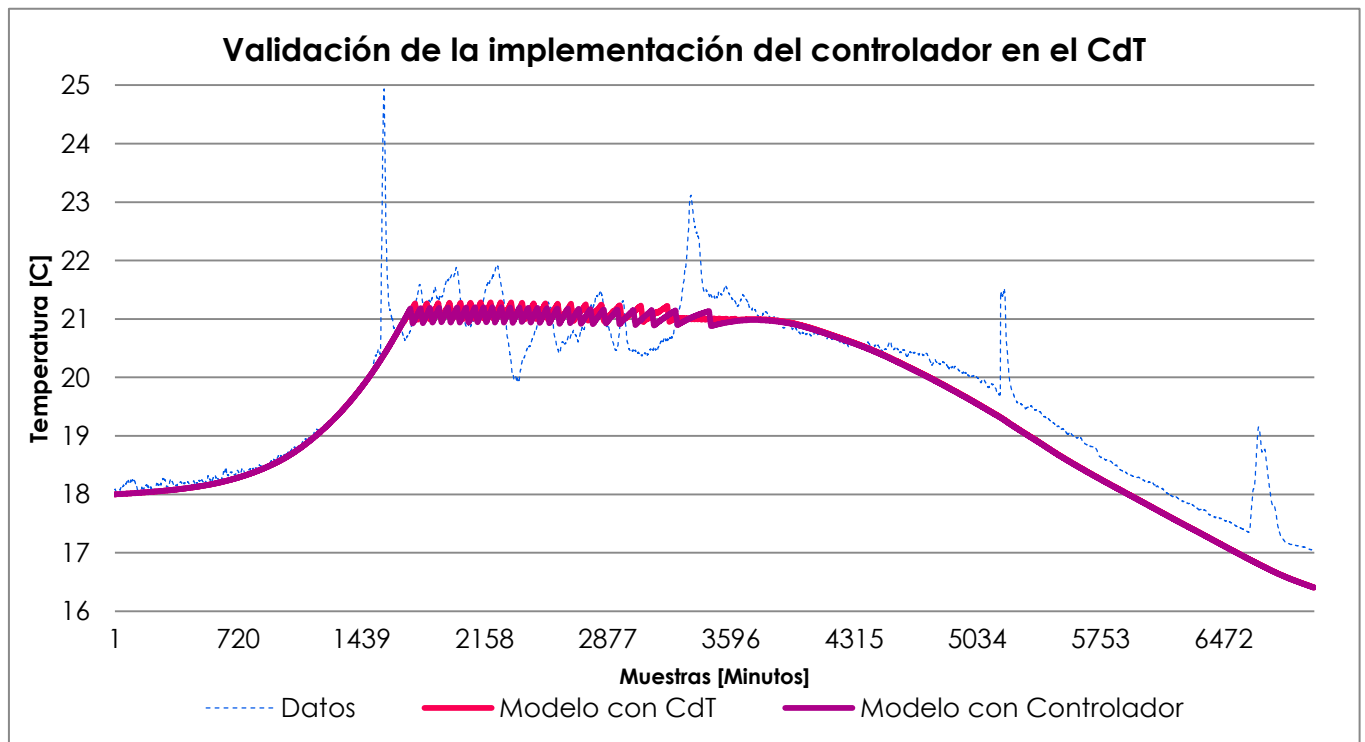


Figura V-3: Comparación del modelo con controlador vs modelo con CdT

Tal como se observa, en la figura VIII-3, la curva "Modelo con controlador" y la curva "Modelo con CdT" mantienen un alto grado de similitud por lo que se puede avanzar sobre la definición de cuales variables modificar para simular diferentes escenarios.

Los parámetros del modelo pueden ser divididos en 3 grupos:

Grupo 1: Son aquellos parámetros que hacen al tipo de levadura que se utiliza.

Grupo 2: Estos definen las condiciones iniciales de la fermentación.

Grupo 3: Definen las características del sistema de refrigeración.

A continuación se muestra una tabla explicativa de todos los parámetros del modelo, divididos en los grupos antes descriptos.

Grupo	Parámetro	Descripción	Efecto de su modificación
1	Kx	Consumo de azúcares	Es una medida de la cantidad de azúcar que consume una célula de levadura, es el limitante principal de la población máxima de levadura y por lo tanto limita el pico máximo de la temperatura del fermentador.
	Bx	Constante de mortalidad	Modifica la tasa de muertes por alcohol, su variación modifica la población máxima de levadura y la duración del proceso de fermentación.
	Ax	Constante de producción de alcohol	Los alcoholes son los principales causante de la muerte de células de levadura.
	P	Población inicial de levaduras	Hace a las veces de delay del proceso de fermentación, si se disminuye el proceso tarda más en comenzar.

	Kt	Constante de producción de calor de la fermentación	Modifica la cantidad de calor que puede generar el proceso de fermentación.
2	A1	Cantidad de azúcar inicial	Junto con el índice de mortalidad por alcohol es el limitante principal de la población máxima que puede alcanzar la levadura y por lo tanto la máxima cantidad de calor que se libera en el fermentador.
	T0	Temperatura inicial	La temperatura está relacionada con el índice de reproducción de la levadura, por lo que aumentar la temperatura inicial acelera el inicio del proceso de fermentación.
3	Ct	Constante de conductividad del reactor al ambiente	Modifica la cantidad de calor transferido desde el interior del fermentador al ambiente, este dependerá del material de construcción del fermentador y su forma principalmente.
	Tc	Constante de conductividad del dispositivo de enfriamiento al reactor	Limita la cantidad de calor que puede remover el sistema de refrigeración del fermentador.
	Tcooler	Temperatura del refrigerante	Cuanto menor es la temperatura del refrigerante mayor es la cantidad de calor que el sistema de refrigeración puede remover del fermentador.

Tabla V-1: Parámetros de simulación

Las variables que influyen al desempeño del proceso de fermentación son:

- *El tipo de levadura:*

Diferentes tipos de levadura pueden poseer mayor o menor resistencia a la muerte causada por alcoholes, pueden producir más o menos calor, etc. Entonces es importante considerar variaciones a los parámetros del grupo 1.

- *La aislación térmica del tanque de fermentación:*

Si la aislación es baja, es decir que la temperatura ambiente gobierna por sobre la temperatura del fermentador puede producir dos escenarios extremos. El primero que la temperatura ambiente sea mucho menor que la temperatura ideal de fermentación, entonces la fermentación necesitaría mucho más tiempo para alcanzar la temperatura ideal (si es que la alcanza) y cada acción de control debería ser minuciosamente aplicada para no retrasar el proceso. Dicho de otra manera, si en una situación normal la acción de control en 1 minuto (mínimo tiempo de acción) produce una disminución de 1 grado de temperatura en una situación de aislación térmica reducida y temperatura ambiente fría, lo que sucede es que en vez de 1 grado, el gradiente sea mucho mayor generando así un escenario donde la fermentación nuevamente tiene que alcanzar la temperatura deseada. El segundo caso, sería que la temperatura ambiente sea superior a la de fermentación, es decir que el problema radica en la capacidad del sistema de refrigeración para quitar calor del reactor. Lo cual no es algo que concierne al controlador.

- *El sistema de enfriado:*

Esto es ajeno al controlador, en el sentido de las capacidades. Lo que el controlador si toma en cuenta para funcionar es la temperatura a la cual trabaja este sistema para prevenir irregularidades como que la temperatura del sistema sea mayor que la de fermentación y en vez de enfriar, caliente. Otro factor importante que puede afectar es el tiempo desde que se inicia la acción de control hasta que se detecta algún cambio y la inercia térmica que esto puede generar una vez finalizada la acción de control. En el desarrollo del

modelo del enfriador³ se tuvo en cuenta un retraso y está considerado no así en el diseño del controlador. Esto requiere que el controlador pueda identificar el retraso analizando el tiempo de respuesta dinámicamente cuestión que no se incluye dentro de los objetivos iniciales.

A modo de referencia, en primera instancia, se realiza la simulación del modelo con controlador, modelo con CdT y el modelo sin controlador utilizando los parámetros ideales.

Legenda de las simulaciones

- Tsim = Temperatura del modelo con el CdT.
- Tmod = Temperatura del modelo con controlador.
- Tplanta = Temperatura del modelo sin ningún control.
- Setpoint = Temperatura de referencia para el controlador.

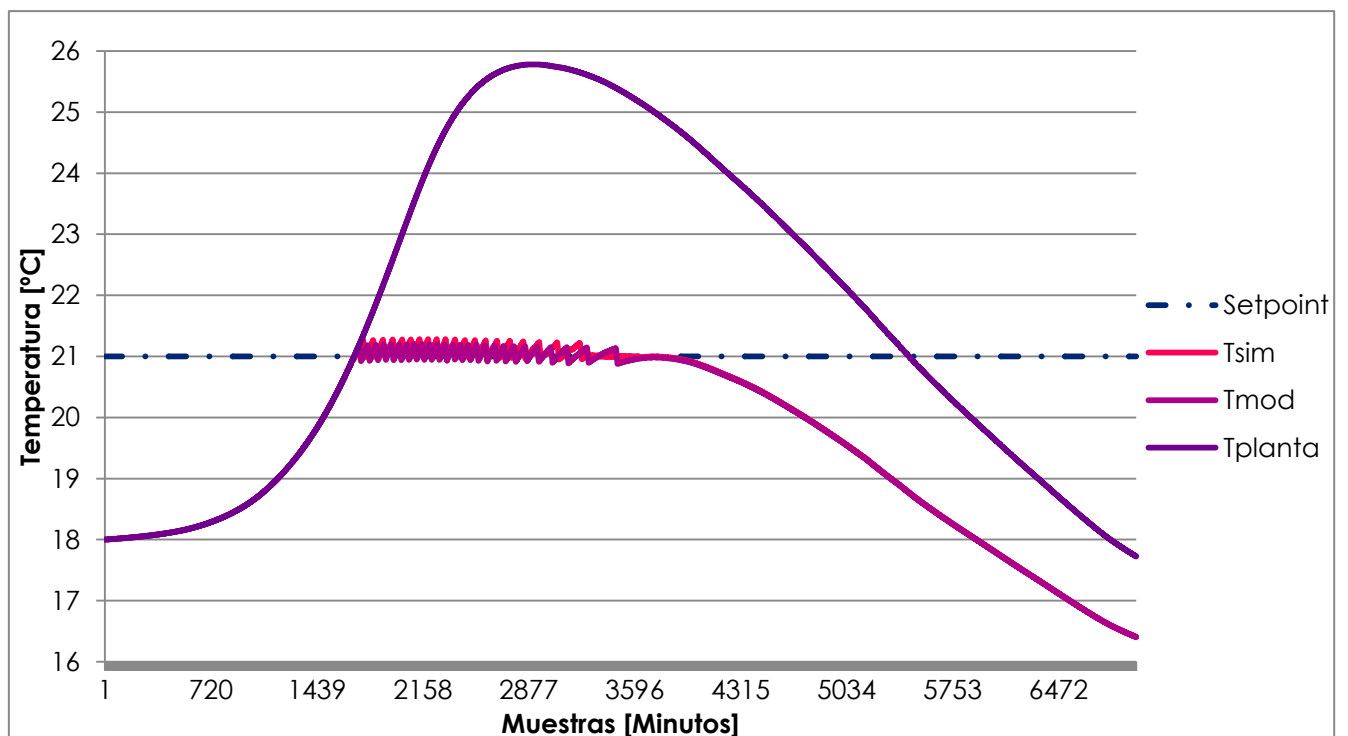


Figura V-4: Parámetros ideales.

El efecto de utilizar otra cepa de levadura se logra modificando la constante de producción de calor de la levadura.

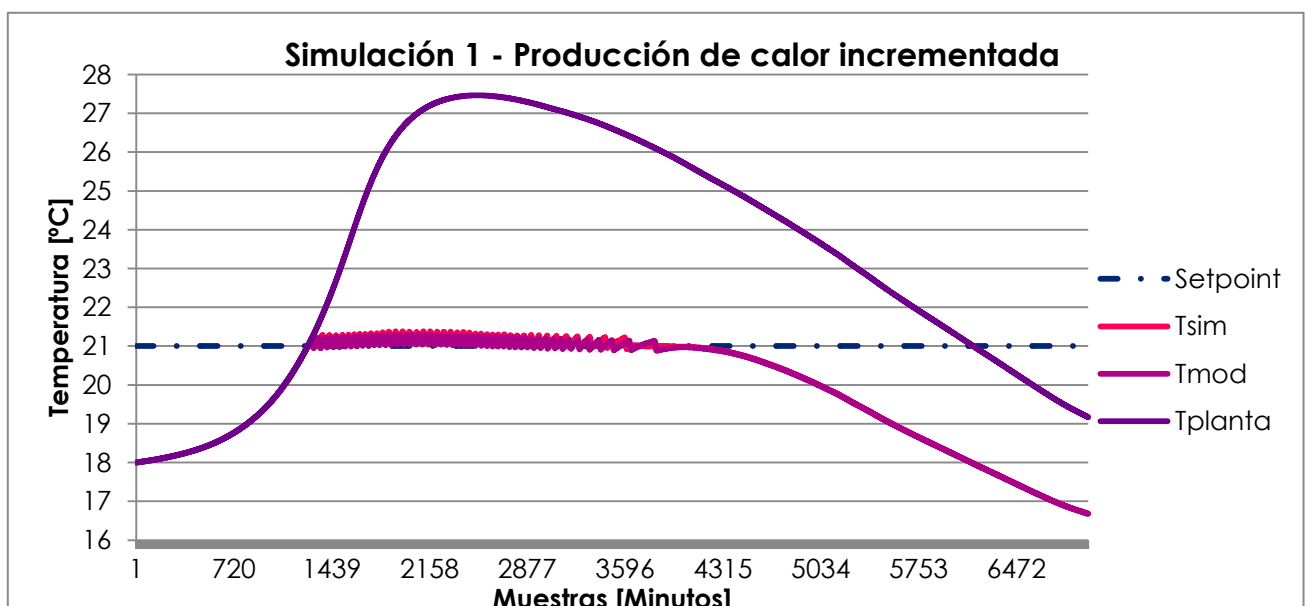


Figura V-5: Sim -Producción de calor incrementada,

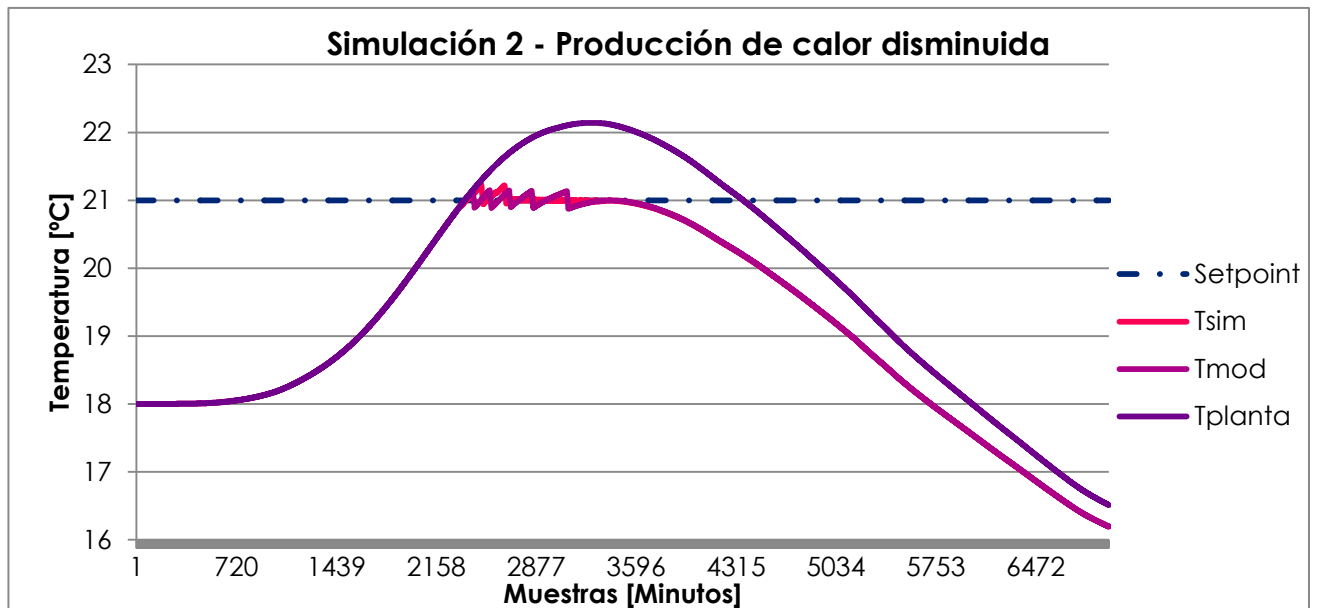


Figura V-6: Sim – Producción de calor disminuida

$$k_{t,levadura} = 0.0009$$

$$k_{t,levadura} = 0.000225$$

Tal como se explicó, la temperatura ambiente es un factor que hace al proceso de fermentación, se simula entonces dos extremos de temperatura. El extremo inferior de temperatura se elige tal que permita al proceso de fermentación alcanzar la temperatura ideal de 21°C. Este límite inferior, de 10°C, se encontró iterando las simulaciones hasta encontrar el resultado deseado. De límite superior se estable 35°C que resulta un valor suficiente para ser considerado como una media. Luego se generan variaciones con media de 10°C y 35°C. Se utilizan curvas para la temperatura ambiente que sinusoidalmente varían entre -5 y 5 grados centígrados alrededor de las temperaturas extremas.

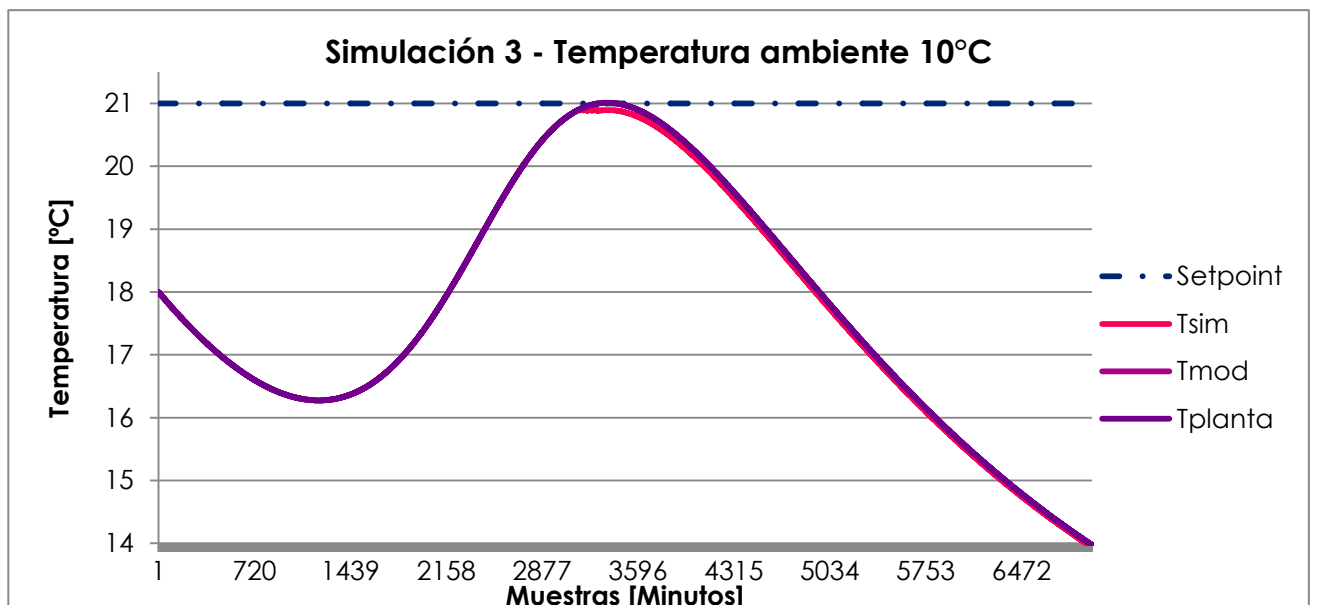
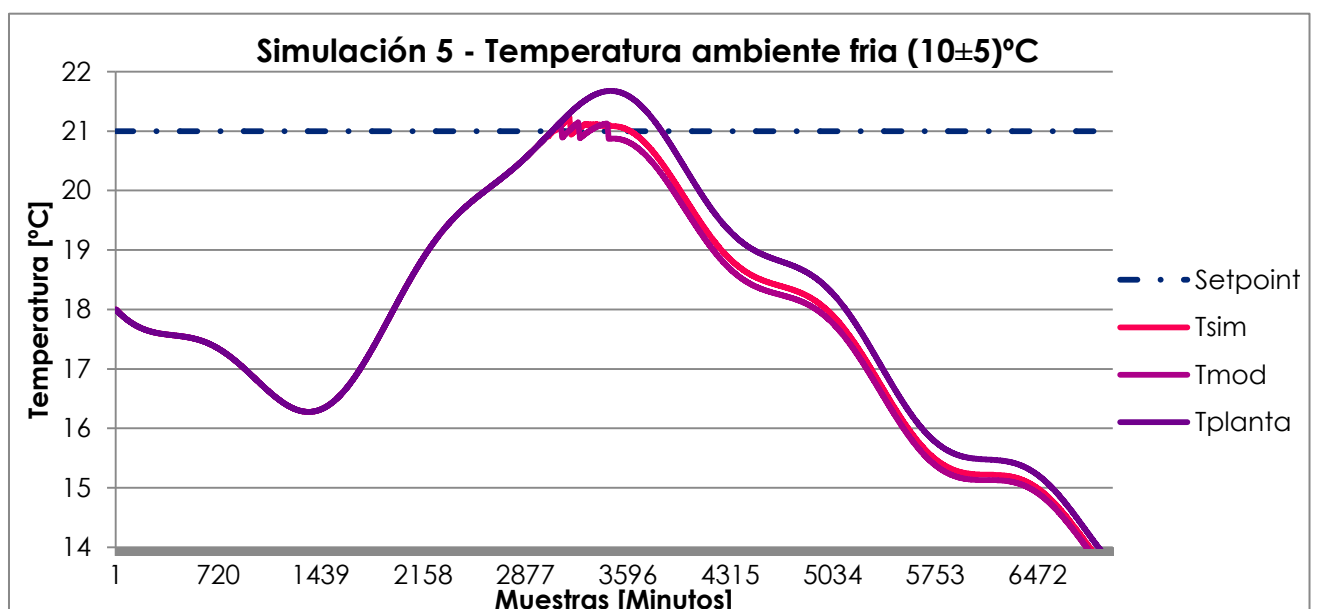
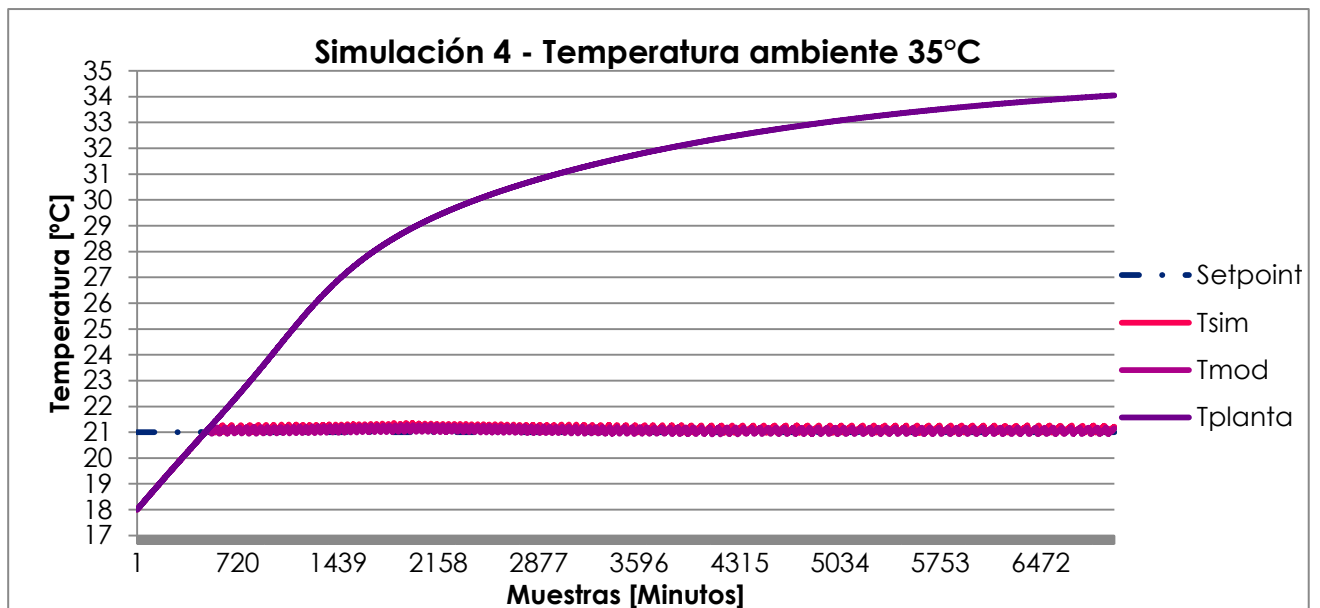


Figura V-7: Sim – Temperatura ambiente límite inferior

$$T_{ambiente} = 10$$



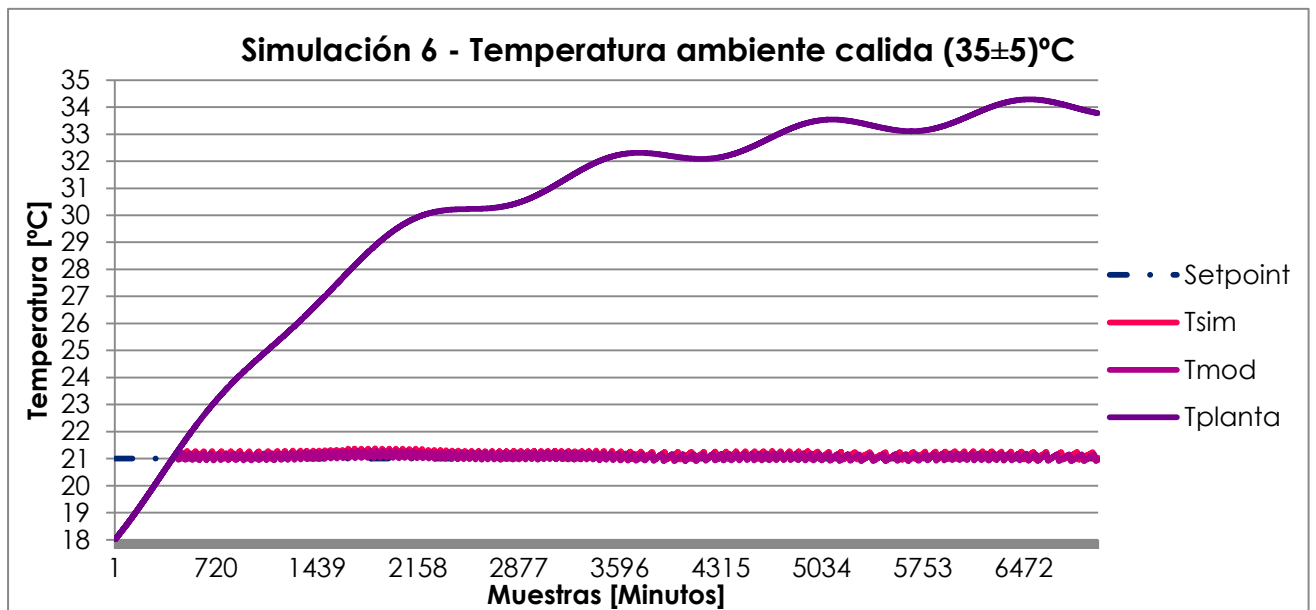


Figura V-10: Sim - Temperatura ambiente límite superior con variación

$$T_{\text{ambiente}} = 35 + 5 \sin\left(\frac{2\pi}{1440} t\right); [1440 \equiv 1 \text{ Día}]$$

Es necesario verificar que sucede en el escenario donde el líquido refrigerante tiene una temperatura muy inferior al setpoint. Necesariamente esta situación presenta una caída abrupta de la temperatura debido al gradiente de temperatura.

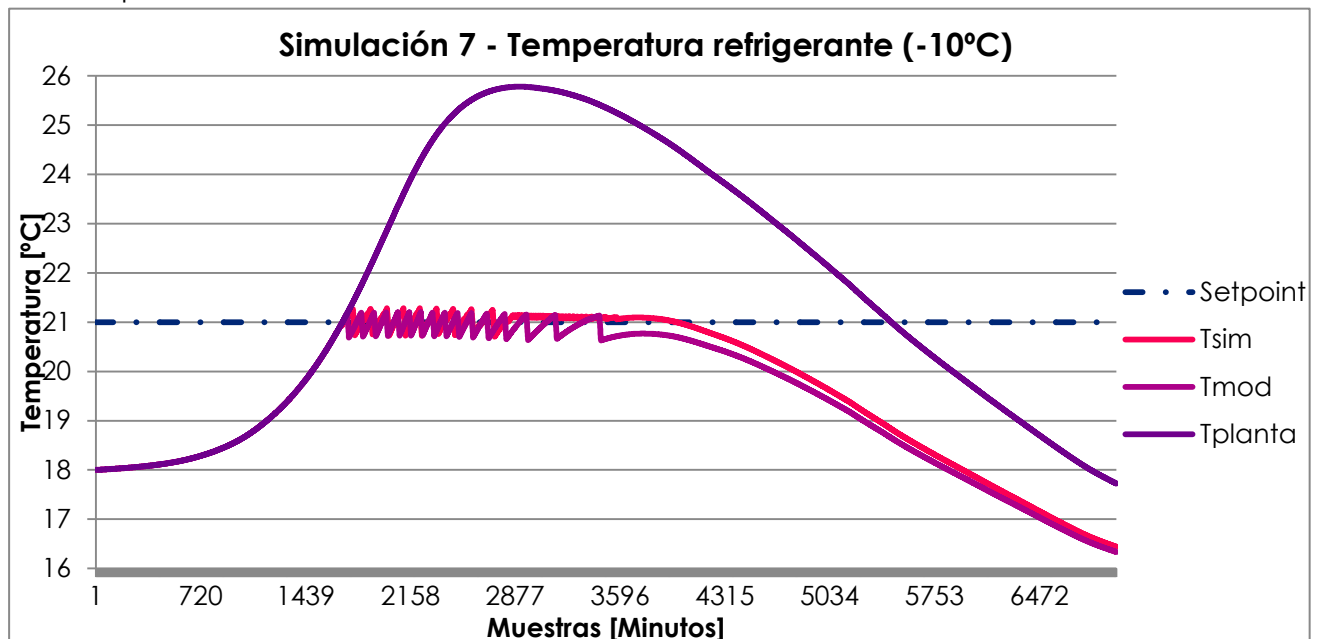


Figura V-11: Sim - Temperatura del refrigerante baja

$$T_c(t) = -10$$

Hasta ahora en cada una de simulaciones la media de la temperatura estuvo por debajo o por sobre la temperatura de setpoint, por lo que durante la totalidad del proceso aumentaba o disminuía la temperatura del fermentador. En esta última simulación se aumentó la conductividad al ambiente para simular una disminución en la aislación del fermentador y utilizando una temperatura media de 20°C con una amplitud de ±5°C se realizó la simulación con el fin de ver la respuesta del controlador a perturbaciones en la temperatura ambiente.

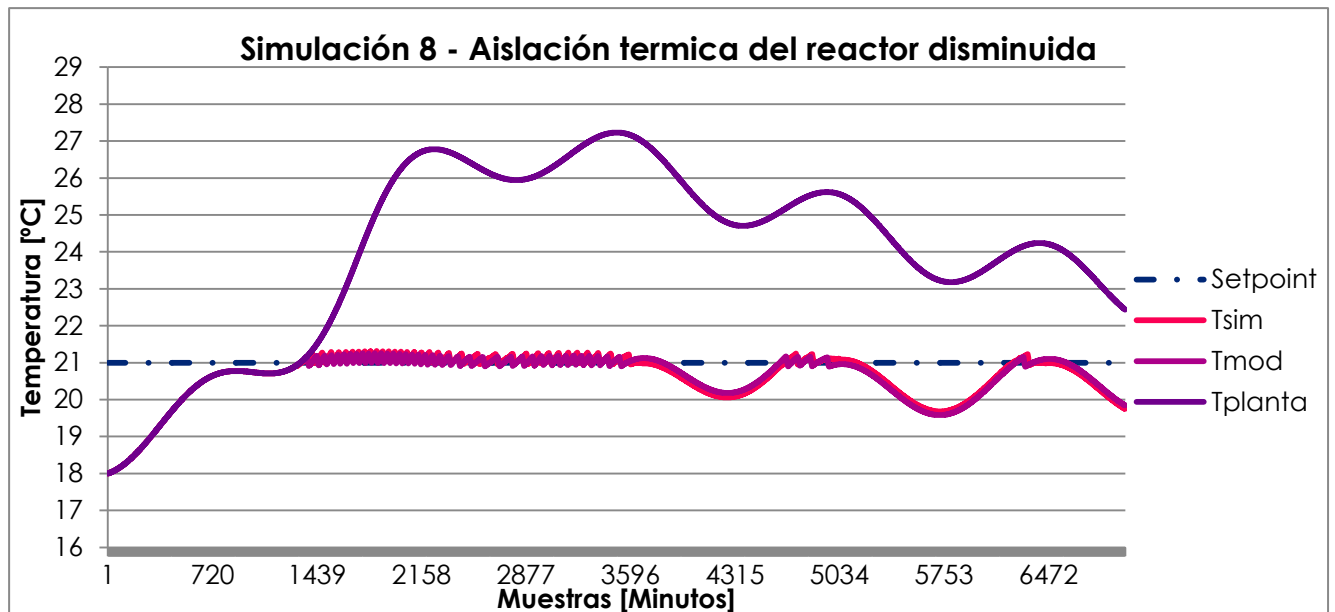


Figura V-12: Sim – Aislación termina del reactor disminuida

$$k_{t,ambiente} = 2.6$$

2. Errores

Se puede observar entonces que en todos los escenarios el modelo de la planta con el controlador funciona correctamente. Se analiza el error del modelo con CdT durante la etapa de fermentación, es decir solo durante el momento donde el controlador actúa. Tal como se observa, este no supera en la mayoría de los casos los 0.35°C.

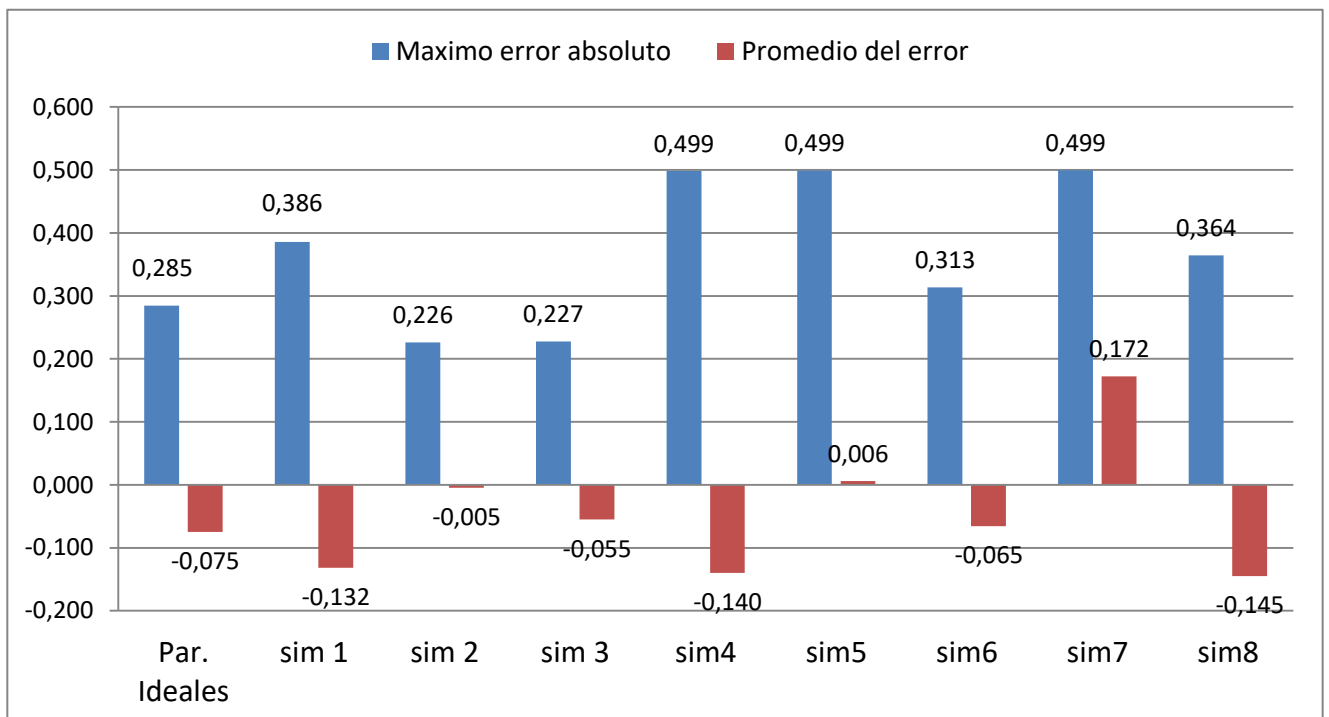


Figura V-13 Gráfica de los errores de simulación

Una variación de la temperatura dentro del rango de $\pm 1^\circ\text{C}$ es admisible para esta primera experiencia ya que este es el error de los controladores mencionados en el capítulo VII, presentes en el mercado argentino. Se puede

entonces, terminar de concluir que las variaciones, incluso de las simulaciones 4,5 y 7 (que alcanzan los 0.5°C) están dentro de lo aceptable.

Si analizamos el porqué de estas variaciones abruptas, podemos distinguir que son debido a que el modelo matemático en cuestión no se ajusta perfectamente a la realidad, la velocidad con la que responde y cambia la pendiente es demasiado rápida. Además, el modo en que se procesa la derivada de la temperatura en el CdT provoca que los valores resultantes estén siempre en los márgenes de saturación de las funciones de membrecía de la derivada del controlador. Esto es debido a que desde la computadora se envían datos entre 0 y 255 que luego son divididos en 10 en el controlador, dando valores entre 0 y 25,5. La resolución resultante es de 0.1°C, al calcular la derivada entre dos puntos⁴ esta, como mínimo, puede ser de 0.1. Si retomamos lo visto en el capítulo IV, el rango de la derivada está entre 0.2 y -0.2.

Las mediciones en la fábrica serán realizadas con un Termistor NTC de 10Kohm a 25°C. Una de las principales desventajas es que se tratan de transductores con muy baja repetitividad en la fabricación. Esto quiere decir que si bien existen hojas de datos donde se caracterizan las NTC, no son necesariamente correctas para todas. La parametrización ajusta la curva de forma precisa aunque esta presenta errores asociados al elemento de calibración (termómetro digital).

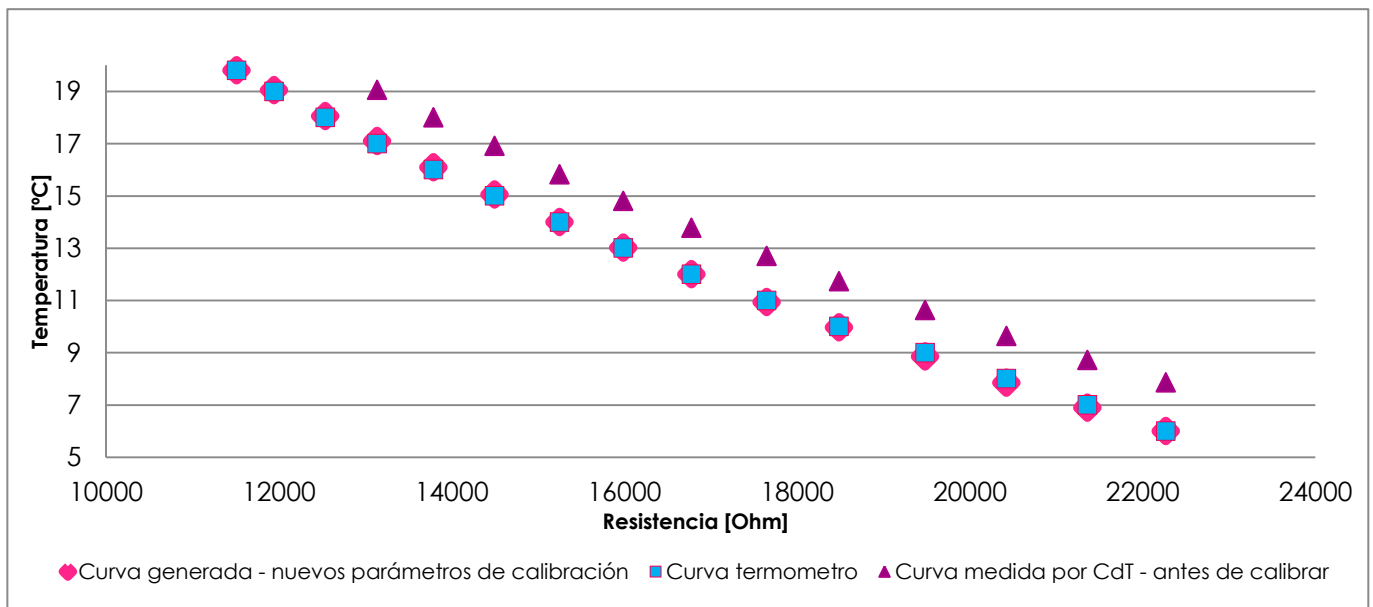


Figura V-14: Calibración del sensor

⁴ Derivada discreta: $f'[x_n] = \frac{f[x_n] - f[x_{n-1}]}{x_n - x_{n-1}} = \frac{f[x_n] - f[x_{n-1}]}{1}$

3. Procesamiento de los datos

Se utilizan tres etapas de pre procesamiento de los valores de temperatura para mitigar los errores debido a perturbaciones

1. Cada segundo un reloj del PIC habilita al ADC a muestrear el valor de resistencia del sensor, en ese momento el ADC realiza 5 mediciones consecutivas que luego se promedian para dar un valor final por segundo.
2. Cada 60 segundos, el CdT verifica si debe realizar una acción de control. Para ello toma las 60 muestras previas y las promedia para obtener un valor de temperatura por minuto.
3. Por último, el valor de temperatura obtenido por el segundo filtro es procesado una vez más para correlacionarlo con las muestras anteriores. Para ello se utiliza un filtro IIR que utiliza un valor anterior para calcular el actual.

$$Tsalida_k = \frac{Tentrada_k - Tsalida_{k-1}}{n} + Tsalida_{k-1}$$

El valor de n determina cuan similar son la entrada con respecto a la salida. Como se aprecia en la figura VIII-13 cuanto más grande el valor de n más se "suaviza" la curva, pero menos se asemeja a la curva original.

Se eligen para el ensayo en la fábrica, como parámetros de los filtros;

- Tomar 5 muestras por segundo y promediar.
- Tomar por minuto, es decir luego de haber registrado 60 muestras, un promedio de las mismas.
- Para el filtro IIR se utiliza un valor de n = 10.

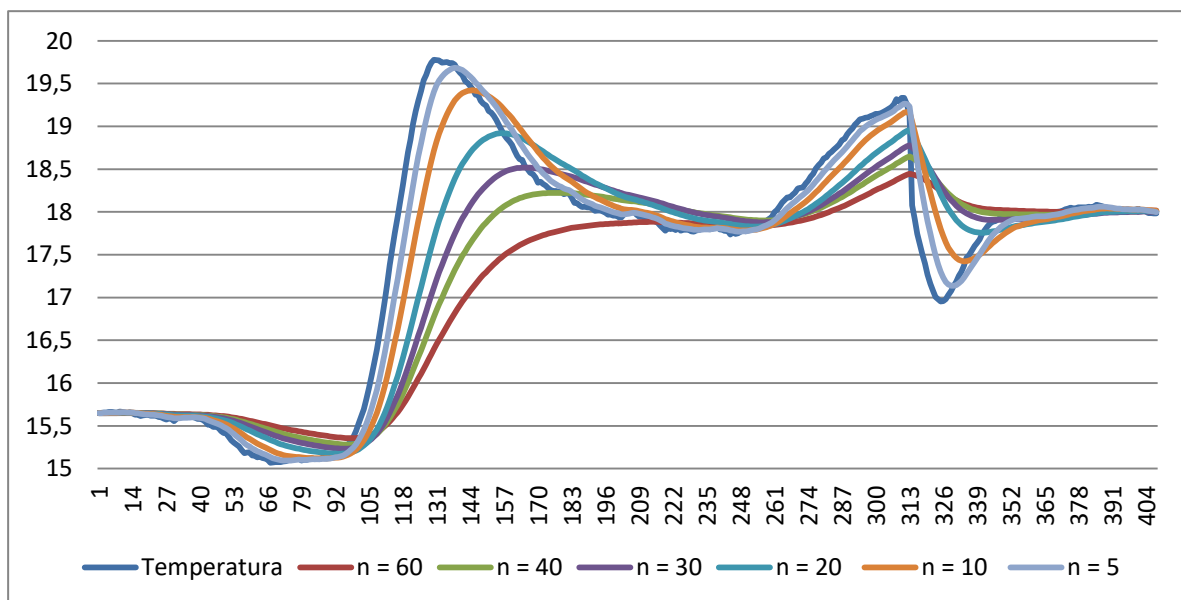


Figura V-15 : Calibración del filtro IIR

4. Ensayo en la fábrica

Primera experiencia

Se realiza en la fábrica de cerveza "La Maroma" ubicada en la localidad de El Bolsón, Rio Negro un ensayo del CdT en condiciones normales de funcionamiento de la fábrica.

El día 29 de mayo de 2019 realizan la cocción de una cerveza tipo ALE-IPA(500 l), la cual tiene un periodo estimado de fermentación de entre 5 y 6 días a una temperatura de 20°C, seguido de un periodo de maduración de 7 días a 15°C y luego 10 días a 4°C. Al momento del transvase de la olla de cocción al reactor de fermentación, la temperatura de la "cerveza" fue aproximadamente de 17,3°C y la temperatura ambiente de 11,2°C.

Se utiliza un sensor de temperatura NTC 10K a 25°C en la vaina del reactor diseñada a tal fin, además se agregan dos sensores extras, de características similares, para medir la temperatura ambiente.

Luego de 12 días en el tanque de fermentación se da por finalizada la prueba, el 11 de junio de 2019. Los operadores deciden cambiar de reactor de fermentación a la cerveza antes de bajar la temperatura, por lo que el último periodo de 10 días no pudo ser registrado. Esto es una operación común que se utiliza para retirar la levadura que decanta al fondo del fermentador, al trasvasarla clarifica la cerveza y evita además excesos de levadura en el producto final.

A continuación se muestra un esquema de cómo esta interconectado el sistema:

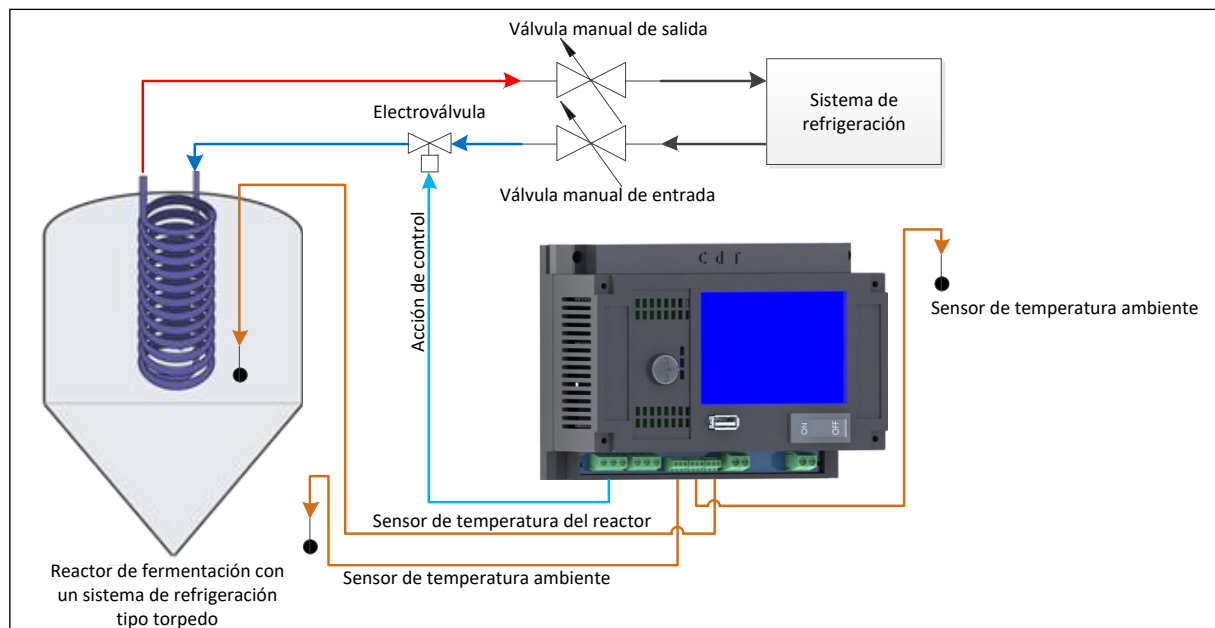


Figura V-16: Esquema de interconexión en la fábrica

Como medida de precaución las válvulas manuales de entrada y salida del torpedo se cierra durante el primer día de fermentación para que este se encuentre completamente vacío y que el líquido refrigerante no retrase la fermentación. Como la temperatura ambiente durante los siguientes días fue baja (alrededor de los 9°C) y las mediciones de temperatura que realizaron sobre el fermentador no mostraban un incremento en la temperatura, los operarios nunca abrieron las válvulas manuales.

A causa de esa decisión es que en la figura V-17 no se ve ninguna variación en la temperatura cuando se abre la electroválvula al tercer día (los operarios no visualizaron esto dado que ocurrió durante el fin de semana). De cualquier manera esta prueba nos permite validar el modelo, verificar que el filtro IIR funcione correctamente y ajustar la injerencia de la temperatura ambiente al proceso. Los sensores de temperatura ambiente estaban ubicados a distintas alturas, por este motivo es que se puede observar una diferencia de temperatura entre las dos mediciones.

El pico en la temperatura ambiente que ocurre en el día 6 fue ocasionado por el calor generado por los quemadores de las ollas, ya que como nos confirmaron los operarios 6 días después de la instalación de nuestro controlador realizaron la cocción de un nuevo lote de cerveza.

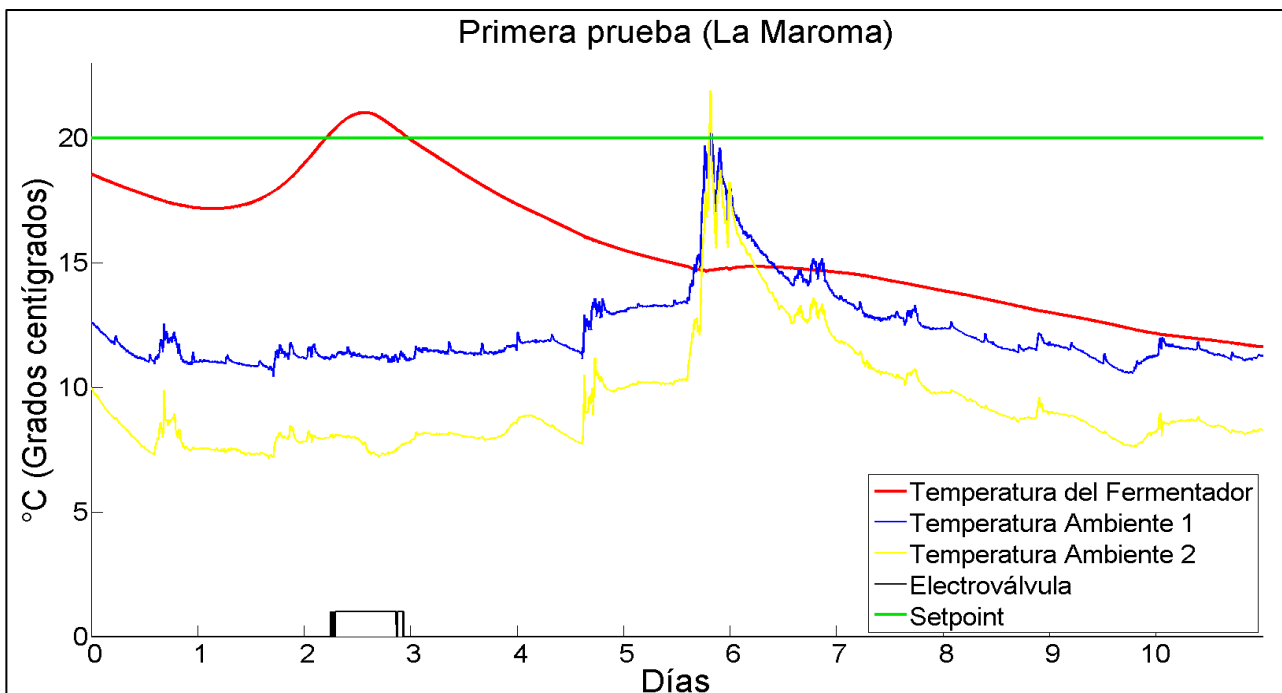


Figura V-18: Datos de la prueba de campo

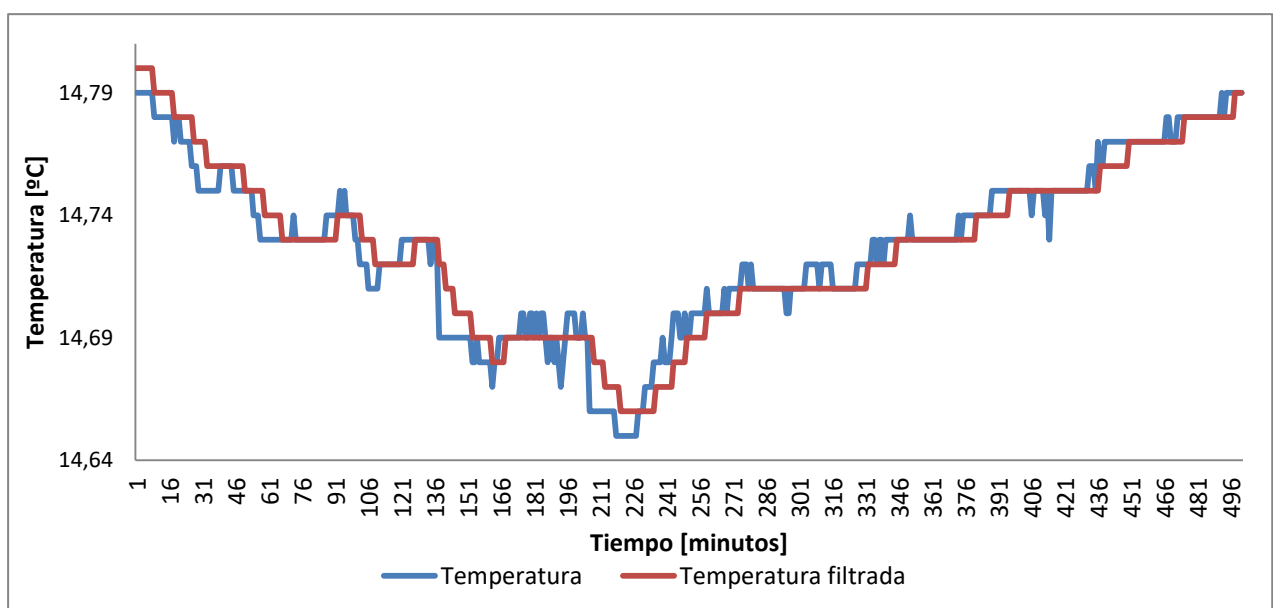


Figura V-17: Filtro IIR en los datos reales

Con respecto al filtro podemos observar como hace lo que se esperaba, suavizando correctamente las muestras, eliminando variaciones indeseadas.

Configuración utilizada y cálculo de error

Dado el sensor elegido para la experiencia, la configuración de la etapa de adquisición se realizó para sensores de dos cables, en la Figura V-19 se puede observar un diagrama de esta configuración.

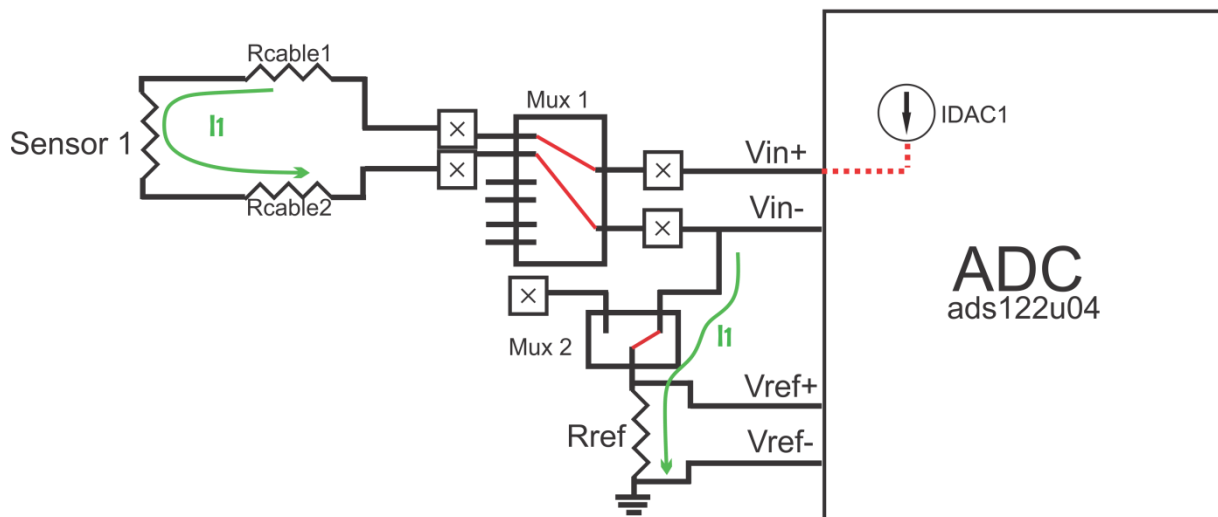


Figura V-19: Configuración utilizada en la prueba en planta.

Una de las fuentes programables de corriente (IDAC) se conecta a la entrada analógica positiva V_{in+} configurada con una corriente de $50\mu A$, la ganancia interna del ADC se configuró en 1 para respetar las restricciones propias del ADC para el rango de voltajes de entrada a medir.

El multiplexor **MUX 1** permite conectar el ADC a uno de los tres sensores a medir, mientras que el **MUX 2** determina el camino que sigue la corriente I_1 hacia GND, en esta configuración en **MUX 2** conecta V_{in-} para permitir el retorno de la corriente por el segundo cable del sensor, utilizando I_1 para generar el voltaje de referencia, por este motivo la medición es radiométrica, es decir que el valor medido es independiente de la corriente utilizada y sus variaciones, siendo solo proporcional a la relación entre la resistencia del sensor y la resistencia R_{ref} como se prueba a continuación.

El voltaje diferencial entre los pines V_{in+} y V_{in-} estará dado por:

$$V_{dif} = I_1 (R_{cable1} + R_{cable2} + R_{sensor} + R_{mux1\ canal1} + R_{mux1\ canal2})$$

Si despreciamos las variaciones en todas las resistencias salvo la resistencia del sensor podemos reescribir V_{dif} como:

$$V_{dif} = I_1 R_{sensor} + I_1 (R_{cable1} + R_{cable2} + R_{mux1\ canal1} + R_{mux1\ canal2})$$

$$V_{dif} = I_1 R_{sensor} + V_{offset}$$

De la misma forma podemos escribir el voltaje de referencia ($V_{ref+} - V_{ref-}$) como:

$$V_{ref} = I_1 R_{ref}$$

Por lo tanto el valor medido por el ADC con la configuración actual y teniendo en cuenta que si bien el ADC posee 24 bits de resolución utiliza un bit para indicar la polaridad del voltaje medido, será:

$$Valor_{medido} = \frac{V_{dif} \times Ganancia \times 2^{23}}{V_{ref}}$$

Lo que es igual a

$$Valor_{medido} = \left(\frac{I_1 R_{sensor}}{I_1 R_{ref}} + \frac{I_1 R_{offset}}{I_1 R_{ref}} \right) \times Ganancia \times 2^{23}$$

$$Valor_{medido} = \left(\frac{R_{sensor} + R_{offset}}{R_{ref}} \right) \times Ganancia \times 2^{23}$$

Como se observa, el valor medido es independiente de la corriente de excitación usada, y si llamamos a $R_{sensor} + R_{offset} = R_{medida}$ podemos despejar su valor de la ecuación anterior.

$$R_{medida} = \frac{(R_{ref} \times Valor_{medido})}{(Ganancia \times 2^{23})}$$

La resistencia medida estará dada por un valor constante R_{offset} , y el valor de interés R_{sensor} , para desafectar R_{offset} de la medición se realiza una medición con los cables del sensor cortocircuitados.

Calculo de error

Los cálculos se realizarán suponiendo una temperatura máxima de trabajo de 40°C, 15°C por arriba de la temperatura ambiente estándar (25°C) utilizada para caracterizar los coeficientes de desviación.

Componente	Tolerancia	Desviación en ppm/°C
Resistencia de referencia	0,05%	10
Ganancia del ADC	0,01%	0,5

Calculo de desviaciones de los valores nominales

$$\sigma_{Rref} = 15^\circ\text{C} \times 10\text{ppm} \times 30\text{k}\Omega = 4.5\Omega \equiv \mathbf{0.015\%}$$

$$\sigma_{Gadc} = 15^\circ\text{C} \times 0.5\text{ppm} \times 50\mu\text{A} = 0.375\text{nA} \equiv \mathbf{0.00075\%}$$

Dado que el error relativo del producto es la suma obtenemos que el error en la medición de resistencia sea:

$$\begin{aligned} \varepsilon_{medición} &= \varepsilon_{Resistencia} + \varepsilon_{GananciaADC} = (0.05\% + 0.015\%) + (0.01\% + 0.00075\%) \\ &= 0.075\% \end{aligned}$$

Dado que para desafectar el valor de offset de la medición realizara una medición del mismo, la resistencia del sensor será calculada como:

$$R_{sensor} = R_{medida} - R_{offset}$$

Y por lo tanto el error absoluto en el valor obtenido será la suma de los errores absolutos, con el fin de poder comparar el error de estas mediciones con los datos recopilados en el capítulo III, el análisis de error se realizara en el entorno de los 21°C, utilizando los datos ya presentados en la sección 2.4.3 del cap. III.

$$R_{21^{\circ}C} = 12040 \Omega$$

$$\frac{d\Omega}{d^{\circ}C_{21^{\circ}C}} \leq \frac{(R_{30^{\circ}C} - R_{21^{\circ}C})}{30^{\circ}C - 21^{\circ}C} = -436,77 \Omega/^{\circ}C$$

$$R_{offset} = 350 \Omega \times 2 \text{ (Resistencia de ambos canales del MUX 2)}$$

Por lo que el error absoluto en R_{sensor} será:

$$\Delta R_{sensor} = (12040 \Omega \times 0.075 \%) + (750 \Omega \times 0.075 \%) = 9,5925 \Omega \equiv 0.075 \%$$

Podemos llegar a que el error en °C en las cercanías de los 21°C será el siguiente:

$$\varepsilon_{medición} = 9,5925 \Omega \equiv 0,0219^{\circ}C$$

Nuevo ajuste del modelo y simulación

El modelo ya desarrollado en el capítulo III, fue ajustado utilizando el promedio de las dos mediciones de la temperatura ambiente, y los parámetros $k_{tcooler}$ y $T_c(t)$ obtenidos en la sección 2.5.3 del capítulo III, con el fin de simular como se hubiera comportado nuestro controlador si las válvulas manuales hubieran estado abiertas.

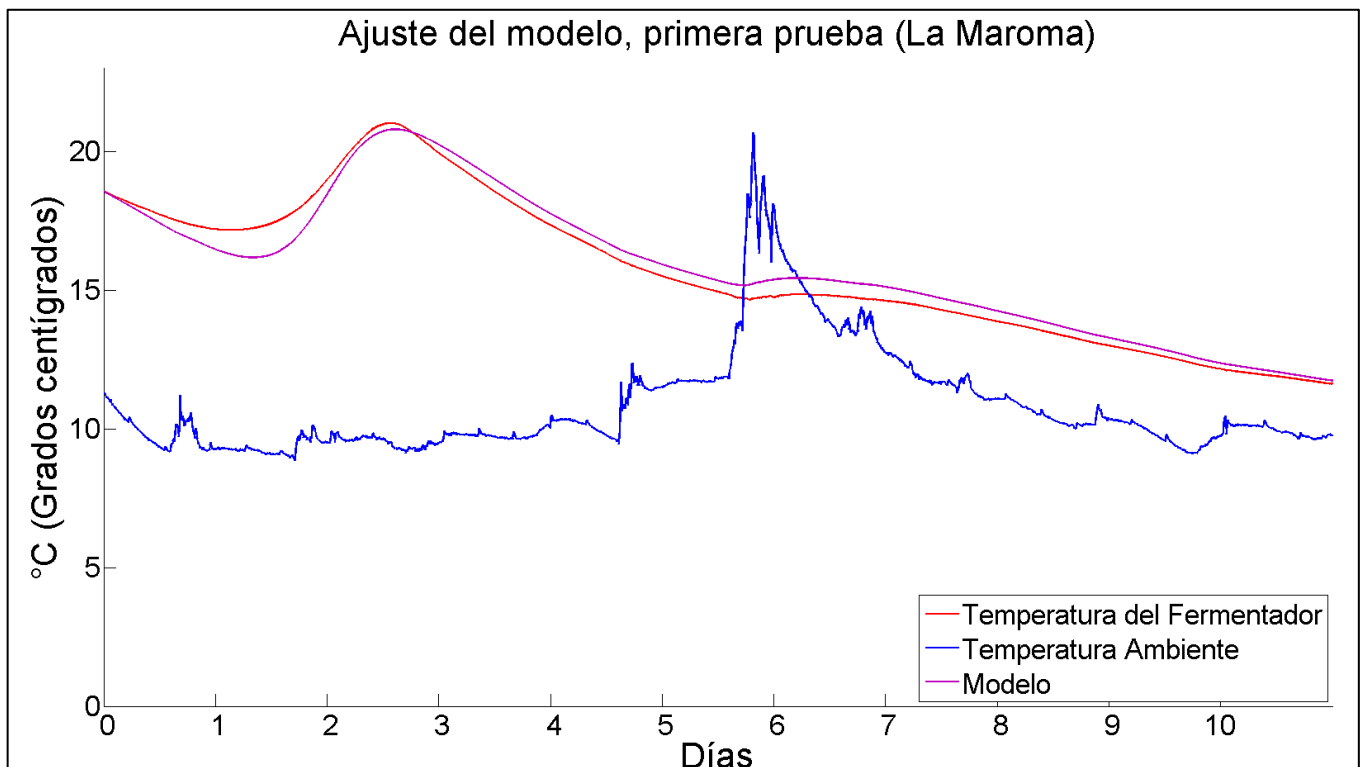


Figura V-20: Ajuste del modelo a la prueba de campo.

Parámetros usados $\eta = 0.007, \beta = 0.001, \alpha = 0.01, N_0 = 65, S_0 = 345, A_0 = 0,$

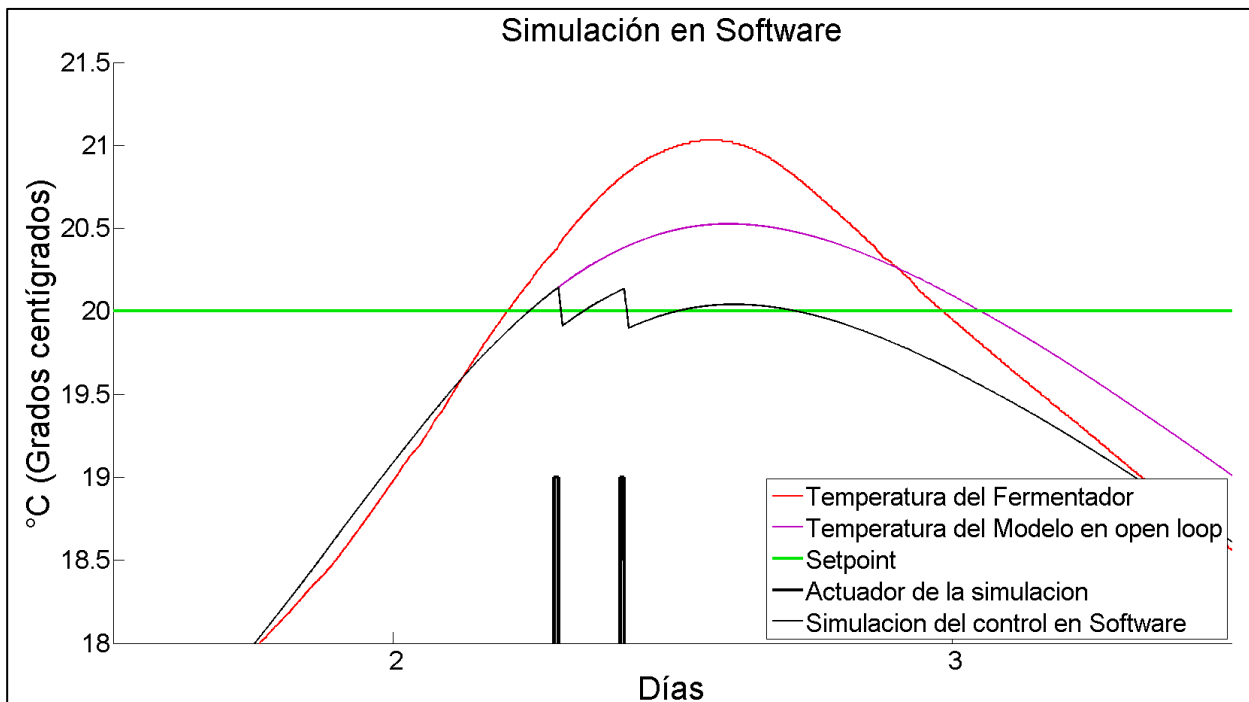
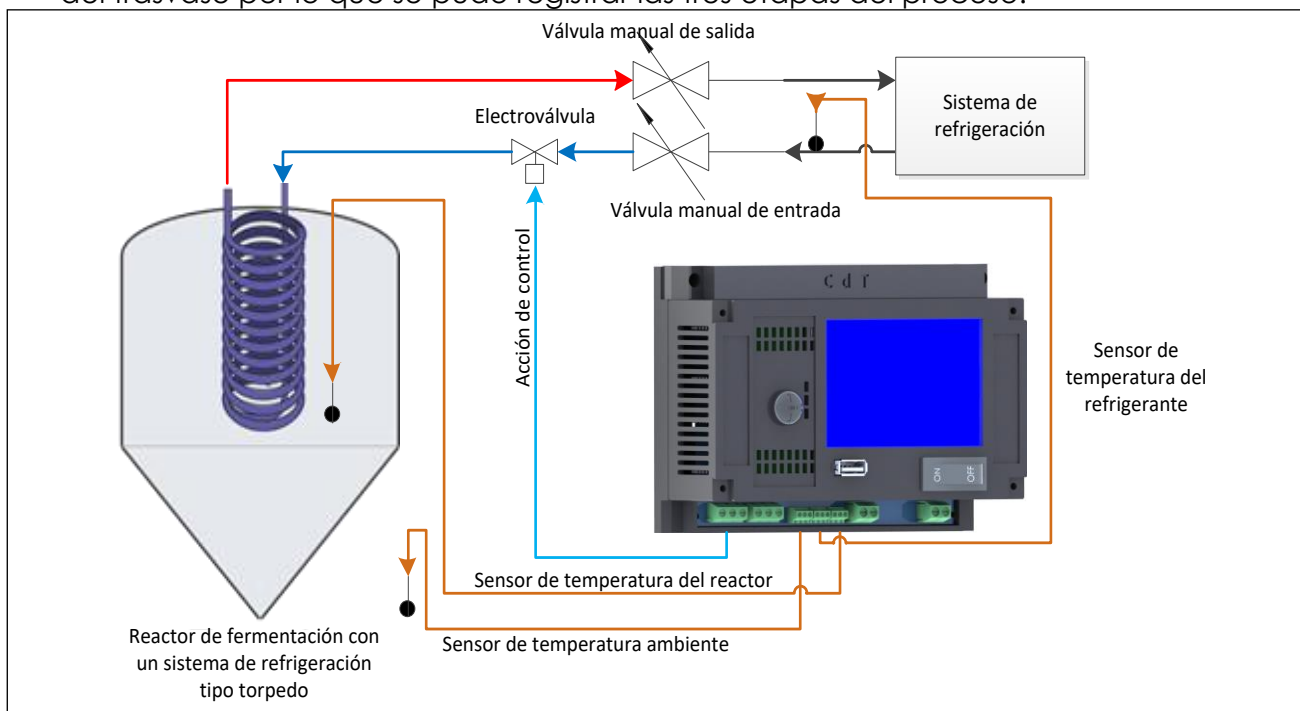


Figura V-21: Simulación de control sobre los datos de la prueba de campo.

$T_0 = 18.56^\circ\text{C}, k_{t,levadura} = 0.00037, k_{t,ambiente} = 0.75, k_{t,cooler} = 1.3, T_c(t) = -1,1^\circ\text{C},$
 $t = 0.000275$

Segunda experiencia

El día 5 de julio de 2019 realizan la cocción nuevamente de una cerveza tipo ALE- IPA (misma de la prueba previa). Se establecen dos diferencias con respecto a la anterior preparación. La primera fue el tiempo de cocción, que de dos horas se redujo a una hora, la segunda fue que no se utilizó la técnica del trasvase por lo que se pudo registrar las tres etapas del proceso.



Se utiliza un sensor de temperatura NTC 10K a 25°C en la vaina del reactor diseñada a tal fin, además se agregan dos sensores extras, de características similares, para medir la temperatura ambiente y del caño del líquido refrigerante.

Luego de 18 días en el tanque de fermentación se da por finalizada la prueba, el 23 de julio de 2019.

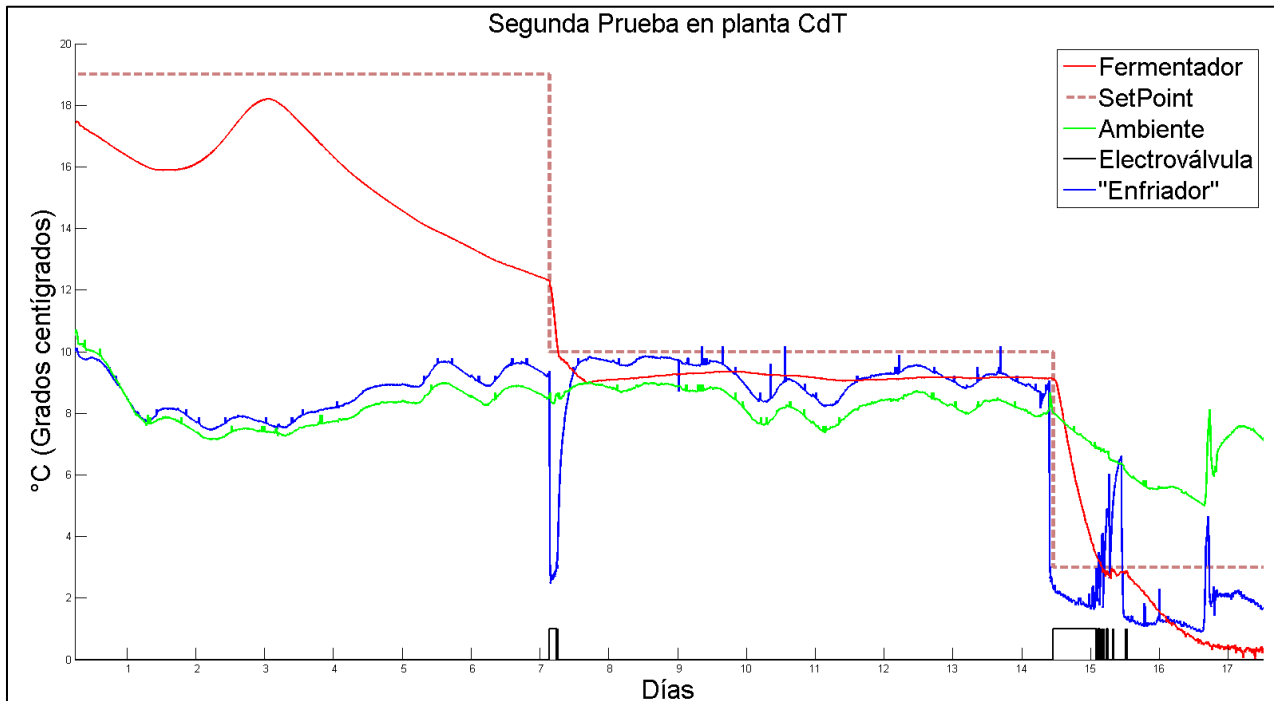


Figura V-22: Datos adquiridos durante la segunda experiencia de campo.

Como se puede observar en la figura V-22, la temperatura del fermentador no alcanzó el setpoint durante la fermentación debido a la baja temperatura ambiente, por lo que nuevamente no podemos comprobar la correcta acción del control durante esta etapa. Sin embargo se puede observar la acción de control durante los dos cambios de setpoint previos a la maduración, los cuales serán analizados a continuación.

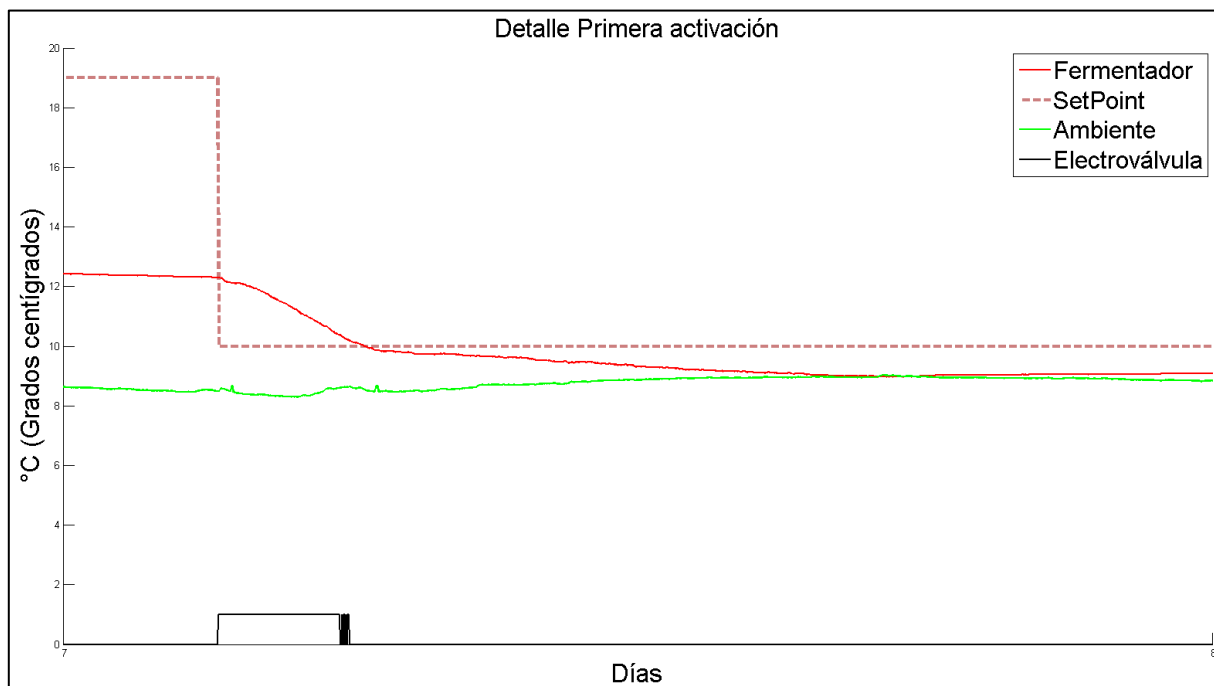


Figura V-23: Detalle del primer cambio de setpoint y consecuente activación del control.

Durante el primer cambio de setpoint (Figura V-23) el cual ocurrió entre el séptimo y octavo día, se puede observar claramente como luego de la apertura de la electroválvula hay una pendiente negativa debido a la circulación del refrigerante, y si bien el algoritmo de control cierra la electroválvula cuando se está por alcanzar el setpoint, la temperatura continúa bajando debido a que dentro del sistema de refrigeración queda refrigerante que todavía está frío, esto se puede observar más claramente en la Figura V-22 en la cual se ve un claro cambio de pendiente luego de el apagado de la electroválvula.

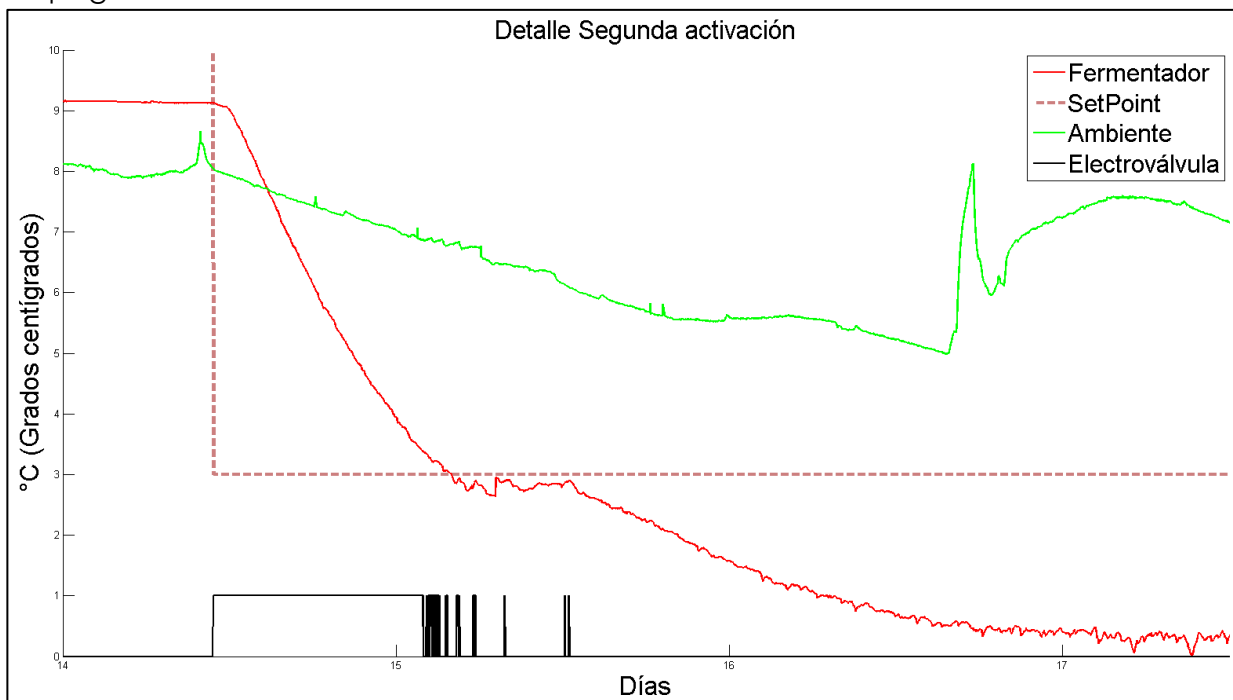


Figura V-24: Detalle del segundo cambio de setpoint y consecuente activación del control.

Debido a que el sistema no cuenta con una forma de calentar el

fermentador, si la temperatura ambiente se encuentra por debajo del setpoint como ocurre en esta ocasión, a medida que transcurre el tiempo la temperatura del fermentador tiende a la temperatura ambiente y el sistema de control pierde la capacidad de controlar el sistema.

El segundo cambio de setpoint genera una situación más interesante ya que fuerza la temperatura del fermentador por debajo de la temperatura ambiente, la cual funciona como una fuente de calor para el fermentador, por este motivo se puede observar en la Figura V-24 como luego de que la temperatura del fermentador alcanza el setpoint, el sistema de control debe continuar accionando la electroválvula para evitar que la temperatura aumente. Lamentablemente en la última activación la electroválvula se traba por lo que el control deja de actuar ya que la temperatura continúa descendiendo asintóticamente a la temperatura del refrigerante aún con la señal de control en "0".

VI - Diseño del sistema

1. Diagrama en bloques

El sistema electrónico para realizar un control térmico digital requiere principalmente de 3 secciones (*Figura V-1*). La adquisición de valores de temperatura, el procesamiento de la información y el actuador.

La adquisición de datos resulta en un driver capaz de interpretar señales provenientes de uno o más sensores de temperatura. A su vez debe traducir dichas señales en información. El procesamiento de información debe implementar un sistema de control que analice los valores de temperatura y realice acciones en concordancia. Así como existen múltiples sensores, a los que la sección de adquisición de datos podría ser capaz de interpretar, existen múltiples tipos de actuadores. Esta sección opera sobre un sistema externo a la plataforma.

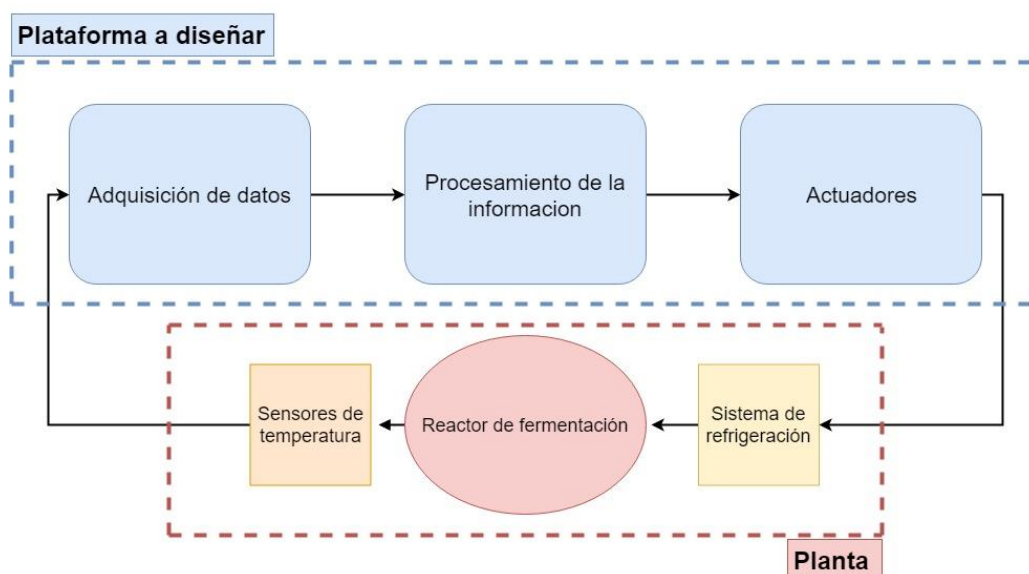


Figura VI-1: Diagrama simplificado del controlador de temperatura

Este proyecto final pretende que el sistema electrónico destaque en conectividad y configurabilidad, por lo que en un diagrama más detallado de lo que es la plataforma se agregan módulos de comunicación.

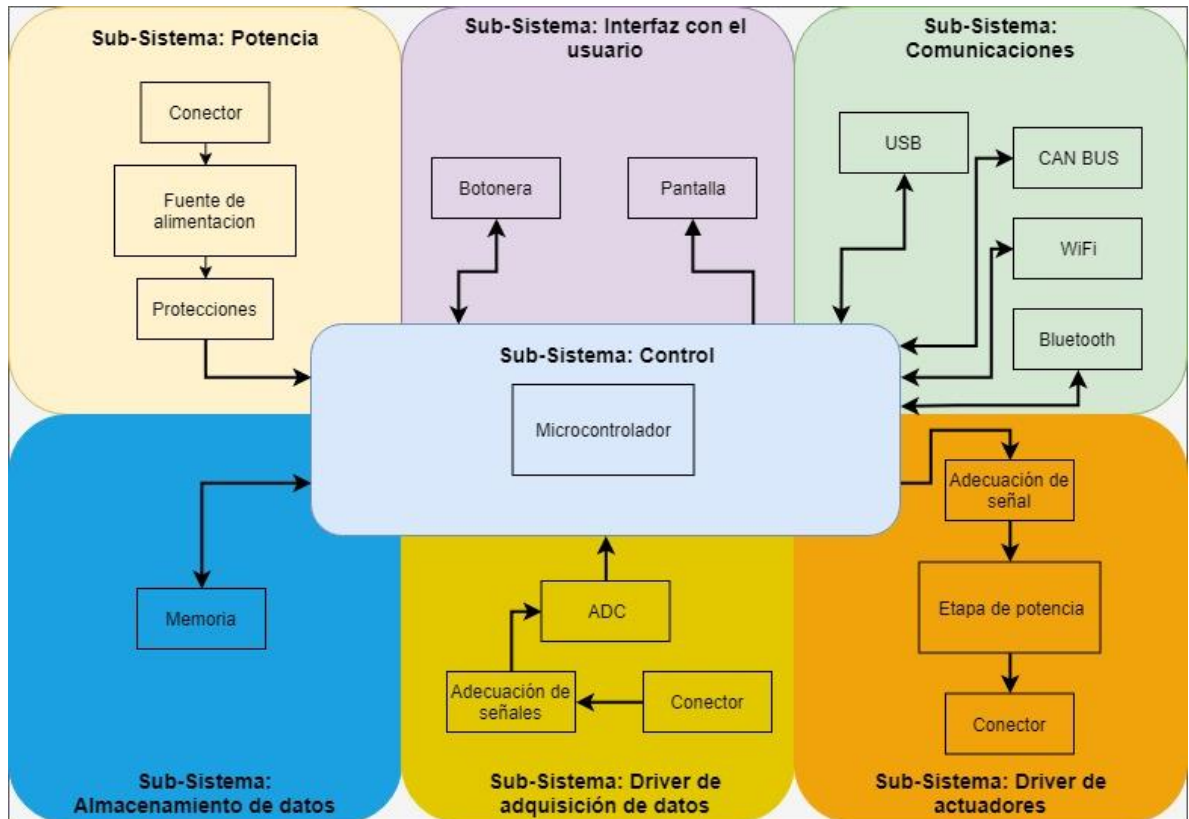


Figura VI-2: Diagrama de la plataforma

2. Requerimientos de la plataforma y selección de componentes

Los requerimientos que se detallan a continuación surgen a partir de la comparación con otros dispositivos que se encuentran en el mercado argentino, con los utilizados por las empresas locales de producción de cerveza artesanal y de la plática con los productores locales de equipamiento de refrigeración.

Para la selección de componentes, los criterios son los siguientes:

- Tipo de componentes: Mixto, con preferencia en SMD.
- Que simplifiquen circuitos para reducir el tamaño del PCB.
- Relación costo-beneficio.
- Para el bloque de adquisición de datos las tolerancias de los componentes menor o igual al 1%.

Visto que el presente proyecto presenta la primera instancia en el desarrollo de un producto se hizo hincapié en ciertos márgenes de seguridad y de estabilidad en el tiempo de los circuitos implementados.

Una consideración importante también fue el costo del dólar que, durante todo el desarrollo PFI, se valuó frente al peso Argentino un 50%.

2.1. Subsistema: Adquisición de datos

En la industria local se utilizan variados tipos de sensores de temperatura. Los más utilizados son los termopares, y los sensores de temperatura resistivos (RTD y termistores).

Los termopares consisten esencialmente en dos tiras o alambres hechos de metales diferentes y unidos en un extremo. Los cambios en la temperatura en esa junta inducen un cambio en la fuerza electromotriz entre los otros ex-

tremos. A medida que la temperatura sube, esta fuerza electromotriz de salida del termopar aumenta, aunque no necesariamente en forma lineal.

Los dispositivos termométricos de resistencia aprovechan el hecho de que la resistencia eléctrica de un material cambia al cambiar su temperatura. Dos tipos de sensores de temperatura clave son los dispositivos metálicos (normalmente conocidos como RTD) y los termistores. Como su nombre indica, los RTD confían en el cambio de resistencia en un metal, con la resistencia aumentando en forma lineal con la temperatura.

Los termistores se basan en el cambio de resistencia en un semiconductor de cerámica; la resistencia cae en forma no lineal con el aumento en la temperatura.

Cada uno tiene sus características físicas asociadas e incluso varían los tipos de sensores dentro de la misma clase. En la tabla V-1 se muestran los principales sensores y características.

Tipo de sensor	Clase	Rango[°C]	Descripción
Termopar	K	-184 a 1260	Uso en vacío o atmósferas reductoras
	N	-270 a 1300	Alternativa a los tipo K con más estabilidad a alta temperatura
	T	-184 a 400	Muy efectivo en ambientes húmedos y bajas temperaturas
RTD	Platino	-200 a 850	Se caracterizan por ser muy lineales y no presentar grandes variaciones a lo largo de su vida útil
	Cobre	-200 a 260	
	Níquel	-80 a 320	
	Molibdeno	- 200 a 200	
Termistores	NTC	-100 a 450	Alineales, comúnmente utilizados como sensores de temperatura. Bajo costo
	PTC	-100 a 350	Se utilizan como protección por sobre corriente.

Tabla VI-1: Sensores y características admisibles (9)

Los productores locales utilizan en su mayoría sensores del tipo Termistor, específicamente los NTC de 10KOhm a 25° C. Por esta razón fue que decidimos acotar la compatibilidad del equipo a sensores de tipo resistivo, pudiendo medir sensores NTC, y RTD tanto de dos como de tres cables.

En cuanto a la medición del sensor, se opta por usar el **ADC ADS122U04** del fabricante Texas Instruments, el cual cuenta con una resolución de 24 bits más que suficiente para medir este tipo de sensores. El factor más importante en esta elección fue el hecho de que el integrado incorporara fuentes de corriente programables, y luego se priorizó en reducir el costo, ya que todos los dispositivos que poseían fuentes de corriente programables contaban con características muy similares y suficientes para nuestros requerimientos.

El ADC seleccionado cuenta con 2 fuentes de corriente emparejadas, lo que posibilita realizar mediciones con sensores RTD de 3 cables, además cuenta con la opción de utilizar una referencia externa, la cual en nuestro diseño es generada por una resistencia externa por la cual circula la misma corriente que excita los sensores, esto permite realizar mediciones ratiométricas, de este modo valor medido es proporcional a la relación entre dos resistencias y se independiza de posibles variaciones en las fuentes de corriente.

Para poder medir tanto NTC, como RTD de dos y tres cables, y además poder utilizar 3 sensores simultáneamente conectados, la etapa de adquisición

cuenta con un conjunto de multiplexores que varían el circuito de adquisición de acuerdo al tipo de sensor a medir, y conectan uno de los 3 sensores del equipo al circuito de adquisición. A continuación se muestra un diagrama explicativo de los dos modos de conexión, a la izquierda se puede ver la configuración para sensores de 3 cables, la principal ventaja de esta configuración es que la resistencia de los cables del sensor es desafectada de la medición, esto es especialmente útil cuando se utilizan sensores como los pt100, en los cuales la resistencia de los cables puede generar un error considerable. La configuración de dos cables por otro lado, es utilizada para medir sensores como puede ser un ntc de 10K Ω , y dado a la dimensión de la resistencia a medir es despreciable el error introducido por la resistencia de los cables, pudiendo entonces utilizar solo dos cables con el sensor.

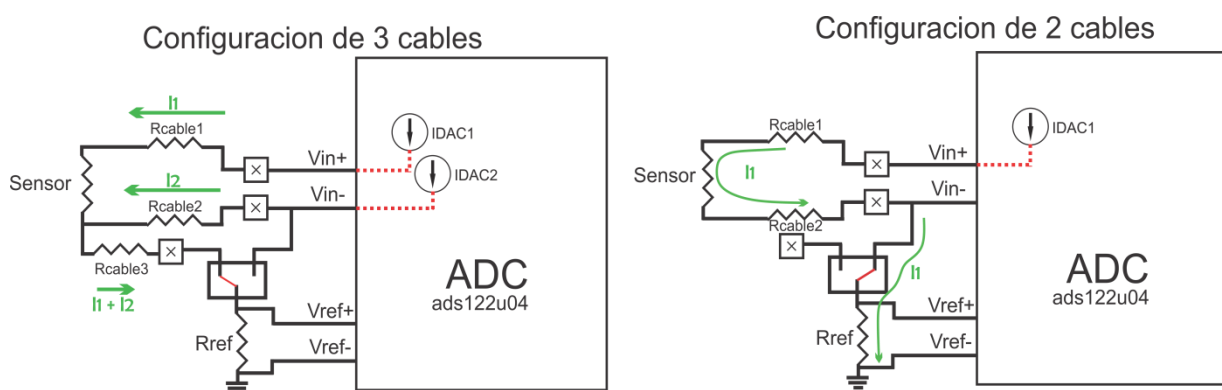


Figura VI-3: Conexiones del ADC según el tipo de sensor.

Los multiplexores fueron elegidos teniendo en cuenta los siguientes parámetros:

- que fueran analógicos y bidireccionales
- La diferencia de resistencia entre un canal y otro fuera pequeña
- La resistencia en estado On fuera pequeña
- El voltaje de excursión de los puertos fuera de al menos 0-5V
- El precio
- La cantidad de canales

Los multiplexores elegidos para alternar entre los 3 sensores del equipo son los **HEF4052B** de NXP, los cuales poseen 4 canales, y para cambiar la configuración del circuito de adquisición entre sensores de 2 cables y 3 cables se utiliza el multiplexor **SN74LVC1G3157** de Texas Instruments.

2.2. Subsistema: Actuadores

Se utilizan seis salidas independientes para los actuadores. De esta manera se puede configurar cada una de ellas para que responda a diferentes situaciones. Por ejemplo se puede configurar 2 salidas por sensor y de esta manera "enfriar" y "calentar" o utilizar alguna salida para activar una señal lumínica o sonora.

Dado que lo que se quiere controlar utiliza una tensión de 220VAC-50Hz se requiere desacoplar al control de los actuadores por cuestiones de seguridad. Además se requiere un actuador capaz de manejar corrientes de hasta 3A lo que asegura el correcto manejo de electroválvulas y de mantas térmicas eléctricas.

Se utiliza el triac BTA212 de NXP con una tensión máxima de 600V y corriente máxima de trabajo de 12A. Se elige este componente ya que cumple cómodamente con los requerimientos y por ser el de más bajo costo. En las tablas 2 y 3 se puede ver los consumos típicos y la temperatura máxima alcanzada por el triac con el consumo máximo.

Para el manejo de los triacs se eligieron los MOC3063S de LITEON el cual tiene una corriente de disparo mínima de 5mA, lo que permitió que no sea necesario componentes extra para poder controlarlo con el micro controlador. Además tiene integrado un detector de cruce por cero lo que simplifica aún más el diseño.

Dispositivo a controlar	Corriente [A]	Potencia [W] ⁵
Electroválvula	0,5	72,6
Manta térmica	3	435,59
Foco	0,068	9,87
Buzzer	0,5	72,6
Corriente máxima de diseño	3	435,59

Tabla VI-2: Consumos típicos

Parámetro	Valor	Unidad	Descripción
T _a	35	°C	Temperatura ambiente
T _{J-Max}	125	°C	Temperatura máxima del encapsulado
R _{J-MB}	1,5	°C/W	Resistencia juntura-encapsulado
V ₀	1,175	V	Tensión de caída
I _{T(AVE)}	2,7	A	Corriente promedio
R _s	0,0316	Ohm	Resistencia triac
I _{T(RSM)}	3	A	Corriente de aplicación
P	3,46	W	Potencia entregada por el triac
R _{HS-A}	24	°C/W	Resistencia disipador-ambiente
R _{MB-HS}	0,5	°C/W	Resistencia encapsulado-disipador con grasa
R _{J-A}	26	°C/W	Resistencia total, juntura-ambiente
T _j	124,91	°C	Temperatura máxima alcanzada

Tabla VI-3 : Temperatura máxima alcanzada por el triac (10)

2.3. Subsistema: Comunicaciones

Uno de los requerimientos que surge de la plática con los productores de cerveza es la falta de conectividad de sus controladores de temperatura. La falta de información sobre el transcurso del proceso y/o el registro de datos automáticos hacen del trabajo de control una tarea tediosa.

Para cumplir con ese requerimiento se emplean dos métodos de acceso remoto. Mediante una red de internet-WiFi y mediante una conexión Bluetooth. Además se agrega una conexión USB y una conexión CAN BUS.

⁵ Se utiliza para los cálculos una tensión de 220VCA y un factor de potencia de 0,85

2.3.1 USB

El Puerto USB tipo A será utilizado por el usuario para, mediante un Pen Drive, cargar o descargar archivos de configuración, curvas de temperatura, etc. Además se le agrega un puerto micro USB, tipo AB, para debugger y programación, este puerto no estará al alcance del usuario.

No se utiliza la funcionalidad de OTG ya que eso requeriría de adaptadores o pen drives especiales, en cambio se opta por la implementación de un circuito de alimentación que garantice los 500mA del estándar USB 2.0 para encender el pen drive. Se incluye además un integrado de protección de las líneas de comunicación D+ y D-.

2.3.2 WIFI

La conexión WiFi tiene como objetivo que el dispositivo pueda conectarse a una computadora y así permitirle al usuario una lectura y visión del proceso más completa. Además si hay más de un dispositivo instalado estos podrán ser visualizados conjuntamente en la pc, centralizando de esta manera el control de temperatura.

Desde la aplicación de computadora (*SO Windows*) será posible operar el controlador de forma completa. En esta primera instancia solo se podrá visualizar las operaciones que realizan los controladores conectados.

Para la implementación de este periférico se optó por el módulo WiFi comercial ESP-07 WiFi Module V1.0 desarrollado por Ai-thinker Team que posee un núcleo ESP8266. Entre sus características destacadas esta la simplificación del circuito (16mmx21.2mm), la antena incorporada (aunque se incluye una antena auxiliar) y bajo consumo. Posee una comunicación UART para la transferencia de datos con el microcontrolador.

2.3.3 Bluetooth

La conexión Bluetooth da al operador la opción de verificar configuraciones, visualizar el estado de funcionamiento del controlador y cambiar configuraciones desde un teléfono celular.

Para la implementación se eligió un módulo HC-06 V2.0 dada su simplicidad de operación, que posee una comunicación UART para la transferencia de datos hacia el micro controlador y su reducido espacio.

2.3.4 Can Bus

La utilización de esta capa física es para brindarle al dispositivo una comunicación industrial robusta para la intercomunicación de dispositivos. De esta manera si entre dispositivos están conectados por CAN BUS solo basta con que uno de ellos se conecte a una red WiFi para tener acceso a toda la red.

El circuito necesario para la implementación de la capa física de CAN BUS se hizo en base al MCP2557FD/8FD. Este transceptor es un integrado de Microchip lo que lo hace altamente compatible con los micro controladores de la misma marca.

2.4. Subsistema: Interfaz con el usuario

Más allá de las diferentes formas remotas de acceder a la plataforma para controlarla se añaden una pantalla y un joystick.

Los controladores comerciales disponen en su mayoría pantallas fabricados con 7 segmentos. Para este dispositivo es necesario que se pueda representar mayor cantidad de información en la pantalla por lo que se eligió una pantalla LCD TFT de 3,5 pulgadas (88.9 mm). Se optó por esta pantalla ya

que fue la pantalla con mayores dimensiones y menor precio que encontramos.

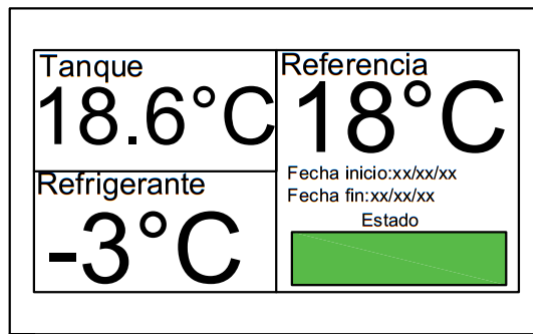


Figura VI-4: Pantalla ilustrativa



Figura VI-5: Render del frente del equipo

2.5. Subsistema: Potencia

Considerando que el dispositivo pretender ser compacto, no es posible utilizar una fuente externa para alimentarlo. Se requiere que pueda ser conectado directamente a la red de distribución de electricidad 220VCA a 50Hz y que posea una eficiencia superior al 70%.

Se identifican múltiples valores de tensión continua, 3.3V para el micro controlador y algunos periféricos y 5V para otros periféricos.

La conversión de corriente alterna a continua se realiza con una fuente IRM-10 de la empresa MEAN WELL. Dicha fuente es del tipo switching de 10W AC-DC en 5V y tiene una eficiencia del 77%. Para el resto de las tensiones se utiliza un regulador lineal a 3.3V (LD1086) y un regulador lineal en 4.7V (LP3985). Se decidió esta configuración para que la disipación de potencia en los reguladores sea mínima y no sea necesaria la utilización de disipadores (tabla 5). El ADC puede ser alimentado con 5V directamente, pero al ser esta tensión proveniente de una fuente switching genera componentes de frecuencia que podrían alterar la medición de temperatura. Para evitar esta situación se utiliza el LP3895.

Además contara con una batería auxiliar de 3.3V para poder mantener al RTC funcionando inclusive ante un corte en el suministro de electricidad de 220V AC.

Regulador 3.3V-LD1084			Regulador 4,7V-LP3985		
Parámetro	Valor	Unidad	Parámetro	Valor	Unidad
T_a	35	°C	T_a	35	°C
T_{jmax}	150	°C	T_{jmax}	125	°C
R_{J-MB}	3	°C/W	R_{J-MB}	3	°C/W
R_{J-A}	62,5	°C/W	R_{J-A}	220	°C/W
I_{out}	0,97	A	I_{out}	0,006	A
P	1,65	W	P	0,002	W
T_j	138,03	°C	T_j	35,38	°C

Tabla VI-4: Temperatura máxima que alcanzan los reguladores

	Consumo [A]	Consumo pico [A]	Cantidad	Potencia [W]	Potencia pico [W]
Tensión = 3,3V					
Micro controlador	0,08	0,35	1	0,264	1,155
Memoria	0,0001	0,025	1	0,0002	0,0825
LED	0,01	0,01	4	0,132	0,132
ADC	0	0	1	0	0
Opto acoplador	0,005	0,005	8	0,132	0,132
WiFi	0,4	0,4	1	1,32	1,32
Pantalla	0,115	0,115	1	0,3795	0,3795
Potencia total [W]				2,23	3,2
Tensión = 4,7V					
HEF4052B	0,00015	0,00015	2	0,0014	0,0014
74LVC1G3	0,0005	0,0005	1	0,0024	0,0024
74LVCH2T45	0,00001	0,00001	1	0,00005	0,00005
ADC	0	0,005	1	0	0,0235
Potencia total [W]				0,004	0,027
Tensión = 5V					
Led	0,015	0,015	1	0,075	0,075
Regulador 3.3V	0,4455	0,6402	1	2,228	3,201
Regulador 4.7V	0,0008	0,0055	1	0,004	0,027
Fuente USB	0,5	0,5	1	2,5	2,5
Transceptor CANBUS	0,07	0,14	1	0,35	0,7
Pantalla	0,13	0,13	1	0,65	0,65
Bluetooth	0,06	0,06	1	0,3	0,3
Potencia total [W]				6,11	7,45

Tabla VI-5: Consumos parciales y totales del dispositivo

Potencia máxima dispositivo [W]	7,45
Eficiencia IRM-10 [%]	77
Potencia máxima requerida a la red [W]	9,68
Corriente máxima requerida a la red [A]	0,044

Tabla VI-6: Consumo a la red eléctrica

2.6. Subsistema: Memoria

Parte de las ventajas que supone esta implementación es la capacidad de analizar los datos recolectados a lo largo del proceso. Además es necesario poder almacenar curvas de temperatura para que el controlador siga a lo largo del tiempo y para ello es necesario incluir una memoria.

	Valor	[Unidad]	Notas
Cantidad de días a registrar	45	días	Ningún tipo de cerveza toma más de 30 días
Información de fecha y hora de inicio	48	bits	
Información de fecha y hora de finalización	48	bits	
Numero de tanque	8	bits	
Tipo de cerveza	6	bits	
Nombre de tabla	6	bits	64 tipos de cerveza
Precisión del sensor	0,01	Centígrados	
Alertas	510	bits	(3bits tipo alerta+48bits fecha y hora)*10
Rango de temperatura	60	Centígrados	
Tiempo de muestreo	1	minuto	
Muestras por hora	60	muestras	
Cálculos			
Cantidad de bits por muestra	16	bits	
Cantidad de bits por día	23.040	bits	
Cantidad de muestras en 45 días	1.036.800	bits	
Información extra necesaria	620	bits	
Cantidad de bits por registro	1.037.426	bits	
Total de bits necesarios	1.013	Kbits	
Total memoria necesaria	0,989	Mbits	Por cada registro.

Tabla VI-7: Tamaño de la memoria

2.7. Subsistema: Control

Al subsistema de control lo integra únicamente un micro controlador de la empresa MICROCHIP que cuenta con una velocidad máxima de CPU de 80Mhz, un módulo CAN, un módulo USB FullSpeed/Host/Device/OTG y hasta 85 pines de entrada/salida. La elección del PIC32MX764F128L está dada en base a los requerimientos listados en la tabla V-8. Además dentro de la amplia gama de micros controladores de Microchip, este es el más económico.

	Señal/Periférico	Tipo de interconexión	Nomenclatura esquemáticos	Pin	Nomenclatura PIC
1	Actuador	Pin digital - Salida	ACT1	78	RD3
2	Actuador	Pin digital - Salida	ACT2	80	RD13

3	Actuador	Pin digital - Salida	ACT3	76	RD13
4	Actuador	Pin digital - Salida	ACT4	68	RD8
5	Actuador	Pin digital - Salida	ACT5	69	RD9
6	Actuador	Pin digital - Salida	ACT6	77	RD2
7	CAN BUS	Módulo CAN	CRx	87	C1RX
8			CTx	88	C1TX
9		Pin digital - Salida	CBx	90	RG0
10	ADC	Pin digital - Salida	AdcDataReady	19	RE9
11		Pin digital - Salida	RSTADC	29	RA10
12		Módulo UART	U3Tx	12	U3TX
13			U3Rx	11	U3RX
14	USB	Módulo USB	USB_P	57	D+
15			USB_N	56	D-
16		Pin digital - Salida	USBEn	20	RB5
17	WiFi	Módulo UART	U2Tx	10	U6TX
18			U2Rx	14	U6RX
19		Pin digital - Salida	RSTWIFI	28	RA9
20		Pin digital - Salida	MEn	35	RB11
21	Bluetooth	Módulo UART	U1Tx	53	U1TX
22			U1Rx	52	U1RX
23		Pin digital - Salida	BluetoothOn	34	RB10
24	Multiplexores	Pin digital - Salida	SensorSEL0	83	RD6
25		Pin digital - Salida	SensorSEL1	79	RD12
26		Pin digital - Salida	MuxSEL1	84	RD7
27	Programación/Debug	Módulo UART	U4Tx	48	U4RX
28			U4Rx	47	U4TX
29		Programación	PGD	25	PGED1
30			PGC	24	PGEC1
31	Joystick	Pin digital - Entrada	BIzq	96	RG12
32		Pin digital - Entrada	BDer	95	RG14
33		Pin digital - Entrada	BCen	97	RG13
34		Pin digital - Entrada	BAr	1	RG15
35		Pin digital - Entrada	BAb	89	RG1
36	Memoria	Módulo SPI	MISO	49	SDI4
37			MOSI	50	SDO4
38			SPI_SCL	39	SCK4
39		Pin digital - Salida	MemCS	38	RA1
40	Comprobación de tensiones auxiliares	Módulo ADC	3V3Test	23	AN2
41			VbatT	26	AN6
42	LED	Pin digital - Salida	LD5	17	RA0
43		Pin digital - Salida	LD6	66	RA14
44		Pin digital - Salida	LD7	67	RA15
45	Pantalla	Módulo paralelo	LcdD0	93	PMD0
46			LcdD1	94	PMD1
47			LcdD3	98	PMD2

48		LcdD3	99	PMD3
49		LcdD4	100	PMD4
50		LcdD5	3	PMD5
51		LcdD6	4	PMD6
52		LcdD7	5	PMD7
53		LcdRd	82	PMRD
54		LcdCs	70	PMCS2
55		LcdRs	71	PMA14
56		LcdWr	81	PMWR
57	Pin digital - Salida	RSTLCD	92	RA7

Tabla VI-8: Lista de señales y periféricos

3. Diseño del PCB

En esta sección se describe el diseño del circuito y algunos criterios que se tuvieron en cuenta.

El circuito fue implementado en dos PCB a causa de tener que adaptarse al gabinete. Dichos PCB se denominan “Placa A” y “Placa B”

3.1. Placa A

En la Figura V-4 se puede ver el modelo 3D de la “Placa A” la cual está hecha en material FR-4 TG130, tiene un espesor de 1.6mm, 2 capas y una dimensión de 128 x 105 mm. Esta placa alberga a casi todos los subsistemas a excepción del subsistema de interfaz con el usuario.

Primó el criterio, al momento de la distribución de componentes, de la separación por etapas. (Figura V-5)

- Alta tensión
- ADC y adquisición de datos
- CAN BUS
- Control en general y periféricos

Además se tuvo en cuenta la corriente máxima de las pistas en la etapa de alta tensión y se las preparo para poder ser estañadas en el proceso de construcción.

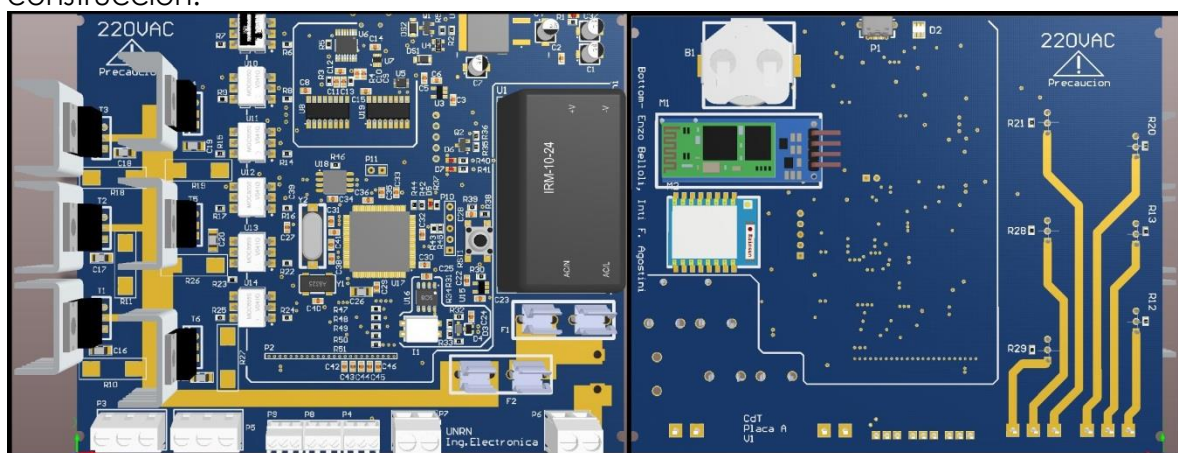


Figura VI-6: Placa A-TOP/BOTTOM.

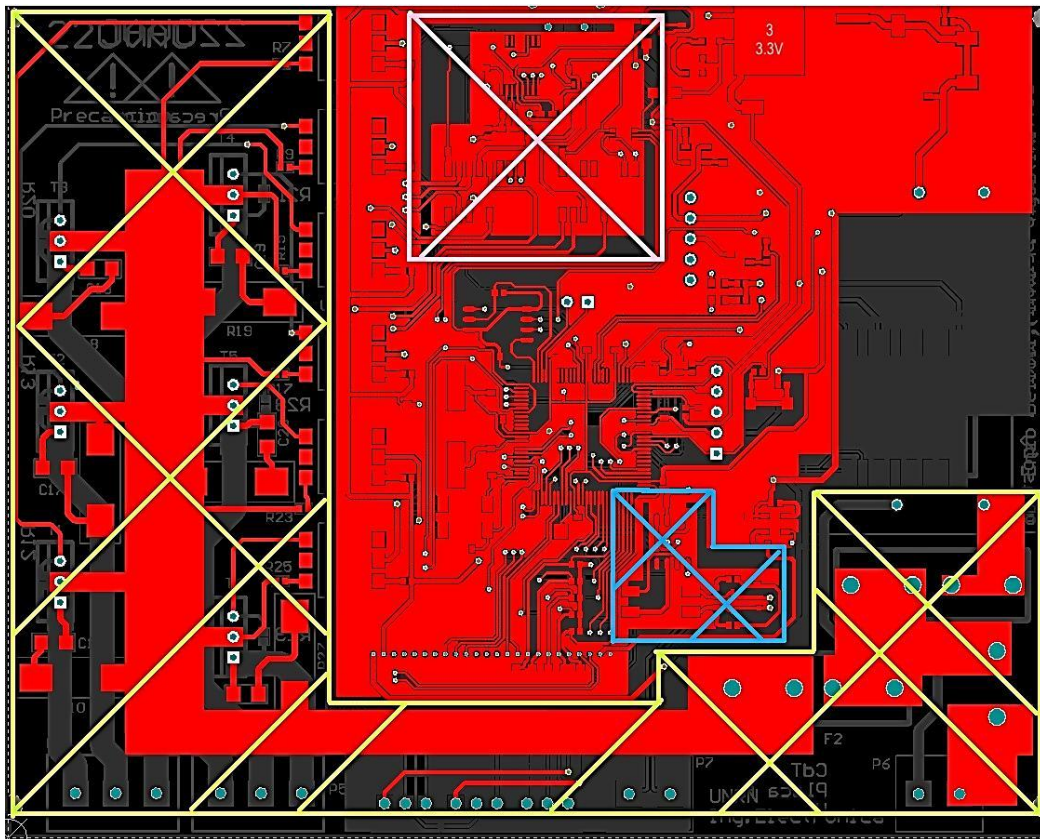


Figura VI-7: Placa A-TOP, Planos de tierra.

3.2. Placa B

La "placa B" también está fabricada en material FR-4 TG130, tiene un espesor de 1.6mm, 2 capas y una dimensión de 131,08 x 71,56 mm. Como esta es la placa del panel frontal, incorpora la pantalla, el joystick y el puerto USB tipo A. la interconexión entre placas se realiza con una cinta de cable de 24 conductores que esta soldada en P2 (Placa A) y P1 (Placa B).

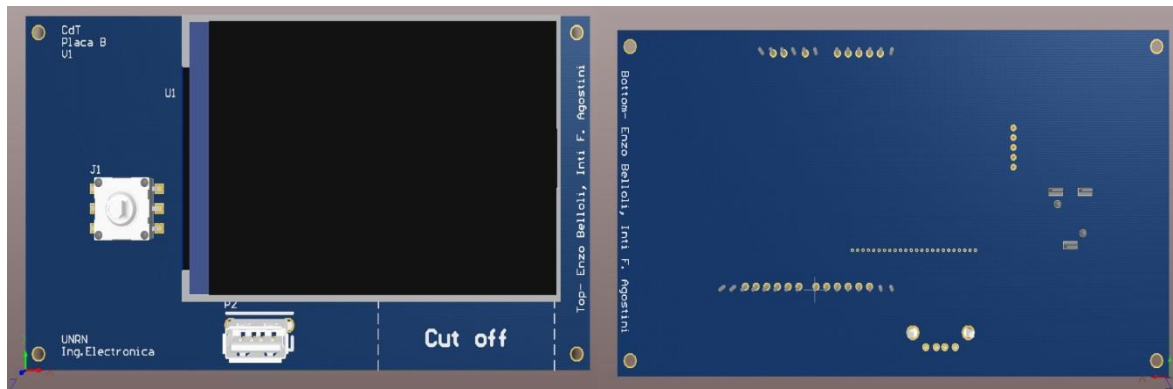


Figura VI-8: Placa B-TOP/BOTTOM.

4. Lista de componentes

Nombre ⁶	Descripción	Valor	Footprint	Código Componente	Código DIGIKEY	Cantidad
B1	Porta pila botón CR2032		SMD	MPD-Battery retainer	BK-912-CT-ND	1
C1, C4, C7	Capacitor electrolítico 16V	10uF	SMD	865080340001	732-8411-1-ND	3
C2, C8, C9, C10, C14, C15, C25, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C42, C43, C44, C45, C46	Capacitor cerámico 16V Y5V	0.1uF	SMD 0603	CL10F104Z08NNNC	1276-1258-1-ND	22
C3, C5, C23	Capacitor cerámico 10V X7R	1uF	SMD 0603	CL10B105KP8NNNC	1276-1946-1-ND	3
C6	Capacitor cerámico 25V C0G/NP0	0.01uF	SMD 0603	C0603C103K3GACTU	399-8987-1-ND	1
C11, C13	Capacitor cerámico 6,3V X7R	10nF	SMD 0201	GRM033R70J103KA01D	490-1262-1-ND	2
C12	Capacitor cerámico 10V X5R	100nF	SMD 0603	CL03A104KP3NNNC	1276-1115-1-ND	1
C16, C17, C18, C19, C20, C21	Capacitor cerámico 450V X7T	0.1uF	SMD 1206	C3216X7T2W104M160AA	445-14838-1-ND	6
C22	Capacitor cerámico 10V X5R	2.2uF	SMD 0603	CL10A225MP8NNNC	1276-1887-1-ND	1
C24	Capacitor cerámico 16V X7R	4700pF	SMD 0603	C0603C472K4RACTU	399-9091-1-ND	1
C26	Capacitor de tantalio 16V-3ohm	10uF	SMD 1206	293D106X0016A2TE3	718-1956-1-ND	1
C37	Capacitor electrolítico 16V	100uF	SMD	UWX1C101MCL1GB	493-2105-1-ND	1

⁶ El nombre por defecto es para la "Placa A", caso contrario se aclara.

C38, C40	Capacitor cerámico 10V C0G,NP0	33pF	SMD 0603	885012006005	732-7749-1-ND	2
C39, C41	Capacitor cerámico 10V C0G,NP0	18pF	SMD 0603	VJ0603A180FXQCW 1BC	720-1660-1-ND	2
D1, D5, D6, D7	LED 2V		SMD 0603	LTST-C190KFKT	160-1434-1-ND	4
D2	TVS diodo 5V15,5V		SC-47	SMS05T1G	SMS05T1GOSCT- ND	1
D3	TVS diodo 5V14V		SOD323	PESD5V0S1BA,115	1727-3837-1-ND	1
D4	TVS diodo 24V50V		SOT323-3	PESD1CAN-UX	1727-1306-1-ND	1
DS1, DS2	Diodo Schottky 20V 1A		SOD-123FL	NRVB120VLSFT1G	NRVB120VLSFT1 GOSCT-ND	2
F1, F2	Fusible vidrio 5mm	5A- 220V				2
I1	Filtro de modo común 2 líneas 100KHz 500mA 300mOhm	51uH	SMD	DR331-513AE	DR331-513AECT- ND	1
M1	Módulo Bluetooth		TH	HC-06		1
M2	Módulo WIFI		SMD	ESP-07/ESP 8266		1
	Antena WIFI 3dBi					1
P1	Conector USB- microAB	USB- uAB	SMD, ángulo recto,TH	475890001	WM17143CT-ND	1
P2	Cable plano 1' 28AWG		TH, 24 pins 1,24mm	3365/24 100	3M157996-1-ND	1
P3, P5	Bornera horizontal 3 posiciones		TH-3x5mm	1935174	277-1578-ND	2
P4, P8, P9	Bornera horizontal 3 posiciones		TH- 3x2.54mm	OSTVN03A150	ED10562-ND	3
P6, P7	Bornera horizontal 2 posiciones		TH-2x5mm	1935161	277-1667-ND	2
P10	Header, 5-Pin		HDR1X5			1
P11	Header, 2-Pin		HDR1X2			1
Q1	Mosfet Canal P 30V 3,8A		SOT23-3	DMP3099L-7	DMP3099L- 7DICT-ND	1
Q2	Transistor NPN 30V 0,6A		SOT23-3	MMBT2222LT1G	MMBT2222LT1G OSCT-ND	1
R1, R6, R8, R12, R13, R14, R16, R20, R21, R22, R24, R28, R29	Resistencia 0,5% 1/5W	220	SMD 0603	ERJ-PB3D2200V	P20205CT-ND	13
R2, R3, R4, R39	Resistencia 5% 1/4W	1K	SMD 0603	ESR03EZPJ102	RHM1.0KDCT-ND	4
R5	Resistencia 0,05% 1/10W	30K	SMD 0603	ERA-3ARW303V	P30KBECT-ND	1
R7, R9, R15, R17, R23, R25	Resistencia 5% 1/10W	330	SMD 0603	ERJ-3GEYJ331V	P330GCT-ND	6
R10, R11, R18, R19, R26, R27	Resistencia 5% 3W	39	SMD 4122	SMW339RJT	A103625CT-ND	6
R30, R36, R38, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51	Resistencia 1% 1/4W	10K	SMD 0603	ESR03EZPF1002	RHM10KADCT- ND	13
R31, R32, R33, R34	Resistencia 0,5% 1/10W	30	SMD 0603	RT0603DRE0730RL	311-2533-1-ND	4

R35	Resistencia 0,5% 1/5W	220	SMD 0603	ERJ-PB3D2200V	P20205CT-ND	1
R37, R40, R41	Resistencia 1% 1/8W	100	SMD 0603	RNCP0603FTD100R	RNCP0603FTD10 0RCT-ND	3
R52	Resistencia 5% 1/4W	100K	SMD 0603	ESR03EZPJ104	RHM100KDCT- ND	1
RST1	Pulsador 12V 0,05A		SMD	PTS645SM43SMTR9 2 LFS	CKN9112CT-ND	1
T1, T2, T3, T4, T5, T6	Triac-compuerta sensible 600V 12A		TH-TO- 220AB	BTA212-600D,127	1740-1324-ND	6
U1	Fuente Switching- entrada 85-264 VCA, salida 5V 2A		TH	IRM-10-5	1866-3032-ND	1
U2	Regulador de vol- taje lineal 3,3V 1,5A		D2PAK- 2lead	LD1086D2T33TR	497-1224-1-ND	1
U3	Regulador de vol- taje lineal 4,7V 150mA		SOT23-5	LP3985IM5- 4.7/NOPB	LP3985IM5- 4.7/NOPBCT-ND	1
U4	Supervisor Push- Pull		SOT143	APX811-46UG-7	APX811-46UG- 7DICT-ND	1
U5	Convertor de nive- les bidireccional		SOT765-1 (VSSOP8)	74LVCH2T45DC- Q100H	1727-7467-1-ND	1
U6	ADC 24bits		TSSOP16	ADS122U04IPWR	296-48886-1-ND	1
U7	Multiple- xor/Demultiplexor 2:1		SOT363	74LVC1G3157GW,12 5	1727-6073-1-ND	1
U8, U19	Multiple- xor/Demultiplexor 4:1		SOT109-1	HEF4052BT- Q100,118	1727-1041-1-ND	2
U9, U10, U11, U12, U13, U14	Opto acoplador salida TRIAC 5KV		SMD	MOC3063S-TA1	160-1723-1-ND	6
U15	Alimentación USB 500mA		SOT25	AP2822AKATR-G1	AP2822AKATR- G1DICT-ND	1
U16	Transceptor CAN FD 8MBps		SOIC8	MCP2557FD-H/SN	MCP2557FD- H/SN-ND	1
U17	Alta performance 80MHz MIPS-Base 32-bit flash micro controlador con USB, CAN y Ether- net, 128KB de memoria de pro- grama , 100 pin		TQFP- PF100_N	PIC32MX764F128L- I/PF	PIC32MX764F12 8L-I/PF-ND	1
U18	Memoria FLASH NOR 16Mb SPI		SOIC8	W25Q16JVSSIQ	W25Q16JVSSIQ- ND	1
Y1	Cristal 32768KHz		SMD	ABS25-32.768KHZ- 6-T	535-10240-1-ND	1
Y2	Cristal 12MHz		SMD	ABLS2-12.000MHZ- D4Y-T	535-9869-1-ND	1
Placa B-P2	Conector USB-A vertical		TH	USB-A1VSB6	ED2988-ND	1
Placa B-J1	Joystick 2 ejes +botón central 20mA 15V		SMD, TH	JS5208	EG4561-ND	1
Placa B-U1	Pantalla LCD TFT 3.5"		TH			1
	Botón ROCKER 20A 277V		Montaje en panel	C6000ALBB-1229W	1091-1034-ND	1
T1, T2, T3, T4, T5, T6	Disipador TO220 24°C/W			V7235-T	A10764-ND	6

Tabla VI-9: Lista de componentes

5. Gabinete

Se investigó la posibilidad de diseñar uno a medida y mandarlo a construir o en su defecto imprimirlo con una impresora 3D. El problema que supuso esta aproximación fue la relación costo-tiempo. Realizar el diseño para luego mandarlo a construir por el método de inyección de plástico tenía un costo demasiado elevado y la impresión en 3D suponía el diseño íntegro del gabinete, lo que conlleva una componente de tiempo también elevada, sin tener en cuenta también que si bien la terminación de este método es suficiente para un prototipo, lo ideal sería luego pasar el diseño a otro método de construcción para el producto final. Por las razones anteriores se decidió utilizar un gabinete prefabricado.

Realizada una búsqueda en el mercado local de gabinetes estándares y no se logró encontrar ninguno que se ajustara a los requerimientos por lo que se compró en el mercado internacional el gabinete de la figura V-7, ya que si bien existían varios gabinetes en el mercado nacional que podían albergar la pantalla, todos tenían dimensiones excesivamente más grandes.

Los criterios de selección estaban principalmente delimitados por el tamaño de la pantalla, pero de igual manera se tuvieron en cuenta los siguientes criterios;

- Volumen mínimo.
- Facilidad de instalación, preferentemente apto para riel DIN.
- Abertura para los conectores de los sensores, actuadores y alimentación.
- Plástico resistente.

Una vez elegido el gabinete, utilizando las medidas que nos proporcionó el fabricante se generó el modelo 3D y en base a este se procedió a dimensionar el PCB, y ubicar los componentes en el mismo, a continuación se pueden observar capturas del modelo.



Figura VI-9: Gabinete

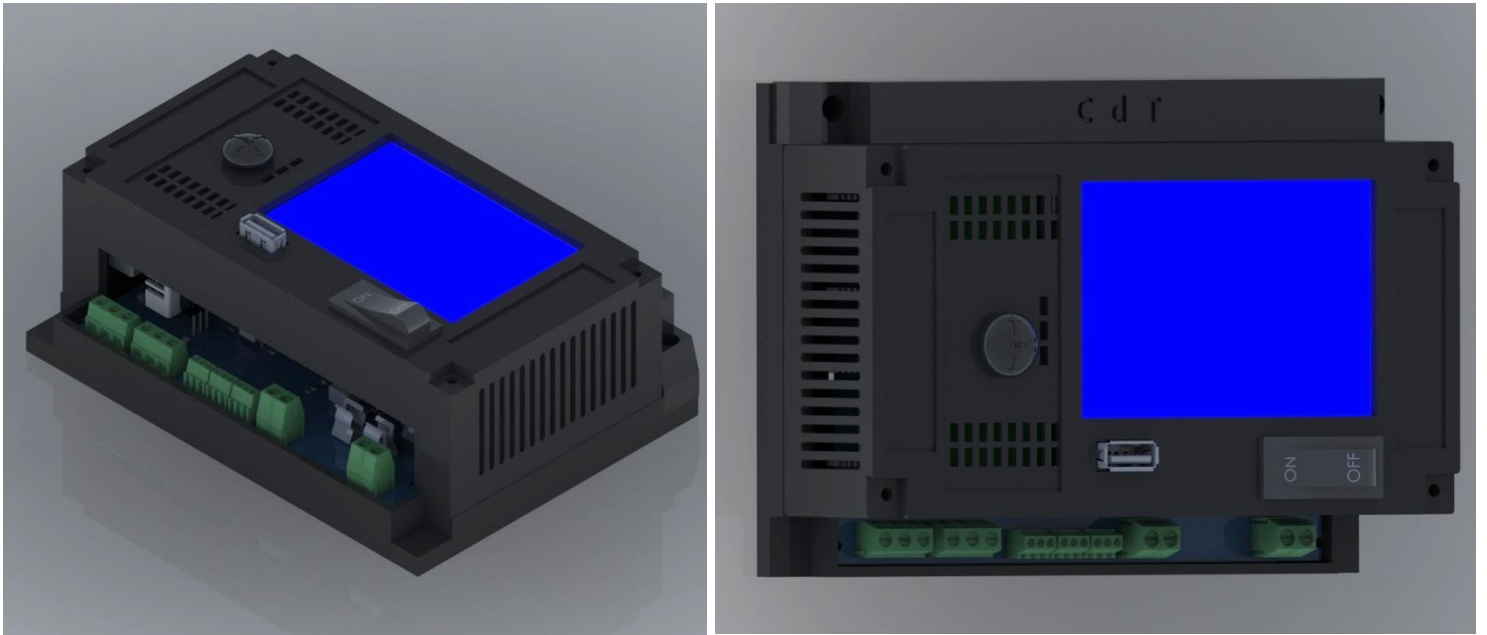


Figura VI-11: Representación del CdT

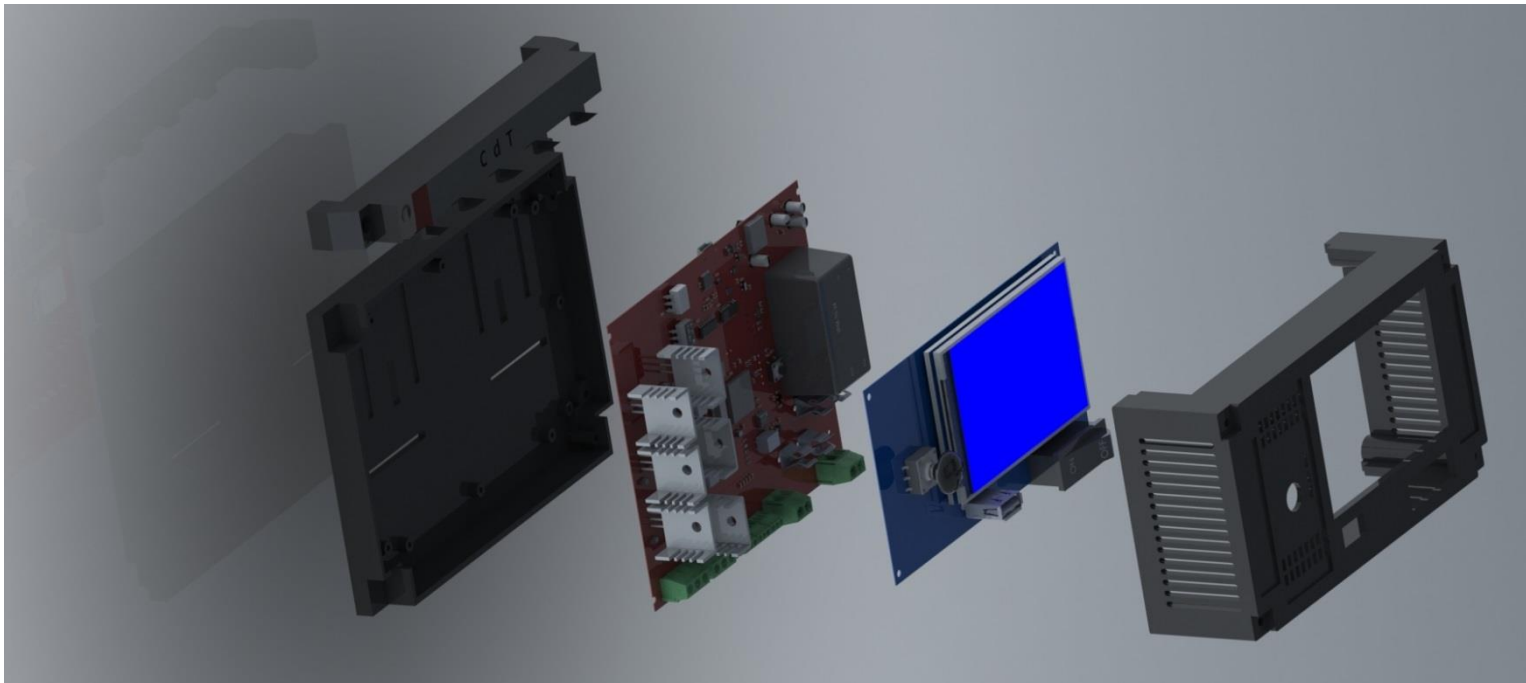


Figura VI-10: Representación vista explosionada del CdT.

VII - Costeo⁷

1. Productos similares en el mercado local

Se analizaron en el mercado local, en Mercado Libre, los precios de distintos controladores de temperatura y alguna de sus características principales.

Marca	Modelo	Dimensión [mm]	Tipo de sensor entrada	Método de control	Salida de control	Comunicación	Alarma	Rampa configuración	Costo
Thermosense Direct	PID CH902	96x96	Cualquiera: JKRS-BENT,RTD,Pt100, Cu50	-----	1-salida-Relé de estado solido	-----	-----	x	US \$99
Novus	Novus N321	75x33x75	pt 100,pt1000,j	PID	1-salida- Relé SPDT 1hp 250vac	RS485 mobdus RTU	x	-----	US \$100
Delta	DTA	48x48- Rail DIN	Cualquiera: JKRS-BENT,RTD,Pt100, Cu50	-----	1-salida- PID, ON-OFF,PWM	RS485 mobdus RTU,ASCII	x	x	US \$73
Omron	E5xC/E5xC-t	48x48x78		-----	1-salida-Relé, voltaje, corriente	RS485	-----	-----	US \$91
MAXWELL	CP48T-SSR	48x48x78	Termocupla universal	PID Fuzzy auto-tunning	1-salida-Relé de estado solido	-----	x	-----	US \$87
Blue Star	ETC 100	48x48x79	NTC 10K	PID	1-salida-Relé 0,5HP	-----	-----	-----	US \$37
Alre	STC-200	48x48x80	NTC 10K	PID	1-salida-Relé 0,5HP	-----	x	-----	US \$37
Elitech	ETC 200	48x48x81	NTC 10K	PID	1-salida-Relé 0,5HP	-----	x	-----	US \$44
Blue Star	MTC-5080	100x51x68, 5	NTC 10K	-----	1-salida-Relé 0,5HP	-----	x	-----	US \$46
ELIWELL	IDPLUS 961	74x32x59	PTC/NTC/PT100 0	-----	1-salida-1SPST 2HP	ModBus	x	-----	US \$47
Selec	TC513	48x48x81	Termocupla J,K,T,R,S/ RTD pt100	PID	1-salida-Relay /SSD	-----	-----	-----	US \$74

Tabla VII-1: Controladores de temperatura comerciales (11)

Se puede observar que estos controladores de temperatura comerciales comparten algunas características principales;

- Dimensiones estándar
- Tipo de control PID, si bien los fabricantes no especifican el método de control en algunos se puede inferir ya que requieren de una "calibración" para su primer funcionamiento.
- 1 sola salida del tipo RELE
- No es común que posean una herramienta de comunicación.
- No es posible cargar una rampa de configuración.
- No es posible relevar datos para su posterior análisis.

Si bien en el mercado argentino es posible encontrar otros tipos de controladores más avanzados también estos requieren de mayor instrucción por parte del operador o mayor infraestructura.

⁷ Para este capítulo se utiliza una cotización del dólar de \$40

2. Compra de materiales

A continuación se muestra el detalle de las compras realizadas para la fabricación de dos controladores de temperatura. Para algunos componentes se compraron componentes de repuesto dado la criticidad de los mismos.

Digikey

Se eligió comprar a Digikey por tener todos los componentes que se buscaban, si bien en ELEMÓN el costo de muchos de los componentes era menor, no disponían de todos ellos.

Nombre	Descripción	Código DIGIKEY	Cantidad necesaria	Cantidad comprada	Precio final[USD]
B1	Porta pila botón CR2032	BK-912-CT-ND	1	2	1,46
C1, C4, C7	Capacitor electrolítico 16V	732-8411-1-ND	3	6	1,02
C11, C13	Capacitor cerámico 6,3V X7R	490-1262-1-ND	2	10	0,33
C12	Capacitor cerámico 10V X5R	1276-1115-1-ND	1	10	0,38
C16, C17, C18, C19, C20, C21	Capacitor cerámico 450V X7T	445-14838-1-ND	6	12	6,80
C2, C8, C9, C10, C14, C15, C25, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C42, C43, C44, C45, C46	Capacitor cerámico 16V Y5V	1276-1258-1-ND	22	100	1,70
C22	Capacitor cerámico 10V X5R	1276-1887-1-ND	1	2	0,24
C24	Capacitor cerámico 16V X7R	399-9091-1-ND	1	2	0,20
C26	Capacitor de tantalio 16V-3ohm	718-1956-1-ND	1	2	0,66
C3, C5, C23	Capacitor cerámico 10V X7R	1276-1946-1-ND	3	10	0,65
C37	Capacitor electrolítico 16V	493-2105-1-ND	1	0	0,00
C38, C40	Capacitor cerámico 10V COG,NP0	732-7749-1-ND	2		0,00
C39, C41	Capacitor cerámico 10V COG,NP0	720-1660-1-ND	2	4	1,32
C6	Capacitor cerámico 25V COG/NP0	399-8987-1-ND	1	2	0,76
D1, D5, D6, D7	LED 2V	160-1434-1-ND	4	8	2,40
D2	TVS diodo 5V15,5V	SMS05T1GOSCT-ND	1	2	1,02
D3	TVS diodo 5V14V	1727-3837-1-ND	1	2	0,96
D4	TVS diodo 24V50V	1727-1306-1-ND	1	2	0,76
DS1, DS2	Diodo Schottky 20V 1A	NRVB120VLSFT1GOSCT-ND	2	4	1,84
I1	Filtro de modo común 2 líneas 100KHz 500mA 300mOhm	DR331-513AECT-ND	1	2	2,06
P1	Conector USB-microAB	WM17143CT-ND	1	2	1,48
P10	Header, 5-Pin		1		0,00
P11	Header, 2-Pin		1		0,00
P2	Cable plano 1' 28AWG	3M157996-1-ND	1	2	2,58
P3, P5	Bornera horizontal 3 posiciones	277-1578-ND	2	4	2,32
P4, P8, P9	Bornera horizontal 3 posiciones	ED10562-ND	3	6	6,18
P6, P7	Bornera horizontal 2 posiciones	277-1667-ND	2	4	1,64
Placa B-J1	Joystick 2 ejes +botón central 20mA 15V	EG4561-ND	1	2	6,96
Placa B-P2	Conector USB-A vertical	ED2988-ND	1	2	1,30

Q1	Mosfet Canal P 30V 3,8A	DMP3099L-7DICT-ND	1	2	0,88
Q2	Transistor NPN 30V 0,6A	MMBT2222LT1GOSCT-ND	1	2	0,26
R1, R6, R8, R12, R13, R14, R16, R20, R21, R22, R24, R28, R29	Resistencia 0,5% 1/5W	P20205CT-ND	13	30	3,25
R10, R11, R18, R19, R26, R27	Resistencia 5% 3W	A103625CT-ND	6	15	5,91
R2, R3, R4, R39	Resistencia 5% 1/4W	RHM1.0KDCT-ND	4	10	0,87
R30, R36, R38, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51	Resistencia 1% 1/4W	RHM10KADCT-ND	13	30	3,78
R31, R32, R33, R34	Resistencia 0,5% 1/10W	311-2533-1-ND	4	8	0,96
R35	Resistencia 0,5% 1/5W	P20205CT-ND	1		-
R37, R40, R41	Resistencia 1% 1/8W	RNCP0603FTD100RCT-ND	3	10	0,30
R5	Resistencia 0,05% 1/10W	P30KBECT-ND	1	2	1,56
R52	Resistencia 5% 1/4W	RHM100KDCT-ND	1	.	0,00
R7, R9, R15, R17, R23, R25	Resistencia 5% 1/10W	P330GCT-ND	6	15	0,81
RST1	Pulsador 12V 0,05A	CKN9112CT-ND	1		0,32
T1, T2, T3, T4, T5, T6	Disipador TO220 24°C/W	A10764-ND	6	12	2,23
T1, T2, T3, T4, T5, T6	Triac-compuerta sensible 600V 12A	1740-1324-ND	6	15	11,07
U1	Fuente Switching-entrada 85-264 VCA, salida 5V 2A	1866-3032-ND	1	2	17,42
U15	Alimentación USB 500mA	AP2822AKATR-G1DICT-ND	1	2	0,92
U16	Transceptor CAN FD 8MBps	MCP2557FD-H/SN-ND	1	2	1,62
U17	Alta performance 80MHz MIPS-Base 32-bit flash micro controlador con USB, CAN y Ethernet, 128KB de memoria de programa , 100 pin	PIC32MX764F128L-I/PP-ND	1	4	24,48
U18	Memoria FLASH NOR 16Mb SPI	W25Q16JVSSIQ-ND	1	2	1,08
U2	Regulador de voltaje lineal 3,3V 1,5A	497-1224-1-ND	1	3	2,76
U3	Regulador de voltaje lineal 4,7V 150mA	LP3985IM5-4.7/NOPBCT-ND	1	3	1,98
U4	Supervisor Push-Pull	APX811-46UG-7DICT-ND	1	2	0,76
U5	Convertor de niveles bidireccional	1727-7467-1-ND	1	2	1,12
U6	ADC 24bits	296-48886-1-ND	1	3	25,56
U7	Multiplexor/Demultiplexor 2:1	1727-6073-1-ND	1	2	0,74
U8, U19	Multiplexor/Demultiplexor 4:1	1727-1041-1-ND	2	4	2,04
U9, U10, U11, U12, U13, U14	Opto acoplador salida TRIAC 5KV	160-1723-1-ND	6	15	8,43
Y1	Cristal 32768KHz	535-10240-1-ND	1	2	0,80
Y2	Cristal 12MHz	535-9869-1-ND	1	2	0,66
	Botón ROCKER 20A 277V	1091-1034-ND	1	2	4,22
Subtotal					173,81 USD

Tabla VII-2: Lista de compra a Digikey

Aliexpress

Nombre	Descripción	Código Componente	Cantidad	Precio Unidad	Precio final[USD]
M1	Módulo Blue-tooth	HC-06	2	2,92	5,84
M2	Módulo WIFI	ESP-07/ESP 8266	2	1,99	3,98
M2	Antena WIFI 3dBi		2	0,50	1,00
Placa B-U1	Pantalla LCD TFT 3.5"		2	9,88	19,76
	Gabinete	szomk -AK-P-13	3	6,80	20,40
Subtotal					30,58 USD

Tabla VII-3: Lista de compras en Aliexpress

Compra de PCBs

Los circuitos impresos fueron construidos en PCBWay, una empresa de origen Chino.

Nombre	Descripción	Cantidad	Precio unidad	Precio final [USD]
Placa A	105x128mm, grosor 1,6mm, 2 capas, cobre 1 oz	10	4	40
Placa B	71,56x131,08mm, grosor 1,6, 2 capas, cobre 1 oz	10	3,2	32
Stencil	SMD stencil frameworkd 370x470mm	1	15	15
Subtotal				87 USD

Tabla VII-4: Lista de compra a PCBWay

Compra varias

Algunos elementos para el armado del dispositivo fueron adquiridos en comercios locales

Nombre	Descripción	Cantidad	Precio unidad	Precio final[US \$]
F1, F2	Fusible vidrio 5mm	2	5	0.25
Cable	1,5mm unipolar Rojo	0,5m	7\$/m	0.087
Cable	1,5mm unipolar Negro	0,5m	7\$/m	0.087
Terminal	metal	4	2	0.2
Termo contraíble	2mm	0,5m	5\$/m	0.062
Cortes láser	Cortes del gabinete	2	60	3
Tornillos	Tornillos para plástico	10	2	0.5
Subtotal				US \$4.19

Tabla VII-5: Lista de compra a comercios locales

3. Costo final del producto

El costo final del dispositivo se conforma de tres partes;

- Insumos
- Materias primas
- Mano de obra

Los insumos son todos aquellos materiales necesarios para la construcción (sin tener en cuenta las herramientas). Este ítem puede considerarse que no varía considerablemente en función de la cantidad de dispositivos que se quieren producir. A término de realizar una proyección simplificada se lo considera constante.

Las materias primas son el principal costo variable, los costos bajan considerablemente en función de la cantidad producida.

La mano de obra es el costo variable que predomina a la hora de aumentar la producción de dispositivos.

Costo de producción					
Insumos					
Nombre	Descripción	Precio			
Alcohol isopropilico	Aerosol-440cc	\$ 160,00			
Estaño en pasta	XG-50 50g	\$ 350,00			
Estaño 0,5mm	250g	\$ 150,00			
Malla desoldante	2mmx1m	\$ 80,00			
Sub total					\$ 740,00
Materias primas					
Gasto	Descripción	Precio	Envío	Impuestos	Subtotal
Digikey	Componentes varios	\$ 6.952,40	\$ 1.600,00	\$ 1.650,00	\$ 10.202,40
Aliexpress	Módulos Arduino + gabinete	\$ 1.223,20	\$ 0,00	\$ 751,60	\$ 1.974,80
PCBWay	PCBs	\$ 3.480,00	\$ 2.400,00	\$ 360,00	\$ 6.240,00
Local	Elementos varios	\$ 167,50	\$ 0,00	\$ 0,00	\$ 167,50
Sub total					\$ 18.584,70
Mano de obra					
Ítem	Valor				
Costo de la hora hombre	\$200				
Horas necesarias	12				
Total	\$ 2.400,00				
Sub total					\$ 4.800,00
Total					\$ 21.724,70 (US \$543.11)

Tabla VII-6: Costo de producción

En la tabla VI-6 se puede observar los valores para la producción de dos dispositivos. En base a dos criterios se armó la tabla, el primero fue el de incluir en el ítem de materias primas la compra de componentes críticos para repuesto y la compra de diez PCBs. El segundo fue el de incluir 12 horas de mano de obra (es decir un día y medio laboral) ya que se considera el armado "artesanal" del sistema para esta simplificación.

Proyección								
Cantidad	Digikey	Aliexpress	PCBWay	Local	Insumos	Mano de obra	Total	Total por unidad
1	\$ 5.101,20	\$ 987,40	\$ 288,00	\$ 83,75	\$ 740,00	\$ 2.400,00	\$ 9.600,35	\$ 9.600,35(US\$240)
2	\$ 10.202,40	\$ 1.974,80	\$ 576,00	\$ 167,50	\$ 740,00	\$ 4.800,00	\$ 18.460,70	\$ 9.230,35(US\$230)
10	\$ 24.171,60	\$ 9.174,00	\$ 2.880,00	\$ 335,00	\$ 740,00	\$ 24.000,00	\$ 61.300,60	\$ 6.130,06(US\$153)
100	\$ 155.244,04	\$ 91.740,00	\$ 14.400,00	\$ 670,00	\$ 1.480,00	\$ 240.000,00	\$ 503.534,04	\$ 5.035,34(US\$125)

Tabla VII-7: Proyección simplificada de producción

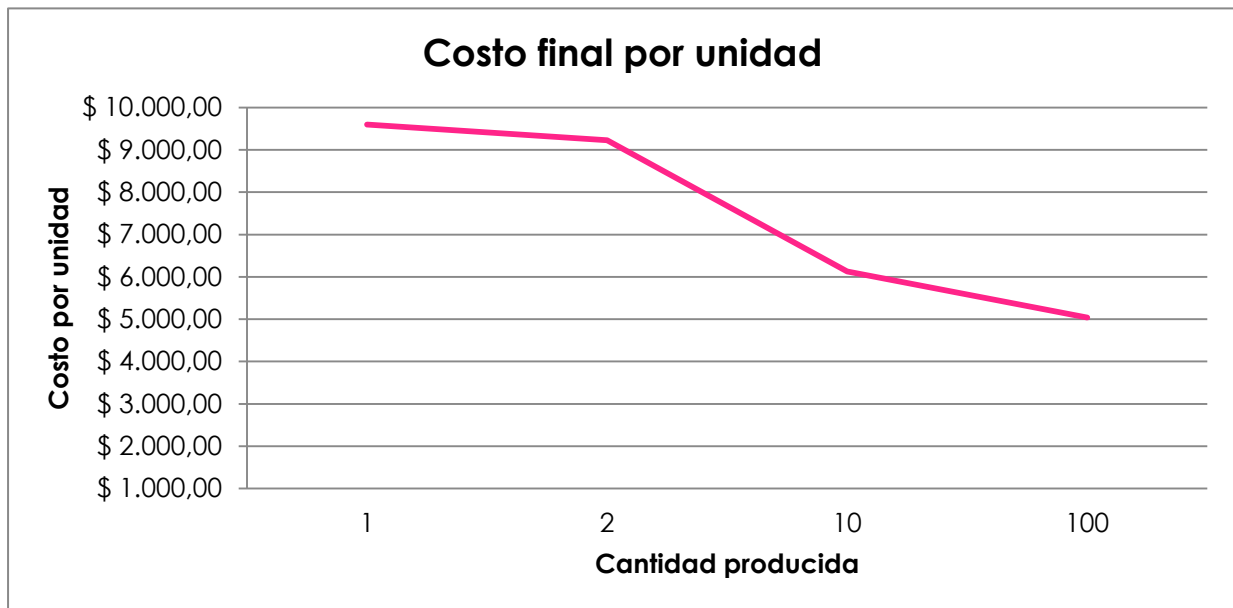


Figura VII-1: Costo final por unidad

Tanto en la tabla VI-7 como en la figura VI-1 se puede observar como el costo final del producto disminuye con la cantidad producida y que el salto es significativo a partir de 10 unidades.

Para poder analizar desde otro punto de vista los costos en la figura VI-2 se puede observar como el porcentaje de la materia prima desciende y el costo de la mano de obra se incrementa porcentualmente en función de la cantidad.

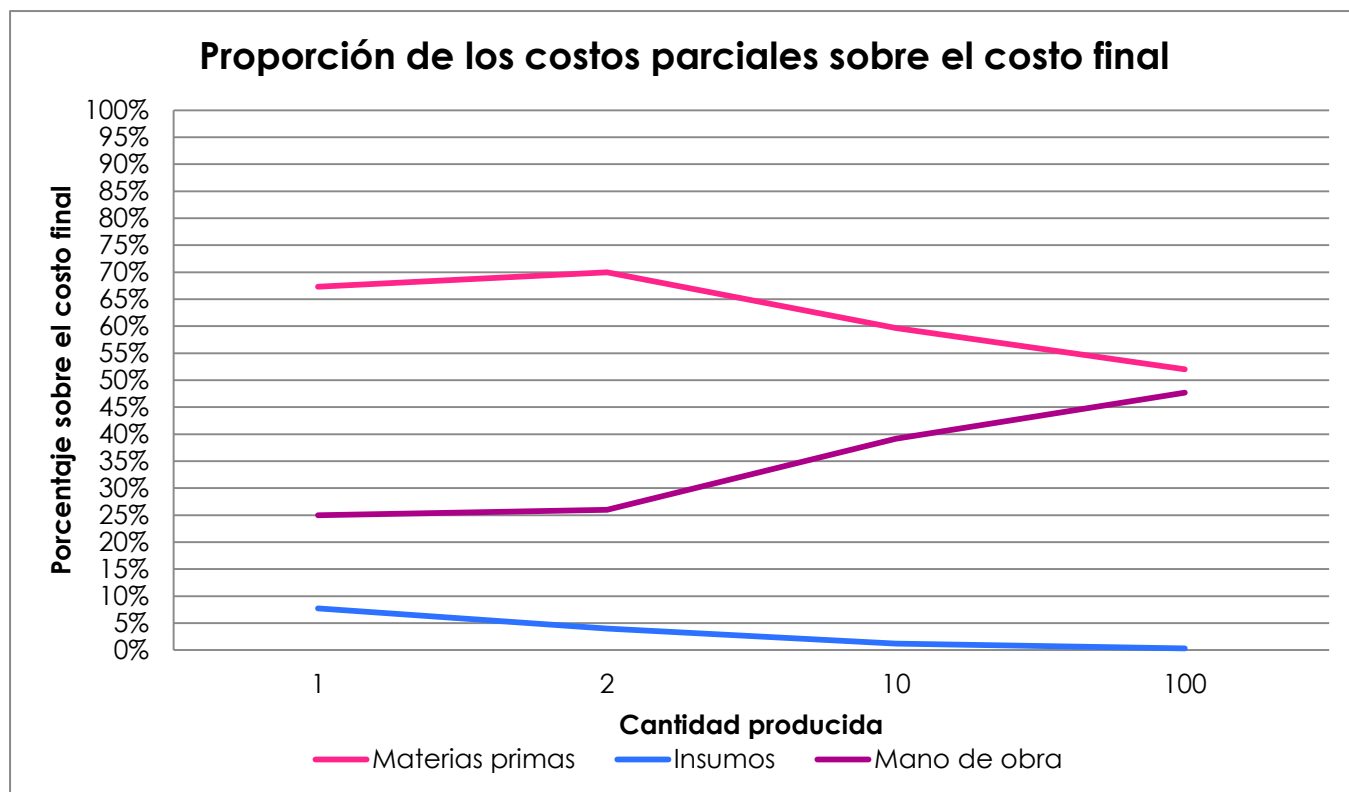


Figura VII-2: Evolución de los costos frente a diferentes cantidades producidas

Precio y ventajas

Dado el elevado costo que tiene la producción (incluso de 10) es necesario realizar un análisis más exhaustivo para llegar a un índice de ganancia deseado.

El mercado local ofrece controladores muy accesibles con precios que promedian los \$1600, que ofrecen 1 única salida de control y la capacidad de leer sensores de temperatura NTC. Luego se pasa a controladores que promedian los \$3500, ofrecen mayor robustez, un abanico más amplio de sensores admitidos, comunicaciones industriales, salida de control con relé de estado sólido. Algunos de estos también permiten utilizar curvas de seguimiento de temperatura y admiten salidas del tipo PWM. De aquí saltamos a PLC cuyos costos van desde unos cuantos miles de pesos a decenas de miles de pesos, que si bien no es exactamente esta línea de producto, puede ser parte de una solución.

El CdT incorpora múltiples salidas y entradas (3 en total), múltiples comunicaciones inalámbricas, la funcionalidad de "datalogger", no requiere ningún tipo de programación como podría ser un PLC y la configuración es simple; no existe producto similar en el mercado argentino.

Suponiendo un margen de ganancia elevado, como podría ser 60%, en la producción de 10 controladores, el precio final sería de \$9800 (US\$ 245), obteniéndose una ganancia neta de \$3678 (US\$ 92) por producto.

VIII - Software

En este capítulo se muestra la estructura del software implementado en el CdT para la prueba del prototipo realizada en una cervecería local. La estructura del código está diseñada en base a la generada por la herramienta de Microchip, Harmony V2.0.6. Dicha herramienta es utilizada para la configuración de los puertos, el reloj, las interrupciones y los periféricos.

1. Arquitectura del software

El software implementado para la prueba del algoritmo de control en la cervecería se valió del uso del módulo Bluetooth HC-05 y el módulo de wifi Esp8266 para implementar una interfaz con el usuario, ya que hasta el momento la pantalla no estaba funcional. Además de las dos interfaces antes mencionadas se utilizó la UART 4 y el puerto USB Device, para poder cargar la configuración inicial del controlador y descargar los datos adquiridos una vez finalizara la prueba.

El sistema utiliza una máquina de estados que controla las diferentes etapas de funcionamiento. Previo a entrar a esta máquina de estados se verifican y configura todo lo necesario para el correcto funcionamiento del CdT en la función **APP_Initialize()**, a continuación se puede observar el diagrama general de la estructura del software.

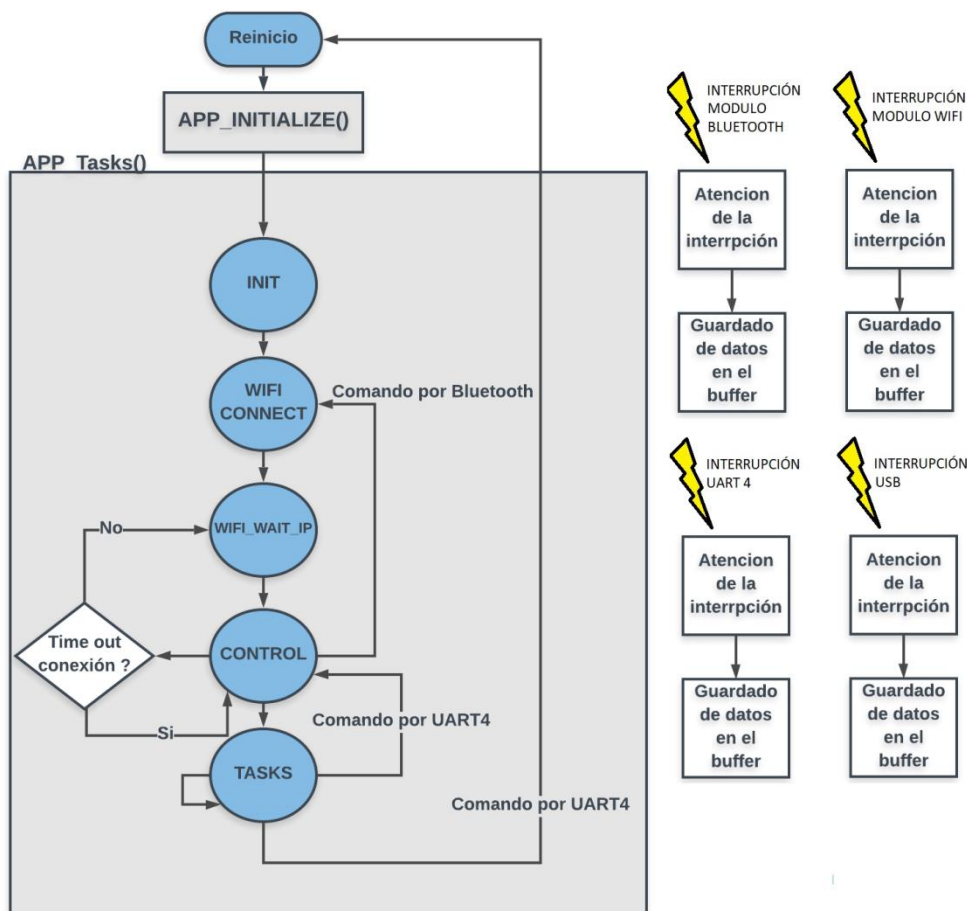


Figura VIII-1: Diagrama de flujo del software implementado en el prototipo.

Descripción de las funciones principales:

APP_Initialize(): esta función inicializa todos los periféricos con sus configuraciones correspondientes, y carga desde memoria la configuración del dispositivo, en el caso de que no haya una configuración válida se crea una configuración con los valores por defecto.

APP_Tasks(): contiene la máquina de estados que maneja todas las funcionalidades del software implementado, a continuación se muestra una tabla con la descripción de cada estado.

Estado	Descripción
INIT	Inicializar los relojes del sistema, utilizados para controlar la frecuencia de ejecución del control y verificar cuando se produce un time-out en alguna función perteneciente a las distintas librerías de periféricos.
WIFI_CONNECT	Envía al módulo ESP8266 (Wifi) el comando y la información necesaria para conectarse a la red Wifi guardada en configuración, y pasa al estado WIFI_WAIT_IP para esperar una IP válida.
WIFI_WAIT_IP	Comprueba si el dispositivo tiene una IP asignada, de ser así se configura el módulo ESP8266 para actuar como servidor, de lo contrario retorna al estado IDLE.
SERVICE_TASK	Como puede verse en la Figura VIII-1 , este modo es solo accesible mediante un comando por UART, esto se debe a que es un estado utilizado para la configuración del dispositivo y la descarga de datos por USB y en esta etapa estas funcionalidades no estaban permitidas para el usuario.
IDLE	Este estado es el encargado de ejecutar las rutinas de muestreo, control, mantener el servidor web y la comunicación Bluetooth.

Tabla VIII-1: Descripción de los estados de la función APP_Tasks().

La secuencia de funcionamiento del dispositivo una vez entro al estado IDLE, es la siguiente, cada un segundo se envía al ADC el comando para muestrear la resistencia, una vez obtenido se calcula la temperatura utilizando los parámetros A,B y C previamente obtenidos con las ecuaciones de la sección III.2.4.1 (caracterización de los sensores). La temperatura calculada se almacena en un vector con capacidad para 60 muestras, este proceso se repite hasta que el vector este completo es decir, que haya pasado un minuto. En este punto se ejecuta el algoritmo de control el cual realiza un promedio de las 60 muestras y el valor resultante lo pasa por un filtro IIR, para luego realizar el proceso de fuzzyficación/evaluación de reglas/defuzzyficación y finalmente se actúa en base al valor resultante.

2. Interfaces de usuario

2.1. Servidor Web

Una vez que el módulo esp8266 está conectado al wifi y es correctamente configurado como servidor, toda petición HTML que recibe es enviada hacia el microcontrolador, esto activa la interrupción de la UART y todos los caracteres de la petición son guardados en un buffer circular. El buffer es revisado periódicamente en el estado IDLE, y en caso de contener información, esta es procesada y en caso de detectarse una petición es

respondida, a continuación se muestra la lista de peticiones que se utilizan el software implementado:

- **GET_HTTP:** Solicitud del código fuente de la página web, se dispara cuando un usuario ingresa la IP del controlador en un navegador web. Al detectar esta petición se carga desde la memoria flash el código fuente y se envía en respuesta.
- **INIT:** Una vez cargada la página se solicita por medio de esta petición los valores iniciales de temperatura del fermentador y del setpoint.
- **GET_TEMP:** Solicitud de la temperatura actual del fermentador, esta solicitud es realizada periódicamente por la página web con el fin de actualizar los indicadores y el gráfico de la temperatura.
- **UP_ONE/ DOWN_ONE/ UP_DEC/ DOWN_DEC:** Esta solicitud es disparada por los botones que permiten al usuario incrementar o decrementar el valor del setpoint en un grado o una décima de grado. Al recibir esta solicitud, el controlador actualiza el valor del setpoint y guarda el mismo en memoria para que ante un corte en el suministro eléctrico al reiniciarse el dispositivo mantenga el setpoint correctamente. Luego se envía como respuesta el valor de setpoint a la página web y esta actualiza el indicador, de esta forma se asegura que la página siempre muestre el valor correcto de setpoint.
- **DATE:** Esta petición es enviada por la página cuando esta es cargada inicialmente por el navegador, seguida de la fecha y hora. El propósito de la misma es dejar en los registros la hora ya que al momento de realizar la prueba del sistema de control el circuito que alterna la fuente de alimentación entre los 220V y la pila no estaba funcional, por lo que en caso de un corte de luz no podíamos mantener la hora.

La página fue creada utilizando RGraph una librería de JavaScript para realizar gráficos, la cual contiene una variedad de tipos de gráficos personalizables y puede ser usada off-line, a esto se le sumó AJAX (Asynchronous JavaScript And XML) para poder mediante peticiones solicitar datos al servidor luego de haber cargado la página y actualizar elementos de la página sin tener que recargarla en su totalidad.

En la Figura VIII-2 se puede observar una captura de la página implementada para el prototipo, esta cuenta con dos indicadores para mostrar al usuario la temperatura y el setpoint del controlador, a la derecha de los indicadores los cuatro botones que permiten modificar el setpoint, y en la parte inferior un gráfico que muestra la evolución de la temperatura en las últimas 24hs.

Controlador de temperatura

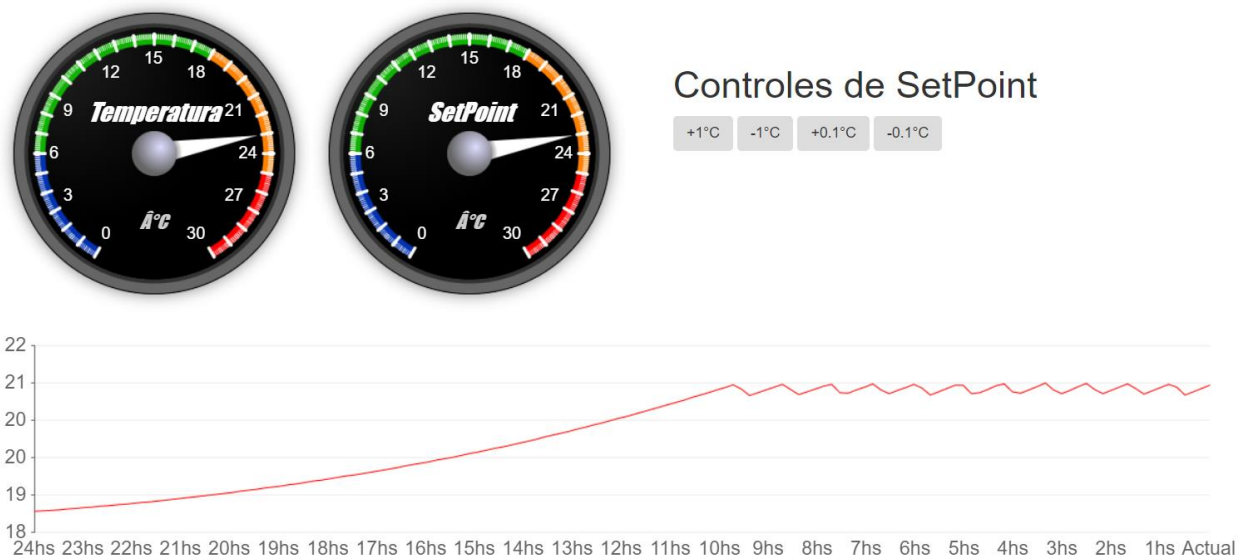


Figura VIII-2: Página web del prototipo.

2.2. Conexión Bluetooth

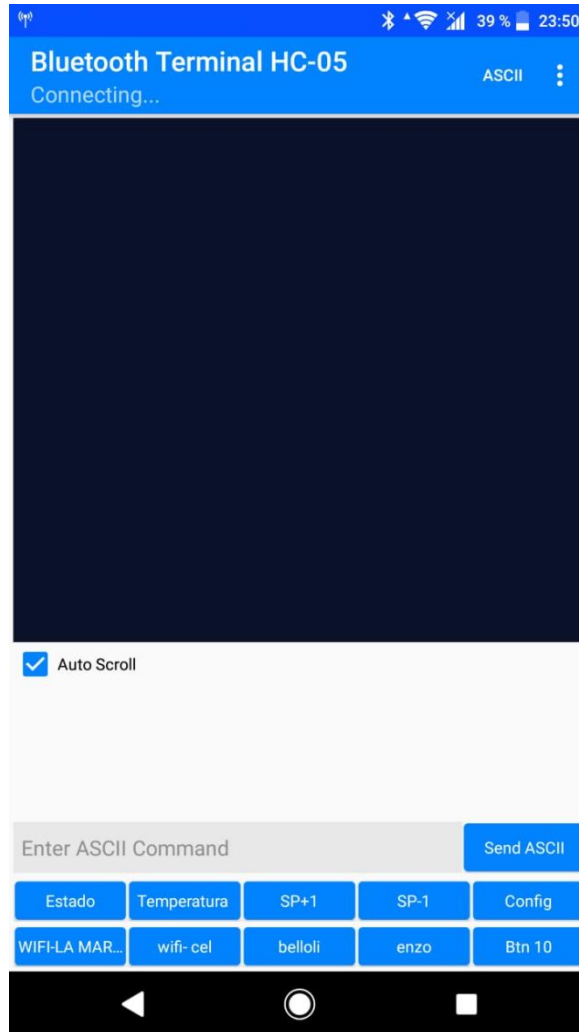
La interfaz Bluetooth se implementó para mostrar al usuario la IP del dispositivo asignada por el enrutador luego de la conexión del mismo a la red wifi, y como una alternativa para utilizar en el caso de que el servidor web no funcionase. Al igual que el módulo de wifi el módulo HC-05 una vez conectado a un dispositivo Bluetooth retransmite todos los caracteres recibidos al microcontrolador por medio de la UART 1, estos datos son almacenados en un buffer ante el disparo de la interrupción y luego en el estado IDLE, el buffer es verificado periódicamente.

Se implementaron los siguientes comandos:

- Comando "Estado": muestra por la pantalla del celular el estado general del control, temperatura, setpoint y si está conectado a una red WiFi, la dirección IP.
- Comando "Temperatura": muestra por pantalla la temperatura filtrada y la temperatura instantánea.
- Comando "SP+1": Incrementa el setpoint en 1°C.
- Comando "SP-1": Decrementa el setpoint en 1°C.
- Comando "Cambio de red": se configuraron 4 redes por defecto.
- Comando "Modo de configuración": Cuando este código es enviado por Bluetooth se cambia el contexto a WIFI_CONNECT. Si previamente a enviar este comando se envía por la UART 4 el carácter 0, el cambio de contexto se realiza a SERVICE_TASKS.

Para enviar los comandos desde un celular se utilizó la aplicación "**Bluetooth Terminal HC-05**", la cual permite diferentes botones para que envíen un carácter determinado, si bien se planea desarrollar una aplicación propia para el dispositivo final en esta instancia se priorizó reducir los tiempos para poder realizar la prueba del sistema de control por lo que se eligió utilizar la aplicación antes descrita. A continuación se puede observar una captura de la

aplicación que utilizaba el usuario para controlar el dispositivo por medio de la



conexión Bluetooth.

Figura VIII-3: Interfaz de la aplicación Bluetooth usada para controlar el prototipo.

3. Estructura de la memoria

El dispositivo cuenta con una memoria SPI flash de 16Mbits marca windbond modelo W25Q16JV, la misma esta subdividida en bloques de 4KB los cuales son la mínima porción de memoria que puede ser borrada. Debido a que la tecnología de la memoria solo permite escribir datos en las posiciones de memoria que hayan sido previamente borradas y que solo se pueden borrar secciones de 4KBytes, 64KBytes o el chip completo, fue que se decidió implementar ese tipo de estructura.

Además de la memoria principal de 16Mbits el chip cuenta con 3 registros especiales de 256 bytes, llamados registros de seguridad, los cuales son independientes de la memoria principal y pueden ser borrados individualmente.

En la memoria es necesario almacenar los siguientes datos:

- Configuración básica del dispositivo, como puede ser credenciales y nombre de la red wifi, nombre de la cervecería, contraseña Bluetooth, etc.

- Configuración de los fermentadores a controlar
- Parámetros de sensores ingresados por el usuario
- Registros de temperatura
- Curvas de setpoint
- Código fuente de la página web

La memoria principal se utilizara dividida en bloques de 64KBytes, y uno de los registros de seguridad se utilizara como un índice ya que al tener 256 bytes es posible mantener un registro de toda la memoria principal.

El índice contendrá por cada bloque un identificador, que permitirá conocer si el bloque está ocupado y que tipo de registro contiene, el primero de los bloques de 64KB estará reservado siempre para la configuración del dispositivo, y cada uno de los bloques de 4KB que lo componen estará almacenado la información de:

- La configuración de los fermentadores a controlar
- Parámetros de sensores
- Configuración Wifi
- Configuración Bluetooth

Por lo tanto la primera posición del índice siempre deberá estar ocupada por el identificador **DEVICE_CONFIGURATION**, en el caso de que este identificador no esté presente se sabrá que la memoria no contiene información valida.

El resto de los bloques estará destinado a almacenar tres tipos de registros, registros de temperatura, curvas de setpoint y la página web, cada uno de los cuales tendrá su propio encabezado con información pertinente según el tipo de registro.

En los registros de temperatura y de setpoint, con identificadores **TEMP_REG** y **SETPOINT_REG** respectivamente, es necesario almacenar valores de temperatura, con una resolución de 0,01 °C en el rango de 0°C a 99°C, y con una frecuencia de muestreo de 1 minuto. Si bien esta resolución y frecuencia de muestreo son superiores a las necesarias para un proceso de variaciones lentas como es la fermentación al ser la primera prueba del dispositivo preferimos utilizar estos valores ya que la cantidad de memoria no era un limitante y podríamos observar con mayor detalle las variaciones en la temperatura ambiente, la cual también sería registrada utilizando esta misma estructura.

Dado que para cada muestra se utilizaron 2 bytes un bloque de 64KB se podrían guardar aproximadamente 32k muestras, lo que equivaldría a 23 días, si bien esta duración sería suficiente para registrar el proceso de fermentación en vista de que el usuario desee registrar también el proceso de maduración el cual suele durar dos semanas, se decidió extender los registros de temperatura y setpoint a dos bloques de 64KB pudiendo registrar 46 días con una frecuencia de muestreo de un minuto.

A los registros que contiene la página web, con identificador **WEB_REG** se los decidió ubicar en los últimos bloques de la memoria, esto permite cambiar fácilmente la cantidad de bloques usados sin tener que reubicar registros de temperatura, dado que la cantidad de memoria disponible es suficiente para almacenar 16 registros de temperatura y por lo tanto es muy probable que las últimas posiciones no se utilicen con este fin.

A continuación se muestra un diagrama del índice y la memoria principal con una hipotética distribución de registros, para ilustrar mejor lo antes descrito.

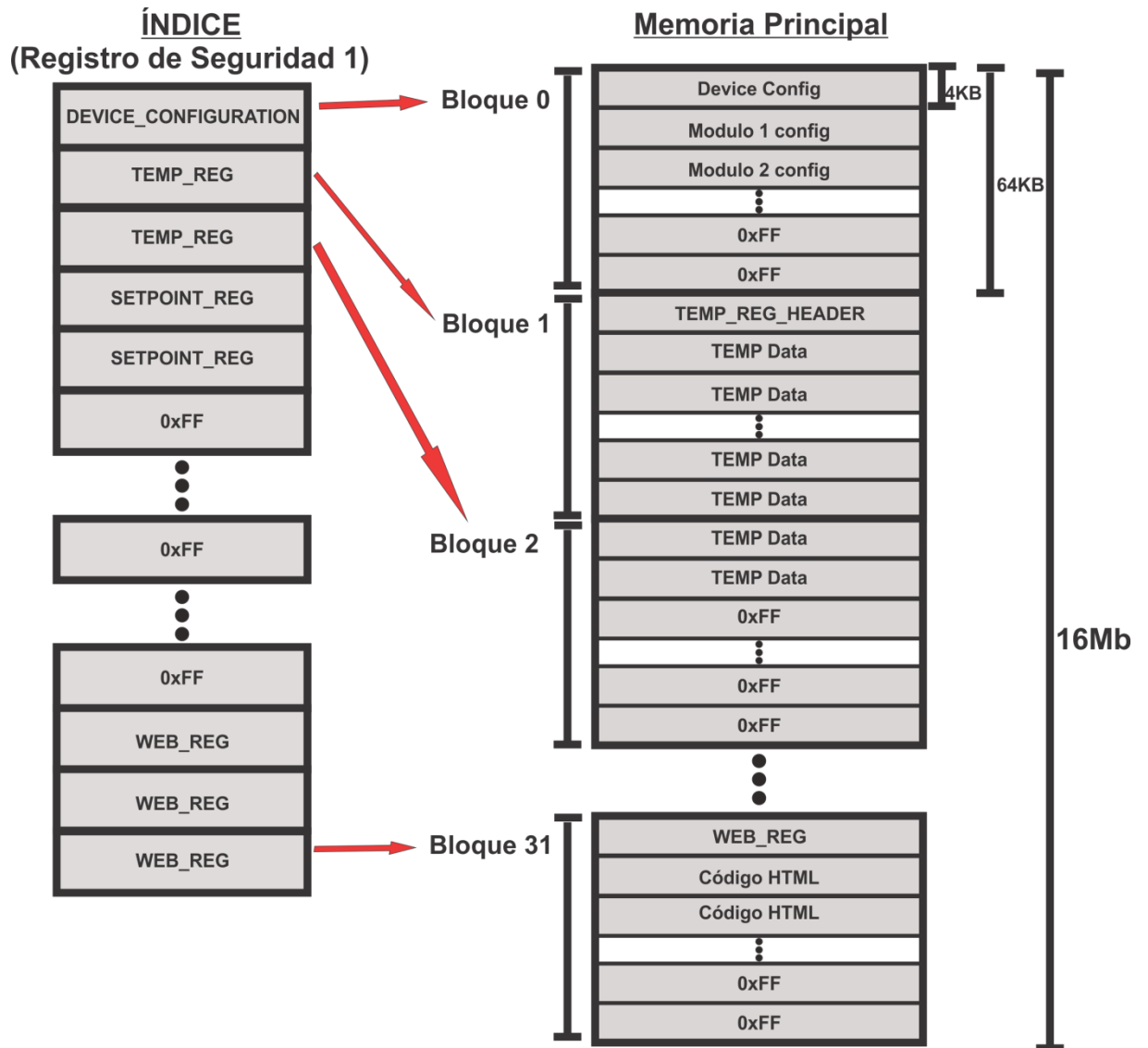


Figura VIII-4: Estructura implementada en la memoria.

IX - Conclusiones

Como se puede observar en el presente documento, se logró diseñar y fabricar un prototipo funcional de un controlador de temperatura para fermentadores de cerveza, el mismo fue probado en producción en una cervecería local y se obtuvieron resultados satisfactorios lo que dio por concluido el objetivo global de este trabajo.

Respecto a la investigación se reunió información suficiente para generar un modelo matemático que tuviera en cuenta los principales fenómenos fisicoquímicos que ocurren durante la fermentación, para luego en base a este modelo diseñar el control y verificar en simulación su desempeño, demostrando la capacidad de tomar un problema de la vida real, analizarlo y simplificarlo aislando la información relevante para el desarrollo del proceso de diseño. Para el algoritmo de control se desarrolló una biblioteca de control difuso genérica que posee la flexibilidad necesaria para modificar los parámetros de funcionamiento tanto para el CdT como para cualquier otro sistema donde se quiera utilizar este método de control, lo que posibilitaría en un futuro utilizar el controlador en otro tipo de planta.

Los resultados de la experiencia en la planta muestran que el modelo es fiel a los datos recolectados, además las pruebas realizadas sobre el mismo permitieron validar la implementación en el controlador. Debido a los inconvenientes que surgieron durante las experiencias en la planta no se pudo observar el accionar del controlador durante la etapa de fermentación pero de todas maneras, con la validación del modelo sumado a las pruebas de laboratorio del control y el accionar observado durante la etapa de maduración en la segunda experiencia en la planta se puede inferir que el controlador trabaja de forma correcta. Como se pudo observar en las simulaciones la implementación final del controlador se adapta a diferentes tipos de escenarios sin la necesidad de que el operador deba configurar parámetros de control, así es que por ejemplo, un cambio en el volumen del fermentador o de las condiciones de aislación no afectan al normal funcionamiento del CdT.

Con respecto al diseño de la plataforma se logró obtener un prototipo operativo mediante la metodología Top-Down, en la cual partiendo de la definición de los requerimientos del dispositivo se realiza una división del mismo en los distintos sistemas y subsistemas, realizando el diseño de cada sistema y la

correspondiente selección de componentes. Finalmente se diseña el PCB para la posterior fabricación, ensamblado y puesta en marcha. Todas las etapas anteriores fueron realizadas y los errores de diseño fueron debidamente detallados e instruidos para subsanarlos en el apartado de “puesta en marcha” de la placa.

Además se hizo un pequeño análisis de mercado a modo orientativo que no constaba en los objetivos iniciales, pero que entendimos ayuda a ubicar el prototipo en el mercado real de Argentina. Es un producto que no existe y que brindaría un mayor control y modernización a la fábrica que decida utilizarlo.

El apartado de software merece un análisis a parte. En principio se establecieron metas, que debido a la falta de experiencia, fueron demasiado grandes. La programación de aplicaciones tanto para el celular como la computadora requiere mucho más tiempo del que le asignamos en nuestro plan de trabajo por lo que no fue posible lograr al 100% este objetivo. De todas maneras se comprobó que tanto la conexión Bluetooth como WiFi funcionan correctamente y se puede establecer una comunicación desde el celular o la computadora. Se diseñó como contrapartida al objetivo, un servidor web en la plataforma. Este permite acceder remotamente al CdT, visualizar información y configurar algunos parámetros. En tanto el firmware de la plataforma se programó todo lo necesario para realizar pruebas de campo y validar los distintos subsistemas necesarios para esto.

Objetivos a futuro.

- Rediseño de placas atendiendo a las distintas fallas encontradas.
- Cambiar el método de conexión entre la placa A y la B a un conector, para disminuir los tiempos de armado del dispositivo (actualmente es un cable soldado).
- Mejorar y actualizar la programación para agregar características de utilidad para el usuario, como por ejemplo implementar un menú de configuración vía pantalla/ joystick para el usuario.
- Evaluar en el rediseño el agregado de una salida o varias pwm para poder controlar otro tipo de actuadores.

- Se encontró que la opción del servidor Web es mucho más pragmática a la hora de establecer una comunicación rápidamente. Se debería entonces enfocar los esfuerzos en la interconexión de las plataformas vía la red y/o utilizando el CAN BUS.
- Analizar la opción de diseñar versión de la plataforma que no incluya la pantalla y así disminuir costos y volumen.
- Diseñar/encontrar un gabinete que sea accesible en la Argentina para posibilitar una fabricación local.
- Realizar más experimentos de campo para validar las pruebas realizadas en simulación (Cap. V) y verificar así la confiabilidad del diseño frente a cambios en la planta.
- Verificar el límite de potencia máximo del driver implementado para los actuadores.
- Robustecer el diseño en función de las condiciones de trabajo (humedad, interferencias, rangos de temperaturas, etc)
- Realizar un plan de negocios para pasar de prototipo a producto.
- Desarrollar un sistema de control adaptativo, que por ejemplo sea capaz de detectar el retraso o inercia del sistema de enfriado y modifique las acciones de control pertinentemente.
- Investigar la viabilidad de utilizar el controlador en otros procesos con el fin de aumentar el mercado al cual está destinado y la posibilidad de agregar otro tipo de sensores necesarios para esto.

X - Anexo I - Esquemáticos y PCB

1. Placa A

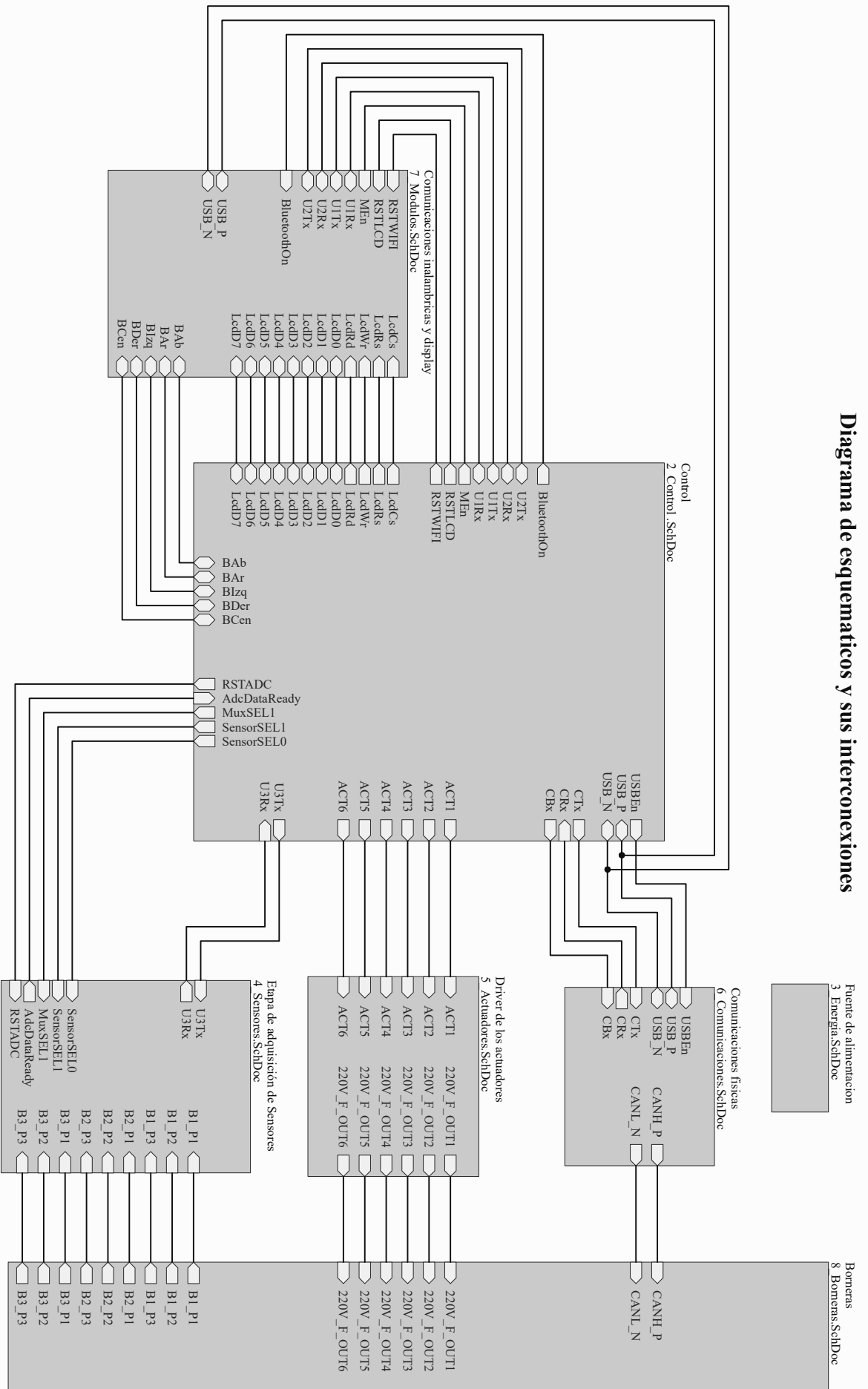
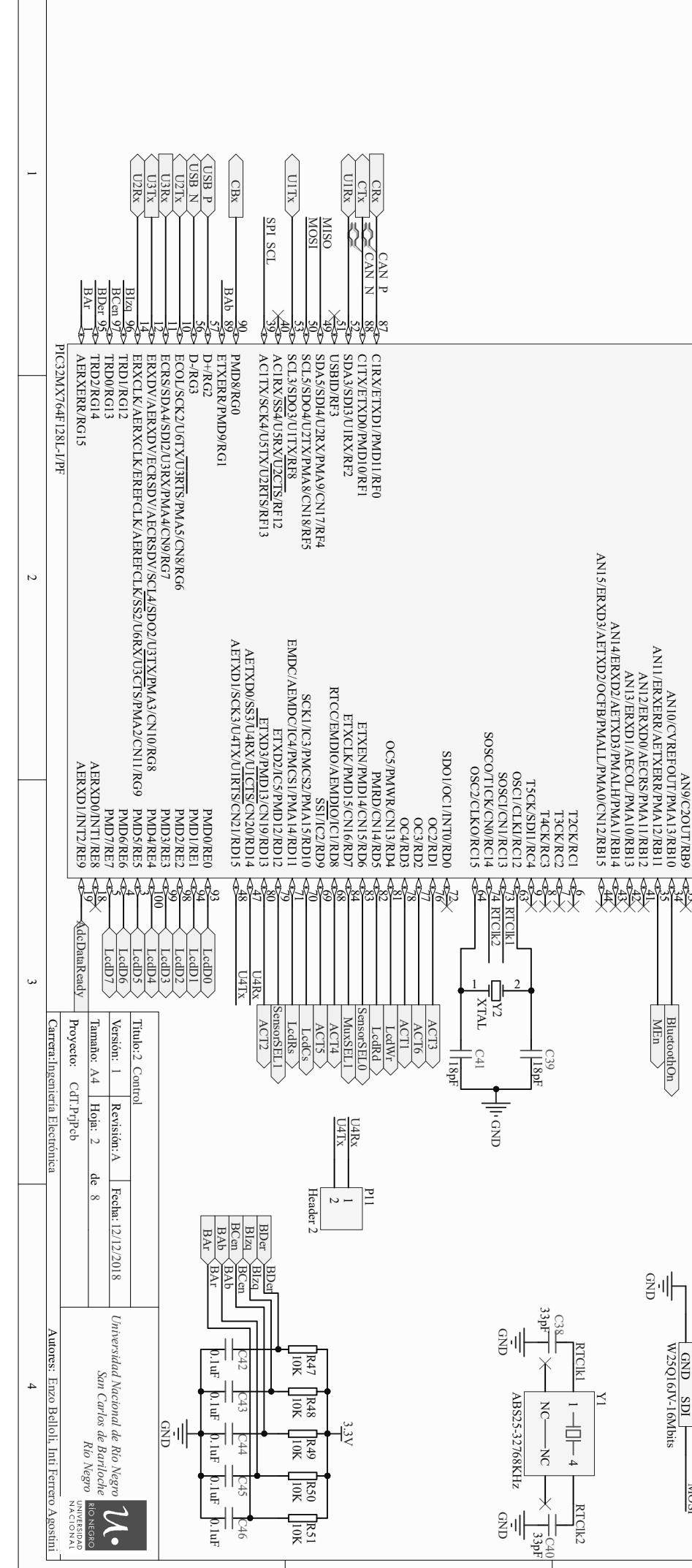
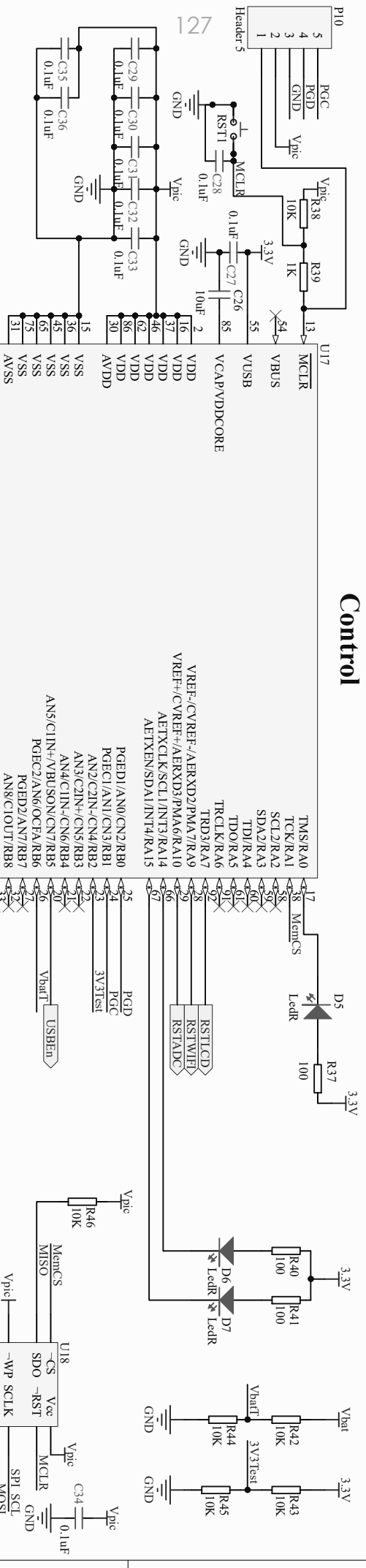


Diagrama de esquemáticos y sus interconexiones

Título: 1 Diagrama en bloques			
Version: 1	Revisión: A	Fecha: 12/12/2018	
Tamaño: A4	Hoja: 1	de 8	
Proyecto: CDT-PpPcb		Autores: Enzo Bellotti, Iní Ferrero Argostini	
Carrera: Ingeniería Electrónica		 Universidad Nacional de Río Negro San Carlos de Bariloche Río Negro UNIVERSIDAD NACIONAL	

Control

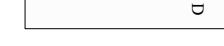


Pin	Signal	Pin	Signal
1	PGC	17	TMS/RA0
2	PGD	18	TCK/RA1
3	GND	19	SCI2/RA2
4	GND	20	SDA2/RA3
5	V _{pic}	21	TDI/RA4
6	V _{pic}	22	TDO/RA5
7	V _{pic}	23	TRCLK/RA6
8	V _{pic}	24	TRD3/RA7
9	V _{pic}	25	TRCLK/RA6
10	V _{pic}	26	TRD3/RA7
11	V _{pic}	27	TRCLK/RA6
12	V _{pic}	28	TRD3/RA7
13	V _{pic}	29	TRCLK/RA6
14	V _{pic}	30	TRD3/RA7
15	V _{pic}	31	TRCLK/RA6
16	V _{pic}	32	TRD3/RA7
17	V _{pic}	33	TRCLK/RA6
18	V _{pic}	34	TRD3/RA7
19	V _{pic}	35	TRCLK/RA6
20	V _{pic}	36	TRD3/RA7
21	V _{pic}	37	TRCLK/RA6
22	V _{pic}	38	TRD3/RA7
23	V _{pic}	39	TRCLK/RA6
24	V _{pic}	40	TRD3/RA7
25	V _{pic}	41	TRCLK/RA6
26	V _{pic}	42	TRD3/RA7
27	V _{pic}	43	TRCLK/RA6
28	V _{pic}	44	TRD3/RA7
29	V _{pic}	45	TRCLK/RA6
30	V _{pic}	46	TRD3/RA7
31	V _{pic}	47	TRCLK/RA6
32	V _{pic}	48	TRD3/RA7
33	V _{pic}	49	TRCLK/RA6
34	V _{pic}	50	TRD3/RA7
35	V _{pic}	51	TRCLK/RA6
36	V _{pic}	52	TRD3/RA7
37	V _{pic}	53	TRCLK/RA6
38	V _{pic}	54	TRD3/RA7
39	V _{pic}	55	TRCLK/RA6
40	V _{pic}	56	TRD3/RA7
41	V _{pic}	57	TRCLK/RA6
42	V _{pic}	58	TRD3/RA7
43	V _{pic}	59	TRCLK/RA6
44	V _{pic}	60	TRD3/RA7
45	V _{pic}	61	TRCLK/RA6
46	V _{pic}	62	TRD3/RA7
47	V _{pic}	63	TRCLK/RA6
48	V _{pic}	64	TRD3/RA7
49	V _{pic}	65	TRCLK/RA6
50	V _{pic}	66	TRD3/RA7
51	V _{pic}	67	TRCLK/RA6
52	V _{pic}	68	TRD3/RA7
53	V _{pic}	69	TRCLK/RA6
54	V _{pic}	70	TRD3/RA7
55	V _{pic}	71	TRCLK/RA6
56	V _{pic}	72	TRD3/RA7
57	V _{pic}	73	TRCLK/RA6
58	V _{pic}	74	TRD3/RA7
59	V _{pic}	75	TRCLK/RA6
60	V _{pic}	76	TRD3/RA7
61	V _{pic}	77	TRCLK/RA6
62	V _{pic}	78	TRD3/RA7
63	V _{pic}	79	TRCLK/RA6
64	V _{pic}	80	TRD3/RA7
65	V _{pic}	81	TRCLK/RA6
66	V _{pic}	82	TRD3/RA7
67	V _{pic}	83	TRCLK/RA6
68	V _{pic}	84	TRD3/RA7
69	V _{pic}	85	TRCLK/RA6
70	V _{pic}	86	TRD3/RA7
71	V _{pic}	87	TRCLK/RA6
72	V _{pic}	88	TRD3/RA7
73	V _{pic}	89	TRCLK/RA6
74	V _{pic}	90	TRD3/RA7
75	V _{pic}	91	TRCLK/RA6
76	V _{pic}	92	TRD3/RA7
77	V _{pic}	93	TRCLK/RA6
78	V _{pic}	94	TRD3/RA7
79	V _{pic}	95	TRCLK/RA6
80	V _{pic}	96	TRD3/RA7
81	V _{pic}	97	TRCLK/RA6
82	V _{pic}	98	TRD3/RA7
83	V _{pic}	99	TRCLK/RA6
84	V _{pic}	100	TRD3/RA7

Title: 2. Control	
Version: 1	Revision: A
Tamaño: A4	Hora: 2
Proyecto: CDT-PpPeb	Fecha: 12/12/2018

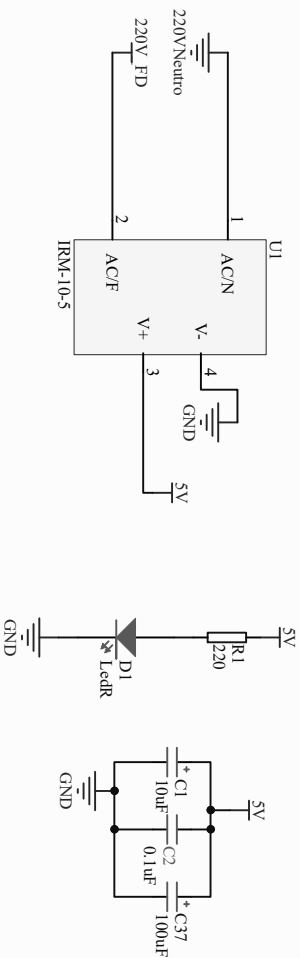
Universidad Nacional de Rio Negro	
San Carlos de Bariloche	
RIO NEGRO	
UNIVERSIDAD NACIONAL	

Autores: Enzo Bellotti, Iní Ferrero Argostini	
---	--

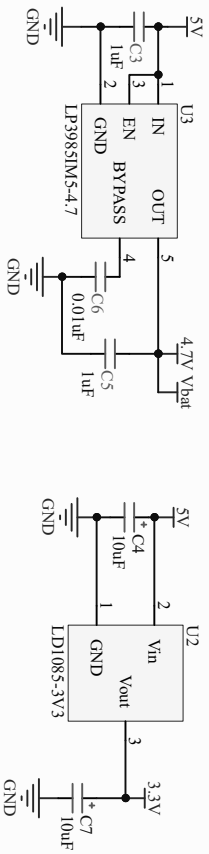


Fuente de alimentacion

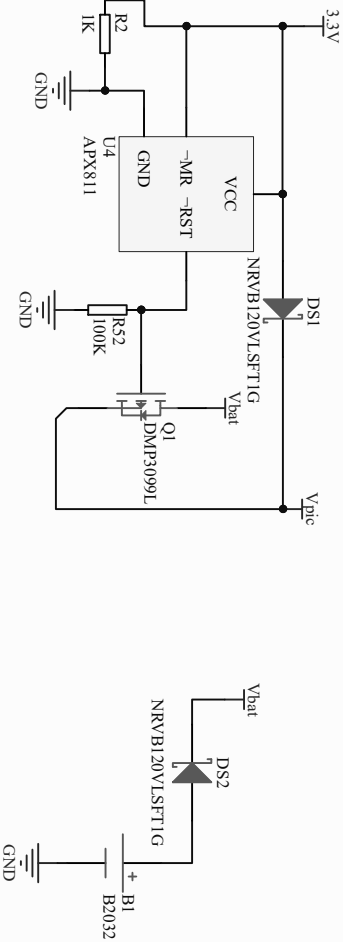
Circuito de Alimentacion primaria (fuente conmutada 220VAC-5VDC/10W)



Fuente regulada



Detectores de tension y alimentacion de respaldo para el PIC



1

2

3

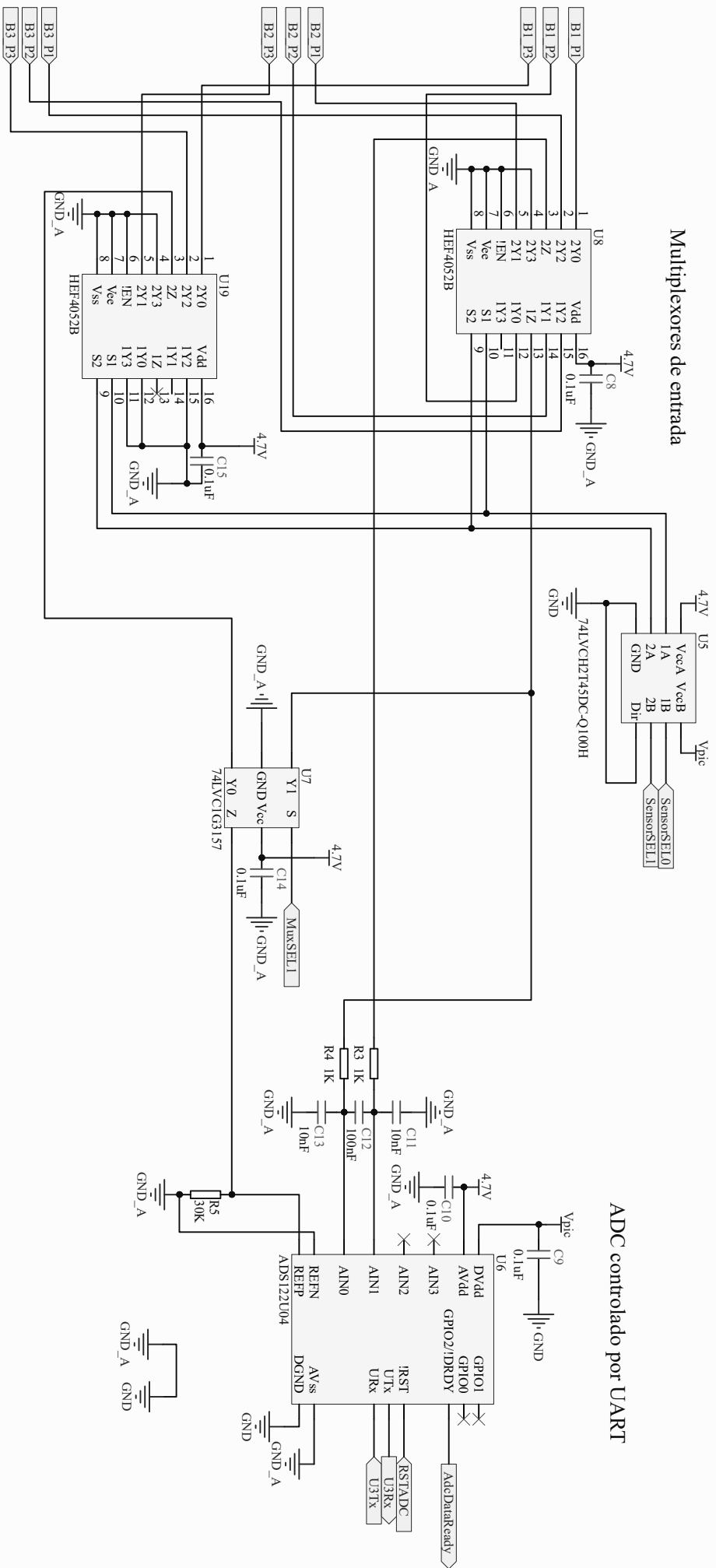
4

Titulo:3. Energia		Fecha: 12/12/2018	
Version: 1	Revision: A	Hoja: 3	de 8
Tamaño: A4	Proyecto: Cdt.PpPeb		
Carrera:Ingenieria Electronica		Autores: Enzo Belloli, Inii Ferrero Argosini	

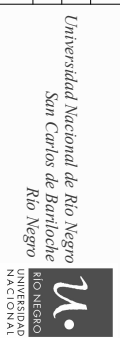
Universidad Nacional de Rio Negro
 San Carlos de Bariloche
 Rio Negro
 UNIVERSIDAD NACIONAL



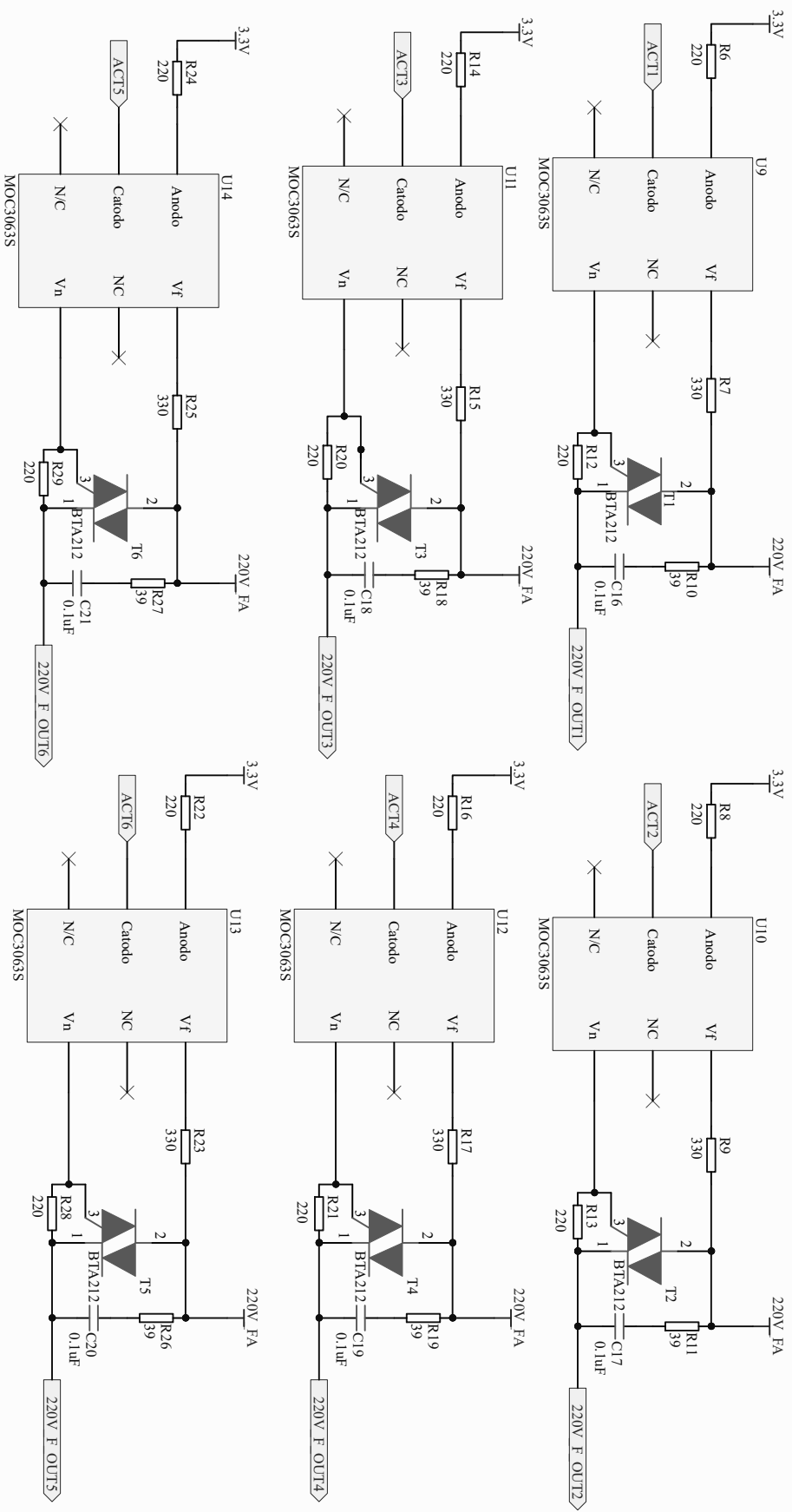
Etapa de adquisición de Sensores



Título: 4. Sensores			
Version: 1	Revisión: A	Fecha: 12/12/2018	
Tamaño: A4	Hoja: 4	de 8	
Proyecto: CDT.PpPcb			
Carrera: Ingeniería Electrónica		Autores: Enzo Bellotti, Iní Ferrero Argostini	



Driver de los actuadores

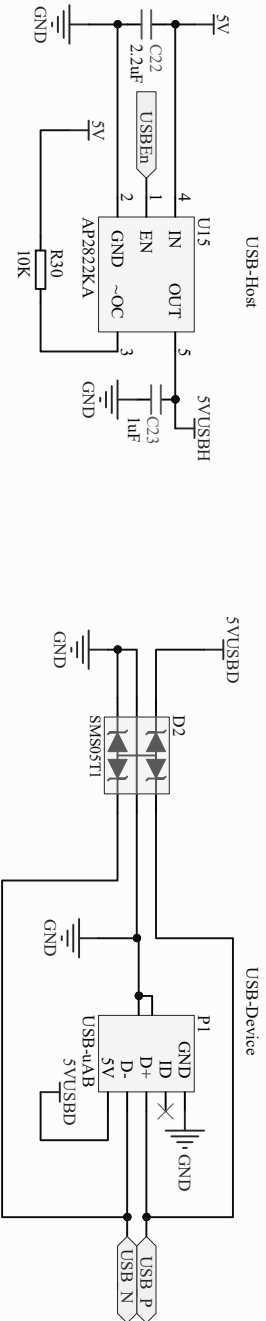


NOTA:
 PCB: Ubicar los tiristores teniendo en cuenta posible disipador
 Hacer plano de GND separado, y marcar en la placa la parte que puede llegar a tener 220V.

Titulo: 5. Actuadores		Fecha: 12/12/2018	
Version: 1	Revision: A	de	8
Tamaño: A4	Hoja: 5		
Proyecto: CDT.PpPeb		Autores: Erizo Bellotti, Iní Ferrero Argosini	
Carrera: Ingeniería Electrónica		UNIVERSIDAD NACIONAL DEL RIO NEGRO San Carlos de Bariloche Rio Negro	

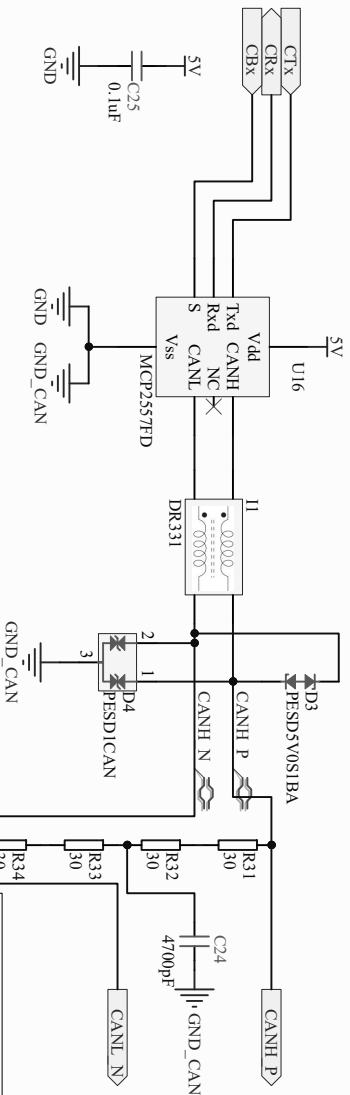
Comunicaciones físicas

Universal Serial Bus (USB)




El dispositivo va a poder mantener comunicaciones USB tanto como Host como Device, esto es pensado para poder utilizar la interfaz USB en el proceso de programación/debugger y luego cuando el producto este terminado los usuarios puedan conectar pendrives para descargar/cargar curvas de temperatura.

Controller Area Network (CAN BUS)

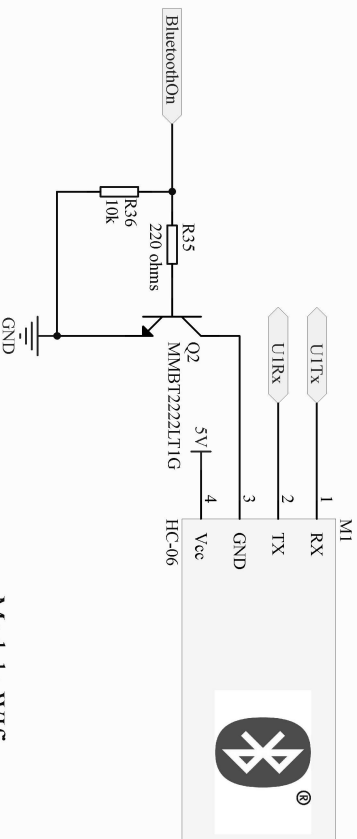
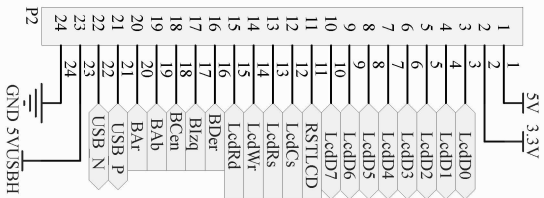


Nota: Hacer un plano de GND independiente para minimizar interferencias con la comunicación CAN

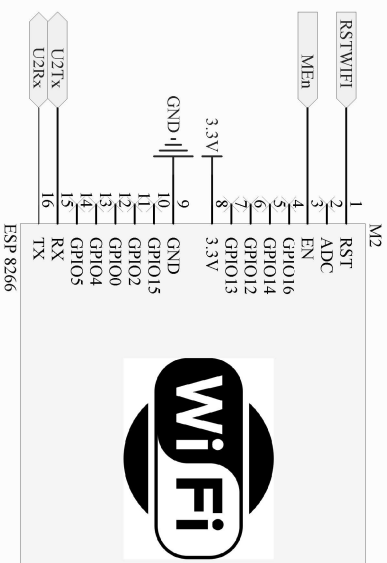
Título: 6. Comunicaciones		Autores: Enzo Bellotti, Iní Ferrero Argostini	
Version: 1	Revisión: A	Fecha: 12/12/2018	 Universidad Nacional de Río Negro San Carlos de Bariloche Río Negro UNIVERSIDAD NACIONAL
Tamaño: A4	Hoja: 6 de 8		
Proyecto: CDT-PpPcb			

Comunicaciones inalámbricas y display

Salida al frente del equipo

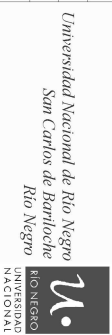


Modulo Bluetooth



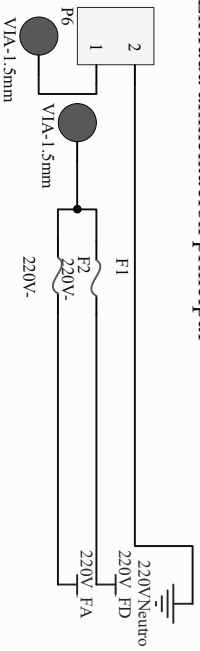
Modulo WiFi

Titulo: 7. Módulos			
Version: 1	Revisión: A	Fecha: 12/12/2018	
Tamaño: A4	Hoja: 7	de 8	
Proyecto: CDT-PpPeb			
Carreera: Ingeniería Electrónica		Autores: Enzo Belloli, Iná Ferrero Agostini	

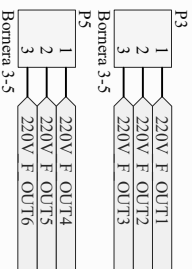


Borneras

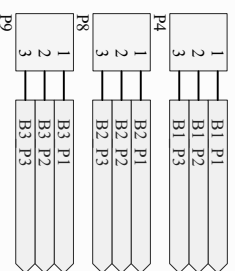
Entrada alimentacion principal



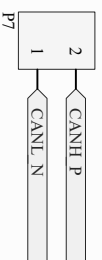
Salidas de actuadores




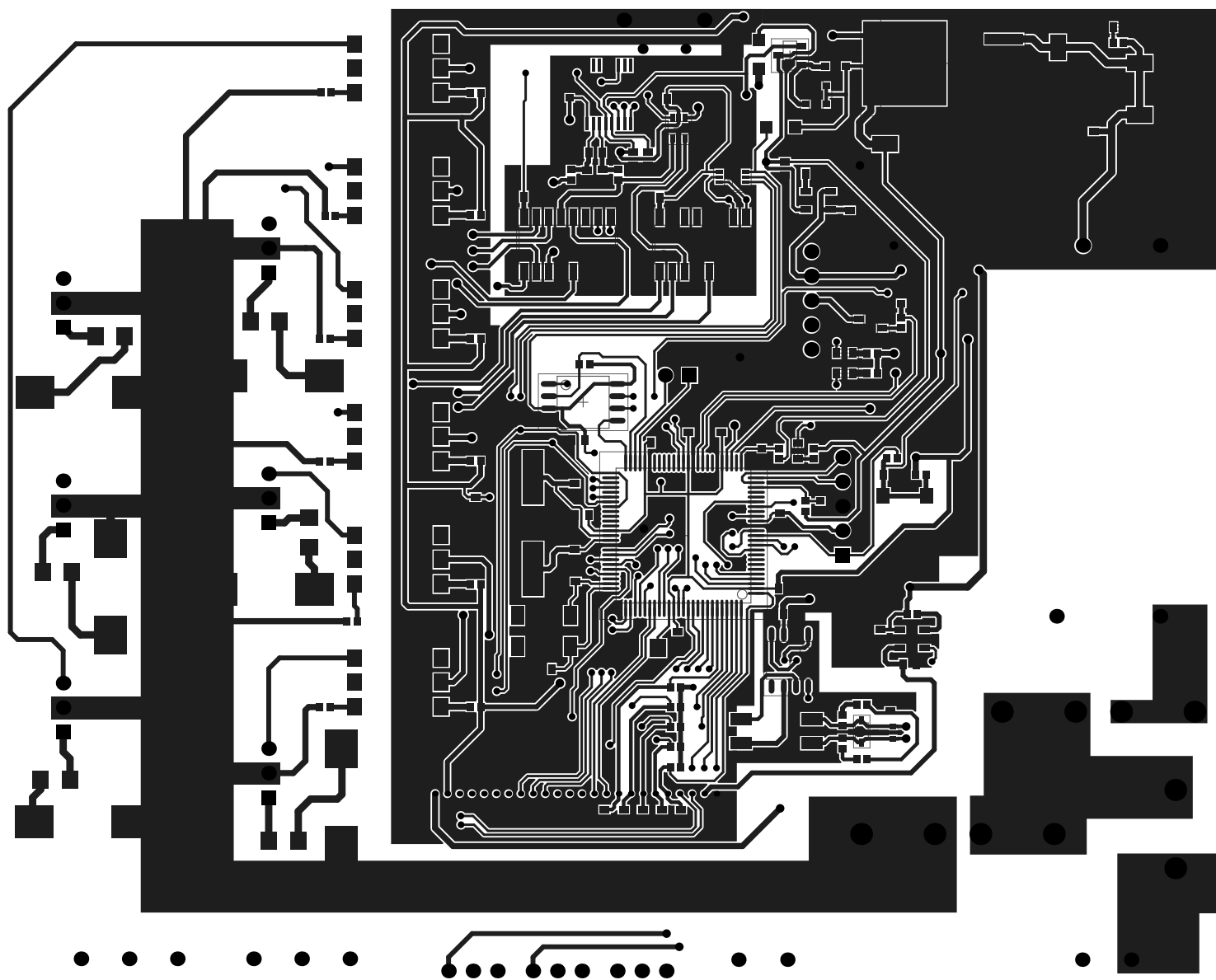
Entrada de sensores



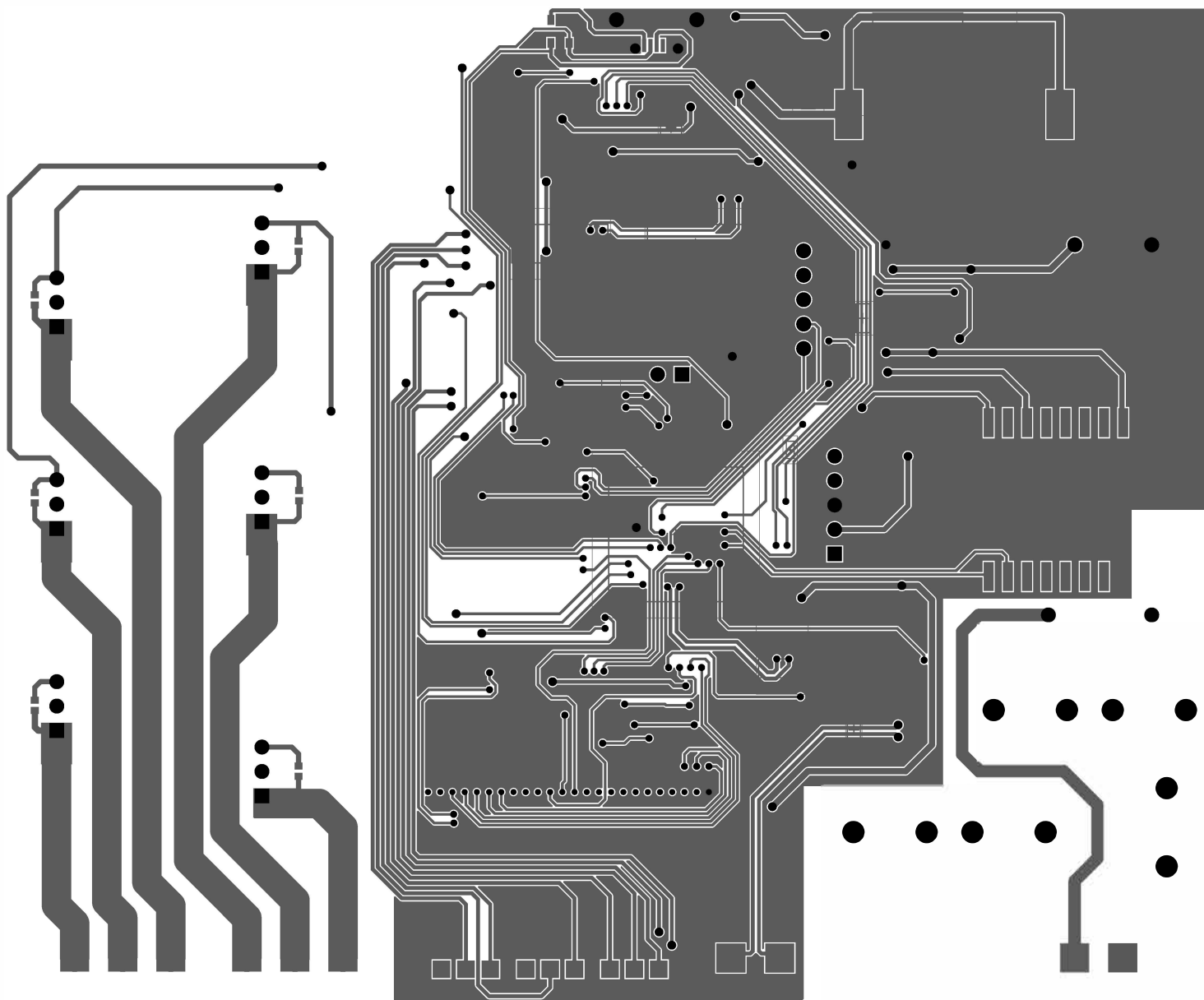
CAN BUS



Titulo: 8 Borneras			
Version: 1	Revision: A	Fecha: 12/12/2018	
Tamaño: A4	Hoja: 8	de 8	
Proyecto: Cdt.PpPeb			
Carrera: Ingeniería Electrónica			
Autores: Enzo Bellotti, Iní Ferrero Argostini			
Universidad Nacional de Río Negro San Carlos de Bariloche Río Negro UNIVERSIDAD NACIONAL			

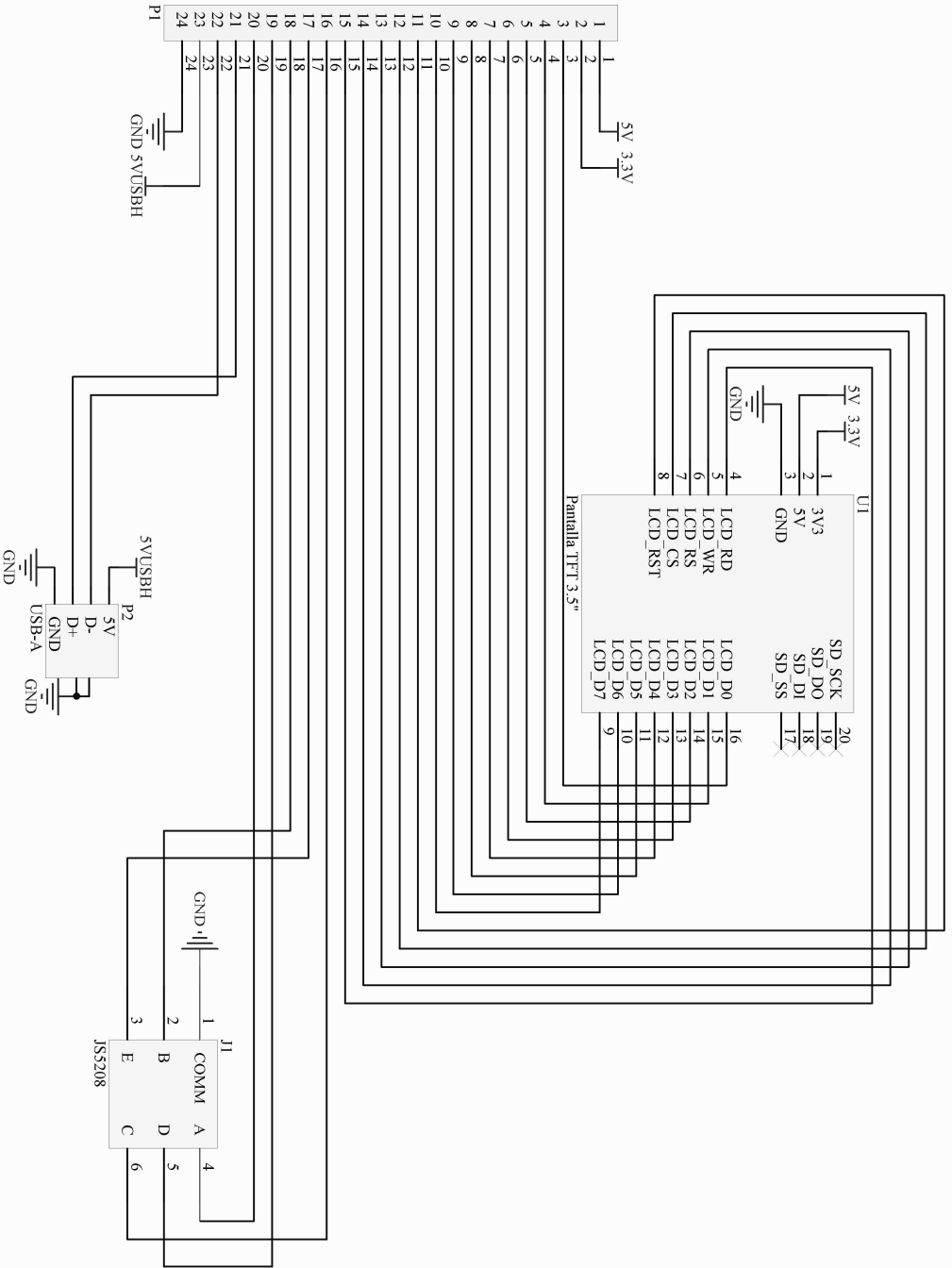


Lado TOP



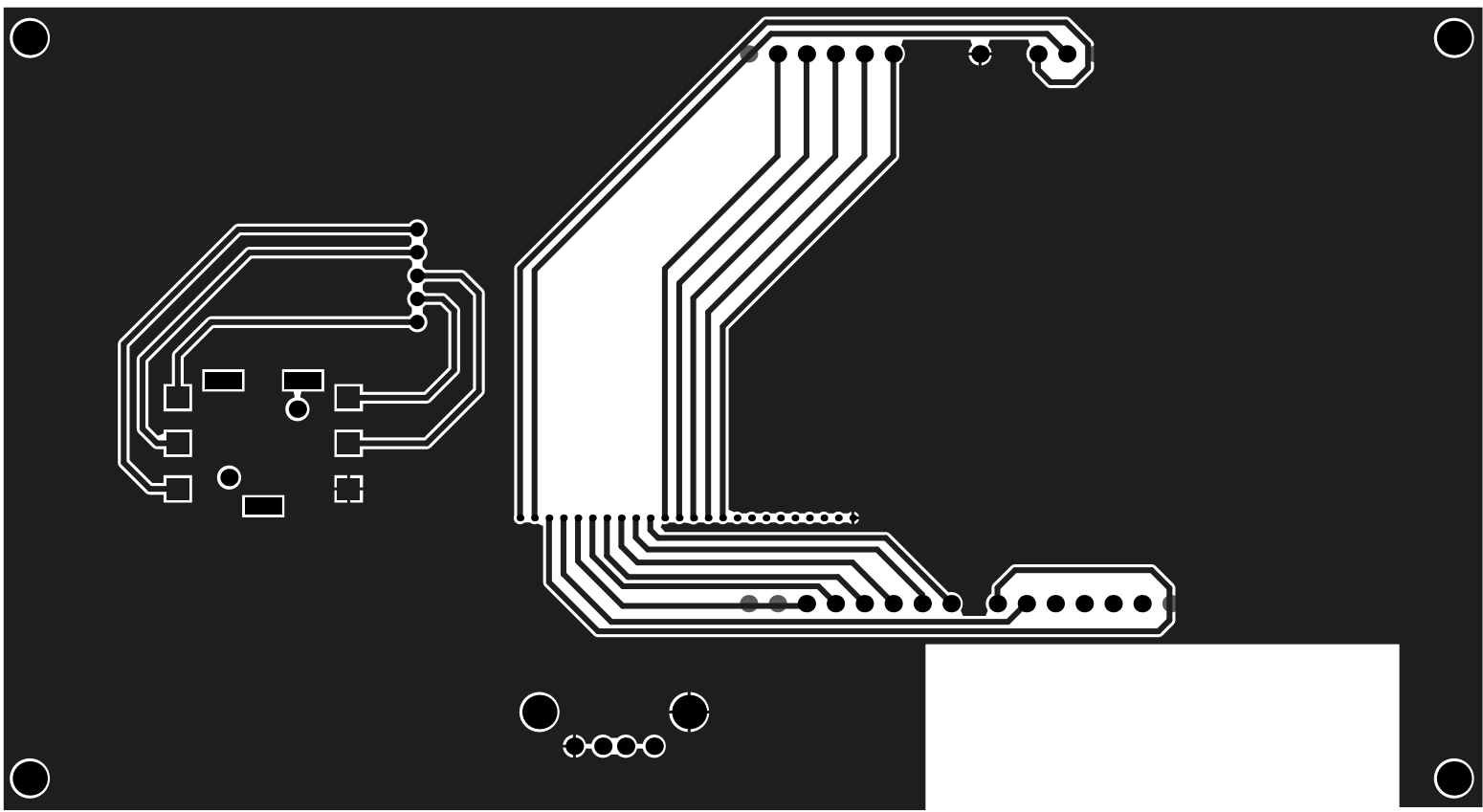
Lado BOTTOM

2. Placa B

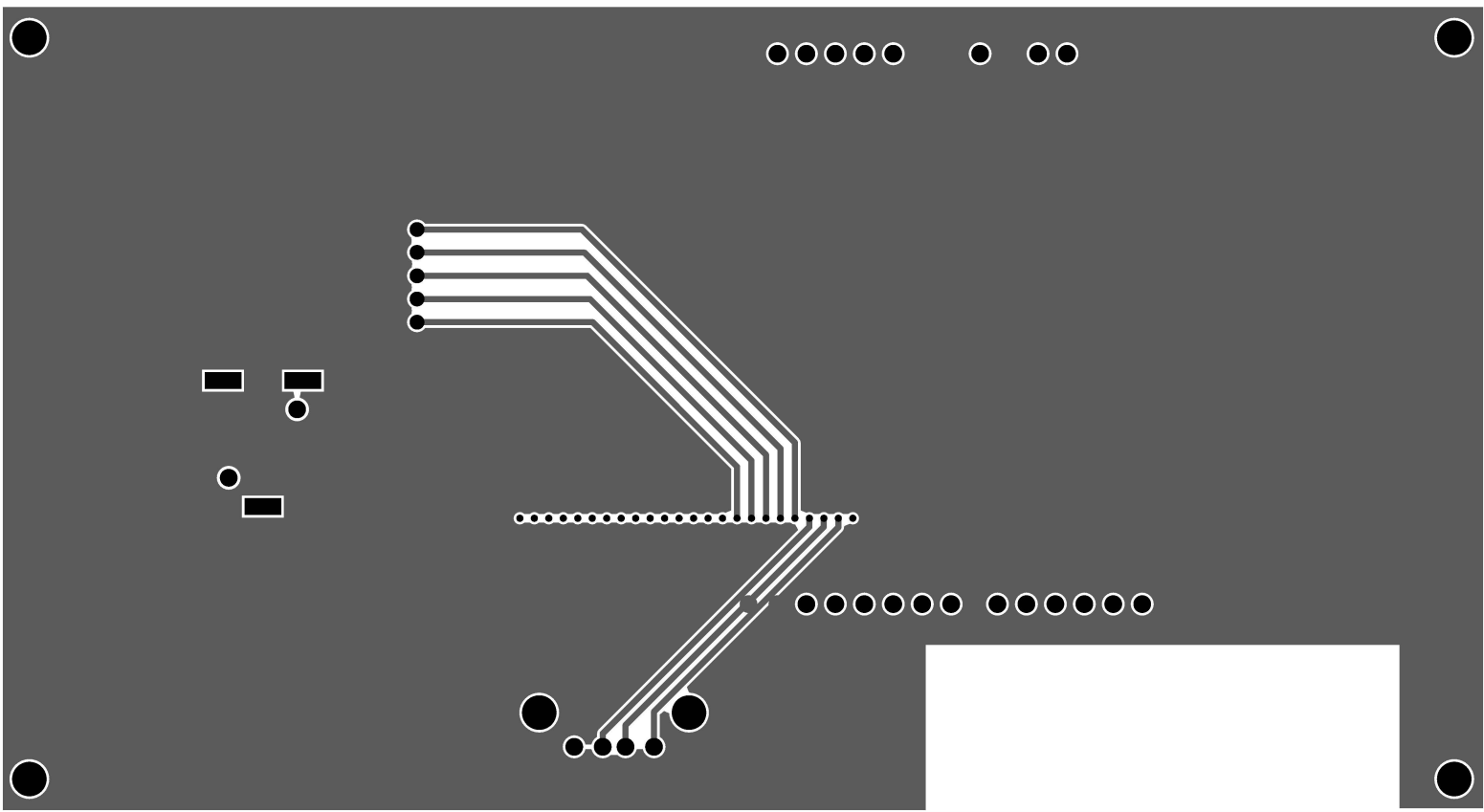


Título: Interfaz usuario		Fecha: 12/12/2018	
Version: 1	Revisión: A		
Tamaño: A4	Hoja: 1	de 1	
Proyecto: CDT_Tapa_PjPeb			
Carerra: Ingeniería Electrónica		Autores: Enzo Belloli, Iná Ferrero Agostini	





Lado TOP



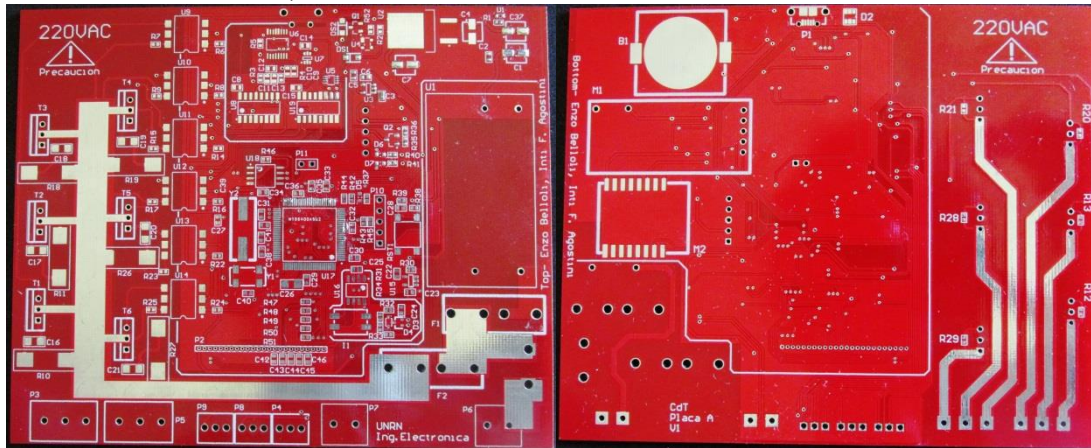
Lado BOTTOM

XI - Anexo II - Ensamblado y puesta en marcha

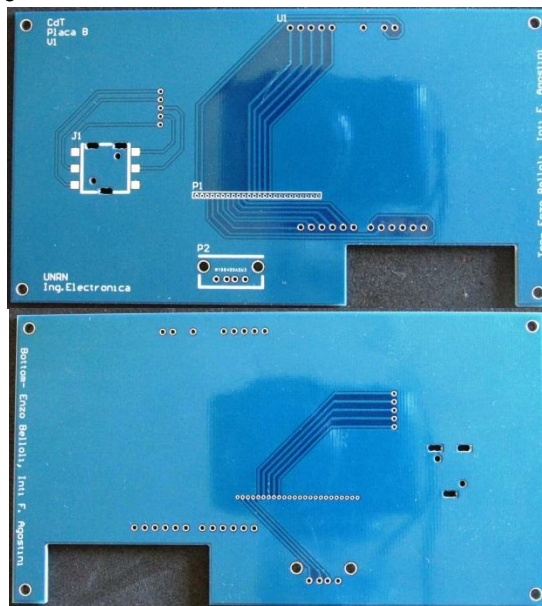
1. Objeto

El presente documento tiene como objeto definir el procedimiento de puesta en marcha de los circuitos impresos del Controlador de Temperatura, CdT por sus siglas;

- Placa A, V1.0



- Placa B, V1.0



2. Alcance

Este documento tiene alcance sobre el CdT V1.

3. Documentación aplicable

1. Capítulo V- subtítulo 4: Lista de componentes.

4. Definiciones

AI Alcohol isopropílico

SA Soldador por aire

MD Multímetro digital

OD Osciloscopio digital

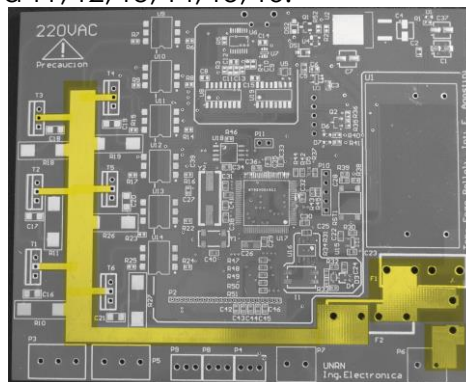
5. Procedimiento de ensamblado

5.1. Equipamiento necesario

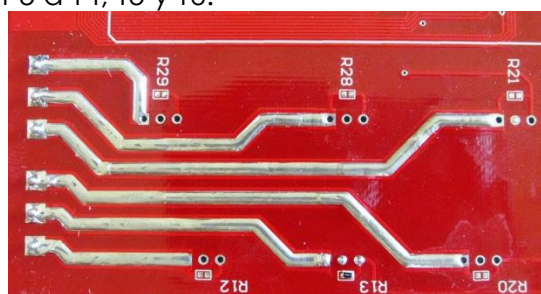
1. Guantes antiestáticos.
2. Manta antiestática.
3. Alcohol Isopropílico en aerosol.
4. Papel.
5. Decapante en gel.
6. Malla desoldante de 1.5mm de ancho.
7. Computadora con el software MPLAB X.
8. Programador PicKit 3 y sus accesorios.

5.2. Placa A

1. Limpiar la superficie de la placa con Al, un cepillo suave y papel hasta que quede completamente limpia.
2. Estañar con soldador de mano las pistas del lado TOP:
 - a. P6
 - b. De F2 a T1, T2, T3, T4, T5, T6.



3. Estañar con soldador de mano las pistas del lado BOTTOM:
 - a. De P3 a T1, T2 y T3.
 - b. De P5 a T4, T5 y T6.



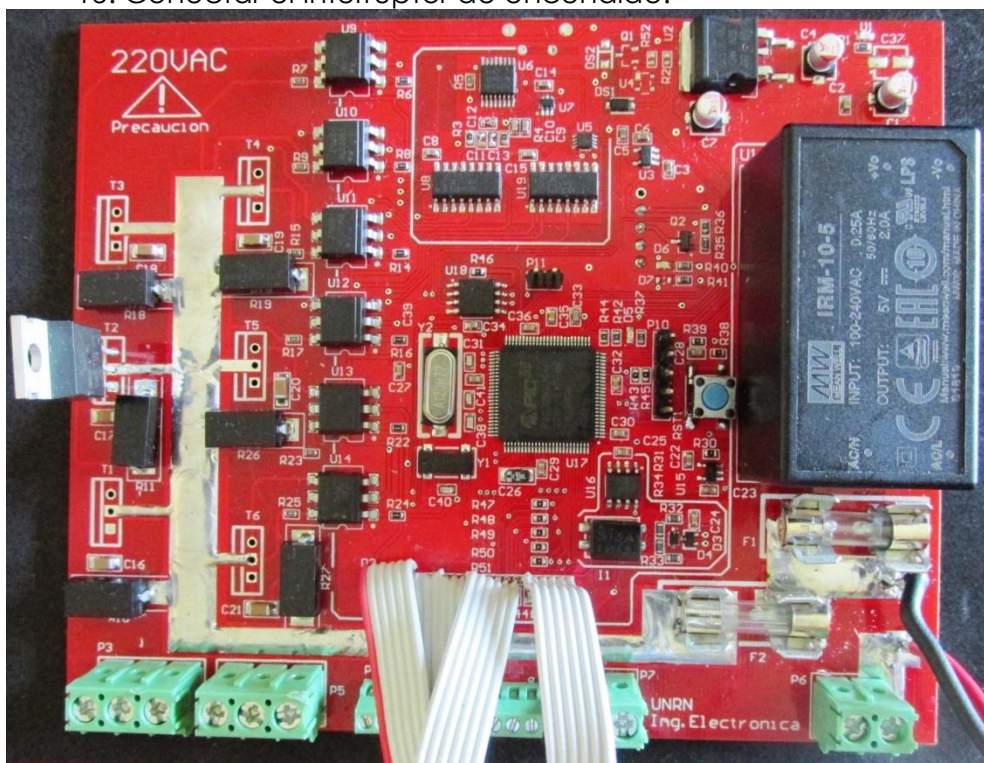
4. Limpiar la superficie de la placa con Al, un cepillo suave y papel hasta que quede completamente limpia.

5.2.1 Lado TOP:

5. Colocar el estencil alineado con los pads del PCB.
6. Colocar con una espátula estaño en pasta.
7. Retirar el estencil y dar comienzo a la colocación de componentes.
8. El criterio de colocación es de componentes más pequeños a más grandes.

- a. **Resistencias.** A excepción de R18, R19, R26, R11, R10, R27, R31, R32, R33 y R34.
 - b. **Capacitores.**
 - c. **Diodos.** Luego de los diodos D3 y D4 se agregan las resistencias R31, R32, R33 y R34.
 - d. **Circuitos integrados.** También se incluye ahora a Y1.
 - e. **Resistencias.** Se agregan R10, R11, R18, R19, R26 y R27.
9. Se reserva para ser colocados posteriormente a Y2.
 10. Realizar rutina de precalentamiento del horno.

Esto solo es necesario cuando el manual del usuario del horno a utilizar así lo requiera.
 11. Seleccionar la curva de temperatura asociada al tipo de estaño en pasta que se haya utilizado y se procede a soldar.
 12. Soldar componentes TH y/o aquellos que no están diseñados para el horno.
 - f. RST1, botón de reset
 - g. P10, Pin de programación.
 - h. T1, T2, T3, T4, T5, T6 .Triacs
 - i. F1 y F2. Porta fusibles
 - j. U1. Fuente de alimentación. *El largo de los pines de +5V y GND deben estar soldados al ras de la placa.*
 - k. P6, P7, P4, P8, P9, P5 y P3. Borneras
 - l. P2. Cable a placa B *(el paso de las vías donde debe conectarse tiene un paso demasiado fino, por lo que se recomienda soldar sobre los pads, en vez de atravesar la placa.)*
 13. Soldar cable de 1.5mm al PAD de P6 y otro cable al compartido por F2 y F1.
 14. Soldar al otro extremo de los cables las fichas para el interruptor de encendido.
 15. Conectar el interruptor de encendido.



5.2.2 Lado BOTTOM:

La cantidad de componentes del lado Bottom es pequeña y requiere que todos sean soldados manualmente.

16. **Resistencias.**

17. **Porta pila. B1**

18. **Módulos, M1 y M2**

19. **P1.** Previo a la colocación del componente poner estaño en pasta sobre los pads SMD. Luego solamente soldar los pins TH del componente.

20. Con un soldador de aire soldar los pads SMD de P1.

21. **D2.**

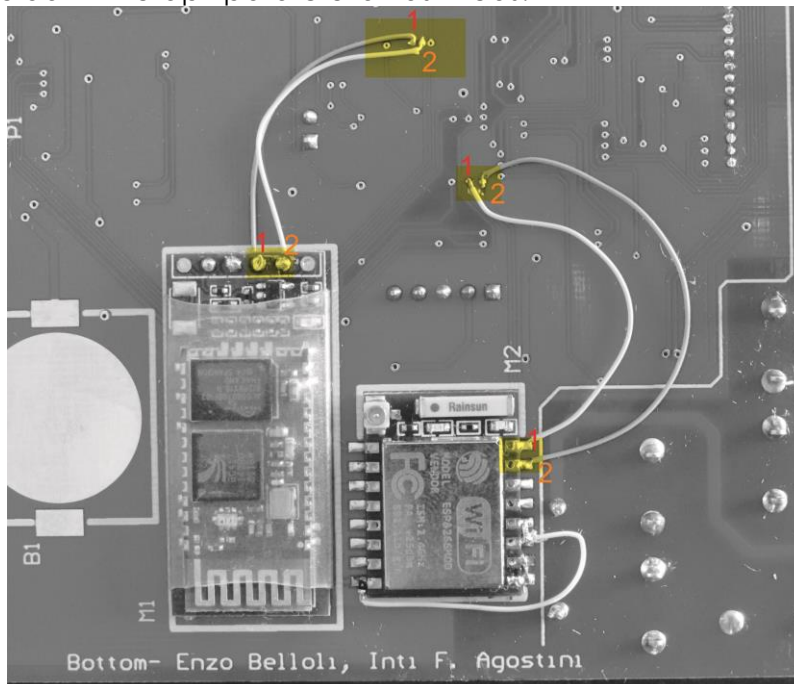
“Hay que tener especial cuidado al momento de soldar el modulo Bluetooth (M1) de no generar un cortocircuito con los pines de U1 (fuente de alimentación).”

5.2.3 Modificaciones

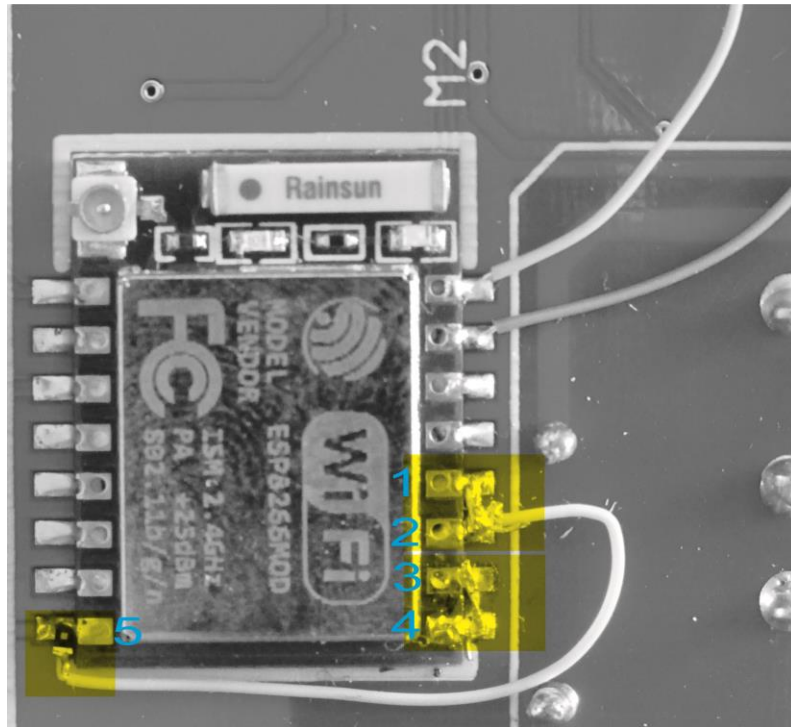
Para el correcto funcionamiento del dispositivo es necesario realizar una serie de modificaciones en la Placa A.

Las líneas de Tx y Rx de las UARTs 1 y 2 deben ser cruzadas y soldadas directamente a los módulos WiFi y Bluetooth.

1. Cortar las pistas del lado BOTTOM lo más cercano a las vias posible
2. Soldar “wirerap” para cruzar las líneas.



Al módulo WiFi es necesario agregar una resistencia de 10K y unir alguno de sus pines.

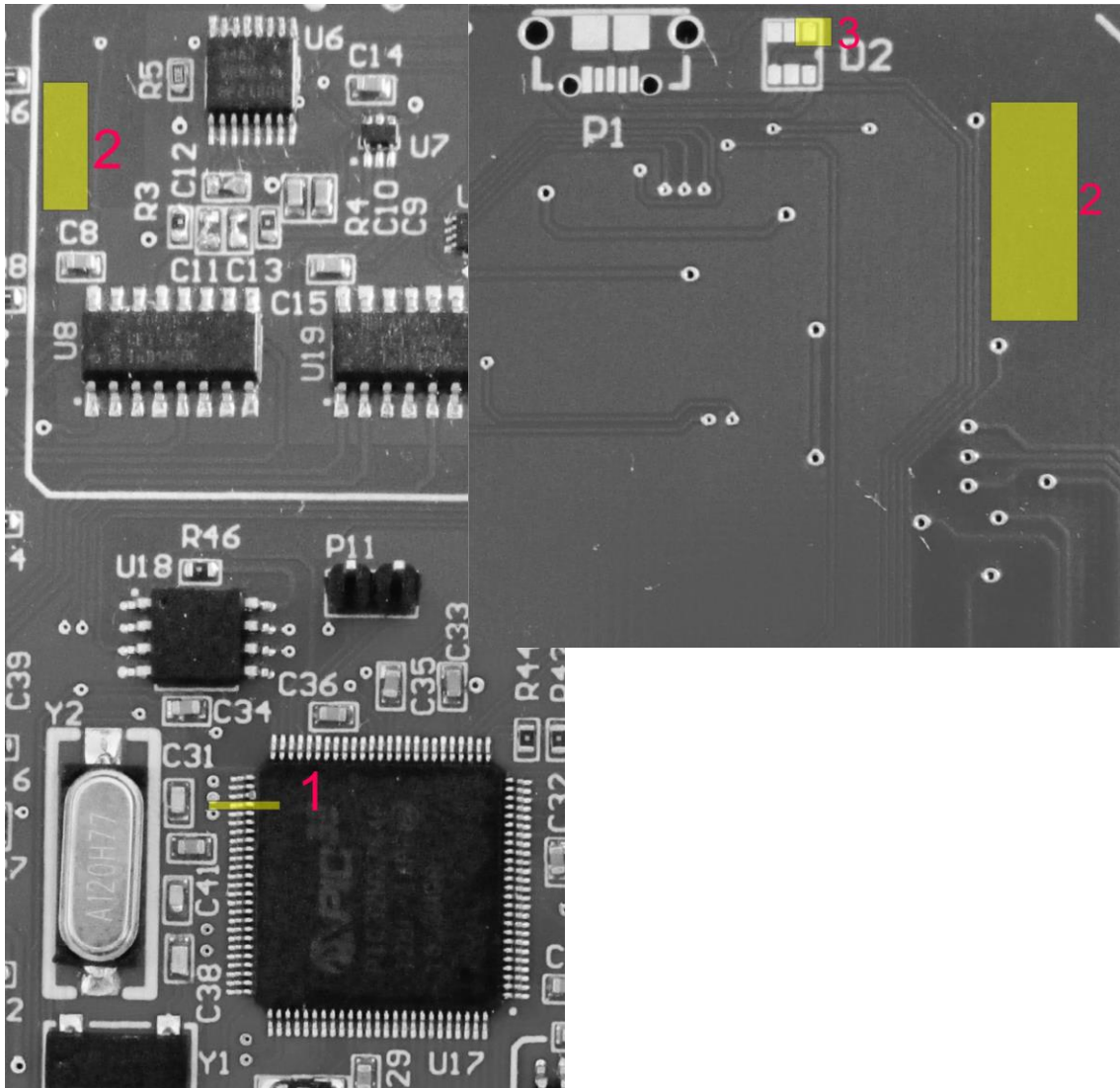


Se debe realizar las siguientes modificaciones:

1. Unir 1 con 2.
2. En 5 soldar la resistencia de 10kohm.
3. Unir el extremo libre de la resistencia con 2.
4. Unir 3 con 4.

Para detectar la conexión de un dispositivo en el puerto micro USB, es necesario que el microcontrolador sense la línea de 5V del conector. Esta línea falta, por lo que es necesario conectar un cable, del estilo wirerap, desde el pin 53 del microcontrolador al pin 1 de la protección del USB.

- a. Realizar un agujero en alguna parte del área 2
- b. Soldar un cable en 1 y atravesar la placa por el agujero hecho en "a".
- c. Soldar el cable en 3.



Al finalizar se procede a realizar una limpieza de toda la placa con Al, un cepillo suave y papel. Es importante que la limpieza sea profunda y exhaustiva.

5.2.4 Inspección

En la inspección se busca encontrar cortocircuitos producto de una soldadura incorrecta, una cantidad excesiva de estaño o de componentes en mal estado.

Visual

1. Se realiza una inspección ocular, con la ayuda un microscopio y/o lupa.

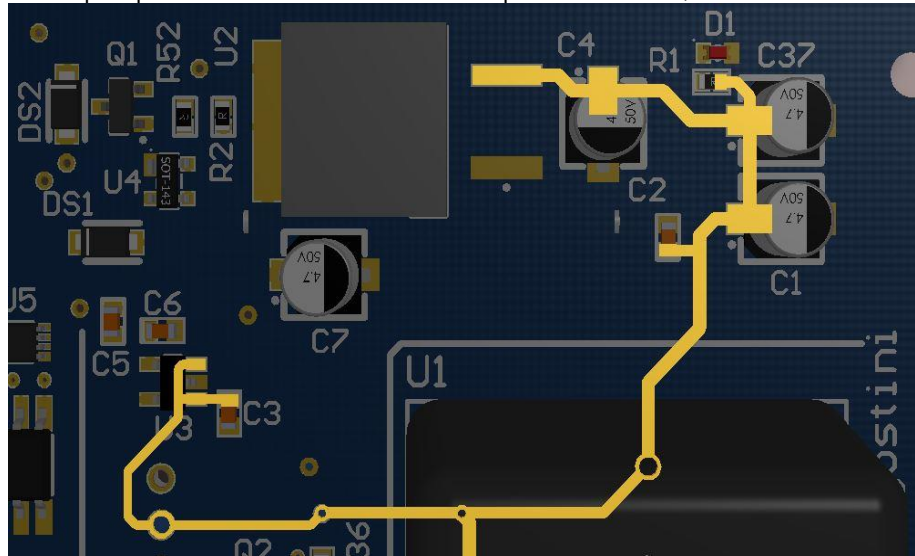
Prestar especial atención a los siguientes componentes: U5, U7, U17, U6, U3, C12, C11y C13.

2. Con la ayuda de un MD revisar la continuidad de las trazas de alimentación y de todos los pines si se observara algún tipo de defecto en la soldadura.

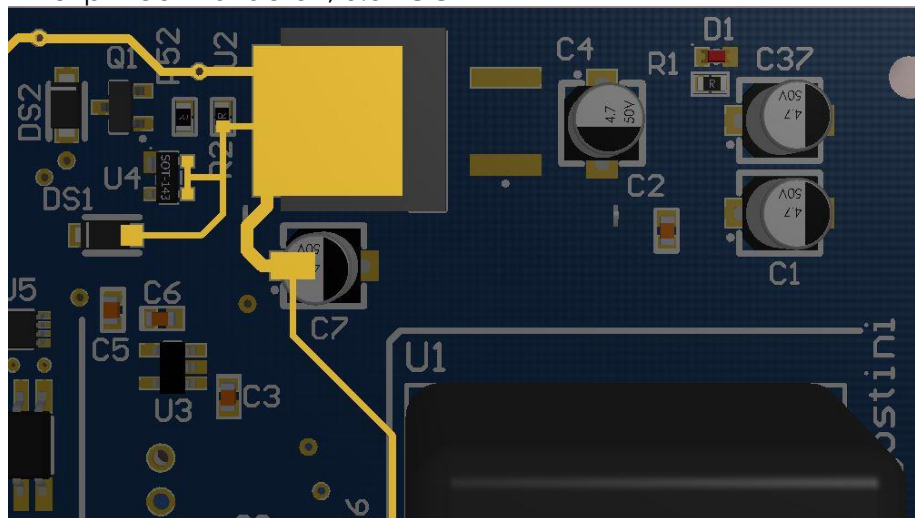
Alimentación.

3. Con el interruptor en la posición de apagado, conectar la placa a la línea de alimentación de 220VCA.

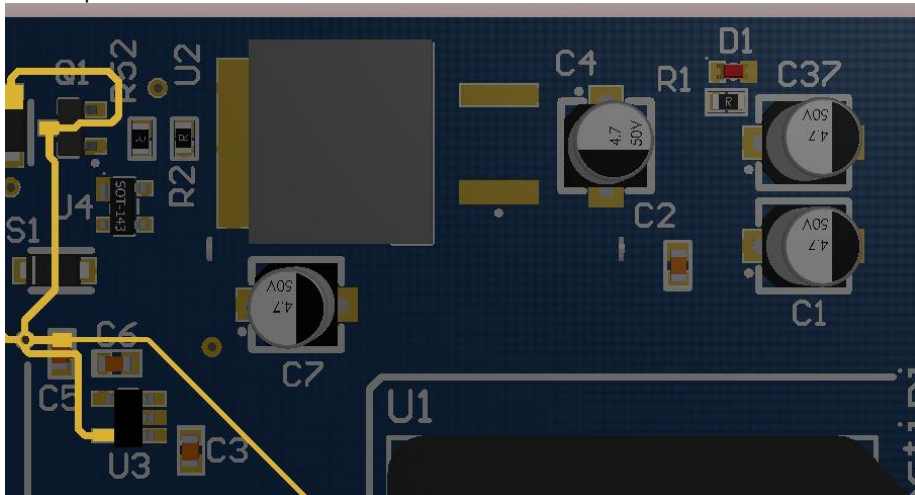
4. Verificar, con la ayuda de un MD, que la tensión en P6 sea la correcta (120VCA a 260VCA).
5. Cambiar el interruptor a la posición de encendido y verificar que el LED, D1, se encienda.
6. Con el multímetro verificar:
 - a. En el pin positivo marcado con un punto de C1, 5VCC.



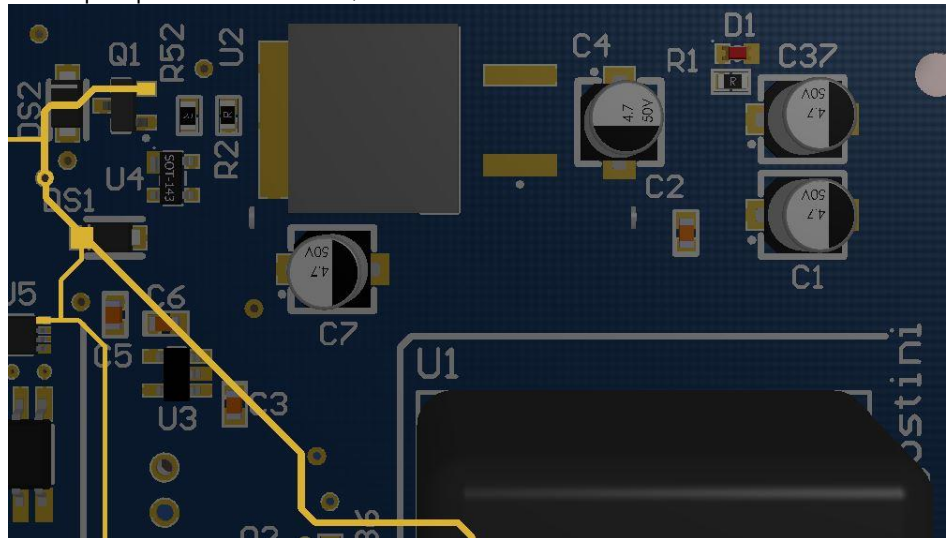
- b. En el pin central de U2, 3.3VCC



- c. En el pin 5 de U3, 4.7VCC



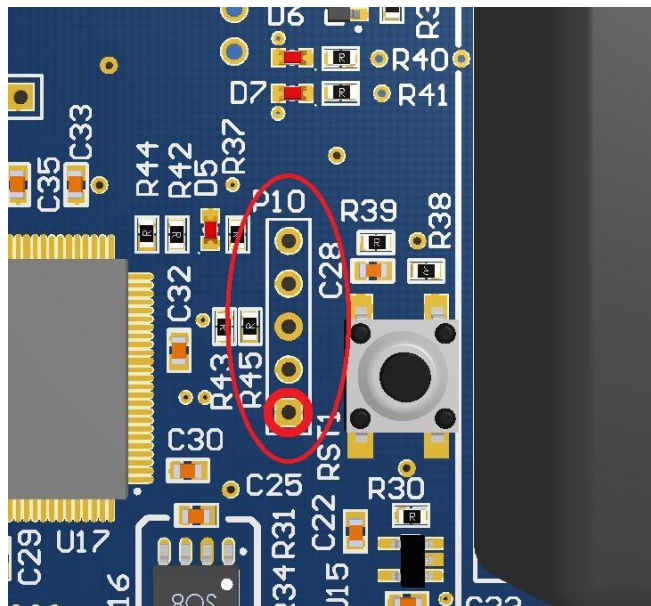
- d. En el pin positivo de DS1, 3.2VCC



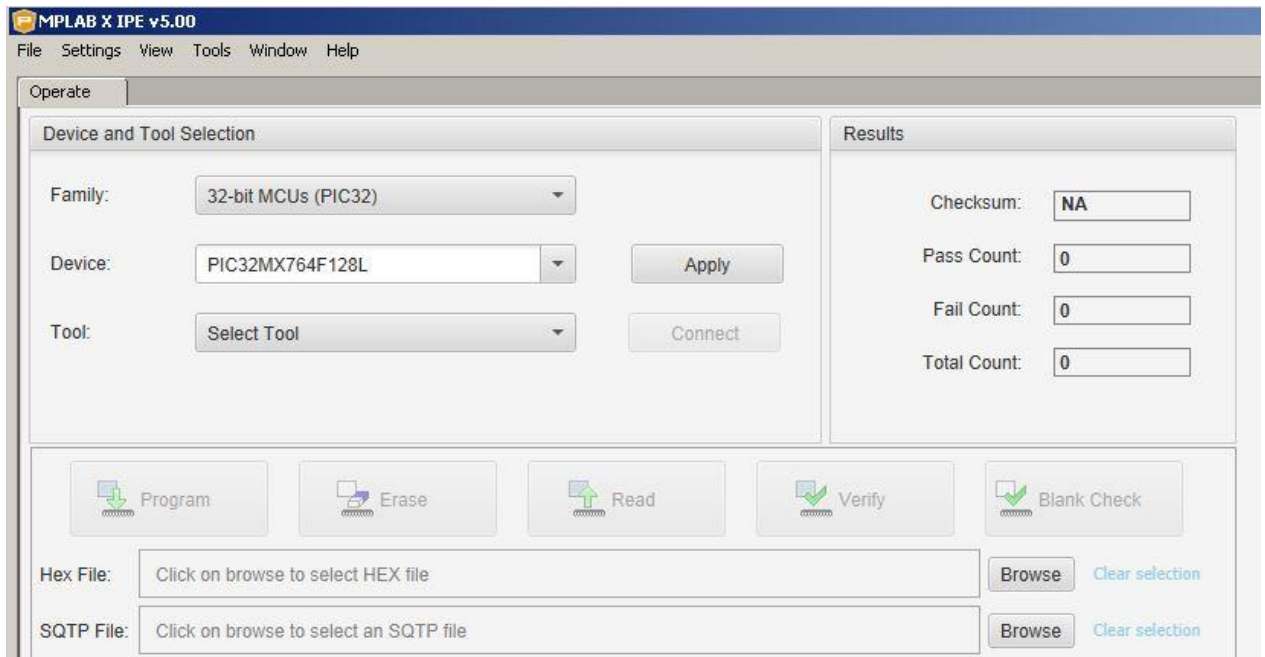
Micro controlador

NOTA: Para este procedimiento se necesitara una PC con el software MPLAB y un puerto USB libre, un programador de PICs y un cable USB a USB mini.

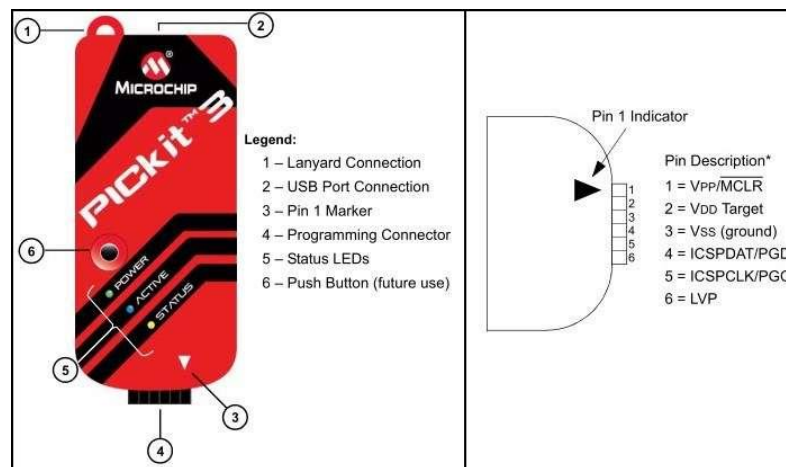
7. Conectar la placa a una computadora utilizando un PicKit3, mediante el conector P10 (pin MCRL primero viendo la placa desde el frente).



8. Con el MPLAB X IPE abierto y configurado, conectar el cable USB del programador a la computadora.



Verificar que el microcontrolador es detectado.



5.3. Placa B

5.3.1 Lado TOP.

1. Limpiar la placa con alcohol isopropílico, un cepillo suave y papel.
2. Soldar en el siguiente orden:
 - a. J1
 - b. P2
 - c. P1

Con un multímetro verificar que cada uno de los pines de P1 estén correctamente soldados y exista continuidad entre la Placa A y la Placa B.

3. Soldar U1.

6. Procedimiento para la verificación del software

6.1. Equipamiento necesario

Una computadora con Windows 7 o superior con los siguientes programas o similares instalados:

- RealTerm: Programa de terminal serie para capturar, controlar y depurar código binario, ASCII y otros flujos de datos de los puertos COM de una PC.
- MPLAB IPE V5.0 o posterior.
- MPLAB Harmony V2.06.

Un celular Smartphone con la siguiente aplicación o similar instalada:

- Bluetooth Terminal HC-05

Un módulo convertidor serial UART a USB.

Cable USB tipo A a tipo Micro AB.

Una Red WiFi configurada con **SSID = redCel** y **pass = 123456en**

6.2. Pasos a seguir

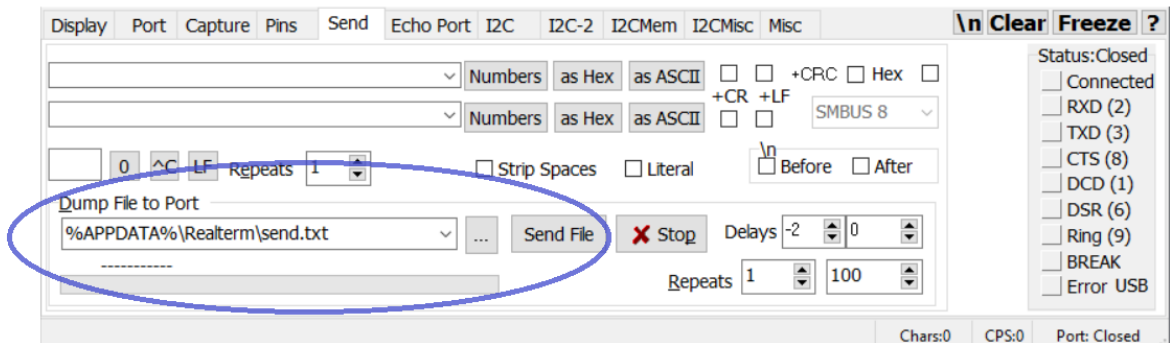
1. Conectar el modulo convertidor a los pines del conector P11 en la placa A. *P11-1* es Rx y *P11-2* es Tx.
2. Abrir RealTerm y abrir el puerto COM asignado al módulo, seleccionando una velocidad de 9600 baudios, 8 bits de datos, sin bit de paridad y un bit de stop.
3. Con la placa encendida y el programador conectado (PICKit 3 o similar) programar el archivo con el nombre de *CdT_VA1.hex* utilizando el MPLAB IPE.
 - a. Una vez que advierta que la programación ha finalizado con éxito verificar que el led D7 parpadea cada un segundo.
4. Una vez programada la placa, por la terminal serie se mostrara información indicando la correcta inicialización de los distintos módulos, y un menú de configuración como se puede observar en la imagen a continuación.

```

UARTs Inicializadas...
Modulo USB inicializado...
Memoria Inicializada...
Se cargaron 0 modulos de Control...
No se encontro codigo HTML...
Inicializacion Finalizada

Envie 0x0C para cargar la configuración por defecto
Envie 0x0E para guardar codigo HTML en la memoria
Envie 0x0D para proceder al loop principal
  
```

- Enviar por medio de la terminal serie el valor 0x0E, seguido del código HTML de la página web del servidor, utilizando la opción "dump file to port" ubicada en la pestaña "send".




- Una vez terminado el envío del código, se mostrará por la terminal el mensaje "Codigo HTML guardado satisfactoriamente".
- Enviar el código 0x0C, el cual guardara en memoria la configuración por defecto y reiniciara automáticamente el dispositivo, por lo que se volverá a mostrar por pantalla la información de inicialización. Esta vez deberá de informar que se encontró un módulo de control y código HTML, como se muestra en la siguiente imagen.

```

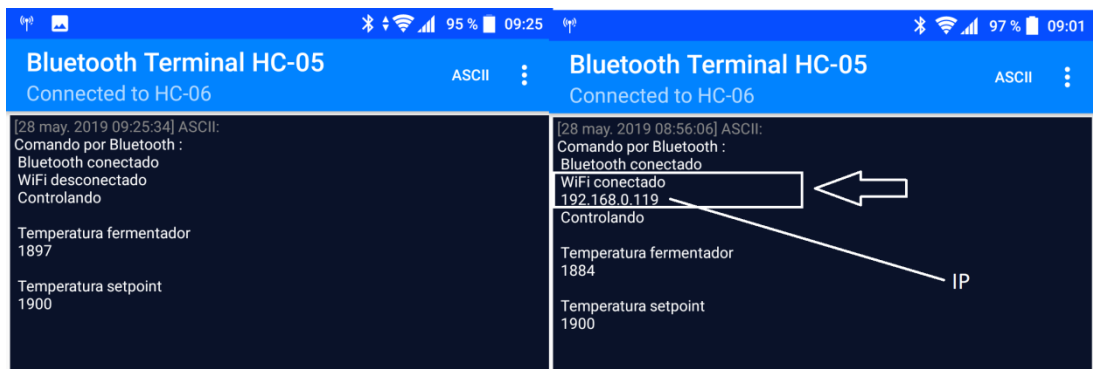
UARTs Inicializadas...
Modulo USB inicializado...
Memoria Inicializada...
Se cargo 1 modulo de Control...
Se encontro codigo HTML...
Inicializacion Finalizada

Envie 0x0C para cargar la configuración por defecto
Envie 0x0E para guardar codigo HTML en la memoria
Envie 0x0D para proceder al loop principal

```

- Conectar un sensor NTC preferentemente de 10kΩ en la bornera P4, utilizando los dos conectores de la derecha y enviar el código 0x0D para proceder al loop principal.
- En el celular, abrir la aplicación  "Bluetooth Terminal HC-05"
 - Si pregunta, permitir que active la conexión Bluetooth.
 - De la lista desplegada debajo de "Paired Devices" seleccionar "HC-06".
 - Enviar "0". Asegurarse que la terminal no adicione los códigos `\n` y/o `\r` al final de la línea.

- d. Si se despliega "WiFi conectado" identificar el número de IP y continuar con el siguiente paso, de lo contrario asegurarse que la red wifi este correctamente configurada y que la intensidad de la señal sea superior a -85dBm y volver a



enviar "0".

e.

10. En una computadora o celular conectado a la misma red wifi, abrir un navegador web y en la barra de navegación introducir la IP obtenida en el paso anterior.

En el navegador se podrá ver el Set-Point, la temperatura actual y la hora en que estos datos se actualizaron por última vez.

Los datos antes mencionados se actualizan cada 4s, si por alguna razón la hora de "Última Actualización" no se actualiza, quiere decir que hubo una falla en el dispositivo, en este caso se debe cerrar la página web, apagar el controlador, esperar 5 segundos y volver a prenderlo, siguiendo nuevamente esta guía de

Control de temperatura

Temperatura [°C]	Set-Point [°C]	Controles de SetPoint
18.95	19	<div style="display: flex; gap: 5px;"> +1°C -1°C +0.1°C -0.1°C </div>
Ultima Actualizacion		
9:16:16		

instrucciones para realizar su conexión mediante wifi.

11. Conectar el cable USB a la placa y la computadora, en la computadora ir a "administrador de dispositivos" e identificar el puerto donde esta conectado el CdT. Si aparece conectado identificar el puerto COM e ir al paso 13. Si aparece con un simbolo de advertencia "Simple CDC Device Demo" ir al paso 12.
12. Hacer click secundario sobre el dispositivo no reconocido y precionar sobre "Actualizar software del controlador", luego seleccionar "Buscar software del controlador en el equipo", dirigirse a la carpeta de instalación de Harmony y seguir la siguiente ruta:

...\microchip\harmony\v2_06\apps\usb\device\cdc_com_port
_single\inf

13. Abrir una nueva instancia de RealTerm en la pc y abrir el puerto COM del USB identificado en el paso 11.
14. En la terminal propia del USB enviar el numero 0x01, el cual iniciara la descarga de los datos registrados por el CdT, desde el momento que se entro al loop principal. Dado que la frecuencia con que se guardan las muestras de de 1 minuto se deberia recibir por la terminal un numero de muestras equivalente a la cantidad de minutos que lleva el CdT en el loop principal.

Al finalizar correctamente los 14 pasos anteriores se comprueba el correcto funcionamiento de la memoria flash, los modulos de wifi y bluetooth, el adc y la comunicación usb. En caso de que alguno de los dispositivos antes mencionados no responda correctamente es necesario realizar una inspección sobre los pines del integrado en cuestion para luego repetir el procedimiento.

XII - Bibliografía

1. **Boulton, Chris and David Quain.** *Brewing yeast and fermentation*. s.l. : Blackwell Science Ltd, 2001.
2. **Salisbury, Alexander.** *MATHEMATICAL MODELS IN POPULATION DYNAMICS*. Sarasota, FL : Division of Natural Sciences, April, 2011. pág. 52.
3. **Caldwell, Alicia.** Caldwell, Alicia, "Students Rise to the Challenge of Modeling Yeast Growth Despite Sour Hiccups from Imperfect Data". [En línea] 2013. [Citado el: 4 de 6 de 2018.] <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1330&context=gradreports>.
4. **Lighthweve, ILX.** "Thermistor Calibration and the Steinhart-Hart Equation, APLICATION NOTE". [En línea] ILX Lighthweve, 2012. [Citado el: 20 de 05 de 2018.] https://www.newport.com/medias/sys_master/images/images/h67/hc1/8797049487390/AN04-Thermistor-Calibration-and-Steinhart-Hart.pdf.
5. **Luis Caspeta, Jens Nielsenb.** "Thermotolerant Yeast Strains Adapted by Laboratory Evolution Show Trade-Off at Ancestral Temperatures and Preadaptation to Other Stresses". [En línea] 21 de 07 de 2015. [Citado el: 5 de 08 de 2018.] http://publications.lib.chalmers.se/records/fulltext/224805/local_224805.pdf.
6. **Guanrong, Chen y Trung, Tat Pham.** *Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems*. Washington D.C : Library of Congress, 2001.
7. **Jantzen, Jan.** *Foundations of Fuzzy Control*. Chichester : John Wiley & Sons Ltd, 2007. ISBN 978-0-470-02963-3.
8. **Viot, Gregory.** *Fuzzy Logic in C. Dr. Dobbs- The world software development*. [En línea] 01 de Febrero de 1993. <http://www.drdoobs.com/cpp/fuzzy-logic-in-c/184408940>.
9. **Areny, Ramón Pallas.** *Sensores y acondicionadores de señal 4ta Edición*. s.l. : Marcombo bolixareu Editores.
10. **Electronics, Koninklijke Philips.** *AN10384: Triacs: How to calculate power and predict Tjmax*. 2005.
11. Mercado Libre Argentina. *Mercado Libre Argentina*. [En línea] www.mercadolibre.com.ar.
12. **UT61, Series.** *Manual Multimetro UT61 Series*. [En línea] [Citado el: 01 de 06 de 2018.] https://www.distrelec.be/Web/Downloads/_t/ds/UT61C_61D_61E_eng_tds.pdf.
13. **US-05, Fermentis-Safale.** *Safale US-05. Safale US-05, Hoja de datos*. [En línea] [Citado el: 15 de 08 de 2017.] <http://www.babrew.com.ar/notas/Safale%20US-05%20-%20Spanish.pdf>.
14. **Alistair D. Rodman, Dimitrios I. Gerogiorgis.** *Dynamic Simulation and Visualisation of Fermentation: Effect of Process Conditions on Beer Quality*. [En línea] [Citado el: 18 de 04 de 2018.] <http://folk.ntnu.no/skoge/prost/proceedings/dycops-cab-2016/proceedings/media/papers/0173.pdf>.
15. **Komorniczak, Michal.** *The bacterial growth curve represents the number of living cells in a population over time*. Wikimedia Commons, s.l. : s.n.
16. **Bruslind, Linda.** *Microbial Growth*. [En línea] Oregon State University, 19 de 9 de 2018. [Citado el: 10 de 3 de 2019.] <http://library.open.oregonstate.edu/microbiology/chapter/microbial-growth/>.
17. **Ulla Saarela, Kauko Leiviskä and Esko Juuso.** *Modelling of a Fed-Batch Fermentation*. Finland : University of Oulu, 2003 - June.