

Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Maestría y Doctorado en Ciencias e Ingeniería



Control de Temperatura en un Medio Confinado

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el
grado de

MAESTRO EN INGENIERÍA

Presenta

EDGAR OMAR CADENA ZEPEDA

Ensenada, Baja California, Enero 2017

Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Control de Temperatura en un Medio Confinado

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN INGENIERÍA

Presenta

Edgar Omar Cadena Zepeda

Aprobada por:



Dr. Manuel Moisés Miranda Velasco

Director de Tesis



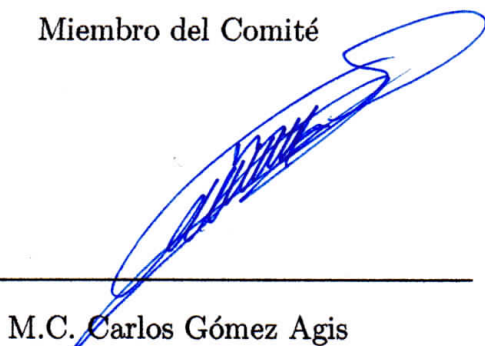
Dr. Miguel Enrique Martínez Rosas

Miembro del Comité



Dr. Horacio Luis Martínez Reyes

Miembro del Comité



M.C. Carlos Gómez Agis

Miembro del Comité



Dr. Juan de Dios Sánchez López

Miembro del Comité

Resumen de la tesis de **Edgar Omar Cadena Zepeda**, presentada como requisito parcial para la obtención del grado de MAESTRO EN INGENIERÍA del programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI) de la UABC. Ensenada Baja California, México, Enero del 2017.

Control de Temperatura en un Medio Confinado

Resumen Aprobado por:



Dr. Manuel Moisés Miranda Velasco

Director de Tesis

En este trabajo se desarrolló e implementó un prototipo de un medio confinado con un control de temperatura que integra telemetría y telecontrol, el sistema también cuenta con consola de control y despliegue de datos mínimos integrada a la caja de control. Se desarrollo una interfaz gráfica para la visualización de los datos medidos, la cual opera desde un servidor web que se ejecuta con el sistema operativo Linux. El usuario tendrá la posibilidad de controlar los parámetros de temperatura e histéresis para la realización de diversos experimentos, así como la posibilidad de monitorizar el estado y parámetros al interior del medio confinado.

El prototipo desarrollado cumple con el objetivo y las metas propuestas al inicio del trabajo y puede ser configurado de acuerdo a las necesidades del usuario final.

Palabras Clave: *temperatura, humedad, control, telemetría*

Dedicatoria

A mi madre Alma Patricia Zepeda, por su impecable papel de mamá, por estar siempre presente, por su constante apoyo emocional y económico, por su complicidad, por su inteligencia, por sus consejos que me han servido en el transcurso de mí andar, por ser la motivación en cada paso que realizó para que de alguna manera devolver todo lo que me ha dado, madre esto es para ti.

A mí querido padre Juan Cadena Jauregui, independientemente de su apoyo económico, siempre ha sido un ejemplo para mí, gracias por enseñarme a ser una persona responsable y trabajadora, por esta ahí apoyarme para seguir superarme académicamente, gracias por tu frase "¡Tú saca esa maestría ya".

A mi hermano André Axel Cadena, el "peque" de la familia por estar en el banco de la paciencia.

A mis familiares, tios, primos y abuelitos por su apoyo moral en el transcurso de mi vida estudiantil dandome ánimos para superarme.

Agradecimientos

Quiero expresar mi agradecimiento:

A mis maestros del posgrado quienes han dado lo mejor de sí, compartiendo sus conocimientos, gracia a sus exigencias he logrado aprender de ellos cosas valiosas, no solo académicas sino personales.

A mi director de tesis Dr. Manuel Moisés Miranda Velasco por sus amistad, apoyo, críticas, observaciones durante el desarrollo de mis estudios de posgrado y la elaboración de esta tesis.

A mis sinodales Dr. Miguel Enrique Martínez Rosas, Dr. Horacio Luis Martínez Reyes, M.C. Carlos Gómez Agis y Dr. Juan de Dios Sánchez López por sus consejos, críticas y apoyo mostrando siempre disposición y tiempo para despejar dudas de temás de interés durante el desarrollo de esta tesis.

A mis compañeros y amigos del posgrado Manuel, Ernesto, Victor por las noches de desvelo terminando trabajos y proyectos de las materias que cursamos.

A mis amigos, Julisa, Daniela, Jessie, Dianna, Frank, Tonny, Daniel, Osmar y Peter por su amistad en los buenos y malos momentos.

Al personal de la Universidad Autónoma de Baja California en especial a la Facultad de Ingeniería, Arquitectura y Diseño por su atenciones, amabilidad y disposición para

ayudarme en lo referente a los trámites escolares.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico brindado, al Cuerpo Académico Comunicaciones e Instrumentación Electrónica” (CACIE) y a la Facultad de Ingeniería, Arquitectura y Diseño (FIAD) de la Universidad Autónoma de Baja California (UABC) por las facilidades otorgadas para la realización de éste trabajo.

Y un agradecimiento especial a la profesora M.C. Juracy Johnson por su confianza, por creer en mí y por su recomendación al ingresar al posgrado.

Índice general

1. Introducción	1
1.1. Justificación	3
1.2. Objetivo General	4
1.3. Objetivos Específicos	4
1.4. Secuencia de la Tesis	4
2. Antecedentes	6
2.1. Cadena de Frío	6
2.1.1. Historia de la Cadena de Frío	7
2.1.2. Principales Usos	7
2.1.3. Poscosecha	8
2.1.4. Importancia de un Ambiente Controlado	9
2.2. Refrigeración	10
2.2.1. Refrigeración por compresión	11
2.2.2. Refrigeración por absorción	13
2.3. Sistemas Frigoríficos	13
2.3.1. Cámaras Frigoríficas	13
2.3.2. Equipo Frigorífico por compresión	14
2.4. Humedad del Aire	16
2.4.1. Humedad Absoluta	17
2.4.2. Humedad Relativa	17

2.5. Medición de Parámetros Ambientales	19
2.5.1. Sensores de Temperatura	19
2.5.2. Sensores de Humedad Relativa	20
2.6. Ethernet	20
2.7. Wi-Fi	21
2.8. Controladores	22
2.8.1. Controlador de acción Todo o Nada (ON/OFF)	22
2.8.2. Controlador de acción Todo o Nada (ON/OFF) con Histéresis	23
2.8.3. Controlador Multivariable	24
3. Control de Variables Ambientales	26
3.1. Propuesta Técnica	26
3.2. Medio Confinado	30
3.2.1. Páneles Interiores	32
3.2.2. Páneles Exteriores	32
3.3. Sistema de Refrigeración	34
3.3.1. Unidad Condensadora	34
3.3.2. Evaporador	35
3.3.3. Gas Refrigerante R134A	36
3.4. Consola de Control	37
3.4.1. Microprocesador	37
3.4.2. Módulo de Temperatura y Humedad Relativa HTM2500LF	38
3.4.3. Circuito de Activación de Etapa de Potencia	44
3.4.4. Interfaz Gráfica en Sitio	47
3.5. Circuito de Protección del Compresor	47
3.6. Sistema de Respaldo Eléctrico	49
3.7. Sistema de Almacenamiento de Datos y Telemetría	50
3.7.1. Aplicación Web	51

4. Pruebas y Resultados	55
4.1. Realización de las Primeras Mediciones	55
4.2. Prototipo Final y Consola de Control	56
4.2.1. Configuración de la Consola de Control	58
4.2.2. Configuración del Sistema de Telemetría	59
4.3. Pruebas en Medio Confinado	60
4.3.1. Mediciones de Temperatura y Humedad	60
4.3.2. Mediciones de Parámetros Eléctricos del Sistema	61
5. Conclusiones Generales	75
5.1. Conclusiones	75
5.2. Aportaciones	76
5.3. Trabajo a Futuro	76
A. Código fuente de sistema de control	78
B. Código fuente de sistema de telemetría	99
C. Esquemático del Circuito de la Consola de Control	113

Índice de figuras

2.1. Esquema donde se representa la etapa de poscosecha.	9
2.2. Ciclo de Refrigeración.	11
2.3. Equipo de Refrigeración por Motocompresor.	15
2.4. Concepto de Humedad.	17
2.5. Humedad Relativa del Aire.	18
2.6. Esquema de Funcionamiento de Control Todo o Nada con Histéresis . .	24
3.1. Sistema con controlador retroalimentado.	27
3.2. Esquema Básico del Prototipo propuesto.	27
3.3. Diseño del prototipo.	29
3.4. Evolución de un medio confinado a un medio aislado.	31
3.5. Paneles tipo Glasliner.	32
3.6. Paneles tipo Foamular.	33
3.7. Montaje del sistema de refrigeración en la estructura.	34
3.8. Unidad condensadora típica para uso comercial.	35
3.9. Equipo evaporador.	36
3.10. Placa de desarrollo Raspberry Pi 2.	38
3.11. Módulo HTM2500LF.	39
3.12. Curva característica del termistor que provee el fabricante.	40
3.13. Interfaz para el termistor y el convertidor analógico-digital.	42
3.14. Convertidor analógico digital MCP3304	42

3.15. Interfaz para el sensor de humedad relativa y el convertidor analógico-digital.	44
3.16. Relé de Estado Sólido.	45
3.17. Circuito de activación de relevador.	46
3.19. Esquema del circuito de protección del compresor.	48
3.20. Compuerta lógica AND	48
3.21. Esquema del Sistema de Respaldo.	50
3.22. Esquema del funcionamiento del Sistema de Telemetría.	51
3.23. Esquema de la Aplicación Web.	52
3.18. Elementos de la Interfaz Gráfica en sitio.	54
4.1. Primer prototipo en placa de pruebas.	56
4.2. Primer prototipo frigorífico.	57
4.3. Unidad Condensadora de 1/3 HP.	58
4.4. Diagrama de flujo aplicación de control.	62
4.5. Circuito impreso de la placa del sistema de control.	63
4.6. Panel de control del prototipo	63
4.7. Página de Inicio.	64
4.8. Página de Monitorización.	65
4.9. Página de Configuración.	66
4.10. Página de Configuración.	67
4.11. Página de Configuración.	68
4.12. Página de Acerca del Cuerpo Académico CACIE.	69
4.13. Mediciones de Temperatura.	70
4.14. Mediciones de Temperatura.	71
4.15. Consumo de corriente eléctrica del sistema.	72
4.16. Consumo de potencia eléctrica del sistema.	73
4.17. Voltaje del sistema.	74
C.1. Esquema del circuito del prototipo.	114

Índice de tablas

3.1. Equivalencia de resistencia vs temperatura obtenida de la hoja de datos.	41
3.2. Respuesta típica en mV del sensor alimentado a 5V.	43

Capítulo 1

Introducción

Actualmente la agricultura demanda de la aplicación de técnicas adecuadas que permitan obtener y comercializar productos de calidad, a los menores costos, con el aprovechamiento de insumos de la zona, que sean amigables con el medio ambiente.

La importancia de la agricultura en el estado de Baja California, es tal que se encuentra en el segundo lugar en producción nacional de tomate rojo, cebolla, fresa, aceituna y dátil. Además de ser el tercer lugar en pepino, espárrago y uva según constan datos del censo económico INEGI (2010) [1], contando con 66,270 hectáreas de aseguramiento agrícola.

Para considerar que un producto agrícola es de buena calidad, éste debe de presentar un buen aspecto visual(color, forma y tamaño) y mantener sus propiedades nutricionales óptimas desde su cosecha hasta su consumo. El no conservar estas características conlleva un riesgo para su comercialización y consumo. Es tal la importancia de la calidad de estos productos que desde el año de 1979 la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) edita un manual técnico titulado "Alimentos para la exportación" [2], el cual se ha convertido en un referente esencial para gobiernos y empresas interesados en garantizar la alta calidad de sus productos agrícolas.

Los productos agrícolas tienen una tendencia inherente a deteriorarse por razones

fisiológicas, por esta razón son denominados productos perecederos. Por lo tanto una vez cosechados se da inicio a un proceso denominado etapa de poscosecha en la cual pasan por unacinspección donde son preparados para su procesamiento mediante una limpieza, preenfriamiento, clasificación, y almacenamiento.

En la etapa de la poscosecha se busca preservar la integridad físico-química de los productos frescos después de ser cortados, es por esto y dada la globalización de los procesos comerciales de exportación que existe la "cadena de frío", esta consiste en una cadena de suministro con temperatura controlada, que inicia desde que el producto se corta, pasando por su distribución por cualquier medio de transporte hasta llegar al estante donde se pondrá en venta al consumidor [3].

Durante el periodo de poscosecha, el control de temperatura es el factor más importante para mantener la calidad de los productos. Es conocido que la variación del valor de la temperatura óptima a la que se debe mantener cada producto en cualquiera de las etapas de la poscosecha, es perjudicial para las propiedades nutricionales y calidad del mismo [4].

En México conviven dos estructuras de poscosecha muy distintas:

- Por una parte, un segmento utiliza de modo cotidiano la cadena de frío para sus productos, por lo general relacionado con los supermercados o sectores de exportación, a los que provee de productos perecederos más seguros y de mayor calidad.
- Por otra parte, hay segmentos tradicionales del mercado, que no recurren a las cadenas de frío. Estas prácticas se encuentran por lo general en los mercados populares. Más de 50 % de los productos perecederos consumidos en México se comercializan sin los beneficios que ofrecen las cadenas de frío [5].

Por lo cual existe una necesidad del sector agrícola local de conocer el comportamiento de sus productos durante la etapa de poscosecha, es decir que se requiere identificar cómo afectan los cambios de las variables ambientales a través del tiempo a los productos desde son cosechados hasta que son adquiridos para el consumo final.

1.1. Justificación

Cuando se separan de la planta madre, las frutas, hortalizas y flores son aún tejidos vivos que respiran. El mantener el producto a la temperatura más baja posible permite conservar las propiedades fisico-químicas del mismo, ya que las temperaturas bajas disminuyen la tasa de respiración, reduciendo además la pérdida de agua [6].

Estimaciones realizadas por SAGARPA (2013) revelan que las pérdidas físicas y de calidad generadas por la incorrecta manipulación, almacenamiento e insuficiente infraestructura de transporte, producen una pérdida total de hasta el 30 % sobre la producción total de productos agrícolas [7].

Es necesario que se le de la importancia que estas cifras arrojan cada año, ocasionadas por condiciones climáticas adversas y por un mal manejo, durante la cosecha como en el período de poscosecha. Por tanto reducir las pérdidas de los productos agrícolas durante su almacenamiento, transporte, distribución y comercialización es de interés nacional como se hace constar en el Plan Nacional México sin Hambre 2013-2018.[8]

Además los miembros del Cuerpo Académico de Comunicaciones e Instrumentación Electrónica (CACIE) en la Facultad de Ingeniería, Arquitectura y Diseño de la Universidad Autónoma de Baja California, trabajan activamente en el diseño y desarrollo de sistemas de medición basados en tecnología electrónica, a través del desarrollo de sistemas de medición de variables ambientales relacionadas con el proceso agropecuario, integrando sistemas de manejo y transmisión de datos con el objetivo de crear bitácoras digitales para la producción con énfasis en aplicaciones agrícolas.

La idea de realizar este proyecto, es concebida debido a que anteriormente se contaba con un medio confinado para el estudio y monitorización de productos agrícolas, pero existe la necesidad de contar con un medio que además de estar confinado, cuente con un aislamiento adecuado y una atmósfera controlada, esta ultima aplicación surge de la necesidad manifestada por lo productores, de tecnificar su producción y poder competir de mejor manera en el mercado global. Es importante señalar, que la aplicación del sistema desarrollado no se restringe a éste sector, por lo que pueden desarrollarse

nuevas aplicaciones en función de las necesidades de distintos sectores.

1.2. Objetivo General

Diseñar y elaborar un prototipo de un medio confinado, con un control automático de temperatura y el monitoreo de la humedad relativa del aire.

1.3. Objetivos Específicos

- *Diseñar y elaborar un medio confinado y aislado para emular condiciones de pos-cosecha de los productos agrícolas.*
- *Elaborar una consola de control en el prototipo.*
- *Diseñar y elaborar un sistema informático de telemetría.*
- *Validar el prototipo.*

1.4. Secuencia de la Tesis

Este trabajo está dividido en los siguientes capítulos:

Capítulo 2 En el capítulo 2 se presenta el marco teórico, incluyendo definiciones y la descripción detallada del objeto bajo estudio, así como las etapas que constituyen el desarrollo del control de temperatura y el monitoreo de la humedad relativa.

Capítulo 3 En el capítulo 3 se describe el diseño e implementación del sistema de control desarrollado, se incluye la información considerada para la implementación física en el sistema y el desarrollo de la interfaz gráfica.

Capítulo 4 En éste capítulo se presentan los resultados obtenidos en la aplicación práctica del prototipo, se indican las condiciones en que fueron realizadas las mediciones y se da una interpretación de las mismas.

Capítulo 5 Finalmente, en este capítulo se dan las conclusiones del trabajo.

Capítulo 2

Antecedentes

Es este capítulo se definirán de manera sencilla los conceptos básicos que se utilizaron en el desarrollo del trabajo; Primeramente se define el concepto de *cadena de frío* y su importancia en los procesos agropecuarios. Posteriormente se explican la refrigeración y sus sistemas de aplicación, los medios de medición de los parámetros de la temperatura y humedad relativa del aire, y los protocolos de comunicación para su monitorización. Por último se explica el control on/off que es uno de los más utilizados dentro de los sistemas de enfriamiento.

2.1. Cadena de Frío

Se define una cadena de frío como una cadena de suministro de productos y mercancías susceptibles al cambio de temperatura, es decir, es una serie de etapas en las que los parámetros ambientales deben mantenerse constantes. Esto se realiza mediante empaquetados aislantes, métodos de refrigeración y planeación de los productos, desde que son elaborados o cosechados hasta que son utilizados en procesos productivos o consumidos por el usuario final [9].

Las cadenas de frío modernas surgen de la necesidad de transportar productos perecederos a mercados lejanos sin que se descompongan, pues de otra manera la venta de estos productos quedaría limitada al lugar de origen.

2.1.1. Historia de la Cadena de Frío

Antes de que existieran las tecnologías de refrigeración actuales, pequeñas cantidades de productos perecederos podían ser transportados mediante procesos muy caros que respondían a las necesidades o caprichos de antiguos reyes o emperadores.

Por ejemplo, se ha documentado que el emperador mexica Moctezuma recibía pescado fresco desde las costas de Veracruz por medio de una cadena de mensajeros que lo transportaban desde la costa. De igual modo, otras culturas antiguas recurrían al almacenamiento de hielo, en cavernas durante las temporadas de invierno con el objetivo de consumir los alimentos de temporadas calurosas, a esto se le llama cosecha de hielo.

Ya en tiempos mas recientes, durante el siglo XIX, en Estados Unidos de Norteamérica surgió la industria de cosecha de hielo y ésta creció de gran manera, gracias al desarrollo de transportes con sistemas de aislamiento térmico. Desde principios del siglo se habían fabricado máquinas de refrigeración que funcionaban con vapor, que evolucionaron hasta que en 1855 se creó el primer sistema de refrigeración por compresión. Esta tecnología fue rápidamente adoptada para la exportación de carnes a Europa.

Durante el siglo XX hubo muchos avances y mejoras en los sistemas de refrigeración, básicamente debido a las necesidades de la industria química moderna y la atención médica que requieren transportar por igual compuestos químicos, muestras médicas o tejidos biológicos.

Actualmente las cadenas de frío son parte fundamental de la vida moderna, dada la importancia que tienen para transportar productos perecederos, como alimentos y medicinas, a grandes distancias; por lo que son esenciales para el crecimiento económico de una nación [9].

2.1.2. Principales Usos

Hospitales, centros y puestos de salud tienen unidades de refrigeración para mantener las vacunas y transportarlas a los puestos de vacunación. Agricultores y empresas de abastecimiento de alimentos cuentan con una logística altamente desarrollada para

el transporte de sus productos por medios refrigerados.

Por otra parte, en cuanto a las principales mercancías transportadas, se trata de productos alimenticios perecederos, como carnes, mariscos, pescados, frutas y verduras; también se trasladan medicamentos, químicos y flores. En el caso del comercio exterior, las exportaciones suelen ser de frutas y verduras, y las importaciones se componen en general por alimentos procesados que requieren refrigeración (embutidos, platos preparados, lácteos, entre otros).

Son los productos químicos, médicos, flores y hortalizas los que demandan mayor calidad en los servicios de transporte refrigerado, ya sea porque se trata de productos peligrosos y de manejo delicado, o bien porque tienen alto valor para la exportación.

2.1.3. Poscosecha

Después de cosecha, los vegetales son los más altamente perecederos y deben ser removidos del campo lo más rápido posible y ser refrigerados o deben ser clasificados y empaquetados para la comercialización.

La maduración y el deterioro de los productos inicia desde el momento de la cosecha, por tanto la vida comercial depende de la temperatura y de cuidado del manejo físico de los mismos, en la figura 2.1 se ilustra de forma general todo el proceso de poscosecha ideal.

Durante el proceso de poscosecha es importante cuidar varios aspectos en el manejo de los productos, ya que una manipulación inadecuada generará efectos que son acumulativos. Varias contusiones pequeñas pueden producir un mal gusto en el alimento. La contusión también estimula el índice de maduración y de tal modo acorta su almacenaje potencial y vida útil, además las contusiones generan pérdida de humedad en los productos [10].

La calidad de los productos se ve favorecida por:

- Cosechar en la madurez o en la calidad óptima.
- Manipular cuidadosamente para evitar lesión mecánica.

- Manipular rápidamente para reducir al mínimo el deterioro.
- Abastecer de envases y empaques protectores.
- Usar: producto químico, tratamiento por calor o tratamientos de atmósfera modificada.
- Hacer cumplir buenos procedimientos del saneamiento en planta.
- Pre-enfriar para quitar calor del campo.
- Abastecer de alta humedad relativa para reducir al mínimo pérdida de humedad
- Abastecer de refrigeración apropiada a través de la comercialización.

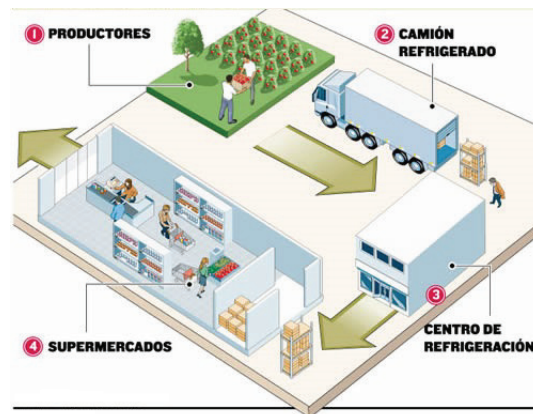


Figura 2.1: Esquema donde se representa la etapa de poscosecha.

2.1.4. Importancia de un Ambiente Controlado

La función primordial de una cadena de frío, y de la cual se deriva su importancia económica y social, es la de conservar las características del producto durante un largo tiempo. Esta función tan básica pero a la vez compleja permite a los empresarios conservar el valor de una mercancía y venderla en cualquier lugar sin que pierda su valor monetario[10].

El enfriamiento rápido después de la cosecha, antes o después del empaquetado, o antes de que se almacene o se transporte, reduce significativamente el deterioro de los vegetales más perecederos. Si el calor de campo se remueve más rápidamente, el producto se puede mantener en buenas condiciones comerciales por mucho más tiempo.

El enfriamiento retarda el deterioro natural, incluyendo el envejecimiento y la maduración; retarda el crecimiento de organismos del decaimiento (y de tal modo el desarrollo de la putrefacción) y reduce la marchites. Esto se debe a que la pérdida de agua ocurre de forma mucho más lenta a bajas temperaturas.

Después de enfriarse, el producto se debe refrigerar a las temperaturas recomendadas, a continuación algunos ejemplos:

- Espárragos (32 a 36 °F ó 0 a 2.2 °C y 95 a 100 % HR)
- Brócoli (32 °F/0 °C y 95 a 100 % HR)
- Repollo (32 °F y 98 a 100 % HR)
- Zanahorias (32 °F ó 0 °C y 98 a 100 % HR)
- Coliflor (32 °F/0 °C y 95 % HR)
- Maíz tierno o elote (32 °F/0 °C y 95 a 98 % HR)
- Pepinos (50 a 55 °F/10 a 13 °C y 95 % HR)
- Lechuga (32 °F/0 °C y 95 a 100 % HR)
- Cebollas (32 °F/0 °C Y 65 a 70 % HR)

2.2. Refrigeración

El proceso de refrigeración consiste en hacer que en una cámara confinada la temperatura llegue a un valor deseado y se mantenga dentro de un margen establecido alrededor del mismo.

Como la cámara no es perfectamente adiabática, para mantener la temperatura, es preciso extraer calor del recinto de forma continua o de manera intermitente y luego enfriarlo.

2.2.1. Refrigeración por compresión

La refrigeración por compresión es un método de refrigeración que consiste en forzar mecánicamente la circulación de un refrigerante en un circuito cerrado dividido en dos zonas, como se aprecia en la figura 2.2, una de alta y otra de baja presión, con el propósito de que el fluido absorba calor del ambiente, en el evaporador en la zona de baja presión y lo ceda en la de alta presión, en el condensador.

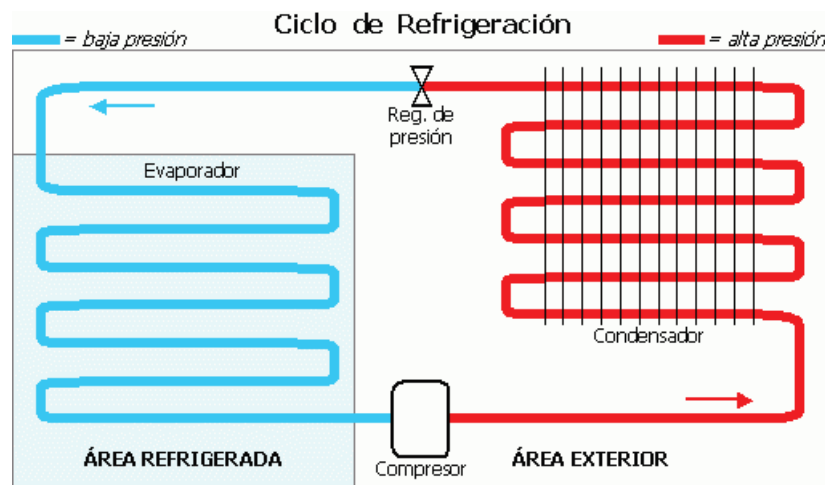


Figura 2.2: Ciclo de Refrigeración.

La refrigeración por compresión se basa en el aprovechamiento de las propiedades de ciertos fluidos, llamados refrigerantes o fluidos frigorígenos, una de sus propiedades es que su temperatura de vaporización a presión atmosférica es extremadamente baja. Los refrigerantes utilizados comúnmente, tienen temperaturas de ebullición en condiciones normales, alrededor de -40°C .

Al incluir un refrigerante con esas características en un circuito frigorífico, se eleva su presión y temperatura, mediante un compresor en un proceso isentrópico, hasta

alcanzar la presión de condensación. En esas condiciones el fluido atraviesa el condensador mientras intercambia calor con el medio exterior. Como consecuencia de la cesión de calor se produce la condensación del fluido, que sale del condensador y alcanza la válvula de expansión totalmente en estado líquido. Esta última parte del proceso, se puede considerar isotérmica, ya que no varía la temperatura durante el cambio de estado. El tramo del circuito comprendido entre el compresor y la válvula de expansión, se conoce como lado de alta o zona de alta presión.

El dispositivo de expansión provoca una caída repentina de la presión y la temperatura sin intercambio de calor, por lo que esta parte del proceso se puede considerar, idealmente, como una transformación adiabática o isoentálpica. El fluido todavía en estado líquido y a la presión de vaporización penetra en el evaporador, intercambiador de calor ubicado en el medio que se pretende enfriar, del cual absorbe la energía térmica correspondiente al calor latente de vaporización, de forma que el fluido sale del evaporador completamente en estado de vapor. La transformación se puede considerar isotérmica por la misma razón que se dio en el condensador. El tramo del circuito desde la válvula de expansión hasta el compresor se conoce como lado de baja o zona de baja presión. A continuación, el vapor es aspirado por el compresor para iniciar de nuevo el ciclo.

En el ciclo de refrigeración ideal, en los balances de energía del equipo, se desprecia cualquier pérdida o ganancia de calor en las tuberías, considerando que los únicos intercambios de calor que se producen en el sistema, ocurren en el evaporador y en el condensador. Sin embargo, en el ciclo real el fluido refrigerante sufre una ligera caída de presión y temperatura debido a las pérdidas por fricción, sobre todo en evaporador y condensador y en las restricciones de las válvulas de admisión y de escape. Esto hace que el ciclo real resulte ligeramente distorsionado respecto del ciclo ideal.

2.2.2. Refrigeración por absorción

El sistema de refrigeración por absorción es un medio de producir frío que, al igual que en el sistema de refrigeración por compresión, aprovecha que las sustancias absorben calor al cambiar de estado, de líquido a gaseoso. Así como en el sistema de compresión el ciclo se hace mediante un compresor, en el caso de la absorción, el ciclo se basa físicamente en la capacidad que tienen algunas sustancias, como el bromuro de litio, de absorber otra sustancia, tal como el agua, en fase de vapor. Otra posibilidad es emplear el agua como sustancia absorbente (disolvente) y amoníaco como sustancia absorbida (soluto).

La técnica nació en 1859, cuando Ferdinand Carré consiguió fabricar hielo con la primera máquina de absorción de ciclo amoníaco-agua.

2.3. Sistemas Frigoríficos

Los sistemas frigoríficos básicamente están integrados por el medio térmicamente aislado en el cual se va a controlar su temperatura, denominado Cámara Frigorífica y el Equipo Frigorífico, que está encargado de controlar la temperatura, empleando un método adecuado de refrigeración.

2.3.1. Cámaras Frigoríficas

Desde la prehistoria, el ser humano sabía hacer fuego y calentar, entregar calor a frutas y hortalizas, pero enfriar, retirar calor, lo aprendió hace poco tiempo. El físico francés Sadi Carnot en los años 20 del siglo XIX, estudiando la máquina de vapor inventada por NOMBRE Watt había creado en FECHA en Inglaterra, desarrolló teorías que fueron la base de la Termodinámica y planteó las fórmulas de la máquina frigorífica.

Posteriormente otros físicos hicieron ensayos de máquinas cada vez más perfeccionados y probando diferentes gases hasta que recién a fines del siglo XIX se construyeron los

primeros frigoríficos. En 1928 se comenzó con la fabricación en masa de refrigeradoras domésticas y luego camiones, vagones de tren y barcos frigoríficos.

Para el almacenamiento de vegetales para su posterior comercialización se desarrollaron instalaciones industriales llamadas cámaras frigoríficas en la cual se regulan las variables físicas del ambiente, normalmente son la temperatura, humedad y circulación del aire [11].

Esta técnica se denomina atmósfera controlada. En años recientes, se ha enfocado una mayor atención a las tecnologías poscosecha que permitan mantener la calidad para alcanzar mercados lejanos con una mejor presentación [12].

Actualmente existen en el mercado contenedores frigoríficos para traslado internacional de productos perecederos y farmacéuticos. Normalmente estos solamente controlan la temperatura en el interior y cuentan con un sistema de ventilación para el flujo de aire en el interior y una pantalla de monitorización de la temperatura y humedad, pero no cuentan con un control de la humedad relativa [13].

2.3.2. Equipo Frigorífico por compresión

Es una máquina térmica diseñada para tomar la energía calorífica de un área específica y evacuarla a otra mediante un proceso termodinámico. Para su funcionamiento, según el Segundo Principio de la Termodinámica, primero es necesario aplicar un trabajo externo, por lo cual el equipo, sea cual sea su principio de funcionamiento, consumirá energía.

Cabe mencionar que este tipo de equipos son los más utilizados en el mercado, de tal forma que se definen un abanico de posibilidades al realizar configuraciones en función de temperatura de trabajo, potencia, caudal de aire, tipo de instalación, volumen de control y otras variables. La figura 2.3 describe su funcionamiento.

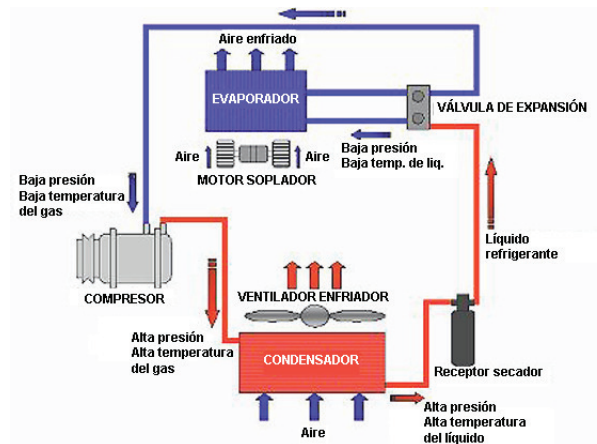


Figura 2.3: Equipo de Refrigeración por Motocompresor.

Los elementos en un equipo frigorífico se pueden agrupar en necesario y opcional, los elementos necesarios son imprescindibles para el funcionamiento, mientras que los opcionales son requeridos en función de la cámara frigorífica. Los elementos necesarios son:

Compresor: es un dispositivo mecánico que bombea el fluido refrigerante, creando una zona de alta presión y provocando el movimiento del refrigerante en el sistema.

Condensador: generalmente es un serpentín de cobre con laminillas de aluminio a modo de disipadores de calor. Su función es liberar el calor del refrigerante al ambiente.

Evaporador: también es un serpentín, pero su presentación varía. El de los equipos de acondicionamiento de aire es muy similar al condensador, pero en los refrigeradores domésticos suele ir oculto en las paredes del congelador. Su función es que el refrigerante absorba calor del área refrigerada.

Dispositivo de expansión: según el caso puede ser una válvula de expansión o un tubo capilar. En cualquier caso, es un punto donde hay una pérdida de carga muy grande, por reducción de la sección de paso; su función es dejar que el refrigerante pase desde la parte del circuito de alta presión a la de baja presión, expandiéndose.

Los elementos opcionales que pueden ser:

Termostato: su función es apagar o encender automáticamente el compresor a fin de mantener el área enfriada dentro de un campo de temperaturas.

Ventilador: su función es aumentar el flujo de aire para mejorar el intercambio de calor. Generalmente está en el área del condensador. Según el tipo de dispositivo que sea, puede haber ventilador (evaporador de aire forzado) o no (evaporador estático) en el área del evaporador.

Filtro de humedad: su función es disminuir o mitigar la humedad en la cámara frigorífica para tener un ambiente con poca humedad.

Depósito de refrigerante líquido

Otro elemento fundamental en estos aparatos es el dispositivo de disipación de calor al exterior, que puede ir desde un simple intercambiador con un ventilador, hasta una torre de enfriamiento.

2.4. Humedad del Aire

El agua es la única sustancia de la atmósfera que puede condensar o evaporarse en las condiciones ambientales que conocemos en la Tierra. Este hecho justifica la división del aire atmosférico, en aire seco y agua, además provocan una gran cantidad de fenómenos meteorológicos como la lluvia, el rocío, las nubes, entre otros.

Un grado de humedad inadecuado (demasiado alto o demasiado bajo) puede causar molestias a algunas personas y deteriorar un gran número de equipos y materiales.

La capacidad de la atmósfera para recibir vapor de agua se relaciona con los conceptos de humedad absoluta y humedad relativa, en la figura [2.4](#) se muestra la relación entre estas.

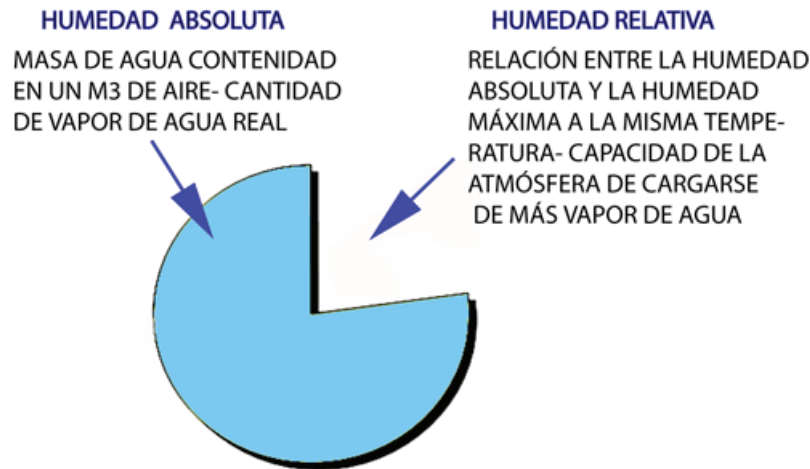


Figura 2.4: Concepto de Humedad.

La humedad del aire se debe al vapor de agua que se encuentra presente en la atmósfera. El aire de la atmósfera se considera normalmente como una mezcla de dos componentes: aire seco y agua. El vapor procede de la evaporación de los mares y océanos, de los ríos, los lagos, las plantas y otros seres vivos. La cantidad de vapor de agua que puede absorber el aire depende de su temperatura, además el aire caliente admite más vapor de agua que el aire frío.

2.4.1. Humedad Absoluta

La humedad absoluta $[g/m^3]$ es la cantidad de agua encontrada en un volumen cerrado de $1m^3$.

$$\text{Humedad absoluta} = \frac{\text{Peso de agua}}{\text{volumen de aire}} [g/m^3] \quad (2.1)$$

2.4.2. Humedad Relativa

Es el cociente de la humedad absoluta y la cantidad máxima de agua que admite el aire por unidad de volumen. Se mide en tantos por ciento y está normalizada de forma que la humedad relativa máxima posible es el 100 %, una humedad relativa del

100 % significa un ambiente en el que no cabe más agua. En la figura 2.5 se muestra el cambio de la humedad relativa ante una humedad absoluta constante y un cambio en la temperatura.

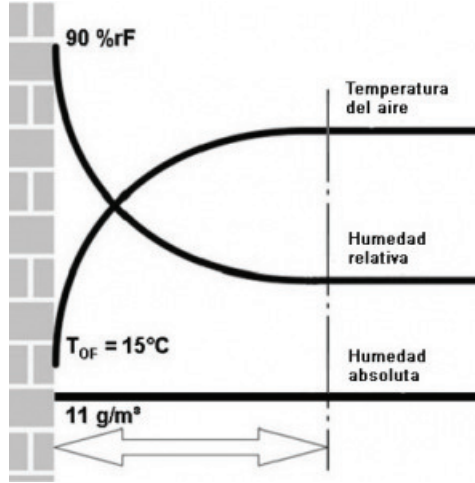


Figura 2.5: Humedad Relativa del Aire.

La humedad relativa F es la relación entre la masa real de vapor de agua en el aire comparada con la masa máxima posible de vapor de agua en el aire:

$$F = \frac{\text{humedad absoluta}}{\text{humedad máxima (indicada normalmente en porcentaje)}}. \quad (2.2)$$

donde F es la humedad relativa, f es la humedad absoluta y f_{max} representa humedad máxima o humedad de saturación, entonces

$$F = \frac{f}{f_{max}} * 100\% \quad (2.3)$$

Como la humedad máxima f_{max} depende de la temperatura, la temperatura cambia con la humedad relativa, aún cuando la humedad absoluta permanezca constante. La humedad relativa aumenta al 100 % cuando se enfría hasta el punto de rocío.

La humedad relativa es una cifra porcentual que especifica el porcentaje de la cantidad máxima posible de vapor de agua actualmente en el aire.

2.5. Medición de Parámetros Ambientales

En las áreas de ciencias exactas es necesario presentar los resultados a través de sistemas de medición que nos permitan entender los estudios y resultados. En este caso las variables a medir son la temperatura y la humedad relativa dentro del medio confinado.

La medición y el control de temperatura y humedad relativa es una aplicación de ingeniería de control sumamente estudiada, actualmente existen aplicaciones con redes de sensores alámbricas e inalámbricas con monitoreo remoto [6] [14].

2.5.1. Sensores de Temperatura

Dentro de los sensores de temperatura podemos destacar los siguientes debido a su amplio uso dentro de los sistemas electrónicos: **Termistores:** Los termistores PTC son resistencias que principalmente están compuestas de bario y estroncio con titanio. La adición de dopantes hace que la componente semiconductora dé una característica de resistencia con respecto a la temperatura, aunque son muy poco utilizados.

Sensores de Temperatura con Salida en Tensión: Es el sensor más elemental que entrega directamente una tensión proporcional a la temperatura. La mayoría no necesitan calibración o ajuste externo asegurando una precisión mejor que 1°C en todo el rango completo de trabajo. Su bajo consumo elimina los efectos indeseados debidos al autocalentamiento, que puede cuantificarse menor que 0.1°C sobre la temperatura ambiente.

Sensores de Temperatura con Salida Digital: La incorporación de dispositivos digitales integrados junto con el sensor reduce el número de componentes externos en las aplicaciones. Así, los circuitos de estos sensores, proporcionan una salida digital serie con modulación de ancho de pulsos, PWM.

Sensores de Silicio con efecto Resistivo: Los sensores de temperatura de silicio muestran una característica casi lineal comparada con la característica exponencial de las NTC. Esto significa que tienen un coeficiente de temperatura que es casi constante

en todo el alcance dinámico de temperatura. Esta característica se puede utilizar cuando el sensor se usa como compensación de temperatura para un microprocesador con conversor de A/D integrado.

2.5.2. Sensores de Humedad Relativa

Existen distintos sensores de humedad, entre los cuales podemos destacar los siguientes en base a su principio de funcionamiento: **Sensores de Humedad Capacitivos:** El sensor lo forma un condensador de dos láminas de oro como placas y como dieléctrico una lámina no conductora que varía su constante dieléctrica, en función de la humedad relativa de la atmósfera ambiente. El valor de la capacidad se mide como humedad relativa.

Sensores de Humedad Resistivos: Un electrodo polímero montado sensa la humedad en el material. Además un circuito acondicionador da una salida estándar.

Estos sensores proporcionan una señal de tensión proporcional a la humedad relativa y puede ser acondicionada por la entrada del convertidor A/D de cualquier microcontrolador.

2.6. Ethernet

El protocolo IEEE 802.3 o Ethernet es un estándar del Instituto de Ingeniería Eléctrica y Electrónica (IEEE, por sus siglas en inglés) para redes de área local de computadoras con acceso al medio por detección de la onda portadora y con detección de colisiones (CSMA/CD). Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

Los orígenes de la tecnología Ethernet se puede rastrear prácticamente hasta principios de la década de 1970, siendo Robert Metcalfe, un ingeniero graduado en MIT y la compañía Xerox los principales precursores de ella. En la actualidad es el méto-

do más simple, seguro, y económico de montar una red entre computadoras, debido fundamentalmente a su flexibilidad, ya que entre otras tantas características es posible utilizarse desde cable coaxial hasta fibra óptica para poder implementar una red con esta tecnología.

La idea básica detrás de Ethernet es que todas las PC's dentro de una red envíen y reciban datos de una forma en que se evite cualquier tipo de superposición. Es por eso que los datos que se envían o reciben mediante este estándar deben ser fragmentados en fracciones más pequeñas y enviados a través de un método conocido como Conmutación de paquetes.

Básicamente esto consiste en que si una de las PC de la red quiere enviar un paquete de datos a otra, debe ser empaquetado, lo que finalmente arroja como resultado un paquete, el cual consiste de varios datos tales como cabecera, dirección del dispositivo en la red a quién va destinado y qué dispositivo de la red lo está enviando. Además contiene datos de control y otras informaciones relativas al mismo como la cantidad de datos que transporta y otros.

Cabe destacar que estos paquetes se envían a todas los dispositivos que conforman la red, siendo los propios equipos los que determinan si el paquete va dirigido a ellos o no, denegando todos los paquetes que no se dirigen a ellos.

Para evitar que estos paquetes colisionarán, se creó CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Mediante CSMA/CD, es posible que los dispositivos escuchen la red para determinar si el canal y los recursos se encuentran libres. En caso afirmativo, se podrá realizar la transmisión para no colisionar con otros paquetes.

2.7. Wi-Fi

El protocolo IEEE 802.11 o Wi-Fi es un estándar de protocolo de comunicaciones del IEEE que define el uso de los dos niveles mas bajos del modelo OSI (capa física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN. En general los protocolos de la rama 802.11x definen la tecnología de redes de área local.

El estándar original de este protocolo data de 1997, el cual alcanzaba velocidades de 1 hasta 2 Mbps y trabajaba en la banda de frecuencia de 2.4 GHz con una modulación de señal de espectro expandido por secuencia directa (DSSS), o con espectro expandido por salto de frecuencia (FHSS).

Una red Wi-Fi permite conectarse a servidores, PC, impresoras, entre otros dispositivos, con la particularidad de hacerlo sin necesidad de cableado.

De manera purista vale a decir que el acrónimo Wi-Fi se utiliza para identificar los productos que incorporan cualquier variación de la tecnología sin cables (wireless) de los estándares IEEE 802.11, que permiten la creación de redes de área local sin cables conocidas como WLAN4, y que son plenamente compatibles con los de cualquier otro fabricante que utilice estos estándares. Las características generales de funcionamiento de una red Wi-Fi son las mismas que las de una red con cableado. La particularidad es que el Wi-Fi utiliza el aire como medio de transmisión.[15]

2.8. Controladores

2.8.1. Controlador de acción Todo o Nada (ON/OFF)

Es la regulación más simple y económica, aplicable en sistemas en los que puede admitirse una oscilación continua entre dos límites, siempre y cuando se trate de procesos de evolución lenta. Como ejemplos podemos citar la regulación de nivel, de presión o de temperatura, todos ellos problemas relativamente sencillos de lógica digital. Numerosos reguladores incorporan esta función básica, que además ofrece la máxima rapidez de respuesta y en ocasiones se recurre a este tipo de control cuando el error es grande, y se pasa de forma automática a otro tipo de regulación cuando el error se aproxima a cero [16].

Estos envían una señal de activación (ON, encendido o 1) cuando la señal de entrada es menor que un nivel de referencia (definido previamente), y desactivan la señal de salida (OFF, apagado o 0) cuando la señal de entrada es mayor que la señal de referencia.

Los controladores ON/OFF son utilizados en termostatos de aire acondicionado. Estos activan el aire frío (ON) cuando la temperatura es mayor que la de referencia (la de preferencia del usuario) y lo desactivan (OFF) cuando la temperatura ya es menor (o igual) que la de referencia.

Los controladores mecánicos de dos posiciones normalmente posee algo de histéresis, por el contrario los controladores electrónicos usualmente funcionan sin histéresis.

2.8.2. Controlador de acción Todo o Nada (ON/OFF) con Histéresis

El controlador Todo o Nada tiene un simple mecanismo de construcción, por esa razón su uso es bastante común en sistemas de regulación de temperatura. Su funcionamiento da como resultado una oscilación de alta frecuencia en la variable a controlar, las cuales dañan los componentes de los sistemas electromecánicos, o recortan la vida útil de los mismos.

La histéresis sirve para disminuir las oscilaciones de alta frecuencia de un sistema todo o nada y de esta forma mejorar el desempeño del sistema a controlar. Pero esta estabilización en los sistemas de control de temperatura, hace que la temperatura se oscile alrededor de la de referencia, ya que la temperatura continúa aumentando antes de descender cuando la carga ya no es alimentada, y después que la alimentación sea restablecida aún sigue disminuyendo antes de aumentar. Por tanto, se tiene siempre un rebasamiento de los límites de temperatura de la histéresis, que no puede ser reducida más allá de ciertos límites con la consecuencia de producir oscilaciones en los circuitos electrónicos. El comportamiento de la variable a controlar con esta clase de control se muestra en la figura [2.6](#).

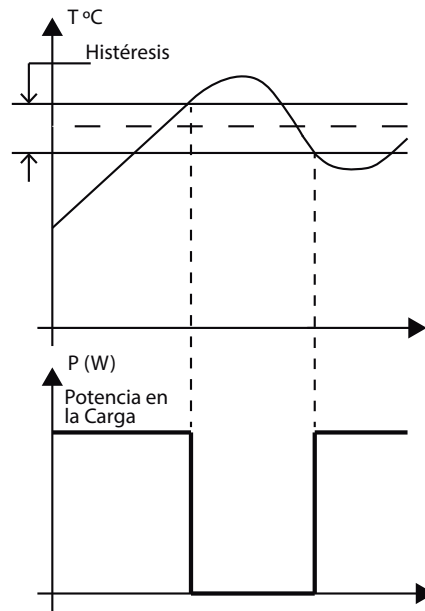


Figura 2.6: Esquema de Funcionamiento de Control Todo o Nada con Histéresis

El control ON/OFF con histéresis se usa generalmente en cámaras frigoríficas donde la temperatura debe estar dentro de un rango y se desea activar y desactivar lo menos posible los motores del sistema de refrigeración.

2.8.3. Controlador Multivariable

Se aplica cuando existen dos o más variables que están acopladas, o lo que es lo mismo, cuando la variación de una ejerce una variación en otra. Como ejemplo se puede citar la climatización de una sala en la cual es preciso controlar la temperatura y la humedad relativa del aire. Si la temperatura de la sala desciende, la humedad relativa sube, puesto que el aire está más frío. Por otra parte, si se introdujera vapor a la sala, no solo aumentaría la humedad, sino que también lo haría la temperatura.

El regulador que controla la inyección de vapor y el que controla la temperatura (por enfriamiento) deberían integrarse para conseguir un objetivo común más que intentar controlar sus respectivas variables por separado. Con el algoritmo adecuado puede lograrse no solo que los dos reguladores operen sin contradicciones, sino también

minimizar el consumo energético o en general optimizar cualquier otra variable de que se disponga.

Capítulo 3

Control de Variables Ambientales

3.1. Propuesta Técnica

Para entender el proceso de refrigeración en este medio confinado primero definiremos los siguientes conceptos de interés[17].:

- Energía Térmica: es la energía interna total de un objeto: la suma de sus energías cinética y potencial molecular.
- Temperatura: se relaciona con la actividad cinética de las moléculas. Aunque no es cierto en todos los casos, un buen principio es definirla como la energía cinética promedio por molécula.
- Calor: calor se define como la transferencia de energía térmica debido a una diferencia en temperatura-
- Enfriamiento: es la disminución de la temperatura de un cuerpo o de un lugar.

Para este sistema en particular se implemento un controlador retroalimentado de tipo todo o nada, el cual por medio de retroalimentación de la variable a controlar calcula el error entre el valor medido y el valor de referencia, para aplicar una acción de control que ajuste el proceso.

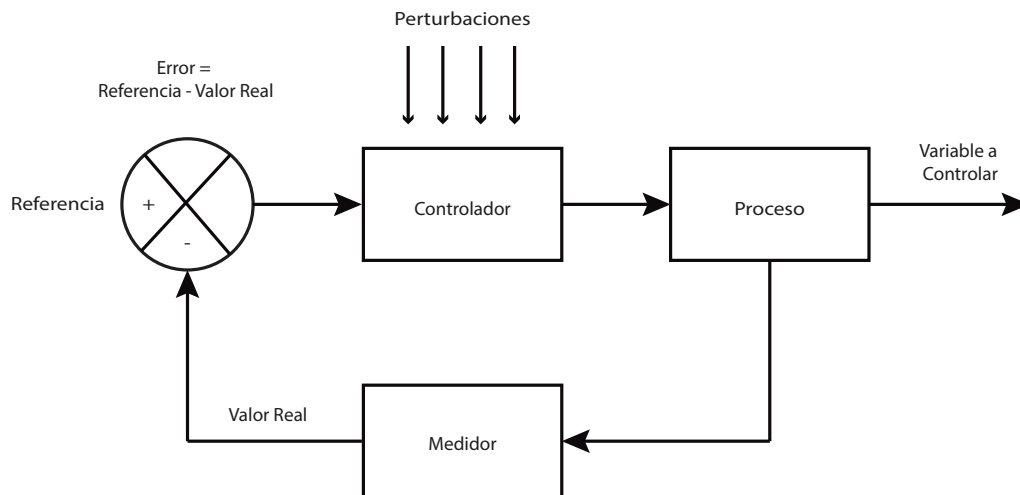


Figura 3.1: Sistema con controlador retroalimentado.

Las ventajas de este controlador son que pueden controlar sistemas inestables y compensar las perturbaciones, sus desventajas es que el uso de sensores induce el problema del ruido al hacer la medición.

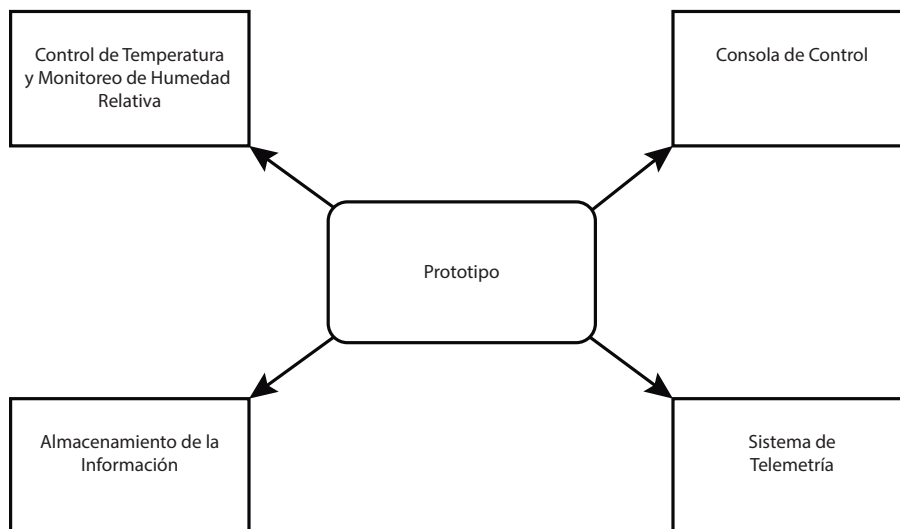
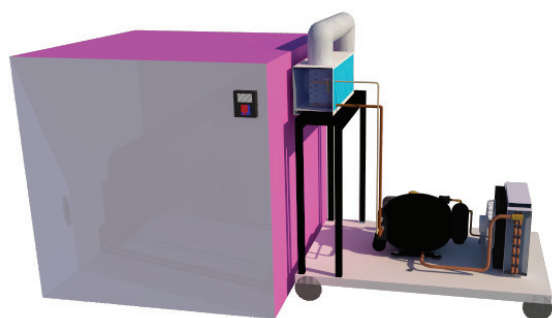


Figura 3.2: Esquema Básico del Prototipo propuesto.

Para hacer el desarrollo del prototipo, se diseñó y elaboró un medio confinado y aislado, sellado en el interior y aislado en el exterior, que cuenta con una compuerta en la parte de arriba removible. Se evitó obstruir el interior al diseñar el sistema frigorífico con sus elementos en el exterior, mediante el uso de una estructura fija para el soporte del equipo de control, asimismo se instaló la unidad de control en su exterior para que sea accesible al usuario.

A continuación en la figura 3.3 se muestra el diseño en el que se basó la construcción del prototipo:



Vista frontal.



Vista trasera.

Figura 3.3: Diseño del prototipo.

3.2. Medio Confinado

El Cuerpo Académico de Comunicaciones e Instrumentación Electrónica contaba con un medio sellado, mas no aislado, el cual constaba de una estructura metálica sobre la cual se montaron paneles de cartón corrugado para sellar el interior como se observa en la figura 3.4, sus dimensiones son 1m de altura, 1m de ancho y 1m de largo dando un volumen interior de 1m^3 , se tomó este medio y para darle las propiedades necesarias para el almacenamiento de productos agrícolas se procedió al montaje de pánels en el interior y exterior, los cuales otorgan las siguientes características al medio: Alta resistencia al impacto y a productos químicos, no permiten el desarrollo de hongos, estabilidad térmica, durabilidad, limpieza fácil y rápida.



Medio confinado.



Medio aislado.

Figura 3.4: Evolución de un medio confinado a un medio aislado.

3.2.1. Páneles Interiores

Se utilizaran paneles en el interior tipo Glasliner de uso industrial, los cuales se muestran en la figura 3.5, son un laminado plástico, plano y opaco elaborado a partir de resinas poliéster y reforzado con fibra de vidrio [18], estos fueron cortados a la medida para cubrir el interior del medio confinado.

Entre sus ventajas podemos encontrar:

- Alta resistencia al impacto y a productos químicos, abrasivos y bacterias.
- No crea hongos.
- Fácil de limpiar.
- Estabilidad térmica.
- Resistencia a la corrosión.
- Máxima protección sanitaria.

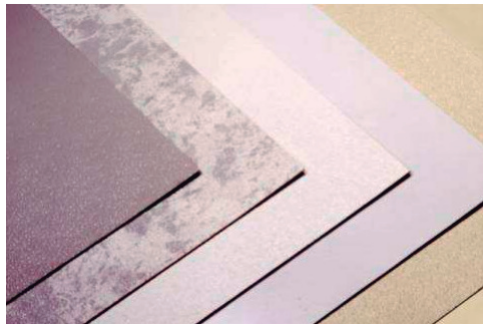


Figura 3.5: Paneles tipo Glasliner.

3.2.2. Páneles Exteriores

Se montaron paneles aislantes marca Foamular en el exterior, estos paneles cuentan con un aislamiento térmico de espuma rígida de poliestireno extruido en paneles manufacturados, como se puede ver en la figura 3.6. Tienen una superficie lisa y una

estructura de celdas cerradas con paredes que se interadhieren unas con otras sin dejar huecos. Dentro de sus aplicaciones se encuentran para aislar térmicamente desde viviendas, bodegas y naves industriales, centros comerciales, hospitales, laboratorios y frigoríficos.[19]

Se tomó en consideración una guía técnica del fabricante, la cual indica que estos paneles aislantes pueden ser utilizados en cualquier aplicación de almacenamiento refrigerado [20], incluyendo los pisos, paredes y techos. Además su resistencia al agua y vapor de agua le permite preservar su baja conductividad térmica aún en un ambiente muy húmedo. Se indica que ha sido sometido a pruebas de congelado y descongelado, y se ha observado que aún a 1000 ciclos se siguen manteniendo sus valores idénticos a cuando se instaló en el cuarto frío.

Entre sus ventajas podemos encontrar:

- Alta resistencia a la humedad y al vapor.
- No filtra el agua.
- No favorece la condensación.
- Es lavable y pintable.
- Garantiza su uniformidad térmica por 5 años.
- Resiste altas temperaturas.



Figura 3.6: Paneles tipo Foamular.

3.3. Sistema de Refrigeración

Para el sistema de refrigeración, debido a consideraciones económicas, se optó por reconstruir un sistema convencional de refrigeración, el cuál consta de una unidad condensadora y su respectivo evaporador. Se desmontaron las partes y equipos de un refrigerador de hogar y se reconstruyó adecuandolo al exterior del medio confinado como un sistema movable y de facil mantenimiento. A continuación en la figura 3.7 se muestran las etapas de construcción y montaje del sistema.



Figura 3.7: Montaje del sistema de refrigeración en la estructura.

3.3.1. Unidad Condensadora

Es la unidad que engloba el compresor, el ventilador y la batería de condensación. Se instaló una unidad condensadora con una potencia de 1/3 HP de equipo de refrigeración comercial, que incluye enfriamiento forzado mediante un ventilador para evitar sobrecalentamiento del compresor.

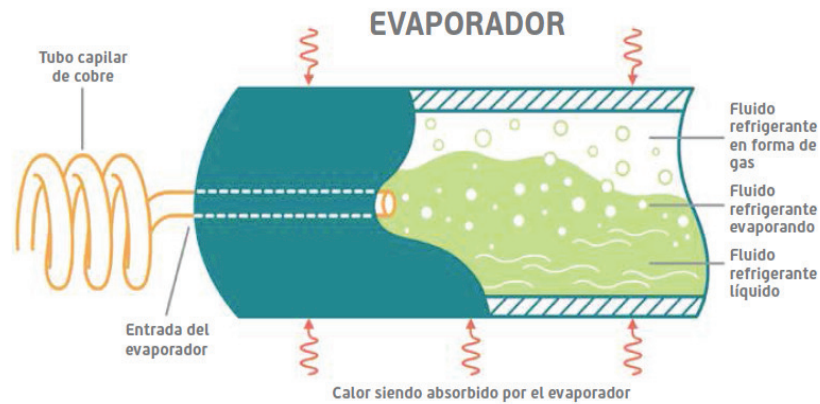


Figura 3.8: Unidad condensadora típica para uso comercial.

3.3.2. Evaporador

El evaporador es el elemento del sistema que se encarga del intercambio de calor entre la etapa de refrigeración y el interior del medio confinado. Su funcionamiento es descrito en la figura [3.9](#).

El evaporador fue tomado de un refrigerador de casa, junto con su abanico que aumenta el intercambio de calor dentro del medio confinado al forzar el intercambio de calor dentro del medio, reduciendo el tiempo de enfriamiento total de la unidad.



Funcionamiento.



Evaporador.

Figura 3.9: Equipo evaporador.

3.3.3. Gas Refrigerante R134A

El refrigerante R134a es un refrigerante amigable con el medio ambiente. Su ODP (potencial de agotamiento del ozono) es cero, por lo que no causa ningún daño a la capa de ozono.

Ventajas:

- El refrigerante R134a no contiene átomos de cloro, por lo tanto, no produce ningún daño a la capa de ozono.
- R134a es seguro de usar, ya que es ignífugo, no explosivo, no tóxico, no irritante y no corrosivo.

- En comparación con el gas refrigerante R12, el R134a tiene una mejor conductividad de calor. Esto reduce considerablemente el consumo de refrigerante. Además, ambos tienen una conductividad térmica similar, así que la modificación de su sistema de refrigeración es mucho más fácil.

3.4. Consola de Control

3.4.1. Microprocesador

Se cuenta con una consola de control implementada mediante una plataforma de hardware libre llamada Raspberry Pi, la cual fue considerada por su versatilidad para el desarrollo de aplicaciones computacionales y su gran comunidad de desarrollo en código abierto, en la cual se programó la rutina de control ON-OFF con histéresis que estará encargada del control de la temperatura y del control de humedad relativa mediante la activación de los relevadores de estado sólido, lo que dará flexibilidad y confiabilidad al sistema. Este sistema activará sus terminales GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General) para activar o desactivar a los actuadores en un circuito cerrado con ductos entrada/salida de aire en un ciclo cerrado sin fugas donde se cambian los parámetros de temperatura en el interior del medio confinado. Se anexa en el apéndice C el esquema completo del circuito del prototipo.

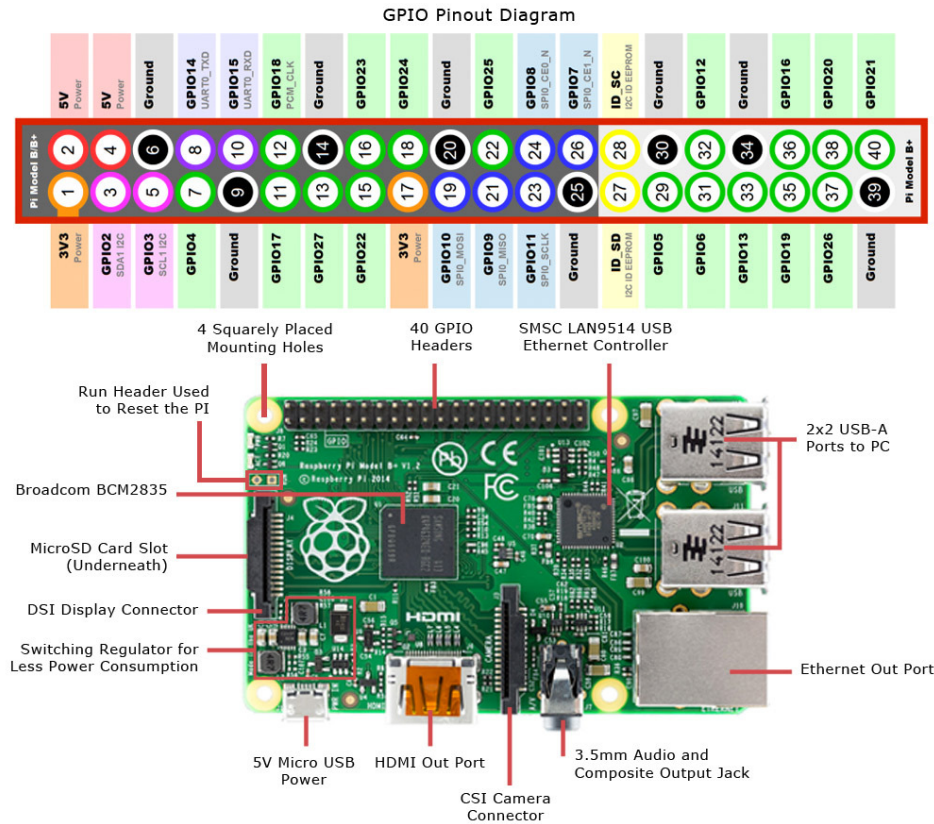


Figura 3.10: Placa de desarrollo Raspberry Pi 2.

3.4.2. Módulo de Temperatura y Humedad Relativa HTM2500LF

Realizando distintas pruebas con varios módulos, pasando por el sensor DS18B20 de tipo digital [21], este tipo de sensores están siendo muy utilizados en proyectos de redes de sensores inalámbricos [14], pero se observó que su respuesta contaba con mayor retardo en la adquisición de la temperatura instantánea, comparándolo con un termistor NTC que al estar siempre excitado tiene una respuesta casi instantánea.

Por lo tanto se instaló un módulo HTM2500LF, que se muestra en la figura 3.11, este incluye un sensor de temperatura, basado en un termistor NTC, e integra un sensor de humedad relativa [22]. El sensor cuenta con las siguientes especificaciones técnicas:

- Voltaje de operación: 3 a 10 V_{dc} .

- Humedad calibrada a $\pm 2\%$ a 55°C .
- Temperatura calibrada a $\pm 1\%$ por medio de termistor NTC $10k\Omega$
- Resistencia al agua.
- Desaturación instantanea después de largos periodos de saturación.
- Rápida respuesta.
- Alta resistencia a productos químicos.



Figura 3.11: Módulo HTM2500LF.

Es importante señalar que las pruebas se realizaron en el interior de la cámara frigorífica, contando con dos sistemas de referencia: un sensor digital DS18B20 y un sistema desarrollado por el CACIE que utiliza 3 termistores 104AT-11.

Sensor de Temperatura

El sensor de temperatura del módulo HTM2500LF corresponde a un termistor NTC, el cual es un dispositivo no lineal y su curva característica varia según sea el fabricante como se aprecia en la figura 3.12. La curva característica puede ser aproximada por medio de la ecuación 3.1 conocida como ecuación de Steinhart-Hart.

La ecuación Steinhart-Hart se ha convertido en el método generalmente aceptado para especificar la relación entre resistencia y temperatura en termistores. La ecuación Steinhart-Hart para la temperatura como función de la resistencia es la siguiente:

$$\frac{1}{T} = A + B \ln(R) + C[\ln(R)]^3 \quad (3.1)$$

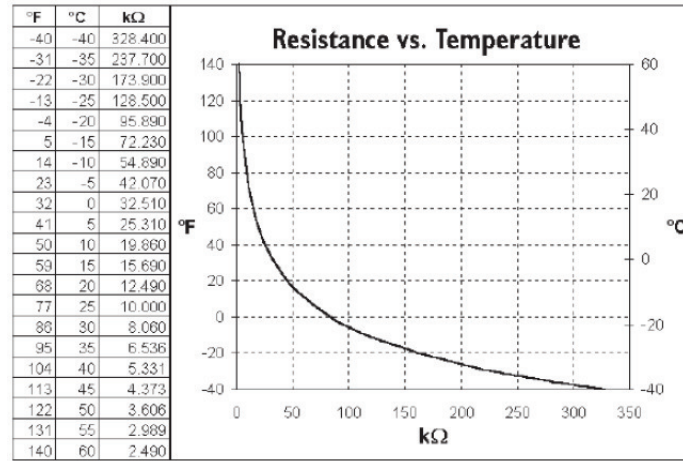


Figura 3.12: Curva característica del termistor que provee el fabricante.

en donde:

T : Temperatura en grados Kelvin

R : Resistencia del Termistor a la Temperatura Medida en Ω

A, B, C : Coeficientes de Steinhart-Hart, que varían dependiendo del termistor

Las constantes A , B , y C pueden ser obtenidas de dos maneras: la primera es utilizando los datos de calibración del fabricante la cual se muestra en la tabla 3.1 o calcularlas seleccionando tres puntos de la tabla o curva se mida con respecto a un sistema de referencia el termistor y resolviendo un sistema de ecuaciones simultáneas de tres incógnitas.

Finalmente se utilizaron los datos de calibración con respecto a un sistemas de referencia para obtener una medición calculada más precisa, se tiene que: $A = 8.54942e - 04$, $B = 2.57305e - 04$, $C = 1.65368e - 07$

Para resolver esta fórmula se utilizó el programa de control que viene incluido en el apéndice A, que se ejecuta en Python. Una vez verificada la ecuación se procedió a realizar la interfaz para el termistor y el convertidor analógico digital, la cual consiste en un divisor de voltaje con el termistor (R_{term}) y una resistencia de $10k\Omega$, se escogió

Temp (°C)	R (Ω)	Temp (°C)	R (Ω)
-40	195652	25	10000
-35	148171	30	8315
-30	113347	35	6948
-25	87559	40	5834
-20	68237	45	4917
-15	53650	50	4161
-10	42506	55	3535
-5	33892	60	3014
0	27219	65	2586
5	22021	70	2228
10	17926	75	1925
15	14674	80	1669
20	12081	85	1452

Tabla 3.1: Equivalencia de resistencia vs temperatura obtenida de la hoja de datos.

este valor porque de acuerdo a la literatura la resistencia de referencia debe ser igual a la que el termistor da a una temperatura de 25°C, tal como se muestra en la figura 3.13. Como voltaje de referencia se utilizan 3.3V los cuales son obtenidos de un regulador de voltaje de 5V a 3.3V modelo TC1262, que se utiliza para alimentar el sistema.

Como se observa en la figura 3.13, se tiene un divisor de voltaje por lo que:

$$V_{in} = \frac{R_{term}}{(R_1 + R_{term})} V_{ref}. \quad (3.2)$$

El ADC que se utiliza es de 12 Bits de resolución, modelo MCP3304 [23], se muestra en la figura 3.14 junto con la descripción de sus patillas de salida cuenta con 8 entradas analógicas, comunicación SPI y tecnología CMOS, opera en un alcance de voltaje de 2.7 a 5.5 Volts, su bajo consumo de corriente en espera es típico de 50 nA y una corriente activa de 450 uA a 5V.

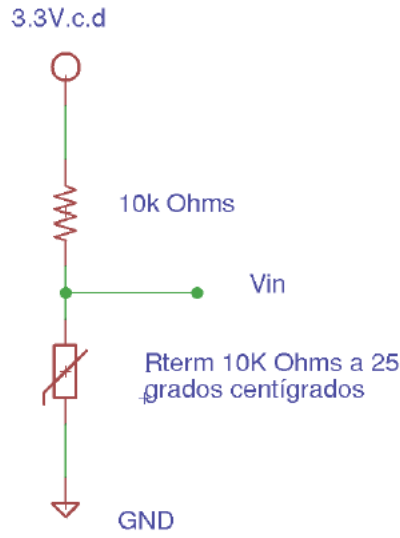


Figura 3.13: Interfaz para el termistor y el convertidor analógico-digital.

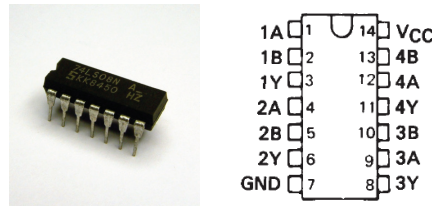


Figura 3.14: Convertidor analógico digital MCP3304

Este dispositivo utiliza un convertidor analógico-digital de aproximaciones sucesivas donde los valores entre 0.0 y V_{ref} son cuantificados como valores de 12 Bits desde 0 hasta $2^n - 1$, los cuales son enviados a la placa Raspberry Pi mediante comunicación serial SPI como la variable L_{ADC} , donde se utilizan las siguientes fórmulas para calcular la temperatura.

$$V_{ad} = \frac{L_{ADC} V_{ref}}{2^n - 1}, \quad (3.3)$$

desarrollando de las expresiones 3.2 y 3.3 R_{term} puede ser expresada como:

$$R_{term} = \frac{R_1 V_{ad}}{2^n - 1 - V_{ad}} \quad (3.4)$$

Conocida R_{term} la temperatura en grados Centígrados ($^{\circ}\text{C}$) puede calcularse usando la expresión 3.1.

Sensor de Humedad Relativa

El sensor de humedad relativa esta protegido por una estructura solida de polímero que es resistente a los químicos industriales, esta especificado para dar mediciones precisas en un rango de 10 % a 95 % de humedad en el aire¹.

El fabricante recomienda en su hoja de especificaciones alimentarlo a 5Vdc y nos provee de una tabla con la salida típica de la señal en base a distintos valores de humedad relativa, que se muestra en la tabla 3.2.

RH (%)	Vout (mV)	RH (%)	Vout (mV)
10	1235	55	2480
15	1390	60	2605
20	1540	65	2730
25	1685	70	2860
30	1825	75	2990
35	1960	80	3125
40	2090	85	3260
45	2220	90	3405
50	2350	95	3555

Tabla 3.2: Respuesta típica en mV del sensor alimentado a 5V.

El cálculo de la humedad relativa responde a una expresión de carácter lineal, la cual se muestra a continuación:

$$V_{out} = 26.65RH + 1006, \quad (3.5)$$

¹El sistema puede realizar mediciones fuera de este rango, pero el comportamiento es no lineal, por lo que el fabricante no recomienda su uso

despejando de la expresión anterior

$$RH = 0.0375V_{out} - 37.7. \quad (3.6)$$

en donde V_{out} es el Voltaje de salida del sensor en mV y RH es la Humedad Relativa en porcentaje (%).

Debido a que la entrada del ADC MCP3304 sólo soporta una entrada de voltaje máxima de 3.3V, se incluyó un divisor de voltaje para que el voltaje máximo de salida del sensor de fuera de 3.3V, el divisor se utiliza en la salida del sensor de humedad que se muestra en la figura 3.15.

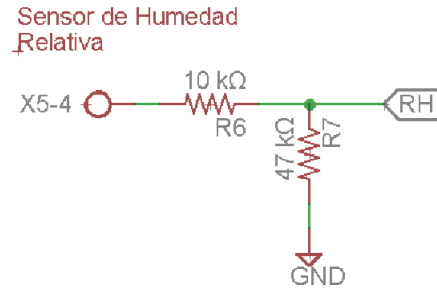


Figura 3.15: Interfaz para el sensor de humedad relativa y el convertidor analógico-digital.

La señal de salida del sensor se envía a un convertidor analógico-digital MCP3304 [23] que a su vez los enviará a la placa Raspberry Pi mediante comunicación serial SPI, dentro de la cual se implementó una fórmula para regresar el valor original antes del divisor de voltaje.

3.4.3. Circuito de Activación de Etapa de Potencia

Para energizar o desenergizar los actuadores del sistema de control se implementó un circuito en el que se actúan las terminales de un relé. Se optó por relés de estado sólido (SSR), los cuales son dispositivos interruptores electrónicos que conmutan el paso de

la electricidad cuando una pequeña corriente es aplicada en sus terminales de control. Básicamente consisten en un sensor que responde a una entrada apropiada (señal de control), un interruptor electrónico de estado sólido que conmuta el circuito de carga, y un mecanismo de acoplamiento a partir de la señal de control que activa este interruptor sin partes mecánicas.[24] Pueden estar diseñados para conmutar corriente alterna o directa. Hacen la misma función que los relés electromecánicos, pero sin partes móviles. Debido a que utilizan semiconductores de potencia como tiristores y transistores pueden conmutar corrientes de hasta 100 amperes.



Figura 3.16: Relé de Estado Sólido.

El modo más sencillo para activar estos relés es mediante un transistor NPN, el cual es un dispositivo que regula el flujo de corriente o de tensión actuando como un interruptor o amplificador para señales electrónica [25]. Una aplicación común de estos dispositivos es como interruptores, conectado como se ve en la figura 3.17, cierra el circuito poniendo tierra el terminal de la etapa de control del relevador mientras que el otro terminal se encuentra conectado a positivo.

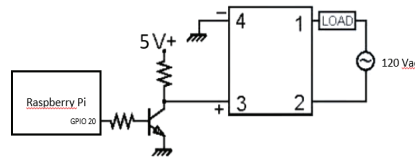


Figura 3.17: Circuito de activación de relevador.

Algunas ventajas sobre al utilizar relés de estado sólido sobre los relés electromecánicos son las siguientes:

- Menor tamaño, permitiendo elementos más compactos y automatizables.
- Menor tensión de trabajo, se activan desde 3.0V.
- Funcionamiento totalmente silencioso.
- Los SSR son más rápidos que los relés electromecánicos; su tiempo de conmutación menor a 10ms.
- Vida útil más larga, incluso si se activa muchas veces, ya que no hay partes mecánicas que se desgasten o contactos que se deterioren a altos amperajes.
- La resistencia de salida se mantiene constante independientemente del uso.
- Sin chispas, no se producen arcos eléctricos, lo que permite ser usados en ambientes explosivos donde es crítico que no se produzcan chispas en la conexión.
- No produce ondas electromagnéticas que puedan producir interferencias en otros equipos.

3.4.4. Interfaz Gráfica en Sitio

Se cuenta con una interfaz gráfica para el usuario en la consola de control en sitio como se aprecia en la figura 3.18, la cual se integra de los siguientes elementos:

Una pantalla LCD gráfica modelo ST7565 [26], que es utilizada para desplegar las últimas mediciones adquiridas de temperatura y humedad por el sistema de medición en tiempo real, además del valor de histéresis de la temperatura.

La cual tiene las siguientes características:

- Pantalla de matriz con 128x64 pixeles
- Tensión de trabajo de 3.3V.
- Interfaz de comunicación SPI.
- Cuenta con luz de fondo programable en colores RGB.

Además se integró un teclado en la consola de control para ajustar los parámetros de configuración del sistema manualmente, este procedimiento se describe en el capítulo 4.

3.5. Circuito de Protección del Compresor

Se introdujo una etapa previa a la salida para el relevador que activa el compresor, la cual protege al sistema en caso de que la temperatura no se encuentre en rangos de operación segura. Este rango de operación segura se definió haciendo pruebas largas donde se observó que la temperatura mínima dentro del sistema no llegaba a menos de 2 C y comenzaba a forzarse el equipo dando lugar a un sobrecalentamiento del compresor, en cambio en caso que el sensor dejara de funcionar o fuera desconectado por error o casualidad, este no funcionaría para evitar que encienda el compresor. Los valores de voltaje de referencia fueron calculados utilizando los datos del fabricante de sensor. En la figura 3.19 se visualiza el esquemático del circuito.

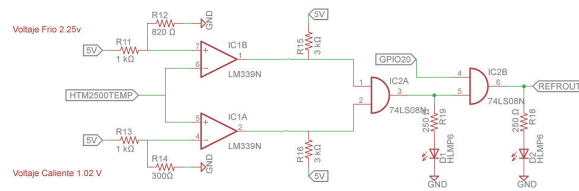


Figura 3.19: Esquema del circuito de protección del compresor.

Para esta etapa se dio solución implementando dos dispositivos electrónicos de fácil adquisición en el mercado.

- Comparador LM-339N: La serie 339 de National Semiconductor es otra familia de comparadores muy utilizada cuando el coste es crítico ya que un mismo encapsulado contiene varios comparadores cada uno de ellos con dos entradas y una salida en colector abierto. La tensión de alimentación es común y todos los comparadores disipan potencia aunque solo se utilice uno de ellos [27].
- Compuerta AND: La compuerta AND es una puerta lógica digital que se comporta de la siguiente manera. Ésta entregará una salida ALTA (1), dependiendo de los valores de las entradas, siendo este caso, al recibir solo valores altos en ambas entradas. Si alguna de estas entradas no son ALTAS, entonces se mostrará un valor de salida BAJA (0) [28]. Su configuración se puede observar en la figura 3.20.

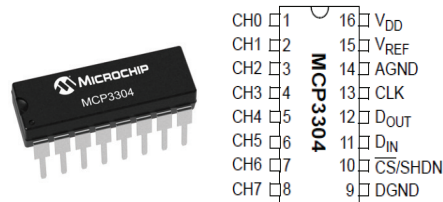


Figura 3.20: Compuerta lógica AND

3.6. Sistema de Respaldo Eléctrico

Los sistemas de respaldo son un conjunto de elementos o equipos que sirven de apoyo a otros, para realizar una actividad o función. Particularmente estos sistemas, permiten mantener un suministro continuo de energía, aun cuando la fuente principal deje de funcionar.

En los tiempos actuales son utilizados sistemas de energía ininterrumpida (UPS) en equipos eléctricos sensibles como pueden ser computadoras, estaciones de trabajo, sistemas de telecomunicaciones, centros de información, sistemas de control de procesos, etc.[29]

El UPS protege los equipos de una baja calidad de la energía de la compañía suministradora, o de una completa pérdida de energía.

Llendo más allá, un UPS puede incluir funciones añadidas como el mejoramiento de la calidad de la energía a una carga crítica.

Existen en el mercado tres tipos de sistemas de respaldo energético:

- Pasivos en Standby
- En Línea Interactivos
- De Doble Conversión

El sistema de respaldo del prototipo consta de una batería de iones de litio acoplada al sistema de telemetría que dará soporte de energía en caso de una falla en el suministro.

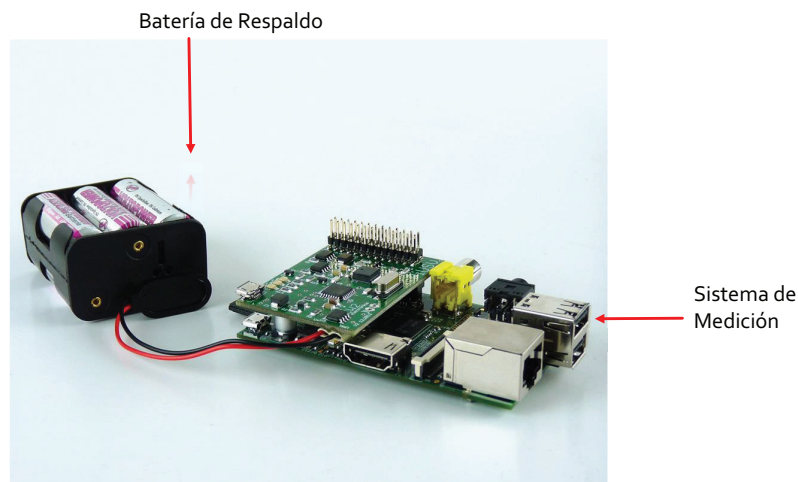


Figura 3.21: Esquema del Sistema de Respaldo.

3.7. Sistema de Almacenamiento de Datos y Telemetría

Se elaboró una rutina de almacenamiento en una base de datos en SQLite y una interfaz de monitoreo remoto por telemetría escrita en lenguaje de programación Python.

Además como parte de los trabajos realizados en el curso de Programación de Dispositivos Móviles, se desarrolló una aplicación móvil funcional en la plataforma de código libre Android, en la cual es posible consultar las variables sensadas actuales de la cámara frigorífica y además se permite graficar un historial entre un intervalo de tiempo para su análisis.

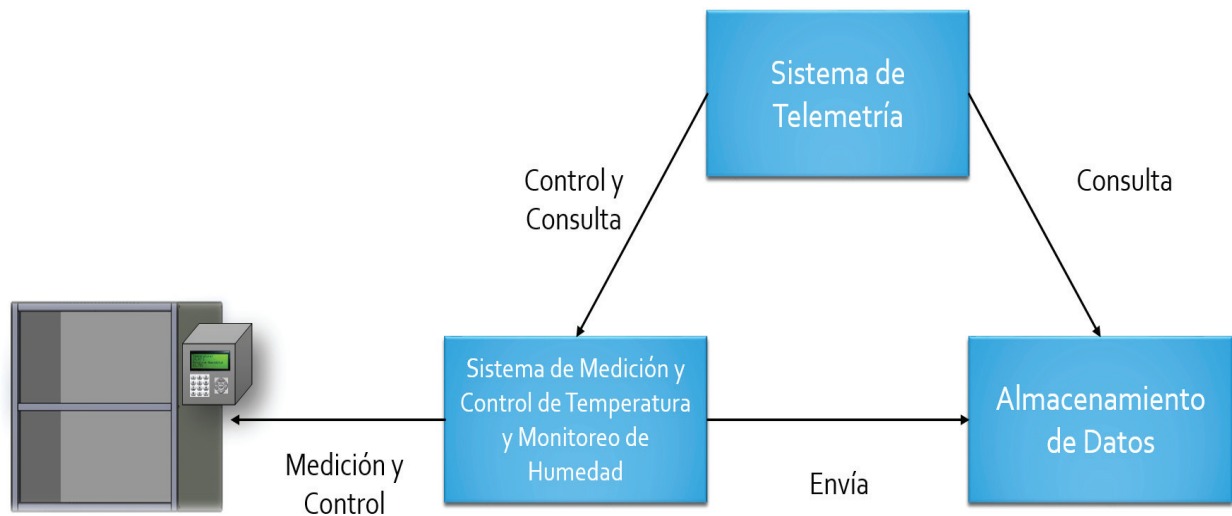


Figura 3.22: Esquema del funcionamiento del Sistema de Telemetría.

3.7.1. Aplicación Web

Se desarrolló la aplicación web basandose en el modelo MVC (Modelo, Vista, Controlador) y utilizando las siguientes herramientas:

Lenguaje de programación Python. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License,¹ que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1.

Flask. Que es un framework minimalista escrito en Python y basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2. Tiene la licencia BSD. Para el manejo de la aplicación web utilizando la Raspberry Pi.

SQLite. Es un sistema de gestión de bases de datos relacional. En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño.

SQLAlchemy. Que es una herramienta SQL de código abierto y mapeador objeto-relacional para el lenguaje de programación Python, cuenta con licencia MIT. Se utilizó

para el manejo de la base de datos mediante la aplicación web.

Los ficheros estáticos fueron programados mediante:

HTML. Es un lenguaje utilizado en la informática, cuyo fin es el desarrollo de las páginas web, indicando cuales son los elementos que la componen.

Javascript. Es un lenguaje que utiliza principalmente para crear efectos atractivos y dinámicos en las páginas web.

CSS. Es un lenguaje que sirve para organizar la presentación y aspecto de una página web.

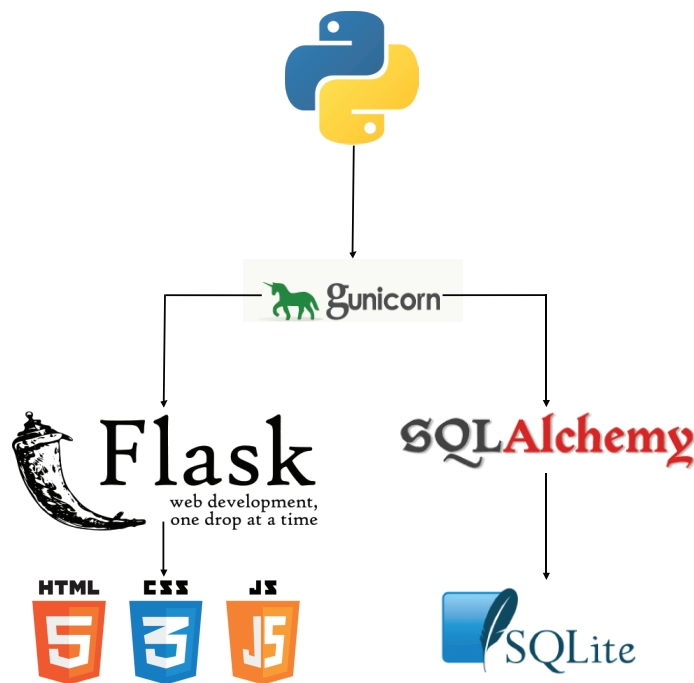


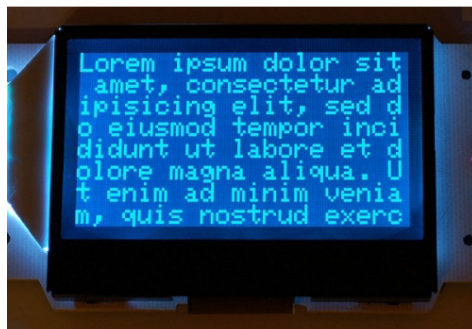
Figura 3.23: Esquema de la Aplicación Web.

El despliegue de una aplicación es una parte compleja puesto que existen muchas formas de hacer el despliegue, se hizo un despliegue usando Nginx, Supervisor y Gunicorn.

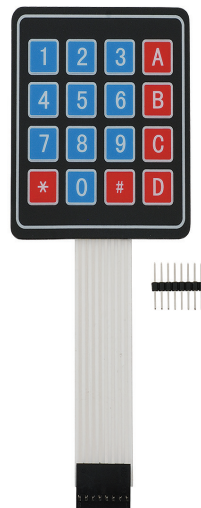
Supervisor. Es un gestor de procesos para linux. Nos permitió crear un proceso en segundo plano de nuestro servidor gunicorn.

Gunicorn. Es un servidor WSGI HTTP para Python (pre-fork de unicorn de ruby). Consume poco y es bastante rápido para aplicaciones basadas en interfaz web.

Nginx. Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Nos ayudará a servir ficheros estáticos.



Pantalla gráfica LCD modelo ST7565.



Teclado de Matriz de Membrana.

Figura 3.18: Elementos de la Interfaz Gráfica en sitio.

Capítulo 4

Pruebas y Resultados

En éste capítulo se presentan las pruebas y resultados obtenidos con el prototipo de medio confinado que emula una cámara frigorífica.

El capítulo se compone de las siguiente secciones; La siguiente se muestra el funcionamiento y características del sensor de temperatura y humedad HTM2500LF, el cual es la base del sistema de medición. Después se detallan las mediciones y funcionamiento de un primer prototipo del sistema que permitieron el diseño y construcción del prototipo final. Se describe el manejo y funcionamiento del sistema construido actualmente. Finalmente se muestra una serie de los datos almacenados de las variables ambientales del medio y se realiza una evaluación del desempeño del sistema.

4.1. Realización de las Primeras Mediciones

Primeramente se realizó un código en lenguaje C utilizando un microcontrolador Arduino Uno.

Básicamente consistía en la lectura de las mediciones de temperatura y humedad, para desplegarlas en una pantalla LCD, como se muestra en la figura 4.1 y la activación de un relevador para el control del compresor en base a una referencia de temperatura constante, sin contar con almacenamiento de datos.

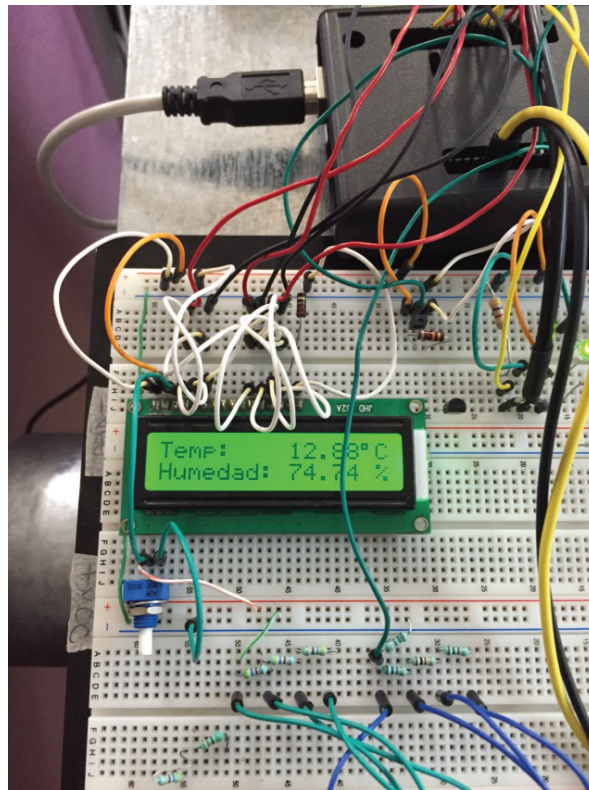


Figura 4.1: Primer prototipo en placa de pruebas.

El prototipo inicial del sistema frigorífico consistió en reutilizar las partes de un refrigerador convencional y adaptarlo a las necesidades del medio. Se contrataron los servicios de un técnico en refrigeración para soldar las tuberías de alta y baja presión y el llenado del compresor con el gas refrigerante.



Figura 4.2: Primer prototipo frigorífico.

Una vez probado este sistema se buscó otra plataforma de desarrollo que permitiera integrar un sistema de almacenamiento de datos con control remoto de la consola, por lo que se escogió una placa de desarrollo Raspberry Pi ya que es posible integrar un servidor web, una base de datos y controlar pines de entrada-salida de uso general (GPIO). Todo esto en un diseño modular y compacto.

4.2. Prototipo Final y Consola de Control

En el prototipo inicial con el sistema Arduino Uno, el equipo de refrigeración era muy pequeño de alrededor de 1/8 HP de potencia por ser de uso doméstico. Sin embargo se presentaron fallas en el condensador y sobrecalentamiento del compresor al dejar

funcionando el sistema por largos periodos de tiempo. Por lo que fue reemplazado por una unidad de 1/3 HP de potencia de un equipo frigorífico comercial, la cual se muestra en la figura 4.3, por otra parte se mantuvo el mismo evaporador existente, sólo se cambio la tubería y la unidad condensadora, que está integrada por el compresor, un abanico y el condensador.

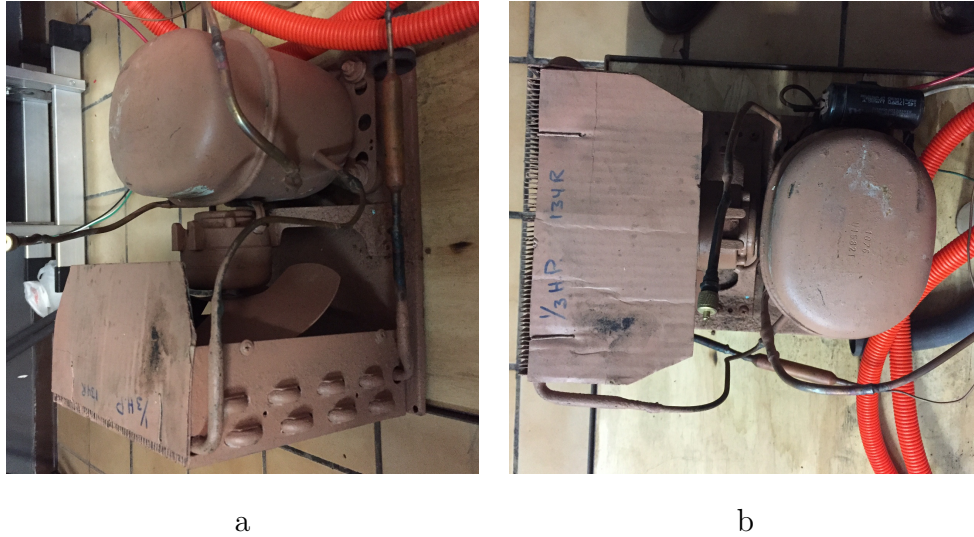


Figura 4.3: Unidad Condensadora de 1/3 HP.

Este nuevo equipo al ser de mayor potencia nos permite llegar a temperaturas más bajas en menos tiempo, evitando el sobrecalentamiento que sufrió el equipo anterior y por lo tanto posibles fallas por exceso de trabajo.

Teniendo la necesidad de integrar el servidor web con su aplicación de telemetría, se configuró la Raspberry Pi con un sistema operativo basado en una distribución de Linux llamada Raspbian Jessie Lite. Además, se selecciono el lenguaje Python para diseñar y elaborar el nuevo programa de control,¹ para tal efecto se realizó el diagrama de flujo que se muestra en la figura 4.4 con el que funciona la consola en sitio.

¹El código fuente se anexa en el apéndice A.

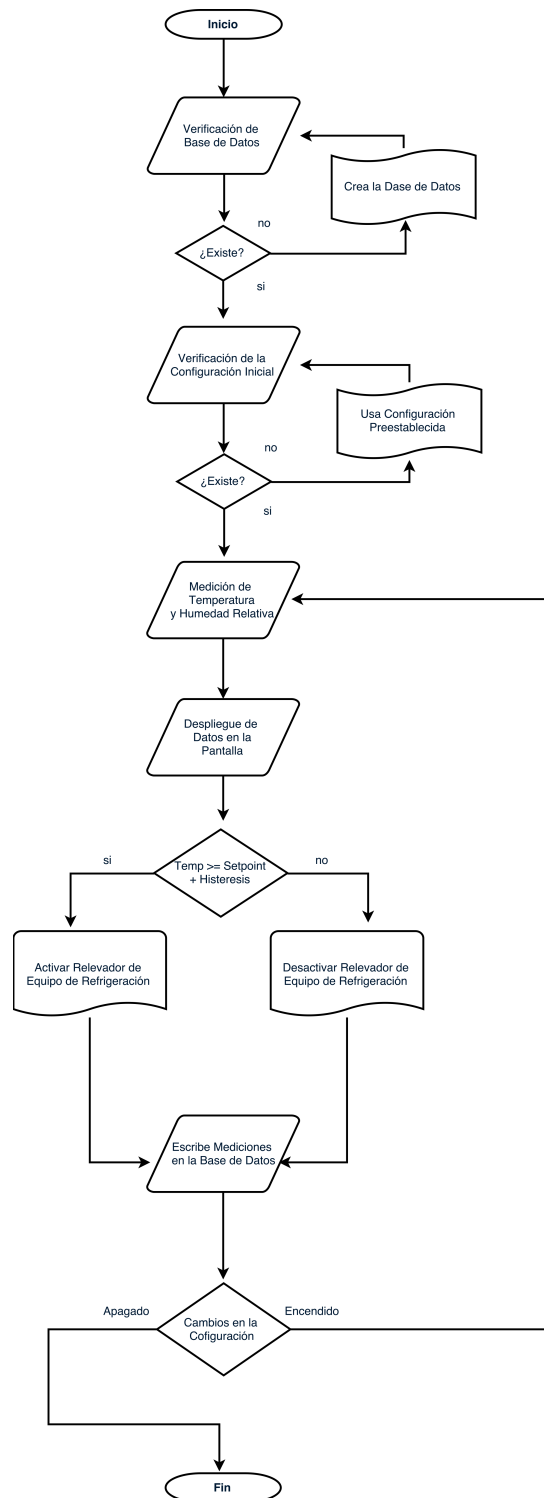


Figura 4.4: Diagrama de flujo aplicación de control.

Dentro de la consola de control se introdujo la placa desarrollada del circuito electrónico de control que se muestra en la figura 4.5, este circuito se encarga de recibir y enviar las señales que realiza el programa ejecutado en la Raspberry Pi.



Figura 4.5: Circuito impreso de la placa del sistema de control.

4.2.1. Configuración de la Consola de Control

La consola de control despliega los datos de medición y puntos de referencia en la pantalla gráfica integrada, para configurar localmente el sistema se hace uso de un teclado, el panel de control se muestra en la figura 4.6. El teclado funciona a base de interrupciones, mientras el programa principal funciona en paralelo por lo que no hay conflicto en la lógica del sistema.

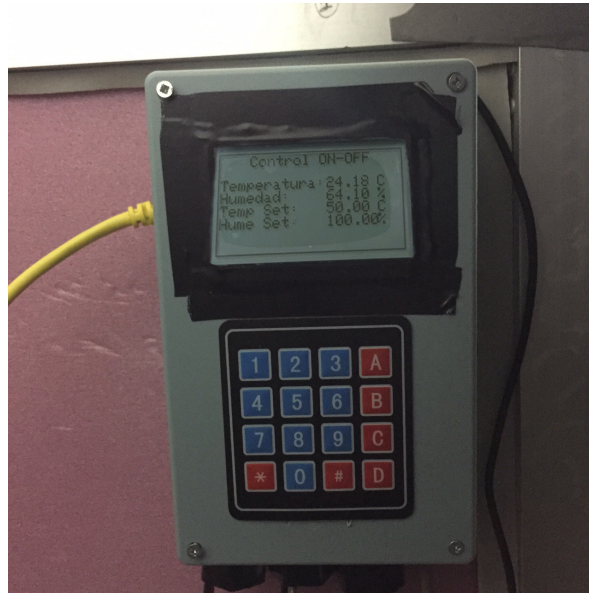


Figura 4.6: Panel de control del prototipo

A continuación se explica el procedimiento de configuración:

Primeramente tecleas un nuevo punto de referencia y dependiendo del parámetro a configurar presionas la tecla deseada para guardarlo en una variable temporal:

- A- Cambias el valor de referencia de Temperatura
- B- Cambias el valor de referencia de Humedad
- C- Cambias el valor de Histeresis de Temperatura
- D- Cambias el valor de Histeresis de Humedad
- *- Punto Decimal
- #- Backspace

Una vez que el usuario haya tecleado sus nuevos parámetros de configuración se deberá teclear el comando: ".. + A" para validar la nueva configuración. En caso de que el usuario haya cometido un error, o simplemente no quiera validar la nueva configuración se deberá teclear el comando: ".. + B" y las variables temporales serán eliminadas, manteniéndose la configuración anterior.

4.2.2. Configuración del Sistema de Telemetría

El sistema de telemetría consta de una aplicación web programada bajo el modelo MVC (modelo, vista, controlador) escrita en lenguaje Python utilizando las librerías Flask y SQL Alchemy para el manejo de la base de datos del sistema, la aplicación cuenta con cinco vistas principales que son servidas mediante el servidor Gunicorn bajo el puerto 8000 mediante el proxy inverso Nginx.

Se tiene el modelo de la aplicación que lo integra la base de datos Sqlite. Las vistas son páginas web que están escritas en HTML5 y CSS. El controlador esta escrito en lenguaje de código abierto Python².

La aplicación web muestra el estado en que se encuentra el equipo en todas las vistas, ya sea encendido o apagado, así mismo permite monitorizar los datos actuales del sistema, graficar o exportar los datos en un archivo de hoja de cálculo, y por último acceder a su configuración y realizar cambios iniciando sesión con una cuenta de administrador. Estas vistas se explican con detalle a continuación.

La página de inicio, que se observa en la figura 4.7 corresponde a la pantalla de inicio donde se describe una introducción del proyecto y se muestran imágenes del renderizado.

²El código fuente de esta aplicación se anexa en el apéndice B.

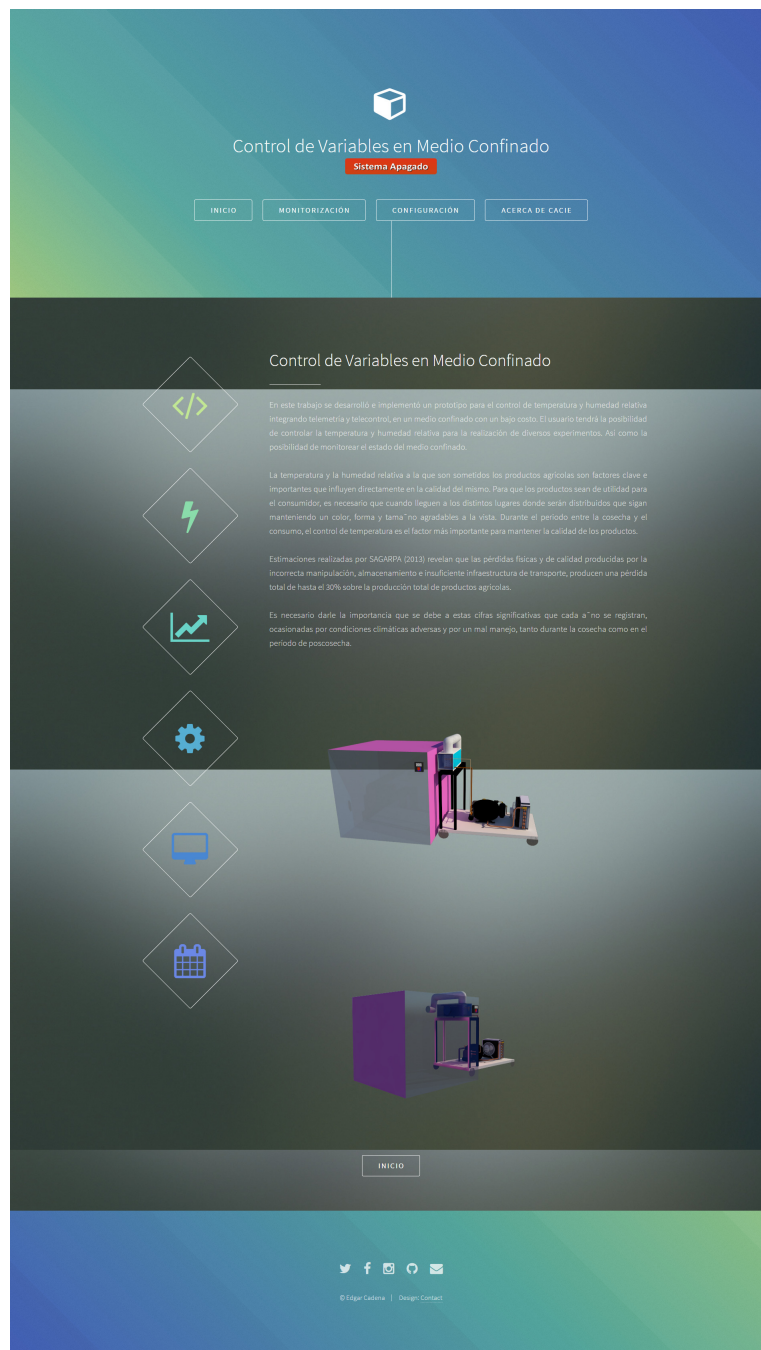


Figura 4.7: Página de Inicio.

La página de datos actuales nos permite visualizar los datos actuales del sistema en tiempo real, como se observa en la figura 4.8.

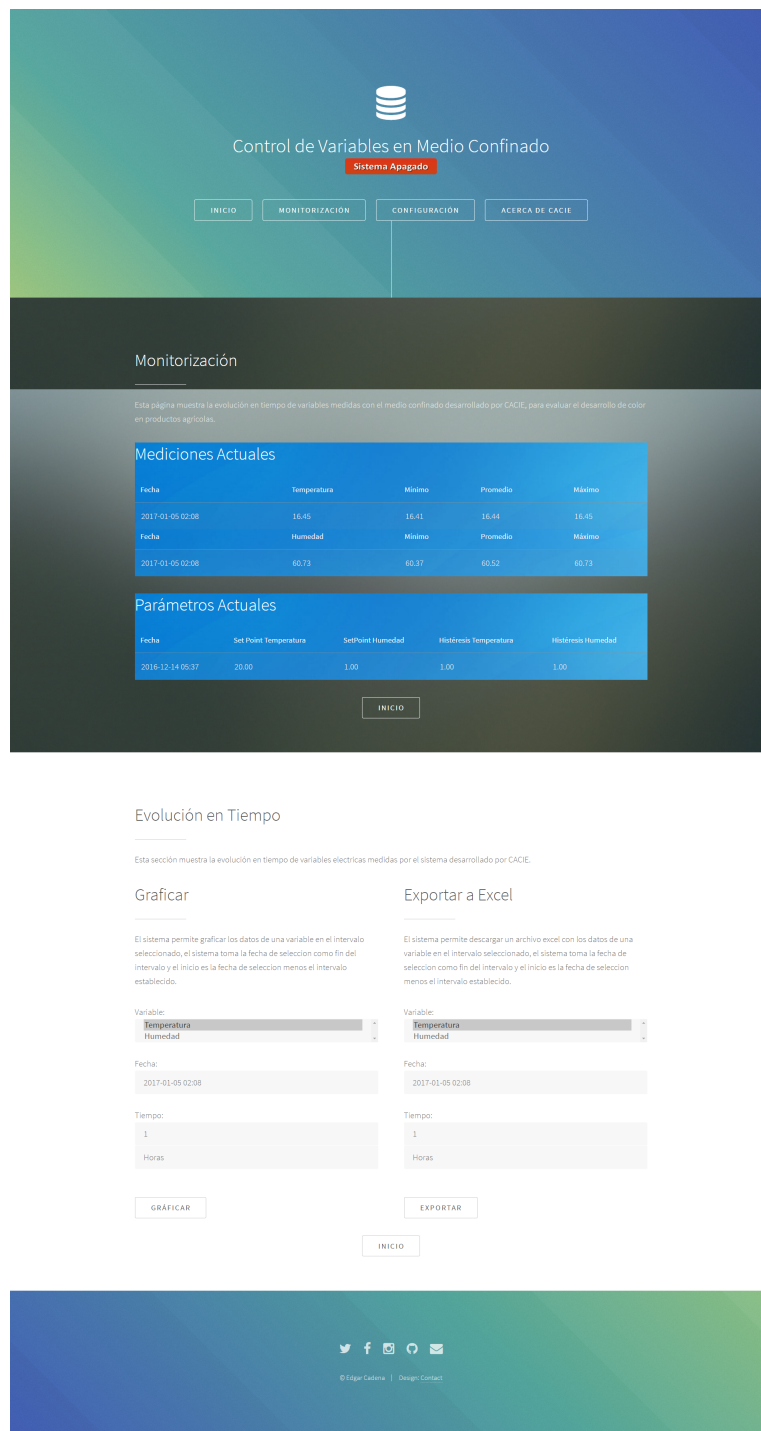


Figura 4.8: Página de Monitorización.

En la página de configuración que se muestra en las figuras 4.9, 4.10 y 4.11, se

permite cambiar la configuraciones del sistema al contar con un usuario con privilegios de administrador del sistema. Además se permite encender o apagar el equipo dde manera remota en el menú oculto. Esta interacción se realiza directamente en la base de datos de configuración del sistema por lo que los nuevos parámetros son interpretados por el sistema en cuanto detecte algún cambio.

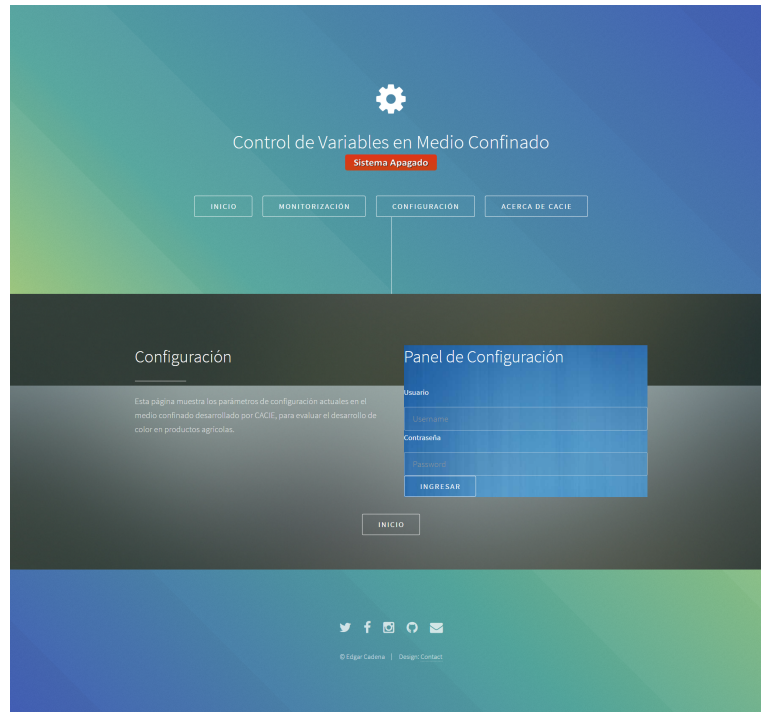


Figura 4.9: Página de Configuración.

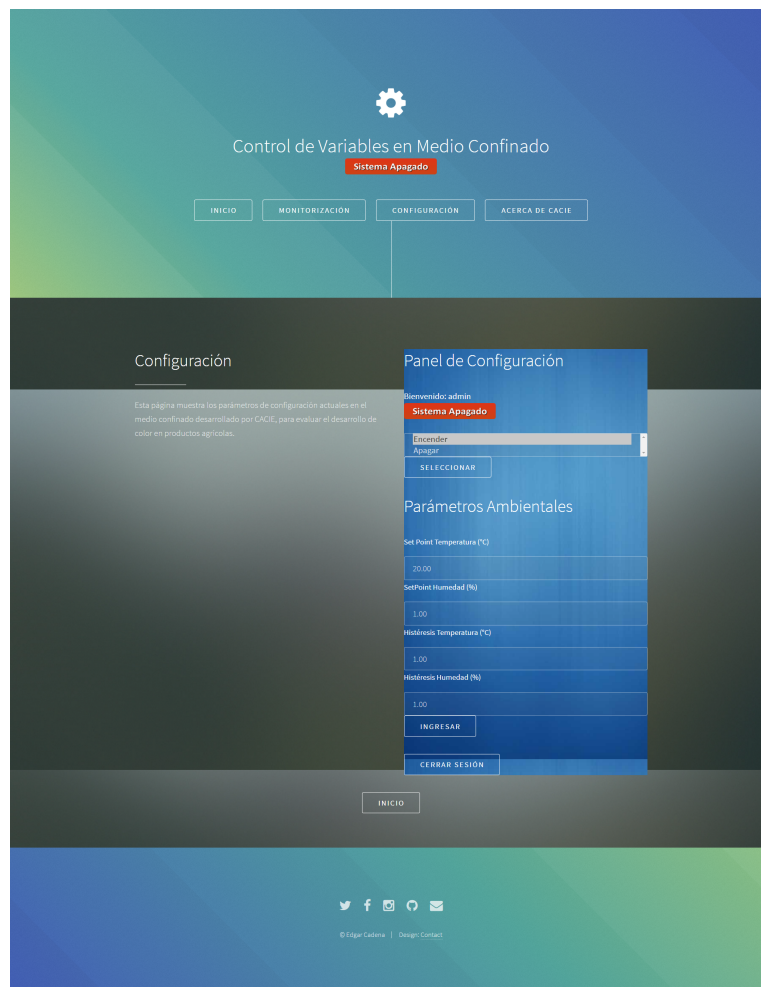


Figura 4.10: Página de Configuración.

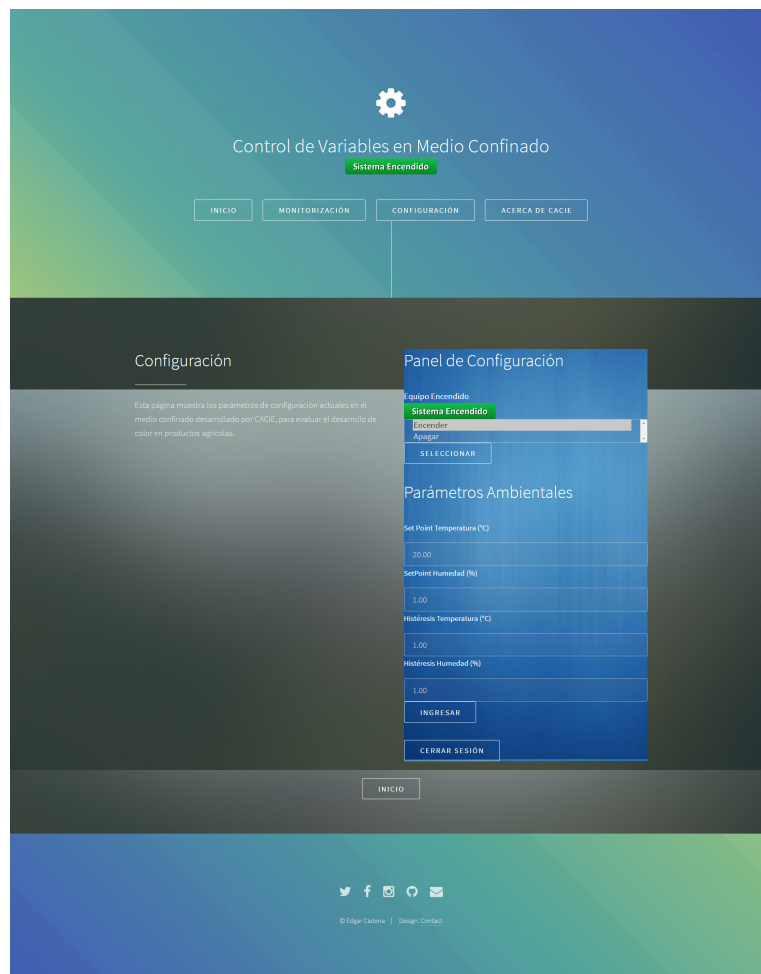


Figura 4.11: Página de Configuración.

Por último se incluye una página donde se habla una breve introducción acerca del cuerpo académico al que pertenece este trabajo de tesis y las actividades que se realizan. Como se muestra en la figura 4.12.



Figura 4.12: Página de Acerca del Cuerpo Académico CACIE.

4.3. Pruebas en Medio Confinado

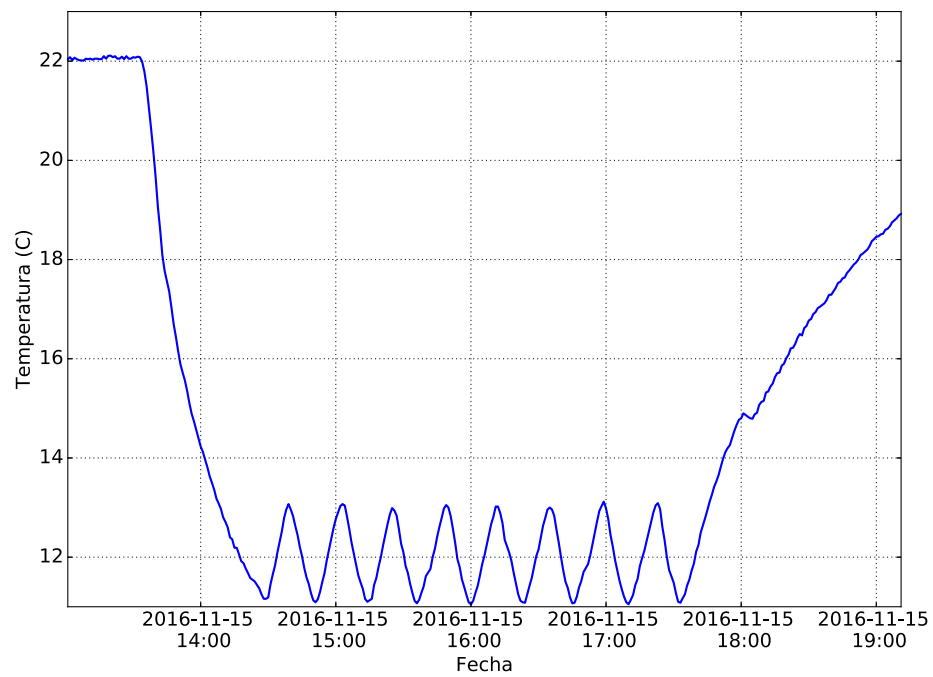
Se validó el sistema completo realizando pruebas con distintos parámetros de control, conectando un sistema externo de medición de consumo de energía eléctrica para caracterizar el consumo del sistema mientras se encontraba en funcionamiento.

4.3.1. Mediciones de Temperatura y Humedad

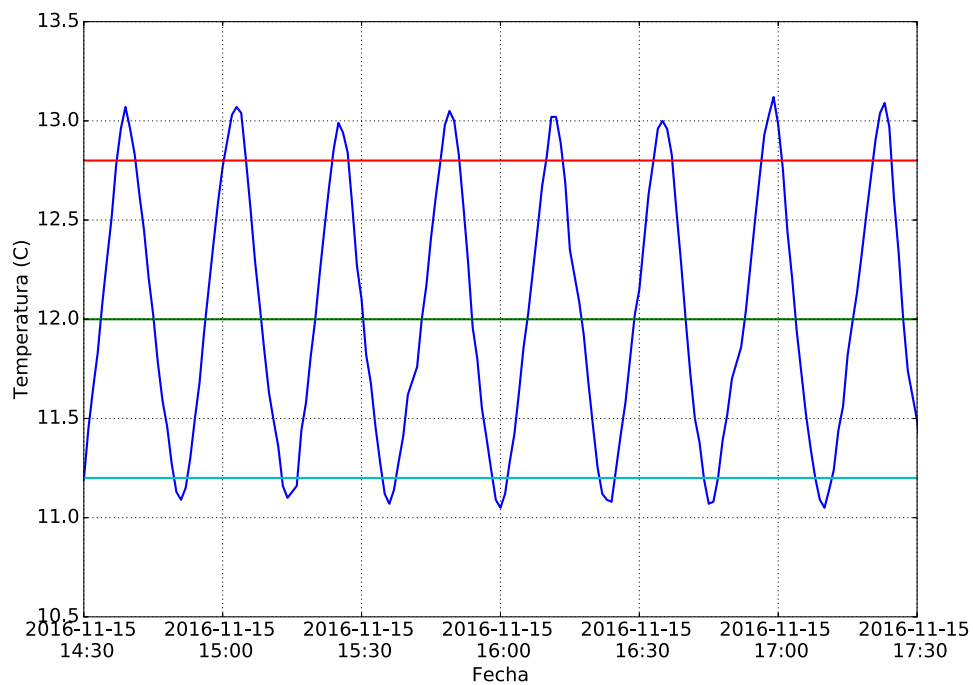
Para dar validez del sistema de control se hicieron varias pruebas dentro de las cuales se observó que el sistema responde como se planeó manteniendo la temperatura dentro del rango de histéresis predefinido.

Se realizaron pruebas con una histéresis a $0.8\text{ }^{\circ}\text{C}$ y un punto de referencia de tem-

peratura de 12 °C como se ve en la figura 4.13 el sistema de control mantuvo el interior del medio confinado a la temperatura de referencia dentro del rango de la histéresis introducida con la mínima variación, en un periodo prolongado de tiempo. Demostrando que el sistema muestra confiabilidad en su actuar.



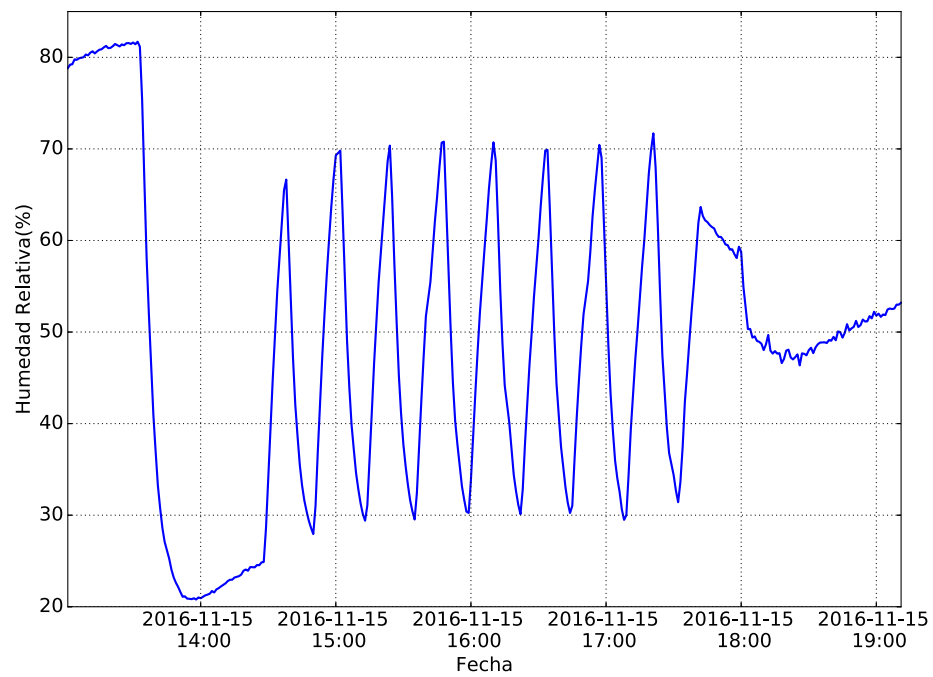
a



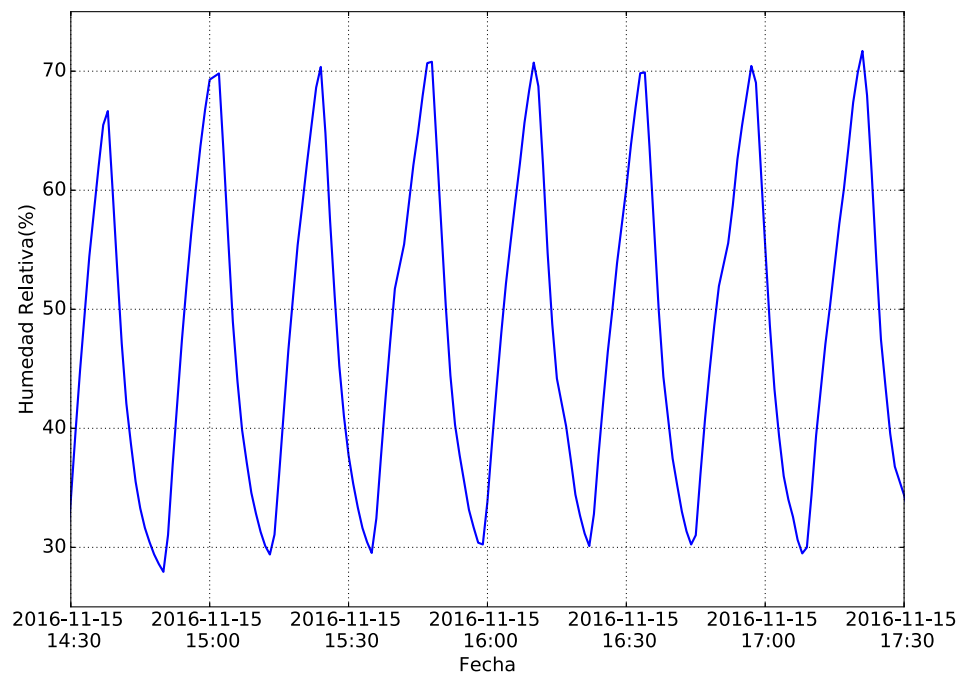
b

Figura 4.13: Mediciones de Temperatura.

Se observa en la figura 4.14 que la humedad relativa del aire desciende al mismo tiempo que la temperatura del interior, esto debido a la condensación que sufren las partículas de vapor de agua suspendidas en la atmósfera interior del medio confinado al momento de que la temperatura desciende.



a



b

Figura 4.14: Mediciones de Temperatura.

4.3.2. Mediciones de Parámetros Eléctricos del Sistema

Se realizaron mediciones de la corriente, voltaje, potencia instantánea del sistema frigorífico, utilizando un sistema digital de medición externo desarrollado por CACIE, el cual mediante un sensor de voltaje y un sensor de corriente calcula mediante aproximaciones los parámetros eléctricos mencionados.

El sistema de refrigeración consume alrededor de 7A como se ve en la figura 4.15 notándose un pequeño incremento durante los arranques pero estabilizándose rápidamente el consumo, para después descender cercano al cero ya que el sistema de control en sitio es de bajo consumo.

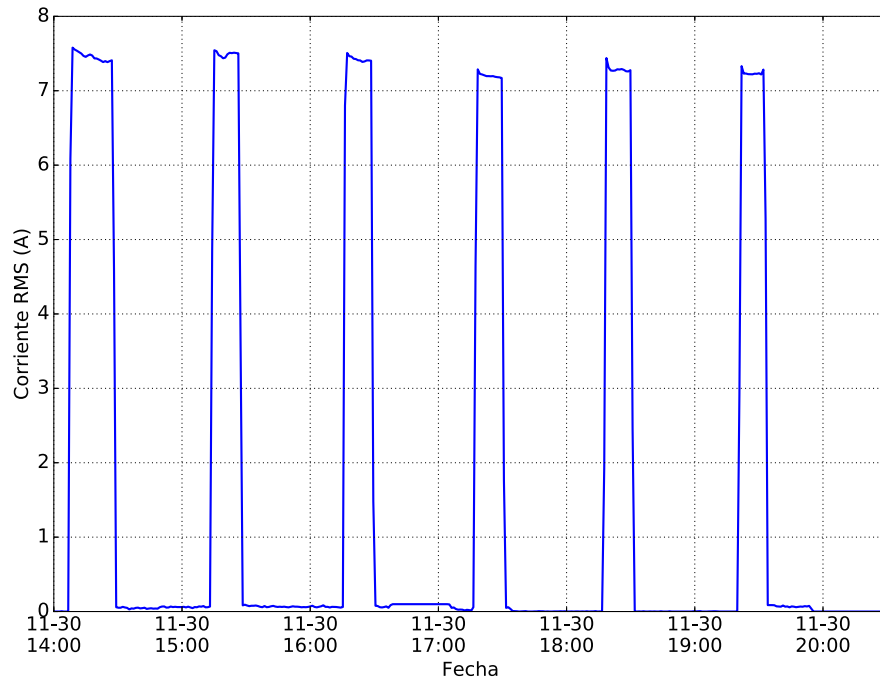


Figura 4.15: Consumo de corriente eléctrica del sistema.

Se observa en las figuras 4.15 y 4.16 que se tiene un comportamiento eléctrico normal para un equipo de refrigeración por compresor, el cual es el esperado al ser un control on-off ya que el compresor consume una potencia constante mientras esta encendido y

nada de potencia al estar apagado, se observó que son intervalos cortos en lo que enfria.

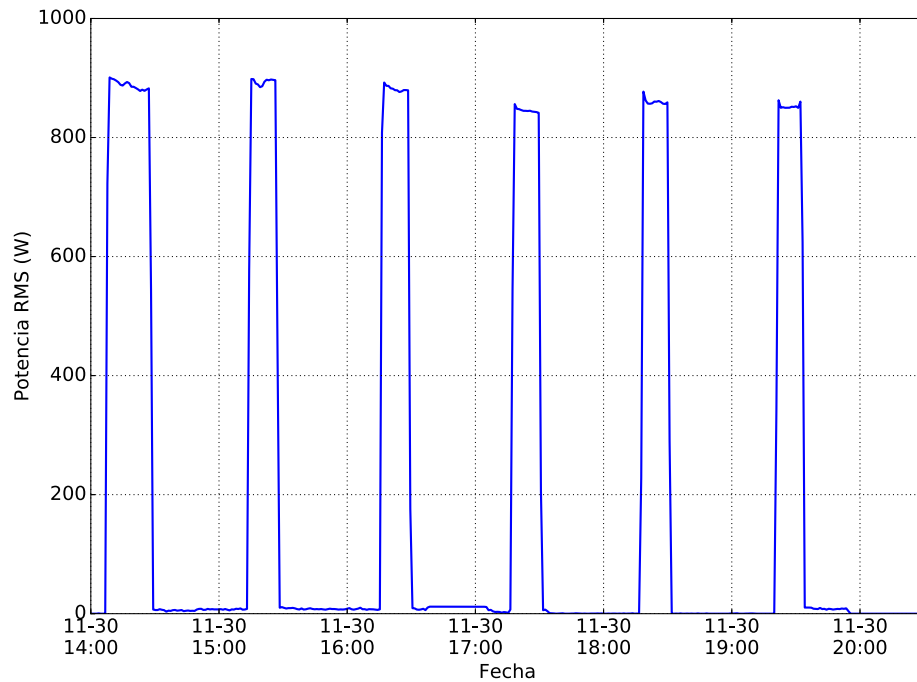


Figura 4.16: Consumo de potencia eléctrica del sistema.

Queda demostrado que el sistema no introduce ruido eléctrico considerable a la línea de corriente alterna ya que la variación del voltaje es mínima como queda documentado en la figura 4.17 a continuación.

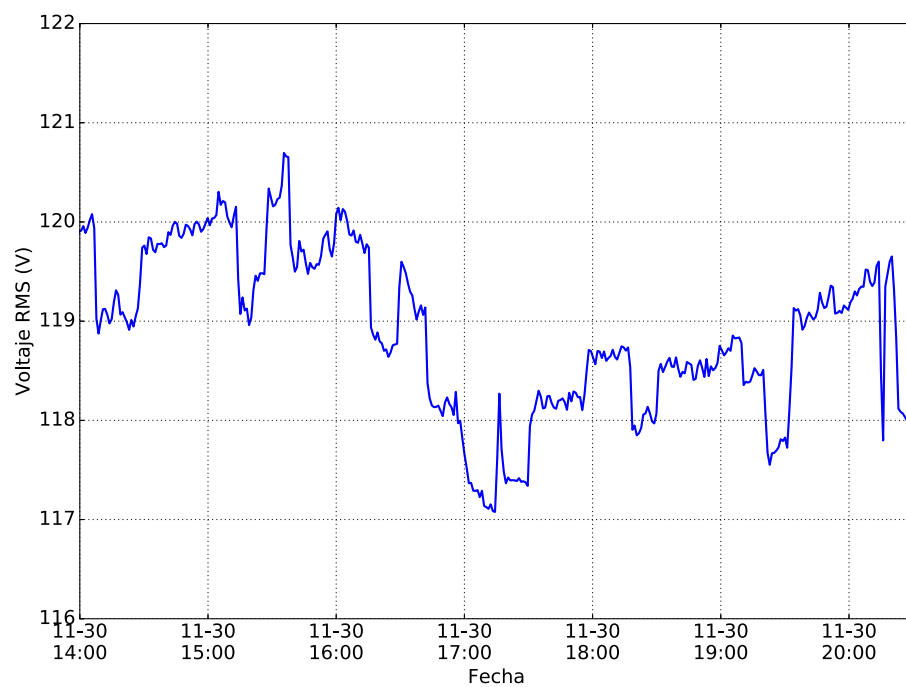


Figura 4.17: Voltaje del sistema.

Capítulo 5

Conclusiones Generales

5.1. Conclusiones

La presente tesis tuvo como objetivo diseñar un medio confinado controlando las variables ambientales, utilizando un controlador digital ON/OFF con histéresis y que contará con telemetría.

En base a la información obtenida, se dividen las conclusiones de acuerdo a los objetivos específicos descritos en el primer capítulo. Además, se establecerán las propuestas para un trabajo a futuro.

La información descrita establece los componentes de la cámara frigorífica y el material del que está elaborada, esto da una idea clara de lo que puede realizar y a los parámetros que puede alcanzar.

Se presentó la metodología para el diseño e implementación de un código realizado en lenguaje Python para el control de los actuadores del sistema, a través de los puertos GPIO del ordenador embebido Raspberry Pi. Se produjo exitosamente el accionamiento de cada uno de los elementos, compresor, abanico, así como el arranque y paro del compresor.

Cabe mencionar que el código realizado esta hecho a la medida, lo que permite el desarrollo de proyectos de forma rápida y adecuada a las necesidades del mismo.

Uno de los problemas con los que se encontró en este proyecto fue sin duda la falta de equipo para trabajar con sistemas refrigeración, por lo que se tuvieron que contratar servicios técnicos para el montaje del equipo frigorífico.

La implementación del control con histéresis se realizó exitosamente, manteniendo temperatura del interior dentro de los parámetros que fueron establecidos de una manera confiable a lo largo de las pruebas.

5.2. Aportaciones

Entre las principales aportaciones de éste trabajo se tiene las siguientes:

- Elaboración de un programa de lenguaje abierto Python para controlar el sistema electromecánico en consola en sitio.
- Elaboración de una interfaz gráfica mediante un código en lenguaje Python para monitorizar el estado y controlar el sistema de manera remota.
- La implementación de un controlador digital ON/OFF con histéresis diferente al de fábrica.

5.3. Trabajo a Futuro

Es importante señalar que aunque se considera que se cumplió el objetivo de este trabajo de tesis, en el desarrollo del mismo se encontraron varios aspectos que salen del alcance del trabajo. Los trabajos futuros que se consideran son:

- Expandir las opciones del sistema de monitorización remoto.
- Realizar un análisis a largo plazo y con los datos obtenidos verificar la confiabilidad del sistema en campo.
- Implementar sensores, como de niveles de presión y porcentaje de distintos gases que conforman la atmósfera al interior del medio confinado.

- Implementar un controlador con histéresis para la humedad relativa en el interior.
- Mejorar el aislamiento del medio confinado.
- Integrar un sistema de medición digital de parámetros eléctricos en el sistema.
- Implementar un sistema de respaldo remoto.

Apéndice A

Código fuente de sistema de control

```
#!/usr/bin/env python3

import spidev
import time
import RPi.GPIO as GPIO
import sys
import sqlite3
import os
import math
import st7565
import xglcd_font as font
from pad4pi import rpi_gpio
import threading

# Versión Final: 01/09/2016

'''
Control ON-OFF por Histeresis para el medio confinado frio
```

Mediante el uso de interrupciones

El teclado 4x4 funciona de la siguiente manera

tecleas un nuevo setpoint y para finalizar

con las teclas:

A- Cambias el Setpoint de temperatura

B- Cambias el Setpoint de Humedad

C- Cambias la histeresis de Temperatura

D- Cambias la Histeresis de Humedad

,,,

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(20, GPIO.OUT)    #compresor
```

```
GPIO.setup(21, GPIO.OUT)    #abanico
```

```
GPIO.setup(19, GPIO.OUT)    #humidificador
```

```
GPIO.setup(26, GPIO.OUT)    #extractor
```

```
spi = spidev.SpiDev()
```

```
spi.open(0, 1)  #bus, device
```

```
glcd = st7565.Glcd(rgb=[13, 6, 5])  #iniciar pantalla
```

```
glcd.init()
```

```
#glcd.set_backlight_color(0, 100, 100)  #color neon
```

```
glcd.set_backlight_color(0, 0, 0)
```

```
fixed = font.XglcdFont('/home/pi/Webapp/fonts/FixedFont5x8.c', 5, 8)
```

```
setTemp = 20.00 #Set points iniciales
```

```
setHume = 50.00
```

```
HistTemp = 1.00 #Histeresis de temperatura en grados centigrados
HistHume = 5.00 #Histeresis de humedad en % de humedad relativa
Centigrados = 0.00
Humedad = 0.00

Vadc = 0.0

A = 8.54942e-04 #proporcionados por fabricante HTM2500LF
B = 2.57305e-04
C = 1.65368e-07

R1 = 10000.0 #resistor fijo
R2 = 0.0 #resistencia variable del termistor

Steinhart = 0.0
comun = 0.0
prom = 0.0
suma = 0.0

newSet = ""
newSetfloat = 0.00
setTempTMP = 0.00
setHumeTMP = 0.00
HistTempTMP = 0.00
HistHumeTMP = 0.00
Cambio = 0
Relays = 1

KEYPAD = [
```

```
[1,2,3,"A"],
[4,5,6,"B"],
[7,8,9,"C"],
["*",0,"#","D"]
]

ROW_PINS = [14, 15, 18, 23] # BCM numbering
COL_PINS = [24, 25, 12, 16] # BCM numbering

#funcion que lee el ADC
def readadc(adcnum):

# read SPI data from MCP3004 chip, 8 possible adc's (0 thru 7)
    if adcnum > 7 or adcnum < 0:
        return -1

    r = spi.xfer2([12+(adcnum >> 1), adcnum << 7, 0])
    adcout = ((r[1] & 15) << 8) + r[2]

    return adcout

#funcion que calcula la temperatura del termistor
def temperatura():

    suma = 0.0
    prom = 0.0

    for i in range(0,10):
        value = readadc(0)
```

```
    suma += value

prom = suma / 10

Vadc = (prom * 3300) / 4095
print (Vadc)
R2 = (Vadc * R1) / (3300 - Vadc) #calculamos R2
#R2 = 25000    #pormientras
#calculamos una vez el logaritmo para ahorrar operaciones
comun = math.log(R2);
#formula de SteinHart-Hart
Steinhart = (A)+(B * comun)+(C* (comun * comun * comun));
Centi = (1 / Steinhart) - 273.15; #Transformados a Centigrados
Centi = round(Centi,2) #redondeamos a 2 digitos
return Centi

def humedad():

    suma = 0.00
    prom = 0.00

    for i in range(0,10):
        value = readadc(1)
        suma += value

    prom = suma / 10

    Vadc = (prom * 3300) / 4095
```

```

Vadc = Vadc * (4.0 / 3.33)    #regresar del divisor de voltaje de 4.0 a 3.3v

#RH = 0.0375 * Vadc - 37.7    # ecuacion para humedad lineal

RH = ( (-1.92e-9 * (Vadc * Vadc * Vadc)) + (1.44e-5 * (Vadc * Vadc))
      + (3.4e-3 * Vadc) ) - 12.4 # ecuacion para humedad polinomial

RH = round(RH,2)

return RH

#funcion que imprime en la pantalla LCD
def drawLCD():

    global Centigrados, Humedad, setTemp, setHume

    TituloTxt = 'Control ON-OFF'
    TemperaturaTxt = 'Temperatura:'
    HumedadTxt = 'Humedad:'
    TempSetPointTxt = 'Temp Set:'
    HumeSetPointTxt = 'Hume Set:'
    CentChar = 'C'    #\u0223 ??
    HumiChar = '%'

    TempValTxT = "{:.2f}".format(Centigrados)    #convierte Float a String
    HumiValTxT = "{:.2f}".format(Humedad)        #convierte Float a String
    SetTempValTxT = "{:.2f}".format(setTemp)     #convierte Float a String
    setHumeValTxT = "{:.2f}".format(setHume)     #convierte Float a String

```



```

glcd.clear_display()          #borra la pantalla
glcd.clear_back_buffer()     #borra el buffer anterior

glcd.draw_rectangle(0, 0, 128, 64)          #dibuja marco
glcd.draw_string(TituloTxt, fixed, 22, 2)   #texto
glcd.draw_string(TemperaturaTxt, fixed, 2, 17) #texto
glcd.draw_string(HumedadTxt, fixed, 2, 25)   #texto
glcd.draw_string(TempSetPointTxt, fixed, 2, 33) #texto
glcd.draw_string(HumeSetPointTxt, fixed, 2, 41) #texto

glcd.draw_string(TempValTxT, fixed, 75, 17)   #texto Temperatura Var
glcd.draw_string(HumiValTxT, fixed, 75, 25)   #texto Humedad Var
glcd.draw_string(SetTempValTxT, fixed, 75, 33) #texto Set Temp Var
glcd.draw_string(setHumeValTxT, fixed, 75, 41) #texto Set Humi Var

glcd.draw_string(CentChar, fixed, 110, 17)
glcd.draw_string(HumiChar, fixed, 110, 25)
glcd.draw_string(CentChar, fixed, 110, 33)
glcd.draw_string(HumiChar, fixed, 110, 41)

glcd.flip() #comando para imprimir en pantalla
return

#funcion de control de los relays
def actuadores():
    global setTemp, setHume, HistTemp, HistHume, Centigrados, Humedad

```

```
print("TempHTM: ", "{0:.2f}".format(round(Centigrados,2)))
print("Hume: ", "{0:.2f}".format(round(Humedad,2)))

if Centigrados > (setTemp + HistTemp):

    GPIO.output(20,GPIO.HIGH)
    GPIO.output(21,GPIO.HIGH)    #abanicos
    print ('Compresor ON')

elif Centigrados <= (setTemp - HistTemp):

    GPIO.output(20,GPIO.LOW)
    GPIO.output(21,GPIO.HIGH)    #abanicos
    print ('Compresor OFF')

if Humedad > (setHume + HistHume):

    GPIO.output(19,GPIO.LOW)
    GPIO.output(26,GPIO.HIGH)

elif Humedad <= (setHume - HistHume):

    GPIO.output(19,GPIO.HIGH)
    GPIO.output(26,GPIO.LOW)

#funciones para el keypad
def digit_entered(key):
```

```

global newSet

newSet += str(key)
print(newSet)

#funcion del keypad
def non_digit_entered(key):
    global newSet, newSetfloat, setTempTMP, setHumeTMP, HistTempTMP,
    HistHumeTMP, Cambio, setTemp, setHume, HistTemp, HistHume

    if key == "*" and len(newSet) >= 0:      #punto decimal
        newSet += '.'
        print(newSet)

    if key == "A" and len(newSet) > 0:
        #al teclear ".." + "A" manda comando para actualizar todos los setpoints modificados
        print(newSet)
        if newSet == "..":
            Cambio = 0
            print("Ingresaste Nuevos Setpoints:", setTempTMP,
                setHumeTMP, HistTempTMP, HistHumeTMP)
            setTemp = setTempTMP
            setHume = setHumeTMP
            HistTemp = HistTempTMP
            HistHume = HistHumeTMP
            drawLCD()
            sqliteWriteCambios()      #almacenar en base de datos
            #Cambio = 0

```

```
elif newSet == ".":
    newSet = ""

else:

    newSetfloat = float(newSet) #modifica setTemp
    setTempTMP = round(newSetfloat,2)

newSet = ""

if key == "B" and len(newSet) > 0:
    #al teclear ".." + "B" manda comando para cancelar las modificaciones de los setpo
    print(newSet)
    if newSet == "..":
        setTempTMP = setTemp
        setHumeTMP = setHume
        HistTempTMP = HistTemp
        HistHumeTMP = HistHume
        print("No cambiaste Setpoints")

elif newSet == ".":
    newSet = ""

else:

    newSetfloat = float(newSet) #modifica setHume
    setHumeTMP = round(newSetfloat,2)
```

```
newSet = ""

if key == "C" and len(newSet) > 0:      #modifica HistTemp
    print(newSet)

    if newSet == ".":
        newSet = ""
    else:
        newSetfloat = float(newSet)
        HistTempTMP = round(newSetfloat,2)

newSet = ""

if key == "D" and len(newSet) > 0:      #modifica HistHume
    print(newSet)
    if newSet == ".":
        newSet = ""
    else:
        newSetfloat = float(newSet)
        HistHumeTMP = round(newSetfloat,2)

newSet = ""

if key == "#" and len(newSet) > 0:      #Backspace en el teclado
    print("Backspace")
```

```
        newSet = newSet[:-1]
        print(newSet)

#funcion del keypad
def key_pressed(key):
    try:
        int_key = int(key)
        if int_key >= 0 and int_key <= 9:
            digit_entered(key)
    except ValueError:
        non_digit_entered(key)

#funcion de sqlite: escritura en la tabla de mediciones de la base de datos
def sqliteWriteMediciones():
    global Centigrados, Humedad, setTemp, setHume, HistTemp, HistHume

    Centigrados = round(Centigrados,2)          #redondeamos a 2 digitos
    Humedad = round(Humedad,2)                  #redondeamos a 2 digitos

    #connect to database file
    dbconnect = sqlite3.connect("my.db")
    #If we want to access columns by name we need to set
    #row_factory to sqlite3.Row class
    dbconnect.row_factory = sqlite3.Row
    #now we create a cursor to work with db
    cursor = dbconnect.cursor()
    #leer el tiempo
    now = time.strftime('%Y-%m-%d %H:%M')
```

```

sql = "INSERT INTO caja (Temperatura, Humedad, SetTemp, SetHume,
    HistTemp, HistHume, Date) VALUES ('" + str(Centigrados).strip('()')
    + "', '" + str(Humedad).strip('()') + "', '" + str(setTemp).strip('()')
    + "', '" + str(setHume).strip('()') + "', '" + str(HistTemp).strip('()')
    + "', '" + str(HistHume).strip('()') + "', '" + now + "')"
print(sql)
#Insert a row of data
cursor.execute(sql)
#Save (commit) the changes
dbconnect.commit()
#close the connection
dbconnect.close()

#funcion de sqlite: escritura en la tabla de configuracion de la base de datos
def sqliteWriteCambios():
    global setTempTMP, setHumeTMP, HistTempTMP, HistHumeTMP, Cambio, Relays

    #connect to database file
    dbconnect = sqlite3.connect("my.db")
    #If we want to access columns by name we need to set
    #row_factory to sqlite3.Row class
    dbconnect.row_factory = sqlite3.Row
    #now we create a cursor to work with db
    cursor = dbconnect.cursor()
    #leer el tiempo
    now = time.strftime('%Y-%m-%d %H:%M')
    sql = "INSERT INTO config (SetTemp, SetHume, HistTemp, HistHume, Date,
    Cambio, Relays) VALUES ('" + str(setTempTMP).strip('()') + "', '" +

```

```
    str(setHumeTMP).strip('()') + "', '" + str(HistTempTMP).strip('()')
    + "', '" + str(HistHumeTMP).strip('()') + "', '" + now + "', '"
    + str(Cambio).strip('()') + "', '" + str(Relays).strip('()') + '"')
print(sql)
#Insert a row of data
cursor.execute(sql)
#Save (commit) the changes
dbconnect.commit()
#close the connection
dbconnect.close()

#funcion que busca cambios en la base de datos
def sqliteSeekChanges():
    global setTemp, setHume, HistTemp, HistHume, Cambio, Relays

    #connect to database file
    dbconnect = sqlite3.connect("my.db")
    dbconnect.row_factory = sqlite3.Row
    cursor = dbconnect.cursor()
    cursor.execute('''SELECT * FROM config ORDER BY Id DESC LIMIT 1;''')
    lrow = cursor.fetchone()

    flag = lrow[6]

    if flag == 1:
        setTemp = lrow[1]
        setHume = lrow[2]
        HistTemp = lrow[3]
```



```
HistHume = lrow[4]
Relays = lrow[7]
print("Hubo cambios en los setpoints via Web:", setTemp,
      setHume, HistTemp, HistHume, Relays)

Cambio = str(Cambio).strip('()')

cursor.execute('''UPDATE config SET Cambio = ? ORDER BY Id DESC LIMIT 1;''', (
dbconnect.commit()

#update cambio a 0 en la tabla despues de obtener nuevos setpoints

else:
    print("Sin Cambios en Setpoints via Web")

dbconnect.close()

def checkTableConfig():
    global setTemp, setHume, HistTemp, HistHume, Cambio, Relays

    dbconnect = sqlite3.connect("my.db")
    cursor = dbconnect.cursor()

    #checa si la tabla mediciones existe, sino la crea
    cursor.execute('''CREATE TABLE IF NOT EXISTS caja(Id INTEGER PRIMARY KEY AUTOINCRE
    Temperatura NUMERIC NOT NULL, Humedad NUMERIC NOT NULL, SetTemp NUMERIC NOT NULL,
    SetHume NUMERIC NOT NULL, HistTemp NUMERIC NOT NULL, HistHume NUMERIC NOT NULL, D
    dbconnect.commit()
```

```
#checa si la tabla config existe, sino la crea
cursor.execute('''CREATE TABLE IF NOT EXISTS config(Id INTEGER PRIMARY KEY AUTOINCR
SetTemp NUMERIC NOT NULL, SetHume NUMERIC NOT NULL, HistTemp NUMERIC NOT NULL,
    HistHume NUMERIC NOT NULL, Date TEXT NOT NULL, Cambio INTEGER NOT NULL, Relays IN

dbconnect.commit()

#checa si la tabla usuarios existe, sino la crea
cursor.execute('''CREATE TABLE IF NOT EXISTS usuarios(Id INTEGER PRIMARY KEY AUTOINCR
    Name TEXT NOT NULL UNIQUE ON CONFLICT REPLACE, Password TEXT NOT NULL, Tipo TEXT

dbconnect.commit()

sql = "INSERT INTO usuarios (Name, Password, Tipo) VALUES ('" + "admin" + "', '" +
#Insertar el primer usuario administrador
cursor.execute(sql)
dbconnect.commit()

dbconnect.row_factory = sqlite3.Row

#busca si existe un renglon de configuracion, sino existe ingresa los datos de def
cursor.execute('''SELECT * FROM config ORDER BY Id DESC LIMIT 1;''')
existrow = cursor.fetchone()

if existrow is None:
    print("No hay datos de configuracion, Autoconfigurando...")

    now = time.strftime('%Y-%m-%d %H:%M')
    Cambio = str(Cambio).strip('()')
    Relays = str(Relays).strip('()')
```

```

        sql = "INSERT INTO config (SetTemp, SetHume, HistTemp, HistHume, Date, Cambio,
        VALUES ('" + str(setTemp).strip('()') + "', '" + str(setHume).strip('()') + "
        + str(HistTemp).strip('()') + "', '" + str(HistHume).strip('()') + "', '" + no
        + str(Cambio).strip('()') + "', '" + str(Relays).strip('()') + "')"
        print(sql)
        cursor.execute(sql)
        dbconnect.commit()

    else:
        print("Hay datos de configuracion...")

    dbconnect.close()

#funcion de autoconfiguracion iniciales
def initConfig():
    global setTemp, setHume, HistTemp, HistHume, setTempTMP, setHumeTMP,
    HistTempTMP, HistHumeTMP, Cambio, Relays

    dbconnect = sqlite3.connect("my.db")
    dbconnect.row_factory = sqlite3.Row
    cursor = dbconnect.cursor()

    cursor.execute('''SELECT * FROM config ORDER BY Id DESC LIMIT 1;''')
    lrow = cursor.fetchone()

    #flag = lrow[6]

```

```
#if flag == 1:

setTemp = lrow[1]
setHume = lrow[2]
HistTemp = lrow[3]
HistHume = lrow[4]
Cambio = lrow[6]
Relays = lrow[7]

print("Configuracion Inicial:", setTemp, setHume, HistTemp,
      HistHume, Cambio, Relays, "")

dbconnect.close()

setTempTMP = setTemp
setHumeTMP = setHume
HistTempTMP = HistTemp
HistHumeTMP = HistHume      #para evitar el error de meter setpoints en ceros

#funcion que apaga relays en caso que el usuario lo desee
def relaysoff():
    GPIO.output(20,GPIO.LOW)
    GPIO.output(21,GPIO.LOW)
    GPIO.output(19,GPIO.LOW)
    GPIO.output(26,GPIO.LOW)

#funcion general de control
```

```
def control():
    global Centigrados, Humedad, Relays

    Centigrados = temperatura()    #leer Termistor
    Humedad = humedad()            #leer Humedad
    drawLCD()                      #funcion de pantalla lcd
    if Relays == 1:
        actuadores()              #funcion de relays
    elif Relays == 0:
        relaysoff()

#programa principal
try:
    GPIO.output(20,GPIO.LOW)
    GPIO.output(21,GPIO.LOW)
    GPIO.output(19,GPIO.LOW)
    GPIO.output(26,GPIO.LOW)

    #lectura
    factory = rpi_gpio.KeypadFactory()
    keypad = factory.create_keypad(keypad=KEYPAD, row_pins=ROW_PINS, col_pins=COL_PINS)
    keypad.registerKeyPressHandler(key_pressed)

    checkTableConfig()
    initConfig()

    while True:
```

```
#start = time.time()

#GPIO.output(21,GPIO.HIGH) #abanico siempre ON

control()                #control
sqliteWriteMediciones()   #almacenar en base de datos cada minuto (tabla medi
time.sleep(15)

sqliteSeekChanges()       #busca cambios de configuracion de la base de datos
control()                 #control
time.sleep(15)

sqliteSeekChanges()       #busca cambios de configuracion de la base de datos
control()                 #control
time.sleep(15)

sqliteSeekChanges()       #busca cambios de configuracion de la base de datos
control()                 #control
time.sleep(15)

#print(time.time()-start)

#salida del programa
except (KeyboardInterrupt, SystemExit):
    glcd.clear_display()
    glcd.clear_back_buffer()
    glcd.cleanup()
    spi.close()
```

```
GPIO.cleanup()
keypad.cleanup()
print("Adios")
sys.exit()
```

Apéndice B

Código fuente de sistema de telemetría

```
from flask import Flask, render_template, redirect, request,
flash, url_for, session, g, abort, jsonify
from flask_sqlalchemy import SQLAlchemy
import flask_excel as excel
from sqlalchemy import text, func
import os
import sys
import time, datetime
```

```
# Fecha de Ultima Modificacion: 29/11/016
# VERSION FINAL
```

```
app = Flask(__name__, static_url_path='')
```



```
app.secret_key = os.urandom(24)

file_path = os.path.abspath(os.getcwd())+"/my.db"
app.config['SQLALCHEMY_DATABASE_URI']='sqlite:///'+file_path

#omite los warnings en la consola
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
#5 minutos dura la sesion
app.config['PERMANENT_SESSION_LIFETIME'] = 300

db = SQLAlchemy(app)

class Caja(db.Model):
    __tablename__ = "caja"
    Id = db.Column(db.Integer, primary_key=True)
    Temperatura = db.Column(db.Numeric)
    Humedad = db.Column(db.Numeric)
    SetTemp = db.Column(db.Numeric)
    SetHume = db.Column(db.Numeric)
    HistTemp = db.Column(db.Numeric)
    HistHume = db.Column(db.Numeric)
    Date = db.Column(db.String(80))

    def __init__(self, Temperatura, Humedad, SetTemp, SetHume,
        HistTemp, HistHume, Date):
        self.Temperatura = Temperatura
        self.Humedad = Humedad
```

```
        self.SetTemp = SetTemp
        self.SetHume = SetHume
        self.HistTemp = HistTemp
        self.HistHume = HistHume
        self.Date = Date

class Config(db.Model):
    __tablename__ = "config"
    Id = db.Column(db.Integer, primary_key=True)
    SetTemp = db.Column(db.Numeric)
    SetHume = db.Column(db.Numeric)
    HistTemp = db.Column(db.Numeric)
    HistHume = db.Column(db.Numeric)
    Date = db.Column(db.String(80))
    Cambio = db.Column(db.Numeric)
    Relays = db.Column(db.Numeric)

    def __init__(self, SetTemp, SetHume, HistTemp,
        HistHume, Date, Cambio, Relays):
        self.SetTemp = SetTemp
        self.SetHume = SetHume
        self.HistTemp = HistTemp
        self.HistHume = HistHume
        self.Date = Date
        self.Cambio = Cambio
        self.Relays = Relays
```

```
class Usuarios(db.Model):
    __tablename__ = "usuarios"
    Id = db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String(80))
    Password = db.Column(db.String(80))
    Tipo = db.Column(db.String(80))

    def __init__(self, Name, Password, Tipo):
        self.Name = Name
        self.Password = Password
        self.Tipo = Tipo

@app.route("/")
@app.route("/index.html")
def home():

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("index.html", TablaRelays = TablaRelays)

@app.route("/acerca.html")
def acerca():

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
```

```
return render_template("acerca.html", TablaRelays = TablaRelays)

@app.route("/actual.html")
def actual():
    ## Calcula Maximo, Minimo y Promedio del ultimo dia

    allDataConfig = Config.query.order_by(Config.Id.desc()).first()
    ConfigSetTemp = allDataConfig.SetTemp
    ConfigSetHume = allDataConfig.SetHume
    ConfigHistTemp = allDataConfig.HistTemp
    ConfigHistHume = allDataConfig.HistHume
    ConfigDate = allDataConfig.Date
    ConfigRelays = allDataConfig.Relays #para boton encendido en la pagina

    TablaConfig = [ConfigSetTemp, ConfigSetHume, ConfigHistTemp,
        ConfigHistHume, ConfigDate, ConfigRelays]

    allDataMed = Caja.query.order_by(Caja.Id.desc()).first()
    MedTemp = allDataMed.Temperatura
    MedHume = allDataMed.Humedad
    MedDate = allDataMed.Date

    TablaMed = [MedTemp, MedHume, MedDate]

    #maximos, minimos, promedios del dia
```

```

dummy = allDataMed.Date
dummy2 = datetime.datetime.strptime(dummy, '%Y-%m-%d %H:%M') - datetime.timedelta(

allDataFunc = Caja.query.with_entities(func.min(Caja.Temperatura).label('min_temp')
func.avg(Caja.Temperatura).label('avg_temp'),
func.max(Caja.Temperatura).label('max_temp'), func.min(Caja.Humedad)
func.avg(Caja.Humedad).label('avg_hume'),
func.max(Caja.Humedad).label('max_hume')).filter(Caja.Date <= dumm

TempMin = allDataFunc.min_temp
TempAvg = allDataFunc.avg_temp
TempMax = allDataFunc.max_temp

HumeMin = allDataFunc.min_hume
HumeAvg = allDataFunc.avg_hume
HumeMax = allDataFunc.max_hume

TablaProm = [TempMin, TempAvg, TempMax, HumeMin, HumeAvg, HumeMax]

#Rutina de Graficado

now = time.strftime('%Y-%m-%d %H:%M')
#checa la fecha del ultimo dato ingresado
last = MedDate

##
return render_template("actual.html", TablaConfig = TablaConfig,
TablaMed = TablaMed, TablaProm = TablaProm, now=now, last = last)

```

```
@app.route("/config.html")
def login_parametros():

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("config.html", allDataConfig=Config.query.order_by(Config.Id.desc()).first(),
                           TablaRelays = TablaRelays)

@app.route("/post_config.html")
def parametros():

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("post_config.html", allDataConfig=Config.query.order_by(Config.Id.desc()).first(),
                           TablaRelays = TablaRelays)

@app.route("/new_config.html")
def new_parametros():

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("new_config.html", allDataConfig=Config.query.order_by(Config.Id.desc()).first(),
                           TablaRelays = TablaRelays)
```

```
@app.route("/loginuser", methods=["GET", "POST"])
def loginuser():

    if request.method == "GET":
        return redirect(url_for('login_parametros'))

    else:
        Username = request.form["username"].strip()
        Password = request.form["password"].strip()

        registered_user = Usuarios.query.filter_by(Name=Username,
            Password=Password).first()

        if registered_user is None:
            flash('Usuario o Contraseña Incorrectos')
            return redirect(url_for('login_parametros'))

        else:
            flash(u'Bienvenido: %s' % Username)
            session['logged_in'] = True
            return redirect(url_for('parametros'))

@app.route("/logoutuser", methods=["GET", "POST"])
def logoutuser():

    session.pop('logged_in', None)
```

```
flash('Cerró su Sesión')
return redirect(url_for('login_parametros'))

@app.route("/modifyconfig", methods=["GET", "POST"])
def configuracion():
    if request.method == "GET":
        allDataRelays = Config.query.order_by(Config.Id.desc()).first()
        ConfigRelays = allDataRelays.Relays    #para boton encendido en la pagina
        TablaRelays = [ConfigRelays]
        return render_template("config.html", TablaRelays = TablaRelays)

    else:
        allDataConfig = Config.query.order_by(Config.Id.desc()).first()
        Relays = allDataConfig.Relays

        SetTemp = request.form["SetTemp"].strip()
        SetHume = request.form["SetHume"].strip()
        HistTemp = request.form["HistTemp"].strip()
        HistHume = request.form["HistHume"].strip()
        now = time.strftime('%Y-%m-%d %H:%M')
        Date = now
        Cambio = "1"

        NuevaConfig = Config(SetTemp=SetTemp, SetHume=SetHume, HistTemp=HistTemp,
                               HistHume=HistHume, Date = Date, Cambio = Cambio, Relays = Relays)

        db.session.add(NuevaConfig)
```



```
db.session.commit()

flash('Nuevos Parámetros Ingresados')

return redirect(url_for('new_parametros'))

@app.route("/relays", methods=["GET", "POST"])
def relaysonoff():
    if request.method == "GET":

        allDataRelays = Config.query.order_by(Config.Id.desc()).first()
        ConfigRelays = allDataRelays.Relays #para boton encendido en la pagina
        TablaRelays = [ConfigRelays]
        return render_template("config.html", TablaRelays = TablaRelays)

    else:

        Seleccion = request.form['opciones'].strip()

        allDataConfig = Config.query.order_by(Config.Id.desc()).first()
        SetTemp = allDataConfig.SetTemp
        SetHume = allDataConfig.SetHume
        HistTemp = allDataConfig.HistTemp
        HistHume = allDataConfig.HistHume

        now = time.strftime('%Y-%m-%d %H:%M')
        Date = now
        Cambio = "1"
```

```
if Seleccion == "Encender":

    flash('Equipo Encendido')
    Relays = "1"

elif Seleccion == "Apagar":
    flash('Equipo Apagado')
    Relays = "0"

NuevaConfig = Config(SetTemp=SetTemp, SetHume=SetHume, HistTemp=HistTemp,
    HistHume=HistHume, Date = Date, Cambio = Cambio, Relays = Relays)

db.session.add(NuevaConfig)
db.session.commit()

return redirect(url_for('login_parametros'))

@app.route("/graph.html")
def graficaweb():
    now = time.strftime('%Y-%m-%d %H:%M')
    #checa la fecha del ultimo dato ingresado
    allDataMed = Caja.query.order_by(Caja.Id.desc()).first()
    last = allDataMed.Date

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
```

```
ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
TablaRelays = [ConfigRelays]

return render_template("graph.html", now=now, last = last, TablaRelays = TablaRelays)


@app.route("/graficar", methods=["GET", "POST"])
def graficarplot():
    if request.method == "GET":
        return redirect(url_for("actual"))

    else:
        fch = request.form["date"]
        lst = Caja.query.all()

        now = time.strftime('%Y-%m-%d %H:%M')

        listvar = request.form.getlist('variable')

        sql = "SELECT Date, "
        sql += ", ".join(listvar)
        sql += " FROM Caja WHERE Date >= " + "datetime('" + request.form.get('date') + " "
        sql += "-" + request.form.get('tiempo') + " " + request.form.get('tipot') + "');"

        sql += " and Date <= " + "datetime('" + request.form.get('date') + " "
        sql += "-" + request.form.get('tiempo') + " " + request.form.get('tipot') + "');"

        result = db.engine.execute(sql)
        salida=[]
        for fila in result:
            salida.append(fila)
```

```

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("graphnew.html",varlis = lst, last = salida, fecha = fecha,
        listvar = listvar, TablaRelays = TablaRelays)

@app.route("/exporta.html")
def exportaweb():
    now = time.strftime('%Y-%m-%d %H:%M')
    #checa la fecha del ultimo dato ingresado
    allDataMed = Caja.query.order_by(Caja.Id.desc()).first()
    last = allDataMed.Date

    allDataRelays = Config.query.order_by(Config.Id.desc()).first()
    ConfigRelays = allDataRelays.Relays          #para boton encendido en la pagina
    TablaRelays = [ConfigRelays]
    return render_template("graph.html", now=now, last = last, TablaRelays = TablaRelays)

@app.route("/export", methods=["GET", "POST"])
def export_records():
    #return excel.make_response_from_array([[1,2], [3, 4]], "csv", file_name="export_data.csv")

    fch2 = request.form["date2"]
    lst2 = Caja.query.all()

```

```
now = time.strftime('%Y-%m-%d %H:%M')

listvar = request.form.getlist('variable2')

sql = "SELECT Date, "
sql += ", ".join(listvar)
sql += " FROM Caja WHERE Date >= " + "datetime('" + request.form.get('date2') + "',"
      + "-" + request.form.get('tiempo2') + " " + request.form.get('tipot2') + "');"

sql += " and Date <= " + "datetime('" + request.form.get('date2') + "');"

result2 = db.engine.execute(sql)
salida2=[]
for fila2 in result2:
    salida2.append(fila2)

return excel.make_response_from_array(salida2, "csv", file_name="export_data")

#return excel.make_response_from_array([[1,2], [3, 4]], "xlsx", file_name="export_

if __name__ == "__main__":
    app.run(debug=True)
    #app.run(debug=True, host="0.0.0.0")
```

Apéndice C

Esquemático del Circuito de la Consola de Control

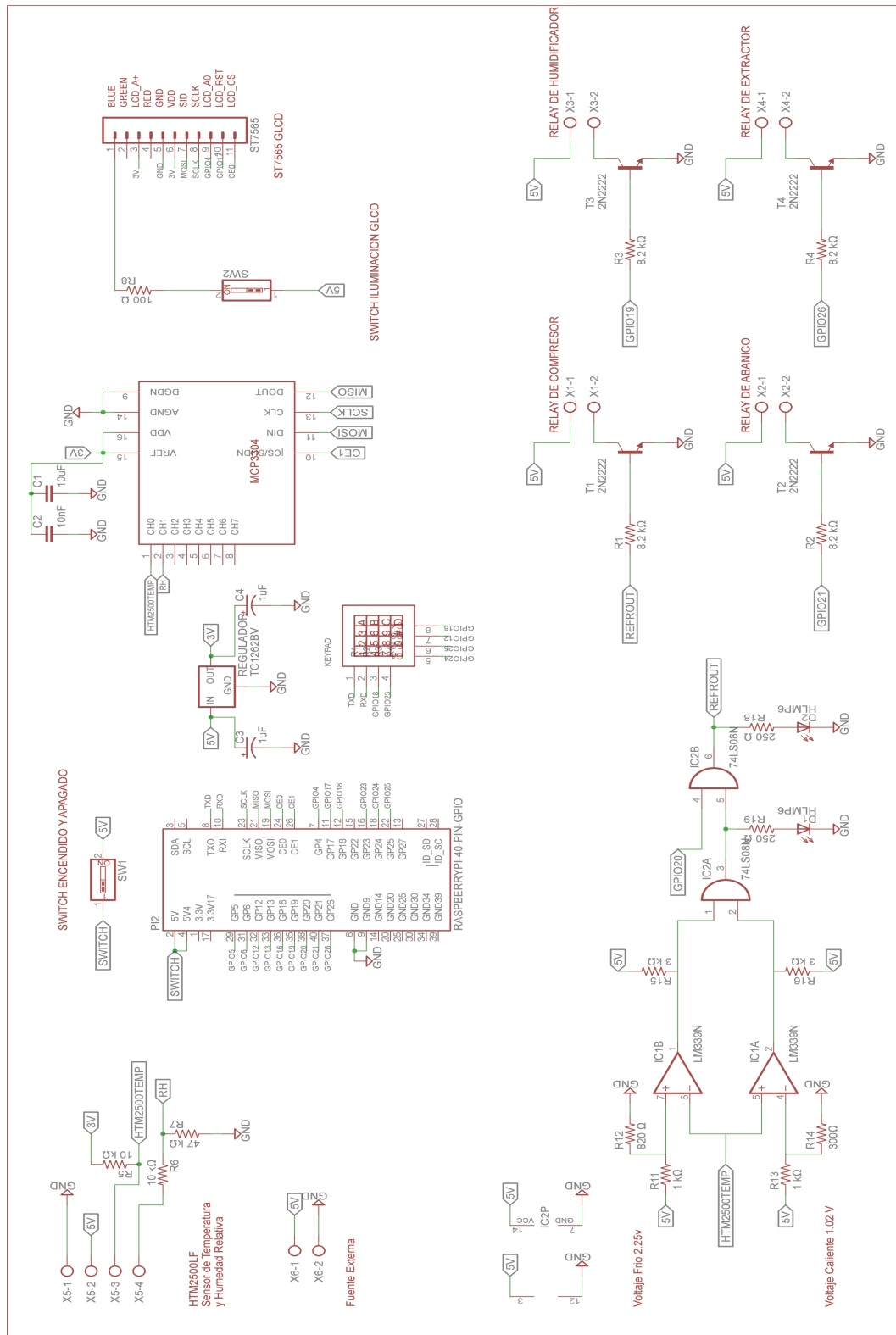


Figura C.1: Esquema del circuito del prototipo.

Bibliografía

- [1] INEGI. Cuentame de México, Agricultura, 2010.
- [2] O Dhamija. *Manual de control de la calidad de los alimentos: alimentos para la exportación*. 1991.
- [3] FAO. Revisión del Código de Prácticas para la Elaboración y Manipulación de los Alimentos Congelados Rápidamente, 2001.
- [4] Eduardo Umaña Cerros. Conservación de Alimentos por Frio. Technical report, 2010.
- [5] Salvador Medina Ramirez. Las cadenas de frío y el transporte refrigerado en México. *Revista Comercio Exterior*, 59(12):1010–1017, 2009.
- [6] Adel A Kitinoja, Lisa; Kader. Capítulo 6: Control de temperatura y humedad relativa, 1996.
- [7] Sagarpa. 1er Informe de Labores Sagarpa 2012-2013. Technical report, 2013.
- [8] Diario Oficial de la Federación. Programa Nacional México sin Hambre 2014-2018., 2014.
- [9] Calixto R Mendoza Roca. *Manual práctico para gestión logística: envase y empaque, transporte y cadena de frío, preservación de productos del agro*. 2015.
- [10] A F L GÓMES, P.A. CAMELO. Calidad postcosecha de tomates almacenados en atmósferas controladas. *Horticultura Brasileira, Brasília*, (1):38–43, 2002.

- [11] María Elizabeth Morales Pérez, Carmen Guadalupe; De la Torre Moreno. Características del transporte en México. 2006.
- [12] L y Sañudo A Báez Sañudo, M., Siller Cepeda J., Muy, D., Araiza E. Contreras. SmartfreshTM: Una novedosa tecnología para extender vida de anaquel en tomate. In *XXXII Congreso nacional de ciencia y tecnología de los alimentos.*, 2002.
- [13] Thermo-King. Contenedores Magnum. Technical report, 2014.
- [14] Zhiyuan Gao, Yingju Jia, Hongwei Zhang, and Xiaohui Li. A Design of Temperature and Humidity Remote Monitoring System Based on Wireless Sensor Network Technology. 2012.
- [15] Pablo Gonzalez Garcia. Diseño de una herramienta de planificación de sistemas WiMAX. 2006.
- [16] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition, 2001.
- [17] Paul E Tippens. *Física : Conceptos y Aplicaciones*. Mc Graw Hill, México D.F., 7a ed. rev edition, 2011.
- [18] Stabilit. Sistema constructivo Glasliner. Technical report, 2014.
- [19] Owens Corning. Foamular Foamular. Technical report, 2006.
- [20] Owens Corning. Extruded Polystyrene Insulation FOAMULAR Extruded Polystyrene Insulation. Technical report.
- [21] Dallas Semiconductor. DS18B20 Temperature Sensor. *Application Note*, pages 1–27, 2002.
- [22] Measurement Specialities. HTM2500LF – Temperature and Relative Humidity Module. Technical report, 2012.
- [23] Microchip. MCP3302/04. Technical report, 2008.

- [24] Fotek SSR Datasheet. Solid State Relay SSR-25DA. Technical report, 2012.
- [25] High Speed Switches. 2N2219a 2N2222a. Technical Report June, 1999.
- [26] Vatronix Holdings Limited. LCD Module User Manual. Technical Report 10.
- [27] Texas Instruments. LMx39-N , LM2901-N , LM3302-N Low-Power Low-Offset Voltage Quad Comparators. Technical Report 1, 2014.
- [28] Texas Instruments. Quadruple 2-Input Positive-Nand Gates Quadruple 2-Input Positive-And Gates. Technical Report January 2001, 2002.
- [29] Xiyu Liu. Grounding system of an Uninterruptible Power System (UPS). 2008.