

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
CARRERA DE ESPECIALIZACIÓN EN SISTEMAS
EMBEBIDOS



MEMORIA DEL TRABAJO FINAL

**Control automático del proceso de
fermentación de vino usando la CIAA**

Autor:
Luis Enrique Chico Capistrano

Director:
Ing. Juan Manuel Cruz

Jurados:
Dr. Ing. Pablo Gomez (FIUBA)
Esp. Ing. Franco Bucafusco
Esp. Ing. Sergio De Jesús Meleán

Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires entre febrero de 2016 y diciembre de 2017.

Resumen

En el presente trabajo se implementó un control automático de temperatura y sistema de alertas para la bodega Chico Zossi. Al mantener la temperatura estable, se mejora la calidad del vino, lo que afecta directamente al precio por litro.

La implementación se hizo sobre la CIAA NXP, mediante un sistema operativo de tiempo real y se hizo uso de la herramienta de control de versiones. Para garantizar la calidad del producto fueron empleadas técnicas de gestión de proyectos y ensayos que permitieron validar y verificar el correcto funcionamiento.

Agradecimientos

A mis padres y hermano que siempre me incentivarón con palabras y ejemplos para seguir creciendo tanto en lo personal como en lo profesional. También a ellos quiero agradecerles por haberme dado a conocer a Dios, quien es mi fuente inagotable de fortaleza, quien me ha ayudado a superar todas las dificultades y especialmente en aquellos momentos donde parece que todo se derrumba, pero Él siempre está a mi lado.

A mi primo Baltazar a quien tanto aprecio y principal motivo de la elección de este proyecto. Por sus aportes para llevar a cabo el desarrollo de este trabajo y por tantos momentos compartidos.

A mis compañeros y profesores de la especialización, quienes me acompañaron a lo largo del año transmitiendo sus conocimientos y experiencias. Especialmente a Horacio Martínez quien me prestó la CIAA-NXP y a Patricio Bos que con sus valiosos aportes hicieron posible que realizara este proyecto.

Al Dr. Ing. Ariel Lutenberg que con tanta paciencia y dedicación hizo posible que yo sea parte de este grupo de personas que buscan día a día el crecimiento del país dentro de nuestra profesión.

A todos ellos, muchas gracias.

Índice general

| | |
|---|------------|
| Resumen | III |
| 1. Introducción General | 1 |
| 1.1. Proceso de elaboración del vino | 1 |
| 1.1.1. La temperatura | 2 |
| 1.2. Motivación | 3 |
| 1.3. Proyecto CIAA | 3 |
| 1.3.1. Objetivos del Proyecto CIAA | 3 |
| 1.3.2. Plataforma CIAA-NXP | 4 |
| 2. Introducción Específica | 7 |
| 2.1. Objetivos y alcances | 7 |
| 2.2. Requerimientos | 8 |
| 2.3. Planificación del proyecto | 8 |
| 3. Diseño e Implementación | 11 |
| 3.1. Hardware | 11 |
| 3.1.1. Módulos requeridos | 12 |
| 3.1.2. Características del hardware | 12 |
| 3.2. Software | 18 |
| 3.2.1. Aplicación | 18 |
| 3.2.2. Plataforma web | 19 |
| 3.2.3. Configuración del servidor | 23 |
| 3.3. Diseño del sistema web embebido | 24 |
| 4. Ensayos y Resultados | 29 |
| 4.1. Banco de pruebas | 29 |
| 4.2. Configuraciones para la PC | 30 |
| 4.3. Pruebas funcionales del hardware | 31 |
| 5. Conclusiones | 35 |
| 5.1. Conclusiones generales | 35 |
| 5.2. Próximos pasos | 35 |
| Bibliografía | 37 |

Índice de figuras

| | |
|--|----|
| 1.1. Esquema de elaboración de vino tinto. | 2 |
| 1.2. Placa CIAA-NXP para la industria. | 4 |
| 2.1. Listado de tareas, y planificación de tiempos de la implementación. | 9 |
| 3.1. Entorno del sistema de control del equipo. | 11 |
| 3.2. Conexión de la CIAA-NXP con el simulador del sistema. | 12 |
| 3.3. Módem SIM800L | 13 |
| 3.4. Circuito de conexión del módulo SIM800L. | 14 |
| 3.5. Fuente de alimentación DC/DC. | 14 |
| 3.6. Circuito regulador de corriente. | 15 |
| 3.7. Circuito regulador de tensión. | 15 |
| 3.8. Circuito del sensor de temperatura. | 15 |
| 3.9. Esquemáticos extraídos de la EDU-CIAA. | 16 |
| 3.10. Capa superior de la placa de simulación de sensores. | 16 |
| 3.11. Capa inferior de la placa de simulación de sensores. | 17 |
| 3.12. Placa simulador de sensores 3D. | 17 |
| 3.13. Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado. | 18 |
| 3.14. Diagrama de flujo de la tarea sensor. | 19 |
| 3.15. Diagrama en flujo de la tarea módem | 20 |
| 3.16. Diagrama de flujo de la tarea control. | 21 |
| 3.17. Primera versión de la web de monitoreo. | 22 |
| 3.18. Versión actualizada haciendo uso del template. | 23 |
| 3.19. Diagrama de secuencia AJAX. | 24 |
| 3.20. Opciones del al presionar sobre el menú desplegable. | 25 |
| 3.21. Activar/Desactivar actuadores. | 26 |
| 3.22. Configurar nivel de descarga para accionar la alerta por batería baja. | 27 |
| 3.23. Configuración de los celulares. | 27 |
| 3.24. Configuración del rango de temperatura y activación del estado de control automático y alarma. | 28 |
| 3.25. Configuración de IP, máscara de red y puerta de enlace. | 28 |
| 4.1. Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado. | 29 |
| 4.2. Placa básica integrada con la CIAA-NXP. | 30 |
| 4.3. Configuración de una red estática para la interfaz Ethernet. | 31 |
| 4.4. Pantalla principal de la web de monitoreo. | 32 |
| 4.5. Activación por control automático debido a temperatura elevada. | 32 |
| 4.6. SMS recibidos debido a temperatura máxima. El segundo SMS recibido es por temperatura mínima. Y el último recibido debido a batería baja. | 33 |

| | |
|--|----|
| 4.7. Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula. | 33 |
| 4.8. Activación manual de actuador 1 y 2, electroválvula y bomba inactivo. | 34 |

Índice de Tablas

| | |
|--|----|
| 2.1. Comparación tiempo planificado vs real. | 10 |
| 3.1. Configuración por default del servidor. | 24 |

Dedicado a Giovanni y Juan Pablo

Capítulo 1

Introducción General

En este capítulo se presenta una breve introducción al proceso de elaboración del vino, a la motivación de este proyecto y al proyecto CIAA.

1.1. Proceso de elaboración del vino

El proceso de elaboración del vino se divide en:

La vendimia o cosecha: consiste en la recolección de la uva, la cual en Argentina se suele realizar entre los meses de febrero y abril.

Pesada: en esta etapa se pesa la uva de la variedad como Cabernet Sauvignon, Malbec, Syrah, Merlot y Tempranilla.

Despalillado: es el proceso mediante el cual se separa la uva del resto del racimo.

Estrujado: desgranando el racimo, las uvas se pasan por una pisadora para conseguir que se rompa la piel de la uva, llamada hollejo. Así se extrae el mosto para facilitar el siguiente paso, pero no se debe estrujar demasiado para evitar que se rompan las semillas de las uvas. Las semillas serán eliminadas.

Fermentación: el mosto ingresa a los tanques de acero inoxidable junto al hollejo y allí comienza el proceso de fermentación a una temperatura de 20-24°C durante 6-8 días.

Descube: en este proceso se separan el hollejo del vino, mediante bombas. El hollejo sufre una prensada y si el vino obtenido es de baja calidad sigue un proceso distinto y si la calidad es buena es agregado al vino obtenido por fermentación, continuando su mismo proceso de vinificación.

Primer trasiego: se eliminan las levaduras y otros tipos de sedimentos, utilizando bombas para pasar de un tanque a un depósito.

Segundo trasiego: el vino es conducido a otro tanque para hacer el segundo trasiego por medio de bombas. El segundo trasiego se efectúa para separar las borras más finas y las levaduras que pueden haber quedado en el vino.

Clarificado: se realiza en otro tanque de acero inoxidable con bombas y con bentonita. Se purifica el vino durante 24 a 72 horas. Después se realiza un trasiego para eliminar la bentonita.

Filtrado: se limpia el vino mediante filtros y con el agregado de tierras filtrantes. Posteriormente, se hacen trasiegos para eliminar los restos de tierras filtrantes.

Fraccionado: el vino de los tanques es transportado mediante bombas hacia la sala de fraccionamiento donde se llenan las botellas, se tapan, se etiquetan y se acomodan en cajas para ser comercializadas.

Se aprecia el esquema de elaboración del vino en la figura 1.1.



FIGURA 1.1: Esquema de elaboración de vino tinto.

1.1.1. La temperatura

La temperatura es un factor vital para que la fermentación siga correctamente su curso. Es importante saber que si la temperatura de fermentación es elevada esos grados de más matarán prematuramente las levaduras que en consecuencia no podrán completar su función metabólica.

Además una temperatura excesiva provocará la pérdida de elementos aromáticos y un incremento del amargor ya que el calor puede desencadenar una reacción química de las partes sólidas procedentes de la vid, como pueden ser los hollejos.

Por este motivo, la temperatura es uno de los aspectos más vigilados en el proceso de fermentación del vino.

1.2. Motivación

En el presente trabajo se busca satisfacer las necesidades de la bodega Chico Zossi, propiedad de un familiar del autor del presente trabajo. Para eso se necesita controlar un proceso delicado como es la fermentación del vino. Por el momento es realizado por personal día y noche, y es difícil conseguir personas responsables que se comprometan con el trabajo.

A su vez se busca también una satisfacción personal, ya que el autor de este trabajo está interesado en implementar en un microcontrolador una interfaz que permita visualizar la información de un sistema mediante una plataforma web, donde se pueda apreciar el estado de las entradas y poder ejercer control sobre ellas.

Para concluir, el hecho de utilizar la plataforma CIAA, permitió cumplir todo estos objetivos personales, haciendo uso de un sistema operativo de tiempo real (RTOS), como sé llamará de aquí en adelante.

1.3. Proyecto CIAA

El Proyecto CIAA cuyas siglas hacen referencia a Computadora Industrial Abierta Argentina, nació en 2013 como una iniciativa conjunta entre el sector académico y el industrial, representados por la ACSE¹ y CADIEEL², respectivamente [5].

1.3.1. Objetivos del Proyecto CIAA

Los objetivos del Proyecto CIAA son los siguientes:

1. Impulsar el desarrollo tecnológico nacional, a partir de sumar valor agregado al trabajo y a los productos y servicios, mediante el uso de sistemas electrónicos, en el marco de la vinculación de las instituciones educativas y el sistema científico-tecnológico con la industria.
2. Darle visibilidad positiva a la electrónica argentina.
3. Generar cambios estructurales en la forma en la que se desarrollan y utilizan en nuestro país los conocimientos en el ámbito de la electrónica y de las instituciones y empresas que hacen uso de ella.

Es importante destacar que La CIAA-NXP es la primera y única computadora del mundo que reúne dos cualidades:

1. Ser Industrial, ya que su diseño está preparado para las exigencias de confiabilidad, temperatura, vibraciones, ruido electromagnético, tensiones, cortocircuitos, etc., que demandan los productos y procesos industriales.

¹Asociación Civil para la investigación, promoción y desarrollo de los Sistemas electrónicos Embebidos.

²Cámara Argentina de Industrias Electrónicas, Electromecánicas y Luminotécnicas.

2. Ser Abierta, ya que toda la información sobre su diseño de hardware, firmware, software, etc. está libremente disponible en internet bajo la Licencia BSD, para que cualquiera la utilice como quiera.

1.3.2. Plataforma CIAA-NXP

La CIAA-NXP es la primer CIAA en ser diseñada, probada y producida en serie. Se puede apreciar en la figura 1.2. Está pensada para utilizarse en (pero no limitada a) equipos de automatización industrial. Se alimenta con 24V (CC). Tiene entradas digitales optoacopladas, entradas analógicas configurables 0-10V o 0-20mA, salidas digitales open-drain y relé, salida analógica configurable 0-10V o 0-20mA, interfaces de comunicación RS232, RS485, CAN, Ethernet y USB-OTG. Está en el microcontrolador LPC4337.



FIGURA 1.2: Placa CIAA-NXP para la industria.

Las especificaciones CIAA-NXP son :

CPU: Microcontrolador LPC4337JDB144³. Dual-core Cortex-M4 + Cortex-M0 @ 204MHz.

Debugger: USB-to-JTAG FT2232H⁴. Soportado por OpenOCD.

Memorias:

- Memorias internas del LPC4337. Ver Hoja de datos del LPC4337JBD144
- SDRAM 128 Mbit (IS42S16800F-7TL o compatible)
- Flash QSPI 32 Mbit (S25FL032P0XMFI011 o compatible)
- EEPROM 1 Mbit y 2 Kbit

Interfaces de comunicación:

- Ethernet con soporte PoE.
- USB On-The-Go.

³http://www.nxp.com/products/microcontrollers/cortex_m4/lpc4300/LPC4337JBD144.html

⁴http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf

- USB Device Auxiliar.
- RS232 (detalles técnicos).
- RS485.
- CAN (detalles técnicos).

Entradas/Salidas:

- 8 entradas digitales optoacopladas.
- 4 entradas analógicas configurables por jumper 0-10V o 0-20mA.
- 4 salidas open-drain de 24V, 1A.
- 4 salidas a relé 24V, 2A.
- 1 salida analógica configurable por jumper 0-10V o 0-20mA.
- Conectores de expansión LV-GPIO, SPI, I2C.

Capítulo 2

Introducción Específica

En este capítulo se detallan las tareas que se realizaron, que contratiempos se tuvieron y cuales fueron los requerimientos.

2.1. Objetivos y alcances

Objetivos

El objetivo de este trabajo es desarrollar un software que permita controlar, monitorear y supervisar el proceso de fermentación del vino en las bodegas en forma automática. Para eso se utilizó la plataforma CIAA ya que está preparada especialmente para aplicaciones industriales.

Alcance

- Sensar la temepratura de un tanque de fermentación de vinos.
- Controlar que la temperatura se mantenga dentro del rango correspondiente según el tipo de vino.
- En caso de corte de energía notificar vía SMS a un celular de contacto.
- Enviar mensajes de alerta vía SMS en caso de temperatura fuera de rango.
- Diseñar una página web para mostrar la información del sistema y realizar las configuraciones necesarias.
- Diseñar una placa básica para mostrar el funcionamiento del sistema.

El proyecto no include:

- Estudio de los sensores y actuadores, se basara dicha información en los datos dados por el cliente.
- Análisis del consumo eléctrico.
- Estudio del proceso de fermentación.
- Definición del rango de temperatura, esto será configurado por el cliente.

2.2. Requerimientos

Los requerimientos fueron consensuados con el responsable de la Bodega Chico Zossi, para trabajar sobre un caso real y son los siguientes:

1. Medición de temperatura:
 - 1.1. Se requiere medir la temperatura de un tanque.
 - 1.2. Se tiene que poder definir un rango entre 7 a 30°C con una amplitud mínima de 1°C.
2. Comunicación:
 - 2.1. Debe transmitir mensajes SMS notificando el estado y los cambios que se produzcan.
 - 2.2. Configuración de los parámetros requeridos de temperatura via web.
3. Funcionamiento del sistema embebido:
 - 3.1. El sistema deberá mantener la temperatura controlada a los parámetros configurados por el usuario. Para ello deberá hacer uso de actuadores.
 - 3.2. Deberá alertar si hay corte de energía mediante un SMS.
 - 3.3. Deberá mostrar el nivel de batería en porcentajes de 0 a 100 % con una resolución mínima del 20 % en la página web.

2.3. Planificación del proyecto

Para llevar a cabo el proyecto se desarrolló la planificación de tareas y se estimaron que tiempos debían emplearse para cada una de ellas. Para esto se aplicó lo aprendido en la asignatura de Gestión de Proyectos y se estableció el diagrama de Gantt que se presenta en la figura 2.1. A su vez se analizaron que tareas debían realizarse primero y cuales eran sus dependencias. De esta forma si se presentase alguna complicación se podría tener más claro cómo afectaría esto al proyecto.

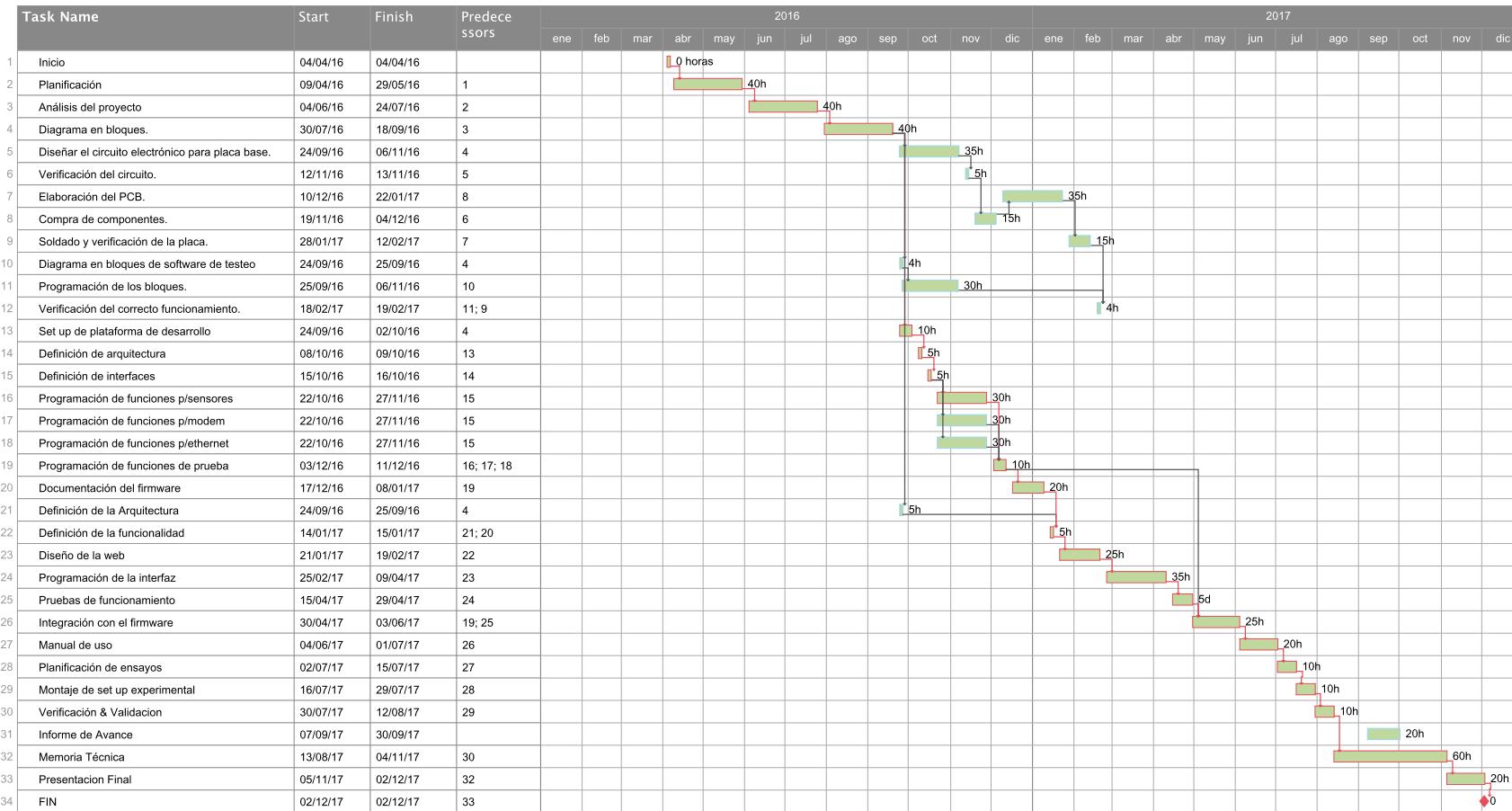


FIGURA 2.1: Listado de tareas, y plafinizacion de tiempos de la implementación.

Al desarrollar el presente proyecto resultó que dada a la falta de experiencia en determinados temas se requirió más tiempo de lo planificado. En la tabla 2.1, se aprecia el tiempo estimado, el tiempo realmente utilizado y cuales fueron las tareas que requirieron más tiempo.

| Nro | Tarea | Tiempo | | % adicional |
|-----|--------------------------------------|----------|------|-------------|
| | | Estimado | Real | |
| 9 | Soldado y verificación de la placa | 15h | 24h | 60 |
| 18 | Programación de funciones p/ethernet | 30h | 50h | 66,7 |
| 23 | Diseño de la web | 25h | 35h | 40 |

TABLA 2.1: Comparación tiempo planificado vs real.

Se observa una diferencia de 39 horas respecto a lo planificado. Siendo que para el desarrollo de todo el proyecto se habían estimado 617 horas se puede concluir que la estimación de los tiempos fue adecuada.

Capítulo 3

Diseño e Implementación

En este capítulo se presenta cómo se desarrolló el hardware básico para simular el entorno de funcionamiento y cómo fue implementado el firmware.

3.1. Hardware

Primeramente se diseñó un circuito electrónico que permite simular el entorno de funcionamiento de una bodega. Para ello se utilizó lo aprendido a lo largo del curso de *diseño de PCBs en KICAD*.

En la figura 3.1 se aprecia el entorno de trabajo del sistema. El mismo interactúa con una bomba de agua que permite llevar líquido refrigerante desde el depósito hacia los tanques que están en proceso de fermentación. Recibiendo como información la temperatura de los tanques, el equipo debe controlar la bomba y la electroválvula correspondiente. De esta forma se controla la temperatura acorde a los parámetros definidos por el usuario.

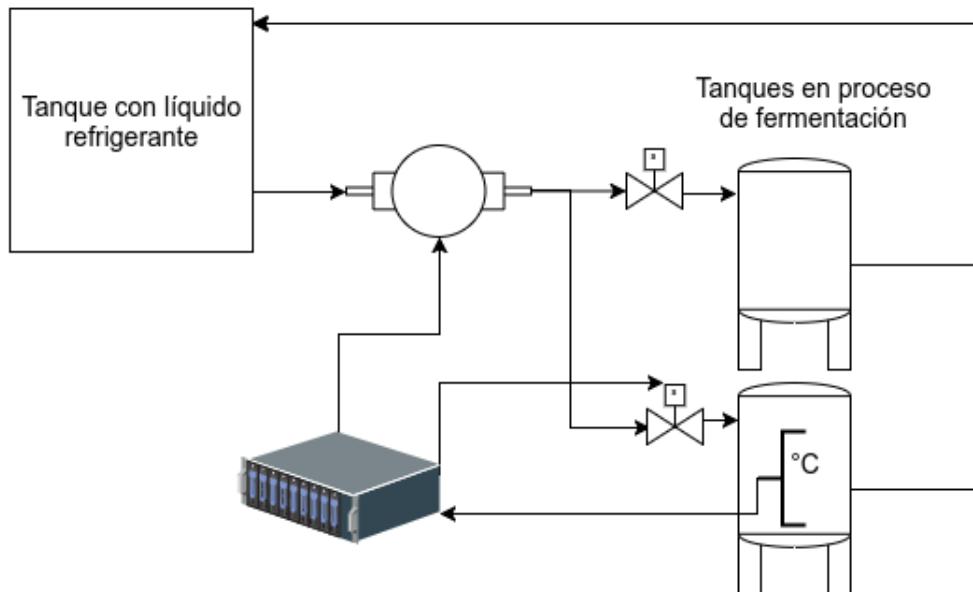


FIGURA 3.1: Entorno del sistema de control del equipo.

3.1.1. Módulos requeridos

Para llevar a cabo el proyecto, se dividió el sistema en cuatro módulos:

Módem: encargado de transmitir las alertas mediante SMS utilizando el modulo SIM800L.

Actuadores: salidas que permiten controlar la bomba y electroválvulas.

Sensores: entradas que permitan simular el comportamiento de los sensores.

CPU: unidad encargada de controlar el funcionamiento del sistema incorporada en la placa CIAA-NXP.

3.1.2. Características del hardware

En la figura 3.1 se mostró que el dispositivo de monitoreo interactúa con la temperatura de los tanques, el control de la bomba y la selección de la electroválvula. Es por esto que se implementó un sistema que está compuesto por el módem y circuitos electrónicos que permiten:

- Simular el estado de la temperatura
- Adaptar los niveles de tensión.
- Mostrar el estado de los actuadores.
- Conectar el módem GPRS.

De esta forma se desarrolló una placa que se denomina "simulador del sistema" y permite conectar el módem y validar el correcto funcionamiento del software implementado. La forma de conexión con la CIAA-NXP se muestra en la figura 3.2.

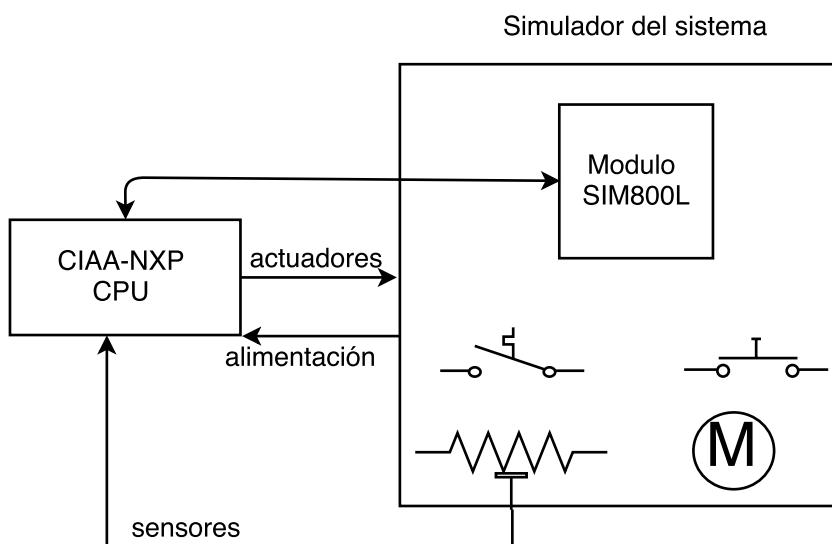


FIGURA 3.2: Conexión de la CIAA-NXP con el simulador del sistema.

La placa de simulación debe contar con las siguientes entradas, salidas y zócalos:

- 1 Salida con regulación de corriente, para simular sensores de 4mA a 20mA.
- 3 Salida de tensión variable.
- 1 Salida de sensor de temperatura LM35.
- 4 Entradas digitales conectadas a leds.
- 4 Salidas conectadas a pulsadores.
- 1 Zócalo para la conexión del modulo SIM800L.
- 1 salida de 5V a 3A y otra de 3.3V a 1A.

El módulo SIM800L se muestra en la figura 3.3 y es un módem GPRS que cuenta con las siguientes especificaciones:



FIGURA 3.3: Módem SIM800L.

- Alimentación: 3.4V a 4.4V (4.0V recomendado)
- CuatriBanda 850/900/1800/1900MHz
- GPRS Multi Slot class 8/10
- Control mediante comandos AT (GSM 07.07 ,07.05 y comandos AT SIM-COM).

En la figura 3.4 se aprecia el esquemático implementado para la conexión del módem GSM. Éste requiere adaptar los niveles de tensión para interactuar con la CIAA-NXP, para el cual se utilizó el max3232.

Para la alimentación del circuito se utilizó el regulador LM2576. El mismo se tomó de la nota de aplicación [2]. Y para regular a 3.3V se utilizó el integrado LM11733 circuito extraído de la nota de aplicación [1]. El esquemático se aprecia en la figura: 3.5.

Para lograr la simulación de distintos tipos de sensores, se utilizaron circuitos como el que se muestra en la figura 3.6 que permiten generar corrientes de 4mA a 20mA. También reguladores de tensión, véase en la figura 3.7 y un sensor de temperatura en la figura 3.8. Para estos circuitos se utilizaron las notas de aplicación de Texas Instruments[3] - capítulo 8 páginas 12 y 13 y sensor temperatura: [4] - 8.2.1 Basic Centigrade Temperature Sensor.

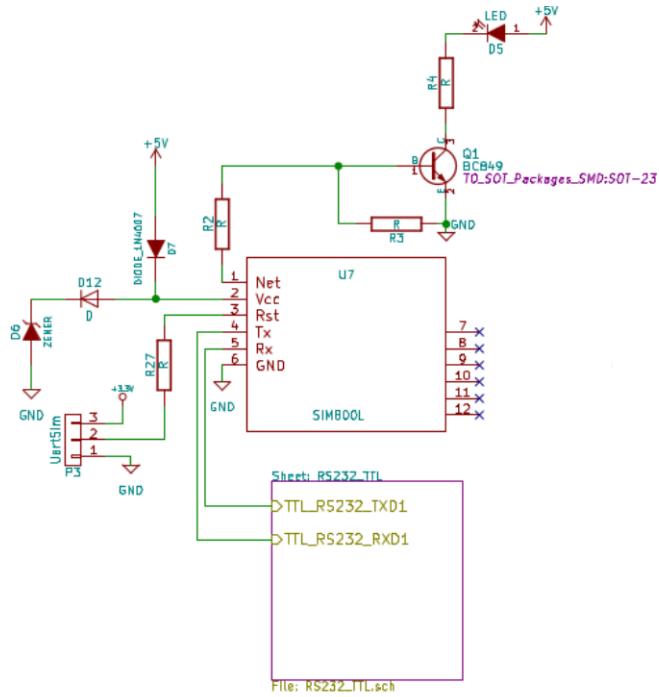


FIGURA 3.4: Circuito de conexión del módulo SIM800l.

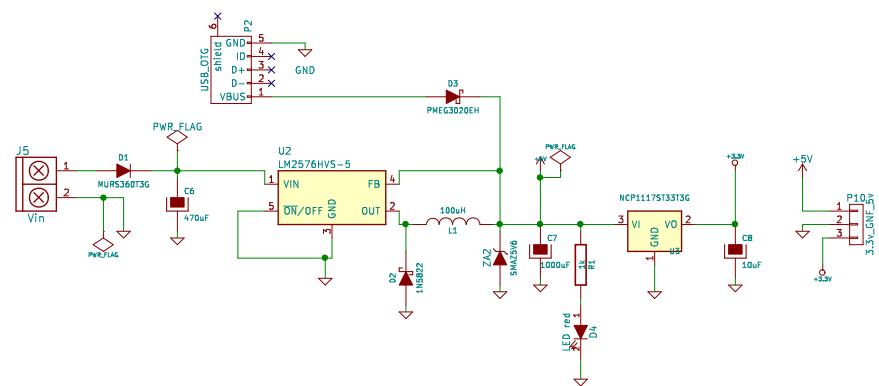


FIGURA 3.5: Fuente de alimentación DC/DC.

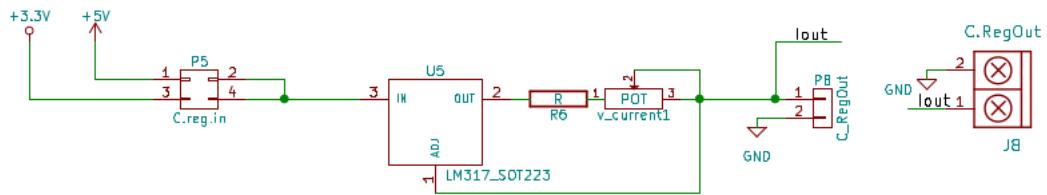


FIGURA 3.6: Circuito regulador de corriente.

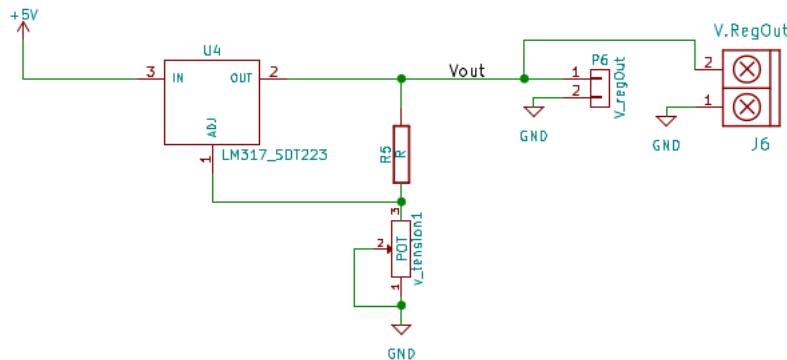


FIGURA 3.7: Circuito regulador de tensión.

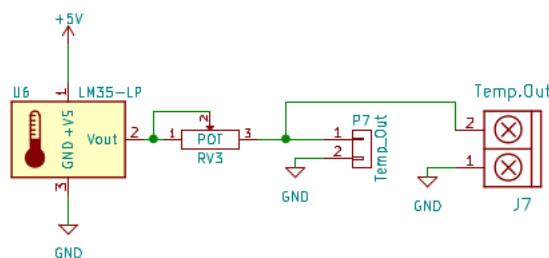


FIGURA 3.8: Circuito del sensor de temperatura.

Para concluir, se utilizaron circuitos básicos que permiten interactuar en forma simple con el sistema y realizar pruebas de funcionamiento. Los mismos fueron extraídos del esquemático utilizado en el proyecto de la EDU-CIAA¹, figura 3.9.

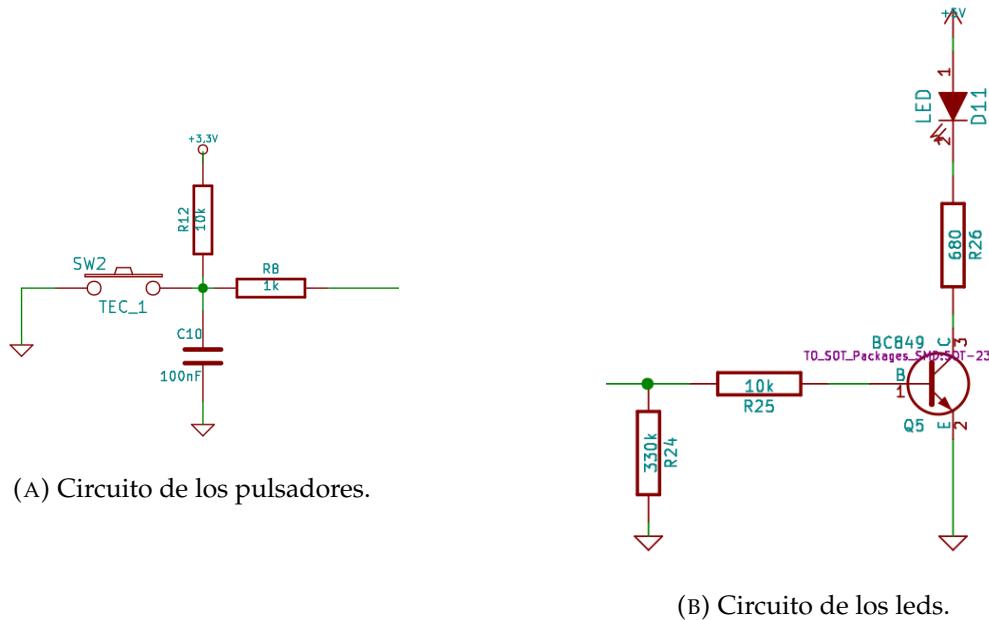


FIGURA 3.9: Esquemáticos extraídos de la EDU-CIAA.

Con los esquemáticos terminados, se pasó a la elaboración del circuito impreso que se presenta en las figuras 3.10, 3.11 y 3.12.

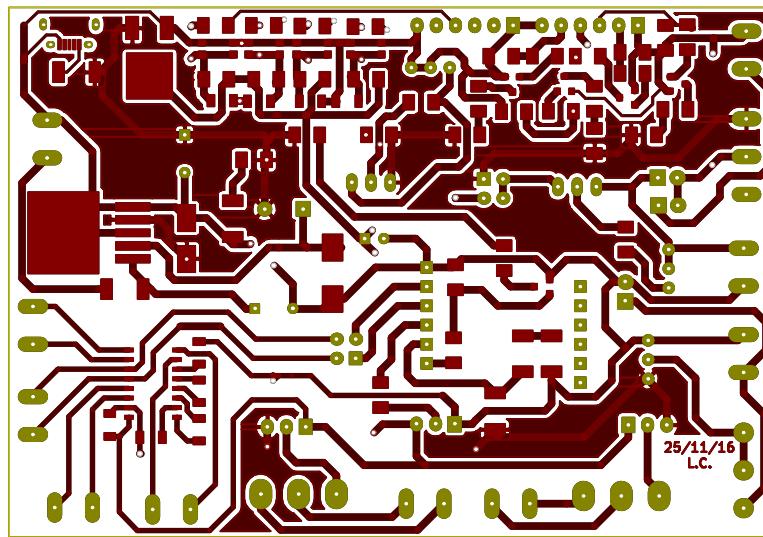


FIGURA 3.10: Capa superior de la placa de simulación de sensores.

¹http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp:edu-ciaa-nxp_color.pdf

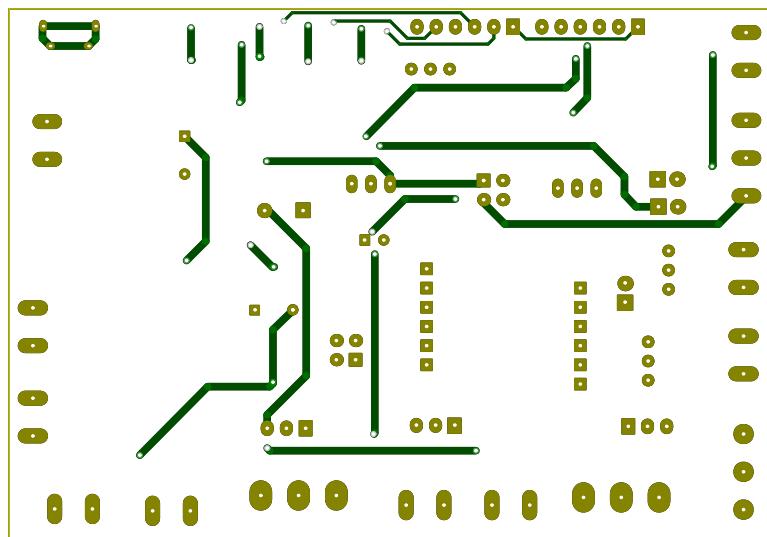


FIGURA 3.11: Capa inferior de la placa de simulación de sensores.

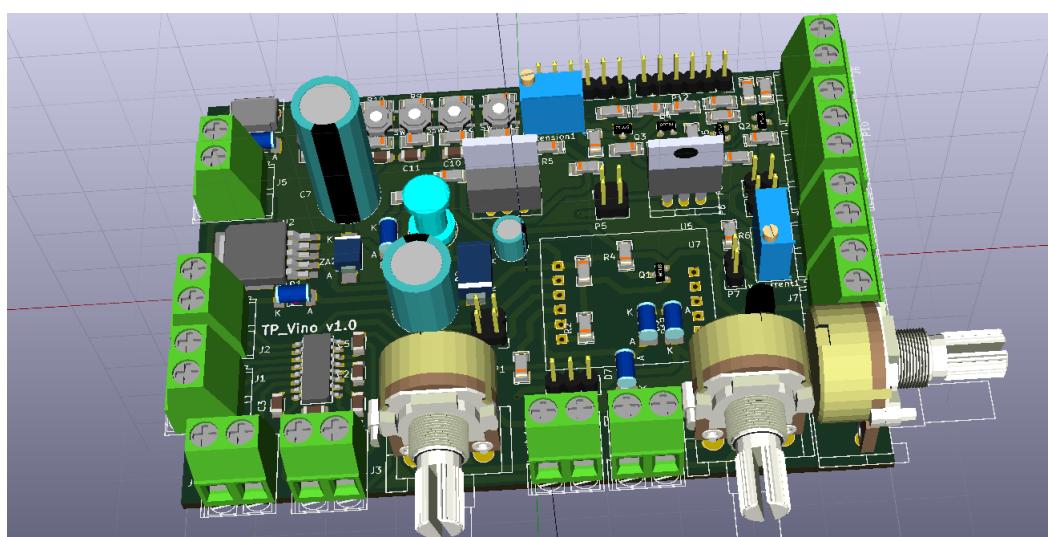


FIGURA 3.12: Placa simulador de sensores 3D.

No obstante, por cuestiones de tiempo no se llegó a implementar la placa que se diseñó del simulador del sistema y se terminó desarrollando una placa experimental que se presenta en la figura 3.13, la cual permitió realizar las primeras pruebas.



FIGURA 3.13: Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado.

3.2. Software

3.2.1. Aplicación

Para la implementación de la lógica del sistema se dividieron en tres tareas: módem, control y sensores.

Sensor

Esta tarea consulta el estado de los sensores y calcula el promedio de las últimas N muestras. Este es el valor que utiliza la tarea de control para definir si es necesario encender los actuadores correspondientes o apagarlos.

El diagrama de flujo de esta tarea se puede apreciar en la figura 3.14.

Módem

En esta tarea se inicializa las configuraciones iniciales para el uso del módem. El cual requiere que esté configurado en modo texto para poder transmitir mensajes SMS. A su vez, es la encargada de interactuar con el módulo GPRS realizando consultas del nivel de señal del módem y de enviar los mensajes de alarmas si estas están activadas y son necesarias.

En el diagrama de flujo de esta tarea se aprecia en la figura 3.15.

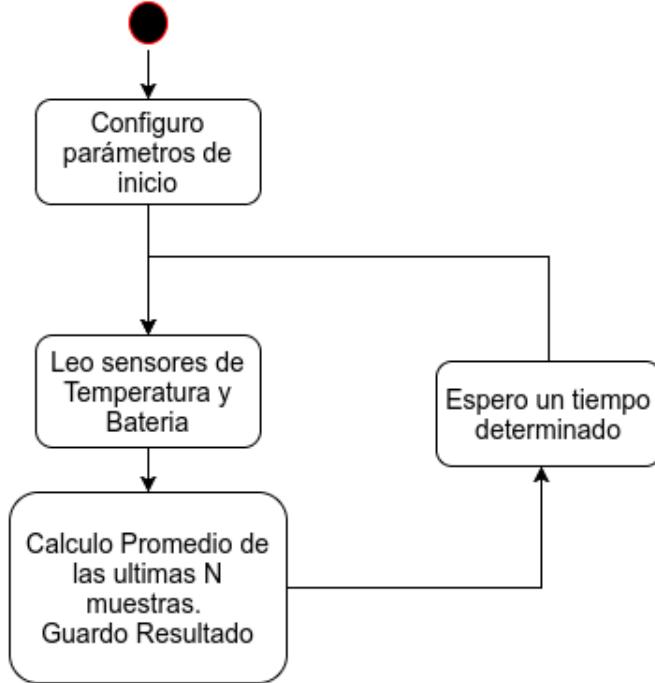


FIGURA 3.14: Diagrama de flujo de la tarea sensor.

Control

La tarea de control es la que:

- Analiza los resultado de los sensores de temperatura y de la batería verificando que estén dentro de los parámetros configurados por el usuario y actúa en consecuencia. Si la temperatura del sistema excede la máxima configurada y el control automático esta activado entonces esta procede a encender la bomba y activar la electroválvula para que circule el líquido refrigerante por el tanque correspondiente. Luego una vez que la temperatura baja hasta la mínima, este deshabilita la bomba y la electroválvula. De no estar activado el control automático, la bomba y la electroválvula continúan en el estado que estaban, es decir no se realiza ninguna acción.
- Es la encargada de setear los actuadores según sean especificados en la plataforma web.

El diagrama de flujo correspondiente a esta tarea se puede apreciar en la figura 3.16.

3.2.2. Plataforma web

Mediante la plataforma web diseñada se busca que el usuario pueda ver los datos de los sensores, tener control sobre los actuadores y realizar las configuraciones necesarias.

Para poder realizarlo fue necesario profundizar los siguientes temas:

HTML: la página web se desarrolló en este lenguaje, que permite darle la estructura básica y se realiza sobre texto plano, es decir no se requiere ninguna

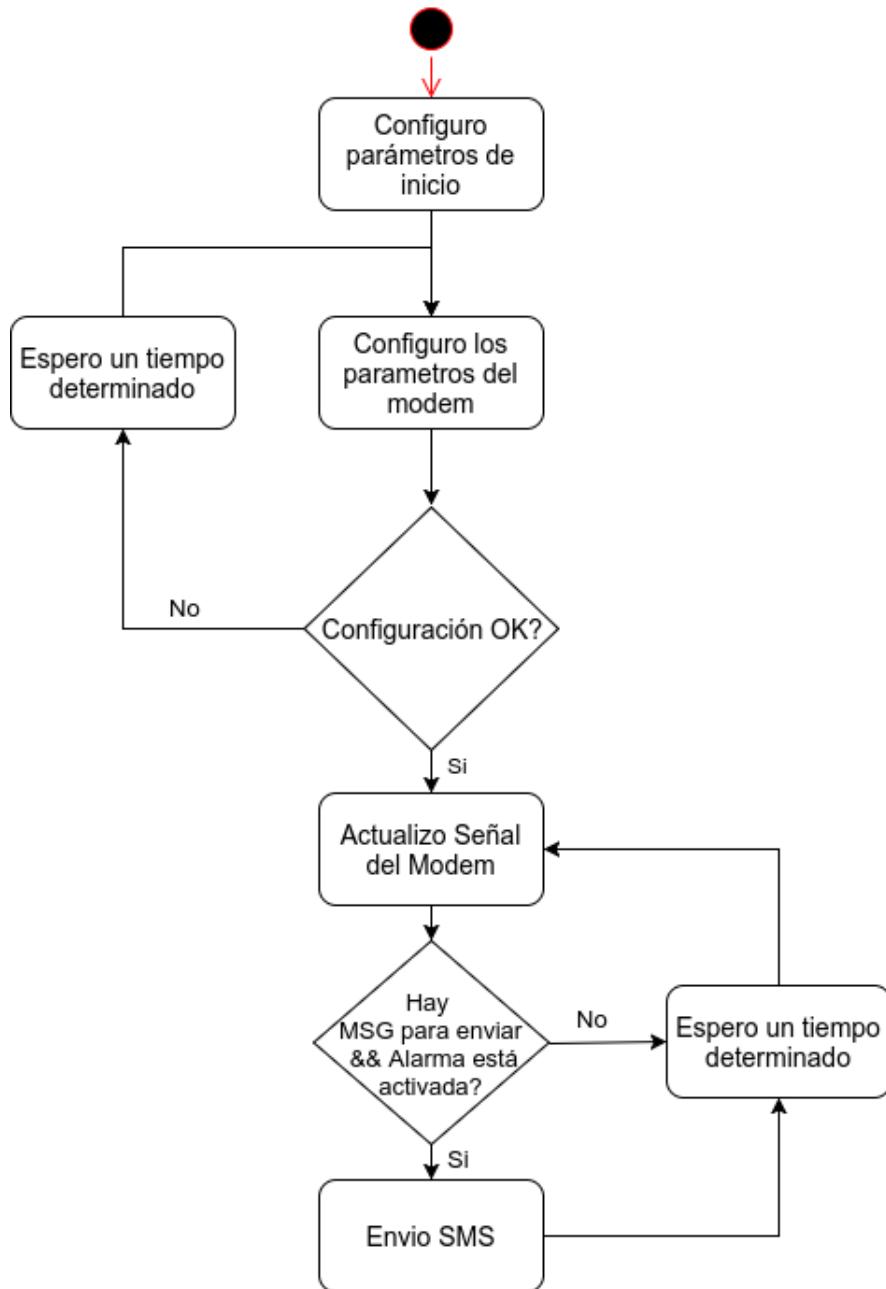


FIGURA 3.15: Diagrama en flujo de la tarea módem

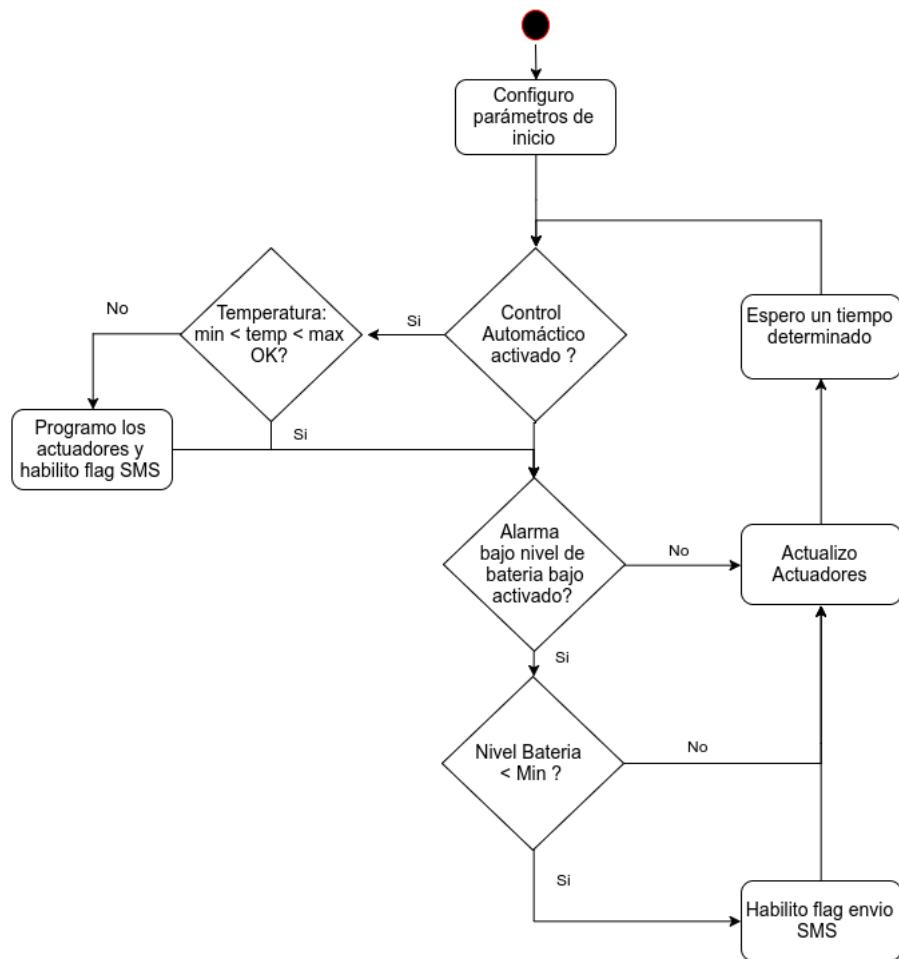


FIGURA 3.16: Diagrama de flujo de la tarea control.

aplicación adicional simplemente un editor de texto. Luego es el navegador web quien tiene la tarea de interpretarlo.

CSS: mediante este lenguaje se realiza la presentación de la página, permitiendo así modificar la visualización de la misma y darle un formato más profesional y elegante.

SVG: mediante este lenguaje se implementaron los gráficos correspondientes para mostrar los resultados de los sensores y el estado de los actuadores.

AJAX: es un recurso muy utilizado cuando se desarrollan web en sistemas embedidos. Permite actualizar el estado de la web sin tener que consultar todo el contenido, simplemente se descarga una primera vez y luego se realizan consultas de los valores que quieren actualizar.

SSI: la implementación que ofrece FreeRTOS es reducida pero más que suficiente para poder agregar contenido a la página en forma dinámica.

CGI: es un mecanismo de comunicación entre el servidor web y una aplicación. Permite al cliente solicitar y enviar datos al servidor. Con esta herramienta es que se obtiene el estado de los sensores y actuadores.

Diseño web

Se buscó información en internet, donde se encontraron varias páginas con cursos de HTML, CSS, y AJAX que son de forma online y gratuita. Las que más se utilizaron fueron:

- <https://www.w3schools.com>
- <https://es.khanacademy.org>
- <https://www.codecademy.com>

Así se logró una primera versión de la web, como se puede ver en la figura 3.17.



FIGURA 3.17: Primera versión de la web de monitoreo.

Cuando ya estaba funcionando en forma correcta, se pasó a mejorar la presentación de la misma. Para ello se usó un template que se ajustara a las necesidades,

es decir mediante un archivo que contiene un estilo ya predefinido que modifica la apariencia de la web. La versión final se muestra en la figura 3.18.

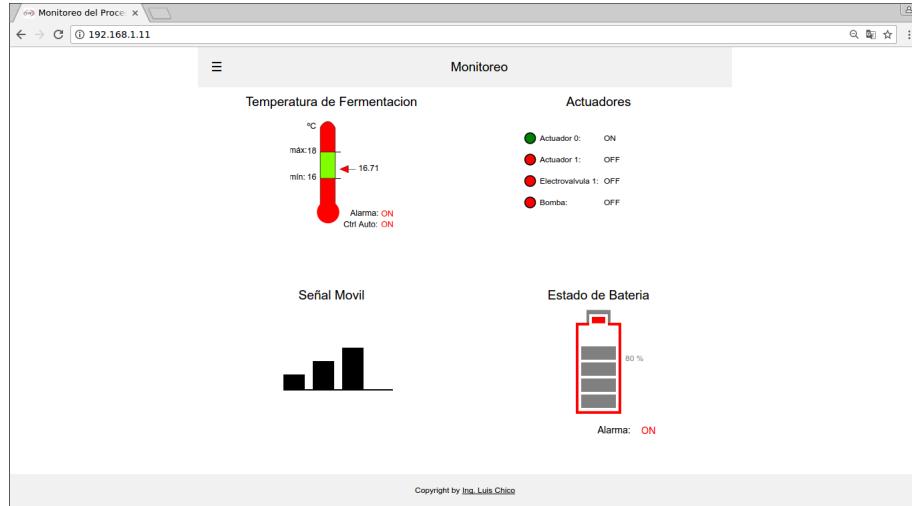


FIGURA 3.18: Versión actualizada haciendo uso del template.

El template que se vio en la versión final, fue encontrando en w3school² y cumplía los siguientes requerimientos:

- Sea amigable.
- Compatible con distintos navegadores, especialmente los más utilizados como ser: Google-Chrome y Firefox.
- Ser cómodo de navegar en distintos dispositivos, como tablets, celulares y PCs.

3.2.3. Configuración del servidor

Al iniciar el servidor se debe tener en cuenta que este será el encargado de gestionar las peticiones del usuario. Es decir una vez que el usuario ingresa la dirección del servidor, este le transmite la información de la web. Para realizar la comunicación entre cliente y servidor se utilizó la API de lwIP [6] que utiliza una serie de callbacks para controlar los eventos debido a la comunicación de la red. Es por ello que para optimizar los recursos se utilizó AJAX, permitiendo que el cliente sea el encargado de realizar las consultas al server en forma periódica.

La configuración del servidor establecida por default es la que se muestra en la Tabla 3.1.

La información se solicita mediante directivas del tipo SSI. Es decir, solo se transmite un archivo ajax.shtml con los parámetros que se van a actualizar. Luego el navegador, mediante una serie de funciones en JavaScript es el encargado de actualizar los cambios realizados. Se puede ver el procedimiento mencionado en la figura 3.19.

²https://www.w3schools.com/w3css/w3css_templates.asp

| Parámetro | Valor |
|------------------|---------------|
| IP | 192.168.1.11 |
| Mascara de red | 255.255.255.0 |
| Puerta de enlace | 192.168.0.1 |
| Puerto | 80 |
| DHCP | Deshabilitado |

TABLA 3.1: Configuración por default del servidor.

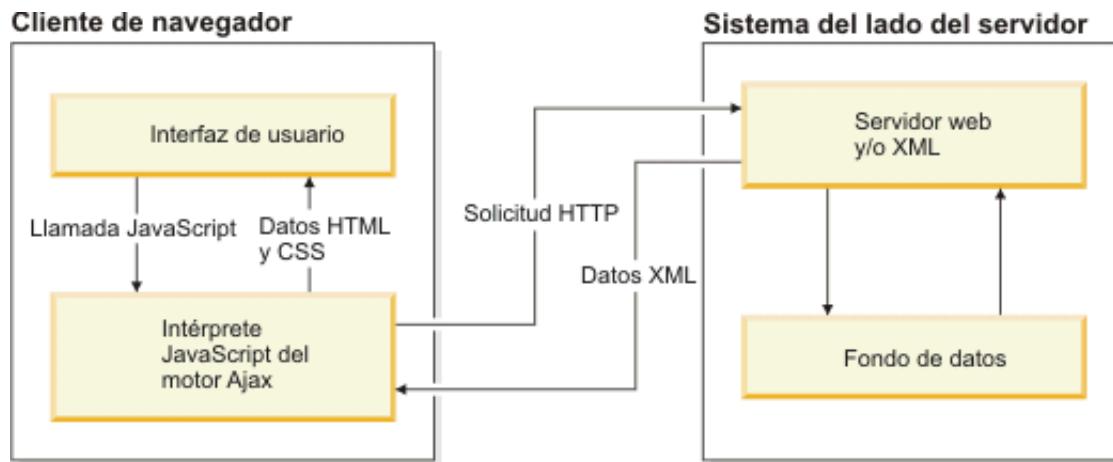


FIGURA 3.19: Diagrama de secuencia AJAX.

3.3. Diseño del sistema web embebido

El menú implementado se muestra en la figura 3.20:

Al presionar donde dice menú, se puede configurar lo siguiente:

1. **Monitoreo:** dirige a la pantalla principal, donde se muestra el estado del sistema. Esto se muestra en la figura 4.4.
2. **Actuadores:** setear en forma manual el estado de los mismos. En la figura 3.21 se presenta esto en mayor detalle.
3. **Batería:** aquí se podrá setear la alarma debido al nivel de descarga de la batería. Esta enviará un SMS indicando que se alcanzó dicho estado. En la figura 3.22 se muestra un ejemplo.
4. **Módem GPRS:** en este menú podremos configurar 2 personas a quienes serán enviadas las alertas debido al accionar de alguna alarma. En la figura 3.23 se presenta esto en mayor detalle.
5. **Temperatura:** aquí se configura el rango de temperatura, la alarma correspondiente y si se activa el control automático de la misma. De estar activado, actuará en forma automática sobre la bomba y la electroválvula para mantener la temperatura dentro de dicho rango. Para apreciar esto con mayor detalle, vea la figura 3.24.
6. **Network:** aquí se configura los parámetros de la red, IP, mascara de red y puerta de enlace. En la figura 3.25 se ve el formulario con los campos mencionados que debemos completar.

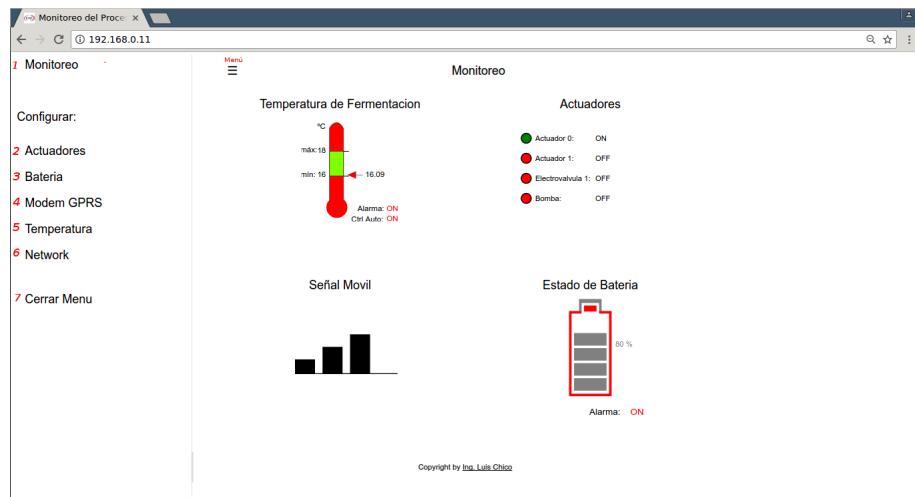


FIGURA 3.20: Opciones del al presionar sobre el menú desplegable.

7. Cerrar Menú: esta opción cierra el menú, permitiendo quitar las opciones de menú.



FIGURA 3.21: Activar/Desactivar actuadores.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
 Monitoreo del Proceso X

192.168.1.11

Bateria

| Valor | | Estado de Bateria |
|--|---|-------------------|
| Mínimo | <input type="text"/> | Máximo |
| Alarma | <input checked="" type="radio"/> ON <input type="radio"/> OFF | |
| <input type="button" value="Submit"/> | | |
| De estar activada la alarma recibirá un SMS | | |
|  Alarma: ON | | |

Copyright by [Ing. Luis Chico](#)

FIGURA 3.22: Configurar nivel de descarga para accionar la alerta por batería baja.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
 Monitoreo del Proceso X

192.168.0.11

Configurar Modem GPRS

| Señal Movil | | Configurar telefonos para reportar alarmas SMS | | |
|---|--|--|-----------------------------------|--|
| | | Nombre | Nro de Celular | Estado |
|  | | Opción_1: <input type="text"/> Test 1 | <input type="text"/> 64958758 | <input checked="" type="radio"/> Activo <input type="radio"/> Inactivo |
| | | Opción_2: <input type="text"/> Person_2 | <input type="text"/> Nro Cel2. | <input checked="" type="radio"/> Activo <input type="radio"/> Inactivo |
| <input type="button" value="Submit"/> | | | | |

Copyright by [Ing. Luis Chico](#)

FIGURA 3.23: Configuración de los celulares.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
 Monitoreo del Proce: x 192.168.0.11

Configuración del control Temperatura

Temperatura de Fermentacion

| Temperatura | Valor |
|--------------------|---|
| Máximo | <input type="text"/> |
| Mínimo | <input type="text"/> |
| Alarma* | ON <input type="radio"/> OFF <input checked="" type="radio"/> |
| Control Automático | ON <input type="radio"/> OFF <input checked="" type="radio"/> |

Configure los valores máximos y mínimos para activar el control automático.
 * Si está activada la alarma, será notificado mediante un SMS.

Submit

Copyright by [Ing. Luis Chico](#)

FIGURA 3.24: Configuración del rango de temperatura y activación del estado de control automático y alarma.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
 Monitoreo del Proce: x 192.168.0.11

Configurar Network

IP:

Mask:

Gateway:

Submit

Copyright by [Ing. Luis Chico](#)

FIGURA 3.25: Configuración de IP, mascara de red y puerta de enlace.

Capítulo 4

Ensayos y Resultados

En este capítulo se expone cuales fueron las pruebas realizadas para determinar que el sistema funciona en forma correcta.

4.1. Banco de pruebas

Para simular el comportamiento de un sensor de temperatura y la descarga de una batería, se implementó la placa que fue presentada en la sección 3.1.2, que para facilidad del lector se vuelve a mostrar en la figura 4.1. Esta consta de dos potenciómetros que permiten simular la variación de la temperatura y la carga de la batería. Sobre la placa se conecto un conversor RS232 a TTL, para lograr la integración con el módem.

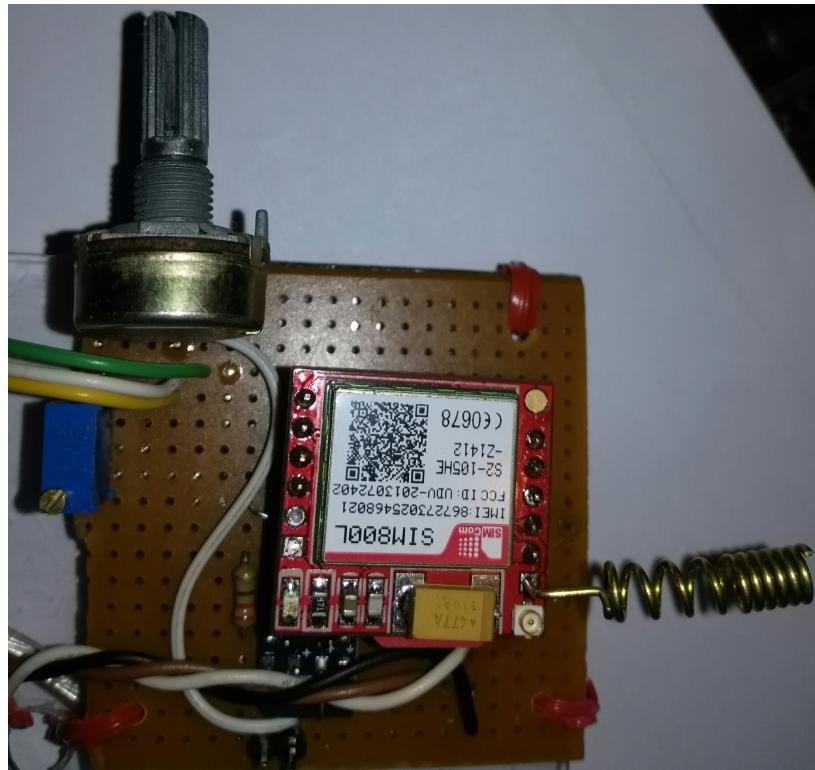


FIGURA 4.1: Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.

En la figura 4.2 se muestra la placa que contiene el módem y los componentes de simulación conectada a la CIAA-NXP.

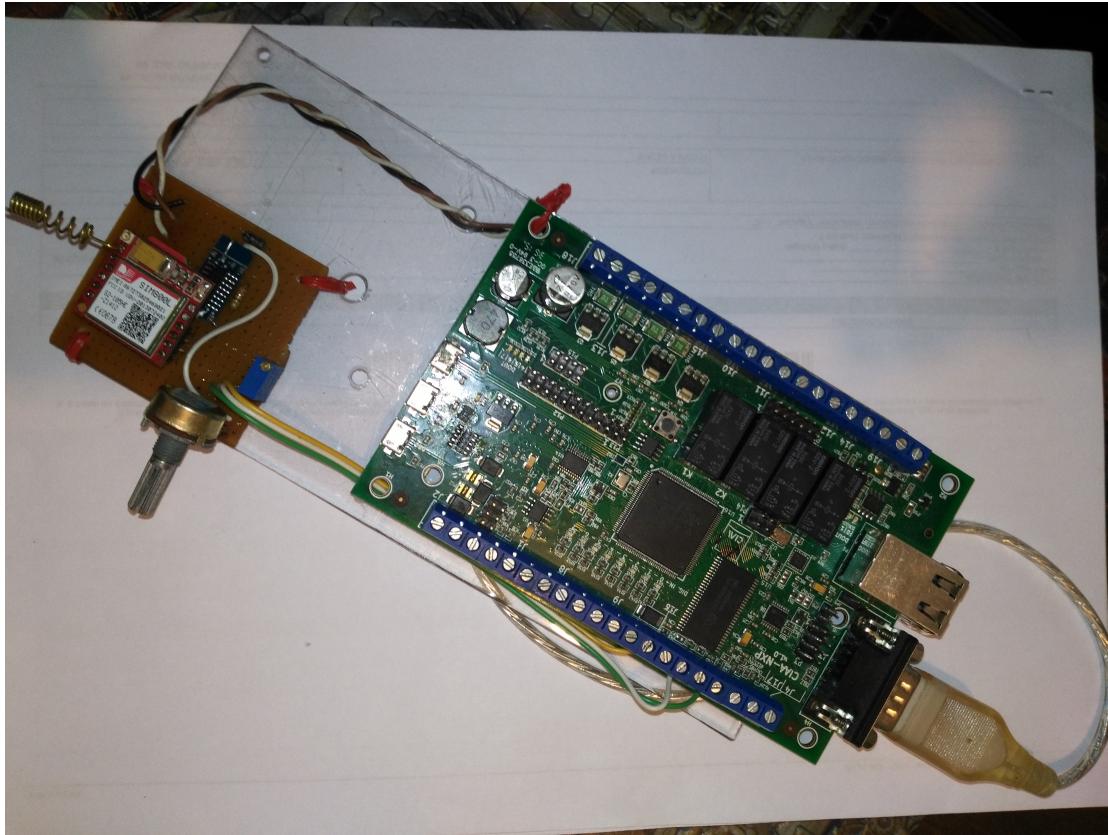


FIGURA 4.2: Placa básica integrada con la CIAA-NXP.

4.2. Configuraciones para la PC

Para la realización de los ensayos, se configuró la PC en forma adecuada. En la figura 4.3 se puede ver cómo configurar en Linux la interfaz de red. Es importante destacar que dichos comandos son ejecutados con el permiso de usuario *root*.

Una vez que se tiene la configuración de red adecuada se obtiene en el navegador web la imagen 4.4 que fue introducida en el capítulo 3.

En esta pantalla se tiene toda la información correspondiente al sistema. Arriba a la izquierda se ve el estado de la temperatura, con sus rangos máximos y mínimos seteados. Y debajo de dicho termómetro podemos ver si esta activada: la alarma y el control automático.

Luego arriba y a la derecha los estados de los actuadores, la electroválvula y la bomba. Estos le permiten al cliente controlar en forma remota dos actuadores y el sistema correspondiente a la bomba de refrigeración con la electroválvula asociada. El sensor de temperatura permitirá informarle al sistema de control automático cuando debe actuar acorde a los rangos establecidos, esto solo en el caso de estar activado dicho control automático.

```

root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:26202 errors:0 dropped:0 overruns:0 frame:0
              TX packets:26202 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1
              RX bytes:1987773 (1.9 MB)  TX bytes:1987773 (1.9 MB)

root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig eth0 192.168.1.2/24
root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
192.168.1.0     *             255.255.255.0   U     0      0        0 eth0
root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig
eth0      Link encap:Ethernet HWaddr a4:ba:db:da:b5:30
        inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::a6ba:dbff:fedab530/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
              RX packets:3969 errors:0 dropped:0 overruns:0 frame:0
              TX packets:6035 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:1460102 (1.4 MB)  TX bytes:765043 (765.0 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:26262 errors:0 dropped:0 overruns:0 frame:0
              TX packets:26262 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1
              RX bytes:1992384 (1.9 MB)  TX bytes:1992384 (1.9 MB)

root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# 

```

FIGURA 4.3: Configuración de una red estática para la interfaz Ethernet.

Abajo a la izquierda el nivel de señal correspondiente al módem GSM, mediante el cual podremos saber si la cobertura de señal de la red de telefonía móvil esta en condiciones de operar y entonces estaría disponible el servicio de alertas mediante mensajes SMS.

Finalmente abajo a la derecha el estado de la batería, el cual va a permitir al sistema continuar en funcionamiento en caso de un corte de energía. El principal uso de este será mantener la posibilidad de enviar un mensaje SMS notificando el corte de energía.

4.3. Pruebas funcionales del hardware

Una vez recorrido todas las opciones que permite la interfaz web y verificar que el comportamiento haya sido el esperado. Se procede interactuar con el sistema, por medio de alteraciones generadas a través de los potenciómetros que representaran la temperatura y el estado de la batería.

Estando activado el control automático y al simular un incremento de temperatura, se comprobó que al exceder la temperatura máxima inicio en forma automática la bomba que permite la circulación del refrigerante y la activación de la electroválvula que interviene sobre el tanque que se está monitoreando. En el navegador web se verá como en la figura 4.5. y en la figura 4.6 se puede observar la alerta correspondiente mediante un SMS.

Se verificó que el estado de la bomba y la electroválvula se mantiene activado hasta llegar a la temperatura mínima, en este instante se procede a desconectar la

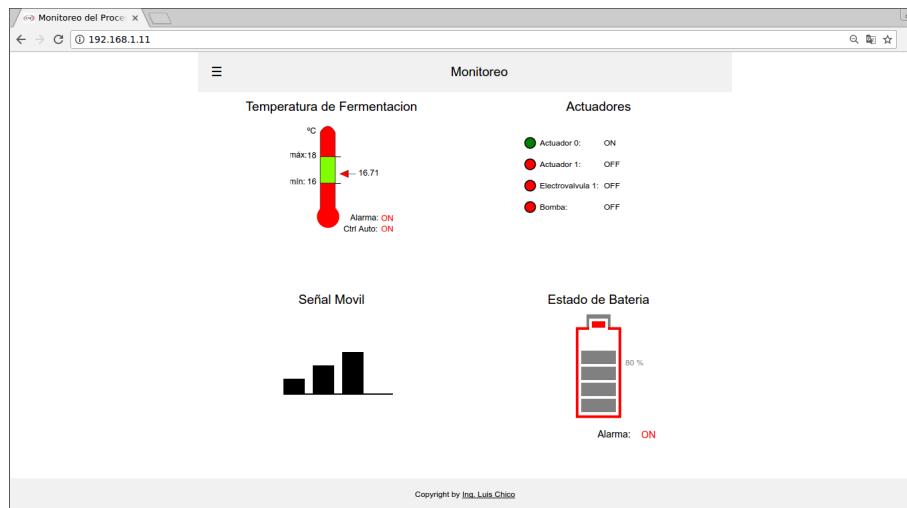


FIGURA 4.4: Pantalla principal de la web de monitoreo.

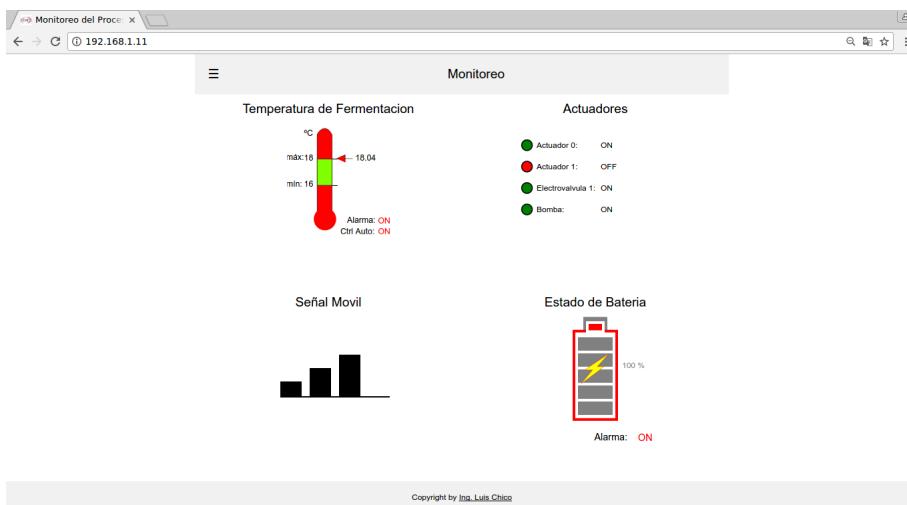


FIGURA 4.5: Activación por control automático debido a temperatura elevada.

bomba y a cerrar la electroválvula. En la web se mostrará como en la figura 4.7. y dado su respectiva alerta mediante un SMS como en la figura 4.6

En el caso de la batería se analiza las variaciones en la carga. Si esta disminuye más de un valor que es determinado, se envía un SMS como en la figura 4.6. Es importante aclarar que en caso de producirse un corte de energía, el sistema no está pensado para que se mantenga en funcionamiento por un tiempo prolongado, sino más bien lo que se busca es que este pueda mantener informado a la personas responsables que se requiere de su atención. Es por esto que no se realizó un estudio del tiempo de vida del sistema.

Para verificar los actuadores, se tiene la sección de configuración de actuadores y para cada uno de ellos se verificó los diferentes estados, encendiendo y apagando. Luego se probó en bloques de varios actuadores. En la figura 4.8 se ve un ejemplo de los casos probados.

De esta forma concluyó que el funcionamiento del sistema es el correcto.

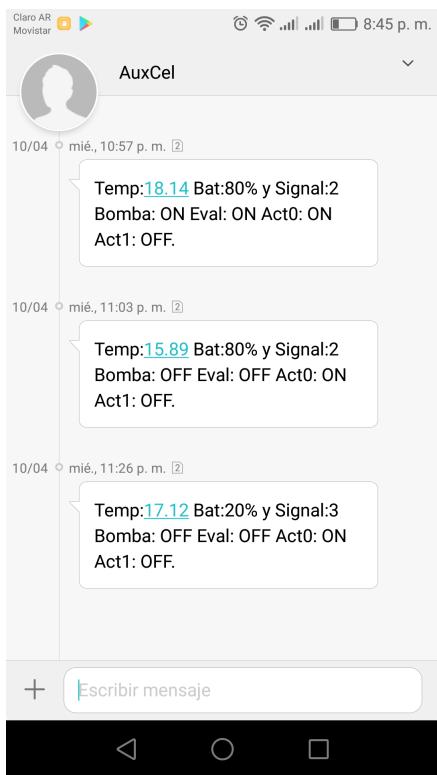


FIGURA 4.6: SMS recibidos debido a temperatura máxima.
El segundo SMS recibido es por temperatura mínima.
Y el último recibido debido a batería baja.

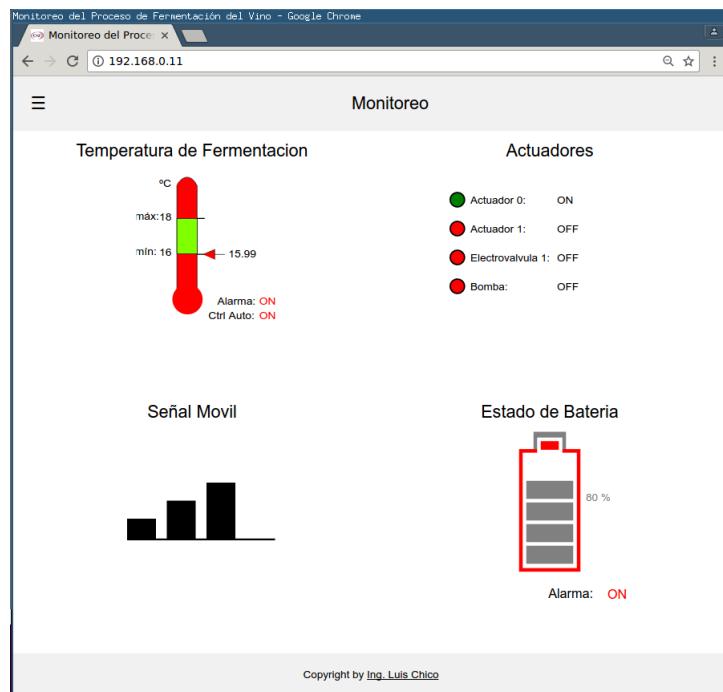


FIGURA 4.7: Temperatura en nivel mínimo, se desactiva la bomba
y la electroválvula.



FIGURA 4.8: Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

Al finalizar este trabajo se logró:

- Acceder mediante una página web al estado del sistema.
- Navegación mediante pc, celulares, tablets compatibles con navegadores como Chrome y Firefox.
- Realizar un control manual y automático del sistema, mediante el uso de un sensor de temperatura y salidas para actuadores.
- Alertar mediante SMS cuando la temperatura excede el rango permitido, y/o para un determinado nivel de batería.
- Implementar mediante el stack TCP/IP de lwIP mediante el RTOS freeRTOS.

Es decir, controlar en forma remota la temperatura de un tanque y recibir alertas en los principales estados críticos del proceso.

5.2. Próximos pasos

Extendiendo la funcionalidad del sistema se pueden optimizar los siguientes puntos:

- Ampliar la capacidad de control a cuatro tanques.
- Mejorar la interfaz web para dichos cambios, y realizar un entorno más amigable.
- Ampliar las funcionalidades de los SMS, no sólo para reportes de alertas sino también para consultas y posibles controles.
- Agregar retroalimentación de sensores y actuadores por hardware para un control más robusto.
- Adicionar una opción de control por intervalos de tiempos.

Bibliografía

- [1] *LM1117 800-mA Low-Dropout Linear Regulator*. Capítulo 8 Figura 16. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm1117.pdf>.
- [2] *LM2576xx Series SIMPLE SWITCHER® 3-A Step-Down Voltage Regulator*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm2576.pdf>.
- [3] *LM317 3-Terminal Adjustable Regulator*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm317.pdf>.
- [4] *LM35 Precision Centigrade Temperature Sensors*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm35.pdf>.
- [5] Proyecto CIAA. *Computadora Industrial Abierta Argentina*. Disponible: 2016-06-25. 2014. URL: <http://proyecto-ciaa.com.ar/devwiki/doku.php?id=start>.
- [6] Savannah. *lwIP - Wiki - Sample Web Server*. Disponible: 2016-06-25. URL: http://lwip.wikia.com/wiki/Sample_Web_Server.