

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. Lic. Agustín Bassi (FIUBA)
Esp. Ing. Patricio Bos (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre enero
de 2016 y agosto 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

Agradecimientos personales. [OPCIONAL]

No olvidarse de agradecer al tutor.

Índice general

Resumen	III
1. Introducción General	1
1.1. Introducción	1
1.2. Producción de vino	1
1.3. Motivación	1
1.4. Proyecto CIAA	1
1.5. Objetivos y Alcances	1
2. Introducción Específica	3
2.1. Objetivos y alcances	3
2.2. Requerimientos	3
2.3. Desglose de tareas - GANTT	4
3. Diseño e Implementación	7
3.1. Hardware	7
3.1.1. Módulos requeridos	8
3.1.2. Características del hardware	8
3.2. Software	14
3.2.1. Aplicación	14
3.2.2. Web	17
3.3. Diseño del sistema web embebido	20
4. Ensayos y Resultados	25
4.1. Banco de pruebas	25
4.2. Configuraciones para la PC	26
4.3. Pruebas funcionales del hardware	27
5. Conclusiones	31
5.1. Conclusiones generales	31
5.2. Próximos pasos	31
Bibliografía	33

Índice de figuras

2.1. Listado de tareas, y planificación de tiempos de la implementación.	5
3.1. Entorno del sistema de control del equipo.	7
3.2. Entorno del sistema de control del equipo.	8
3.3. Módem SIM800L.	9
3.4. Módulo SIM800L e índice de esquemáticos.	10
3.5. Fuente de alimentación DC/DC.	10
3.6. Regulador de corriente, de tensión y del sensor de temperatura.	11
3.7. Esquemáticos extraídos de la EDU-CIAA.	11
3.8. Capa superior de la placa simulador de sensores.	12
3.9. Capa inferior de la placa simulador de sensores.	12
3.10. Placa simulador de sensores 3D.	13
3.11. Placa de simulación de temperatura y estado de la batería con el módulo SIM800L integrado.	14
3.12. Diagrama de flujo de la tarea que controla el módem GPRS.	15
3.13. Diagrama de flujo de la tarea de control.	16
3.14. Diagrama de flujo del sensor.	17
3.15. Primera versión de la web de monitoreo.	18
3.16. Versión actualizada haciendo uso del template.	19
3.17. Diagrama de secuencia AJAX.	20
3.18. Opciones del al presionar sobre el menú desplegable.	21
3.19. Activar/Desactivar actuadores.	21
3.20. Configurar nivel de descarga para accionar la alerta por batería baja.	22
3.21. Configuración de los celulares.	22
3.22. Configuración del rango de temperatura y activación del estado de control automático y alarma.	22
3.23. Configuración de IP, máscara de red y puerta de enlace.	23
4.1. Placa de simulación de temperatura y estado de la batería con el módulo SIM800L integrado.	25
4.2. Placa básica integrada con la CIAA-NXP.	26
4.3. Configuración de una red estática para la interfaz Ethernet.	27
4.4. Pantalla principal de la web de monitoreo.	28
4.5. Activación por control automático debido a temperatura elevada.	28
4.6. SMS recibidos debido a temperatura máxima. El segundo SMS recibido es por temperatura mínima. Y el último recibido debido a batería baja.	29
4.7. Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula.	29
4.8. Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.	30

Índice de Tablas

2.1. Comparación tiempo planificado vs real.	6
3.1. Configuración por default del servidor	20

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción General

1.1. Introducción

1.2. Producción de vino

1.3. Motivación

En el presente trabajo se busca satisfacer las necesidades de la bodega de mi primo. La cual se necesita controlar un proceso delicado como es la fermentación del vino y se requiere un cuidado continuo. Por el momento es realizado por personal día y noche. Teniendo en cuenta que en la zona de trabajo es difícil conseguir personas responsables que se comprometan con el trabajo.

A su vez se busca también una satisfaccion personal, dado que hacía tiempo que buscaba poder implementar en un microtrolador 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.

Y para concluir, el hecho de utilizar la plataforma CIAA, la cual me permitia cumplir todo estos objetivos, haciendo uso de un sistema operativo de tiempo real,(RTOS) como lo llamaremos de ahora en adelante..

1.4. Proyecto CIAA

1.5. Objetivos y Alcances

Capítulo 2

Introducción Específica

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

2.1. Objetivos y alcances

Objetivos

Desarrollar un software que permiea controlar, monitorear y supervisar el proceso de fermentación del vino en las bodegas en forma automática. Aprovechando el proyecto CIAA, se utilizó esta plataforma ya que está preparada especialmente para aplicaciones industriales, siendo esta libre y gratuita.

Alcance

- Sensar la temepratura de un tanque de fermentacion de ya se para vinos tintos o blancos.
- Controlar que la temperatura se mantenga dentro del rango correspondiente según el tipo de vino.
- En caso de corte de energia se notificar vía SMS a un celuar de contacto.
- Mensajes de alerta vía SMS en caso de temperatura fuera de rango.
- Diseño de una página que permita mostrar la informacion del sistema y realizar las configuraciones necesarias.
- Diseño de una placa básica para mostrar el funcionamiento del sitema.

2.2. Requerimientos

Estos fueron consensuados con el responsable de la Bodega Chico Zossi, para trabajar sobre un caso real.

1. Medición de Temperatura:
 - 1.1. Se requiere medir la temperatura de un tanques.
 - 1.2. La resolución de debe ser de $\pm 1^{\circ}C$.

2. Comunicación:

2.1. Debe poder transmitir mensajes SMS notificando el estado/cambios producidos.

2.2. Acceso web para la configuración de los parámetros requeridos de temperatura.

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 del actuadores.

3.2. Deberá alertar si hay corte de energía mediante un SMS.

3.3. Deberá mostrar el nivel de batería.

2.3. Desglose de tareas - GANTT

Para llevar el proyecto a cabo se desarrollo la siguiente planificación de tareas y se estimaron que tiempos debían emplearse para cada una de ellas.

A su vez se analizaron que tareas debian realizarse primero y cuales eran sus dependencias. De esta forma, si se presentara alguna complicacion poder tener un análisis mas completo de como podria esto afectar a la fecha de entrega del proyecto.

Todo este estudio fue aprovechado de haber realizado la asignatura de Gestión de Proyectos. Podemos ver en la siguiente figura 2.1 el plan de trabajo.

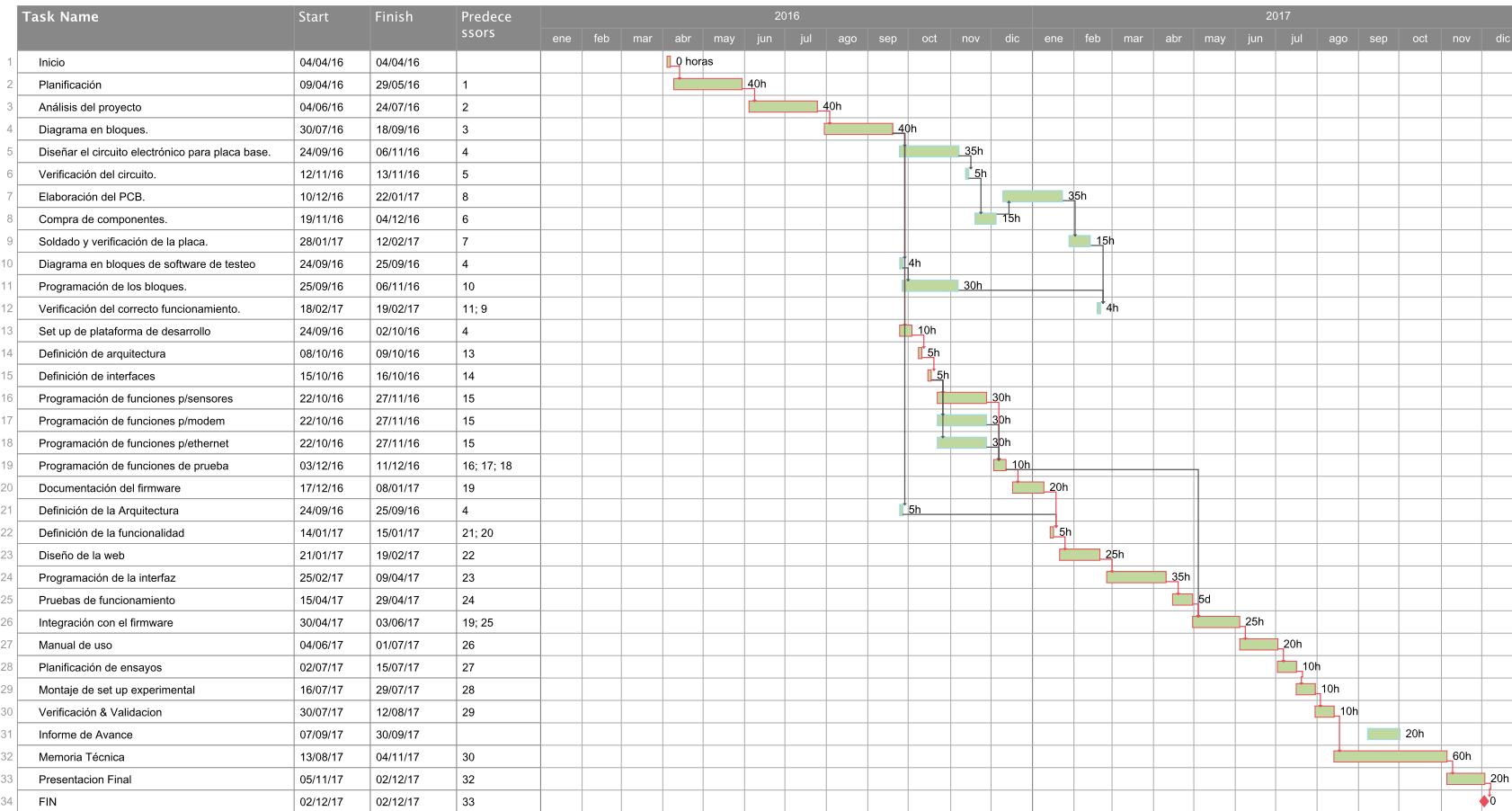


FIGURA 2.1: Listado de tareas, y plafinizacion de tiempos de la implementación.

Pero debido a la falta de experiencia en determinados temas, que fueron más complejos de lo que se estimo. Es por ello que en la siguiente tabla 2.1, se va a mostrar el tiempo real y cuales fueron las tareas que requirieron más tiempo de lo planificado.

Nro	Tarea	Tiempo		% a más
		Estimado	Real	
9	Soldado y verificacion de la placa	15h	24	60
18	Programacion de funciones p/ethernet	30h	50h	66,7
23	Programacion de funciones p/ethernet	25h	35h	40

TABLA 2.1: Comparación tiempo planificado vs real.

Con lo cual hay una diferencia de 39 horas a más de lo planificado. Siendo que para el desarrollo de todo el proyecto se habían estimado 617 horas podemos concluir que la estimación de los tiempos no estuvo mal. Dado que habían temas que no se conocian con profundidad y de haber sido crítica la fecha de entrega se hubiera buscando ayuda de otro programador para poder avanzar en forma paralela.

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 desde el deposito del líquido refrigerante a 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 controlada la temperatura acorde a los parámetros definidos por el usuario.

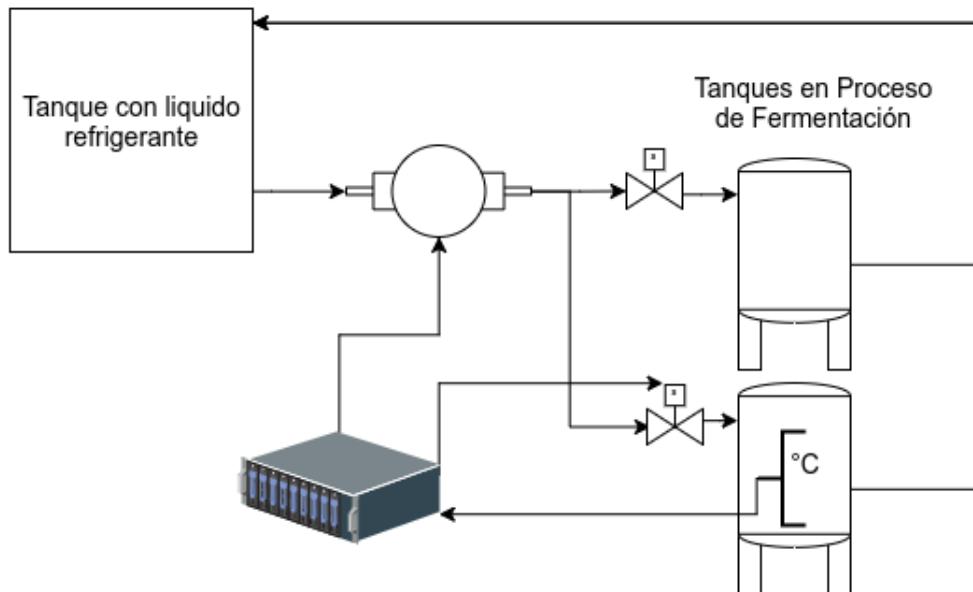


FIGURA 3.1: Entorno del sistema de control del equipo.

3.1.1. Módulos requeridos

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

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

Actuadores: salidas que permiten controlar la bomba y electroválvulas. Debido a que se utilizo la CIAA-NXP esto ya está resuelto.

Sensores: entradas que permitan simular el comportamiento de los sensores, para esto se utilizaron potenciómetros.

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

3.1.2. Características del hardware

La placa diseñada que va a reemplazar el entorno del sistema que vimos en la figura 3.1 por el sistema que va a estar compuesto por el modem y circuitos electronicos que permiten:

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

La forma de conexión con la CIAA-NXP se muestra en la figura 3.2.

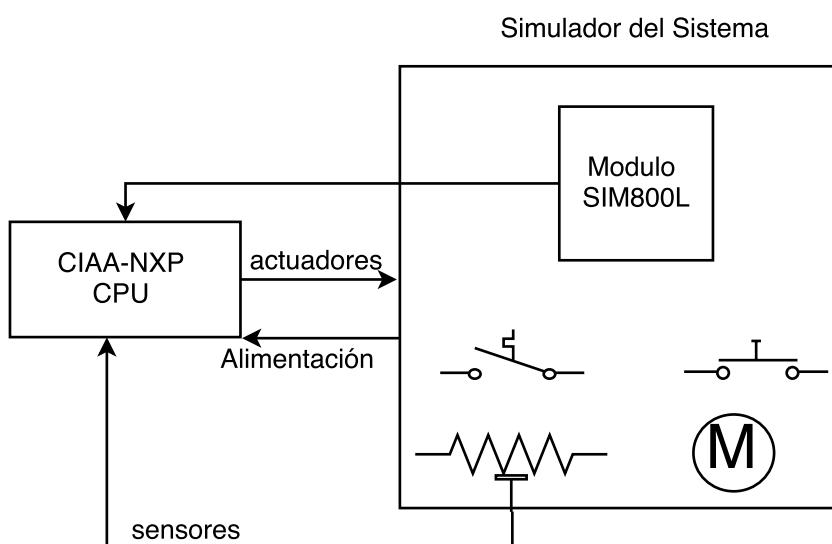


FIGURA 3.2: Entorno del sistema de control del equipo.

Los especificaciones de la placa de simulación son:

- 1 Salida con regulación de corriente, para simular sensores de 4mA a 20mA.

- 1 Salida con tensión variable, utilizando el regulador LM317.
- 1 Salida del sensor de temperatura LM35.
- 2 Salidas con tensión variable, utilizando potenciómetros.
- 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 que se muestra en la figura 3.3, 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 puede apreciar el esquemático implementado para la conexión del módem GSM. Éste requiere adaptar los niveles de tensión para interactuar con la CIA-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]. Para regular a 3.3V se utilizó el integrado LM11733 circuito extraído de la nota de aplicación [1].

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, y hasta un sensor de temperatura. 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.

Para concluir, se utilizaron unos 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-CIA¹.

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

UART + MAX232 + SIM800L

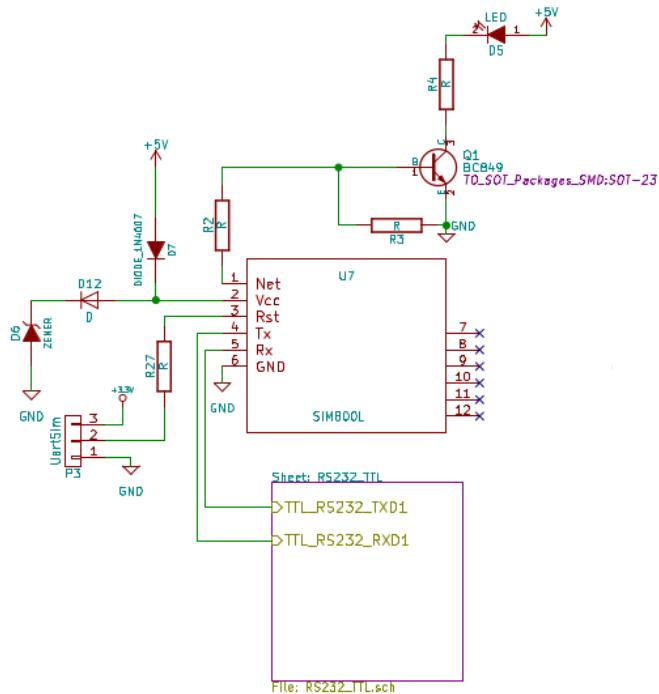


FIGURA 3.4: Módulo SIM800l e índice de esquemáticos.

**Fuente de alimentación con rango aproximado de entrada de 8 a 30V (CC).
Tensiones de salida: 5V, 3A y 3.3V, 1A.
Posibilidad de alimentar la CIAA a través del puerto USB mediante la conexión con D1.**

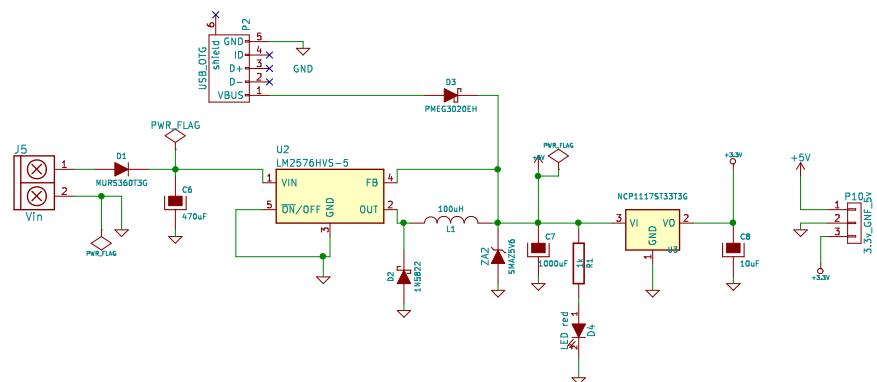


FIGURA 3.5: Fuente de alimentación DC/DC.

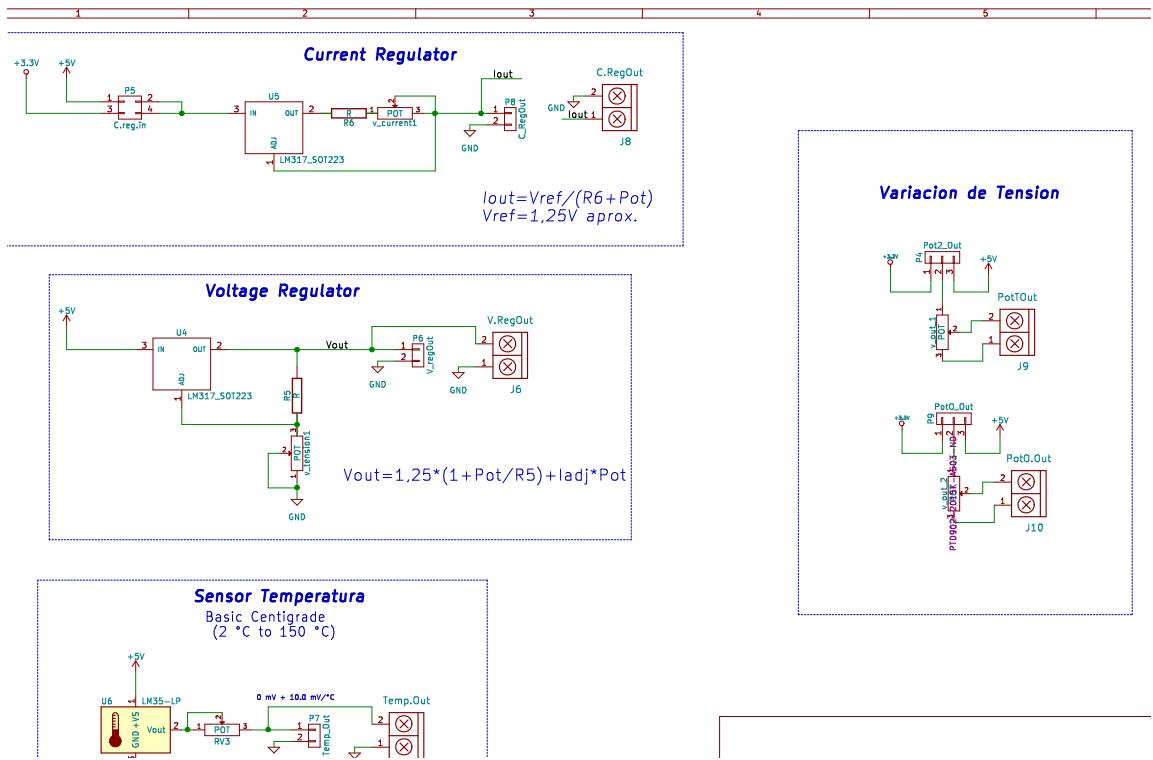


FIGURA 3.6: Regulador de corriente, de tensión y del sensor de temperatura.

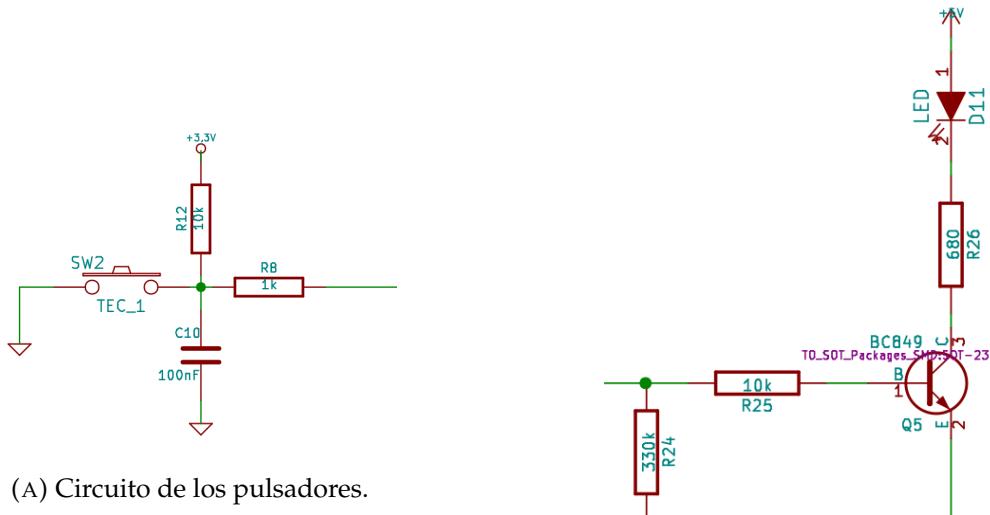


FIGURA 3.7: Esquemáticos extraídos de la EDU-CIAA.

Con los esquemáticos terminados, se pasó a la elaboración del PCB. Se realizaron los ruteo necesarios, los cuales podemos verlos en las siguientes figuras 3.8 - 3.9.

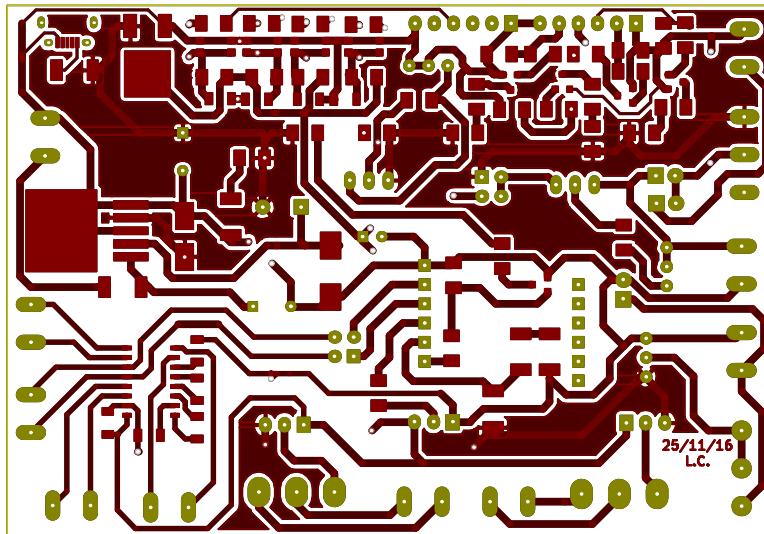


FIGURA 3.8: Capa superior de la placa simulador de sensores.

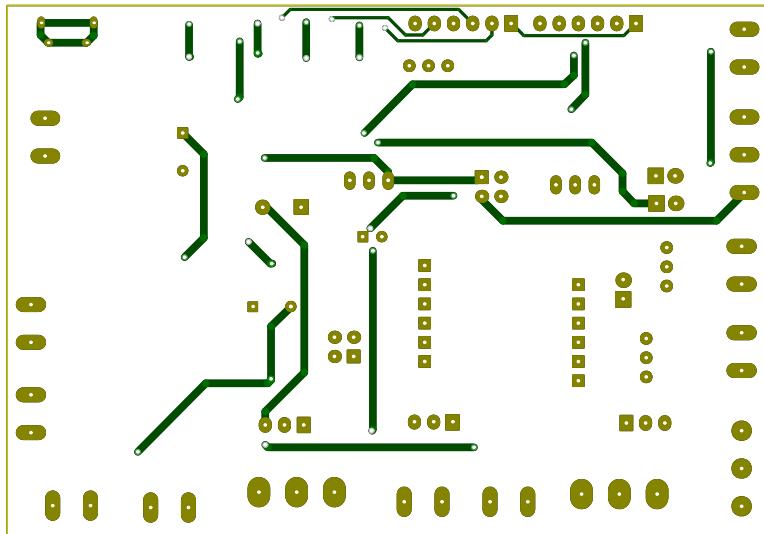


FIGURA 3.9: Capa inferior de la placa simulador de sensores.

Realizando una vista previa de como quedaría el esquema y aprovechando el potencial de la herramienta de KICAD, se realizó una vista en 3D de como quedaría. La que se puede apreciar en la figura 3.10.

No obstante, por cuestiones de tiempo no se llegó a implementar. Simplemente quedó en una primera etapa para una posterior implementación. Se terminó desarrollando una placa experimental que permitió realizar las primeras pruebas. Podemos verla en la siguiente figura 3.11.

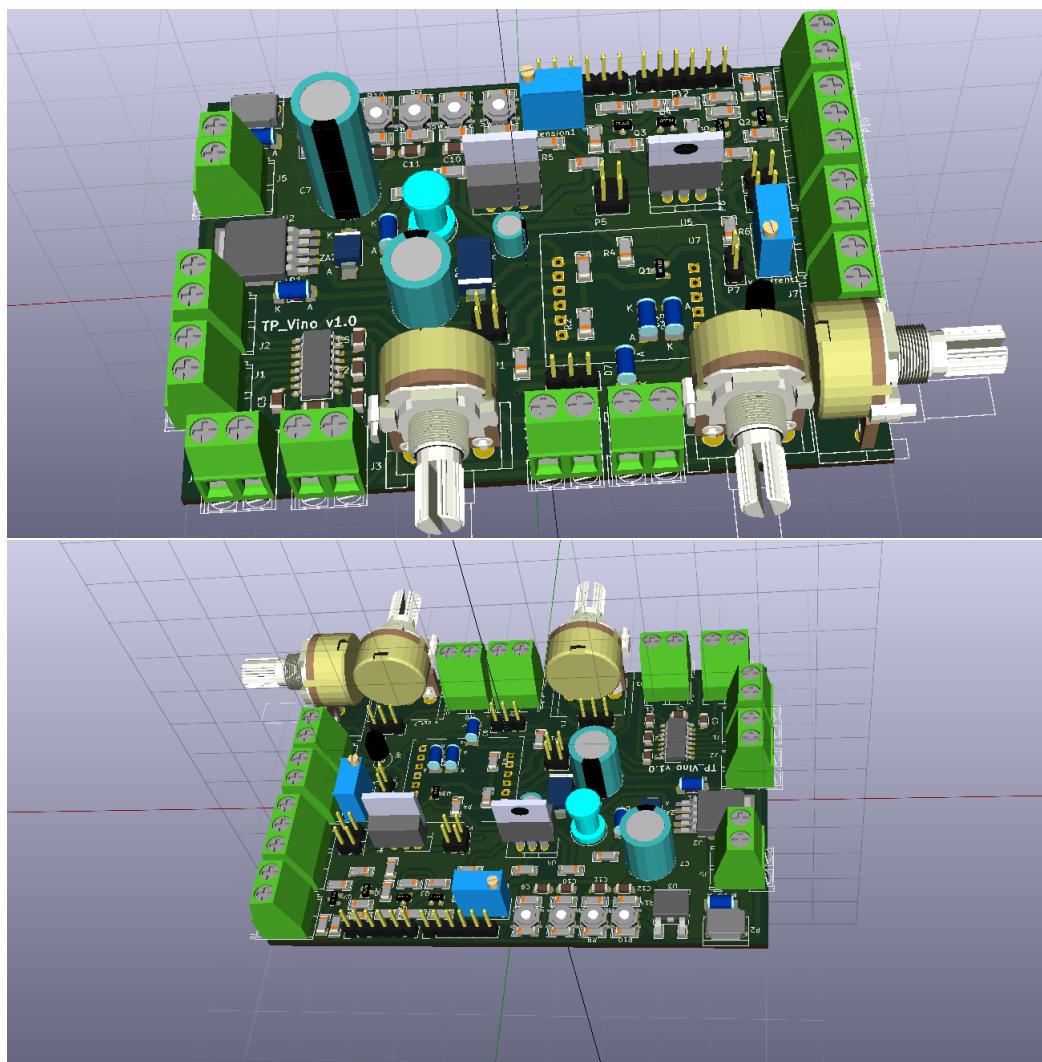


FIGURA 3.10: Placa simulador de sensores 3D.



FIGURA 3.11: Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado.

3.2. Software

3.2.1. Aplicación

Se dividieron las tareas por recursos:

- Módem
- Control
- Sensores

Módem

En esta tarea se inicializa las configuraciones iniciales para el uso del módem. Es la encargada de interactuar con el módulo GPRS, realizando consultas del estado de señal y de enviando los mensajes de alarmas en caso de ser necesarios.

Podemos ver como fue implementado en el diagrama de flujo en la siguiente figura 3.12.

Control

Con esta tarea se analiza el resultado los sensores de temperatura, y de la batería. Se verifica que estén dentro de los parámetros configurados por el usuario y se actúa en consecuencia. Siendo que el sistema de control actua cuando: la temperatura del sistema excede la máxima configurada, este procede a encender la bomba para que circule el líquido refrigerante y la electroválvula correspondiente. Luego una vez que el sistema llega a la temperatura mínima, este deshabilita la bomba y la electroválvula.

También es la encargada de setear los actuadores según sean especificados en la plataforma web.

La implementación se puede apreciar en la siguiente figura 3.13.

Sensor

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

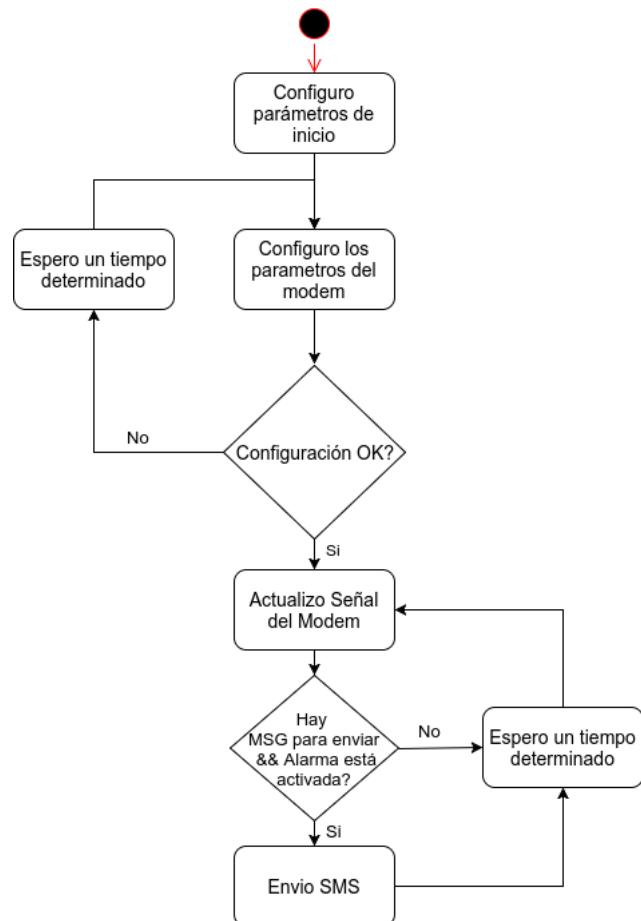


FIGURA 3.12: Diagrama de flujo de la tarea que controla el módem GPRS.

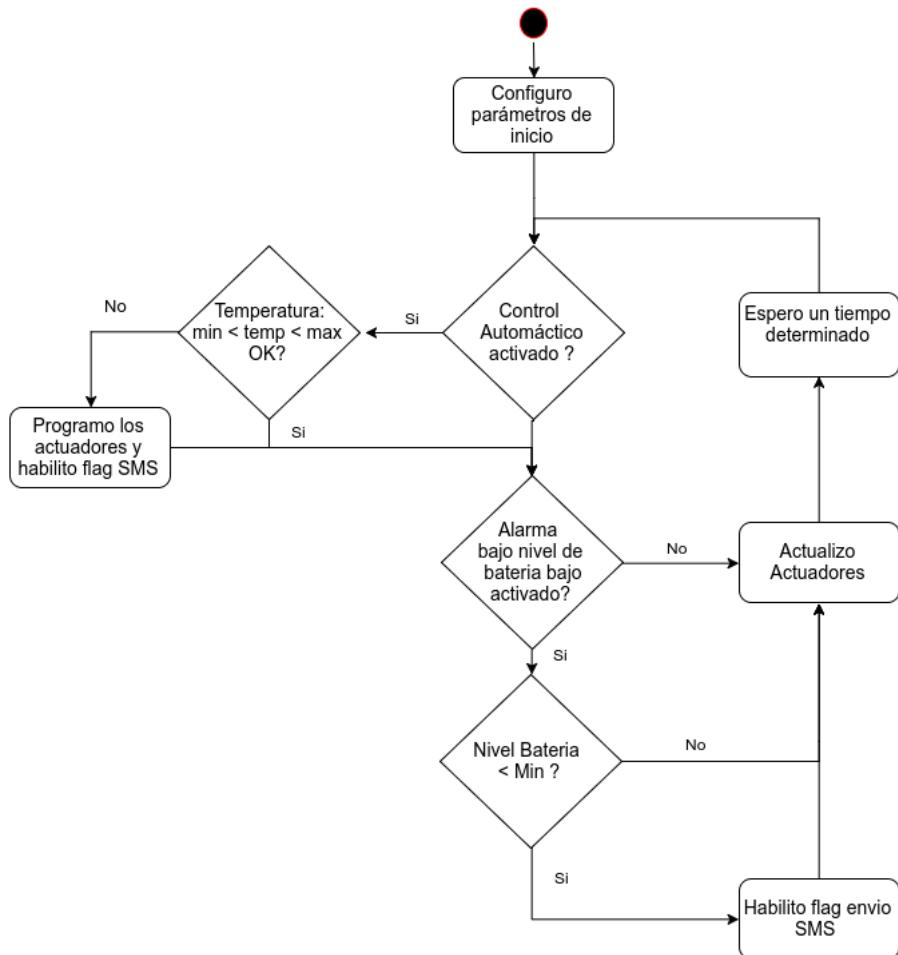


FIGURA 3.13: Diagrama de flujo de la tarea de control.

Como vemos en el siguiente diagrama de flujo, figura 3.14.

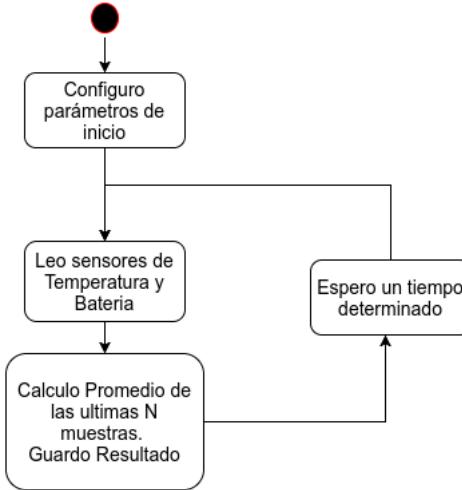


FIGURA 3.14: Diagrama de flujo del sensor.

Dado que por falta de tiempo no se elaboró el PCB diseñado, pero se realizó una versión simplificada del mismo que se verá más adelante.

Surgieron varios imprevistos, principalmente con el modulo SIM800L, el cual posee grandes inconvenientes a la hora de detectar la SIM. Luego de una serie de búsquedas en internet, donde muchos habían tenido un problema similar dado que:

- La fuente de alimentación fue subdimensionada.
- Módulo es muy sensible y se debe limpiar bien los contactos de la SIM.
- O problemas con las soldaduras del módulo.

Finalmente, como el módulo ya había sido comprado y se encontró la falla a tiempo, se utilizó este en el prototipo. Pero este no se recomienda por los problemas mencionados anteriormente. A futuras implementación este módulo será reemplazado.

3.2.2. 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: En dicho código se desarrolla la página web, este permite darle la estructura básica y se realiza sobre texto plano, es decir no se requiere ninguna aplicación adicional simplemente un editor de texto. Luego es el navegador web quien tiene la tarea de interpretarlo.

CSS: Mediante este código 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 código 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: Si bien la implementación que nos ofrece FreeRTOS es reducida. Es más que suficiente para poder agregar contenido a nuestra pagina en forma dinámica.

CGI: Por medio de este el cliente que navegar por nuestra plataforma puede enviar y solicitar datos, al servidor.

Diseño web

Al iniciar el diseño de la web, no tenía experiencia previa. Es por ello que se inició buscando información en internet, donde se encontraron varias páginas con cursos de HTML, CSS, y AJAX que son de forma online y gratuita. Las que mas se utilizó fueron:

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

con lo cual se logró una primera versión de la web, como se puede ver en la figura 3.15.

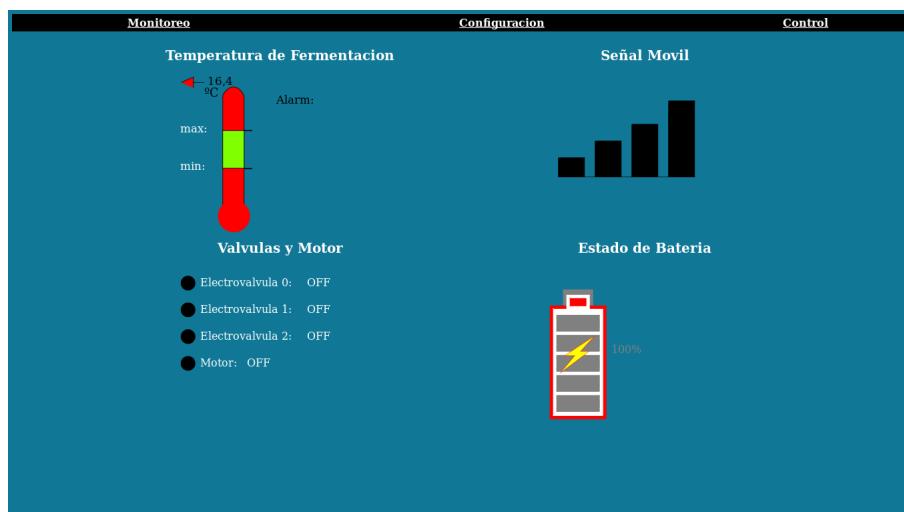


FIGURA 3.15: Primera versión de la web de monitoreo.

Cuando ya estaba funcionando en forma correcta, se pasó a mejorar las 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 modifica la apariencia de la web. Pudiendo ver dicho cambio en la siguiente figura 4.4.

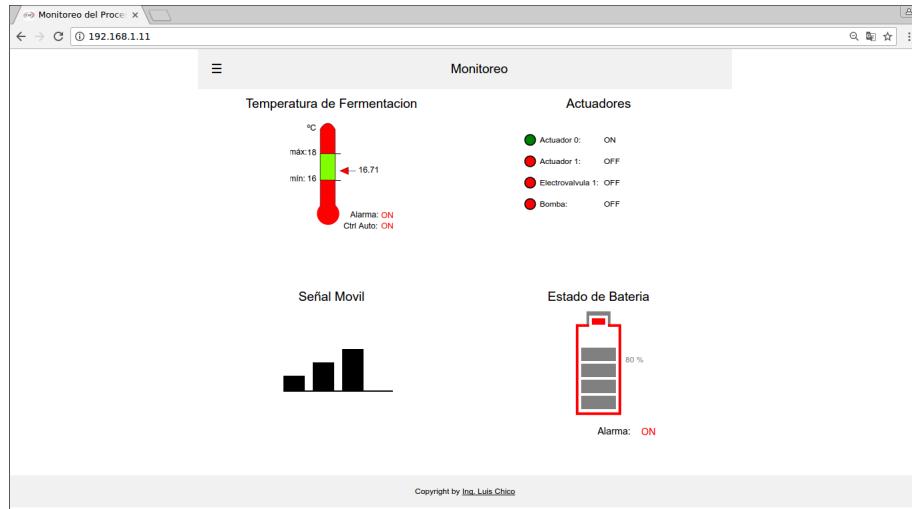


FIGURA 3.16: Versión actualizada haciendo uso del template.

Este template fue encontrando en w3school². Siendo que cumplía los siguientes requerimientos:

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

Configuración del servidor

Al iniciar el servidor debemos tener en cuenta que este sera 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 utiliza la API de lwIP 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 utilizo AJAX, permitiendo que el cliente sea el encargado de realizar las consultas al server en forma periódica. 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. Podemos ver el procedimiento mencionado en la siguiente figura³ 3.17.

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

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

³<http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Trabajo-Final-CESE-Patricio-Bos-2016.pdf>, figura 3.9.

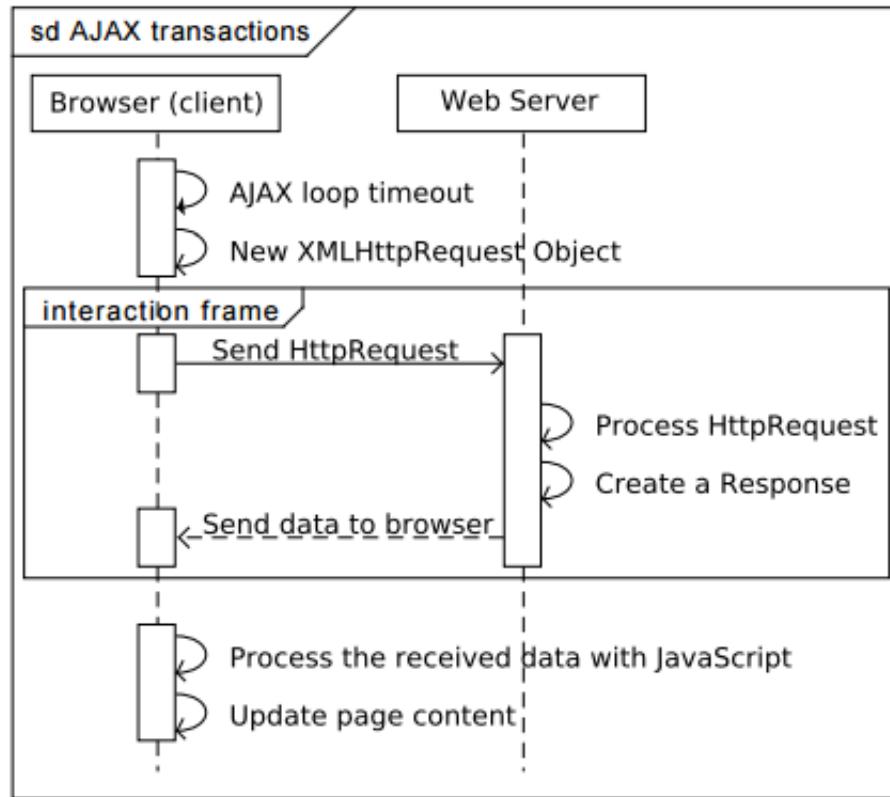


FIGURA 3.17: Diagrama de secuencia AJAX.

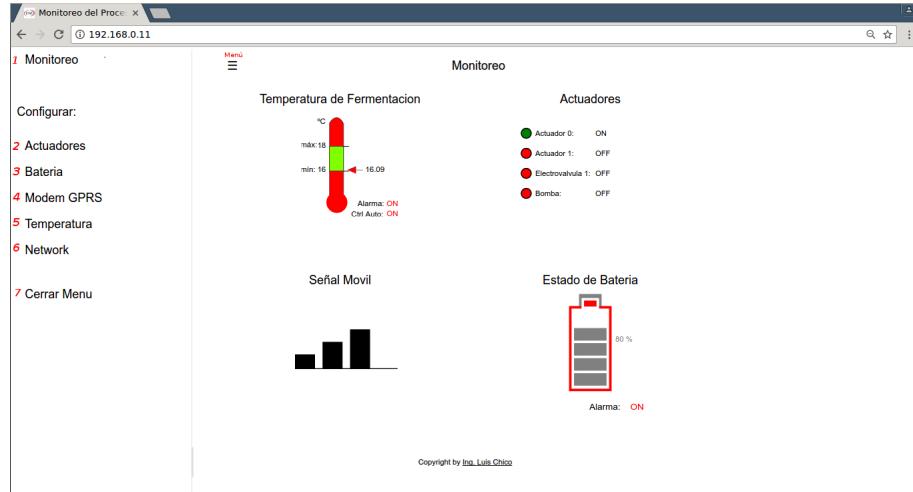
Parámetro	Valor
IP	192.168.1.11
Máscara de red	255.255.255.0
Puerta de enlace	192.168.0.1
Puerto	80
DHCP	Deshabilitado

TABLA 3.1: Configuración por default del servidor

3.3. Diseño del sistema web embebido

Se procede a navegar por el menú, figura 3.18, para constatar que todo el sistema está funcionando en forma correcta. Al presionar donde dice menú, se puede configurar lo siguiente:

- 1. Monitoreo:** Dirige a la pantalla principal, donde ver el estado del sistema. figura 4.4.
- 2. Actuadores:** setear en forma manual el estado de los mismos. figura 3.19.
- 3. Batería:** Aquí se podrá setear la alarma debido al nivel de descarga de batería. Esta enviará un SMS indicando que se alcanzó dicho estado. figura 3.20.
- 4. Módem GPRS:** En este menú podremos configurar 2 personas a quienes serán enviadas las alertas debido al accionar de alguna alarma, figura 3.21.



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. figura 3.22.
6. **Network:** Aquí se configura los parámetros de la red, IP, máscara de red y puerta de enlace. figura 3.23.
7. **Cerrar Menú:** Esta opción cierra el menú, permitiendo quitar las opciones de menú.



FIGURA 3.19: Activar/Desactivar actuadores.

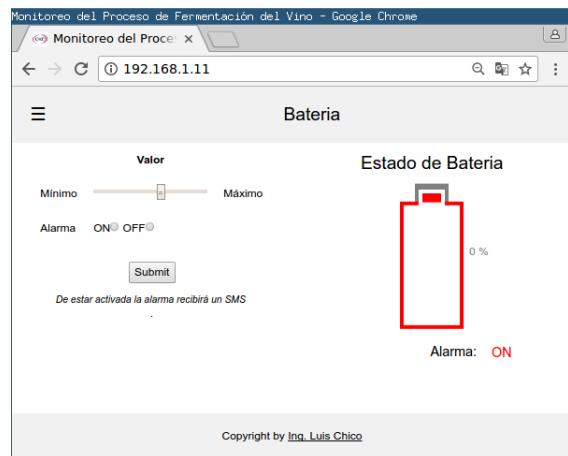


FIGURA 3.20: Configurar nivel de descarga para accionar la alerta por batería baja.

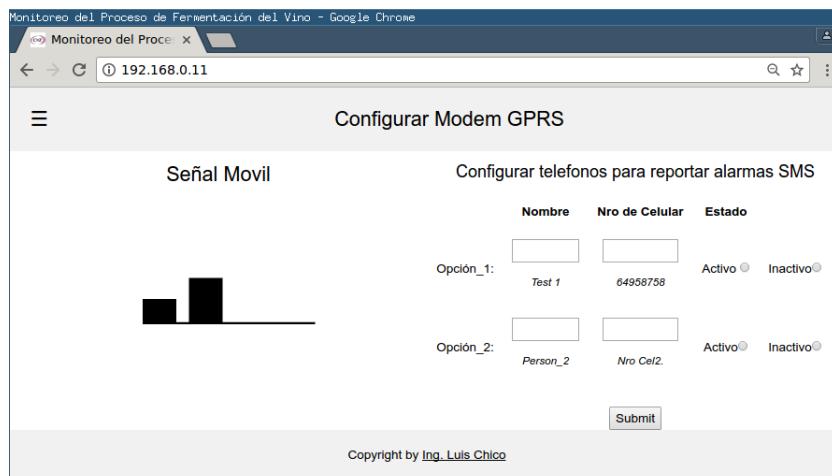


FIGURA 3.21: Configuración de los celulares.

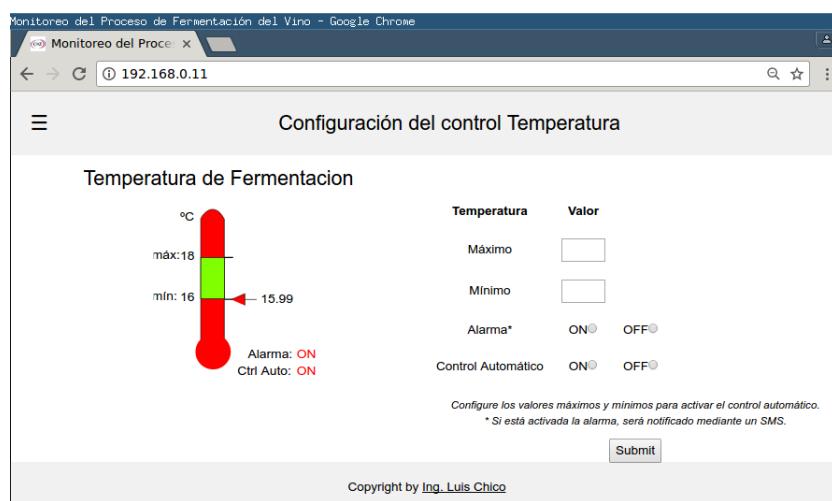


FIGURA 3.22: Configuración del rango de temperatura y activación del estado de control automático y alarma.

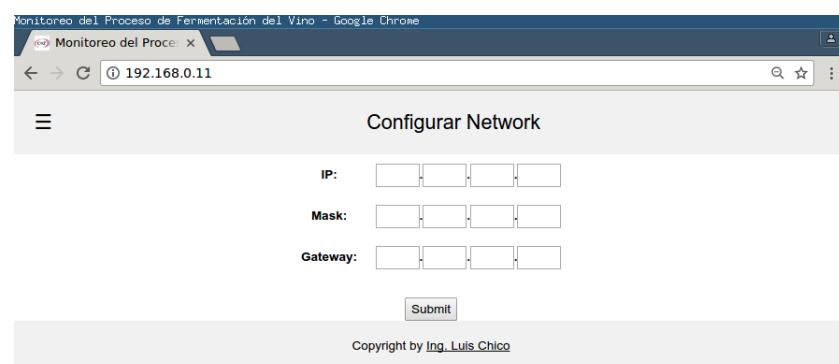


FIGURA 3.23: Configuración de IP, mascara de red y puerta de enlace.

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 una placa básica la cual ya fue vista en el capítulo 3.2, que para facilidad del lector se vuelve a mostrar en la figura 4.1, consta de dos potenciómetros que permiten simular la variación de estos dispositivos. Sobre la misma se conecto un conversor RS232 a TTL, para lograr la integración con el módem.

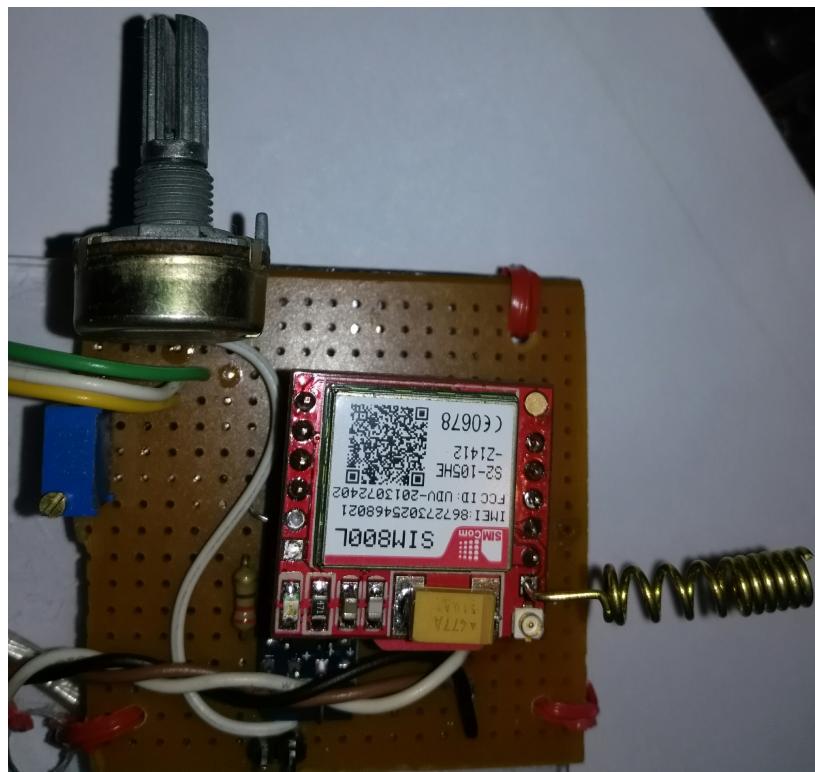


FIGURA 4.1: Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.

En la figura 4.2 se muestra la placa básica que contiene el módem y los componentes que me permiten simular las variaciones de temperatura y el estado de la batería integrado a la CIAA-NXP.

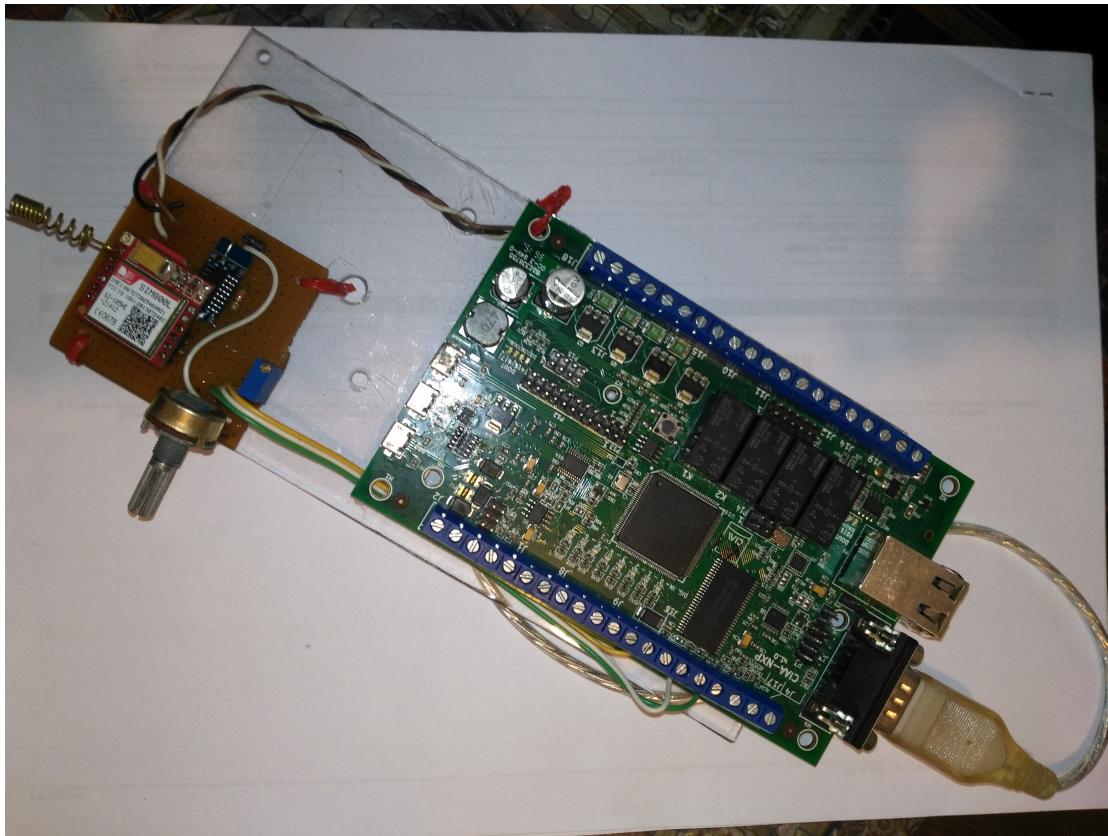


FIGURA 4.2: Placa básica integrada con la CIAA-NXP.

4.2. Configuraciones para la PC

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

Una vez que se tiene la configuración de red adecuada se obtiene en el navegador web la imagen 4.4 que fue vista 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 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 estamos monitoreando. En la web se ve cómo en la figura 4.5. y se recibe la alerta correspondiente mediante un SMS como se puede observar en la figura 4.6.

Se verificó que éste se mantiene activado hasta llegar a la temperatura mínima, en este instante procede a desconectar la bomba y a cerrar la electroválvula. En la

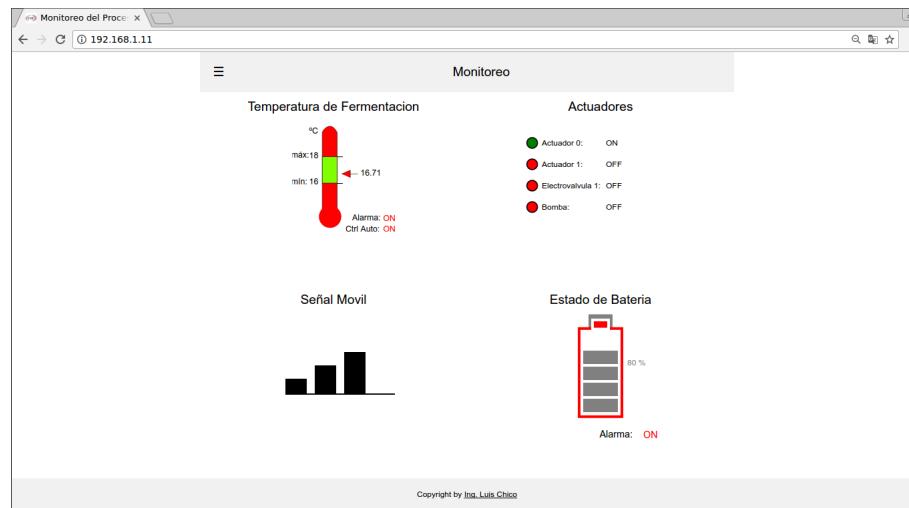


FIGURA 4.4: Pantalla principal de la web de monitoreo.

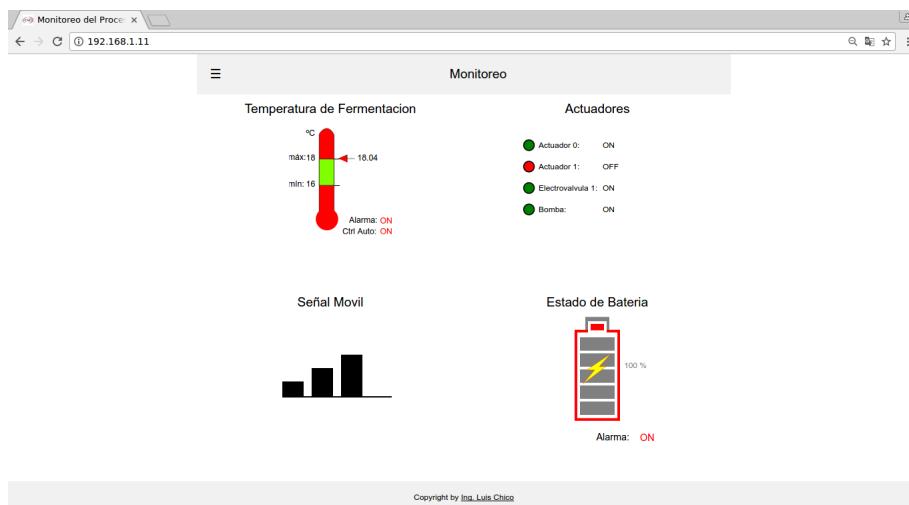


FIGURA 4.5: Activación por control automático debido a temperatura elevada.

web se verá como en la figura 4.7. y dado su respectiva alerta mediante un SMS como vemos 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 enviará SMS como el que observamos en figura 4.6.

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.

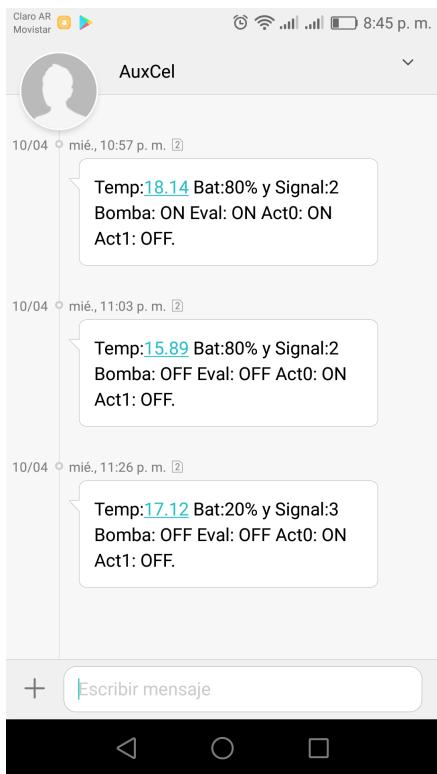


FIGURA 4.6: SMS recibidos debido a temperatura máxima.
El segundo SMS recibido es por temperatura mínima.
Y el último recibido debido a batería baja.

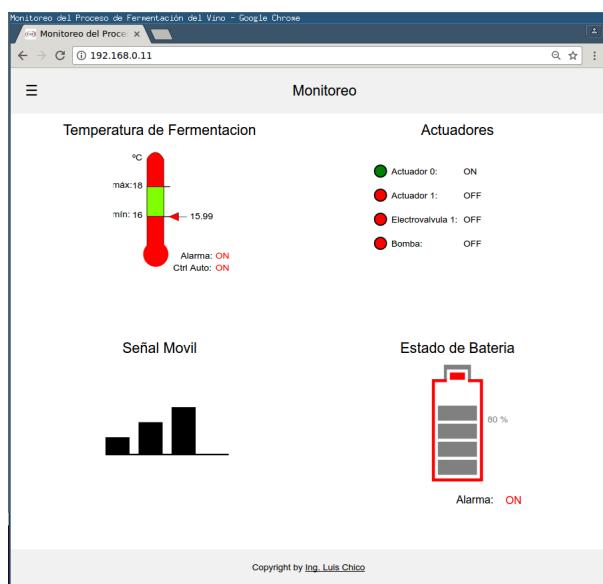


FIGURA 4.7: Temperatura en nivel mínimo, se desactiva la bomba
y la electroválvula.



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

Capítulo 5

Conclusiones

5.1. Conclusiones generales

Al finalizar esta etapa se logró:

- Acceder mediante una página web accedemos 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 sus funciones se puede 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 de mas 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>.