

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
2. Diseño e Implementación	3
2.1. Hardware	3
2.1.1. Diagrama en bloque	3
2.1.2. Módulos requeridos	4
2.1.3. Características del hardware	4
2.2. Software	7
2.2.1. Aplicación	7
2.2.2. Web	8
3. Ensayos y Resultados	15
3.1. Banco de pruebas	15
3.2. Configuraciones para la PC	16
3.3. Verificación del sistema web embebido	17
3.4. Pruebas funcionales del hardware	19
4. Conclusiones	23
4.1. Conclusiones generales	23
4.2. Próximos pasos	23

Índice de figuras

2.1. Esquema de interaccion del dispositivo.	4
2.2. Modem SIM800L	5
2.3. Módulo sim800l e índice de esquemáticos.	5
2.4. Fuente de alimentación DC/DC.	6
2.5. Regulador de corriente, de tensión y del sensor de temperatura.	6
2.6. Entradas a leds y salidas de los pulsadores.	7
2.7. Capa superior de la placa simulador de sensores.	7
2.8. Capa inferior de la placa simulador de sensores.	8
2.9. Placa simulador de sensores 3D.	9
2.10. Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.	9
2.11. Diagrama de flujo de la tarea que controla el modem GPRS.	10
2.12. Diagrama de flujo de la tarea de control.	11
2.13. Diagrama de flujo del sensor.	11
2.14. Primera versión de la web de monitoreo.	12
2.15. Versión actualizada haciendo uso del template.	12
2.16. Diagrama de secuencia AJAX.	13
 3.1. Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.	15
3.2. Placa básica integrada con la CIAA-NXP.	15
3.3. Configuración de una red estática para la interfaz Ethernet.	16
3.4. Pantalla principal de la web de monitoreo.	16
3.5. Opciones del al presionar sobre el menú desplegable.	17
3.6. Activar/Desactivar actuadores.	18
3.7. Configurar nivel de descarga para accionar la alerta por batería baja.	18
3.8. Configuración de los celulares de quienes van a recibir las alarmas por SMS.	18
3.9. Configuración del rango de temperatura y activación del estado de control automático y alarma.	19
3.10. Formulario de configuración de IP, mascara y puerta de enlace.	19
3.11. Activación por control automático debido a temperatura elevada.	20
3.12. SMS recibidos debido a temperatura máxima. El segundo SMS recibido es por temperatura mínima. Y el último recibido debido a bateria baja.	21
3.13. Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula.	21
3.14. Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.	22

Índice de Tablas

2.1. Configuracion por default del servidor	14
---	----

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción General

```
\include{Chapters/Chapter1}
\include{Chapters/Chapter2}
\include{Chapters/Chapter3}
\include{Chapters/Chapter4}
\include{Chapters/Chapter5}
```


Capítulo 2

Diseño e Implementación

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

2.1. Hardware

Se diseño de un circuito, 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*. Si bien, por falta de tiempo, no se llevó a cabo el desarrollo de la placa diseñada que veremos a , a fines de poder testear el sistema se elaboró una versión simplificada.

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 debido a:

- Fuente de alimentación subdimensionada.
- Módulo muy sensible, limpiar bien los contactos de la sim.
- Problemas en la soldadura del módulo.

Finalmente, si bien dicho modulo ya había sido comprado y se encontró la falla a tiempo se utilizó en el prototipo, pero este no se recomienda. Ya que trae consigo los problemas mencionados anteriormente.. A futuras implementaciones este módulo será reemplazado.

2.1.1. Diagrama en bloque

En el siguiente esquema, Figura 2.1 podemos apreciar el entorno de trabajo del dispositivo. Donde el mismo deberá interactuar con una bomba de agua que permite llevar desde el depósito del líquido refrigerante a los tanques que están en proceso de fermentación. Recibiendo como información la temperatura de los tanques, este debe controlar la bomba y la electroválvula correspondiente. De esta forma permitirá tener controlada la temperatura acorde a los parámetros establecidos por el usuario.

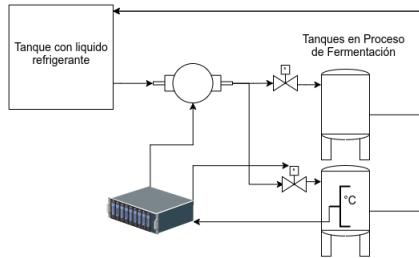


FIGURA 2.1: Esquema de interacción del dispositivo.

2.1.2. Módulos requeridos

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

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

Actuadores: salidas que nos permiten controlar la bomba y electroválvulas. Debido a que se utilizó la CIA-NXP 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 CIA-NXP.

Con lo cual podemos ver que el CPU y los actuadores ya están desarrollados al haber elegido la CIA. Nos quedará ver los sensores y el modem en la próxima sección.

2.1.3. Características del hardware

El sistema cuenta con:

- 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 Sócalo 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 siguiente Figura 2.2, es un módem GPRS que cuenta con las siguientes especificaciones:

- Alimentación: 3.4V a 4.4V (4.0V recomendado)
- CuatriBanda 850/900/1800/1900MHz
- GPRS Multi Slot class 8/10



FIGURA 2.2: Modem SIM800L.

- Control mediante comandos AT (GSM 07.07 ,07.05 y comandos AT SIM-COM).

En la figura 2.3 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 utilizamos un max3232.

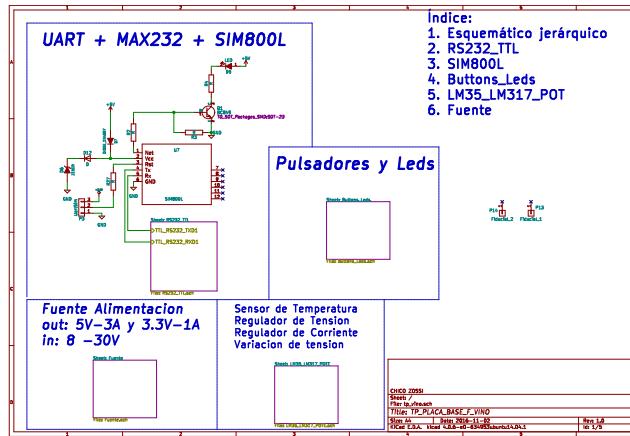


FIGURA 2.3: Módulo sim800l e índice de esquemáticos.

Para la alimentación del circuito se utilizó el regulador LM2576. El mismo se tomó de la nota de aplicación el diagrama típico.¹ Para regular a 3.3V se utilizó el integrado LM11733 circuito extraído de la nota de aplicación.²

Para lograr la simulación de distintos tipos de sensores, se utilizaron circuitos como se muestra en la figura 2.5 que permiten generar corrientes 4mA a 20mA, variaciones de la tensión, y hasta un sensor de temperatura. Para los circuitos se utilizaron las notas de aplicación de Texas Instruments³

¹Nota de aplicación Texas Instruments: <http://www.ti.com/lit/ds/symlink/lm2576.pdf>

² Texas Instruments: <http://www.ti.com/lit/ds/symlink/lm1117.pdf> - Capítulo 8 Figura 16

³Reguladores de Tensión y Corriente <http://www.ti.com/lit/ds/symlink/lm317.pdf> - capítulo 8 páginas 12 y 13 y sensor temperatura: <http://www.ti.com/lit/ds/symlink/lm35.pdf> - 8.2.1 Basic Centigrade Temperature Sensor

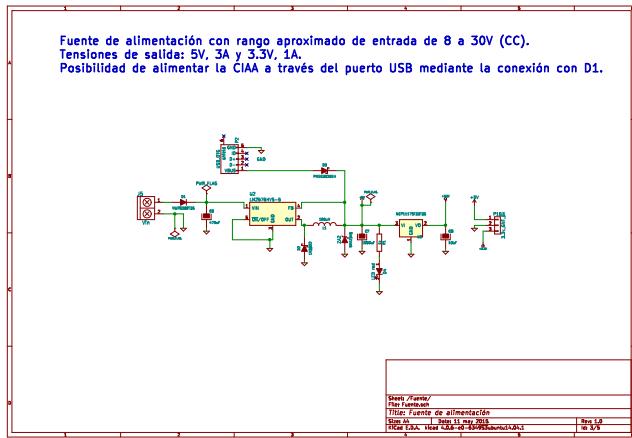


FIGURA 2.4: Fuente de alimentación DC/DC.

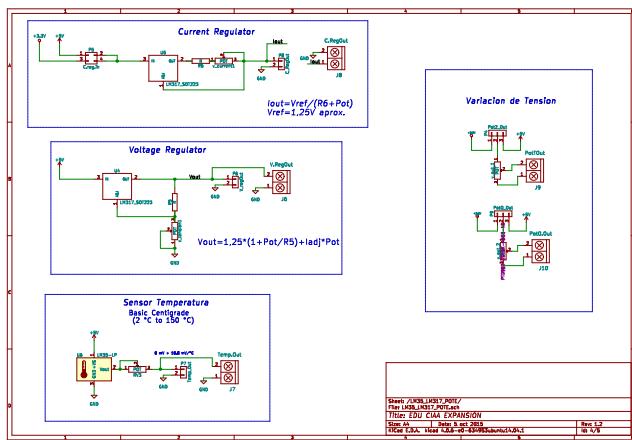


FIGURA 2.5: Regulador de corriente, de tensión y del sensor de temperatura.

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-CIA⁴

Una vez realizados los esquemáticos, se pasó a la elaboración del PCB. Se realizaron los ruteo necesarios. Y como se puede observar en la siguientes figuras 2.7 - 2.8

Realizando una vista previa de como quedaría el esquema y aprovechando el potencial de la herramienta de kicad, se realizó una vista 3D de como quedaría, la que se puede apreciar en la figura 2.9.

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 2.10

⁴http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp:edu-ciaa-nxp_color.pdf

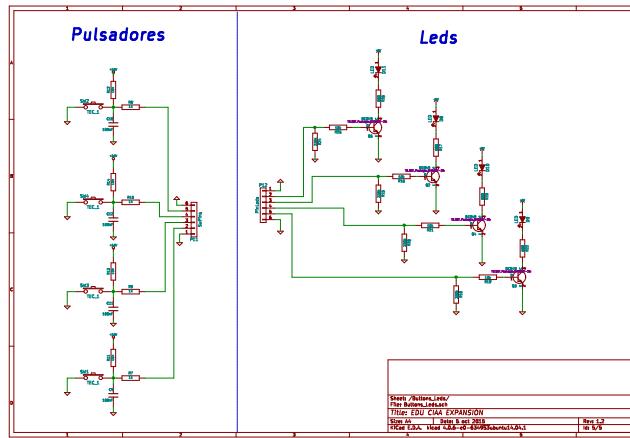


FIGURA 2.6: Entradas a leds y salidas de los pulsadores.

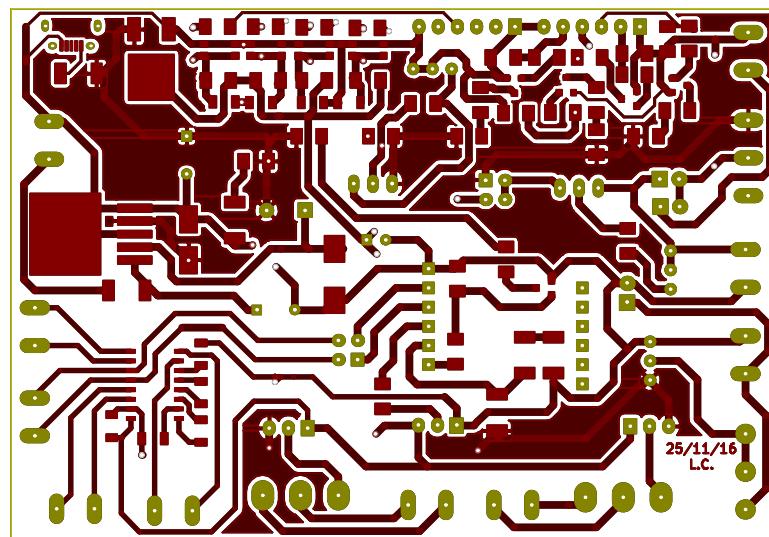


FIGURA 2.7: Capa superior de la placa simulador de sensores.

2.2. Software

2.2.1. Aplicación

Se dividieron las tareas por recursos:

- Modem
- Control
- Sensores

Modem

En esta tarea se inicializa las configuraciones iniciales para el uso del modem. Es la encargada de interactuar con el módulo GPRS, realizando consultas del estado

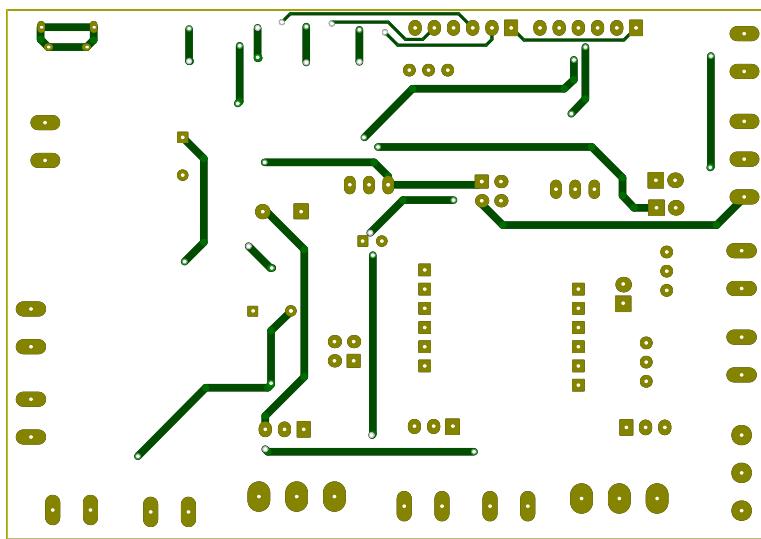


FIGURA 2.8: Capa inferior de la placa simulador de sensores.

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 2.11.

Control

Con esta tarea se analiza el resultado los sensores de temperatura, y de labatería. Se verifica que esten dentro de los parametros configurados por el usuario y se actua en consecuencia.

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

La implementacion se puede apreciar en la siguiente figura: 2.12.

Sensor

En este caso la tarea es más simple, consulta el estado de los sensores y calcula el promedio de las ultimas N muestras, este es el valor que usa el control para definir si es necesario encender los actuadores correspondientes.

Como vemos en el siguiente diagrama de flujo, figura: 2.13.

2.2.2. 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:

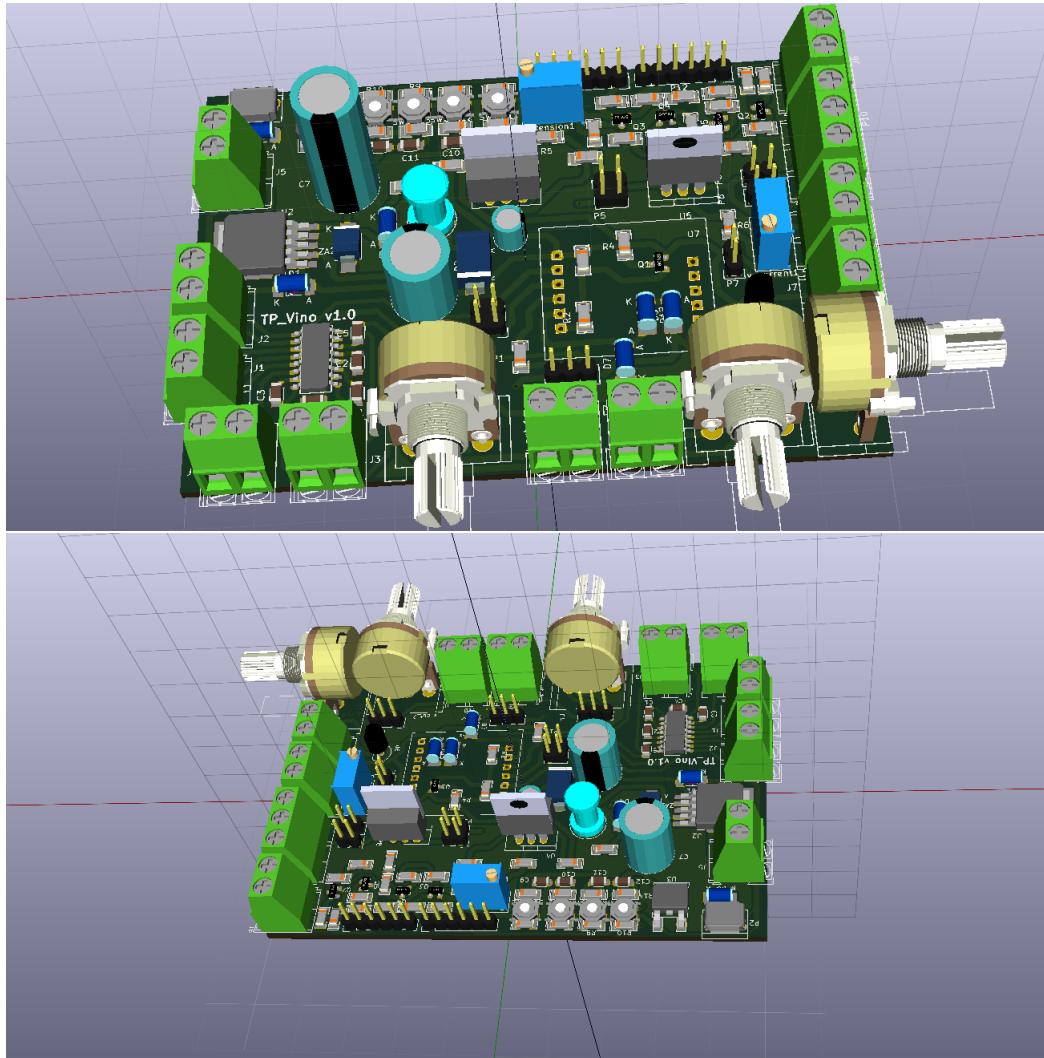


FIGURA 2.9: Placa simulador de sensores 3D.



FIGURA 2.10: Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.

HTML: En dicho código se desarrolló 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.

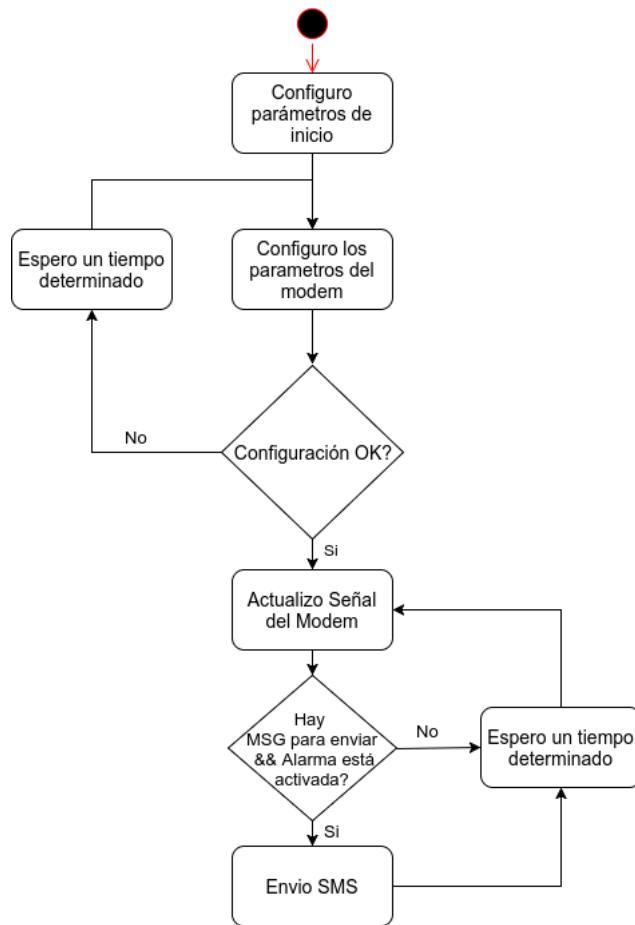


FIGURA 2.11: Diagrama de flujo de la tarea que controla el modem GPRS.

SVG: Mediante este código se implementarán los gráficos correspondientes para mostrar los resultados de los sensores y el estado de los actuadores.

AJAX: es un recurso muy utilizado para cuando desarrollan web en sistemas embebidos. Permite actualizar el estado de la web sin tener que consultar todo el contenido, simplemente se descarga una primera vez y luego se puede realizar consultar 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 página 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, varias páginas facilitaban algunos cursos de HTML, CSS, y AJAX de forma online y gratuita. Las más usadas fueron:

- <https://www.w3schools.com>

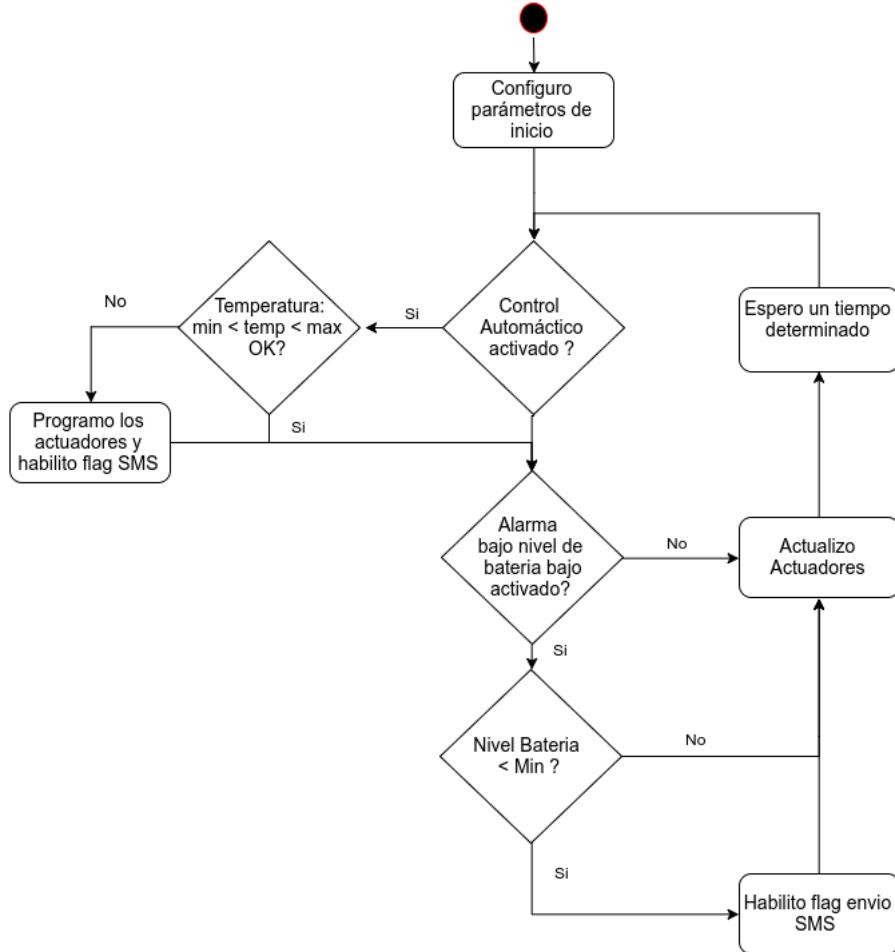


FIGURA 2.12: Diagrama de flujo de la tarea de control.

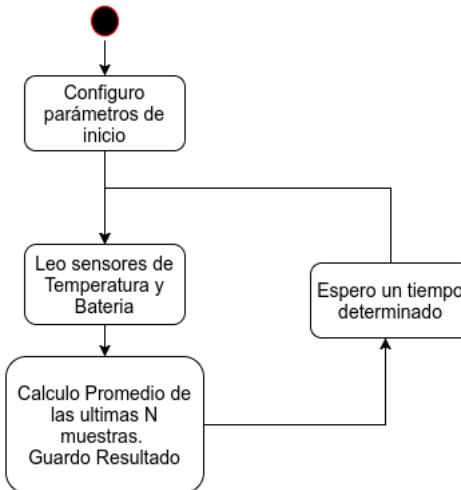


FIGURA 2.13: Diagrama de flujo del sensor.

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

con lo cual se logró una primera versión de la web, ver figura:2.14.

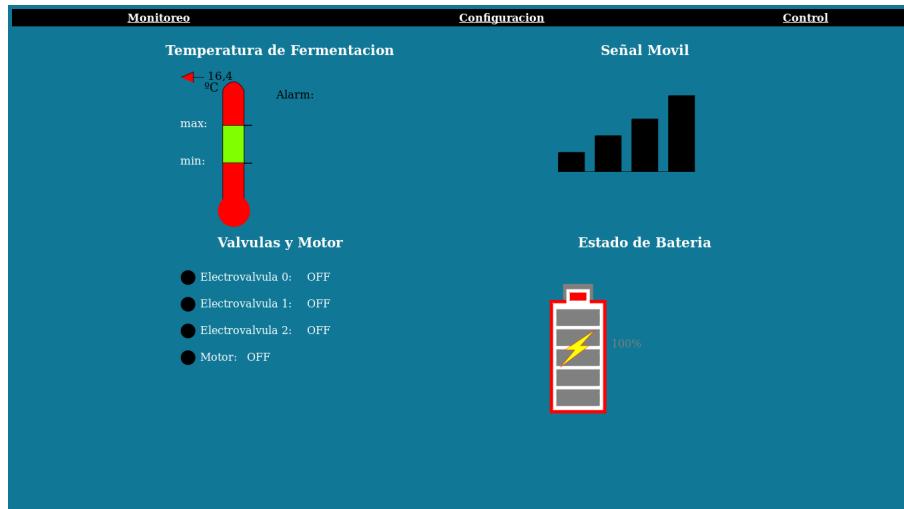


FIGURA 2.14: Primera versión de la web de monitoreo.

Luego una vez que esta, estaba funcionando en forma correcta, se paso a mejorar las presentación de la misma, y 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:3.4.

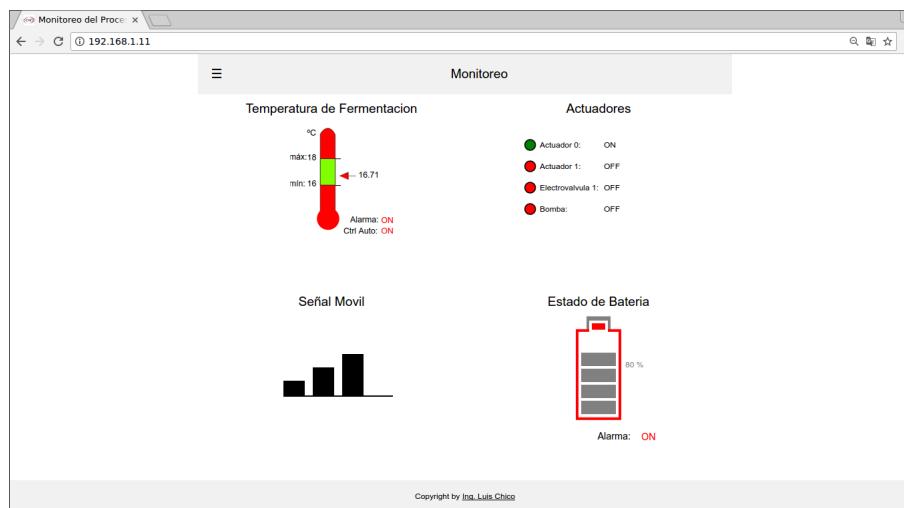


FIGURA 2.15: Versión actualizada haciendo uso del template.

Este fue encontrando en w3school⁵. Siendo que cumplia los siguientes requerimientos:

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

⁵https://www.w3schools.com/w3css/w3css_templates.asp

Configuración del servidor

Al iniciar el servidor debemos 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 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 utiliza 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 browser, mediante una serie de funciones en JavaScript es el encargado de actualizar los cambios realizados. Podemos ver el procedimiento mencionado en la siguiente figura⁶:2.16

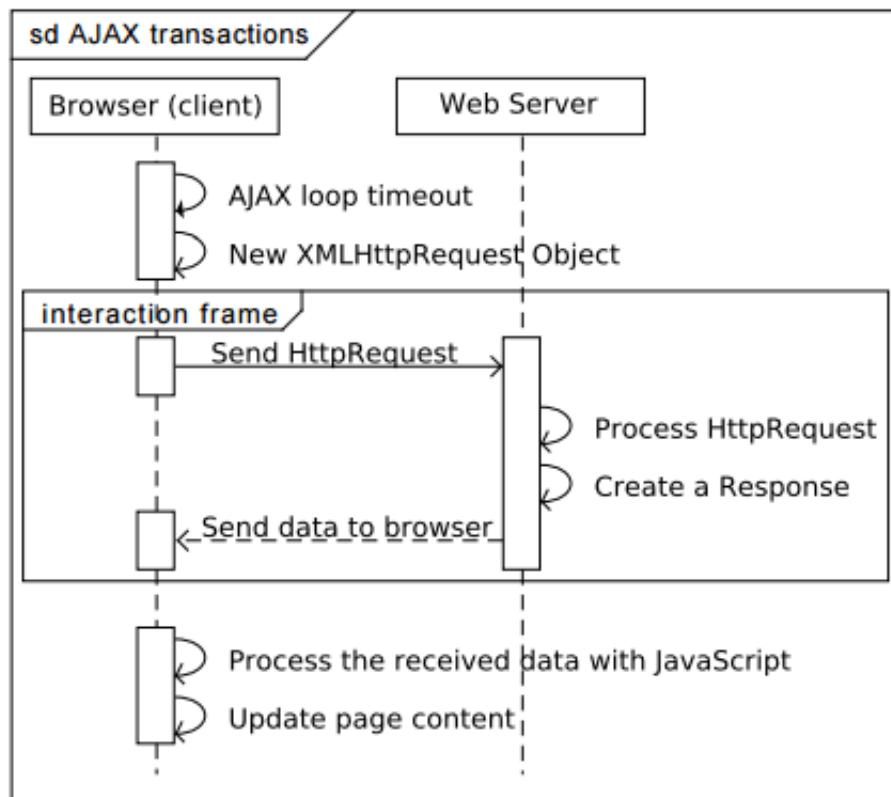


FIGURA 2.16: Diagrama de secuencia AJAX.

La configuración establecida por default es la que se muestra en la siguiente tabla:2.1.

⁶<http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Trabajo-Final-CESE-Patricio-Bos-2016.pdf>, figura:3.9.

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 2.1: Configuracion por default del servidor

Capítulo 3

Ensayos y Resultados

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

3.1. Banco de pruebas

Para poder simular el comportamiento de un sensor de temperatura y la descarga de una batería. Se implemento una placa básica la cual ya fue vista en el capítulo 3.2, que para facilidad del lector volvemos a mostrarla en la siguiente figura 3.1, la cual 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 del módem GPRS SIM800L.



FIGURA 3.1: Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.

Todo esto se integra la Plataforma de la CIAA-NXP, véase figura 3.2, para verificar el correcto funcionamiento del firmware desarrollado.

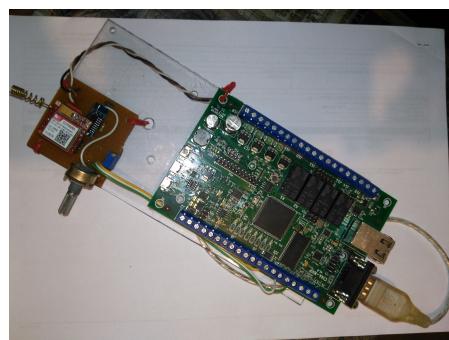


FIGURA 3.2: Placa básica integrada con la CIAA-NXP.

3.2. Configuraciones para la PC

Para la realización de los ensayos, comenzamos configurando la PC en forma adecuada. Para ello en la siguiente figura 3.3 vamos a ver como configurar en Linux la interfaz de red. Es importante destacar que dichos comandos son ejecutados con el permiso se súper usuario, *root*.

```
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:a4:b5:30
        inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::a4ba: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 3.3: Configuración de una red estática para la interfaz Ethernet.

Una vez que tenemos la configuración de red adecuada vamos a proceder a dirigirnos a un navegador web. Y como vemos en la imagen 3.4 que fue vista en el capítulo anterior y que para facilitar al lector se ingreso nuevamente, donde el numero de ip correspondiente al dispositivo de control es: 192.168.1.11

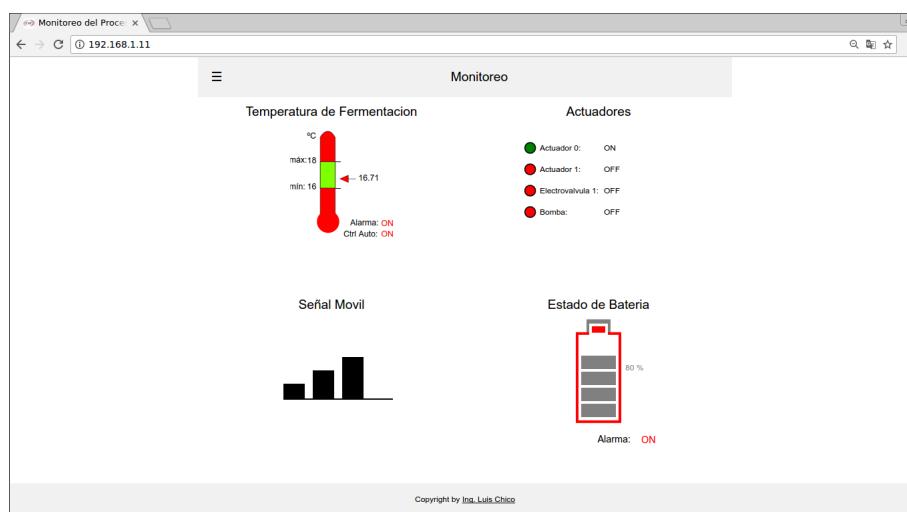


FIGURA 3.4: Pantalla principal de la web de monitoreo.

En esta pantalla vamos a tener toda la información correspondiente al sistema. Arriba a la izquierda podemos ver 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 podemos ver 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.

Abajo a la izquierda tenemos en 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 permitirnos el servicio de alertas mediante mensajes SMS.

Finalmente abajo a la derecha tenemos 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.

3.3. Verificación del sistema web embebido

De lo explicado en la sección anterior, se procederá a navegar por el menú, véase figura 3.5, las diferentes configuraciones para constatar de que todo el sistema esta funcionando en forma correcta. Al presionar donde dice menú, obtendremos las opciones de configuración. Donde:



FIGURA 3.5: Opciones del al presionar sobre el menú desplegable.

- 1. Monitoreo:** Nos dirige a la pantalla principal, donde podemos ver el estado del sistema. figura 3.4.
- 2. Actuadores:** Podemos setear en forma manual el estado de los mismos. figura 3.6.
- 3. Batería:** Aquí podrá setear la alarma debido al nivel de descarga de batería que configuremos. Esta enviara un SMS indicando el estado. figura 3.7.

- 4. Módem GPRS:** En este menú podremos configurar 2 personas a quienes serán enviadas las alertas debido al accionar de alguna alarma.3.8.
- 5. Temperatura:** Aquí se configura el rango de temperatura, la alarma correspondiente y si deseamos activar 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 3.9.
- 6. Network:** Aquí se configura los parametros de la red, IP, mascara de red y puerta de enlace. figura 3.10.
- 7. Cerrar Menú:** Esta opción cerrar el menú, permitiendo quedarse en la sección que estaba antes de presionar en menú.



FIGURA 3.6: Activar/Desactivar actuadores.



FIGURA 3.7: Configurar nivel de descarga para accionar la alerta por batería baja.

FIGURA 3.8: Configuración de los celulares de quienes van a recibir las alarmas por SMS.



FIGURA 3.9: Configuración del rango de temperatura y activación del estado de control automático y alarma.

Este formulario web para configurar la red. Se piden los datos IP, Mask y Gateway en formato de cuadruplo-octeto (IP: 192.168.0.11). Hay un botón 'Submit' y una nota que dice 'Copyright by Ing. Luis Chico'.

FIGURA 3.10: Formulario de configuración de IP, mascara y puerta de enlace.

3.4. 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 monitorizando. En la web no muestra figura 3.11. y dado su respectiva alerta mediante un SMS como vemos en la figura 3.12

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 web se verá como en la figura 3.13. y dado su respectiva alerta mediante un SMS como vemos en la figura 3.12

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 3.12.

Para verificar los actuadores nos dirigimos a la web en la sección de configuración de actuadores y seteamos para cada uno de ellos los diferentes estados, encendiendo y apagando. Luego se probó en bloques de varios actuadores. En la siguiente figura 3.14 se ve un ejemplo de los casos probados.

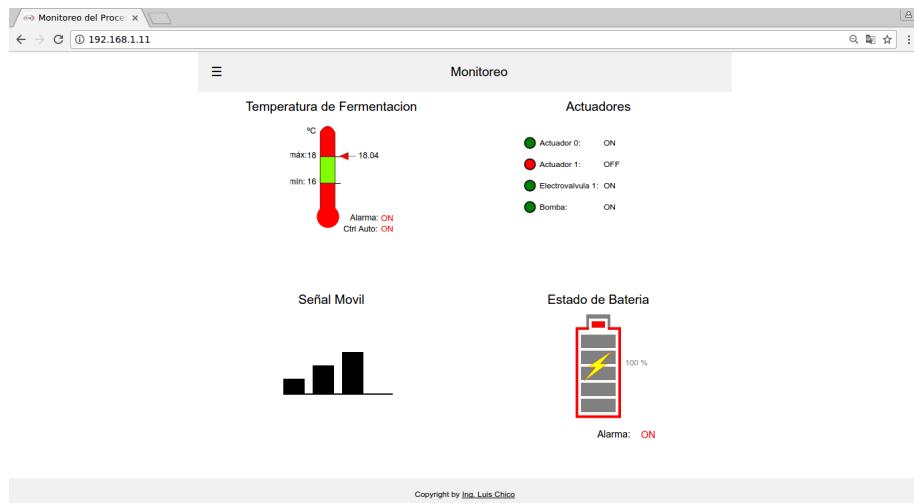


FIGURA 3.11: Activación por control automático debido a temperatura elevada.

De esta forma podemos concluir que el funcionamiento del sistema es el correcto.

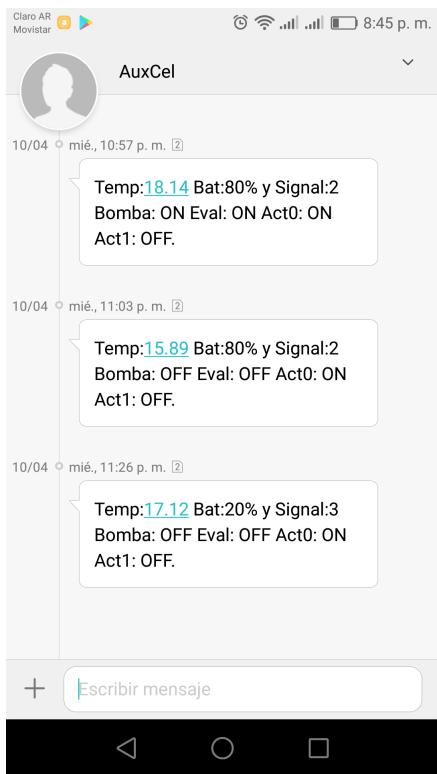


FIGURA 3.12: SMS recibidos debido a temperatura máxima.
El segundo SMS recibido es por temperatura mínima.
Y el último recibido debido a batería baja.

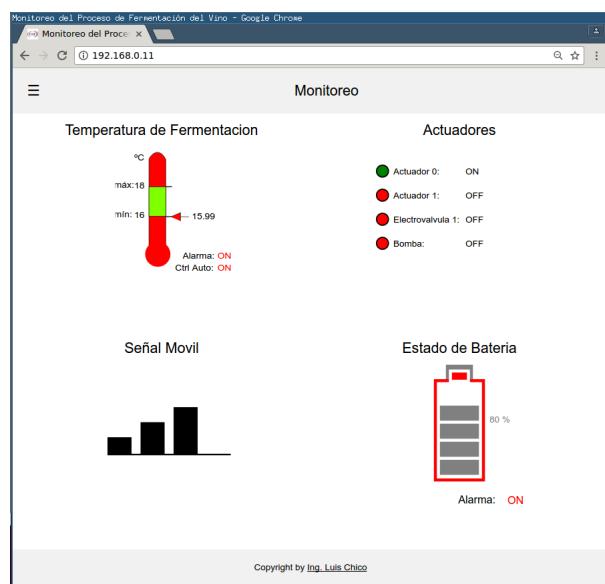


FIGURA 3.13: Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula.



FIGURA 3.14: Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.

Capítulo 4

Conclusiones

4.1. Conclusiones generales

Al finalizar esta primera etapa del prototipo, podemos concluir que se logró:

- Mediante una página web accedemos al estado del sistema.
- Permite navegación mediante pc, celulares, tablets compatibles con navegadores como Chrome y Firefox.
- Haciendo uso de un sensor de temperatura y salidas para actuadores. Permite un control manual y automático del sistema.
- Alertar mediante SMS cuando: la temperatura excede el rango permitido, y/o para un determinado nivel de batería.
- Implementar mediante el stack TCP/IPlwIP con el RTOS freeRTOS.
- Uso del lenguaje C, HTML, CSS y JavaScript.
- Hacer uso de un sistema de control de versiones, GitHub.

Es decir, ya podemos controlar la temperatura de un tanque y recibir alertas en los principales estados críticos del proceso. Tenemos incorporado una interfaz de monitoreo que a su vez nos permite actuar en forma remota.

4.2. Próximos pasos

En esta primera etapa se dió una primera interacción con todo lo que involucra controlar el proceso de fermentación. Ahora estamos en condiciones de perfeccionar este producto extendiendo sus funciones. Para ello podríamos optimizar los siguientes puntos:

- Ampliar la capacidad de control a 4 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.